Proceedings of the American Control Conference
Anchorage, AK May 8-10, 2002

Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming [1]

Arthur Richards [2] and Jonathan P. How [3]

Abstract

This paper describes a method for finding optimal trajectories for multiple aircraft avoiding collisions. Recent developments in spacecraft path-planning have shown that trajectory optimization including collision avoidance can be written as a linear program subject to mixed integer constraints, known as a *mixed-integer linear program* (MILP). This can be solved using commercial software written for the operations research community. In this paper, an approximate model of aircraft dynamics using only linear constraints is developed, enabling the MILP approach to be applied to aircraft collision avoidance. The formulation can also be extended to include multiple waypoint path-planning, in which each vehicle is required to visit a set of points in an order chosen within the optimization.

1 Introduction

Path-planning for aircraft has been the subject of much recent interest. Two major applications drive this research: air traffic management [2] and autonomous Unmanned Aerial Vehicles (UAVs) [3]. Both areas require path-planning methods for multiple vehicles, avoiding obstacles and each other. Finding optimal solutions to these problems is intrinsically hard, since the optimization is non-convex. However, it can be written involving discrete decisions between linear constraints, which in turn can be expressed as linear constraints upon a mixture of continuous and integer variables [8, 9]. The resulting mixed-integer linear program (MILP) can be solved using efficient, commercial software employing the branch-and-bound algorithm [7]. This approach has been applied to spacecraft maneuvering, for which linear approximations of relative dynamics are well-known [5, 10]. In this paper, a linear approximation of aircraft dynamics will be developed, allowing the MILP method of collision avoidance to be used in the trajectory optimization.

A centralized algorithm is presented to perform the co-ordinated guidance of multiple aircraft flying at a certain altitude. The aircraft are modeled as vehicles moving in two dimensions with constant speed and limited turning rate. Previous work in this area has developed necessary conditions for optimality using nonlinear aircraft dynamics [1]. Another method performed a simplified optimization based on a single heading decision [4]. The method described in this paper provides direct generation of the optimal trajectory. A minimum time formulation, also seen in [11], is combined with the collision avoidance constraints from [5], and extended to include aircraft dynamics.

The ability of MILP to include discrete decisions in the optimization allows some very flexible mission problems to be solved. In particular, this paper includes the extension of the formulation to include multiple waypoint path-planning. Instead of having a single terminal point, each aircraft is required to visit a number of points. The order of those visits is selected within the optimization to give the shortest overall flight time.

This paper begins with the formulation of the trajectory optimization, including the dynamics model, collision avoidance constraints and multiple waypoint extension. An example is then presented to demonstrate that the approximation of the dynamics performs acceptably. Further examples demonstrate co-operative planning for multiple aircraft and the flexibility of the multiple waypoint method.

2 Problem Formulation

2.1 Approximation of Vehicle Dynamics

This paper considers problems in which aircraft fly at constant altitude, resulting in planar motion. This is a common restriction in air traffic models, as air space is commonly structured in layers [1]. Also, in UAV problems, altitude is often determined by mission constraints, such as sensor resolution or radar visibility, resulting in a 2-D guidance problem only.

Furthermore, for many cases of interest, an aircraft can be modeled as moving at constant speed. The rate of change of heading angle is limited by the maximum bank angle of the aircraft. Writing these constraints exactly results in nonlinear expressions, which cannot be handled in a MILP framework. Therefore, the aircraft is approximated here as a point mass, moving with limited speed and acted on by a force of limited magnitude. The optimization seeks the minimum time

[1]Funded under Air Force grant #F49620-01-0453

[2]Space Systems Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139 arthurr@mit.edu

[3]Space Systems Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139 jhow@mit.edu

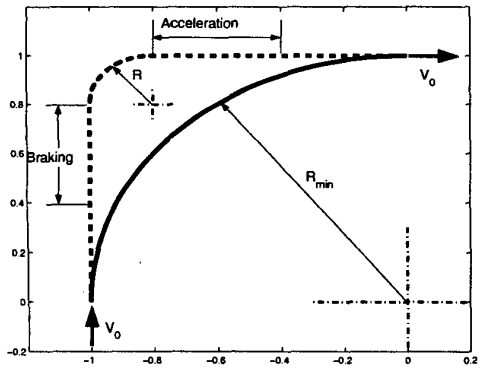

Fig. 1: Cornering paths allowed by the constraints. $R_{\min}$ is the designed minimum radius of curvature. The dashed path is allowable within the constraints, but will always take longer than the solid path.

solution, so it is favorable to remain at, or near, the maximum speed throughout the entire maneuver.

