Optimizer Algorithms for Supervisory Control : Enabling Supervisor Input

Document information	
Project title	SUPEROPT
Project N°	E.02.01.
Project Manager	Company
Deliverable	Optimizer Algorithms for Supervisory Control : Enabling Supervisor Input
Deliverable ID	D2
Edition	1.1
Template version	02.00.00
Task contributors	
University of Bristol	

Abstract

This document reports the methods developed by the SUPEROPT project for supervisor interaction with optimizers. Two mathematical forms for an optimizer to support a Multi-Sector Controller are presented and compared, including various constraint forms to enable flexible but intuitive supervisor input. Enabling supervisor input to the optimization ensures the supervisor retains control of the high-level decision making while leaving the optimizer to perform the low-level detailed trajectory design.

Authoring & Approval

Prepared By		
Name & company	Position / Title	Date
Oliver Turnbull / UoB	Research Assistant	20/12/2012

Reviewed By		
Name & company	Position / Title	Date
Gillian Clare / University of Bristol	Research Assistant	21/12/2012

Approved By		
Name & company	Position / Title	Date
Arthur Richards / University of Bristol	Senior Lecturer / Project Lead	21/12/2012

Document History

Edition	Date	Status	Author	Justification
A.1	22/06/2012	Draft	ODNT	New Document
A.2	12/07/2012	Draft	ODNT	Revised including input from AGR
A.3	14/07/2012	Draft	AGR / ODNT	Final revisions
A.4	21/08/2012	Draft	ODNT	Inclusion of SESAR comments including separation of “Enabling Input” and “Informing” into separate documents.
1.0	20/12/2012	Final	ODNT	Updated document to include recent progress for final deliverable.
1.1	20/12/2012	Final	ODNT/AGR	Reviewed and updated.

IPR (foreground)

This deliverable consists of SJU foreground.

Table of Contents

List of tables

List of figures

1 Introduction

1.1 Purpose of the document

The goal of SUPEROPT is to investigate the interaction between an optimization and a human supervisor. "E.02.01-SUPEROPT-D1-PDLR Problem Definition and Literature Review" defined two challenge scenarios, the "Network Manager" (NM) and the "Multi-Sector Controller" (MSC), through which such interactions could be investigated. Table 1 presents a high-level review of some of the key parameters of the NM and MSC roles as implemented in this version of the models. Note that the purpose of SUPEROPT is to use these as case studies to investigate how humans and trajectory optimizers can cooperate: they are not intended to be a complete design of future roles in SESAR.

Role	Network Manager	Multi-Sector Controller
Candidate optimizer(s)	Simulated Annealing Genetic Algorithm	MILP Non-linear
Scale	FAB Flight-by-Flight Flow	Multi-Sector Area (MSA) Individual Aircraft Trajectories (with multiple, moving obstacle avoidance)
Planning Horizon	Up to 2 hours	20 minutes
Time-step	1 minute	2 minutes
How the supervisor can manipulate the optimizer	N/A	Selection of intuitive constraints Defined go-via points or corridors
How the optimizer informs the supervisor	List of ranked distinct solutions Load-Time profile	Active Constraints Trade-space of solutions (multi-objective)
Constraints	Runway capacities	Collision Avoidance Aircraft Dynamics

Table 1: Summary of NM and MSC roles

This document focuses on methods through which a human supervisor can influence the result of an optimization with the aim of separating the high-level decision making (human supervisor) from the low level detailed trajectory generation (optimizer). Specifically, this document uses the MSC case study to demonstrate novel forms of constraints to achieve these goals.

The remainder of this document is structured as follows: Section 3 presents a Mixed Integer Linear Program (MILP) optimization of the MSC scenario including a sub-section on conflict resolution sense constraints for high-level supervisor input; Section 4 mirrors the MILP MSC solution section but using a non-linear model of the aircraft dynamics with a polar set representation for collision avoidance; Section 5 presents some initial results on the relative computational complexity of each method.

1.2 Intended readership

This document is intended primarily for EUROCONTROL readership to report the principal project findings but may be of interest to other SESAR projects that wish to utilise any of the results.

1.3 Inputs from other projects

The Challenge Scenarios are developed around roles outlined in the PHARE [7] and ADAHR [8] EUROCONTROL projects.

