

ENABLING AUTONOMOUS FLIGHT CAPABILITIES ONBOARD COMMERCIAL AIRCRAFT TO IMPROVE SAFETY

Pritesh Narayan, Sanja Dogramadzi

University of the West of England, Bristol, United Kingdom

Abstract

This paper presents the development and implementation of a novel Flight Management System (FMS) to ensure onboard passenger safety of commercial aircraft in the event of an emergency scenario where the pilot must relinquish control of the aircraft (e.g. hijacking or loss of cabin pressure). In such situations, the flight management system autonomously pilots the aircraft to a predefined destination whilst taking into account potential hazards such as terrain (in low altitude scenarios), aircraft dynamic constraints, and flight objectives with real-time constraints present. Furthermore, the FMS must ensure that the Human Decision Maker (HDM) (e.g. Air Traffic Control or ATC) has sufficient time to intervene if they do not agree with automated onboard decisions. The proposed FMS shows that the inclusion of intelligent planning capabilities onboard commercial aircraft can assist in improving safety for the aircraft and onboard passengers in the event of emergencies by allowing the aircraft to continue operating autonomously in a management by exception paradigm. This approach has been demonstrated in this paper through simulation using a commercial aircraft model operating in low altitude partially known environments.

Introduction

Civil aviation is now an essential aspect in our lives, providing such benefits as high-speed travel (commercial) and shipment of goods (transport). Globalization is expected to continue into the 21st century and beyond as the number of passenger and freight aircraft in operation grows [1].

Due to the potentially catastrophic results of aircraft accidents, civil aircraft follow very stringent manufacturing and maintenance standards. Furthermore separation services are provided by air

traffic control during Instrument Flight Rules (IFR) flight in vicinity of airports or at high altitudes.

After the events of September 11th 2001, airport security has been further improved through the use of advanced scanning and detection [2]. However, the safety of passengers onboard, or that of the aircraft platform, cannot be guaranteed, as the pilot may have to relinquish control of the aircraft in emergency scenarios (e.g. in the event of a hijacking or loss of cabin pressure). This is due to the fact that the pilot is solely responsible for providing the control inputs for aircraft planning and guidance during low altitude flight. Most literature indicates that this capability can also be replicated to some extent via intelligent control architectures [3]. Intelligent control architectures are generally structured in a hierarchical manner where high level (complex and abstract) tasks are decomposed into a series of time critical low level tasks (data rich and precise) [4].

Intelligent control architectures have been successfully applied to airborne systems [5, 6]. Boskovic [6] for example, presents an Intelligent Control based Flight Management System (FMS), which is optimized for Unmanned Aircraft (UA) navigation and incorporates high level abstract layers (e.g. Path Planning and Trajectory Generation) with time critical low levels layers (e.g. Flight Control).

Low altitude autonomous aircraft operations in the National Airspace System (NAS) are expected to be more challenging, as onboard flight management system must automate the planning and piloting aspects whilst ensuring that the aircraft remains safe from the threat of collision with terrain or other aircraft (if present). The proximity of obstacles to the aircraft may also place real-time constraints on the available computation time. An automated computationally adaptive trajectory planner within the proposed FMS may allow for the computation of an optimized solution in the presence of real-time

deadlines onboard commercial aircraft operating at low altitudes.

Automating the trajectory planning component however, is non-trivial and some challenges include: incorporation of complex platform dynamics, trajectory optimization to meet pilot objectives, real-time constraints on computation time imposed by obstacles in the flight path, and the guarantee that the generated solution is collision free. Furthermore, civil aircraft operations in the NAS are monitored by ATC, therefore the ATC must have the ability to safely intervene autonomous operations if they do not agree with the FMS decisions.

This paper details the development and implementation of a new trajectory planning approach, which allows for the computation and optimization of feasible 3D flight trajectories within real time planning deadlines, to ensure onboard passenger safety of commercial aircraft, in the event of emergency scenarios during flight in low altitude partially known environments. An outline of automated trajectory generation approaches and related work is given in the following section. The candidate trajectory generation process is also outlined, where the solution is generated through the concatenation of primitives through the application of Maneuver Automaton (MA) theory.

