Collision avoidance within flight dynamics constraints for UAV applications

R. W. Penney
QinetiQ
Malvern, UK

ABSTRACT

Avoiding collisions with other aircraft is an absolutely fundamental capability for semi-autonomous UAVs. However, an aircraft avoiding moving obstacles requires an evasive tactic that is simultaneously very quick to compute, compatible with the platform's flight dynamics, and deals with the subtle spatio-temporal features of the threat.

We will give an overview of a novel prototype method of rapidly generating smooth flight-paths constrained to avoid moving obstacles, using an efficient trajectory-optimisation technique. Obstacles are described in terms of simple geometrical shapes, such as ellipsoids, whose centres and shapes can vary with time. The technique generates a spatio-temporal trajectory which offers a high likelihood of avoiding the volume in space-time excluded by the predicted motion of each of the known obstacles. Such a flight-path could then be passed to the aircraft's flight-control systems to negotiate the threat posed by the obstacles. Results from a demonstration implementation of the collision-avoidance technique will be discussed, including non-trivial scenarios handled well within 100ms on a 300MHz processor.

NOMENCLATURE

$\mathbf{a}_\mu$	trajectory basis-function weighting vector
H	energy measure of current UAV trajectory
$\mathbf{r}(t)$	position of UAV centre-of-mass as function of time
t	time co-ordinate
α_μ	centre-time of basis function μ
β	temporal-extent of trajectory basis function
λ_ρ	Lagrange-multiplier associated with constraint ρ
$\phi(t)$	basis-function for describing UAV flight-path
$\psi(\mathbf{r},t)$	bounding surface of obstacle

1.0 INTRODUCTION

There are many applications of unmanned air vehicles where enormous practical benefits can be achieved with un-piloted aircraft operating in relatively benign environments, for example high-altitude surveillance. In such circumstances, involving largely unrestricted airspace, remote piloting or relatively basic auto-piloting functions can be entirely satisfactory. However, as applications become more ambitious, and as the reliability of communications becomes more precarious, the need for a greater degree of autonomous behaviour becomes almost inevitable. This is especially true for military applications, where terrain-obscuration and congested airspace are endemic, and remote piloting is much less viable. Collision avoidance is an absolutely basic functionality that a semi-autonomous UAV must possess if it cannot rely on a human operator dealing with 'difficult' unexpected situations[1,2].

When dealing with collision-avoidance scenarios, human pilots will often quite effortlessly use an enormous amount of skill and intuition in devising an evasive manoeuvre. In common with many of the problems of artificially emulating human behaviour, it can be

Paper No. 2903. Manuscript received 22 July 2004, revised version received 23 February 2005, accepted 8 March 2005.
This paper was first presented at the 19th Bristol International UAV Systems Conferences.

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

a considerable technical challenge to construct an algorithm that can achieve a similar level of flexibility and robustness for even such a 'primitive' capability[3]. While one can imagine an approach based on codifying what human pilots say they do when faced with a threat, it is not clear that this is actually the best starting point given how much understanding and intuition will always be implicit in the human's descriptions.

In the rest of this paper we will discuss one possible approach to generating evasive manoeuvres for a UAV, based on an efficient trajectory-optimisation mechanism. We will begin in Section 2 with a brief discussion of the sense and avoid problem, followed in Section 3 with an overview of how we have chosen to formulate the avoidance aspect of this. Section 4 will then give a more technical description of our proposed algorithm, with some illustrative results in Section 5. We will then consider, in Section 6, how such an algorithm might be improved and the other allied capabilities a UAV might need to use it. Our conclusions are offered in Section 7.

2.0 SENSING AND AVOIDING

Clearly, a UAV needs to have some mechanism of recognising a potential collision if it is to make any attempt to avoid it. While some collisions may be avoidable through the use of static on-board information such as a terrain database[4] or through careful pre-planning of routes, in practice potential collisions must be dealt with dynamically. This necessitates both a method of acquiring information about potential hazards, and a mechanism for responding to them, neither of which is trivial. Indeed, there are many open questions about the precise regulatory specifications of a sense-and-avoid system, even though the necessity of such a capability and outline design requirements are clear[5].

2.1 Co-operating aircraft

When the main threat comes from cooperative aircraft, it is possible that warnings could be generated simply on the basis of each aircraft broadcasting its position (e.g. from GPS), heading etc. The information from other aircraft should then give each UAV a fairly accurate picture of the hazards in its surrounding airspace.