1.4 Acronyms and Terminology

Term	Definition
AM	Airspace Manager
AMC	Airspace Management Cell
AOA	Aircraft Operator Agent
AOC	Airline Operations Centre
AOP	Airspace Operations Plan
ATFM	Air Traffic Flow Management
ATM:	Air Traffic Management
CDM	Collaborative Decision Making
DOC	Direct Operating Cost
EC	Executive Controller
E-ATMS	European Air Traffic Management System
FAB	Functional Airspace Block
GHA	Ground Handling Agent
LTM	Local Traffic Manager
MILP	Mixed Integer Linear Programming
MPC	Model Predictive Control
MSA	Multi-Sector Area
MSC	Multi-Sector Controller
MSP	Multi-Sector Planner
NM	Network Manager
PWA	PieceWise Affine
PC	Planner Controller

Term	Definition
RBT	Reference Business Trajectory
ROCD	Rate Of Climb/Descent
SBT	Shared Business Trajectory
SESAR	Single European Sky ATM Research Programme
SESAR Programme	The programme which defines the Research and Development activities and Projects for the SJU.
SJU	SESAR Joint Undertaking (Agency of the European Commission)
SJU Work Programme	The programme which addresses all activities of the SESAR Joint Undertaking Agency.
TBC	To Be Completed

1.5 Nomenclature

Symbol	Definition
N_a	Number of aircraft
a	Index of aircraft
$\boldsymbol{r}(t,a)$	Position of a at time t (decision variable)
$r_d(t,a)$	Position of a at time t in dimension d
$\boldsymbol{r}^R(t,a)$	Reference trajectory of a at time t (fixed parameter)
$\ddot{\boldsymbol{r}}(t,a)$	Acceleration of a at time t
k	Index of time-step
t_k	Sample time k
N_t	Number of time-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)

2 Introduction to MSC

The ATM environment encompasses a range of requirements: some are hard constraints, e.g. collision avoidance, while others may depend on user preferences, e.g. minimum cost or minimum time trajectories. One of the themes of the SUPEROPT project is developing methods to provide this flexibility within a supervised optimisation.

The primary way of enabling such inputs investigated in this section is the development of new forms of constraints. These are intended to capture supervisor desires in a way that is specific enough to avoid "surprises" in the results, yet flexible enough to ensure good performance overall.

This section presents two implementations of the MSC role using Mixed Integer Linear Programming (MILP) and Collocation optimizers, respectively. The MILP offers fast solutions and global optimality but a limited dynamics model (linear or piecewise linear) and discretized time. Meanwhile the collocation method allows a more thorough dynamics model and could enable the inclusion of noise or emissions models in the cost function but can be more difficult to solve, especially for the global optima.

2.1 MSC Cost Function

The cost function aims to capture mathematically the metrics of performance that the MSC wishes to "make small", but without any specific upper limit on their acceptable values. They should be made as small as possible within the constraints. (Alternatively some metrics may be desired to be maximized – this is easily handled with a minus sign.) Both formulations of the MSC share the cost function proposed in Equation 1.

Assume there are N_a aircraft in the MSA and that each aircraft a has Reference Business Trajectory (RBT) $r^R(t,a)$ 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 $r^R(t_f^R(a), a)$. This would typically refer to the pre-determined point, e.g. in the SBT, at which the aircraft is expected to exit the MSA. The optimizer designs for each aircraft a trajectory $r(t,a)$ from time t_0, i.e the current time, to $t_f(a)$, the new chosen time at which *a* exits the MSA. 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.

$$J = \sum_{a=1}^{N_a} \begin{bmatrix} \alpha t_f(a) \\ +\beta\left(r\left(t_f(a),a\right) - r^R\left(t_f^R(a),a\right)\right)^2 \\ +\gamma \sum_{k=1}^{N_t} \ddot{r}\,(t_k,a) \\ +\delta \sum_{k=1}^{N_t} \left(r(t_k,a) - r^R(t_k,a)\right)^2 \end{bmatrix} \qquad \text{Equation 1}$$

where: the weighting on final time reflects the desire to avoid delay; the weighting on deviation from exit point penalizes coordinations with the adjoining MSA and long term deviations from the RBT; and the acceleration ($\ddot{r}$) weighting is included to reduce manoeuvring and increase passenger comfort; 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 $(\alpha, \beta, \gamma, \delta)$. How these weights affect the result is an important question and is well studied in the field of multi-objective optimization. Further investigation into this aspect of SUPEROPT is on-going.

