Computationally Adaptive Multi-Objective Trajectory Optimization for UAS with Variable Planning Deadlines

Pritesh Narayan*, Duncan Campbell, Rodney Walker
*PhD Candidate
Queensland University of Technology
2 George Street, Brisbane
p.narayan@qut.edu.au

Abstract—This paper presents a new approach which allows for the computation and optimization of feasible 3D flight trajectories within real time planning deadlines, for Unmanned Aerial Systems (UAS) operating in environments with obstacles present. Sets of candidate flight trajectories have been generated through the application of maneuver automaton theory, where smooth trajectories are formed via the concatenation of predefined trim and maneuver primitives; generated using aircraft dynamic models. During typical UAS operations, multiple objectives may exist, therefore the use of multi-objective optimization can potentially allow for convergence to a solution which better reflects overall mission requirements. Multiple objective optimization of trajectories has been implemented through weighted sum aggregation. However, real-time planning constraints may be imposed on the multi-objective optimization process due to the existence of obstacles in the immediate path. Thus, a novel Computationally Adaptive Trajectory Decision (CATD) optimization system has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters to ensure that the trajectory solution search can generate a feasible solution, if one exists, within a given length of time. The inclusion of the CATD potentially increases overall mission efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities. This approach has been demonstrated in this paper through simulation using a fixed wing UAS operating in low altitude environments with obstacles present.[1][2]

TABLE OF CONTENTS

1——————————
[1]978-1-4244-2622-5/09/$25.00 ©2009 IEEE
[2]IEEEAC paper#1237, Version 5, Updated 2009:1:06

1. INTRODUCTION

Unmanned Aerial Systems (UAS) have been previously employed in a diverse range of military applications including surveillance and strike deployment [1]. With respect to civilian applications, geographically sparse countries, such as Australia have great potential for utilization of UAS in asset management, search and rescue, remote sensing operations and atmospheric observation [2]. In order to realize this potential, seamless operation of UAS within the National Airspace System (NAS) is required [3, 4]; this is a difficult problem.

Operation of UAS in the NAS creates a new set of challenges that are not applicable to many military applications. From a regulatory perspective, UAS need to: (i) demonstrate an Equivalent Level Of Safety (ELOS) to that of a human piloted aircraft, (ii) operate in compliance with existing aviation regulations and (iii) appear transparent to other airspace users [5].

The majority of UAS operations still require human operators to perform mission management and piloting tasks through real time communications links with the unmanned platform. This results in high operator workload and places greater reliance on the communications link. The inclusion of automated planning systems onboard can potentially improve mission efficiency and allow for continued operations in the presence of communications failures. In particular, the automation of global and local path planning components assist in ensuring that the flight occurs in accordance with the rules of the air; a key ELOS requirement.

Local path planning provides a navigation strategy for safe traversal through cluttered environments. The desired track, represented as a collision free flight trajectory, ensures that the platform remains within platform performance bounds. Automating the local path planning process is non-trivial and some challenges include: incorporation of complex platform dynamics, trajectory optimization to meet mission requirements, real-time constraints on computation time imposed by obstacles in the flight path, and the guarantee that generated trajectories are collision free.

During operations, civilian UAS may have multiple objectives to meet. The use of multi-objective optimization allows for the generation of a solution which better reflects the overall mission requirements. Additionally, if operations are undertaken at lower altitudes, the environment may present several challenges not encountered during high altitude flight. Terrain and urban structures become hazards to the safety of the UAS. The proximity of obstacles to the UAS places real-time constraints on the re/planning computation time available.

This paper presents a new framework for the Computationally Adaptive Multi-Objective Flight Management of UAS in civilian environments. An outline of UAS trajectory generation approaches and related work is given in section 2. Section 3 presents an overview of the trajectory optimization process, and section 4 outlines the real-time re/planning requirements of UAS operating in cluttered requirements. Simulation results presented in section 5 demonstrate how the addition of the CATD can allow for the generation of feasible trajectories within given real-time deadlines. Finally, conclusions are presented in section 6.

