Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)

4-D Trajectory Optimizers for Conflict Avoidance Using Speed Advisories

Oliver Turnbull, Arthur Richards
Department of Aerospace Engineering
University of Bristol
Bristol, UK
Oliver.Turnbull@bristol.ac.uk, Arthur.Richards@bristol.ac.uk

***Abstract*—This paper extends 4-D trajectory optimizers to resolve conflicts through "speed advisories", separating high-level decision making from the detailed trajectory optimization. Details of Mixed Integer Linear Programming (MILP) and collocation optimizers are briefly reviewed before the additional constraints are developed to force resolution only through speed changes. Results for a multi-sector area over Wales and North-West England illustrate how the method can be used. A brief evaluation of the computational complexity of the method is shown.**

Keywords- 4-D trajectory optimization; air traffic control, conflict resolution; speed advisory

I. INTRODUCTION

The SESAR Concept of Operations [4] for future Air Traffic Management (ATM) calls for "extensive use of automation support to reduce operator task load, but in which *controllers remain in control as managers*". The method presented in this paper follows a paradigm of separating high-level decision making, such as "flight A should pass above flight B", from the low level detailed trajectory design, i.e. 4-D waypoints.

The authors' previous work [15] developed two 4-D trajectory optimizers for aircraft conflict avoidance. The idea of *sense constraints* was introduced to allow a human supervisor to make high-level decisions regarding the resolution of conflicts between aircraft. Constraint forms to enforce behavior such as horizontal and altitude conflict resolution as well as the relative location of each aircraft., e.g. above/below or ahead/behind, were presented. The idea is to provide constraints for a selection of high-level resolution strategies: the human chooses the strategy and the optimizer finds a corresponding trajectory.

Another form of conflict resolution is for one or more conflicting aircraft to alter only their airspeeds to ensure safe separation is maintained at all times. This could be implemented by issuing a "speed advisory" to one or more of the aircraft. Speed resolution has been studied previously, e.g. [20],[21]. Vela et al [20] developed a MILP for Flight-Level and speed assignment on fixed routes while Cruck and Lygeros [21] demonstrated in the ERASMUS project [22] that over long time scales, autonomous small speed changes can be effective enough to avoid conflicts ever requiring resolution by the human controller. Speed changes therefore present an attractive strategy for conflict resolution, since they can resolve conflicts without significant path changes and knock-on controller workload. This paper presents constraints that can be added, if requested by a controller, to 4-D trajectory optimizers such that selected conflicts are resolved through speed advisories.

There has been a large volume of research into optimal trajectory optimization for conflict avoidance [1],[2],[3]. However, there has been comparatively little consideration of how humans can be included within such systems [21]. Most work focuses on exploiting as much flexibility as can be modelled in an optimizer. This paper focuses on constraining the optimizer to meet a human requirement – the choice of speed change only – while retaining as much flexibility as possible in other decisions.

The first part of this paper reviews a Mixed-Integer Linear Programming (MILP) approach [3], to solve the global, non-convex conflict resolution optimization including a representative 3-D dynamics model [15], [16] from BADA [10]. MILP captures the discrete decision making within the problem, such as conflict avoidance, with binary decision variables. It has been chosen here as it is extensible and, with commercial software, e.g. [6], reliable to solve. Nonlinear optimization has also been proposed for trajectory generation [7], [8], although convergence can be a challenge and global optimality is not guaranteed. The second part of this paper takes the collocation method proposed in [9] and the obstacle avoidance method of Patel and Goulart based on polar sets [7] and adds different constraints to implement speed advisories. Although harder to solve for the global optimum than MILP, the nonlinear model does admit a wider variety of dynamics and costs, such as noise [12].

The remainder of this paper is organized as follows: Section II defines some nomenclature; Section III outlines the general problem statement; Section IV reviews the MILP optimizer and develops the additional constraints for conflict resolution through speed changes; Section 0 mirrors section IV but for the collocation optimizer; Section VI applies the MILP method to a Multi-Sector Area; Section VII presents some initial computational complexity results; Finally Section VIII draws some conclusions.

This paper describes results from the SUPEROPT project that is part of SESAR WorkPackage E, which is addressing long-term and innovative research.

II. NOMENCLATURE

The nomenclature used in this paper is defined in TABLE I.

TABLE I. NOMENCLATURE

Symbol	Definition
N_a	Number of aircraft
a,b	Index of aircraft
$\mathbf{r}(a,t)$	Position of *a* at time *t* (decision variable)
$r_d(a,t)$	Position of *a* at time *t* in dimension *d*
$\mathbf{r}^R(a,t)$	Reference trajectory of *a* at time *t* (fixed parameter)
$\mathbf{f}(a,t)$	Acceleration of *a* at time t
k	Index of time step
t_k	Sample time *k*
N_t	Number of times-steps
$t_f^R(a)$	Final/exit time of *a* on reference trajectory (fixed parameter)
$t_f(a)$	Final/exit time of *a* (decision variable)

III. PROBLEM DESCRIPTION

Fig. 1 illustrates a typical problem instance. Define N_a aircraft and assume that each aircraft *a* has Reference Business Trajectory (RBT) $\boldsymbol{r}^R(t,a)$, as described in the SESAR Concept of Operations [4], running from $t=t_0$, the current time, to $t=t_f^R(a)$, the reference time at which the RBT ends; the immediate destination of aircraft *a* is defined by $\boldsymbol{r}^R(t_f^R(a),a)$. This would typically refer to the pre-determined point, e.g. in the Shared Business Trajectory (SBT), at which the aircraft is expected to exit the sector or multi-sector area. The optimizer designs for each aircraft a trajectory $\boldsymbol{r}(a,t)$ starting also from time t_0, i.e the current time, to $t_f(a)$, the new chosen time at which *a* exits the area. Finally, since a numerical optimizer can only have a finite number of constraints, define discrete time step variable *k* to index a set of N_t sampling times between t_0 and $t_f(a)$. Constraint and cost evaluation will be performed at these points.