A further term could be added to the cost function (Equation 1) to directly account for each aircraft's fuel consumption. Figure 1 shows a clear relationship between fuel flow and the flight phase and flight level. In the case of a linear solver (such as that used for MILPs), the nonlinearity of the relationship could be approximated using a PieceWise Affine (PWA) function. An example of a MILP implementation of a PWA can be seen in Section 3. In the case of a nonlinear solver, such as that used with the collocation model, then the underlying functions could be implemented directly.

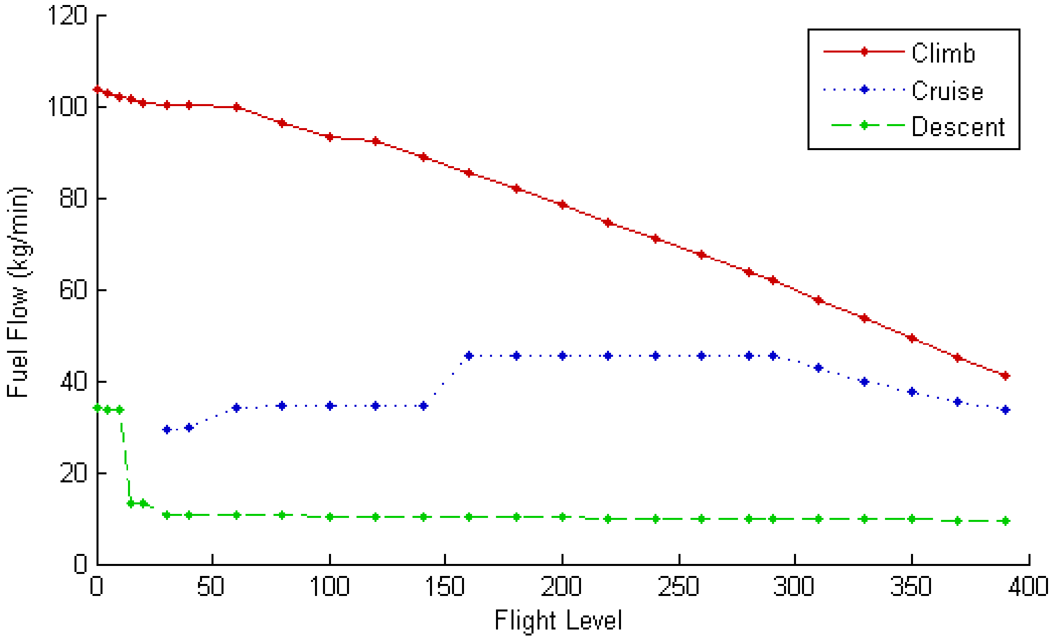


Figure 1: Variation of A319 fuel flow with flight phase and level

3 MSC MILP Solution

3.1 MILP MSC Model

This section presents the constraints required to approximate the aircraft dynamics in a Mixed Integer Linear Program. This section is primarily a review of [1] which contains the basic method and [2] which originally presented the 3-D model. The full model is included here for completeness.

First we constrain the vehicle kinematics and dynamics assuming a point mass model:

$$\boldsymbol{r}(a,k+1) = \boldsymbol{r}(a,k) + \boldsymbol{v}(a,k)dt \qquad \text{Equation 2}$$

$$\boldsymbol{v}(a,k+1) = \boldsymbol{v}(a,k) + \dot{\boldsymbol{v}}(a,k)dt \qquad \text{Equation 3}$$

$$\forall \ a \ \in \{1,\dots,N_a\}, k \ \in \{1,\dots,(N_t-1)\}$$

Where $\mathbf{r}$ is the aircraft's position; $\mathbf{v}$ the velocity, $\dot{\mathbf{v}}$ the acceleration; dt the length of each time-step; $\mathrm{N_a}$ the number of aircraft in the problem; and $\mathrm{N_t}$ the number of time-steps in the planning horizon.