This is essentially the approach used to good effect by the FAA's traffic alert and collision avoidance system (TCAS)[6]. In general, this style of system would fit well with the existing navigation and communications capabilities likely to be part of any UAV, so impose little extra burden. However, for military operations wherein covertness may be critical, any collision-avoidance system based on extensive communication will prove problematic.

2.2 Active sensing

Without extensive inter-aircraft co-operation, quite a sophisticated sensing system may be required. For example, recent experiments on the NASA Proteus experimental aircraft have used a high-frequency radar scanning system[2].

For a basic collision-avoidance system, one can imagine using simple acoustic or optical sensors that purely indicate, on the basis of back-scattered intensity being above some threshold, whether the UAV is heading towards a potential threat. This could then be used as a stimulus for the UAV to turn until the sensor registers clearer airspace ahead. Naturally, this 'turn and hope' strategy might avoid some collisions with stationary or slow-moving bulky obstacles, but it is a rather limited safeguard.

Ideally, one would like the UAV to have at least position, velocity and bounding-volume information for all obstacles in its neighbourhood. This level of detailed information is probably most easily provided with a pulse-Doppler radar, or perhaps a pulsed laser system, combined with relatively standard tracking techniques[7]. One might imagine alternatives based on ultrasound, but these are likely to prove awkward in view of the high levels of acoustic noise, and the large Doppler-shifts expected. A stereoscopic imaging system might also be considered, but would likely require moderately sophisticated image-processing techniques to extract even quite basic range information about the obstacles, and might produce rather modest accuracy on the important quantity of range-rate.

2.3 Recognising the threat

If we assume that our UAV has a suitably capable sensing system to identify potential obstacles, there is still the question of whether and in what way the UAV reacts to them. The standard pilot's guideline is that a collision is inevitable if another aircraft remains at a fixed bearing (unless it is on a parallel path). What this lacks, and where a human pilot's intuition plays a key role, is in allowing a safe margin around other aircraft and in deciding how to manoeuvre so as to reduce the threat. It is also very difficult to generalise this principle to a scenario involving more than two aircraft, because it is no longer sufficient to describe things in terms of a single displacement from the UAV.

As it is not improbable that an aircraft will have to deal with more than well-separated obstacles, it would certainly be undesirable to have a mechanism that was unable to take account of inter-related threats. Things become more subtle when one recognises that the aircraft's current flight-path was probably chosen with a view to moving towards a particular direction to which it would like to return. Hence, the collision-avoidance manoeuvre really needs to be seen in context of near-term navigation, not just as a totally isolated reaction. Here again, a human pilot may well be able to make a rapid, subconscious, identification of those general directions towards which it would be safest to turn.

3.0 EFFICIENT GENERATION OF TACTICS

Let us now consider how one might formalise the process of responding to a potential collision, assuming that the UAV has a relatively detailed picture of the threats around it. To be as robust as possible, the process of generating the evasive tactic must meet the following criteria:

- the tactic must be quick to compute;
- it must be compatible with the aircraft's flight dynamics;
- obstacles must be treated as occupying a non-zero volume of space;
- it must be possible to use information about obstacles' expected future motion.

What we require is a simple representation of both the UAV's intended tactic, and of the excluded volume around obstacles: together these should allow us to rapidly test whether a prospective tactic is viable. This is perhaps simplest to achieve if we describe the tactic as a smooth flight-path in space-time, rather than in terms of quantities more directly related to control-surface positions, such as heading or banking demands. While this may mean that we need an unconventional form of control-law to translate the flight-path into actual control-surface positions, it does make it far easier to test whether a prospective manoeuvre avoids the obstacles, without having to repeatedly integrate cumbersome aerodynamic differential equations[8].

We would also like to avoid having to use a detailed model of the UAV's flight dynamics when testing whether a prospective flight-path is actually flyable.

One might be tempted to represent the UAV's trajectory using way-points or piecewise-linear trajectories, but the vertices are certainly not faithful models of the turns an aircraft is actually capable of, and this makes assessing the viability of such a trajectory

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

rather more awkward. Certainly, given enough vertices[9], or in larger-scale routing problems[10], such an approach can be useful, but it is preferable to work with trajectories that are intrinsically smooth.

By representing the UAV's prospective trajectory in terms of smooth basis functions (such as Gaussians or polynomials) and associated control-weights, it is much easier to produce a smooth guide-path that the actual aircraft is more likely to be able to follow closely. Moreover, with suitable basis functions, one can directly compute the velocity and acceleration at points along the trajectory, and these provide a convenient link with the aerodynamic limits on the airframe.