Autonomous Trajectory Planning

A smooth, nominal, feasible and collision free trajectory is required for safe guidance of the UA from its current state to the desired goal state. The following sections outline the candidate methods applied for the inclusion platform constraints and collision detection to ensure that the trajectories generated are both safe and feasible.

Feasible Trajectory Planning

The inclusion of aircraft dynamic constraints during the trajectory planning process, allows for the generation of feasible flight trajectories where aircraft performance and stability bounds are taken into account. Flight trajectories can be represented through a variety of methods, including, spline based or geometric approximations. Polynomial or spline based techniques [7] place control points in a particular order to generate the desired trajectory

whilst, geometric trajectory generation methods require the concatenation of segments to create a smooth optimal pathway for non-holonomic vehicles (such as fixed wing aircraft) [8, 9].

Inclusion of dynamic constraints within spline-based trajectories requires modifying control point parameters and testing to see if platform constraints are met in an iterative manner. Geometric based techniques implicitly allow for inclusion of dynamic constraints within each segment during its formulation. It must be noted that geometric planning methods such as Dubins curves [8] only consider platform dynamics during periods when the vehicle remains in a state of equilibrium [10, [11] extends Dubins' research to form smooth feasible trajectories through the concatenation of motion primitives; referred to as Maneuver Automaton (MA) theory.

Motion primitives represent aircraft maneuvers, which are generated off-line and stored into either of two classes: trim and maneuver primitives. Trim primitives represent flight maneuvers where the platform is in a state of equilibrium (platform velocity is constant and attitude rates remain zero). Maneuver primitives represent flight maneuvers where the platform transitions to a state outside of equilibrium (attitude rates are non-zero). The inclusion of attitude rate constraints allows for the generation of trajectories, which better encapsulate overall platform dynamics and can allow for improved platform tracking [12].

Figure 1 presents the concatenation of cruise and helical ascent trim primitives. The inclusion of a maneuver primitive} between the trim primitives allows for the inclusion of the fixed wing roll ($\dot{\theta}$) and pitch rate($\dot{\phi}$)constraints. For a detailed overview of MA theory and primitive representation, refer to [11, 13]. For additional mathematical proof on the concatenation of trim and maneuver primitives for fixed wing aircraft, refer to [14].

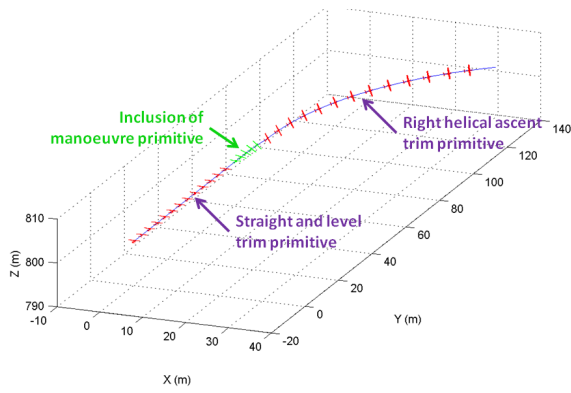


Figure 1. Concatenation of Right Helical Ascent Trim to Straight and Level Trim

The following section presents the consideration of platform safety during operations in low altitude environments through the incorporation of safe states [15].

Safe Trajectory Planning

Safe autonomous aircraft operations in cluttered environments require the generation of collision free trajectories. One way to achieve is through the discretization of the continuous flight track generated and testing that the sampled points to ensure they are collision free. A collision is deemed to take place if any part of the computed trajectory lies within an obstacle or terrain.

The inclusion of trajectory collision detection alone however, does not guarantee platform safety during autonomous operations in partially known environments. During autonomous flight where a management by exception control paradigm [16] is applied, in order to maintain full authority, the HDM (e.g. Air Traffic Controller or ATC) should be able to veto the onboard FMS decisions and command the aircraft to safely enter a loiter state at any time if they disagree with the FMS decisions. Schouwenaars [15] has applied a safe state formulation to 2D partially known environments, which allows a platform with non-holonomic constraints to enter a loiter state if the vehicle becomes trapped in local minima, thus preserving platform safety.

It was found that planning in 3D could be conducted [17] to reduce the possibility of becoming trapped in local minima as the UA has the additional option of traversing over obstacles at a higher

altitude. This section presents the formulation of loiter maneuvers for fixed wing aircraft using MA theory, and the application of safe state theory to trajectory planning in 3D partially known environments.