The cost function is defined as:

$$J = \sum_{a=1}^{N_a} \alpha t_f(a) + \beta \|\mathbf{r}(t_f(a),a) - \mathbf{r}^R(t_f^R(a),a)\| + \sum_{a=1}^{N_a}\sum_{k=1}^{N_t} \gamma \|\mathbf{f}(a,t_k)\| + \delta \|\mathbf{r}(t_k,a) - \mathbf{r}^R(t_k,a)\| \quad (1)$$

where: the first term with a weighting on the final time reflects the desire to avoid delay; the second term penalizes co-ordinations with the adjoining area with a weighting on deviation from the exit point; the third time is included to reduce manoeuvring and increase passenger comfort with a weighting on acceleration *f*; and the final term reflects the desire to stay close to the RBT throughout the MSA. The relative importance of the different terms is adjusted via the weights α,β,γ,δ.

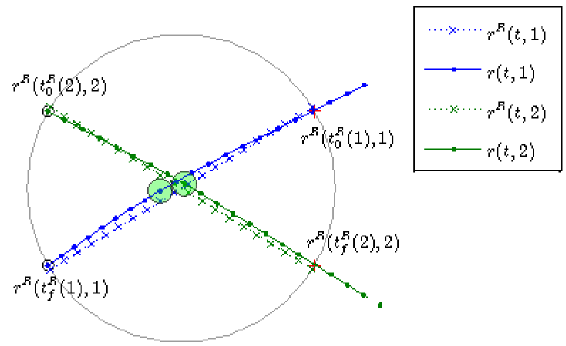


Figure 1: Example of trajectory definitions

IV. MILP

Both the MILP and collocation based optimizers have previously been described in detail [15], [16], [19] but are reviewed here for completeness.

A. Review of Method

The dynamics model is based on the EUROCONTROL Base of Aircraft Data (BADA) [10] which provides both an analytical model and a database of aircraft performance for typical commercial aircraft. A simple point mass model is assumed such that:

$$\mathbf{r}(a,k+1) = \mathbf{r}(a,k) + \mathbf{v}(a,k)\Delta t\,, \quad (2)$$

$$\mathbf{v}(a,k+1) = \mathbf{v}(a,k) + \mathbf{f}(a,k)\Delta t\,, \quad (3)$$

where $\boldsymbol{r}(a,k)$ and $\boldsymbol{v}(a,k)$ are the position and velocity of aircraft *a* at time step *k* respectively. Further constraints are placed on the upper and lower limits of the aircraft velocity and accelerations, the rates of climb and descent as well as each aircraft's arrival and departure points (see [15] for more details).

Conflict avoidance between each pair of vehicles, a_1 and a_2, is enforced by N_θ constraints in the horizontal plane and two in the vertical direction:

$$cos\left(\frac{2\pi\theta}{N_\theta}\right)[r_x(a_1,k_1) - r_x(a_2,k_2)] - sin\left(\frac{2\pi\theta}{N_\theta}\right)[r_y(a_1,k_1) - r_y(a_2,k_2)] \geq D_H - Mb_a(a_1,a_2,k_1,k_2,\theta), \quad (4)$$

$$r_z(a_1,k_1) \leq r_z(a_2,k_2) - D_V + Mb_a(a_1,a_2,k_1,k_2,N_\theta+1), \quad (5)$$

$$r_z(a_1,k_1) \geq r_z(a_2,k_2) + D_V - Mb_a(a_1,a_2,k_1,k_2,N_\theta+2), \quad (6)$$

$$\forall\ a_1 \in \{1,\ldots,N_a\}, a_2 \in \{1,\ldots,N_a\}, k_1 \in \{2,\ldots,N_t\}, k_2 \in \{2,\ldots,N_t\}, \theta \in \{1,\ldots,N_\theta\} : (k_1 = k_2) \cap (a_1 > a_2),$$

where D_H and D_V are the horizontal and vertical separation distances respectively; *M* is a large constant; $b_a(a_1,a_2,k_1,k_2,\theta)$ is a binary variable to selectively relax the constraint between aircraft a_1 and a_2 at time-step k_1 (of a_1's) trajectory and k_2 (of a_2's trajectory) in direction θ. To ensure conflict avoidance, at least one of the constraints must hold at each time step:

$$\sum_{\theta=1}^{N_\theta+2} b_a(a_1,a_2,k_1,k_2,\theta) = (N_\theta+1) \quad ,(7)$$

Finally, avoidance between time-steps [16] is ensured by enforcing the avoidance binaries at all combinations of the current and previous time-steps k_1, k_2, k_1-1 and k_2-1:

$$cos\left(\frac{2\pi\theta}{N_\theta}\right)[r_x(a_1,k_1)-r_x(a_2,k_2-1)]-sin\left(\frac{2\pi\theta}{N_\theta}\right)[r_y(a_1,k_1)-r_y(a_2,k_2-1)] \geq D_H - Mb_a(a_1,a_2,k_1,k_2,\theta), \quad (8)$$

$$cos\left(\frac{2\pi\theta}{N_\theta}\right)[r_x(a_1,k_1-1)-r_x(a_2,k_2)]-sin\left(\frac{2\pi\theta}{N_\theta}\right)[r_y(a_1,k_1-1)-r_y(a_2,k_2)] \geq D_H - Mb_a(a_1,a_2,k_1,k_2,\theta), \quad (9)$$

$$cos\left(\frac{2\pi\theta}{N_\theta}\right)[r_x(a_1,k_1-1)-r_x(a_2,k_2-1)]-sin\left(\frac{2\pi\theta}{N_\theta}\right)[r_y(a_1,k_1-1)-r_y(a_2,k_2-1)] \geq D_H - Mb_a(a_1,a_2,k_1,k_2,\theta), \quad (10)$$

