Proceedings of

ATACCS'2013

3rd International Conference on Application and Theory of Automation in Command and Control Systems

Naples, Italy
28 - 30 May 2013
University of Naples Federico II

Editors
Guillaume Brat, Eduardo Garcia, Antonio Moccia, Philippe Palanque, Alberto Pasquini, Francisco Javier Saez & Marco Winckler
FOREWORD

Technological and operational developments in Air Traffic Management (ATM) are more based on industrial and national interests than on a consensus approaches. This contributes to the lack of cohesion and interoperability of the ATM system and tends to worsen the problem of airspace fragmentation. Current Air traffic management inefficiency, which is one of the main challenges of SESAR (Single European Sky ATM Research), could be drastically reduced by increasing levels of automation and having a common assessment framework for all roles, interactions and functions of ATM agents.

Research and development in automation and command and control systems can pave the way for new tools and technologies. These, coupled with new operational procedures and legislation, will enable a paradigm shift towards a global transport traffic management system capable of supporting highly efficient and environmentally friendly air transport operations. As a good example, two large regions of the world are currently embarked in major ATM modernization initiatives, SESAR (Single European Sky ATM Research) in Europe and NexGen (Next Generation Air Transportation System) in the US. These initiatives are focused on exploring how increased levels of automation enabled by new technology can lead to a sustainable and more efficient ATM system.

Information sharing and systems interoperability will be key enablers to improve transport processes. However, it is not clear how the sheer amount of information that will become available to stakeholders can be turned into useful knowledge to improve operations. It is not even clear what kind of knowledge and actors are required. The automation of processes using new technology will involve answering many important questions, such as: how can uncertainty be managed in automated systems? in which command and control scenarios (centralized or autonomous) will automation provide a higher overall system performance? do high traffic density and complexity limit the potential benefits of higher levels of autonomy? are the current frameworks for automation, cognition and human factors adequate to explain the singularities of ATM? is a highly automated air transport system socially/psychologically acceptable?

The ATACCS (Application and Theory of Automation in Command and Control Systems) series of conferences is organized by the SESAR’s HALA! Research Network with the aim of fostering research in the domain of automation. The ultimate objective of ATACCS is to provide an open forum to present and discuss relevant research contributing to answer some of the questions above.

These proceedings provide an overview of the 3rd ATACCS Conference, held at Università degli Studi di Napoli Federico II, Naples (Italy), on 28th-30st of May 2013. The contributions to the conference gathered in these proceedings contain the most up-to-date research in the automation domain, focusing on ATM applications. The topics of the research include Human-Automation Collaboration, Strategic Trajectory Planning, Trajectory Optimization, Unmanned Aircraft Systems & Collision Avoidance, Autonomy and Automation and Decision Support and Tools.

Many individuals have contributed to the success of this international conference. Our sincere appreciation goes to all authors including those whose papers were not included in the program. Many thanks to our distinguished General Conference Chairs, Professors Antonio Moccia and Francisco Javier Saez. Special thanks to the Research papers chairs, Professors Philippe Palanque and Guillaume Brat. Thanks also are due to the Doctoral consortium chairs, Professors Chris Johnson and Hartmut Fricke. Thanks to the Local conference chair, Professor Domenico Accardo and to the Posters and Demonstrations Chair, Dr. Thomas Feuerle. Special thanks go to the Publicity and Publication Chair, Dr. Marco Winckler for his good and timely work. Finally, we also want to give a special thanks to the Program Committee, whose effort and hard work reviewing the papers are one of the greatest assets of the ATACCS conference.

Alberto Pasquini & Eduardo García
ATACCS’2013 Technical Programme Chairs
The HALA! Network

The conference ATACCS is supported by the HALA! Network which is a Research Network established within the framework of SESAR WP-E to spearhead long term and innovative research in Air Traffic Management (ATM) in pursuit of the SESAR 2020 vision and beyond. HALA! is the acronym for “Higher Automation Level in ATM” which is also the focus of the network. It started its activities on September 3rd 2010 and is currently formed by over 450 researchers. The HALA! network is managed by the following organizations:

- Universidad Politécnica of Madrid (UPM), Spain (coordinator)
- Imperial College London, United Kingdom
- KTH Royal institute of Technology, Sweden
- TU Braunschweig, Germany
- TU Dresden, Germany
- Institute of Research in Informatics (IRIT)/University of Toulouse, France
- ATM R&D+innovation Reference Centre (CRIDA), Spain
- National Aerospace Laboratory (NLR), The Netherlands
- Boeing, R&TE, Spain
- Deep Blue, Italy
- EADS Innovation Works
- GMV SKY, Portugal
- Pildo Labs, Spain
- Università degli Studi di Napoli “Federico II”, Italy
- Delft University of Technology, Netherlands
- TU Darmstadt, Germany
- University of Roma 3, Italy
- Technical University of Catalonia, Spain
- University Kassel, Germany
- Istanbul Technical University (ITU), Turkey

In order to reap the expected benefits and to ensure that there are no delays in the implementation of SESAR (Single European Sky ATM Research), research resources need to be applied to the exploration of novel, unconventional and high risk areas, involving new technologies, concepts or ideas. However, the need to deliver an on-time and on-budget system constrains the possibility of the SESAR projects to invest in high-risk, novel or unconventional areas. HALA! will provide a flexible environment in which ideas on Automation in Air Traffic Management (ATM) will flow using a common approach and removing most of the constraints on the research carried out in SESAR.

The organizations participating in HALA! cover all the knowledge areas related to automation and their experts collaborate actively with SESAR. As Innovation is a main objective of HALA!, Doctoral (PhD) level research involving leading universities in collaboration with each other and the ATM industry, are the heart of HALA!. In addition to the traditional university supervision, PhD researchers are guided by organizations that represent the ATM industry so that the developed knowledge can be optimally translated into effective tools. Finally, HALA! participants have a long proven experience on collaborating in complex projects and provide HALA! with the necessary skills and experience to achieve the challenge of automation.

We invited people interested in joining HALA! to visit our web site http://www.hala-sesar.net/ and to participate of the upcoming ATACCS conferences.

HALA! means in Spanish “Go on! Get moving!”

Francisco Javier Saez
HALA! Management Committee
ATTACS’2013 Technical Committee

GENERAL CONFERENCE CHAIRS
Antonio Moccia, University of Naples 'Federico II', Italy
Francisco Javier Saez, UPM, Spain

TECHNICAL PROGRAMME CO-CHAIRS
Alberto Pasquini, Deep Blue, Italy
Eduardo Garcia, UMP/CRIDA, Spain

RESEARCH PAPERS CHAIRS
Philippe Palanque, ICS-IRIT, University Paul Sabatier, France
Guillaume Brat, NASA, United States

POSTERS AND DEMONSTRATIONS CHAIR
Thomas Feuerle, TU Braunschweig, Germany

DOCTORAL CONSORTIUM CHAIR
Hartmut Fricke, TU Dresden, Germany
Chris Johnson, University of Glasgow, Scotland

PROGRAM COMMITTEE
Domenico Accardo, University of Naples "Federico II", Italy
Paola Amaldi, School of Engineering and Information Sciences Middlesex University, UK
Yamine Ait Ammeur, ENSEEIHT, France
Junia Anacleto, Federal University of São Carlos, Brazil
Rosa Maria Arnaldo, UPM, Spain
Fredrik Barchéus, KTH Stockholm, Sweeden
Valentina Beato, DeepBlue, Italy
Ellen J. Bass, University of Virginia, USA
Matthew Bolton, San Jose State University, USA
Sue Bogner, Institute for the Study of Human Error, LLC
Marc Bourgois, EUROCONTROL
Guy Boy, Florida Institute of Technology, USA
José Ramón Casar, ETSIT, Spain
Jan Davies, University of Calgary, Canada
Frederic Dehais, ONERA, France
Michael Cooke, Aerospace Psychology Research Group, Trinity College Dublin, Ireland
Asaf Degani, General Motor, Israel
Michael Feary, NASA Ames, USA
Thomas Feuerle, Braunschweig Technical University, Germany
Dimitra Giannakopoulou, NASA, USA
Michele Grassi, University of Naples "Federico II", Italy
Peter Hecker, Braunschweig Technical University, Germany
Erik Hollnagel, Ecole des Mines, France
Gokhan Inalhan, Istanbul Technical University, Turkey
Shelly Jeffcott, Centre of Research Excellence in Patient Safety, Australia
Chris Johnson, University of Glasgow, Scotland
Kota Kageyama, ENRI, Japan
Barry Kirwan, EUROCONTROL
Thomas Kunze, TU Dresden, Germany
Peter Bernard Ladkin, Bielefeld University, Germany
Javier López Leonès, Boeing R&TE, Spain
Max Mulder, Delft University of Technology, The Netherlands
José Neves, GMV SKY, Portugal
Bernd Oreschko, TU Dresden, Germany
Miguel Angel Piera, Universidad Autonoma de Barcelona
Amy Pritchett, School of Aerospace Engineering, USA
Marcel Quintana, Consultant
Marta Quintães, GMV SKY, Portugal
Catherine Tessier, ONERA, France
Christel Seguin, ONERA, France
Dirk Schaefer, EUROCONTROL
Nicolás Suarez, CRIDA, Spain
Pernilla Ulfvengren, KTH Stockholm, Sweden
Pim van Leeuwen, NLR, Netherlands
Maria de Fatima Queiroz Vieira, Universidade Federal de Campina Grande, Brazil
Miguel Vilaplana, Boeing R&TE, Spain
Amaya Atencia Yepez, GMV, Spain
Washington Yotto Ochieng, Imperial College, UK
Helle Oltedal, Nautical Institute gjenoppstar i Haugesund, Norway
Organizing committee

LOCAL ORGANIZATION CHAIR
Domenico Accardo, University of Naples 'Federico II', Italy

PUBLICITY AND PUBLICATION CHAIR
Marco Winckler, ICS-IRIT, University Paul Sabatier, France

Acknowledgments

The conference chairs and conference organizers would like to thank our sponsors:

Industry and Organization Sponsors
- Hala Research Network
- SESAR
- EUROCONTROL
- The Boeing Company
- Selex
- Sicta

Scientific Sponsors
- IFIP W.G. 13.5 on Human Error, Safety and System Development
- ACM SIGCHI Special Interest Group on Computer-Human Interaction
- University of Naples "Federico II"
- interact.design.org
# Table of Contents

## RESEARCH PAPERS

1. **4D Trajectory Planning in ATM with an Anytime Stochastic Approach**  
   Jose Antonio Cobano, David Alejo, Guillermo Heredia and Aníbal Ollero

2. **Assessment of Impact of Degree of Automation on Human Roles: The Experts’ Analysis using Gaming**  

   Damiano Taurino, Stefano Bonelli and Marco Ducci

4. **Autonomous Management of a UAV Airfield**  
   Rosa M. Rodríguez, Francisco Alarcón, Daniel Santamaria Rubio and Aníbal Ollero

5. **Aviation Safety: Modeling and Analyzing Complex Interactions between Humans and Automated Systems**  
   Neha Rungta, Guillaume Brat, William J. Clancey, Charlotte Linde, Franco Raimondi, Chin Seah and Michael Shafto

6. **Development of Numerical Sensor Models for Cooperative and Non-Cooperative Collision Avoidance**  
   Anna Elena Tirri, Giancarmine Fasano, Domenico Accardo, and Antonio Moccia and Vincenzo Baraniello

7. **Effect of Flight Plans Predictability and Accuracy on Traffic Demand Forecast**  
   Esther Calvo Fernández and José Manuel Cordero

8. **Flying by the Seat of their Pants: What can High Frequency Trading learn from aviation?**  
   Gordon Baxter and John Cartlidge

9. **Model-Based Dynamic Distribution of User Interfaces of Critical Interactive Systems**  
   David Navarre, Célia Martinie, Philippe Palanque, Alberto Pasquini and Martina Ragosta

10. **Modeling Aircraft Pushback Trajectories for Safe Operations**  
    Franziska Dieke-Meier, Thomas Kalms, Hartmut Fricke and Michael Schultz

11. **Multiobjective Tactical Planning under Uncertainty for Air Traffic Flow and Capacity Management**  
    Gaetan Marceau-Caron, Pierre Savéant and Marc Schoenauer

12. **Multiphase Mixed-Integer Optimal Control applied to 4D Trajectory Planning in Air Traffic Management**  
    Alberto Olivares, Manuel Soler and Ernesto Staffetti

13. **The Legal Case**  
    Giovanni Sartor, Giuseppe Contissa, Hanna Schebesta, Migle Laukyte, Paola Lanzi, Patrizia Marti and Paola Tomasello

14. **Transforming Time - Towards an intuitive time constraint depiction**  
    Jendrick Westphal, Nima Barraci, Uwe Klingauf and Jens Schiefele

## POSTERS

129 **A Decomposition-based Optimal Control Approach for Aircraft Conflict Avoidance Performed by Velocity Regulation**  
   Loïc Cellier, Sonia Cafieri and Frédéric Messine

132 **Aircraft Trajectory Simulator Using a Three Degrees of Freedom Aircraft Point Mass Model**  
   Javier García-Heras Carretero, Francisco Javier Sáez Nieto and Ricardo Román Cordón

136 **An OpenFlow-based Architecture for IaaS Security**  
   Antonio Marotta, Gabriella Carrozza, Stefano Avallone and Vittorio Manetti

140 **Control and Stabilization Applied to Micro Quadrotor AR.Drone**  
   Kamel Boudjat and Chevif
| 146 | Automated Task Allocation | Rocío Barragán Montes, Eduardo García González and Francisco Javier Sáez Nieto |
| 150 | Designing Alarm Device for Socio-Technical Systems: Co-active or Coercive Design? | Paola Amaldi and Sara Khalil |
| 154 | Towards an Integrity Monitoring for 4D-trajectories | Alexander Schwithal and Peter Hecker |

**DOCTORAL CONSORTIUM**

| 170 | A Multi-ModelS Based Approach for the Analysis and Design of Usable and Resilient Partly Autonomous Interactive Systems | Martina Ragosta |
| 174 | An Approach for Aircraft Turnaround Decision Making to Minimize Outbound Delay | Thomas Kunze, Michael Schultz and Hartmut Fricke |
| 179 | A Property-Graph-Based Knowledge Representation for Decision Support Systems | Lutz Bretschneider and Peter Hecker |
| 184 | AUTOFLY-Aid: Multi-Modal Trajectory Projection Approach for Airborne Collision Detection and Avoidance | Emre Koyuncu, Eduardo García and Gokhan Inalhan |
| 158 | Information Analysis for a Future Flight Deck Design in the Context of 4D Trajectory Based Operation | Sven Hartmann |
| 162 | Traffic-Separation & Collision-Avoidance Strategy to Promote Civil Applications of UAVs | Giovanni Migliaccio, Giovanni Mengali and Roberto Galatolo |
| 166 | User-Interface Design for Highly Automated Systems - A structured approach | Giuseppe Frau |
| 189 | Effects in Fuel Consumption of Assigning RTAs into 4D Trajectory Optimisation upon Departures | Santi Vilardaga and Xavier Prats |
| 193 | Distance at Closest Point of Approach for Airborne Collision Avoidance | Anna Elena Tirri, Giancarmine Fasano, Domenico Accardo, Antonio Moccia |
| 197 | Modeling External Disturbances for Aircraft in Flight to Build Reliable 4D Trajectories | Tobias Pabst, Thomas Kunze, Michael Schultz and Hartmut Fricke |
| 202 | Sensitivity of Continuous Climb Departure Predictions to Aircraft Intent Uncertainties | Enrique Casado, Miguel Vilaplana and Colin Goodchild |
4D Trajectory Planning in ATM with an Anytime Stochastic Approach

José Antonio Cobano  
Robotics, Vision and Control Group (University of Seville)  
Camino de los Descubrimientos, s/n 41092 Seville (Spain)  
jacobano@cartuja.us.es

David Alejo  
Robotics, Vision and Control Group (University of Seville)  
Camino de los Descubrimientos, s/n 41092 Seville (Spain)  
dalejo@cartuja.us.es

Guillermo Heredia & Aníbal Ollero  
Robotics, Vision and Control Group (University of Seville)  
Camino de los Descubrimientos, s/n 41092 Seville (Spain)  
{guiller,aollero}@cartuja.us.es

ABSTRACT
This paper presents the Anytime Stochastic Conflict Detection and Resolution system (ASCDR), which automatically identifies conflicts between multiple aircraft and proposes the most effective solution 4D trajectory considering the available computation time. The system detects conflicts using an algorithm based on axis-aligned minimum bounding box and resolves them cooperatively using a collision-free 4D trajectory planning algorithm based on a roundabout fast initial solution and a stochastic optimization technique named Particle Swarm Optimization (PSO) to modify the 4D initial trajectories of the aircraft with an overall minimum cost. Moreover, an anytime approach using PSO is applied because determining optimal trajectories with short time intervals in the flight phase is not feasible. Thus, trajectories whose quality improves when available computation time increases are yielded. The method could be applied to Medium-Term and even Short-Term Conflict Detection and Resolution depending on the look-ahead times. The method has been validated with simulations in scenarios with multiple aerial vehicles in a common airspace.

Author Keywords  
Conflict detection and resolution; air traffic management; cooperative aerial vehicles; stochastic optimization

ACM Classification Keywords  
J.2 [Physical Sciences and Engineering]: Aerospace.

General Terms  
Algorithms; Management; Reliability.

INTRODUCTION
The workload of the air traffic controllers is increasing and, despite the upgrades in the onboard systems, humans still constitute the core of the future ATM system. The automation technologies will play an important role in this future ATM to satisfy high air traffic demand and hence decreasing controller workload. Thus, the implementation of an efficient, safe, and reliable Air Traffic Management (ATM) system [1] is one of the main goals of SESAR to address this increasing traffic volume in the next years. Therefore, higher levels of automation in ATM should be explored. These automation technologies should be supported by new decision support systems based on algorithms to allow the controller to timely react to dangerous situations by facilitating collision-free trajectories [2]. Therefore, 4D trajectory planning algorithms are required to ensure the safety of the system.

A review of these methods on path planning with a comprehensive mathematical discussion is presented in [3]. Among these methods could be highlighted: potential fields [4], graph search like A* and D* [5] and Rapidly-exploring Random Trees (RRT) [6]. Other kind of methods have been proposed such as evolutionary techniques [7][8][9], particle swarm optimization [10] and multi-objective evolutionary algorithms [11].

The problem of trajectory planning is NP-hard [12][13]. Sampling-based techniques, as opposed to combinatorial planning, are usually preferred in these NP-hard problems. These planning schemes match particularly well when the solution space is hard to model or unknown a priori because of its dynamic nature.

Furthermore, planning optimal collision-free trajectories for multiple aircraft leads to optimization problems with multiple local minima in most cases and, thus, local optimization methods as gradient-based techniques are not well suited to solve it. The application of evolutionary techniques or particle swarm optimization is an efficient and effective alternative for this problem, since they are global optimization methods.

The methods on Conflict Detection and Resolution (CDR) are characterized depending on the following factors: dimensions of the state information, technique for dynamic state propagation, conflict detection threshold, conflict resolution technique, maneuvering dimensions, and
management of multiple aerial vehicles conflicts. These methods have also been studied extensively and different types of techniques have been proposed [14]: based on the speed assignment [15] or mixed-integer linear program (MILP) [16] [17] [18], [19] analyses the influence of speed uncertainty on the efficiency of mid-term speed adjustments. Other method resolves pairwise conflicts [20] but do not consider more aircrafts. The method for multiple-aircraft conflict avoidance proposed in [21] assumes that aircrafts cruise at constant altitude with varying velocities and that conflicts are resolved in the horizontal plane using heading change, velocity change, or a combination thereof. Methods based on Ant Colony Optimization algorithm have also been proposed [22]. In [23], the application of a game theory approach to airborne conflict resolution is presented. These techniques present a disadvantage: they are not well suited for applications that require a high level of scalability for the application to many aircrafts.

This paper presents the Anytime Stochastic Conflict Detection and Resolution system (ASCDR). First, the system automatically identifies conflicts between multiple aircraft and then proposes the most effective solution 4D trajectory considering the look-ahead time, that is, the available computation time. The detection algorithm is based on axis-aligned minimum bounding box. The cooperative collision-free 4D trajectory planning algorithm is based on a roundabout technique and a stochastic optimization technique named Particle Swarm Optimization (PSO). The roundabout technique is considered to quickly compute an initial but non-optimal solution. The choice of PSO is based on its low computational overheads and faster solution convergence compared to genetic algorithms and other evolutionary algorithms [24]. Each aircraft changes its 4D trajectory to solve the detected conflicts collaborating with the rest of aircraft. The maneuvers allowed are change of heading and speed to meet the Estimated Time Arrival (ETA). The ASCDR system presents two main advantages: its low execution time and its scalability.

The computing time is an important issue in flight execution phase. The ASCDR system considers an anytime approach to resolve the conflicts. Computing time of most optimization methods is not deterministic, and most published works either do not consider computing time or simply show that in average it is much less than the available time to get a solution, thus wasting available computing time. The proposed system adopts an anytime approach, it is able to provide a flyable solution at any time, and it will be more or less close to the optimum depending on the available time.

Thus, the ASCDR system always yields valid 4D trajectories whose quality improves when available computation time increases. Therefore, the ASCDR system can be used with low look-ahead times although the quality of the solution will not be optimal.

The paper is organized into five sections. The formulation of the problem is presented in Section II. The ASCDR system is described in Section III. Section IV presents the simulations performed. Finally the conclusions are detailed in Section V.

PROBLEM FORMULATION
The problem considered in this paper is the collision-free 4D trajectory planning of airspace users in a dense ATM scenario. A 4D trajectory is defined by a sequence of waypoints and its corresponding ETA. The aircraft share a common airspace and the separation between aircraft should be greater than a given safety distance. The maneuvers allowed to solve a collision are the addition of intermediate waypoints and the changes of speed. After a collision is detected, the problem is solved when a collision-free 4D trajectory for each aircraft is computed.

All aircraft cooperate to solve the problem changing their initial trajectory. The information that the system needs in order to solve the problem is the following:

- Initial 4D trajectory of each aircraft
- Model of each aircraft
- Initial location and goal location of each aircraft
- Look-ahead time to know the available computation time

The objective is to find collision-free 4D trajectories that minimize the probability of having a collision while minimizing the changes of waypoints and speed for each aircraft to meet the ETA.

ASCDR SYSTEM
This section describes the blocks of the proposed system to resolve the detected conflicts (see Figure 1). First, a detection algorithm based on axis-aligned minimum bounding box is implemented to detect collision. Then, a 4D trajectory planning algorithm based on roundabout techniques and PSO is implemented.

![Figure 1. Blocks of the ASCDR system.](image)

Axis-aligned minimum bounding box for detection
The detection algorithm is based on axis-aligned minimum bounding box. This technique presents as advantages the low time of execution and the need of few parameters to describe the system. On the other hand, it presents two disadvantages: it is not very accurate and it depends on the coordinate axes.
Each aircraft is represented with two boxes, horizontal and vertical box, joined in order to detect the conflicts (see Figure 2). Each box is defined by the intersection of three intervals, one by axis. The measurement of the horizontal box is related to the minimum horizontal separation between aircraft and the vertical box is related to the vertical separation. Thus, the minimum separation, $S_{\text{min}}$, between two aircrafts is defined by the dimension of both joined boxes (see Figure 2). A collision is detected when there is an overlapping between the intervals that define each box. Thus, the 3D problem is reduced to three problems of overlapping, one in each coordinate. Let us consider the intervals in one coordinate $A=[A_x,A_y]$ and $B=[B_x,B_y]$. The condition of overlapping for this coordinate is given by:

$$A_x > B_x \land A_y < B_y$$  \hspace{1cm} (1)

Figure 2. Detection algorithm based on axis-aligned minimum bounding box: A and B overlap (collision).

4D Trajectory planning algorithm

The collision-free 4D trajectory planning algorithm is based on a roundabout technique and the PSO method.

The goal of this algorithm is to obtain collision-free 4D trajectories by adding one intermediate waypoint in the aircraft trajectory. The speed of the aircraft is also changed to meet the ETA at the goal waypoint. The following cost function is proposed in order to obtain desirable solutions:

$$J = \sum_{i=1}^{N} (L_i + D_i) + \omega_c$$  \hspace{1cm} (2)

where $N$ is the number of aerial vehicles, $L_i$ is the total length of trajectory $i^{th}$, $D_i$ is related to the changes of velocity in order to meet the ETA of each aerial vehicle, and $\omega_c$ is the collision penalty that will be added if the new trajectories still lead to collisions in the system. These trajectories should be discarded.

This optimization problem is solved by using PSO. PSO is a heuristic global optimization algorithm and was first proposed in [27]. It is developed from swarm intelligence and is based on the research of bird and fish flock movement behavior. It works by maintaining a swarm of particles that move around in the search-space influenced by both the improvements discovered by the other particles (social behavior) and the improvements made by the particle so far (greedy behavior). Its main advantages are its simplicity, easy implementation and the existence of few parameters to tune when comparing with other evolutionary algorithms.

The algorithm implemented is based on [28]. Let $S$ be the number of particles in the swarm, each particle is defined by a state vector $x_i$ in the search space and a velocity vector $v_i$. The state vector is given by a position and speed. Let $p_i$ be the best state vector of particle $i$ and let $g$ be the best known state vector of the entire swarm. In first place, the swarm is initialized by assigning random initial locations and velocities. A uniform distribution in the search space has been chosen for this step.

Then the exploration loop is executed. In each iteration, both the state vector and the velocity of each particle is updated by applying the formula indicated in steps 10 and 11 (see Algorithm 1).

$$x_i \leftarrow x_i + v_i$$  \hspace{1cm} (3)

$$v_i \leftarrow \omega v_i + \phi_p r_p (p_i - x_i) + \phi_g r_g ((g - x_i))$$  \hspace{1cm} (4)

The most important parameters in this formula are the social weight, $\phi_g$, and the local weight, $\phi_p$. $\omega$ is the inertia weight. $r_g$ and $r_p$ are vectors where each component is generated at random with a $U(0,1)$ distribution. Local and global best positions are also updated if necessary (steps 13-15).

The exploration loop can be finished by using many different termination criteria. Among these criteria a timeout condition and a convergence condition (most of the individuals lay in to a tight region of the search space) are the common approaches. In this paper, the algorithm concludes when the available computation time is reached.

The parameters $\phi_g$ and $\phi_p$ have been tuned by performing several tests with the same conditions and only changing one parameter at a time. These parameters are usually selected in the interval [0,1]. In our case, the best values found were $\phi_g=0.9$ and $\phi_p=0.1$.

The main problem of these evolutionary algorithms, is that the results vary deeply as a function of the initial population that is randomly obtained. Moreover, a local minimum is present when all of generated population leads to conflict.

For these two reasons, a method with a predictable execution time and that always leads to a feasible solution is necessary. This solution will be added to the initial population of the PSO algorithm.
We propose the use of the Generalized Roundabout Policy to solve this. This technique was introduced in [25] and provides quick and effective solutions for large-scale problems. The main idea is to make the involved aircrafts circle the conflict in the same direction. The radius of the circle should be greater to ensure collision-free trajectories and to make flyable trajectories. We refer to the reader to [26] for a more in-depth discussion of the Generalized Roundabout Policy.

**Algorithm 1 Basic PSO algorithm**

1. for Each particle do
   2. Initialize each particle’s state vector $x_i$ with the desired probability function
   3. Initialize particle best state vector $p_i \leftarrow x_i$
   4. If $f(p_i) < f(g)$ update the swarm best state vector $g \leftarrow x_i$
   5. Initialize each particle’s velocity vector $v_i$
      An uniform distribution is usually used.
   6. end for
7. repeat
8.   for Each particle do
9.      Pick random numbers $r_g, r_p$ with $U(0, 1)$
10.     Update the particle’s velocity:
       $v_i \leftarrow \omega v_i + \phi_p r_p (p_i - x_i) + \phi_g r_g (g - x_i)$
11.     Update the particle’s state vector:
       $x_i \leftarrow x_i + v_i$
12.    if $f(x_i) < f(p_i)$ then
13.        Update the particle’s best known state vector
14.     if $f(x_i) < f(g)$ then
15.        Update the swarm’s best known state vector $g \leftarrow x_i$
16.    end if
17. end if
18. end for
19. until A termination criterion is met

**SIMULATIONS**

In order to evaluate and validate the ASCDR system a comprehensive set of simulations generated randomly should be carried out.

**Test set design**

The definition of a metric plays on important role to evaluate the results. In cases of difficult path or motion planning problems for only one mobile object, there are some de facto benchmark standards in the academic context, like the bug trap or the alpha test [31]. However, this is not the case when dealing with planning of multiple mobile objects.

A test set has been developed in a given scenario to validate the ASCDR system and to measure the properties of the system regarding time of execution, optimization and level of scalability with number of vehicles. Furthermore, the test set and the design methodology could be useful for comparison with other methods developed.

The considered scenario has a base of 10x10 dimensional units. Different problems are defined considering the same scenario, as well as the same random problem generation process. Each problem is formulated as a set of entry and exit points located in one of the lateral sides of the scenario.

The adopted strategy is regressive. Random candidate solutions are generated and the problem is defined using them when they are found.

The random generation process of the tests is performed following the Algorithm 2. For each vehicle, an entry side is randomly chosen, selecting a uniformly random number between 1 and 4 (line 4). Then, the exit side is randomly selected from the resting 3 sides (line 5). Entry and exit points are randomly selected from the corresponding side (line 6). A certain number, $M$, of intermediate waypoints inside of the scenario along with the entry and exit points define the flight plan.

The algorithm should ensure the following (see line 8):

- The solution is valid, i.e. vehicles do not collide.
- The initial plans generate a contrast, i.e. the vehicles initial plans lead to collision.

The test set consists of 40,000 different problems grouped by the number of vehicles involved, from 2 to 5, in subsets of 10,000 tests. This classification, using the number of vehicles, is useful to study the scalability characteristics of the method.

**Algorithm 2 Random test generation algorithm**

1: for each test do
2:   while test is not valid do
3:     for each aircraft do
4:        Choose a random entry side
5:     choose a random exit side from the resting 3
6:     choose entry and exit points from the corresponding entry and exit sides
7:     Add $M$ random intermediate waypoints
8:     Check for the flight plan validity
9:   end for
10: end while
11: end for

**Simulations results**

Many simulations have been carried out from the test set in order to check the properties of the ASCDR system. The tests have been performed in the same computer and under the same conditions.
The algorithms have been run in a PC with a 2GHz Dual Core processor and 2 GB of RAM. The operating system used was Kubuntu Linux with kernel 2.6.32. The code was written in C++ language and compiled with the gcc-4.4.1.

A model of the aircraft is needed to simulate and evaluate the suitability of the generated trajectories. Complex models can be used [29][30]. However, the trade-off with the required computing time should be considered. Other models could be used since the developed system can be adapted to different mathematical models of the aircrafts. In this paper we consider a small scenario whose dimensions could be used to perform experiments in the future. Moreover, small aerial vehicles are considered with some constraints. These constraints can be adapted a limits of maneuverability if aircrafts are considered.

The minimum horizontal separation between aerial vehicles in XY plane is $S_{xy}=0.8m$. The dimensions of the simulation scenario are 10mx10m. Two hundred tests have been performed for each subset. The number of intermediate waypoints, $M$, is set to 1 in both methods. Therefore, each solution trajectory is composed of two segments. The allowed speed for each aerial vehicle is between 0.3m/s and 2m/s. The speed is computed randomly in each segment. Each aerial vehicle should meet its ETA considering a margin, $m_{eta}=8s$. If a particle does not meet $m_{eta}$, it is penalized.

Figures 3, 4 and 5 show the mean time of execution, mean minimum cost and the deviation of the mean ETA, respectively. Each box of the figures depicts statistics of the 200 test performed for a given number of aerial vehicles. The central mark is the median, the edges of each box are the 25th and 75th percentiles, and the whiskers extend to the extreme data points.

Figure 3. Time of execution against number of aerial vehicles after 100 iterations.

Figure 4. Minimum cost against number of aerial vehicles after 100 iterations.

Figure 5. Deviation of ETA of each aerial vehicle against number of aerial vehicles after 100 iterations.

The relation between the time of execution and the number of iterations performed, and its dependence with the number of aerial vehicles is also analyzed (see Figure 6). The slope usually depends on the number of aerial vehicles and the relation is linear and additive.

Figure 7 shows the evolution of the mean minimum costs with the iterations. The system finds a better solution as time passes, i.e. a smaller cost each iteration. The mean of the minimum costs computed in the population of all the tests has been chosen as statistical indicator.

Figure 6. Time of execution vs. number of iterations depending on the number of vehicles.
The indicator shows how much time it would cost to achieve a solution with certain level of optimality. This relates the cost in a given iteration to the obtained minimum cost in the corresponding problem.

Figure 8 shows a normalization of the cost against the number of iterations. A line that marks the required number of iterations to compute for a 90% level of optimality is drawn. As the test set is executed in the same computer where both methods have been implemented, Figure 6 will provide an estimation of the time needed for that number of iterations, and therefore, that level of optimality.

For the cost normalization, a linear transformation, \( f(x) = ax + b \), is applied to the actual cost values to set them in the range \([0,1]\). Therefore, \(a\) and \(b\) are chosen in such a way that the maximum cost equals to 1 and the minimum cost equals to 0. Therefore,

\[
a = \frac{1}{\text{Cost}_{\text{max}} - \text{Cost}_{\text{min}}} \quad (5)
\]

\[
b = \frac{\text{Cost}_{\text{min}}}{\text{Cost}_{\text{min}} - \text{Cost}_{\text{max}}} \quad (6)
\]

Depending of the number of aerial vehicles, a solution of great quality, 90%, is computed when forty or sixty iterations have been performed. Moreover, this algorithm based on PSO presents better results than the algorithm based on genetic algorithms presented in [32].

This study shows clearly the advantages of the 4D trajectory planning algorithm based on PS of the ASCDR system. Thus, the advantage of this method can be observed because a good quality of the solution can be obtained with much less than a hundred iterations. The quality of solution improves as the time increases showing the interest of the anytime approach.

Furthermore, the reliability of ASCDR system is demonstrated because it always finds a solution by considering two hundred simulations for each number of aerial vehicles.

Figures 9, 10, 11 and 12 show how the quality of the solution improves as the time increases by using the ASCDR system. The quality of solution is reported in ten instants. The time of execution depends on the number of vehicles and the quality of the solution always improves with time increases; that is, the anytime approach yields trajectories whose quality increases with the available computation time.
Figure 12. Anytime approach with five AVs.

CONCLUSION
The ASCDR system, a 4D trajectory planning algorithm based on an anytime stochastic optimization approach, have been presented. This system detects conflicts in 4D trajectories of multiple aerial vehicle using an algorithm based on axis-aligned minimum bounding box, and resolves them cooperatively using a collision-free 4D trajectory planning algorithm based on a roundabout to quickly compute a feasible but non-optimal initial solution and Particle Swarm Optimization to improve incrementally the initial solution. The ASCDR provides a valid collision-free set of 4D trajectories at any time, so it can be used with short look-ahead times, although a sub-optimal solution will be obtained. Thus, the ASCDR is well suited to situations with variable look-ahead times, depending on the number of aerial, the involved flying segments and the distance to potential conflicts.

The ETA of each aerial vehicle is also met considering a deviation margin. This characteristic is important in the future ATM system.

The algorithm has been validated with many simulations performed in different scenarios and several studies to analyze the characteristics of the algorithm.

Future work will include the validation of these techniques with a larger number of aerial vehicles (up to 10). Moreover, a validation of the system experimentally with aerial vehicles will be performed. Also a simulation considering real traffic will be execute by considering complex models and constraints of maneuverability.

ACKNOWLEDGMENTS
This work was supported by the European Commission FP7 ICT Programme under the project ARCAS 287617 and the CLEAR project (DPI2011-28937-C02-01) funded by the Spanish Research and Development Program. Also the partial funding from the Junta de Andalucia project P09-TEP-5120 is acknowledged.

REFERENCES


31. “Alpha test,” http://parasol.tamu.edu/groups/amatogroup/benchmarks/mp

Assessment of impact of Degree of Automation on Human Roles: The Experts’ Analysis using Gaming

R. Casar, A. Arranz, B. Escribano
Isdefe
racasar@isdefe.es

A. Grosskreutz, R.Garcia
CRIDA
argroskreutz@crida.es

R. Suikat, M. Jipp
DLR
Reiner.Suikat@dlr.de

G.D.R. Zon, M. Joosse
NLR
Rolf.Zon@nlr.nl

ABSTRACT
This paper describes the results of the two assessments performed to evaluate the impact of increasing levels of automation (LoA) on Human Factors aspects and the impact on their responsibilities and interactions. This activity has been developed within the “Assessment of Degree of Automation on Human Roles” (ADAHR) project inside the framework of SESAR Work Package E, which is addressing long-term and innovative research (http://www.adahr.eu).

These assessments have been carried out in two environments, “Airspace Organization and Management” and “Airport Operations Centre”, using experts’ performance in role-based games in two steps: a preparative “Paper Based” gaming session followed by a complementary “Hardware Based” gaming session on a hardware platform.

The results of these two exercises have been gathered from the participation of experts in the different games. They will be presented by environment and taking into account the roles involved in each environment. Aspects such as trust in the system, situational awareness (SA), confidence and performance were addressed.

Author Keywords
Gaming; Human Factors, Automation, ATM

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

General Terms
Human Factors, Automation, Assessment, ATM.

INTRODUCTION
The Air Traffic Management (ATM) requires changes to support expected future demand. Modernization of the ATM systems includes increased automation as part of the solution (for example with the system Airborne Collision Avoidance System, ACAS, to reduce the risk of mid-air collisions [1]). Thus the impact of the automation and new technologies on the human operator should be considered to ensure the success of the implementation and deployment of those systems (examples can be found in the literature ([2] and [3]).

Automated systems must be compatible with human capabilities and the development of effective and usable automated ATM systems requires Human Factors input throughout the design life cycle, from concept formulation, through detailed design, to implementation and operation.

At this point, the goal of ADAHR project is enlarging the knowledge about the impact of automation on human roles, aiming at investigating how the automation impacts on the interactions and SA among several actors with different interests (e.g. airspace users and air navigation service providers) and focusing on the long-term phase (beyond 2020 horizon).

The necessary LoA for 2020 in a Single European Sky ATM Research (SESAR) environment is studied in the SESAR Joint Undertaking (SJU) programme, but this level of automation is expected to be higher in the ATM environment after the SESAR development phase. Therefore the assessments carried out embrace three different time horizons: 2020, which will be the starting point used as reference to allow a comparison between the different time frames, 2035 and 2050. It is assumed that the LoAs will increase over time and the baseline scenario, which is the 2020 timeframe, will be the expected situation when improvements of SESAR development phase are in operation.

Due to the limitations of time and effort within the ADAHR project, the complete list of ATM roles could not be assessed. The boundaries were set out to assess a suite of roles in two different environments: “Airspace Organization and Management” and “Airport Operations Centre”. The analysis focused on the Demand and Capacity Balance (DCB) processes where imbalances are produced by abnormal situations such as bad weather or a special event such as the final of a football match when passenger demand increases in a short timeframe.
The results of these assessments were obtained from three different sources: expert’s judgment of each of the environments, a statistical analysis performed by students in the Airport Operations Environment, and a global assessment independent of the environment. This paper presents the results from the experts’ assessment.

GAMING AS ASSESSMENT TECHNIQUE

Human-In-the-Loop (HIL) Gaming techniques are “serious games”, designed for a specific purpose other than pure entertainment. These games are played with persons (mostly, experts) acting as actors and allow the exploration of concepts and definition of roles and processes in a structured way focusing the players’ attention on the flow of information and responsibilities associated with the processes.

Gaming was selected to assess the impact of automation on human operators as it has previously been successfully used to explore situation awareness and the human-human and human-machine interactions in automated environments [5, 6].

A combination of two types of gaming techniques, (Paper-based gaming and Platform-based gaming), were utilised because experience in ATM assessment has proved that this combination of role-based games provides a good quality assessment of the process involved in the concept under test as shown in [5, 6]. More details about the methodology can be found in [4].

Paper-based games (PBG) are performed using basic office material. They are basically board games where the rules are designed according to the processes and roles interactions, (for example, needs for communicating between them), to be studied/clarified. Hardware-based games (HBG) are performed in the same way as the paper-based ones, but the means/tool used to play is a hardware platform. The HBG use and complement the results obtained from PBG with the following benefits:

- the platform allows the execution of the exercises in a more realistic context, showing the information as it will be in real life.
- platform results are more reliable and accurate. The players judged HBG as more realistic than PBG. This issue is supported taking into account the results from other gaming exercises performed within the ATM environment as described in [4] and [5].
- the performance analysis is easier,
- and for this project, there is a strong link with automation and human roles.

Concretely for the assessment carried out in ADAHR, the combination of both techniques allowed the definition and exploration of roles and their responsibilities and the interaction of these roles within an automated environment:

- the paper-based games produced high-level and preliminary outcomes to support the platform's configuration and they served as training about the technique and the scenarios to be analyzed.
- the hardware-based games completed the results and supported quantitative data analysis (only for Environment 2).

The platforms used in the hardware-based games were:

- CHILL (Collaborative Human-In-The-Loop Laboratory) to analyse the “Airspace Organization and Management” environment, henceforth Environment 1. CHILL is a versatile collaborative ATM validation platform in which different categories of actors can work together to efficiently manage traffic demand and capacity, exchange ATM data and share information in support of a collaborative Air Traffic Flow and Management planning process.
- ACCES (Airport operations and Control Centre Simulator) to analyse the “Airport Operation Centre” environment, henceforth Environment 2. ACCES provides several operator working positions as well as a means to provide a common overview of the situation to the operators. This platform has a flexible infrastructure with up to ten operator working positions as well as a large screen to show a situation overview to all operators.

APPROACH OF THE ASSESSMENT

The assessments performed were focused on the achievement of the following objectives:

- Identification of mechanisms that enhance the trust of the human actors in automation.
- Analysis of the impact of automation on the interaction between human actors.
- Analysis of the impact of automation on the new roles and responsibilities foreseen by the LoAs addresses, especially in terms of workload and SA.
- Analysis of the tools supporting the ATM human actors’ duties

In order to reach these objectives, the first stage was the definition of the new roles and responsibilities of human operators for the two environments, starting from the SESAR definitions of human roles in 2020 and the expected technical capabilities of automation [7]. Next, the scenarios were defined for the two different environments and for three future automation levels (corresponding to a situation around approximately 2020, 2035 and 2050) [8]. After this the design of the different exercises through the development of different games was planned and reported [9,10,11,12] and subsequently, the impact of the automation described in the scenarios on human roles was assessed through the execution of the designed games per each environment and finally the outputs were analyzed and reported [13,14]. The process is shown in Figure 1.
Figure 1: Process flow followed to carry out the ADAHR assessments

To analyze the impact of automation on human roles the following metrics were measured: Trust and confidence, Workload, Performance, SA and Teamwork. These were assessed using questions extracted from standard scales of validated questionnaires that they generally applied in ATM. The questionnaires used as baseline were: NASA Task Load Index (TLX) [15], Controller Acceptance Rating Scale (CARS) [16], SHAPE Teamwork Questionnaire [17], SA Rating Scale (inspired by the Cooper Harper scale) [18], Scale of Trust in Automated Systems [19] and the IT Suspicion Scale [0].

Given the limited number of participants, (three participants per environment), the assessment performed by the experts was purely qualitative assessment. To enrich the nature of the results of the project, additional platform-based gaming sessions focused on Environment 2 were planned to provide a quantitative assessment which could give more value to the analysis of the results. Due to the issue to find expert’s available for the assessments and due to the big number of players needed to obtain a significant sample (60 persons in 20 groups of three participants were requested to allow drawing statistically valid results), the participants of these gaming sessions were finally students or non-experts.

The outputs of the execution of the expert’s games were collected through: observations and notes taken by the management team during the execution of the games, questionnaires completed by the actors and notes taken during the debriefing session after the run had been carried out.

The different gaming sessions were performed consecutively and the duration of each gaming session was between two and three days. The sessions for Environment 1 (paper-based and subsequently platform-based) were followed by the sessions for Environment 2 (also paper-based and then platform-based). This methodological sequence allowed the applications of lessons learnt from each session to subsequent ones, improving the quality of the results obtained and reducing the time needed for the preparation and execution of the exercises.

Next section deals with the description of the environments under study, through the description of the scenarios and their roles. This description will facilitate the readers’ understanding of the results.

ENVIRONMENTS DESCRIPTION

Environment 1: Airspace organization and management

The scenarios of Environment 1 featured the impact of a non-severe capacity shortfall due to forecast bad weather (i.e. the capacity shortfall would not be so severe to lead to the application of a queue management process [10]). The bad weather was expected to occur over Madrid, impacting the capacity of Madrid-Barajas airport and the Madrid Air Traffic Control Centre (ACC). This happened during Madrid’s high season, when unexpected events might be more likely to disrupt operations and the agreed service level might fail to be met. The scenario encompassed the whole process - from the detection of the imbalance until the approval of a solution to be implemented.

More specifically, the scenario consisted of the following core processes:

- Build/Refine Reference Traffic Demand;
- Detect Airspace Demand Capacity Imbalance;
- Select/Refine/Elaborate on a DCB Solution at Network Level;
- Assess Network Impact of the DCB Solution;
- Start User Driven Prioritization Process (UDPP) on Shared Business Trajectories (SBTs).

Environment 1 Roles

The following roles took part in the scenarios:

- Regional Network manager (RNM);
- Local Traffic Manager (LTM);
- Airport collaborative decision making (CDM) Manager
- Airspace Manager

The list of responsibilities of these roles was provided in detail in [9,10]

Environment 1 Scenarios

The 2020 scenario was used as the baseline. It is assumed that SESAR was in place and that airspace users expressed their preferences which were reflected in the priority level of a flight. The DCB tools were automated to provide a complete set of solution/action alternatives on demand, which did not completely solve the situation, but simplified it. These solutions took the priority of a flight into account.

In the 2035 scenario the DCB tools were automated to provide a ranked list of solutions/action alternatives on demand, which completely solved the situation, but had different cost of solution indexes. The cost of solution index took into consideration the business needs of the different actors and the airlines. This scenario was run twice, with different locations and impacts of the storm.

In the 2050 scenario the DCB tools were automated to identify and solve demand capacity issues and try to optimize capacity usage. They provided the solution that
best fitted the pre-agreed parameters reflecting the business needs of the different actors and airlines, and an equity parameter taking into account previous penalties. This solution had to be approved by the actors. The 2050 scenario had two different runs with different location and impact of the storm.

It is important to highlight that UDPP process could not be performed in the games of Environment 1 due to the lack of availability of experts and also because the lack of maturity of UDPP process definition.

**Environment 2: Airport Operation Centre**

The scenario designed for the games was based on a special event taking place at a European airport with high demand. This special event was a football match, the Champions League final, featuring a British and a Spanish team. This event meant that there would be an airport demand increase, due to special flights chartered for the occasion.

Reception of the extra flight plans and the search for a solution to the capacity problem took place during the day of operations, several hours before the match. It consisted of the following core processes:

- Detect Airport Demand/Capacity Imbalance
- Select/Refine/Elaborate a Demand Capacity Balancing (DCB) Solution at airport level

**Environment 2 Roles**

- Airport Agent
- Major Airline Agent
- Charter Airline Agent

The list of responsibilities of these roles was detailed in [11,12]

**Environment 2 Scenarios**

The 2020 scenario was used as baseline to compare the other two scenarios, 2035 and 2050. The LoA was supposed to be the lowest and thus most similar to current situation. For this scenario each actor had a different what-if tool to assess solutions. The information shared with all actors was limited. For this timeframe, coordination and communication between actors acquired greater importance.

In the 2035 scenario, the system created and proposed a set of possible solutions based on Key Performance Indicators (KPIs) such as punctuality or throughput, but not on the cost models of the actors, which were still confidential for each of them only. Then each actor selected his/her own solution and they had to agree to a common solution for all actors involved in the process.

In the 2050 scenario, System Wide Information Management (SWIM) was assumed to be fully implemented and all actors shared all the information in real time, so that communication required between them was limited. The LoA for this scenario was the highest and the system knew the cost model of all the stakeholders, so that it looked for the solution with the highest total profit for the whole operation. This scenario assessed the acceptance of the solution proposed by the system, which was supposed to be the best, globally speaking. Agents had to accept or reject the solution proposed by the system.

For this environment, different runs were planned to minimize the effects of the order of execution of the scenarios. In environment 1, the order was 2020-2035-2050. Then nine runs were planned and the variants of this scenario were:

- Scenario 1: a set of additional charter flights was announced on the morning of the day of the match, resulting in a heavy overload of the airport, as the available parking capacity was not sufficient for all flights without creating additional parking space by closing a runway.
- Scenario 2: a heavy storm was forecast for the peak arrival traffic period, resulting in required increased separation of arriving flights and hence in a reduction of the runway capacity.
- Scenario 3: the airport announced a planned closure (for maintenance) of one of the runways during the peak traffic hour. The closure could be moved in time by some margin.

In all three cases the task of the stakeholders was to make the best possible use of the available resources and to conduct traffic as efficiently as possible.

**PLAYERS**

Five experts participated as actors in the experts’ gaming sessions for the two environments. They were all males and they all had 10-25 years’ experience in ATM. This ensured that experts had the appropriate knowledge and skills to provide reliable results. They were of different nationalities (British, Italian, Romanian, German and Spanish) and four of the actors had no previous experience of using gaming techniques.

Three actors participated in each gaming session. This means that each of the actors played a unique role except one of them who was able to play two different roles, taking into account his previous experience: Regional Network Manager for Environment 1 and Charter Airline for Environment 2. Furthermore the same set of actors participated in both, Paper-Based and Platform-Based gaming sessions of each environment.

As mentioned the number of participants in the expert’s gaming sessions allowed to obtain only qualitative results. These descriptive statistics showed the trend of the indicators by roles and by timeframe, (i.e. LoA).
RESULTS
This section presents the main results obtained for each of the two environments from the gaming exercises performed by ATM experts. More details can be found in the report documents [13, 14].

Following considerations must be taken into account for the proper understanding of these results:

- The assessment was made taking into account two issues: the interpretation of the implementations expected for 2020, 2035 and 2050 timeframes, stated in [8] and; the interpretation of the impact of those implementations on the roles and the responsibilities, stated in [7].
- The results of the expert’s gaming sessions were obtained through a qualitative assessment and based on the opinion and experience of the five experts. The numerical data obtained from the questionnaires were interpreted as trends in the behavior of the role over the different timeframes, (i.e. LoAs).
- Failures in the system were out of the scope of the assessment.

Environment 1
The study of this environment was performed in two different gaming sessions, one using PBG and another one using HBG.

The paper-based sessions resulted in two types of outputs:

- Feedback for the design and configuration of the platform-based games
- Qualitative results related to the ADAHR objectives assessment.

The main outputs of the paper-based gaming sessions to the platform-based gaming sessions were:

- Refinement of the processes involved in the different timeframes, and the actions performed by each actor: the experts pointed out some actions or processes that should be changed or improved to be more realistic and adapted to a future real scenario.
- Improvement of the definition of the LoA that should be in place in each year: the experts considered the LoAs, based upon their current experience, proposed for 2020 and 2035, as very low;
- Improvement of the definition of the solutions to be provided by the automation: the experts commented the information that they thought was missing in order to have sufficient confidence in the system and to facilitate their operation (for example, the solutions that the system proposed in the 2035 timeframe were not considered realistic conforming to the expected LoA for this timeframe).

- Initial awareness of the expected outcomes from the simulation: At the beginning of the gaming sessions, the experts were informed about the objectives of the project. After the execution of the different game runs and after they completed the corresponding questionnaires, they had gained a deeper knowledge of the objectives of the exercise and the assessment to be done, in terms of the questions and metrics to be evaluated.
- Initial knowledge by the expert participants on the scope of the simulation: The paper-based gaming sessions in this environment represented the players’ first experience with the gaming technique and also with the scenarios. This served as training for the general performance of a game and, more importantly, for the processes and responsibilities of the roles that they performed.

These outcomes were translated into a refined set of requirements for the hardware platform [10], refined validation scenarios, and a better background document for the participants.

Environment 1 Overall results
The results of the paper-based gaming sessions suggested that actors felt more comfortable with the 2050 environment than with the intermediate 2035 level, (based on their answers to the questionnaires and also taking into account the comments of the experts during the execution of the games and during the debriefing). The results of the platform-based gaming sessions seemed to confirm this, but in the end; the experts pointed out the need to remain in control of the situation. This assessment could be explained by two facts: the order effects of the performance of the games (2020→2035→2050) or, although the experts requested that the system proposed more solutions so as to have more options, they did not understand the rationale behind them.

The following provides a brief description of the most remarkable aspects found during the different timeframes of Environment 1.

The 2020 timeframe: The scenario related to 2020 timeframe was the first to be performed and was used as training for the actors. In addition, discussion about processes and use of tools took place during these sessions.

The 2035 timeframe: In one of the runs two actors were working in parallel to find a solution with the data of the system and finally, one of them was working with old information. So that it was suggested adding an indicator displaying when an actor was working on the solution. The actors felt more comfortable than in 2020 timeframe but they could not tell if it was because they knew the system better or because automation had solved the whole situation.

Another interesting issue was brought along in this timeframe: the RNM took one solution influenced by human kindness. The actors questioned whether or not this
parameter would be lost when automation was increasing and the systems were more and more involved developing the solutions and making the decisions.

The 2050 timeframe, the actors were told that there was a conference where all the parties in the ATM/airport reviewed the performance of the network in the previous season and agreed the performances of the next season. During the 2050 debriefings, the actors said that the related parameters should include different preferences (These related mainly to cost and fuel/lack of fuel by 2050).

One of the main discussions was about the need to clearly know the limits of the ATM network. The situation was provoked by a situation in which a small network effect produced an overload of two flights in one sector over Europe. The RNM approved the solution because the overload was very low (and he knew that sector’s capacity was above the official one that allowed the controllers to deal with this type of situations). Therefore the related solution cost was the best one. The APOC indicated that he was talking about current capacity definition and that probably in 2050 the capacity would be the real limit or very close to it. The flexibility of the human compared to the machine was discussed.

Environment 1 Metrics assessment

The assessment on the impact of automation on human roles was based on qualitative results from the comments of the experts and also based on the analysis of some metrics assessed in the questionnaires. To do this, the outputs of the questionnaires were used to assess the metrics:

Role of automation – The participants rated the LoA as adequate to perform their activities. They positively assessed the substitution of human tasks for the performance of their activities although the 2050 timeframe was considered (by a slight margin) to be the best one. Furthermore the participants accepted the increasing LoAs. They thought that, given the level of traffic in the 2035 and 2050 exercises, a high LoA was required to be able to undertake the work. However the LTM felt more comfortable with the LoA in 2035, where he had to review the solution and had several options available to him. The RNM showed a slight preference for the 2050 situation but he still wanted to be able to review the solution, checking that everything was ok.

Active Involvement of humans – The participants perceived a decrease in the degree of the human involvement in the tasks but they did not experience it as inconvenient for their performance.

Confidence and self-efficacy – The participants rated their perception of the execution of their tasks as positive and they felt confident with the solution given by the system. These data were based on the low rates of the degree of frustration in their performance, with the 2020 situation being the most frustrating.

Situational Awareness – the participants stated that their SA was sufficient for the given LoA and traffic density for that year. The LTM rated SA highest for 2035; the RNM and CDM Airport Manager rated their SA lower.

Trust in the system – It was seen that the level of trust was maintained as the LoA increased. Feelings of system reliability, accuracy, usefulness and confidence increased over the periods of time. However, the RNM stated that he would have liked to be able to perform a final "check" on the solution before accepting it.

Workload – the actors rated their workload as acceptable for the given automation and traffic density of that year, although there was a difference for each participant in the degree of decrease in workload as the LoA increased. As there were only three participants, there can be no speculation as to the cause of this difference. Besides it was suggested that since the automation was solving all the local DCB conflicts, the role of LTM approval of the solution could be incorporated into the duties of the RNM.

Needs of possible future support tools – Since the general public becomes more accustomed to communicating textually (IM, SMS/texting, etc.), the acceptance of this mode of communication in a DCB planning environment will increase and could be used to advantage. This was appropriate since the activities occurred during the planning phase of flight, and not execution, so that there was no time pressure. Other needs revolved around the visualization of the solution and each actor’s ability to see what had been implemented, if he/she so desired. Some proposals for improvements were:

- Ability to show multiple scenario data: it was recommended that future DCB systems aiming at this LoA incorporate means for the actors to visualize the scenarios being proposed for implementation and the changes involved.
- The rules used by the tools to calculate the best/optimal solution should be known by all the participants.

Environment 1 Lessons learnt

As result of the performance of these gaming sessions the following conclusions were made to be taken into account in next environment assessment:

- The combination of both gaming techniques was very suitable to perform this type of assessment: The HBG platform offered the realism that players needed to perform complete the assessment started with the paper-based games. These latter also supported the familiarisation with the concept and the configuration of the platform. This allowed players to concentrate on the assessment on the most important and interesting issues.
- Training was crucial to achieve results in which one is fully confident. The training period must be extensively planned so that participants can feel comfortable with the
platform or tool operations and they can be focus on the concept or issue to be assessed.

- It was recommended that individuals participate in both paper and platform exercises and that they assume the same roles. This allowed them to acquire the experience in the gaming technique and gained knowledge about the scenarios, thus the actors were able to be focused on evaluation of the objectives of the simulation.

- Changing roles in the same scenario did NOT demonstrate to work well. There was an attempt to change the roles of the actors to have more independent results, (i.e. results about the same role performed by different persons), but it proved to be unsuccessful when low numbers of runs are scheduled. In this case, it was better to maintain the same roles to facilitate familiarisation with the technique and the concept.

- The order effects (2020 - 2035 - 2050) should be taken into consideration for any future games of this nature. There was a feeling that learning and other positive carry-over effects would be stronger than negative effects, such as fatigue, which would lead to results which were positively biased for the later scenarios.

Environment 2

Similarly to the Environment 1, the analysis was performed in two different gaming sessions, one using paper-based games and another one using platform-based games.

The difference with the environment 1 is that the qualitative results of the gaming sessions carried out by Experts will be complemented with the statistical analysis of the data obtained from the performance of the students gaming session. The analysis of these results is still on-going at the moment in which this paper is being developed.

Taking advantage of the lessons learnt from the Environment 1 gaming sessions, the following aspects were considered:

- the execution of several runs per timeframe (3 per each timeframe) was planned to minimize the order effects of the different scenarios.
- the same actors participated in both gaming sessions.
- the actors assumed the same roles for all runs.
- Training was carefully planned and the performance of the game (i.e. the way in which the game is played) was simplified to avoid the need for the actors to have detailed knowledge of interaction with the tool.

The paper-based sessions resulted also in two types of outputs:

- Feedback for the design and configuration of the platform-based games
- Qualitative results related to the ADAHR objectives assessment.

The main contributions of the paper-based gaming sessions to the platform-based gaming sessions were:

- The solution to cancel a regular scheduled flight in favour of additional charter flight was considered as unacceptable: The scenario and cost model were modified such that cancellations of scheduled flights were not necessary to achieve an acceptable solution.
- The ground handler role, as separated role, was missed by the participants: The tasks of ground handler were included in the Airport Agent duties. However, providing a separate ground handler role could not be done for the PBG, as this would change the team and furthermore, the ACCES platform did not have the capabilities to manage the ground handler aspects sufficiently.
- The actors found the system provided solutions that they considered were not sufficiently transparent, i.e. they could not see the rationale behind them. This effect was considered a limitation of the PBG; therefore, the HBG would give the actors more options to explore and evaluate the proposed solutions.

Environment 2 Overall results

From the global analysis of the paper-based and platform-based gaming sessions, the main aspects about the impact of automation on the Environment 2 roles were:

- Personal opinions about automation and whether or not automation results in an optimal solution affected the average rating given by the actor. The rating over time of the actors was comparable.
- The actors stated it was not clear who was responsible for the final decision.
- Trust in automation over time remained the same, but more insight in the reasoning of the tool would be needed to improve trust and understand the solution provided by the tool.
- One comment about the importance of User/Human Centred Design was considered crucial for trust in the systems and confidence in the performance: “I have to imagine that although I don't know the logical implementation of my requirements in the tool, it should comply with all of them since I assume that the system has been developed and tested by myself and the rest of the stakeholders since the very beginning of its lifecycle”
- The ratings by the two Airlines were often comparable while the Airport ratings differed somewhat. The Airport Agent tended to assess more positively the impact of the LoA on his own role than did the Airline Agents.

The most remarkable aspects found during different timeframes of the Environment 2 are detailed in the following paragraphs.

The 2020 scenario: The actors were still in command using a tool to calculate the benefits of the solution. However they were not very familiar with the tool and finding an
optimal solution took a lot of time in comparison with real life. The actors commented on the need for more training with the tool so as to have more confidence in the results. In fact, the actors experienced a training curve effect with the system over the runs for each timeframe. In addition to this the actors stated several times the large amount of effort and time spent in negotiation: “At the end of the day I lost money as the airport agent had to ultimately make a final decision. We worked at nine different solutions and in the end, my business lost out. There was nothing I could do to resolve it”.

Other comments noted for this timeframe included:

- some information about the airport was crucial for decision making processes, (e.g. location and alternative airports)
- the possibility of swapping slots seemed not to be feasible, although the auction of slots resulted in a better more possibility.
- the airport had more confidence in the information provided by the Major Airline than Charter Airline. This confidence was based on familiarity and knowledge of the Major Airline’s systems

The 2035 timeframe: Although the actors had less influence on the possible options (defined by the system) than in the 2020-timeframe and a lower SA, they were still in command as they could choose an alternative. The actors felt responsible and the confidence and self-efficacy was rated higher than in the other time-frames. The Airport Agent felt that the information provided by the system was correct and that he could access to the information that he needed: “Solutions are presented in a desired and clear way. It is easy to choose within the three options with the rest of participants, and if one needs further details, the tools present a lot of additional information”.

Other comments noted for this timeframe included:

- It was assumed that in 2035 airlines would have to negotiate directly (without the intervention of the airport), both airline representatives considered that the behavioural change would be as important as the change in systems and procedures.
- For Charter Airline, the optimisation of global KPI was less important than maximizing their profit. This might not be the case for the Major Airline.

The 2050 scenario: The actors were not convinced the system provided the optimal solution and as the actors had no way to check or optimize the provided solution, they felt powerless and without control over their jobs and with very low SA. One of the actors defined as a “the leap of faith “ the acceptance of the solutions that a high-automated system can provide.

Additional feedback was provided about the information that should be shared and known by all the actors. The actors playing the role of the Airlines pointed out that only global numbers should be shared. The information related to the Airport should, however, be shared by all partners due to its importance in decision-making.

Environment 2 Metrics assessment
Similarly to Environment 1, the assessment on the impact of automation on human roles was based on qualitative results coming from the comments of the experts and also based on the analysis of some metrics. To do this, the outputs of the questionnaires were used. The analysis of the metrics is described:

Role of automation – There was an overall increase in the positive role of automation over time. However the participants assessed as more adequate the LoAs in 2035 and 2050. There were some differences between the roles:

- the Airport Agent gave the assessment that, in general, the LoAs proposed for all timeframes were more than adequate in comparison with the Airline Agents.
- the Major Airline considered the adequacy of the LoA in 2020 very low.

Active Involvement of humans – The participants perceived a decrease in the degree of human involvement in tasks but they did not make any negative comments.

Confidence and self-efficacy – The participants did state that their confidence and self-efficiency was better in the 2035 timeframe. The 2050 scenario was the worst one evaluated by the players in terms of confidence and self-efficacy. Globally the Airport Agent felt more confident and efficient in comparison with the Airline Agents.

Situational Awareness – this indicator was differently perceived by the actors. The Airport Agent rated his SA as good, and highest for 2035. The Airline Agents experienced a decrease in their SA as the timeframes progressed, with this assessment being highlighted by the low SA that the Charter Airline declared in 2050

Trust in the system – Trust in the systems remained stable during all timeframes and no differences over time were stated by any of the actors. Again the Airport Agent declared himself to have a high level of trust in the system. However the Airline Agents were uncertain about the degree of trust for all three timeframes.

Workload – it was not directly measured. However one indicator could be the time that took to reach an agreement to find a solution. So that the 2020 was the timeframe in which this time was longer and this was, therefore, the timeframe in which workload was higher.

Needs of possible future support tools – There were some proposals for improvement on the systems:

- more insight in the reasoning of the tool is needed to trust and understand the solution provided by the tool
• the view of the overall performance was rated as very positive since this is a feature nowadays missed. However the possibility to influence on specific flights in addition to the influence on details of Airport Operation Plan (AOP) was requested.

• since the process to make a solution was performed through the modification of different indicators, the visualization of the operational changes as consequence of this change was also requested.

VALIDITY OF THE RESULTS
As mentioned before, the expert’s assessment was only performed in a qualitative basis and the results obtained were based only in the opinion of the five experts who participated in the gaming sessions.

In order to assess the validity of the results, some questionnaires about the expectations and the confidence in the results were performed to collect the actors’ feedback. To do this, the questionnaires were completed before and after the gaming sessions. The questions were about the suitability of the gaming technique for this type of assessments and about the confidence in the results.

The analysis of the questionnaires showed that despite the fact that the most of the experts had no experience in the Gaming Technique, none of them expressed scepticism with respect to the technique for either of the environments.

The belief in suitability of Gaming Technique for assessing new concepts decreased slightly for paper-based games. This can be attributed to the fact that these games do not “feel” realistic, as some of the participants remarked. After the platform-based games the opinions were – as expected – considerably more positive, mainly because of the higher degree of reality owing to using a platform. Another positive effect may come from the fact that the same actors as during the paper-based gaming session were participating in the platform-based gaming session, which was considered essential for the overall assessment by all three actors.

This latter fact increases the validity of the results because the actors rated as positive the performance of the technique and the usefulness of the results that they provided. Furthermore the assessment on the metrics was made based on validated and standard scales to give more value to the results obtained.

CONCLUSION
The impact of increasing LoA on the interaction between human actors and on their roles and responsibilities was assessed for the “Airspace Organization and Management” and “Airport Operation and Centre” environments by consecutive paper-based and platform-based gaming sessions by each environment. The participants in these games were five ATM experts.

The paper-based games were used to refine the scenario, to provide the platform-based gaming with the right input and to do a preliminary assessment of the effect of higher LoAs on the involved ATM actors. The platform-based gaming sessions were completely focused on assessing that effect. The gaming sessions met the expectations and the combination of the paper and platform-based gaming sessions has again demonstrated their suitability for this type of assessments.

The results of Environment 1 showed that:

• The increasing role of automation was positively perceived over the time. The adequacy of the LoAs was better evaluated for the 2050 and 2035 timeframes. This was supported by the actors’ confidence and self-efficacy in these timeframes.

• Situational Awareness was assessed by all the actors in all timeframes as sufficient, and being significantly better for 2035 and the LTM airport.

• Trust in the system was maintained over the timeframes but highlighting the preference that humans were still in command to accept and apply the solutions

• The active involvement of humans decreased over the time, as expected.

Other remarkable comments were:

• Parameters such as flexibility and human kindness are better used by humans than by automated systems and this should be taken into account in the design of the systems, so as to provide the possibility for interactions between human and machines.

The results of Environment 2 showed that:

• Although the adequacy of the different LoAs was rated as positive, the actors pointed out the best situation for the 2035 environment, being the best perception by the Airport Agent role. However the least well perceived LoA was for the 2050 environment by the Major Airline Agent. This was supported by the assessments on confidence and self-efficacy.

• Situational Awareness was assessed differently per role. The most positive assessment was for the 2035 environment by the Airport Agent role whereas the Airline’s Agents found their SA insufficient and decreasing over time.

• Trust in the system was also maintained as the LoA increased equally as in Environment 1. And the actors requested the need to have the final decision to accept and apply one solution as well. This would have allowed the actors to perceive the situation as being under control.

• The active involvement of humans decreased over the time, as stated in Environment 1.
Other remarkable comments were:

- Importance of the involvement of the end-user in all systems lifecycle, (especially in early stages, design and development and testing), for ensuring trust in the systems with increasing LoAs.
- Importance of insight into the reasoning of the system and the resultant required training. This is crucial for trust and understanding of the solutions provided by those systems when increasing LoAs.
- Responsibilities must be clearly defined when high LoAs are implemented. The authority and accountability should be clearly defined as humans could feel less responsible because their tasks will be simply monitoring situations and systems will provide the solutions.

These results about the impact of automation on human roles will be further reinforced by the results of the students’ gaming sessions, which will provide a quantitative data analysis.

ACKNOWLEDGMENTS

The work presented here is carried out in the project ADAHR, which is a consortium of Ingeniería de Sistemas para la Defensa de España (Isdefe), Centro de Referencia de Investigación, Desarrollo e Innovación ATM (CRIDA) in Spain, Deutsches Zentrums für Luft- und Raumfahrt (DLR) in Germany and National Aerospace Laboratory (NLR) of the Netherlands. ADAHR is co-financed by EUROCONTROL on behalf of the SESAR Joint Undertaking in the context of SESAR Work Package E. This paper reflects only the authors’ views and EUROCONTROL is not liable for any use that may be made of the information contained here.

Special thanks must be given to the five ATM experts who participated in the gaming sessions. None of the assessments could have been carried out without their proactivity, their experience and their willingness.

REFERENCES

1. Eurocontrol site (http://www.eurocontrol.int/acas/)
7. E.02.09-ADAHR Roles and Responsibilities D1.1, V00.01.00, 17/08/2011
8. E.02.09-ADAHR Scenarios Description D1.2, V00.00.02, 27/04/2012
9. E.02.09-ADAHR-D2.1-Exercise Plan Paper-Based Gaming ENV1 Ed00.01.01, 20/06/2012
10. E.02.09-ADAHR-D2.2-Exercise Plan Platform-Based Gaming ENV1 Ed00.01.01, 26/09/2012
11. E.02.09-ADAHR-D2.3-Exercise Plan Paper-Based Gaming ENV2 Ed00.01.02, 13/11/2012
12. E.02.09-ADAHR-D2.4-Exercise Plan Platform-Based Gaming ENV2 Ed00.01.01, 28/02/2013
13. E.02.09-ADAHR-D3.1-ADAHR- Gaming Report ENV1 Ed00.01.01, 29/01/2013
14. E.02.09-ADAHR-D3.2-ADAHR- Gaming Report ENV2 Ed00.00.02, 20/02/2013
17. SHAPE (Solutions for Human-Automation Partnerships in European ATM), http://www.eurocontrol.int/humanfactors/gallery/content/public/docs/SHAPE_questionnaires/SHAPE%20User%20Guide%20v0.1.pdf
## Automated Support for Separation Assurance: a Common Design Approach for General Aviation and Remotely Piloted Aircraft Systems

### Damiano Taurino
Deep Blue Research and Consulting
Piazza Buenos Aires 20, 00198 Rome, Italy
damiano.taurino@dblue.it

### Stefano Bonelli
Deep Blue Research and Consulting
Piazza Buenos Aires 20, 00198 Rome, Italy
stefano.bonelli@dblue.it

### Marco Ducci
Deep Blue Research and Consulting
Piazza Buenos Aires 20, 00198 Rome, Italy
marco.ducci@dblue.it

### ABSTRACT
In this paper we present a novel approach to automation design based on the joint design of two different tools, with different automation levels, different maturity levels, different application domains and different users characteristics. We show the opportunities and challenges brought by a joint design of Separation Assurance support tools for General Aviation and Remotely Piloted Aircraft Systems. We propose a methodology and tools for identifying and validating common requirements, thus creating a common core which forms the grounding for the development of two dedicated tools.

### Author Keywords

### ACM Classification Keywords

### General Terms
Human Factors; Design.

### INTRODUCTION
The idea of the airspace user is a key concept for the SESAR Programme: airspace users are considered to be at the center of future Air Traffic Management (ATM) system and they are among the main beneficiaries of the technical and operational improvements foreseen in the SESAR framework. The definition of airspace users includes several different categories of civil, military and state operators, flying by fixed wing aircraft, rotorcraft, balloons etc. This research paper focuses on two specific classes of airspace users: General Aviation (GA) and Unmanned Aircraft Systems (UAS), with particular reference to the category of Remotely Piloted Aircraft Systems (RPAS), as defined by the International Civil Aviation Organization (ICAO) in [1].

With GA one refers to all but commercial and military aviation GA therefore includes business aviation, touristic non-scheduled flights, private flights and air schools flights, so including a wide range of aircraft types that fly under very different conditions and rules, both IFR (Instrumental Flight Rules) and VFR (Visual Flight Rules). This class of airspace users is worth of attention because it is considered of strategic relevance for the achievement of the mobility and capacity standards foreseen for the year 2020 by EUROCONTROL [2].

The RPAS are another very interesting case study: although their use in military operations is consolidated and considered as routine, their involvement in civil missions in non-segregated areas with a consequent integration within the ATM system is a current and 'hot' topic. In the next few decades civil RPAS applications are supposed to be one of the fastest growing markets. Nevertheless the definition of RPAS as brand new airspace users represents a very challenging and critical issue to be investigated. For this reason Europe is financing several research projects, working groups and other activities to address the opportunities and challenges related to the introduction of RPAS as airspace users in the present and future ATM system. An interesting example is the ICONUS (Initial CONOps for Uas in Sesar) project [3], funded by SESAR.

RPAS and GA are very different categories of airspace users, with their own peculiar characteristics, history and evolution. Nevertheless some common high level challenges and requirements can be identified for their effective and proficient introduction in the future ATM system, as envisaged by SESAR. One of these is the need for automatic support of the pilot’s capability to detect and avoid the surrounding traffic, during all phases of flight. According to the EUROCONTROL Performance Review Report 2010 [4], GA flights are more prone to airspace infringements than other classes of flights. In fact, GA VFR flights are conducted outside controlled airspace and are in general flown by less-experienced leisure pilots. For the
RPAS class, the presence of an embedded Detect and Avoid system to support the remote pilot is a stringent requirement and a fundamental enabler for Beyond Line Of Sight (BLOS) operations [1], that are expected to be a great part of the RPAS civil missions in the next few decades.

This paper aims at: identifying the challenges and the potential opportunities brought by the design of a common interoperable concept of Detect and Avoid tool for GA and RPAS; providing a theoretical model of the problem; proposing design methodologies and solutions to tackle the challenges and exploiting the opportunities.

In the first section the background of this research will be presented: a description of the novel algorithmic approach to Collision Detection and Resolution problem we are using will be provided and a prototypal tool we designed for GA will be introduced. In the second section the research question we are addressing and the novelty of our approach will be discussed. In the third section the potential opportunities (wider market, uniform training, etc.), together with the challenges brought by a common design (different users, different levels of automation, operational context, etc.) will be identified and analyzed. In the fourth section solutions using a theoretical modeling of the problem will be proposed and possible design and validation methodologies will be introduced. Finally some conclusions and potential next steps for our research will be provided.

BACKGROUND

Separation assurance is the capability to maintain a certain safety distance (either vertical, horizontal or both) from the surrounding traffic and it is a key concept for Air Traffic Management and Aviation in general. The task of keeping separation among aircraft in controlled airspaces is performed by the Air Traffic Control, by coordinating appropriate strategies with the pilots and monitoring their accomplishment. In other conditions, like uncontrolled airspaces, in the case of VFR flights, separation by ATC is not provided and the aircraft pilot has the responsibility to deal with it. Planning, communicating and monitoring separation maneuvers are among the most demanding tasks for the Air Traffic Controllers. The idea of delegating part of them to the aircraft crew, while still maintaining the same level of safety, is one of the challenges accepted by the scientific community to provide the ATM system with effective tools to enhance safety and performance. In the literature a wide number of different approaches to the problem, often referred as “airborne self-separation”, has been provided [5] and most of these are based on algorithmic solutions to the problem of Conflict Detection and Resolution (CD&R). Other studies (e.g. SESAR WP E project SUPEROPT) [6] have focused on the problem of effectively presenting to the pilot the information provided by the algorithms. The mentioned research therefore encompasses design approaches for the Human Machine Interface (HMI).

Our research moved from the design of a Collision Detection and Resolution algorithm based on the framework of multi-agent systems modeling to its application in different operational domains, from autonomous UAS to General Aviation. In the following sub-sections two research projects approaching the problem of self-separation and proposing innovative solutions will be introduced: the ARCA Project and the SepaRA project. The experience brought by these two projects will be the starting point for introducing the common approach to automation design for separation support tools in different domains, proposed in this paper.

ARCA Project: Conflict Detection and Resolution for Autonomous UAS

A key factor for the diffusion of UAS is clearly the ability to safely perform their mission. In this context it is necessary that the UAS be capable of maintaining separation from other flights and obstacles. To address this issue, in the framework of the project ARCA (Adaptive Routing and Conflict mAnagement for UAS, funded by European Commission for the timeframe 2009-2012) [7], a CD&R algorithm for autonomous UAS has been developed, to ensure their self-separation from other aircraft, both in segregated and in non-segregated airspaces. The goal of ARCA project was in fact to develop an on-board flight system able to guide a UAS towards a specific destination modifying its own flight trajectory in reaction to external events. The system is supposed to be fully automated, with no human intervention.

The algorithm developed for ARCA is based on a Multi-agent System approach and the Satisfying Game Theory (SGT)[8]. Multi-agent Systems approach is a new computational framework which has proven to be effective in the resolution of several real-world problems dealing with multiple actors (e.g. packet routing, transport logistics, Airborne Self-Separation). The different agents, while pursuing their own specific goals, collaborate to optimize the system global efficiency. Multi-agent Systems are capable of generating robust solutions with cheap computational resources. The SGT gives instead the global efficiency criterion that the Multi-agent System has to optimize. It does not seek the optimal solution for all agents simultaneously, but the agents simply obtain an adequate solution, the most satisfying one. The adequacy of a choice is determined by comparing two different utility functions representing respectively the benefits and the costs of the considered choice. For the problem of airborne self-separation, benefits are related to the optimality of the possible trajectory while costs are proportional to risk of collisions with other manned or unmanned aircraft. In this way each aircraft defines its own preferences on the basis of other agents’ preferences, thus enabling a collaborative approach to the conflict resolution problem.

The Conflict Detection and Resolution (CD&R) algorithm developed for ARCA [9], in the following referred as
“ARCA algorithm”, can be classified according to the scheme presented in [5]. The first classification is between a centralized and a distributed conflict resolution strategy. To ensure autonomous operations the system developed for ARCA was based on a distributed approach in which each UAS determines its own trajectory to avoid a conflict. This approach can also be easily extended to a greater number of UAS involved and guarantees better performances in terms of overall delay and fuel consumption. Another key characteristic of the ARCA CD&R algorithm is the choice of a global resolution approach. This solution is based on the assumption that each aircraft exchange data with all other aircraft in its range. In the case of ARCA this is obtained through the use of ADS-B transponders [10] installed on every aircraft. The flight data exchange ensures that any arising conflict can be solved in a coordinate way among all the aircrafts involved. Another issue is related to the maneuvers that the aircraft will have to perform to obtain the required separation from the other aircrafts. These maneuvers have to be realistic and feasible and should take into account environmental factors such as weather conditions. In the ARCA algorithm a full 3-D trajectory model has been implemented, allowing each aircraft to perform maneuvers combining variations of speed, direction and altitude. The algorithm has been tested by running a software tool, which simulated the behavior of the SGT algorithm in a 3D environment. In a first batch of experiments UAS were considered in a segregated airspace and no data from ADS-B were used. These simulations in a synthetic environment contributed to test the algorithm performances in terms of speed and computational load. A second batch of experiments integrated the features of the first test campaign with ADS-B flight data from commercial aircrafts thus realizing a realistic simulation of a mixed environment with manned and unmanned flights. Finally hardware-in-the-loop simulations were carried out to test the integration and performances of the algorithm on the devices that would be actually installed on the UAS. In all three cases different scenarios were simulated by varying the number of UAS involved and the starting configuration: a choke point scenario with all flight distributed on a circle of radius 60 nautical miles and head directly through the circle center towards the opposite side and a random flight scenario where UAS were randomly flying in a square of side 120 nautical miles [9].

The main quality of the CD&R method developed for ARCA is the global approach to the conflict resolution problem. As a consequence, the separation between aircrafts is assured without the risk of generating other conflicts. Moreover the tests carried out demonstrated that the algorithm is efficient both from a computational point of view and in the conflict resolution process. On the other hand, the algorithm did not go under extensive testing and its capabilities in different scenarios have to be further assessed. Moreover the effects of the external environment on the aircraft trajectories were not taken into account leading to a not completely realistic simulation of the problem. Finally, given that the avoidance maneuvers are determined on the basis of an optimization process, the final solution cannot be estimated in advance. This prevents comparison of the obtained results with a predicted solution, thus it is not possible to forecast the final outcome of the detections and avoidance process.

In the following Section an application of the same algorithmic approach to the problem of Conflict Detection and Resolution for the General Aviation will be shown.

SeparA Project: supporting General Aviation pilots

The SeparA (Separation for general Aviation) project is an Italian national project funded by the Regione Lazio in the timeframe 2010-2012. The objective of this study was to develop a device to support General Aviation pilots in the process of conflict detection and resolution as well as in the flight routing, based on ADS-B [11]. The ARCA algorithm was here refined and adapted to the specific operational domain, while an HMI for mobile devices has been designed (Figure 1). In order to provide the pilot with traffic information and separation maneuvers suggestions. The tool has been validated both on the HMI side and the algorithm's logic side, with methods ranging from expert analysis to real time simulation, from feature inspection to logs analysis. The HMI has encountered strong appreciation from GA pilots, and the acceptability of the concept in the several GA communities (leisure pilots, private pilots, business pilots) has been rated as high. In the scope of this article, the main results to be presented are: the interface has been judged highly usable, the tool has been proven to enhance pilots' situation awareness both for general traffic picture and conflicts detection; the tool supports pilots' decision making process during conflicts; the behaviour of the algorithm has been considered realistic, and the solutions proposed are equivalent or equal to the ones the pilots would have performed in a real operational context, thus enhancing pilots' trust in the system; it was easy for pilots to interpret system's behaviours, understanding why the system was acting in a certain way. Finally no issues related to safety have been identified.