Left and right coordinated-turn trim primitives were performed in an alternate manner at sampled points along each trim primitive. Attitude rate constraints are implicitly considered through the inclusion of a transition maneuver between the current primitive and the loiter maneuver executed (Figure 2).

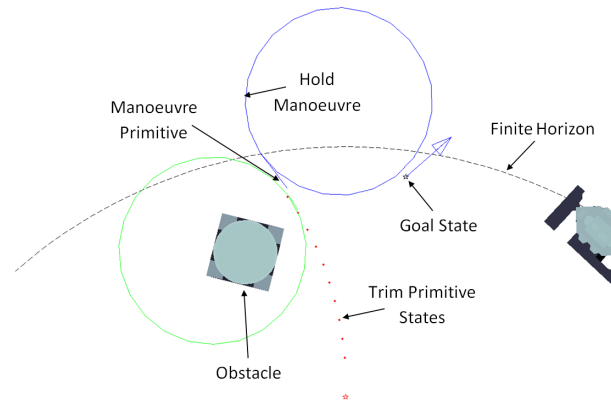


Figure 2. Execution of Fixed Wing Aircraft Hold Maneuver in Proximity to Obstacles

Using a finite receding horizon strategy, a safe state is deemed to be collision free and capable of safely executing a minimum turn loiter maneuver if the loiter maneuver does not traverse outside the finite horizon. This research applies a cylindrical region with a predefined radius R_c centered at the current platform state (\mathbf{s}_p) position ((x_p, y_p, z_p) or \mathbf{p}) to represent the finite horizon or known part of the operational environment.

Figure 3 presents a series of visual representations of the safe state formulation where:

1. Fixed resolution automaton is generated using platform dynamics.
2. Left and right loiter maneuvers generated from automaton state locations as initial states; all loiter maneuvers traversing outside the receding horizon are not considered safe.

3. All states and loiter maneuvers resulting in collisions are culled.
4. Remaining valid states form the feasible collision free automaton with implicit consideration of platform safety.

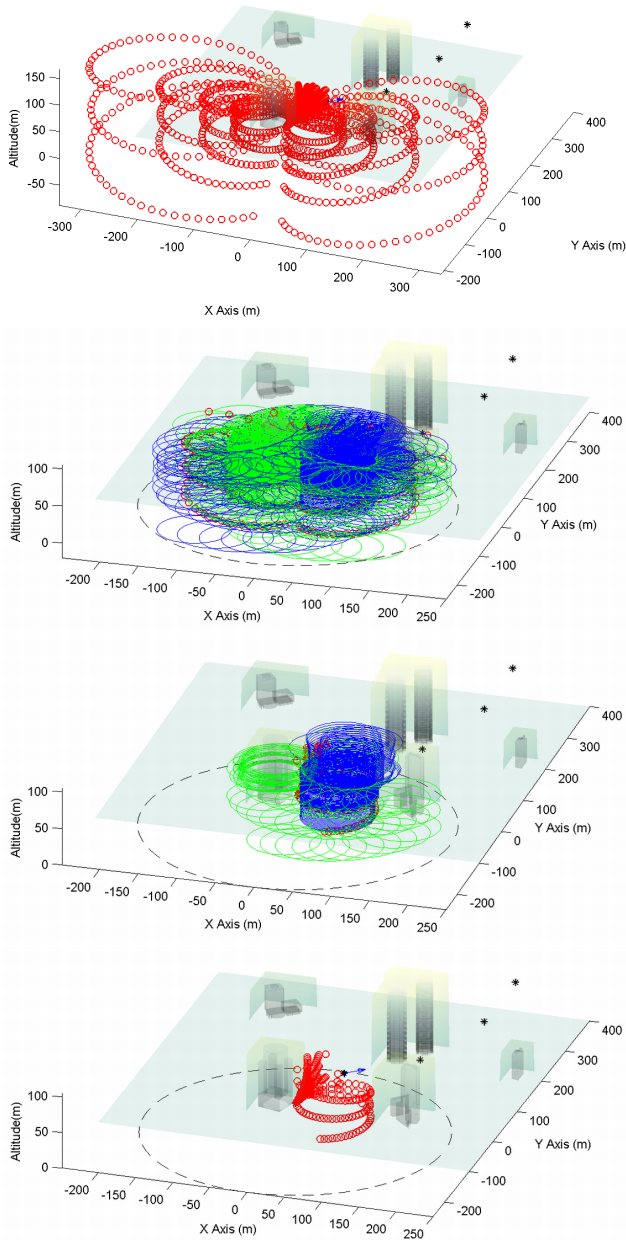


Figure 3. Safe State Formulation for Fixed Wing Aircraft in 3D Partially Known Environments Using a Finite Receding Horizon Strategy