B. *Speed Advisory Constraints*

Implementation of a speed advisory in implies that the every point on the new trajectory should be somewhere on the RBT, but not necessarily at the corresponding time step. It is relatively trivial to force a waypoint to lie on a given line segment but as we do not know in advance if an aircraft needs to accelerate or not; or by how much, then we must also select *which* segment of the RBT the new point should lie on. This is implemented as a piecewise affine (PWA) function. The line segment that the new trajectory of aircraft a occupies at time-step k is identified with the binary variables $b_s(a,k,j)$ and selected with the following constraint:

$$\sum_{j=1}^{N_t-1} b_s(a,k,j) = 1 \quad , \quad (31)$$

$\forall\ a \in \{1, \ldots, N_a\}$, $k \in \{1, \ldots, N_t\}$ meaning that the optimizer must choose one line segment. The following constraints require $\mathbf{r}(a,k)$ to be within a margin Δr of the chosen segment:

$$\sum_{j=1}^{N_t} c_s(a,k,j) = 1 \quad , \quad (32)$$

$$\begin{aligned} c_s(a,k,1) &\leq b_s(a,k,1) \\ c_s(a,k,j) &\leq b_s(a,k,j-1)+b_s(a,k,j),\ j=2,\ldots,N_t-1 \\ c_s(a,k,N_t) &\leq b_s(a,k,N_t-1) \end{aligned} \quad (33)$$

$$\begin{aligned} \mathbf{r}(a,k) &\leq \sum_{j=1}^{N_t} \mathbf{r}^R(a,j)c_s(a,k,j)+\mathbf{1}\Delta r \\ \mathbf{r}(a,k) &\geq \sum_{j=1}^{N_t} \mathbf{r}^R(a,j)c_s(a,k,j)-\mathbf{1}\Delta r \end{aligned} \quad , \quad (34)$$

$\forall\ a \in \{1, \ldots, N_a\}$, $k \in \{1, \ldots, N_t\}$, where $c_s(a,k,j)$ is the relative distance of aircraft a at time k from point j of its RBT. The margin Δr is a small tolerance applied to ensure that the new trajectory remains dynamically feasible. For example, exactly fixing the spatial (3-D) trajectory fixes the turn radii, but these may not be compatible with an altered speed profile.

To illustrate the function of these constraints, consider the effect of choosing that the new position for a_1 at $k = k_1$ will lay on the segment between the 2nd and 3rd waypoints on the RBT, i.e. $b_s(a_1, k_1, 2) = 1$ and $b_k(a_1, k_1, j) = 0$ for all other j. Then according to (33), only the weights $c_s(a_1, k_1, 2)$ and $c_s(a_1, k_1, 3)$ can be non zero. Furthermore, according to (32), they must sum to one. Thus the summation terms in (34) must be a point somewhere between $\mathbf{r}^R(a_1,2)$ and $\mathbf{r}^R(a_1,3)$ and the point on the new trajectory can be more than Δr away.

Fig. 3 shows the effect of the above method on a 2D example of two aircraft. Fig. 3a shows the original RBTs and Fig. 3b shows the standard spatial avoidance described in Section IVA. Fig. 3c shows the affect of applying the additional speed advisory constraints; examination of the figure shows that both aircraft remain on their original flight paths but that F001 accelerates and F002 decelerates in order to ensure safe separation. It should also be noted that as the cost penalizes any deviation from the RBT, then after the conflict resolution, the aircrafts' velocities will be modified again to return the aircraft as close as possible to the original 4-D RBT.

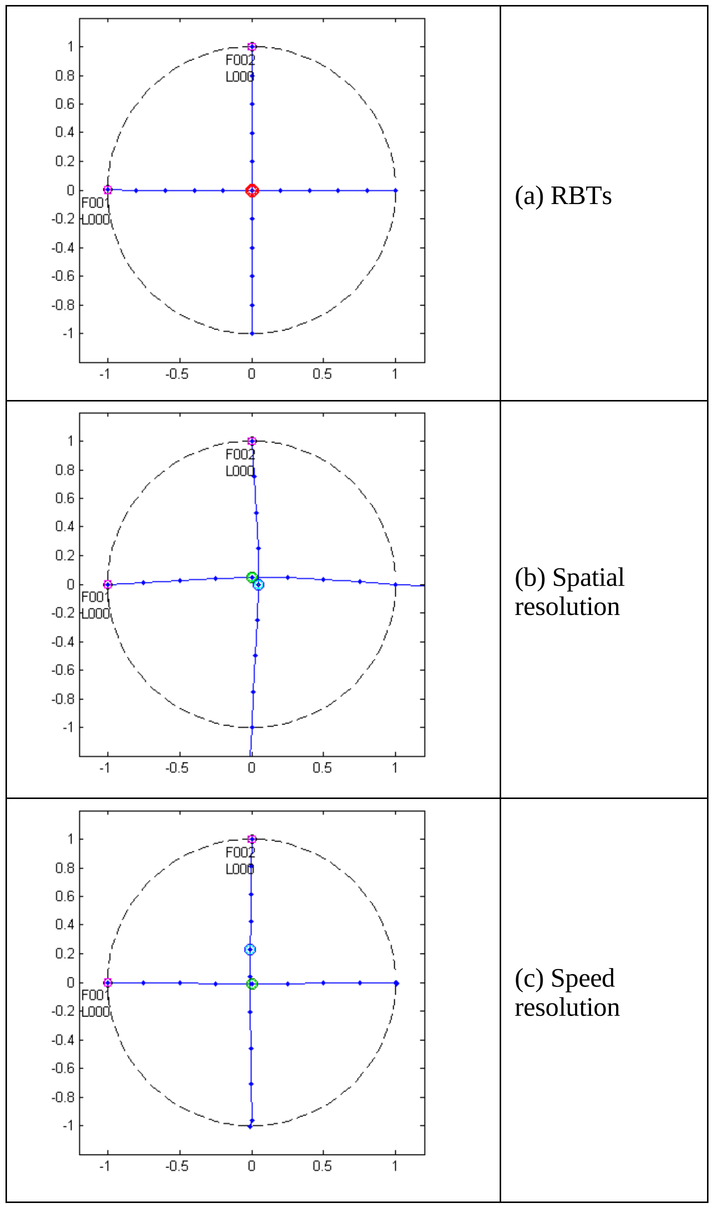


Figure 3: Trajectories from MILP Optimization in 2-D

V. COLLOCATION METHOD

A. *Review of method*

Collocation combined with a polar set representation of obstacles or other aircraft can be used for multi-vehicle conflict avoidance as described in [15], [19].