An upper bound is placed on the aircrafts' airspeed:

$$\|\boldsymbol{v}(a,k)\|_2 \leq V_{max} \ \ \forall \ a \ \in \{1,\dots,N_a\}, k \ \in \{1,\dots,N_t\} \qquad \text{Equation 4}$$

This is non-linear so is implemented using a number of linear approximations:

$$\boldsymbol{v}(a,k) \cdot \boldsymbol{e}(\theta,\alpha) \ \leq \ V_{max}$$

$$\forall \ a \ \in \{1,\dots,N_a\}, k \ \in \{1,\dots,N_t\}, \alpha \ \in [-\pi,\pi], \theta \ \in [0,2\pi] \qquad \text{Equation 5}$$

Where the angle ranges both the inclination, α, and azimuth, θ are approximated by $\mathrm{N_c}$ discrete samples**.**

The initial position is fixed:

$$\boldsymbol{r}(a,0) = \boldsymbol{r}^{\boldsymbol{R}}(t_0,a) \ \ \forall \ \ a \ \in \{1,\dots,N_a\} \qquad \text{Equation 6}$$

taking the start point of the trajectory to be the current point of the aircraft.

Similarly we ensure that the destination is reached. Due to the fixed length of the time-step we allow a degree of flexibility in defining the target as an area rather than a point:

$$\sum_{k\in\{1,\dots,N_t\}} b_f(a,k) = 1 \quad \forall \ a \ \in \{1,\dots,N_a\} \qquad \text{Equation 7}$$

$$\boldsymbol{r}(a,\mathrm{k}) - \boldsymbol{r_f}(a) \leq \boldsymbol{R_f}(a) + \mathrm{M}(1-\mathrm{b_f}(\mathrm{a},\mathrm{k})) \qquad \text{Equation 8}$$

$$\boldsymbol{r_f}(a) - \boldsymbol{r}(a,\mathrm{k}) \leq \boldsymbol{R_f}(a) + \mathrm{M}(1-\mathrm{b_f}(\mathrm{a},\mathrm{k})) \qquad \text{Equation 9}$$

$$\forall \ \ a \ \in \{1,\dots,N_a\}, k \in \{1,\dots,N_t\}$$

$$\boldsymbol{R_f}(a) \leq \boldsymbol{R_{f_{max}}} \quad \forall \ \mathrm{a} \ \in \{1,\dots,\mathrm{N_a}\} \qquad \text{Equation 10}$$

Where $\mathrm{b_f}$ is a set of binary variables to indicate the time-step at which each aircraft arrives at its destination, hence $t_f(a)$ in Equation 1 can be represented as $\sum_k kb_f(a,k)$; r_f is the destination (or finish) for each aircraft; $\boldsymbol{R_f}(a)$ is a decision variable representing the distance to the destination at the arrival time-step for each aircraft, a; M is a vector of large positive constants; $\boldsymbol{R_{f_{max}}}$ is the maximum permitted distance in each dimension of any aircraft from its destination at the arrival time-step, *i.e.* the size of the target area.

Equation 7 ensures that one of the time-steps is taken as the arrival time at the destination while Equation 8 and Equation 9 determine the distance to the destination at that time; the binary variable set in Equation 7 combined with M relax the constraint at all time-steps except that chosen as the finish time. Finally Equation 10 ensures that the distance to the destination at the arrival time is within a specified bound. The arrival time term in the cost can be captured via $t_f(a) = \sum_k b_f(a,k)t_k$.

3.1.1 Aircraft Performance Model

In order to extend the previous 2-D model [1] to 3-D while maintaining realistic aircraft dynamics, it is necessary to introduce a performance model to the trajectory generator/optimization.

The EUROCONTROL Base of Aircraft Data (BADA) provides both an analytical model and a database of aircraft performance for typical commercial aircraft. The BADA User Manual [3] states that the longitudinal and normal acceleration for civil airliners is limited to 2 and 5 fps^2 respectively.

Acceleration constraints are applied in the horizontal and vertical directions, respectively. Since MILP requires the use of a global frame of reference, it is impossible to distinguish between longitudinal and lateral acceleration so the lower limit of 2 fps^2 is applied in all horizontal directions, represented by a limit $\mathrm{A}_{\mathrm{horiz}}^{\max}$. Vertical acceleration is constrained to be less than 5 fps^2 up or down.