The turning rate constraint is effected by a force magnitude limit. Consider a mass m traveling with speed v subject to a force of magnitude f. The instantaneous turning rate ω will be greatest if the force is perpendicular to the velocity. It is therefore limited by

$$\omega \leq \frac{f}{mv} \tag{1}$$

Furthermore, if the magnitude of the force is limited to $f_{\max}$ and the speed is a constant $v_{\max}$, the rate is limited throughout the problem by

$$\omega_{\max} = \frac{f_{\max}}{mv_{\max}} \tag{2}$$

In this implementation of the problem, there is only an upper bound on the speed. The inclusion of this magnitude limit as linear constraints is shown in Section 2.2. It is feasible for the speed to fall below $v_{\max}$, allowing tighter turns than the bound in Eqn. 2. However, for the minimum time solution, it is favorable to remain at maximum speed and obey the specified turn rate limit.

For example, consider the situation shown in Fig. 1, in which an aircraft must turn through 90^o from one trajectory on to another. Two of the possible paths are shown in this figure. Following the solid line, the aircraft remains at maximum speed and turns at the prescribed maximum turning rate $\omega_{\max}$, therefore following the prescribed minimum radius $R_{\min}$. It is also feasible to follow the dashed path, decelerating first, then applying the maximum force to achieve a smaller radius of curvature, before accelerating back to maximum speed and rejoining the solid path. Although it is allowed in the linear model, this trajectory is not possible for a real aircraft. Fig. 2 shows the variation in total maneuver time with the radius of turn used. This was found analytically by calculating the time for

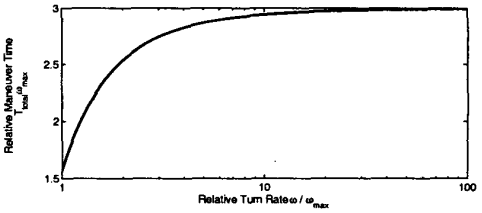

Fig. 2: Variation of execution time with turn rate for the maneuver shown in Fig. 1.

the turn itself, the necessary deceleration and acceleration times, and the adjoining segments of maximum speed travel. It is clear from the figure that the fastest turning maneuver is achieved by remaining at maximum speed and obeying the nominal maximum turning rate $\omega_{\max}$. Using higher turning rates leads to a slower overall maneuver, due to the additional deceleration required. This result matches intuition, because the solid path in Fig. 1 is shorter in length than any others and has a higher average speed.

When avoidance constraints are added, some arrangements of obstacles could cause the model to favor the tighter turn. Therefore, it is necessary to post-analyze each trajectory to ensure it is flyable by the real aircraft. If not, the problem can be rerun with a lower force limit until an acceptable solution is found.

2.2 Dynamics Constraints in the Problem

To analyze the types of problems discussed in the Introduction, assume that there are N aircraft, each approximated as a point mass moving in 2-D. The position of aircraft p at time step i is given by (x_{ip}, y_{ip}) and its velocity by $(v_{x_{ip}}, v_{y_{ip}})$, forming the elements of the state vector $\mathbf{s}_{ip}$. Each aircraft is assumed to be acted upon by control forces $(f_{x_{ip}}, f_{y_{ip}})$ in the X- and Y-directions respectively, forming the force vector $\mathbf{f}_{ip}$. The discretized dynamics of the overall system, applied to all N vehicles up to T time steps, can be written in the linear form

$$\begin{gathered} \forall p \in [1 \ldots N] \forall i \in [0 \ldots T-1] \\ \mathbf{s}_{(i+1)p} = \mathbf{A}_p \mathbf{s}_{ip} + \mathbf{B}_p \mathbf{f}_{ip} \end{gathered} \tag{3}$$

In all cases, the initial conditions are specified as

$$\mathbf{s}_{0p} = \mathbf{s}_{I_p} \tag{4}$$

where $\mathbf{s}_{I_p}$ is the initial state of vehicle p.