Consider a set of N_a aircraft with dynamics and constraints:

$$\dot{\mathbf{s}}(a,t) = f(\mathbf{s}(a,t)), \mathbf{u}(a,t)), a) \ \forall \ a, t, \tag{36}$$

$$g(\mathbf{s}(a,t), \mathbf{u}(a,t)), a) \leq 0 \ \forall \ a, t, \tag{37}$$

$$\mathbf{s}(a,0) = \mathbf{s}^R(a,0) \ \forall \ a, \tag{38}$$

$$\mathbf{s}(a, t_f(a)) = \mathbf{s}^R(a, t_f(a)) \ \forall \ a, \tag{39}$$

where ***s*** is the complete dynamic state of the aircraft, including position terms ***r*** and the velocity *etc*. The above constraints are solved using a typical collocation method such as [8] and as shown in more detail in [15] and [16]. This method has explicit decision variables for both spatial coordinates and the times $t_f(.)$, exploited later for the speed resolution constraints.

Having modelled the dynamics, the avoidance criteria are included. [15] and [16] extend the approach of Patel and Goulart [7] for obstacle avoidance to multi-vehicle conflict avoidance such that the avoidance criteria can be written as:

$$\mathbf{y}_{k,j}(a,b)^T \begin{pmatrix} \mathbf{r}_k(a) - \mathbf{r}_j(b) \\ t_k(a) - t_j(b) \end{pmatrix} \geq 1 \ \ \forall k, j, a > b, \tag{40}$$

$$\mathbf{y}_{kj}(a,b)^T \begin{pmatrix} \mathbf{r}_{k-1}(a) - \mathbf{r}_j(b) \\ t_{k-1}(a) - t_j(b) \end{pmatrix} \geq 1, \tag{42}$$

$$\mathbf{y}_{kj}(a,b)^T \begin{pmatrix} \mathbf{r}_k(a) - \mathbf{r}_{j-1}(b) \\ t_k(a) - t_{j-1}(b) \end{pmatrix} \geq 1, \tag{43}$$

$$\mathbf{y}_{kj}(a,b)^T \begin{pmatrix} \mathbf{r}_{k-1}(a) - \mathbf{r}_{j-1}(b) \\ t_{k-1}(a) - t_{j-1}(b) \end{pmatrix} \geq 1. \tag{44}$$

$$[\mathbf{I}_3 \mathbf{0}_{3\times1}]\, \mathbf{y}_{kj}(a,b)^T \in \mathcal{A}^0. \tag{45}$$

where $\boldsymbol{y}_{k,j}(a,b)$ is the normal vector of a 4-D hyperplane separating the trajectories of *a* and *b* and is a decision variable; A^0 is the "polar set" the avoidance region, e.g. the separation cylinder, or a set of acceptable separation normals; in this optimization, r_k is the kth position along the path and t_k is the time at which it will be reached. The reader is directed to [7] and [19] for full coverage of polar sets.

The optimization can be solved using a third party solver such as IPOPT [18]. IPOPT uses an interior point algorithm to solve the nonlinear optimization and as such provides no guarantee of global optimality and the rate of convergence and quality of the solution are highly dependent on the starting point, or initial solution, in the search space.

B. *Speed Advisory Constraints*

Due to the explicit inclusion of time as a decision variable and constraints (40) - (44), implementation of speed advisories in collocation can be achieved simply by fixing the location of the collocation points in all three spatial dimensions to force resolution by speed (achieved by adjusting the times associated with each point):

$$\mathbf{r}_k(a) = \mathbf{r}_k^R(a) + \delta r \ \ \forall \ \ k, a, \tag{46}$$

$\forall\ a \in \{1, \ldots, N_a\}$, $i \in \{1, \ldots, N_c\}$, where the δr term is a small tolerance included to ensure the trajectory remains dynamically feasible and N_c is the number of collocation points.

Fig. 5 shows a 2D example of velocity resolution using collocation. Fig. 5a shows the original RBTs, i.e. the inputs to the optimization and Fig. 5b shows the relative position of the two aircraft and it is clear that they pass closer than the avoidance distance indicated by the red circle. Fig. 5c and Fig. 5d show the conflict resolved spatially using the collocation model of Section 0A where it is clear that both aircraft have deviated from their RBTs in order to avoid one another. Fig. 5e and Fig. 5f show the same conflict resolution but including the additional speed advisory constraint (46) which causes both aircraft to remain on their RBTs but to adjust their velocities to avoid a conflict which is as expected.

Fig. 6 shows a similar example for 3 vehicles with the relative location of each aircraft plotted over a fine scale relative to each other aircraft.