Trajectory generation using MILP requires the use of a global frame of reference. If we assume small angles of attack and bank angles then we can approximate the longitudinal and normal accelerations as horizontal and vertical limits respectively:

$$\mathrm{e}(\theta, 0) \cdot \dot{\boldsymbol{v}}(k,a) \le \mathrm{A}_{\mathrm{horiz}}^{\max} \qquad \text{Equation 11}$$

$$-\mathrm{A}_{\mathrm{vert}}^{\max} \le [0\ 0\ 1]\dot{\boldsymbol{v}}(k,a) \le \mathrm{A}_{\mathrm{vert}}^{\max} \qquad \text{Equation 12}$$

$$\forall\, a \in \{1..N_a\}, k \in \{1..N_t - 1\}$$

Where N_a is the total number of aircraft; N_t is the number of time-steps. The angular range is again approximated over a discrete set of N_c angles.

To model individual aircraft dynamics more precisely, the Rate Of Climb/Descent (ROCD) has been limited according to BADA. Taking the data for a typical aircraft (Airbus A319), operating at its nominal weight, it is clear that the permitted ROCD is dependent on flight regime and level [3] and that this data can be approximated by suitable Piecewise Affine (PWA) functions for climb and descent.

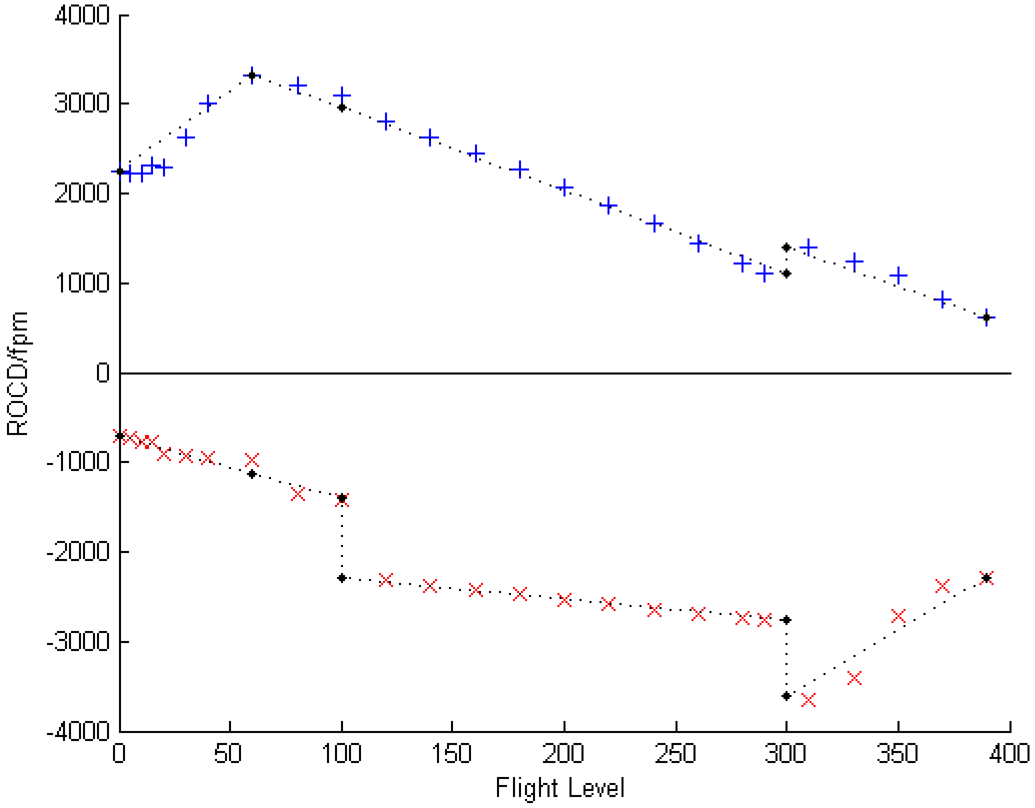


Figure 2: A319 Rate Of Climb/Descent approximated with a piecewise affine function