2. FEASIBLE TRAJECTORY REPRESENTATION

A local path planning process is generally described as a system which generates a smooth trajectory representing the aircraft track through a set of mission level waypoints; typically generated by a global planner. The trajectory generated is required to be feasible and collision free to ensure that UAS flight track is safe and within platform performance bounds.

UAS Platform Constraints

The inclusion of vehicle dynamics during the trajectory planning process, allows for the generation of flight trajectories which take platform constraints into account. Vehicle dynamics are used to calculate the performance envelope which the aircraft must remain within to ensure that platform does not operate outside performance bounds. In the presence of a Stability Augmentation System (SAS) onboard, trajectories which do not consider platform performance bounds may lead to poor tracking.

Flight Trajectory Representation

Flight trajectories are generally represented through the use of either spline based or geometric approximations. Polynomial or spline based techniques [6, 7] place control points in a particular order to generate the desired trajectory. Geometric based techniques require the concatenation of aircraft flight maneuvers to form a smooth flight track [8-11]. However, these flight maneuvers are usually limited to cruise and constant radius turns and roll/yaw coupling effects are not considered; an essential flight characteristic of fixed wing platforms.

During the execution of a constant radius turn for a fixed wing aircraft, the consideration of roll/yaw coupling allows for the inclusion of platform roll rate as a component of the overall aircraft performance envelope. However, this requires the additional tracking of the platform attitude (roll component) during the trajectory planning process. One candidate method which allows for the inclusion of roll rate performance bounds is maneuver automaton theory.

Maneuver Automaton Theory

Maneuver Automaton (MA) theory, proposed by Frazzoli et al. [12, 13] can be used in the generation of feasible flight trajectories through the sequential concatenation of predefined motion primitives (Figure 1). MA employs two types of primitives: trims and maneuvers. Trim primitives represent the vehicle during a state of equilibrium whilst maneuver primitives characterize the vehicle operating outside a state of equilibrium. Primitives are generated using a dynamic model of the vehicle, thus platform stability can be implicitly guaranteed through generation of primitives which ensure that the vehicle remains within performance bounds.

Trajectory Representation Implementation

For this paper, MA theory is used to describe a time-invariant non-linear, dynamical system S , described as a set of ordinary differential equations (ODE) as:

$$\dot{x}(t) := \frac{d}{dt} x(t) = f(x(t), u(t)) \tag{1}$$

Where u is the control input (execution time, maneuver type) = $\{\tau, primitive\}$ and x is the state vector.

Trim Primitive Representation

Trim Primitives represent the UAS platform operating in a state of equilibrium. Using MA theory, trim primitives can be generated by placing the body fixed roll $(\dot{\phi})$ and pitch $(\dot{\theta})$ rates to zero and maintaining a constant velocity (V), roll (ϕ) and pitch (θ) angle for the duration (τ_q) of the primitive execution.

Trim primitives were generated using a 6 Degree of Freedom (DOF) flight dynamics model based on the Aerosonde UAS platform data set available in the Aerosim Blockset [14]. Six predefined trim primitives have been implemented in simulation including: cruise, coordinated turn, climb, descent, helical climb and helical descent.

The initial platform state $x(t_i) = x_i$ reaches a final state $x(t_f) = x_f$ due to the execution of a given trim primitive (q); this can be represented as:

$$x_f = x_i + \tau_q \dot{x}_q$$
$$t_f = t_i + \tau_q \qquad (2)$$

Where $\{V, \phi, \theta\}$ are constants and $\{\dot{\phi}, \dot{\theta}\} = \{0,0\}$

It is of importance to note, that for a platform to enter a state of equilibrium (execution of a trim primitive), the initial platform attitude must equal the attitude requirements of the trim primitive to be executed; $\{\phi, \theta\}_i = \{\phi, \theta\}_q$. If the initial platform attitude does not equal the attitude required to execute the given trim primitive, a maneuver primitive must be inserted to ensure that body fixed attitude rate constraints are included within performance bounds.