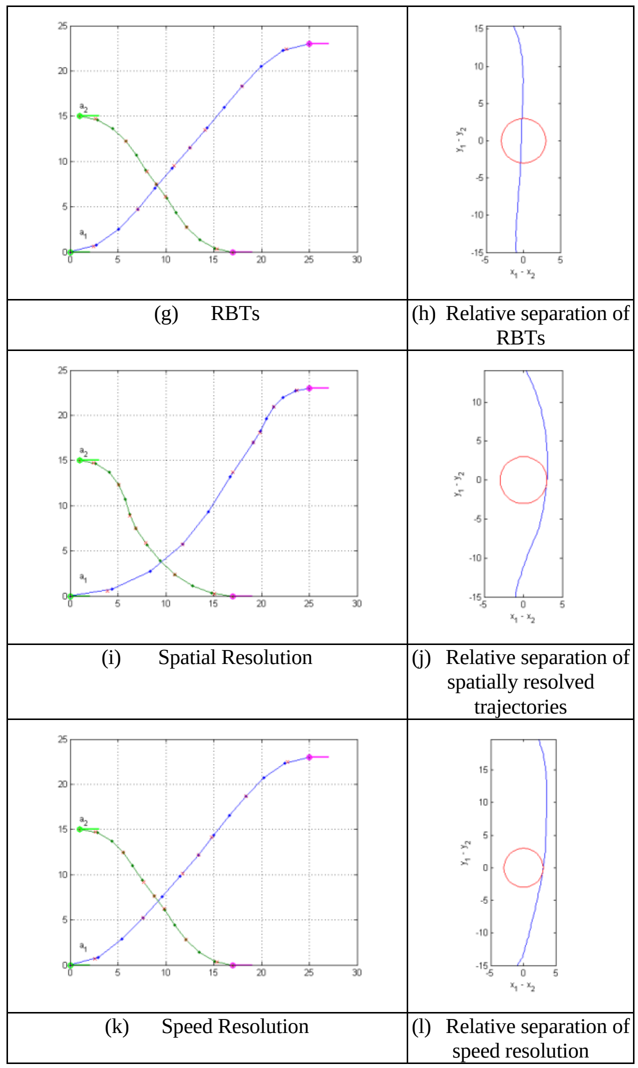
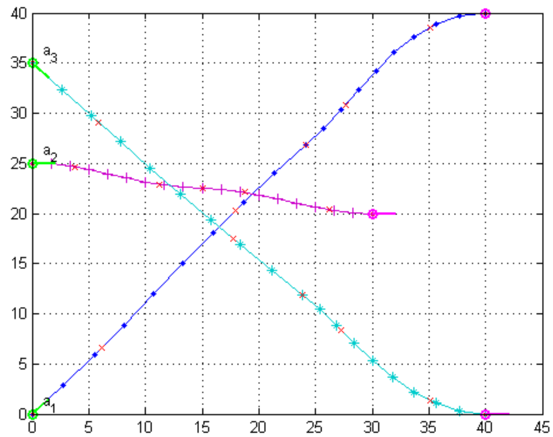
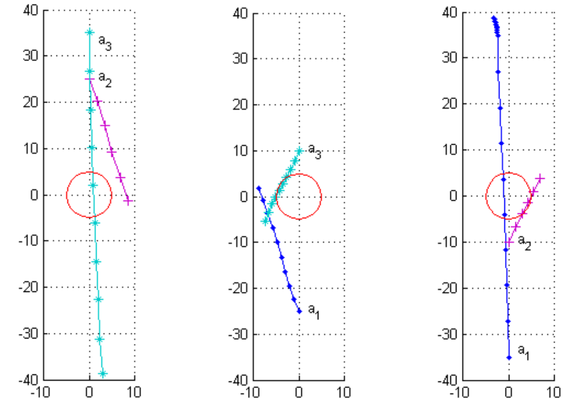


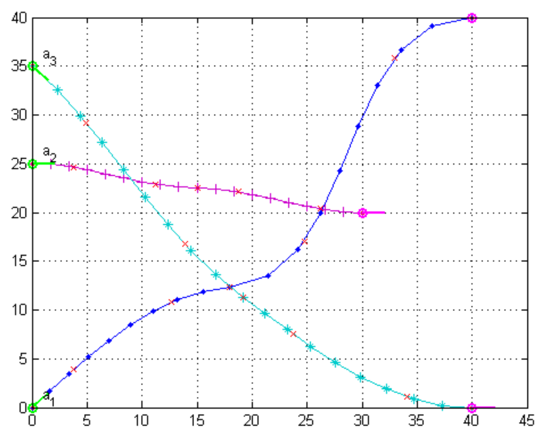
Figure 5: Velocity resolution results (2D, 2 vehicles)



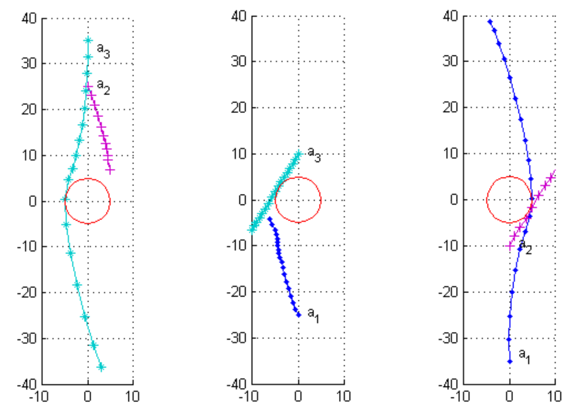
(a) RBTs



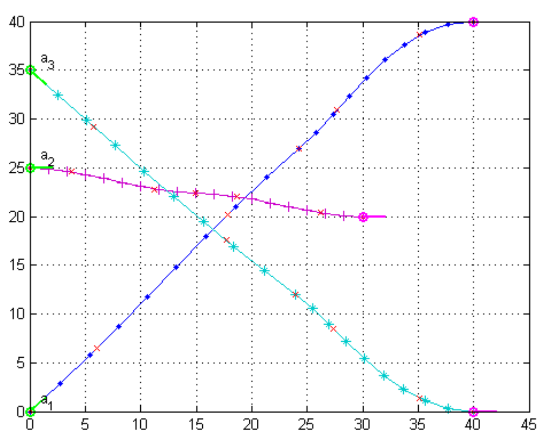
(b) Relative separation of RBTs to a_1, a_2 and a_3 respectively