At its simplest, we might imagine that a flight-path should try to minimise the typical acceleration of the airframe, as this will be indicative of the typical structural forces. More ambitiously, we might wish to constrain the trajectory to have hard limits on the maximum acceleration and on the maximum and minimum speed. We will see shortly (Section 4) that minimising the typical acceleration is very easy to achieve with the basis-function representation, and is already well-aligned with considerations of flight dynamics.

In order to represent the obstacles, it is natural to encapsulate these as some form of bounding volumes, and simple geometrical figures such as spheres or ellipsoids will be particularly convenient. Such forms not only allow one to very rapidly check whether a prospective trajectory passes through an obstacle, but are also well matched to the Gaussian distribution of estimated obstacle location that might emerge from a Kalman-filter tracker[7] within the UAV's sensing system.Thus, we could describe each obstacle as an ellipsoid whose centre position, radii, and orientation have values given, as functions of time, by the predictions of the sensing system.

With the UAV's trajectory parameterised by the weightings assigned to each of the smooth basis functions, the task of avoiding the obstacles consists of adjusting those weightings until the resulting trajectory satisfies the constraint of lying outside all the obstacles while having as small a value of typical acceleration as possible.

4.0 COLLISION-AVOIDANCE ALGORITHM

Now that we have given a broad overview of our formulation of the collision-avoidance problem, we can turn to considering its implementation in slightly greater detail.

4.1 Representing smooth trajectories

Firstly, in order to allow us to work with intrinsically smooth trajectories, we will assume the existence of a set of basis-functions $\phi_\mu(t)$, each of which might resemble a localised disturbance (near $t = \alpha_\mu$) such as

$$\exp -\beta(t-\alpha_\mu)^2 \quad (t-\alpha_\mu)\exp -\beta(t-\alpha_\mu)^2 \qquad \ldots (1)$$

which are illustrated in Fig. 1. The width-parameter β and the choice of values for $\{\alpha_\mu\}$ will depend on the typical timescales associated with the UAV's dynamics.

Given such a basis, we can represent a smooth trajectory by combining our chosen $\phi_\mu(t)$ with a set of vector-valued weightings $\{\mathbf{a}_\mu\}$:

$$\mathbf{r}(t) = \sum_{\mu=1}^{k} \mathbf{a}_\mu \phi_\mu(t) \qquad \ldots (2)$$

The optimisation of the shape of the trajectory then reduces to adjustment of the $\{\mathbf{a}_\mu\}$, which might typically consist of about a few dozen vectors.

We will also have to arrange that our trajectory starts at the UAV's current position, and ends somewhere desirable, for example the next way-point provided by a higher-level routing or navigation system. With straightforward choice of the spacing of the centres $\{\alpha_\mu\}$ of the basis-functions Equation (1), one can arrange that only a small number of the $\{\mathbf{a}_\mu\}$ contribute significantly to these boundary values. This allows one directly to prescribe values for those weights so that these boundary conditions are immediately (closely) satisfied.

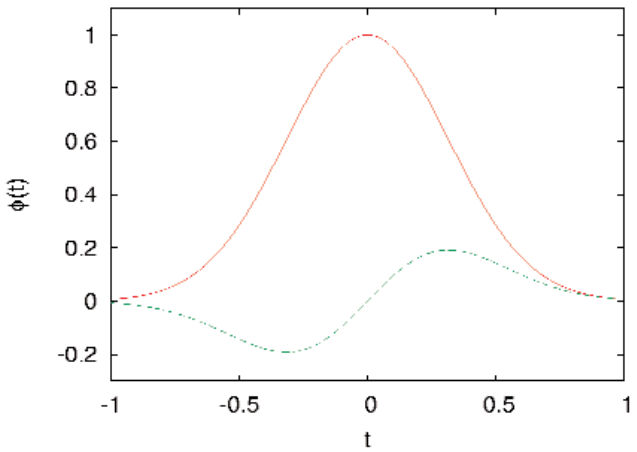


Figure 1. Example basis functions $\phi(t)$ used for representing the UAV trajectory.

In the absence of any obstacles, we need some measure of the quality of a trajectory. As discussed in Section 3, we choose a cost-function that is related to the typical acceleration over the trajectory:

$$H_0 = \frac{1}{t_1 - t_2} \int_{t_0}^{t_1} \left| \ddot{\mathbf{r}}(t) \right| \mathrm{d}t \qquad \ldots (3)$$

Thus, H_0 measures the mean-square acceleration over the time-interval $[t_0, t_1]$, which will certainly be large if our choice of $\{\mathbf{a}_\mu\}$ produces an unduly oscillatory trajectory.

