On-Board Trajectory Generation for Collision Avoidance in Unmanned Aerial Vehicles

Chi-Kin Lai, Mudassir Lone, Peter Thomas, James Whidborne and Alastair Cooke
Dynamics, Simulation and Control Group
Cranfield University
Cranfield, Bedfordshire, MK43 0AL, UK
{c.k.lai, m.m.lone, p.thomas, j.f.whidborne, a.cooke}@cranfield.ac.uk

Abstract— This paper addresses the problem of collision avoidance with moving obstacles for unmanned aerial vehicles. It is assumed that obstacle detection and tracking can be achieved 60 seconds prior to collision. Such a time horizon allows on-board trajectory re-planning with updated constraints due to intruder and ownship dynamics. This trajectory generation problem is solved using a direct method, meaning the problem is transcripted to a nonlinear programming problem and solved with an optimization method. The main challenge in trajectory generation framework is to reliably provide a feasible (safe and flyable) trajectory within a deterministic time. In order to improve the method's reliability, a Monte Carlo analysis is used to investigate the convergence properties of the optimization process, the properties of the generated trajectories and their effectiveness in obstacle avoidance. The results show that the method is able to converge to a feasible and near-optimal trajectories within two seconds, except in very restrictive cases. Moreover, the dynamic feasibility of the generated trajectories is verified with nonlinear simulations, where the trajectory generation is integrated with the six degree-of-freedom nonlinear model of a fixed-wing research vehicle developed at Cranfield University. The results show that the generated trajectories can be tracked with a proposed two-degree-of-freedom control scheme. The improved convergence, fast computation and assured dynamic feasibility pave the way for on-board implementation and flight testing.

TABLE OF CONTENTS

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are particularly useful for Dangerous, Dull and Dirty tasks (3D tasks), yet, they have not been certified for routine use in non-segregated airspace, due to regulatory issues. One of the important issues is the ability to see and avoid other traffic while removing the pilot on-board [1]. UAVs themselves need to have at least the same level of safety as a manned aircraft in order to be certified. In other words, it requires an on-board Autonomous Collision Avoidance (ACA) system reliable enough for certification. The potential of UAVs in 3D tasks will only be fully exploited upon their successful insertion into civil non-segregated airspace.

A successful ACA system should be capable of sensing obstacles, decision making about conflict determination and guidance for conflict resolution. Thanks to remarkable developments in sensor technology and processing capabilities, motion estimation of obstacles, based on sensor fusion techniques, is reasonably accurate nowadays, e.g. Fasano et al [2] described a multi-sensor tracking algorithm which provides reliable estimates for not only intruders' positions but also their motion. With the kinematic information of intruders, collision conditions can be checked continuously based on fuzzy logic or geometry and kinematics, e.g. Angelov et al [3] proposed a collision detection method that uses bearing information only. Once the collision conditions are met, a conflict resolution has to be invoked and it should immediately generate the corresponding guidance commands to get rid of the collision conditions. Broadly, there are at least two types of resolution methods: guidance-law-based and online-optimization-based.

The main idea of guidance-law-based methods for conflict resolution is to resolve the collision conditions by manipulating the velocity vector command according to some predetermined feedback guidance laws. Shin et al [4] derived the collision conditions from two-dimensional geometry and then resolved them using differential geometry analysis. Two resolution guidance laws were obtained: for heading only and for heading and ground speed. Their local stability and existence were evaluated with Lyapunov theory. Luongo et al [5] obtained the collision conditions by intersecting a cylindrical safety bubble of an intruder with the relative velocity

[1] 978-1-4244-7351-9/11/$26.00 ©2011 IEEE.
[2] IEEEAC Paper #1251, Version 4, Updated 16/12/2010.

vector between the *ownship*[3] and intruder. The conditions then provide inequality constraints for a kinematic optimization problem of finding a minimum velocity change in order to avoid the collision. Under assumption of no dynamic constraints, it admits an analytical solution for the constrained optimization problem and the analytical solution itself serve as a resolution guidance law. Angelov et al [3] and Zak [6] derived two necessary conditions for ownship to keep a safe distance from an intruder, based on a collision model constructed with complex motion principles. The necessary conditions and equations of motion serve as equality constraints in a time-optimal control problem. With the help of an analysis of Euler-Lagrange equations for the functional minimization problem, 'bang-bang' resolution guidance laws were obtained. The advantages of such methods, in comparison to iterative methods, lies in the deterministic property of the guidance law. However, their effectiveness and optimality may fade out as uncertainties appear in environment and vehicle dynamics.

While available processing power is more than enough for predetermined guidance laws, online-optimization-based methods attempt to repeatedly solve an optimization problem for conflict resolution at each sample time. Although this idea from Model Predictive Control (MPC) or Receding Horizon Control (RHC) is well known, the success of its application to obstacle avoidance problems is limited due to an inherent nature of the problem: a non-convexity stemming from the fact that a feasible trajectory has to lie outside an obstacle area which generally approximated as a convex region. This non-convexity would affect the convergence and computation speed of the optimization. In order to get rid of the poor convergence and not-fast-enough computation speed of such non-convex constrained optimization problem, Shim et al [7] formulated the avoidance problem in a Nonlinear Model Predictive Control (NMPC) framework. In the NMPC framework, the controls of a detailed discrete model are chosen to minimize the augmented cost function with the obstacle constraint as penalty functions. This implies that even a non-convergent solution should be dynamically feasible and staying away from the obstacle. Although the obstacle constraint embodied as soft constraint is not guaranteed to be satisfied, the avoidance capability is shown in [8] and [9].

On the other hand, a Trajectory Generation (TG) framework utilizes direct methods for Optimal Control Problems (OCPs) while keeping obstacles as hard constraints. Betts [10] categorized direct methods as methods that deal with optimality directly with mathematical programming (optimization), rather than deal with the complex necessary conditions for optimality derived from theory. Thus, the main idea of direct methods is to transcribe an OCP to a Nonlinear Programming (NLP) problem and then solve it with some developed optimizer. A point-mass model, as a trade-off between computational complexity and required accuracies, is usually used for prediction of vehicle dynamic. Singla et al [11] converted the obstacle avoidance problem to a convex one with coordinate transformation and then solved the problem with a sequential linear programming approach. Patel et al [12] applied a direct multiple shooting method for aircraft avoidance maneuvers in the context of an anti-hijack system. It is reported that most cases succeed in providing a ten-second avoidance trajectory within a fraction of second while implemented in C++ , however, there exist some cases in which an additional constraint would cause the optimization to fail to converge.

The main challenge in the TG framework is to reliably provide a feasible trajectory within a deterministic time. In a previous paper, Lai et al [13] applied a direct method, called Inverse Dynamic in Virtual Domain (IDVD) [14], to aircraft re-routing problems for a pop-up obstacle scenario. The convergence property was investigated with a Monte Carlos analysis, and the results show a considerable convergence improvement when with a better-initial-guess optimization routine. However, the convergence in more restrictive cases still remains unsatisfactory: it sometimes failed to converge to a feasible trajectory within given maximum iterations. Furthermore, the dynamic feasibility of generated trajectories has not been verified as they are only evaluated based on a point-mass model. The relationship between tracking performance and constraint setting becomes particularly important in collision avoidance applications. This paper further investigates the method's convergence properties and the generated trajectories' properties which include their dynamic feasibility, effectiveness in obstacle avoidance and ability to provide functional trajectories.

The paper is organized as follows: Section 2 describes a proper problem formulation for the TG and its interactions with other components in the ACA system. Section 3 introduces the ideas and corresponding operations for a direct method and a better-initial-guess optimization routine to solve the problem. Section 4 presents an investigation of the method's convergence properties and functionalities in a Monte Carlo analysis. The six Degree-of-Freedom (6 DoF) simulation results for the TG's application to a research UAV are presented in section 5. Finally, Section 6 contains conclusions and suggestions for further work.

2. TRAJECTORY GENERATION FOR COLLISION AVOIDANCE

This section first presents a functional architecture[4] of a general ACA system within a close-loop vehicle system. This is followed by the presentation of an OCP formulation for collision avoidance guidance.