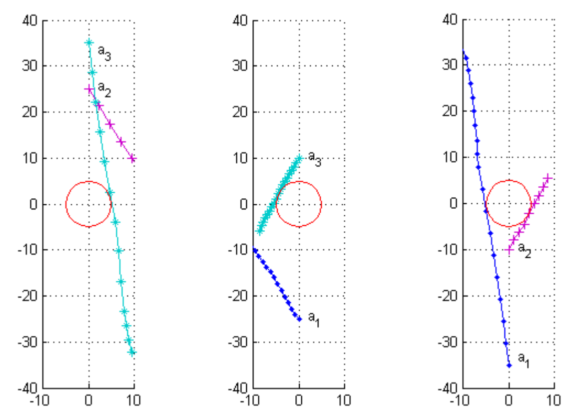
(c) Spatial Resolution



(d) Relative separation of spatially resolved trajectories to a_1, a_2 and a_3 respectively



(e) Velocity resolution



(f) Relative separation of velocity resolution trajectories to a_1, a_2 and a_3 respectively

Figure 6: Velocity resolution results (2D, 3 vehicles)

VI. Example Application to a Multi-Sector Area

To demonstrate the method on a more realistic dataset, historic trajectory data was downloaded from www.flightradar24.com to represent a set of Reference Business Trajectories from three sectors over the U.K. and the Irish Sea, selected to include crossings between transatlantic flights and short-hauls bound to or from northern Britain. In order to introduce conflicts to the data, two sets of flight-paths were downloaded for two 20 minutes periods and super-imposed onto a single arbitrary time period, the complete set of 71 trajectories are shown in Fig. 7, where the upper plot is a plan view and the lower plot represents a vertical projection.

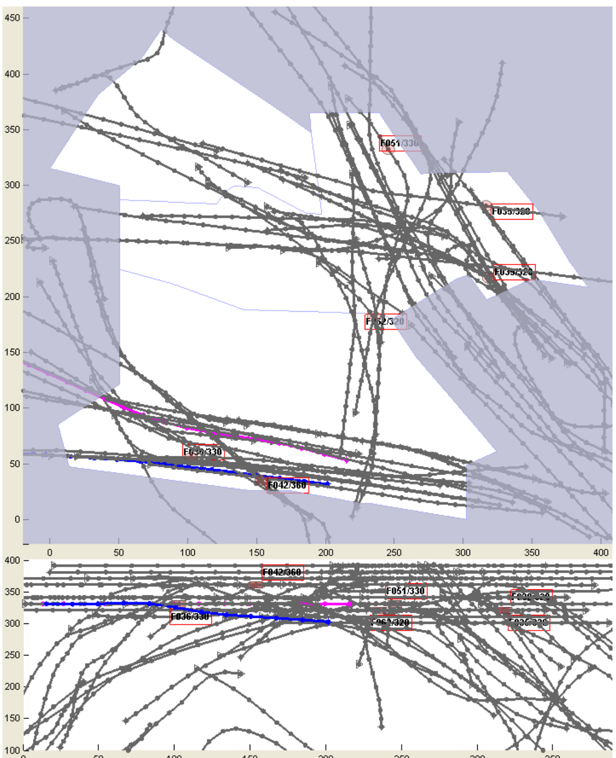


Figure 7: Complete set of flights

Fig. 8 shows a sub-set of the RBTs filtered to show only the trajectories for aircraft with identified conflicts; the time-steps with conflicts are indicated with red cylinders.

A MILP implementation of the speed advisories was implemented on a larger scale and used to re-plan the conflicting trajectories as shown in Fig. 9. Non-conflicting aircraft are fixed. The optimizer is constrained to avoid all aircraft, not just those in conflicts. The optimization must therefore re-plan for the 6 aircraft in conflict but those are also required to avoid conflicts with the other 65 aircraft in the problem.

The trajectories of aircraft with detected conflicts were re-planned using MILP, including the speed advisory constraint for all new trajectories.

Fig. 9 shows the resolved trajectories overlaid on the RBTs and it can be seen that the time-steps where there was previously a conflict (marked with a green cylinder) now have sufficient separation to ensure that the corresponding line-segments do not intersect, i.e. are conflict free. This is confirmed by plotting the separation between aircraft at each time step as in Fig. 10 where the minimum horizontal and vertical separation are plotted for each aircraft between each pair of adjacent time-steps both before (+) and after (x); the shaded region on the graph represents the conflict region between aircraft. From Fig. 10 it is clear that the optimization has increased the horizontal separation between aircraft sufficiently to ensure no conflicts remain. Fig. 11 shows the airspeed of flight F036 before and after optimization. The changes are obvious and to be expected, since this is the only variable available to resolve the conflict. Observe how the aircraft has accelerated to pass ahead of F030 and then slowed to revert to its original RBT.

This example highlighted one issue with speed resolution: it is more effective over long distances. The subliminal control concept of [21] worked over very long distances with only small speed changes. Here, with only a short distance to work with and many other potential conflicts, greater speed changes were needed and feasibility was lost in some other cases. A suggestion then is to "soften" the speed resolution constraints (34) such that the margin Δr can grow, but at a penalty. With appropriate choice of weights, this would find a speed resolution when possible but revert to spatial only when forced.

VII. Computational Complexity

As an indication of the complexity of the proposed method, Table II compares the solution times for the scenarios presented in this paper both with and without the additional constraints to implement a speed advisory.

TABLE II. EXAMPLE SOLUTION TIMES

Scenario	**Solution Time (s)**	
	Spatial Resolution	***Speed Advisory***
2D MILP	0.25	0.95
2D Collocation (2 vehicles)	1.78	2.72
2D Collocation (3 vehicles)	16.80	31.93
MSA Example (MILP)	124.77	900*

Given that the speed resolution adds both additional constraints and binary variables to the MILP, it is not surprising that the MILP examples are slower to compute with them added. Note that in the MSA example, the solver reached the 15-minute time limit and returned the best solution it had found. Accelerating the solution times is the subject to on-going work.