Without inclusion of states which result in loiter maneuvers outside of the finite horizon, it was found

that the minimum finite horizon radius R_c is limited by the aircraft's maneuverability. To ensure platform safety and provide ATC with the authority to veto and safely execute a hold maneuver at anytime, R_c is required to be greater than the minimum platform turn radius (Equation 1).

$$R_{\min} = \frac{u_0^2}{g \tan \phi_{\max}} \quad (1)$$

The following section presents the concatenation of primitives to form a smooth trajectory in an optimized manner. An overview of previous optimization methods applied to trajectories generated using MA theory is presented.

Trajectory Optimization

MA theory [11] like other geometric trajectory generation methods [8] requires the sequential selection of individual segments to form a smooth continuous trajectory. The final trajectory is formed through concatenation of a set of selected trim primitives (and corresponding maneuver primitives, if required) where each trim primitive selected for execution can be considered as a stage.

Without an optimization strategy in place, there is no guarantee that trajectories generated will meet HDM and mission criteria, thus an optimization strategy is required in order to generate trajectories which best meet one or more mission objectives. Dynamic Programming (DP) [18] has been previously employed in related research [10, 13] for the optimization of feasible trajectories generated through the application of MA theory. DP is a sequential optimization method, which finds the least cost solution (optimum) from a set of alternatives over one or more stages.

In comparison to the application of DP to trajectory planning with respect to a generic graph search implementation [19] \mathbf{p} can be treated as the current node. Each possible state the platform can reach through the execution of currently stored trim primitives must be treated as a neighboring node. Expanding each neighboring node would cause the algorithm to grow exponentially in computational complexity for each additional stage considered in the overall optimization process.

MA theory is a method for the discretization of platform dynamics rather than planning space; an accurate representation of platform motion requires a higher automaton resolution resulting in a larger number of neighboring nodes. Therefore, the use of DP in autonomous motion planning can become very computationally expensive as the number of stages in the optimization process increases.

The DP formulation requires selection of the optimal maneuver and corresponding jump time from a predefined set of maneuvers. This paper uses the DP search algorithm but limits the search to single stage optimization. Optimization using DP search over one stage involves selection of the optimal trim primitive (q) and corresponding jump time or primitive duration (τ_q) from a predefined set of trim primitives (\mathbf{q}) and possible jump times for each stage in an iterative manner.

$$\text{Let } \tau_q = \{ \tau_{q,1}, \tau_{q,2}, \dots, \tau_{q,i} \} \in [0, t_f - t_i]$$

represent the discrete possible set of possible jump times for each q . Each discrete τ_q , for a given q , can be represented as a unique decision alternative, as it will result in a different final state if executed. Let A be the set of such alternatives. The optimal maneuver and corresponding jump time (i.e. the most desirable alternative) will have the least cost, where the cost is an aggregated value representing the desirability of a given alternative with respect to given criteria.

Frazzoli applies two specific criteria during maneuver and jump time selection: minimizing euclidean distance of current platform (\mathbf{p}) and goal (\mathbf{g}) positions (criterion $\text{crit}_{\|\mathbf{g}-\mathbf{p}\|}$); and minimizing the difference between the platform heading (ψ_p) and desired heading angle at goal (ψ_g) (criterion $\text{crit}_{|\Delta\psi|}$) during maneuver selection. This research applies both criteria during the selection of the optimal maneuver and jump time selection with equal weighting before aggregating using a weighted sum decision optimization [19] For a more detailed overview of criteria representation applied to MA theory, refer to [14].

For operations in environments where planning must be performed within a limited time frame; the onboard processing capabilities may not be sufficient

to generate a feasible optimized collision free trajectory plan before the imposed deadline for a fixed automaton size.