Autonomous Collision Avoidance Systems

As shown in Figure 1, the ACA system takes as input the nominal trajectory and current state from the Mission and

[3]The host vehicle equipped with the ACA system is named as ownship hereafter.

[4]This architecture is based on the architecture in [15], making sure compatibilities of the trajectory generation subsystem.

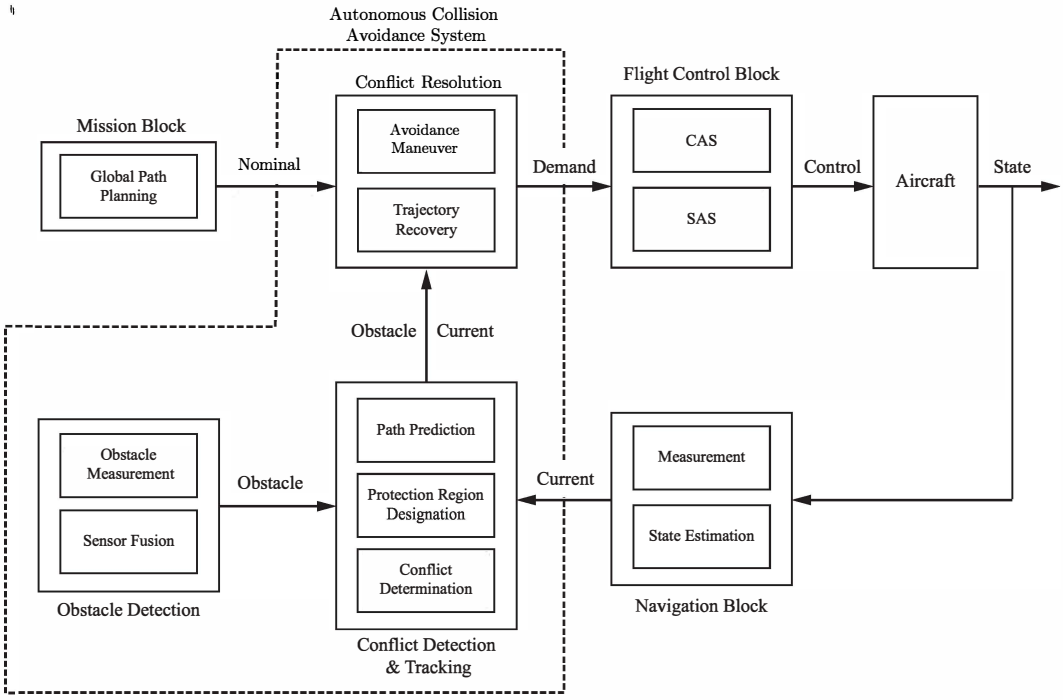

Figure 1. Functional architecture for an Autonomous Collision Avoidance system.

Navigation Block respectively, and then commands the Flight Control Block with the corresponding guidance control signals. To achieve collision avoidance autonomously, it has to detect, determine and resolve any conflicts on-board with the following sub-systems: Obstacle Detection, Conflict Detection & Tracking and Conflict Resolution.

Obstacle Detection—To detect surrounding obstacles with its on-board sensor suite, which can be comprised of active (e.g. radar) and/or passive (e.g. electro-optical) types of sensors. It is desirable to have multiple sensory suites on-board, thus allowing data fusion techniques to improve the accuracy of estimations [2]. Furthermore, while active sensors are able to provide range and better resolution information, passive sensors provide an aspect of covertness, and protection against other small aircraft or hot air balloons [3] [16]. The output of this block would be the estimated position and velocity of the obstacle with bounded uncertainty.

Conflict Detection & Tracking—To track intruders' motions and determine whether a conflict is emerging. The intruder's future trajectory is predicted and a protection zone around the intruder is prescribed according to the relevant class of national airspace. Based on this information, a decision making algorithm, such as in [16] and [15], assesses the risk of conflict between the ownship and the intruder. This collision check should run continuously with a short sampling time. When an upcoming conflict is predicted, the Conflict Resolution sub-system is triggered with the information of the predicted collision; this would include time-to-collision, closest point of approach, and the kinematic information of the intruder and ownship.

Conflict Resolution—To calculate required guidance control commands to achieve an avoidance maneuver with updated kinematic information of the intruder and the ownship while taking the following considerations into account:

1. Dynamic feasibility: performance limitations of the ownship cannot be violated or it may lose control of the vehicle and thus, further endanger itself or other airspace users.
2. Minimum Separation Distance (MSD): a regulatory requirement for the vehicles to maintain a MSD in all circumstances to reduce the risk of an actual collision.
3. Rules of the Air: according to the regulations, there are some priority rules that should be followed, e.g. turn right to avoid a head-on collision in most cases.[5]
4. Cost of maneuver: cost of the avoidance maneuver can be evaluated. Depending on the mission objectives, these can be minimum deviation from nominal trajectory, minimum path length or minimum fuel consumption.

Based on the estimated time-to-collision, the avoidance maneuver can be divided into two: an *emergency evasive maneuver* (avoiding the obstacle at all cost in a shorter time horizon) and an *optimal re-routing* (de-conflicting with cost minimization in a relatively longer time horizon). No matter what means are taken a *trajectory recovery* must be carried out once the conflict is resolved. This means rejoining the nominal trajectory from any deviations caused by the avoidance maneuver.

[5] The results in this paper are obtained with no considerations of the Rules of the Air.

Problem Formulation for Trajectory Generation

Although the MPC and TG framework share the same idea of on-line optimization, they differ in their prediction models and time horizons. MPC uses a relative shorter time horizon and behaves in a more reactive manner. Therefore, this is more suitable for emergency evasive maneuvers. On the other hand, the TG formulation attempts to simultaneously achieve avoidance and recovery by optimal re-routing with a longer time horizon.

Model—Usually in a TG context, a detailed point mass model is used to capture vehicle dynamics. However, a kinematic model is used here for the following reasons:

1. Due to certification requirements, flight control and guidance systems are designed separately. Hence, the ACA system can only control the vehicle's velocity vector through the flight control system.
2. An ACA system based on a more general kinematic model implies an interoperability across multiple platforms. More importantly, future tactical UAVs may have novel airframes or control strategies whilst maintaining the same kinematics.
3. The current purpose is to investigate convergence properties when constraints become more restrictive; a kinematic model offers appropriate restrictiveness for such investigation.

The system of differential equations for the kinematic model is as follows:

$$\begin{aligned}\dot{x} &= v\cos\gamma\cos\chi \\ \dot{y} &= v\cos\gamma\sin\chi \\ \dot{z} &= -v\sin\gamma\end{aligned} \tag{1}$$

where, $\vec{x} = [x(t), y(t), z(t)]^T$ is the state with the three components of inertial positions; $\vec{y} = \vec{x}$ is the output; $\vec{u} = [v(t), \gamma(t), \chi(t)]^T$ is the control with speed, Flight Path Angle (FPA) and heading.