The PWA function can easily be implemented using MILP as follows:

$$\sum_m b_{ROCD}(i,k,m) = 1 \qquad \text{Equation 13}$$

$$\sum_m \lambda_{ROCD}(i,k,m) \geq 0 \qquad \text{Equation 14}$$

$$\sum_m \lambda_{ROCD}(i,k,m) \leq \begin{cases} b_{ROCD}(i,k,m), & for\ m = 1 \\ b_{ROCD}(i,k,m-1) + b_{ROCD}(i,k,m), & for\ m \in \{2,\dots,N_{P_{ROCD}}\} \\ b_{ROCD}\left(i,k,N_{P_{ROCD}}(i)-1\right), & for\ m = N_{P_{ROCD}} \end{cases}$$

Equation 15

$$\sum_m \lambda_{ROCD}(i,k,m) B_{ROCD}(i,m) = r_z \qquad \text{Equation 16}$$

$$v_z(i,k) \leq \sum_m \lambda_{ROCD}(i,k,m) A_{ROC}(i,m) \qquad \text{Equation 17}$$

$$v_z(i,k) \geq \sum_m \lambda_{ROCD}(i,k,m) A_{ROD}(i,m) \qquad \text{Equation 18}$$

$\forall\ i \in \{1,\dots,N_a\}, k \in \{1,\dots,N_t - 1\}, m \in \{1,\dots,N_{P_{ROCD}}\}$

where $\boldsymbol{B}_{\boldsymbol{ROCD}}(i)$ is a vector of flight levels at which 'the rate of climb or descent function changes' and $A_{ROC}(i,m)$ and $A_{ROD}(i,m)$ are the maximum rate of climb or descents respectively at the altitudes specified in $\boldsymbol{B}_{\boldsymbol{ROCD}}$.

3.1.2 Avoidance Constraints

A final modification to the 2-D formulation [1] was to establish independent horizontal and vertical separation distances; this reflects current ATM practice where typically horizontal separation is 1000ft compared to 5nmi (approximately 30400ft) horizontally [14]. The avoidance constraints are formulated in MILP as follows:

$$r_d(a_1,k) \le r_d(a_2,k) - D_H + M\big(1 - b_a(a_1,a_2,k,d)\big) \qquad \text{Equation 19}$$

$$r_d(a_1,k) \ge r_d(a_2,k) + D_H - M\big(1 - b_a(a_1,a_2,k,d+N_d)\big) \qquad \text{Equation 20}$$

$$\forall\, a_1 \in \{1,\dots,N_a\}, a_2 \in \{1,\dots,N_a\}, d \in \{1,\dots,2\}, k \in \{2,\dots,N_t\} \;:\; a_1 > j$$

Where $r_d(a_1,k)$ is element *d* of the position vector of aircraft a_1, at timestep k ; D_H is the horizontal avoidance distance (5nmi); M is a large positive scalar; and $b_a(a_1,a_2,k,d)$ is a binary variable used to relax the avoidance constraint in all but one of the directions $+x,+y,+z,-x,-y,-z$, i.e. it is only necessary to avoid the obstacle in one direction at each time-step. Equation 19 and Equation 20 constrain the distance between a_1 and a_2 to be greater than the horizontal avoidance distance in the positive and negative direction, respectively, for a given dimension d, i.e.:

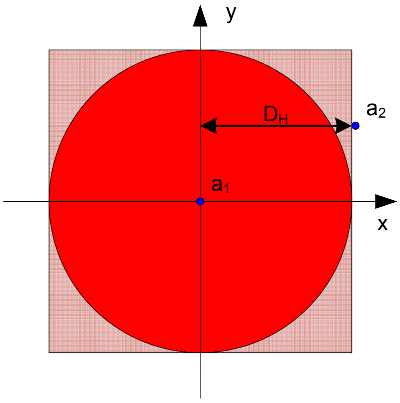


Figure 3: MILP approximation to horizontal avoidance criteria

In the vertical direction we require just two binary variables:

$$r_z(a_1,k) \le r_z(a_2,k) - D_V + M\big(1 - b_a(a_1,a_2,k,3)\big) \qquad \text{Equation 21}$$

$$r_z(a_1,k) \ge r_z(a_2,k) + D_V - M\big(1 - b_a(a_1,a_2,k,6)\big) \qquad \text{Equation 22}$$

$$\forall\, a_1 \in \{1,\dots,N_a\}, a_2 \in \{1,\dots,N_a\}, k \in \{2,\dots,N_t\} \;:\; a_1 > a_2$$

Where D_V is the vertical avoidance distance (1000ft). Equation 21 and Equation 22 constrain the vertical separation in the same manner as Equation 19 and Equation 20 do horizontally.