As discussed previously, both the velocity of the aircraft and the force acting upon the aircraft are subject to magnitude constraints. The exact representation of these constraints would be nonlinear, but they can be approximated by linear inequalities. The true magnitude constraints enclose a circle on the X-Y plane, as shown in Fig. 3. A simple magnitude limit on each element, as shown in Eqn. 5 for the speed constraint,

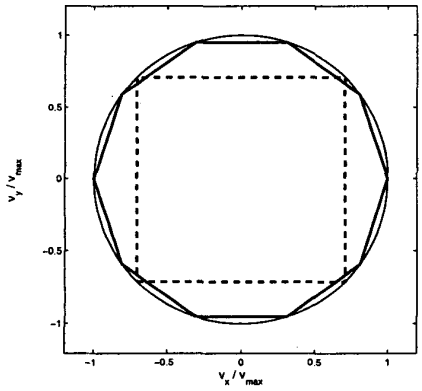

Fig. 3: Approximations to magnitude limits for 2-D vectors. The circle is the feasible region for true magnitude limits. The square and polygon are two ways of approximating these regions with linear constraints.

restricts the velocity within the square shown in Fig. 3.

$$|v_x| \leq v_{\text{lim}} \text{ and } |v_y| \leq v_{\text{lim}} \tag{5}$$

This is obviously a poor approximation to the circle (magnitude limit). However, an arbitrary number (M) of constraints can be used to achieve better approximations. For both velocity and force constraints these are given by

$$\forall i \in [0 \ldots T-1] \forall p \in [1 \ldots N] \forall m \in [1 \ldots M]$$

$$f_{x_{ip}} \sin\left(\frac{2\pi m}{M}\right) + f_{y_{ip}} \cos\left(\frac{2\pi m}{M}\right) \leq f_{\max} \tag{6}$$

$$\forall i \in [1 \ldots T] \forall p \in [1 \ldots N] \forall m \in [1 \ldots M]$$

$$v_{x_{ip}} \sin\left(\frac{2\pi m}{M}\right) + v_{y_{ip}} \cos\left(\frac{2\pi m}{M}\right) \leq v_{\max} \tag{7}$$

The feasible region formed by ten constraints ($M = 10$) is shown in Fig. 3, forming a better approximation to the circle. The additional constraints have little effect on the computation time since they include only linear variables.

2.3 Collision Avoidance

To safely avoid collision, each vehicle has a rectangular exclusion region around it which no other vehicle can enter. For example, consider two vehicles p and q, using the same position notation as in the previous section. Let the safety distance be denoted by d, assumed to be the same in both X and Y directions for simplicity. The constraints for collision avoidance were developed in [5, 10] and can be written as

$$\begin{array}{lrcl} \forall i \in [1 \ldots T] & \forall p, q \; | q > p & & \\ & x_{ip} - x_{iq} & \geq & d - Rc_{ipq1} \\ \text{and} & x_{iq} - x_{ip} & \geq & d - Rc_{ipq2} \\ \text{and} & y_{ip} - y_{iq} & \geq & d - Rc_{ipq3} \\ \text{and} & y_{iq} - y_{ip} & \geq & d - Rc_{ipq4} \\ \text{and} & \sum_{k=1}^{4} c_{ipqk} & \leq & 3 \end{array} \tag{8}$$

where c_{ipqk} are a set of binary variables (0 or 1) and R is a positive number that is much larger than any position or velocity to be encountered in the problem. The first four inequalities in Eqn. 8 represent the four lines enclosing the exclusion region around each vehicle. If $c_{ipqk} = 0$, there is at least d distance between vehicles p and q in the k^{th} direction (of the four directions $+X$, $-X$, $+Y$, $-Y$) at the i^{th} time step. If $c_{ipqk} = 1$, the constraint is relaxed. The final inequality ensures that no more than three of the constraints are relaxed at any time step, so there must always be safe separation in at least one direction.

Eqn. 8 becomes an additional constraint on the trajectory optimization problem. The binaries c_{ipqk} become decision variables for the optimization.