Constraints— The model is subject to the following initial conditions, at t_0 when the optimal re-routing is engaged:

$$\vec{x}_0 = \begin{bmatrix} x(t_0) \\ y(t_0) \\ z(t_0) \end{bmatrix}; \vec{u}_0 = \begin{bmatrix} v(t_0) \\ \gamma(t_0) \\ \chi(t_0) \end{bmatrix} \tag{2}$$

and final conditions, at t_f when the vehicle reaches the given critical final position and velocity:

$$\vec{x}_f = \begin{bmatrix} x(t_f) \\ y(t_f) \\ z(t_f) \end{bmatrix}; \vec{u}_f = \begin{bmatrix} v(t_f) \\ \gamma(t_f) \\ \chi(t_f) \end{bmatrix} \tag{3}$$

The admissible control space is defined based on the vehicle's capability by:

$$\mathbb{U} = \{\vec{u} \in \mathbb{R}^3 : v \in [v_{min}, v_{max}] \text{ and } \gamma \in [\gamma_{min}, \gamma_{max}]\} \tag{4}$$

The admissible state space is defined based on the intruder's position, $\vec{r}_I(t)$, and the size of its protection zone, R_{PZ}:

$$\mathbb{X} = \{\vec{x} \in \mathbb{R}^3 : \|\vec{x} - \vec{r}_I\| \geq R_{PZ}\} \tag{5}$$

Cost Function—The following three cost functions are used in this paper to compare their effects on optimization and investigate the properties of resulting trajectories:

Path Length, L, between the starting position and final position:

$$L = \int_{t_0}^{t_f} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} dt \tag{6}$$

Maneuver Time, T, required to achieve the final position:

$$T = \int_{t_0}^{t_f} dt = t_f \tag{7}$$

Maximum Deviation, D, of ownship from its nominal trajectory:

$$D = \max\{\|\vec{x}(t) - \vec{x}_{ref}(t)\| : \forall t \in [t_0, t_f]\} \tag{8}$$

Avoidance Trajectory Generation Problem—Subject to all the above constraints, to find an optimal control $\vec{u}^*(t)$ to drive the system onto the optimal trajectory $\vec{x}^*(t)$ that minimizes the selected functional J (in this case L, T or D):

$$OCP: \min_{\vec{u}(t)} J, \quad s.t.\ (1) - (5), \forall t \in [t_0, t_f]. \tag{9}$$

Remark 1—Further investigations about suitable constraint settings for dynamic feasibilities can be carried out using a more detailed model. The direct method described below is able to extend to a detailed point-mass model by only changing the constraints in the optimization problem as in Lai et al [13] and Drury et al [17] [18].

Remark 2—This is the basic problem needed to be solved at every sample step. A regular update, as in Cowling et al [19] and Berry et al [20], to repeatedly solve the problem is introduced in future work.

3. DIRECT METHOD FOR TRAJECTORY GENERATION

This section first introduces the idea of the IDVD direct method followed by three operations that convert the OCP to an optimization problem. Lastly, the optimization process to solve the resulting NLP problem is proposed.

IDVD Direct Method

Figure 2 shows the process and all required operations for IDVD direct method. The process can be read from the right to left as it converts the OCP to an optimization problem. Each operation is briefly explained in the following subsections; detailed descriptions can be found in Lai et al [13] and Yakimenko [21]. The explanation starts from the OCP in (9) and finishes with the NLP in (17).

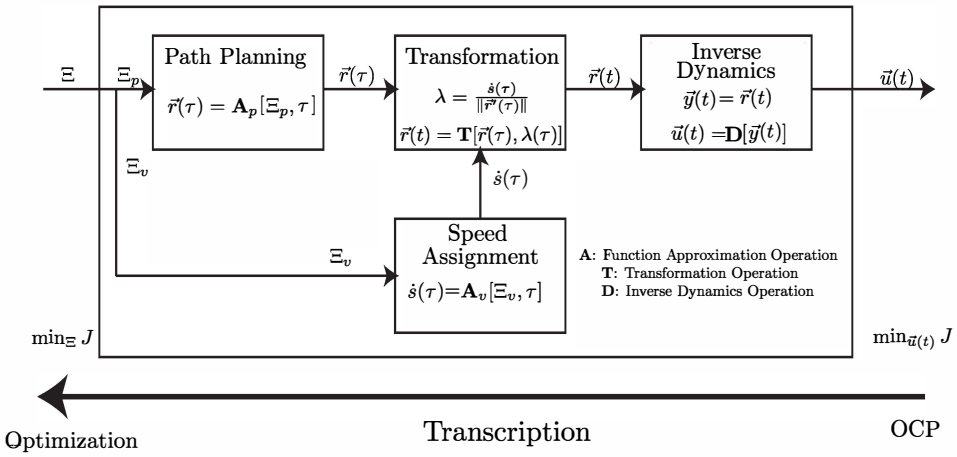

Figure 2. Process of IDVD direct method transcribing an optimal control problem to a nonlinear programming problem

Inverse Dynamics—The concept of inverse dynamics for a system is to express control functions with output information only. Therefore, required controls for a system to achieve some desired output trajectories can be readily obtained. The inverse dynamic of the kinematic model in (1) can be obtained as follows:

$$\begin{aligned} v &= \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \\ \gamma &= -\arctan \frac{\dot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \chi &= \arctan \frac{\dot{y}}{\dot{x}} \end{aligned} \tag{10}$$

Therefore, the mapping from outputs to controls, $\vec{u}(t) = \mathbf{D}[\vec{y}(t)]$, can be achieved with the above relation. As shown by the Inverse Dynamics block in Figure 2, the optimal control problem of searching for $\vec{u}(t)$ can be converted to a search for a reference trajectory in time domain, $\vec{r}(t)$.

Transformation—The main idea here is to introduce an intermediate variable called *virtual arc*, τ, to decompose the reference trajectory determination into two tasks: path planning and speed assignment. A *virtual speed* is then defined as the derivative of the virtual arc with respect to time as follows:

$$\lambda(\tau) \triangleq \frac{d\tau}{dt} \tag{11}$$

Using the chain rule, the virtual speed can also be expressed as follows:

$$\frac{ds}{dt} = \frac{ds}{d\tau}\frac{d\tau}{dt} \Longrightarrow \lambda = \frac{\dot{s}}{s'} \tag{12}$$

where, s is the path length and the prime represent the derivative with respect to virtual arc.

Using the chain rule again, the following relationship between virtual arc and time domain can be obtained:

$$\begin{aligned} &\dot{\vec{r}} = \lambda \vec{r}'; \quad \ddot{\vec{r}} = \lambda(\vec{r}''\lambda + \vec{r}'\lambda') \\ &\dddot{\vec{r}} = \lambda^3 \vec{r}''' + 3\lambda^2\lambda'\vec{r}'' + (\lambda^2\lambda'' + \lambda\lambda'^2)\vec{r}' \end{aligned} \tag{13}$$

where, it can be seen that the information about the virtual speed and reference trajectory in virtual domain ($\lambda(\tau)$ and $\vec{r}(\tau)$) are required in order to obtain the reference trajectory in time domain, $\vec{r}(t)$. This is represented by $\vec{r}(t) = \mathbf{T}[\vec{r}(\tau), \lambda(\tau)]$.

On the other hand, the virtual speed in virtual domain is determined by the speed assignment and reference trajectory in virtual domain by Equation (12).

Finally with all the above transformations, the time domain reference trajectory is uniquely defined by the speed assignment and reference trajectory in virtual domain, as shown in the Transformation block in Figure 2. Thus, the search is now for $\vec{r}(\tau)$ and $\dot{s}(\tau)$.

Function Approximation—In order to avoid a search in an infinite function space, function approximations are used to convert it to a search in a parameter space of finite dimension.

As shown in the Path Planning block in Figure 2, the reference trajectory in virtual domain is first defined by three seventh-degree polynomials:

$$\vec{r}(\tau) \triangleq \sum_{k=0}^{7} [a_k, b_k, c_k]^T \frac{\tau^k}{\max(1, k(k-1))} \tag{14}$$

in which, a_k, b_k, and c_k are coefficients of the three coordinates' polynomials.