![Figure 1. A screenshot of the HMI of the SeparA tool](image-url)
A COMMON DESIGN APPROACH FOR GENERAL AVIATION AND REMOTELY PILOTED AIRCRAFT SYSTEMS

After investigating the Conflict Detection and Resolution methods and tools for two very different classes of aircraft (fully automated UAS and GA) we moved the focus of our research to RPAS.

Figure 2. The relationships among the selected projects

RPAS are considered by ICAO as the only class of UAS that will be inserted and integrated in the air traffic in the next years [1], since the Autonomous UAS (AUAS) are foreseen to be flown only as experimental and scientific flights in segregated airspaces. An RPAS is expected to be fully controlled by the remote pilot on the ground, who is in charge of all the flight operations. He/she can directly (manually) pilot the aircraft, typically by a stick control, or monitor the behaviour of an auto-pilot, keeping the ability to switch promptly to direct control in the case of malfunctions and/or unexpected events. The remote pilot is indeed responsible for avoiding collisions with other aircraft when flying in non-segregated airspaces by following ATC indications, when available, or by autonomously adopting appropriate separation strategies where traffic control service is not provided.

SARA (Sistema Avanzato per la sicuRezza Aerea di RPAS, Advanced System for Safety in RPAS) is an Italian project funded by FILAS (financing Agency of Regione Lazio) for the timeframe 2012-2014. The objective of the SARA project is to provide the remote pilot of an RPAS with an effective tool for supporting the traffic separation task. This tool has to be integrated with the remote station and shall provide information about surrounding traffic and effective (safe and suitable) suggestions for the pilot, presented as collision free trajectories on the screen. The previous experiences from the ARCA and SeparA projects are potentially very useful and a valuable basis to draw on when designing this new tool: the ARCA project provided us with an innovative and effective CD&R algorithm suitable for manned-unmanned mixed traffic scenarios, while SeparA project produced a decision support prototypal tool for pilots, even though in a GA domain (Figure 2).

In this paper we analyse the challenges and the opportunities associated to an idea of joint design for concepts at a different level of readiness (requirement collection phase vs. prototype), with different operational applications (RPAS vs. GA), different levels of automation and different users (remote pilots vs. GA pilots), but still sharing a set of common core requirements and needs (CD&R support for the pilot).

In the following the main opportunities and challenges brought by this approach will be introduced, then the problem of joint design will be analysed and modelled. Finally we will propose a novel methodology and a set of tools for supporting the process of joint design of new concepts.

OPPORTUNITIES AND CHALLENGES

Developing two systems (for RPAS and GA) sharing as much as possible characteristics between them, provides some interesting opportunities from different points of view.

From an economic point of view, for example, development costs could be reduced, as a common development phase can be shared by two systems that will be proposed to two different communities. The two systems also share the base technology (i.e. ADS-B) thus also reducing costs. Moreover, RPAS pilots could be recruited among GA pilots, with well developed and tested skills in using the system and in understanding the underlying philosophy and logic; this means less training needs.

Figure 3. RPAS and GA concepts' independent design
The validation process, both form a technical and a human factors point of view, can be performed by a wider community, and with a wider base of testers. This means more resilience and more consideration of usability and safety issues.

From the point of view of the researchers community, there is the opportunity to develop and test methods aimed at merging and integrating requirements coming from different domains and different development processes.

The main challenges we identified for the design of a common interoperable CD&R tool for GA and RPAS are due to the differences in terms of technologies, operations and human component (pilot) between the two systems (Figure 4).

The first challenge is related to the different level of automation required by a CD&R tool in the two domains. We decided to classify the automation level by using the Parasuraman et al. classification [12]. Although this classical model has been recently criticized for a presumed poor ability to describe all the complex and dynamic processes underlying the interaction between human and automated systems (see [13]), we decided to use it for two reasons. First of all it is a quite simple and at the same time expressive model for providing a picture of the analyzed system or tool in terms of automation. Second then Parasuraman classification is quite well known so providing a common and agreed basis for a fruitful collaboration among designer, users, experts and stakeholders in general, which is one of the main focuses of the here proposed methodology. In the following we provide a brief analysis of the different levels of automation for the two reference domains, GA and RPAS:

- In the VFR GA context the tool is used as a Decision Support System, i.e. the pilot receives from the system information about surrounding traffic and a suggestion for a “safe” trajectory to keep separation from it. The pilot can decide whether to follow the suggestion or not, since he is still able to keep separation by seeing outside the aircraft. Following the classification provided by Parasuraman et al., this scenario corresponds to Level 4 ("The computer suggests an alternative");

- In the RPAS context we have two different scenarios. If the aircraft is flown in Visual Line of Sight (VLOS) the situation is comparable to the GA case, since the pilot is still able to keep adequate separation by seeing the potential encounter(s). If the aircraft is flown BLOS the pilot can take decisions about separation maneuvers basing only on the information provided by the tool. The remote pilot can still decide whether following or not the suggested maneuver, but his/her “sensing” is strongly dependent from the information provided by the system and more trust has to be put in the system. In the RPAS case the pilot can also decide if the proposed separation maneuver has to be directly executed by the aircraft auto-pilot, so in this case the role of the pilot is to accept the suggestion and monitor its execution. We can also suppose a more automated scenario in which the CD&R tool identifies a potential separation infringement, it informs the pilot and proposes a resolution maneuver and it waits for the pilot intervention. If the pilot does not timely adopt a resolution strategy by manually performing the maneuver or by enabling the auto-pilot to perform it, the system acts automatically, accomplishing the maneuver. This behavior is suitable to address the non-nominal condition in which an high latency or a malfunction on the Command and Control Link between Remote Station and Remotely Piloted Aircraft prevents the pilot from promptly adopt a safe strategy. Here the Automation Level raises to 5 ("The computer executes the suggestion if the human approves") or eventually 6 ("The computer allows the human a restricted time to veto before automatic execution") in the more automated scenario.

![Image](https://via.placeholder.com/150)

**Figure 4.** RPAS and GA concept design, from reutilization of requirements to interoperable design
Another important consideration about the peculiarity of RPAS has to be made: while in a GA context (and in manned aviation in general) the pilot has an immediate feedback about the success of a maneuver, this is not obvious when the piloting is performed remotely through a Communication Link. In the case of a RPAS, the pilot performing a “safety” maneuver for assuring separation from traffic has to wait for the feedback and monitor it. This “latency” can be non-negligible especially in BLOS operation.

The presence of a Communication Link as part of the systems implies some considerations about the separation maneuvers suggested by the CD&R tool: it shall never consider solutions leading the aircraft to fly beyond the radio communication link, resulting in a loss of control from the pilot station. At the same time for GA flying in VFR mode, the tool shall never propose trajectories leading the aircraft in air spaces (or altitudes) where this mode is not allowed.

Basing on these considerations we can assume that the quantity and the content of the information provided by the CD&R tool for RPAS will be quite different from the one provided by the GA tool (for instance: radio range vs. precluded air spaces).

**SOLUTIONS**

Based on the User Centred Design (UCD) framework [14], we have defined solutions in the form of consolidated tools. These tools allow the generation of requirements, prototyping and evaluation in a methodology aimed at identifying, as much as possible, common requirements for two different systems, thus reducing the gap represented in Figure 5. This methodology is currently under validation within the SARA project.

The proposed methodology can be activated at several levels of development and different levels of maturity of the concepts. For example in the case presented in this article, the selected projects are at a quite different level of maturity: SepaR has reached V3 of E-OCVM [15] phases (Pre-Industrial development & Integration), while SARA is still at V1 (Scope): some requirements have been collected and are under validation. In order to apply the proposed methodology, it is necessary that the projects selected have not only a set of requirements, but also a sat that has been somehow validated.

Within the SARA project, it is expected that requirements will diverge as development evolves from the early stage (concepts, non interactive prototypes) to final development (interactive prototypes).

**The proposed methodology**

The envisaged methodology is articulated in five different phases:

1. **Collection of validated requirements from both domains:** coherently with UCD spirit, the proposed methodology starts with the analysis of users’ needs, focusing on the ones RPAS and GA pilots have in common (Figure 4). This analysis is based on selected projects’ results: requirements coming from previous projects are gathered;

2. **Filtering of the requirements:** it is necessary to identify which requirements can be considered of common interest. The proposed tool is a focus group with two operational experts coming from the two domains;

3. **Feasibility study:** identification is necessary for the feasibility of each requirement, from a technical and human performance perspective. The proposed tool is a technical meeting among Human Factors experts and Engineers;

4. **Prototyping:** the requirements are implemented into prototypes. The type of prototype is related to the maturity level of the concept, and can range from paper based mock-ups to interactive simulators;

5. **Evaluation:** the same prototype is evaluated in two different test (each for domain). The evaluation methods are related to the maturity level, and can range from Heuristic Analysis to User Tests.

![Figure 5. RPAS and GA common needs](image)

The focus group has been chosen as the preferred tool for phase 2 because, as shown in literature [16], it has been successfully used in past activities and proved to be suitable and effective for requirements selection and prioritization. The technical meeting is a more structured tool in which, for each requirement, the Human Factors and the Engineer evaluate the potential issues related to its implementation, both in terms of human performance (e.g. impact on workload) and technologies limits (e.g. lack of computational power).
Coherently with UCD framework, the process is iterative, and several cycles of Analysis of the activity, Prototyping and Evaluation can be performed.

CONCLUSION
In this paper we present a novel approach to the design of automated tools for supporting self-separation of different classes of airspace users. In particular we propose a methodology and tools for eliciting common requirements and developing potential solutions to challenges brought by the design of two different tools, with a common purpose (supporting separation assurance) but different operational environment (RPAS vs. GA), users (remote vs. GA pilots) and different levels of automation (5/6 vs. 4 in the Parasuraman et al. [12] categorization). The objective of the methodology introduced here is to identify as wide as possible a set of common requirements, by eliciting them with final users and validating them using consolidated tools (like focus group with experts). The identification of a common set of requirements enables the opportunity of design interoperable solutions, leading to clear benefits (lower costs, common training for pilots, wider user community for tests and improvements). The methodology is now being used in the development of SARA (RPAS) and SeparA (GA) tools, at different levels of maturity, so the application to these real cases will provide evidences and feedbacks about the validity of the proposed methodology. Our idea of interoperable design of tools by identification of common requirements can be applied to different application domains which share the need to develop tools with different characteristics but a clearly recognizable common purpose. In the future we intend to exploit new application fields where our methodology can be applied, widened, refined and improved upon.

REFERENCES
6. SUPEROPT Project Consortium (2010), SUPEROPT Website: http://www.hala-sesar.net/superopt
Enhancing Autonomy in Unmanned Aerial Vehicles. 
International Journal of Artificial Intelligence, Special 
Issue in Intelligent Unmanned Systems, 316-328.

10. Stamper, W. Understanding mode S technology. 

ADS-B Based Separation Support for General Aviation. 
Bercelona, Spain.


Autonomous Management of a UAV Airfield

Rosa M. Rodríguez
Boeing Research & Technology
Europe
Av. Sur del Aeropuerto de Barajas 38
28042 Madrid - Spain
rosamaria.rodriguez@boeing.com

Francisco Alarcón,
Daniel Santamaria Rubio
FADA-CATEC
C/ Wilbur y Orville Wright 17
41309 Seville (España)
falarcon@catec.aero
dsantamaria@catec.aero

Aníbal Ollero
Universidad de Sevilla.
Av. Camino de los Descubrimientos, s/n
41092 Seville (Spain)
aollero@cartuja.us.es

ABSTRACT
In this paper, a cooperating objects-based architecture for autonomous management of an Unmanned Aerial Vehicle (UAV) airfield is presented. Although UAVs are gaining autonomy, they still need human intervention prior to take off and landing. The implementation of the architecture described in this paper composed by a set of networked sensors and actors would allow the existence of a completely unattended airfield in which UAVs are remotely managed by end users. It relies on a completely autonomous operation from the mission planning to the mission executions and completion. The development of the platform that makes this possible is part of an on-going FP7 European project, PLANET, whose main components, functionalities and first field tests are presented in this paper.

Author Keywords
Unmanned Aerial Vehicle (UAV); Cooperating objects; sensors and actors networks; deployment; autonomous management;

ACM Classification Keywords

General Terms
Design; Experimentation;

INTRODUCTION
As the use of Unmanned Aerial Vehicles (UAV) is expanding from military to civil applications, we could expect its extension for commercial usage. One main problem that still remains to be solved is the co-existence of manned and unmanned aviation. Thus, many efforts have been devoted to the area of unmanned aviation. Unmanned AircraftSystems (UAS) have evolved in the last years providing more and more autonomy to the vehicle, i.e., the UAV. Such autonomy includes not only the capability to fly autonomously over specific areas but also, in some cases, the capability to take off and land without human intervention. However, most of the UAVs require important operator interaction before the take off or after landing.

The concept of a “Highly Automated Airfield” originates from the idea of increasing the autonomy of UAV operations from pre-flight on the ground segment, to take off and landing and post flight ground phases. The objective during those segments of the mission is to minimize the operator presence relying in a completely autonomous operation mostly unattended from the mission planning to the mission execution and completion.

In the context of the European Project PLANET – PLAtform for the deployment and operation of heterogeneous NETworked cooperating objects (FP7-2009-5-257649) [1] –, a software platform is being developed to support the operation of Cooperating Objects (COs) [2]. COs refer to embedded computing devices equipped with communication as well as sensing or actuation capabilities, able to cooperate and organize themselves autonomously into networks to achieve a common task [3]. As such, COs cover a wide range of devices. The PLANET concept is being validated in two complementary scenarios, the first case is the monitoring of the Doñana Biological Reserve located in Andalucia, Spain. [4] This environment has a high ecological value and is very sensitive to the impact of pollution. The second scenario is a Highly Automated Airfield in which security plays an important role and where wireless communication and cooperative techniques pose significant challenges. This paper refers to the second application. Next section introduces the PLANET framework and its main characteristics and functionalities. Then, several specific use case scenarios related to the autonomous operation in a UAV airfield are presented. The paper continues describing the first experimental tests performed the airfield: sensors used and communication network. The last section explains the results and conclusions of the first experimental tests.
THE PLANET FRAMEWORK

PLANET is an integrated platform that is being developed to enable the deployment, operation and maintenance of large-scale complex systems of heterogeneous networked cooperating objects, including wireless sensors network, actuator networks and mobile objects.

An overview of the architectural components (Figure 1) is described as follows. In the first step, the user introduces the characteristics of the target scenario: i.e. geographical data, parameters to measure. With these requirements, the visualization, modeling and simulation tool process the information and suggest an initial sensor deployment map in the given scenario. Through a planning and pre-deployment interface, PLANET produces the first suggestion for deployment task (number and type of sensors, deployment points, sequence of the deployment). This task is then validated and optimized before the real deployment is executed. The platform has also been designed with diagnosis, healing and management functionalities that monitor the deployment and operation status of the networked sensors and actors and produces the necessary actions to substitute or repair sensors in case of failure detection. These actions can include the deployment, if necessary of autonomous mobile vehicles to help in executing a given action.

![Figure 1: The PLANET Concept [1]](image1)

Apart from the software framework described before, cooperating objects (mobile and static sensor and actors) are the main components of the system. A sensor node is a static node that provides physical measurements and collaborates with other Cooperating Objects to perform data processing or data collection based on the application scenario. The network is complemented by the presence of mobile vehicles: Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicles (UAV). An unmanned vehicle plays multiple roles in PLANET. It can be used as a deployment tool [5] of sensors in specific locations, a mobile sensor node or a data mule performing different tasks. As a CO, the vehicle is also monitored by the system software and the user. Therefore, the vehicle reports its operational status at the same time that performs the instructions specified by the user during the runtime. The application of this framework to enable the autonomous management of a UAV airfield is described in the next sections.

HIGHLY AUTOMATED AIRFIELD

In the hypothetical scenario proposed in this paper, several different end-users with one or more UAS are presented. Those platforms are located in a highly automated airfield. Those users can be Biological or Environmental Research Groups, Fire Departments, Homeland Security units or Civil Protection authorities. The UAVs mission definition depends entirely on the remote users that program the mission of the UAV autopilot. Then the requested date and time slot are sent to the airfield which is represented by the Airfield Management System (AMS) (Figure 2). The end-users communicate the planned mission, e.g., flight path for collecting biological data in difficult access areas, or unplanned ones, e.g., urgent deployment to a new fire. The AMS responds by providing a set of services. These services include some pre-flight arrangements (i.e. refuelling/charging, de-icing, payload adjustment or calibration, etc.), take-off service (including the taxi and safety measures) and landing service (gate provision, blocks on).

![Figure 2: The Airfield and User Interaction](image2)

Once the mission request has arrived to the airfield, the networked COs managed by the PLANET framework allows the UAV autonomous operations in the airfield with none or minor human intervention. The Cooperating Objects are heterogeneous and include: sensors nodes capable of measuring airfield conditions (weather, radar, presence of obstacles on the runway, intrusion in airfield perimeter), and mobile vehicles: fixed wing UAVs, rotary wing UAVs and UGVs. These vehicles can have onboard sensors or serve as platforms that help in the deployment of other sensors. The networked coordination of the COs allows the autonomous execution of pre-flight and post-flight operations, as well as autonomous taking off, flying and landing. The PLANET framework does not perform mathematical data fusion. Data from different sensors are compared with databases containing the ranges of values that allow the operation (for example limits in wind speed...
or temperatures for UAVs). The AMS relies on a Petri net. Once the values received from the sensors are compared with the data base reference, a series of signals are generated to understand if the airfield changes its operational state. For example, if the airfield initial state is ‘operational’ and one of the sensors detects the presence of an obstacle on the runway, the airfield would move to an ‘emergency’ state. Mission would be canceled until the obstacle would be removed and the airfield would recover its ‘operational’ state. This is how the airfield provides resources (service areas, parking areas and runways) and services (weather forecasting, obstacle detection and removal services). The Airfield Management System (AMS) also schedules and shares the resources and services of multiple external users, and therefore UAVs. In order to ensure a successful completion of UAV operations, requirements on interactions between UAVs, autonomous airfield service COs, environment conditions and security/safety needed, should be clearly identified.

The Airfield Management System (AMS) analyzes the UAV’s capabilities, and accepts the services requested if it is possible to offer them. Once they are accepted, AMS provides to the UAS data and resources depending on the airfield equipment. The airfield is highly automated and may consequently be unattended; hence physical security measures must be implemented. These security measures must be based on automated systems which should guarantee secure and safe operations. For this reason, the airfield will have some devices to prevent, detect and deter possible intruders compromising the level of safety operations.

**PLANET architecture in a Highly Automated Airfield**

The PLANET framework comprises a set of architectural components, each of which contributes to the main functionalities of PLANET framework applied in the automation of an UAV airfield.

The *PLANET Control* is the most critical component in the network operation phase. It is responsible for monitoring the operational status of the deployed CO network and is required to ensure the correctness and continuity of all network operations. *PLANET Control* is in the centre of the PLANET framework architecture interacting with different components in different tasks. It interacts with another component of PLANET framework, *PLANET Platform* which is embedded in all the CO, to collect airfield data and to monitor the sensor network status. PLANET Platform is the interface between *PLANET Control* and COs. *PLANET Platform* gathers sensing data or status monitoring data of each CO, and provides this information to *PLANET Control*. This information is shown to the User through a third component of PLANET, the *PLANET Visualizer Command and Control (VCC)*.

**Cooperating Objects**

In PLANET, the term “Cooperating Objects (COs)” refers to collaborating entities that are involved in the PLANET work flow in order to perform automated operations in the UAV airfield. Examples of COs include the autonomous UAVs and UGVs, and a set of heterogeneous sensor nodes to be deployed or already deployed. In general, all COs are categorized into three types: Unmanned Ground Vehicles (UGVs), Unmanned Aerial Vehicles (UAVs) and Sensor Nodes.

Unmanned Ground Vehicles (UGVs): An UGV is an unmanned ground vehicle that can roll autonomously. In PLANET, an UGV can play one of three roles: an autonomous deployment tool, a mobile sink and a sensor node. When, the UGV acts as a deployment tool, it carries sensor nodes to the locations where the nodes need to be deployed; when acting as a mobile sink, the UGV acts like a data mule and navigates through the monitored area for data collection; finally, when, the UGV acts like a mobile sensor node, it moves to the assigned area and performs sensing tasks using equipped sensors.

Unmanned Aerial Vehicles (UAVs): An UAV (Figure 3) is an unmanned aerial vehicle that can fly autonomously. Based on the features and functionalities, UAVs used in PLANET can be categorized into two types: fixed-wing and rotary-wing UAVs. Like UGVs, a UAV can also play one of three roles: a deployment tool, a mobile node and a mobile sensor node. In the specific application of the automated airfield, only fixed wing UAVs will be used.

![Figure 3: Unmanned Aerial Vehicles](image)

Sensor Nodes: Sensor nodes are devices with sensing, computation and communication capabilities. Sensor nodes can provide physical measurements of phenomenon events, process the sensor data, and deliver related data about the monitor events. Note that, sensor nodes in PLANET are viewed as static nodes, which remain stationary once they are deployed.

**HIGHLY AUTOMATED AIRFIELD SCENARIOS**

To illustrate this highly automated airfield scenario and for PLANET framework demonstration purposes, several specific use cases have been designed:

- Automated Mission Service
- Perimeter Security Service
- Sensor Healing Service
The first one, Automated Mission Service, provides the automated airfield with the basics settings required for UAVs to perform takeoff and landing operations; considering the availability of resources, services and safety concerns (weather conditions, free runway). The second one represents the automatic response of the airfield when, prior to a take-off or landing, an incident related to an unauthorized intrusion into the airfield perimeter is detected. In the third case, a malfunction of one of the sensors is reported and the airfield activates the deployment of mobile vehicles to repair or substitute the damaged sensor before continuing with the scheduled missions operation.

**Scenario 1 Automated Mission Service**

This use case describes the fundamental concept of the Highly Automated Airfield where all possible automated services are provided to allow autonomous take-off and landing operations of UAVs. An end user (the UAS Operator) that can be remotely located introduces a mission to the Airfield Management System (AMS) with scheduled takeoff and landing date and time. A concrete UAV with a specific configuration (primary type of payload) must be also specified by the user. Upon receiving the request from the user, the Airfield Management System plans a series of operational procedures with automated services and resources to allow autonomous completion of the mission. This means that the AMS must maintain total situational awareness of all ongoing activities of the airfield, autonomous vehicles, automated service operations as well as environmental conditions such as weather parameters. Apart from weather conditions, in both taking off and landing actions, an important task is to detect the presence of obstacles (animals, static objects) on the runway and to clear the obstacle if any is detected. Thus, the Airfield Management System provides an automated obstacle detection and removal system. Such system is composed of a set of sensors monitoring the paths for the UAV operations. The system will notify to AMS the existence of obstacles. If the sensors report such obstacles, external help is required to remove them. As long as an obstacle blocks a certain area on the airfield, UAVs will not be allowed to use these areas for starting or landing operations. To ensure the safety of running operations, the AMS should be able to interrupt every operation at any time. This also includes the suspension of missions for safety reasons. If the mission is allowed, and there is a safe take off, this first use case considers also the process of safe landing and proper return to parking lot.

**Scenario 2 Perimeter Security Services**

The safe operation of UAVs in the airfield can only take place when levels of security conditions are met. In this sense, the airfield is considered a critical infrastructure and as soon as a security breach is detected, operations must be stopped. In this scenario, the airfield has a security perimeter protection system which detects and tracks intruders inside the airfield perimeter. If an intrusion is reported to the AMS, mobile and fix sensors are activated to provide a close range vision of the unknown threat. The airfield security perimeter protection system takes the coordinates of the incident location with the mobile cooperating objects which are able to reach the incident location. They obtain the image and send it back to the system so the threat can be cleared or confirmed. Once the incident is clarified and resolved, the AMS can continue with the regular and scheduled take-off and/or landing operations.

**Scenario 3 Sensor Healing Services**

This scenario addresses the requirements for the maintenance of the sensors that allow the autonomous operation of the airfield. On the automated airfield, a large-scale of collaborative, networked sensor nodes need to be maintained. The integrity and operational correctness of the system must guarantee safe operations. A first requirement on the airfield is then the capability of substituting any malfunctioning sensor by a temporary one while the permanent sensor is repaired. Moreover, this substitution should be performed autonomously using unmanned mobile vehicles such as UGVs. In order to implement this scenario, the airfield infrastructure must incorporate the following sensors and functionalities: autonomous checking and calibration procedures to ensure that the entire airfield infrastructure is working properly and mobile systems to replace malfunctioning or compromised sensors. As in the previous scenario, once the maintenance operation is finished, the AMS can continue with the regular and scheduled take-off and/or landing operations.

**EXPERIMENTAL TESTS**

To validate the PLANET framework several field tests were performed in a real environment, at the aerodrome of Villamartin in Cádiz (Spain). The aim was testing the maturity of the first version of PLANET and its application in the management of an UAV airfield. The scenario chosen in these experiments was the first use case of Automated Mission Service described previously. This section deals with the description of the networked sensors, mobile vehicles and the type of communication used in the experiment.

**Cooperating objects in an highly automated airfield**

Key elements for the automation of the airfield are the PLANET framework computing infrastructure that facilitates the cooperation between sensors and actors, and the Airfield Management System computing infrastructure that receives the mission plan and starts the coordination with the PLANET framework. Regarding sensors nodes and actors, the equipment deployed in a highly automated airfield are the following:

- Weather Sensor System
- Obstacle Detection System
Radar System
- Intrusion Detection System
- UAS operating in the airfield
- UGV’s for obstacle removal and intrusion identification

Weather Sensor System
Unmanned Aerial Vehicles are very sensitive to weather conditions; therefore a constant control of wind speed, temperature and other weather parameters in the airfield is essential. In the first experiments, a real airfield meteorological station has been emulated with a meteorological portable station that consists of a handheld sensor: the Kestrel 4500. This sensor is capable of monitoring and reporting an exhaustive list of environmental parameters and store and chart up to 1400 data points for analysis. The list of the conditions and main characteristics of this sensor is the following: Heading (true & magnetic), Wind direction, Crosswind, Headwind/tailwind, Altitude, Pressure trend, Barometric pressure, Wet bulb temperature, Relative humidity in %, Heat stress index, Dewpoint, Wet bulb temperature, Density altitude, Wind chill. Depending on the UAV type and size, and the environmental condition measured, AMS can allow or reject the UAV mission requested by the external user.

Obstacle Detection System
The main element of the Obstacle Detection System is a LIDAR LD-MRS-400001 connected to a PC-104 computer. The LIDAR serves for directional contact-free detection on the sensor surroundings, basically of the objects located within the radial field of view. The object detection is done with laser beams emitted in four stacked planes. The device measures the distance and the direction (the angle to the sensor) of the object. From the measured data, the LD-MRS calculates the position of the object in the sensor coordinate system. In this way, this sensor will work in two ways, first as an obstacle detector and also as a movement sensor.

Radar System
If the Obstacle Detection System previously described provides a good surveillance in short distances, a HARRIER Air Surveillance Radar provides a high reliable wide area surveillance support for UAV operations, with detection and tracking for civil aircraft up to ranges of 20-28 miles. The radar would be used in airspace surveillance area of 30x30 km around the airfield facility so as to provide reliable “sense-and-avoid” operations support, to make sure that aircraft are not in the UAV operating path and to prevent the UAV from flying into them.

This system provides full surveillance coverage up to 30 miles away and from ground level to altitudes up to 20000 feet. The data of this radar is displayed in real time with high system update rates less than 1.2 seconds. This device is fully networkable via secure TCP/IP and other protocols e.g. via fiber optic, Ethernet, wireless, cellular and satellite systems. The data display is available in real time for multiple remote user displays. This radar also includes the ability to define pre-set site perimeters and zones, providing automated alert notification when an intruder is detected crossing or entering the zones. All system parameters can be fully remote controllable.

Intrusion Detection System
The Intrusion Detection System provided in this scenario is based on SELEX Sistemi Integrati solution for situational awareness and early warning. This product called MasterZone is based on unattended ground sensors technology. It supports the surveillance of target areas and detects hazards in different operational scenarios. The sensors can be deployed in large quantity to cover a wide area. They detect movements, sound, magnetic fields, terrain vibrations and routinely report field information to a central monitoring station. The Monitoring Station performs data acquisition from the network, as well as data processing of detected events and alarms and sends them to the AMS.

Unmanned Aerial Vehicle
The UAV used is an X-Vision. It has 8Kg payload capacity and an endurance of 3 hours. It is a fixed-wing UAV with a combustion engine and with large space to integrate new systems and processing units. It is completely autonomous (Piccolo autopilot and Ground Control Station from Cloudcap) with a maximum range of 40Km (limited by the datalink). This vehicle needs a runway of at least 200m for take-off and landing. One important characteristic of this UAV is the fact that it is able to perform taxi operations. The UAV has integrated a mode A/C transponder that also has ADS-B capabilities. The default payload is a gimbal with visual and InfraRed cameras. However, more payloads could be integrated in the UAV. It can be operated
autonomously from the Ground Control Station, which will be connected with PLANET platform in order to communicate with the rest of the PLANET COs (WSN, UGVs, etc.).

Unmanned Ground Vehicle
The Unmanned Ground Vehicle used in PLANET project is a 4x4-wheel Mobile robotic platform, designed for indoor and outdoor operation when a higher ground clearance and fast maneuverability is required. Its wheels are rugged, with alight weight20Kg vehicle, with a maximum velocity of 6.5km/hr, with high ground clearance (88mm), compact design and weather and water resistant. The UGV dimensions are: Height: 255mm; Width: 530mm; Length: 570mm; Weight: 19.5Kg. The platform is driven by four powerful (80W, 65Kg.cm/wheel) motors, one for each wheel and have a payload of 30Kg, with a dragging payload of max 50Kg. It is designed for tough terrains and is capable of running over vertical step up to 155mm and climbing up low rise stairs (up to 110mm step). The UGV is wirelessly (802.11G) interconnected. It integrates outdoor GPS and onboard sensors, such as Gyro Accelerometer Compass for autonomous navigation, temperature sensing and voltage monitoring. The integrated high resolution video/audio provides remote operator detail information of the area surrounding.

Communication Network
In a Highly Automated Airfield Scenario, a number of wireless and wired networks are needed to support the equipment and to allow the communications. The wired connections will be mainly based on standard Ethernet and a few USB based links. For the wireless connections most links are based on standard Wi-Fi and specific radio modems are used for special data links like UAV’s control or video stream. A scheme containing all the connections is shown in Figure 5

Depending on the geographical deployment, different routers, access points and antennas will be used. Most of the connections are made through WLAN (2.4 Ghz), except the 900MHz datalink between UAV and its ground control station. This device has a 400 MHz link between the unattended ground sensors and its central station and a 1.3 GHz link to send video images from the onboard camera to the UAV ground control station.

The communications networks are the following:
- The Airfield Management System and the UAS need to communicate:
  - UAV position
  - Routes for the UAV
  - Clearance commands
  - Abort commands
- The Airfield Management System and the Obstacle Detection System need to communicate:
  - Obstacle presence and location estimation
- The Airfield Management System and the UGV for obstacle removal need to communicate:
  - Obstacle location
  - UGV position
- The Airfield Management System and the weather sensors need to communicate:
  - Weather conditions

In these communications, the framework PLANET plays a key role since it provides the means for a collaborative network between COs and AMS. In Figure 6, the network infrastructure and the systems that were involved in the experiments is shown.

Figure 5: Physical Architecture and Connectivity.

Figure 6: Network Infrastructure.

PLANET Control collects the information from the PLANET Platform embedded in every COs and monitors their operational status. This information is sent to AMS.
that interprets all gathered data and commands the needed actions: i.e. removal of obstacles, allow or abort airfield operation.

**Experiment description**

The field experiments for the HAA took place the first week of July 2012 in the Villamartin Aerodrome (Cádiz – Spain) to test the PLANET framework and all the systems that were integrated. A simplified version of Scenario 1 was chosen for these first experiments. No radar was integrated in these first experiments. The deployed sensors and mobile vehicles had to be capable of checking that both weather and runway were in optimal condition to allow the take off of a planned UAV mission.

The systems and roles of these systems were:

- **AMS**: Airfield Management System for managing the airfield and commanding the different actions to the COs.
- **Weather Station (WS)**: measure of meteorological conditions.
- **LIDAR**: detects the presence and location of obstacles in the runway.
- **UGV**: Unmanned Ground Vehicle for removing the obstacles which is managed through its Ground Control Station (UGV-GCS)
- **UAV**: Unmanned Aerial Vehicle. As well as the UGV, it is controlled through its Ground Control Station
- **CTL**: PLANET Control. Component of the PLANET framework in charge of gathering data from COs
- **VCC**: Visualization Command Control. Component of the PLANET framework for visualization.

For the communication between the different systems, a WIFI antenna was used to provide a local network (Figure 6). In the current implementation of the PLANET Platform and PLANET Control, all COs communicate using Data Distribution System (DDS) channels.

For the experiment, a scenario was prepared in order to evaluate the safety of take off and landing operation in the runway. The scenario was separated in some steps:

1. An UAV is required from a user for a specific mission.
2. Initialization procedure where all the components register in the PLANET Control and they are configured.
3. An object is dropped in the runway.
4. The Obstacle Detection System using the LIDAR detects the object and sends this information to PLANET Control and this module addresses this information to the AMS.
5. AMS analyzes the message of the LIDAR and commands to a UGV to remove the object. The UGV begins the mission and the take off command is blocked.
6. When the UGV reaches the waypoint, the object is autonomously removed and the AMS unlocks the alert in the runway, so it can work in a normal operation state.
7. AMS checks again the information of the Weather station and LIDAR. When everything is correct, it commands the required UAV to take off.
8. UAV takes off and sends telemetry to the AMS and the VCC.

In this particular experiment, the AMS is the component that gathers information and takes decisions based on the logic programmed inside. However, more complex situations described in the other scenarios require the cooperation of CO without the intervention of the AMS. The DDS communications make this possible. Every time a CO wants to be part of the PLANET framework, it must register in the system. If the CO needs to know some information about another CO, it must be registered to that specific CO. Hence, every time this second CO publishes some information/data, the other interested CO would automatically receive it. A logic implemented in the sensors allows them to act in the scenario if necessary. For example, sensor damage would interact with UGV for sensor replacement; in this case the AMS is aware of the situations and stops operation in the airfield if necessary.

Regarding the robustness of the system, the fact of using DDS has some advantages. In case of a failure of the AMS, PLANET Control would notice it and would communicate to the rest of the CO that the airfield is not available. This is possible because PLANET Control knows all messages exchanged in the system and the state of each component.

**RESULTS OF THE EXPERIMENTS**

With respect to the communication between AMS, COs and PLANET framework, the experiments revealed that there is a large amount of information that needs to be exchanged between components so new typology of messages had to be defined in addition to those initially implemented. Regarding the registration procedure, it was noticed the importance of the configuration of the different elements. This allows the AMS and the PLANET Control to know how many COs and elements were involved in the airfield. In addition, new sensors and systems can be incorporated in any moment during the operation. During the experiments, it was also necessary to redesign the logic associated to the Petri net implemented inside the AMS. On the other hand, the DDS middleware facilitates the data communication.
and allows interactive changes in the network. DDS was a good choice for the communication infrastructure.

When an object was dropped in the runway (Figure 7) the Obstacle Detection System that uses the LIDAR sensors was able to detect the obstacle, so it was tested that the behavior of the detections algorithms worked properly.

When the LIDAR detected the obstacle, the AMS blocked the take off command and it was only possible to command an UGV to remove the object. For future experiments, in case of having small objects to be removed, it is important to take into account the GPS accuracy. This operation is meant to guarantee the safety conditions in the airfield. Once the obstacle was removed, the AMS returned to the normal state of operation. Then, the take-off request for the XVision UAV initiates. The AMS checked that the weather conditions were OK and also the state of the runway and then it commanded the take-off to the UAV. This can be seen in the Figure 8.

![Figure 7: Removal Process](image1)

![Figure 8: Take off](image2)

In the experiments the PLANET Visualization Command and Control module (VCC) was also tested. It was possible to see how the VCC was receiving constantly the UAV telemetry: latitude, longitude, altitude, GPS time stamp, velocity North, East and Down, Acceleration X,Y,Z, Euler Angles (roll, pitch, yaw), gyroscopes, angular rate (roll, pitch and yaw) and barometric altitude and indicated airspeed. Figure 9 shows the VCC receiving data:

![Figure 9: VCC receiving data.](image3)

**CONCLUSIONS**

The first field experiments have shown that the idea of autonomous management of an UAV airfield seems possible. The framework PLANET, an architecture based on networked cooperating objects, allows the interaction of heterogeneous cooperating objects to provide the services expected from an autonomous airfield. The communications using DDS worked properly and each system was able to connect with each object involved in PLANET. Also, the different modules and its algorithms allow separating the different functionalities of the airport in more specialized systems that use the communication network for publishing the data; in this way, the modules in charge of the management of the airport were able to fuse the different data in relevant information for command the appropriated actions to the different cooperating objects. In this way the airport is modularized and it is possible to add new sensors and functionalities for future implementations and needs. A first demonstration scenario has been shown in this paper. Future experiments will also tests more developed and complex situations involving a larger network of sensors and actors. The increasing complexity of future experiments aims at understanding the feasibility of implementation of the Highly Automated Airfield idea in a real scenario.

**ACKNOWLEDGEMENTS**

This work has been partially supported by PLANET, PLAtform for the Deployment and Operation of Heterogeneous NETworked Cooperating Objects, funded by the European Commission under FP7 with contract number FP7-2009-5-257649 (www.planet-ict.eu).

**REFERENCES**

1. FP7 Project PLANET webpage www.planet-ict.eu
Aviation Safety: Modeling and Analyzing Complex Interactions between Humans and Automated Systems

Neha Rungta, Guillaume Brat, William J. Clancey, Charlotte Linde, Franco Raimondi†, Chin Seah, Michael Shafto

NASA Ames Research Center, Moffett Field, CA 94035, USA
\{neha.s.rungta,guillaume.p.brat,william.j.clancey,charlotte.linde,chin.h.seah,mike.shafto\}@nasa.gov
†Middlesex University, London, UK, f.raimondi@mdx.ac.uk

ABSTRACT

The on-going transformation from the current US Air Traffic System (ATS) to the Next Generation Air Traffic System (NexGen) will force the introduction of new automated systems and most likely will cause automation to migrate from ground to air. This will yield new function allocations between humans and automation and therefore change the roles and responsibilities in the ATS. Yet, safety in NexGen is required to be at least as good as in the current system. We therefore need techniques to evaluate the safety of the interactions between humans and automation. We think that current human factor studies and simulation-based techniques will fall short in front of the ATS complexity, and that we need to add more automated techniques to simulations, such as model checking, which offers exhaustive coverage of the non-deterministic behaviors in nominal and off-nominal scenarios. In this work, we present a verification approach based both on simulations and on model checking for evaluating the roles and responsibilities of humans and automation. Models are created using Brahms (a multi-agent framework) and we show that the traditional Brahms simulations can be integrated with automated exploration techniques based on model checking, thus offering a complete exploration of the behavioral space of the scenario. Our formal analysis supports the notion of beliefs and probabilities to reason about human behavior. We demonstrate the technique with the Überlingen accident since it exemplifies authority problems when receiving conflicting advices from human and automated systems.

Author Keywords
Aviation Safety, Authority and Autonomy, Brahms, Verification, Überlingen collision.

ACM Classification Keywords
H.1.2 Models and Principles: User/Machine Systems; D.2.4 Software/Program Verification: Model Checking

INTRODUCTION

Over the past few years, the US has embarked in a transformation of the Air Transportation System (ATS) to address the expected increase of air traffic in the US. The original prediction is that the traffic in 2025 will be between two and three times greater than the current traffic. There is no consensus on the actual size, but everybody agrees that the US needs to modernize the ATS and to implement NexGen (Next Generation Air Transportation System) to accommodate the traffic increase over the next 15 years; Europe is going through a similar effort with SESAR. An important goal of NexGen is to increase efficiency without compromising safety. The implementation of NexGen, described in the Integrated Working Plan (IWP), will see the introduction of new automated systems (e.g., ADS-B, GPS-based navigation) and new air traffic paradigms (e.g., 4D trajectory), which will cause some Air Traffic Management (ATM) functions to migrate from ground to on-board, and possibly vice versa. The new automation will cause a change in function allocation as well as a change in roles and responsibilities for air traffic controllers and pilots. As a consequence, it poses new challenges in assessing the safety of the overall system. This is the focus of our work.

The US National Airspace System (NAS) is currently quite safe and accidents are at a record low. NexGen needs to provide at least the same, if not a better, level of safety. The NexGen IWP has a requirement (R-1440) that calls for new and improved verification and validation (V&V) techniques for complex systems. This is often understood as applying only to the software systems that will be used in NexGen. However, we also need to recognize that NexGen is a complex system in which humans and autonomy (or automation: we do not need to make a distinction in this work) are interacting in quite subtle ways. Therefore, we also need new safety evaluation techniques to verify and validate the interactions of humans and autonomy in the complex system that is NexGen. Moreover, there is a large consensus that the earlier in the lifecycle this V&V is done, the easier it is to detect and fix errors [5]. In our work, we are focusing on developing methods for evaluating early, in the design phase, models of complex interactions in which there are multiple, different, simultaneous, situation-dependent assignments of
authority and autonomy (A&A) among humans and automation. In order to ensure safety in NexGen, there is a need for well-defined formalizations of procedures, possible actions of the actors involved, and the consequences of the actions. The formalization should facilitate the analysis of various interactions between the actors; the analysis can either be simulations or formal search-based techniques such as model checking. We specifically need to develop methods that allow us to articulate the authority bounds (limits) and behavior in terms of ownership: who has authority in any situation and how it may affect safety. The definitions that we adopt simplified definitions for authority and autonomy in this work are as follows:

- **Authority** refers to having the right, or power, to exercise controls or issue air traffic commands that impact the position, velocity, and/or attitude of aircraft during operations.

- **Autonomy** (or automation) refers to a function or system that can operate independently of pilot or air traffic controller intervention.

The ATS, especially with future NexGen concepts of operation, is a complex system involving dynamic interactions among multiple actors that are largely governed through formal assignment of roles and responsibilities. These A&A assignments are made at the design level, but are executed at the operational level according to each actor’s view of its roles and responsibilities. Operationally, the system continuously adjusts for shortcomings in the assignment of authority and autonomy, for shortcomings in the capacity of actors to perform their assigned roles and responsibilities, and to optimize various performance factors such as capacity, environmental impact, and safety. This suggests that system safety should be derived not only from a predictable execution of assigned roles and responsibilities but also from checks and balances to ensure that the system operates as designed in the face of failures, disturbances and degradations. The ability of the system to operate in off-nominal conditions as a result of the checks and balances extent in it provides resilience, a critical characteristic for system safety.

Assessing safety in human-automation systems can be done using several techniques. Historically, human-in-the-loop studies have been the most prominent ones [4, 20]. They are quite costly to perform (they are real-time studies in which humans interact with ATS simulations), somewhat limited in scale (it is difficult to pull many controllers into a study) and often incomplete in the sense that they can explore only a restrictive set of behaviors. A few user interface rules have been extracted from these studies and can be used as building blocks to design the system interfaces. However, these rules fall short in helping in the context of a highly dynamic and complex system such as the ATS. Another way of analyzing human-machine systems is to create models for the humans (often based on the procedures that need to be performed by the humans) and run simulations [15, 18, 21] with the shortcoming that simulations can only examine a restricted set of behaviors. There has also been a growing interest and research in using formal methods for assessing safety in human-automation systems, particularly in the aviation domain. They have the potential of exploring all possible behaviors given an sufficiently complex model for the human and the systems. Early examples of the use of model checking for analyzing human-machine interactions are described in [8, 13, 22]. More recent examples try to bridge the gap between simulations and the use of model checking [3, 6, 7]. The analysis method is model checking but the representation of the problem (i.e., models for the human and the automation) uses simulation languages instead of fairly simple finite state models. These techniques can even be expanded to the design and the verification of aerospace systems [2]. Our work falls in the category of using both simulation languages and formal methods. Our innovation resides in using a simulation language defined for representing multi-agent systems, which is what the ATS really is: a complex system of interacting agents some of which are humans and some of which are automated systems. But we also integrate the simulation language with formal verification techniques based on model checking.

Concretely, we model systems in the Brahms multi-agent framework [11, 24]. Brahms is a multi-agent simulation system in which people, tools, facilities, vehicles, and geography are modeled explicitly. The air transportation system is modeled as a collection of distributed, interactive subsystems such as airports, air-traffic control towers and personnel, aircraft, automated flight systems and air-traffic tools, instruments, and flight crew. Each subsystem, whether a person or a tool such as the radar, is modeled independently with properties and contextual behaviors. Brahms facilitates modeling various configurable realistic scenarios that allows the analysis of the airspace in various conditions and reassignment of roles and responsibilities among human and automation. We then apply formal methods to the proposed concepts and configurations early in the development process to identify promising candidates for safe solutions, as well as find design problems when they are easier to fix. This combination of modeling and formal methods will increase assurance of safety and motivate adoption of advanced automation and associated operations protocols. To motivate our approach we present a generalized air transportation system model based on the Überlingen collision.

The rest of the paper is organized as follows: to motivate our work we first describe the conditions that led to the Überlingen collision. Then, we describe how humans and automation, as well as their interactions, are modeled. Finally, we then present simulation results and a description the verification framework, and discuss related work.

ÜBERLINGEN COLLISION OVERVIEW

The Überlingen accident [1], involving the (automated) Traffic Collision Avoidance System (TCAS), is viewed as a very good representative example illustrating the problem of authority versus autonomy (A&A) [10]. The Überlingen collision is a paradigmatic example of A&A conflicts. In particular, TCAS has the ability to reconfigure the pilot and air traffic control center (ATCC) relationship, taking authority from the air traffic control officer (ATCO) and instructing the pilot.
TCAS

TCAS is an onboard aircraft system that uses radar transponder signals to operate independently of ground-based equipment to provide advice to the pilot about conflicting aircraft that are equipped with the same transponder/TCAS equipment. The history of TCAS dates at least to the late 1950s. Motivated by a number of mid-air collisions over three decades, the United States Federal Aviation Administration (FAA) initiated the TCAS program in 1981. The system in use over Überlingen in 2002 was TCAS II v.7, which had been installed by US carriers since 1994.

TCAS II issues the following types of aural annunciations:

- Traffic advisory (TA)
- Resolution advisory (RA)
- Clear of conflict

When a TA is issued, pilots are instructed to initiate a visual search, if possible, for the traffic causing the TA. In the cases when the traffic can be visually acquired, pilots are instructed to maintain visual separation from the traffic. When an RA is issued, pilots are expected to respond immediately to the RA unless doing so would jeopardize the safe operation of the flight. The separation timing, called TAU, provides the RA unless doing so would jeopardize the safe operation of the flight. The separation timing, called TAU, provides the RA unless doing so would jeopardize the safe operation of the flight. The separation timing, called TAU, provides the RA unless doing so would jeopardize the safe operation of the flight. The separation timing, called TAU, provides the RA unless doing so would jeopardize the safe operation of the flight.

Überlingen Collision Narrative

On July 1, 2002, a midair collision between a Tupolev Tu-154M passenger jet travelling from Moscow to Barcelona, and a Boeing 757-23APF DHL cargo jet manned by two pilots, travelling from Bergamo to Brussels, occurred at 23:35 UTC over the town of Überlingen in southern Germany. The two flights were on a collision course. TCAS issued first a Traffic Advisory (TA) and then a Resolution Advisory (RA) for each plane. Just before TCAS RA to the Tupolev to climb, the air traffic controller in charge of the sector issued a command to descend, which the crew obeyed. Since TCAS had issued a Resolution Advisory to the Boeing crew to descend that they immediately followed, both planes were descending when they collided.

The decision of the Tupolev crew to follow the ATC’s instructions rather than TCAS was the immediate cause of the accident. The regulations for the use of TCAS state that in the case of conflicting instructions from TCAS and ATCO, the pilot should follow the TCAS instructions. The conflict in the Überlingen scenario represents the conflict between the authority of automated systems (TCAS) and people (crews and ATC), as well as their autonomy (freedom to act independently). The reason this conflict came into being is because the loss of separation between the two planes was not detected or corrected by the ATCO. The loss of separation between airplanes are frequent occurrences; it is part of the normal work of air traffic control to detect and correct them accordingly.

There were a set of complex systemic problems at the Zurich air traffic control station that caused the ATCO to miss detecting the loss of separation between the two planes. Although two controllers were supposed to be on duty, one of the two was resting in the lounge: a common and accepted practice during the lower workload portion of night shift. On this particular evening, a scheduled maintenance procedure was being carried out on the main radar system, which meant that the controller had to use a less capable air traffic tracking system. The maintenance work also disconnected the phone system, which made it impossible for other air traffic control centers in the area to alert the Zurich controller to the problem. Finally, the controllers workload was increased by a late arriving plane. An A320 that was landing in Friedrichshafen required the ATCO’s attention, who then failed to notice the potential separation infringement of the two planes.

The Überlingen collision proves that methods used for certifying TCAS II v7.0 did not adequately consider human-automation interactions. In particular, the certification method treated TCAS as if it were flight system automation, that is, a system that automatically controls the flight of the aircraft. Instead, TCAS is a system that tells pilot how to maneuver the aircraft, an instruction that implicitly removes and/or overrides the ATCs authority. Worldwide deployment of TCAS II v7.1 was still in process in 2012, a decade after the Überlingen collision.

MODELING THE ÜBERLINGEN WORK SYSTEM

Overview of Brahms

Brahms is a full-fledged multi-agent, rule-based, activity programming language. It is based on a theory of work practice and situated cognition [11, 24]. The Brahms language allows for the representation of situated activities of agents in a geographical model of the world. Situated activities are actions performed by the agent in some physical and social context for a specified period of time [9]. The execution of actions is constrained (a) locally: by the reasoning capabilities of an agent and (b) globally by the agents beliefs of the external world, such as where the agent is located, the state of the world at that location and elsewhere, located artifacts, activities of other agents, and communication with other agents or artifacts. The objective of Brahms is to represent the interaction between people, off-task behaviors, multi-tasking, interrupted and resumed activities, informal interactions and knowledge, while being located in some environment representative of the real world.

The Brahms agent language can also be used to develop executable software agents that are based on models of situated behavior. This allows for the development of intelligent agents that can act and react to specific situations that occur during its execution, and that have been modeled as the agent’s activity-behavior.

At each clock tick the Brahms simulation engine inspects the model to update the state of the world, which includes all of the agents and all of the objects in the simulated world. Agents and objects have states (factual properties) and may have capabilities to model the world (e.g., a radar’s display is modeled as beliefs, which are representations of the state of the aircraft). Agents and objects communicate with each other; the communications can represent verbal speech, reading, writing, etc. and may involve devices such as telephones,
radios, displays, etc. Agents and objects may act to change their own state, beliefs, or other facts about the world.

**Constructs in the Brahms Überlingen Model**

In a Brahms model, the system being modeled is the entire work system, including agents, groups to which they belong, facilities (buildings, rooms, offices, spaces in vehicles), tools (e.g., radio, radar display/workstation, telephone, vehicles), representational objects (e.g., a phone book, a control strip), and automated subsystems (e.g., TCAS), all located in an abstracted geography represented as areas and paths. Thus the notion of human-system interaction in Brahms terms is more precisely an interaction between an agent and a subsystem in the model; both are behaving within the work system.

A workframe in Brahms can model the interaction between an agent’s beliefs, perception, and action in a dynamic environment, for example, these characteristics are leveraged when modeling how a pilot deploys the aircraft landing gear. A pilot uses the on-board landing control and then confirms that the landing gears are deployed while monitoring the aircraft’s trajectory on the Primary Flight Display. This is modeled in Brahms as follows: a pilot (e.g., the DHL pilot) is a member of the PilotGroup, which has a composite activity for managing aircraft energy configuration. For further details about how the different Brahms constructs are used to model the various aspects of the Überlingen collision we refer the reader to our technical report [10].

A specific instance of a conceptual class is called a conceptual object. A particular flight (e.g., DHX611, a conceptual object) is operated by a particular airline and consists of a particular crew (a group) of pilots (agents) who file a particular flight plan document (an object), and so on. Each instance of an agent and object have possible actions defined by workframes where each workframe contains a set of activities that are ordered and often prioritized. Certain workframes are inherited from their group (for agents) or class (for objects). The set of possible actions are modeled at a general level and all members of a group/class have similar capabilities (represented as activities, workframes, and thoughtframes); however, at any time during the simulation, agent and object behaviors, beliefs, and facts about them will vary depending on their initial beliefs/facts and the environment with which they are interacting. The model incorporates organizational and regulatory aspects implicitly, manifest by how work practices relate roles, tools, and facilities.

A Brahms simulation model configuration consists of the modeled geography, agents, and objects, as well as their initial facts and beliefs of agents and objects. The different configurations allow us to perform a what-if analysis on the model. The time of departure for a flight might be an initial fact in a Brahms model. One can modify the model to assign a different time of departure for a flight in each simulation run. Another example of configurable initial facts may include work schedules for air traffic controllers. In one configuration of the work schedules an air traffic controller may...
be working alone in the ATCC, while in another configuration, two controllers would be present in the ATCC. Initial beliefs of an agent might be broad preferences affecting behavior (e.g., TCAS should overrule the ATC), thus initial beliefs can be used as switches to easily specify alternative configurations of interest. Alternative configurations are conventionally called scenarios. Thus for example, a scenario might be a variation of the Überlingen collision in which two aircraft have inter-route flight times that put them on an intersecting path over Überlingen; the only other flight is a late arriving flight for Friedrichshafen and maintenance degrades the radar, but the telephones are operative.

In general, a model is designed by the model builder with sufficient flexibility to allow investigating scenarios of interest. The set of causal factors of interest (e.g., use of control strips when approving aircraft altitude changes, availability of telephones) constitute states of the world and behaviors that can be configured through initial facts and beliefs. The initial settings define a space of scenarios. Using Brahms to evaluate designs within this space, while using formal methods to help modelers understand its boundaries so they can refine the model to explore alternative scenarios, constitutes the main research objective of this work.

The simulation engine determines the state of a modeled object (e.g., aircraft). It determines the state of its facts and beliefs. Some objects are not physical things in the world, but rather conceptual entities, called conceptual classes in the Brahms language. These represent processes, a set of people, physical objects, and locations (e.g., flights), and institutional systems (e.g., airlines) that people know about and refer to when organizing their work activities.

**High-level Structure of the Brahms Überlingen model**

An overview of the agents, objects, classes in the Brahms Überlingen model are shown in Figure. 1. All of the systems that are mentioned in the BFU Report, [1], and play a role in accident have been modeled; a partial list follows:

1. Agents:
   - (a) Pilots in each aircraft
   - (b) Two ATCOs at Zurich

2. Geography:
   - (a) Airports: Moscow, Bergamo, Barcelona, Brussels
   - (b) Control Centers at Zurich and Karlsruhe that includes layout of physical workstations
   - (c) Aircraft interior layout

3. Objects:
   - (a) Aircraft: DHL, BTC, AEF, and other aircraft in the sector during the simulated time period (flights are conceptual objects associated with these).
   - (b) Flight Management Computer (FMC) with Cruise & Standard Terminal Arrival Route (STAR) modes for DHL & BTC

   - (c) Control center workstations including radio frequencies and sectors.

4. Activities:
   - (a) Flight Take-off Phase: Clock in ATCC announcing time for departure ATCO communicates departure approval; FMC guides with Standard Instrument Departure; Pilot activities and communications.
   - (b) Flight Cruise Phase: FMC flying in auto-pilot mode using flight plan; Pilot activities and communications.
   - (c) Flight Phase: Pilot flying in auto-pilot mode; ATCOs handoff and accept flights.
   - (d) ATCOs handoff and accept flights
   - (e) Flight Landing Phase: Pilot requests permission to land and ATCO communicates approval; FMC guides with Standard Terminal Arrival Route; Pilot activities and communications.

**Key Subsystems and Conditions**

The following key subsystems and conditions are modeled in the Brahms Überlingen model:

1. Interactions among Pilot, Flight Systems, and Aircraft for climb and cruise with European geography for one plane, the DHL flight plan.
2. BTC flight, flight plan (two versions: on-time and delayed with collision) and geography – this is independent of ATCO actions, to confirm that simulation reproduces collision with flight paths actually flown.
3. Radar Systems and Displays with ATCOs, located in Control Centers, monitoring when flights are entering and exiting each European flight sector in flight plans.
4. Handover interactions between Pilot and ATCOs for each flight phase.
5. Two ATCOs in Zurich, Radar Planner (RP), and ARFARadar Executive (RE), assigned to two workstations (RE has nothing to do under these conditions).
6. Add TCAS with capability to detect separation violations, generate Traffic Advisory (TA) and Resolution Advisory (RA). DHL and BTC are delayed (on collision course, which tests TCAS)
7. Pilots follow TCAS instructions.
8. ATCO may intervene prior to alert depending on when ATCO notices conflict in Radar Displays since ATCO is busy communicating with other flights, moving between workstations, and trying to contact Friedrichshafen control tower on the phone.
9. AEF flight and flight plan so Zurich ARFA RE performs landing handoff to Friedrichshafen controller.
10. Third plane, the AEF flight, arrives late, requiring ATCO communications and handoff to Friedrichshafen: (a) Handled by ATCO in Zurich at right workstation (ARFA sector) and not left East and South sector workstation. (b)
Phone communications for handovers, (c) Methods used by ATCO when phone contact does not work:

(a) Ask Controller Assistant (CA) to get another number (pass-nr); requires about 3 minutes for CA to return
(b) After pass-nr fails, discuss with CA other options about 30 sec
(c) When not busy handling other flights, try pass-nr again.
(d) When plane is at Top-Of-Descent waypoint, as specified in STAR, for landing at airport, within N nm of airport, method of last resort is to call pilots on radio and ask them to contact the tower directly

11. STCA added to ATCO workstations (modeling normal and fallback mode without optical alert). The ATCO responds to alert by advising Pilot to change flight level based on next flight segment of flight plan.

12. Reduce to one Zurich ATCO which triggers the sequence of variations from the nominal situation; now Zurich ATCO must operate flights from two workstations.

Note that fig: key does not show the geography, facilities, and flights.

PROPERTIES OF INTEREST
The question that the analysis tries to answer, using both simulation and verification, is why under certain conditions, a collision is averted, while in others it is not? In the analysis we try to gauge how the temporal sensitivity and variability of the interactions among ATCO, TCAS, and the pilots impacts the potential loss of separation and collision of the planes. Concretely, the questions that we ask during the analysis are:

- Given that the arrival of the AEF flight is disrupting the ATCOs monitoring of the larger airspace (e.g., if it arrives sufficiently late, no collision occurs), what is the period (relative to the BTC and DHL flights paths) when AEF’s arrival can cause collision?
- During this period, does a collision always occur or are there variations of how the AEF handoff occurs, such that sometimes the separation infringement is averted?
- Is there evidence that high-priority activities such as monitoring the sector are repeatedly interrupted or deferred, implying the ATCO is unable to cope with the workload?

SIMULATION OF THE ÜBERLINGEN SCENARIOS
The Brahm's Überlingen Model defines a space of work systems (e.g., is STCA optical functioning? are there two ATCOs?) and events (e.g., the aircraft and flights). Every configuration of model, which involves configuring initial facts, beliefs, and agent/object relations, constitutes a scenario that can be simulated and will itself produce many different outcomes (chronology of events), because of non-deterministic timings of agent and object behaviors. The model was developed and tested with a variety of scenarios (e.g., varying additional flights in the sector; all subsystems are working properly). The Überlingen accident is of special interest, in which systems are configured as they were at the time of the accident and the DHL and BTC planes are on intersecting routes.

Setting up the Simulation
The key events that occur during simulation are logged chronologically in a file that constitutes a readable trace of the interactions among the ATCO, pilots, and automated systems. The log includes information about the following: (a) ATCO-pilot interaction regarding a route change, including flight level and climb/descent instruction, (b) Separation violation events detected by TCAS, including TAU value, (c) Closest aircraft and separation detected by ATCO when monitoring radar, (d) STCA optical or aural alerts, including separation detected, (e) Agent movements (e.g., ATCO shifting between workstations), (f) Aircraft movements, including departure, entering and exiting sectors, waypoint arrival, landing, collision, airspeeds, vertical, etc., (g) Aircraft control changes (e.g., autopilot disengaged), (h) Radio calls, including communicated beliefs, (i) Phone calls that fail to complete.

Summary of Results
The outcome of ten simulation runs of Brahm’s Überlingen model configured for the collision scenario are shown Figure 2. In the simulation runs 1, 2, and 3 shown in Figure 2, the ATCO intervenes before TCAS TA, but planes have not separated sufficiently, TCAS will take BTCs descent into account, advising DHL to climb. In the simulation runs 4, 5, 7, 8, and 9, the ATCO intervenes between TA and RA. In these runs whether the planes collide depends on timing. As shown in Figure 2 two of the five runs results in a collision. Note that in our model a collision is defined as occurring when the vertical separation between the planes is less than a 100 feet. Finally, in the simulation runs 6 and 10, the ATCO intervenes about 10 seconds after TCAS RA—which BTC pilots ignore (or might be imagined as discussing for a long time)—BTC continues flying level while DHL descends, so they miss each other, separated by more than 600 ft at the crossing point. In other runs, we have also observed that ATCO intervenes so late, he actually takes the pilots’ report about TCAS RA instructions into account.

When ATCO intervenes in the period between the TA and RA in runs 4, 5, 7, 8, and 9 collision is possible, as at Überlingen. That is, ATCO has to intervene before TA advising BTC descent for BTC to respond sufficiently for TCAS to advise DHL to climb. In runs 4 and 7, collision is narrowly averted because BTC begins to descend four or five seconds after the TCAS RA, which is sufficient for a narrow miss (just over 100 feet). In run 9 the BTC descent begins 5 seconds before the RA, hence the aircraft miss by more than 200 feet). Runs 5 and 8 (Figure 9-3) lead to collision because the TCAS RA and BTC AP disengage occur at the same time, as happened at Überlingen. Because the model uses the Überlingen descent standards to control the BTC and DHL aircraft during the emergency descent, simulation matches the paths of the aircraft at Überlingen guaranteeing a collision (within defined range of error). In both cases, TCAS didn’t instruct DHL to climb because BTC was above DHL at that time and of course had not begun its descent.
When ATCO intervenes after the RA, the BTC pilots in the simulations ignore the RA advice and continue level flight, which itself averts the collision—even though ATCO advises BTC to descend (which implies not considering that DHL is below them). We of course do not know what the BTC pilots would have done if ATCO had not intervened. With more than one pilot interpreting TCAS correctly, it appears possible the BTC would have climbed.

The final AEF hand-off (directing the pilots to contact the tower) always occurs in the simulation after the TCAS RA; at Überlingen it occurred prior to the TA. This discrepancy raises many questions about what variability is desirable. In the verification of the system we were able to find certain cases where the final AEF hand-off occurs before the TCAS TA and the planes still collide.

The simulation results for other configurations of the Brahms Überlingen model are described in the technical report [10].

**Formal Verification**

We use verification techniques to systematically explore the various behaviors in collision scenario of the Brahms Überlingen model configuration in addition to the simulation experiments.

**Background**

In [16] we present an extensible verification framework that takes as input a multi-agent system model and its semantics as input to some state space search engine (or a model checker). The search engine generates all possible behaviors of the model with respect to its semantics. The generated behaviors of the model are then encoded as a reachability graph $G := \langle N, E \rangle$ where $N$ is a set of nodes and $E$ is a set of edges. This graph is automatically generated by the search engine. Each node $n \in N$ is labeled with the belief/facts values of the agents and objects. In the work in [16] we generate the reachability graph using the Java PathFinder bytecode analysis framework. An edge between the nodes represents the updates to beliefs/facts and is also labelled with probabilities. The reachable states generated by the JPF are mapped to the nodes in a reachability graph. The verification of safety properties and other reachability properties is performed on-the-fly as new states and transitions are generated in JPF. Additional verification activities can be performed on the reachability graph after all the JPF states have been generated.

**Limitations**

The JPF-based MAS connector requires a complete implementation of the Brahms semantics to generate the intermediate representation. The current implementation of the Brahms semantics presented in [16] only supports a limited set of constructs. Furthermore, JPF is a stateful analysis engine that stores the generated model in memory. Capturing the state of all the agents and objects in Brahms including their workframes and thoughtframes can lead to large memory requirements. Additionally, for large systems it is often intractable to generate and capture even just the intermediate representation in memory.

**Stateless Brahms Model Checking**

![Figure 2. Outcomes of ten simulation runs of Überlingen scenario. Bold indicates greatest potential for collision (ATCO intervenes between TA and RA; both aircraft descending).](image-url)
To overcome the limitations just described, in this work we adopt a *stateless* model checking approach. *Stateless model checking* explores all possible behaviors of the program or model without storing the explored states in a visited set. The program or model is executed by a scheduler that tracks all the points of non-determinism in the program. The scheduler systematically explores all possible execution paths of the program obtained by the non-deterministic choices. Stateless model checking is particularly suited for exploring the state space of large models. In this work we instrument the Brahms simulator to perform stateless model checking. The instrumented code within the Brahms engine generates all possible paths (each with different combinations of activity durations) in depth-first ordering. Stateless model checkers like VeriSoft [14] do not in general store paths; however, in order to perform further analysis of the behaviors space the Brahms stateless model checker can store all the generated paths in a database.

**Non-determinism in Brahms**

There are two main points of non-determinism in Brahms models. The first point of non-determinism is due to durations of primitive activities. The different primitive activities in Brahms have a duration in seconds associated with them. The duration of the primitive activity can either be fixed or can vary based on certain attributes of the primitive activities. When the random attribute of a primitive activity is set to true the simulator randomly selects the primitive activity duration between the min and max durations specified for the activity. The second point of non-determinism arises from probabilistic updates to facts and beliefs of agents and objects. Updates to facts and beliefs are made using *conclude* statements in Brahms. An example of a conclude statement is: `conclude((Pilot.checkStall = false), bc:70, fc:70).` This states that the belief and fact, checkStall, in the Pilot agent will be updated to false with a probability of 70%. Here bc represents belief certainty while fc represents fact certainty.

In the Überlingen model currently there are only deterministic updates to facts or beliefs. The updates to facts and beliefs are asserted with a 100% probability. Nevertheless, there is a large degree of non-determinism due to variations in activity durations. The difference in minimum and maximum durations ranges from 2 seconds to a few hundred seconds. This can potentially lead to a large number of timing differences between the various events. In future work we plan to extend the Brahms Überlingen model to support probabilistic variations in order to account for errors by humans and automated systems.

**Behavior Space**

The scheduler within the stateless Brahms model checker generates all possible paths through the different points of non-determinism in the Brahms model. Note that in describing the output of the Brahms stateless model checker we use the terms path and trace interchangeably. Intuitively, a path (or trace) generated by the Brahms stateless model checker is equivalent to the a single simulation run. More formally, a path or trace is a sequence of events executed by the simulator \((e_0, e_1, e_2, \ldots, e_i)\). Each event in the trace is a tuple, \(\langle a, t, (u, val) \rangle\) where \(a\) is the actor id, \(t\) is the Brahms clock time, \(u\) is the fact or belief updated to the value \(val\). For each trace we generate a sequence of nodes in the intermediate representation \(n_{init}, n_0, n_1, n_2, \ldots, n_t\). The initial node in the sequence, \(n_{init}\) is labeled with the initial values of belief/facts values for the various agents and objects. The event \(e_0 := \langle a_0, t_0, (u_0, val_0) \rangle\) is applied to the initial node \(n_{init}\) where the value assigned to \(u_0\) is updated to \(val_0\). Each event is applied in sequence to a node in the intermediate representation to generate \(n_{init}, n_0, \ldots, n_{t-1}, n_t\).

**Summary of the Results**

There are several activities in the Brahms Überlingen model with a specified range of minimum and maximum durations. Due to the size and complexity of the model, generating a single trace takes approximately 15 minutes. It would in all likelihood take a few weeks to generate all possible traces within the system. In order to mitigate this computational bottleneck, we scope the verification of the model. We non-deterministically explore the minimum, median, and maximum durations for each activity in the model. In the traces generated by the stateless Brahms model checker, approximately a third of the generated traces lead to a collision. If the collision was an undesired property (a fault) in the model, then the results of the model checking would indicate a very high error density. It is, however, important to note that the goal of the collision configuration in the Brahms Überlingen model was to faithfully recreate the conditions that led to the planes colliding. The verification results demonstrate that even with the timing variations a large number of paths (one-third of the generated paths) lead to the collision due to the fact that the AEF flight was distracted with the ATCO flight, the short-term collision avoidance system (STCA) which provides optical and audible alerts for the ATCO was under maintenance, and the fact that there was only one ATCO on duty.

We present an overview of the verification results for the two properties of interest described earlier: (a) how does the arrival of the AEF flight impact the ATCO’s ability to monitor the large airspace and (b) does a collision always occur in this period? Some of the results described in the simulation also hold true for the traces generated during the verification. We were able to study other interesting aspects of the model with respect to the properties that were not observed in the simulation. In the simulation runs of the Überlingen model the final AEF handoff (directing the pilots to contact the tower) always occurs in the simulation after the TCAS RA; in the verification runs, however, the final AEF hand-off can occur before the TA is ever issued. Some of the cases observed are as follows:

1. The final AEF hand off occurs before the TA, the separation infringement is detected and resolved.
2. The final AEF hand off occurs before the TA, and the planes still collide. Note that this is a very interesting scenario because the Überlingen accident report states that the final AEF hand off occurred before the TA for either of the planes.
The verification results indicate that while in some cases the AEF flight arrival can exacerbate the problem for the ATCO, it is not the only cause of the accident. From a wider systemic perspective, the separation violation did not occur at Überlingen only because of the arrival time of the AEF flight. Rather, the Skyguide company had tolerated a deviant form of SMOP during night operations: consequently nobody was responsible for the system, particularly during the maintenance process. Otherwise ATCO would have been informed that STCA Optical alert was not functioning and that the backup phones had been disabled. We can encode the output of the Brahms Stateless model checker into a PRISM model, [17], and check various probabilistic properties of the system. The updates to the facts and beliefs represent the probabilistic updates to the system. Note that the output of the Brahms stateless model checker can be encoded as the intermediate representation in the work in [16].

The Überlingen collision scenario does not provide opportunity for sophisticated properties since a large number of paths lead to the collision of the planes. The model, however, lends itself to be extended to other general cases and scenarios present in the aviation domain. For example, most pilots in practice commonly ignore the TA alert issues by TCAS, but are trained to react immediately to an RA. Rather than being specified as initial configuration we can extend the model to support probabilistic updates that indicate whether or not the phones are down, whether the STCA is in maintenance, and the other ATCO officer is on a break.

DISCUSSION

Our overall goal is to model and analyze interactions between humans and automated systems, and apply this methodology to the safety analysis of NextGen. It is our conjecture that such an analysis needs to be done early in design before deploying any new automation. The problems that are related to safety which can be detected early in the design phase are easier to fix. In order to achieve this goal, we need to reason about how humans perform their tasks in conjunction with complex, thus hard to grasp in its entirety, automation. It led us to making the following choices.

To model the interactions between humans and automation we chose the Brahms modeling language. Brahms has the ability to reason about agents and objects that can represent humans as well as automated systems. The agents can have varying levels of intelligence which provides us the flexibility to model agents at varying granularity. The simulation of agent behavior can range from rational procedure following to simulating how people actually doing their work, i.e., their practice, or simulating reactive behavior that is fragmented, unfocused, incomplete, etc. We can encode non-deterministic choices in the model and even assign probabilities to these choices. We can also express the notion of belief, which is quite important when a human interacts with a complex system. For example, the pilot in charge during the Air France 447 accident described in [16] had wrong assumptions about the pitch of the plane and being able to model his belief as to the state of the system is important. Brahms also gives us the added benefit of being able to model precisely a working environment (e.g., a controller console is two yards away from another one, which implies some time is needed to switch from one to the other). Early in the design phase, details, including those about work settings, are not necessarily known. However, it is advantageous to have this feature when one wants to refine the analysis as one gets closer to deployment at particular locations. Quite often FAA ground systems require adaptations when they are fielded at new locations. Using Brahms’ capabilities, one can tune a generic model to the details of a particular location and verify that no new safety issues can appear.

From an analysis point of view, we are taking a pragmatic approach, adopting both simulations and model checking techniques. In this work, we experimented with generating the behaviors using a stateless model checker. The reachability graph can also be generated by the JPF model checker as described in [16] where currently a subset of the Brahms semantics are implemented as a Java library. After completing the implementation of the Brahms semantics in JPF we can leverage the several extensions of JPF to facilitate the scalability of the analysis. Another important criterion is to provide the ability to reason about beliefs and probabilistic behaviors. As described in the verification framework of [16], we can encode the reachability graph into inputs for different model checkers such as PRISM, SPIN, and NuSMV. This allows us to leverage state of the art verification technologies and check properties related to probabilities, liveness, and beliefs.

With respect to the methodology, it is important to reconcile the need for details in an analysis with the fact of performing an analysis early in design when details are not necessarily known. Our first answer is based on using fairly generic models for controllers and pilots based on the current literature (which includes the body of work in human factor studies). These generic models can then be refined as we progress towards implementation, adaptation, and finally deployment. Scalability might still be an issue, and we will address it by using proper abstractions and, if possible, compositional verification techniques. Similarly, we will have to address the scalability of the models when it comes to analyzing larger parts of the National Airspace System (e.g., multiple airports, multiple sectors, many airplanes). Fortunately JPF is being extended with capabilities to address abstractions and compositional verification. So, we will have a good base from which to draw.

RELATED WORK

In addition to the approaches mentioned in the Introduction, there is a large body of work dealing with the verification of human-machine interactions and with the verification of avionic systems. The DO-178B titled Software considerations in airborne systems and equipment certification is the official guideline for certifying avionics software. Several model checking and formal verification techniques have been employed to verify avionic software in [19, 2] in accordance with the DO-178B. Recent work describes how changes in aircraft systems and in the air traffic system pose new challenges for certification, due to the increased interaction and
In [19] the authors present a framework that supports multiple input formalisms to model avionic software; these include MATLAB Simulink/Stateflow and SCADE. These formalisms are then translated into an intermediate representation using Lustre, a standard modelling language employed to model reactive systems with applications in avionics. Finally, Lustre models are translated to the input language of various model checkers, including NuSMV, PVS, and SAL. The key difference with the approach we describe for formal verification is that the translation is purely syntactic. In our work, instead, we do not translate the modelling language, but we operate at the level of the Brahms simulator. This allows us to consider the full semantics of Brahms, and not a subset of the language compatible with the verification tools. More importantly, we explicitly consider a hybrid system composed of software and humans, and we are able to reason about beliefs and probabilities, while the work in [19] is limited to temporal properties.

There is a vast literature to model human-machine interactions. Recently, Combefis et al. [12] have employed Java Pathfinder as a model checker to verify human-machine interactions. The modelling language is based on Statecharts but, as in the work of [19], this formalism does not allow us to reason about probabilities or beliefs. We refer to the references available in [12] for an overview of other similar approaches.

The work of Yasmeen and Gunter [25] deals with the verification of the behaviour of human operators to check the robustness of mixed systems. In this approach the authors employ concurrent game structures as the modelling language and translate the verification problem to a model checking instance using SPIN. As in the previous cases, our approach is different in that we do not perform syntactic translations and we reason explicitly about probabilities and beliefs. Additionally, we also provide a detailed and complex case study.

The Enhanced Operator Function Model (EOFM) is another modelling language developed to model and verify interactions between humans and automated systems [7]. Similarly to the other works described above, EOFM is translated into the input language of the model checker SAL to perform verification of properties encoded in linear temporal logic. The authors describe the application of their framework to the verification of a cruise control system for cars. The main limitation of this approach is that it currently supports single-operator systems only and, as in the case of [19] and [25], there is no support to reason about probabilities and beliefs.

Acknowledgements This work is supported by the System-Wide Safety and Assurance Technologies Project in the Aviation Safety Program of NASA’s Aeronautics Research Mission Directorate. The views expressed in this paper are those of the authors and do not necessarily represent the views of NASA or the U.S. Government. Author names are alphabetically ordered following the first author. Sections of this paper describing Überlingen collision, Brahms framework, Brahms Überlingen model, and simulation scenarios are adapted from [10]. We thank Ron van Hoof providing support for the Brahms simulation system.

REFERENCES


Development of Numerical Sensor Models for Cooperative and Non-Cooperative Collision Avoidance

Anna Elena Tirri, Giancarmine Fasano, Domenico Accardo, and Antonio Moccia
Università degli Studi di Napoli “Federico II”
Piazzale Vincenzo Tecchio 80
Napoli, I80125, Italy
anna.elena.tirri@unina.it
+39 0817682149

Vincenzo Baraniello
Onboard Systems and ATM Department
Centro Italiano Ricerche Aerospaziali (CIRA)
Via Maiorise snc
Capua (CE), I81043, Italy
v.baraniello@cira.it
+39 0823623791

ABSTRACT
One of the most hazardous consequences of the growth of air traffic volume foreseen in the next decade is undoubtedly the threat of Mid Air Collision in proximity of major airports. Besides this consideration, another great-involved factor has to be addressed to the integration of Unmanned Aerial Vehicles in the National Airspace. Thus, in order to reduce the probability of collision risk, both manned and unmanned aircraft should be equipped with Detect, Sense and Avoid (DS&A) systems able to ensure enhanced situational awareness capabilities. Modern aircraft can use several types of collision avoidance systems to prevent unintentional contact with other aircraft, obstacles, or the ground. These systems include cooperative and non-cooperative sensors in standalone or multi-architectural configuration, such as: radars, electro-optical, Automatic Dependence Surveillance Broadcast (ADS-B) and Traffic Alert Collision Avoidance (TCAS) systems. In this framework, the European MIDCAS (MID Air Collision System) project aims at demonstrating solutions for Mid Air Collision Avoidance functions. Within this framework, numerical models of cooperative and non-cooperative systems have been realized by University of Naples “Federico II” in collaboration with the Italian Aerospace Research Center (CIRA). This paper focuses on the analysis and description of these models in Simulink™ environment. In the last part of the paper, some simulation results are presented and analyzed.

Author Keywords
Unmanned Aircraft Systems, Collision Avoidance, Cooperative and Non-cooperative Systems.

ACM Classification Keywords

INTRODUCTION
The operation of manned and unmanned aircraft in the same airspace, including civil airspace, is an important feature that will enable growth in the industry and commercial sectors, and greater utility for all the operators. In today’s National Airspace System (NAS) there are a number of technologies, process, and procedures, which together ensure that the risk of collision for manned aircraft is consistent with an acceptable level of safety. The introduction of unmanned aircraft systems (UAS) places significant attention on developing technologies that would enable UAS to satisfy the requirements to “see and avoid” other aircraft in the flight path [2, 16]. A Detect, Sense and Avoid (DS&A) system has to operate in a widely varying environment of different UAS types, different airspaces, different operating conditions and different obstacles to avoid.

The UAS types vary in size from the palm sized aircraft to the airliner-sized aircraft. The environment in which these UAS are expected to fly varies from daylight to darkness, and from clear weather to cloudy, foggy, rainy, plus in a variety of airspace types, from uncontrolled to fully controlled.

As a consequence, an unmanned aircraft must be properly equipped for the conditions and airspace in which it is intended to operate. To fly in controlled airspace, an aircraft must have specific navigation and surveillance functions. Due to the large range of UAS types, the mission of these aircraft and equipage requirements vary as well.

In all cases, Collision Avoidance is a key issue to be addressed and determining the appropriate type of sensor is a challenging, multidimensional problem.

The surveillance function for collision avoidance can be performed through two fundamental methods: cooperative sensors, wherein an aircraft is equipped with a transponder to interrogate and/or broadcast information; non-cooperative sensors, which are able to detect targets autonomously. Non-cooperative systems include radar and electro-optical sensors that provide information independently by other information sources. In particular, radars represent a suitable option to provide the required
situational awareness in the case of medium/large UAS platform, which have to attain a reliable full autonomy from ground. These sensors provide accurate range estimations and cannot be much affected by weather conditions, though they are not very accurate for angular information. In this case, the electro-optical sensors can be a good solution for increasing angular accuracy in a multi-sensor architecture.

Automatic Dependence Surveillance Broadcast and Traffic Alert and Collision Avoidance systems are classified as cooperative systems. The former is expected to provide an improvement of the Separation Assurance function. In fact, today this function is performed by controllers and it is based on measurements of surveillance radars. In the future, the ADS-B will enable pilots to perform self-separation assurance maneuvers [8,12]. It can be foreseen that ADS-B will play a key role for the integration of UAS into civil airspace. Moreover, General Aviation aircraft will be provided with a system that will ensure a remarkable increase in the overall situational awareness and a reduction in the number of collision threats. As for TCAS, it has been used for many years as main collision avoidance system for airliners, providing a significant reduction of mid-air collision risk.

All these systems have been considered within MIDCAS project. This European project aims at demonstrating the baseline solutions for the UAS Midair Collision Avoidance function (including separation), acceptable by manned aviation community and being compatible with UAS operations in non-segregated airspace. In particular, three different steps with high levels of interaction and interdependency need to be realized: a progress on standards for Sense and Avoid; design of a generic S&A function to be tested in simulations and finally design of a S&A demonstrator to be tested in manned and UAS flight. Within this framework, CIRA and the university of Naples “Federico II” collaborated for the realization of numerical models for S&A systems to be tested in simulation environment. In fact, the realization of numerical software is very important for saving time and effort prior to conduction of real flight tests. In particular, simulation-based testing should be adopted whenever possible to validate the software under realistic conditions [7,14].