Maneuver Primitive Representation

During the execution of a maneuver primitive, the UAS does not have to remain in a state of equilibrium. For a fixed wing platform, the body fixed attitude rate constraint becomes $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{\max}, \dot{\theta}_{\max}\}$. In this paper, maneuver primitives (p) are employed to connect two trim primitives, if required, in the formation of feasible trajectories. This allows for the consideration of attitude rates as an additional platform constraint during periods where the UAS is not in a state of equilibrium, where $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$.

If $\{\phi, \theta\}_i \neq \{\phi, \theta\}_q$, the UAS platform dynamic model is propagated until the platform reaches the desired state configuration $\{\phi, \theta\}_i = \{\phi, \theta\}_q$ making the execution of the next trim primitive feasible.

While $\{\phi, \theta\}_i = \{\phi, \theta\}_q$

$$x_{k+1} = x_k + \dot{x}_p \Delta T$$
$$t_{k+1} = t_k + \Delta t \qquad (3)$$

Where $\{\dot{\phi}, \dot{\theta}\} = \{\dot{\phi}_{\max}, \dot{\theta}_{\max}\}$

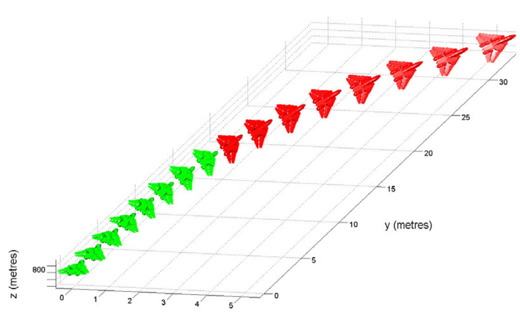

Figure 1 – Visual Representation of Trim and Maneuver Primitive Concatenation (Coordinated Turn)

Generating Collision Free Trajectories

Safe UAS operation in cluttered environments requires the generation of collision free trajectories. This has been accomplished through the inclusion of collision detection algorithms. The transition maneuver must be deemed collision free before collision detection along the maneuver primitive takes place. Due to the sequential nature of maneuver concatenation, a collision free candidate trajectory does not guarantee vehicle safety during the next maneuver. Safe state maneuvers [15] are executed at each sampled point along the candidate flight mode and then tested for collisions. This ensures that the UAS can enter a safe state if no collision free trajectory is determined during the optimization of the following stage (Figure 2).

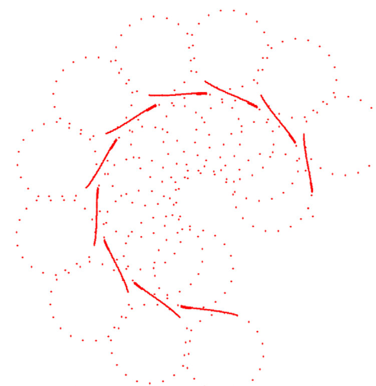

Figure 2 – Safe States Generated for a Candidate Coordinated Turn Trim Primitive

3. Trajectory Optimization

Dynamic programming (DP) [16] has been previously employed in related research [17, 18] for the optimization of feasible trajectories which have been generated using MA theory. DP is a sequential optimization process where each trim primitive selected for execution can be considered as a stage. Thus the final trajectory is formed through sequential concatenation of a set of selected trim primitives (and corresponding maneuver primitives, if required) for all stages used in the computation.

DP is a very computationally expensive algorithm for the motion planning application. In comparison to the application of DP to trajectory planning with respect to a generic graph search implementation, the current UAS platform position 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 neighboring nodes. Expanding each neighboring node would cause the algorithm to grow exponentially in computational complexity for each additional stage considered in the overall optimization process. Due to the inclusion of maneuver primitives, it is difficult to calculate how many stages are required before a solution is found (if one exists).

In a typical UAS scenario, constant trajectory replanning maybe required if operations take place in partially known environments (e.g. active onboard sensing is predominantly used for navigation). To decrease the computational complexity and resulting time to plan during DP optimization over multiple stages, hybrid architectures involving DP with Rapidly Exploring Random Trees (RRT) [12] and DP with Model Predictive Control (MPC) [19] have been implemented.