2.4 Solving for the Minimum Time Trajectory

This section shows how MILP constraints can be used to include variable finishing times within a linear optimization. A concurrent development of a similar formulation can be seen in [11], applied to off-line design of minimum-time regulators. In this problem, vehicle p is required to reach its destination (x_{F_p}, y_{F_p}) at some time-step before the maximum T. This problem is solved by introducing the binary variables b_{ip}, which have a value of 1 if the p^{th} vehicle reaches its destination at the i^{th} timestep, and 0 otherwise. The resulting MILP constraints are

$$\begin{array}{lrcl} \forall p \in [1 \ldots N] \; \forall i \in [1 \ldots T] & & & \\ & x_{ip} - x_{F_p} & \leq & R(1 - b_{ip}) \\ \text{and} & x_{ip} - x_{F_p} & \geq & -R(1 - b_{ip}) \\ \text{and} & y_{ip} - y_{F_p} & \leq & R(1 - b_{ip}) \\ \text{and} & y_{ip} - y_{F_p} & \geq & -R(1 - b_{ip}) \end{array} \tag{9}$$

$$\forall p \in [1 \ldots N] \sum_{i=1}^{T} b_{ip} = 1 \tag{10}$$

where R is the same large, positive number used in Eqn. 8. It can be seen that if $b_{ip} = 1$, Eqn. 9 forces the aircraft state to equal the final state. However, if $b_{ip} = 0$, then the constraints are inactive. Eqn. 10 enforces the logic that each vehicle must reach its target at one time point, but does not require that the vehicles finish at the same time. The minimum time solution for each aircraft is sought by minimizing the sum of the finishing times for each vehicle:

$$\min_{\mathbf{s},\mathbf{u},\mathbf{b},\mathbf{c}} J = \sum_{i=1}^{T} \sum_{p=1}^{N} T_i b_{ip} \tag{11}$$

where T_i is the actual time elapsed at step i, $\mathbf{b}$ and $\mathbf{c}$ are the binary variables.

Unfortunately, the cost function in Eqn. 11 alone leads to an inefficient formulation. Since time has been discretized, there can be multiple solutions that finish at each time step. In addition, the states and inputs for

time steps after the selected finishing time have no effect on the cost. These redundancies do not affect the final result, but finding that solution can be slower as irrelevant regions of the solution space are searched. This problem can be remedied by adding a small input penalty to the cost function

$$\min_{\mathbf{s,u,b,c}} J = \sum_{p=1}^{N} \left(\sum_{i=1}^{T} T_i b_{\mathrm{ip}} + \epsilon \sum_{i=0}^{T-1} (|f_{x_{ip}}| + |f_{y_{ip}}|) \right) \tag{12}$$

where ϵ is a positive number, small enough to ensure that the fuel penalty never exceeds the value of one time step. The complete problem is then to minimize Eqn. 12 subject to the constraints in Eqs. 3, 4, 6, 7, 8, 9 and 10. In this case, there is a unique optimal solution, and the solution algorithm appears to perform more efficiently. Due to the complexity of the CPLEX solution procedure used, the exact cause of this improvement is unclear, but later results will show that it has a significant effect on the solution times.

2.5 Multiple Waypoints

The problem formulation so far plans trajectories from a fixed start point to a fixed end point. This section extends the minimum time formulation to include multiple waypoints. These results address the scenario in which each vehicle must visit a set of points during the maneuver, but the order of those visits is not prespecified. Instead, the order will be selected within the optimization to minimize the overall cost of the maneuver.

Assume that there are W waypoints for each vehicle and the position of the k^{th} waypoint for the p^{th} vehicle is $(x_{\mathrm{W_{kp}}}, y_{\mathrm{W_{kp}}})$. To solve this problem, introduce the new binary variables b_{ipk}, which have the value 1 if the p^{th} vehicle reaches its k^{th} waypoint at the i^{th} time step, and 0 otherwise. In this case, the MILP constraints from Eqs. 9 and 10 are modified to

$$\begin{aligned} &\forall p \in [1 \ldots N] \, \forall i \in [1 \ldots T] \, \forall k \in [1 \ldots W] \\ &\quad x_{ip} - x_{\mathrm{W_{kp}}} \leq R(1 - b_{ipk}) \\ &\text{and} \quad x_{ip} - x_{\mathrm{W_{kp}}} \geq -R(1 - b_{ipk}) \\ &\text{and} \quad y_{ip} - y_{\mathrm{W_{kp}}} \leq R(1 - b_{ipk}) \\ &\text{and} \quad y_{ip} - y_{\mathrm{W_{kp}}} \geq -R(1 - b_{ipk}) \end{aligned} \tag{13}$$

$$\forall p \in [1 \ldots N] \, \forall k \in [1 \ldots W] \, \sum_{i=1}^{T} b_{ipk} = 1 \tag{14}$$