The paper focuses on the description of the developed sensor models highlighting the needed input variables and the outputs of the systems. The last part of the paper is devoted to some results relevant to model validation.

SENSOR MODELS

The developed numerical sensor models have been realized and tested by means of MATLAB/Simulink™ tools. These models simulate radar, electro-optical, ADS-B and TCAS systems.

In the following section, the developed sensors models are described after a briefly description of their functionalities.

Radar Sensor

Active microwave sensors represent a suitable option to provide the required situational awareness capability in case of medium/large UAS platforms. In fact, airborne radars provide direct and typically accurate range estimates (also range rate if Doppler processing is used). Moreover, they can guarantee large detection range, low levels of missed detection (ground echoes have to be properly filtered), and cannot be much affected by weather conditions, so that the all-time all-weather operations can be guaranteed.

It is worth noting that, in the choice of wavelength, maximizing detection range, minimizing sensor dimension to enable installation on-board a light aircraft and improving as much as possible angular resolution are contradicting requirements.

In fact, radars operating at low frequencies are relatively unaffected by atmosphere, but are large in size and unable to provide required spatial resolution, due to the main lobe width that is directly proportional to operating wavelength. A higher frequency radar, instead, is smaller in size and provides better resolution for given aperture size, but is more susceptible to atmospheric and weather effects, and in particular to rain. Frequencies ranging from C-band (about 6 GHz) to Ka and W band (35 and 94 GHz, respectively) have been used and/or proposed in sense and avoid applications.

The developed sensor model is able to simulate all these operating frequencies by proper selection of model inputs.

The Radar Simulink main block is reported in Figure 1.

**Figure 1. Radar Simulink Block.**

The input parameters are:

- Number of intruders;
- Ownship (Own_data) and intruders motion (intruder_1, intruder_2, intruder_3, intruder_4);
Radar/Tracking model parameters that consist of: Probability of false alarm, number of integrated pulses, peak power, radar frequency, antenna gain, effective noise temperature, receiver noise figure, noise bandwidth, radar mounting angles, radar field of regard, radar accuracy and biases.

- Environmental conditions;
- A flag indicating the number of intruders considered in the simulation (N_Intr_Flag_Vec);
- UTC (Coordinated Universal Time) time that is in general a time measurement to provide measurement time tagging (UTC_time).

The radar model simulates a scenario in which the own aircraft and one or more (up to 4) intruders are presented. The position and velocity of these aircraft are passed to the main block of the software on 4 different parallel channels. Before this operation, the radar main parameters (defined above), the weather conditions, mean radar cross section (RCS) and Swerling type [15] for each intruder, must be defined.

The system is structured in different layers, each of them contains several blocks and performs a particular functions. The simulated radar allows obtaining both radar measurements and estimates position and velocity of the intruders (whose number can be properly selected by means of an input variable) since the model includes a simulated tracking algorithm. Simulated accuracy on tracking-based position and velocity for each intruder are consistent with radar input parameters. Considering now the internal block operations, first of all a data conditioning block is used to calculate exact baseline vectors in the own ship body reference frame (BRF), baselines and their first order derivatives in the North-East-Down (NED) reference frame, and the true range rate for all the intruders. Radar cross-section and Swerling type are also stored in proper variables for later usage in Simulink blocks.

The relative position and velocity vectors are first calculated in NED, and then attitude angles are used for conversion in body coordinates. Subsequently, the model is comprised of a sensor and a tracking model.

The radar sensor model simulates target detection to produce raw radar measurements for each intruder. In particular, the model works by following this logic: input data (DP_true_body) are converted into the radar reference frame (Intruder_DP_radar), taking into account sensor position onboard the own-ship and alignment of radar axes (DP_RADAR block). Then, the available signal-to-noise ratio (SNR) is computed for a single pulse and for each intruder, on the basis of intruder RCS, radar and relative geometry parameters. The SNR is a fundamental variable for the calculation of the Probability of Detection (Pd) that depends also on the probability of false alarm, number of integrated pulses and the intruder Swerling type. Figure 2 shows the developed simulink blocks that perform the above-mentioned functions.

The Logical detection block (Figure 3) allows establishing if an intruder is detected in the current scan by means of a Monte Carlo approach [1] and if an intruder lies within radar field of regard. In the case it is not detected, the probability of false alarm (Pfa) is used to generate an eventual false alarm on the intruder channel. Overall numbers of detected intruders, number of real intruders that have been detected, and eventual presence of false alarms, are given in output of this block. On each channel, intruder “true” range and angles are selected and they will be corrupted with simulated sensor noise. In case of false alarms, random range and angles are generated in this block. The sensor noise is simulated by a Monte Carlo approach.

Track status is handled on the basis of the following rules/assumptions:
- False alarms are eliminated by the tracking algorithm;
- Two consecutive detections determine the creation of a tentative track;
- Three-out-of-five detections determine the creation of a firm track;
- A track is deleted after three negative scans.
The number of deleted tracks on a channel is monitored so as to give unique track identification (Track ID) to each track.

Track data in NED are corrupted on the basis of track status. An output creator block is used to provide the complete output for each intruder channel. Definitively, the outputs of the model are based on: track identification, track status, intruder position and velocity in NED as estimated by the tracking algorithm, range, azimuth, elevation and range rate as provided by radar, estimated intruder RCS and estimated intruder RCS confidence.

**Electro-optical Sensor**

The detection of objects in an image is the capability to discriminate the object itself from the background. This function can be usually realized if two conditions are satisfied: the obstacle has an apparent angular size in the image that is greater then the instantaneous field of view (IFOV) of the camera; the amount of light scattered from the obstacle to the EO system is larger than its internal noise.

In order to simulate these conditions, several parameters have to be defined, such as focal length, detector size and environmental operating conditions. Several atmospheric phenomena have impact on EO system detection performance, such as diffusion and absorption [5, 11]. For example, foggy weather can reduce to zero detection capabilities since water vapor particles have a size of the order of wavelength of EO systems.

The simulated EO system has been developed taking into account these considerations. The model enables testing both infrared and daylight cameras whose parameters are defined externally and passed as inputs to the Simulink model (Figure 4).

In particular, the input parameters to be set are:
- Number of intruders;
- Ownship (Own_data) and intruders motion (intruder_1, intruder_2, intruder_3, intruder_4);
- Electro-optical model parameters that consist of: Probability of false alarm, EO mounting angles, EO field of view, EO accuracy and biases, sensitivity, effective focal length, detector (pixel) size, Number of cycles for 50% detection probability (N50=3), Noise Equivalent Delta Temperature (NEDT), signal loss, background temperature/intensity, target typical size and average temperature/intensity;
- Environmental conditions;
- A flag indicating the number of intruders considered in the simulation (Num_Intr_Flag_Vec);
- UTC (Coordinated Universal Time) time that is in general a time measurement to provide measurement time tagging (UTC_time).

The model is able to accept up to 4 intruders’ inputs on 4 different parallel channels.

Considering the internal structure of the model, first of all a data conditioning block calculates exact baseline vectors in the ownship body reference frame.

Then, EO detection is simulated through the following procedure (the logical architecture is similar to the radar model): the input data are converted in the camera reference frame taking into account sensor position onboard the ownship and alignment of radar axes (DP_true_EO). The output together with target temperature (tT1), target size (tS1), and background temperature is then passed to the logical detection blocks in which the probability of detection is calculated (Figure 5). Then, if an intruder lies within camera field of view, by means of a Monte Carlo approach it is established if the intruder is detected in the current frame. In the case it is not detected, the probability of false alarm is used to generate an eventual false alarm on the intruder channel. As in the radar case, overall numbers of detected intruders, number of real intruders that have been detected and eventual presence of false alarms (Fa_on_ch), are given in output of this block. For each intruder, the “true” angles are selected and then corrupted with simulated sensor noise (EO_measurement_output) (Figure 6).
In case of false alarm, random angles are generated. At the end, the “true” intruder data, or data relevant to simulated false alarms, are corrupted with sensor noises simulated by a Monte Carlo approach. The outputs of the EO Simulink models comprise intruder azimuth and elevation angles.

**Automatic Dependence Surveillance-Broadcast System**

The ADS-B system was developed in order to support aircraft operations so that the limits of ground based radar surveillance could be overcome [6, 10]. These limits consist in the maximum number of aircraft that could be handled at the same time and the absence of feedback to the pilots about surrounding traffic. The basic principle is that each aircraft is equipped with a certified Satellite Navigation receiver and it broadcast its current position by means of a radio link so that it can be received by surrounding aircraft and ground controllers. This is an alternative to surveillance performed by ground radars. This logic is the one implemented in Simulink environment.

In particular, the ADS-B Simulink model has been realized in order to create a 3-D map of surrounding aircraft in the airspace. The model is able to accept up to 100 intruders, each of them with an own message protocol reporting the intruders’ position in terms of latitude, longitude and altitude, the intruder velocity in North-East-Down reference frame, the ICAO address, the accuracy categories, the aircraft identification code, the general aircraft emergency state, the ADS-B message category, Geometric and Pressure Altitude, Aircraft length and width, GPS Antenna Offset, Rate of Climb, Active ACAS Resolution Advisories, Barometric Pressure Altitude. The ADS-B Simulink model is shown in Figure 7.

As in the case of radar and EO models, ADS-B model is composed of different layers. Internally, the model allows broadcasting the intruder message only if it is ADS-B equipped (Intruder_ADS_B_flag). If this check is true, the intruders and own aircraft position (Intruder_pos_vel and own_pos_vel) are converted in North-East-Down coordinates and the relative range and altitude between them are calculated. After that, another check is executed to verify if the relative range and altitude are less than a threshold value. The intruders that are comprised in this surveillance volume limit are passed to the output model where the position and velocity variables are corrupted with sensor noises generated by a Monte Carlo Approach (Figure 8).

**Traffic Alert and Collision Avoidance System**

The Traffic Alert and Collision Avoidance System (TCAS) provides a solution to the problem of reducing the risk of midair collision between aircraft. Each TCAS-equipped aircraft interrogates all other aircraft in a determined range about their position (via the 1,030 MHz radio frequency) and all other aircraft reply to other interrogations (via 1,090 MHz). This interrogation-and-response cycle may occur several times per second [3]. The TCAS system builds a three dimensional map of aircraft in the airspace, incorporating their range (garnered from the interrogation and response round trip time), altitude (as reported by interrogated aircraft), and bearing (by the directional antenna from the response). Then, by extrapolating current range and altitude difference to anticipated future values, it determines if a potential collision threat exists.

During surveillance, the TCAS output depends on the type of transponder that is mounted onboard other aircraft: mode A provides only ID information and range; mode C provides ID, range and altitude; mode S provides ID, range, altitude, call-sign and squawk code. In Figure 11, the block representing mode A interrogation is reported as an example. The Simulink TCAS block is reported in Figure 9.
The input parameters are based on: Intruders’ Position based on WGS84, Aircraft Call-Sign, Squawk Code, TCAS Mode based on Transponder installed onboard intruders, TCAS surveillance simulated uncertainty (intruder_data).

TCAS developed model is based on this logic. The model allows creating a map of up to 100 intruders in the airspace surrounding the own aircraft. To this end, both intruders’ and own aircraft information are passed to the model. The output is provided only if the intruders are equipped with a transponder. After this check, the position variables defined in terms of latitude, longitude and altitude are converted in NED coordinates and relative positions are calculated (intruder_data_ned). Inside the main block, all the interrogation modes outputs are implemented but only one is activated depending on the transponder installed onboard the aircraft (NO_TCAS, TCAS_mode_A, TCAS_mode_C, TCAS_mode_S blocks) (Figure 10). The model outputs the relative aircraft information corrupted by sensor noises simulated by a Monte Carlo approach.

PERFORMANCE PARAMETERS EVALUATION
In this section the most important parameters for radar and ADS-B simulation are discussed. They comprise the radar Probability of Detection (Pd) and GPS angular accuracy as function of range. These parameters are the core of the developed systems.

In the radar case, the detection range for a given target can be calculated in probabilistic terms in the basis of achievable signal-to-noise ratio (SNR) and the number of impulses integrated to perform target detection [13].

The basic equation within this framework is the radar equation that gives single pulse SNR for assumed transmitted power (Pt), antenna gain (G), impulse duration (influencing the bandwidth B), receiver noise figure (F), losses (L), target radar cross section (\(\sigma\)), effective noise temperature (Te), and the target range (R) as follows:

\[
SNR = \frac{PtG^2\lambda^2\sigma}{4\pi} \frac{kTeBFR^4}{L}
\]  

On the basis of this equation, of the assumed probability of false alarm, number of integrated pulses, and Swerling type it is possible to calculate the Pd [1].

As an example, two different radar frequencies (Ka and X band) are analyzed in the following. In particular, the performance has been evaluated for different atmospheric conditions.

In case of Ka-band, radar parameters have been set as follows: antenna size 0.3 x 0.3 m, peak power 1 Kw, the impulse duration of 100 ns with 2 pulses coherently integrated, mean obstacle radar cross section of 1 m\(^2\), obstacle Swerling type 2 (propeller aircraft), the receiver noise figure set to 6 dB, antenna efficiency is 0.5 and probability of false alarm is \(10^{-9}\).

Figure 12, reports the probability of detection as function of range in two different weather conditions: clear air and quantified by the rain rate in millimeters per hours. In this case, 10 mm/h has been chosen for the second condition.
The same analysis has been conducted considering a X-band radar. The parameters have been set equal to the previous one and only the radar frequency has been changed. Of course, this means that a worse angular accuracy is obtained because of the wider antenna main lobe. Figure 13 reports the probability of detection as function of range for both clean air and rain.

Concerning the ADS-B, the quality of service is related to the performance of the GPS systems installed onboard in terms of accuracy, integrity, availability and continuity of service. In fact, the accuracy changes passing from standalone GPS to differential one. In standalone conditions, average GPS positioning accuracy at 95% probability level (2 sigma) can be assumed to be of 7.1 m for horizontal position and 13.2 m for vertical position. In case of ground-based (baseline of a few km) and satellite-based differential GPS, the 1-sigma accuracy are 0.99 m (horizontal) and 1.85 (vertical), and 1.27 m (horizontal) and 2.37 m (vertical), respectively. These values have been obtained referring to [4, 9].

Once positioning uncertainties are fixed both for the own-ship and the intruders, relative positioning accuracy can be estimated to a first approximation by assuming uncorrelated measurements. However, this is highly over-conservative approach. In fact, in collision avoidance scenarios the distance among the aircraft is of the order of a few kilometers, and it is likely that significant cancellation of common errors when calculating relative positions: the resulting accuracy can be only slightly worse than range-based differential GPS when considering standalone GPS receivers.

For the sake of concreteness, correlation among position measurements can be taken into account by applying a 0.5 factor to the uncertainty estimated assuming uncorrelated measurements.

Given a linear uncertainty $\sigma_l$ on relative positioning, to a first approximation it can be converted into angular error $\sigma_\theta$ on the basis of the Equation (2):

$$\sigma_\theta = 2 \arctg \frac{\sigma_l}{2R}$$

that can be approximated to (Equation (3)):

$$\sigma_\theta \approx \frac{\sigma_l}{R}$$

The Figure 14, Figure 15 and Figure 16 report the angular accuracy as function of range. The plots have been obtained considering the relative range between and intruder and own aircraft.

In the first case, both aircraft are equipped with standalone GPS; in the second case, the own aircraft is equipped with a ground based differential GPS and the last case, an EGNOS (European geostationary navigation overlay system) based GPS has been considered for both aircraft.

The plots show that the angular uncertainty is small compared with typical non-cooperative sensor performance (except for very small range), especially if ground-based or satellite-based differential GPS is adopted.
Naples, Italy, May 28-30, 2013

Figure 15. Angular uncertainty as a function of range using a differential GPS (own-ship) and a standalone GPS (intruder) for relative position estimation.

Figure 16. Angular uncertainty as a function of range using an EGNOS-based GPS for position estimation.

CONCLUSION
This paper focused on the simulation of cooperative and non-cooperative sensor models. These systems are very important sources for the realization of Collision Avoidance logic onboard Unmanned Aircraft systems. In the first part of the paper, the different systems have been described highlighting their advantages in terms of system sizes, costs and performance. Then, the developed systems models have been presented and described in details, specifying the inputs and outputs variables and the operations executed inside the models. In the last part of the paper, some key parameters for performance evaluation of radar and ADS-B systems have been analyzed, such as radar probability of detection and GPS accuracy. Future papers will report and discuss the results of the simulated sensor models integrated in a real time environment.

ACKNOWLEDGMENTS
The PhD grant of Miss Tirri is co-funded by EUROCONTROL through the research network HALA! in the framework of WP-E SESAR project.

REFERENCES
3. Federal Aviation Administration, Introduction to TCAS II, version 7.1.
Effect of Flight Plans Predictability and Accuracy on Traffic Demand Forecast

Esther Calvo Fernández
R&D Engineer
CRIDA
ecalvof@e-crida.aena.es

José Manuel Cordero
Principal Researcher
CRIDA
jmceddeo@cridda.es

ABSTRACT
Traffic Demand Forecast is a key aspect of Air Traffic Management (ATM), even more relevant recently as becomes increasingly more important to effectively scale capacity and associated resources (Air Traffic Controllers amongst them) to match the actual demand.

In the case of ATM, many different demand forecasts can be considered, from the simplest to the most complicated ones, all aiming to provide a reliable forecast with the maximum time anticipation, thereby allowing the Network Managers at the different network levels to anticipate the needs to fulfill. However, there is no real indicator assessing the real accuracy of these forecasts.

In this scenario it has been launched by CRIDA a long-term research activity with the goal to define reliability indicators that can be applied to traffic forecasts to fill in the described gap. This study will be performed through the availability of samples of both real-time flight plan information and post-flight data, and divided into several analysis phases. This reliability index is envisioned as a valuable complementary tool in combination with any traffic forecasts, as is intended to provide the user improved awareness of the results obtained by using a particular demand forecast.

Thus, this paper addresses an analysis of real ATM system data in order to determine both predictability and accuracy indicators which will eventually combine for a reliability index. The promising results of the first stage of this research activity are presented here, showing some already applicable conclusions.

Authors keywords
ATM, Predictability, Accuracy, Reliability, Traffic Demand Forecast.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
As new ATM systems implementing new and more accurate models (able to support the necessary functionalities for an efficient service in a more demanding environment) arise, it appears as a constant requirement the need of better information to feed these systems (not always achievable because of technological, procedural or other restrictions). In particular, in systems such as Demand and Capacity Balancing (DCB) modules it is a key aspect the availability of a reliable demand forecast for effective system functionality.

Additionally, the appearance of new systems generates new demands in this direction, reinforcing the needs of reliable data sources which eventually contribute to traffic demand forecasts improvement.

Traffic forecasts, on the other hand, are extensively obtained through a wide range of different methodologies, including from the ones with a higher degree of realism (e.g., those based on historical data) to other that need to be less refined [1, 2]. All of them may better apply a particular purpose (e.g., Demand and Capacity Balance Models [3], Sector Configuration Optimizers [4] or Performance Monitoring tools [5]) but the common situation with all of them is that currently there is no post-use evaluation of the accuracy and real effectiveness of them for their intended use (more than the operator own experience in the provided results they provide). Additionally, the degree of uncertainty of these forecasts may vary depending on the time anticipation at which they are obtained.

In this scenario, CRIDA (ATM Research and Development Center, associated with AENA, the Spanish Air Navigation Service Provider) has launched a research activity that eventually aims to identify methodologies and metrics to define an applicable reliability index (final goal of the research) that can be used from the simplest flight plan snapshot to a whole traffic demand for a determined interval. Thus, this reliability index aims to provide complementary assessment information associated to any forecast, thereby allowing the system operator to know how good the output obtained with that demand can be considered.

In many knowledge areas with similar needs of reliable and accurate information sources, data quality assessment techniques have been applied to this particular problem. These techniques are able to determine the necessary information quality to get the required performance from a system (e.g., a specific algorithm). While in the ATC domain very limited data quality applications have been
deployed until the moment [7], it is quite common to find them in business environments, among others [8-10]. The research presented in this paper will take into account data quality assessment for the development of a reliability index based both in accuracy and predictability.

**Background**

In order to perform this task it is necessary to compare a particular forecast with the real post-flight data (which contains what has really happened inside the system). Therefore, two data sources are necessary: an estimated traffic forecast and a real post-operation flight data. The comparison between them will be done at flight plan level, as it considered that a flight plan is the minimum unit for any traffic forecast.

CRIDA has wide experience in analyzing the post-flight log data of the real Spanish ATC Platform and extracting the relevant information from it, from flight plan to radar tracks. For the scope of this first stage of the project a traffic sample of one month has been considered for analysis, being considered representative enough for this stage (will be widely increased in the following stages).

This set of information contains all the flight plan data for every flight in the Spanish Peninsular FIR (Flight Information Region) in one month, excluding the Canary Islands (managed as an independent FIR). For this study all the flight plans messages are recorded, not only the initial and final ones. Thereby, given a specific timestamp it can be known the exact information that the ATC Platform had at that moment, obtaining a snapshot of the demand forecast that was expected by the system at that particular moment.

While the results of this research can be extended to any traffic demand forecast based on flight plans, it is particularly relevant in terms of comparison and coherency that both sources of information are originated at the same platform, thereby avoiding the effects of potential differences between data sources features.

**Scope**

As stated, this study is based on one month of data of the Spanish peninsular FIR, containing: i) the real flown data, obtained post-flight from an operational ATC platform log repository, and including real flight plan evolution as well as radar data; and ii) all the flight plan messages received by the ATC system for every flight, from which it can be seen the evolution of the estimated flight plan times and routes during real operation.

As it has been described, this project aims to provide a reliability index for demand forecasts. Obviously, this index has sense merely in the planning (pre-flight) phase, no matter which time horizon in advance is considered. Thereby, the flight plan messages that are of interest for the study are those received by the system while the flight is not yet inside the FIR, but they are already known by the system (with forecasted flight plan information).

Additionally, in this pre-flight phase two different flight situations must be distinguished: i) those flights departing from an airport inside the FIR, for which their pre-flight flight plan messages are always update messages before their off-block time, when the flight gets activated and “enters the FIR”; and ii) those flight coming from outside the FIR and, in most cases, already flying; for them “pre-flight” is intended as “before flying inside the FIR”. These two cases show complementary behaviors, both contributing to the traffic forecast known by the system.

In particular, the Barcelona FIR (a subregion of the Peninsular FIR) will be studied. The reason is that it corresponds to an Area Control Center (ACC) region, which is the minimum unit for which sector configurations and resource allocation (immediate effects of an improved demand forecast) can be performed.

**Objectives**

As stated, the main goal of this long-term research is to define a usable reliability index illustrating the forecasted flight plans data quality at different pre-flight times.

The results presented here explore the concepts of predictability and accuracy (later defined in this paper) as key factors affecting reliability. Additionally, their evolution for a same flight plan during all their pre-flight phase has been studied, in search of reliability levels that ensure enough data quality. These already usable results are the output of the first stage of work, currently ongoing. Thus, this paper addresses: (i) the methodology used for predictability and accuracy evaluation, (ii) the definitions of these two indexes which impact data quality and reliability, and (iii) presentation of the obtained results.

**METHODOLOGY**

**Strategy**

A key aspect in this study is the analysis of the evolution of real flight plans messages in order to provide data quality indicators in terms of accuracy and predictability, as the really flown data can be compared with the different stages of the flight plan during the pre-flight phase (when forecasts are required), thus providing patterns of behaviour that can be later used either for improved modelling (impacting positively during the planning phase), or for corrective actions into the system, or even (as in this study) to estimate the goodness of the information used for planning in terms of a potential reliability index.

The differential factor of this study is the availability of both the real time data (thereby having accurate snapshots of the information in the system at each time) and the real flight data obtained in the post-flight phase (and which analysis makes possible to have a measure of how good each of the mentioned snapshots were). The work presented here corresponds to the first phase of this study.
Data used
The data used in this study has been obtained from the Spanish ATC Platform, in particular from the GIPV (Flight Plan Information Manager) subsystem, containing real operational data from Spanish ACCs. This initial analysis has been done with one month of data, without identifying the particular month and year used for the data sample. In further stages the amount of data will be largely increased.

System Architecture
The mentioned GIPV is a subsystem from the Spanish ATC Platform (SACTA) that contains online flight plan information of those flights that are already flying or are going to fly in the near future (up to 15 hours, nominally) in the airspace of responsibility of the Central Processing Flight Plan subsystem to which it is linked to, and that are already known by the system (key aspect).

Currently in Spain two GIPVs exist: the first one manages information on all Flight Plans affecting peninsular FIR (Flight Information Region), while the other deals exclusively with the Canary Islands FIR.

This flight plan information is continuously updated by receiving, processing and properly combining messages received from a range of sources of information:

- Initial Flight Plan System (IFPS)
- Central Flow Management Unit (CFMU)
- Collateral Centers
- Manual actions performed from the control positions
- Multisensor Surveillance Data Processing (TDVM)

For the scope of the current analysis, all of the messages emitted by the GIPV are stored in a database for further study, as in the original system no data persistence is done.

Workflow
In search of data quality indicators, two initial analyses have been performed to the mentioned online GIPV data. In particular, both predictability and accuracy analysis (later defined in this paper) have been done for two relevant sets of flight plan data:

i) All the Flight Plans from the Peninsular FIR.

ii) Only the Flight Plans with segments of their route in the Barcelona ACC region under study.

Each one of the real time Flight Plan messages contains a lot of information; for the purpose of this study only the following fields have been considered relevant:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Flight Plan Change Event status</td>
</tr>
<tr>
<td>Log time</td>
<td>Flight Plan system registration Date and Time</td>
</tr>
<tr>
<td>Callsign</td>
<td>Flight Identification</td>
</tr>
<tr>
<td>Departure</td>
<td>ICAO location indicator of the departure aerodrome</td>
</tr>
<tr>
<td>Destination</td>
<td>ICAO location indicator of the destination aerodrome</td>
</tr>
<tr>
<td>IOBT</td>
<td>Initial EOBT (Estimated Off-block date and time)</td>
</tr>
<tr>
<td>HFIR time</td>
<td>First FIR estimated entry date and time</td>
</tr>
<tr>
<td>Region entry time</td>
<td>Spanish ATC Region estimated entry date and time</td>
</tr>
</tbody>
</table>

Table 1. Fields from the flight plan messages considered for this study

The procedure’s first step aims to univocally identify each particular flight, getting rid of repetitive callsigns along the data sample, in some cases (e.g., military) even the same day. For this identification the Callsign, Departure, Destination and IOBT fields of the flight plan message are required. It has been found that 1.55% of flight plan messages of the sample don’t have IOBT information, consequently being removed from the traffic sample.

The usual behavior of a Flight Plan in the system includes both a creation and a cancellation event, being all the intermediate events flight plan modifications/updates, activations (i.e., when the flight enters the FIR) or terminations (when the flight has either landed or left the FIR). Only flights including at least a creation and cancellation event are included in the traffic sample.

Thus, while the one month raw data sample considered included a total of 808,386 flight plan messages, this preprocessing to ensure data coherency reduced the sample to 784,467 flight plan messages (97.06% of the original one). These valid flight plan messages correspond to 107,462 effective flights (or flight plans), being 42,801 of them with some segment of their trajectory inside the Barcelona ACC controlled airspace.

A key aspect in the study is to correctly determine if a flight passes through the ACC airspace just from the flight plan waypoint information, which is not trivial. The airports managed by the Barcelona FIR are Girona-Costa Brava, Sabadell, Barcelona-El Prat, Lleida, Reus, Valencia, Alicante, Palma de Mallorca, Menorca and Ibiza. Additionally, CNS (Communication, Navigation and Surveillance) service is provided in the civil-military airports of Murcia and Albacete. This region includes also
other 13 general aviation aerodromes not managed by the Spanish ANSP but that must be considered for this work.

RESULTS
The results of the extensive data analysis performed are divided into two areas, each one of them adding some specific vision of the data quality assessment. This section presents them separately, describing the scope of each one.

Initial flight plans
As the main goal in an ideal demand forecast system would be to know accurate flight plan information with a great advance, for planning purposes it becomes clear that the initial flight plan (or the first notice that the system has about a particular flight) is key in terms of predictability and accuracy of the flight plan data. From this perspective, the initial flight plan message becomes the most relevant one for a specific flight, as after its reception by the system the flight is taken into account for planning purposes.

As stated, in order to analyze the data quality of these flight plans “snapshots” their predictability will be calculated for every message of each flight (thereby not taking into account the real happened data). Thus, predictability indicator (in terms of the initial flight plan message for each flight) is defined as the difference between the time at which the initial flight plan message enters the system (Log time) and the FIR entering time (HFIR) indicated in this initial flight plan message:

\[
Predictability = (HFIR)_{first}^{FP} - (Log \ time)_{first}^{FP}
\]

This indicator measures the anticipation of the flight plan initial messages respect to the FIR entering information they provide. Figure 2(A) graphically shows the flight plan initial messages percentages respect to their predictability.

On the other hand, the accuracy indicator for the initial flight plans messages is defined as the difference between the indicated FIR entering time for that flight included in the initial flight plan message and the real FIR entering time (obtained from post-flight information):

\[
Accuracy = (HFIR)_{first}^{FP} - (HFIR)_{last}^{FP}
\]

This indicator gives a valuable measure of how precise the information in the initial flight plan was on the light of real behavior of the flight. Figure 2(B) illustrates the histogram of the accuracy, in minutes. When the difference is negative it means that the flight has delayed its entrance in the FIR from what was initially planned; has been decided not using absolute values for better illustration of the system.

While Figure 2 shows an aggregated view for all Flights, it has been observed a clear difference between those flights departing from the FIR and those arriving to it, as in one case all the information is dependent on the FIR local system while in the other (incoming flights) the behavior is highly dependent on the information coming from collateral airspaces and systems. Thus, Figures 3 and 4 show separated views of predictability and accuracy for these two cases (47.8% of flights are departing from inside the considered FIR, while 52.2% are departing from outside). Every predictability and accuracy graph contains a vertical arrow, indicating the 80% value of accumulated flights, which has been observed as a representative value.

Figure 2. Analysis of all Peninsular FIR initial flight plan messages. (A) Distribution of predictability (in minutes) per initial flight plan message. (B) Distribution of accuracy per initial flight plan message. (C) Combined histogram of initial flight plan messages respect to both predictability and accuracy
Figure 3. Analysis of departing from Peninsular FIR initial flight plan messages. (A) Distribution of predictability per initial flight plan message. (B) Distribution of accuracy per initial flight plan message.

Figure 4. Analysis of arriving at Peninsular FIR initial flight plan messages. (A) Distribution of predictability per initial flight plan message. (B) Distribution of accuracy per initial flight plan message.

Figure 2(A) shows four peaks in the predictability graph, meaning that at those advance times (3, 11, 13.5 and 16 hours) a great number of initial flight plans enter the system. Additionally, looking at the 80% accumulated initial flight plans mark it can be determined that this percentage of flights is known by the system at approximately 5 hours of advance (this first result is considered as a relevant parameter for planning purposes).

The graph only includes initial flight plans with positive predictabilities, meaning that their initial flight plan is previous to their FIR entrance time. However, it has been found that 5.83% of initial flight analyzed showed negative predictability, that is, a later initial message time than FIR time, implying that the system knows about these flights after they enter the FIR, thereby not being able to consider them for planning (most of these flights are military flights).

Regarding accuracy graphs, it can be observed in Figure 2 (B) a clear Poisson distribution shape. In particular, is remarkable that the 0 minutes value (meaning total accuracy for the initial flight plan) shows a peak clearly over the Poisson distribution shape. These two facts imply useful results with a potential application for planning purposes and further data reliability assessment, line where the work of this study is ongoing on a new phase.

The disaggregated view for incoming and outgoing flights illustrated in Figures 3(A) and 4(A), regarding predictability, show remarkable differences for those flights departing from the FIR and those arriving or overflying the FIR. Thus, the departing flights (Figure 3(A)) show clear predictability peaks conforming a stepped accumulated initial flight plans graph, while the incoming initial flights plan, not dependant on the Spanish ATC platform show a more constant and linear behavior. It can be observed, additionally, that those flights arriving to the FIR have a better predictability value, as this is higher for the 80% of flights than the FIR departing flights one. It is obviously desirable that this predictability value becomes greater, in combination with proper accuracy (relationship to be explored in further analysis).

For accuracy graphs (Figures 3(B) and 4(B)), it can be observed that the initial flight plans values for the FIR departing flights are better, as the 80% accumulated flight plans mark is minor in absolute value (thereby more precise). Observation of the histogram shapes also shows that FIR departing flights are more right-shifted on the graph (positive values), meaning that typically flights tend to anticipate their entrance to the FIR from the initially planned one, while those flights arriving to the FIR after departing from airports outside the FIR tend to delay their effective entrance from that initially planned (left-shifted histogram shape of the graph). Both effects have an obvious impact on the system that will be studied.

Going deeper, the Barcelona FIR has been applied the same initial flight plan analysis, but only taking into account the flights with trajectory segments contained into the Barcelona FIR airspace. The purpose is to determine feasibility and potential benefits of doing a refined planning per airspace unit, as these units are the ones for which sector configurations and staffs plans apply, thereby the ones for which demand must be forecasted and assessed. Thus, both the predictability and accuracy formulas have been adapted to fit the particular Barcelona (LECB) FIR.
\[ \text{Predictability} = (\frac{\text{LECB}}{\text{entry time}})_{FP} - (\frac{\text{Log} \text{time}}{\text{entry time}})_{FP} \]

\[ \text{Accuracy} = (\frac{\text{LECB}}{\text{entry time}})_{FP} - (\frac{\text{LECB}}{\text{entry time}})_{FP} \]

As in the general case, the analysis have been done both aggregated (considering both FIR incoming and outgoing flights together, Figure 5) and separately, in Figures 6 and 7. In this last case the flights departing from the Barcelona FIR are 18,285 flights (42.7% of total), for 24,516 flights departing from outside the Barcelona FIR and arriving to it (57.3%).

The results observed in the specific case of Barcelona FIR have proven to be similar to those extracted of the overall Peninsular FIR, fact that needed to be checked for proper adapted planning. Thus, the initial flight plans predictability and accuracy graphs are very similar to those obtained in the global case.

---

**Figure 5.** Analysis of all FIR Barcelona initial flight plan messages. (A) Distribution predictability of per initial flight plan message. (B) Distribution of accuracy per initial flight plan message. (C) Combined histogram of initial flight plan messages respect to both predictability and accuracy.

**Figure 6.** Analysis of departing from FIR Barcelona initial flight plan messages. (A) Distribution of predictability per initial flight plan message. (B) Distribution of accuracy per initial flight plan message.

**Figure 7.** Analysis of arriving at FIR Barcelona initial flight plan messages. (A) Distribution of predictability per initial flight plan message. (B) Distribution of accuracy per initial flight plan message.
In this case the flights with negative predictability are 5.58% (slightly less than in the global FIR).

The main difference with the results obtained in the global FIR is the increased predictability of the flights departing from inside the FIR; thereby the accumulated 80% of the flights are in the system at a earlier position than in the global FIR.

In terms of accuracy, the Poisson-shaped distribution is again observed in this case; however, a relevant aspect is that the desirable total accuracy value (0 minutes in the graph) is no longer a noticeable peak, smoothing itself to the shape of the distribution. Is also noticeable the fact that for the flights reaching the FIR after departing outside it, the accuracy graph is more shifted to the right side, meaning that they get less delayed that in the global FIR (the ideal situation for this graph would be a delta function at 0 minutes, being better the system behavior the more similar the observed Poisson distribution becomes to it; however, the clear patterned shape becomes an useful tool for demand modelling and forecast reliability assessment).

Subsequent flight plan messages

Here, the previous analysis is extended to the first two update messages of the flight plan, in order to study the flight plan predictability and accuracy evolution during pre-flight phase. As only flight plan messages with positive predictability values are considered (meaning that they have yet not entered the FIR and are still considered “pre-flight” in the scope of this study). Thus, Figure 8 shows that only 43.32% of the second flight plan messages enter the system before the flights effectively enter the FIR (thereby the rest is characterized during all its pre-flight/planning phase only by its initial flight plan message, that don’t get updated in this phase); in the case of the third flight plans update this value is only 15.58%. In this graph the 100% of flights is not reached as it includes only flights with positive predictability values (initial flight plan message prior to their FIR entrance time).

Both the predictability and accuracy indicators for the second and third flight plan messages are represented in Figure 9, and calculated by the following formulas (particular extensions of the general formula applicable to every flight plan message):

\[
\text{Predictability} = (\text{HFIR})_{\text{second}} - (\text{HFIR})_{\text{last}}
\]

\[
\text{Accuracy} = (\text{HFIR})_{\text{second}} - (\text{HFIR})_{\text{last}}
\]

\[
\text{Predictability} = (\text{HFIR})_{\text{third}} - (\text{HFIR})_{\text{last}}
\]

\[
\text{Accuracy} = (\text{HFIR})_{\text{third}} - (\text{HFIR})_{\text{last}}
\]
As shown in Figure 9(A) and 10(A) second and third flight plan messages have lower predictability compared to the initial flight plan message. They also show that one hour before the FIR estimated entry time a peak of second and third flight plan messages enter the system, with much smaller values at the rest of time advances. In this case, the 80% accumulated flight plans vertical mark is noticeable lower than in the previous case, meaning that these updates are quite proximal to their FIR entrance time.

**Last Flight plan message with positive predictability**

After analyzing the first three messages individually, the additional goal is to calculate the quality of the last flight plan message for each flight before it enters the FIR (reduced to the first three, which cover the vast majority of flights in the considered traffic sample). As in the previous cases, both predictability and accuracy of these flight plan messages are represented in Figure 11.
Accuracy of all flight plan messages in function of different predictabilities

As described, the previous analysis covered the first three flight plan messages, considering only those with positive predictability, i.e. those whose log time is previous to their FIR estimated entry time. Additionally, it has been considered relevant to explore the accuracy of the last message modification for different predictabilities taken into account all the flight plan messages.

The aim is to analyze messages data quality depending on the planning time at different pre-flight horizons. Thus, different target predictabilities are set for which only last pre-flight flight plan message is taken into account.

Figures 12, 13, 14 and 15 show accuracy of last flight plan messages depending on its predictability, considering: (A)

Table 2. Percentage of Flight plans which have messages with determinate predictability and percentage of this messages with exact accuracy

all the Peninsular FIR messages, or (B) only flights with segments contained into the Barcelona FIR messages. As shown below, when message predictability increases the standard deviation of its accuracy with respect to total accuracy (0 minutes) also increases.
CONCLUSIONS
This paper addresses the study of real on-line flight plan data compared to real post-flight data, with the goal of analyzing accuracy and predictability values that eventually will merge into a demand forecast reliability index.

A whole set of real-time operational data has been analyzed, corresponding to the internally expected demand of the Spanish operational ATC Platform. No relevant difference has been found for the data analyzed for the whole Spanish peninsular FIR and for its subregion Barcelona FIR. This fact allows an early validation of the obtained results at ACC level (considering it as a representative minimum unit in terms of sector configuration and ATM resource allocation).

When comparing the FIR flight plans departing from the FIR with those arriving to the FIR from collateral regions, in the first case clear peaks on predictability respect to the stationary behavior of those reaching the FIR space are observed. Additionally, it has been demonstrated that FIR flight plans departing from the FIR are more accurate than those arriving to the FIR.

It is observed that the majority of flights get fully characterized in their demand forecast during all the pre-flight phase for their initial flight plan, with reasonable values of accuracy.

The study of the evolution of flight plans messages for a particular flight, with focus on the last pre-FIR message, has shown that accuracy slightly increases when predictability reduces. The main conclusion of the analysis done is that an approximated predictability value of between 5 and 10 hours is considered as good enough to have a reliable demand forecast as the number of flights is sufficient for a practical purpose (from the 50% to 80% of Flight Plans); thereby, this interval is a good time horizon to use this specific demand forecast. While different thresholds and their effect on particular systems will be later analyzed (together with a wider traffic sample), this analysis already allows to roughly determine the most suitable pre-flight phase to use it, already allowing some early reliability indication.

The results obtained clearly suggest that a relationship exists between accuracy and predictability that can be expressed as a reliability index. Additionally, the observed flight plans messages peaks and either linear or stepped accumulated graphic immediately apply to better demand modeling (e.g., in effectively combining at different time horizon historic data with real system data). Additional demand modeling benefits, also applicable, have been observed from the evolution and behavior of consecutive flight plans for a same flights.

The results obtained settle the basis for the ongoing research activity, which in further steps will advance to relate both predictability and accuracy concepts into a proposed reliability index, as well as in studying more aspects from the analyzed data.

REFERENCES
Flying by the Seat of their Pants: What can High Frequency Trading learn from aviation?

Gordon Baxter  
University of St Andrews  
St Andrews KY16 9SX, UK  
gordon.baxter@st-andrews.ac.uk

John Cartlidge  
University of Bristol  
Bristol, BS8 1UB, UK  
john@john-cartlidge.co.uk

ABSTRACT
As we build increasingly large scale systems (and systems of systems), the level of complexity is also rising. We still expect people to intervene when things go wrong, however, and to diagnose and fix the problems. Aviation has a history of developing systems with a very good safety record. Domains such as high frequency trading (HFT), however, have a much more chequered history. We note that there are several parallels that can be drawn between aviation and HFT. We highlight the ironies of automation that apply to HFT, before going on to identify several lessons that have been used to improve safety in aviation and show how they can be applied to increase the resilience of HFT in particular.

Author Keywords  
Ironies of automation; human-in-the-loop; high frequency trading; flash crash; socio-technical systems.

ACM Classification Keywords  
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
We are building ever larger scale IT based systems (and systems of systems) and these systems now permeate much of society. Many of these systems incorporate levels of complexity that make it difficult for an individual to get a good understanding of how they really work. Recent advances allow the technology to achieve levels of 99.999% reliability. These systems are invariably socio-technical systems, operated by teams of people, and we expect people to intervene and save the day when the technology fails.

This situation has persisted since we started introducing technology into the workplace. It is now 30 years since Bainbridge [2] published her paper “Ironies of Automation” which analysed the basic irony that as control systems get more advanced, the contribution of the human operator seems to become more important. Bainbridge’s work predates many significant technological developments—distributed systems, personal computers, the advent of the Internet and so on. We are still not giving the operators the resources to fulfil their role, so the ironies of automation still prevail, and we can still learn from the underlying arguments, which are all founded in psychology [4].

The characteristics of aviation and the process industries
Bainbridge focused her attention on monitoring and control activities in the process industries (chemical production, steel manufacturing and so on), and aviation to illustrate the problems. These domains are characterised by being complex and highly dynamic. Although it may be possible (in some cases, at least) to still manually control the processes involved, automation is almost invariably involved. The automation that is used, however, is not always transparent or predictable, which can give rise to automation surprises [35] where the operator’s start to ask questions such as “Why did it do that?” and “What is it doing now?” Indeed, the need to oversee the automation is probably best exemplified by aviation. To achieve the appropriate levels of skill required to fly an aircraft with its vast array of instrument displays, dials, switches and levers in the cockpit, pilots were typically taught that they had to aviate, navigate and communicate. The advent of the glass cockpit, where the functionality of many devices was incorporated into computer systems changed the nature of the job of flying an aircraft such that pilots are now taught to aviate, communicate and manage systems.

The changes in technology across complex, dynamic domains changed the role of the operator from one of manual control, to one of monitoring and supervisory control. The net effect is that people have become less directly involved in controlling processes as tasks have been automated.

At the point where automation started to become more widespread, many systems designers regarded the operators as a major source of variation and unpredictability in system performance. Their solution was to automate tasks and essentially remove the human operators from the system. This was in contradiction to the body of evidence showing the importance of the interdependencies between systems of systems, the level of complexity is also rising. We still expect people to intervene when things go wrong, however, and to diagnose and fix the problems. Aviation has a history of developing systems with a very good safety record. Domains such as high frequency trading (HFT), however, have a much more chequered history. We note that there are several parallels that can be drawn between aviation and HFT. We highlight the ironies of automation that apply to HFT, before going on to identify several lessons that have been used to improve safety in aviation and show how they can be applied to increase the resilience of HFT in particular.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ATACCS'2013, 28-30 May 2013, Naples, Italy. 
people and technology and how these are intrinsic to getting work done [e.g., 11, 13].

In most domains where systems are safety critical there are constraints and bottlenecks on how the system is allowed to perform. In process control, for example, the laws of chemistry and physics constrain how quickly some processes can happen, and in aviation, airports only have a limited number of runways, so a scheduling mechanism is used to maintain an efficient throughput of aircraft. The systems have checks and balances in place to assure safety to a very high level.

**From aviation to financial trading**

In the 30 years since Bainbridge published her work, technology has become ubiquitous. In some domains, such as financial trading, technology now pervades where it never existed before.

For hundreds of years, financial trading was a largely low-tech human activity, involving buying and selling face-to-face on “open-outcry” trading floors (and originally in London’s coffee shops) using little more technology than a pencil and paper. Then, on October 27th, 1986, the “Big Bang” deregulation of UK financial markets [e.g., 7] ushered in a move from open-outcry to screen based electronic trading on the London Stock Exchange (LSE). For the first time, geographically dispersed traders could now trade *en masse* from separate financial institutions. New, anonymous computer-based trading platforms enabled faster transaction speeds, better price discovery and increased liquidity. The markets flourished and huge profits were made. Trading technology was here to stay.

Although financial trading may seem a far cry from domains like aviation, and industrial process control, there are similarities. In some ways markets can be likened to airspace. In airspace, there are multiple airlines interacting and competing for the best slots to make money, whilst in the financial markets there is intense competition between high frequency traders looking to exploit fleeting arbitrage opportunities to make money. Each aircraft that flies through the airspace is monitored and controlled by human pilots whilst each trading algorithm deployed in electronic markets is also monitored and, to a lesser extent, controlled by human traders. In aviation there are regulations that try to ensure safety and efficiency, whilst still providing an environment in which the airlines can make money; in the markets there are also regulations in place to try to make sure that markets achieve efficiency and resilience, whilst allowing trading companies to make money [19].

In the past three years there have been several highly visible failures in financial trading. In the next section we detail one particular type of trading, high frequency trading (HFT), focusing on three significant failures in financial trading, to indicate the types of problems and the issues involved in HFT. We then describe the results that came out of the UK Foresight project which examined the future of automated trading in financial markets. This is followed by a consideration of more human factors related issues, describing the ironies of automation as they apply to financial trading. After looking at the lessons that HFT can learn from aviation, we conclude by re-emphasising the need to consider financial trading in the markets as a socio-technical system, learning lessons from other domains where similar problems have already been resolved.

**HIGH FREQUENCY TRADING**

In traditional financial markets, the success of a trader depended on their making timely decisions. These were based on their knowledge of market fundamentals and dynamics, and knowing when to continue to hold a particular position and when to get out. There is invariably a lot of information flowing in the markets, so it has always been difficult for a lone individual to successfully monitor and anticipate events, rather than just react to them as they happen.

After 1986’s Big Bang, further deregulation and technological innovation combined to radically change the landscape of financial trading beyond all recognition. Accelerated by the EU’s 2007 Markets in Financial Instruments Directive (MiFID)\(^1\), there has been a proliferation of Alternative Trading Systems (ATSs), including Multilateral Trading Facilities (MTFs) and Electronic Communication Networks (ECNs), enabling trading to take place away from the traditional exchanges. This has produced market fragmentation.

The introduction of technology made it possible to monitor larger amounts of information more quickly than people can, and provided the foundation for algorithmic and automated trading systems. These systems use software to automate some (and sometimes all) of the trading process. They were developed to assist and, in many cases, replace human traders. These systems allow decisions about buying and selling to be made more quickly (and automatically) to exploit fluctuations in markets and individual prices.

At the extreme end of automated trading lie HFT systems that habitually trade relatively small quantities of stocks and shares, often only holding positions for a fraction of a second. If the system can generate a net profit of a few pennies in that time, this can quickly lead to a steady stream of income by carrying out a large number of similar trades.

HFT systems are designed to exploit fleeting arbitrage opportunities that arise between market venues. Their main strategy depends on speed of execution: if another trader manages to execute first, an opportunity will often be lost. This competition has produced a race to zero [19] among HFTs as they try to minimise latency at all costs.

---

\(^1\) MiFID was proposed to offer new opportunities and innovation, but few anticipated how dramatically it would alter the landscape.
Consequently, HFTs utilise relatively simple/naïve strategies, because they cannot afford the time required to perform a series of complex calculations before they act.

In order to minimise the time it takes to execute, automatic trading systems are normally situated on servers that are physically located as close as possible to the digital stock exchange.

The majority of financial trading is now automated. It has been estimated that in the US markets HFT could yield an annual income of at least the order of $10bn [23], although this is currently quite small compared to the overall trading volume (which was about $50 trillion in 2008).

Whilst the human traders have not completely disappeared, they may now be based in offices situated across the globe. The role of the human trader, however, has been reduced in the same way as has happened in process control and aviation where the operators and pilots are now less directly involved in performing control actions. Nowadays traders are mainly concerned with setting trading strategies and monitoring their execution. Even in the relatively rare situations where the humans are still making decisions, the trades are still executed algorithmically. It is a fiercely competitive world, however, and in the time it takes a trade to execute, there is a risk that another algorithm may have identified that the trade is happening and intervene before the trade completes to make its own profit.

The naivety of HFTs combined with their immensely fast trading times can have profound effects. Here we describe three events to illustrate the deleterious dynamics that HFT can cause in the financial markets. These examples vary in scale of interaction between HFT firms: from the micro-scale of interaction between HFTs in an individual stock (Knight Capital’s “technology breakdown”); the mid-level interaction between HFTs in an individual stock (stock price “fractures”); and the macro-level multi-instrument, market-wide interactions (the “flash crash”). It is interesting to note that all of the events we describe bear the elements of the aviation automation surprises we described earlier.

**Knight Capital’s “Technology Breakdown”**
The Knight Capital Group is an American global financial services firm engaging in market making, electronic execution, and institutional sales and trading. In 2011, Knight ranked number 1 in secondary trading of US equities by share volume among all securities firms; and in the first three quarters of 2012 Knight’s US Equity Market Making traded an average of 128,000 shares per second.³

On 1st August 2012, Knight Capital started live trading using their new Retail Liquidity Provider (RLP) market making software on the NYSE. Immediately they started losing millions of dollars a minute. It was forty-five minutes before the software was stopped, by which point Knight had lost a total of $440 million [20]. The following day Knight’s own share price plummeted on the news, erasing 75% of Knight’s equity value. Within six months, a rescue deal was put together by a group of Wall Street firms to prevent Knight having to file for bankruptcy. The downfall of this highly successful HFT firm was entirely due to one disastrous autonomous technology breakdown.

While doubt remains as to the exact cause of Knight’s trading loss, Nanex Research’s [28] analysis offers the most compelling insight. By forensically analyzing millisecond trade data on the New York Stock Exchange (NYSE), Nanex Research demonstrated that there was a frenetic period where almost all trades alternated between buying at the offer (the lowest price offered by a seller) and then immediately selling at the bid (the highest price offered by a buyer), each time losing the difference in the bid-ask spread. “In the case of EXC [Exelon Corporation], that means losing about 15 cents on every pair of trades. Do that 40 times a second, 2400 times a minute, and you now have a system that’s very efficient at burning money” [28]. It appears that Knight had inadvertently deployed their test software as well as RLP!

The test software was designed to fire patterns of buy and sell orders at RLP inside a development platform, but was now doing just that on the live exchange using real money. Neither the traders at Knight, nor the RLP had any idea that anything was wrong because the test software was not designed to feedback any information about profit and loss. The two separate units of Knight software were both buying and selling without any idea of what the other was doing [28].

Alternative explanations for Knight’s trading loss include the suggestion that the trading malfunction involved Knight Capital buying $5 billion of stock in a trade that was intended to take place over five weeks but was actually executed in just 20 minutes [12]. Whatever the ultimate cause of Knight’s loss, one thing is certain: in the time it took for the HFT system’s error to be spotted and the algorithm pulled, it was already far too late for the firm to recover from the devastating consequences.

**Stock Price “Fractures”**
In February 2012, Johnson et al. [22] published a working paper that immediately received widespread media attention, including coverage in eFinancial-News [33], New

---

² Market makers provide liquidity to a market by issuing simultaneous quotes to buy and sell a financial instrument or commodity, with the hope of making a profit on the bid-ask spread: the difference between the buy and sell price.

³ http://www.knight.com/ourfirm/liquidity.asp
Scientists [17] and Wired [24]. Having analysed millisecond-by-millisecond stock-price movements between 2006 and 2011, Johnson et al. argued that there was evidence for a phase transition in the behaviour of financial markets at the sub-second time-scale. At the point of this transition, the market dynamics switch from a domain involving interactions among a mix of human traders and robot automated algorithmic trading systems, to a domain newly-identified by Johnson et al. in which the automated trading systems interact only among themselves, with no human traders involved. This abrupt system-wide transition from mixed-human-algorithm phase to a new all-algorithm phase has been named the “robot phase transition” [5].

At sub-second timescales, below the robot transition, the robot-only market exhibits “fractures”—ultra-fast swings in price—that are undesirable, little understood, and intriguingly appear to be linked to longer-term instability of the market as a whole. In particular, Johnson et al. [22] showed that the cumulative number of fractures observed across the entire market increased sharply during the period that the S&P500 fell most rapidly. Subsequently, as the index began to recover, fewer fractures were observed. This discovery has the potential for significant impact in the global financial markets. If the short-term micro-effects can indeed give some indication of longer-term macro-scale behaviour then it is possible that new methods for monitoring the stability of markets could be developed, offering early-warning systems for future flash-crashes. We return to this point in the discussion on ex post circuit breakers in the discussion of the findings of the Foresight report.

In March 2012, a series of laboratory-style experiments where human traders interacted with algorithmic trading agents (i.e., robots) in a minimal experimental model of an electronic financial exchange were conducted [5]. The aim was to see if correlates of the two regimes suggested by Johnson et al. occur in such laboratory conditions. Results indicated that when trading robots act on a super-human timescale of 100ms,

4 the market starts to fragment, with statistically fewer human-robot interactions that we would expect from a fully mixed market. In contrast, when robotic trader agents are slowed to a thinking-and-reaction speed similar to that of humans (of the order of hundreds of milliseconds, up to 10000ms), less fragmentation is observed. Cartlidge and Cliff [5] conclude that this is the first evidence for the robot transition occurring in controlled experimental financial market systems. This discovery and methodology opens the way for a principled research program to dynamically study the inter-relationships between the low level behaviour of automated trading systems and the global impact they have on market stability. Interestingly, an inadvertently introduced “spread jumping” bug that caused the robot agents to trade at prices far from equilibrium was introduced in the initial round of experiments. Despite the relative simplicity of the market, the bug (which had interesting parallels with the Knight Capital bug) was not spotted in real-time and was only discovered through extensive post-experimental analysis [5].

The “Flash Crash”

The Flash Crash happened in the USA on May 6th, 2010. The US’s Dow Jones Industrial Average Index (aka “the Dow”) was down by over 300 points on the day, but then fell a further 600 points between in the five minutes between 14:42 and 14:47, effectively wiping $1 trillion from the value of the market. In the subsequent 20-minute period, the Dow recovered most of the 600-point fall. The Flash Crash was the largest within day fall on the index but, perhaps more importantly, it was the unprecedented speed at which the crash occurred that was truly stunning.

That crashes occur in financial markets is self-evident. However, the nature of crashes, in particular the speed of crashes, has changed over time as technology has been introduced. For instance, in 1929, the well-documented Wall Street Crash was the tipping point that plunged the Western world into economic depression [16]. On “Black Thursday”, Oct 24th 1929, decades before the invention of the digital computer, the Dow opened at 305.85. By Nov 13th, it had fallen to 199; a 35% decrease in market value in just 3 weeks. Five decades later, on Oct 19th 1987, when electronic trading systems and computer-generated trading were still in their infancy, “Black Monday” saw the largest one-day decline in the Dow’s history (22%). The Flash Crash of May 6th 2010, in comparison, saw the Dow plummet 9% and then largely recover in the space of just 20 minutes. Clearly, as technology pervades, and markets become more dynamic, market crashes can occur at ever-greater speeds.

It was not just the speed of the flash crash that raised concern. This was a new kind of crash that had dynamics previously unseen. For instance, within the space of 14 seconds, more than 27,000 E-Mini S&P futures contracts were bought and sold; yet the aggregate net purchases was a mere 200. Ultra-fast algorithms had simply been passing contracts back and forth between themselves at lightning speed in what was described as a hot potato effect.

Contemporaneous with the Dow’s Flash Crash, individual stock prices behaved extremely erratically. Some stocks, like Accenture, plummeted to just 1 cent, while others, such as Sotheby’s, traded at $100,000. At this value, Sotheby’s had a net worth greater than the entire Chinese GDP! To compensate, the NYSE retrospectively cancelled all trades executed between 14:40-15:00 that were more than 60% away from last print at 14:40. This arbitrary cut-off point.
resulted in lots of arbitrary winners but, more importantly, lots of arbitrary losers. Trade dynamics such as these and the resulting ad-hoc interventions severely damaged investor confidence. Traders are much less likely to invest in a company’s stock if they cannot be sure whether the share price in 10 minutes will be 1 cent or $100,000; or if the exchange is likely to cancel trades after the deal has been made.

Furthermore, the Flash Crash has turned out to be a far from isolated incident. Since the Flash Crash, there have been repeated mini-flash crashes all over the world, such as the commodities crash of May 5th, 2011, where Brent Crude Oil suffered a record intraday 13% drop, Copper slid 5%, and Cotton fell 8% [34].

After 2010’s Flash Crash, it took the US Securities and Exchange Commission (SEC) and Commodity Futures Trading Commission (CFTC) almost five months to publish its official report [6]. They attributed the event to Waddell & Reed’s large mutual fund selling an extraordinarily large number (75,000) of E-mini S&P contracts which exhausted the number of available buyers. This was followed by HFTs aggressively selling, thereby exacerbating the effects of the large sale, and contributing to the sharp fall in prices. In other words, the CFTC/SEC blamed a combination of fat fingers (a trader hitting the wrong button) and HFTs.

The CFTC/SEC report has been widely condemned for its explanation of events. Nanex [29], for example, conclusively showed using millisecond tick data that the Waddell & Reed algorithm “was very well behaved; it was careful not to impact the market by selling at the bid, for example”. In simple terms, this means that the Waddell & Reed algorithm waited for buyers to accept its selling price each time it sold, rather than (as the CFTC/SEC suggested) aggressively dumping stock into the market at any price it could take. The mutual fund’s algorithm will have had some influence on the market, however, as it was targeting volume in its strategy [36]. Also, the claim that somebody inadvertently sold more stock in Proctor & Gamble than intended has been refuted, and the role of HFTs remains a matter of contention. Several alternative explanations for the Flash Crash have been advanced. Some are still the subject of debate, such as whether Waddle & Reed’s massive sale of 75,000 E-mini S&P contracts led to a major dislocation in the futures market too.

What is clear, however, is that prices only stabilised when the Chicago Mercantile Exchange’s Stop Logic Functionality was triggered to prevent a cascade of further falls in the price of E-mini S&P contracts. This injected a five second pause in trading, which was accompanied by a reduction in market pressures. A short time later, the price of the E-mini contracts began to recover, along with the Dow.

In the USA, trading curbs, known as circuit breakers, were subsequently introduced. These are designed to halt trading in any S&P 500 stock that fluctuates up or down by more than 10% within a five minute period. On the day of the Flash Crash, the process for breaking a trade was not clear to those traders in the market, and trades were only being halted when they were over 60% away from the reference price.

These new circuit breakers, which halt trade to provide a five-minute cooling off period, were initially only introduced for the S&P 500 stocks listed on the NYSE. They have subsequently been extended to other areas of the market, using trigger levels appropriate to that market. Although the circuit breakers may prevent re-occurrence of an identical Flash Crash, they do not eliminate the risk of other sorts of crashes, such as a Splash Crash, where a stock market event splashes out into the currency markets and beyond. This could happen because of the intricate interconnections between trades across markets as people try to keep their trading portfolio risk neutral by balancing it across sectors, markets, asset classes and so on [18].

FORESIGHT ANALYSIS OF THE FUTURE OF COMPUTER TRADING IN FINANCIAL MARKETS

A proposal to establish a project to look at the future of computer based trading in global financial markets was made in early 2010, before the Flash Crash (Cliff, personal communication). The UK Government Office for Science’s subsequently commissioned an international Foresight Project on The Future of Computer Trading in Financial Markets to look at two major challenges. The first was to explore the effects of the pace of technological change which, coupled to the continual rise in complexity of financial trading and markets makes it problematic to understand the role of HFT (and automated trading in general) on financial markets. The second was to create good evidence and sound analysis of the issues as a basis for informing the development of new regulations for the market.

After two years of extensive examination of evidence from over 20 countries, the final report was published in October 2012 [15]. The report explores how computer generated trading in financial markets will evolve over the next 10 years, using independent academic analysis of the evidence on the actual and potential effects of computer-based trading on financial markets.

Computer trading has transformed the way financial markets operate. Today, over one-third of UK based equity trading is HFT. In the US it may be as high as 60% or more. HFT has been implicated by some as a contributory factor in the Flash Crash, and in other failures as noted above.

The Foresight project found evidence that computer based trading and HFT has had several beneficial effects on financial markets. Firstly, there has been a positive contribution to liquidity, as measured by bid-ask spreads: the difference between the lowest price a trader is willing to sell and the highest price a trader is willing to buy.
Secondly, due to increased market venue competition and greater liquidity, transaction costs for both retail and institutional investors have reduced. Finally, there is no direct evidence that computer based trading and HFT has increased volatility or market abuse.

In specific circumstances such as the Flash Crash, however, it was noted that HFT can have negative effects on the markets. In periods of uncertainty the need for liquidity, which is one of the roles of the market makers, can be critical. HFT market makers, however, tend to leave the market, leading to a disappearance of liquidity, making the situation even more uncertain [36]. Furthermore, self-reinforcing feedback loops can amplify risks and lead to financial instability. The Foresight report proposes that mechanisms for managing and modifying potential adverse effects of computer based trading and HFT should be assessed and introduced. The mechanisms with the strongest supporting evidence and weakest opposing evidence include: (i) the introduction of coordinated circuit breakers; (ii) a coordination of tick sizes across venues; and (iii) market wide standards including coordinated, synchronized and accurate timestamps across multiple trading venues.

**Co-ordinated Circuit Breakers**

Circuit breakers are designed to temporarily halt trading, thus attempting to restore order in the market by dampening feedback loops to reduce further adverse movement. The breakers can be implemented in two ways: *ex post* and *ex ante*. *Ex post* circuit breakers trigger when a share price has fluctuated above or below a predefined safe threshold. These mechanisms monitor simple price data and activate only *after* the share price has moved out of bounds.

In contrast, *ex ante* circuit breakers are designed to halt trading *before* things go bad. These preventative measures use metrics other than price to monitor the market for precursory indications that instability is more likely to occur. Such circuit breakers can then warn regulators, venues and participants in advance to take appropriate action. One such metric is Easley et al.’s [10] Volume-synchronised Probability of Informed trading (VPIN™) flow toxicity metric. VPIN provides an estimate of the probability of informed trading based on volume imbalance and trade intensity. The value of VPIN was extremely high (suggesting low liquidity) in the run up to the Flash Crash. Easley et al. suggest that VPIN could be used: (i) as an *ex ante* indicator to warn about impending volatility/crashes; and (ii) as a tradable index (like the Chicago Board Options Exchange Market Volatility Index, VIX™) to enable HFT firms (liquidity providers) to hedge their risk as VPIN accumulates before a crash.

Irrespective of whether circuit breakers are *ex ante* or *ex post*, it is critical that they are harmonized across trading venues. If they are not, traders could simply switch to another trading venue when one venue gets halted.

**Co-ordinated Tick Sizes**

The tick size is the smallest price increment allowable at a trading venue. For instance, if the tick size is 10 cents and the current best bid (highest offer to buy) is €4.50, then the minimum price a buy order can have to post a new best bid is €4.60. Hence, the smaller the tick size, the easier it is (the more opportunities there are) to narrow the spread; i.e., to place a new buy order that is higher than the current best bid or a sell order that is lower than the current best ask. Smaller tick sizes offer more trading flexibility and are thus very attractive to HFT. For this reason, competition between trading venues to encourage HFT participation has led to an arms race between venues offering ever-smaller tick sizes.

Identifying the right tick size involves making a trade-off between two opposing forces, however. On the one hand, a coarser grained tick size offers more incentive for investors to place limit orders—orders to buy or sell at a limit price, i.e., buy at the limit price or lower, or sell at the limit price or higher—thereby boosting the liquidity displayed in a limit order book [1]. The coarser tick leads to a wider minimum bid-ask spread. This makes market making more attractive by increasing its profitability, which should increase liquidity as the number of market makers rises. On the other hand, higher minimum bid-ask spreads raise investors’ transactions costs, which leads to reduced trading and a corresponding reduction in liquidity.

As with circuit breakers, the Foresight report suggests that there should be a policy to harmonize the tick size across venues. If they are not, traders could simply switch venues with smaller tick sizes to reduce costs.

**Co-ordinated Market-Wide Standards**

The Foresight report makes the case for market-wide standards. These include the need for coordinated, synchronized and accurate timestamps across multiple trading venues.

In addition the report notes the need for accurate, reliable data in order to better understand the effects of computer based and HFT and hopefully also prevent further adverse events. It therefore calls for the introduction of a European financial datacenter. This would be responsible for receiving, warehousing and repurposing financial data across all primary European markets.

**THE IRONIES OF AUTOMATION IN FINANCIAL TRADING**

As in other domains, automation has changed the role of the human (traders), leaving them with two main types of task. The first is to configure algorithms, monitor trades and evaluate results. The real problem here is that it typically takes a human about 150-200ms to respond to a simple stimulus such as a sound or a light. Given that the lower limit for trade execution times is currently around 10µs [19], this means that the system could have made tens of thousands more trades before the trader can respond.
The traders’ skills for controlling how trades take place are likely to be out of date as a combination of erosion through lack of practice and changes in the nature of trading across several exchanges. Given that the trading systems are most likely to fail in unexpected situations, the traders may have to perform specialised, rarely (and possibly never before) used actions to regain control. In other words, the operators require more skill and need time and resources in order to work out what to do, possibly from first principles.

The second type of task is diagnosing problems with the systems, and determining how to fix them. This is particularly important, given that other systems will attempt to exploit these problems to generate a profit. Diagnosing and fixing the problems requires a combination of cognitive skills, which Bainbridge [2] categorised as long term knowledge and working storage. As long as the traders have a detailed, up to date understanding of the systems they are controlling they may be able to develop novel strategies to deal with new situations as they arise. The context in which the traders make decisions will be encapsulated in a mental model [26], which is updated as the situation changes. Since the traders are usually no longer involved in controlling the trades, however, it becomes harder both to develop and maintain their mental models, and the less they use their knowledge, the harder it becomes to retrieve. So any interventions will often be based on a minimal amount of information until they have had the chance to investigate further, to update their mental model, and to consider the available options.

If the traders are reduced to simply monitoring what the systems are doing, this creates another type of problem. For the most part, and under normal market conditions, the system will run smoothly and predictably. When the information the traders are watching is more or less unchanging, however, they are likely to have problems maintaining visual attention for more than 30 minutes. As their visual attention fades, it becomes harder to detect any visual anomalies. Automated alarms may help, but then the issue of who monitors the alarms arises. One of the classic ironies of automation is that the human has to monitor the system to make sure that it is working correctly, when the whole point of introducing the automation was because it was believed that it would do a better job than the human. Having the traders monitor the automation introduces two problems.

The first is that the trader will require specialised knowledge—aquired through either training, or dedicated displays—in order to be able to monitor the system effectively. The second is that the systems are processing more information at a faster rate than the traders can in order to make decisions. It therefore becomes impossible for the trader to adequately track the system’s behaviour in real time. Instead, they will only be able to check the system at a higher level of abstraction and at a potentially considerable time lag.

LESSONS FOR HFT FROM AVIATION

The ironies of automation in financial trading can be overcome, but the solutions—like those for other domains such as aviation—are, as Bainbridge [2] acknowledged, highly dependent on factors such as the size, complexity and speed of the system. We believe that HFT, where the solutions are dependent on the trader’s skills and abilities, can learn something from aviation, in particular. Somewhat ironically, several of the solutions are technology based. We fully accept, however, that HFT should not just blindly follow aviation and that great care is needed in finding appropriate lessons and applying them. We are aware of the shortcomings of following checklists, for example, which can make a bad situation worse, as happened in the Swissair Flight 111 air accident [9]. Like aviation, HFT is really a system of systems, so there are potentially lessons to be learned at several levels. Below we highlight some of the lessons we have identified so far.

Lessons from systems of systems

In aviation the way that problems are dealt with requires decision making on several levels. The technology may decide that a faulty piece of equipment should be shut down, or the decision could be made by the pilots. The decision to allow an aircraft that has declared an emergency to land out of turn at an airport requires much more manual co-ordination and intervention between the flight crew and air traffic control. In HFT the decision to shut down a single system down after a failure could be taken automatically, but the decision to shut down one or more trading venues, or even close the markets should require some degree of manual control and co-ordination between traders, regulators and those operating the exchanges. Indeed, the SEC has recently called for the introduction of kill switches, which may reside at the exchanges, to instantly disable an errant trading system [8].

One of the reasons that air transport works on a global basis is because of bottlenecks in the system. Aircraft regularly fly through the airspace of many countries en route from one airport to another without incident. They have to safely end up at airports, however, and how they do so depends on co-operation and co-ordination between pilots, air traffic control, airlines and the regulators, including the independence of ATC from the airlines, and regulations that govern the vertical and horizontal separation of aircraft. Even under free flight conditions, the aircraft still have to form an orderly queue as they approach an airport before they can land. In HFT, however, the traders are the people who oversee how trades progress, and have a vested interest in exploiting any anomalies they may spot. The speed of the trades makes it impossible for the traders to interact with the trading systems in real time. This means that the traders cannot detect a single failure until the effects have become large enough to be noticeable by a human. In the time it takes to diagnose and repair the failure, however, many more trades may have been executed, and possibly have
exploited that failure. Haldane [19] suggests the possibility of imposing minimum resting periods on all trades, which would place a lower level time limit on each trade, and would reintroduce an element of collaboration and communication into the trading process. He argues that this would help restore the balance between market efficiency and market stability; to date regulatory changes have tended to favour market efficiency.

**Lessons from systems monitoring**

Part of the burden for handling some aspects of aviation safety and efficiency has been passed to the automation. The detection of other air traffic in the aircraft’s vicinity, for example, is nowadays handled by the aircraft’s Traffic Collision Avoidance System which automatically generates alarms on several levels. If technology is to provide at least part of the solution within HFT, however, it becomes even more important that any technology failures are immediately obvious to the traders and the markets as a whole. If a system is frequently generating alarms, for example, then the traders will become quite experienced at routinely handling them. This highlights Bainbridge’s final irony which is that the best automated systems which rarely require manual intervention require the biggest investment in training to ensure that the people can appropriately respond when things do go wrong.

In air traffic control, the vigilance problem is dealt with by only allowing controllers to spend limited time at their displays overseeing a sector of airspace. Although this idea could be applied to HFT, it would not overcome the fact that the trades are happening at a rate faster that the traders can track. So they could only monitor trading at a higher level of abstraction.

Accident investigation in aviation relies on forensic evidence from the aircraft’s cockpit voice, and flight data recorders. These are used to piece together what happened in the aftermath of the accident. Up until very recently the SEC simply did not have access to enough data to be able to forensically examine why crashes were happening in the market. They have now employed technology from one of the HFT firms to address this problem. Up until now, the SEC has relied on the official trading record, referred to as the consolidated tape, which details the prices of all trades made on any of the US’s stock exchanges. The sophisticated trading firms do not wait the extra milliseconds for the consolidated tape to be released but instead buy the data directly from the exchanges. This allows them to build their own record before the official record is released, and it is more comprehensive because it includes details of orders that were submitted but never completed. Even with the new stream of information from Tradeworx, however, the SEC will still not have a completely comprehensive picture of the market. For example, it will not have access to data for trades executed in dark pools—trading venues that do not require adherence to the reporting rules used by the public exchanges [32]. Furthermore, the details of who is placing the trades will only become available once the consolidated audit trail is introduced in the next few years.

**Lessons from regulation and standards**

The need for effective regulations and regulators is critical to aviation. The role of regulators like the Civil Aviation Authority in the UK, for example, includes explicit objectives addressing safety and efficiency. In HFT the role of the regulators like the SEC focuses on protecting investors, but without explicitly mentioning safety issues. Nanex Research [27] recently highlighted that the regulators appeared not to be enforcing Regulation National Market System (NMS) and subsequently suggested that rather than being enforced, it had been rescinded [30]. Regulation NMS covers the issue of the National Best Bid or Offer which is supposed to assure investors that they are getting the best price for any stocks they buy and sell. The emphasis on speed at all costs in automated trading has made it virtually impossible to show a definitive audit trail for whether an investor received the best price. The regulations also exist to prevent quotes being generated to manipulate other traders in the market—so called quote stuffing—but are not being applied. Nanex Research suggests that quotes should have a minimum lifespan of 50ms.

There are recognised standards for developing software for aviation. RTCA/DO-178B (version B) (also known as EUROCAE ED-12B), is the de facto standard used by regulators like the FAA to decide whether software will perform reliably in an airborne environment. This standard, which was published in 1992, provides guidelines for assuring that the software and equipment will perform its intended function with a level of safety that is compliant with airworthiness requirements. In HFT, the overall safety of the market has effectively been ignored, with traders switching from one algorithmic trader to another to exploit anomalies as they arise in the live market regardless of the effect they may have on that market. The SEC has recently called for new regulations on software testing and reliability after the Knight Capital fiasco, including new software standards [31].

In aviation, there is often a long lead time between the conception of a piece of equipment and its being introduced into the industry and made mandatory. Rigorous testing and certification are required before the new equipment is deployed. S-mode datalink, for example, was originally conceived in 1975. It has only been mandatory for aircraft flying under visual flight rules in Europe since 2005, however [3]. In stark contrast, the lifetime of trading algorithms is very short, with traders typically introducing new algorithms every few weeks. A system of governance requiring evidence of testing, or a system of certification would help to regulate the appearance of rogue algorithmic traders in the markets.
Lessons from organisational learning
The aviation industry has generally been very good at managing the effects of the ongoing introduction of automation. Many of the important issues associated with glass cockpits in the mid 1990s were encapsulated by the FAA’s Human Factors team’s report, *The interfaces between flightcrews and modern flightdeck systems* [14], for example. Flight deck technology has evolved considerably in the intervening period, but the skills needed to deal with the changes in technology have not. At the same time manual skills have been eroded as the pilots rely increasingly on the technology to fly the aircraft, making it harder for pilots to know how to (and be able to) recover from a stall, and carry out a go-around in the event of a missed approach when coming into land. Even though regulations for recurrent training of pilots exist, there have been recent calls for changes to the recurrent training regulations in order to reconcile pilot skills with the newer technologies [25]. Manual control skills, such as being able to recover from a stall, and carrying out a go-around in the event of a missed approach are being eroded. These examples show that the regulators need to self-monitor, and regularly revisit the regulations to learn which ones are still applicable, and whether they are still being appropriately policed and enforced.

In the aftermath of an aviation accident, there is invariably an accident investigation, carried out by an agency that is independent of the regulator. In the UK, for example, the Air Accident Investigation Board (part of the Department for Transport) would produce a report which it would send to the Civil Aviation Authority which regulates aviation and is a public company, rather than a government agency. The accident report produces clear and timely findings, identifying lessons that can be learned, and where changes may be needed to improve safety. In most cases there is general agreement with the findings, and where there is disagreement, it is often a matter of degree. In contrast, the CFTC/SEC report on the Flash Crash was widely condemned for being late and inaccurate. The SEC is now considering the need for external retrospective assessment. It has been noted that “[w]ithout some assessment ... we may never know what went wrong—and we run the risk of trying to prevent the wrong problem” [21].

SUMMARY
On the face of it, high frequency trading and aviation could hardly be more different. Both are striving to achieve resilience--in the markets, and in air transportation respectively-- whilst still allowing companies to make a profit. We have, however, identified several underlying similarities in the ways that HFT and aviation work.

Up until the Flash Crash in 2010, HFT emphasised profits over resilience but since then there has been an increased focus on improving resilience. Based on the identified similarities between HFT and aviation we have highlighted several lessons where we believe that HFT can learn from aviation in the areas of technology, regulation and software development. We regard these lessons as the start of the process of improving resilience in the HFT markets (and potentially, beyond). Our intention is to build up a comprehensive list of lessons that can be used to improve and maintain the resilience in the HFT markets.

ACKNOWLEDGMENTS
Special thanks go to Dave Cliff for discussion of the major findings of the Foresight report. John Cartlidge was supported by EPSRC grant number EP/H042644/1. Gordon Baxter was supported by funding from the UK’s EPSRC as part of the Large Scale Complex IT Systems (LSCITS) project (Grant no.: EP/F001096/1).

REFERENCES


21. Heusser, M. Software testing: Lessons learned from the Knight Capital fiasco CIO.


24. Keim, B. Nanosecond trading could make markets go haywire Wired Science.


27. Nanex. Is the National Best Bid or Offer (NBBO) being ignored? Nanex.


30. Nanex. Reg NMS has been rescinded! Nanex.


32. Popper, N., and Protess, B. To regulate rapid traders, SEC turns to one of them The New York Times.


Model-Based Dynamic Distribution of User Interfaces of Critical Interactive Systems

David Navarre, Célia Martinie, Philippe Palanque
Institute of Research in Informatics of Toulouse
University of Toulouse
Interactive Critical Systems (ICS) team
118, route de Narbonne
31042 Toulouse Cedex 9, France
{martinie, navarre, palanque}@irit.fr

ABSTRACT
Evolution in the context of use requires evolutions in the user interfaces even when they are currently used by operators. This paper proposes a model-based approach to support proactive management of context of use evolutions. By proactive management we mean mechanisms in place to plan and implement evolutions and adaptations of the entire user interface (including behaviour) in a generic way. This generic model-based approach is exemplified on a safety critical system from the space domain. It presents how the new user interfaces can be generated at runtime to provide a new user interface gathering in a single place all the information required to perform the task. These user interfaces have to be generated at runtime as new procedures (i.e. sequences of operations to be executed in a semi-autonomous way) can be defined by operators at any time in order to react to adverse events and to keep the space system in operation. Such contextual, activity-related user interfaces complement the original user interfaces designed for operating the command and control system. The resulting user interface thus corresponds to a distribution of user interfaces in a focus + context way improving usability increasing efficiency and effectiveness.

Author Keywords
Model-Based approaches, formal description techniques, interactive software engineering, automation, distributed user interfaces, dynamic reconfiguration of user interfaces.

ACM Classification Keywords
D.2.7 Distribution, Maintenance, and Enhancement, D.2.11 [Software] Software Architectures - Languages (e.g., description, interconnection, definition), H.5 [Information Systems] Information Interfaces and Presentation

General Terms
Design, Automation, Reliability, Human Factors.

INTRODUCTION
In the early days, the basic design rationale for User Interfaces for control rooms was to assign one display to each component to be monitored and one physical input to each command to be sent to one component of the controlled system. This resulted in very large command and control rooms being rather easy to design and build but rather cumbersome to operate. Such difficulties have been largely studied and reported in scientific work looking at the design aspects (e.g. [31] and [9]), at the implication on operations (see typical image of controls customization where operators add beer labels on top of control levers p. 95 [27] from [32]) and safety when incident or accident occurred ([30] p 193 on Chernobyl accident). In order to overcome such constraints, design drivers for command and control systems have been targeting at concentration and integration of both displays and controls. In several domains such as control rooms and aviation, such concentration was achieved by adding computing resources for concentrating data from multiple displays into a single (or sometimes several in case of large and complex systems) display unit. In aeronautics such concentration of display is known under the notion of “glass cockpit” as computer screens were replacing previous analog displays.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1 By concentration we refer here to the terms coined by J. Vanderdonckt in [35]
loss of situational awareness, or skill degradation, whereas not enough automation can lead to an unmanageable, unsafe or problematic workload [29]. This is the reason why, for instance, the SESAR (Single European Sky ATM Research) programme targets higher levels of automation in aviation in order to improve safety and efficiency of ATM operations.

Work on function allocation such as the ones described in [11] or [4] aim at supporting the design of automation and more precisely at identifying and assessing candidate functions to be automated. Beyond that, if the use of the system is highly dynamic i.e. evolves regularly (for instance in order to handle unexpected adverse events such as malfunctions, faults, malicious attacks …), there is a need for dedicated support to anticipating evolutions and for providing adequate solutions. This paper proposes a model-based tool-supported approach for the design and development of distributed user interfaces in the context of highly dynamic complex systems requiring repetitive and systematic activities to be allocated to the system in order to allow operators to be focussing on more analysis and decision related tasks. This approach embeds automatic generation of distributed user interfaces allowing operators to monitor the execution of semi-autonomous procedures. Next section presents with more details the context that has been introduced above. The following section presents the process associated with the approach exhibiting why there is a need of distributing the operators’ user interfaces in two different parts, one being the standard command and control interface and the other one being an additional UI generated for handling a dedicated adverse event. The last section presents a case study about satellite ground segments applying step by step the approach. Finally a conclusion and directions for future work are presented.