With our choice of the form of $\mathbf{r}(t)$ Equation (2), much of the calculation of the cost-function Equation (3) for any prospective trajectory can be pre-computed in terms of the following matrix:

$$M_{\mu\nu} = \frac{1}{t_1 - t_0} \int_0^1 \ddot{\phi}_\mu(t) \ddot{\phi}_\nu \mathrm{d}t \qquad \ldots (4)$$

which is entirely determined by (derivatives of) the (fixed) basis functions.

We now have a cost-function of the following form:

$$H_o = \sum_{\mu,\nu=1}^{k} M_{\mu\nu} \mathbf{a}_\mu \cdot \mathbf{a}_\nu \qquad \ldots (5)$$

In the absence of any obstacles, or additional aerodynamic constraints, the values of the $\{\mathbf{a}_\mu\}$ not fixed by the starting/ending boundary conditions can now be very readily found by minimising a quadratic function Equation (5).

4.2 Adding obstacles

Naturally, when obstacles are present, the optimisation of the $\{\mathbf{a}_\mu\}$ is less straightforward. The first step is to define a function $\psi_i(\mathbf{r},t)$ that defines the edges of each obstacle i. For example, for a spherical obstacle of radius d, one can define a function $\psi_i(\mathbf{r})$ that is zero on the surface and greater than zero outside:

$$\psi(r) = |\mathbf{r} - \mathbf{c}|^2 - d^2 \qquad \ldots (6)$$

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

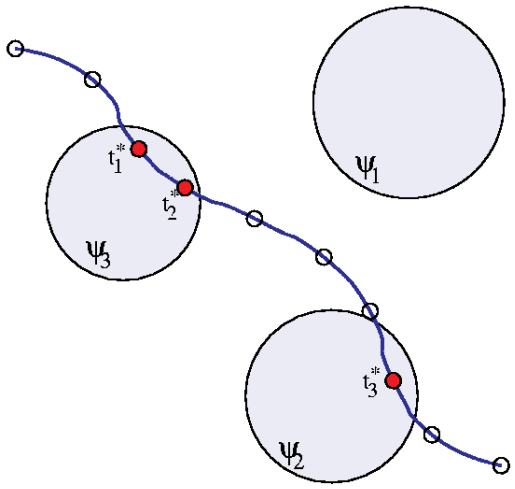


Figure 2. Schematic illustration of the process of identifying which obstacle-constraints are active. The current trajectory (blue, heavy, line) is examined at a selection of points (small circles), and where these lie within the interior of an obstacle (shaded larger circles), they are added to the list of active constraints (red, filled, small circles).

In general, we will also want to allow for the centre-position c and radius d to be functions of time, and extend this to ellipsoidal obstacles —- all these steps are straightforward and not essential for our subsequent discussion.

We need to treat differently circumstances where the trajectory lies outside an obstacle and is therefore essentially unconstrained by it, *versus* circumstances where the trajectory needs to be constrained to lie on the surface of an obstacle. As is standard in the handling of inequality constraints[11], this leads to the idea of an active set of constraints. This active set of constraints can be identified by examining a selection of regularly-spaced points over the trajectory in the interval $[t_0,\ t_1]$, and checking whether each $\psi_i(\mathbf{r},t) > 0$ (see Fig. 2).

Where a particular trajectory passes through the obstacle, the associated ψ_i and time t_i^* are noted and used to define a constraint whereby the trajectory is forced to lie on the surface of the obstacle.

This leads to the construction of a cost-function of the following form:

$$H = H_0 + \sum_{\rho} \lambda_\rho \psi_\rho \left(\mathbf{r}\left(t_\rho^*\right), t_\rho^* \right) \qquad \ldots (7)$$

wherein the mean-square acceleration of H_0 is augmented by a set of Lagrange multipliers $\{\lambda_\rho\}$ and constraint functions ψ_ρ. To avoid a profusion of unwanted indices, we will assume that the index ρ labels both the obstacle itself and the point where it touches the trajectory at time t_ρ^*, and the summation over ρ covers all the active constraints. Naturally, the contact-times t_ρ^* are defined implicitly in terms of the weightings $\{\mathbf{a}_\mu\}$ which define the shape of the trajectory.

4.3 Hard flight-dynamics constraints

The use of the UAV's mean-square acceleration as the foundation of our cost-function Equation (3) already means that the trajectory-optimisation process is likely to favour trajectories which do not place excessive aerodynamic loading on the airframe unless the obstacle-avoidance constraints Equation (7) force it to do otherwise. Thus, provided the UAV is not operating in extremely hazardous circumstances, it may be unnecessary to explicitly constrain the trajectory to obey any additional dynamic constraints.