The coefficients are restricted by the derivatives of the coordinates on the boundaries ($\tau = 0$ and $\tau = \tau_f$). For instance, a_k are determined by the following set of linear equations:

$$\vec{a} = \mathbf{C}^{-1}\vec{x}_{BC} \tag{15}$$

where,

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{\tau_f^2}{2} & \frac{\tau_f^3}{6} & \frac{\tau_f^4}{12} & \frac{\tau_f^5}{20} & \frac{\tau_f^6}{30} & \frac{\tau_f^7}{42} \\ 0 & 1 & \tau_f & \frac{\tau_f^2}{2} & \frac{\tau_f^3}{3} & \frac{\tau_f^4}{4} & \frac{\tau_f^5}{5} & \frac{\tau_f^6}{6} \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 & \tau_f^4 & \tau_f^5 \\ 0 & 0 & 0 & 1 & 2\tau_f & 3\tau_f^2 & 4\tau_f^3 & 5\tau_f^4 \end{bmatrix}$$

$$\vec{a} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{bmatrix}^T$$

$$\vec{x}_{BC} = \begin{bmatrix} x_0 & x_0' & x_0'' & x_0''' & x_f & x_f' & x_f'' & x_f''' \end{bmatrix}^T$$

Thus, the coefficients are uniquely defined by τ_f and the boundary values. After imposing the initial and final conditions, (2) and (3), all other values still need to be determined in the optimization process. The optimization variable vector for path planning in this case is as follows:

$$\vec{\Xi}_p = [x''_{0,f}, y''_{0,f}, z''_{0,f}, x'''_{0,f}, y'''_{0,f}, z'''_{0,f}, \tau_f]^T \qquad (16)$$

With respect to the Speed Assignment block in Figure 2, the same operation of function approximation can be applied to parameterize the speed profile explicitly or implicitly as shown in Kaminer et al [22]. Alternatively, Drury et al [18] evaluate the maximum feasible airspeed directly from the constraints and spatial path. Note that such speed assignment may introduce additional decision variables, $\vec{\Xi}_v$.

As shown in the Transcription block of Fig.2, by concatenating $\vec{\Xi}_v$ and $\vec{\Xi}_p$ to $\vec{\Xi}$, the search is now converted to a nonlinear programming problem:

$$NLP: \min_{\vec{\Xi}} J(\Xi), \quad s.t.\ \vec{c}(\vec{\Xi}) \leq 0, \quad \vec{c} \in \mathbb{R}^{\mathbb{M}} \qquad (17)$$

where, $\vec{c}(\vec{\Xi})$ represents the converted constraints from (4) and (5). This can be solved by any standard nonlinear constrained optimization method.

Optimization

From the optimizer's point of view, only a cost and constraints generator is required for optimizations. For each optimization variable vector, the corresponding control and state vectors can be obtained from the process of IDVD method, then the constraints and cost are evaluated at each node with equations given in Section 2.

Lai et al [13] proposed an optimization routine attempting to combine efficiency in gradient-based optimization methods and the better-initial-guess idea in sequential applications. The routine based on the same cost and constraint generators involves two steps with two optimizers.

In the first step, a random initial guess Ξ_{ran} and a zero-order unconstrained optimizer `fminsearch` in $Matlab^{TM}$ are used to minimize the constraint violation in NLP in (17) alone; this is $\eta = \sum_{i=1}^{M} \max(c_i, 0)^2$. As the purpose here is to get a better initial guess Ξ_{bet} for the subsequent optimizations, the process is truncated after a certain number of iterations. The second step takes the resulting Ξ_{bet} and calls a second-order constrained optimizer `fmincon` in $Matlab^{TM}$ to solve the complete NLP problem.

Remark 1—All constraint violations are normalized by their limitation values because the degree of constraint violation is better represented by this relative value, especially in the cases with different types of constraints at the same time.

4. CONVERGENCE & FUNCTIONALITY ANALYSIS

In order to improve the method's reliability, a Monte Carlo analysis is designed and used to investigate the convergence properties of optimization process with respect to constraints and cost functions. The functionalities of generating minimum-cost and obstacle-avoidance trajectories are also evaluated in this framework.

Monte Carlo Analysis Setting

A simulation framework is designed based on that of Luongo et al's [5], which is designed to meet standard specifications given in [23]. It consisted of 500 collision scenarios and four performance measures as detailed below.

Ownship's Motion—The ownship is modeled as a kinematic point-mass having limitations in speed and FPA. It is originally traveling along a nominal straight trajectory going from point A to B with the following approach velocity vectors:

$$P_A = [0, 0, -1500]^T(m), \vec{V_A} = [20, 0, 0]^T(m/s) \quad \text{and}$$
$$P_B = [1400, 0, -1500]^T(m), \vec{V_B} = [20, 0, 0]^T(m/s)$$

Intruders' Motion— Intruders are modelled as kinematic point-masses and assumed to travel always on a straight line. 500 intruders' initial positions, P_I, are randomly generated (with a uniform distribution) on the boundary of an assumed sensor range; this is ± 110 degrees in azimuth, ± 15 degrees and 1.5 km in range.

Guaranteed Collision Scenario—To assure a scenario where a collision will happen if no avoidance maneuver is performed, the following calculations are used to determine the intruder velocity vectors:

1. Generate a random position on the ownship's original nominal trajectory as a desired collision point, C, and calculate the time for ownship to reach this point, t_c.
2. Connect the intruder P_I and point C to obtain the direction of intruder's velocity, $\hat{\vec{V}}_I$, and the distance between two points, d_{IC}.
3. Calculate the required speed to collide, $|\vec{V}_I|$, by dividing d_{IC} with t_c.

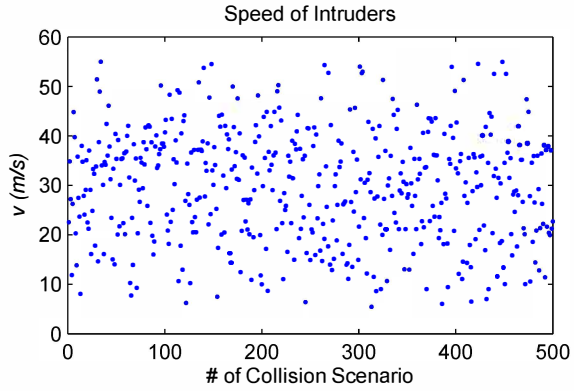

Figure 3. Intruders' speed in 500 collision scenario.

The resulting velocities of intruders for 500 cases covers a wide range of possible directions and speeds as shown in Figure 3.

Performance Measures—Three cost functions defined in section 2 are used to evaluate the quality (optimality) of the generated trajectories: path length, maneuver time and maximum deviation. The Closest Distance of Approach, CDA, between the ownship's position and the intruder's protection zone is used to evaluate the closeness between two vehicles. CDA less than zero implies a high risk of collision as the ownship fails to maintain the MSD from an intruder.

Optimization Setting

Constraints on speed, FPA and obstacle avoidance were used to investigate convergence properties. The optimization is made more restrictive by only allowing variation in FPA constraint. The values used as constraint limitations can be found in Table 1.

Table 1. Limiting values for constraints

$v \in [15, 25](m/s)$,	MSD $= 150(m)$,
$\gamma \in [-\gamma_{lim}, \gamma_{lim}](^\circ)$,	$\gamma_{lim} \in \{4, 6, 8, 10, 12, 14, 16\}(^\circ)$

Table 2 summarizes the optimizer's parameters used in this paper. Within the specified maximum number of iterations or function evaluations, a *convergent solution* is a local minima whose constraint violations is less than the specified tolerance. A *successful avoidance trajectory* is the convergent solution that can be obtained within 5 seconds.

Table 2. Values for optimization setting

Constr. Tol.	$1e-5$	Fun. Tol.	$1e-5$
Max. Iter.	1000	Max. Fun. Eval.	3000
# of nodes	$N = 100$	Algorithm	active-set

Simulation Results

Convergence Properties—Table 3 summarizes three sets of results for 500 collision scenarios with seven different FPA constraints and three different cost functions. The convergence rates, success rates and average computation times are presented. It can be seen that the convergence rates decrease from about 96% to about 70% as the maximum FPA constraint varies from 16° to 4°. Such a drop in convergence rate is investigated fully in the paragraph below. With respect to the computation times, almost all cases with minimum path length and minimum time cost functions are able to converge to solutions within 2 seconds, whereas cases with minimum deviation took more time; up to 9 seconds for optimization.