AUTOMATION IN THE CONTEXT OF COMPLEX SYSTEMS

There are many different levels for implementing design decisions in order to include autonomous behaviors in a computing system. The first one (static level) consists in defining and designing the allocation at design time and to design and build the interactive system according to this allocation of functions. This is for instance the case in automotive industry with the ABS (anti-lock braking system). This autonomous system prevents vehicles wheel from blocking while the driver is breaking. Even though the autonomous system is triggered by the user, its behavior is “hard coded” and cannot be altered. The second one (dynamic execution level) consists in designing and defining flexible and redundant functions as in the aeronautics domain with the auto pilot. All the functions that are available in that autonomous system (such as climbing to a certain altitude) can also be performed manually by the pilot. The decision to allocate the execution of the function to the autonomous system remains in the hand of the user. The last level (dynamic execution and definition level) allows the user to define the behavior of the automation and also to decide when such autonomous behavior will be executed. Such level corresponds for instance to the definition and execution of macros in Microsoft Excel or the text styles in Microsoft Word.

The current paper addresses the last level (presented above) applied to command and control systems for satellite control rooms. Indeed, in case of malfunction the operator is required to define a procedure in charge of solving the identified problem. Such procedures are then tested and executed either in an autonomous or manual way. However, even in the case of autonomous execution some information might be required from the operator to complete the execution. Such information can be values of some parameters (presented on some display units) of the satellite or go/no go that contacted experts in the domain of the failure (e.g. engines, electricity …) have provided to the operator. One of the issues related to that problem is that the information required from the operator can be distributed amongst many displays making this activity cumbersome, time consuming or even error-prone. The objective of this research work is to exploit the content of the procedure defined by the operator to generate and additional user interface dedicated to the management of the procedure. This user interface gathers all the information that has to be checked and provided by the operator throughout the execution of the procedure. How such user interfaces can be generated from the definition of the procedure is presented in details in the following section. It is important to note that the point is not here to modify the existing user interface of the application but to generate an additional, contextual user interface. This prevents difficulties that may occur and which are known under the term “automation surprises” [28] if the routine interface was unpredictably altered by the generation process. Indeed, currently the new interface generated can be simply ignored, at no cost, by the operators.

USER INTERFACE GENERATION FOR DYNAMIC PARTLY AUTOMATED SYSTEM

As presented in the previous section, in the area of complex command and control systems, some of the user tasks and activities cannot be identified beforehand i.e. at design time. In addition to that issue, these tasks can be complex and/or inadequate for a human being (requiring for instance, management of a large amount of information, execution of multiple commands under strong temporal constraints, …). Such tasks are thus good candidates for delegation to an autonomous sub-system. In order to address those issues there is a need to provide operators with meta-level systems able to combine multiple commands and to delegate their execution to an

---

2 Air Traffic Management
autonomous agent. The design of this part of the partly-autonomous command and control system requires the same level of reliability and usability as the rest of the application. While the reliability aspects of user interfaces can be addressed using standard dependability and fault-tolerance techniques such as the COM/MON architecture proposed by [14] and applied/extended to user interfaces in interactive cockpits [33], the usability aspects have to be addressed according to the work done in the area of automatic generation of User Interfaces as described in [34] or more recently in [26].

Several model-based approaches and toolkits aim at designing and implementing Distributed User Interfaces (DUIs) reconfigurable at runtime. Fröberg et al [10] present a framework called Marve in order to support graphical components reallocation across platform. Their work particularly focuses on event communication structure management. Melchior et al. [21] introduce a toolkit to deploy DUIs and then a framework based on state transition diagrams to represent distribution states of a DUI [22]. Kjeldsen et al. [13] also present a system architecture for widget interaction reconfiguration on planar surfaces. Another set of contributions dealing with dynamic reconfiguration of distributed user interfaces layout are based on the CAMELEON framework [5]. Manca and Paterno present a dialog model description language which aims at supporting dynamic distribution of user interfaces elements across various devices [15]. Other contributions deal with runtime architectures. Clerckx et al. [6] propose a design process and runtime architecture supporting partial dynamic redistribution of the user interface at runtime.

These contributions do not take into account or partially (in the case of state transition diagrams to represent the distribution states [21]) the behavioural part of the distributed interactive applications. This is a critical aspect when dealing with command and control of safety critical systems which might lead to deadlocks. We previously addressed that aspect by proposing fault-tolerant architectures dedicated to the dynamic reconfiguration of user interfaces in the context of cockpits of large civil aircrafts. This reconfiguration supports distribution as well as relocation of user interfaces of critical applications to other displays unit when the default one is faulty [24] and [23].

AN AUTOMATED DESIGN PROCESS FOR GENERATING INTERFACES FOR PARTLY AUTOMATED SYSTEMS

Generation of user interfaces can be envisioned if behavioural description of the automation is available and if a generic mechanism for distribution is available. However, such generation of the user interface must not have a negative impact on monitoring activities, so distribution to another display and/or to another window is required. This distribution allows decoupling the introduction of new interfaces (generated) from the set of existing ones. The design process presented in this section aims at guaranteeing the continuity of operation so that the predefined set of interfaces for monitoring and control is not altered by the generated ones.

Overview of the process

Figure 1 presents the generic process involving dynamic generation of part of the User Interface. That Figure is split in three parts.

![Figure 1. General overview of the approach](image)

The first part (called Design and development time) on top corresponds to the design and development of the User Interface that is done following a classical user-centered development process. The only difference is located in the phase called (Design Automation (function allocation as defined in [4])) dedicated to the attribution of functions either to the partly-autonomous system or to the operator. Of course the description of the process remains on purpose abstract not even showing the iterations as we only highlight here the main principles. The interested reader can find a more complete and precise description of such a user-centered design process in [20]. This part is split into two threads of developments represented by the two swim lines. The right-hand side corresponds to the standard development aiming at producing a usable user interface.

The underlying concept behind this process is that there are two types of user interfaces that will be used by the operator. A generic user interface allowing the operator to perform the main tasks assigned to him/her and a set of specific user interfaces aiming at supporting specific
activities defined by procedures. The generic user interface corresponds to the UI of the command and control system allowing managing the entire system while the specific UI are dedicated to procedure (that might have been defined after the UI of the command and control system has been finalized). This process is rather generic in critical systems where modification of the command and control systems might involve time and resource consuming activities such as certification by external authorities.

The other two boxes in Figure 1 correspond to the design and development of the specific user interfaces dedicated to the management of specific procedures. The one on the right-hand side corresponds to procedures that have been identified during the design phases of the command and control system and follow the standard user-centered design process. The one at the bottom of Figure 1 corresponds to the generation of a user interface while the command and control system is in operation. Indeed, in many cases e.g. change is usage processes or handling of unexpected adverse events not envisioned during the design phases of the command and control system. The resulting user interface of the command and control system is thus the sum of these 3 interfaces. It is important to note that the generated part does not replace the existing one but is proposed as a kind of contextual help to the operators.

Figure 2. Generic generation process for the user interface of procedures

Distribution and generation
Figure 2 refines the user interface generation process presented at the bottom of Figure 1. It starts with a manual activity carried out by the operator consisting at modifying an existing (or potentially creating a new one).

- To describe the procedure (as explained with more details in the case study section) operators are provided with behavioral description languages such as YAWL [12]. Our process is based on another language called ICOs (Interactive Cooperative Objects) [25] which combines Petri nets and Object-Oriented constructs allowing manipulating values within the Petri net-based behavioral description. Beyond that, activation and rendering functions in ICO make it possible to connect this behavioral description to the graphical user interface it describes. This activity is represented as a manual and automated process as it is performed using dedicated editing tool. The ICO description of the procedure provides the grounding of the behavioral part of the user interface that will be generated.

- To describe the operators’ activities that cannot be inserted in ICO models, HAMSTERS (Human-centred Assessment and Modelling to Support Task Engineering for Resilient Systems) notation is used. HAMSTERS is a task modelling notation designed for representing the decomposition of human goals into activities (perceptive, cognitive, motor, interactive…).

- The ICO procedure is then automatically analyzed using a Petri net pattern detector based on a collection of patterns descriptions. These patterns correspond to the basic bricks that constitute the procedure behavior and depend on the application it is related to. The
product of this pattern extraction is a logical structure of the targeted application as a collection of instantiated patterns (an instantiated pattern contains attributes that directly relate it to the part of the ICO description it corresponds to). As within our generation process this description is only transient, we do not handle it as a model per se, even if it would be possible.

- For each of these instantiated patterns, the UI generation phase associates a concrete component using a predefined mapping and these components are then composed within a generic graphical canvas, creating a default layout of these components. The production of this phase is a model that does not describe the behaviour of the generated application (the behaviour being provided by the ICO model in the next step). This is not presented on Figure 2 but the components, the generic canvas and the produced application are customizable, allowing a fine tuning of the produced user interface. This would be needed for instance when maintenance is performed of the application thus going back to the design process.
- Lastly, the generated model and the ICO procedure are put together to provide the final interactive user interface (using the activation function and the rendering function of ICO introduced above).

This generation process is instantiated and illustrated on a case study in the following section.

Figure 3. Examples of textual and graphical synoptics

Relationship with Previous Work
As presented above this work build on top of previous work we have done on the formal description and prototyping of user interfaces. While that previous work was focussing on supporting developers in a) identifying users activities and goals (using the notation HAMSTERS [17]) b) describing in a complete and unambiguous way both the interface and the associated interaction techniques using the ICO formal description technique [25] c) a set of case tools called CIRCUS integrating HAMSTERS case tool and ICO case tool called PetShop [1].

HAMSTERS\(^3\) is a tool-supported graphical task-modeling notation aiming at representing human activities in a hierarchical and ordered way. Goals can be decomposed into sub-goals, which can in turn be decomposed into activities, and the output of this decomposition is a graphical tree of nodes. Nodes can be tasks or temporal operators.

The ICO formalism is a formal description technique dedicated to the specification of interactive systems [25]. It uses concepts borrowed from the object-oriented approach (dynamic instantiation, classification, encapsulation, inheritance, client/server relationship) to describe the structural or static aspects of systems, and uses high-level Petri nets to describe their dynamic behavioral aspects.

As this paper only focusses on the process and the benefits of generating specific user interface while the system in under use, next section will not present detailed models of the case study.

\(^3\)http://www.irit.fr/recherches/ICS/softwares/hamsters/index.html
CASE STUDY
The example presented in this section belongs to the category of complex command and control systems from the space domain. Such interactive systems are less time constrained than other ones (such as aircraft cockpits). Beyond that, such systems are less safety critical (the only possible safety issue would correspond to a spacecraft falling on earth and injuring people). However, the potential cost of a failure is far beyond the development cost of these systems making them belong to the category of critical systems. This case study aims at highlighting how to automate the distribution of interface for the operators, providing a particular focus on the design of procedures and the generation of interactive means to control the automation. These concepts as well as the development process presented above have been applied to satellite ground segment applications within the context of the ALDABRA (Architecture and Language for Dynamic And Behaviorally Rich interactive Application) Research & Technology project.

PICARD ground segment overview
The PICARD satellite mission is dedicated to solar activity observation. Operators are in charge of two main activities: observing periodically the vital parameters of the satellite and performing maintenance operations when a failure occurs. They may have to lead concurrent activities such as monitoring satellite state and parameters, detecting failures and recovering from them, preparing and following up TeleCommand plans. To support the task of failure detection and recovery, the Operation Ground Systems is made up of two relatively unconnected components. Amongst the interactive systems used within the control room of PICARD, synoptic (see Figure 3) represent an important support to the operators’ activities. Synoptic gather a set of parameters to propose a general overview of them, these parameters being used by the operators to monitor the state of the satellite. The PICARD operation control centre uses more than fifty synoptic containing around 10 000 parameters (such as battery status, communication link status…), and the number of procedures for possible maintenance operations goes beyond one hundred. As illustrated in Figure 3, synoptic may contain graphical representation of parameters, but most of them represent parameters as text (such as the central part of Figure 3).

Figure 4. Procedure manager

Another important part of the ground segment system is the procedure manager. It aims at triggering TeleCommands, i.e. uploading commands onto the board system in order to change its current configuration and makes the parameters evolve (see Figure 4). When operating a satellite (for instance when executing a particular procedure), such a quantity of screens and density of information makes it difficult for the operators to find a particular parameter navigating amongst the synoptic. This activity may be critical when the operator tries to solve a satellite failure, where he/she has to precisely analyse the relevant parameters. The complexity of a satellite makes it difficult to design a dedicated synoptic for each kind of failure, so that when an
unexpected event occurs, dedicated procedures must be redesigned, but not the interactive system itself which remains the same (and is thus design as generic as possible).

**Operational procedures as partly automated systems**

Satellites and spacecraft are monitored and controlled via ground segment applications in control centres with which satellite operators implement operational procedures. A procedure contains instructions such as sending teleCommands (TC), checking teleMetry (TM), waiting, providing required values for parameters, etc. The definition of operational procedures may be found in the ECSS-E-70-32A standard [8] and defines the elements that an operational procedure must contain (declaration of the local events raised within the procedure, preconditions, instructions...). Procedures are the main mechanism used in control rooms to manage the spacecraft during both test and operations phases.

**Software environment and modelling tools associated to the generation and distribution process**

The targeted platform (due to the project requirements) is Java and more specifically the Java technology called JavaFX ([http://javafx.com](http://javafx.com)) which allows the description of the graphical part of an interactive application with an XML file (called FXML) and which allows customisation of the graphical rendering using CSS styling ([http://www.w3.org/Style/CSS/](http://www.w3.org/Style/CSS/)).

---

**Figure 5. Generation process for interactive synoptics**

HAMSTERS and ICO notations, presented in previous section, have associated CASE (Computer Aided Software Environment) tools. HAMSTERS associated software tool (also called HAMSTERS) enables to edit task models and simulate their execution. ICO (Interactive Cooperative Object) is Petri nets based and associated to a supporting tool, Petshop. It enables to edit application behavioural models and to connect them to the presentation part of the user interface (graphical widgets and frames for example). It also enables to execute the application with the underlying behavioural models. Additionally, HAMSTERS task models and Petshop system models can be connected at edition time as well as at runtime in order to ensure consistency between operator tasks and system behaviour [1, 18, 19]. This synergistic use of the two tool-supported notations also provide support for assessment of function allocation between operator and system [16].

**Application of the process to PICARD ground segment applications (synoptic and procedure manager)**

The main idea we illustrate with this case study is how to take benefits from the model-based generation process to support the generation of customizable interactive synoptic, and to associate them to the original interfaces (synoptic and procedure manager) that are required to support most of the activities of the operators. The generic distribution process (Figure 2) has been instantiated (Figure 5) to reflect the use of our targeted platform and modelling tools:

- The starting point of the process (top-left part of Figure 5) is the original operational procedure from which we manually produce an ICO model (and a Hamsters model that is not represented here due to space constraints).
- The ICO procedure represents the behaviour of the being generated interactive synoptic and the
modifications performed on it introduces iterations in the generation process.

- The ICO procedure is automatically analysed with a Petri net pattern detector (bottom-left part of Figure 5), associated to a collection of patterns descriptions, which embed algorithms to detect the basic bricks that constitute a procedure such as parameter update, checking of these parameters, messages and choices proposed to operators. The result of this pattern extraction is a logical structure of the synoptic in form of a list of instantiated patterns (with the list of monitored parameters and a list of elements of the control flow of the procedure).

- A JavaFX component is then associated to each of this instantiated patterns, using a predefined mapping. These components are then integrated within a generic synoptic canvas, producing a JavaFX application (with no behaviour, the behaviour being provided by the ICO model in the next step). The customisation of the JavaFX components, generic canvas and produced JavaFX application is additionally supported by the use of CSS styling to precisely adjust graphical attributes of the generated synoptic.

- Lastly, the JavaFX synoptic and the ICO procedure are put together to provide the final interactive synoptic.

Examples of the models and interactive synoptic produced during this generation process are presented in Figure 6:

- The left part is an excerpt of the ICO model of the corresponding procedure where two parts are highlighted, corresponding to two behavioural patterns corresponding to the two parts on the right side of the figure.
- The centre part represents the generic graphical canvas.
- The bottom-right part is the resulting interactive synoptic.

**CONCLUSION**

This article has presented how model-based approaches can be used for the automated generation of contextual user interfaces and how they can provide operators of ground segments with focus and context information. This approach exploits a formal behavioural description technique (the ICO notation [25]) for the description of both the operational procedures and thus the behaviour of the generated user interface. The graphical presentation is produced using an XML dialect called FXML which
belongs to the JavaFX technology. This contribution presents a unique case study where the generation of user interfaces provides important benefits for operators of critical interactive systems. Furthermore, the distribution of generated user interface across another display guarantees segregation with the standard command and control system thus preventing possible fault propagation to the ground segment.

The current work corresponds to the final contribution of the research project ALDABRA and is under consideration for inclusion in the next generation of ground segment operations. While informal testing with ground segment operators has received very positive feedback, the critical system nature of the application domain requires adoption by regulatory authorities prior to development (by certified companies) and deployment in operational satellite ground segment. Such work is being undertaken and lead by CNES via ISIS (Initiative for Space Innovative Standards) targeting at standard, generic and innovative ground segments (http://www.iafastro.net/iac/archive/browse/IAC-09/B4/7/4801). This work is part of a more ambitious research programme aiming at defining processes, methods and tools for the design and development of safety critical interactive systems. While function allocation is critical for most (partly-) autonomous systems, the current paper only referred to a context of automation where allocation is previously defined and does not evolve. Future work intends to extend previous work on automation design [16] and aims at exploiting the tasks models to identify potential migrations and to assess the impact of such migrations on operations’ performance.

ACKNOWLEDGMENTS

This work was partly sponsored by CNES (French National Space Studies Center) R&T Tortuga R-S08/BS-0003-029 and Aladabra ETS-CT-R&T TTGA-196-CN as well as EUROCONTROL research network HALA! on Higher Automation Levels in Aviation and WPE project SPAD (System Performance under Automation Degradation).

REFERENCES


15.Manca, M., Paterno, F. Extending MARIA to support Distributed User Interfaces. In J.A. Gallud et al. (eds), Distributed User Interfaces: Designing Interfaces for the


Modeling Aircraft Pushback Trajectories for Safe Operations
Franziska Dieke-Meier, Thomas Kalms, Hartmut Fricke, and Michael Schultz
Chair of Air Transport Technology and Logistics
Technische Universität Dresden
01062 Dresden, Germany
{dieke-meier, fricke, schultz}@ifl.tu-dresden.de, thomaskalms@gmx.de

ABSTRACT
Pushback operations are safety critical as historic incident and accident data show. The consequences are severe injuries of the ground personnel and expensive aircraft damages. As such, pushback process optimization by means of precise tug/aircraft trajectory prediction is considered as valuable risk mitigation strategy. We present a mathematical model to predict these pushback trajectories under various geometric constellations which is a complex solver activity. Our proposed model relies on a kinematic approach to generate the pushback trajectory as a function of aircraft wheelbase and geometric angles for curved segments. The conclusive concept validation is based on commonly provided ICAO standard planning data and shows promising results which will be subject of further field validation tests.

Author Keywords
Pushback; safety; pushback trajectory; ground operations.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
A pushback of an aircraft is necessary when an aircraft is unable or not authorized to leave the aircraft stand with its own engine power because of the modality of the aircraft stand design. In that case a tug (also tractor) is used to push the aircraft to a target position on the centerline of the taxi lane/taxiway or to a safe area at the apron. The tug driver must push the aircraft within a narrow passageway, which is restricted by obstacles like other aircraft on adjacent or opposite stands, buildings, fences, or lighting poles.

A continued analysis of incidents/accidents during pushback at U.S. airports [1] shows that 31.8% of the 189 identified pushback-related occurrences resulted in damages caused by collisions between the pushed aircraft and other moving/pushed/stationary aircraft or fixed objects (category 1 occurrence) between 1991-2011. Figure 1 points out the increasing rate of incidents (accidents included) for this kind of event over the last years. A slightly reduced, but also increasing trend is recognizable for the separate illustrated rate of accident. It may be noticed that this rates cannot be correlated with weather caused visual conditions, because the main part of the investigated pushback events occurred under sufficient visual conditions in the daytime. Obviously other parameters, e.g. narrow spaces combined with operational time pressure, determine the increasing rate of incidents/accidents presumably.

The number of 21 (out of 60) category 1 occurrences at U.S. airports resulted in significant aircraft damages, which are classified as accident by the National Transport Safety Board (NTSB) corresponding to the ICAO definition of an accident [2]. Consequentially, an accident rate of 1.48E-07 is given for the event collision with objects and the investigated period. A second investigation of pushback occurrences at U.K. airports from 2005-2011 intensifies the presented situation: four of the seven pushback-related collisions (registered by the Air Accidents Investigation Branch) resulted in substantial aircraft damage and in an accident rate of 5.1E-07 events per pushback operation (rate of incidents: 7.7E-07, accidents included). The presented rates illustrate an optimistic view of the real situation, because the determination of the rates is based on the very conservative approach, that 100% of the outbound traffic needs pushback. Assuming 75% pushback operations
for outbound traffic the determined accident and/or incidents rates raise by the factor of 1.33. For calculations of the rates, (commercial) traffic data of the national boards (US: Air Traffic Activity System of the FAA, UK: Civil Aviation Authority) are used.

Fortunately, pushback occurrences are typically not attended by onboard fatalities. But the pushback process holds a particularly high risk of injury of ground personnel. In today’s operations, the pushback of an aircraft is usually assisted by a dispatcher and/or wing walkers. The risk of injury comes along with the close proximity of the assigned ground personnel to the nose gear, main gear and/or pushback tug in conjunction with the kinetic energy of the moving aircraft and the tug [3, 4]. Three fatalities of ground personnel were registered in the context of the mentioned U.S. airport study [1], which result in a fatal accident rate of 2.12E-08 fatalities per pushback operation for the period 1991-2011. Further 32 ground personnel members (and three flight crew members) were injured seriously or lesser during pushback operations during the same time. Because of the existing risks, some main airlines reduce the assisting personnel significantly and instruct the dispatcher to ride in/on the tug. But this is not a standardized procedure.

The results of the study emphasize the assumption that current implemented instruments/procedures seem to be no (more) sufficient to prevent collisions during pushback. Especially the determined fatal accident rate shows quite clearly the urgent necessity for action because it violates the formal A-SMGCS target level of safety of 1E-08 per operation [5]. Without improvements, the pushback process as part of the taxi-out process foils efforts in the context of A-SMGCS regarding implementing safer aircraft operations at airports. Additionally, the primary aim of A-SMGCS implementations at airports (low visibility surface operations) is not completely achievable without a significant change of current visibility-dependent pushback procedures. This shall be accomplished by providing an appropriate system support [6], which is a main motive of the presented research.

As Figure 2 illustrates, there are three general options to improve the pushback process. Providing both sufficient maneuvering space and standardized assistance procedures seem to be the best option. But, improvements of the obstacle situation are typically complicated and expensive to achieve, especially at large airports with a complex infrastructure. Further, any reduction of the walk-out assistants needs to be compensated from the safety perspective: to ensure redundancy of information (e.g. distances between aircraft and objects) thus preventing incidents.

The presented mathematical model of pushback trajectories will consider this improvement strategy by providing essential contributions to a) assess and show any potential for optimizing a given obstacle situation, b) identify safe pushback trajectories and c) allow deriving a guidance and support system for the operator both on ground and on flight deck.

**MODELING AIRCRAFT PUSHBACK TRAJECTORIES**

The mathematically modeling of trajectory aims at predicting aircraft positions relative to each other or to other obstacles within a pre-set area of investigation. Based on these data, improved, safe pushback procedures will be formulated. The challenge for the modeling part relies on the necessity, that the aircraft cannot be represented as a point mass model with one reference center of gravity as it might be appropriate for the flight or perhaps even taxi phase. Because of its complex maneuvering and turning behavior, the movement of an aircraft has to be described by the positions of the nose and main gear reference points depending on the nose wheel steering angle. Consequently, a pushback trajectory always consists of two trajectories to reproduce the aircraft position exactly regarding the obstacle environment: the trajectory of the nose gear reference (center) point and the trajectory of the main gear reference (center) point.

**Review of existing models**

Two general approaches are mainly used to model aircraft ground trajectories: the dynamic approach and the kinematic approach.

Dynamic models provide a very detailed view of real processes. Numerous dynamic model approaches exist, e.g. from Barnes and Yager [7] and Simoni [8]. The compendium published by Bates and Hagström [9] focuses on special closed-loop control problems for aircraft on the ground. The publications of Coetzee, Rankin, Krauskopf and Lowenberg [10-12] provide a complete force-predicated model with the use of nonlinear analysis and synthesis techniques (especially bifurcation). The authors mainly focus on the effects of an altering longitudinal center of gravity position and tire properties. Primarily the dynamic models are used for flight simulators and aircraft designing. The validity of these models depends on the coverage and the quality of multi-variant input parameters as well as the quality of the solving methods for the partly
complex calculations. Most of the published models only refer to a single aircraft type, where the A320 is often used as a reference.

Kinematic models are based on geometric conjunctions neglecting all forces, which enable quite simplified approaches for planning aspects. Adopting the kinematic approach of Fossum and Lewis [13], Coetzee describes the accompanied kinematic towing stability criteria [14]. The ICAO Aerodrome Kinematic Manual (ADM) provides an approach for the determination of the path followed by the main undercarriage to design the extra taxiway width (or fillet) in case of taxiway curves or junctions [15].

**Fundamentals of the pushback trajectory model**

Because of the unknown, but mandatory input parameter of different aircraft types, an application of the dynamic models to design pushback trajectories is not considered as appropriate. Comparing to taxiing maneuvers, the low velocities and accelerations during pushback operations rather call for kinematic models to predict the pushback trajectory. To develop the pushback trajectories model the following definitions are made:

- Each wheel of the aircraft touches the ground surface at one virtual reference point, located in the horizontal center of the wheel.
- The contact points of all wheels belonging to the nose landing gear are merged into one steerable wheel contact point $P_{ng}$. The touch-points of each wheel of the main gear are equally merged into one fixed reference wheel contact point $P_{mg}$. Both aggregations are calculated by arithmetic averaging.
- Any effects of potential tire slippage and inertia are neglected.

During a standard pushback, the tug’s forces are transferred to the aircraft by the strut of the nose landing gear, located at $P_{ng}$. Therefore, $P_{ng}$ acts as the point of traction, causing the connected aircraft and his main landing gear center $P_{mg}$ to follow on a dependent path. Such simplifications to a kinematic model are standard practice for airport planners.

Aircraft manufacturers provide reference turn radii for the design of taxiways matching the so-called tractrix curves of their aircraft. Equivalent methods are used in road planning for the layout of junctions and gateways, but also in automobile equipment like rear view cameras, assisting drivers backing up a vehicle. It seems desirable to adapt those existing models of tractrix curves to compute the movement of the main landing gear, depending on the trajectory of the nose gear. However, an appropriate model needs to be transferable to the kinematic conditions of a towed aircraft and the special case of pushing an aircraft backwards at the same time.

The presented model adopts the kinematic approach of Zoebel [16]. For a motor truck, connected to a single-axis trailer and moving on a circular path, Zoebel describes a suitable kinematic computation of the movement executed by the trailer. Simplified and transferred to the field of road traffic, a pushed or towed aircraft acts similar to a single-axis trailer pushed or pulled by a towing vehicle. Transferring Zoebel’s approach back on the tug-aircraft system, the following variables describe the aircraft movement (see Figure 3):

- Point $P_0$ represents the turning center of a circular path followed by the nose gear.
- Wheelbase $l_{wb}$ measures the distance between the nose gear center $P_{ng}$ and main gear center $P_{mg}$.
- Rolling radius $r_{ng}$ defines the radius followed by the nose gear center $P_{ng}$.
- Angle $\gamma$ defines the horizontal relation between tug and aircraft.
- Angle $\varphi$ represents the angular position of the nose landing gear on its circular path.

![Figure 3. Geometric model of an aircraft.](image)

**pushed/towed by a tug along a circular path**

By pushing or towing the nose gear center $P_{ng}$ on a circular path around $P_0$ with a nose wheel steering angle $\beta$ and a constant radius $r_{ng}$ (see Figure 3), the main gear center $P_{mg}$ will move on a certain path. Regardless of its starting position, the motion of a towed aircraft main gear center $P_{mg}$ will converge to a stable circular equilibrium around $P_0$, depending on $r_{ng}$ and $l_{wb}$. Two cases of conjunction could be distinguished. The inner case describes the movement of the aircraft’s main gear center starting inside the equilibrium (Figure 4) while the outer case represents a main gear center moving starting outside the equilibrium (Figure 5). After reaching the equilibrium the main gear center $P_{mg}$ will also move on a respective circular path as the nose gear center $P_{ng}$. Contrary to a towing movement, a simultaneous circular movement of the pushed nose and
main gear center is nearly impossible in practice. Because of the unstable movement of the pushback small inputs (e.g. by the surface or driver) will result in an appreciable divergence of the main gear center from the circular path. Hence, especially these outer and inner cases are interesting for modeling of pushback trajectories. However, the equilibrium is essential for the mathematical definition of the tractrix, described in the following.

The main gear center’s $P_{mg}$ change in position is derived from $\varphi$ and $\gamma$ as given in equation 4. The difference $\varphi - \gamma$ represents an angle, which consists of the nose wheel steering angle $\beta$ and the real deviation angle of the main gear.

$$\frac{\delta(y_{mg})}{\delta(x_{mg})} = \tan(\varphi - \gamma) \quad (4)$$

The left part of equation (4) is converted into a partial derivative while the tangent on the right side is replaced by sine and cosine.

$$\frac{\delta(y_{mg})}{\delta y} + \frac{\delta(y_{mg})}{\delta \varphi} \frac{\delta \varphi}{\delta y} = \frac{\sin(\varphi - \gamma)}{\cos(\varphi - \gamma)} \quad (5)$$

By further conversions and simplifications, the equation can be reduced to the following term:

$$\frac{\delta \varphi}{\delta y} = \frac{l_{wb}}{l_{wb} + r_{mg} \cdot \cos \gamma} \quad (6)$$

The derivation of $\varphi$ with respect to $\gamma$ leads to equation (7) (according to [17]):

$$\varphi(\gamma) = \frac{l_{wb}}{r_{crit} \sqrt{r_{ng}^2 - l_{wb}^2}} \ln \left( \frac{(r_{ng} - l_{wb}) \cdot \tan \frac{\gamma}{2} + \sqrt{r_{ng}^2 - l_{wb}^2}}{(r_{ng} - l_{wb}) \cdot \tan \frac{\gamma}{2} - \sqrt{r_{ng}^2 - l_{wb}^2}} \right) \quad (7)$$

Substitution with the critical radius $r_{crit}$, $\varphi(\gamma)$ leads to:

$$\varphi(\gamma) = \frac{l_{wb}}{r_{crit} \sqrt{r_{ng}^2 - l_{wb}^2}} \ln \left( \frac{(r_{ng} - l_{wb}) \cdot \tan \frac{\gamma}{2} + r_{crit}}{(r_{ng} - l_{wb}) \cdot \tan \frac{\gamma}{2} - r_{crit}} \right) \quad (8)$$

Solving $\varphi(\gamma)$ for $\gamma$ results in two solutions $\gamma(\varphi)$ due to the absolute value within the logarithmic function. The solutions represent the above-mentioned specific cases of the circular tractrix, the inner and outer case.

$$\gamma(\varphi) = \begin{cases} 2 \cdot \arctan \left( \frac{r_{crit}}{r_{ng} - l_{wb}} \cdot \frac{e^{\frac{r_{crit} \varphi}{l_{wb}}} + 1}{e^{\frac{r_{crit} \varphi}{l_{wb}}} - 1} \right), & \text{inner case} \\ 2 \cdot \arctan \left( \frac{r_{crit}}{r_{ng} - l_{wb}} \cdot \frac{e^{\frac{r_{crit} \varphi}{l_{wb}}} - 1}{e^{\frac{r_{crit} \varphi}{l_{wb}}} + 1} \right), & \text{outer case} \end{cases} \quad (9)$$

In the special case of a nose gear moving on a radius smaller than the wheelbase, the solution of the equation (1) of the critical radius $r_{crit}$ becomes imaginary. To exclude this case, the radicand in equation (1) is replaced by its absolute value. Zoebel’s approach [16] of the inner and outer tractix is extended by a third solution of $\gamma(\varphi)$ by substituting $r_{crit}$ in the equation for the outer tractrix by an imaginary number $r_{crit} = iy$. 

![Figure 4. Inner tractrix of an aircraft towed along a circular path ($r_{ng} > l_{wb}$)](image)

![Figure 5. Outer tractrix of an aircraft towed along a circular path ($r_{ng} > l_{wb}$)](image)
The derived tractrix curves are aligned for each segment and serve as a basis for approximating the circle segments. The radius rng can now be computed as:

\[
r_{ng} = \frac{\Delta s}{\Delta HDG_{ng}} = \frac{\Delta s}{HDG_{ng}(t_{i+1}) - HDG_{ng}(t_i)}
\]

Further, straight segments can be approximated this way by using an accordingly large radius r_{ng} = 10^5 \text{ m}. By doing this for all n segments, the nose gear path is converted to a sequence of circle segments, which can now be processed with a circular tractrix.

In summary, the circular movement of an aircraft can be described by three tractrix curves. Choosing the right tractrix for a certain movement depends on the following fundamental decision criteria:

- The special tractrix curve in the case of r_{ng} ≤ l_{wb}.
- The outer tractrix curve in the case of r_{ng} > l_{wb} and the main gear center outside the equilibrium.
- The inner tractrix curve in the case of r_{ng} > l_{wb} and the main gear center inside the equilibrium.

Handling of complex trajectories
The kinematic model is thus able to compute the main gear movement of an aircraft that is pushed or towed by its nose gear on a fixed circular trajectory. In consideration of the fact that a typical pushback trajectory includes straight segments and curves of non-fixed bending, the next step is about gaining access to more complex trajectories.
introduced for $\gamma$ at the beginning of the circle segment (Figure 8). A counter clockwise movement of the nose gear ($\Delta HDG_{ng} < 0$) leads to $\gamma_{start} = 180^\circ - \gamma$ while a clockwise movement ($\Delta HDG_{ng} > 0$) results in $\gamma_{start} = \gamma$.

2. The choice between special and normal tractrix curves requires a comparison of the aircraft’s wheelbase $l_{wb}$ with the actual rolling radius $r_{ng}$. The special tractrix curve represents the case of $r_{ng} \leq l_{wb}$ whereas $r_{ng} > l_{wb}$ leads to the inner or outer tractrix respectively.

3. A decision between the inner or outer tractrix requires a unique characterization of the aircraft’s main gear position in relation to the equilibrium. A useful approach is the calculation of the angle $\gamma_{crit}$ representing the value for $\gamma$ for the special case where the main gear center is moving on the equilibrium. Equally to point 1, $\gamma_{crit}$ depends on $\Delta HDG_{ng}$:

$$\gamma_{crit} = \begin{cases} 2 \cdot \text{atan} \left( \frac{r_{crit}}{r_{ng} - l_{wb}} \right), & \Delta HDG_{ng} > 0 \\ 180^\circ - 2 \cdot \text{atan} \left( \frac{r_{crit}}{r_{ng} - l_{wb}} \right), & \Delta HDG_{ng} < 0 \end{cases}$$

(14)

The final decision between inner and outer tractrix curve depends on a simple comparison of $\gamma_{crit}$ and $\gamma_{start}$. The case of $\gamma_{crit} \leq \gamma_{start}$ leads to the use of the inner equation, otherwise the outer equation is chosen.

Figure 8. Determination of $\gamma_{start}$

At this point, the proper type and size of the tractrix can be chosen and calculated for the considered circle segment. The angle $\gamma_{start}$ represents a certain entry point on the selected tractrix by its corresponding angular position $\phi_{start}$:

$$\phi_{start} = \begin{cases} \frac{l_{wb}}{r_{crit}} \cdot \ln \left( \frac{r_{ng} - l_{wb}}{r_{crit}} \frac{\gamma_{start}}{2} \right), & \text{for } r_{crit} > l_{wb} \\ 2 \cdot \frac{l_{wb}}{r_{crit}} \cdot \text{atan} \left( \frac{r_{ng} - l_{wb}}{r_{crit}} \frac{\gamma_{start}}{2} \right), & \text{for } r_{crit} \leq l_{wb} \end{cases}$$

(15)

The angular movement of the nose gear center is already given with $\Delta HDG_{ng}$. Therefore, the final angular position $\phi_{end}$ of the nose gear center can be calculated by:

$$\phi_{end} = \begin{cases} \phi_{start} + |\Delta HDG_{ng}|, & \text{for pushed aircraft} \\ \phi_{start} - |\Delta HDG_{ng}|, & \text{for pulled aircraft} \end{cases}$$

(16)

Inserting $\phi_{end}$ back into the chosen tractrix curve leads to the angle $\gamma_{end} = \gamma(\phi_{end})$ which needs to further be transferred back to the angle $\gamma(t_{i+1})$ by computing $\gamma(t_{i+1}) = 180^\circ - \gamma_{end}$ in case of steering counter clockwise to the left and $\gamma(t_{i+1}) = \gamma_{end}$ in case of steering clockwise to the right.

Finally the new earth-fixed position of the main gear can be easily calculated by the known position of the nose gear center and $\gamma(t_{i+1})$:

$$x_{ng} = x_{ng} + l_{wb} \cdot \sin(HDG_{ng} + \gamma - 90^\circ)$$

(17)

$$y_{ng} = y_{ng} + l_{wb} \cdot \cos(HDG_{ng} + \gamma - 90^\circ)$$

(18)

Figure 9 illustrates the procedure of selecting and adapting the tractrix to a specific curve segment of length $\Delta s$ and radius $r_{ng}$. Repeating this process for all following segments of the nose gear trajectory generates a continuous track of the main gear center. Obviously, the accuracy of the generated main gear track strongly depends on the segment-length $\Delta s$.

Figure 9. Computing the movement of the main gear center

Implementation

The model is implemented as a Java application to calculate the pushback trajectory (Figure 10) and display the output of relevant model parameters: heading of the tug and aircraft tail, the nose wheel steering angle, the radius of nose gear turn and $\gamma$ (see Figure 11).
Figure 10. Tool display: Pushback Trajectory Data

Figure 11. Display for monitored Pushback Data

CONCEPT VALIDATION
To prove the validity of the developed model the specification of the ICAO Aerodrome Design Manual (ADM) [15] are used as a validation reference. The ADM uses a similar kinematic approach while focusing on forward movements only. Nonetheless, it provides some model parameters and plotted charts (e.g. see Figure 14) allowing a validation of the developed pushback trajectory model. Two maneuvering scenarios are chosen and the corresponding data, determined by the model and the ADM approach, are compared.

CONCEPT VALIDATION
To prove the validity of the developed model the specification of the ICAO Aerodrome Design Manual (ADM) [15] are used as a validation reference. The ADM uses a similar kinematic approach while focusing on forward movements only. Nonetheless, it provides some model parameters and plotted charts (e.g. see Figure 14) allowing a validation of the developed pushback trajectory model. Two maneuvering scenarios are chosen and the corresponding data, determined by the model and the ADM approach, are compared.
Manoeuvring scenario A), which is illustrated in Figure 13, represents a towed aircraft following an arc with a radius \(r_{mg} = 1.5 \cdot lb\). At the beginning of the forward maneuver the main gear center \(P_{mg}\) deviates 67.5\% of the wheelbase. Assuming that the aircraft passes polar angles of 66.5° and 114.5° the corresponding deviations of \(P_{mg}\) are determined. Using Figure 14 the ADM related deviations could be read. For this, an initial polar angle of 27.5° has to be considered, which represents the theoretic completed angle from an origin point (with deviation = 100\%, polar angle = 0°) to \(P_{mg}\) with a deviation of 67.5\% of the wheelbase. The results are shown in Table 1.

This kind of forward maneuver is not similar to a backward pushback maneuver because of the differed circle maneuver geometry of a pushback. In practice the driver focuses the movement of the main gear center primarily, while the nose gear describes the necessary curve to position the main gear at the desired curve line. Regarding on forward maneuvers exclusively, the scenario corresponding charts of the ICAO ADM are not applicable to a backward maneuver. But our presented model is also usable for calculations of forward maneuvers and provides a wise addition to the ICAO ADM by the case of backward maneuvers.

![Figure 14. Nose wheel steering angle \(\beta\) and deviation of the main gear center \(P_{mg}\) when the nose gear center \(P_{nm}\) follows an arc of circle (cont.), valid for \(r_{mg}/lb = 1.5\), [15]](image1)

The second maneuvering scenario B) comprehends a pushback maneuver (backwards). The main gear deviates two meters from the straight movement line of the nose gear at the beginning of the maneuver. The nose gear center follows a straight line, whereas the main gear center deviation increases.

![Figure 15. Backward maneuvering scenario B)](image2)

The corresponding ADM chart (see Figure 16) plots the deviation of the main gear center depending on the distance travelled by the nose gear center (axis of abscissae) and an initial deviation (axis of ordinates). Towing the aircraft with a deviated main gear center along a straight line, results in a decreasing deviation of the main gear relating to the straight movement line of the nose gear. For this special case the ADM chart is also applicable for the backward maneuver. Using the chart for the backward maneuver the initial deviation is not an input variable (as in the case of a forward maneuver) but an output variable. The sought deviation after 20 meters travelling is determined with the value of distance travelling \((A340-600: 20 \text{m}/32.883 = 60.8\%)\) and the deviation of the main gear center \((A340-600: 2 \text{m}/32.883 = 6.1\%\). Table 2 shows the determined deviation of the main gear center after a maneuver distance of 20 meters.

![Figure 16. Deviation of the main gear center \(P_{mg}\) when the nose gear center \(P_{nm}\) follows a straight line, [15] (reduced to the relevant data curves)](image3)

**Table 1. Compared results of concept validation scenario A)**

| A/C   | \(l_{ob}\) [m] | Polar angle | ADM [m] | Model [m] | \(|A|\) [m] |
|-------|----------------|-------------|---------|-----------|----------|
| A340-600 | 32.883         | 66.5°       | -3.979  | -3.935    | 0.044    |
|        |                | 114.5°      | -9.372  | -9.365    | 0.007    |
| B787-8  | 22.78          | 66.5°       | -2.756  | -2.726    | 0.030    |
|        |                | 114.5°      | -6.492  | -6.488    | 0.004    |
| A320-100 | 12.64          | 66.5°       | -1.529  | -1.513    | 0.016    |
|        |                | 114.5°      | -3.602  | -3.600    | 0.002    |
alidati
calculating the essential

e.g. implemented in an operator

2.

trajectories provides

advisory system concept,

aircraft instead of the aircraft engines) could enable an autonomous

by electrical motors in the wheel of the main or nose gear

maneuvers

provide

requirements

has to be noticed, that t

derive collision free

step

environment on apron

the model enable

pushback operations (of

of the pushback process to allow for a significant mitigation

Our

CONCLUSION

Comparing the determined results of both validation scenarios a very good compliance is given. Of course, we consider performing reliable field measurements to finally ensure the quality of the trajectory model.

CONCLUSION

Our proposed trajectory approach provides a reliable model of the pushback process to allow for a significant mitigation of pushback risks and provides a solid base for advanced pushback operations (e.g. simultaneous pushback to alternative parallel taxilanes). By calculating the essential aircraft centers of reference during a pushback maneuver the model enables placing the aircraft exactly in an obstacle environment on aprons to assess the given situation and identify safe pushback trajectories/ maneuvering corridors. In the context of this research and development the next step is to consider the specific obstacle situation on the apron inside the developed simulation environment to derive collision free routes. Validating the model data against the ICAO ADM data shows promising results, which will be further subject to the upcoming field tests. It has to be noticed, that the ADM only considers requirements for forward maneuvers, but our model will provide a solid base to cover a broad range of future apron maneuvers also. For example, eTaxi (aircraft taxi powered by electrical motors in the wheel of the main or nose gear instead of the aircraft engines) could enable an autonomous aircraft pushback [6] and would require an adequate aircraft stand planning and a certification of the corresponding aircraft procedure. The pushback trajectory model provides a planning support, which are currently not addressed by the ICAO ADM. Finally, implemented in an operator advisory system concept, our model of aircraft pushback trajectories provides a significant contribution towards to both the increasing automation/control of pushback operations and future aircraft operations.

REFERENCES

1. Diecke-Meier, F., Fricke H. The need for a collision


2. International Civil Aviation Organization (ICAO).

Aircraft Accident and Incident Investigation. Annex 13
to the Convention on ICAO, 9th ed., ICAO, Montreal,

2004.

3. Dell G. Pushback accidents reviewed to identify causes

and prevention. Flight Safety Foundation - Airport


4. Flight Safety Foundation. Communication from the

pushback-tractor seat helps prevent serious injuries.


5. International Civil Aviation Organization (ICAO).

Advanced Surface Movement Guidance and Control


2004.

6. Diecke-Meier, F., Fricke H. Expectations from a steering

control transfer to cockpit crews for aircraft pushback.


7. Barnes, A. G., Yager, T. J. Enhancement of aircraft

ground handling simulation capabilities. Neuilly sur


8. Simoni, S. Development of a multi-physics model of a

civil aircraft for ground maneuvers using modelica.

Università di Pisa, 2006.

9. Bates, D., Hagström, M. Nonlinear Analysis and

Synthesis Techniques for Aircraft Control. Springer,


10. Rankin, J., Coetzee, E., Krauskopf, B., Lowenberg, M.

Bifurcation and Stability Analysis of Aircraft Turning

on the Ground. Journal of guidance, control, and

dynamics (2009), 500-511.

11. Coetzee, E., J., Krauskopf, B., Lowenberg, M.,

Nonlinear Aircraft Ground Dynamics. International

Conference on Nonlinear Problems in Aviation and


12. Rankin, J. Bifurcation Analysis of Nonlinear Ground


Trailer-Truck Jackknifing. SIAM Review Vol. 23 (1981),

no.1, 95-99.

14. Coetzee, E. Modelling and Nonlinear Analysis of

Aircraft Ground Manoeuvres. University of Bristol


15. International Civil Aviation Organization (ICAO).

Aerodrome Design Manual, Part 2 Taxiways, Aprons


16. Zöbel, D. Mathematical Modeling of the Kinematics of

Vehicles. In: Hrubina, Kamil: Mathematical Modeling

of Technical Processes. Socrates/Erasmus Summer


17. Bronstein, I. N., Semendjaev, K.A., Musiol, G.,

Muehlig, H. Taschenbuch der Mathematik. Frankfurt am

Main, Verlag Harri Deutsch (2005), 1075.

18. International Civil Aviation Organization (ICAO).

Aerodromes, Volume I Aerodrome Design and

Operations. Annex 14 to the Convention on ICAO,

5th ed., ICAO, Montreal, 2009

| A/C   | l_{wh} [m] | ADM [m] | Model [m] | |A| [m] |
|-------|------------|---------|-----------|------|
| A340-600 | 32.883     | 3.486   | 3.666     | 0.180 |
| B787-8  | 22.78      | 4.602   | 4.768     | 0.166 |
| A320-100| 12.64      | 8.481   | 8.156     | 0.325 |

Table 2. Compared results of concept validation scenario B)

Comparing the determined results of both validation scenarios a very good compliance is given. Of course, we consider performing reliable field measurements to finally ensure the quality of the trajectory model.
Multiobjective Tactical Planning under Uncertainty for Air Traffic Flow and Capacity Management

Gaétan Marceau Caron  
Thales Air Systems  
Rungis, France  
gaetan.marceau-caron@inria.fr

Pierre Savéant  
Thales Research & Technology  
Palaiseau, France  
pierre.saveant@thalesgroup.com

Marc Schoenauer  
TAO project-team  
INRIA Saclay Ile-de-France  
Université Paris Sud  
Orsay, France  
marc.schoenauer@inria.fr

ABSTRACT
We investigate a method to deal with congestion of sectors and delays in the tactical phase of air traffic flow and capacity management. It relies on temporal objectives given for every point of the flight plans and shared among the controllers in order to create a collaborative environment. Uncertainty is modeled at the trajectory level with temporal information on the boundary points of the crossed sectors and with a probabilistic occupancy count. Therefore, we can model the accuracy of the trajectory prediction in the optimization process. On the one hand, more accurate is our prediction; more efficient will be the proposed solutions. On the other hand, when uncertainty is not negligible, the proposed solutions will be more robust to disruptions. Also, a multiobjective algorithm is used to find the tradeoff between the delays and congestion. Finally, the flow management position can choose automatically or manually the adequate solution.

Author Keywords
Probabilistic Model; Uncertainty Handling; Evolutionary Computation; Multiobjective Optimization.

ACM Classification Keywords
J.2 Applied Computing: Physical Sciences and Engineering—Aerospace

General Terms
Design; Algorithm; Experimentation; Performance.

INTRODUCTION
Nowadays, delays in air traffic management are a major problem, which is mainly caused by capacity limits, particularly in Europe where the flight density is high. This relates directly to the regulations taken in the Air Traffic Flow and Capacity Management (ATFCM), because their impact on the trajectories are more consequent than the ones taken in Air Traffic Control. According to [1], the Network Manager is responsible for planning the demands issued by the airlines on the air traffic infrastructures, including runways and control sectors. This is done in the strategic and the pre-tactical phase, where the different actors are implied in the Collaborative Decision Making process. Then, the ATFCM daily plan is published, which describes a set of tactical measures, e.g. routeing scenarios or regulations, and the departure slots are assigned to the aircraft operators. The latter consist in intervals of 15 minutes that are supposed to encompass the uncertainty at the boarding phase. Thereafter, the evolution of the flight is assured through the Flow Management Position, linking every control centers to the Network Manager. This position is responsible to implement local procedures, to monitor their effect and to provide the relevant information to the Network Manager. In return, the Enhanced Tactical Flow Management System is responsible to provide the predicted occupancy count for each sector. Then, the network manager and the flow management position will agree if the activation of regulations is required or not.

Of course, the sector occupancy is calculated from a trajectory prediction with potential errors. Nevertheless, if the predicted sector occupancy is higher than the capacity, regulations can be activated and generate delays. This will effectively disrupt the initial slot allocation and generate delays. The major drawback from this procedure is that the effect of regulations becomes effective a few hours later and will drastically impact the workload. In some cases, when uncertainty about the future is high, ineffective regulations might be issued. This is the main reason for the introducing of the Short-term ATFCM measures in the process. These are intended to solve small disruptions locally in time and space, and encompass minor ground delays, flight level capping and minor rerouting. In the following, we present adaptive target times of arrival as a mean to stabilize the network impacted by small disruptions.

In air traffic control, the limits of ground trajectory prediction are known, especially when the information concerning the aircraft state or the pilot intent is not available. Consequently, the air traffic controllers have to work with a bounded time horizon to fulfill the separation assurance task. Beyond this horizon, the situation becomes unclear and anticipation is difficult, if at all possible. Nevertheless, with the progress of flight management systems, one can expect that these limits will be pushed...
back. This new generation of navigational tools opens the way to new opportunities for planning during the tactical phase of Air Traffic Flow and Capacity Management (ATFCM). During this phase, flights can be scheduled temporally over geographical points in order to minimize a complexity criterion and/or the delays of flight arrivals. The basic idea is to globally schedule beforehand all the flights in order to facilitate the work of the air traffic controllers. For now, the implementation of these target times of arrival could be part of the air traffic controller’s job in the first place, but will eventually be integrated into the 4D trajectory concept. Because the ATFCM tactical temporal horizon can go beyond 2 hours, it must deal with the inherent uncertainties relative to the environment. These can be the consequences of late departure, wind variations, conflict resolution or, more importantly as far as security is concerned, of a requested avoidance of a hazardous weather phenomenon area. Hence, it is mandatory for any automated system that is designed to optimize the scheduling of the flights to be aware of these sources of uncertainty in order to ensure robustness of the decisions.

This paper introduces an original methodology to tackle uncertainty regarding aircraft trajectories and airspace sector crossings. Probability theory is used as a formal framework to capture the essence of uncertainty in air traffic management. Then, the necessary tools for propagating the uncertainty temporally and spatially, through the waypoints of flight plans, are explained. From this propagation, we can infer the probability of sector congestion with a closed-form equation, avoiding costly Monte-Carlo simulations of the complete system that is usually the only way to numerically estimate uncertainties. Then, the probabilistic model is used within an optimization algorithm for scheduling all flights at boundaries of the sectors in order to minimize the expected cumulated delays and the expected sector congestion. To the best of our knowledge, the novelty of this work is to provide the inference mechanism to propagate the uncertainty from the trajectories to the sectors and to use the resulting probabilistic model as a black-box for multiobjective optimization.

The paper is organized as follows: first, we present a literature survey on the formulations and the techniques used to solve the related air traffic flow management problem. Then, we present the motivation of the paper and the mathematical formulation of the uncertainty model. Finally, we give the experimental results that validate the model and the optimization process, followed by a discussion on the possible extensions of the model and sketches further directions of research.

RELATED WORK

The Operational Research community has studied many variants of the air traffic flow management problem since the beginning of the 90s. Today, we can distinguish two families of formulations: the static approach and the dynamic approach. The former, also known as the single-stage approach or open-loop approach, is simpler in terms of formalism and computation since its goal is to find an optimum once and for all. The latter, also known as the multi-stage approach or closed-loop approach, deals with uncertainty and incoming information about time estimates, sector capacities and unpredictable phenomenon.

To our knowledge, the most comprehensive formulation for the static approach is the Air Traffic Flow Management Rerouting Problem [2], which integrates all phases of a flight, different costs for ground and air delays, rerouting, continued flights and cancellations. Instances of the size of the National Airspace of the United States were used to validate the approach. Also, with the same mathematical framework, [3], [4] have formulated the problem in terms of routes instead of nodes. The latter includes a stochastic formulation with discrete probabilities associated to scenarios for sector capacities. In the same manner, there is also the work of [5], which describes an optimization problem to minimize directly the probability of congestion in the sectors with the concept of chance constraint.

The works mentioned so far solve the ATFCM problem in the pre-tactical phase. The technique used, binary integer programming, is powerful enough to address large-scale scenarios. Besides, other techniques were used to solve similar problems. [6] uses stochastic optimization methods for handling sector congestion with take-off delays and rerouting. Constraint programming was also used by [7] and [8]. The former solves the slot allocation problem with sector capacity constraints and the former minimizes an air traffic complexity metric for multiple sectors.

From the Optimal Control community, [10] proposes a solution to the dynamic problem when weather disruptions occur. They apply network regulation strategies to a dynamical model representing the problem as a network of queues with load-dependent service rates on sector boundaries. This macroscopic view, also known as Eulerian Model, is well adapted for instances of the size of the National Airspace System. The purpose of the given algorithm is to change the incoming throughput or to reroute flows when a sector capacity degradation occurs.

A multiobjective optimization approach has been used in air traffic control by [8] to minimize an aggregated complexity metric, designed and validated by Eurocontrol, over sectors. In this case, the dimensions of the multiobjective problem are the complexity for each sector and thus, they use the weighted sum as a scalarization function to aggregate the complexity over all sectors. In this case, the multiobjective problem is transformed into a mono-objective problem and the weights are used to generate different trade-offs between sectors. Also, [9] use the multiobjective approach to model the trade-off between sector congestion and delays. In this case, the objective function space is in two dimensions and the decision space consists of the takeoff time of the flights and the chosen routes. A multiobjective
genetic algorithm is used to generate a pool of solutions with a diversity measure in order to distribute them uniformly on the Pareto front.

Besides, a study of the uncertainty was conducted by [11] with an analysis of the prediction error of the time of arrival of the aircraft. The main hypothesis of the study is that the random variable of the prediction error follows a Gaussian distribution. The parameters of this distribution were estimated from real data. The mean error is fixed to zero and the standard deviation is 4 minutes for active flights and 15 minutes for proposed flights.

To the best of our knowledge, this article is the first to tackle the problem of optimization of the air traffic flow and capacity management with a probabilistic model used to monitor the flights in real-time and a multiobjective algorithm to find the adequate actions to respond to disruptions.

**MOTIVATION**

The goal of this work is to propose an automated method to enhance the robustness of the network to small disruptions in the tactical phase in ATFCM. To do so, the temporal uncertainty of flight trajectories are modeled in a dynamical way since the estimated time over (ETOs) can be updated, i.e. via a data-link transmission between ground control and the flight management system of the aircraft or with updates from a ground trajectory prediction. Through a monitoring process, any change in the sector capacities will result in the evaluation of the congestion from these temporal trajectories. If need be, an optimization process is launched in order to change the trajectories of the aircraft according to the magnitude of the disruption. For this part, a multiobjective algorithm is used in order to find multiple plans, which give the tradeoff between generated delays and congestion reduction. This is a way for the FMP to be able to choose manually or automatically the most adapted solution to the disruption.

The basic idea of the approach relies on the concept of organized traffic. During the arrival phase, there is an inherent bottleneck, because of the required separation on runway due to the wake vortex. A solution to the complexity of the converging aircraft toward the runway was to enforce a sequence at the beginning of the arrival phase. This sequence organizes the traffic in a queue and reduces the four-dimensional space problem into a simpler. In situation of heavy traffic, such structure shall be generalized globally. As in [8], it can be done by establishing targets of arrival on boundary points of the sectors. Then, airborne measures, such as miles-in-trail, can be applied to entire flow and reduce considerably the complexity.

The main novelty of this paper is to consider the uncertainty at a trajectory level by the intermediate of the target time of arrival, then to propagate the uncertainty into the sectors by using a closed-form equation and finally, to use the probability marginals in order to infer the expected cost of delays and the expected cost of congestion. Then, the targets are tuned with a multiobjective algorithm in order to reduce the expected cost of congestion, if it is higher than a threshold. Also, we include ground delays since the actions on airborne flights are clearly restricted by justified economic reasons from the airlines.

Contrary to static and deterministic approaches, a dynamic and stochastic approach, like the one presented here, can generate a plan that is robust to changes as long as the uncertainty is well estimated. On one side, an optimal schedule for the deterministic approach is characterized by the tightness between the target times of arrival. In effect, to minimize the delays, an optimal solution should imply that an aircraft enters in the sector as soon as another aircraft leaves it. With deterministic targets, the difference in time between these targets will be of the same order than the time discretization, e.g. one minute. With a probabilistic approach, we use safety margins between these moments that are function of the presence of uncertainty in the system. The main difference with robust approaches, which considers the worst-case scenario, is that the plan is not too conservative. In effect, the probability that the worst case arises might be so low that it will systematically lead to a suboptimal behavior. Instead, it is more interesting to consider the scenarios proportionally to their probability of occurrence.

Besides, applying multiobjective optimization in this context is a way to gather multiple alternatives on the schedules of arrivals in the sectors, each corresponding to a trade-off between minimizing the use of regulations and reducing the complexity. In this work, the complexity refers to the probabilistic occupancy count of a sector, which is a refinement of the deterministic one used today. Usually, the monitoring value or capacity of the sector is determined from the feedbacks of the controllers. Then, a monitoring process is responsible to raise an alarm when the occupancy count reaches 90% of the capacity at any time. Since it is a scalar value for the entire sector, it does not account of the geometries of the trajectories for particular configurations. As a matter of fact, increasing the number of flights by one in the sector might increase drastically the workload of the controllers depending on the current airspace. If the traffic is organized, the increase will be small, but if the flights have many crossing trajectories or with many flight level changes, the disruption might be significant. Consequently, the occupancy count is not sufficient alone to evaluate the impact of the number of flights on the workload of the controllers and so; additional air traffic control tools are necessary to manage locally and spatially the trajectories.

Therefore, we believe that a global optimization system for air traffic management shall be composed of multiple components. In this article, we expose important concepts on the tactical phase of ATFCM. This component should acknowledge the optimization done previously by the pre-
tactical phase. As a matter of fact, this could be the initial plan and all the work done in the tactical phase is to update the plan following the unexpected perturbations. Then, a tactical tool in Air Traffic Control should use the information given by our component in order to do conflict detection and resolution. As an example, [12] provides the necessary concepts to do conflict resolution and scheduling for aircraft converging toward a fixe by taking into account the airline preferences. Other works as [13] and [14] gives a methodology to solve the problem in general airspace and the considerations in the implementation of such systems respectively. The combination of the three approaches could decrease substantially the workload associated to the coordination phase.

MATHEMATICAL FORMULATION

Notation

Essentially, the mathematical formulation will rely on the probability theory and stochastic optimization. Here, the events are target times of arrival at georeferenced points. Let us consider flight plan \( f \in F \) with \( n \) waypoints denoted by \( X_1^f, \ldots, X_n^f \) and associated to \( n \) random variables \( T_1^f, \ldots, T_n^f \), where \( T_i^f \) represents the time of overfly of flight \( f \) over waypoint \( X_i \), and let us call \( p_f \) the probability density of \( T_i^f \), dropping the superscript \( f \) when there is no ambiguity.

According to standard definition, the marginal probability is:

\[
P[T_j \in \Delta t|T_i = t_i] = \int_{\Delta t} p_{ji}(\tau|t_i) d\tau
\]  

where \( p_{ji}(\tau|t_i) \) is the conditional probability that the flight is over \( X_j \) during the time interval \( \Delta t \) given that the flight is over the point \( X_i \) at time \( t_i \).

\[\text{Figure 1: Bayesian Network for a flight plan}\]

Trajectory Model

Let’s define an uncertainty model for any trajectory. To expose easily the concepts presented here, we rely on the graphical model on Figure 1, namely a Bayesian Network, to represent the interactions between our random variables, illustrated with green circles. An arrow between \( T_i \) and \( T_{i+1} \) shows that the former influences the latter, or more precisely, that the two random variables are not independent. The joint density function of \( T_i \) and \( T_{i+1} \) is:

\[
p_{i,i+1}(t_i, t_{i+1}) = p_{i+1|f}(t_{i+1}|t_i) \cdot p_f(t_i) \quad (2)
\]

This equality represents the propagation of the information in the same direction than the sequence of waypoints. As a first physical constraint, in order to respect the arrow of time along the sequence, we impose:

\[
p_{i,f}(t_i, t_j) = 0, \text{ if } t_i \geq t_j, \forall j > i
\]

Now, let’s generalize the joint distribution for an arbitrary number of waypoints:

\[
p_{1:N}(t_{1:N}) = p_{N|1:N-1}(t_N|t_{1:N-1}) \cdot p_{1:N-1}(t_{1:N-1}) = \ldots = \prod_{i=2}^N p_{i|i-1}(t_i|t_{i-1}) \cdot p_f(t_1) \quad (3)
\]

The first equality is obtained with the definition of the joint probability; the second equality requires the assumption of conditional independence also known as the Markov assumption. Then, the process is iterated for each \( T_i \) from \( N-1 \) to \( 1 \) in order to obtain the last equality. Equation 3 is the markovian uncertainty model for the flight plan. On the one hand, \( p_f(t_1) \) is the density function associated to the arrival of the flight in the airspace. On the other hand, \( p_{i|i-1}(t_i|t_{i-1}) \) is the density function, which contains information on the intents of the pilot to arrive at a point given the time of arrival on the previous points.

Sector Model

Here, we give the closed-form equation for computing the exact probability that a sector is congested, which requires \( S_{sf}^t \), the Bernoulli random variable that the flight \( f \) is in the sector \( s \) at time \( t \), and \( S_{sf}^t \) its complementary. Notice that \( \{S_{sf}^t: t \in \Omega\} \) is a stochastic process where \( \Omega \) is the time horizon.

Then, the probability to not be in the sector during the time interval \( \Delta t = [t_{\min}, t_{\max}] \) is the probability to enter after \( t_{\max} \) or the probability to exit before \( t_{\min} \). Because of the arrow of time constraint, these two events are mutually exclusive and one obtain:

\[
P(S_{sf}^t = 0) = P(T_{f\min}^f > t_{\max}^f) + P(T_{f\max}^f \leq t_{\min}^f)
\]

\[
= [1 - P(T_{f\max}^f \leq t_{\min}^f)] + P(T_{f\max}^f \leq t_{\min}^f)
\]

\[
= 1 - F_{T_f}^f(t_{\max}^f) - F_{T_f}^f(t_{\min}^f)
\]

\[
\Rightarrow P(S_{sf}^t = 0) = F_{T_f}^f(t_{\max}^f) - F_{T_f}^f(t_{\min}^f) \quad (4)
\]

When \( (t_{\max}^f - t_{\min}^f) \to 0 \), we obtain the values for \( S_{sf}^t \).

Now, inference on the presence of many flights in a given sector during an interval can be undertaken. To do so, let \( K_s \) be the random variable of the number of flights in the sector \( s \) at time \( t \). Then, by using a multi-index notation, we have:
\[ P(K^t_s = n) = \prod_{|a| = n / f} P(S^t_{s,f})^a \cdot P(S^t_{i,f})^{1-a} \]  

(5)

where \( a = (a_1, ..., a_{N^t_s}) \subseteq (0,1)^{N^t_s}, \quad |a| = a_1 + ... + a_{N^t_s} \) and \( N^t_s = \left\{ t | P(S^t_{s,f}) \neq 0 \right\} \). Again, \( (K^t_s; t \in \Omega) \) corresponds to a stochastic process and these are depicted with diamonds on the graphical model.

**Poisson Binomial Distribution**

Since [15], we know that the congestion model (eq. 5) is a Poisson Binomial distribution. This model is close to the one proposed by [11], except that they are interested by the mean and the variance around the error of prediction to correct the occupancy count. Here, we are interested in the expected number of aircraft in the sector, and so, we need to compute the probability mass function. As an example, if we consider three flights, the equation becomes:

\[ P(K^t_s = 1) = P(S^t_{s,1}) \cdot P(S^t_{s,2}) \cdot P(S^t_{i,3}) \]
\[ + P(S^t_{s,1}) \cdot P(S^t_{i,2}) \cdot P(S^t_{s,3}) \]
\[ + P(S^t_{s,1}) \cdot P(S^t_{s,2}) \cdot P(S^t_{i,3}) \]

As a first remark, the number of conjunctions (products) is determined by the number of combinations \( \binom{N}{n} \) where \( N \) is the total number of flights crossing the sector and \( n \) is an arbitrary number of flights. Consequently, the associated computational burden attains its maximum value at \( n = N/2 \) and decreases when \( n \) goes to 0 or \( N \). As an example, the number of conjunctions is equal to 1.1826e17 when \( N = 60 \) and \( n = 30 \), which is intractable. Also, notice that we are interested in the cumulative density function for every timestamp \( t \). In order to avoid the burden of computation of the direct method, we need to rely on the characteristic function, as demonstrated by [16]. As a matter of fact, the probability can be expressed by:

\[ P(K^t_s = 1) = \frac{1}{N^t_s} \sum_{i=0}^{N^t_s} \exp(-i\omega ln) \]
\[ \cdot \prod_{f=1}^{N^t_s} \left( 1 - P(S^t_{s,f}) + P(S^t_{s,f}) \cdot \exp(i\omega l) \right) \]

where \( i = \sqrt{-1} \) and \( \omega = \frac{2\pi}{N^t_s+1} \). Therefore, this equation can be computed efficiently with the use of a Fast Fourier Transform.

**Flight Intents**

At this point, one would like to manipulate the flight intents more directly, i.e. for the generation of the conditional probabilities and during the optimization process. To do so, let \( \gamma_{t+1} \) be the target time of arrival on the waypoint \( X_{t+1} \) of an arbitrary flight. Now, we make the assumption that the flights have a unique target time of arrival on each waypoint. Then, the conditional probability can be parameterized as \( p_{i+1}(t_{i+1} | t_i; \gamma_{i+1}) \). An acceptable constraint on the space of possible conditional probabilities is to restrain it to unimodal functions where their mode is centered at the target value. Furthermore, we require that its support be bounded to denote the physical constraints of the aircraft, i.e. its flight envelope and its finite amount of fuel. Good candidates for such properties are triangular and gamma probability density functions, already used in project management tools, like PERT, for characterizing the length of a task in a scheduling problem. Then, as depicted on the graphical model, the rectangular nodes act as interfaces between the optimization algorithm and the model for computing the expected cost functions. This corresponds exactly to a black-box optimization approach.

**Optimization Formulation**

Now, we have all the elements to define our multiobjective optimization problem. Because of the stochastic context, one way to define the cost functions is to use their expected values. For the expected cost of delays, let \( \phi_f: \Omega \rightarrow \mathbb{R}_+ \) be the function associated to the cost of delays for the flight \( f \). Then, the expected cost of delays for this flight is:

\[ \mathbb{E}_{\phi_f}(t_{n}^f; \gamma_{|f|}) = \int_{\Omega} \phi_f(t) \cdot p_{n}^f(t; \gamma_{|f|}) dt \]

where \( n^f \) is the number of waypoints in the flight plan \( f \) and so, \( p_{n}^f \) refers to the marginal density function associated to the arrival point \( X_{n}^f \). The inequality ensures that the cost function is bounded for the values in the support of the probability density function. When aggregating these individual functions in order to obtain the associated objective function, one question that immediately arises is equity. In this work, we define the same cost function for every flight and use the super-linear trick, from [2], in order to penalize exponentially any delays. As a consequence, we avoid the case where a flight will be constantly penalized for the benefit of the others. So, we use \( \phi(t) = (t - A_f)^\beta \) where \( A_f \) is the scheduled time of arrival of flight \( f \), \( \beta > 1 \) is the super-linear coefficient and the plus index refers to the positive part. One can also find other relevant cost functions without changing the optimization formulation. In our work, we use:

\[ C_{1}(\gamma) = \sum_{f \in \mathcal{F}} \left[ \int_{\Omega} (t - A_f)^\beta \cdot p_{n}^f(t; \gamma_{|f|}) dt \right] \]

(6)

as the first objective. Here, \( \gamma_{|f|} \) denotes the vector of intents restricted to the ones concerning the flight \( f \). Notice that \( p_{n}^f(t; \gamma_{|f|}) \) is the resulting marginal density function obtained from the marginalizing the joint probability distribution obtained with eq. 2 where all the components of \( \gamma_{|f|} \) are implied in the propagation of the uncertainty. Because \( \mathcal{F} \) is finite and the support of \( p_{n}^f \) is bounded, we know that eq.6 is finite for any targets. In the same manner, we define the cost of congestion of sector \( s \) by \( \psi_s: \mathbb{N} \times \Omega \rightarrow \mathbb{R}_+ \) as: 

\[ \psi_s(\gamma) = \sum_{f \in \mathcal{F}} \left[ \int_{\Omega} (t - A_f)^\beta \cdot p_{n}^f(t; \gamma_{|f|}) dt \right] \]
\( \mathbb{R}^+ \) with the number of congestion flights and the time as arguments.

The expected cost of congestion is:

\[
E_{\psi_s(\cdot, t)}(K_s^t) = \sum_{n=C_s^t}^{N^t_s} \psi_s(n, t) \cdot P(K_s^t = n)
\]

where \( C_s^t \) is the capacity of the sector \( s \) at time \( t \). Let \( \Omega \) be the temporal interval from now to the convergence point.

\[
C_2(\gamma) = \int_\Omega \sum_{s \in S} \sum_{n=C_s^t+1}^{N^t_s} (n - C_s^t)^\lambda \cdot P(K_s^t = n; \gamma) \, dt
\]

Again, \( P(K_s^t = n; \gamma) \) is the resulting probability distribution of the inference done with equation 5, which depends on the intents \( \gamma \). Here, the parameter \( \lambda \) denotes the risk aversion of the controllers when exceeding the capacity. Because \( S \) is finite and the stochastic process will eventually converge toward zero, \( C_2 \) is finite for any targets.

\( C_1 \) and \( C_2 \) are the two criteria of our bi-objective optimization problem. Let \( D \subseteq \mathbb{R}^n \) be the decision space and \( f: D \rightarrow \mathbb{R}^2 \) be the vector-valued cost function. Let \( x \in D \) be a point in our decision space, each dimension of \( f(x) = (C_1(x), C_2(x)) \) denotes a cost associated to the decision. When there is a heavy demand on the airspace, the two costs are antagonist, i.e. reducing the delays will induces more flights in the airspace and, as a consequence, will increase the congestion probability. The relation of Pareto dominance captures this idea. Let \( x, y \in D \) be two decisions, then \( x \) dominates \( y \), denoted by \( x < y \), iff \( C_i(x) \leq C_i(y), \forall i \in \{1, 2\} \) and \( \exists j \in \{1, 2\} | C_j(x) < C_j(y) \).

**Constraint**

From the optimization algorithm point of view, the intents shall be bounded with the flight envelope. However, assigning a value to a target will impact the bounds of the subsequent targets. To obtain independent decision variables, the optimization algorithm works with flight durations in sectors. These bounds are hard constraints, which cannot be violated in order to find better solutions and define feasible intervals. There is a distinction between feasible intervals and probable intervals defined by the supports of the marginal distributions. So, for a given point, a probable interval must be a subset of the feasible interval.

Therefore, here, we only consider box constraints

\[
\gamma_i \in \left[ \frac{\beta_i \cdot V_{i+1}}{d_{i+1}}, \frac{\alpha_i \cdot V_{i+1}}{d_{i+1}} \right],
\]

which are easily taken into account in evolutionary algorithms in general. As in [8], an en-route flight can have a maximum of speed up rate of 1 minute per 20 minutes and a maximum slow down rate of 2 minutes per 20 minutes. During experimentations, we notice that speed changes only are insufficient to reduce the cost of congestion, considering that the mean duration in sectors is around 20 minutes. Also, we add the possibility to increase the distance in sectors by a factor \( \delta \) and we introduce the possibility for ground delays. So, we obtain the intervals:

\[
\gamma_i \in \left[ \frac{\beta_i \cdot V_{i+1}}{d_{i+1}}, \frac{\alpha_i \cdot V_{i+1}}{d_{i+1}} \right] \quad \text{with } \delta = 1.05
\]

**Monte-Carlo Verification**

In order to verify experimentally the inference equations, we rely on Monte-Carlo approximation. First, we need to be able to sample from the flight model of each flight (see Equation 3). Because of the structure of our graphical model, a simple forward sampling technique can be used with the conditional probabilities of the model. A sample is expressed as a vector \( x_f^{(k)} \sim p_f (t_{1:N}) \) with a time of overflight for each point of the flight \( f \). Then, we can construct an indicator function per flight to represent the fact that the flight is in the sector at time \( t \). We will denote these by \( \phi_f^{(k)}(t) = \chi_{[\delta f(t_1), \delta f(t_N)]}(t) \), where \( \chi \) is the indicator function. Then, we can count the number of flight in the sector for any time: \( s^{(k)}(t) = \sum_{f \in F} \phi_f^{(k)}(t) \). To determine if the sector is congested, we use again the indicator function with the set \( C \) of values higher than the capacity: \( c^{(k)}(t) = \chi_C(s^{(k)}(t)) \). Finally, the Monte-Carlo approximation is given by:

\[
c_C(t) = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} c^{(k)}(t) \quad \text{where } \mathcal{K} \text{ is the set of Monte-Carlo simulations.}
\]

At this point of the research, the proof of convergence of the Monte-Carlo Routine toward the inference equation for the probability of congestion is unknown. The main difficulty comes from the fact that the closed-form equation concerns one timestamp at a time instead of the Monte-Carlo routines, which generate time intervals. Therefore, in this study, we solely verify experimentally that the two methods return the same result.

**Experiment**

In this section, the experiment on reducing the congestion with a multiobjective optimization algorithm on a probabilistic model of the trajectories and the sectors congestion is described. The first goal is to numerically validate the theoretical model defined above, and to assess the propagation of the uncertainty from the trajectories to the sector. The second goal is to assess whether NSGA-II can actually solve the multiobjective optimization problem. All design variables here are continuous, and the variation operators have been chosen according to [17]. More precisely, all experiments use the simulated binary crossover operator (SBX) and the polynomial mutation, which can handle directly the box constraints. The other parameters of the algorithm are the population size, the probability of crossover, the probability of mutation, the exploration coefficients for the operators, and the maximum number of generations before termination.

In this section, the experiment on reducing the congestion with a multiobjective optimization algorithm on a probabilistic model of the trajectories and the sectors...
congestion is described. The first goal is to numerically validate the theoretical model defined above, and to assess the propagation of the uncertainty from the trajectories to the sector. The second goal is to assess whether NSGA-II can actually solve the multi-objective optimization problem. All design variables here are continuous, and the variation operators have been chosen according to [17]. More precisely, all experiments use the simulated binary crossover operator (SBX) and the polynomial mutation, which can handle directly the box constraints. The other parameters of the algorithm are the population size, the probability of crossover, the probability of mutation, the exploration coefficients for the operators, and the maximum number of generations before termination.

**Assumption**
First of all, inside the probabilistic model, we need to discretize the temporal horizon in order to compute numerically the integrals. We choose a time step of one minute because we believe that it is under the order of magnitude of the precision in real world. This choice affects the accuracy of the evaluation of the uncertainty and therefore, is an internal parameter for the probabilistic model and can be used to control the trade-off between accuracy and computational burden. This parameter is completely hidden from the optimization algorithm because of the black-box optimization approach. For this reason, we can use a continuous domain for the decision variables of the optimization problem that will permit any arbitrary small value for this parameter. Thereafter, we assume that the feasible interval length for the first point is from 5 minutes to 60 minutes and the support length of the probability distribution cannot be less than 15 minutes. For the next points, we consider solely flights with constant level flight above FL300 with a true airspeed of 460 knots (MACH 0.78). With $\alpha = 0.9$ and $\beta = 1.05$, the flight can lower its speed to 414 knots (MACH 0.72) and increase it to 483 knots (MACH 0.82).

Also, we need to define the conditional probability for every flight at every point, which can require a lot of information. Instead, an interesting way to do it is to assume a nominal policy. The chosen one is, for each flight at each point, the flight management system of the aircraft tries to maximize the probability to arrive at the next point at the given target. A way to encode the policy is to take a probable interval and to map a triangular distribution over it where the intent is the mode of the distribution. In our case, we choose a probable interval length equals to a confidence interval of 95% on the distribution obtained in [11] that is a length of 24 minutes.

**Experimental Setting**
The chosen instance for this study (Figure 2) implies 10 flights and 5 sectors. There is a central sector that every flight must cross. The capacity of this central sector is only three, two for the northwest and southeast sectors and one for the southwest and northeast sectors. We also show the departure nodes with the routes toward the sectors. The underlying decision space consists in a 60 dimensional space. A second instance was used to assess the performance of the inference algorithm in the probabilistic algorithm.

The instance comprises 300 flights, regrouped in 10 flows, each crossing 4 sectors. These are defined on a 4x4 grid and the flows arrives from the north and the east, including the diagonals from northwest and southeast. The model was coded in C++ and simulated on a 2.2GHz processor. For the algorithm using the characteristic function, we used the FFTW library [18]. The time required to simulate the model, i.e. for a function evaluation, is around 113ms with the direct method for the first instance and 3 seconds for the second with the FFT algorithm. As a matter of fact, we can see from on the log-scale of Figure 3 that the FFT algorithm is clearly faster than the direct one. For NSGA-II, we used the Paradiseo Library also in C++. This solver can be freely downloaded on the INRIA’s forge at: http://paradiseo.gforge.inria.fr/. In order to assess the performance of the algorithm, we used the hypervolume indicator implemented in PISA [19]. So we choose a population size of 100 individuals, a crossover probability of 0.8 with an expanding coefficient of 20 for the SBX operator, a mutation probability of 0.2 and an expanding coefficient of 20 for the polynomial mutator. Also, the maximum number of generations allowed is 400. To verify the model, we used the Monte-Carlo routine defined previously. The associated results are depicted on the different figures with crosses. As we will see, Monte-Carlo coincides with the closed-form equations.
Analysis

Now, we will analyze each step of the methodology. First, we need to compute the marginal probabilities over waypoints, as depicted on, with the trajectory model defined by Equation 3 (Figure 4). We can see that the value of the modes decreases and the support of the distributions increases with time. This simply translates the fact that uncertainty on the target time of arrival increases with time. Then, we notice that the mean difference between the modes of the marginals is around 7 minutes higher than the expected 30 minutes required for crossing a sector. This is mainly due to the lower bounds of the feasible and probable intervals. When the computed lower bound of the probable interval is lower than the feasible interval, the difference is added to the upper bound and the lower bounds of the two sets become equal.

In summary, the expected time of arrival can be shifted in time, but after a while, there is no more information on if the flight will arrive before or after this expected value.

Then, with Equation 4, we compute the stochastic process for a flight to be in a sector over time (Figure 5). We can see that the value of modes decrease with time. Also, the rate of change of the stochastic process depends on the value of the marginals and also, it decreases with time. Intuitively, it corresponds to the fact that there is more uncertainty in the moment of coordination between sectors, that is to say the overlapping regions of the different curves grow in time. We can also visualize the fact that the probability for an aircraft to be in one of the sectors is equal to 1 by summing the probability at every timestamp. Besides, we can expect that the function defining the stochastic process will be unimodal since there is no reason for the probability to vary while the aircraft is inside the sector.

Thereafter, we need to compute the probability that the sector is congested at a given time, using Equation 5. Figure 6 shows the stochastic process associated to the northeast sector. Again, even if the instance is symmetric in distance, the probability for this sector to be congested decreases with time since the variance of the marginals increases with time. It means that, with luck, some aircraft will cross the sector before the others without any regularization.
Figure 8: Hypervolume Indicator
As a matter of fact, this model is a way to quantify the effectiveness of regularizations. Combined with an optimization algorithm, we are able to put priority on actions affecting events that will occur before others, or at least, on events with more confidence of occurrence. An interesting fact is that it is accomplished naturally without the use of a discount factor.

When all these distributions are known, the probabilistic model can compute the expected cost of delays and congestion using equations 6 and 7 respectively. One way to understand the cost functions from a computational point of view is to add, for all possible timestamps on the temporal horizon, the cost function at a given timestamp multiplied by the probability at this timestamp. Consequently, minimizing the probability of congestion for every timestamp will effectively minimize the expected cost of congestion. As a matter of fact, this computation is the bottleneck in terms of computational burden of the evaluation function, but can greatly benefit from a parallelization.