If it is necessary to have greater assurance that the optimised trajectory will not violate the capabilities of the aircraft, for example by having segments that are too slow to achieve adequate lift, then additional constraints must be imposed. These can be handled with a very similar approach to the obstacle-avoidance constraints, although certain types of dynamical constraint are better suited to the representation we have chosen for the UAV's trajectory Equation (2).

We can easily evaluate the velocity and acceleration of the platform at any point along its trajectory, for example:

$$\dot{\mathbf{r}}(t) = \sum_{\mu=1}^{k} \mathbf{a}_\mu \dot{\phi}_\mu(t) \qquad \ldots (8)$$

This means that we can express the constraint that the speed of the UAV at any particular time t is at least v_0 by an expression of the form:

$$\sum_{\mu\nu} C_{\mu\nu}(t) \mathbf{a}_\mu \cdot \mathbf{a}_\nu > v_0^2 \qquad \ldots (9)$$

where the matrix $C_{\mu\nu}(t)$ can be pre-computed for a selection of sample-times $\{t\}$ in terms of derivatives of the basis functions $\{\phi_\mu(t)\}$.

Hence, we can use a similar process of examining a set of sample points along the trajectory in order to check whether dynamical constraints of the form Equation (9) are satisfied. As before, those inequality constraints that are not satisfied are then treated as equality constraints through additional terms in the augmented cost-function:

$$H = H_0 + \sum_{\rho} \lambda_\rho \psi_\rho \left(r\left(t_\rho^*\right), t_\rho^* \right) + \left\{ \sum_{\mu,\nu=1}^{k} C_{\mu\nu}(t) \mathbf{a}_\mu \cdot \mathbf{a}_\nu - v_0^2 \right\} \qquad \ldots (10)$$

4.4 Trajectory optimisation

Having constructed the augmented cost-function Equation (10), the collision-avoidance problem now reduces to adjusting the weightings $\{\mathbf{a}_\mu\}$ so that H is extremised.

In practice, this must be achieved through an iterative process involving:

- adjusting the $\{\mathbf{a}_\mu\}$ to modify the shape of the trajectory;
- adjusting the $\{\lambda_\rho\}$ to enforce those contact constraints that are active;
- and re-examining the set of active constraints to establish which obstacles λ_ρ and times t_ρ^* need to be active.

The process of adjusting the $\{\mathbf{a}_\mu\}$ & $\{\lambda_\rho\}$ involves computing the derivatives of H Equation (10), and trying to make these derivatives zero, which can be achieved through well-known methods[12]. Naturally, as constraints become active or are deactivated, the set of Lagrange multipliers $\{\lambda_\rho\}$ may grow or shrink, but this is largely only a matter of bookkeeping. When the set of active constraints remains stable, and the $\{\mathbf{a}_\mu\}$ & $\{\lambda_\rho\}$ are changing insignificantly, the optimisation can be deemed to have converged at least to a local optimum. If time permits, one can perform a succession of such optimisations from different starting values for the $\{\mathbf{a}_\mu\}$ and pick that optimum which has the lowest value of H_0 Equation (3).

5.0 EXAMPLE RESULTS

We can now turn to examining how our proposed collision-avoidance mechanism deals with some stylised scenarios. To try to keep our discussion generic, and not tied to the capability of particular aircraft, we will rescale times and distances to be of order

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

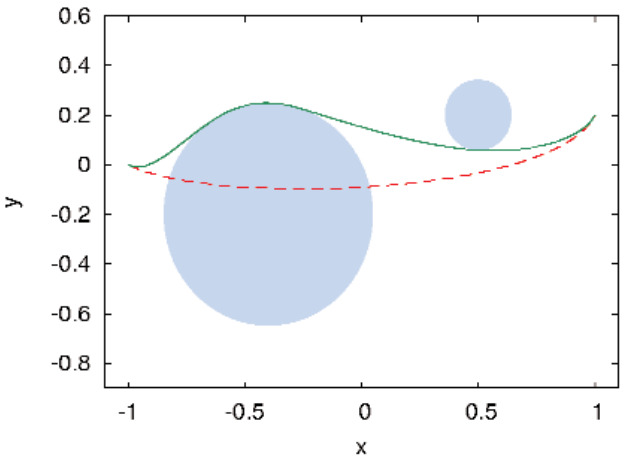


Figure 3. Schematic illustration of a two-dimensional collision-avoidance problem. The obstacles are represented by the shaded areas, and the initial trajectory is shown by the (dotted) red curve. After optimisation, the (solid) green trajectory is seen to neatly skirt the edges of the obstacles, while still obeying the same boundary-conditions as the original trajectory.