The research presented in this paper uses the DP search algorithm but limits the search to single stage optimization. This converts the DP algorithm to a greedy search implementation, which essentially chooses the most optimal trim primitive, trim execution time and maneuver execution time required to execute the optimal trim primitive for a single stage. The UAS position after execution of the optimal trim primitive is taken as the next node for expansion, and continues until the goal is reached is reached (Figure 3).

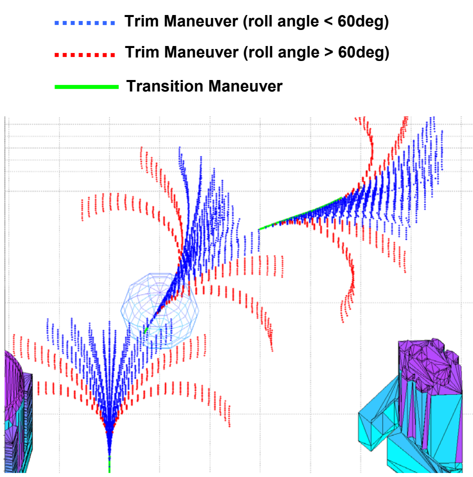

Figure 3 - Greedy Search Algorithm Implementation

Executing a DP search algorithm iteratively over each stage significantly decreases search time. However, not considering all stages during the optimization process means that global solution optimality and completeness cannot be guaranteed. Additionally, this may lead to scenarios where the platform becomes trapped in local minima. UAS motion planning in 3D space has the advantage for allowing the execution of certain motion primitives (e.g. helical ascent) to escape local minima and continue operations [20]. In addition, during operations in dynamic and partially known environments, a greedy motion planning implementation can suffice as it may not be possible to find a global solution (e.g. due to limited environment representation). Furthermore searching for a globally optimal solution may be infeasible as there can be real-time constraints placed on the finite replanning time available.

Multi-Objective Optimization Process

During operations, civilian UAS may have multiple objectives to meet including platform safety; successful completion of the mission; minimizing fuel, time, and/or distance; or minimizing deviation from the current path. The use of multi-objective optimization allows for the generation of a solution which may better reflect the overall requirements of the mission. For example, by placing greater emphasis on safety, operations in populated environments may benefit from the inclusion of additional objectives which minimize platform control loss.

During each stage, the utility value is calculated using a weighted sum aggregation for all feasible trim primitives. The objectives included in the optimization process are, minimization of distance to goal and minimization of vehicle heading with respect to goal. Two additional objectives have been included to generate trajectories which are less likely to lead to loss of platform control. These objectives include: minimizing wing loading; and minimizing the transition length required to execute next flight mode. The optimal solution for each stage is the trim primitive with the highest aggregated weighted sum value.

$$\mu_T = \sum_{i=1}^{n} w_i \mu_i \qquad (4)$$

Where μ_T is the total utility value, w_i is the objective weighting and μ_i is the objective utility value.

The following section provides an overview of real time considerations during the optimization process.

4. Real Time Optimization

In the presence of real time deadlines, there is a finite length of time available (Finite Planning Window) for the UAS to complete the trajectory solution search before a predefined safety maneuver must be executed to ensure collision free flight. Convergence to a solution, if one exists, within this Finite Planning Window (FPW) is dependent on current system execution parameters and computational power available.

The time required to perform an optimal trajectory solution search during maneuver generation is dependent on system execution parameters such as search resolution (number of primitives available); maneuver resolution (number of points representing primitive). Scenarios may occur where a feasible solution cannot be generated within the FPW if the search and resolution settings are too great. Consequently, solution completeness may be further diminished if the settings are too low.

A novel Computationally Adaptive Trajectory Decision (CATD) optimization system has been developed and implemented in simulation to dynamically manage, calculate and schedule system execution parameters. This ensures that the trajectory generator can complete the trajectory solution search and generate a feasible solution, if one exists, within the FPW.

CATD is an expert system which composed of two components. The offline component benchmarks the computational performance of the system using sets of predefined execution parameters. The computational performance can be estimated as the algorithm is deterministic in nature. However, the offline component must be re-executed if the computation capabilities of the system are modified.