The collocation method also became slower with the addition of speed resolution constraints. This is perhaps surprising as decision variables are removed from the problem in this case, but the more constrained nature of the problem does make it more difficult to solve. This is also the subject of on-going work, with particular attention to the generation of the "initial guess" for the solver.

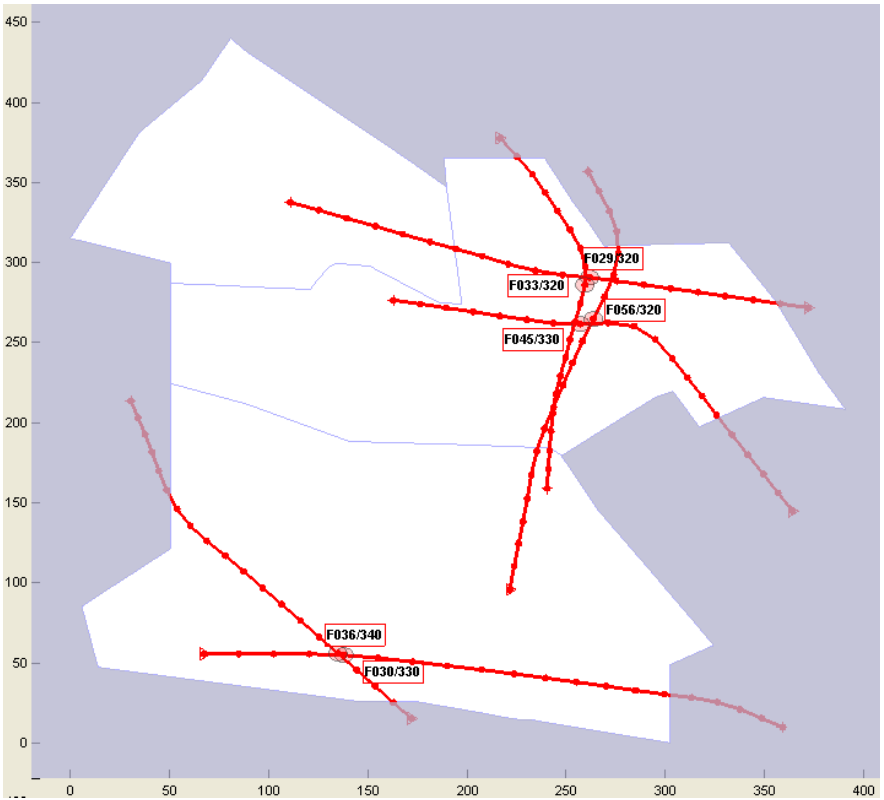


Figure 8: Set of flights with conflicts (require re-planning)

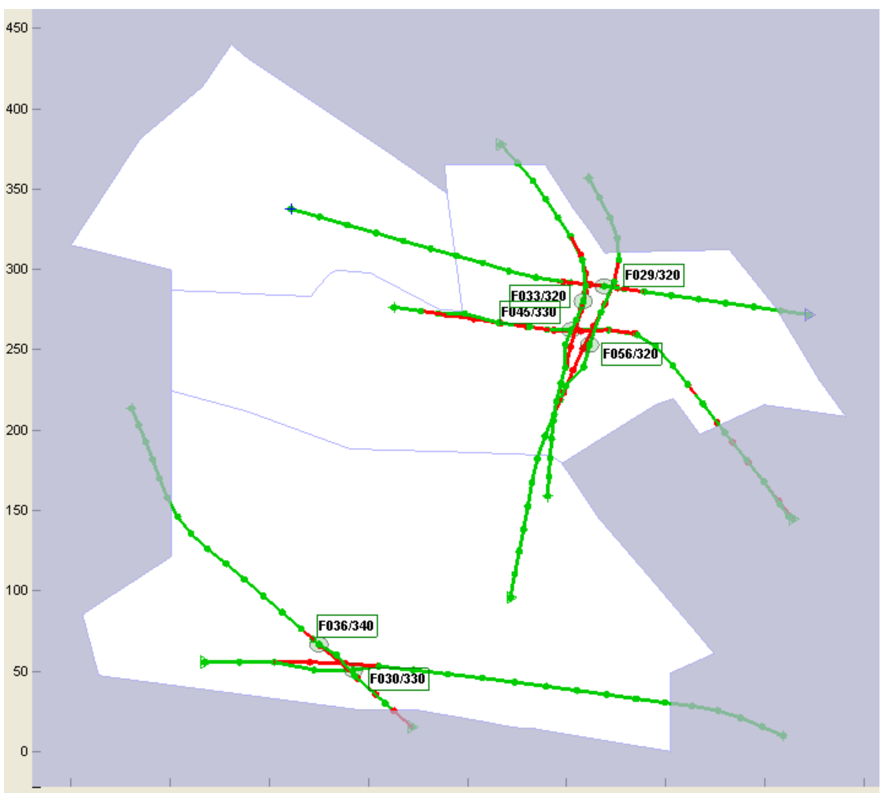


Figure 9: Resolved trajectories

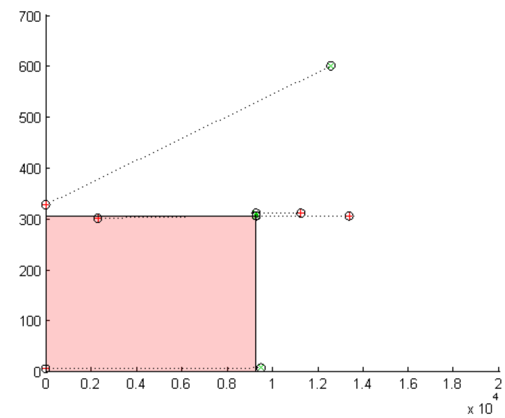


Figure 10: Separation of aircraft before and after resolution

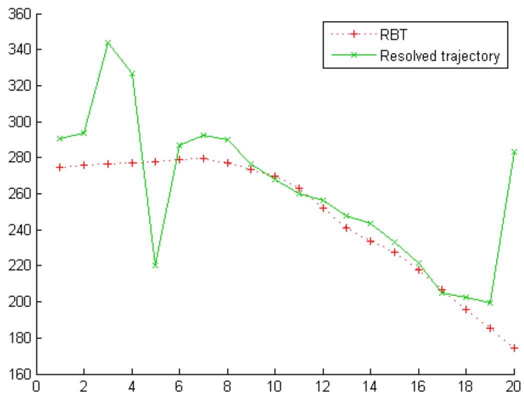


Figure 11: Distribution of airspeed for planning aircraft before and after conflict resolution