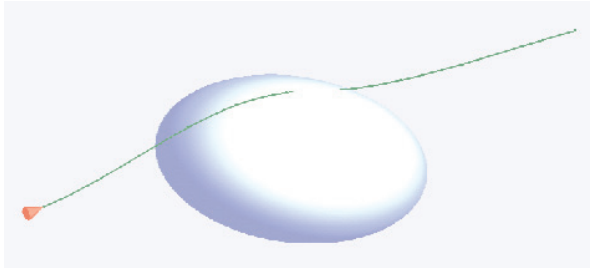


Figure 4. Illustration of the optimised trajectory for avoiding a static ellipsoid in three dimensions. The initial trajectory (unshown) passes roughly through the centre of the ellipsoid, but requires only about 35ms of computation to be converted into a smooth evasive manoeuvre. The slight remaining penetration of the trajectory into the obstacle is a consequence of the finite number of control parameters available to the algorithm.

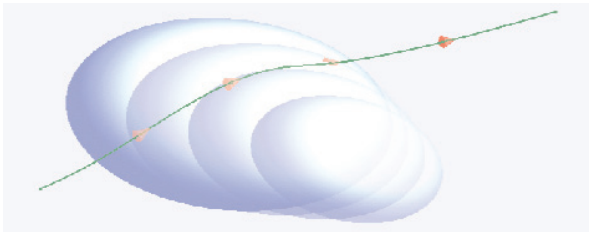


Figure 5. Time-lapsed illustration of the optimised trajectory for a moving ellipsoidal obstacle whose radii and orientation change over time. The ellipsoid is moving into the picture, with its radii growing as the UAV passes from left to right.

unity, and choose the speeds of any moving obstacles to be comparable to that of the UAV.

The results here are from an implementation of our algorithm in C++, based on the ideas presented in Section 4. In each case, our trajectory will be defined by 24 nodes ($k = 24$ in Equation (2), four of which are fixed by the initial/endpoint conditions imposed at t = t_0, $t = t_1$. Where we indicate computation time of our algorithm, these refer to a 300MHz Pentium II running Linux 2.4.10. Naturally, these timings are strongly influenced by the degree of care taken in implementation, but should be indicative of what could be achieved in a deployable system.

5.1 Two dimensions

Consider first a two-dimensional obstacle-avoidance problem, as shown in Fig. 3. This consists of two circular static obstacles, and a vehicle whose starting and ending locations and velocities are set to make collision almost inevitable. The initial vehicle trajectory is constructed by optimising purely the mean-square acceleration Equation (3), which indeed produces a smooth flight-path which neatly asymptotes to the endpoint conditions, but passes through the larger of the obstacles.

If this initial trajectory is passed again to the optimisation routine, but taking account of the obstacles Equation (7), then various points along the trajectory are immediately added to the set of active constraints and the trajectory is progressively deformed to avoid the obstacles. Indeed (Fig. 3), after only 80ms processing, the optimised trajectory is seen to neatly manoeuvre around the edges of the obstacles without any unreasonably dramatic turns.

5.2 Three dimensions

Let us now examine more ambitious scenarios in three dimensions, which will naturally be more difficult to depict on paper.

As an elementary example of how the algorithm handles three-dimensional obstacles, consider manoeuvring around a static ellipsoid (Fig. 4). Clearly, with only a limited number of degrees of freedom, the algorithm will be unable to perfectly track the edges of a smooth obstacle, so it is unsurprising that the optimised trajectory slightly penetrates the surface of the ellipsoid (by less than about 10% of its minimum radius). Nevertheless, the optimisation has clearly generated a smooth and effective manoeuvre that is likely to be close to the best that is achievable with the available degrees of freedom, and required only 35ms of computation.

It is interesting to note that the computational load for this scenario is substantially less than that of the earlier two-dimensional problem (Fig. 3). While the increased spatial dimensionality tends to increase the computational cost, the number of iterations required by the algorithm, and the number of changes in the active set of constraints, will strongly depend on the arrangement of the obstacles. It is probable that finding a trajectory that neatly meanders between two two-dimensional obstacles (Fig. 3) is actually significantly more difficult than manoeuvring around relatively benign three-dimensional obstacles.