Now, we can analyze the Pareto front obtained with the last population of 11 runs (Figure 7). At the first glance, the runs generate different fronts, which cover a large region, especially when minimizing the expected cost of congestion. So, minimizing the delays is relatively easy when we do not consider congestion. This is why the upper part of the Pareto fronts overlap. But, when we try to minimize the congestion, it seems that the algorithm falls in local optimum. To verify this explanation, we used the hypervolume indicator, which gives the volume determined by the area enclosed by the Pareto front and the worst possible point. Figure 8 confirms the premature convergence since the hypervolume indicator stabilizes around generation 150 for each run. In future work, we will tune the parameters of the algorithm according to the hypervolume indicator. We expect to find good parameters, maybe by simply increasing the exploration ratio, to avoid these local optima.

Discussion
One clear limit of this work is that the proposed theoretical framework has been validated on a single instance, assessing at the same time both the uncertainty model and the optimization process. Further experiments are mandatory to draw any firm conclusion regarding either the model or the optimization algorithm, and these experiments must involve several other instances. Here, the second instance was created to assess the computation time of the probabilistic model. We notice that the mean cost functions decrease during the optimization, but further investigation is necessary to understand how the algorithm impact decision variables that are not related to the disruption. We think that the second instance is a good starting point to create different variant of disruptions and to assess that the approach is able to tackle the problem.

Even if the algorithm returns many solutions, a human operator will not be able to understand each of them in real-time. Consequently, the multiobjective algorithm shall return only a subset of interesting solutions. All the difficulty comes from the definition of what is an interesting solution. One way to circumvent this difficulty is to use the diversity operator used in several population-based multiobjective algorithms. A diversity operator is used to filter solutions, but at the same time, to preserve the shape of the Pareto front. This prevents from having solutions in the same region. A necessary parameter is the size of the archive containing the solutions. Here, we used the crowding distance measure, inherent to NSGA-II with an archive size of 100 solutions.

The triangular distribution is an interesting choice because its variance increases as the mode approaches the bounds. In our case, the cost function is very similar to the variance formula, the target replaces the mean and we take only the positive part. Hence, we verified experimentally that the cost function increases as the target approaches the boundaries. This is relevant for the optimization algorithm since we penalize any target that would be close to the boundaries of the feasible set.

Regarding the probabilistic model, because the uncertainty on sectors are evaluated using the closed-form equations, the expected cost functions can be computed exactly for any instance. We tested the inference mechanism with instances with a maximum number of flights equals to 1000. Even with such a number of variables, the congestion was evaluated exactly with a computational time inferior to 0.15 seconds. Regarding the optimization algorithm, there is very little hope to ever formally prove its convergence. Hence methods from experimental sciences must be used here. Statistics over many random instances of the size of a real operational context are the way to go, assessing how often and in which contexts the method can fails.

The choice of NSGA-II was motivated by the fact that we still do not fully understand all the properties underlying the probabilistic model. Hence, instead of trying to guess possibly wrong assumptions, we have chosen to use a robust and general optimization algorithm. Nevertheless, it is important in the future to compare NSGA-II with other,
more recent multiobjective optimization algorithms. Therefore, an extensive statistical study is needed, in order to find the most suited algorithm for this kind of problem.

Moreover, the formulation of the tactical planning problem shall also be extended to include more operational constraints and their effects on the algorithms. Also, experiments and data mining shall be done in an operational context in order to model more accurately the underlying uncertainty. Moreover, the uncertainty could represent the expected errors of the trajectory prediction. This would create a smooth transition between both phases. Since we are in an online context, particle filters could also be used at the trajectory level in order to estimate the probability to be at the next points and therefore, estimate the probability of congestion. With recent statistical studies on real trajectories, we know that the uncertainty is different from one phase to another. Indeed, there is important uncertainty on the real departure time and on the climbing phase. Afterward, the uncertainty is low in en-route phase and so, the values used here are certainly too high. In the following, we should configure the model to reflect the uncertainty for the different phases. Finally, we have mentioned the use of stochastic processes for the probability that the flight is in a sector and for the probability of congestion. We believe that there are interesting issues that can be addressed with the novel techniques from this domain.

Conclusion
This article has introduced a probabilistic model to handle the propagation of the uncertainty from the trajectories to the sectors. The prerequisites of the model are the marginal of the initial arrival in the airspace, the policy of the flight management system when trying to stick to the target schedule, and the potential external disruptions. Thereafter, the equation for computing the probability for a flight to be in a sector can be computed. From there on, the closed-form equation to compute the probability of congestion can be derived.

Then, some general formulations for the expected cost of delays and the expected cost of congestion were given. We used a well-known trick to ensure equity that was naturally integrated in the model. Finally, because the congestion measure is clearly not the only criterion that should be used to decide for a schedule, the well-known multiobjective algorithm NSGA-II was proposed to solve the bi-objective problem of minimizing both the congestion and the cumulated delays of the flights, i.e., to approximate the non-dominated solutions of the Pareto front.

Furthermore, in order to illustrate how the theoretical model can be useful in practice, we presented some results on two instances. On-going and further work will investigate other MOEAs to replace NSGA-II, and, more importantly, several different instances. One crucial issue is how well this algorithm scales with the problem complexity (number of flights and number of sectors). Nevertheless, we are confident that further studies will demonstrate the robustness of the proposed approach of using multiobjective evolutionary algorithms to solve the stochastic and dynamic optimization problem of air traffic flow and capacity management.

ACKNOWLEDGMENT
Gaétan Marceau is funded by the scholarship CIFRE 710/2012 established between Thales Air Systems and the TAO project-team at INRIA Saclay Ile-de-France, and the scholarship 167544/2012-2013 from the Fonds de Recherche du Québec - Nature et Technologies.

REFERENCES


Multiphase Mixed-Integer Optimal Control Applied to 4D Trajectory Planning in Air Traffic Management

Alberto Olivares*, Manuel Soler*, and Ernesto Staffetti*.
*URJC: Department of Signal Theory and Communications. Fuenlabrada, Madrid, Spain
manuel.soler, alberto.olivares and ernesto.staffetti@urjc.es

ABSTRACT
In this paper an approach to aircraft trajectory optimization is presented in which integer variables and continuous variables are considered. Integer variables model decision making processes, and continuous variables describe the state of the aircraft which evolves according to differential-algebraic equations. The problem is formulated as a multiphase mixed-integer optimal control problem. It is transcribed into a mixed integer nonlinear programming problem by applying a 5th degree Gauss-Lobatto direct collocation method and then solved using a nonlinear programming based branch-and-bound algorithm. The approach is applied to the following en-route flight planning problem: given an aircraft point mass model, a wind forecast, a 3D airspace structure, and the relevant flying information regions with their associated overflying costs, find the control inputs that steer the aircraft from the initial fix to the final fix following a route of waypoints and performing step climbs, while minimizing certain performance indexes in which fuel based, environmental based, time based, and overflying based costs are considered during the flight. The decision making process arises in determining the optimal sequence of waypoints and the optimal sequence of flight levels. The optimal times at which the step climbs are performed and the waypoints are to be overflown are also to be determined. Numerical results are presented and discussed, showing the effectiveness of the approach.

Categories and Subject Descriptors
D.4.8 [Performance]: modeling & prediction, simulation; J.2 [Physical Sciences and Engineering]: Aerospace

General Terms
Simulation

Keywords
4D Trajectory Optimization, Air Traffic Management, Multiphase Mixed-Integer Optimal Control, Minlp.

1. INTRODUCTION
In the future Air Traffic Management (ATM) system the trajectory becomes the fundamental element of a new set of operating procedures collectively referred to as Trajectory-Based Operations (TBO) [1]. TBO will provide the capabilities, decision-support tools, and automation to manage aircraft movement by trajectory. This shift to a trajectory-based Air Traffic Control (ATC) will enable aircraft to fly negotiated flight paths in which ATC will issue restrictions to be met and the airline will decide the most economical way to meet them. In this way, the TBO concept will allow more flexible and efficient airspace usage, resulting in more efficient trajectories.

With such underlying motivation, this paper presents an approach to aircraft trajectory optimization in which both continuous and integer variables are included. Integer variables model decision making processes and continuous variables describe the state of the aircraft which evolves according to differential-algebraic equations.

Trajectory optimization has been studied for several decades. A survey on numerical methods for trajectory optimization is given in [2]. The trajectory optimization problem can be formulated as an optimal control problem [3, Chap. 2-3]. Practical issues for solving optimal control problems are discussed in [4, Chap. 4].

In the flight of an aircraft, several flight phases can be distinguished. Thus, different flight models can be used for different phases, resulting in a multistage or multiphase trajectory optimization problem [5], which can be typically formulated as multiphase optimal control problems. Many instances have been solved for practical applications in aerospace engineering. In [6] and [7], a unmanned air vehicle flight mission is solved considering the route as a given sequence of waypoints. In [5] and [8] the minimum fuel trajectory problem for commercial aircraft is discussed for a given sequence of phases constituting the flight plan. Multi-stage problems for space flight trajectory optimization have also been solved. They include multi-stage launch [9], multi-stage ascent [10], or multi-stage orbit transfer [11].

However, due to an increasing sophistication of both manned and unmanned space missions, and due to the complex nature of the airspace structure in atmospheric flights, decision variables may be needed for better modeling of trajectory optimization problems [12]. Therefore, integer or binary variables are introduced to model decision making processes in optimal control problems. Few works considering decision variables have been presented in the scope of aircraft trajectory optimization: In [12], two examples are solved, an asteroid mission and a refueling mission, using a pseudospectral knotting method to generate a mixed-variable programming problem; in [13, 14], evolutionary algorithms including decision variables are derived for space mission planning. Indeed, the inclusion of binary variables into nonlinear optimal control problems has remained a fundamental limitation. However, this has proven to be doable under an
ATM constrained environment in [15, 16], where a multi-phase mixed-integer optimal control approach to trajectory optimization was presented. In this work, a horizontal flight with waypoint allocation was solved while minimizing fuel and overfly charges.

Thus, the main contribution of this paper is to extend the multiphase mixed-integer optimal control approach in [15] to solve a 4D trajectory optimization problem with 0-1 binary decision variables for flight level allocation and waypoint allocation considering a trade-off between different cost sources in the objective function.

The problem to be studied can be described as follows: given an aircraft point mass dynamical model, a wind forecast, an airspace structure, and the Flying Information Regions (FIRs) with their associated overflying charges, find the control inputs that steer the aircraft from the initial fix to the final fix following a route of waypoints and performing step climbs, while minimizing certain performance indexes in which fuel based, environmental based, time based, and overflying based costs are considered during the flight. The decision making process arises in determining the ordered set of waypoints and flight levels to be overflown. The times at which they are reached are also to be determined.

Such a problem can be formulated as a multiphase Mixed-Integer Optimal Control Problem (MIOCP) [17], [18]. The multiphase MIOCP is transcribed into a MINLP problem, first converting the multiphase optimal control problem into a conventional optimal control problem by making the unknown switching times part of the state [19], then applying a collocation method based on high-order Gauss-Lobatto quadrature rules [20] to convert the dynamic equations of the system into constraints, and finally introducing binary variables to model the decision making process. Thus the resulting optimization problem is a MINLP problem, which is solved using the solver Bonmin [21] which implements a NLP Branch & Bound algorithm. Details of the algorithm in a multiphase MIOCP framework can be found in [15, 16].

The general multiphase MIOCP is stated in Section 2. The solution approach is presented in Section 3. In Section 4, the application to en-route trajectory optimization is presented. In Section 5 results are reported and discussed. Finally Section 6 contains the conclusions.

2. PROBLEM FORMULATION

The multiphase motion of an aircraft can be modeled by a set of differential-algebraic dynamic subsystems:

\[ \dot{x}(t) = f(x(t), u(t), z), \]
\[ 0 = g(x(t), u(t), z), \]

where \( q = 1, \ldots, N \) is the index of phases, \( t \in [t^I, t^F] \subset \mathbb{R} \) is the time, \( x(t) \in \mathbb{R}^{n_x} \) is the state variable in phase \( q \), \( u(t) \in \mathbb{R}^{n_u} \) is the control variable in phase \( q \), \( z \in \mathbb{R}^{n_z} \) represents a vector of parameters. Let \( t^I = \bar{t}^I \leq \cdots \leq \bar{t}^N = t^F \) be the switching times between phases. Thus, in the time interval \( [\bar{t}^{q-1}, \bar{t}^q] \) \( q = 1, \ldots, N \) the system evolution is governed by the differential-algebraic subsystem \( q \), being \( \bar{t}^q = t^q \) and \( \bar{t}^{q-1} = t^{q-1} \). At time \( \bar{t}^q \) with \( q = 1, \ldots, N - 1 \) the algebraic-dynamic system corresponding to \( q \) to \( q + 1 \). The decision making processes are modeled by means of one time independent vector of binary variables \( v^q \in \{0,1\}^{n_v^q}, q = 1, \ldots, N - 1 \), whose components are \( v^q, j = 1, \ldots, n_v^q \).

2.1 Formulation of the multiphase MIOCP

The multiphase MIOCP can be stated as follows [17, Chap. 1]:

\[ \min J(x, u, v^q, z) = \sum_{q=1}^{N-1} \int_{t^{q-1}}^{t^q} L^q(x(t), u^q(t), z) \, dt \] (1a)
\[ + \sum_{q=1}^{N-1} \Psi(x(\bar{t}^q), v^q, z). \] (1b)

subject to:

\[ \dot{x}(t) = f^q(x(t), u^q(t), z), \quad t \in [\bar{t}^{q-1}, \bar{t}^q], q = 1, \ldots, N, \] (1c)
\[ g^q(x(t), u^q(t), z) = 0, \quad t \in [\bar{t}^{q-1}, \bar{t}^q], q = 1, \ldots, N, \] (1d)
\[ \nabla g^q(x(t), u^q(t), z) \nabla x \geq 0, \quad t \in [\bar{t}^{q-1}, \bar{t}^q], q = 1, \ldots, N, \] (1e)
\[ x^{I+1}_{\ell} = x^I, \quad \Psi(x^{N+1}) = 0, \] (1f)
\[ r^{ineq}(\bar{x}^q, \bar{v}^q, z) \leq 0, \quad q = 1, \ldots, N, \] (1g)
\[ r^{suc}(\bar{x}^q, \bar{v}^q, \bar{z}) = 0, \] (1h)
\[ r^{suc}(\bar{x}^q, \bar{v}^q, \bar{z}) = 0, \] (1i)
\[ \int_{t^{q-1}}^{t^q} E^q(x^q(t), u^q(t), z) \, dt = \Phi, \] (1j)
\[ \int_{t^{q-1}}^{t^q} F^q(x^q(t), u^q(t), z) \, dt = \Phi, \] (1k)

subject to: (1a) \( \bar{t}^q = t^q \) \( q = 1, \ldots, N - 1 \). The decision making processes are modeled by means of one time independent vector of binary variables \( v^q \in \{0,1\}^{n_v^q}, q = 1, \ldots, N - 1 \), whose components are \( v^q, j = 1, \ldots, n_v^q \).

The terms of the objective functional (1b) are in Bolza form and contain a Lagrange term \( \int_{t^{q-1}}^{t^q} L^q(x(t), u^q(t), z) \, dt \) and a Mayer term \( \int_{t^{N-1}}^{t^N} F^q(x^q(t), u^q(t), z) \, dt \). Equation (1d) and Equation (1e) with \( f^q \in \mathbb{R}^{n_x} \) and \( g^q \in \mathbb{R}^{n_z} \) are the equations of the differential-algebraic system in phase \( q \). Equations (1f) with \( c^q \in \mathbb{R}^{n_c} \) are the inequality constraints in phase \( q \). Equations (1g) and Equation (1h) are the boundary conditions of the problem. Equations (1i) with \( r^{ineq} \in \mathbb{R}^{n_{ineq}} \) and Equations (1j) with \( r^{suc} \in \mathbb{R}^{n_{suc}} \) are the inequality and equality interior-point constraints, respectively.

Equations (1k) are the transition conditions between phases to ensure continuity, which are usually of the form \( x^{q+1}(t^{q+1}) = x^q(t^q) \), for \( q = 1, \ldots, N - 1 \). These conditions are also referred to as the linkage conditions. \( f^q \in \mathbb{R}^{n_x} \) is assumed to be piecewise Lipschitz and \( \partial f^q/\partial x \) is assumed to be regular. \( L^q, F^q, c^q, r^{ineq}, \) and \( r^{suc} \) are assumed to be twice differentiable. The dimensions \( n_x, n_u, n_z, n_c, n_{ineq}, n_{suc} \) and \( n_y, n_g \) can be different for each phase.

The solution of this problem is composed by the functions \( x^q(t), u^q(t), q = 1, \ldots, N, t \in [t^I, t^F] \), the switching times \( \bar{t}^q, q = 1, \ldots, N - 1 \), the vector of binary variables \( v^q, q = 1, \ldots, N - 1 \), and the vector of parameters \( z \). Note that \( t^I = t^F \) is also a variable of the problem if the final time is unknown.

A more general form of multiphase mixed-integer optimal control problem that includes time dependent binary control functions can be found in [17] and [18].

3. MIOCP SOLUTION APPROACH

3.1 Treatment of the switching times

The multiphase MIOCP is tackled first making the unknown switching times part of the state vector and then introducing a new independent variable with respect to which the switching times are fixed. In this reformulated problem, there is a linear relation between the new variable and time, but the slope of this linear relation changes on each interval between two switches. These slopes are actually time scaling factors that determine the optimal switching times. The reformulated problem is a conventional MIOCP. More details on this technique can be found in [19].
5th degree Gauss-Lobatto collocation method

A collocation approach has been used in which integration rules are based on a particular family of Jacobi interpolating polynomials that give rise to the so called Gauss-Lobatto quadrature rules. In particular, the fifth-degree Gauss-Lobatto integration scheme has been used [20]. The motivation behind the use of 5th degree Gauss-Lobatto quadrature rules is the reduction of the number of variables of the problem to achieve an acceptable computational time. In this class of numerical methods the optimal control problem is converted into a NLP problem.

The mathematical details of this formulation can be consulted in [20]. Also, in [16] they can be found with an ad-hoc formulation for the multiphase Miocp approach.

3.3 MINLP resolution

Consider fixing the binary vectors $v$ for $q = 1, \ldots , N - 1$. In such a case, the multiphase Miocp would become a conventional optimal control problem due to the transformation of Section 3.1 and can be transformed to an NLP problem as explained in Section 3.2.

If the sequence of modes is allowed to vary, the NLP becomes a MINLP due to binary decision variables. MINLP is the mathematical problem of minimizing a function in a feasible region described as the intersection of a nonlinear set and integrity. In all generality, MINLP is an undecidable problem but if one assumes that the feasible region is bounded, which is the case here, it is NP-Hard. Although bounded MINLP problems can be solved in theory, they are one of the most challenging problems in computational optimization, in particular, when the nonlinearities are not convex, as it is the case for the aircraft dynamics and constraints. A simple algorithm for determining the mode sequence could be to enumerate all possible values for $v$ and solve the associated optimal control problems and pick the best solution. Unfortunately, this method is impractical for more than a few binary variables. A common approach to address larger problems is to do an implicit enumeration via the Branch & Bound algorithm. We provide a sketch of this method to stress the particularities that arise in applying it to solve the Miocp.

3.3.1 Branch & Bound Algorithm

Branch & Bound is a divide-and-conquer method. The problem is divided by partitioning the set of feasible solutions into smaller and smaller subsets. The conquering is done by computing bounds on the cost of the best feasible solution in each subset and discarding subsets whose lower bound exceeds a known feasible solution. Branch & Bound is an exact algorithm when the branch used in each subproblem is a valid lower bound. In our case, to find a lower bound for the multiphase Miocp, we relax the binary variables, that is, let $w \in [0, 1]$ and solve the associated NLP. However, obtaining a true lower bound on the value of the multiphase Miocp is a difficult task due to presence of nonconvex dynamics and constraints in the NLP. Thus, our solution is heuristic. In our case, we have used the solver Bonmin [21] which implements a NLP Branch & Bound algorithm. Bonmin is an open-source MINLP solver implementing several different algorithms for solving mixed integer nonlinear optimization problems. Source code and binaries of Bonmin are available from COIN-OR (http://www.coin-or.org). We call Bonmin through the AMPL modeling language.

Mathematical details on the the transformation into a MINLP problem and on the NLP Branch & Bound employed algorithm can be found in [15, 16].

4. CASE STUDY

We optimize the trajectory of A330-301 aircraft performing the en-route part of a flight New York-Rome between the waypoints YAHOO ($p^t = (11.60\degree E, 41.60\degree N)$) as initial fix and the waypoint AMTEL ($p^e = (26.74\degree N, 45.60\degree W)$) as final fix. The route to be flown allows two step climbs to be performed and seven different flight levels (FL285, FL300, FL315, FL330, FL345, FL360 and FL375). The initial flight level is FL300. The initial boundary conditions of the problem are: $V(t_i) = 235 \text{ [m/s]}, \gamma(t_i) = 0\degree, \chi(t_i) = 0\degree, m(t_i) = 174000 \text{ [Kg]}$.

In order to show the effectiveness of the approach, three different problems have been solved considering different objective functions:

- Fuel costs and overfly charges.
- Fuel costs, time-based costs, and overfly charges.
- Fuel costs, overfly charges, and contrail-based costs.

4.1 Dynamics of flight

In order to plan fuel optimal aircraft trajectories, it is common to consider a 3 degree of freedom dynamic model that describes the point variable-mass motion of the aircraft over a spherical flat-earth model. We consider a symmetric flight, that is, we assume there is no sideslip and all forces lie in the plane of symmetry of aircraft. Wind is included due to its considerable effects on fuel consumption. BADA 3.6 is used as aircraft performance model [22]. The equations of motion of the aircraft are:

$$ (R_e + h) \cos \theta \chi = V \cos \gamma \sin \chi + V_{wx}, \quad (2a) $$
$$ (R_e + h) \dot{\theta} = V \cos \gamma \cos \chi + V_{wy}, \quad (2b) $$
$$ h = V \sin \gamma + V_{wb}, \quad (2c) $$
$$ n = -\eta T, \quad (2d) $$
$$ mV \cos \gamma = L \sin \mu, \quad (2f) $$
$$ mV \sin \gamma = L \cos \mu - mg \cos \gamma, \quad (2g) $$

![Figure 1: Aircraft state and forces](image)

In the above, the three kinematic equations (Eq. (2a-2c)) are expressed in a ground based reference frame $(x_e, y_e, z_e)$ and the three dynamic equations (Eq. (2e-2g)) are expressed in an aircraft-attached reference frame $(x_w, y_w, z_w)$ as shown in Figure 1. The states are: $x, y, z$ referring to the aircraft 3D position (longitude, latitude, altitude); $V, \gamma, \chi$ referring to the true airspeed, heading angle, and flight path angle respectively. $R_e$ is the radius of earth; $m$ is aircraft mass, $\eta$ is the speed dependent fuel efficiency coefficient. $V_{wx}, V_{wy}, V_{wb}$ are components of the wind, $T$ is the thrust, and $\mu$ is the bank angle. Lift $L = C_L S q$ and drag $D = C_D S q$ are the components of the aerodynamic force, $S$ is the reference wing surface area and $\frac{q}{2} \rho V^2$ is the dynamic pressure. A parabolic drag polar $C_D = C_{D0} + K C_L^2$ and a standard atmosphere are assumed. $C_L$ is a known function of the angle of attack $\alpha$ and the Mach number. The aircraft position in 2D is approximated as $x = (R_e + h) \cos(\theta) \lambda$ and $y = (R_e + h) \theta$. The bank angle $\mu$, the engine thrust $T$, and the coefficient of lift $C_L$ are the inputs, that is, $u(t) = (T(t), \mu(t), C_L(t))$. For further details on aircraft dynamics, please refer to [23].
4.1.1 Horizontal flight dynamics

The 3DOF equations governing the translational horizontal motion of an airplane are the following:

\[ m\ddot{V} = T - D, \quad \lambda R_c \cos \theta = V \cos \gamma \cos \chi + W_x, \]
\[ mV \dot{\gamma} = L - mg \sin \gamma, \quad \theta R_c = V \cos \gamma \sin \chi + W_y, \]
\[ L \cos \mu = mg, \quad \dot{m} = -\eta T, \]

In general, the engine thrust \( T \) and bank angle \( \mu \) are the control variables of the aircraft, that is \( u = (T, \mu) \). The thrust is commanded by the engine throttle and the bank angle is commanded combining rudder and ailerons trims. The state vector red is \( x = (\lambda, \theta, V, \gamma, m) \), where \( \lambda \) is de longitude, \( \theta \) the latitude, \( V \) the True Air Speed, \( \gamma \) the flight path angle and \( m \) the mass of the aircraft. \( W_x \) and \( W_y \) correspond to the east and north wind components respectively. \( R_c \) is radius of earth and \( \eta \) corresponds to specific fuel consumption.

4.1.2 Vertical flight dynamics

The 3DOF equations governing the translational vertical motion of an airplane are the following:

\[ m\ddot{V} = T - D - mg \sin \gamma, \quad \lambda R_c \cos \theta = V \cos \gamma \cos \chi + W_x, \]
\[ mV \dot{\gamma} = L - mg \cos \gamma, \quad \theta R_c = V \cos \gamma \sin \chi + W_y, \]
\[ \dot{m} = -\eta T, \quad \dot{h} = V \sin \gamma. \]

In general, the engine thrust \( T \) and bank angle \( C_L \) are the control variables of the aircraft, that is \( u = (T, C_L) \). The thrust is commanded by the engine throttle and the coefficient of lift is commanded with elevator trims. The state vector red is \( x = (\lambda, \theta, V, \gamma, m) \), where \( \lambda \) is de longitude, \( \theta \) the latitude, \( V \) the True Air Speed, \( \gamma \) is the flight path angle and \( m \) the mass of the aircraft. \( W_x \) and \( W_y \) correspond to the east and north wind components respectively. \( R_c \) is radius of earth and \( \eta \) corresponds to specific fuel consumption.

4.1.3 Path constraints

The path constraints of the problem are those that define aircraft’s flight envelope given in BADA database manual [22]:

\[ 0 \leq h(t) \leq \min[h_{\text{MO}}, h_u(t)], \quad \gamma_{\text{min}} \leq \gamma(t) \leq \gamma_{\text{max}}, \]
\[ M(t) \leq M_{\text{MO}}, \quad \dot{m}_{\text{min}} \leq \dot{m}(t) \leq \dot{m}_{\text{max}}, \]
\[ V(t) \leq \dot{a}_u, \quad C_L V_c(t) \leq V(t) \leq V_{\text{MO}}, \]
\[ \dot{\gamma}(t)V(t) \leq \dot{a}_u, \quad 0 \leq C_L(t) \leq C_{L_{\text{max}}}, \]
\[ V_{\text{min}}(t) \leq T(t) \leq V_{\text{max}}, \quad \mu(t) \leq \mu. \]

In the above, \( h_{\text{MO}} \) is the maximum reachable altitude, \( h_u(t) \) is the maximum operative altitude at a given mass (it increases as fuel is burned); \( M(t) \) is the Mach number and \( M_{\text{MO}} \) is the maximum operating Mach number; \( C_L \) is the minimum speed coefficient, \( V(t) \) is the stall speed and \( V_{\text{MO}} \) is the maximum operating calibrated airspeed; \( \dot{a}_u \) and \( \dot{a}_u \) are respectively the maximum normal and longitudinal accelerations for civilian aircraft. Note that several flight envelope constraints are nonconvex.

4.2 Wind data

To take into account the influence of wind, wind forecast provided by the National Oceanic and Atmospheric Administration (NOAA) Forecasts System Laboratory (FSL) via GRidded Binary (GRIB) files are considered. GRIB files provide wind forecasts as tabular data that give the three components (vertical, north and east) of the wind vector at each node of the grid. Vertical wind is neglected due its low influence.

Wind forecast tabular data are fitted into analytical functions by means of nonlinear regression analysis [24]. A 4-th degree polynomial is fitted to the data. In this way, the east component can be expressed as:

\[ W_x = a_x + b_x \lambda + c_x \theta + d_x \lambda^2 + \ldots + p_x \theta^3 \lambda + q_x \theta^4, \]

and the north component as:

\[ W_y = a_y + b_y \lambda + c_y \theta + d_y \lambda^2 + \ldots + p_y \theta^3 \lambda + q_y \theta^4, \]

where \( a_x, b_x, \ldots, p_x, q_x \) and \( a_y, b_y, \ldots, p_y, q_y \) correspond to the coefficients of the regressions applied to the east and north components of the forecast tabular data, respectively.

The wind forecast of July the 3th, 2012 is considered. Analytic functions resulting from regression (3)-(4) are valid within a domain covering the North Atlantic and some part of Europe, i.e., \( \lambda \in [-70^\circ, 12^\circ] \) and \( \theta \in [40^\circ, 55^\circ] \). Functions (3-4) can be included in the set of equations (2). The goodness of fit, measured in terms of R-Squared parameter, yielded 0.78 for \( W_x \) and 0.71 for \( W_y \). Figure 2 shows both forecast tabular data (blue dots) and analytic functions (surfaces) for \( W_x \) and \( W_y \) at 200 [Hpa] \((h=11769 \text{ [m]})\).

![Figure 2: Wind values at 200 [HPa] (h=11769 [m]).](image)

4.3 Airspace structure

The airspace is structured to ensure the safe development of aircraft operations. To this end, a network of routes has been established and equipped with navigation aids, so that the aircraft can navigate following them. The routes of this network are referred to as ATS (Air Traffic Services) routes, and they are composed of waypoints and airways. Waypoints may be simple named points in the space or may be associated with existing navigational aids, intersections, or fixes. Airways are imaginary corridors connecting waypoints.

The ATS routes are published in the basic manual for aeronautical information called Aeronautical Information Publication (AIP), which is usually updated once a month coinciding with the Aeronautical Information Regulation and Control (AIRAC) cycle. Ocean tracks might change twice a day to take advantage of any favorable wind. In free flight areas the path is defined by the user, and thus finding the optimal path considering the effects of wind is a crucial issue in these zones.

In the model assumed in this paper, airways are not considered and it is supposed that the aircraft can fly an arbitrary route among waypoints. However considering a complete directed graph structure over the set of waypoint is redundant because aircraft must fly through closer waypoints before reaching farther waypoints. A graph structure which is capable to reflect this simple observation is the multipartite graph structure.

Thus, the airspace structure is modeled as a complete multipartite graph \( G = (V, E) \), whose vertex set \( V \) is partitioned into pairwise disjoint independent subsets which are
called partite sets. In this model nodes represent waypoints and arcs represent possible transitions between them. In a complete multipartite graph vertices are adjacent if and only if they belong to different (adjacent) partite sets. The complete multipartite graph considered is composed of a sequence of $N + 1$ partite sets, $V_0, V_1, \ldots, V_N$, where $V_0$ and $V_N$ contain 1 node each, and the initial and final waypoints $p^I$ and $p^F$, respectively, and $V_i, q = 1, \ldots, N - 1$, contains $n_0$ nodes. Let $\mathcal{P} = \{p^I, p^{1:n_1}, \ldots, p^{N-1:n_N}, p^F\}$ be the collection of waypoints of the partite sets $V_1, \ldots, V_{N-1}$. See Figure 3.

### 4.4 Waypoints selection

The waypoints and navais of the AIRAC cycle published in June 2012 have been considered and a set $\mathcal{P}$ of 8 × 5 waypoints have been selected from them. For those phases entering or exiting oceanic regions, the waypoints have been selected manually coincident with the FIR/UIR bounds. This was the case of the first, second, and third partite sets. On the contrary, for those phases overflying the intra-European area, the waypoints have been selected randomly according to an algorithm that can be consulted in [15, 16]. $\mathcal{P}$ of waypoints is given in Table 1.

**Table 1: Coordinates of the Waypoints**

<table>
<thead>
<tr>
<th>$p^I$</th>
<th>$p_1^I$</th>
<th>$p_2^I$</th>
<th>$p^F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(41.11, 67)</td>
<td>(41.11, 67)</td>
<td>(44.11, -67)</td>
<td>(44.11, -67)</td>
</tr>
<tr>
<td>(41.78, -67)</td>
<td>(41.78, -67)</td>
<td>(42.63, -67)</td>
<td>(42.63, -67)</td>
</tr>
<tr>
<td>(43.56, -67)</td>
<td>(43.56, -67)</td>
<td>(44.93, -67)</td>
<td>(44.93, -67)</td>
</tr>
<tr>
<td>(45.83, -51)</td>
<td>(45.83, -51)</td>
<td>(46.87, -51)</td>
<td>(46.87, -51)</td>
</tr>
<tr>
<td>(47.84, -51)</td>
<td>(47.84, -51)</td>
<td>(48.77, -51)</td>
<td>(48.77, -51)</td>
</tr>
<tr>
<td>(48.82, -8)</td>
<td>(48.82, -8)</td>
<td>(49.5, -8)</td>
<td>(49.5, -8)</td>
</tr>
<tr>
<td>(49.5, -8)</td>
<td>(49.5, -8)</td>
<td>(50.5, -8)</td>
<td>(50.5, -8)</td>
</tr>
<tr>
<td>(45.5, -0.94)</td>
<td>(45.5, -0.94)</td>
<td>(45.61, -0.94)</td>
<td>(45.61, -0.94)</td>
</tr>
<tr>
<td>(45.32, -3.69)</td>
<td>(45.32, -3.69)</td>
<td>(45.08, -3.69)</td>
<td>(45.08, -3.69)</td>
</tr>
<tr>
<td>(45.69, -1.41)</td>
<td>(45.69, -1.41)</td>
<td>(45.03, -0.78)</td>
<td>(45.03, -0.78)</td>
</tr>
<tr>
<td>(46.05, -2.25)</td>
<td>(46.05, -2.25)</td>
<td>(45.73, -1.06)</td>
<td>(45.73, -1.06)</td>
</tr>
<tr>
<td>(45.56, -1.12)</td>
<td>(45.56, -1.12)</td>
<td>(45.78, -1.12)</td>
<td>(45.78, -1.12)</td>
</tr>
<tr>
<td>(45.54, 1.36)</td>
<td>(45.54, 1.36)</td>
<td>(45.33, 1.23)</td>
<td>(45.33, 1.23)</td>
</tr>
<tr>
<td>(44.12, 2.19)</td>
<td>(44.12, 2.19)</td>
<td>(44.05, 2.99)</td>
<td>(44.05, 2.99)</td>
</tr>
<tr>
<td>(44.38, 4.84)</td>
<td>(44.38, 4.84)</td>
<td>(44.37, 5.17)</td>
<td>(44.37, 5.17)</td>
</tr>
<tr>
<td>(45.66, 4.89)</td>
<td>(45.66, 4.89)</td>
<td>(45.10, 5.16)</td>
<td>(45.10, 5.16)</td>
</tr>
<tr>
<td>(46.5, 4.95)</td>
<td>(46.5, 4.95)</td>
<td>(42.89, 8.67)</td>
<td>(42.89, 8.67)</td>
</tr>
<tr>
<td>(44.59, 8.66)</td>
<td>(44.59, 8.66)</td>
<td>(44.04, 8.03)</td>
<td>(44.04, 8.03)</td>
</tr>
<tr>
<td>(45.15, 7.99)</td>
<td>(45.15, 7.99)</td>
<td>(43.45, 7.79)</td>
<td>(43.45, 7.79)</td>
</tr>
</tbody>
</table>

### 4.5 En-Route Overflying Charges

ATS routes go through flight regions, which are portions of the airspace in which a single national aviation authority provides navigation, surveillance and control services. These regions are referred to as FIR/UIR (Flying Information Region/Upper Information Region). Figure 4 shows the FIR/UIR structure of the North Atlantic airspace. In general, national aviation authorities apply overflying fees for the services they provide. Very different charging schemes are applied including purely traveled distance-based charges, aircraft weight and traveled distance charges, flat rate charges ($FR$), or communication rate charges ($CR$) [25].

The charging methodologies in the relevant regions for the flight to be analyzed in the experiment, namely, United States of America (USA), Canada, and Europe, including the North Atlantic oceanic regions, are briefly presented.

In Europe, the standard EUROCONTROL charge formula for en-route services in the EUROCONTROL members countries is:

$$
R_{EU}^\text{En} = UR_{i} \cdot \frac{GCD_i}{100} \cdot \sqrt{\frac{MTOW}{50}}
$$

where $UR_{i}$ is the service unit rate in FIR$_{i}$ (referring to member country $i$), $GCD_i$ is the great circle distance in Kilometers [Km] traveled in FIR$_{i}$, and $MTOW$ is the maximum take-off weight in metric tonnes [t] of the aircraft. The unit rates of en-route charges are established by each EUROCONTROL member state and updated every month [26]. In the USA, the Federal Aviation Administration (FAA) only charges overflight fees to operator that fly in the USA controlled airspace, but neither take off nor land in the USA. In the continental airspace, the en-route charges $r_{USA,OCA}^\text{En}$ are $38.44$ per 100 nautical miles (measured in Great Circle Distance (GCD)). In the oceanic airspace, the fee $r_{USA,OCA}^\text{En}$ is $17.22$ per 100 nautical miles (in GCD)$^1$. Nav Canada applies different fees for its oceanic and continental airspaces. Canadian oceanic charges in Gander Oceanic FIR are based on a flat rate that can be decomposed into navigation fee $FR_{CAN,OCA}^\text{Gander}$ of C$ 93.24$ and a communication fee $CR_{CAN,OCA}^\text{Gander}$ of C$ 22.94$ [27]. Canadian continental airspace charges are based on aircraft weight and traveled distance as follows:

$$
R_{CAN,CON}^\text{En} = UR \cdot GCD \cdot \sqrt{MTOW}
$$

where the Unit Rate (UR) is $8.03445$, the travelled GDC in [Km] and MTOW in [t]. Charges for services provided in the Shannon Oceanic FIR comprise a flat communication rate $CR_{SHO}^\text{En}$ of C$ 45$ (charged by Ireland) and a flat navigation fee $FR_{SHO}^\text{En}$ of C$ 65.70$ (charged by United Kingdom) [28].

---

4.6 Contrails

In this section we follow [29]. Contrails form when a mixture of warm engine exhaust gases and cold ambient air reaches saturation with respect to water, forming liquid drops that quickly freeze. Contrails form in the regions of airspace that have ambient relative humidity with respect to water ($RH_w$) greater than a critical value $r_{constr}$. Regions with $RH_w$ greater or equal than 100% are excluded because clouds are already present [30]. Contrails can persist when the environmental relative humidity with respect to ice ($RH_i$) is greater than 100% [31]. Thus, persistent contrail favorable regions are defined as the regions of airspace that have: $r_{constr} \leq RH_w < 100\%$ and $RH_i \geq 100\%$.

The estimated critical relative humidity for contrail formation at a given temperature $T$ (in degrees Celsius) can be calculated as:

$$r_{constr} = \frac{G(T - T_{constr}) + e_{sat}(T)}{\epsilon_{sat}(T)},$$  (5)

where $e_{sat}(T)$ is the saturation vapor pressure over water at a given temperature. The estimated threshold temperature (in degrees Celsius) for contrail formation at liquid saturation is:

$$T_{constr} = -46.46 + 9.43 \log(G - 0.053) + 0.72 \log^2 (G - 0.053),$$  (6)

where

$$G = \frac{EI_{H_2O}C_p P}{\epsilon Q(1 - \eta)}.$$  (7)

In equation (7), $EI_{H_2O}$ is the emission index of water vapor, $C_p$ is the isobaric heat capacity of air, $P$ is the ambient air pressure, $\epsilon$ is the ratio of molecular masses of water and dry air, $Q$ is the specific heat combustion, and $\eta$ is the average propulsion efficiency of the jet engine.

$RH_i$ is calculated by temperature and relative humidity using the following formula [32]:

$$RH_i = RH_w \frac{6.0612 \exp \frac{102 - T}{290.69}}{6.1162 \exp \frac{77.38}{237.4 + T}}$$  (8)

where $T$ is the temperature in degrees Celsius.

Data of air temperature and relative humidity for June the 30th, 2012 at time 18.00 Z (10.00 a.m. PST) have been retrieved from the NCEP/DOE AMIP-II Reanalysis data provided by the System Research Laboratory at the National Oceanic & Atmospheric Administration (NOAA) [3]. The data have a global spatial coverage with different grid resolutions. Our data have a global longitude-latitude grid resolution of $2.5^\circ \times 2.5^\circ$. Regarding the vertical resolution, the data are provided in 17 pressure levels (hPa): 1000, 925, 850, 700, 600, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10.

We compute the latitude-longitude grid points that are favorable to persistent contrail formation at different barometric altitudes (which defines the pressure). We do so based on gathered data of air temperature and relative humidity, and using equations (5-8) with the following: $EI_{H_2O} = 1.25$; $C_p = 1004 \ [J/KgK]$; $\epsilon = 0.6222$; $Q = 43 \cdot 10^6 \ [J/Kg]$; and $\eta = 0.15$. The longitude-latitude grid points with favorable conditions for persistent contrail formation are represented as red dots in Figure 5 for different barometric altitudes. In order to analyze the regions of persistent contrail formation in our case study, we estimate the values of temperature and relative humidity for the flight levels (converted into barometric altitudes) subject of study by running a linear interpolation.

2Z-hour corresponds to Universal Time Coordinates (UTC). The Pacific Standard Time (PST) is given by UTC - 8 hours.
3The data have been downloaded from NOAA website @ http://www.esrl.noaa.gov/psd/
4.7 Cost factor determination

In this section, we mainly follow [33].

Where there is little doubt about the importance of appropriate estimates for different climatic and cost components in identifying optimal flight strategies, attention generated in this area remains rather limited. Reference [34] assumes infinite cost when a flight flies into a persistent contrail region. This issue is not consistent with airliners economical perspectives, which might be willing to consider environmental cost but not as the only driver. If flight travel time increases as a result of contrail avoidance, not only will extra fuel consumption impose penalties on aircraft operations, but flight crew will work longer hours, also increasing the operating cost to airlines. Moreover, since climate impact from CO$_2$ and contrail formation would clearly be borne by the society rather than air carriers alone, passenger travel time cost should also be included in the overall cost consideration. In [35], the authors argue that the altitude changes rest on the fact that the radiative forcing from CO$_2$ and contrails are of comparable magnitude, which indeed is not completely accurate.

These give rise to two potential issues. First, operational strategies should consider a trade-off cost involving environmental and operational costs. Second, as admitted by [36, 37], using radiative forcing as the measure for climate abatement will be very difficult since radiative forcing includes the impact of all historic flights and does not account for the resident timescales of emissions. Given the distinct physical characteristics of different greenhouse gas agents, global warming potential (GWP) represents a better metric to quantify the true climate impact of different gases. While contrail is not categorized as a gas agent, we follow [33] in attempting to unify the climate impact of CO$_2$ and contrail formation using the GWP concept. We incorporate it together with the cost of fuel consumption into the flight planning design. Notice that we also consider other costs such as air navigation fees, and time-based cost related to crew, passenger or maintenance.

4.7.1 Fuel cost

Fuel consumption is determined based upon the differential equation (24), which provides the fuel burnt as the flight evolves over time. In order to quantify the cost, one simply has to multiply the consumed fuel and the cost of the jet fuel. The average kerosene price in June 2012 was $403.7 cents/gallon. Given that the typical aircraft jet fuel, Jet A, has a density of 0.820 Kg/L, we can assume the fuel cost as:

\[ C_F = 1.30048 \text{$/kg} \]

4.7.2 CO$_2$ emissions cost

The estimation of CO$_2$ cost per second is based on the fixed ratio of jet fuel consumption and CO$_2$ emissions, and estimates of the social cost of carbon. We follow [38] and use $35 as the mean social cost for one ton of CO$_2$ emitted. The per Kg based cost yields:

\[ C_{CO_2} = 0.11 \text{$/kg} \]

4.7.3 Time-based costs

Unit crew cost can be calculated by dividing the total pilot and flight attendant costs by the total block time across one year of operation. Using information from the Bureau of Transportation Statistics Form 41 P-5.2 database, we estimate a cost of $0.1573/sec.

The travel cost per unit time for all passengers onboard is the product of passengers’ value of time (VOT), aircraft seat capacity, and load factor. Following the U.S. Department of Transportation guidance on the economic value of passenger travel time, we use wage rate as a proxy for passenger VOT. The mean wage rate of $21.35/hr in the U.S. in 2010 is used [39]. Assuming 335 seats on a typical A330 and an 80% load factor, the unit travel time cost for all passengers onboard amounts to $1.73/sec.

\[ C_t = 1.73 \text{$/sec} \]

4.7.4 Contrail formation cost

Quantifying the unit cost of contrail formation involves two steps: first estimating the climate impact of formed contrail for one second of flight time and then converting it into monetary cost.

Following the discussion in the introduction to this section, we use GWP to quantify the climate impact of different greenhouse gas agents. For the sake of simplicity, details are omitted. The reader is referred to [53] and [40]. The contrail formation cost for the different time horizons is given in Table 2. Notice that the time horizon has to do with the fact that persistent contrails remain in the atmosphere for a couple of hours, whereas CO$_2$ remains for ever.

Table 2: Contrail cost for different time horizon

<table>
<thead>
<tr>
<th>FL</th>
<th>mi [Kg/min]</th>
<th>$H=20$ Years</th>
<th>$C_{flu}$ [FL]</th>
<th>$$/hr$</th>
<th>$H=100$ Years</th>
<th>$$/hr$</th>
<th>$H=500$ Years</th>
<th>$$/hr$</th>
</tr>
</thead>
<tbody>
<tr>
<td>279</td>
<td>66.7</td>
<td>0.883</td>
<td>0.2337</td>
<td>0.0706</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>290</td>
<td>61.7</td>
<td>0.8976</td>
<td>0.2375</td>
<td>0.0718</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>62.7</td>
<td>0.9121</td>
<td>0.2414</td>
<td>0.0729</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>330</td>
<td>62.2</td>
<td>0.9049</td>
<td>0.2395</td>
<td>0.0724</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>350</td>
<td>60.1</td>
<td>0.8743</td>
<td>0.2314</td>
<td>0.0699</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>370</td>
<td>58.7</td>
<td>0.8539</td>
<td>0.2260</td>
<td>0.0683</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>390</td>
<td>58.1</td>
<td>0.8452</td>
<td>0.2217</td>
<td>0.0676</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>410</td>
<td>58.0</td>
<td>0.8438</td>
<td>0.2233</td>
<td>0.0675</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Therefore, the cost associated to contrail formation will be given time based unitary cost given in Table 2 for the different flight levels multiplied by the time the aircraft overflies through regions of persistent contrail formation at those flight level in the different legs of the flight. Notice that we only consider the $H = 20$ time-horizon. In order to obtain the contrail-based cost in the flight level of our case study, we run a linear interpolation between flight levels.

4.7.5 Overfly charges

The overflying cost for a flight from a USA airport to Europe through Canadian continental airspace, Gander Oceanic, and Shanwick Oceanic FIRs can be expressed as

\[ r_{CanCon} + r_{GanOcc} + r_{Eur} \]

where \( r_i \) refers to the ith relevant European FIR/UIR. The components of this cost have been defined in Section 4.5. Notice that, since the flight departs from JFK airport, the USA do not apply any navigation fee. The unit rates employed in the European regions are those corresponding to European regions adjusted unit rates applicable to April 2012 flights. For continental Canada and oceanic regions the rates are those given before. All rates have been converted to $.

\[^4\text{Santa Maria: } \varepsilon \ 9.79; \text{ United Kingdom France: } \varepsilon \ 83.23; \text{ Spain (continent): } \varepsilon \ 71.84; \text{ France: } \varepsilon \ 64.63; \text{ Italy: } \varepsilon \ 78.69; \text{ Portugal (Lisbon UIR): } \varepsilon \ 33.06.\]
5. RESULTS

We have run three simulations considering different objective functions: Fuel costs and overfly charges [Exp. 1]; Fuel costs, time-based costs, and overfly charges [Exp. 2]; Fuel costs, overfly charges, and contrail-based costs [Exp. 3].

For all three cases, the flight plan is given in Table 3.

Table 3: Flight plan model

<table>
<thead>
<tr>
<th>Phase</th>
<th>Name</th>
<th>Mode</th>
<th>Cost</th>
<th>Time</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>leg 1</td>
<td>Mode H</td>
<td>10000</td>
<td>25000</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>Step Climb 1</td>
<td>Mode V</td>
<td>54000</td>
<td>54000</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>leg 2</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Step Climb 2</td>
<td>Mode V</td>
<td>54000</td>
<td>54000</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>leg 3</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>leg 4</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>leg 5</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>leg 6</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>leg 7</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>leg 8</td>
<td>Mode H</td>
<td>20000</td>
<td>20000</td>
<td>6</td>
</tr>
</tbody>
</table>

The horizontal profile is equal for all three cases. The computed sequence of waypoints gives rise to the following route is: YAHOO, DOVEY, VODOR, RIVAK, PEPET, BNC11, RANTRA, XIRBI, LBN32, AMTEL. The obtained optimal path has been depicted in Figure 6, where the dots represent the computed discrete samples and the triangles the waypoints of the route.

The value of the binary variables gives also rise to the selection of the optimal flight level after performing the different step climbs. The optimal flight level allocation is given in Table 4. The vertical profiles are given in Figure 7. Notice that Exp.1 and Exp. 3 have the same flight level allocation. In Figure 7, the horizontal lines correspond to the different flight levels and the vertical-dashed lines correspond to the switching instants for Exp. 3.

Table 4: Flight level allocation

<table>
<thead>
<tr>
<th>Mode</th>
<th>Exp. 1</th>
<th>Exp. 2</th>
<th>Exp. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>L375</td>
<td>FL375</td>
<td>FL375</td>
<td>FL375</td>
</tr>
<tr>
<td>L350</td>
<td>FL350</td>
<td>FL350</td>
<td>FL350</td>
</tr>
</tbody>
</table>

The switching and final times of the optimal solution are given in Table 5 together with the accumulated consumed fuel at the end of each phase. The optimal evolution of both state and control variables within the time domain is represented in Figure 8, where the dots represent the computed discrete samples and the vertical lines correspond to the switching times.

As expected, Exp.1 presents the least consumption, whereas Exp. 2 presents the fastest flight. It is important to mention that Exp. 2 results are distorted since compressibility effects are not taken into account. In this way, the simulation permits to fly faster than real with less consumption than real. This phenomena could be avoided with a speed dependent drag curve. Exp. 3 present the greatest consumption with the slowest flight. This is also logical, since the aircraft seeks to fly those flight levels in which contrail-based cost are lower. This is why the vertical profile is different.

The computation time was on a Mac OS X 2.56 [GHz] laptop computer with 4 GB RAM was always within one hour: 1656.83 sec for Exp. 1, 2628.26 sec for Exp. 2, 3323.54 sec for Exp. 3.

Finally, a discussion on the quality of the heuristic approach is pertinent. For the sake of space, we omit it. We refer the reader to [15] for a thorough discussion on this end.

6. CONCLUSIONS

In this paper we have effectively studied the 4D trajectory planning problem considering 0-1 binary variables to model decision-making on the optimal flight level and waypoint allocation. Such problem has been tackled using a multi-phase mixed-integer optimal control approach, running simulations with different cost functions that include fuel expenses, environmental-based costs, and operative cost. We have shown the potentiality of the approach as a flight planning tool considering different cost preferences.
Figure 8: State and control variables: Exp. 1 (orange solid-dotted line); Exp. 2 (red solid-dotted line); Exp. 3 (green solid-dotted line)
7. REFERENCES


The Legal Case

Giovanni Sartor,* Giuseppe Contissa,*
Hanna Schebesta,* Mijle Laukyte*
*European University Institute, Florence, Italy
giovanni.sartor@eui.eu,
giuseppe.contissa@eui.eu,
hanna.schebesta@eui.eu,
migle.laukyte@eui.eu

Paola Lanzì*, Patrizia Marti*,†,
Paola Tomasello*
†University of Siena
DeepBlue, Rome, Italy
paola.lanzi@dblue.it, patrizia.marti@dblue.it,
paola.tomasello@dblue.it

ABSTRACT
This paper presents the first release of the Legal Case, recently developed by the ALIAS Project and still under refinement. The Legal Case is a methodological tool intended to address liability issues of automated ATM systems: it provides for a legal risk management process that can be applied either proactively or retroactively. Used in a proactive way, the Legal Case aims to address the liability issues that may emerge in the design of new technologies. Used in a retroactive way, it is meant to assess liability issues pertaining to an existing piece of technology that has already reached the deployment stage. The Legal Case is mainly designed to be used by a Legal Analyst who is a member of an interdisciplinary project team dealing with the development of new automated technologies (in the case of the proactive approach) or with the accident investigation (in the case of the retroactive approach). Although not yet finalized, the Legal Case methodology is gathering great interest from the ATM community.

Author Keywords:
legal risk; ATM; automation; case-based approach; argument-evidence-based approach.

ACM Classification Keywords
K.5.m. Legal Aspects of Computing: Miscellaneous

General Terms
Human Factors; Design; Legal Aspects

INTRODUCTION:
WHAT IS THE LEGAL CASE?
The Legal Case is a methodological tool intended to support the integration of any technology in complex systems, particularly in ATM. Its purpose is to address the liability issues emerging from the interaction between the human and the ATM automated tools, ensuring that these issues are clearly identified and taken into consideration at the right stage of the design, development and deployment process.

The Legal Case is currently being developed by the ALIAS Project (Addressing the Liability Impact of Automated Systems), that is co-financed by EUROCONTROL on behalf of the SESAR Joint Undertaking as part of Work Package E. The project focuses on the legal implications of automation, exploring the relation between automation and liability in complex socio-technical systems as fundamental issue in human-technology interaction.

The Legal Case has been called in this way in analogy with the other Cases currently available, which are the Safety Case and the Human Factors Case. In fact, the notion of Case recalls the case-based approach, which has been proposed in the framework of the European Operational Concept Validation Methodology (EOCVM) [6] to provide key stakeholders with evidence constructed from targeted information. The Legal Case is in line with the case-based approach, first of all, because it enables grouping diverse information into a clear structure, in order to describe the potential and the risks of the concept under evaluation. Secondly, the Legal Case shares the idea that generally concepts can be more easily and less costly modified at an early stage of their development process; while at a late stage of development, it will likely be more complicated, more expensive or even impossible to fix them. This idea featured in the ALIAS approach since its very beginning and stimulated as main philosophical background the legal research conducted so far [13]. In line with this perspective, it is not a coincidence that the Legal Case was initially conceived as a proactive process, while the other possible applications have been identified during the further progress of its design. The term Legal specifies the kind of content addressed by the methodology, which deals with the liability issues potentially raised by the interaction between the human operator and the automated systems. This aspect represents a highly innovative development as no methodological tool is currently available to address the liability impact of new technologies during the design phase of project’s lifecycle. In fact, initially, EOCVM proposed the development of the following five Cases to be applied in R&D: 1) the Safety Case; 2) the Environment Case; 3) the Human Factors Case; 4) the Standards and Regulatory Case; 5) the Business Case. However, only the Safety Case and the Human Factors Case have been developed so far and are currently available. Of the others, the Environment Case and the Business Case will be developed within the framework of the SESAR Programme, while no attempt has been carried out so far to develop the Standards and Regulatory Case. By addressing certain
key legal aspects, the Legal Case developed by the ALIAS Project can ideally be a way to bridge the gap between such cases and the Standards and Regulatory Case, the designing of which has not yet started. Although coherent with the other Cases, the Legal Case is complementary to them, as it may potentially be integrated with the other Cases into the Business Case. In fact, even if the Legal Case methodology has been designed to be applied at project level, horizontal use of its results is also feasible. The horizontal application of the Legal Case would consist in combining the results of different instances of its application, which focus on technologies with some common elements. The process of combining the outcomes has the advantage of highlighting the similarities and differences of results achieved by applying the Legal Case to different technologies. This would contribute to the identification of common patterns or issues of liability attribution that otherwise may remain unnoticed or exclusively associated to a certain technology or disaster. In fact, the horizontal application of the Legal Case would highlight that different automated technologies may raise similar issues of liability attribution, which may in turn need to be addressed in a systemic way and require refinement of the current legal framework.

In addition to this, in the framework of SESAR, the results of the Legal Case can be combined with those of the other Cases being developed, which are following analogous processes inspired by the Generic Transversal Areas Assessment Process. In this way the Legal Case may contribute to enlarge the scope of the Business Case, thus not excluding that “Legal Implications” could start to be considered as a Transversal Area.

This paper is intended to present the first release of the Legal Case methodology, which is still under refinement. Within the ALIAS project the development of Legal Case started with the study of the legal framework, focusing on liability in air law. However, the focus of this paper is to present the workflow of the process, rather than the results of the legal analysis.

The paper is articulated in five main parts, as follows:

• The first part describes the scope of the Legal Case: first of all, it explains why the increasing of automation expected within SESAR is in a certain way “imposing” the need to address legal issues of automated systems in a pragmatic and systematic way; secondly, it describes how the Legal Case is supposed to provide an answer to this demand and which are the expected benefits of applying the legal case;

• The second part aims to describe the generic process of the Legal Case, presenting the steps of this process and introducing the main ways in which the Legal Case can be used, i.e. its proactive and the retroactive application;

• The third and the fourth part provide a detailed description respectively of the proactive and of the retroactive application of the Legal Case, outlining for each of them the scope, the purpose and the features of the process;

• The fifth and conclusive part explores the forthcoming activities in support of the refinement of the Legal Case methodology. As the Legal Case is a very innovative methodology, thus requiring in a certain way a cultural change in the ATM community, these will be mainly based on close cooperation with and feedback collection from the stakeholders. For the scope of the Legal Case methodology, key stakeholders include ANSPs, Regulatory Agencies and the major industries of the civil aviation (systems’ designers).

1. AUTOMATION AND RESPONSIBILITY IN THE FUTURE SCENARIO: WHY WOULD A LEGAL CASE BE NEEDED?

Although not yet finalised, the Legal Case methodology is gathering great interest from the ATM community. Industrial suppliers, ANSPs, research centres and authorities are unanimous in recognizing the need to address the liability impact of automated systems as early as possible during the project’s lifecycle, as they all agree that the more the innovation will progress, the more the theme of liability attribution will be crucial. The feedback collected so far including the discussions currently active on the ALIAS network [1] confirm that legal issues investigation is one of the main subjects needed to be addressed when facing with the integration of highly automated systems in current operations.

In fact, the introduction of higher levels of automation expected with SESAR 1616] will bring about a change in the way the machine supports human performance, thus potentially implying a new allocation of the decision making tasks between the human and the automation, in the sense that tasks previously carried out by the human (for example the provision of separation) are supposed to be partially delegated to the system. In particular, highly automated systems are expected to take over operators’ repetitive tasks, while the human role is expected to be focused on strategic planning, intervening on exceptions and monitoring the system’s behaviour [7][10]. In general, rather than governing flight operations directly, pilots and controllers will supervise the automated systems doing the job. This will raise significant legal questions, thus requiring a critical revision of the allocation of tasks, roles and responsibilities in the context of complex socio-technical systems.

---

1 The legal framework addressed is composed of international conventions, such as the 1944 Convention on International Civil Aviation (the Chicago Convention), The 1929 Warsaw Convention, superseded by the 1999 Montreal Convention, The 1952 Convention on Damage Caused by Foreign Aircraft to Third Parties on the Surface (the Rome Convention); EU Regulations such as (EC) No 2027/97 amended by Regulation (EC) No 889/2002; legislative packages developing a European policy (Single European Sky) in ATM; case law such as Überlingen 2002; Linate 2001; Cagliari 2004; Flight AF447 over Mid Atlantic 2009, etc.
Does the introduction of higher level of automation imply any change in the responsibility of the actors involved? And, at the end, who is going to be legally liable for damage resulting from human, technical or organisational failures of an automated socio-technical system? In particular, who is to be held liable if an automated system makes or induces wrong choices? Our previous research [3] has shown that when automation is increasingly used (in aviation as in other fields), then defective technologies or the defective deployment of technologies, while hopefully reducing human error, will possibly become the main cause of injuries to people and things. More importantly, in the future scenario, where ATM systems will rely on embedded, automated and deeply interwoven technologies, liability may be distributed among the stakeholders in a way which is different from the current one. Thus liability for damages will be increasingly transferred to the organizations creating the technologies, defining the contexts for their deployment, using, or maintaining them. In this perspective liability should be considered in a systemic way, i.e., as being inseparable from the design and implementation of concepts into a socio-technical system. Furthermore the allocation of liability (within the boundaries of the applicable public law), will need to be defined through a legal framework that is accepted by the involved parties, and that provides a reference both when designing the system and when investigating its subsequent failures.

The Legal Case is intended to meet the needs coming from the stakeholders, in the sense that it is meant to supply the ATM community with a standard process to identify and manage liability risks associated to automated technologies in a structured, coherent and pragmatic way. The main purpose of the tool is to help preventing the possibility that liability issues would act as showstoppers for the development and deployment of new technologies, as often happens in current industrial processes. Moreover, it may bring significant benefits to the European Communications, Navigation Surveillance / Air Traffic Management (CNS/ATM) community in the long term, changing the dominant culture and approach to legal liabilities.

2. THE LEGAL CASE PROCESS

The Legal Case methodology outlines a generic legal risk management process to systematically identify and address liability issues of automated ATM systems. The process is compliant with the Generic Transversal Areas Assessment Process, which the Transversal Areas of SESAR\(^2\) are required to follow in conducting assessments and building cases.\(^3\)

The generic process of the Legal Case consists of four steps (Figure 1):

1. **Understand the context.** This step requires the collection of a set of background information about the object of the study (which may be an operational concept, a system, a service, or an accident in which a piece of technology played a crucial role);

2. **Identify liability issues.** This step defines the legal implications of the object of the study on the basis of the understanding of its socio-technical aspects.

3. **Perform the analysis.** This step analyses the stakeholders’ acceptability of the legal implications defined in previous step, proposes ways to deal with all involved legal risks, and proposes possible mitigations and recommendations for the design.\(^4\)

4. **Provide results and recommendations.** This step presents the results of the study, highlighting the liability issues associated with the object of the study, the ways to deal with legal risks and further recommendations.

Figure 1 The Legal Case generic process

The Legal Case risk management process is combined with software based tools (in particular, tools for argumentation mapping) [11] to support legal risk management in ATM scenarios. The substantive focus on automation provides the angle from which the Legal Case incorporates existing modeling techniques of human-machine interaction and connects them to the legal perspective.

The Legal Case is mainly designed to be used by a Legal Analyst, namely a person having a legal background in aviation and liability law, which enables him/her to understand the legal issues involved in a project or accident. We assume the Legal Analyst to be a member of an interdisciplinary project team; this would enable the Analyst to appeal to others’ skills available within the project when additional technical knowledge is required. In fact, legal knowledge is necessary to deal with the liability topics while engineering knowledge and human factors are essential combination of the various cases into the business case at programme level. Its adoption in the Legal Case intends to highlight the transversal nature of liability assessment and the need to consider this topic in a systemic and structured way across the innovation process, in the same ways as safety, security, human performance, environment and cost-benefit.


\(^3\) The Generic Transversal Areas Assessment Process has the purpose to ensure that a specific topic (i.e. safety, security, environment, human performance, cost-benefit) is addressed in a coherent way across the programme, thus guaranteeing the quality, the comparability and the compatibility of the results produced at project level and favouring the final

\(^4\) In the proactive application of the Legal Case, the main stakeholder is the developer (manufacturer) of the new technology, while in the retroactive application is based on the acceptability of the all stakeholders.
for the understanding of the technical and operational features of the object of the analysis, i.e., the automated system under examination. In this respect, the Legal Case can be conceived as key communication tool between experts belonging to different communities. The generic process of the Legal Case can be used with a twofold purpose: either in a proactive way, to systematically identify, address and manage the liability issues of automated systems during their design, or in a retroactive way, to support the identification of liabilities associated with automated tools that are already in operation. So, the Legal Case offers two viewpoints on legal risk management: the proactive and the retroactive viewpoints. In a proactive viewpoint, risk management is meant to prevent or mitigate legal risk, that is, it is anticipatory. Risk management from a retroactive viewpoint provides a strategic response to legal risks that have already taken place (or may take place in the future) and provides a structure for their containment.

The process of the Legal Case includes aspects that are common to proactive and retroactive applications, although differences are envisaged in the specific tasks carried out in the two cases. A significant difference between proactive and retroactive processes emerges in Step 1, as the collection of background information carried out at this stage has a different focus: the legal issues of the new ATM system being developed (in the proactive process), and the possible legal issues arising in the future from potential accidents or malfunctions associated to a technology already in operation, as well as the legal impacts of specific accidents that have taken place (in the retroactive process). The same difference characterizes also Step 2 and 3. In those steps it mainly reflects the different techniques used for the identification of legal issues and for the subsequent analysis.

3. PROACTIVE APPLICATION OF THE LEGAL CASE

Used in a proactive way, the Legal Case has the purpose of contributing to the design of automated ATM systems that are associated to schemas of liability attribution. This is achieved by ensuring that aspects such as the legal framework which they are based on, the ways in which liabilities are attributed and distributed among the stakeholders, and their possible effects in terms of stakeholders’ acceptability are properly taken into account during the design process.

There are many cases of technologies (within and outside the ATM domain) that, although technically mature and reliable, did not go in operation due just to users’ or all the stakeholders’ resistances related to liability issues. An interesting case concerns the Automated Highway System (AHS), which has been described in our previous research. This is a concept referring to a set of designated lanes on a limited access roadway for unmanned vehicles controlled by an automated system. This concept implies a shift of driving and control functions from the driver to the automated car with the aim to provide safer and more convenient travel. Notwithstanding the promising trial carried out in 1994, the US Department of Transportation withdrew financial support to the project. The decision was motivated by concerns about the acceptability of this automated technology and, in particular, about the acceptability of its legal implications.

In the European ATM domain the concept of Free Flight underwent a similar story. The discussion about this new operational concept started in the United States in the early 90s, and quickly extended to Europe in the following years. The basic idea of Free Flight is to shift from active air traffic control of aircraft to passive control with intervention in case of exceptions. Aircraft flying en route no longer have to use existing predefined airways, but are free to fly a preferred route. Problems, such as bad weather or loss of separation with other traffic along the preferred route, have to be resolved by the flight crews, normally without intervention from Air Traffic Controllers. According to Eurocontrol [5], in Free Flight the responsibility for separation assurance from other aircraft remains within the aircraft in almost all circumstances, although ground-based ATM can still undertake some responsibility (e.g. in emergencies). Several studies on this operational concept [15] have been undertaken, concluding that, although the pilot is free to select the trajectory and responsible for separation assurance, Air Traffic Control (ATC) remains the ruling entity in separation arbitration. Free Flight thus consolidates the principle of the final authority of the pilot in command while maintaining ATC’s responsibility for separation arbitration. The Free Flight has never fully convinced the European stakeholders for a number of different reasons, among which the schema of liability attribution between air and ground, and thus it has not been applied in Europe (yet).

The purpose of a proactive application of the Legal Case is to prevent such kind of scenarios. Discovering at the end of the design process that a new technology implies a distribution of liability that is not feasible in the current legal framework and/or not acceptable for the stakeholders, may have enormous systemic costs, induce sub-optimal mitigations, and risk to make the whole system inefficient.

Expected benefits of the proactive application of the Legal Case

The proactive application of the Legal Case is expected to produce legal insights that could be used with a twofold purpose: first of all to feed into the technology design process and contribute to the design of acceptable systems, secondly to highlight gaps of the current legal framework in dealing with highly automated systems. Anticipating the identification of this kind of gaps during the design process offers the opportunity to address the issue when the system is still being designed and developed, allowing to investigate the problem and potentially address it before the new system is ready for the deployment.

The Legal Case methodology looks at liability allocation as one of the inherent properties of the ATM
systems. This approach moves from the consideration that—as well as other inherent properties of the ATM systems, such as safety and human performance—legal liabilities are likely to affect the stakeholders’ acceptability and to constrain the application of a new technology. As a consequence they should be taken into account during the design process of new operational concepts in order to early address issues that later on may reveal themselves as show-stoppers for the success of the technology. In other words, together with the need to design for safety and human performance, we assume that there is a need to “design according to liabilities.” The idea is that addressing liability issues earlier in the development process will make it easier, less costly and controversial to address legal issues at a later stage, when the system is deployed. In addition, the earlier in the development process we identify the liability issues, the more uniform the approach to liability attribution could be across projects developing new automated technologies. This is expected to provide the advantage of having new technologies that in this respect are compliant and easy to combine.

The Figure 2 shows the benefits of a proactive application of the Legal Case. The x-axis represents the timeline of the project lifecycle, including all the E-OCVM phases from research and development to deployment [6]; the ordinate symbolizes the costs, both in terms of time and in terms of effort and resources.

Figure 2 Benefits of the proactive application of the Legal Case

Scope of the proactive application of the Legal Case: the notion of “Design according to liabilities”

The “design according to liabilities” is the innovative approach that we propose to the design of automated systems. The innovative aspect here regards the application of the user-centred design process to new topics, i.e., the legal aspects, that traditionally were not addressed in such methods. The notion of “design according to liabilities” enlarges the scope of the operational concept design and aims to ensure the legal feasibility and acceptability of the new automated tools. In this regard, it is also expected to avoid that liability issues come out at the end of the whole design process as show-stoppers that impair the actual implementation and deployment of the technology. Finally, it aims to speed up the deployment of the technology and its adoption in the operational environment.

In analogy with the user-centred design approach, the design according to liabilities tries to optimize the technology around how stakeholders can, want or need to benefit from it, rather than forcing them to change their behaviour to accommodate the product. The figure 3 shows how the proactive application of the Legal Case methodology can interactively feed the operational concept evaluation and validation process.

Figure 3 The Legal Case in the design process

The figure recalls the generic user-centered design process (represented by the spiral) [2] that we propose to apply to the liability topics. It highlights that the user-centered design can be characterized as a multi-stage problem solving process that not only requires designers to analyze and foresee how users are likely to use a product (design & development), but also to test the validity of their assumptions with regards to user behaviour in real world tests with actual users (evaluation & validation). The Legal Case is intended to apply such an iterative process to liability issues, in the sense that the legal evaluation and validation phase feeds with insights and recommendations the design and development phase of the concerned concept; afterwards, another design and development phase may implement these recommendations and potentially lay the basis for a further evaluation and validation phase, and so on iteratively until needed. In analogy with the user-centered design methodology, the needs, wants and limitations of the stakeholders are given extensive attention at each stage. Without these tests, it would be very difficult for the designers of a product to understand intuitively what liability impact of the concerned tool may look like.

Moreover, this means that the Legal Case process can be flexibly applied at any stage of the project lifecycle, from the initial prototyping up to intermediate and high maturity levels, and potentially repeated as the maturity level increases, the unique differences being the kind of analysis to be performed and the output obtained at each step.

This is what we call “design according to liabilities”.

The proactive process

The figure 4 shows the process of proactive application of the Legal Case. The representation of the process is circular in order to highlight its iterative nature. The application of the proactive process in fact is not a one shot operation. It is expected to be systematically and periodically applied during the design process in order to check, at different levels of concept maturity, that the ATM system being developed does not raise major issues from the legal point of view.
In its proactive version, legal risk management deals with hypothetical risks and the identification of possible risk sources. After detection, the risks become assessable by attributing rough estimations of probabilities to them. These legal risks can relate to compliance by monitoring the pertinent body of law, such as international, European and national legislation, case law and what can broadly be identified as ‘soft law’ such as non-binding guidelines and recommendations. However, contractual monitoring (clauses in contracts of sale, insurance) make up part of a preventive approach as well. Where a legal risk is identified, it is presented within the broader project team with a view to develop a joint risk treatment strategy.

In particular, the four steps of the process are:

1. Understand the ATM concept;
2. Identify liability issues;
3. Perform the legal analysis (improve and validate the liability allocation);
4. Collect findings and produce results.

They are featured as follows.

Step 1 of the proactive application of the Legal Case has the threefold purpose to i) collect background information about the ATM concept being designed, ii) classify the level of automation of the associated system, and iii) identify the possible failures regarding the new operational concept. The classification of the level of automation of the concerned ATM concept (point ii) is performed on the basis of the Level Of Automation Taxonomy (LOAT) developed by SESAR 16.05.01 [14]. The LOAT is a tool for addressing the kind of support that the automated system provides to the human performance: it helps to identify the division of tasks between the human and the machine. The taxonomy will alert the Legal Analyst to significant changes in the allocation of tasks between the human and the automation, thus inducing him/her to examine their legal significance.

Step 2 concerns the assessment of liability risks on the basis of existing laws and contractual arrangements. The assessment is carried out with the help of two kinds of maps, namely legal risks maps and legal analysis maps.

A legal risk map is a support tool for highlighting the liability risks associated to the possible failures identified in the previous step. It links a particular factual constellation (in particular a kind of failure) to a possible legal liability (a classification map). The purpose of risk-liability maps is to suggest kinds of legal liabilities to be investigated for each possible failure identified in Step 1. The legal risks maps are classification maps; the main kinds of failures (first level of the mapping structure) are connected to the possible legal liabilities (second level of the mapping structure) resulting from them. The following map (figure 5) shows a list of potential technical latent conditions, and related liabilities emerging from them.

Figure 4 Proactive application of the Legal Case

Figure 5 Legal risks map

For instance, technical latent conditions, which could lead to an accident involving the Traffic Collision Avoidance System (TCAS), could be those regarding TCAS processor’s insufficient capacity to compute advisories updates. This could engender product, organisational or managerial liabilities. Similarly, in the System Wide Information Management (SWIM) system, damage could result from prolonged technical shortcomings, deriving from the fact that the system had been upgraded with new functions without considering side effects and without ensuring compatibility with pre-existing functions. In this case organisational and managerial liability may be at issue,

---

5 In the future version of the Legal Case we consider to expand the legal analysis to cover the accidents produced by unknown causes.

6 In the current version of the Legal Case, we address four main types of possible failures "Organizational Latent Condition", "Technical Latent Condition", "Technical Active Error", and "Human Active Error", and their sub-types. In the next release of the Legal Case the categorization of the failures will be refined, and new kinds of failures (and related liabilities) will be added, such as regulatory failures or failures performed by the individuals outside the socio-technical system in question such as passengers or persons on the ground and their contributory negligence leading to the accident.

7 Classification maps are meant to provide taxonomies of the objects within a certain domain. They consist of boxes linked simply by lines flowing either from top to bottom or from left to right. The boxes can be expanded and collapsed at different levels, since the classification maps can be multi-level.
together with (software) product liability and with various contractual liabilities. Technical latent conditions could also threaten the functioning of Remotely Piloted Aircraft Systems (RPAS) in case in which the software—calculating avoidance maneuvers—was malfunctioning, because it wasn’t adequately tested. Here organization, managerial and product liability may be at issue, with regard to user, maintainer and the developer.

After having built the legal risk map, the Legal Analyst needs to examine the possibility that a legal risk concerning a particular actor occurs in different contingencies. To do this, s/he can rely on the legal analysis maps (supported by the relevant legal and empirical knowledge). A legal analysis map links a possible legal liability to the preconditions of its existence (following a map based on legal arguments)\(^8\). Legal analysis maps reflect our understanding of the law on liability as it is represented in the current legal framework concerning air law, product liability, insurance and contract law\(^9\). The answer which the Legal Analyst looks for through the map is whether there is the risk of a particular kind of liability, and this will be established by checking whether the conditions for that kind of liability may exist under some possible circumstances. The following map (figure 6) represents product liability. It shows that the manufacturer is liable on the basis of the product liability doctrine when their technology falls within the definition of a product, the technology is defective and the defect is the reason why the accident occurred.

![Legal analysis map for product liability](image)

**Figure 6 Legal analysis map for product liability**

The two analyses have to be performed in sequence: first, we need to identify whether there is the risk of a possible liability and, secondly, under what conditions this liability can be incurred. Thus we suggest the following sub-process:  
1. Preliminarily identify the liability risks on the basis of legal risk maps (maps connecting kind of failures in different socio-technical context to kinds of liabilities);  
2. Examine the possibility that a legal risk occurs\(^10\) and concerns a particular actor, in different contingencies on the basis of liability maps (supported by the relevant legal and empirical knowledge). The expected outcome of this sub-process is to determine the approximate probability that a liability is incurred by a certain actor (on the basis of a very rough scale, for example: very unlikely, unlikely, likely, very likely), and to annotate the corresponding probability on the liability risk map.  

The **Step 3** consists in engaging in legal design on the basis of the results of the legal analysis performed in the previous step. By “legal design” we mean proposing possible mitigations and recommendations for the system’s design. Such mitigations and recommendations are targeted towards optimal acceptability of the liability risks for all stakeholders.

This involves complementing the outcomes of the legal responsibilities analysis with private (contractual) legal regulations meant to ensure an allocation of liabilities which is acceptable to the parties. Three fundamental liability-design measures can be decided upon at this stage:

- Liability mitigating measures;
- Liability enhancing measures;
- Liability displacing measures.

The first (liability mitigating measures) are meant to reduce liability between the parties through contracts. This happens in particular through liability exclusion clauses, according to which the parties agree that one of them will not be bound to compensate damages caused to the other. For instance, in case the parties decide that the contractual liability of the software producer for failures of software is too burdensome, the software purchaser may agree on a liability limitation clause, according to which there would be no compensation for the damage to the purchaser resulting from software defects.

On the contrary, a liability enhancing measure could provide that damage should be compensated although according to the public law there should be no compensation in a certain case (for instance, the parties may agree on a strict liability of a manufacturer for damages caused by the malfunctioning of the equipment, covering also cases where the manufacturer could invoke the state of the art exception).

Liability displacing measures would involve instead the commitment of one party to cover damages

---

\(^8\) Argumentation maps are visual representations of the structure of arguments. Arguments are represented as diagrams with boxes corresponding to claims (propositions put forward in the argumentation) and arrows corresponding to relationships between such claims. Premise-claims can support or attack conclusion-claims. Premises supporting a conclusion are in green, while those attacking a conclusion are in red.

\(^9\) As to the cases for which no specific regulation is available, we made reference to the general norms of liability and general principles of the international law. However, the Legal Case methodology doesn’t aim to provide complete legal maps covering the different national legislations, but rather to use the maps as guidelines that the Legal Analyst should follow to solve the concrete issue at stake.

\(^10\) The ‘legal risk management’ considers legal risk as one of the components of risk management. It addresses legal risk as a distinct type of risk, which is an isolatable part of the overall risks faced by a stakeholder. By a legal risk we understand the probability of an unwanted legal outcome, being triggered by uncertain factual circumstances and/or uncertain future legal decisions. The valuation of legal risks, obviously, depends on the viewpoint of the stakeholder (namely, on whether s/he will have to suffer the consequences of such unwanted outcomes).
resulting from a liability of another party. For instance, the purchaser could commit to cover the damages that the producer should pay in case the software was faulty, or to provide insurance to that effect. The objective of Step 4 is to produce the results of the legal analysis and to submit them to all the interested stakeholders for their approval and re-use in the design process.

4. RETROACTIVE APPLICATION OF THE LEGAL CASE

Used in a retroactive way, the Legal Case intends to assess liability issues pertaining to an existing piece of technology that has already reached the deployment stage. The retroactive application can address:

- possible legal issues arising in the future from potential accidents or malfunctions, or
- the legal impacts of specific accidents that have taken place.

When the Legal Case is retroactively applied to address possible legal issues arising in the future, the process of its application will largely correspond to the proactive process described above. The main differences will be the following:

- rather than being performed by the design team, the analysis will be entrusted to an assessment team;
- the input information will consist, in particular, of data concerning the current deployment of the technology, as obtained by documentation (such as handbook of the technology, procedures for its usage in a certain organisation, incident reports, near misses and past similar accidents), but also by on-field analysis of the ways in which the technologies are used by the operators;
- the focus of the results will be on changes to the current legal arrangements (in particular, the modification of contracts and the adoption of insurance policies) and organisational arrangements, rather than on suggestions for changes to the technology.

Conversely, when the Legal Case is applied to an accident in which existing technologies played a crucial role, the process of its application may have some specificities, as shown in the figure 7.

![Figure 7 The retroactive application of the Legal Case](image)

The retroactive application in case of accident (like the proactive application described above) follows these 4 steps:

1. **Understand the Accident.** This step requires the collection of background information about the disaster and any relevant systems and/or services involved in the accident dynamic. In particular, background information to be gathered will regard: the operational context in which the disaster took place, the events of the accident, the main attributed causes of the accident, the actors and the technology involved in the dynamic of the disaster. This step includes the identification of the level of automation of the technologies in question, and it concludes with the identification of the failures that led to the accident.

2. **Identify liability issues.** This step identifies the implications on liability attribution associated with the technology involved in the accident.

3. **Perform the analysis (address liabilities).** This step performs a detailed analysis of the aspects of liability attribution previously identified.

4. **Provide results and recommendations.** This step presents the results of the study carried out in the previous steps, highlighting the aspects of liability attribution associated with the ATM concept or system involved in the accident.

**The retroactive application of the Legal Case as support to accident investigation**

We already anticipated that in an increasingly automated environment, such as the one of aviation, the liability for any damage caused by technological tools needs to be reconsidered: the more automated the system becomes, the more organizations and individuals are involved in building, testing, and developing the system. This is why the failures caused by highly automated tools will enlarge the scope of liability distribution extending it among the developers involved in building it. This would mean that experts will be called to establish what went wrong in the tool, and who was responsible for that particular part of it which went wrong: the developer of hardware, the developer of software, the maintenance service provider, the software engineer who had the task to ensure the frictionless integration of different parts of the tool, etc. The liability may be completely or partially transferred to the enterprise using the technology or to the technology developers (programmer, manufacturer...), who created it. Introducing highly automated systems into operational environments brings about a new dimension of liability attribution among the stakeholders, where, in case of an accident, according to the socio-technical perspective, all the stakeholders (including the developers of the system) have to be included into the investigation process.

The research conducted in the framework of the ALIAS project has shown that the socio-technical perspective has been often taken into account in the accident investigation but not in any ensuing litigation and that this has shaped the attribution of liability. In line with this perspective, it is interesting to point out that the International Air Transport Association (IATA) has recently expressed its concerns with respect to an increasing trend toward the criminalization of accidents [8], mainly consisting in the fact that judges and courts are likely to attempt to override accident investigators through the acquisition
of safety information reported prior to the accident as evidence in court, and often having as a result the prosecution of a frontline individual. Such a "criminalization" may be either subsequent to or concurrent with the safety investigation itself [12]. The main issue at stake here is that the nature of these parallel investigations is often of potentially conflicting agendas.