Figure 4 shows two avoidance trajectories with maximum FPAs of 4° (blue) and 8° (green) respectively. The comparison illustrates a more restrictive situation for an avoidance maneuver with a smaller value of FPA. Now consider the bird's eye view of the collision scenario in Figure 4. The intruder's trajectory (red) is almost perpendicular to the nominal trajectory (black). This implies that there is a higher risk of collision in the lateral plane of the ownship. It can be seen that the trajectory with 4° FPA constraint (blue) is unable to climb over the intruder; unlike the trajectory with 8° FPA constraint (green). It therefore, conducts a spiral transition maneuver to avoid the intruder. This avoidance maneuver would lead to higher cost, longer path length and larger deviation. Thus the optimization process may be resistant to such type of solutions and attempt to tune an infeasible solution around the straight nominal trajectory.

Figure 5 further shows the feasibility of two avoidance maneuvers shown in Figure 4. It can be seen from the time history of the corresponding controls that all controls stay within their limitation values. They also shows the aggressiveness of the maneuvers as they always try to make most use of the capability by reaching their limit. In this collision scenario, a vehicle with lower capability can only choose an avoidance maneuver with higher cost.

Cost Functions Properties—Table 4 summarizes the actual values for the performance index: path length, maneuver time and maximum deviation. Each is used in-turn as the cost function to be minimized. Note that, as an illustrative example, the actual values shown are the statistical values for 500 trajectories with a maximum FPA constraint of 16°. These are obtained from the Cumulative Distribution Function (CDF) plot shown in Figure 6. The diagonal values of the table show that the selected cost is always being minimized. Although the optimality of solutions are not determined, the resulting trajectories are always feasible with good performance.

Moreover, while the flexibility in cost functions selection can be utilized according to various mission objectives, caution has to be taken during the selection because it has a considerable effect on the optimization process. As shown in Table 3, for some cases the optimization time increases when max-

Table 3. Convergence, success rates and computation times for different FPA constraints.

Cost function	FPA (°)	16	14	12	10	8	6	4
Minimum path length	Convergence (%)	98.4	93.0	88.0	75.2	76.8	75.8	72.6
	Success (%)	98.4	93.0	88.0	75.0	76.8	75.8	72.6
	Time (s)	1.17	1.65	1.23	0.97	1.14	0.95	1.26
Minimum time	Convergence (%)	96.2	91.4	86.8	72.6	68.2	68.2	71
	Success (%)	95.4	91.4	86.8	72.6	68	68.2	70.8
	Time (s)	1.25	0.85	1.64	0.86	0.85	1.18	1.20
Minimum deviation	Convergence (%)	97.8	89.8	85.4	75.0	76.4.2	73.8	72.4
	Success (%)	75.6	73.6	65.4	58.2	66.4	67.2	66.0
	Time (s)	3.52	9.58	9.40	7.81	1.38	1.12	1.86

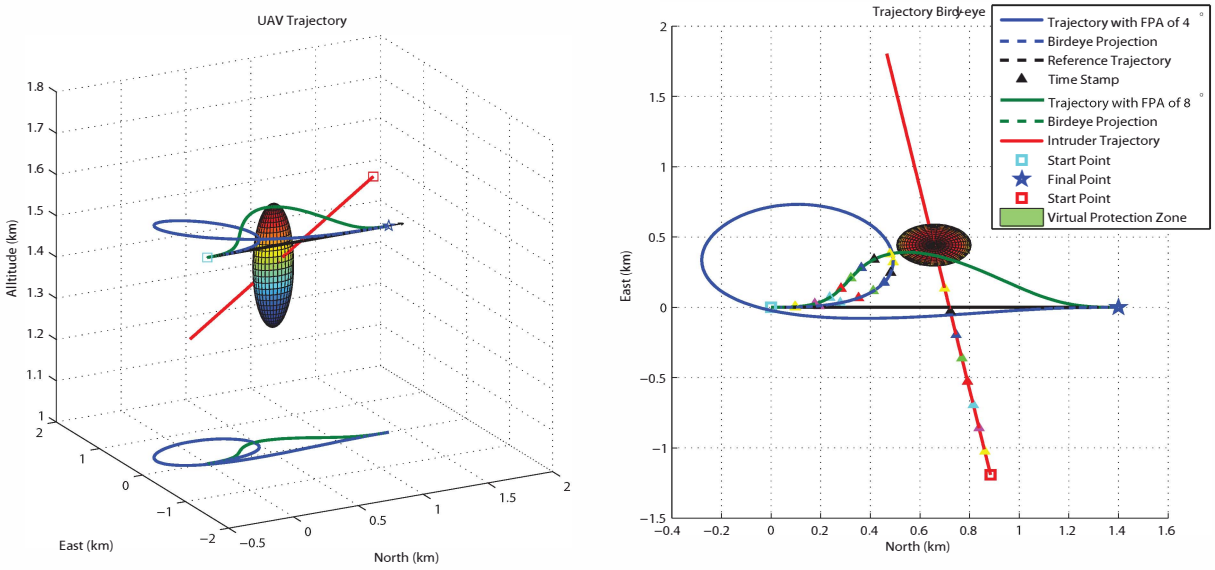

Figure 4. Trajectories for a collision avoidance scenario with 4° and 8° FPA constraints.

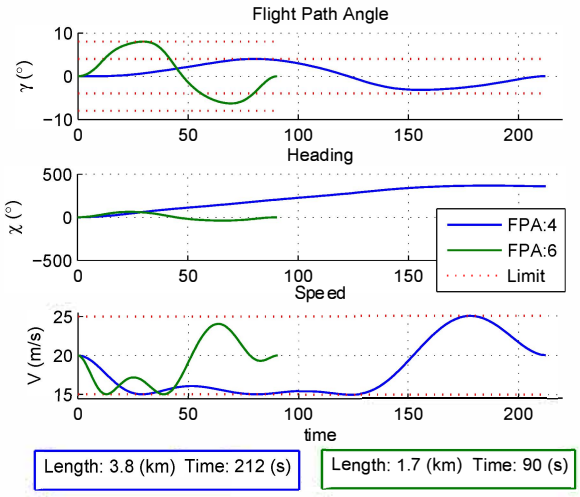

Figure 5. The corresponding control functions for Figure 4.

imum deviation is used as a cost function.

Table 4. Maneuver cost with different cost functions.

Cost Function	Time (s)	Length (m)	Deviation (m)
Time	**69.7**	1508.3	236.7
Length	81.9	**1440.0**	252.0
Deviation	77.6	1460.0	**160.0**

Avoidance Capability—Figures 6 and 7 summarize the Monte Carlo analysis for two extreme cases: maximum constraint FPA of 4° and 16° (both with the path length as a cost function). From Figure 6 it can be seen that the CPA of the convergence scenarios (first 492 scenarios) are always greater than zero, which means the closest distance between two vehicles is always greater than the required MSD and therefore no collisions occur. However, when the optimization fails to converge, as for the last 8 scenarios, the CPA ranges from 0 m to -10 m. This means that the ownship endangers itself by entering the intruder's protection zone. Moreover, in