The online component dynamically computes the most optimum set of execution parameters with respect to the available computational power and FPW. Multi-objective theory is used to find a best compromise solution where the conflicting objectives are maximization of search and resolution and minimization of search time.

The inclusion of the CATD potentially increases overall mission efficiency and may allow for the implementation of the system on different UAS platforms with varying onboard computational capabilities. The following section presents the results for the generation of feasible trajectories with the CATD both enabled and disabled. A 3D environment representation was setup in MATLAB the UAS assignment included safe and efficient navigation through a set of mission level waypoints.

5. RESULTS

During the simulation the platform operates at a constant velocity of 30 m/s. The simulation has been performed on a computer with an Intel Core 2 quad core processor operating at 3.4GHz to simulate the how the inclusion of the CATD can allow for the generation of feasible trajectories within a given FPW. The FPW is calculated as the time taken to complete the current stage. The FPW value is has a maximum value of ranging from 3 to 5 seconds to simulate a finite horizon (FH) between 90 and 150m

Simulated Results – CATD Not Enabled

The first set of results show the algorithms performance without the CATD enabled for each computing setup. The maneuver generation algorithm finds a feasible solution (Figure 4 and Figure 5) using a predefined set of maneuver and search resolution parameters (Table 1).

Table 1 – Algorithm Run Time: CATD Not Enabled

FH (m)	Maneuver Resolution	Search Resolution	Avg. Utility	Min. FPW (s)
90	80	89	0.52	-0.7
120	80	89	0.52	0.1
150	80	89	0.52	1.2

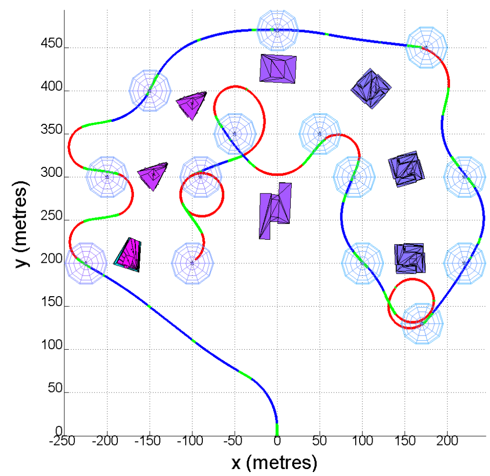

Figure 4 –Top View of Trajectory

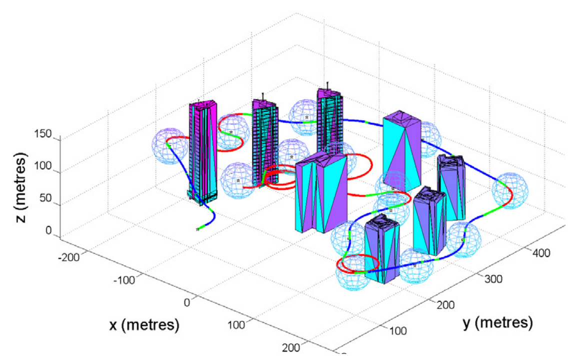

Figure 5 – 3D View of Trajectory

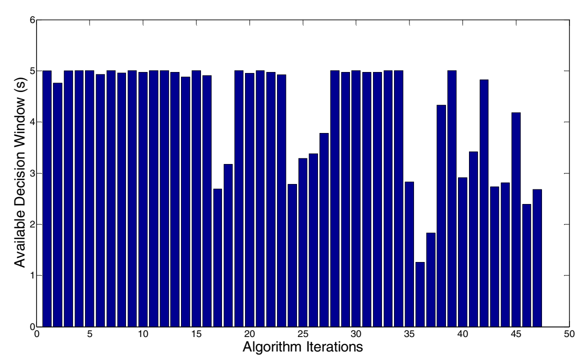

Figure 6 – FPW per Iteration (FH = 150m)