Frazzoli [10] states, “The selection of the trajectory primitives is currently done manually: it would be desirable to obtain formal criteria defining the ‘optimal’ choice of primitives, trading off the complexity of the resulting automaton with the achievable performance. A dynamic resizing of the automaton is also conceivable: in critical situations, when a decision has to be taken in a very short time, the automaton could be reduced to a few maneuvers, whereas in a more secure situation the set of possible maneuvers could be expanded”. To the authors best knowledge, no research has been currently published which explores the inclusion of methodologies to automate automaton resizing for MA based trajectory planning with respect to available computational power available for fixed wing aircraft.

Real-Time Planning Capability

In the presence of real time deadlines, there is a finite length of time available (Finite Planning Window) for the FMS to complete the trajectory solution search before a predefined safety maneuver must be executed to ensure collision free flight. The author has previously proposed a novel Computationally Adaptive Trajectory Decision System (CATDS) for the dynamic selection of optimum automaton sizes, allowing for the computation and optimization of feasible flight trajectories within real-time planning deadlines [20].

[16] states that with increasing levels autonomy onboard UA, operators move away from direct control of the platform towards a management by exception control paradigm. Management by exception would take place onboard autonomous commercial aircraft as the proposed FMS is expected to perform planning and execution and then inform ATC of its current and future actions if real-time constraints are present.

The ATC operator must have sufficient time (FPW_{min}) to veto or override the current plans and revert to a lower control paradigm if required. Without consideration for future stages, HDM may have insufficient time to veto the current trajectory segment being traversed if the time to traverse the

following trajectory segment ($\tau_{q_{\min,k+1}}$) is less than minimum planning window specified for the current segment ($FPW_{\min,k}$). The following section presents the inclusion of the FPW_{\min} constraint within the trajectory-planning layer of the FMS.

Inclusion of Minimum FPW Constraint

An FPW_{\min} constraint can be implemented within the trajectory planner by either; applying a minimum trajectory segment execution time constraint ($\tau_{q_{\min}}$), or starting the computation for the following trajectory segment before the aircraft traverses the current segment. Applying a $\tau_{q_{\min}}$ constraint reduces the number of feasible alternatives available to the planner and may force the selection of sub-optimal trajectory segments. Thus, an FPW_{\min} was applied by starting the computation for the following segment (q_{k+1}) before current trajectory segment had been completely traversed.

The algorithm uses a buffer (t_b) to ensure that enough time is available during computation of the current segment to generate the following stage with an FPW of FPW_{\min} (Equation 2). This results in more conservative use of the available FPW during planning to ensure that the FPW_{\min} constraint can be met. It must be noted that the trajectory planner must also account for the change in the location of the finite horizon centre during simulation (Equation 3). Figure 4 presents a visual representation of the inclusion of the FPW_{\min} constraint applied to a partially known environment problem. Determining t_B where t_{bc} is a matrix of time required to compute automata of varying resolutions offline.

$$t_B = FPW_{\min} - \tau_{q_{\min}} \quad (2)$$

$$\tau_{q_{\max,k+1}} = \max_q (t_{bc} < (FPW_k - t_B))$$

Change in finite horizon centre (C_{k+1}) where p_c is the platform position on q_k with a jump time of τ_c

$$\tau_c = FPW_{\min} - \tau_{q_{\max,k+1}} \quad (3)$$

$$C_{k+1} = p_c$$

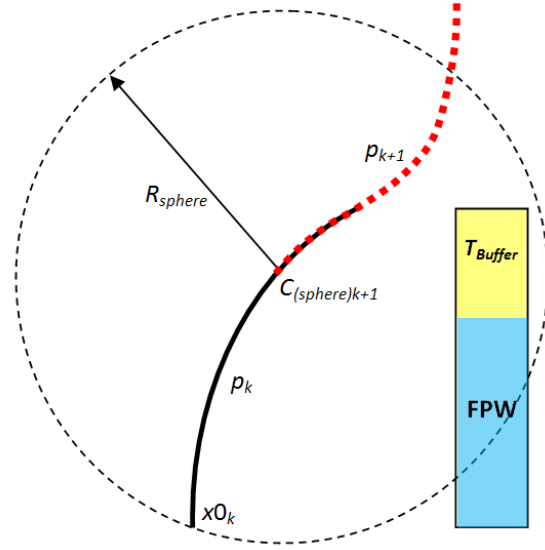


Figure 4. FPW_{\min} Constraint

The following section presents simulation results, which examine real-time trajectories generated by the proposed FMS with the inclusion of an FPW_{\min} constraint.