If we further increase the complexity of our scenario to allow the ellipsoid to move, change radii and even slightly rotate (Fig. 5), then the situation more closely resembles that of a UAV handling the output of a sensor system described in terms of a Gaussian probability density function. Here again, the algorithm requires only about 35ms to compute an optimised trajectory that adroitly moves around the edge of the assumed motion of the obstacle, with minimal penetration into the excluded volume.

So far we have been examining cases where one might argue that a carefully crafted prescriptive tactic might be sufficient to reduce the threat posed by an obstacle. Given that it would be rather ambitious to achieve this for situations involving more than a single obstacle, let us briefly consider the scenario of Fig. 6. Here the UAV is confronted by two moving obstacles each of which crosses its original flight-path. Although the algorithm requires more computation to tackle such a combination of threats (here about 110ms), the tactic that emerges is again smooth and economical in its response.

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

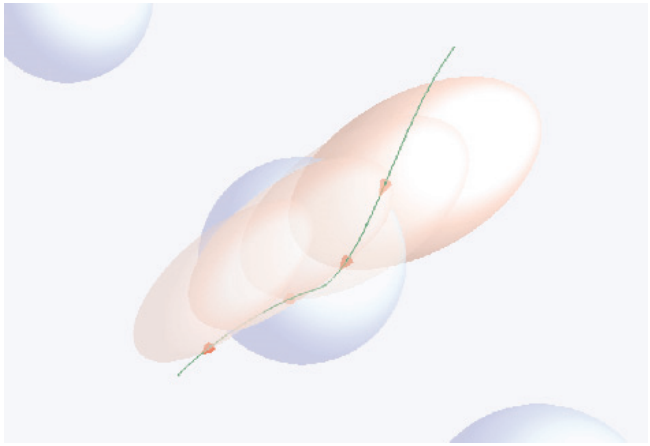


Figure 6. Time-lapsed illustration of a scenario involving two obstacles. The ellipsoid grows in size at it moves from bottom left to top right, while the sphere also grows as it moves more quickly from top left to bottom right. (Only the first three frames of the sphere's motion are visible.) The optimised trajectory shows the UAV moving from top right to bottom left, with minimal penetration into the volume excluded by the obstacles.

6.0 FURTHER WORK

Clearly, there are many aspects to producing an operational collision-avoidance system, and we have here concentrated only on the manoeuvre-generation problem. We will briefly explore how our proposed approach might fit into an overall system, and some further issues that need to be addressed within its development.

6.1 Platform interface

We have already noted that our collision-avoidance algorithm generates merely a guide-path for the aircraft, specifying an intended position at each future time, dependent on the assumed future motion of the obstacles at the instant when the optimisation was performed.

For such a guide-path to be of use, one must have a control law that can manoeuvre the aircraft actually to follow this trajectory. While one can imagine simply breaking-down the trajectory into many way-points and adopting a fairly standard auto-piloting function, this is likely to be quite precarious.

Our preliminary experiments using a fairly detailed aerodynamic model of a small UAV have provided some reassurance that it is possible to formulate a low-cost control-law that is able to steer a UAV along a sinuous flight-path such as would be generated by our algorithm. While this work is at an early stage, it is likely that a more polished version of such a control-law would not pose fundamental challenges. Indeed, the algorithm of Section 4 should generally produce trajectories sufficiently flyable that they would not require an overly sophisticated control-law in order to be followed quite accurately.

6.2 Terrain avoidance

A significant feature absent from the example scenarios of Section 5 is terrain. For UAVs operating close to the ground, it would be essential to include the terrain surface as another 'obstacle' to avoid.

There is no fundamental difficulty in simply defining another function $\psi_i(\mathbf{r})$ that measures the height of the UAV above the ground, as determined from a terrain database, and proceeding exactly analogously to the case of simpler-shaped obstacles. However, such an approach is likely to be unnecessarily computationally costly, as interrogating the terrain database is likely to be far more expensive than testing whether a given point lies outside an ellipsoidal obstacle.

In practice, therefore, it is likely that the terrain surface would have to be pre-processed into a set of simpler bounding shapes, such as planes, large ellipsoids, etc. Although this does not seem to be a major difficulty, it is still a subject of future work as regards our approach to collision-avoidance.

6.3 Flight clearance

Flight clearance and safety must obviously be central to practicable UAV systems, and collision-avoidance perhaps poses some particular difficulties[5]. By definition, a collision-avoidance mechanism would be redundant unless the UAV was in some danger of causing damage or being damaged.

Certainly there are safeguards that one would expect to impose, such as checks that the mechanism doesn't trigger outside well-defined circumstances, and that the proposed avoidance tactic doesn't attempt to go under the ground etc. But given that the threat of collision will not be under the UAV's exclusive control, it would be difficult to specify performance criteria based on rigid guarantees that a collision-avoidance algorithm will always successfully avoid collision — some form of probabilistic assessment would seem more appropriate.