The case of the dramatic runway incursion that occurred in Linate in 2001 is one of the most representative examples. In fact, besides human errors, crucial latent conditions and organisational failures contributed to the occurrence of the accident on the ground. However, the Italian legal system failed to take into account the importance of institutional-organisational aspects in allocating criminal liabilities and put the greatest part of responsibility on the individual closest to the accident (the controller).

Another example is the criminal case that followed the Air France Concorde disaster in Paris on July of 2000 and in which 113 people lost their lives. Ten years after the accident, a single mechanic from Continental Airlines was found guilty of manslaughter because a small titanium strip fell off a Continental Airlines aircraft on the runway before Concorde took off. He received a suspended sentence. Charges against Air France and the designers of Concorde were dropped, and Paris Airport was not held to account over its failure to perform a runway inspection. The mentioned Court decisions show how a complex and fragmented legal framework may make it difficult to distinguish tasks, roles and related responsibilities. According to the results presented by the IATA, the countries with strong civil law cultures are particularly susceptible to this movement and they are cited as countries in which it is more likely that judges and courts will attempt to override accident investigators through the acquisition of safety information reported prior to the accident as evidence in court. Such a trend may have a twofold impact:

- on safety culture and reporting system: the use of data contained within voluntary occurrence reports to prosecute or to attribute blame may represent a sizeable barrier to the generation and submission of reports in the first place. Evidently, an increasing trend of this nature has the potential to significantly impact one of the core foundations upon which the safety culture is built: the Just Culture (open reporting systems being the key to the realization of the Just Culture) [4]. To grant free access to this data would be to erode the foundations of a Just Culture, ultimately serving to obstruct the influx of safety-related information and, as a result, instead pose what is perhaps the greatest threat to aviation safety [8]. So, even if recent years, especially 2011, were the safest in aviation history [9], there are still challenges to be faced in order to improve the compliance of the justice processes with the safety processes;
- on the investigation of the causes of the accidents and consequently on building appropriate lessons learnt in order to prevent similar events to occur in the future: it is evident that the prosecution of frontline employees will not prevent similar accidents to occur in the future because it does not address the underlying safety issue that acts as a latent condition contributing to the fatal event.

The purpose of a retroactive application of the Legal Case to accidents is to prevent such kind of situations from happening. In this respect, the Legal Case is intended to serve as support tool for analysing accidents from a socio-technical perspective and to allocate liability accordingly; plus, it is intended to represent a standardized methodology of analysis that could prevent the fragmentation that often characterizes legal investigations. The final purpose of the process is to feed the current legal approach to accident investigations and to the legal trial with a systematic methodology that may represent a reference for improving legal analysis of accidents taking place in complex socio-technical systems.

As additional potential resulting benefit, the retroactive application of the Legal Case may provide an input back for the redesign of the concerned technology.

5. CONCLUSIONS
This paper presented the first release of the Legal Case methodology recently developed by the ALIAS Project. Further steps of the Project will concern the refinement of the tool. This will profit of the feedback coming from the stakeholders. In this regard, the ALIAS Project is going to propose an iterative process of validation and redesign of the Legal Case, in which a User Group empanelled in the project will be involved. Panelists will be stakeholders interested in applying the Legal Case and will establish a stable and active cooperation with the project team, which will allow us to collect feedback from users and improve the Legal Case methodology. In fact, although the Legal Case is compliant with the case-based approach, thus relying on a consolidated methodology in the ATM community, special attention is needed for ways to facilitate its use, due to the innovative topic it deals with, i.e. the legal aspects, that are traditionally not addressed in the framework of systems’ design and development. This is the reason why the ALIAS Project is intended to pay extensive attention to the feedback coming from the stakeholders.

References
1. ALIAS Project, the ALIAS network, www.aliasnetwork.eu
5. EUROCONTROL http://www.eurocontrol.int/eeq/public/standard_page/proj_MFF.html
6. EUROCONTROL, European Operational Concept Validation Methodology (E-OCVM), edition 2, March 2007


8. IATA, Safety, The Blame Game


16. SESAR JOINT Undertaking, The SESAR Target Concept.

This work is co-financed by EUROCONTROL acting on behalf of the SESAR Joint Undertaking (the SJU) and the EUROPEAN UNION as part of Work Package E in the SESAR Programme. Opinions expressed in this work reflect the authors' views only and EUROCONTROL and/or the SJU shall not be considered liable for them or for any use that may be made of the information contained herein.
Transforming Time
Towards an intuitive time constraint depiction

Jendrick Westphal, Uwe Klingauf
Flight Systems and Automatic Control
TU Darmstadt
Darmstadt, Germany
Westphal-, Klingauf@fsr.tu-darmstadt.de

Nima Barraci, Jens Schiefele
Market and Solutions Research
Jeppesen GmbH
Neu-Isenburg, Germany
Nima.Barraci-, Jens.Schiefele@jeppesen.com

ABSTRACT
Future ATM environments will require the pilot to ensure adherence to 4D trajectories. This paper proposes an intuitive time constraint depiction to optimize the information needed to be perceived and evaluated by the pilot. The developed system can be applied to any 4D trajectory concept. For this time constraints are transformed from the temporal to a lateral domain. After defining the concept, a possible integration into an electronic charting application is described.

Author Keywords
4D Trajectory; Air Traffic Management; Time Constraints; SESAR; NextGen

ACM Classification Keywords
J.2 PHYSICAL SCIENCES AND ENGINEERING - Aerospace; H.4.2 INFORMATION SYSTEMS APPLICATIONS: Types of System - Decision Support; H1.2 MODELS AND PRINCIPLES: User/Machine Systems - Human Factors

INTRODUCING TIME
The current Air Traffic Management (ATM) System is reaching its capacity limit [10]. To allow higher density air traffic and at the same time more efficient operations of each individual flight are the challenges to be solved in the future [5].

Addressing this problem, the SESAR [2] and NextGen [9] research projects are well under way. A set of new tools will be introduced to the ATM domain shortly. One of these tools is the usage of 4D Trajectories, that in contrast to current operations allow not only the precise coordination of a flight in position and altitude but also in time.

4D Trajectories are only of use when shared among the stakeholders of the ATM system. For this the trajectories have to be negotiated and monitored [4]. The monitoring of time adherence to a trajectory is an unusual task for the flight crew. Currently the flight crew only monitors its lateral deviation, for example to a given Required Navigation Performance (RNP) procedure [1]. The performance is monitored by achieving a Total System Error (TSE) lower than the RNP value for the given procedure [7]. The vertical Trajectory adherence is monitored by ensuring the adherence to vertical constraints on Trajectory Change Points (TCPs) which are usually defined waypoints along the Route. The constraints can have different modes (at, at or above, at or below, between) and are discrete for the TCP they are defined at.

A similar handling will be proposed for the handling of time constraints. However time is not a variable directly controlled by the pilot, instead it is the result of the aircraft velocity over a distance and controlled by the pilot through speed modifications. Thus the introduction of time is not adding an additional dimension to be controlled but rather limiting the existing dimensions to positions over time in order to achieve the time constraint. This rather complex interaction between the dimensions demands for an abstract representation with which the pilot can monitor the adherence to constraints in all dimensions.

This Paper is proposing a way towards an intuitive time constraint depiction along the trajectory. For this, first a common understanding of a trajectory definition has to be achieved. In this paper three trajectory definitions are differentiated with a focus on one. The mathematics needed to transform the time constraints from the time to a lateral domain are defined. Last integration into an electronic charting application is demonstrated and future research explained.

DEFINING TIME
An aircraft trajectory and thus the times along this trajectory can be defined in many ways. To define a common understanding of time and time constraints, three trajectory definitions are differentiated. A Single Time Constraint is set to the trajectory; this is the case for example when trying to reach a fix in the terminal area at a required Time of Arrival (RTA). Multiple Time Constraints are put on a trajectory to allow strategic planning, increase enroute as well as airport capacity and improve the

1 The FTE is influenced by the flight crew through the Flight Technical Error (FTE)
planning of the overall ATM system. A temporal RNP is put on an aircraft's trajectory when high precision temporal navigation is needed such as in the final approach to an airport.

These three concepts are described in respect to their temporal flexibility between discrete constraint TCPs.

**Single Time Constraint**

Figure 1 Single Time Constraint

A single time constraint or RTA is the simplest 4D Trajectory integration possible. In many modern Flight Management Systems (FMS) the pilot can define a RTA for a waypoint along the Route. The Flight Guidance of the FMS will adjust the speed to fulfill this constraint as close as possible [3].

Figure 1 depicts the Δt over a distance s an aircraft flies between TCP₀ and TCPₙ in relation to the reference trajectory. The reference time of a point s on the trajectory can be calculated by eq. 1.

\[ t(s) = t_{TCP_n} - (d_{TCP_n} - s) \cdot \frac{\delta \Delta t}{\delta s} \]  

(1)

Starting at the last constrained waypoint, \( \frac{\delta \Delta t}{\delta s} \) describing the change of time to the nominal time over the distance s.

Assuming that:

\[ \frac{\delta \Delta t}{\delta s} = \text{const.} \]  

(2)

results in:

\[ \frac{1}{\Delta \tau} = V_G \]  

(3)

for constant wind and air speeds along the trajectory. This simplification is made for a first integration described in this paper. The slopes of the graphs shown in Figure 1 can be calculated by the minimum and maximum \( \frac{\delta \Delta t}{\delta s} \) see eq. 4.

\[ \frac{1}{(\frac{\delta \Delta t}{\delta s})_{min}} = V_{G,min} \]

\[ \frac{1}{(\frac{\delta \Delta t}{\delta s})_{max}} = V_{G,max} \]  

(4)

The minimum and maximum values can either be the current minimum and maximum selectable ground speeds, global minima and maxima considering altitude changes and wind predictions, or limitations set by the airlines in form of minimum and maximum Cost Indices for an efficient flight.

**Multiple Time Constraints**

Trajectories with multiple constraints are handled similar to trajectories with only one constraint, but need to be deconflicted. Time constraints also do not have to be fixed to one exact time but can rather have one of the following four states (compare [11], similar [6]):

- At the TCP has to be overflown exactly at the specified time
- At or above the TCP has to be overflown at the time or later
- At or below the TCP has to be overflown at the time or earlier
- Between defines two times between which a TCP should be overflown

The deconfliction of multiple time constraints means that it has to be ensured that one time constraint is either as restricting or more restrictive than a following constraint. For this a forward and backward propagation of the trajectory with the minimum and maximum achievable groundspeeds \( V_{G,min} \) and \( V_{G,max} \), the distance s between the constraint TCP and the limiting times for each constraint TCP has to be performed.

Figure 2 Multiple Time Constraints

Figure 2 shows two 'between' constraints at TCPₙ and TCPₘ and the achievable times at TCP₀. As a result of the forward propagation it can be seen that \( \Delta t_{TCP(n,min)} \) is more restrictive than \( \Delta t_{TCP(m,min)} \) and thus further restricting the achievable time window at TCPₙ. For the maximum limit it can be seen from the backward propagation that \( \Delta t_{TCP(m,max)} \) is more restrictive thus the achievable time window at TCPₙ has to be reduced. \( \Delta t_{TCP(m,max)} \) is also defining the achievable maximum time \( \Delta t_{TCP(0,max)} \) at TCP₀.
Temporal RNP

Figure 3 Time RNP

The concept of a temporal RNP is the most advanced of all. Figure 3 shows that the minimum and maximum time deviations are constant for the applicability of one temporal RNP value. When the temporal RNP value changes at a TCP, the minimum and maximum time deviations are set to the new constant temporal RNP value (in Figure 3 from T-RNP1 to T-RNP2 at TCP). As this concept requires a continuously defined trajectory, a transformation from the temporal to the lateral domain is straightforward and not further described in this research.

TRANSFORMING TIME

After having established a working basis of trajectory descriptions with Time constraints, efforts for an intuitive depiction of these time constraints can be undertaken. To transform the time constraint from the time into a lateral domain, the minimum and maximum times to the next most constraining waypoint have to be identified. The most constraining waypoint can differ for a trajectory description with multiple time constraints. TCP\textsubscript{n} is the next most constraining time constraint for the lower limit, and TCP\textsubscript{m} for the upper limit. For one time constraint the most constraining TCP is always identical for upper and lower limit \((n = m)\), for multiple time constraints \(n\) and \(m\) can but don't have to be identical.

The available time to reach a waypoint can be calculated as the difference between the minimum or maximum time at the constraining waypoint and the present time.

\[
\begin{align*}
\Delta t_{\text{min}} &= t_{TCP_{n,\text{min}}} - t_{\text{present}} \\
\Delta t_{\text{max}} &= t_{TCP_{m,\text{max}}} - t_{\text{present}}
\end{align*}
\]

Solving Eq. 1 for the distance and differentiating between the minimum and maximum distance from the constraint TCP with consideration of the times in Eq. 5 results in Eq. 6.

\[
\begin{align*}
s_{\text{max}} &= d_{TCP_{n}} - \frac{\Delta t_{\text{min}}}{\frac{d_{TCP}}{d_{s}}} \frac{\Delta t_{\text{min}}}{\frac{d_{TCP}}{d_{s}}} \\
s_{\text{min}} &= d_{TCP_{n}} - \frac{\Delta t_{\text{max}}}{\frac{d_{TCP}}{d_{s}}} \frac{\Delta t_{\text{max}}}{\frac{d_{TCP}}{d_{s}}}
\end{align*}
\]

where \(d_{TCP_{n}}\) and \(d_{TCP_{m}}\) are the distances to the constraint TCP\textsubscript{n} and TCP\textsubscript{m} from TCP\textsubscript{0}. \(\frac{d_{TCP}}{d_{s}}\)\textsubscript{min} and \(\frac{d_{TCP}}{d_{s}}\)\textsubscript{max} are calculated for a first integration as:

\[
\begin{align*}
\frac{1}{\frac{d_{TCP}}{d_{s}}} &= T A S_{\text{min}} + V_{\text{wind}} \\
\frac{1}{\frac{d_{TCP}}{d_{s}}} &= T A S_{\text{max}} + V_{\text{wind}}
\end{align*}
\]

where \(T A S_{\text{min}}\) and \(T A S_{\text{max}}\) are the minimum and maximum selectable true airspeeds of the aircraft in current configuration and \(V_{\text{wind}}\) is the currently acting wind. This rather simplistic approach results in:

\[
\begin{align*}
s_{\text{max}} &= d_{TCP_{n}} - \Delta t_{\text{min}} \cdot (T A S_{\text{min}} + V_{\text{wind}}) \\
s_{\text{min}} &= d_{TCP_{n}} - \Delta t_{\text{max}} \cdot (T A S_{\text{max}} + V_{\text{wind}})
\end{align*}
\]

More strategic integrations are thinkable and are currently being developed. However, this rather tactical integration has already yielded good results.

The focus of this research however, is the trajectory concept with multiple time constraints.

DEPICTING TIME

From the calculations described above, a depiction has been developed and implemented into Gate to Gate, Jeppesen's next Generation EFB charting application. Gate to Gate is a data driven, seamless and integrated paper chart replacement. Functionalities include graphical NOTAM and Weather depiction with inflight updates, as well as connectivity to the aircraft FMS and the airlines ground systems [12].

In this framework the Precision Aircraft Control enhancing Route (PACeR), an implementation of the algorithm described above has been integrated. To allow a modification of existing time constraints, that were received from a 4D capable FMS or ground systems, as well as adding new constraints, a constraint editing system (CES) has been also integrated as is shown in Figure 4.
The CES allows the user to set or manipulate time constraints at any waypoint along the route. First the mode of the constraint has to be set. The constraint can be fulfilled either 'at', 'at or above', 'at or below' or between two defined times. To modify a constraint the hour, minute or second of a time can be selected and manipulated with 'up' and 'down' arrows. The second can be altered in 5 second steps, to limit the number of inputs required and still ensure a precise Trajectory definition.

The altered trajectory with the modified time constraint now has to be negotiated either directly with Air Traffic Control (ATC) or with the Airline Operations Center (AOC). As the Gate to Gate application has data link access capabilities all this can be done directly in the EFB application. Once agreed the new Trajectory can be made directly available to the FMS, either by a direct connection from the EFB or through Controller Pilot Data Link Communication (CPDLC) or an Aircraft Communications Addressing and Reporting System (ACARS) message.

As soon as either a time constraint has been received or set in the CES of Gate to Gate, the PACeR depiction becomes active. This is shown in Figure 5 as magenta line along the Route. Figure 5 depicts a flight from EDDV (Hannover, Germany) back to Hannover using an RNP approach. A time constraint was added when passing 'OSN' to coordinate with other traffic arriving to EDDV and allowing a Continuous Descent Approach along the RNP procedure. The Magenta line along the green Route depicts an 'area to be' in which the Aircraft, depicted by the Ownship symbol as white Arrow, would have to stay to adhere to the time constraint set at 'OSN'.

As this 'area to be' which is the result of the calculation of $s_{\text{min}}$ and $s_{\text{max}}$ is highly dynamic, Figure 6 shows how the PACeR depiction evolves over time. In Figure 6 the aircraft flies towards the waypoint 'OSN' which is restricted by a time constraint to be crossed between "14:14:57" and "14:15:03". Advancing a little in time Figure 7 shows how the extent of the PACeR 'area to be' decreases the closer the current time is approaching the constraint time of 'OSN'. After having flown over 'OSN' the next time constraint at 'WPTIF' is the most constraining on the trajectory. The aircraft is required to pass 'WPTIF' between "14:27:57" and "14:28:03" as is shown in Figure 8. As can be seen in all three figures, the aircraft stays well within the PACeR and thus well within the agreed trajectory at all times.
TIME AHEAD

To verify and validate the PACeR depiction integrated into the Gate to Gate application in an operational environment, simulator and flight trials were conducted. As part of the German Heterogeneous complex Air Traffic (HETEREX) project, Jeppesen and TU Darmstadt have conducted flight trials in cooperation with the Deutsches Zentrum für Luft- und Raumfahrt (DLR) in their Airbus A320 Advanced Technology Research Aircraft (ATRA). During these trials the Gate to Gate EFB application was connected to the DLR 4D capable FMS [8]. 4D Trajectories could be transferred from the EFB to the FMS, where the guidance from the FMS ensured adherence to these constraints.

Future developments will focus on a more strategic implementation of the transformation algorithm. Currently the algorithm is very tactical taking the currently available minimum and maximum speeds and the current wind into consideration. The assumption that \( \frac{V_{\text{Min}}}{V_{\text{Max}}} = \text{const.} \) which was made for the work in this paper is only valid for a constant wind and a constant minimum and maximum air speed along the trajectory.

To verify the impressions collected during the HETEREX trials with objective and subjective measures, on a larger group of participants, simulator trials with the enhanced system are planned for late 2013.

REFERENCES

A Decomposition-based Optimal Control Approach for Aircraft Conflict Avoidance Performed by Velocity Regulation

Loïc Cellier
ÉNAC-Univ. of Toulouse
F-31055 Toulouse, France
loic.cellier@enac.fr

Sonia Cafieri
ÉNAC-Univ. of Toulouse
F-31055 Toulouse, France
sonia.cafieri@enac.fr

Frédéric Messine
ÉNSÉEIHT-Univ. of Toulouse
F-31071 Toulouse, France
frederic.messine@n7.fr

ABSTRACT
One of the decisive tasks within the air traffic management is the resolution of aircraft conflict avoidance problems. To avoid conflict, aircraft have to preserve a minimal safety distance between them. In this paper, we present optimal control models and approaches based on speed regulation to perform aircraft conflict avoidance. We consider some aircraft configurations with separable trajectories, i.e., such that trajectories of aircraft pairs exhibit conflict zones which are each other separated in terms of time and/or space. We propose a decomposition of the problem in such a way to solve independently subproblems of the original one.

Keywords
Collision avoidance; air traffic management; velocity regulation; optimal control; decomposition-based approach.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
To prevent the risk of collision, the main purpose of Air Traffic Control (ATC) is to ensure a minimum distance of separation between each pair of aircraft. This norm corresponds to 1000ft vertically and 5NM horizontally (with the units: 1NM (nautical mile) = 1, 852m; 1ft (feet) = 0.3048m). In this context, a pair of aircraft is said in conflict if this norm is not respected.

Different approaches for aircraft Conflict Detection and Resolution (CD&R) have been proposed (see, e.g., Kuchar and Yang [6] for a survey of existing approaches). Many research works focus on aircraft trajectory deviations, with altitude and/or heading changes. Recently, the European En-Route Air Traffic Soft Management Ultimate System (ERASMUS) project (Bonini et al. [1]) has pointed out the interest in aircraft separation maneuvers based on velocity changes. Thus, new models and solution approaches based on small velocity variations have been developed (see, e.g., Pallottino et al. [7], Cafieri [2]).

We consider optimal control to perform aircraft conflict avoidance based on speed regulation (for an introduction to the optimal control theory, see e.g., Trélat [9]). Starting from our approach combining direct and indirect optimal control methods (Cellier et al. [3]), we propose a decomposition of the problem which exploits the aircraft trajectory topology. This allows us to obtain a more computationally affordable approach for CD&R.

We recall optimal control for CD&R based on speed regulation in the next section. We then present the decomposition-based approach. A few remarks and statements of future work conclude the paper.

OPTIMAL CONTROL APPROACH
MODEL WITH ACCELERATION AS COMMAND
We formalize aircraft conflict resolution using optimal control. Taking into account energy criteria, we minimize a quadratic cost penalizing the speed variations, on the set of all the n aircraft (i), during the whole time horizon (from t0 to tf). A dynamic system depending on time (t), allows us from the command – the acceleration variable (ui) – to deduce the velocity (vi) then, using the trajectory direction (di), the position (xi) for each aircraft. Moreover, for operational reasons, the acceleration and velocity are bounded. We consider initial conditions and (free or fixed) terminal conditions on velocities and/or positions. The main constraint is the separation which guarantees, for each pair of aircraft, the fulfillment of the requested norm (Di). We start from the following optimal control model, using small speed variations only, and keeping the aircraft trajectories unchanged, to get the aircraft separation.

\[
\begin{align*}
\min_u & \sum_{i=1}^{n} \int_{t_0}^{t_f} u_i^2(t) dt \\
\text{s.t.} & \quad \forall i \in [t_0, t_f], \forall i \in I \\
& \quad \forall i \in [t_0, t_f], \forall i \in I \\
& \quad \forall i \in [t_0, t_f], \forall i \in I \\
& \quad \forall i \in [t_0, t_f], \forall i \in I \\
& \quad \forall i \in [t_0, t_f], \forall i \in I \\
& \quad \forall i \in [t_0, t_f], \forall i < j
\end{align*}
\]
Resolution Strategy

Optimal Control problems can be numerically solved by using two kinds of methods. First, one can use direct methods. They are generally based on time-discretization and numerical integrators to replace the differential equations. They transform the initial optimal control problem into a large scale NonLinear (continuous) optimization Problem (NLP). A second class of methods is the one of indirect methods, which make it possible to obtain an analytical solution via the Pontryagin Maximum Principle. Nevertheless, for this last one category of numerical optimal control methods, theoretical difficulties exist to manage constraints on the state variables (for more details, see, e.g., Trélat [9]).

In our model, we have to deal with numerous constraints on the state variables. We have constraints on the state variables of first order (velocities), i.e., the velocity bounds, and constraints on the state variables of second order (positions), i.e., the separation conditions. In this context, a direct method appears more easy to implement. It can be applied by performing a time-discretization and replacing the differential equations (representing the system dynamic) by numerical integrators of Euler type. This leads to the resolution of a NLP. In this NLP, the number of variables and constraints largely increases with respect to the number of aircraft involved (n) and the number of considered time-subdivisions (p). More precisely, the computational complexity is given by $O(np)$ variables (resp. $O(n^2p)$ constraints). In order to reduce the number of variables and constraints, in Cellier et al. [3], we proposed to combine direct and indirect numerical optimal control methods, as briefly recalled below.

We recall the definitions of zone and postzone as follows. Let zone be the region where for an aircraft pair separation constraints have to be verified and postzone be the following region where all the conflicts have been solved and when the aircraft are already separated.

From the spatial point of view, let $x_{\text{enter}}^{i}$ be the first (by chronological order) trajectory point of the aircraft $i$ for which the distance between this point and another point of the trajectory of aircraft $j$ corresponds to the separation norm. Reciprocally, by projections, we can denote $x_{\text{exit}}^{i}$ the last (by chronological order) trajectory point of aircraft $i$ for which the distance between this point and another point of the trajectory of aircraft $j$ corresponds to the separation norm. Similarly, for aircraft $j$, we can define the trajectory points $x_{\text{enter}}^{j}$ (resp. $x_{\text{exit}}^{j}$) to enter (resp. to exit) to the zone.

The same decomposition can be formulated from the point of view of time. Let $t_{1,\text{min}}^{i}$ be the minimum instant time for aircraft $i$ (using upper velocity bound) to reach its first trajectory point $x_{\text{enter}}^{i}$ (with $j$ another aircraft). Let $t_{1,\text{max}}^{i}$ be the maximum instant time for aircraft $i$ (using lower velocity bound) to reach its last trajectory point $x_{\text{exit}}^{i}$ (with $j$ another aircraft).

Let $t_1$ be the zone entry time:

$$ t_1 := \min_{i \in \{1,\ldots,n\}} \ t_{1,\text{min}}^{i}, \quad \text{s.t.} \ i_{1} := \arg\min_{i \in \{1,\ldots,n\}} \ t_{1,\text{min}}^{i}, $$

Let $t_2$ be the zone exit time:

$$ t_2 := \max_{i \in \{1,\ldots,n\} \setminus \{t_{2,\text{max}}^{i} \}} \ t_{2,\text{max}}^{i}, \quad \text{s.t.} \ i_{2} := \arg\max_{i \in \{1,\ldots,n\}} \ t_{2,\text{max}}^{i}. $$

The zone (from $t_1$ to $t_2$ ) corresponds to an unique period for all the $n$ aircraft. We limit ourselves to satisfy the separation constraint within the zone to guarantee the separation overall.

From $t_0$ to $t_2$, we apply a direct method. From $t_2$ to $t_1$, we do not need to impose the original state variable constraints (velocity bounds and separation constraints). We apply an indirect optimal control method, which gives an analytical solution via the Pontryagin Maximum Principle (Pontryagin et al. [8]).

DECOMPOSING THE PROBLEM EXPLOITING AIRCRAFT TRAJECTORY TOPOLOGY

We develop the optimal control approach described in the previous section, proposing, for certain aircraft trajectory configurations, a problem-decomposition into small subproblems which can be solved independently.

As a basic assumption, we consider aircraft flying along linear trajectories at the same altitude-level, and focus on tactical phases (i.e., short-term potential conflict flight phases). For the present study, we consider in particular air traffic configurations with separable trajectories, i.e., such that trajectories of aircraft pairs exhibit conflict zones which are each other separated in terms of time and/or space. In practice, aircraft defining a conflict zone do not have any interaction with aircraft defining a different zone.

In this way, we can exploit the topology of the aircraft trajectory configurations to decompose the problem and solve it independently on subproblems, considering zone and postzone, and correspondingly direct and indirect numerical optimal control methods, on each of such subproblems.

We decompose the initial set of aircraft into subsets of aircraft as follows. For a pair of aircraft, we can define a potential concourse if a conflict may appear between them as a consequence of a speed change. We consider aircraft subsets such that if an aircraft has a potential concourse with another, the two aircraft are belonging to the same subset. We refer to such a subset as a cluster.

For each cluster of aircraft (c), we can define its own time range $[t_{c}^{s}, t_{c}^{e}]$ corresponding to its zone (defined in the previous section for a set of aircraft). Assuming all the zones separated from each other, we can solve independently all the aircraft subproblems using their respective zone only. This reduces significantly the computational effort to solve the whole problem.
Computational experiments are carried out using the AMPL [5] environment and the interior point method solver IpOpt [10] for large-scale nonlinear problems. We consider air traffic configuration involving aircraft flying along their straight paths and potential conflicts. The horizontally separation norm is 5NM. Acceleration are bounded, based on Eurocontrol’s Base of Aircraft Data (BADA) [4], namely $u_i^{ub} = -u_i^{lb} = 4000\text{NM/h}^2$. Velocities are bounded, based on the ERASMUS project, by a small speed range: $[v_i^{t0} - 6\% v_i^{t0}, v_i^{t0} + 3\% v_i^{t0}]$ (where $v_i^{t0}$ is the initial velocity of aircraft $i$). Terminal conditions are returning to the initial velocities ($v_i^{t0} = 447\text{NM/h}$) at final time ($t_f = 1\text{h}$).

![Figure 1. Trajectory configuration involving 30 aircraft and 15 conflicts.](image)

We give an example (see Figure 1) validating our approach. We consider an aircraft configuration involving 30 aircraft and 15 conflicts. The number of pairs of different aircraft corresponds to 435. Applying the previously described strategy based on the zone and the postzone, we are not able to solve the problem. This is due to the large number of variables and constraints in the NLP problem corresponding to the zone. Applying the proposed decomposition approach, based on aircraft clusters of potential concourses, we are able to obtain an efficient solution. The number of constraints is reduced by more than 96%, due to the fact that only 15 pairs of aircraft have to be considered to satisfy the separation conditions. We obtain a (local) optimal solution in 232 seconds (on a laptop with 2.53GHz and 4Go RAM), with all conflicts solved (all aircraft separated).

**CONCLUSION**

We presented an approach for aircraft conflict avoidance based on optimal control, with an acceleration command, where a decomposition of the problem in such a way to solve independently subproblems of the original one is proposed. Future work will address a further development of this approach to identify subproblems of aircraft involved in more general configurations.

**ACKNOWLEDGMENTS**

The authors gratefully acknowledge financial support under grant ANR 12-JS02-009-01 "ATOMIC".

**REFERENCES**

Aircraft Trajectory Simulator Using a Three Degrees of Freedom Aircraft Point Mass Model

Javier García-Heras Carretero †, Francisco Javier Sáez Nieto †, Ricardo Román Cordón ‡
Air Navigation Research Group
Polytechnic University of Madrid
Madrid, Spain
† javier.garciaheras@upm.es, † franciscojavier.saez@upm.es, ‡ ricardo.roman@upm.es

ABSTRACT
Aircraft Operators Companies (AOCs) are always willing to keep the cost of a flight as low as possible. These costs could be modelled using a function of the fuel consumption, time of flight and fixed cost (over flight cost, maintenance, etc.). These are strongly dependent on the atmospheric conditions, the presence of winds and the aircraft performance. For this reason, much research effort is being put in the development of numerical and graphical techniques for defining the optimal trajectory.

This paper presents a different approach to accommodate AOCs preferences, adding value to their activities, through the development of a tool, called aircraft trajectory simulator. This tool is able to simulate the actual flight of an aircraft with the constraints imposed. The simulator is based on a point mass model of the aircraft.

The aim of this paper is to evaluate 3DoF aircraft model errors with BADA data through real data from Flight Data Recorder FDR. Therefore, to validate the proposed simulation tool a comparative analysis of the state variables vector is made between an actual flight and the same flight using the simulator. Finally, an example of a cruise phase is presented, where a conventional levelled flight is compared with a continuous climb flight. The comparison results show the potential benefits of following user-preferred routes for commercial flights.

Author Keywords
Trajectory Based Operation; Optimal Control; Business Trajectory; Point Mass Model; Trajectory Optimization; Aircraft Model Validation

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
Over the past few years, the common practice within ATM has been that commercial aircraft must fly by following a set of predefined routes to reach their destinations. Currently, aircraft operators (AOCs) are requesting for more flexibility to fly according to their preferences, in order to help them achieve their business objectives. AOCs generally wish to keep the cost of a flight as low as possible. These costs depend mainly on: the amount of fuel needed; the actual time of flight and also the over flight charges.

The present paper shows a different approach to accommodate AOCs preferences and add value to their activities, through the development of a tool/algorithm that calculates optimal trajectories.

To accomplish this objective, an aircraft trajectory simulator (TS) based on 3DoF is presented. This model considers the aircraft as a point mass, as well as the aircraft performances and physical constraints, and also atmospheric information. Many works on aircraft trajectory modelling or trajectory optimization have used a 3DoF aircraft model, as an example [1, 2, 3, 4, 5].

In order to accept the model as an aircraft TS, an adequate validation is needed. According to [6] a validation process involves: the evaluation of the software used, the airframe and the pilot. This paper is focused on the software and aircraft model validation. To evaluate the accuracy of the model, it could be developed from an experimental flight test to an aircraft observation during several parameters are modified.

For this reason, the uncertainties of the model should be delimited. The data used to perform the validation of the model is obtained from actual flights, and it allows checking the differences between the real variables (from real Flight Data Recorder FDR) and the ones defined by the aircraft TS. A similar method has been used by [7], where an Unmanned Aerial Vehicles (UAV) model based on a 6DoF has been validated through the comparison at the same inputs of the real aircraft response and the software model one. Also, in [8] a flight test program was designed to validate a 6DoF flight model.

Once the validation of the aircraft TS has been accomplished an example is presented. In this example we compare two different trajectories: one representing a cruise phase of a real flight following a levelled flight in three steps and the second one performing a continuous climb.
The movement of an aircraft could be expressed by a 6 Degrees of Freedom (6DoF) or a 3DoF [9]. The first model is the most complete model due to the fact that it has into consideration both rotational and translational motion.

Commercial aircraft trajectories involve small aircraft rotation axes and also the angle of sideslip could be considered negligible because of the turn coordinator system that is installed into almost the whole commercial aircraft. The resultant of that approximation is the 3DoF model.

Therefore, this paper proposes an aircraft TS based on a Point Mass Model (PMM) due to it is a simple one and the errors of that model are affordable, as it will be shown in the follow part. Equation (1) shows in the three first elements the kinematic equations, the following three the dynamic equations and the final element is the fuel consumption equation.

\[
\begin{bmatrix}
X' \\
Y' \\
H' \\
\frac{d}{dt} V' \\
\frac{d}{dt} \psi' \\
\frac{d}{dt} \gamma' \\
\frac{d}{dt} W'
\end{bmatrix} =
\begin{bmatrix}
V \cos(\psi) \cos(\gamma) + \omega_x, \\
V \sin(\psi) \cos(\gamma) + \omega_y, \\
V \sin(\gamma) + \omega_z, \\
\frac{g}{W} \left[ (T \cos(\theta - \gamma) - D) - W \sin(\gamma) \right], \\
\frac{g}{W V'} \left[ L \sin(\phi) - D \right], \\
\frac{g}{W V'} \left[ L + T \sin(\theta - \gamma) \right] \cos(\phi) - W \cos(\gamma), \\
-C T
\end{bmatrix}
\]

Equation (1) could be simplified because in commercial aircraft, the roll angle, the flight path angle and the flight path angle derivative are small. Also, it must be observed that the thrust has been considered as a vector pointing in the longitudinal aircraft axe direction. With all these considerations, the equation (1) could be rewritten as it is shown in (2).

\[\text{(2)}\]

Where:

- \(C\) and \(T\), are the specific fuel consumption and the engine thrust respectively, determined using BADA 3.9 information [10]
- \(W\) is the weight
- \(\gamma\) is the flight path angle
- \(g\) is the gravity of value 9.81 m/s\(^2\)
- \(V\) is the velocity of the airplane relative to the air
- \(\alpha\) is the angle of attack
- \(\beta\) is the angle of sideslip
- \(\theta, \phi, \psi\) are the airplane pitch, roll and heading angles respectively
- \(X, Y, h\) are the component of the position vector along XYZ
- \(\omega_x, \omega_y, \omega_z\) are the component of the wind vector along XYZ
- \(L\) is the aircraft lift force, which is considered equal to \(W/\cos(\phi)\)
- \(D\) is the aircraft drag force, which is evaluated in each phase of flight with drag polar coefficients from BADA 3.9 [10]

Figure 1: Aircraft Trajectory Simulator Scheme.
For the design and implementation of the aircraft TS, MATLAB® SIMULINK® [11] software environment was used.

The aircraft TS has four main sections: Flight Management System (FMS), Flight Control System (FCS), aircraft model in a quasi-stationary flight and a geographic variables converter, as is depicted in figure 1.

The FMS manages the reference variables (Velocity, altitude and roll angle) which inputs of the FCS. For this purpose information from the mission (route 2D/3D or 4D) or the aircraft guidance law are needed.

The FCS is based on three control loops to regulate: velocity deviation, altitude deviation and lateral deviation from the defined mission. These control loops will act on Throttle Lever Position (TLP), altitude derivative and roll angle, respectively, which are the inputs in the aircraft model.

In the Aircraft model in a quasi-stationary flight section the kinematics, dynamics and the fuel consumption equation are implemented (see equation 2). It is also taken into account the quasi-stationary hypothesis where lift is considered equal to the weight component. That section needs, besides the inputs, the aircraft performances, power plant information and aircraft initial condition.

$$\begin{bmatrix} X \\ Y \\ h \\ V \\ \psi \\ W \end{bmatrix} = \begin{bmatrix} V \cdot \cos(\psi) + \omega_x \\ V \cdot \sin(\psi) + \omega_y \\ V \cdot \gamma + \omega_z \\ g \cdot \frac{T - D - W \cdot \gamma}{W} \\ \frac{gL \cdot \sin(\phi)}{W} \\ -C \cdot T \end{bmatrix}$$

Finally, the coordinates should be converted from Local Level System (LLS) to Latitude, Longitude, and Height (LLH) coordinates with respect to the same system of coordinate. This will allow in further works the analysis of a group of trajectories and their possible interactions within the airspace. For this porpoise, the eccentricity and the mayor axis of the ellipsoid WGS-84 are needed.

**VALIDATION AIRCRAFT TRAJECTORY SIMULATOR**

The validation process is based on the comparison between real data from Flight Data Recorder (FDR) and variables obtained from the aircraft TS. For this purpose, the FDR route is introduced into the Mission section of the aircraft TS (see Figure 1). The information of the FDR selected for this validation example corresponds to a Iberia flight IB6826 from São Paulo – Guarulhos (23°25′55″S 46°28′10″W) to Madrid - Barajas (40°29′36.80″N 3°34′0.035″W) flown on the 14th of September 2009 (ETOT 19:00 local time). This flight lasted 9 hours, 58 minutes and 21 seconds (i.e. 35901 seconds) and was flown by an Airbus A340-600.

In Figure 2 the trajectory obtained by the aircraft TS (blue line) is compared with the actual trajectory that was flown (red line-FRD) with respect to global coordinates (longitude and latitude). The vertical trajectories of the TS and the actual flight are also compared with respect to time in Figure 2. These comparisons are made for the three phases in which this flight was performed.

The deviation between both trajectories is presented numerically in Table 1, showing these results, the trajectory defined by the aircraft TS can be considered accurate enough to represent a real flight, except for the weight variable witch have shown great deviations. This could be due to the winds uncertainties that have not been considered.

<table>
<thead>
<tr>
<th></th>
<th>Take-off</th>
<th>En-Route</th>
<th>Landing</th>
</tr>
</thead>
<tbody>
<tr>
<td>μ (ft)</td>
<td>0.0072</td>
<td>36.634</td>
<td>5.49e-5</td>
</tr>
<tr>
<td>V (kts)</td>
<td>-2.779</td>
<td>73.384</td>
<td>-0.325</td>
</tr>
<tr>
<td>φ (deg)</td>
<td>-0.008</td>
<td>106.67</td>
<td>-0.849</td>
</tr>
<tr>
<td>γ (deg)</td>
<td>0.1985</td>
<td>0.7414</td>
<td>0.0007</td>
</tr>
<tr>
<td>W (kg)</td>
<td>1205.6</td>
<td>31760</td>
<td>1150.8</td>
</tr>
</tbody>
</table>

Table 1: Mean (μ) and Variance (σ) differences for every phase of flight

**PRACTICAL OPTIMIZATION EXAMPLE**

Once the aircraft TS has been validated as an adequate tool for obtaining the aircraft variables along the defined route, a practical example could be presented. In this section the Aircraft TS is used for comparing a flight cruise phase based on three steps (which have been used in the real flight presented in the validation process, see figure 2) and a flight based on a continuous cruise phase. The results obtained...
with this example allow measuring the quantity of fuel that is saved with the latter using the same time of simulation and a constant True Air Speed (TAS) of 480kts.

![Graph](image)

**Figure 3. Step vs. continuous climb cruise trajectories**

The figure 3 shows the vertical profiles, the TAS, the TLP and the weight derivative of the both trajectories. The TAS is close 480kts except in the trajectory based on steps when the aircraft changes the flight level, which is the same time of a peak of consumption or a maximum TLP.

In conclusion, the cruise based on steps spend 351.4104kg more fuel than the cruise base on continuous climb, because the cruise based on a continuous climb uses more constant TLP than the other one. The more constant use of TLP is associated to constant fuel consumption and besides the optimal aircraft altitude has to be increased due to the waist of mass associate to the fuel consumption.

**CONCLUSION AND FURTHER WORKS**

In the present research it is shown that the most simple aircraft model (3DoF in a quasi-stationary flight) can simulate with accuracy the aircraft trajectory taking into account the computational advantages (less processing time) that it gives to the demanded problem. That is important because in future works it will be required to study in a simulated airspace the optimised Business Trajectories (BTs) provided by the Aircraft TS presented on this paper.

The goal of this second activity will be minimize the probability of ATC tactical intervention. It will receive all the Shared Business Trajectories (SBTs), display them, and use an algorithm to evaluate the possible interactions and reduce them to an admissible minimum. In case there are problematic SBTs, i.e. SBTs that are involved in interactions with hazards, they are taking through the whole process again with the new constraints, producing perturbed SBTs, until they tally with the agreed SBT. The agreed SBT are finally adopted by the system as Reference Business Trajectory (RBT).

Moreover, a simple optimization example is presented. A sign of how important is keep studying the trajectory optimization is that only making a continuous climb in the cruise phase a decreasing of fuel consumption is obtained. Therefore, the future objective is to design full optimal 4D trajectories: take-off, climb, cruise, descent and landing phases. Wind must be taken into account in future works.

**REFERENCES**


An OpenFlow-based Architecture for IaaS Security

Antonio Marotta¹, Gabriella Carrozza², Stefano Avallone¹, Vittorio Manetti²

¹University of Naples Federico II (DIETI) Via Claudio 80125 Naples, Italy
{antonio.marotta, stavallo}@unina.it
²SESM s.c.a.r.l.Via Circumvallazione Esterna, Giugliano in Campania 80014 Naples, Italy
{gcarrozza, vmanetti}@sesm.it

ABSTRACT
Cloud Computing technology and its service model, Infrastructure as a Service, are emerging as the leading approaches to encourage the scalable and efficient utilization of resources and the convenient consumption of elastic services. Despite all the advantages that derive from the application of Cloud Computing IaaS model, when dealing with mission and safety critical infrastructures “built in the cloud”, it is needed to be also aware of the security gaps and concerns. In this work we present our proposed architecture to tackle cloud security issues and we describe the first results of our experimental campaign.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

General Terms
Security, Design, Experimentation, Measurement, Performance

INTRODUCTION
Cloud Computing technology and its service model, Infrastructure as a Service, are emerging as the leading approaches to encourage the scalable and efficient utilization of resources and the convenient consumption of elastic services. The IaaS paradigm allows to deploy, configure and run heterogeneous applications without the need to be conscious about the underlying physical infrastructure. The use of a Cloud Computing platform to create a virtualized testbed in order to reproduce a real operational environment allows to obtain a lot of benefits in terms of:

- chance to reproduce real world scenarios in house to perform testing campaigns;
- availability of automatic procedures to implement backup and disaster recovery of entire testbeds;
- possibility to configure and manage the testbed components and the testbed versioning through automatic mechanisms.

Despite all the advantages that derive from the application of Cloud Computing IaaS model, when dealing with mission and safety critical infrastructures “built in the cloud”, it is needed to be also aware of the security gaps and concerns. Based on the analysis of the literature, Cloud Computing security issues can be related to different scopes.

The way authentication, authorization and accounting are handled assumes a very high influence: security threats are often originated from internal users, so there is the need to be sure that only an authenticated user can access his granted resources, according to clear-cut global policies. The actions performed by users in relation to the platform’s resources should be also registered and accounted for further analysis in case of policy violations. Another important task is the management of the security principles, which are availability, integrity and confidentiality of the cloud data storage. In this case, advanced encryption schemes can be used to guarantee that the proper users are able to access, modify and delete given information.

Virtualization technology, which is the heart behind IaaS model, has rapidly changed the needs and the requirements for network security. Traditional security means, like internal security devices and access control lists are not sustainable when dealing with virtualized servers and resources, due to the strains for making them up-to-date with the rapid changes in the topology. Only authorized hosts and devices should be able to communicate in the virtualized networks, while malicious ones have to be identified and somehow confined. The virtualization layer also poses new security challenges, because virtual guests can be easily compromised in different ways and they can also damage other virtual machines. So, one of the possible remedies is to check virtual machines’ behavior by intercepting attempts in the modification of sensible code. At the same time, virtual machines’ images can be checked, in order to verify their integrity.

In order to deal with security issues in the cloud and with the dynamism, typical of IaaS approach, we propose an OpenFlow-based [1] architecture which uses classical intrusion detection mechanisms to identify patterns of attacks and that realizes mitigation and recovery strategies in reaction to them. The architecture has been designed and implemented in a virtualized testbed, deployed on an IaaS
platform, namely OpenNebula [2], which represents a real world Air Control Center (ACC). The nature of the application fulfilled by the components of the testbed, really stresses out the lack of robust security solutions and the need for automatic procedures of disaster and attacks recovery. Here we present the first experimental activities that were conducted for the design of the architecture and that cope with:

- the performance comparison among different Open Source OpenFlow Controllers;
- the characterization of three different Open Source IaaS platforms on the basis of the Provisioning Time metric;
- the implementation in the selected Controller of a new functionality in order to provide L2 VLAN encapsulation/de-encapsulation.

OPENFLOW AND THE SOFTWARE DEFINED NETWORKING PARADIGM

The way networking is handled and configured in the virtualized testbed is based on the Software Defined Networking [3] (SDN) paradigm, which can be considered as a new way of thinking about the network. It is basically founded on a sharp distinction between data plane, which is still related to the network devices, and the control plane, which is external and logically centralized. The main benefits that derive from its adoption are in terms of a complete isolation for the application layer and the global view of the network. In the first case, researchers can build their own applications on top of the control layer, so that they are completely isolated from the network devices. Therefore you can write new protocols or applications without affecting the internals of the devices. The second advantage deals with the availability of a global view of the network itself, so it is easy to react to events and changes in the topology. OpenFlow is one implementation of this approach and embodies the interface between the control and data layers. It defines all the messages that are exchanged through a secure channel established between the network switches and an external Controller, that determines the logic according to which traffic flows are forwarded. Nowadays, SDN paradigm is extremely appealing to Cloud Computing Networking as a Service, since it represents a flexible way to create virtual network on the fly and to guarantee multi-tenancy L2 isolation or other network services. Furthermore, results obtained from previous conducted analysis and experiments, lead us to confirm that OpenFlow can allow to reach great flexibility in the network, by assuring dynamism and security policy enforcement, without the need to change the internal architecture of the network components. That is why OpenFlow can be considered as an effective mean to face vulnerabilities, even in a dynamic context like the one of Cloud Computing IaaS, and to implement automatic mitigation/recovery strategies, in the case of security attacks.

THE PROPOSED ARCHITECTURE

The architecture we propose has to be analyzed considering three different layers. The Cloud Layer shows two data-centers, which are geographically connected through a private enterprise backbone network. With the aim to further increase the security level in the connection between the data-centers, we use a splitting mechanism based on MPLS [4], MultiProtocol Label Switching Protocol, that splits packet into parts and redirects them to disjoint paths, so that malicious users that intercept traffic are not able to reconstruct the messages. Each data-center has its own IaaS cluster and there is one main node which is in charge of managing the overall infrastructure. In the Virtualization Layer, the view is independent from a particular platform deployed in one of the data-centers. Regarding the organization, every physical machine, namely a “compute” node, hosts a virtual switch to which all the network interfaces of the guests are plugged. In the virtual switching layer we use the OpenvSwitch [5] technology, which offers a set of functionalities, among which the OpenFlow protocol (v1.0) is also implemented. The flow tables of the switches are programmed by an OpenFlow Controller: when a packet generated by a virtual guest arrives to the switch and there is no match with the available rules, it is sent to the controller, which can decide to install a new flow rule in the switch to forward or discard it. All the traffic produced by the virtual machines is controlled and checked against some well-known patterns of malicious traffic to identify possible attacks. When an anomalous network activity is detected, the alarm generated by Snort [6] is sent through a TLS (Transport Layer Security) socket to an Alarm Correlator that performs the following actions:

- event storage;
- notification process after the extraction of information needed to determine the severity level of the attack;
- identification of the mitigation strategy to implement on the basis of the aforementioned severity level. Such a strategy will be triggered by also interacting with the IaaS manager and the OpenFlow Controller.

The mitigation strategy we intend to implement when an attack against a node of the virtualized testbed is detected, consists in migrating the attacked VM to a different data-center which belongs to the same infrastructure. After the migration process is accomplished, the Correlator can instruct the Controller to change the flows in the virtual switch of the physical node where the guest was previously hosted, in order to assure the transparency of its location.
EXPERIMENTAL CAMPAIGN

We carried out the first experimental work with the aim of selecting an Open Source solution among several OpenFlow Controllers. The comparison of the controller’s performances was accomplished through OFlops [7], namely OpenFlow Operations Per Second, which is composed by two software packages:

- OFlops, a particular controller that allows to benchmark lots of features of the switches;
- Cbench (Controller benchmarker), that generates packet-in events for the controller by emulating switches’ connection. It is able to calculate the maximum packet-in message generation rate, the delay between packet arrival and packet-in event and the processing delay.

Table 1 Controllers comparison

<table>
<thead>
<tr>
<th>Flow-mod/s</th>
<th>Nox Hub</th>
<th>Nox Switch</th>
<th>Nox Learning switch</th>
<th>Trema</th>
<th>Floodlight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>7427</td>
<td>19225</td>
<td>7127</td>
<td>52207</td>
<td>56368</td>
</tr>
<tr>
<td>Max</td>
<td>7440</td>
<td>20109</td>
<td>7147</td>
<td>56204</td>
<td>57535</td>
</tr>
<tr>
<td>Avg</td>
<td>7435</td>
<td>19916</td>
<td>7137</td>
<td>54333</td>
<td>57142</td>
</tr>
<tr>
<td>Stdev</td>
<td>3,55</td>
<td>299,31</td>
<td>6,84</td>
<td>1285</td>
<td>407,81</td>
</tr>
</tbody>
</table>

Table 2 Provisioning time

<table>
<thead>
<tr>
<th></th>
<th>Provisioning Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudStack</td>
<td>17,5714</td>
</tr>
<tr>
<td>OpenNebula</td>
<td>22,4789</td>
</tr>
<tr>
<td>OpenStack</td>
<td>27,6996</td>
</tr>
</tbody>
</table>
CONCLUSION AND FUTURE WORK
In this work we firstly discussed the context which deals with the challenge of the main security issues related to the Cloud Computing environment. Then we proposed a SDN-based approach to guarantee network security and to undertake selected reactions in case of attacks, by describing all the components needed by our architecture. As future work we aim at using more sophisticated intrusion detection mechanisms in order to be able to detect unknown and unusual traffic patterns. Furthermore, we intend to extend the experimental campaign by performing a more accurate comparison among Cloud Computing IaaS platforms, that is based on other metrics such as: Elasticity, Agility, network stressing and CPU/memory usage.

REFERENCES
1. Nick McKeown, Tom Anderson, Hari Balakrishnan et other “OpenFlow: Enabling Innovation in Campus Networks” Whitepaper
2. www.opennebula
3. Open Networking Foundation “Software-Defined Networking: The New Norm for Networks” Whitepaper
5. Ben Pfaff, Justin Pettit, Teemu Koponen, “Extending Networking into the Virtualization Layer”. In 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII) New York City October 2009
Control and Stabilization Applied to Micro Quadrotor AR.Drone

(1) K. Boudjit
kamelboudjit@gmail.com
(1) Department of Electrical Engineering
University of Science and Technology Houari Boumediene (U.S.T.H.B.),
BP N°32 EL ALIA, Algiers, Algeria

(2) Chevif Larbes
larbes_cher@yahoo.fr
(2) Ecole Nationale polytechnique Alger
10Avenue hassan Badi BP182, EL-Harrach Alger 16200.

ABSTRACT
The research an autonomous miniature flying robots has intensified considerably, thanks to the recent growth of civil and military interest in Unmanned Aerial vehicles (UAV). In this paper a simple control algorithm to stabilize the attitude of a quadrotor aircraft subjected to various disturbances is presented. First, we present a dynamic model of the quadrotor followed by simulation and implementation of a classical PID controller; the results were inconclusive and show the stability of the quadrotor for different disturbances. Secondly, we present the navigation and control technology embedded in a recently commercialized micro Unmanned Aerial Vehicle (UAV) AR.Drone. Test flight stabilization will be conducted on the AR.Drone, experimental results are discussed and interpreted.

Authors keywords
AR.Drone; UAV; Stabilization; Quadrotor; SDK.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

I. INTRODUCTION
The important progress over the last years in sensing technologies, high density power storage, and data processing have made the development of micro unmanned aerial vehicles (UAV) possible. In the field of sensing technologies, industry can possible currently a new generation of integrated micro IMU composed generally of MEMS technology inertial sensors and magneto-resistive sensors [1]. The last technology in high density power storage offers about 180 W/Kg which is a real jump ahead especially for micro aerial robotic. This technology was originally developed for handheld applications and is now widely used in aerial robotics. The cost and size reduction of such systems makes it very interesting for the civilian market in several application like for small- area monitoring and building exploration. Simultaneously, this reduction of cost and size implies performance limitation and thus a more challenging control [1]. Moreover, the miniaturization of the inertial sensors imposes the use of MEMS technology which is still less efficient than the conventional sensors because of noise and drift. The use of low-cost IMU is synonym of less efficient data processing and thus a bad orientation data prediction in addition to a weak drift rejection. On the other hand, and in spite of the latest progress in miniature actuators, the scaling laws are still unfavourable and one has to face the problem of actuators saturation. That is to say, even though the design of micro aerial robots is possible, the controls still a challenging goal.

Autonomous flying robots have gained enormous commercial potential during the last years. Recent developments in high density power storage, integrated miniature actuators and MEMS technology sensors have made autonomous miniaturized flying robots possible [3]. This new situation has opened the way to several, complex and highly important applications for both military and civilian markets. Military applications currently represent the lion’s part of the unmanned flying vehicle market, and this industrial sector is growing strongly. Are of the advantages of quadrotors is the payload augmentation. They have more lift thrusts than conventional helicopters therefore they offer better payload. Moreover, they are potentially simpler to build and highly manoeuvrable. These advantages qualify a quadrotor as a good platform for autonomous Unmanned Aerial Vehicle research.

In 2004, the Parrot company starter a project named AR.Drone aiming at producing a micro Unmanned Aerial Vehicle (UAV) for the mass market of videos games and home entertainment [2]. The targeted goals of the AR.Drone project go way beyond conventional usages commonly considered in both civilian and military applications. They encompass augmented reality (AR), videos games, and interactivity. As is discusses in this article, this land marking project is a prime example of sophisticated use of low-cost sensors (MEMS and cameras) for mass markets where retail price is of major importance [2]. This project and the embedded algorithm have the particularity of being highly stable, robust and very user-friendly. In order words, the technology yields the way to the enjoyment of playing. The underlying complexity can be completely forgotten.

In this paper a simple control algorithm to stabilize the attitude of a quadrotor aircraft subjected to various disturbances is presented. First, we present a dynamic model of the quadrotor followed by simulation and implementation of a classical PID controller; the results were inconclusive and show the stability of the quadrotor for different disturbances. Secondly, we present the navigation and control technology embedded in a
recently commercialized micro Unmanned Aerial Vehicle (UAV) AR.Drone. Test flight stabilization will be conducted on the AR.Drone, experimental results are discussed and interpreted.

II QUADROTOR MODEL

The quadrotor model is shown in figure 1. A bidy-fixed frame (B) is assumed to be at the center of gravity of the quadrotor, where the z-axis is pointing upwards. This body axis is related to the inertial frame (O) by a position vector \( \rho = (x, y, z) \in O \) and a rotation matrix \( R \in SO(3) \). A ZYZ Euler-Angle representation has been chosen to represent the rotations. It is composed of 3 Euler-Angles, \( (\phi, \theta, \omega) \), representing yaw, roll (rotation around y-axis) and pitch (rotation around x-axis), respectively [1].

A spinning rotor produces moment as well as thrust. Let \( F_i \) be the thrust and \( M_i \) be the moment generated by rotor \( i \), that is spinning with rotational speed of \( w_i \). Let \( V_b \in B \) be the linear velocity in body-fixed frame and \( w_b \in B \) the angular velocity. Therefore the velocities will be,

\[
V_b = R^T \rho
\]

\[
s kem(w_b) = R^T R
\]

Where \( s kem(w) \in SO(3) \) is the skew symmetric matrix of \( w \). To represent the dynamics of the quadrotor [1], we can write the Newton-Euler equations as follows:

\[
mV_b = F_{ext} - w_b \times mV_b \tag{3}
\]

\[
I_b w_b = M_{ext} - w_b \times I_b w_b \tag{4}
\]

\( F_{ext} \) and \( M_{ext} \) are the external forces and moments on the body-fixed frame. \( I_b \) is the inertia matrix, and \( m \) is the mass of the helicopter. Drag on a moving object is given by

\[
\text{Drag} = \frac{1}{2} C_d \rho v^2 A,
\]

in which \( \rho \) is the density of air, \( A \) is the frontal area, \( C_d \) is the drag coefficient, and \( V \) is the velocity. Assuming constant \( \rho \), the constants as the above equation can be combined to from \( C \), which simplifies drag to \( \text{Drag} = CV^2 \).

The force generated by a rotor which is spinning with rotational velocity of \( w \) is given by

\[
F = bL = \frac{\rho}{4} w^2 R^2 abc (\theta_i - \phi_i), \quad \text{where } b \text{ is the number of blades on a rotor, } \theta_i \text{ is the Pitch at the blade tip, } \phi_i \text{ is the inflow angle at the tip. By combining the constant terms as constant variable D, this equation simplifies to } F_i = DW_i^2 \text{ therefore } F_{ext} \text{ and } M_{ext} \text{ will be}
\]

\[
F_{ext} = -C_x \hat{x} + C_y j + (T - C_z \hat{k})K - R^* mg K \tag{5}
\]

\[
M_{ext} = M_x \hat{x} + M_y \hat{j} + M_z \hat{k} \tag{6}
\]

Where \( C_x, C_y, C_z \) are the drag coefficients along x, y and z axes, respectively. \( T \) is the total thrust and \( M_x, M_y, M_z \) are the moments generated by the rotors. The relation of thrust and moments to the rotational velocities of rotors is given as follows:

\[
\begin{pmatrix}
T \\
M_x \\
M_y \\
M_z
\end{pmatrix} =
\begin{pmatrix}
D & D & D & D \\
-DI & DI & DI & -DI \\
CD & -CD & CD & -CD
\end{pmatrix}
\begin{pmatrix}
w_x^2 \\
w_y^2 \\
w_z^2
\end{pmatrix}
\tag{7}
\]

The above matrix \( M \in \mathbb{R}^{4x4} \) is full rank for \( l, C, D \neq 0 \). The rotational velocity of rotor \( i(w_i) \), can be related to the torque of motor \( i(\tau_i) \) as

\[
\tau_i = I_r w_i + Kw_i^2 \tag{8}
\]

Where \( I_r \) is the rotational inertia of rotor \( i \), \( K \) is the reactive torque due to the drag terms. Motor torques \( \tau_i \) should be selected to produce the desired rotor velocities \( (w_i) \) in equation 8, which will change the external forces and moments in equation 5 and 6 [1]. This will produce the desired body velocities and accelerations in equations 3 and 4.

III SIMULATION AND CONTROL QUADROTOR

A controller should pick suitable rotor speeds \( w_i \) for the desired body accelerations. Let’s define the control inputs to be:

\[
u_1 = (F_1 + F_2 + F_3 + F_4)
\]

\[
u_2 = l(-F_1 + F_2 + F_3 - F_4)
\]

\[
u_3 = l(-F_1 - F_2 + F_3 + F_4)
\]

\[
u_4 = C(F_1 - F_2 + F_3 - F_4)
\]

\( C \) is the force-to-moment scaling factor. The \( u_i \) represents a total thrust on the body in the \( z \)-axis,
$u_2$ and $u_3$ are the Pitch and Roll inputs and $u_4$ is a Yawing moment. Backstepping controllers are useful when some states are controlled through other states [1]. Since motions along the x and y axes are related to tilt angles $\theta$ and $\phi$ respectively, backstepping controllers can be used to control tilt angles enabling the precise control of the x and y motions (inputs $u_2$ and $u_3$). The altitude and the Yaw, can be controlled by PD controllers.

$$u_1 = \frac{g + K_p (Z_d - Z) + K_d (Z_d - \dot{Z})}{\cos \theta \cos \omega} \quad (10)$$

$$u_2 = K_p (\phi_d - \phi) + K_d (\dot{\phi_d} - \dot{\phi})$$

Setting different PID coefficients is done in the following order. We began by well adjusting the Pitch and Roll then Yaw, as the Pitch and Roll act on two rotors while the Yaw it is on all four rotors. The goal is to obtain an optimal response system (rise time and the smallest possible answers).

After several tests we noticed that PD is sufficient to stabilize our quadrotor, the parameters chosen are: For the roll and pitch: $kp = 10$, $k = 0$, $Kd = 20$

After having stabilized pitch and roll stabilization Yaw becomes easy, $kp = 2$ and $ki = 0$, $kd = 1$ than enough for optimal response.

Figure 2 shows the simulation block our quadrotor.

![Fig.2. Block Simulink](image)

The parameters used in our simulation are:

- $m=0.638$ the total weight of the quadrotor (Kg)
- $g=9.81$ gravity (m/s²)
- $gOffset=0.0$
- $lx=0.2839$ The moment of inertia along the X (m².Kg)
- $ly=0.3066$ The moment of inertia along the Y (m².Kg)
- $lz=0.0439$ The moment of inertia along the Z (m².Kg)
- $lRotor=(8.3)*(10^{-5})$ The moment of the rotor (m².Kg)
- $l=0.21$ Half the size of the quadrotor (m)
- $b=(3.13)*(10^{-4})$ (m.Kg/rad²)
- $d=(3.2)*(10^{-6})$ (m².Kg/rad²)

Figure 3, shows the simulation results.
It is clear that the stabilization of the pitch and roll and yaw is very fast (≈ 5 seconds). However, these results are very interesting position close to balance "roll, pitch and yaw, close to zero." As in the case of an initial orientation of the distant state of equilibrium by PD control can not stabilize the system is nonlinear. It becomes unstable. The problem of the real system is the same, to compensate for initial conditions far from equilibrium. It takes a significant gain, but nevertheless, a gain too large will cause the engine to saturation.

IV EXPERIMENTAL SETUP

The AR.Drone (Figure 4) is a remote-controlled consumer quadrotor helicopter developed by Parrot. The body is made of a carbon fiber tube structure and high resistance plastic. A protection hull is made of expanded Polypropylene (EPP) farm, which is durable, light in weight and recyclable. The hull provides protection during indoor flights. The propellers are powered by four brushless motors (35,000 rpm, power 15 W) [2]. Energy is provided by a Lithium Polymer Battery with a capacity of 1000 mAh, which allows a flight of approximately 10 minutes.

Fig4. AR.Drone quadrotor helicopter with protection hull attached.

The AR.drone carries an internal computer with a 468MHz ARM9-Processor and 128 MB of RAM, running a custom Linux operating system. A mini-USB connector is included for software flashing purposes and to attach add-ons (e.g., GPS sensor). An integrated 802.11g wireless card provides network connectivity with an external device that controls the vehicle. A remote control is not included instead, a regular Wi-Fi enabled device was initially designed for Apple platform (e.g., iPhone, iPad, and iPod) and because available on other platform during 2001. It is also possible to control the AR.Drone from a Linux or Windows PC with the software designed for application developers [2].

IV.1. Open Application programming Interface

The AR.Drone (Application Programming Interface) is the reference project for developing applications for the AR.Drone. It includes SDK (Software Development Kit) source code written in C, multiplatform examples and documentation. The API does not include software that is embedded on the AR.Drone [2]. Communication with the AR.Drone is done through four main communication services which are implemented in the SDK.

1- Controlling and configuring the drone is done by sending AT commands on regular basis;
2- Information about the drone (e.g., status, altitude, attitude, speed) is called navdata the navdata also includes filtered and raw sensor measurements. This information is sent by the drone to its client at a frequency of approximately 30 (demo mode) or 20 times per second;
3- A video stream is sent by the AR.Drone to the client device. Images from this video stream are decoded using the codec’s (decodes) include in the SDK;
4- Critical data is communication over a channel called the control port. This is a TCP connection to provide reliable communication. It is used to retrieve configuration data and to acknowledge important information such as the sending of configuration information.

Figure 5 shows the principle of communication AR.Drone.

Fig5. Layered architecture of a client application built upon the AR.Drone SDK.

The AR.Drone library is part of the SDK and provides high level APIs to access the drone. Its content can be summarized as following:

SOFT: includes headers files describing the communication structures;
- ARDRONETOOL: a set of tools to easily manage the drone, e.g., communication initialization, input/output device management, high-level drone control functions and navdata receiving and decoding system;
VPSDK: a set of general purpose libraries video processing pieces, multiplatform wrappers for system level functions, multiplatform wrappers for communication functions and helpers to manage video pipelines and threads.
VLIB: the video processing library. It contains the functions to receive and decode the video stream.

IV.2. Tests and results

The use of a ground control interface is essential for the proper conduct tests of this project, particularly powerful and easy to use consists of various wireless communication protocols “Wi-Fi”. The current system allows observing the quad-rotor free movement. This requires the presence of a person who seeks to control
interface settings and sending orders in real time through a keyboard, joystick or gamepad joystick type and receives in return parameters position, speed, acceleration, altitude and even videos and images taken by the two cameras fitted to the drone. To do this, you need a PC with Wi-Fi antenna, a joystick and of course a drone. The following figure shows the GUI Linux used in our tests.

Fig.6. GUI Linux used.

You should see a GUI showing the current drone state. If connection with the drone is successful, the central graphs entitled theta, phi and psi should change according to the drone angular position. In the following screenshot the drone is bending backwards.

The GUI also gives us the ability to view the 4 PWM control signals generated by our microcontroller and send the controller to each motor. These PWM signals are sent to motor commands. They are generated by the Output Compare modules in ARM 9 with a pulse width variation between 0% and 100%.

Fig.7. Signal PWM

From these results we see that the engines 1 and 3 are responsible for pitching and engines 2 and 4 roll. Subsequently several tests will be performed on the pitch and roll to confirm the exact layout engines.

Figure 8 shows the reaction of the four rotors as a disturbance in the functioning of the right roll.

Fig.8. Roll right depending on the PWM.

Figure 9 shows the results obtained during a disturbance to roll to the left, and the reaction of the four motors.

Fig.9. Roll left depending on PWM

Figure 10 shows the results in a disturbance in front of a quadrotor in how our pitching, with PWM control signals generated with four motors.

Fig.10. Front pitch based on PWM

Based on the results obtained during tests with Pitch and Roll functioning disturbance, can be seen to roll (Fig.8, 9), when we gave a disturbance right and left our quadrotor, we have noticed that the motors 2 and 4 are represented by the colors green and black respectively,
reacted immediately to maneuver given and this is confirmed by the PWM control signals generated by our microcontroller.

Better visualized angular velocities along the axes X and Y, tests were carried out on the quad-rotor. Figures (11, 12) clearly show the results.

According to the figure 13 we see when sending a set point, the reaction is instantaneous engine along the Y axis, where the accelerations were immediate same for small perturbations caused this confirms the stability of the quad-rotor efficiency and 3-axis Accelerometer implanted in our devices that allow us to measure in real time the different accelerations.

V CONCLUSION
The purpose of this paper is to present the navigation and control technologies embedded in the commercial micro UAV AR.Drone.

Current developments are now focused on video games to use this aerial platform in interactive augmented reality game plays. Now that low-level control algorithms have been developed and have proven to stabilize the system in sufficiently various flight modes, one can expect control design to play a key-role by means of guidance in such applications. This a subject for future works.

REFERENCES
2. P.J. Bristeau, F; Callou, and all; “The Navigation and Control Technology Inside the AR.Drone micro UAV”, Preprints of the 18th IFAC World Congress Milano (Italy) August 2011.
Automated Task Allocation

Rocío Barragán Montes  Eduardo García  Francisco Javier Sáez Nieto
CRIDA  CRIDA  UPM
Grupo de Investigación Navegación Aérea (GINA)  Universidad Politécnica de Madrid (UPM)
Pza. de Cardenal Cisneros, nº 3, 28040 Madrid, Spain  rocio.barragan@upm.es  egarcia@crida.es  franciscojavier.saez@upm.es

ABSTRACT
The goal of the paradigm shift in Air Traffic Management (ATM) is to increase its overall performance by means of redesigning processes, evolving to a more automated, autonomous and predictable system. Nevertheless, when dealing with automation, it is important to determine until what extent can any part of the system or process be automated, that is, to determine the right player. It is also likely to be tempted to make the system as autonomous as possible, avoiding the stiffness introduced by centralised processes, which means to choose the right place to drive it. Finally, it is also commonly accepted that the sooner an activity is planned, the more predictable the system will be, i.e., to determine the right time. However, reality creates constraints that make it impossible to reach the ideal status: fully automated, completely autonomous and totally anticipated. Considering the ATM system as a set of tasks and functions, its allocation can be defined as their placement in time, place and player. This paper presents operational research methodologies to estimate the best time, the best place and the best player for optimal performance of the ATM system.

Author Keywords
Task Allocation; ATM; Level of Automation; Anticipatory; Compensatory; Centric; Autonomous; Optimisation; Operations Research; Decision Support Tools; Performance Metrics.

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
Three Characteristics
In the context of civil air navigation, where SESAR is aiming at improving the ATM global efficiency, there are key features to achieve the paradigm shift [8, p. 3-4]:

- Moving from airspace towards trajectory-based operations, such that each aircraft follows its preferred route and arrives at its desired time of arrival; the so-called Reference Business Trajectory (RBT);
- Dynamic airspace management, facilitated by a central network, to enhance coordination between aviation authorities;
- New and innovative technologies for more precise navigation and surveillance in order to optimise airspace capacity and maintain a sufficient level of safety in complex, high-traffic, and time-critical situations.

As it is stated in HALA's Position Paper [2, chapter 4.2], the shift will be focused on ATM invariant processes and new role assignments based on three interdependent criteria, having overall system performance as main driver for ATM automation:

- Its anticipatory/compensatory components [4],
- The degree of ATM agents involvement (autonomous vs. centric), and
- The level of automation [7].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. ATACCS’2013, 28-30 May 2013, Naples, Italy.

Figure 1. 3-D task allocation.

Figure 1 represents a 3-D perspective of the task allocation characteristics. The optimum state would be highly automated (10 in Parasuraman, Sheridan and Wickens levels), totally anticipatory in Hollnagel Extended Control Model (ECOM) and fully autonomous. Nevertheless, reality
creates constraints that make it impossible to reach the ideal status. The PhD thesis will be devoted to develop a methodology to optimise the task allocation.

The inception in the concept of task allocation, is the improvement of the overall system performance. This performance will be estimated by Key Performance Indicators (KPIs), gathered into eleven Key Performance Areas (KPAs), established in [6, Appendix D]:

- KPA-01: Access and equity
- KPA-02: Capacity
- KPA-03: Cost-effectiveness
- KPA-04: Efficiency
- KPA-05: Environment
- KPA-06: Flexibility
- KPA-07: Global interoperability
- KPA-08: Participation by the ATM community
- KPA-09: Predictability
- KPA-10: Safety
- KPA-11: Security

There is a factor which could be added to the list: resilience as defined in [1], [5], [10] is the intrinsic ability of a system to adjust its functioning prior to, during or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions. Recently it has been included as a part of the topics studied in SESAR ComplexWorld network, which is developing the project MAREA to support effective application of resilience engineering. Hence, it will be comprised in this PhD thesis as a component to be taken into consideration.

The KPIs will supply the variables to the equations which will be used to optimise the system, while the position in the 3-D coordinates presented in Figure 1 are the means to represent the results.

**Why Operational Research**

Since the World War II, there has been many applications of Operational Research (OR), Operations Research in US English, in strategic-level Decision Support Tools (DST). It is stated in [9] that although OR originally concentrated on solving problems at the operational research, support tools for strategic-level decision-making have become increasingly popular not only in academia but also in the industrial world. This acquaintance supports the decision of using OR as the mathematical drive to solve the proposed problem.

The first step in approaching an OR problem is the posing. The modelling will retain the most relevant elements of the reality into mathematical equations, where the unknowns, are the Decision Variables (DV). This part of the problem is the key, whereas if the equations are ill posed, the results will be completely wrong. There are two kinds of mathematical expressions: the Objective Function (OF) which must be optimised (maximise or minimise) and the existing restrictions, which are the Constraint Set (CS).

Secondly, in order to solve a set of OR obtained equations, there are different methods which can be used as explained in [3], depending on the nature of the mathematical statements. With the help of a mathematical programming tool, the equations are expressed and solved, and the values of the unknown DV determined in order to optimise the function.

Finally, once the first results have been found, they must be validated in order to show their applicability to reality, analysing their effect on the system. The results of this assessment process may show that the posing was not correct, so it may be possible to be forced to reformulate some of the equations. This phase is known as post optimality analysis.

**PHD APPROACH**

**The Objective Function**

Building up the mathematical model will take into account the different KPAs and will use its KPIs to determine the OF, or in our case, OFs, and its DV, among which, the coordinates in the Figure 1 (3-D task allocation) will be established.

In order to simplify the posing of the OF, the KPAs will be grouped into three related OR. See Table 1.

<table>
<thead>
<tr>
<th>Operational Research Phase</th>
<th>KPAs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operational level (OOR)</strong></td>
<td>KPA-02 – Capacity</td>
</tr>
<tr>
<td></td>
<td>KPA-06 – Flexibility</td>
</tr>
<tr>
<td></td>
<td>KPA-09 – Predictability</td>
</tr>
<tr>
<td></td>
<td>KPA-10 – Safety</td>
</tr>
<tr>
<td></td>
<td>Resilience</td>
</tr>
<tr>
<td><strong>Efficiency level (EOR)</strong></td>
<td>KPA-03 – Cost-effectiveness</td>
</tr>
<tr>
<td></td>
<td>KPA-04 – Efficiency</td>
</tr>
<tr>
<td></td>
<td>KPA-05 – Environment</td>
</tr>
<tr>
<td><strong>Feasibility level (FOR)</strong></td>
<td>KPA-01 – Access and equity</td>
</tr>
<tr>
<td></td>
<td>KPA-07 – Global interoperability</td>
</tr>
<tr>
<td></td>
<td>KPA-08 – Participation by the ATM community</td>
</tr>
<tr>
<td></td>
<td>KPA-11 – Security</td>
</tr>
</tbody>
</table>

Table 1. Operational Research Phases applied to ATM

The three levels indicated above can be substantially analysed independently by following the sequence shown in Figure 2.
The OFs in each level will be stated but they must be checked, before the following step. The validation process will proceed as described in Figure 2.

At the first loop, an Operational level OR (OOR) analysis will be conducted to provide the optimised task allocation from the operational point of view, by using detailed KPIs from capacity, predictability/flexibility and safety/resilience areas. Only generic indicators from the other two levels will be considered at this stage.

![Flow diagram of the operational research objective function definition](image)

Figure 2. Flow diagram of the operational research objective function definition

Sensitivity analysis would be carefully performed at the OOR level to identify operational DVs that strongly affect the other two levels, i.e., Efficiency and Feasibility.

A similar process (EOR) will be developed for the second level (Efficiency) by considering as DVs, those related to economical efficiency and environmental impact KPAs, plus those that are also sensible DVs derived from the previous sensitivity analysis. The obtained result will provide relevant information about the satisfying Operational and Efficiency indicators.

Then, depending on whether the satisfying test is considered positive, a final process (FOR) will be conducted for the third level (Feasibility). In this case, DVs related to liability issues, social and institutional effects KPAs, plus the sensible DVs derived from the two previous sensitivity analyses will be considered.

The steps presented in Figure 2 for deriving the solution will iterate until the global solution is found.

The main setback of the process is due to the fact that the KPAs in each level are interconnected. The relationship could be direct or inverse or there could be no immediate connection at all. Figure 3 shows the relations of the areas included in the OOR as an example. The arrows denote the direction of the relationship while the position of the cones indicates whether the characteristic is increased or decreased for the sake of the connection. For instance, the relationship between capacity and safety is: the more capacity the less safety. However, the relationship between flexibility and capacity is: the more flexibility, the more capacity.

![OOR Areas Relationship](image)

Figure 3. OOR Areas Relationship

Following SESAR recommendations, optimising some KPAs would seem to be detrimental to other KPAs. Nevertheless, conceiving the system as a whole, the overall performance will be maximised. For instance, enabling a 3-fold increase in capacity would mean decreasing safety. Using this methodology, it will be exhibited that other factors affecting safety can be refined to reach the improvement of safety performance by a factor of 10, for example, those belonging to predictability.

The Constraint Set

Other factors that will be taken into account and will define the CS will be the scenario constraints:

- Airports capacity,
- Atmosphere behaviour and
- Airspace limitations.

Also the state of the art of technical enablers, established by robustness and quality performance indicators for:

- Communications, navigation and surveillance sensors,
- Information technologies,
- Automation and Human-Machine Interfaces (HMI).

And finally, the invariants in ATM.
Additionally, there are several agents that compose the system as shown in Figure 4 and that determine the point of view under which the CS will be stated: the Network Manager (NM), Air Navigation Service Providers (ANSP), Airports, Airlines Operation Centre (AOCs), Meteo Info Suppliers and Military Operations. There is also a starring "actor", which is the Aircraft, shown in yellow in Figure 4, the one that plays the "subject", which is the trajectory.

As it is difficult to extract the right answer from the wrong problem, a post optimality analysis must be carried out at this point. Once the CS is established, and carefully analysed in order to make sure it matches reality, the system must be resolved.

**The Expected Conclusions**
Using a Mathematical programming tool, the system will be solved and with a friendly user interface, it could be used as a "What if" tool, in order to determine the DV that can make the improvement of the system grow significantly.

A possible extension of the research would be considering that the constraints are not always fixed and in the medium and long term some of them will change. What will happen then and how the overall performance could be improved if something in the present is changed could also be a reason to reallocate certain task.

**ACKNOWLEDGMENTS**
The PhD grant of Ms. Barragán is supported by CRIDA.

**REFERENCES**
8. SESAR Master Plan D5, April 2008, document no. DLM-0710-001-02-00.
Designing Alarm Device for Socio-Technical Systems: Co-active or Coercive Design?

Paola Amaldi  
School of Life and Medical Sciences  
Hatfield University, AL10 9AB  
p.amaldi@herts.ac.uk

Sara Khalil  
School of Life and Medical Sciences  
Hatfield University, AL10 9AB  
sarakhalil71@gmail.com

ABSTRACT
The behavior of an onboard alarm system called Air Traffic Collision and Avoidance System (ACAS) has been scrutinized to provide some evidence with respect to its role in monitoring and controlling Air Traffic Management. Its behavior was then contrasted to the ‘interdependent matrix’ proposed by Johnson, Bradshaw, Feltovich, Jonker, (2010) whose aim is to characterize coactive design. ACAS turned out to be more ‘useful’ to monitor traffic than ‘necessary’ to avoid mid air collision. However it scored low on those dimensions associated with coactive design.

Author Keywords
Alarm system, ACAS, coactive design

ACM Classification Keywords
Human automation interaction

General Terms
Human Factors; Design.

INTRODUCTION
Improving or maintaining safety in socio-technical systems is a constant challenge. The main challenge is to strike a balance between main competing pressures like compliance, “creative adaptation”, workload management, resource management, economic and political pressures to name a few (Dekker, 2005; Rasmussen, 1997). Other challenges come from introducing automated tools that while supposedly enhancing safety by offering new protection, they introduce new requirements for co-ordination.

The challenge introduced by automated tools is that critical co-ordination/co-operation patterns are disrupted if the interactions between tools, procedures, tasks and agents are not enough understood at the time a new device is being designed and developed. Traditional approaches to human automation interaction emphasize levels of automation and the sharing of tasks/responsibility as to avoid unwarranted interference in each other spheres of activity.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ATACCS’2013, 28-30 May 2013, Naples, Italy.

COACTIVE DESIGN
A recent approach in supporting co-ordination for human and machine join activity is to decompose automation along two orthogonal dimensions: self-sufficiency and self-directedness (Feltovich, Bradshaw, Clancey, & Johnson, 2007; Johnson, Bradshaw et al., 2010). Self sufficiency refers to the degree of autonomy of the automated agent form the human input and self-directness refers to the degree of its autonomy from the human control.

When both self-sufficiency and self directedness are low, most of the control is manual; as self-sufficiency increases but the automated agent is not given more autonomy, the automation becomes underused; a complacent mode of interaction will result when the system is highly self-directed but not enough self-sufficient meaning that it is over-trusted; finally the risk of opacity arises when the automation is highly self sufficient and self-directed but not fit for joint activity with human agents thus making it difficult for other members to maintain awareness of the transformations induced by the automation.