Results

Simulated 3D environments were constructed in MATLAB to represent low altitude urban terrain Figure 5. The aircraft is tasked with generating an optimized feasible collision free trajectory through a set of predefined waypoints, where each waypoint is represented by a spherical capture region and ψ_g (Figure 6). The simulations have been performed on System 1 (Intel Core2 CPU at 3GHz) and System 2 (Core2 CPU simulated at 1.5GHz).

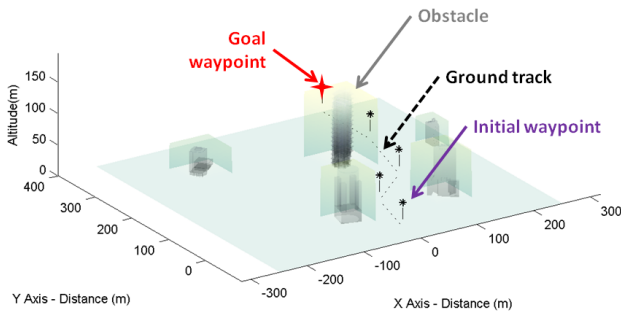


Figure 5. Simulated 3D Environment Representing Low Altitude Urban Terrain

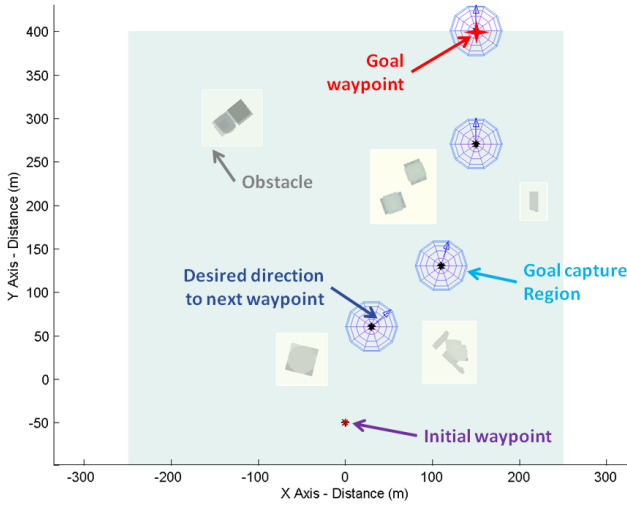


Figure 6. Waypoint Capture Represented as Spherical Regions

The main system execution parameter is the total number of alternatives included in the decision process. The number of alternatives is dependent on the platform ψ_p , ϕ_p and the number of discrete τ_q samples representing each q selected. The ψ_p and ϕ_p resolutions indicate the sampling distance (in degrees) between trim maneuvers (Table 1).

Table 1. A (i.e Automaton Sizes) Applied during Simulation

A	ψ_p res. (deg)	ϕ_p res. (deg)	q resolution
1350	15.74	7.48	50
1950	10.49	7.48	50
5700	6.996	7.48	100
9100	10.49	2.49	100
13300	6.996	2.49	100
41800	6.996	1.49	200

The simulations are conducted in partially known environments where the known environment abstraction is represented as a finite horizon scheme. During planning in partially known environments, the planner has limited knowledge of the planning environment. This places a limit on the length of the primitives, which can be executed, as platform safety cannot be guaranteed if the selected maneuver state is outside the environment abstraction available.

Segment FPW

Table 2 presents the real-time trajectory generation results for the simulated urban scenario (Figure 5) with the inclusion of FPW_{min} constraints of 1 and 2 seconds. Figure 7 details segment computation times for an FPW_{min} of 2 seconds on system 1.