These constraints require that each vehicle visits each of its assigned waypoints once, but does not imply any ordering of the visits. Since the ordering of the waypoints is not specified, any one of them might be the final point visited. The finishing time is therefore the maximum of the visiting times for each waypoint. Thus we define a new variable, $T_{\mathrm{F_p}}$ to represent the finishing time for the p^{th} vehicle. It must satisfy the following constraint

$$\forall p \in [1 \ldots N] \, \forall k \in [1 \ldots W] \, T_{\mathrm{F_p}} \geq \sum_{i=1}^{T} T_i b_{\mathrm{ipk}} \tag{15}$$

Finally, the cost function in Eqn. 12 is modified to become

$$\min_{\mathbf{s,u,b,c}} J = \sum_{p=1}^{N} \left(T_{\mathrm{F_p}} + \epsilon \sum_{i=0}^{T-1} (|f_{x_{\mathrm{ip}}}| + |f_{y_{\mathrm{ip}}}|) \right) \tag{16}$$

subject to the constraints in Eqs. 3, 4, 6, 7, 8, 13, 14 and 15. It can be seen that this formulation will provide the minimum time solution to visit all the waypoints in the optimal order.

3 Implementation

The optimization problems shown here can be easily translated into the AMPL modeling language [6]. An AMPL model file contains the constraint forms for all instances, while the data is written to an AMPL data file by a Matlab script. CPLEX optimization software is used to solve the problem [13]. A series of scripts in Matlab and AMPL allow the entire path-planning problem to be invoked by a single command. The problems were solved on a 1GHz PC with 256MB RAM.

4 Examples

4.1 Single Aircraft

The example in this section demonstrates that the linear constraints form an acceptable approximation to aircraft dynamics. A vehicle with mass $m = 5$kg moves through a field of fixed obstacles. The constraints for avoiding fixed obstacles were developed in [5] and are similar to the collision avoidance constraints earlier in this paper. The dynamics were discretized with a time step of two seconds. The initial position was $(5, 5)$ and the final position $(-5, 4)$, with a required velocity of $(-0.2, 0)$ at both points.

The real vehicle has a maximum speed $v_{\max} = 0.225$m/s and turn rate $\omega_{\max} = 15°/\mathrm{s}$. Using Eqn. 2, this corresponds to a force limit $f_{\max} = 0.294$N. A force penalty weighting $\epsilon = 0.001$ was used. The circles in Fig. 4 mark the trajectory design using these values. It avoids all the obstacles and reaches its target after 60 seconds or 30 time steps. The solid line in Fig. 5 shows the speed of the vehicle during the maneuver. As expected, it remains close to the maximum throughout. The turn rate, shown by the dash-dot line in Fig. 6, is either near its maximum or zero. The trajectory is approximately a series of straight lines joined by the tightest possible turns, all at constant speed. This satisfies the necessary conditions derived in [1], which found that the extremal path consists of segments of minimum radius turns joined by straight lines. Note that the trajectory cuts the corners of the obstacles, since collision avoidance is enforced only at

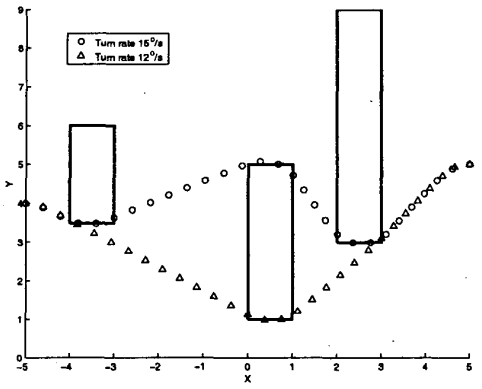

Fig. 4: The designed trajectories for an aircraft moving through obstacles. Paths for two different turn rate settings are shown.

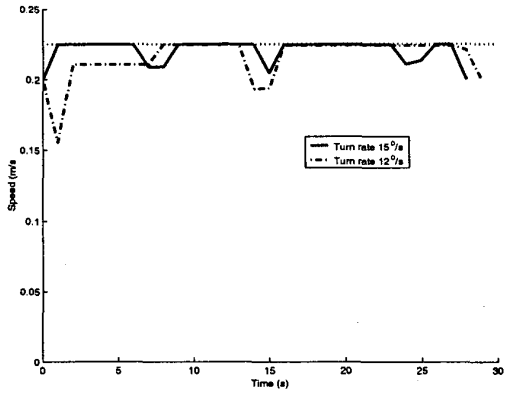

Fig. 5: Speed of vehicle during maneuvers shown in Fig. 4. The dashed line shows the specified limit.