Adding a set of requirements to support interdependent activity can counteract issues of ‘opacity’. To overcome the ‘autonomous matrix’ pitfalls, coactive design advocates for a conceptual shift in human automation interaction. The shift is a turn away from a focus on defining and re-defining of rules for function allocations among autonomous (or semi) systems, to an emphasis on interdependence among activities distributed over autonomous or semi-autonomous agents. Such interdependence has, however to be supported by some regulatory mechanisms that while ensuring some order and predictability among activities, allow for coordination among agents. This inter-predictability allows agents for example, to be able to monitor and being monitored, be dependent and being depended on.

Self-directedness and self sufficiency have to be supplemented by a capability of taking into account higher level goals or other interdependent tasks that while falling outside the scope of the agent’s responsibilities might be affected or might affect the achievement of the agent’s main goals. Coactive design includes, therefore, a third axis, interdependence among agents whose requirements include providing assistance, transparency and feedback. We shall refer to this as the interdependence axis.
In other words, inter-predictability which supports coordination within joint activity, is the extent to which agents can be “mutually predictable, understandable and directable.” (Feltovich, Bradshaw, Clancey & Johnson, 2007, pg. 178). In what follows we highlight the purpose of our investigation

OUR STUDY

Our investigation aims at characterizing the management of an on board warning system ACAS (Airborne Collision Avoidance System) that monitors trajectory separation of pairs of aircraft. ACAS is supposed to be a ‘last resort safety net’ and it provides two kinds of warning: a Traffic Advisory (TA) supposed to alert pilots to possible incoming loss of separation and a Resolution advisory (RA) to which pilots have to comply no matter how pilots and/or ATC assess the situation. Through an analysis of 1636 Incident Reports we intended to uncover what roles does the onboard safety device (ACAS) fulfill. In particular given Bradshaw et al framework of coactive design, we ask how does score on the dimensions constituting the interdependence matrix? In other words, is ACAS high in self directedness and self sufficiency and at the same time have enough knowledge of the operational environment to support coordination of joint activity? The Incident Reports analysed were filled in during the year 1991 to 2008 anytime ACAS triggered off pilots had to fill in a template requesting information about the type of maneuver, the type of a/c whether there was any traffic information given by ATC before or after the triggering of the triggering off of the safety device.

Main Findings

In most cases we analysed data starting from the year 2004 to better scrutinized ACAS behaviour when the new longitudinal separation regulation became enforced i.e., Reduced Vertical Separation Minima—RVSM). In one case, we looked at the whole database because the answer to the question would not be affected by the regulated longitudinal separation.

Data underwent basic quantitative analysis and qualitative narrative analysis. Following is a number of questions we used query our database.

(I) Pilots’ compliance to RAs was 90% and the distribution of different RAs was as follows: in 40% of the cases pilots were expected to respond to a ‘corrective’ RA, implying a change in current flying altitude (see percentages reported for “Climb” and “Descent” RAs in Fig 1). These are most disruptive for ATC as under RA instruction, flight level change cannot be coordinated with ATC. The remaining 60% of RAs required either adjusting or simply monitoring the current vertical speed (see AVS and MVS in fig 1). These are called preventive RAs as they do not imply a change in the flying trajectory. However change in vertical speed might need coordination, notably during the approach phase. Again under RAs instruction no coordination is allowed.

(Figure 1. Interdependent matrix. Adapted from Johnson et al. 2010)

In almost 60% of the cases the alert triggered off when longitudinal separation was within safety standards (see item ‘above 1000’ in table 1); (III) In about 90% of the times RAs triggered off while aircraft were changing flight level (see fig. 2, all of the phases except ‘cruise’) This suggests that the alarm is ‘sensitive’ to lack of knowledge of “intent” of the aircraft (particularly as it concerns Flight level changes not included in the (seasonal) plan) that could trigger the alarm.

Figure 2 Distribution of RAs

(II) In almost 60% of the cases the alert triggered off when longitudinal separation was within safety standards (see item ‘above 1000’ in table 1);
(IV) The high rate of compliance to RAs (90%) was not mirrored by an overwhelming positive assessment of the RA which was rated ‘necessary’ less than 20% of the times (Amaldi, Chellappah & Mansour, 2007). Further analysis of the narratives shows the distribution of ‘useful’ ‘nuisance’ and ‘necessary’ RAs across the different conditions evoked by pilots.

(V) Reasons provided by pilots for lack of compliance to RAs, confirmed that ACAS was one among the several sources of information that pilots consulted to assess hazard and making decisions. The textual analysis of the narratives showed the lack of pilot’s intent knowledge of the alarm and at the same time, pilots used several cues as confirmatory evidence to account for their lack of complacency. The following list refers to common reasons.

- Vertical separation as reliably assessed and displayed by ACAS. Pilots reported reading the value on the navigation display;
- Visual search. They monitored the position of the “intruder” visually;
- Pilots received traffic information from ATC about intents of the intruder; particularly important when the “intruder” is a military aircraft not visible on the ACAS display.
- Pilots made use of the “party line”, i.e. “listen in” the conversation between the “intruder” and ATC to infer intent of other parties.

CONCLUSION
Overall these findings suggest that ACAS does not behave primarily as a ‘last resort safety net’ as it triggers off in the 90% of the times when aircraft are changing flight level and in 60% of the times separation minima are not violated. Further its lack of knowledge of intent of other agents is recognized and judged an important condition of its triggering off. Nevertheless the advisory provided along with visual information is considered useful and used in conjunction with other information (like visual acquisition, ATC, pilots’ communication) to make decisions. Our analysis of the narratives and a previous set of interviews we conducted with airline pilots (McMorrow, Amaldi, and Boiardi, 2005) indicated that that in a number of cases pilots justified the occurrence of the RAs when they could understand its logic, as for example, in the case of a 1000 ft level off, i.e., when ACAS ignores that the two aircraft will settle well before their ascending and descending trajectories intercept. To conclude in terms of the interdependence matrix, ACAS scores quite high in terms of sufficiency, being completely autonomous (from ‘manual input”) in detecting traffic problems and issuing resolution. However its actual self directedness is quite low: it issues advisories but it is completely dependent on human input for the execution of the maneuver. Concerning its interdependence it provides good assistance as revealed by the ‘useful” and “necessary” appraisals and some feedback during the implementation of the corrective maneuver, but it scored quite low in terms of its predictability, detectability, transparency, and knowledge of intent of the other agents including higher level goals or other aspects of the task environment. These findings might suggest a shifting away from classifying ACAS as a compelling (coercive?) alarm to an attention alerting device (coactive?)

ACKNOWLEDGMENTS
We are grateful to the ACAS Eurocontrol team led by Dean Garfield and to Anusha Chellappah and Heba Mansour for their help in the data analysis.

REFERENCES


Towards an Integrity Monitoring for 4D-trajectories

Alexander Schwithal
Institute of Flight Guidance
Technische Universitaet Braunschweig
27, Hermann-Blenk-Str.
Braunschweig, 38108 Germany
+49 531 391 9863
a.schwithal@tu-bs.de

Peter Hecker
Institute of Flight Guidance
Technische Universitaet Braunschweig
27, Hermann-Blenk-Str.
Braunschweig, 38108 Germany
+49 531 391 9802
p.hecker@tu-bs.de

ABSTRACT
For the years 2030 and beyond aircraft are expected to fly optimal trajectories that are defined in the form of three dimensional waypoints plus associated required times of overfly. In order to maintain separation and exploit the benefits of these 4D-trajectories, aircraft must stay within very small volumes around the reference track. Challenging uncertainties arise when it comes to unpredictable events like for example fast changes of meteorological conditions. Some of those imperfections during the prediction phase of the trajectory can be later compensated by closed loop control. Therefore, this work introduces the notion of uncertainty prior and after closed loop control. That approach splits the predicted trajectory tracking performance into traditional aircraft performance and other input parameter prediction, together with their associated uncertainty sources and another block that models the capability of controls to soften the impact of certain input variations. This aims at determining what input uncertainties could be compensated by applying closed loop or manual control and finally to come up with a more reliable aircraft state prediction not only in terms of accuracy but also with respect to integrity. In order to do so, uncertainties on the relevant input factors must be identified. Their impact on the state estimate has to be quantified in scenarios with and without closed loop control. The paper will present an approach towards a quantification of the predicted state variable, exemplarily for the Estimated Time of Arrival (ETA) prediction. The outcome is a prerequisite for specifying accuracy and integrity of the predicted future aircraft states that will provide benefit to a more reliable trajectory tracking in the context of future Air Traffic Systems. It will further discuss the remaining gaps between the integrity monitoring of today’s aircraft trajectory tracking compared to fully 4D contained trajectory operation as it is envisaged by the SESAR master plan [1].

Authors keywords
4D-Trajectory, RNP, R4DNP, Contract Tube, Aircraft navigation, Navigation Performance, Closed-loop control

ACM Classification Keywords
J.2 [Physical Sciences and Engineering]: Aerospace.

INTRODUCTION
When it comes to a change towards 4D trajectory operation there are two kinds of information that are vital for the global Air Traffic System (ATS) to have. The first is a precise knowledge of the aircraft state vector at the moment and for the future. The second is an associated measure of trust that describes the accuracy and integrity of the aircraft system performance to stick to the predicted (or planned) aircraft states. Knowing those two things would provide a huge benefit to the overall Air Traffic System in terms of efficiency, throughput, and complexity as well as controller workload. Right now, a significant amount of controller workload is created by the fact that controller have to spend time and attention to resolve conflicts just because they simply do not know where exactly the aircraft is going to be in the close future [2]. They are missing reliable information about the aircraft tracking performance, especially in the fourth dimension. The latter presents the aircraft’s capability to meet required times of overfly (RTOs) for a finite number of waypoints along the flight path. This uncertainty directly leads to an increased sector complexity which again causes the controller workload to rise dramatically. Therefore, the goal of this work is to come up with a measure of trust that over-bounds possible errors inside the whole 4D trajectory tracking system to a dynamically calculated volume, which, at the same time, guarantees autonomous performance monitoring to keep the integrity risk below a well defined value.

MOTIVATION
Looking at current generations of auto pilot systems shows that many of them follow the idea of certifying a total system tracking performance for controlling the flight path in the horizontal plane. In a first approximation the total system accuracy is given by the navigation accuracy on one hand and flight control accuracy on the other. While the first represents the accuracy of the current navigation means, the second refers to what the flight control is capable of doing in the presence of an assumed level of disturbances, e.g. wind. Other approaches like the required navigation performance (RNP) even go a step further and specify not only stringent values for accuracy but also
define requirements for an integrity monitoring onboard the aircraft. Recent SESAR activities extend the existing 2D plus lateral guidance strategies of modern Flight Management Systems (FMS) to initial 4D capabilities [2],[3]. These systems already evaluate the current errors between planned and actual aircraft states in all four dimensions. Furthermore, they provide ETA estimates at least for the metering fix up to which the whole Controlled Time to Arrival (CTA) procedure is executed. However, there is no value that indicates the quality of the estimates, for instance in the form of standard deviation. It is exactly that missing piece of information that, if available, would allow the controller to focus on aircraft that for some reason provide inaccurate estimates rather than dealing with aircraft that have a high probability to fulfill their estimates, knowing that these predictions meet well defined requirements in terms of accuracy and integrity.

The following sections describe the approach that this work makes towards a quantification of the ETA estimation quality. One major difference in this approach compared to current uncertainty estimation techniques is the time frame. Current 2D RNP performance evaluation only monitors the momentary tracking error, consisting of a navigation and a control uncertainty budget. In contrast, a performance monitoring for an ETA point in 4D involves monitoring the current aircraft tracking performance and considering effects that will influence the flight up to that point in the future, like disturbances but also control maneuver for compensation. In other words, the new approach has to make predictions for the future, which is not the case for current 2D RNP performance monitors. This paper ties to current techniques and suggests ways to get to an error overabounding for 4D trajectories.

TRAJECTORY PREDICTION

In this paper, it is assumed that a full 4D trajectory prediction in the form of a time series of suitable aircraft states can be achieved if the following prerequisites are fulfilled:

1) The aircraft intent information is given, for example in the form of a flight plan. It defines the planned trajectory that has been negotiated with Air Traffic Control. Furthermore, it can be considered conflict free until further notice. Thus uncertainties that are related trajectory changes are not taken into account.

2) An aircraft model is available that allows the propagation of aircraft states over time, including basic functions of an auto pilot system. Additionally, it shall provide estimates of aircraft performance measures \( (V_{TAS \text{ min}}, V_{TAS \text{ max}}) \). Those can be used to determine the capability of closed-loop control and the resulting input uncertainty mitigation.

3) The current and future navigation sources (and modes) are known or can be assumed in form of a state machine. For example, start with GBAS, then switching to SBAS and finally going back to GBAS again. Those define the uncertainties for the position inputs.

4) A model that estimates the probabilities that input parameters that have been used for generating the initial trajectory, are different during the trajectory tracking phase, for example differing wind. Here, not the absolute value is necessary but only the difference between assumed and real condition as well as the probability for this situation to occur. Those values affect the propagation of momentarily uncertainties to future points along the trajectory, which is a new but necessary step for contained 4D operation.

For this work, the focus lies on aircraft states that are essential for 4D trajectory operation, thus those that represent the aircraft evolution with respect to ground. They include aircraft position (latitude, longitude, altitude, true airspeed (as it represents the aircraft performance under given wind), vertical flight path angle, course angle, and the ground speed. Moreover, it includes the exact time stamp \( t \), for which the vector \( \mathbf{x} \) is valid.

\[
\mathbf{x} = (p_{\text{lat}}, p_{\text{lng}}, p_{\text{alt}}, V_{\text{TAS}}, \gamma, x, V_{\text{ground}}) \quad (1)
\]

The reason for the numbered points mentioned above is explained in the following. As long as no disturbances occur, the actual flight path will be identical with the initially planned flight path. It can be considered as the reference track. Very small volumes around this reference track can be guaranteed with a high probability. However, when it comes to variations of the input parameters like for example wind during the tracking phase of the trajectory operation, errors occur that widen up the volumes and increase the uncertainties for the prediction.

UNCERTAINTY PROPAGATION

This section describes how to propagate uncertainties on the initial position to the ETA estimate for a straight line segment. Therefore, we calculate the arrival time by the nonlinear function

\[
t_{ET\Delta i+1} = t_i + \frac{1}{V_{GS}} \sqrt{(p_{H,i+1} - p_H)^2 + (p_{V,i+1} - p_V)^2} \quad (2)
\]

, where \( t_{ET\Delta i+1} \) denotes the ETA for waypoint \( w_{i+1} \), \( V_{GS} \) the groundspeed and \( p_H, p_V \) the horizontal/vertical position again for waypoint \( w_i \), here considered the initial waypoint. The splitting into horizontal and vertical position components is due to the fact that most navigation systems use different means for horizontal and vertical positioning or at least perform very different in those two components due to geometry reasons.
The uncertainty of the inputs are transferred to the output using GAUSSIAN error propagation, here in a linearized form via first order TAYLOR series expansion. Therefore, we calculate the JACOBIAN matrix $J$, which is evaluated at the working point, here the mean value.

$$J = \begin{pmatrix}
    \frac{\partial J_{ETA}}{\partial p_{H}} & \frac{\partial J_{ETA}}{\partial p_{V}} & \frac{\partial J_{ETA}}{\partial V_{GS}} \\
    \end{pmatrix}$$ (3)

The input uncertainties are defined to

$$\sigma^{2}_{IN} = \begin{pmatrix}
    \sigma_{p_{H}}^2 & \sigma_{p_{V}}^2 & \sigma_{V_{GS}} \sigma_{p_{H}} \\
    \sigma_{p_{V}}^2 & \sigma_{p_{H}}^2 & \sigma_{V_{GS}} \sigma_{p_{V}} \\
    \sigma_{V_{GS}} \sigma_{p_{H}} & \sigma_{V_{GS}} \sigma_{p_{V}} & \sigma_{V_{GS}}^2 \\
    \end{pmatrix}$$ (4)

and indicate the variances and covariances of the input parameter. In general it can be assumed that $\sigma^{2}_{IN}$ is symmetric, thus

$$\sigma_{p_{H},p_{V}} = \sigma_{p_{H},p_{V}}; \sigma_{V_{GS},p_{H}} = \sigma_{V_{GS},p_{H}}; \sigma_{V_{GS},p_{V}} = \sigma_{V_{GS},p_{V}}.$$ (5)

Finally, the output uncertainty for the ETA estimated is given by

$$\sigma^{2}_{eta} = J \cdot \sigma^{2}_{IN} \cdot J^{T}$$ (6)

If additionally the time step for the initial position measurement is also considered a variable that is affected by uncorrelated uncertainty, equation 6 is extended to:

$$\sigma^{2}_{eta} = J \cdot \sigma^{2}_{IN} \cdot J^{T} + \sigma^{2}_{t}$$ (7)

TRAJECTORY PREDICTION

So far we derived the predicted output variance for the case that no control algorithm has been implemented. At this point we introduce the notion of uncertainties prior and after closed loop control. Clearly, uncertainty on one of the input parameters does not necessarily cause uncertainty on the output, i.e. the state prediction. As long as a closed loop control is active, either in the form of an engaged auto-pilot or a human-in-the-loop pilot who follows indicated countermeasures, those disturbances on the input may have no visible effect on the output. We therefore distinguish between uncertainty consequences prior and after closed loop control. Going back to our initial goal of providing probabilities keeping the reference trajectory, it becomes clear that the performance of the closed loop control is a key element for that. In this case control performance does not refer to the regular sense of control theory that defines control stability in terms of gain or phase margins. They are rather defined in the sense of what capabilities and countermeasures are currently available to react to disturbances. It is the aircraft performance in a given environment that defines the margins for the aircraft to react. Therefore, those aircraft performance measures and optional other compensation techniques have to be considered, in our case as a part of the aircraft model. Other approaches do not rely on adapting the ground speed but try to match the remaining distance instead in order to meet time requirements. They make use of precise navigation capabilities for along-track error compensation, for example in the form of flight path tunneling, stretching etc. This approach is described for example in [7] and only mentioned for the sake of completeness. During first initial 4D evaluations, pilots stated that “… they should be in charge of speed control, and the controllers should facilitate or propose lateral changes if speed adjustments were not sufficient.”[2] Consequently, in practice the control will be a combination of speed and lateral maneuvers. The latter is not part of this paper

GROUND SPEED MONITORING

Another influence to a contained trajectory tracking is the integrity performance of the navigation sensors that provide the groundspeed feedback to the 4D trajectory control system.

While nowadays, the aircraft’s ground speed is a nice to have value for predicting approximate arrival times, it is going to be crucial when it comes to the envisaged 4D-trajectory operation. There, ground-speed is a key aspect for calculating the accuracy and compliance for along-track components. Ground speed control is not only important for estimating arrival times for different points throughout the trajectory. It is also important to have to apply the right amount of true airspeed in order to compensate for a given wind disturbance. The correlation here is given by

$$V_{GroundSpeed} = V_{TAS} + V_{Wind}$$ (8)

Consequently, uncertainties in the wind speed directly affect the groundspeed as long as those are not compensated by adapting the airspeed.

$$\sigma_{V_{GS}} = (1 - k) \cdot \sigma_{V_{Wind}}$$ (9)

In this representation (equation 9) index k models the ability to soften variability in the assumed wind. It is defined in the range $k \in \mathbb{R} (0,1)$, from 0 for scenarios with no control, up to 1 for scenarios where control can compensate wind deviations completely. The k value itself is a function of nominal airspeed $V_{TAS}$, minimum and maximum possible airspeeds $V_{TAS_{min}}$ and $V_{TAS_{max}}$ as well as the remaining flight time $t_{remain}$.

$$k = f(V_{TAS}, V_{TAS_{min}}, V_{TAS_{max}}, t_{remain})$$

$$k \approx S \cdot t_{remain} \cdot \frac{(-V_{TAS_{max}} - V_{TAS_{min}}) \cdot (V_{TAS_{min}} - V_{TAS_{max}})}{\sqrt{R}}$$ (10)

The term denoted with R models the robustness of the initially negotiated reference airspeed relative to the extreme limits. Because of the integrating nature of speed control for ETA tracking, its capabilities increase with...
higher remaining flight time. The scalar \( S \) scales the value \( k \) to match the domain of definition.

The control limitations \( V_{TAS_{\min}}, V_{TAS_{\max}} \) change with the evolution of the aircraft states over the planned trajectory and can be derived from an aircraft performance model.

Additional dependencies could be modeled by considering a cost index that models the control effort that one is willing to apply in order to soften deviations.

Up to this point we only considered ground speed uncertainty as a consequence of variations in the wind speed. Modeling the closed loop error propagation assumed a perfect measurement of the groundspeed to feedback and match the reference speed. However, the ground speed itself must be considered a measurement that subject to sensor noise and sensor integrity. If those measurements are taken for integer to ensure the integrity of the output value ETA, then the input variable “groundspeed” must have some kind of error over bounding itself. This is a gap that somehow has to bee filled.

Therefore, we suggest applying similar monitoring measures as for current integrity navigation systems for example in the form of groundspeed protection level. Those techniques have been put into place for example for SBAS or GBAS systems [5] and provide means of guaranteeing specific degrees of accuracy and integrity by monitoring worst-case error scenarios.

The challenge here is not to assume worst-case errors for all aircraft types, model performances and input parameter variations but instead having different error bounds depending on the situation. The goal is to provide tighter error bounds having at the same time a high degree of integrity but without penalizing availability.

Two different techniques could be used for monitoring the 4D tracking performance. First, analogue to the known horizontal protection level, we define a longitudinal protection level (LPL) that shall over-bound the 4D trajectory tracking performance in the longitudinal component, or in other words the time domain. For example one approach could be to define the longitudinal protection level as follows.

\[
LPL = \max(LPL_{H0}, LPL_{H1})
\]  

(11)

In this case, three cases are considered, where

- \( LPL_{H0} \) defines the bound for fault free system scenario
- \( LPL_{H1} \) defines the bound for a faulty scenario having a single error in the current ground speed tracking system coming from the navigation or control system.

At every time, the maximum both cases would be used to determine the upper bound for the protection volume.

A completely different approach would be to do a so called “sigma inflation”. The idea here is to have standard variation as a measure of trust for each of the aircraft states and to inflate them according to every error or uncertainty that arises in the system, including ground speed uncertainties. Again, a functional closed loop control with enough margins for countermeasures could deflate the standard variations as a function of the control margin. This research project will try to identify appropriate mathematical functions for this inflation and deflation depending on occurring disturbances, their probability and finally the amount of control margins.

CONCLUSION

Aiming at future 4D trajectory tracking concepts as envisaged by SESAR this paper presented an architecture that could implement a first step towards an integrity monitoring onboard the aircraft for a fully contained 4D trajectory operation for the example of an ETA prediction. Necessary components have been derived and different approaches have been presented to fill the gaps between the contained 2D operation in the current days and the 4D operation with full 3D waypoint plus associated RTAs.

REFERENCES

3. SESAR i4D+CTA Validation Report - Step A Project ID 04.03, May 2012
Information Analysis for a Future Flight Deck Design in the Context of 4D Trajectory Based Operation

Sven Hartmann
Technische Universität Darmstadt
Petersenstraße 30, 64287
Darmstadt
hartmann@fsr.tu-darmstadt.de

ABSTRACT
Due to continued growth of the current air transportation system, in several areas, the demand of the system often exceeds the available capacity of the system, resulting in significant negative consequences [1]. The transformation of the current clearance-based air traffic management (ATM) system into a future Trajectory based Operation (TBO) system is a consequence of the overarching goals defined by the European Sky ATM Research (SESAR) and Next Generation Air Transportation System (NextGen). The achievement of an increase in capacity and efficiency while maintaining safety requires a tough integration of new procedures and technologies as figured out in 4D TBO. This doctoral paper presents an overview on the PhD question of integrating the required information for 4D TBO, enabling the flight crew to work under this full 4D TBO concept. A key point of this approach is to analyze what information is required and how the required information can be integrated into the flight deck using automation. Starting with a current flight deck information analysis this PhD thesis will use expert surveys and real time simulator trials to evaluate the required information and information presentation. This PhD research work is aligned with the SESAR objectives of ATM and Automation and will produce results, which can interlink with other projects performed with the HALA! Network.

Author Keywords
Trajectory based Operation; Flight deck design; SESAR, NextGen

ACM Classification Keywords
Documentation

General Terms
Air Traffic Management; Information

INTRODUCTION
The SESAR program and the NextGen, both deal with the same challenge, that changes in current Air Traffic Control (ATC) technologies and procedures are necessary to facilitate growth in air traffic while maintaining the current safety level. In order to accommodate an increased air traffic demand, SESAR [1, 3] and NextGen [4, 5] describe the transformation process from current clearance based ATM system to Trajectory based ATM. More in detail, both concepts describe the negotiation of a trajectory, including constrains like Controlled Times (CT) between the airspace user and the Air Navigation Service Provider (ANSP) that will be agreed from both and followed by the aircraft. The Term “4D TBO” is used to describe that times are also included in the trajectory to improve its predictability. The task of the cockpit crew is to fly the negotiated trajectory and the function of the ATC is to manage the trajectories [1, 2, 4, 5].

The introduction of the 4D TBO concept calls for new ATM technologies and procedures as the concept of 4D TBO cannot exist without accurate Target Times of Arrivals (TTAs) and trajectory negotiation [3, 4]. This shift from clearance-based operation to strategic trajectory negotiation creates completely new information requirements on the flight deck and on the ground [3].

The research work to be conducted within my thesis in the context of HALA will investigate how the flight crew on the flight deck has to be supported with information in order to be able to manage and fly a 4DT in the context of TBO. The goal of the PhD work is to develop an advanced information presentation demonstrator to present the required information for 4D TBO quickly and in an understandable format, enabling the flight crew to improve the mission performance under the new ATM system environment. In order to analyze the required information and to evaluate the information processing and presentation on the flight deck, this PhD work will first define the working environment.

The objective of this paper is to provide an overview on the PhD activities by describing the working environment, providing the overall methodology of the research work and constitute the current status.

The paper is structured as follows: The next section is dedicated to the changes in the air traffic management (ATM) system, including the two main concepts SESAR and NextGen. Then the methodology planned to use in this PhD research work is described. After that the current PhD
thesis status is given, including tools and methodologies possible to use for the information analysis. At last, a conclusion outlines this position paper and sets out the next phases to be taken during this PhD.

FUTURE AIR TRANSPORTATION SYSTEM

The air transportation system will transform in the next 20 years conducted by the ICAO vision of an integrated and globally interoperable ATM system [2]. The programs, SESAR in Europe and NextGen in the US, are currently the two most significant programs, supporting the modification of the ATM system. The key change of both concepts is the transformation from current clearance based ATM system to a TBO system. To identify the research environment and to define the TBO definition both programs were investigated to specify the most appropriate working concept environment [1, 3, 4, 5, 6].

SESAR

The key concept of SESAR is the implementation of a 4D trajectory based ATM system and the relevant information sharing process between all stakeholders [1]. This ATM Target Concept is centered around a business trajectory (BT) which is representing the users intention. The 4DT is defined by three spatial dimensions, time and all possible constrains containing capacity limitations or weather. The trajectory precision increases during the lifecycle and during flight execution the BT is changing into the Reference Business Trajectory (RBT), shown in figure 1 [1].

![Figure 1. Trajectory Lifecycle [3]](image)

One fundamental tool to implement the concept is a information network for data sharing and Collaborative Decision Making (CDM) [1]. This information sharing and decision-making process between all stakeholders will be supported using a System Wide Information Management (SWIM). The new airspace design will be reflected through the Network Operations Plan (NOP). In this future ATM system the human will act as decision maker and Trajectory manager. The overall SESAR Operational Concept target time is 2020 [3].

NEXTGEN

NextGen identifies eight key capabilities to achieve the transformation of the ATM system and one of these areas is the definition of the aircrafts flight path in four dimensions, the three space coordinate plus target times and appropriate time buffers to take the uncertainty into account [5]. In addition the 4DT is defined with representative waypoints, constrains and time windows over waypoints, shown in figure 2.

![Figure 2. 4D Trajectory with Waypoints (WPs) [5]](image)

During the planning and construction phase of these 4D trajectories, the users preferences were taken into account using a collaborative ATM (C-ATM) system. To share the required information for 4DTs, NextGen defines a network-enabled information access system in real time for all stakeholders including decision making and risk management tools, represented in figure 3 [5, 6].

![Figure 3. NextGen ATM Infrastructure [5]](image)

Through this fundamental change the people involved will get new roles and responsibilities. Especially the flight crew will be more integrated into the ATM process.

PHD CONCEPT ENVIRONMENT

In transformation process of the overall ATM system is fundamentally the same in SESAR and NextGen. Both concepts describe a network based information process sharing the required information for 4D TBO between the stakeholders [6]. The review of both programs clearly identify, that the transformation to TBO is the basic element of SESAR and NextGen [6]. Although the capability to negotiate the trajectory between the involved stakeholders, will be a key element of both concepts [6]. The representation of the aircrafts flight path as 4D trajectory, including all required information, instead of a flight plan, is the necessary information on future flight decks [6]. This key element of 4D TBO will be used in this PhD research work to analyze how 4D TBO concept can be integrated into future flight decks without increasing the human’s workload.
In order to analyze the required information for this 4D TBO concept on the flight deck the SESAR definition of a 4D BT, including Trajectory Change Points, will be used as the TBO environment for my PhD.

**PHD METHODOLOGY**

Differrent Human Machine Interface concepts, such as the Tunnel-in-the-Sky concept [7], have shown that the visualization of 3D trajectories could be integrated into the cockpit to support the flight crew. The overall approach of this PhD work will be to develop a concept to display and comply to the 4D trajectory defined by the SESAR Concept of Operation into the flight deck, in order to meet the users intention and to fly conflict free 4D trajectories with a special focus on the time constraints.

The overall PhD research question is to analyze, how automation can be used to integrate the required information for 4D TBO into the flight deck without increasing the cockpit crews workload. The thesis will be a human centered cockpit system analysis to design and develop a concept, supporting all needs of the cockpit crew under 4D TBO and maintaining the workload. The research work is grouped into three main activities. The first part is to evaluate and analyze the required information for 4D TBO on the flight deck including a hierarchical task analysis, the second part will be a conceptual phase to define the information processing and presentation on the flight deck and the last element will be an experimental scenario using flight simulator trials. The information processing and implementation will use a completely new information presentation instead of integrating additional information into the current flight deck design. The information processing and presentation, including the conceptual phase will be based on the results of the information analysis and expert surveys to support the flight crew workflow and the process of decision-making.

**CURRENT PHD STATUS**

This PhD research work started on the beginning of June 2012. To provide an overview over the PhD status, the tools and methods to analyze the information on the flight deck are described first, followed by a short insight into the analysis and completed with an outlook on the next tasks.

**TOOLS SUPPORTING THE INFORMATION ANALYSIS**

Out of amount of engineering methods and human factors available, the cognitive work analysis (CWA) and the hierarchical task analysis (HTA) are the most popular ones [7]. Both methods can be used to analyze the required information on the flight deck, the HTA represents an analytical approach and the CWA represents the system design framework [8].

**Cognitive work analysis**

The CWA is a very flexible approach to analyze complex systems and it is more a framework than a methodology. The focus of the framework is based on constrains and can be consists of five phases, shown in figure 4 [8].

**Figure 4. Phases of CWA [8]**

The CWA framework can be used as a human factor design method.

**Hierarchical task analysis**

HTA is a systematic methodology for the identification of goals, tasks and provides a framework to classify the goals and tasks with the functional model of the overall system using a detailed task description process. The outputs of this HTA process can be used to inform additional methods. The HTA can be applied to a wide range of complex system through its flexibility [8].

**Comparison**

The CWA framework describes how targets and functions of a system could be achieved sufficiently whereas HTA specifies and describes the goals. The second difference between both methods is the focus, CWA focuses on the constraints and HTA focuses on the system goals [8]. Due to the focus that HTA describes how goals should be and how they can be achieved, HTA will be used through the entire design and development process of this PhD, from information analysis to the operational system demonstrator to describe the overall system.

**FLIGHT DECK INFORMATION ANALYSIS**

The HTA process is first used to review the available information on the flight deck. The scenario used to perform the HTA is based on current operations under ICAO standards, using a normal daytime flight from one destination to another with standard weather conditions. The different operations and tasks are defined using goals and sub-goals. The overall goal of the flight crew is to transport passengers from origin to destination safely, efficient and on schedule. This overall goal is divided into different sub-goals. The description of each goal is based on the required tasks, the operational requirements and the required information. Depending on the actor different tasks and operational requirements are necessary therefore the HTA was done for the pilot flying (PF) and the pilot-non-flying (PNF). As an example figure 5 presents an extract of the PF HTA. This task model is hierarchical and the different goals and sub goals are categorized. In addition a terminology was developed to highlight the kind of task execution to be able to categorize the information.
Based on this HTA results, interviews with experts were done to identify which information they would require for a 4D TBO working environment and for what purpose they would use the information. Based on these information and indications, a questionnaire was developed to do an extensive expert survey. The information processing and presentation, including the conceptual phase will be based on the results of the information analysis and expert survey.

**Conclusion and Outlook**

This PhD research work started at the beginning of June 2012. The analysis of the available information on the flight deck is finished, the questionnaire based on these results developed and the next step will be to analyze and evaluate the results. Parallel the integration and level of automation for the information processing and presentation on the flight deck will be developed. The next step will be a first demonstrator to present the results of the analysis carried out.

**REFERENCES**

3. SESAR, The ATM Target Concept D3, 2007
4. SESAR, ConOps Step 1, Edition 1.0, 2012.
10. Paul Salmon et. al., Hierarchical task analysis vs. cognitive work analysis: comparison of theory, methodology and contribution to system design, 2008
Traffic-Separation & Collision-Avoidance Strategy
To Promote Civil Applications of UAVs

Migliaccio Giovanni
University of Pisa
Via G. Caruso, 8, Pisa, Italy
giovanni.migliaccio.it@gmail.com
+ 39 320 5611694

Mengali Giovanni
University of Pisa
Via G. Caruso, 8, Pisa, Italy
g.mengali@ing.unipi.it
+ 39 050 2217 258

Galatolo Roberto
University of Pisa
Via G. Caruso, 8, Pisa, Italy
r.galatolo@ing.unipi.it
+ 39 050 2217 211

ABSTRACT
Traffic separation and collision avoidance are safety critical requirements to operate aircraft in prescribed environments. UAVs also need to meet these requirements to be integrated in such environments. In this paper a strategy is proposed to develop a system that can provide an airborne capability to autonomously avoid a conflict. This strategy is based on a concept that complies with the ICAO vision of a modern ATM system. Such strategy is also implemented through an algorithm that makes use of a four-dimensional approach in the time-space domain to detect and resolve a conflict using kinematic constraints. Each intruder or obstacle is modeled through a moving ellipsoid which represents a region where another aircraft must avoid to go. The proposed approach is suitable to detect and resolve potential conflicts real-time.

Author Keywords
Traffic Separation; Collision Avoidance; Conflict Detection And Resolution Strategy; Autonomous Airborne Sense And Avoid Systems; Autonomous UAVs.

ACM Classification Keywords

INTRODUCTION
Many investigations have pointed out an increasing interest in using UAVs within civil and commercial markets, with a result that a wide diffusion of their operations is foreseen by 2025. The integration of UAVs in non segregated airspaces presents regulatory and technical issues. In fact, it is known that aircraft shall not be operated in such proximity to other aircraft as to create collision hazards [4] and vigilance shall be maintained so as to see and avoid other aircraft [3].

Manned aircraft comply with these requirements by a pilot on-board that is the ultimate responsible of taking actions. For operating in the same airspaces, UAVs need a way to replace the human capabilities to see and avoid a potential conflict with an equivalent level of safety [1-6].

With regard to an UAV, the aeronautical community refers to the mentioned equivalent capability with the term Sense And Avoid (SAA). The term Sense refers to the capability to acquire information from the environment to get situational awareness. Instead, the term Avoid refers to the capability to assess the risk of conflict and to define actions to avoid it. The aim of these avoid-actions is to maintain separation minima from the surrounding traffic (Separation-Provision) or to implement a so-called last-minute maneuver to avoid an imminent collision (Collision-Avoidance).

Over the years, different approaches have been proposed to automate the process of maintaining traffic separation and avoiding collisions; interesting reviews and several methods can be found in the literature [7-9].

In this paper a conflict detection and resolution strategy is proposed. It is based on the concept of conflict management envisioned by ICAO [5]; kinematic algorithms perform the actions of conflict detection and solution formulation. The implementation of the actions of execution and monitoring has also to be performed through suitable algorithms.

The proposed approach is also innovative. It makes use of a generic moving ellipsoid to represent an intruder. Also, this approach is able to manage the differences in the horizontal and vertical minima of separation, detect conflict situations, provide tree-dimensional actions of conflict resolution over the time, and operate in real-time.

This paper is organized as follow. The so-called Sense And Avoid problem is first introduced. The concept of Conflict Management is presented according to ICAO vision. Some important results from requirements and functional analysis are illustrated. Such results enable one to identify how an Avoid Function should work and how such function could be developed. Then, a solution is proposed to automate the functionalities of separation and collision-avoidance and a conceptual demonstration of the conflict-avoidance strategy is also provided through the results of a set of simulations.
**SENSE AND AVOID**

**Introduction to the general Problem**

The Sense And Avoid problem involves the development of technical solutions to enable UAVs to meet the requirement for traffic separation and collision avoidance as required to operate in prescribed environments.

Worldwide agencies, industries and research centers have participated to research programs with the final objective to develop a so-called Sense And Avoid Standard. This should represent a method to demonstrate UAVs compliance with the regulations to operate in some environments.

To achieve a solution, several factors should be considered, such as airspace classification, regulations and procedures, environment conditions, sensors and aircraft performances, and also the compatibility with other systems.

**A concept for Conflict Management**

To begin, it is very important to understand how a situation of conflict should be managed within a modern system of Air Traffic Management (ATM).

The current ATM system is already organized to limit the risk of collision under a target level of safety. However, the future needs, along with the integration of some UAVs in prescribed environments, require to define a modern ATM system. The vision of ICAO foresees a future ATM system formed by a number of concept components that have to be integrated [5]. Among these, the Conflict Management is the component of primary interest for what concerns the problem of mid-air collision avoidance.

According to ICAO, the Conflict Management should limit the risk of collision between aircraft through three different layers. These layers are the Strategic Conflict Management, the Separation Provision, and the Collision Avoidance.

A Strategic Conflict Management should ensure adequate minima of separation between aircraft. This layer should also limit the application of the so-called tactical changes to the aircraft flight trajectories.

If a failure occurs in the strategic process, and separation cannot be ensured, tactical actions should take place. These Tactical Actions are the so-called Tactical Separation (i.e. a first attempt to maintain or restore the adequate minima of separation) and the Collision Avoidance (i.e. a last resort to avoid an imminent collision).

Figure 1 provides an illustration of this concept. It is based on the airspace structure, the traffic synchronization, etc., for ensuring a strategic separation. Instead, an ATC, a pilot and an autonomous Sense And Avoid System (for an UAV) are responsible for the application of the tactical actions.

The Tactical Conflict Management should comply with the indications of ICAO. Then, it should comprise a number of key-phases, see the Figure 2: Detect Conflict; Formulate Solution; Implement Solution; Monitor Execution.

**Figure 1. Conflict Management Layers – A possibility to integrate Manned and Unmanned traffics**

For what concerns the application of the tactical actions to UAVs, a fundamental hierarchy exists. The first responsible to take actions to avoid a conflict is the ATC. The ATC should communicate to the PIC, which is responsible to implement ATC indications. The PIC is also responsible to take actions to avoid a potential conflict, even if the ATC doesn’t provide indications. Finally, a last resort to avoid a mid-air collision is the so-called airborne Sense And Avoid System (SAAS). A SAAS should provide the capability to an UAV to perform autonomous and safe operations.

In fact, a SAAS should provide the navigation capability to operate autonomously also within a surrounding traffic; this should ensure the adequate level of safety to operate UAVs in prescribed environments. In fact, consider for example a case in which the PIC is not able to communicate with the corresponding UAV or the communication delays are such that to prejudice the safety: in situations like these, a SAAS should replace the capabilities of a human pilot to see and avoid a surrounding traffic, through the implementation of actions of conflict avoidance. For what concerns this paper, the focus is only on the Avoid Function.
AVOID FUNCTION

Requirements and Architecture

This section presents the results of an analysis conducted to identify how an Avoid Function should behave and which should be its sub-functions, interfaces and constraints; see the Figure 3. A set of important requirements to develop an Avoid Function have also been identified [1-6].

General statements require that UAVs carry functionalities for flight, navigation and communication in a similar way to manned aircraft. Furthermore, UAVs operations should not increase the risk to the other airspace users. Also, ATM procedures and the services to an UAV should mirror those applicable to manned aircraft.

Technical statements require that a Separation Function has to separate UAVs from other traffic by a minimum distance of 0.5 nm in the horizontal plane or 500 ft in the vertical plane. Also, a Collision Avoidance Function has to ensure other aircraft are avoided by a minimum distance of 500 ft in the horizontal plane and 350 ft in the vertical plane.

In addition, the field of regard to be overseen should extend ±110° in the horizontal plane and ±15° in the vertical plane; instead, a reference distance to detect a conflict is assumed here to be of the order of 5 nm.

Other requirements state that a Separation Function should provide the capabilities to maneuver while minimizing the deviations from a datum flight path and complying with the rules-of-the-air. Instead, a Collision Avoidance Function should provide the capability to autonomously maneuver while minimizing the time to reach a condition of safety. In any case, a maneuver has to comply with the performances of an aircraft. Finally, a capability to autonomously return-to-course should also be made operative after a conflict has been avoided.

PROPOSED SOLUTION

Avoid Strategy

An new conflict detection and resolution approach is now proposed to automate the conflict avoidance capability.

Begin to consider the Figure 4: B is the aircraft that has to maneuver to avoid a conflict, while A is the intruder to be avoided. The ellipsoid around A represents the separation region. A conflict is considered to occur if and only if B is predicted to enter the ellipsoid of A. To perform a conflict, B is considered to perform a maneuver to change its original velocity and to reach a new constant velocity.

A procedure to determine the timing and intensity of the above mentioned velocity change is now illustrated.

Begin to consider the problem from a kinematics point of view. Suppose the objective is to find the deviation to set an autopilot for implementing a suitable maneuver. To achieve this, consider a broken-line to schematize the trajectory of B, see the Figure 5: the point P0 is the initial position of B; the point P1 is the position of B after a time-delay; the point P2 is a virtual position of B and is only used to calculate the velocity deviation to avoid a conflict; finally, the point P3 is where to check the effectiveness of the strategy.

Figure 3. Functional Architecture of a Sense And Avoid System – Sense Function & Avoid Function

Figure 4. Scenario Schematization – B is the reference Aircraft and A is the Intruder to be avoided

Figure 5. Avoid Strategy – Detect Conflict, Formulate Solution, Implement Solution, Monitor Execution
The strategy can be summarized through a number of key actions to be executed sequentially. The first phase starts at P0 and is performed through a dedicated algorithm to detect a conflict (DETECT). The second phase is a schematization of the situation and the trajectory of B through a broken-line (SCHEMATIZE). This schematization is used to calculate the velocity-deviation of B in a point P2 (CALCULATE). A continuous trajectory is then simulated to closely follow the broken-line and to implement the deviation (SIMULATE). A final check is conducted in a point P3 to ensure that the strategy can effectively avoid the conflict (CHECK).

When the solution is declared effective to avoid a conflict it can be implemented by an autopilot (EXECUTE). A final action of monitoring the evolution of the situation can be helpful to manage unforeseen or unpredictable change to the aircraft flight trajectories (MONITORING).

A previous work of the authors [10] discusses an analytical solution that solves the problem of avoiding a conflict. The work utilizes the strategy herein illustrated and provides an algorithm to evaluate the mentioned path deviation.

Simulation of the Strategy
Figures 6-7-8 illustrate how the proposed strategy (through its algorithms) could generate different conflict avoidance trajectories; the simulations start from a common condition of conflict and show different potential solutions.

CONCLUSION
The proposed approach provides a new general method to automate the functionalities of self-separation and collision-avoidance for the development of an Avoid Function.

The approach is innovative: it utilizes a generic ellipsoid to represent a moving intruder; it is based on the concept of conflict management envisioned by ICAO; it makes use of generic dedicated algorithms to detect a potential conflict; the formulation of a solution comes from the avoid-strategy illustrated before; the corresponding avoidance algorithms are able to manage differences in the horizontal and vertical minima of separation, provide tree-dimensional actions of conflict resolution and operate in real-time.

The proposed approach represents a first step in the design of control laws which will provide the required navigation capabilities to integrate UAVs in prescribed environments.

REFERENCES
10. Migliaccio, G., Mengali, G., and Galatolo, R., Conflict Detection And Resolution Algorithms for UAVs Collision Avoidance, paper under review, please ask to the author at giovanni.migliaccio.it@gmail.com.
User-Interface Design for Highly Automated Systems
A structured approach

Giuseppe Frau
Deep Blue srl & University of Rome 3
Piazza Buenos Aires, 20 00198 Rome - Italy
giuseppe.frau@dblue.it

ABSTRACT
Guidelines and theoretical frameworks have been provided by studies concerning the design of interfaces in automated systems. Such studies propose how to define the appropriate level of automation or to assign the proper roles and tasks to human and automation. Nevertheless, failures can still occur in the real interactive actions due to poorly designed interfaces. It is necessary to identify how design intents influence the actual interaction and to map them to the most suitable user-interfaces and interaction techniques. This work puts the basis for a research aimed at building a framework for interface design in future ATM systems. A brief literature review is presented to support the need of this kind of approach. Afterwards the structure of the research project is outlined along with the conclusion in which I state the expectations for the results.

Author Keywords
HCI; User-Interface Design; Human Automation Interaction; Design; ATM; Automation;

ACM Classification Keywords
Design; HCI; Human Factors; Interaction Styles; Input devices and strategies;

INTRODUCTION
Overcoming the traditional human-automation issues is a challenge that has been addressed by many research works. Design choices like picking the right level of automation (or deciding for an adaptive automation), or correctly distributing roles and tasks etc. are very important decisions that deserve robust foundations in order to “build the system right”. However, what do we do after we have made those choices? Choosing the right level of automation doesn’t guarantee us a sound interaction. Neither the right task allocation does. Failures can happen in the real interactive actions due to poorly designed interfaces.

To some extent interface design in safety critical systems has been supported by a UCD (User Centered Design) approach and EID (Ecological Interface Design) – see Literature Review section. As the review will point out, those complementary approaches may provide a better support if the automation dimension was taken into consideration. Moreover a guidance for the usage and benefit of the new interaction techniques is by now a very important tool for the designers.

Figure 1 - Qualitative representation of Hardware, Interface Design and ATM progresses over time

The above picture shows a qualitative trend of the progresses made by the HW technology (blue), Interface Design (red) and ATM domain (green). As Moore predicted, we had a linear HW progress over time. Interface design instead has been characterized by constant periods followed by rapid changes. The last one can be identified in the introduction of the Natural User Interfaces. The ATM field, on the other hand, is going to introduce a big amount of innovation thanks to the SESAR program. The contemporary rise of those trends represents new challenges and opportunities.

LITERATURE REVIEW
Human Automation Interaction research field has made a wide effort to produce theoretical frameworks and guidelines for the design of highly automated systems. Researchers and designers have been trying to use these results in order to develop effective proof of the actual correctness of the underlying theories.

This brief literature review will go through the main theoretical findings of the automation field and through some of the main practical realizations trying to highlight the gaps between these two words.
We can identify several dimensions of automation. In [6] for example, O’Hara and Higgings, identified 6 dimensions (applied to the nuclear-power plant cases): levels, functions, processes, modes, flexibility and reliability. However not all of them are applicable in different fields.

From the level dimension of automation point of view, the work of Parasuraman [7] it is wildly used as the framework of reference at least in the ATM world. As the authors state, their model does not claim to provide comprehensive design principles but it can be used in the early stages of the design to get a support about what types and levels of automation implement and what issues the designers are more likely to face as consequences of their choices. Moreover several limitations -in some extent anticipated by the authors- of the practical applicability of this works have been found recently [11].

The same seminal work of Parasuraman [7] can also represent the base for the function dimension of automation. The classes of functions to which automation can be applied are identified as information acquisition, information analysis, decision and action selection and action implementation. The automated functions are mapped with the cognitive functions of the human being.

The automation mode dimension would deserve a dedicated literature review itself, as there is no commonly accepted precise definition of the word “mode”. Being over simplistic we can use the O’Hara statements “mode involve performing the same functions in different ways. They provide capacity for a system to do different tasks or to accomplish the same task using different strategies under changing conditions”. Vicente and Jamieson [4] provide a sufficient overview of the mode challenges as mode transition design and mode awareness and they offer a set of principles for the automation design facing those issues.

The flexibility dimension regards the capacity of automation to change the responsibility and the allocation of a certain task or activity. The flexibility can be characterized in

- the change of the levels of automation for a given task depending on some context parameters or some operator selections

- the change of the task allocation of a subset of tasks: for example a task belonging to the action implementation class can be dynamically assigned to the operator or to the system on the basis of the context information or, again, some operator selections.

Lastly, the reliability dimension regards the degree of correctness in the results or behaviour of the automated system. Studies [2] and [9] report effects on performance and workload due to low automation reliability and they give general guidelines for mitigating those effects in case full reliability cannot be guaranteed.

There’s a considerable amount of works that tries to identify and quantify how the operator is affected by the several dimensions of automation. Those works also define the guidelines to mitigate bad effects saying WHAT designers should do but there’s a lack of information about HOW those guidelines should be implemented. Despite this, researches and designers did realize -of course- innovative interfaces for ATM leveraging on the most recent technologies. Hurter in [3] with Strip”Tic realized a novel system for ATC that mixes augmented paper and digital pen. In some ways he overcame the problem of not having a physical artefact originally highlighted by [5] and he got positive feedbacks from controllers both for the augmented paper functionality and for the selection technique using the digital pen. Pitman and Cummings developed MAV-VUE (Micro Aerial Vehicle Exploration of an Unknown Environment) [8] allowing operators with minimal training to effectively fly a MAV to explore their environment using a mobile interface for a hand-held device.

Sarter [10] reviews the guidelines for multimodal information presentation exploring possible benefits (and challenges too like mode synchronization) deriving from the usage of rarely used channels like touch and olfaction. Sarter also highlights the limitations of certain guidelines like the fact that “they do not appear to be specific for the design of multimodal interfaces but rather repeat earlier general design guidelines”. More specific guidelines and recommendations for ATM interfaces need to be obtained following the automotive examples in [1] in which very detailed solution have been tested in order to understand the most suitable design for the task.

PROJECT DESCRIPTION

The project is focused on the Air Traffic Control separation assurance activity and is structured in two main parts. The first one aims to develop the core of a simple Air Traffic Control simulator that will be used in the second phase. The core/engine will have the ability to instantiate as many aircraft as specified in a given scenario and play them at the same time allowing the control with external actions. Each aircraft, on the other side, will be constituted by

- A main “playback” functionality that allows it to follow the planned scenario
- An action interface towards the external to allow the control
- An autonomous functionality
- A module for the performance measurements
The automation part of the aircraft will be represented by a self-separation algorithm. As it is not the goal of this project to assess what is the best algorithm for a self-separated scenario, a fairly simple one will be chosen from the existing literature and implemented in the core simulator in order to support several levels of automation.

The second part of the project is the development of the user interfaces for the separation of the aircraft in the simulator. The development will be carried out independently and iteratively for each selected level of automation.

Each resulting HMI will be validated with trained and possibly real users. This strategy will allow the identification of detailed needs/requirements for each analyzed level of automation. A similar approach can be applied for all the dimensions regarding human automation (i.e. levels, functions, adaptivity etc).

Method
The main objective of this research is to study how the presence of automation influences the development of the user interfaces. Capturing this aspect is essential to answer the fundamental question which is “how do we develop user interfaces suitable for highly automated environments?” The chosen method to contribute to this answer is to elect the automation level as independent variable and then to analyze the resulting HMIs (thus considering the HMI as an independent variable). Small-scale human in the loop simulations and expert judgment will be used to validate each HMI development. At the end, results about the final HMIs will be summarized trying to recognize new interface design patterns, guidelines and methods mapped with the original levels of automation for which the interface has been generated.

A further contribution of this project will be the explorative investigation of suitable interaction techniques enabled by the most recent input devices. Since the growing importance assumed in the last year, a free-hand gesture control device (see www.leapmotion.com) will be used to design a natural user interface for controlling the system.

Forthcoming Activities
In parallel with the development of the simulator engine, a documentary research about the future tasks of the ACTOs and the associated levels of automation is needed in order to obtain realistic scenarios to implement. Furthermore, one or more methods to analyze the followed development process and the final HMIs and obtain generalized results are required.

CONCLUSION
The proposed research project aims to have a two-fold contribution in the topic of user interface design for automated systems. From one side it will try to give more detailed guidance about the development of the interfaces in order to overcome/mitigate the well-known issues introduced by the automation. On the other hand, this will be done with the support of the most recent interaction techniques that are just approaching the consumer world and that are more likely to be the future technologies we are going to use.

REFERENCES


A Multi-ModelS Based Approach for the Analysis and Design of Usable and Resilient Partly Autonomous Interactive Systems

Martina Ragosta

Abstract

In this paper I present the progress made during the second year of my PhD which is developed in collaboration with the ICS team of the IRIT lab in Toulouse and Deep Blue s.r.l. consulting & research in Roma. This PhD is funded by HALA! Research Network under the joint supervision of Prof. Philippe Palanque and Alberto Pasquini. It is related to System Performances under Automation Degradation – SPAD project financed by EUROCONTROL.

Author Keywords

Models based approach; automation; partly autonomous interactive systems.

ACM Classification Keywords

D.2.2 [Software] Design Tools and Techniques - Computer-aided software engineering (CASE), H.5.2 [Information Interfaces and Presentation]: User Interfaces - Interaction styles

INTRODUCTION

Until the 90’s the driving direction of research and engineering in aviation, space and in computer science was to design systems aiming to automate as much functions as possible. This direction changed due to many studies [10] that have demonstrated that fully automated systems are out of the grasp of current technologies. On one side, it should be taken into account that migrating functions [1] from the operator to the system might have impact on operations both in terms of safety and usability, on the other side that the operator should be able to cope with failures in case of automation degradation [5].

This paper is structured as follows. Next section describes the objective of this PhD thesis. Section 2 proposes a brief discussion about task modelling techniques and presents the one which has been adopted. Section 3 is dedicated to explain the approach that we plan to use throughout the PhD. This is exemplified on a case study in Section 4. Section 5 presents the future steps. The last Section provides a list of selected publications.

OBJECTIVE

The objective of this PhD thesis is to provide an unifying view on the analysis and the design of partly autonomous interactive systems taking into account both the usability and also the resilience. The past research work on automation for these systems can be divided according to the 3 different perspectives: 1) the designing perspective focuses on how to engineer the systems offering automation features, 2) the evaluation perspective focuses on how to assess the operational aspects of automation including performance impact of automation on operations, and 3) the human perspective focuses on how to understand the role of the operators who deal with a system. This PhD targets at offering an integrated view on these 3 different perspectives by providing notations and related tools able to support each phase of development process from the design to the valuation. The selected models and the adopted approach (please refer to Section 3) can guarantee an accurate description of automation of the partly autonomous interactive systems, of the operators’ task while interaction of the system both when it properly works and also in case of automation degradation, and of the features of this interaction in terms of usability and resilience.

Such automation design activity requires at first a complete understanding of operators’ tasks and a task models can be a good way of representing such information. There are many different types of task modelling techniques, supported or not by a tool, that can support this activity. The next Section offers a brief discussion about task modelling techniques and presents the one which has been adopted.

TASK MODELLING TECHNIQUES

The most recent contributions [7, 12] confirm that there is a clear interest from the scientific community for task modelling techniques and their notations. Previous works in
this field have mainly targeted at describing the operators’ tasks as a sequence of events which have been accomplished to achieve a goal such as HTA [8], GTA [11], UAN [3] and AMBOSS [2]. This procedural description does not allow to represent all the information and objects needed to the operators who interact with a partly autonomous interactive systems i.e. information required for performing the tasks and information produced while performing the tasks, explicit objects, knowledge and its different types representation. In order to benefit from a detailed representation of operators’ tasks, in this PhD thesis has been adopted as task modelling technique the Human-centered Assessment and Modelling to Support Task Engineering for Resilient Systems (HAMSTERS) in its latest version.

HAMSTERS
HAMSTERS 1 is a tool-supported graphical task modelling notation aiming at representing human activities in a hierarchical and ordered way. Goals can be decomposed into sub-goals, which can in turn be decomposed into activities, and output of this decomposition is a graphical tree of nodes. Nodes can be tasks or temporal operators.

Tasks can be of several types and contain information including a name and information details. Temporal operators are used to represent temporal relationships between sub-goals and between activities (as illustrated in Figure 1 and Figure 2 in the next Section).

Tasks can also be tagged by temporal properties to indicate whether or not they are iterative, optional or both [7].

One main element of this notation is the subroutine. A subroutine is a group of activities that a user performs several times possibly in different contexts and which might exhibit different types of information flows. A subroutine can be represented as a task model and a task model can use a subroutine to refer to a set of activities. This element of notation enables the distribution of large amount of tasks across different task models and factorization of the number of tasks. HAMSTERS also provides support for representing how particular objects (data, information...) are related to particular tasks. These relationships (input, output or both) between objects and tasks that can be expressed with HAMSTERS notation. As for an object, it is possible to represent the relationships between a task and information which is represented in a non-ambiguous way.

This notation is able to describe the complexity of the operators’ tasks and it can support the identification of which tasks in the task model are good candidate for automation and which ones should remain operator driven [6]. However, dealing with a partly autonomous interactive system requires to represent not only the operators’ tasks but also the system behaviour and the interactions between this two components.

A MULTI-MODELS BASED APPROACH
To be able to represent each component that is involved in this interaction and the interaction by itself, a multi-models based approach has been adopted. The first two selected models (HAMSTERS and ICO&PetShop) can provide a synergistic integration of the tasks and system models and represent each component at different levels of granularity.

In order to analyze the interactions and all the components as a whole, the Functional Resonance Analysis Method has been chosen. This approach is also adopted in SPAD project through a federation of models [5]. This Section provides a brief presentation of the other two adopted models, ICO&PetShop [9] and the Functional Resonance Analysis Method (FRAM) [4].

ICO&PetShop
ICO is a formal description technique dedicated to the modeling of interactive applications. This formalism makes it possible the entire interactive application including both the behavioral aspects (states and state changes) and the interaction aspects (events triggered on the user interface and graphical rendering). It is based on objects Petri nets. An example of the behavioral description in ICOs is given in Figure 1. PetShop 2 is the CASE tool associated CASE to the ICO formalism. It allows editing models and their execution. The models have been edited using PetShop.

Thanks to this technique and its tool, it is possible to describe the complexity of the interactive system.

FRAM
It is a safety management method aiming to support both accident investigation and risk assessment processes. It is based on four principles related to complex socio-technical systems structure and dynamic. It describes a failure as a resonance of the normal variability of functions which are characterized by six basic parameters (Input, Output, Preconditions, Resources, Time, and Control). An analysis using FRAM comprises the following five steps: 1) define the purpose of modelling and describe the situation being analyzed, 2) identify the essential functions that make up the event characterizing each by six basic aspects, 3) characterize the actual/potential variability of functions, 4) define functional resonance based on potential/actual dependencies (couplings) among functions, 5) propose ways to monitor and dampen performance variability (indicators, barriers, design/modification, etc.).

This PhD thesis adopts a color code to make a distinction between the different functions (i.e. orange for the functions which are carried out by humans, blue for the technological and violet for the interactive ones). Additionally, some

1http://www.irit.fr/recherches/ICS/softwares/hamsters/index.html

2http://www.irit.fr/recherches/ICS/softwares/petshop/
functions can themselves be refined in other sub-functions according to the required level of detail. This refinement is illustrated in Figure 4 where the “Checking weather conditions” function is refined into: “Read ND information”, “Analyze current weather map”... This refinement provides support for the representation of a larger number of functions while keeping the model representation understandable.

This method can be the appropriate glue to connect the two different views provided by the previous models.

**ILLUSTRATIVE CASE STUDY**

Due space constraints we can briefly illustrate a case study, the Weather Radar, to exemplify the approach.

**The Weather Radar (WXR)**

The WXR is an application currently deployed in many cockpits of commercial aircrafts. It provides support to pilot’s activities by increasing their awareness of meteorological phenomena (Figure 1 shows the HAMSTERS model related to this activity) during the flight journey, allowing them to determine if they may have to request for a trajectory change (illustrated in Figure 2), in order to avoid storms or precipitations for example. The WXR is implemented in the FCU and its control panel provides two functionalities to the crew members. The first one is dedicated to the mode selection of weather radar and provides information about status of the radar, in order to ensure that the WXR can be set up correctly. The second functionality is dedicated to the adjustment of the WXR orientation, named Tilt angle, that can be done in an automatic way or manually.

![Figure 1. “Keep awareness of weather situation” includes the sub-goal “Checking weather conditions” in the red box](image)

While, “Keep awareness of weather situation” presents all abstract tasks and several subroutines meaning that it can be redefined in deeper way, “Change heading” shows different high-level task types and object related to them.

![Figure 2. Pilot goal “Change heading”](image)

Also, in Figure 2 there are system tasks (in blue box) which can be modeled using ICO&PetShop notation such as the instrumentation to manage heading which is also implemented in the FCU. In particular, it is composed of two controls, 1) Heading Selection to select by the knob the heading value (Figure 3 shows the corresponding ICO model) and 2) the Horizontal Situation Indicator (HSI) which indicates the current heading.

As illustrated by Figure 3, there are four different events (“incrementHeading”, “decrementHeading”, “engage” and “disengage”) handled by dedicated transitions (named by adding _T* to the name of the event).

![Figure 3. Behavior of the heading selection](image)

By the use of HAMSTERS notation is possible estimate the complexity of operators’ tasks while by the use of ICO&PetShop of the interactive system.

As suggested by the approach, after having created the task and the system model, it is possible to adopt FRAM making different scenarios in order to analyze all the components and their interactions as a whole.

As illustrated in Figure 4, the main function “Checking weather conditions”, which corresponding to the sub-goal in HAMSTERS model, includes several sub-functions of different types (human, technological or interactive). These
are connected via Input-Output arrows which can change according to the scenario.

Figure 4. “Checking weather condition” function consists of several sub-functions

A failure or an automation degradation in a function can affect all the connected ones exhibiting different levels of variability, and consequently they can show resonance (i.e. according to time and to resources if they are shared between the functions). By the suggested approach, we can quantify and analyze these aspects.

FUTURE WORK
The next step will be to assess and evaluate the approach by a wider case study (AMAN and/or RPAS). For each case study we can create several scenarios which can contribute to improve the existing notations and better analyze, understand and faced both with the task and system complexity and also with automation degradation.

SELECTED PUBLICATIONS


REFERENCES

173
An Approach for Aircraft Turnaround Decision Making to Minimize Outbound Delay

Thomas Kunze, Michael Schultz, and Hartmut Fricke
Technische Universität Dresden
Hettnerstr. 1-3, 01062 Dresden
{kunze, schultz, fricke}@ifl.tu-dresden.de

ABSTRACT
While the current turnaround handling shows potential for prediction and reliability improvement, the turnaround management approach of the Department of Air Traffic Technology and Logistics at TU Dresden describes a scientific foundation using a stochastic approach for process description and delay modelling. Based on recent air traffic network and delay analysis, new delay input data could be derived for European airports. In a first step to integrate open and closed-loop process control for higher automation levels in turnaround management, the subprocesses of aircraft cleaning and boarding have been modelled and implemented, showing great potential to minimize aircraft ground time in case of disturbances.

Author Keywords
Airport operations, turnaround operations, delay modelling, stochastic modelling

INTRODUCTION
The aircraft turnaround, connecting two flight legs of an aircraft, has been identified as a crucial process for keeping up to tight schedules and economic productivity [1-3]. This does not only apply to airlines but also to ground handling companies and airport operators, all of them trying to maximize the utilization of their respective resources. Unlike the en-route segment of a flight, the turnaround is more complex in terms of involved parties and given restrictions due to technical, legal and operational aspects. Close process dependencies and restrictions also make this “flight” segment vulnerable to process uncertainties, disruptions and disturbances, which will be carried onto the next flight leg and therefore affect the whole air traffic system. This was also recognized by the future ATM research programs, SESAR in Europe and NextGen in the USA, by integrating turnaround operations into the Shared Business 4D-Trajectory (SBT) of a flight. Consequently, the aircraft trajectory on ground does not change spatially but advances in time. The Airport Collaborative Decision Making (A-CDM) initiative shall enable a further increase of predictability in ground operations by introducing multiple milestones for a flight and sharing information between all involved participants of the turnaround.

The aim of our research is to provide airline or ground handlers with an optimal decision point for interventions (using control theory approaches) in case of deviations from the actual planned turnaround. While airlines keep record of process performance and adjust the planning accordingly on a regular base, expert interviews showed that in case of disturbances, decision making is mostly based on the experience of ground handling or airline company’s staff rather than a standardized process strategy leading to the propagated target times. Our proposed stochastic model allows the transition from today’s commonly used buffer strategies to automated environments by using intelligent prediction and controlling strategies.

Previously analysed field data indicates that the punctuality of an arriving aircraft has significant influence on turnaround operations and the turnaround time. A dynamic buffer strategy has been found, allowing airlines to partly absorb deviations from planned turnaround times of disruptive events. However, no strategic systematic pattern for the implementation of buffers was found, showing that efficiency relies again on operator experience and information availability at the time the disturbance occurs [4].

Additional influences on the turnaround time have been investigated in [5]. Different airport categories have been found to cause variations in the process time. Also, the varying skill level of local staff due to different training methods and expertise is contributing to distinct process characteristics. Several other studies focused on the stochastic approach aiming to describe detailed turnaround sub-process characteristics such as boarding or cleaning [6, 7].

MODEL DESCRIPTION
Based on our previous studies, it became evident that the turnaround process and its sub-processes show a stochastic behaviour rather than the deterministic one they are planned against. As of this reason, a stochastic approach has been developed to model ground operations.
For each sub-process, characteristic parameters describing the shape of the progress over time have been identified and classified based on previously gathered field data. Using this empirical data, each class of parameters has been fitted to a corresponding stochastic distribution function describing the process duration and its start time relative to the on-block time of the aircraft. Finally, the turnaround time is determined by combining all sub-processes using a Monte-Carlo simulation, resulting in a distribution of turnaround times. By applying different confidence intervals, specific times for the later application of control loop theory can be derived.

As previously mentioned, a dynamic buffer strategy between sequentially running processes has been found in turnaround operations to absorb up to a third of the occurring inbound delay. Based on the work done in [4], the process behaviour has been adapted and integrated into this turnaround model by using different process start-time distributions for each delay category.

For each sub-process, characteristic parameters describing the shape of the progress over time have been identified and classified based on previously gathered field data. Using this empirical data, each class of parameters has been fitted to a corresponding stochastic distribution function describing the process duration and its start time relative to the on-block time of the aircraft. Finally, the turnaround time is determined by combining all sub-processes using a Monte-Carlo simulation, resulting in a distribution of turnaround times. By applying different confidence intervals, specific times for the later application of control loop theory can be derived.

As previously mentioned, a dynamic buffer strategy between sequentially running processes has been found in turnaround operations to absorb up to a third of the occurring inbound delay. Based on the work done in [4], the process behaviour has been adapted and integrated into this turnaround model by using different process start-time distributions for each delay category.

![Figure 1: Used approach for transformation of empirical single flight data to universal distribution](image1.png)

**Figure 1: Used approach for transformation of empirical single flight data to universal distribution**

The lower graph shows the actual arrival delay measured by the difference between scheduled time of arrival (STA) and

![Figure 2: Process interactions and buffer time](image2.png)

**Figure 2: Process interactions and buffer time**

The right side of the figure shows the remaining distribution of buffer times. It can be observed that the process interactions increase significantly while in parallel the mean value and standard deviation of the time buffer decrease in case of a rising inbound delay. The turnaround model therefore requires the inbound delay as an input parameter. It can be derived from real flight plan data by comparing actual in-block time (AIBT) and scheduled in-block time (SIBT). However, as this data is not always available, a disturbance model has been developed allowing the application of the turnaround model regarding airports missing this input data. The initial model described in [6] uses a data analysis of arrival delay at American airports and derived adequately fitted distributions for different airport sizes which would then be used as an input for the turnaround model. As the American air traffic network is less slot-restricted (with exception of the metropole area), the portability of the found distributions is not guaranteed. Therefore the disturbance model has been refined with recently acquired delay data for major European airports. An example set of the delay distribution for Munich airport (MUC/EDDM) is pictured in Figure 3: Enroute delay compensation and arrival delay for Munich airport

![Figure 3: Enroute delay compensation and arrival delay for Munich airport](image3.png)

**Figure 3: Enroute delay compensation and arrival delay for Munich airport**

The lower graph shows the actual arrival delay measured by the difference between scheduled time of arrival (STA) and
actual time of arrival (ATA). The upper one shows the delay compensation (gained or lost delay) during the actual flight, measured by arrival delay at the destination and the departure delay at the origin airport. A negative value in delay compensation therefore implies an en-route delay occurred for this specific flight. The approach of modelling the delay distribution based on two different functions, one for positive delay and one for early arrivals, as presented in [6] can still be applied to European airports as the lower graph in Figure 3 suggests. A further contributor to deviations from a scheduled in-block time can be found in the taxi process. However, it is highly depending on the airport and taxiway layout, which is currently not the focus of our research. The data provided by the EUROCONTROL Central Office of Delay Analysis is therefore used to describe the taxi time as a Gaussian distribution based on the given mean taxi time and standard deviation for the respective airport [9].

Our research currently focuses on the intra-process disturbances. For each turnaround sub-process, a detailed chain of events in staff, equipment, required information and corresponding time stamps is elaborated. This process map and description allows the identification of points where disruptive events can occur as well as a general understanding of how the sub-process reacts on disruptions in terms of sensitivity.

The first assessed process has been the aircraft boarding process, as it is always the last element of the turnaround and therefore is always part of the critical path. Several boarding strategies and procedures have been investigated on how they influence the progression of the boarding process. A detailed description of the analysis can be found in [11]. Figure 4 shows the expected boarding time and progress of different procedures for an Airbus A320. The 1-door random boarding, using only one door and passengers enter the plane in any order, is used as baseline procedure. It has been found that the shape of the progress function is identical for every strategy and procedure, it only scales with time.

The microscopic approach for the cleaning process models these four steps. The process times for each of these steps are listed in Table 1 below. To derive the duration of the sub step, we assume a Gaussian distribution with the parameter \( \mu \) and \( \sigma \). Each single sub-step has the same characteristics, so the assumption of the Normal distribution results in: \[ \mu_{\text{step}} = 0.25 \mu_{\text{process}} \text{ and } \sigma_{\text{step}} = 0.5 \sigma_{\text{process}}. \]

Table 1: Process times for cleaning process

<table>
<thead>
<tr>
<th>Process</th>
<th>Duration [s]</th>
<th>Standard Deviation [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seat cleaning</td>
<td>4.1 (per seat)</td>
<td>1.4</td>
</tr>
<tr>
<td>Lavatory cleaning</td>
<td>115</td>
<td>17</td>
</tr>
<tr>
<td>Galley cleaning</td>
<td>150</td>
<td>45</td>
</tr>
<tr>
<td>Vacuum</td>
<td>120</td>
<td>36</td>
</tr>
</tbody>
</table>

To allow efficient decision making, the aircraft progress needs to be monitored before it arrives at the destination airport. This tactical component may enable different decisions for airlines and stakeholders, e.g. if shorter turnaround is required, a remote parking position allowing two door boarding procedures can be arranged by the airport, while this may not be possible after the aircraft has landed. To define appropriate control strategies, three different target functions are currently looked into:

(a) Delay minimization

The first obvious target function is the minimization of inbound delay. Up to one third of the delay can be absorbed by the existing buffers between processes as described before. To compensate the remaining delay, stakeholders have to deploy additional resources in terms of staff or equipment, also adding additional cost to the turnaround operation. The final cost (1) can be determined by the sum of resource cost (2) and delay cost (3) under the constraint to minimize the delay cost implying the use of additional resources.

\[
F = c_R + c_D \tag{1}
\]

\[
c_R = c_{R,1} \cdot t_{R,1} + c_{R,2} \cdot t_{R,2} + \cdots + c_{R,n} \cdot t_{R,n} \tag{2}
\]

\[
\min c_D = c \cdot (t_{\text{remaining delay}})^a \tag{3}
\]

(b) Cost minimization

While the minimization of delay at a single airport may be reasonable with regard to the ATM network, it comes with a certain cost for airlines and/or ground handlers, who both need to work cost effective. Therefore, the second target approach is to minimize

Figure 4: Boarding duration and progress shape using different procedures

A similar microscopic analysis has been conducted for the cleaning process. The process can be split into four main steps (remove rubbish, clean, restock items and rearrange) for all seats, lavatories, galleys, and crew rest. Therefore the microscopic approach for the cleaning process models these four steps.
the cost of the turnaround operations and find a trade-off in cost liability and outbound delay.

\[ \min F = c_R + c_d \]  \hspace{1cm} (4)

\[ c_R = c_{R,1} \cdot t_{R,1} + c_{R,2} \cdot t_{R,2} + \cdots + c_{R,n} \cdot t_{R,n} \]  \hspace{1cm} (5)

\[ \min c_d = c \cdot (t_{remaining\ delay})^a \]  \hspace{1cm} (6)

(c) Risk minimization

A third possible target function is the minimization of risk in case of a tight schedule and is a derivate from case (a) with no actual delay occurring. As shown in the lower graph in Figure 5, the predicted turnaround distribution may only finish with a probability of around 60-70% in the required time. In case the flight is critical to depart on the planned time due to slot restrictions or important connecting flights at the destination, stakeholders have the possibility to deploy additional resources to finish turnaround operations in the critical time frame with an acceptable probability of 90% or higher. The final cost therefore consists only of additional cost for resources, while delay cost \((c_d)\) equals zero.

Figure 5: Possible target functions for process control

APPLICATION

The prototype simulation environment, which was introduced in [10], has been extended with the microscopic process models for boarding and cleaning as well as basic control strategies based on the above mentioned target functions. To determine the cost of delay, individual costs categorized by aircraft type have been integrated based on values defined in [12]. The cost for use of additional resources has been taken from different ground handling and airport fee schedules. Simulation results show a high potential for reducing delay while adding only moderate costs for airlines. Using a cost minimization approach, costs can actually be reduced by adding additional resources caused by the reduced delay cost which has a progressive behaviour with raising delay while resource cost is a fixed value multiplied by time it is used (see Figure 6).

Figure 6: Delay and cost reduction using cost minimization approach

While a small initial delay of 10 minutes can be compensated by approximately 90%, higher delays cannot achieve such compensation. Naturally, this comes also due to the basic strategies currently implemented. High delays of 50 minutes have different possibilities, such as e.g. aircraft change, which are currently not modelled.

Figure 7: Variation in cleaning steps

By modifying the steps in aircraft cleaning (see ‘Fast’ in Figure 7) and using different boarding procedures (outside-in procedure being most beneficial independent from used doors) the approach of risk minimization for flights with critical schedule could be achieved (see Figure 8).

Figure 8: Example for risk minimization using different turnaround strategies
OUTLOOK
Single parts of the presented turnaround management system have already been verified in the field. The next step is to validate the turnaround time prediction using stochastic process descriptions and basic control options in a live airport environment. Therefore two projects with a small and a hub airport are in progress – giving the opportunity to verify the system behavior and adaptability in different sized environments.

The presented control options mainly represent a tactical and basic approach to adapt turnaround operations in case of disturbances. In the future, control options will be extended beyond the two processes of boarding and cleaning and more advanced control strategies including strategies for actively running processes will be developed allowing an automated decision support for finding the optimal intervention option by means of delay and cost minimization.

ACKNOWLEDGMENTS
The work presented is sponsored as part of the HALA! Ph.D. programme.

REFERENCES
A Property-Graph-Based Knowledge Representation for Decision Support Systems

Lutz Bretschneider  
Institute of Flight Guidance  
Technische Universität Braunschweig 27, Hermann-Blenk-Str.  
Braunschweig, 38108 Germany  
+49 531 391 9819  
l.bretschneider@tu-bs.de

Peter Hecker  
Institute of Flight Guidance  
Technische Universität Braunschweig 27, Hermann-Blenk-Str.  
Braunschweig, 38108 Germany  
+49 531 391 9802  
p.hecker@tu-bs.de

ABSTRACT
This paper addresses the investigation of the performance of a property-graph-based knowledge representation in the context of a decision support system. It is part of a PhD project focusing on the development of a knowledge-based flight state evaluation system that supports the pilot with the goal to improve the situation awareness in a more complex system environment. The depiction of requirements and knowledge principles are followed by a Complexity Analysis that evaluates the order of performance in the worst case of several given tasks. These tasks are selected from typical issues of the knowledge management in the intended system. The investigation compares a property-graph-based system with a schema-based and hybrid approach. It is shown that the high performance is achieved especially in complex knowledge access and discovery of knowledge. Furthermore, a qualitative comparison shows the advantages of the knowledge representation with a property-graph. Finally, limitations of the investigation are discussed and improvements and future work are presented.

Author Keywords  
Complexity Analysis; Decision Support System; Property Graph; Knowledge-Based System; Situation Awareness

ACM Classification Keywords  
D.2.8 Software Engineering - Complexity measures  
I.2.4 Knowledge Representation Formalisms and Methods

INTRODUCTION
Motivated by the increase of automation and system complexity in the cockpit, the related PhD focuses on the development of a knowledge-based flight state evaluation system that supports the pilot to ensure the Situation Awareness in complex system environment.

Situation Awareness is defined as “a state of knowledge, from the processes used to achieve that state” [2]. In the model of Situation Awareness the process is divided into a) perception of elements, b) comprehension of current situation and c) perception of future status. One approach of understanding how the process is managed by the human is named the Mental Model. The challenge for the human in the process of Situation Awareness is a situation depending focus. Endsley (1995) refers Rouse and Morris (1985) who describe the Mental Model as “mechanisms whereby humans are able to generate descriptions of system purpose and forms, explanations of systems functioning and observed system states, and predictions in future states” [2]. In the context of knowledge the Mental Model is depicted by amount and content of knowledge the human uses for processing Situation Awareness. This covers with the “Construction-Integration Theory” of Knitch (1998) which describes the process of Situation Awareness as context sensitive knowledge activation [4]. This means that the perceived information and background knowledge results in an activation of relationships in the knowledge net which represents the situation model. The integrity of this situation model defines the level of Situation Awareness.

Based on these findings the related PhD project aims at defining a knowledge which can describe a Mental Model with the ability to discover situation models and adaptive knowledge structures for several scenarios. To achieve this objective the paper describes a graph-based knowledge representation that is analyzed and compared to other possible approaches.

KNOWLEDGE REPRESENTATION
The knowledge representation describes the way how the knowledge is managed. Knowledge in system engineering can be specified by facts, rules, heuristics and metaknowledge [1]. The depiction of knowledge is a result of data, the context of data, information and the dependencies between information. To develop a system for flight state evaluation all kinds of knowledge, information and data should be regarded in the definition of the knowledge representation. To prevent a loss of knowledge caused by the separation of different knowledge types or information a unique representation is required.
With the focus on the automation of the Situation Awareness process as a human support system, the comprehension of the system is important. That means the knowledge representation – a major part of the system – shall support the requirement of comprehension. Additionally, the knowledge has to be managed in an efficient way to allow fast access of the knowledge in the dynamic context of flight.

In the theory of knowledge representation two approaches can be named. On the one hand the use of a well-defined structure where knowledge, information and data is classified and assigned and on the other hand the network- or graph-based approach where knowledge, information and data are depicted in dependencies.

**Graph-Based Knowledge Representation**

The intended knowledge representation is based on a Property Graph (cp. Figure 1). It combines the idea of semantic networks with the theory of graphs. Data can be represented in a form of nodes, edges respectively relationships and properties. The combination of nodes and relationships results in an information base. The specification of nodes and relationships with properties adds information and results in a definition of knowledge.

The characteristic of the Property Graph allows depiction of knowledge with a less defined structure that easily enables a depiction of different kind of entities. Moreover, the principle of a local search, that means search from one node via the relationships to other nodes, implies a scalable performance of knowledge access in contrast to a global search executed by a structured knowledge representation. To prove this assumption the following Complexity Analysis is conducted.

**COMPLEXITY ANALYSIS**

A Complexity Analysis is a common method to analyze the performance of algorithms. In the context of knowledge representation the analysis focuses on the performance of the Property Graph and its knowledge management. To analyze that, the Property Graph and two comparable representations are presented. With the definition of typical tasks to write, change, delete and especially access the knowledge the algorithms are developed and analyzed.

Inside the investigation the required memory and runtime can be scored. In the evaluation of the knowledge representation the amount of memory is not analyzed because sufficient size of memory is assumed. The performance can be evaluated by runtime measurement or identification of order in worst case, average case or best case. The measurement depends on the computation system and requires an implementation of the comparable representations. Therefore, the identification of order is preferred. Another constraint of the investigation is the focus on the worst case. Because the best case is no selection criteria and the average case is not definable. This is also justified by the uncertain knowledge about the implementation of the used technologies. To classify the results of the scenarios the big O-notation is used. Therefore a literature research provides the order of performance for given data access and search algorithm. [3]

**Scenarios**

In this section the scenarios are defined. They include the definition of exemplary representations as well as the definition of assumptions and tasks.

**Representations**

Subsequently, three possible representations are depicted. They are illustrated to describe the algorithms for executing the knowledge management tasks. The figures have no demand on completeness. Furthermore, some definitions are simplified respectively exemplary.