Table 2. Computation Results with FPW_{min} Constraint for System 1

FPW_{min} (s)	A Res. [Min Max]	Min. stage Compute time (s)	Time to Plan (s)	Flight Time (s)
1	[600 5320]	0.53	14.22	19.75
2	[600 3420]	0.53	8.5	19.91

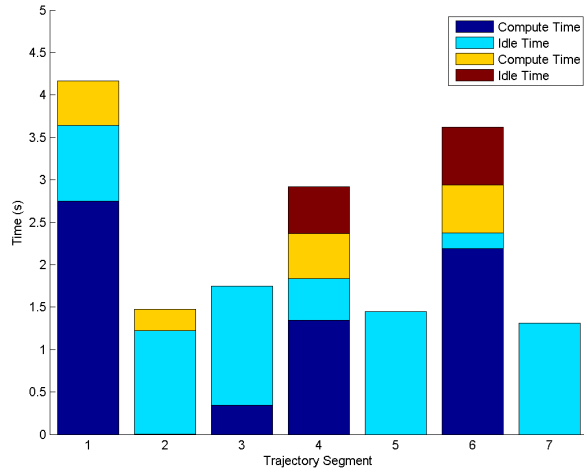


Figure 7. Segment FPW with $FPW_{min} = 2s$ on System 1

The inclusion of an FPW_{min} constraint whilst the ensured that the trajectory planner took $\tau_{q_{k+1}}$ into account to provide the supervisory ATC with sufficient time to veto if they did not agree with the current trajectory segment selected by the onboard FMS. However, it was also found that the inclusion of a longer FPW_{min} of 2 seconds required a longer t_b resulting in a more conservative approach to the allocation of segment computation time (i.e. more idle time).

The following section presents Monte-Carlo simulation results for further analysis the performance of the proposed FMS to generate optimized trajectories for a wide range of possible paths (or waypoint sets).

Monte-Carlo Simulation

The simulation uses nine starting locations with three possible altitudes and a depth of three waypoints resulting in a total of 263 waypoint set permutations (Figure 8).

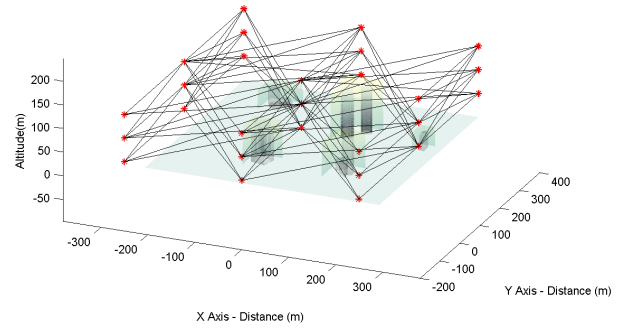


Figure 8. Waypoint Sets Applied during Monte-Carlo Simulation

The FMS was tasked with generating feasible collision free trajectories through all waypoints sets whilst meeting real time planning requirements. The results including Alternative (A) resolution and mean (μ) and standard deviation (σ) data for the FPW are presented in (Table 3).

Table 3. Monte-Carlo Simulation Results

FPW_{min} (s)	CPU (GHz)	A Res. [μ ; Max]	FPW [μ ; σ ; min] (s)
2	3	[2670; 20900]	[2.05; 0.26; 1.6]
2	1.5	[1382; 13300]	[1.91; 0.32; 1.1]
1	3	[4209; 20900]	[1.24; 0.28; 0.6]
1	1.5	[2242; 9100]	[1.20; 0.10; 0.4]

The monte-carlo simulation demonstrated that the FMS was capable of generating optimized, feasible, collision free trajectories across a wide range of waypoint sets with real-time constraints present. However, whilst the average FPW experienced during the mone-carlo simulation set is relatively close to the desired FPW_{min} there is some variance which means that the minimum time available to ATC veto certain segments is less than the FPW_{min} constraint. Thus it is desirable for onboard systems to have greater computational capabilities in conjunction with a higher FPW_{min} constraint.

Conclusions

This paper presented a new approach to assist in the onboard passenger safety of commercial aircraft through the implementation of a novel FMS to enable autonomous flight in the event of an emergency

scenario. Feasible trajectories were generated by concatenating common fixed wing aircraft flight maneuvers formed through the application of MA theory. Aircraft safety was demonstrated through the application of safe state theory to 3D partially known environments.

The MA based automaton resolution was dynamically adjusted to ensure that a feasible solution could be found within the available FPW. Furthermore, it was demonstrated through monte-carlo simulation that the FMS was capable of generating optimised, feasible, collision free trajectories within real-time constraints for a wide variety of waypoint sets.