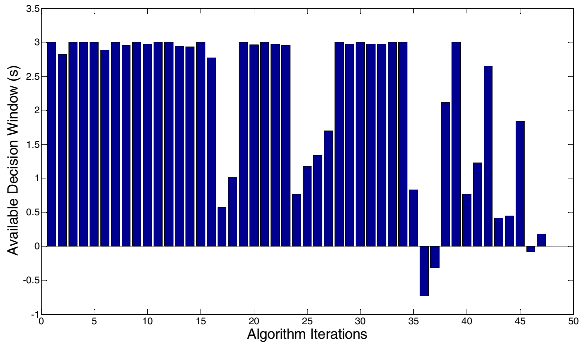

Figure 7 - FPW per Iteration (FH = 90m)

Without the CATD enabled, there is not guarantee that feasible trajectories will be generated within a given FPW. Using predefined search and maneuver resolution parameters may use of the computation time available inefficiently in scenarios where the FH is relatively large (Figure 6). In scenarios, where the given FH is shorter (Figure 7), the platform may not be able to compute a feasible solution within the available FPW.

Simulated Results – CATD Enabled

Enabling the CATD dynamically adjusts the maneuver and search resolutions with respect to the available FPW. Table 2 presents the results for the simulated results with the CATD Enabled.

Table 2 - Algorithm Run Time - CATD Enabled

FH (m)	Maneuver Resolution	Search Resolution	Avg. Utility	Min. FPW (s)
90	Dynamic (Figure 10)	Dynamic (Figure 11)	0.93	1.6
120	Dynamic	Dynamic	0.93	0.3
150	Dynamic (Figure 14)	Dynamic (Figure 15)	0.9	0.5

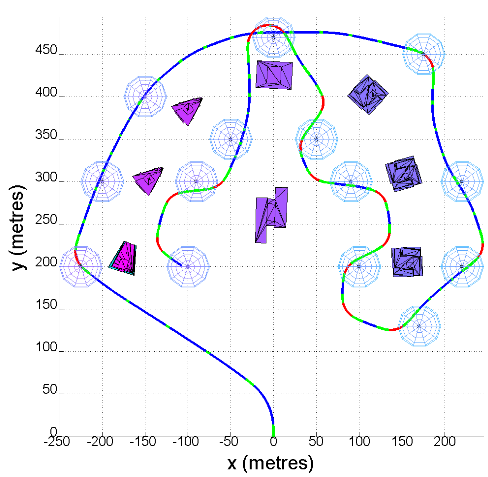

Figure 8 –Top View of Trajectory (FH = 150m)

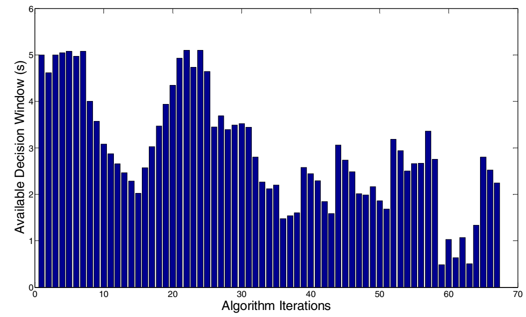

Figure 9 - FPW per Iteration (FH = 150m)

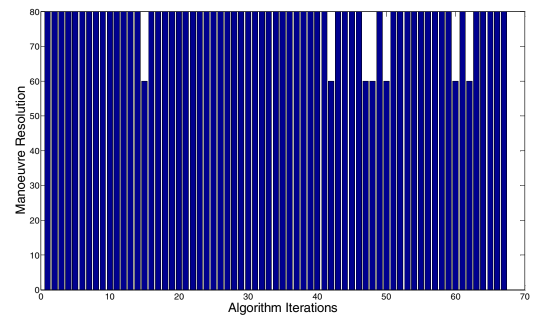

Figure 10 - Maneuver Resolution (FH = 150m)

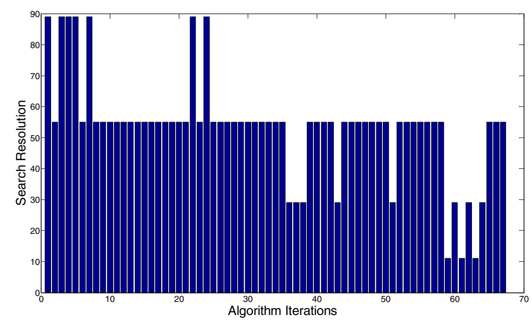

Figure 11 - Search Resolution (FH = 150m)