Finally:

$$\sum_{m=1}^{6} b_a(a_1,a_2,k,m) = 1 - \sum_{l=1}^{k-1} \sum_{a\in\{a_1\cup a_2\}} b_f(a,l) \qquad \text{Equation 23}$$

$$\forall\, a_1 \in \{1,\dots,N_a\}, a_2 \in \{1,\dots,N_a\}, k \in \{2,\dots,N_t\} \;:\; a_1 > a_2$$

Where $b_f(a,l)$ is set of binary variables indicating the finishing time from Equation 7. Equation 23 ensures that at least one of the avoidance binaries previously constructed is enforced at each time-step between each pair of aircraft; the sum at the end of the equation is required to ensure that the optimization does not plan for an aircraft once it has reached its destination.

In cases where the avoidance distance is large in comparison with the time-step, the above formulation is sufficient. However, to reduce the number of decision variables when planning over longer intervals, the length of each time-step is often increased. Once the distance that an aircraft can travel in a single time-step is larger than the avoidance distance then the above formulation can lead to optimized trajectories containing conflicts due to "corner-cutting". Consider the case shown in Figure 4 where the avoidance constraints are valid at times k and $k-1$ but the two aircraft clearly come into conflict between the time-steps.

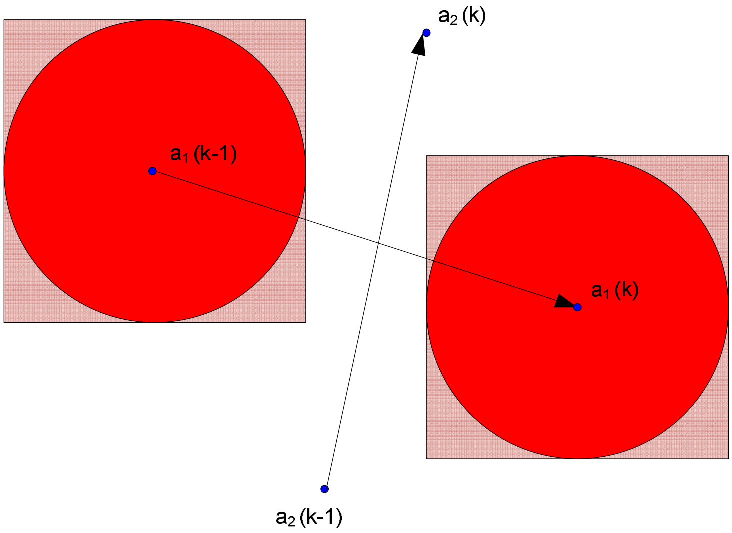


Figure 4: MILP "corner cutting"

To overcome this problem, it is proposed to enforce the constraints at the current time of each aircraft step and also the previous time-step of both aircraft *with the same binaries*, i.e. between: $a_1(k)$ and $a_2(k)$; $a_1(k)-1$ and $a_2(k)$; $a_1(k)$ and $a_2(k-1)$; $a_1(k-1)$ and $a_2(k-1)$. This approach is equivalent to ensuring that the line segments, representing the aircraft trajectory during an entire time-step, do not intersect, by ensuring a common separating plane between the two segments. Note that this introduces more constraints but does not introduce more binary variables.

Due to the approximation of the MILP avoidance constraints the solution can be slightly conservative: rather than only the line segments not intersecting, we must also consider the overall bounding box as shown in Figure 5 where we wish to avoid the darker region (a fixed distance perpendicular to the trajectory) but the above formulation prohibits other aircraft passing anywhere within the entire shaded region. To alleviate this conservatism it is simple to approximate the circular avoidance regions at each time step with a greater number of constraints. The effect of an increased number of constraints on inter-sample avoidance is shown in Figure 6.