**Figure 1 - Neo4j Property Graph [6]**

**Figure 2 - Relational Representation**

Figure 2 shows the schema representation organized by the principles of relational database design. This include the depiction of n:m dependencies with a reference index like the ‘States’ table has, as well as the use of an extra table to depict n:n dependencies like the ‘Connections’ table. It contains unidirectional connections between systems, states, tools, and records. The aircraft systems are categorized into four tables: ‘Avionics’, ‘Human Machine Interfaces’, ‘Aircraft’ and ‘Control Systems’. These tables
consist of a unique index and a name. In further implementation additional information like the aircraft code in the ‘Aircraft’ table are possible. In the consideration of this schema properties are implemented as columns. The ‘Records’ table contains data, measured of control system sensors or provided by other external sources. The ‘Tools’ table is an index of available heuristics, rules, etc. To define the order the number of tables is defined as $t$, the index of a table as $i$, the number of table entries as $e$, the number of entries in a specific table as $e_i$ and the number of columns as $c$.

**Figure 3 - Hybrid Representation**

Figure 3 illustrates a hybrid representation. The principles of a Property Graph are managed in a schema that includes a ‘Nodes’ table, a ‘Properties’ table and a ‘Relationships’ table. The properties are joined by a reference index of a node or relationship. The relationship is defined by a node A and a node B and an additional attribute ‘direction’. This representation is characterized by the number of nodes $n$, the number of relationships $r$ and the number of properties $p$.

**Assumptions**

The following assumptions are defined for the evaluation and to enable a comparison:

I. the queries using an index in the SQL database and the graph database have a similar performance,

II. the number of table entries $e$ is equivalent to the sum of nodes and relationships:

\[ e = n + r, \]

III. the number of connection table entries equals the number of relations:

\[ e_T = r, \]

IV. the total number of table entries minus the connection table entries are equal to the number of nodes:

\[ e - e_T = n, \]
the number of tables is less than the number of nodes and less the number of relationships:

\[ t < n, \]

\[ t < r, \]

VI. the number of relationships approximately is three times the number of nodes:

\[ r = 3n. \]

The assumption VI is an approximation because more than one relationship exists between the nodes. Some are unidirectional and then twice and some connections are additional. In the worst case the graph is closed so the number of relations is

\[ r_{max} = n(n - 1). \]

But this assumption is neglected in this investigation because it is against the characteristic of a complex knowledge network where knowledge gaps exist or uncertain knowledge exists.

**Tasks**

The following tasks are defined:

(1) **Store knowledge**

The first task of a knowledge management is the storage of acquired knowledge, information and data. This includes the storage of new data, e.g. the information of a new engine type (a), the depiction of a new relationship, e.g. a new connection between the flight management system and the engine (b) and a new property that can be an engine product number (c).

(2) **Read and find knowledge**

The second task is the most important part for flight operations. The ability to access the knowledge fast is evaluated with the following tasks. Firstly, find information e.g. the engine (a). Another task is reading of connected information to information A, e.g. to find all input information on a system (b). Finding all paths from information A to B is a task to identify all possible information flows between the engine and the pilot (c). In case of a system failure the pilot has to know which systems are affected so the general task is to find all unidirectional paths from information A (d). Finally, the representation should allow finding an alternative path from information A to information B with the same type of relations. The example in the investigation is the loss of the connection between the pilot and the engine warning light. In that case the path via the flight management system has to be identified (e).

(3) **Change and improve knowledge**

The third part of the knowledge management evaluates the performance to improve knowledge. Therefore, an issue should be concretized, e.g. changing the engine information with a static power property to a thrust-, power- and consumption-characteristic (a). It is also possible to add information, e.g. the parts of the engine (fan, compressor, turbine, combustion chamber, tailpipe and housing) to the system (c). Finally, an information property should be changed, e.g. change the engine state from running to stopped (d).

**Results**

Table 1 shows the reduced order of performance grouped by type of representation in columns and by tasks in rows. The reduced order means that constants and low order summands are eliminated from the functions.

<table>
<thead>
<tr>
<th>Schema Representation</th>
<th>Hybrid Representation</th>
<th>Graph Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a ( O(\log e_3) )</td>
<td>( O(\log n) ) ( O(\log n + \log r) )</td>
<td></td>
</tr>
<tr>
<td>1b ( O(\log e^7) )</td>
<td>( O(\log n + \log r) ) ( O(\log n + \log r) )</td>
<td></td>
</tr>
<tr>
<td>1c ( O(e_3) )</td>
<td>( O(\log n + \log p) ) ( O(\log n + \log p) )</td>
<td></td>
</tr>
<tr>
<td>2a ( O(\log n) )</td>
<td>( O(\log n) ) ( O(\log n) )</td>
<td></td>
</tr>
<tr>
<td>2b ( O(e^7 \log (e - e_7)) )</td>
<td>( O(r \log n) ) ( O(n + r) )</td>
<td></td>
</tr>
<tr>
<td>2c ( O(e^7 \log r) )</td>
<td>( O(r^2) ) ( O((n + r) \log r) )</td>
<td></td>
</tr>
<tr>
<td>2d ( O(\log n) )</td>
<td>( O(\log r) ) ( O(n \log r) )</td>
<td></td>
</tr>
<tr>
<td>2e ( O(e^7 \log r) )</td>
<td>( O(r^2) ) ( O(n \log r) )</td>
<td></td>
</tr>
<tr>
<td>3a ( O(y \log e_3) )</td>
<td>( O(y \log p) ) ( O(n + y) )</td>
<td></td>
</tr>
<tr>
<td>3b ( O(\log e_3) )</td>
<td>( O(\log n + \log r) ) ( O(\log n + \log r) )</td>
<td></td>
</tr>
<tr>
<td>3c ( O(\log e^7) )</td>
<td>( O(\log n + \log r) ) ( O(\log n + \log r) )</td>
<td></td>
</tr>
<tr>
<td>3d ( O(\log e_3 + e^7) )</td>
<td>( O(\log n + p) ) ( O(\log n + \log p) )</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1 - Order of Performance**

Before the results that are shown in Table 1 are discussed task 1b is described as an example. The algorithm of task 1b to insert a relationship in the schema representation has to search for the engine id (in table 3) and the flight management system id by using the index (in table 1). Afterwards, the connection can be inserted in table 7 and registered in the index of the connection table. The resulting order of the example is

\( O(\log e_3 + \log e^7 + \log e^7) \).

As a result of the assumption III and IV the order is reduced to \( O(\log e^7) \) because the number of entries in table 3 and table 1 is less than the number in table 7. If the task is fulfilled in the hybrid representation, in a first step the engine entry id and the flight management system id are searched in the node table with use of the index. The result is a logarithmic order (\( \log n \)). To insert the relationship into the relationship table the order of one is used to insert the relationship in the table and an additional logarithmic order \( O(\log r) \) is used to add the entry to the index of the table. For this task the algorithm to insert the relationship in the knowledge graph is equal to the algorithm of the hybrid representation.
This example results in a logarithmic order for all representations. Considering the assumption VI the logarithmic order of the schema representation is less high then the order of the hybrid and graph representation. The logarithmic order of the first task differs more. And the linear order of the schema algorithm in task 1c is comparable to the logarithmic order because of the huge difference between the number of table entries \(e\) and the number of nodes \(n\) or relationships \(r\). Due to these results the graph representation has a slightly performance in storing knowledge.

However, the results of task 2 present a significant lower order of performance for the graph representation. If one information is accessed a logarithmic order is given. If a task gets complex (e.g. task 2b) the runtime increases up to the linearithmic order \(O(r \log n)\) for the schema representation. The results of task 2c, 2d and 2e show an even quadratic order for the schema representation. However the linearithmic order is the worst case of the graph representation. It is a result of the graph search methods. Task 2c is solved with the use of a Dijkstra algorithm having the order \(O((n + r) \log r)\) and task 2e is solved with a two times depth-first search having a linear order \(O(n + r)\) [183].

The results of scope 3 – change and improve knowledge – is equal to the writing of knowledge. The schema representation has a slightly higher performance if the knowledge about the structure is assumed. Otherwise the order of performance would increase linear with the number of tables \(t\) and the number of columns \(c\) respectively the number of properties \(p\).

This finding verifies the assumption that the graph representation has an advantage in the access and discovery of knowledge. Although the schema representation depicts knowledge clearly, knowledge changes often forces a change in the schema structure. Moreover, a minimum of knowledge about the structure is required to request information in a schema implementation. The enhanced graph using properties defined by indexed key-value pairs indicates an advantage in the knowledge management.

**REFERENCES**


**SUMMARY AND FUTURE WORK**

With a description of the objective and motivation, the demands on the knowledge representation resulting from the focus of the related PhD are presented. In addition, the intended property-graph-based knowledge representation is depicted and the expectations are specified. Further, the paper presents the Complexity Analysis method with an explanation of the limitations and assumption of the knowledge representation investigation. The definition of the scenarios including the exemplary representations, the assumptions and the tasks is followed by the results of the Complexity Analysis. Although the graph-related higher performance in knowledge access and discovery could be shown, the theoretical analysis of the worst case is limited by the uncertain assumptions and the uncertain knowledge about the used technologies. Therefore, a comparison of exemplary implementations is intended. Furthermore a guide line for the interface of the knowledge representation has to be defined. In addition to that, the PhD project will focus on the research of a task- and state-based knowledge activation to enable the context sensitive use of knowledge.
AUTOFLY-Aid: Multi-Modal Trajectory Projection Approach for Airborne Collision Detection and Avoidance

Emre Koyuncu
Controls and Avionics Lab.,
Istanbul Technical University
emre.koyuncu@itu.edu.tr

Eduardo Garcia
CRIDA and Department of Air Navigation, Polytechnic University of Madrid
eduardo.garcia.gonzalez@upm.es

Gokhan Inalhan
Faculty of Aeronautics and Astronautics, Istanbul Technical University
inalhan@itu.edu.tr

ABSTRACT
AUTOFLY-Aid PhD project focuses on Near and Mid Air Collision (NMAC) resolution with 4D trajectory management functions seen from the airborne perspective to be implemented as automated (active) or decision support (passive) implementation of improved ACAS solution using the onboard avionics and the SESAR enhanced flight deck situational awareness. These new capabilities utilize the new CNS (primarily ADS-B and its enhancements) services and SWIM network. The dynamic 4D trajectory management approach in the project is based on a hybrid and stochastic airspace model not only representing uncertainties associated with sensed-and-received airspace traffic and intent information, but also representing limitations come from weather, terrain, no-fly-zone conflict hazards and air congestion. This paper explains the trajectory propagation approach of the AUTOFLY-Aid project based on Modal Maneuver method and its integration to probabilistic conflict/collision search process in order to provide the pilots with understanding of potential collision risks in probabilistic term.

Author Keywords
Conflict and Collision Detection, Trajectory Projection, Maneuver Modes, Pilot Decision Support Systems

INTRODUCTION
This PhD project tries to explore a new approach to developing collision avoidance logic that has the potential to significantly improve safety while reducing the rate of unnecessary alerts. The approach involves recent algorithmic advances based on probabilistic models of aircraft behavior and performance metrics. These probabilistic models can be modified to accommodate the anticipated evolution of the airspace, and the logic may be re-optimized as necessary with involving game theoretic approach. The overall automation support system which embeds dynamic 4D trajectory management is envisioned to a) provide the pilots with alternative trajectories as tunnels-in-the-sky through avionics displays on the console and head-up displays in real-time, b) provide the flight crew with quantified and visual understanding of collision risks in terms of time and directions and countermeasures, and c) provide autonomous conflict resolution as an autopilot mode, thus, ensuring highly responsive and adaptive airborne collision avoidance in face of ever challenging scenarios that involve blunders, weather/ terrain/ obstacle/ new conflict hazards.

The conflict detection methodology is based on the idea of spatial search phenomena for potential conflicts including aircraft-to-aircraft conflicts and collisions with the obstacles (rather these obstacles are soft weather hazards or hard earth objects). This search method to be investigated will rely on creation of probabilistic flight trajectory (4DT) envelopes for the aircrafts in the traffic for every predefined time window. These envelopes also include uncertainty factors existing in weather patterns and the flight models. The flight models naturally embed the stochastic nature in which the rationality (or irrationality) of the flight crews within the common airspace is presented with probabilistic action patterns. Trajectory envelope search process hinges on using multi-modal approach utilizing distinct flight modes. These flight modes can be combined to generate maneuvers within the flight envelope of the aircraft.

**Figure 1. AUTOFLY-Aid conflict detection and resolution algorithm architecture**

The main idea behind the Modal Maneuver Based Planning is to divide an arbitrary flight maneuver into smaller maneuver segments (called maneuver modes) and associate them with maneuver parameters (called modal inputs).
AUTOFLY-Aid uses this methodology to create 4D trajectory distribution maps for the composite Air Space picture (which includes other aircraft, weather patterns, terrain/ground and obstacles) at real-time (on the order of seconds) and identify/classify collision risks and potential additional risk factors. Overall methodology in conflict detection and avoidance is illustrated in Figure 1.

MULTI MODAL TRAJECTORY PROJECTION
Multi Modal control framework basically consists of decomposition of the arbitrary maneuvers into set of maneuver modes and associated maneuver parameters. Complexity of maneuver planning part has been reduced by reducing the dimension of the problem (modal sequence has strictly lower dimension than state space description) and control part was relaxed by designing specific controllers for each mode and switch between them in order track maneuver mode sequence instead of designing a single controller for maneuver tracking over full flight envelope [2, 3].

Motion planning problem for aerospace vehicles are complicated by the fact that, planners based on optimal performance begins to fail in means of computation, when one takes into account of constraints related with dynamical equations of aircraft. To reduce the complexity of this problem, motion description languages and quantized control concept have been adopted into motion planning [4]. More recently, closed loop hybrid control systems were developed based on linear temporal logic for the same purpose by [5]. For aerospace vehicles, a hybrid model for aircraft traffic management was developed in [7]. Frazzoli [4] suggested a maneuver automaton, which uses a number of feasible system trajectories to represent the building blocks of the motion plan of the aircraft, and a trajectory based (based on maneuver regulation principle) control system, which asymptotically regulates the actual trajectory to the trajectory generated by maneuver automaton. However, motion plans and controllable trajectories are restricted to the library of the maneuver automaton. Such libraries can be built by using interpolation between feasible trajectories [8]. [9] extended this system for online planning of feasible trajectories in partially unknown environments by using receding horizon iterations. In the approach presented in [10], parameterized sub maneuvers builds up complex maneuver sequences and makes possible to cover almost any arbitrary maneuver and the entire flight envelope by this approach.

Maneuver Modes and Modal Inputs
Configuration of a maneuvering aircraft can be described in terms of a single state trajectory, however, it is also possible to describe the maneuver by representing it as a sequence of predefined maneuver modes and associated parameters. In contrast to current approaches that build up a maneuver library by extracting specific maneuvers from either simulations or manual flight. In [11], it is build a parameterized maneuver library where each maneuver mode is represented by a set of state constraints and state equations that evolves according to the modal inputs applied to that mode.

In this approach maneuver modes are modeled as a nonlinear dynamical system with reduced order dynamics in terms of state constraints that rises due to description of the maneuver mode. For original coordinates, following complete state vector structure can be given;

\[ X = \begin{bmatrix} V_T & \alpha & \beta & \varphi & \theta & \psi & P & Q & R & n_p & e_p & h \end{bmatrix}^T. \]

In Modal description, states on maneuver mode are decomposed into three subspaces;

\[ O \in \mathbb{R}^{n-l-k}, D \in \mathbb{R}^l, M \in \mathbb{R}^k. \]

In this decomposition, O represents the states that are constrained to be kept constant during execution of the mode. D represents the states that are driven by modal inputs and M is the set of modal inputs. In this context, a maneuver mode is a reduced order dynamical system given by equations;

\[ \dot{O} = 0, \]
\[ \dot{D} = h_1(D, M), \]
\[ \dot{M} = h_2(D, M) + g_1(u). \]

Given a nonlinear system and a set of maneuver modes, the state space trajectory of the system is replaced with the trio sequence given by;

\[ (q_{ji}, m_{ji}, \tau_i), i = 1, 2, ..., N, j = 1, 2, ..., M, \]

where N is the number of modes in mode sequence, M is the number of maneuver modes in the library. qi is the i-th maneuver mode of the system executed in it h order. Here i is referred as the i-th flight mode that the aircraft is executing with the maneuver sequence. mi is the set of modal input values associated with i-th flight mode, and \( \tau_i \) is the time duration of the i-th flight mode. Since the system dynamics of each mode is considerably simpler and of lower dimension than the full state space dynamics, both the flight planning and the flight control problems are expected to become less challenging using this framework.

Cruise modes
Cruise modes class consists of level flight and climb/descent modes, which are characterized by straight flight path with fixed heading and zero body angular rates.

Level Flight Mode: Level flight mode is the basic aircraft maneuver, where the aircraft flies at a fixed heading and altitude while keeping wings level with zero roll angle. A commercial aircraft spends most of its route in this mode, since it is the most convenient way to fly from one point to another while keeping altitude constant. q1 is assigned to represent the level flight mode. Velocity is the only degree
of freedom of level flight mode; hence the only modal input and altitude rate is constrained to zero. Note that the velocity is allowed to be time varying in this case which provides the capability to perform maneuvers such as accelerating level flight. Using the context of the equations state representation, the variables and the constraints of this mode are:

\[
O_q = \{e_p, h, \phi, \psi, \beta, P, Q, R\},
\]
\[
D_q = \{n_p, \theta = \alpha\},
\]
\[
M_q = \{V_T(t)\}.
\]

**Climb/Descent Mode:** Climb/descent mode is also one of the most commonly encountered maneuver used for gaining or losing altitude. In commercial flight this maneuver is performed during take-off and landings. The variables and the constraints of this mode are:

\[
O_{q_2} = \{e_p, \phi, \psi, \beta, P, Q, R\},
\]
\[
D_{q_2} = \{n_p, h, \theta\},
\]
\[
M_{q_2} = \{V_T(t), \gamma = \theta - \alpha\}.
\]

### Three Dimensional (3D) Modes

Out-of-plane maneuvers may be an essential in face of conflict. These maneuvers involve angular motion around all axes.

---

**Longitudinal Loop Mode:** The maneuvers consists of drawing full and half circles that are denoted as loops in the longitudinal plane where it can be defined as a loop around an imaginary cylinder with a base in longitudinal plane and extends into lateral axis of the aircraft can be executed. Longitudinal loop mode is denoted as q4. The dynamics of the longitudinal loop mode are:

\[
O_{q_4} = \{r_{lon}, \beta\},
\]
\[
D_{q_4} = \{\eta_{lon}, e_p, \alpha, \phi_w, P, Q, R\},
\]
\[
M_{q_4} = \{V_T(t), \theta_w, \psi_w\}.
\]

---

**Figure 3. Longitudinal loop modes with zero and nonzero wind axes yaw angles**

**Attitude Transition Modes**

Cruise modes, along with loop modes, serve as a modular library to navigate in a three dimensional environment. Since the path of an aircraft is not a modal input to any of the modes, path continuity is automatically satisfied. However, attitude continuity in terms of aircraft attitude and velocity vector attitude, is not automatically satisfied due to the fact that the maneuver sequence can embed discontinuous attitude transitions and switches. To resolve this issue, two controlled attitude transition modes are presented in [11].

**Pitch/Yaw Transition Mode:** The main aim of the pitch/yaw transition mode is to point an aircraft in the right direction in terms of body axes Euler pitch and yaw angles. This mode is referred as q5. The dynamics are given by;

\[
O_{q_5} = \{n_p, e_p, h, V_T, \phi\},
\]
\[
D_{q_5} = \{\theta, \theta_w, \psi_w\},
\]
\[
M_{q_5} = \{\theta(t), \psi(t)\}.
\]

Angular rates of pitching and yawing angles are modal inputs to this mode and this selection of modal inputs is analogous to the selection of translational velocity components on cruise and looping modes.
Roll Mode: Rolling the aircraft between switching maneuvers is a common practice in lateral and longitudinal loop maneuvers due to fact that most of the loops result in inverted or unusual roll attitude. This mode is referred as q6. The dynamics of this mode are:

\[
O_{q6} = \{n_p, e_p, h, \psi, V_T, \beta\}, \\
D_{q6} = \{\alpha, \dot{\phi}\}, \\
M_{q6} = \{\phi, (t)\}.
\]

Probabilistic Conflict Detection
Because unnecessary alerts are undesirable, such systems must be able to determine whether an alert is required. The common approach in the warning systems involves computing probability of the undesirable event and giving alert if the conflict probability exceeds the predefined thresholds.

A wide variety of different collision avoidance algorithms have been proposed and they differ in how they model of aircrafts’ motion. The proposed method in this research can be categorized in Probabilistic Methods. These type methods examine all possible future trajectories and accounts for their relative likelihoods. Assigning different likelihoods to different trajectory sets can result in robustness against low-probability threats and may prevent unnecessary alerts [13, 14, 15]. In this sense, probabilistic approaches contain both nominal and worst-case methods, which are special cases of the probabilistic approach. Nominal propagation assigns probability one to the most likely trajectory (to linear motion), and worst-case propagation assigns equal probability to all maneuvers can be performed by aircraft. In probabilistic approach, by assigning different likelihoods to different maneuvers, it is possible to reduce false alarm rate while managing low-possible threats.

The probabilistic flight envelope including uncertainties is seen as probabilistic distribution in Figure 4 generated by Multi Modal Maneuver Search Algorithm. The multimodal maneuver search relies on a hybrid approach, which chooses maneuvers from a finite maneuver set and then chooses their parameters from a continuous dynamically feasible region (Line 4 and 5). This selection is made randomly in order to cover the whole flight envelope. The trajectory distribution map, which is the set of the generated maneuvers in a probabilistic distribution, represents all potential positions of the aircrafts in the future (in Figure 4). If the generated 4D trajectory distribution maps conflict with ownships flight intent algorithm will issue an alert for potential collision in a predefined unit time (or less).

CONCLUSION
This PhD project tries to explore a new approach to developing collision avoidance logic that has the potential to significantly improve safety while reducing the rate of unnecessary alerts. The conflict detection methodology is based on the idea of spatial search phenomena for potential conflicts including aircraft-to-aircraft conflicts and collisions with the obstacles such as weather hazards or hard earth objects. This search method to be investigated will rely on creation of probabilistic flight trajectory (4DT) envelopes for the aircrafts in the traffic for every predefined time window. These envelopes also include uncertainty factors existing in weather patterns and the flight models. The main idea behind the Modal Maneuver Based Planning is to divide an arbitrary flight maneuver into smaller maneuver segments (called maneuver modes) and associate them with maneuver parameters (called modal inputs). AUTOFLY-Aid uses this methodology to create 4D trajectory distribution maps for the composite Air Space picture (which includes other aircraft, weather patterns, terrain/ground and obstacles) at real-time (on the order of seconds) and identify/classify collision risks and potential additional risk factors.

ACKNOWLEDGMENTS
This work and Emre Koyuncu is supported in part by SESAR WP-E HALA! Research Network Ph.D.
Fellowship. A part of the AUTOFLY-Aid is to be carried in CRIDA, UPM and Boeing RTE.

REFERENCES
Effects in Fuel Consumption of Assigning RTAs into 4D Trajectory Optimisation upon Departures

Santi Vilardaga
ASCAMM-CTAE
Cerdanyola del Vallés (Barcelona), Spain
santi.vilardaga@ctae.org

Xavier Prats
Technical University of Catalonia (UPC)
Castelldefels (Barcelona), Spain
xavier.prats@upc.edu

ABSTRACT
4D trajectory optimisation has showed good potential to reduce environmental impact in aviation. However, a recurrent problematic is the loss in air traffic capacity that these pose, usually overcome with speed and time advisories. This paper aims at the quantification in terms of fuel consumption of implementing suboptimal trajectories to preserve capacity. Via an own developed optimisation framework, we deliver results on how imposing a non-optimal RTA to a trajectory increases the fuel burned. We show how advancing a metering fix in an example departure trajectory translates to an increase of up 15Kg of fuel burned. Similarly, postponing it 50s, will burn around 23Kg more. Also, imposing a level phase (due to incoming traffic) will typically consume around 25Kg more. Different scenarios and situations are studied for the fairest comparison.

Author Keywords
trajectory optimisation; 4D trajectories; non-linear programming; optimal control; fuel; performance models; BADA; RTA; metering fix;

INTRODUCTION
The improvement of air transport efficiency (in terms of economic and environmental impact) is one of the major drivers for research and development in the SESAR and NextGen programmes. New technologies and procedures for future ATM and on-board systems and operations are being investigated and proposed. Initiatives such as Continuous Climb Departures (CCD), Continuous Cruise Climb (CCC), and Continuous Descent Approaches (CDA) propose good fuel reduction in specific phases of the flight. However, such operations usually come with a negative impact in air traffic capacity, given the vast topologies of aircraft and hence diversity in vertical and speed profiles.

Uncertainty in Top of Descent (TOD) for example has been studied in [1]. Solutions to this issue usually come with the acceptance of sub-optimal trajectories given dynamic speed requests [2], multiple flight path angle (FPA) phases [3], requested time of arrivals in a specific point (RTA) [4], etc. For example, several research has been done in the integration of CDA in dense TMAs [5,6]. The Oceanic Tailored Arrivals program, currently in place in San Francisco airport [4], is another relevant example. These arrivals are supported by the Efficient Descent Advisor (EDA) developed by NASA-AMES, which is able to compute conflict-free optimal descent trajectories and satisfy a given arrival fix metering [7].

Even if this research is indeed very promising and is setting the foundations for future applications, we are still far to fulfil SESAR objectives in terms of significantly improving the trajectory efficiency in terminal airspace. Additionally, in order to preserve air traffic capacity, it is usually assumed that a sub-optimal solution is implemented. This paper aims at the quantification in terms of fuel consumption of such sub-optimal trajectories via an own developed optimisation framework.

OPTIMISATION FRAMEWORK
Trajectory modelling and optimisation has been a subject widely researched in the last decades. Analytically, this optimisation problem can be formally written as a continuous optimal control problem and extensive research on its resolution can be found in the literature. However, realistic trajectories are hardly impossible to solve analytically and a wide variety of numerical solutions have arisen. One of the most relevant ones involves the direct transcription of the problem, leading to a Non-Linear Programming (NLP) problem with a finite set of decision variables [8]. This approach will set-up the basic theoretical background for the research proposed in this paper.

Equations of Motion
In this paper we have taken a Point-Mass representation of the aircraft, where forces apply at its Centre Of Gravity (COG). For the initial assessment proposed in this paper, a winds calm situation, in a flat non-rotating earth has been assumed. The equations of motion are written as follows:

\[ \dot{v} = \frac{1}{m} (T - D - mg \sin \gamma) \]  
\[ \dot{\gamma} = \frac{v}{n_z} (n_z \cos \phi - \cos \gamma) \]  
\[ \dot{\psi} = \frac{g \sin \phi}{v \cos \gamma} n_z \]  
\[ \dot{x} = v \cos \gamma \cos \psi \]  
\[ \dot{y} = v \cos \gamma \sin \psi \]  
\[ \dot{h} = v \sin \gamma \]
where \( x \), \( y \) and \( h \) are the spatial location of the aircraft, \( v \) is the velocity, \( \gamma \) the flight path angle, \( \psi \) the heading and \( \phi \) the bank angle. The load factor \((n_\alpha)\) is defined as the relation between the aerodynamic lift force and the aircraft weight as follows:

\[
n_\alpha = \frac{L}{m g}
\]  

(7)

To the calculation of the aerodynamic and propulsive forces, we use BADA performance models [9]:

\[
C_L = \frac{2 m g}{\rho v^2 S \cos \phi}
\]  

(8)

\[
C_D = C_{D_0} + C_{D_2} \beta^2
\]  

(9)

\[
D = \frac{1}{2} \rho S v^2 C_D
\]  

(10)

\[
T = C_{TA} \left( 1 - \frac{h}{C_{TA}} + C_{TA} h^2 \right)
\]  

(11)

where \( C_{TA}, C_{TA_2} \) and \( C_{TA_3} \) are climb coefficients specified in the BADA tables; \( C_{D_0} \) is the parasitic drag coefficient; \( C_{D_2} \) is the induced drag coefficient; and \( \rho \) is the air density at altitude assuming ISA atmosphere as calculated in BADA. To model the throttle position we define \( \mu \) as a percentage multiplying \( T \).

Additionally, BADA defines the fuel flow as follows:

\[
ff(v, T) = \eta T
\]  

(12)

where the thrust specific fuel consumption is modelled as:

\[
\eta = C_{f_1} \left( 1 - \frac{v}{C_{f_2}} \right)
\]  

(13)

being the coefficients \( C_{f_1} \) and \( C_{f_2} \) also defined in BADA.

**Problem Formulation**

Optimal control problems are usually nonlinear and generally do not have analytic solutions and it is required to employ numerical methods to solve them [8, 10]. The approach is to convert the infinite-dimensional original problem into a finite-dimensional optimisation by iteratively applying three fundamental steps [8]: collocation (namely Euler, Trapezoidal and Pseudospectral are the most used [11]), solving the NLP (with NLP Solvers such as SNOPT or IPOPT) and re-dimensioning the problem (packages such as for instance GPOPS\(^1\), SOCS\(^2\) or PSOPT\(^3\) will automatically iterate the three steps).

The optimisation framework developed in this paper uses GPOPS, which implements a pseudospectral collocation method. It handles multiphase optimal control problems and can automatically resize the number of collocation points in each iteration (i.e. it does not rely on the number of points given in the guess). GPOPS is developed in MATLAB and is open source and free for research purposes.

\(^1\) General Pseudo Spectral OPtimal control Software. http://www.gpops.org


\(^3\) PseudoSpectral Optimiser, http://www.psopt.org/

Using the Equations of Motion described above, we have formulated an optimal control problem, the solution to which minimises the fuel consumption as:

\[
f(t) = \int ff(v, T) \, dt
\]  

(14)

The state \((x)\) and control \((u)\) vectors of the problem are defined as follows:

\[
x = [v \ \gamma \ \psi \ x \ y \ h]; \quad u = [n_\alpha \ \phi \ \mu]
\]  

The following table depicts the constraints considered in the optimisation problem:

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum altitude</td>
<td>( h \leq h_{MAX} )</td>
</tr>
<tr>
<td>Minimum operation</td>
<td>( v \geq v_{MIN} )</td>
</tr>
<tr>
<td>airspeed</td>
<td></td>
</tr>
<tr>
<td>Maximum operation</td>
<td>( v \leq v_{MAX} )</td>
</tr>
<tr>
<td>airspeed</td>
<td></td>
</tr>
<tr>
<td>No deceleration</td>
<td>( \dot{\psi} \geq 0 )</td>
</tr>
<tr>
<td>allowed</td>
<td></td>
</tr>
<tr>
<td>No descent allowed</td>
<td>( h \geq 0 )</td>
</tr>
<tr>
<td>Procedure Design</td>
<td>( \frac{h}{z} \geq 3.3 % )</td>
</tr>
<tr>
<td>Gradient (PDG)</td>
<td></td>
</tr>
<tr>
<td>Load factor</td>
<td>0.85 ( \leq n_\alpha \leq 1.15 )</td>
</tr>
<tr>
<td>Bank angle</td>
<td>(-25 \leq \phi \leq 25 )</td>
</tr>
</tbody>
</table>

**Table 1. Constraints in the optimal control problem**

Many of these are operational constraints, either to stay within the flight envelope or comply with ATM constraints such as ground obstacle avoidance (PDG). Since BADA defines \( v_{MIN} \) and \( v_{MAX} \) on CAS speeds, we do the conversion from TAS to ensure we stay within the limits. Additionally, the box constraints on \( n_\alpha \) and \( \phi \) where defined following usual civil aviation standards. More information on optimal control formulation techniques used in this research can be found in [12] and [13].

**Generating the optimal trajectory**

BADA defines different flight phases for a departing trajectory, with specific performance values to each phase. In it, there is a first phase where the aircraft must be at TOGA thrust climbing up without the possibility of turning or making changes in the aerodynamic configuration. In many studies, this phase is not contemplated given the low degrees of freedom in it due to operational constraints. After that, the following phases are defined by the time of aerodynamic changes. BADA defines a first flap retraction (from TO to IC) and a second flap retraction into clean configuration (IC to CL). Since flap retraction will change aircraft performance, each phase has different aerodynamic drag coefficients among other particularities. In our framework, these change with defined speed steps [14].

To take into consideration the changes in aerodynamic configurations, we use continuous and twice differentiable *switching functions*. This method has the negative impact that it adds complexity (non-linearities) to the model (hence greater calculation times and convergence difficulties), and
the minor side effect of having a transition effect around the switching value. Both issues are directly related, since the less steep is the function (and thus smoother for the NLP Solver) the bigger is the transition effect, and vice versa. Hence, a trade-off must be sought [13].

With this, we are able to compute a full trajectory from a set of initial conditions to a set of final conditions, including one or more RTA in waypoints along the route using a multi-phase optimal control problem.

SCENARIOS
For the sake of this paper, we have envisaged a set of scenarios that cover different operational constraints: from close-to-current operations to fully optimal scenarios. To this end, we have defined two baseline scenarios that will be compared to N futuristic trajectories with RTA. A first optimal departure scenario (A) has been defined from ground to cruise altitude (FL360 in the example) without traffic constraints. The second scenario (B) tries to closely reproduce a current ATCO conflict resolution with a step climb. In it, we specify a level-off segment at 10000ft from an along track distance of 20Nm to 30Nm (i.e. the aircraft reaches a point where it is told to maintain altitude up to the point where it is cleared to climb again). The optimal vertical profile of the two scenarios (A and B) is seen in Figure 1.

![Figure 1. Optimal vertical profile for scenarios A and B](image)

It is important to notice that in order to define the correct reference conditions for all scenarios, we have defined the same ending conditions for all in position, speed and altitude. Therefore we are able to fairly compare the burned fuel and the time spent in the flight. Besides, these ending conditions are defined taking into account that the aircraft must end with enough energy to continue with the subsequent phases of the flight.

Once the reference scenarios have been defined, we are able to create the following subset of scenarios in the direction of the objectives of the current paper. In a futuristic scenario, with the capability to cope with 4D trajectories we want to quantify the impact in fuel that assigning 4D metering fixes (RTA) due to traffic constraints will pose as compared to the optimal (A) and current operations (B).

The optimal trajectory (A) cannot be flown due to a potential loss of separation with other traffic. For this reason, the ATCO has taken a conservative approach leveling-off the aircraft at 10000ft, before entering in conflict (B). In a 4D futuristic scenario, we could envisage that better separation assurance techniques could be in place, such as giving an RTA to the conflicting point (P) that prevents the conflict. Thus, the separation of the two aircraft will be sought by making one (or both) of them arrive to P at an earlier or later time.

Even if assuming a static geometry of the conflict is naïve, it is out of the scope of the paper to account for dynamic geometries due to uncertainties in the conflicting trajectories. Nevertheless, this allows us to isolate one trajectory and quantify its increase in fuel with the different assigned RTA.

For the following scenario (C) we make the assumption that P is found at 25Nm north. In the reference scenario A, we find out at what altitude (12227ft) and time (271s) the trajectory reaches P. On the other hand, we calculate the fastest possible time to reach it (presumably a bit faster than the optimal reference). Then, starting from the fastest feasible time, we define an RTA and iteratively move it back in time with 10s time steps. Each RTA will give different fuel consumptions and altitudes at P (see Figure 2). These should be further studied in order to detect if the conflict is avoided and if they provide more optimal results.

![Figure 2. Optimal vertical profile for scenario C as compared to A and B](image)

RESULTS
The fuel burned for scenario B has been quantified with an increase of 25,85Kg of fuel burned, as compared to the optimal reference trajectory A.

Results for trajectories in scenario C are depicted in the chart in Figure 3. Starting from the fastest feasible trajectory, -13 seconds earlier than the optimal (calculated with the same optimization framework, changing only the objective function to time), to +50 seconds. It relates the fuel burned, the total duration of the flight and the altitude...
when reaching P.

![Figure 3. Results for trajectories in Scenario C](image)

In the previous figure, zero denotes the optimal trajectory (as calculated for scenario A) and the positive and negative values are the increase or decrease in fuel or time in comparison to the latter. In this case, if a conflict can be avoided reaching the given point P at time -13, the fuel increase will be of 5,87Kg, almost 20Kg less than B. Likewise, delaying the RTA 50s, the increase in fuel will be 22,93Kg (little less than B), reaching the point at a higher altitude (2650ft higher), though as it can be seen, the total duration of the departing trajectory will be about 47s longer.

**CONCLUSION**

This paper presents a generic optimisation framework that has been used to quantify the effect of different RTA in a departing trajectory. The results can be useful in the process of choosing a sub-optimal trajectory in respect with separation assurance, time pressure and fuel reduction.

In this paper we have experimented with different scenarios that quantify the effect in fuel consumption that sub-optimal departing trajectories pose. To this end, we have isolated one trajectory on the conflict to be able to study the increase in fuel. Effectively, these results should be used within a collaborative conflict resolution to find the optimal RTA for all conflicting trajectories.

Finally, this framework proposes a generic approach to this issue. Further research will have to be applied to specific real-life examples in order to assess the potential in minimising fuel.

**ACKNOWLEDGMENTS**

We thank the contributors to the paper and the funds that come from SESAR WP-E HALA! Network, Ascamm Private Foundation and Technical University of Catalonia.

**REFERENCES**

Distance at Closest Point of Approach for Airborne Collision Avoidance

Anna Elena Tirri, Giancarmine Fasano, Domenico Accardo, and Antonio Moccia
Università degli Studi di Napoli “Federico II”
Piazzale Vincenzo Tecchio 80
Napoli, I80125, Italy
annaelena.tirri@unina.it
+39 0817682149

ABSTRACT
This paper focuses on algorithm description and test results from an Obstacle Tracking algorithm based on a Particle Filter developed in Spherical coordinates, aimed at demonstrating Unmanned Aerial systems (UAS) autonomous collision detection capability in terms of reliable estimation of Distance at Closest Point of Approach (DCPA). In fact, in the framework of UAS Sense and Avoid problem, the estimate of DCPA constitutes a fundamental requirement for the collision risk assessment. Since assessed techniques, such as Extended Kalman Filter (EKF), can cause some loss of accuracy in obstacle tracking performance in case of non-linearities in obstacle dynamics, innovative methodologies are expected to provide more accurate estimates of tracking performance. In particular, Particle Filter technique is likely to improve the estimate of DCPA and potentially reduce the delay in collision detection.

The developed software has been tested in off-line simulations based on flight data gathered during a test campaign conducted with a very light aircraft in the framework of TECVOL project. In particular, the algorithm performances have been evaluated in radar-only configuration since the main interest was addressed to the analysis of the impact of innovative technologies on tracking capabilities.

Author Keywords
Unmanned Aerial System, Sense and Avoid, Collision Avoidance, Particle Filter.

ACM Classification Keywords
I.6. Simulation and modeling, I.6.4 Model Validation and Analysis, I.6.5 Model Development.

INTRODUCTION
In the framework of UAS Sense and Avoid problem, the estimate of Distance at Closest Point of Approach is a critical issue for collision risk assessment. Regulatory agencies prescribe that this parameter must not be less than 500 ft in case of light aircraft [10] to guarantee a level of safety compared to manned aircraft.

The Distance at Closest Point of Approach indicates the minimum distance between two aircraft that keep their velocity vector constant; it is defined by equation (1) [3]:

$$DCPA = \frac{\vec{r} \cdot \vec{V}}{\|\vec{V}\|} - \vec{r}$$

(1)

where \(\vec{r}\) and \(\vec{V}\) are the relative position and velocity between the own aircraft and the intruder.

In order to obtain an accurate estimation of DCPA, an adequate sensor setup must be chosen depending on size, speed, and maneuverability. These systems include cooperative and non-cooperative technologies [5,6,9], such as radars, electro-optical, ADS-B, and TCAS systems.

In the framework of TECVOL project carried out by the Italian Aerospace Research Center (CIRA) and the University of Naples “Federico II”, an integrated hardware architecture with a radar as the main sensor and electro-optical as auxiliary sensors has been preferred to cooperative systems since the project aimed at demonstrating autonomous UAS flight. The software system was based on an Extended Kalman Filter.

During the flight campaign, different geometries with the two aircraft in chasing and frontal encounter configuration have been performed. In this paper, a single-quasi-frontal encounter has been analyzed.

Due to the presence of multi-sensor configuration and non-linearity in obstacle dynamics, assessed EKF techniques can cause some loss of accuracy in obstacle tracking performance. Innovative methodologies such as Particle Filter are expected to provide much more accurate obstacle dynamics than EKF and reduce the delay in the assessment of the collision risk.

In this paper, after a description of the developed software, the obstacle tracking results are reported and analyzed.
DEVELOPED OBSTACLE TRACKING SYSTEM

A Detect, Sense and Avoid system plays an important role onboard UAS. In fact, this system must be able to detect and avoid other aircraft in the flight path. Indeed, the main goal of an Airborne Obstacle tracking system is to provide the best estimate of obstacle dynamics in terms of DCPA.

The developed software is based on Sampling Importance Resampling (SIR) particle filter [7] constituted by three main steps: generation of particles, calculation of the weights associated to the particles and resampling procedure to avoid degeneracy phenomenon. In this case, the resampling procedure is based on a Systematic Resampling scheme [1].

In order to obtain accurate estimate of a target position and velocity, the obstacle dynamic has to be properly characterized. Several models exist to describe the obstacle trajectory [8], in the developed algorithm a Nearly Constant Velocity model has been chosen and implemented [2].

The state vector is comprised of 7 components that are the obstacle position and velocity in terms of range, azimuth, elevation, their first time derivatives and DCPA in North-East-Down (NED) reference frame. After the initialization phase, the state vector is propagated through a non-linear dynamic equation.

ALGORITHM RESULTS

The Obstacle Tracking algorithm has been tested in off-line simulations based on real data gathered during an intensive flight test campaign. The scenario analyzed has a duration of about 20 s in a single-quasi frontal encounter geometry between the own aircraft and the intruder. The number of particle has been set equal to 500 in order to avoid a high computational load to evaluate the software performance.

The algorithm outputs are based on obstacle position and velocity in terms of range, azimuth, elevation and their first time derivatives. Besides these variables, the DCPA has been considered as an additional state vector variable in order to exploit the PF capability to handle non-linear dependencies.

The analysis of the software outputs has been performed taking into account only radar measurement since the main interest was to evaluate the impact of new filtering techniques on tracking capabilities.

The following figures show the algorithm outputs as estimated by Particle Filter tracking software, by radar, and by GPS, which has been used as reference.

In Figure 1 the obstacle range estimate is reported. The plot shows that after few seconds a firm track is generated on the basis of radar measurements the tracker is able to track the obstacle trajectory. In all figures, the colored lines represent the particle trajectories.

Figure 1. Range in NED reference frame as estimated by radar, by radar-only tracking, and by GPS, and estimation error as function of GPS time of the day.

In Figure 2 and Figure 3, the obstacle angular dynamics are reported. The tracker is accurate in estimating these variables, in fact the estimation errors in terms of root mean square is of 3.7° and 2.5° for azimuth and elevation, respectively. These values are comparable to EKF ones. For more details on EKF algorithm and performances, the reader is referred to Ref. [4].

Figure 2. Azimuth in NED reference frame as estimated by radar, by radar-only tracking, and by GPS, and estimation error as function of GPS time of the day.

The obstacle range and angular rates are reported in Figure 4, Figure 5 and Figure 6. These variables are very important information sources for the estimation of the DCPA.

Figure 3. Elevation in NED reference frame as estimated by radar, by radar-only tracking, and by GPS, and estimation error as function of GPS time of the day.
In Table 1, the Particle Filter performance are summarized. In particular, the root mean square of the estimated errors is reported.

In Figure 7, the Distance at Closest Point of Approach as estimated by radar-only tracking, by radar and by GPS (used as reference) is reported.

In order to highlights the improvement obtained by using Particle Filter techniques on DCPA estimation, the EKF performance is also reported. In this case, the Particle filter tracker is able to assess the risk of a collision faster than EKF. In particular, the PF delay is reduced of about 4 s with respect to EKF.
Figure 7. Distance at Closest Point of Approach in NED as estimated by radar, by radar-only tracking, and by GPS, and estimation error as function of GPS time of the day.

CONCLUSION
The paper focused on algorithm and test results from an obstacle tracking software based on Particle Filter technique. The algorithm performance have been evaluated taking into account only radar measurement since the main interest was addressed at innovative techniques impact on tracking capabilities.

The software has demonstrated the ability to accurately detect and track the obstacle trajectory. In particular, the PF algorithm is able to assess the collision risk faster than EKF.

Future works foresee the introduction of electro-optical sensor in the developed algorithm since these systems work at higher frequencies with respect to radar. Moreover, multi-sensor configuration allows taking advantages of information from both sensors sources.

ACKNOWLEDGMENTS
The PhD grant of Miss Tirri is co-funded by EUROCONTROL through the research network HALA! in the framework of WP-E SESAR project.

REFERENCES
Modeling External Disturbances for Aircraft in Flight to Build Reliable 4D Trajectories

Tobias Pabst*, Thomas Kunze, Michael Schultz, Hartmut Fricke
Technische Universität Dresden
Dresden, Germany
*tobias.pabst@tu-dresden.de

ABSTRACT
The introduction of trajectory based operation in future Air Traffic Management Systems is driven by the European SESAR and American NextGen Initiatives and relies on a common view for four-dimensional trajectories. Herein, Decision Support Tools will play an important role as automation will certainly increases over time to allow predicting and amending current 3D trajectories in a real-time environment by taking into account external disturbances to the aircraft such as weather or wind. This paper presents a model to generate 4D trajectories based on stochastic disturbances acting on the aircraft in flight. Definitions for momentary position uncertainty and its projection over time as a corridor of uncertainty are presented in this paper paving the way for a model transferring external disturbances into position uncertainty adhering to the trajectory. Results from the implemented model show quantified position uncertainties along the corridor of uncertainty based on the Eurocontrol BADA aircraft performance model. The research shows that the presented approach to transfer stochastic disturbances into position uncertainty is suitable for dedicated applications but not in general.

Author Keywords
Air Traffic Management; Trajectory Based Operations; 4D Trajectory Management; Uncertainty Management; Corridor of Uncertainty; Position Uncertainty.

INTRODUCTION
The objective of this study is to introduce a method to describe four-dimensional trajectories affected by non-deterministic disturbances as it occurs in day-to-day air traffic operations due to unforeseen weather and wind impacts or airspace constraints on short notice. Several definitions actually co-exist for a four-dimensional trajectory (4DT) description, especially within the concepts of operation for SESAR [2] and NextGen [5]. We start in favor of the NextGen version, as it considers the 4DT including a limited position accuracy prediction resulting from e.g. weather or clearance disturbances [5]. The purpose of the study is not to determine and categorize stochastic disturbances but to present an approach to transform the impact of disturbances into position uncertainty (PU) forming what can be called a corridor of uncertainty (CoU) over the time enclosing the 4DT. Conceptual definitions of the CoU and PU are introduced in the next section while taking ICAO PBN terms into consideration. After having set the definitions, an approach on how to quantify CoU and PU is presented in the main part of this paper. Following this, the simulation used to calculate the 4DT is introduced and the application of the methodology is explained. Then, the simulation outcomes and results are presented. In the last section concluding remarks are discussed and aspired future work is announced.

METHODOLOGY
Recent studies focus on the categorization and determination of the impact and stochastic nature of external, stochastic disturbances affecting 4DTs. Table 1 shows the two main, slightly different categorizations according to [1] and [3]:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Conditions</td>
<td>Initial Conditions</td>
</tr>
<tr>
<td>Intent Uncertainties</td>
<td>Intent Errors</td>
</tr>
<tr>
<td>Weather forecast</td>
<td>Errors in environmental information</td>
</tr>
<tr>
<td>Modeling errors</td>
<td>Modeling Errors</td>
</tr>
<tr>
<td>Flight technical errors</td>
<td>Aircraft-specific Errors</td>
</tr>
</tbody>
</table>

Table 1: Categorization of disturbances

Although the categorization criteria are almost identical, the calculation of the individual stochastic disturbance differs strongly and is not complete in terms of mapping all emerging factors. However, the sources of uncertainties mentioned in Table 1 result in deviations between the actual flown flight path and the predicted (desired) 4DT. For client applications (DSTs) in future ATM Systems it is important to understand the effects of the influence of stochastic disturbances to compute and handle 4DT with certain levels of uncertainty. Therefore it is essential to discover the nature of each stochastic disturbance and determine the
most appropriate representation to characterize them mathematically as introduced by [1]. Furthermore it should be clear how the specific disturbance is acting on 4DTs. In [3] the impact of disturbances is decomposed into three orthogonal directions (longitudinal, lateral and vertical) and it is defined which directions are affected for a set of given disturbances. This general approach is adopted and advanced into a definition for the position uncertainty and corridor of uncertainty. [6]

Position uncertainty (PU) is the confidence area with a given stochastic boundary for the future position of the aircraft at any given time between off- and on-block. PU is assumed to be three-dimensionally normal distributed with the respective standard deviations referenced to an aircraft-carried earth axis system. Meaning that the center of the axis system is located at the aircrafts center of gravity and the longitudinal axis of the aircraft represents the x-axis. The y and z axes are earth referenced, meaning that y is horizontal and z is vertical located. The geometric shape of the three-dimensional position uncertainty can be described by an ellipsoid. [4, 6]

The Corridor of Uncertainty (CoU) is represented exclusively by the vertical plane of the PU over the time. Thus, the y and z axis forming the CoU at a given time and can further be described through an ellipse. [4, 6]

The following terms are introduced to denominate the three orthogonal components of the PU:
- Along Track Uncertainty (ATU) i.e. x-axis
- Cross Track Uncertainty (XTU) i.e. y-axis
- Vertical Track Uncertainty (VTU) i.e. z-axis

Figure 1 below illustrates the appropriate components of the PU. The CoU could be illustrated by XTU and VTU (ATU is not applicable as per definition).

Figure 1. Illustration of the position uncertainty PU, decomposed into the three orthogonal components cross track, vertical track and along track uncertainty at a fixed point in space.

The introduced terminology is strongly related to the ICAO PBN terminology but in another context. Following the ICAO terminology, along track tolerance (ATT), cross track tolerance (XTT) and vertical track tolerance refer to the fact that the aircraft knows its present position with a certain degree of uncertainty. The introduced terminology however, expects a probable position with a certain degree of uncertainty for a given look-ahead time. For the actual time (t = 0) or present position the meaning of the competing terms is equal and the position uncertainty can be assumed equal to the actual navigation performance (ANP). The geometric shape of the three-dimensional position uncertainty for the special case t = 0 was shown to be described by an ellipsoid [4]. In previous work [9] it was shown for several flight segments, that the actual cross track tolerance (XTT) and vertical track tolerance (VTT) of modern aircraft lead to normal distributed envelopes around the desired track as anticipated through the ICAO PBN concept. For the along track tolerance (ATT), time to schedule adherence is the dependent variable which in this paper is initially supposed to be also normal distribution according to the following equation:

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x}\right)^2}$$

An approach to transfer stochastic disturbances into PU is seen in the utilization of the invariance of the normal distribution against convolution (multivariate normal distribution) and may be used to convolute different independent, normal distributed random variables. The approach is suitable for the determination of the PU/CoU by knowing longitudinal, lateral, and vertical standard deviations of displacement caused by specific, normal distributed and independent stochastic disturbances. The following assumption was made: Displacements along the respective axis caused by disturbances are independent and normal distributed, as this is essential for determining the emerging position uncertainty and respectively the corridor of uncertainty. For normal distributed independent random variables the following mathematical approach can be used: If X and Y are normal distributed independent random variables with the mean values $\mu_x$, $\mu_y$, standard deviations (SD) $\sigma_x$, $\sigma_y$ and a, b as any real constants, then the sum of X and Y is also normally distributed:

$$aX + bY \sim N(a\mu_x + b\mu_y, a^2\sigma_x^2 + b^2\sigma_y^2)$$

Given that the random variables are somehow correlated, the following approach is suitable: If X and Y are jointly normal distributed random variables, then X + Y is also normal distributed. The mean value is simply the sum of the means. Variances are however not additive due to the correlation $\rho$. For more than two variables, the covariance matrix can be used:

$$\sigma_{X+Y} = \sqrt{\sigma_x^2 + \sigma_y^2 + 2\rho\sigma_x\sigma_y}$$

The components of the PU (ATU, XTU and VTU) are characterized by twice the standard deviation of the displacement resulting from the impact of the specific disturbances along the appropriate axes.
\[\begin{align*}
\text{ATU} &= 2 \times \sigma_{\text{ATU}} \\
\text{XTU} &= 2 \times \sigma_{\text{XTU}} \\
\text{VTU} &= 2 \times \sigma_{\text{VTU}}
\end{align*}\]

As mentioned above, this approach is suitable for independent and normal distributed random variables and therefore limited in application. The limitation due to dependent variables could be overcome by introducing the correlation parameter, but assuming a normal distribution instead of a de facto uniform distributed random variable will probably cause a significant error.

**SIMULATION**

The approach described above was implemented into “Testbench for agent-based air traffic simulation” (TABATS), which was developed at TU Dresden within the UTOPIA Project [7]. Using a BADA-based trajectory predictor and an Airbus A320 based flight guidance model, the simulation allows precise analysis of trajectory data including the newly implemented ATU, XTU and VTU components.

<table>
<thead>
<tr>
<th>Stochastic Uncertainties</th>
<th>Stochastic nature</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind</td>
<td>Normal distributed</td>
<td>Weather/Environment</td>
</tr>
<tr>
<td>Speed Intent</td>
<td>Normal distributed</td>
<td>Intent</td>
</tr>
<tr>
<td>Track Change</td>
<td>Normal distributed</td>
<td>Intent</td>
</tr>
</tbody>
</table>

Table 2. Considered stochastic uncertainties

The considered stochastic disturbances are shown in Table 2. The wind is categorized by [1] and [3] as disturbance value belonging to weather. Wind can be decomposed into wind speed and wind direction. According to [1] the wind can be modeled by normal distributions. Speed intent and track change are categorized as intent disturbance and are assumed to be normally distributed. Table 3 illustrates the affected components of the PU for the considered disturbances. Note that a coupling may occur between components, for example uncertainties in the track change will lead to displacements relating to XTU and thereby producing ATU displacements. As the ground speed depends on the wind component, ATU and VTU are affected.

<table>
<thead>
<tr>
<th>Stochastic Disturbances</th>
<th>ATU</th>
<th>XTU</th>
<th>VTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 3. Components of the PU affected by disturbances

The representative flight used for the simulation is originating in Stockholm-Arlanda (ESSA) with Frankfurt/Main (EDDF) as destination and executed by an Airbus A320. The route is defined as follows: ESSA19R, NOSLI, BAGOS, KUXOD, MIC, ULSEN, ALOSI, RIMET, LESMO, TUNIV, KERAX, EDDF25C with NOSL4G as Standard Instrument Departure Route and KE25N as Transition. The total distance is about 685.3 nm with 20 track changes included. [6]

The following equations are derived from the approach mentioned above to determine the standard deviations of displacement for the respective axes.

\[
\begin{align*}
\sigma_{\text{ATU}}(\Delta t) &= \sqrt{\sigma_{\text{ATU},1}(\Delta t)^2 + \sigma_{\text{ATU},2}(\Delta t)^2 + \ldots + \sigma_{\text{ATU},m}(\Delta t)^2} \text{ in nm} \\
\sigma_{\text{XTU}}(\Delta t) &= \sqrt{\sigma_{\text{XTU},1}(\Delta t)^2 + \sigma_{\text{XTU},2}(\Delta t)^2 + \ldots + \sigma_{\text{XTU},m}(\Delta t)^2} \text{ in nm} \\
\sigma_{\text{VTU}}(\Delta t) &= \sqrt{\sigma_{\text{VTU},1}(\Delta t)^2 + \sigma_{\text{VTU},2}(\Delta t)^2 + \ldots + \sigma_{\text{VTU},m}(\Delta t)^2} \text{ in ft}
\end{align*}
\]

The standard deviations of displacement are calculated for each segment represented by the time interval \(\Delta t\) and for each specific disturbance value \(i\). Furthermore, for the conversion of disturbances into PU, it is assumed that the normal distribution is suitable to represent the stochastic nature of the displacement along the axes caused by each disturbance \(i\) and for each segment \(\Delta t\). Therefore it is tolerable to use the approach introduced in the methodology section. The SD of displacements caused by wind and speed intent are related to the time of the corresponding segment \(\Delta t\). The SD of displacement due to track changes only depends on the specific turn characteristics. The SD due to track changes is assumed to be 0.35 percent of the difference between straight and curved lateral path. The SD of displacement caused by speed intent is assumed by 3 percent of the mean speed of the segment. The standard deviation of displacement due to the wind, however, is modeled as mean of 8.8 knots ground speed difference with 1 knot standard deviation. Finally, the following equations are used to calculate ATU, XTU and VTU.

\[
\begin{align*}
\text{ATU}(t_T) &= \text{ATU}(t_{T-1}) + 2 \times \sigma_{\text{ATU}}(\Delta t) \\
\text{XTU}(t_T) &= 2 \times \sigma_{\text{XTU}}(\Delta t) \\
\text{VTU}(t_T) &= \text{VTU}(t_{T-1}) + 2 \times \sigma_{\text{VTU}}(\Delta t) \\
\text{with } \Delta t &= t_T - t_{T-1}
\end{align*}
\]
Figure 2: Progression of the longitudinal displacement for specific uncertainties and ATU in total.

Note that ATU and VTU are the sum of the respective component at the previous time $t_{r-1}$ and twice the standard deviation of the displacement for the actual segment, showing that the displacement propagates with time. In contrast, the XTU component can be assumed as constant over time and to be mainly depending on the flight management system installed in the aircraft. The following simulation results focus on ATU only. As mentioned above XTU is more or less assumed to be constant over time and VTU strongly depends on the operational context and increases when vertical changes occur i.e. climb and descent maneuvers.

RESULTS

The results presented in this section focus on the Along Track Uncertainty (ATU) component only and use the described methodology for the determination. Figure 2 illustrates the progression of the longitudinal displacement, in particular the ATU over flight time and can be seen as the initial prediction with the aircraft on ground and a look-ahead time equal the flight time. The solid line represents the progression of the ATU. It can be observed that the ATU increases almost linear with the time. A growth of approximately 0.367 nautical miles per minute of look-ahead time can be assumed. Furthermore, the figure shows the progression of the specific components over time, assuming the specific disturbance would exclusively cause the longitudinal displacement. With 0.285 nm/min, wind contributes significantly to the ATU compared to speed intent (0.189 nm/min) and track change (0.041 nm/min). The slight grey vertical lines represent the location of the waypoints. The waypoint LESMO and the waypoints inside the Transition KE25N (trombone) cause the biggest displacement due to specific turn characteristics. The progress of ATU in general is in accordance with the expected range mentioned by [8] of about 0.13 to 0.20 nm per minute look-ahead time with a one sigma confidence interval.

If the calculation would be updated periodically during the flight, the ATU is expected to take the shape of a sawtooth wave. In case of a periodic in-flight calculation the initial state ($t = 0$) of the displacement could be assumed to be in the range of the actual navigation performance of the corresponding aircraft.

CONCLUSION AND OUTLOOK

This paper presented an approach to transfer stochastic disturbances into position uncertainty. Therefore, the definitions needed and general methodology was introduced. First results for the ATU component were discussed. It was found that the main difficulty to determine the PU and CoU is to determine the standard deviation of displacement caused by a specific disturbance. Further, it has to be validated that the displacements caused by disturbances are normally distributed and independent.

Further research activities will focus at revealing the disturbances contributing to the components of the PU and determining the standard deviation caused by them. In addition, it will be evaluated, which distributions are most suitable for characterizing the stochastic nature of the displacement due to the impact of the appropriate disturbance. The progression of ATU, XTU and VTU will be investigated further by using the simulation. The theory of stochastic approaches will be considered to advanced or refine the approach for the calculation of the PU to overcome the assumption that displacements caused by disturbances are independent and normal distributed. The outcomes of future analysis should be verified through comparison against results of recent studies. Measured real flight data could be used for validation purposes.
REFERENCES
2. Eurocontrol, Initial 4D-4D Trajectory Data Link (4DTRAD) Concept of Operations. 2008
3. Federal Aviation Administration / Eurocontrol, Aircraft Trajectory prediction Errors: Including a Summary of Error Sources and Data (Version 0.2). 2006
6. Pabst, T. Development and implementation of a model for the description and synchronization of trajectories, taking into account stochastic disturbances (diploma thesis), Technische Universität Dresden 2012
Sensitivity of Continuous Climb Departure Predictions to Aircraft Intent Uncertainties

Enrique Casado  
PhD Candidate  
School of Aerospace Engineering  
University of Glasgow  
1013313c@student.gla.ac.uk

Dr. Miguel Vilaplana  
ATM Technologies Manager  
Boeing Research & Technology Europe  
Madrid, Spain  
miguel.vilaplana@boeing.com

Dr. Colin Goodchild  
Honorary Research Fellow  
School of Aerospace Engineering  
University of Glasgow  
c.goodchild@aero.gla.ac.uk

ABSTRACT
With the goals of higher capacity, efficiency and safety and lower environmental impact, the Air Traffic Management (ATM) system is evolving worldwide from tactical airspace management to strategic trajectory management. This paradigm shift is founded on the introduction of the Trajectory-Based Operations (TBO) concept, which establishes the procedures for optimal Trajectory Management (TM) thanks to the use of advanced automation tools. These tools will rely on the capability of accurately knowing the actual aircraft position and the intended aircraft trajectory for precisely predicting the future aircraft evolution with the time. In future TBO environment, onboard and on-ground systems will be able to predict aircraft trajectories based on the best knowledge of the aircraft intent, weather conditions, aircraft performances and initial aircraft state. However, uncertainties related to such data cannot be fully eliminated, and thus, deviations between the actual and predicted trajectories are unavoidable. The study of the prediction uncertainties will allow to evaluate their effects on the predicted trajectories, and therefore, will improve the strategic and collaborative decision making process by providing relevant information about most likely future aircraft states. This paper focuses on the analysis of the impact of aircraft intent uncertainties on the predictions of continuous climb departure procedures.

Author Keywords
Air Traffic Management; Aircraft Trajectory Prediction; Continuous Climb Operations; Uncertainties in 4D Trajectory Management; SESAR; NextGen.

ACM Classification Keywords
F.4.3; G1.10; G.3; I.6.8; I.6.5; J.2.

General Terms
Algorithms; Experimentation; Measurement

INTRODUCTION
A modernization of the ATM system is now being developed to increase its capacity and efficiency while improving the operational safety and reducing the global environmental impact. The Single European Sky (SES) in Europe [1], NextGen in United States [2] and the Australian ATM Strategic Plan (AATMSP) in Australia [3] are the main initiatives which nowadays have initiated the implementation of the TBO concept. This new ATM paradigm basically establishes the processes and procedures needed for negotiating, agreeing, executing and updating the Reference Business Trajectory (RBT) whenever safety is not jeopardized [4]. The Air Navigation Service Providers (ANSPs) will facilitate to the airspace users the required means for ensuring that such RBTs are flown within the appropriate accuracy. To do so, a new generation of advanced automation tools for supporting the trajectory management activities (e.g., conflict detection and resolution, traffic sequencing and merging or arrival scheduling) will be developed. These tools will leverage any available information shared by the Airline Operation Centers (AOCs) or directly provided by the aircraft’s Flight Management System (FMS) about actual or predicted trajectory information for enabling future procedures in the context of TBO.

Since the trajectory will be the cornerstone of the future paradigm, the need of computing accurate trajectory predictions, both on-ground and onboard, becomes paramount for reaching the expected benefits of TBO and exploiting them to the maximum. Any aircraft prediction requires from information about the aircraft intent1 (AI) which best describes the intended trajectory to be executed by the pilot or the FMS; the weather conditions, including atmospheric temperature and pressure and wind filed, within the trajectory will take place; a model of the aircraft performances which describes the aircraft behavior under the considered AI and weather; and the initial aircraft state which determines the trajectory starting conditions [6].

1 Unambiguous description of how the aircraft will be guided during the time interval for which a predicted trajectory is computed [5].
Naples, Italy, May 28-30, 2013

The aircraft intent information can be considered a prime source of uncertainty, especially in climbing or descending trajectories. For instance in [7], the lack of accurate AI information is considered the main error source in ground-based predictions, concluding that the air-ground exchange of AI data improve dramatically the accuracy of those predictions.

The work presented herein is part of a PhD studies entitled Application of the Theory of Formal Languages to the Modeling of Trajectory Uncertainty and the Analysis of its Impact in Future Trajectory-Based Operations [8] whose main objective is the identification and characterization of the sources that add uncertainty to the trajectory prediction process. The study presented in this paper is focused on the analysis of the impact of the AI uncertainty throughout the trajectory prediction process. For that purpose, a continuous climb departure has been modeled and used for evaluating the influence of those identified elements considered as sources of uncertainty.

The remainder of this paper is structured as follows. Next section exposes the methodology applied in the presented work. This section also summarizes the trajectory prediction definition used for describing the continuous climb procedures to be analyzed, and also lists the hypothesis used for predicting those trajectories. Following section gathers all outputs of the proposed experiments, where the impact of identified uncertainty sources is assessed. Finally, main conclusions and further work are included in the last section.

**METHODOLOGY**

Any trajectory prediction is affected by the uncertainty introduced by the inputs required for such prediction. This section summarizes the theoretical approach followed for describing the trajectory prediction framework used throughout the proposed study. Based on this framework, a continuous climb procedure is described as a model for evaluating the effects of the aircraft intent uncertainties on the prediction process.

![Figure 1: Block diagram of a Trajectory Prediction Architecture](image)

**Figure 1: Block diagram of a Trajectory Prediction Architecture**

**Trajectory Prediction Architecture**

For analyzing the influences and the impact of uncertainty sources it is required a formulation of the aircraft motion problem where the effects of all inputs can be considered individually. Figure 1 depicts the architecture proposed in [9], which has been assumed for the work presented afterwards. This approach decouples each data set required for trajectory computation, and therefore, allows to study the influences of each one independently from the others.

**Aircraft Intent Description Language**

Since the study aims at evaluating the impact of the uncertainty related to the AI in the final computed trajectory, a formal methodology for defining such input is needed. The Aircraft Intent Description Language (AIDL) [10] provides the capability of univocally describing the AI which represents the trajectory to be computed during a time interval.

The AIDL is a formal language intended to express aircraft intent information in a univocal, rigorous, and standardized manner. It is composed by a finite set of instructions which defines the alphabet and a set of rules which determines when a sentence is well formed. Instructions are characterized by two attributes: the effect or mathematical expression which constrains the aircraft motion; and the time interval during which this effect is active. The grammar establishes the rules for combining instructions appropriately both sequentially (instructions with contiguous, non overlapping execution intervals) and simultaneously (instructions with overlapping intervals). A correct AIDL sentence is formed by instructions organized along 3 Configuration Profiles (for describing the aerodynamic configuration of the aircraft during each time interval) and 3 Motion Profiles (2 for describing the longitudinal path and 1 for the lateral path). In total, 6 simultaneous instructions correctly ordered as stated by the grammar rules are required for defining an operation (unambiguous description of the aircraft behavior during a time interval).

All the hypotheses and mathematical formulation considered during the development of the AIDL are widely discussed and justified in [5, 6].

**Continuous Climb Operations**

Future ATM operations will be characterized by more efficient procedures which provide in return fuel savings and a lower noise and emissions impact. One of the envisioned transformations is the implementation of continuous climb operations (CCO) which will support optimal departure profiles. CCO are operation, enabled by the airspace design, and by the appropriate release of Air Traffic Control (ATC) clearances, in which departing aircraft climbs without interruption, to the greatest possible extent, by flying an optimal vertical profile defined by a continuously climbing path [11].

The efficiency of such procedures derives from the fact that level-off segments at lower altitudes rather than the cruise altitude require extra fuel consumption because the fuel burn by the engine for providing the same thrust decreases
with the altitude. Therefore, reducing the ATC interventions to the minimum, always obeying the safety requirements, will allow to execute optimal climbing profiles until initiating the cruise phase.

Departing and arriving traffic are in many cases interdependent. Usually arrivals are performed at higher flight levels than departures. Thus, lifting ATC restrictions for arrivals would benefit departures in the same way. For example, the design of Continuous Descent Operations (CDO) at Dallas/Fort Worth International Airport would elevates the arrivals by 2,000ft to 4,000ft (depending on where the streams cross) above the departure stream [12]. This change in the airspace structure would facilitate CCO with the associate improvement of fuel burn, noise abatement and emissions reductions for both climbing and descending traffic.

This interdependency implies that ATC requires precise information about both traffic flows in order to maintain safety, while providing airspace users with the means for executing their preferred profiles. Inaccuracies or deviations from actual and predicted trajectories potentially could affect both traffic flows, impacting negatively the global system performances.

The analysis presented herein considers a continuous climb procedure (described in the following subsection) as the operational procedure used for studying the impact of the AI uncertainties into the aircraft trajectory prediction.

**Description of a Continuous Climb Departure Procedure**

A continuous climb departure describes the departure procedure from the initial rotation until the top of climb (TOC) characterized by a continuous climb at optimal engine ratings without any intermediate level-off segments. The execution of such procedures depends strongly on the type of aircraft, the take-off weight, the weather conditions (especially wind and atmosphere temperature) and the aerodrome altitude.

There are many different strategies for performing such procedures. A general formulation would be characterized by:

- Use of the optimal engine rating according to the aircraft weight, weather conditions and aerodrome altitude.
- Constant ascending profile until reaching the cruise altitude.
- Lateral path according to navigation charts and ATC restrictions unless any indication is specifically release by the controller.
- Optimal flaps and landing gear retraction according the airline recommendations.

Based on these high level assumptions and the information about take-off procedures included in the airline’s aircraft operation manuals [13], and making use of the versatility of the AIDL, a continuous climb departure can be described as follows:

- **Initial conditions:**
  - Aircraft: B737-800
  - Weight: 79.000kg
  - Aerodrome Altitude: 300ft
  - Calibrated Airspeed (CAS): 156kn
  - Flaps position: 5º
  - Landing gear: Deployed
- **Maximum Climb (MCMB) engine rating from the initial conditions until the TOC.**
- **Linear acceleration from the initial CAS until 310kn, speed at which it is planned a CAS/Mach transition. This acceleration has been modeled assuming that the aircraft will be flying at 250kn CAS at 10.000ft of pressure altitude.**
- **Cruise Mach speed equal to 0,80.**
- **Constant heading through the whole trajectory.**
- **Flaps retraction to 1º at 180kn CAS and fully retraction at 200kn CAS.**
- **Landing gear retraction at 200kn CAS.**
- **Pressure altitude of 33.000ft at the beginning of the cruise phase.**

Due to the focus of the study is on the AI, null wind and null deviations from the international standard atmosphere (ISA) have been considered.

**Description of Aircraft Intent Uncertainties**

Although there are many different ways for describing how an aircraft can be commanded and guided for executing the desired continuous climb procedure, the nominal case considered in this work is that exposed in previous subsection. This AI will be subject to the effects of stochastic perturbations to try to elucidate how each element introduces uncertainty to the process.

The first step is to differentiate the invariants of the AI, or those elements which are fixed under any circumstances. For this nominal case, the list of invariants is:

- **Initial conditions. It is assumed that the initial state of the trajectory is accurately known.**
- **All the AIDL instructions and their specifiers.** For example, the engine rating in ADIL is expressed by a Throttle Law (TL) instruction whose specifier is MCMB. For the complete AI, both instructions and specifiers are maintained invariant.
- **Flaps scheduling.** Under certain circumstances, it would be possible to completely retract the flaps from 5º without passing through the 1º stage. Nevertheless, a two stages retraction is always performed.
• Cruise altitude and Mach speed.

Once the invariants are identified, it is plausible to consider that the remaining elements of the AI are the uncertainty sources which can affect the prediction of the continuous climb trajectory. The list of uncertainty sources can be grouped as follows:

- Configuration uncertainties:
  - Speed target for the flaps retraction to 1º (ST1)
  - Speed target for the complete flaps retraction (STCL)
  - Speed target for the landing retraction (STLG)

- Motion uncertainties:
  - Linear acceleration\(^2\) (LA) as function of the pressure altitude (H\(_p\))
  - Speed target at the end of the linear acceleration (ST)
  - Cruise Mach target (MT)

Therefore, the AI uncertainties can be expressed as stochastic variations around the nominal values of the indentified parameters.

### SENSITIVITY OF TRAJECTORY PREDICTION ERRORS TO AIRCRAFT INTENT UNCERTAINTIES

Form the ATM point of view, the most relevant state variables required for both traffic management and control activities are the 4D position (time, altitude, longitude and latitude) and aircraft speed, while for accurate ground-based predictions the mass variation is also a relevant factor. This section compiles the results of the sensitivity analysis on such state variables to stochastic AI variations. To obtain such results, a set of different Monte Carlo simulations, in which the defined parameters were varied according to a predefined probability density function (PDF), was designed. The trajectory predictions generated for each set of inputs were obtained from an initial version of a stochastic trajectory predictor (STP). This STP is based on an AIDL enabled tool developed by Boeing Research & Technology – Europe (BR&T-E) which makes use of the aircraft performance models provided by BADA 3.9 [14].

Due to the pressure altitude and the Mach speed at the end of the climb are invariants, the state variables to be analyzed are reduced to time, flown distance and mass variation. Following Table 1 shows the nominal values at the end of the climb which will be used as baseline data.

<table>
<thead>
<tr>
<th></th>
<th>time (t)</th>
<th>distance (d)</th>
<th>mass (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOM</td>
<td>1.315s</td>
<td>253.766m</td>
<td>76.971,3kg</td>
</tr>
</tbody>
</table>

### Table 1. Nominal trajectory prediction outputs

#### Trajectory Prediction Sensitivity to Configuration Uncertainties

The configuration uncertainties only affect the drag polar curve that has to be considered for computing each flight operation\(^3\). This means that the drag polar curve, provided by the aircraft performance model (APM) when the flaps are deployed at 5º and the landing gear is out, is different to that used during clean configuration operations. The uncertainty introduced by the stochastic parameters has been described by means of uniform probability distributions whose means are the nominal values and the upper and lower limits have been selected for an interval spread equal to the 5% of the nominal values. This hypothesis ensures that the aircraft performance limitations (basically buffet and placard speeds) are respected in all cases and that no overlapping between the two flap retraction stages occurs. Table 2 shows the values considered for the Monte Carlo runs.

<table>
<thead>
<tr>
<th></th>
<th>CAS(_{\text{Min}})</th>
<th>CAS(_{\text{MEAN}})</th>
<th>CAS(_{\text{Max}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST1</td>
<td>175,5kn</td>
<td>180kn</td>
<td>184,5kn</td>
</tr>
<tr>
<td>STCL</td>
<td>195kn</td>
<td>200kn</td>
<td>205kn</td>
</tr>
<tr>
<td>STLG</td>
<td>175,5kn</td>
<td>180kn</td>
<td>184,5kn</td>
</tr>
</tbody>
</table>

#### Table 2. Min, Mean and Max values of the stochastic configuration parameters

In this case, the motion uncertainties were considered not applicable in order to evaluate solely the effect of the configuration uncertainties. The following experiments were conducted:

- Exp1: [ST1\(_{\text{Min}}\), ST1\(_{\text{Max}}\)] – STCL – STLG
- Exp2: [ST1\(_{\text{Min}}\), ST1\(_{\text{Max}}\)] – [STCL\(_{\text{Min}}\), STCL\(_{\text{Max}}\)] – STLG
- Exp3: ST1 – STCL – [STLG\(_{\text{Min}}\), STLG\(_{\text{Max}}\)]
- Exp4: [ST1\(_{\text{Min}}\), ST1\(_{\text{Max}}\)] – [STCL\(_{\text{Min}}\), STCL\(_{\text{Max}}\)]

A summary of the results obtained after computing 1,000 trajectories based on the generation of the same number of AIs according to the stochastic description of the selected parameters can be found in Table 3, where the maximum and minimum deviations regarding the nominal case are shown.

---

\(^2\)CAS = a + b \cdot H_p

\(^3\)An operation is the elementary aircraft behavior that arises due to the simultaneous effect of a set of compatible AIDL instructions that unambiguously determine the aircraft motion during a certain time interval [5].

---
As considered for the configuration uncertainties, the variability of motion uncertainties was described by uniform probability distributions with a deviation between the maximum and minimum values equal to the 5% of the nominal values. Table 4 shows the values considered for the Monte Carlo runs.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>ST</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td>76.788 kn</td>
<td>0.0159 kn/ft</td>
<td>302.25 kn</td>
<td>0.78</td>
</tr>
<tr>
<td>78.757 kn</td>
<td>0.0163 kn/ft</td>
<td>310 kn</td>
<td>0.80</td>
</tr>
<tr>
<td>80.726 kn</td>
<td>0.0167 kn/ft</td>
<td>317.75 kn</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Table 4. Min, Mean and Max values of the stochastic motion parameters

For evaluating separately the influence of the motion uncertainties, the configuration uncertainties were not considered for the following experiments:

- **Exp5**: \([a_{\text{min}}, a_{\text{max}}] - [b_{\text{min}}, b_{\text{max}}] - \text{ST} - \text{MT}\)
- **Exp6**: \(a - b - [\text{ST}_{\text{Min}}, \text{ST}_{\text{Max}}] - \text{MT}\)
- **Exp7**: \([a_{\text{min}}, a_{\text{max}}] - [b_{\text{min}}, b_{\text{max}}] - [\text{ST}_{\text{Min}}, \text{ST}_{\text{Max}}] - \text{MT}\)
- **Exp8**: \(a - b - \text{ST} - [\text{MT}_{\text{Min}}, \text{MT}_{\text{Max}}]\)
- **Exp9**: \([a_{\text{min}}, a_{\text{max}}] - [b_{\text{min}}, b_{\text{max}}] - [\text{ST}_{\text{Min}}, \text{ST}_{\text{Max}}] - [\text{MT}_{\text{Min}}, \text{MT}_{\text{Max}}]\)

Following the same process as described above, 1,000 runs for a set of randomly generated AIs were executed. Relative maximum and minimum deviations regarding the nominal case for the considered state variables are shown in Table 5.

<table>
<thead>
<tr>
<th>t</th>
<th>d</th>
<th>m</th>
<th>(\text{MT})</th>
<th>(\text{ST})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp5</td>
<td>Exp6</td>
<td>Exp7</td>
<td>Exp8</td>
<td>Exp9</td>
</tr>
<tr>
<td>min</td>
<td>max</td>
<td>min</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>(-0.585%)</td>
<td>(+0.667%)</td>
<td>(-0.063%)</td>
<td>(+0.053%)</td>
<td>(-0.021%)</td>
</tr>
<tr>
<td>(-0.249%)</td>
<td>(+0.431%)</td>
<td>(-0.974%)</td>
<td>(+1.300%)</td>
<td>(-0.024%)</td>
</tr>
<tr>
<td>(-0.801%)</td>
<td>(+0.948%)</td>
<td>(-1.123%)</td>
<td>(+1.398%)</td>
<td>(-0.043%)</td>
</tr>
<tr>
<td>(-3.511%)</td>
<td>(+7.031%)</td>
<td>(-4.908%)</td>
<td>(+9.391%)</td>
<td>(-0.144%)</td>
</tr>
<tr>
<td>(-3.630%)</td>
<td>(+8.163%)</td>
<td>(-4.655%)</td>
<td>(+11.93%)</td>
<td>(-0.177%)</td>
</tr>
</tbody>
</table>

Table 5. Relative Min and Max deviations from Nominal Predicted Trajectory (Motion Uncertainties)

The maximum absolute time and flown distance deviations for the Exp7 are 12.5s and 3.547m, while for the Exp8 are 92.47s and 23.854m. Thus, the influence of the linear acceleration (LA) and the CAS at which the transition CAS/Mach is performed are much less important than the effect of the cruise Mach speed.

Deviations are wider for the Exp9 where all the uncertainties are combined. The maximum absolute time and flown distance deviations are 100.7s and 30.274m.

After analyzing the results, it can be stated that the lack of precise information about the cruise Mach target impact dramatically the trajectory predictions. Variations like those used in this work are incompatible with safety regulations (minimum separation of 10NM through traffic) [15]. Under current safety rules, the maximum spread from the nominal value admissible would be \(\pm 1\%\), which would lead to a minimum Mach speed of 0.7910 and a maximum of 0.8089.

**CONCLUSIONS AND FUTURE WORK**

The main conclusions of the work presented in this paper can be summarized as follows:

- Configuration parameters do not add a meaningful stochastic impact into the predicted trajectory mainly because of non-clean configuration interval is very short compared to the whole trajectory. These parameters could produce a higher impact in descent approaches, where the non-clean configuration interval is considerably longer.
- The cruise Mach target can be considered the most important source of uncertainty. Small variations on that...
parameter generate relevant deviations in time and flown distance.

- In all cases, the fuel consumption, equivalent to the aircraft mass variation, is almost not affected. This is due to the engine rating is fixed to MCMB during the whole trajectory. The differences are caused by longer or shorter paths, which in turns implies longer or shorter flight durations.

Beside of extending the analysis to the cruise and descent phases, future work will tackle with the influence of different weather conditions in the final output. It is necessary to note that all speeds (True Airspeed (TAS), Indicated Airspeed (IAS), CAS or Mach) except ground speed are strongly linked to the atmosphere conditions and could potentially add uncertainty to the AI.

ACKNOWLEDGMENTS
Thanks to the HALA! Research Network (SESAR WP-E) and BR&T-E for their support and sponsorship of the PhD studies this work is part of.

REFERENCES
11. ICAO. Continuous Climb Operation (CCO) Manual. DOC 9993 AN/495.
Organized by

HALA! RESEARCH NETWORK
http://www.hala-sesar.net/

In Cooperation with

SUPPORTED BY
SESRAR
JOINT UNDERTAKING

In Cooperation with

SICHTI

Sponsored by

IFIP WG 13.5: Human Error, Safety, and System Development

University of Naples "Federico II"

IRIT Press