It would require substantial further work to give a meaningful estimate of the probability of our algorithm successfully saving an aircraft from a given type of collision, but experience across a good range of simulated scenarios strongly suggests that it might offer a UAV a very worthwhile chance of escape.

One important aspect within such a validation process would be the performance of an automatic collision-avoidance process when operating near piloted aircraft. While it will always be extremely difficult for the UAV to make accurate predictions about the future motion of other aircraft, especially if they themselves are reacting to the impending collision, there are certain conventions that human pilots are likely to follow. For example, knowledge that a human pilot may well turn to starboard rather than to port, could be quite significant in narrowing-down the volume that the UAV assumes it has to avoid. Within our current formulation, such prior knowledge will primarily influence the tracking process that provides the estimated obstacle trajectories, but it may also be advantageous to try to bias the algorithm to generate avoidance tactics that also comply with convention. The implications of both these aspects require further study.

7.0 CONCLUSIONS

We have outlined the challenges faced by a UAV in responding to a potential collision, and noted that the process of rapidly generating a successful evasive tactic may need to be quite sophisticated. A low-cost algorithm based on a constrained trajectory-optimisation process has been proposed, using a formulation that guarantees a smooth flight-path will emerge. We have considered how obstacles and flight dynamics constraints may be represented within the formulation. Some stylised example scenarios have been discussed, involving both stationary and moving obstacles, where the algorithm appears able to generate successful yet economical evasive tactics with limited computational resources.

Current work is extending the approach to multi-vehicle scenarios, and integrating it with higher-level navigation algorithms. Further work will be needed to ensure that the algorithm can be integrated effectively into an actual aircraft system, and to provide the safeguards and assurances needed for flight clearance.

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press

ACKNOWLEDGEMENTS

The author would like to thank Richard Evans, Tom Cooper and David Walker (QinetiQ) for their assistance and stimulating discussions in connection with this work. We are also grateful to Andy Walster (Cranfield Aerospace) for informative discussions of the Observer UAV, and to Phill Smith (Blue Bear Systems Research) for bringing various references to our attention. Comments from the anonymous reviewers of this paper have also been very helpful.

Financial support by the UK Ministry of Defence Corporate Research Programme is gratefully acknowledged.

REFERENCES

1. LAKE, M.F. Collision avoidance – a pilot's perspective, Proceedings 17th International Conference on UAV Systems, 2002, Bristol, UK.
2. LOPEZ, R. Avoiding Collisions in the age of UAVs, *Aerospace America*, **27**, June 2002.
3. KUCHAR, J.K. and YANG, L.C. Review of conflict detection and resolution modeling methods, IEEE trans *Intelligent Transportation Systems*, 2000, **1**, (4), pp 179-189.
4. BARFIELD, A.F., PROBERT, J. and BROWNING, D. All terrain ground collision avoidance and maneuvering terrain following for automated low level night attach, *IEEE Aerospace and Electrical Systems Magazine*, 1993, **8**, (3), pp 40-47.
5. *CAA Directorate of Airspace Policy; Unmanned Aerial Vehicle Operations in UK Airspace – Guidance*, 2004, Civil Aviation Authority, CAP722.
6. WILLIAMSON, T. and SPENCER, N.A. Development and operation of the traffic alert and collision avoidance system, Proceedings of IEEE, 1989, **77**, (11), pp 1735-1744.
7. BAR-SHALOM, Y. and FORTMANN, T.E. *Tracking and Data Association*, 1988, Academic Press.
8. COOK, M.V. *Flight Dynamics Principles*, 1997, Butterworth Heinemann.
9. HU, J., PRANDINI, M., NILIM, A. and SASTRY, S. Optimal coordinated maneuvers for three dimensional aircraft conflict resolution, 2001, Proceedings of AIAA conference on Guidance, Navigation and Control, Montreal.
10. BOKADIA, S. and VALASEK, J. Severe weather avoidance using informed heueristic search, 2001, Proceedings of AIAA conference on Guidance, Navigation and Control, Montreal.
11. FLETCHER, R. *Practical Methods of Optimization*, 1987, Wiley.
12. PRESS, W.H. TEUKOLSKY, S.A., VETTERLING, W.T. and FLANNERY, B.P. *Numerical Recipes in C*, 1992, Cambridge University Press.

https://doi.org/10.1017/S0001924000000695 Published online by Cambridge University Press