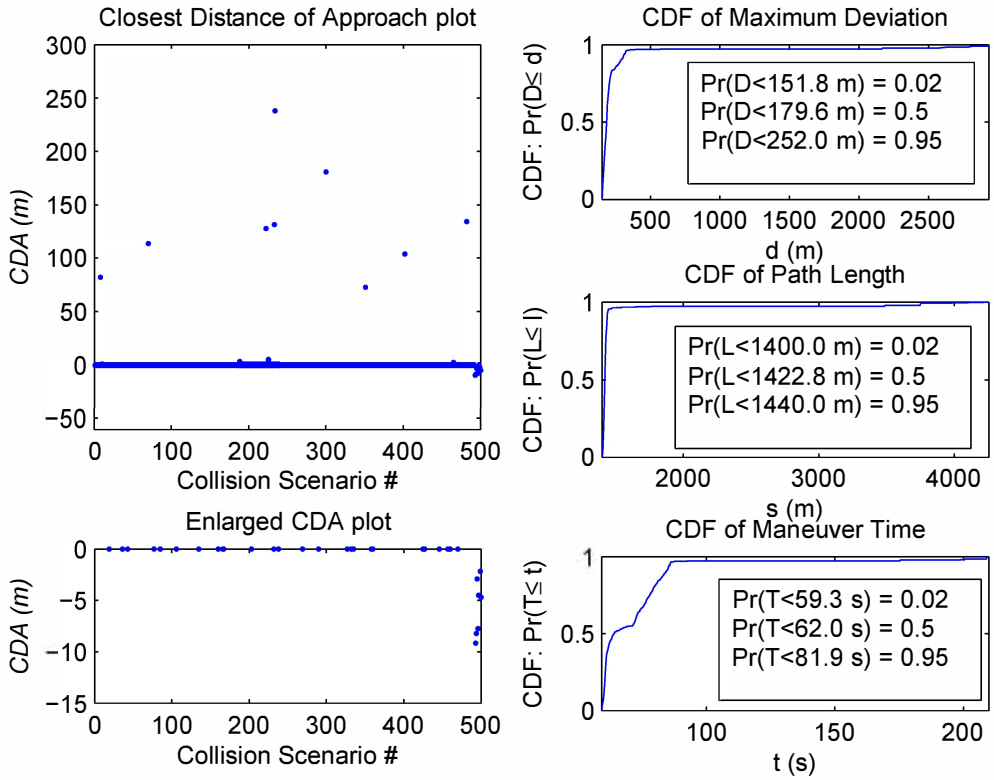

Figure 6. Monte Carlo analysis result over 500 collision scenarios with FPA constraint of 16° and path length cost function.

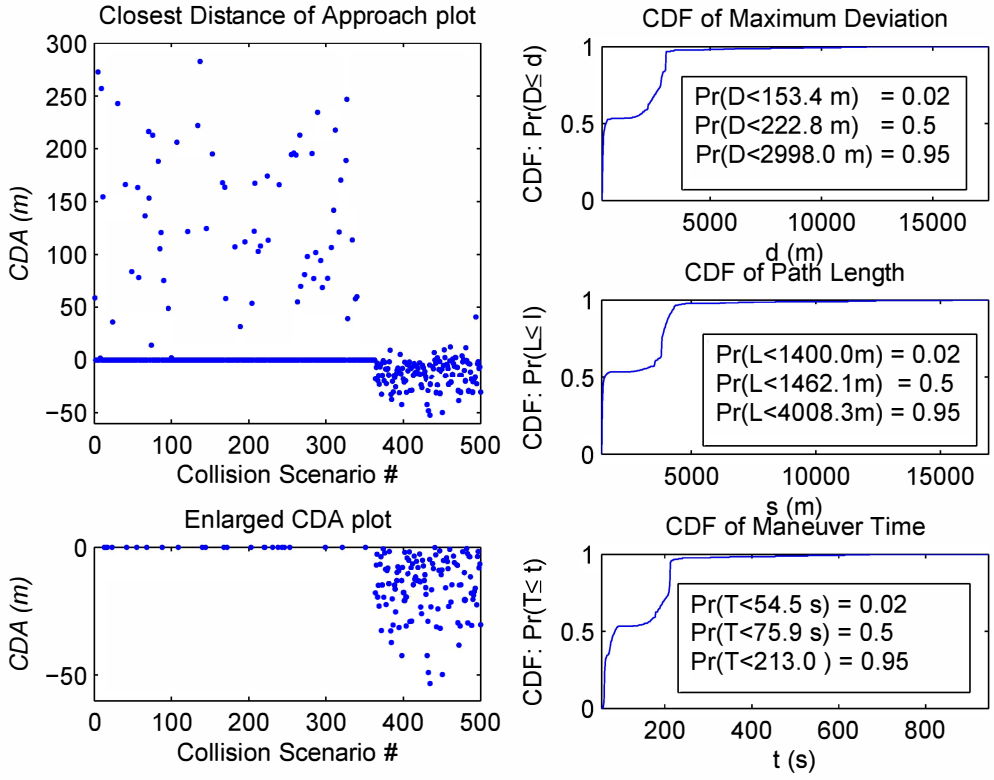

Figure 7. Monte Carlo analysis result over 500 collision scenarios with FPA constraint of 4° and path lenth cost function.

most cases, the CPA is a low value very near zero, implying that the resulting avoidance trajectories always try just to miss the protection zone for shorter path length. On the right-hand-side of Figure 6, three CDFs associated with three variables of D, L and T are shown. The CDF is constructed by cumulating the performance index over all 500 cases and it shows the probability distribution of the random variable; in this case the performance index. It was found that 95% of cases have a maximum deviations less than 252.0 m, path lengths less than 1440.0 m and maneuver times less than 81.9 seconds.

Figure 7 shows the other extreme of 4° FPA constraint. 123 collisions occur, significantly more than that for the 16° FPA constraint. This is consistent with the non-convergent cases shown in Table 3. There are more cases with CDA ranging from 50 m to 300 m due to the spiral transition maneuver as described in the previous section. With respect to performance, 95% of cases have maximum deviations less than 2998.0 m, path lengths less than 4008.3 m and maneuver times less than 213.0 seconds. The performance is poorer than the other extreme as there are a certain number of high-cost spiral transition maneuvers in this more restrictive situation. Furthermore, the shapes of three CDF plots imply that there are two main types of maneuvers dominating the distribution and the performance of 50% cases are comparable to that of the 16° FPA constraint cases. Therefore, it is further demonstrated that one of the maneuvers is a normal avoidance maneuver while the other is a spiral transition maneuver, like the green and blue one in Figure 4 respectively.

5. Application to the 6 DoF Model

This section presents the integration of the TG and the verification of dynamic feasibilities. It first describes a high fidelity 6 DoF model of the research UAV and its flight control system with conventional autopilot functions. A two-degree-of-freedom control scheme and trajectory tracking is then introduced. Lastly, the dynamic feasibility of the trajectories generated is verified with 6 DoF simulation and the tracking performances are presented.

Controlled Vehicle

6 DoF Model Description—The Cub, shown in Figure 8, is a small (4 kg) remotely operated vehicle with a high wing trainer-like configuration. It is a modified off-the-shelf radio controlled scale model of the Piper Cub J-3 aircraft, which is used for technology demonstration and evaluation. It is powered by either a four stroke internal combustion engine or, an equivalent-sized electric motor, providing up to 820 W of power.

The 6 DoF model is an implementation of the six rigid body aerodynamic stability and control equations from which, the forces and moments acting on the vehicle's center of gravity

Figure 8. Cub UAV research platform.

are obtained (more details can be found in Thomas [24]):

$$C_D = C_{D_\bullet} + C_{D_M} M + C_{D_\alpha} \alpha + \frac{c}{2V}\left(C_{D_q} q + C_{D_{\dot\alpha}} \dot\alpha\right) + C_{D_{\delta_e}} \delta_e + C_{D_{\delta_a}} \delta_a + C_{D_{\delta_r}} \delta_r \quad (18)$$

$$C_Y = C_{Y_\beta} \beta + \frac{b}{2V}\left(C_{Y_p} p + C_{Y_r} r\right) + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \quad (19)$$

$$C_L = C_{L_\bullet} + C_{L_M} M + C_{L_\alpha} \alpha + \frac{c}{2V}\left(C_{L_q} q + C_{L_{\dot\alpha}} \dot\alpha\right) + C_{L_{\delta_e}} \delta_e \quad (20)$$

$$C_l = C_{l_\beta} \beta + \frac{b}{2V}\left(C_{l_p} p + C_{l_r} r\right) + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \quad (21)$$

$$C_m = C_{m_0} + C_{m_M} M + C_{m_\alpha} \alpha + \frac{c}{2V}\left(C_{m_q} q + C_{m_{\dot\alpha}} \dot\alpha\right) + C_{m_{\delta_e}} \delta_e \quad (22)$$

$$C_n = C_{n_\beta} \beta + \frac{b}{2V}\left(C_{n_p} p + C_{n_r} r\right) + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \quad (23)$$

The propulsion system is modeled with a database containing the power and fuel flow values as functions of throttle setting and engine speed. The combined forces and moments are used in the solution of the equations of motion in the usual way to obtain the acceleration, speed, and position of the body center.