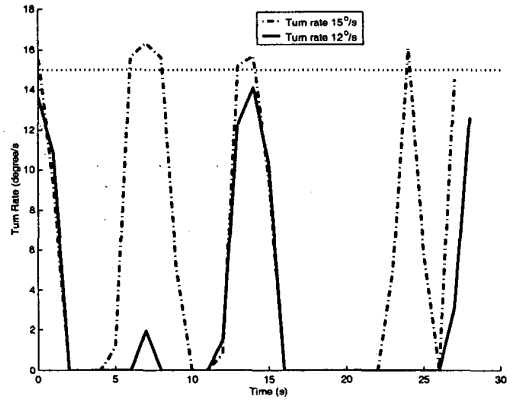

Fig. 6: Turning rate of vehicle during maneuvers shown in Fig. 4. The dashed line shows the real aircraft limit of $15°/s$.

the discrete time points marked. The obstacle regions in the optimization must be slightly larger that the real obstacles to allow for this margin.

Fig. 6 shows that the turn rate exceeds the 15°/s limit during some of the tighter turns around obstacle corners. This possibility was discussed in Section 2.1. In this case, the infringement of the turn rate limit is small, but if the resulting trajectory is unflyable by the real aircraft, post-processing would indicate the need to find a different solution. Reducing the force limit in the model by 20%, corresponding to a nominal turn rate limit of 12°/s, gives the result shown by the triangles in Fig. 4. Since the available force has been reduced, the tight turns in the first maneuver would require greater deceleration. This makes the new path more favorable. The solid line in Fig. 6 shows the turning rate during the redesigned maneuver, which now remains below the 15°/s limit.

The optimization was solved in 49 seconds with the first settings and 65 seconds with the reduced force. To illustrate the importance of the force penalty, the same problems using the cost function in Eqn. 11, without the force penalty, were solved in 142 and 277 seconds respectively. In a further experiment to test the problem-dependency of solution times, 783 randomly-generated, feasible problems, each involving similar obstacles and operating region as in this example, were solved using the cost function with force penalty. The average solution time was 11 seconds, with a maximum of 68 seconds.

4.2 Multiple Aircraft

Having verified the model in the previous section, this example applies it to a collision avoidance problem involving multiple aircraft. Three aircraft, similar to those used in the previous example, are required to traverse different diameters of a circle, as shown in Fig. 7. Their starting points are marked by squares and destinations by stars. The straight line paths to the destinations, shown dotted in the figure, would clearly lead to a collision. The designed trajectories, also shown in Fig. 7, form a 'roundabout' maneuver and successfully avoid collision with minimal deviation. The square exclusion region is 1m across. A similar result was shown in [1], found by an iterative process involving the necessary conditions for optimal avoidance. This paper has repeated that result by direct optimization. The trajectories also demonstrate the co-operative nature of this optimization method.

The heavy dots mark the positions of the aircraft at the 18^{th} time step. The exclusion regions around these positions are shown by the dotted boxes. Observe that the vehicles are separated by exactly the safety distance in the X-direction, illustrating the efficiency of the formulation and the direct physical significance of the avoidance distance. This contrasts with penalty methods such as potential functions [12], in which the avoidance weighting is not so obviously related to the achieved distance and may need tuning.

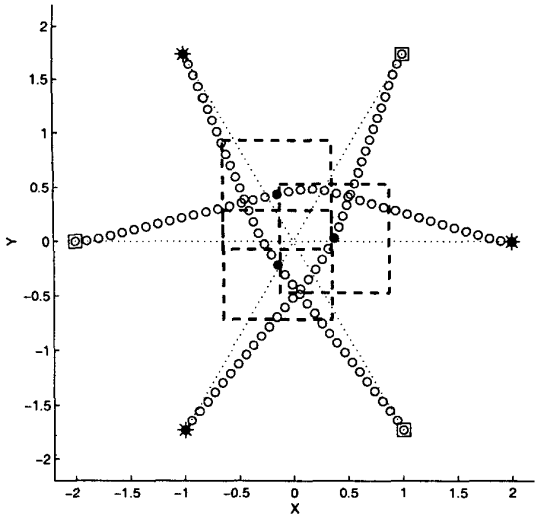

Fig. 7: The designed trajectories for the aircraft. The stars mark the target positions.