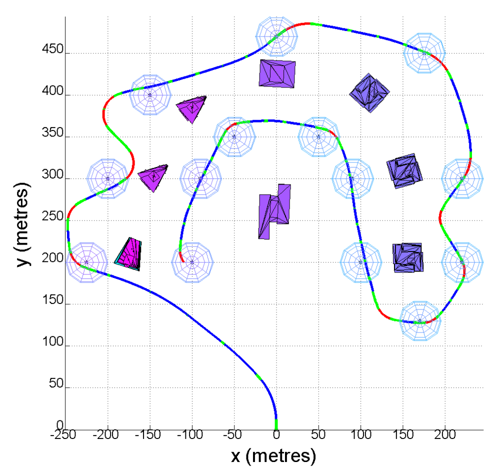

Figure 12 -Top View of Trajectory (FH = 90m)

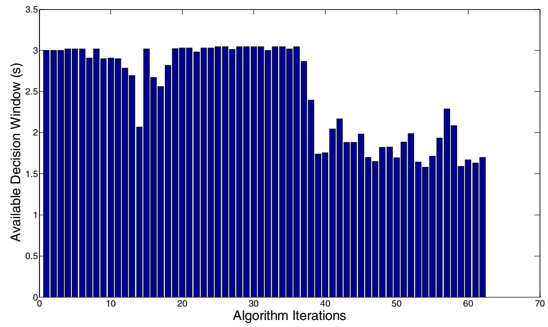

Figure 13 - FPW per Iteration (FH =90m)

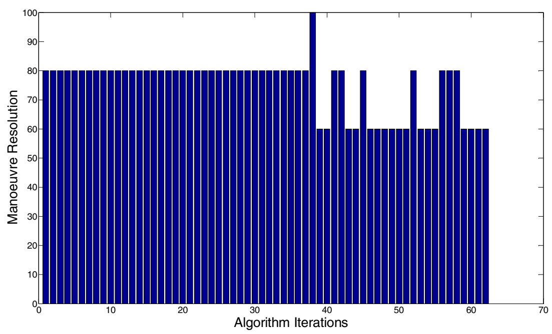

Figure 14 - Maneuver Resolution (FH = 90m)

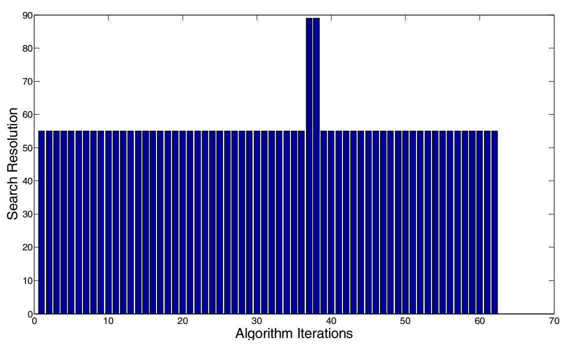

Figure 15 - Search Resolution (FH = 90m)

The inclusion of the CADT ensures that a feasible solution is generated within the given FPW. By dynamically adjusting the search and maneuver resolution parameters, the system compromises search completeness for time required to generate a solution. However, systems with greater onboard computational capabilities and/or longer FH (simulating onboard sensors) (Figure 9), benefit from the ability to complete a search at a higher resolutions. Systems without lower computational resources and/or Shorter FH can continue to generate feasible trajectory solutions (Figure 13) within the given FPW. However, this requires the search to be conducted at lower resolutions.

6. CONCLUSIONS

This paper has presented a new framework which allows for the computation and optimization of feasible 3D flight trajectories within real time planning deadlines, for UAS operations in cluttered environments. A novel real time flight management subsystem (CATD) was implemented to dynamically adjust maneuver and search resolution parameters to ensure that a feasible trajectory solution could be generated (if one existed) within a given FPW.