Flight Control System—Inner-loop control is implemented with PI and PID controllers providing altitude, bank, sideslip, and speed regulation. Altitude control is preceded with a flight path angle, track-and-hold proportional gain, whilst a wing-leveler with inner-loop roll rate regulation constitutes the bank angle controller.

Integration of Trajectory Generation

Two-degree-of-freedom control design—In order to achieve a robust guidance function for conflict resolution, a two-degree-of-freedom control scheme is used to integrate the TG with the controlled vehicle. This scheme is applied to missile guidance by Lukacs et al [25] and to local motion planning for micro aerial vehicles by Berry et al [20]. Figure 9 shows the block diagram of this control scheme, which constitutes a TG and a Trajectory Tracking (TT) system. The TT is required to close the loop when the optimization process within the TG is not fast enough to provide feedback in real time.

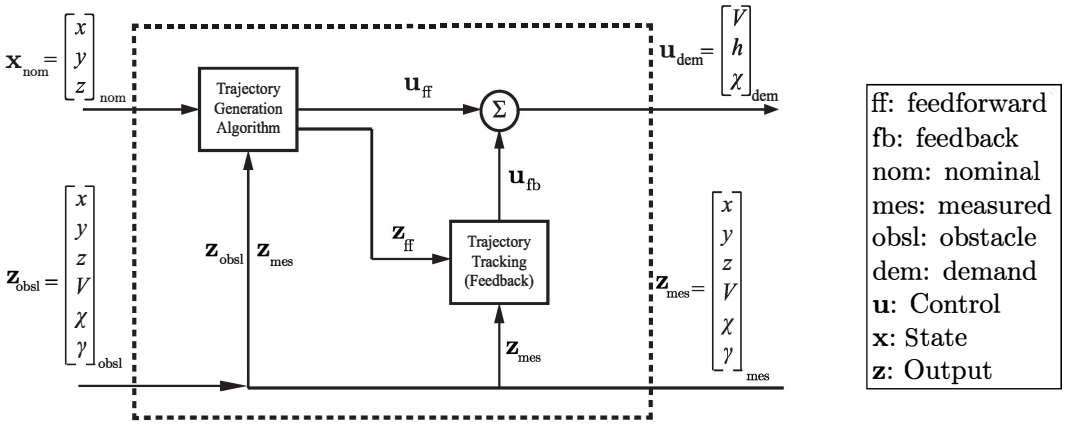

Figure 9. Schematic of the two-degree-of-freedom control design.

Furthermore, uncertainties between models used in TG and actual vehicle dynamics is inevitable. The TT can compensate for such uncertainties to provide better overall performance. Therefore, command signals to FCS are comprised of a feedforward element from TG and a feedback element from TT.

Trajectory Tracking Design—A PI controller is used for TT, which introduces feedback control via the following control laws:

$$\begin{aligned} V_{fb} &= K_v(V_{ff} - V) + (K_x + \frac{K_{xi}}{s})(x_{ff} - x) \\ h_{fb} &= K_h(h_{ff} - h) + K_\gamma(\gamma_{ff} - \gamma) \\ \chi_{fb} &= (K_\chi + \frac{K_{\chi i}}{s})(\chi_{ff} - \chi) + (K_y + \frac{K_{yi}}{s})(y_{ff} - y) \end{aligned} \tag{24}$$

The control law gains were initially designed via frequency-domain techniques using the linear approximations of the augmented aircraft. These gains, as shown in Table 5, were later verified and tuned through simulations with the nonlinear aircraft model. Simulations at various flight conditions showed negligible loss in performance and therefore, no gain scheduling was incorporated.

Table 5. Values used for control law gains.

Gain	Value	Gain	Value
K_v	1.00	K_x	1.20
K_{xi}	0.01	K_γ	0.90
K_χ	1.20	$K_{\chi i}$	0.001
K_y	0.02	K_{yi}	0.002
K_h	-0.01		

Verification

The dynamic feasibility of the generated trajectories in Section 4 is verified with the above integration. 100 trajectories were randomly selected from 500 collision scenarios and served as feedforward commands, as shown in Figure 9. It should be noted that, since the dynamic feasibility for the whole trajectory is being investigated, there is no feedback to the TG in this paper.

Table 6 summarizes the tracking performance observed during the 6 DoF simulations, where the FPA constraint ranged from 4° to 10°. It can be seen that 90% of cases with 4° maximum FPA were tracked with a maximum error less than 5 m[6]. The tracking performances worsens as the maximum value for FPA is increased. This can be contributed to the limited capability of the PI controller for longitudinal trajectory tracking. When a considerable error occurred and could not be compensated by the PI controller, the feedforward signals became ineffective because the TG is not updated. Thus, the tracking error accumulated as time went by.

Table 6. Tracking performance for 100 avoidance trajectories

Constraint on FPA ($^\circ$)	4	6	8	10
Maximum Error $\leq$ 5 m ($\%$)	96	75	71	64

Figure 10 shows an example of the trajectory tracking results. It is evident that, the actual trajectory (blue) is very close to the feedforward reference trajectory (red). The tracking performances for speed, FPA and heading are shown in Figure 11. The maximum error occurs around 100 seconds, when the aircraft has to recover from a dive.

6. CONCLUSIONS

The collision avoidance problem for conflict resolution has been addressed in a trajectory generation framework with a direct method. The convergence of the method, properties of the generated trajectories and effectiveness in obstacle avoidance are investigated through a Monte Carlos analysis. The results show that the method is able to converge to a feasible and near-optimal trajectory within two seconds, except in very restrictive cases. Future work is being carried out

[6]The aircraft has a wingspan of approximately 2.5m. Therefore, 5m being twice the wingspan, is a measure of error based on wingspan.

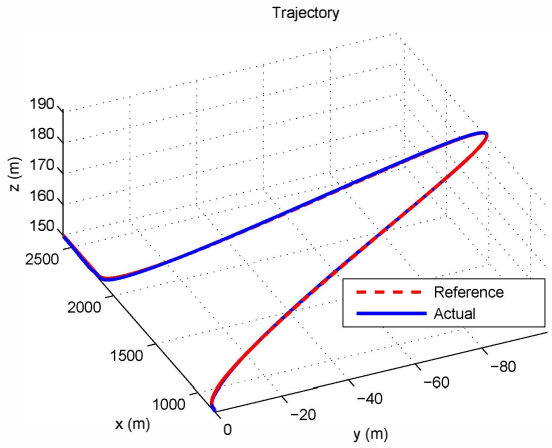

Figure 10. Comparison between the actual trajectory and the feedforward reference trajectory.

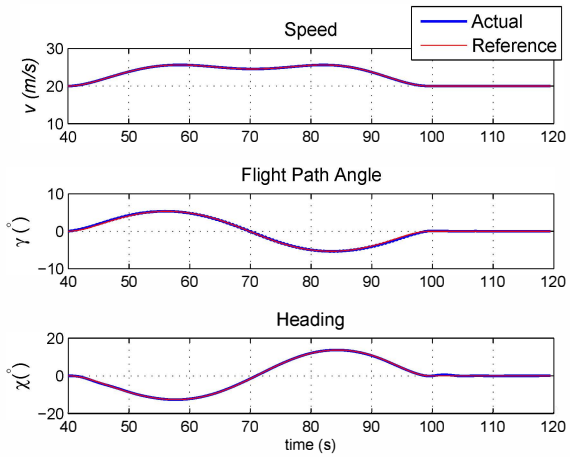

Figure 11. Feedforward reference speed, FPA and heading for a collision avoidance maneuver.

to deal with the non-convergent situations to further enhance the reliability. Moreover, the dynamic feasibility of the generated trajectories is verified via nonlinear simulations, where the trajectory generation is integrated with a 6 DoF nonlinear model of a fixed-wing research vehicle developed at Cranfield University. The results show that the generated trajectories can be tracked with the proposed two-degree-of-freedom control scheme. A regular update of the trajectory generator is being implemented to improve the tracking performance.