VIII. Conclusions

This paper builds on the previous work and presents a novel method for implementing a *speed advisory* in two trajectory optimizers: Mixed Integer Linear Programming (MILP) and a non-linear collocation method. Each method offers different advantages and limitations: MILP provides a globally optimal solution but more limited forms of objective functions and dynamics; collocation can account for a more accurate dynamics and complex cost functions but provides no guarantee of global optimality. Consequently, both methods may be of interest in different applications. In order to obtain the benefits of both methods, if timescales allowed, the MILP could even be used to initialize the collocation method.

The paper has presented an intuitive method for implementing speed advisories in both a linear and non-linear optimizer. Combined with previous work [15] this provides a comprehensive set of tools for making high-level decisions regarding aircraft conflict resolution to aid human controllers.

References

[1] Y. Kuwata and J. How, “Three dimensional receding horizon control for UAVs,” in AIAA Guidance, Navigation, and Control Conference, 2004.

[2] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L. T. Biegler, “Milp and nlp techniques for centralized trajectory planning of multiple unmanned air vehicles,” in Proceedings of the 2006 American Control Conference, IEEE, 2006.

[3] A. Richards and J. How, “Aircraft trajectory planning with collision avoidance using mixed integer linear programming,” in American Control Conference, 2002. Proceedings of the, pp. 1936–1941, 2002.

[4] Eurocontrol, “SESAR concept of operations,”

[5] A. Richards, “Constraining the sense of conflict resolution: Supervision of route optimization,” in SESAR Innovation Days, 2011.

[6] IBM, “IBM ILOG CPLEX website,” March 2012.

[7] R. B. Patel and P. J. Goulart, “Trajectory generation for aircraft avoidance maneuvers using online optimization,” Journal of Guidance, Control, and Dynamics, vol. 34, pp. 218–230, JanuaryFebruary 2011.

[8] A. Eele and A. Richards, “Path-planning with avoidance using nonlinear branch-and-bound optimization,” Journal of Guidance, Control, and Dynamics, vol. 32, pp. 384–394, Mar. 2009.

[9] F. Fahroo and M. I. Ross, “Direct trajectory optimization by a chebyshev pseudospectral method,” in Proceedings of the American Control Conference, 2000.

[10] Eurocontrol, “Base of aircraft data,” 2010.

[11] J. Latombe, Robot Motion Planning. Kluwer international series in engineering and computer science: Robotics, Springer, 1990.

[12] R. F. de Oliveira and C. B¨uskens, “Benefits of optimal flight planning on noise and emissions abatement at the frankfurt airport,” in AIAA Guidance, Navigation, and Control Conference, 2012.

[13] B. Fornberg, A practical guide to pseudospectral methods,” vol. 1 of Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1996.

[14] Q. Gong, F. Fahroo, and M. Ross, “Spectral algorithm and for pseudospectral and methods and in optimal and control,” in Journal of Guidance, Control, and Dynamics, vol. 31, May–June 2008.

[15] O. Turnbull and A. Richards, “Examples of Supervisory Interaction with Route Optimizers”, Proceedings of the SESAR Innovation Days (2012)

[16] A. Richards and O. Turnbull, “Efficient Use of Time Steps in Collision Avoidance Optimization”, unpublished

[17] J.-P. Berrut and L. Trefethen, “Barycentric lagrange and interpolation,” SIAM Review, vol. 46, no. 3, pp. 501–517, 2004.

[18] A. W¨achter and L. Biegler, “Global and local convergence of line search filter methods for nonlinear programming,” Tech. Rep. B-01-09, Carnegie Mellon University, 2002.

[19] O. Turnbull and A. Richards, “Collocation methods for multi-vehicle trajectory optimization”, unpublished

[20] Vela, A.; Solak, S.; Singhose, W.; Clarke, J.-P.; "A mixed integer program for flight-level assignment and speed control for conflict resolution," In Proceedings of the 48th IEEE Conference on Decision and Control, 2009.

[21] Cruck, E.; Lygeros, J.; , "Subliminal air traffic control: Human friendly control of a multi-agent system," American Control Conference, 2007. ACC '07 , vol., no., pp.462-467, 2007

[22] ERASMUS Project, Concept of Operation, 2008, available at http://www.atm-erasmus.com

Author BiographIES

Oliver D.N. Turnbull, PhD. Engineering Mathematics, University of Bristol, 2008; MEng Avionic Systems Engineering, University of Bristol, 2004. After completing his PhD he worked as a space systems engineer before returning to the University of Bristol as a Research Assistant. His research interests include optimization for vehicle guidance, path planning and decision making in the presence of uncertainty.

Arthur G. Richards received the M.Eng. degree from Cambridge University in 2000 and the SM and PhD degrees from MIT in 2002 and 2004, respectively. Since 2004, he has been with the Department of Aerospace Engineering, University of Bristol, Bristol, U.K., where he is currently a Senior Lecturer. His research interests include trajectory optimization, model predictive control, and their combination to develop high-performance guidance for aircraft and spacecraft.