The inclusion of the CATD coupled to a multi-objective maneuver automaton based trajectory planner can potentially allow for more efficient use of the computational time available. Additionally, the utilization the offline component of the CATD to evaluate the performance of a given system, may potentially allow for the implementation of CATD on different platforms with varying onboard computational capabilities and Finite Planning Windows.

ACKNOWLEDGEMENTS

The author would like to thank and acknowledge the support of the Australian Research Centre for Aerospace Automation (ARCAA) and the Queensland University of Technology (QUT) throughout this research project.

REFERENCES

[1] "Unmanned Aircraft Systems Roadmap," Secretary of Defense 2005.

[2] S. Wegener, "UAV Autonomous Operations for airborne Science Missions," American Institute of Aeronautics and Astronautics 2004.

[3] "Advisory Circular - Unmanned Aerial Vehicle Operations, Design Specification, Maintenance and Training of Human Resources," Civil Aviation Authority of Australia July 2002.

[4] "EUROCONTROL Specifications For The Use Of Military Unmanned Aerial Vehicles As Operational Air Traffic Outside Segregated Airspace," EUROCONTROL 25 April 2006.

[5] M. T. DeGarmo, "Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace," MITRE, Center for Advanced Aviation System Development, McLean, Virginia 2004.

[6] K. B. Judd and T. W. McLain, "Spline based path planning for unmanned air vehicles " in *AIAA Guidance, Navigation, and Control Conference and Exhibit,*. vol. AIAA-2001-4238 Montreal, Canada, 2001

[7] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 33, pp. 898-912, Dec 2003.

[8] E. P. Anderson, R. W. Beard, and T. W. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *IEEE Transactions on Control Systems Technology,* vol. 13, pp. 471-477, 05 Jan 2005.

[9] E. Gagnon, C. A. Rabbath, and M. Lauzon, "Heading and position receding horizon control for trajectory generation," in *The Proceedings of the American Control Conference*, 2005, pp. 134-139.

[10] Y. Kuwata, "Real-time Trajectory Design for Unmanned Aerial Vehicles using Receding Horizon Control," in *Aeronautics and Astronautics* Massachusetts: Massachusetts Institute of Technology, 2003.

[11] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *21st Digital Avionics Systems Conference. Proceedings (Cat. No.02CH37325)*, vol.2 ed USA Piscataway, NJ: IEEE, 2002, pp. 8D2-1-12.

[12] E. Frazzoli, M. A. Dahleh, and E. Feron, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999, pp. 2471-2476 vol.3.

[13] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on],* vol. 21, pp. 1077-1091, 2005.

[14] "Aerosim Blockset," Unmanned Dynamics, 2003.

[15] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the American Control Conference*, 2004, pp. 5576-5581 vol.6.

[16] S. M. LaValle, *Planning Algorithms*. New York: Cambridge University Press, 2006.

[17] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Robust Motion Planning Using a Maneuver Automaton with Built-In Uncertainties," in *AIAA Aerospace Sciences and Exhibit*, Reno Nevada, 2003.

[18] L. Singh, J. Plump, M. W. McConley, and B. D. Appleby, "Software Enabled Control: Autonomous Agile Guidance and Control for a UAV in Partially Unknown Urban Terrain," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, 2003.

[19] T. Schouwenaars, B. Mettler, E. Feron, and J. How, "Hybrid Model for Trajectory Planning of Agile Autonomous Vehicles," *Journal of Aerospace Computing, Information and Communication,* vol. 1, Dec 2004.

[20] D. Anisi, J. Robinson, and P. Ögren, "On-Line Trajectory Planning for Aerial Vehicles: A Safe Approach with Guaranteed Task Completion," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006.

BIOGRAPHY

Pritesh Narayan *completed his bachelor's degree in Aerospace Avionics Engineering with first class honors at QUT in 2005. His final year project included the development of autonomous capabilities for unmanned airborne platforms. His PhD research is focused on the generation of feasible flight trajectories with multi-objective optimization in scenarios with finite replanning time deadlines.*