The improved convergence, fast computation and assured dynamic feasibility show potential for on-board implementation of the trajectory generator. Therefore an autonomous capability that accounts for changing environments can be achieved by in-flight reconfiguration of cost functions, models or constraints. Further simulations with uncertainty in obstacle sensing and environments (e.g. sensor noise and gust) will be carried out to pave the way for future flight testing.

ACKNOWLEDGMENTS

The authors would like to acknowledge Darren Ansell of BAE Systems for valuable discussions. This work is funded by BAE Systems and the EPSRC.

REFERENCES

[1] J. R. Wilson, "The time is now for sense and avoid," in *Aerospace America*. AIAA, June 2006.

[2] G. Fasano, D. Accardo, A. Moccia, C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles," *Journal of Aerospace Computing, Information, Communication*, vol. 5, pp. 338–360, 2008.

[3] P. Angelov, C. D. Bocaniala, C. Xideas, C. Patchett, D. Ansell, M. Everett, and G. Leng, "A passive approach to autonomous collision detection and avoidance in uninhabited aerial systems," in *Tenth International Conference on Computer Modeling and Simulation*, 2008.

[4] H.-S. Shin, A. Tsourdos, B. A. White, M. Shanmugavel, and M.-J. Tahk, "UAV conflict detection and resolution for static and dynamic obstacles," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, no. AIAA 2008-6521, Honolulu, Hawaii, August 2008.

[5] S. Luongo, F. Corraro, U. Ciniglio, V. D. Vito, and A. Moccia, "A novel 3d analytical algorithm for autonomous collision avoidance considering cylindrical safety bubble," in *IEEE Aeropsace Conference*, BigSky, Montana, March 2010.

[6] B. Zak, "The problems of collision avoidance at sea in the formulation of complex motion principles," *International Journal Applied Mathematics and Computer Science*, vol. 14, no. 4, pp. 503–514, 2004.

[7] D. H. Shim and S. Sastry, "An evasive maneuvering algorithm for UAVs in see-and-avoid situations," in *American Control Conference*, New York City, USA, 2007.

[8] J. M. Eklund, J. Sprinkle, and S. Sastry, "Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft," in *American Control Conference*, Portland, OR, USA, June 2005.

[9] D. H. Shim and S. Sastry, "A situation-aware flight control system design using real-time model predictive control for unmanned autonomous helicopters," in *AIAA Guidance, Navigation and Control Conference*, no. AIAA 2006-6101, Keystone, Colorado, August 2006.

[10] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, March-April 1998.

[11] P. Singla and T. Singh, "A novel coordinate transformation for obstacle avoidance and optimal trajectory planning," in *AIAA Astrodynamics Specialist Confer-*

ence and Exhibit, Honolulu, Hawaii, August 2008.

[12] R. B. Patel, P. J. Goulart, and V. Serghides, "Real-time trajectory generation for aircraft avoidance maneuvers," in *AIAA Guidance, Navigation and Control Conference*, no. AIAA 2009-5623, Chicago, Illinois, August 2009.

[13] C.-K. Lai and J. Whidborne, "Aircraft route re-planning for a pop-up obstacle using a direct method," in *UKACC International Conference on Control*, Coventry, UK, September 2010.

[14] O. A. Yakimenko, Y. Xu, and G. Basset, "Computing short-time aircraft maneuvers using direct methods," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, August 2008.

[15] D. W. Ansell, "Collision avoidance system," United State Patent US20 080 027 647A1, 2008.

[16] C. Patchett and D. Ansell, "The development of an advanced autonomous integrated mission system for uninhabited air systems to meet UK airspace requirements," in *International Conference on Intelligent Systems, Modelling and Simulation*, 2010.

[17] R. G. Drury and J. F. Whidborne, "A quaternion-based inverse dynamics model for real-time UAV trajectory generation," in *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, August 2009.

[18] R. Drury, A. Tsoudros, and A. Cooke, "Real-time trajectory generation: Improving the optimality and speed of an inverse dynamics method," in *IEEE Aeropsace Conference*, BigSky, Montana, March 2010.

[19] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an antunomous controller for a quadrotor UAV," in *European Control Conference*, Kos, Greece, July 2007.

[20] A. J. Berry, J. Howitt, D.-W. Gu, and I. Postlethwaite, "Continuous local motion planning & control for micro air-vehicles in complex environments," in *AIAA Guidance, Navigation and Control Conference*, no. AIAA 2010-7874, Toronto, Ontario Canada, August 2010.

[21] O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 865–875, September-October 2000.

[22] I. Kaminer, O. Yakimenko, V. Dobrokhdov, A. Pascoal, N. Hovakimyan, C. Cao, A. Young, and V. Patel, "Coordinated path follwing for time-critical missions of multiple UAVs via l1 adaptive output feedback controllers," in *AIAA Guidance, Navigation and Control Conference*, 2007.

[23] *Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System (F2411-04)*, ASTM International Std., 2004.

[24] P. R. Thomas, "Tools and techniques for the low cost flight testing of unmanned aerial vehicles," Ph.D. dissertation, Cranfield University, 2010.

[25] J. A. Lukacs, IV and O. A. Yakimenko, "Trajectory-shaping guidance for interception of ballistic missiles during the boost phase," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1524–1531, 2008.

BIOGRAPHIES

Chi-Kin Lai is currently a Ph.D. student at Cranfield University investigating real-time trajectory generation for guidance and control of UAVs. He received his M.Sc (Distinction) in Systems and Control from City University, London in 2008. He is a Student Member of the IEEE, the AIAA and the Royal Aeronautical Society. His research interests lie in the fields of optimal control and embedded optimization along with their applications for developing autonomous UAVs.

Mudassir Lone received his BA and MEng degrees in Aerospace and Aerothermal Engineering from the University of Cambridge in 2008. He is doing his PhD at Cranfield University in the field of man-machine interaction and aircraft handling qualities. His research interests include aeroservoelastics, optimal control and flight dynamics and control of unconventional aircraft. He is currently working as an aeroelastics engineer at Airbus UK. He is a Student Member of the Royal Aeronautical Society and the American Institute of Aeronautics and Astronautics (AIAA). He is also serving on the AIAA Atmospheric Flight Mechanics Technical Committee.

Peter Thomas completed his Ph.D. at Cranfield University in 2010, investigating applications of low cost tools and techniques for flight testing UAVs. He has developed and implemented both bespoke and off-the-shelf solutions for modeling, simulating, testing, and analysis of unmanned systems. This included the programming of attitude heading reference systems, graphical user interfaces, data parsing and analysis tools. He has also built several small remotely piloted research platforms for research and experimentation. He received a M.Eng. in Mechanical Engineering from Warwick University in 2006 and is an Associate Member of both the IMechE and the Royal Aeronautical Society.

James Whidborne received a B.A. degree in Engineering from Cambridge University in 1982, and an M.Sc. and a Ph.D. in Systems and Control from UMIST, Manchester, in 1987 and 1992 respectively. From 1982 to 1985 he worked in industry, and from 1986 to 1991, he was with the Control Systems Centre, UMIST. From 1991 to 1994, he held a position of Research Associate with the Department of Engineering, at the University of Leicester. From 1994 to 2003, he was a Lecturer, then Senior Lecturer with the Department of Mechanical Engineering, King's College London. He is currently a Senior Lecturer in the Department of Aircraft Engineering at Cranfield University, UK. He has over 160 research publications, including three books. His research interests include flight control, control of UAVs, robust control design and fluid flow control. He is a Chartered Engineer, a Member of the IET and a Senior Member of the IEEE.

Alastair Cooke is a Senior Lecturer in the Department of Aircraft Engineering within the School of Engineering, Cranfield University. He is a member of the Dynamics, Simulation and Control Group with a primary interest in the application of modelling, simulation and parameter estimation to handling qualities assessment for the purposes of clearance and qualification. He joined Cranfield University in September 2003 having previously been responsible for rotorcraft performance flight test at QinetiQ, Boscombe Down. Prior to this he worked for several years as the rotorcraft flight dynamics tutor at the Empire Test Pilot School, one of the premier flight test training establishments.