Finally, the inclusion of a minimum FPW within the FMS provided a minimum period of time to veto onboard decisions and potentially allows the aircraft operate at a higher level of autonomy during the emergency scenario using a management by exception paradigm in the NAS.

References

- [1] Japan Aircraft Development Corporation. (2013). Worldwide Market Forecast 2013-2013. Japan Aircraft Development Corporation.
- [2] Blalock, G., Kadiyali, V., & Simon, D. H. (2007). The Impact of Post-9/11 Airport Security Measures on the Demand for Air Travel. *Journal of Law and Economics*, 50 (4), 731-755.
- [3] Narayan, P., Wu, P., Campbell, D., & Walker, R. (2007). An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Airspace System (NAS). 2nd International Unmanned Air Vehicle Systems Conference.
- [4] Applications of Intelligent Control. Boston: Kluwer Academic.
- [5] Freed, M., Bonasso, P., & Dalal, K. (2005). An Architecture for Intelligent Management of Aerial Observation Missions. AIAA Aerospace Conference. Virginia: AIAA.
- [6] Boskovic, J., Prasanth, R., & and Mehra, R. (2002). A Multilayer Control Architecture for Unmanned Aerial Vehicles. American Control Conference, 3, pp. 1825-1830. Cincinnati.
- [7] Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., & N., a. K. (2003). Evolutionary Algorithm Based Offline/Online Path Planner for Uav Navigation. *IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics*, 33 (6), 898-912.
- [8] Dubins, L. (1957). On Curves of Minimal Length with a Constraint on Average Curvature with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79, 497-516.
- [9] Anderson, E., Beard, R., & and McLain, T. (2005). Real-Time Dynamic Trajectory Smoothing for Unmanned Air Vehicles. *IEEE Transactions on Control Systems Technology*, 13 (3), 471-477.
- [10] Frazzoli, E., Dahleh, M., & and Feron, E. (1999). A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopters. 38th IEEE Conference on Decision and Control. 3, pp. 2471-2476. IEEE.
- [11] Frazzoli, E., Dahleh, M., & and Feron, E. (2005). Maneuver-Based Motion Planning for Nonlinear Systems with Symmetries. *IEEE Transactions on Robotics and Automation*, 21 (6), 1077-1091.
- [12] Bottasso, L., L. D., & and Savini, B. (2007). Path Planning for Autonomous Vehicles by Trajectory Smoothing Using Motion Primitives. *IEEE Transactions on Control Systems Technology*
- [13] Schouwenaars, T., Mettler, B., Feron, E., & and How, J. (2003). Robust Motion Planning Using a Maneuver Automaton with Built-in Uncertainties. AIAA Aerospace Sciences and Exhibit.
- [14] Narayan, P. P., Meyer, P., & and Campbell, D. A. (2013). Embedding Human Expert Cognition into Autonomous UAS Trajectory Planning. *IEEE Transactions on Systems and Cybernetics: part b*, 43 (2), 530-543.
- [15] Schouwenaars, T., How, J., & Feron, E. (2004). Receding Horizon Path Planning with Implicit Safety Guarantees. *American Control Conference*, 6, pp. 5576-5581.
- [16] Franke, J., Zaychik, V., Spura, T., & and Alves, E. (2005). Inverting the Operator/Vehicle Ratio:

Approaches to Next Generation Uav Command and Control. Unmanned Systems North America.

[17] Anisi, D., Robinson, J., & Gren, P. (2006). On-Line Trajectory Planning for Aerial Vehicles: a Safe Approach with Guaranteed Task Completion. AIAA Guidance, Navigation, and Control Conference and Exhibit . AIAA.

[18] Bellman, R. (1952). On the Theory of Dynamic Programming. National Academy of Sciences , 38, 716.

[19] Lavalle, P. (2006). Planning Algorithms. New York: Cambridge University Press.

[20] Narayan, P. P., Campbell, D. A., & Walker, R. A. (2008). Computationally Adaptive Multi-Objective Trajectory Optimization for Uas with

Variable Planning Deadlines. IEEE Aerospace Conference. Big Sky, Montana.

Acknowledgements

The authors would like to thank and acknowledge the support of the University of the West of England (UWE) throughout this research project. This project was funded by UWE's Early Career Research Starter Grant scheme (SPUR4).

32nd Digital Avionics Systems Conference

October 6-10, 2013