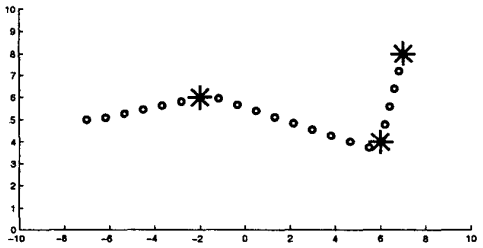

Fig. 8: Designed trajectory for an aircraft to visit three waypoints, marked by stars, in any order. The aircraft starts at the left.

4.3 Multiple Waypoints

This section presents an example of the multiple waypoint formulation, in which the aircraft must visit several different points in an unspecified order. In Fig. 8, the waypoints are marked by stars. The algorithm has chosen the ordering to give the fastest trajectory visiting all three points. In Fig. 9, an obstacle blocks the path used in Fig. 8. The algorithm now selects a different ordering to visit the points in minimum time. This illustrates the potential of this method for solving planning problems including aspects of high-level co-ordination.

5 Conclusion

It has been shown by theory and example that a constant speed, limited turn rate vehicle, such as an aircraft, can be modeled as a point mass with limited speed and subject to limited force in a minimum time problem. This has been combined with collision avoidance constraints to form a mixed integer linear program, which can be solved by a commercial software package. A further extension generates trajectories passing through a set of waypoints, the order of which is determined in the optimization for minimum completion time. Examples have been given to show that the approximate modeling is valid and to demonstrate the technique applied to various collision avoidance problems.

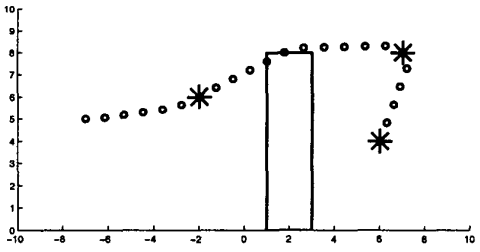

Fig. 9: Designed trajectory for the same task as in Fig. 8 but with an obstacle blocking the path used previously. The order of visiting the waypoints is changed.

References

[1] A. Bicchi, L. Pallottino "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 221-231, Dec. 2000

[2] T.S. Perry, "In Search of Future of Air Traffic Control," *IEEE Spectrum*, vol. 34, pp. 18-35, Aug. 1997.

[3] P.R. Chandler, M. Pachter, S. Rasmussen, "UAV Cooperative Control," *IEEE ACC*, June 2001.

[4] Z.-H. Mao, E. Feron, K. Bilimoria "Stability of Intersecting Aircraft Flows under Decentralized Conflict Avoidance Rules," *AIAA GN&C Conf.*, Aug. 2000.

[5] T. Schouwenaars, B. DeMoor, E. Feron and J. How, "Mixed integer programming for safe multi-vehicle cooperative path planning," *ECC*, 2001.

[6] Robert Fourer, David M. Gay, and Brian W. Kernighar. *AMPL, A modeling language for mathematical programming.* The Scientific Press, 1993.

[7] C. A. Floudas *Nonlinear and Mixed-Integer Programming - Fundamentals and Applications* Oxford University Press, 1995.

[8] A. Bemporad and M. Morari, "Control of Systems integrating Logic, Dynamics and Constraints," *Automatica* 35(1999), Pergamon Press, UK, pp. 407-427

[9] H. P. Williams and S. C. Brailsford, "Computational Logic and Integer Programming," in *Advances in Linear and Integer Programming*, Editor J. E. Beasley, Clarendon Press, Oxford, 1996, pp. 249–281.

[10] A. Richards, J. How, T. Schouwenaars and E. Feron, "Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming," *AIAA GN&C Conf.*, Aug. 2001.

[11] H. P. Rothwangl, "Numerical Synthesis of the Time Optimal Nonlinear State Controller via Mixed Integer Programming," *ACC*, 2001.

[12] J.-H. Chuang, "Potential-Based Modeling of Three-Dimensional Workspace for Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 5, October 1998.

[13] *ILOG CPLEX User's guide.* ILOG, 1999.