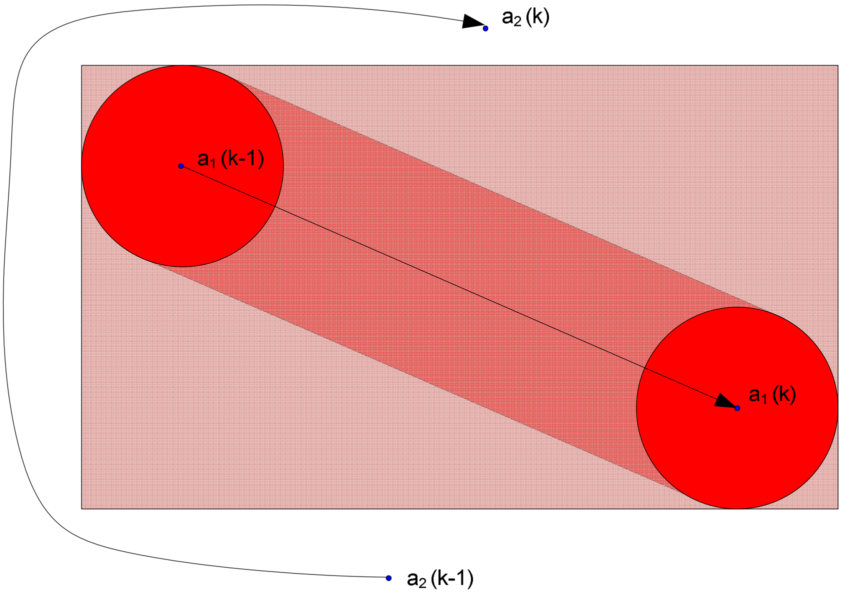


Figure 5: Inter-sample avoidance with MILP avoidance constraints

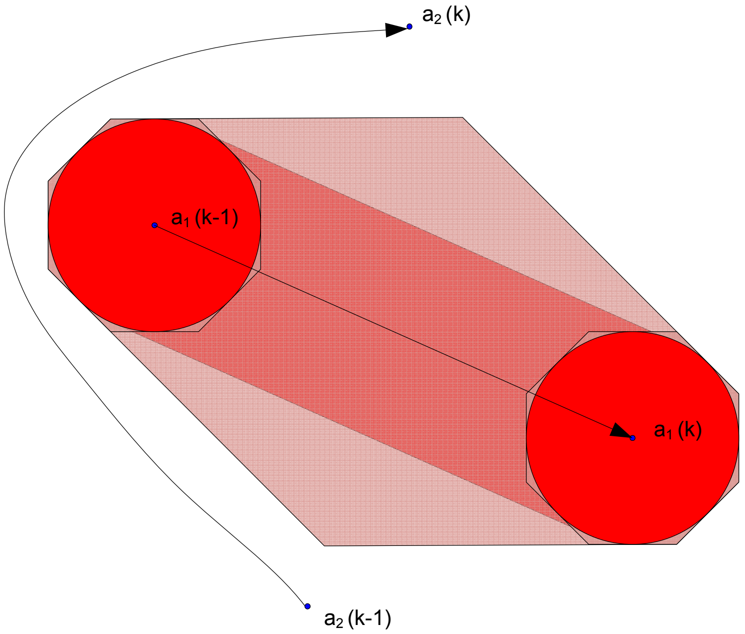


Figure 6: Refined inter-sample MILP avoidance constraints

Formalizing the above concepts requires us to define the avoidance binaries between all aircraft at all time-steps along both trajectories, i.e.:

$$\cos\left(\frac{2\pi\theta}{N_\theta}\right)(r_x(a_1,k_1)-r_x(a_2,k_2))-\sin\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_y(a_1,k_1)-r_y(a_2,k_2)\right)\geq D_H-M\left(1-b_a(a_1,a_2,k_1,k_2,\theta)\right)$$

Equation 24

$$\cos\left(\frac{2\pi\theta}{N_\theta}\right)(r_x(a_1,k_1)-r_x(a_2,k_2-1))-\sin\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_y(a_1,k_1)-r_y(a_2,k_2-1)\right)\geq D_H-M\left(1-b_a(a_1,a_2,k_1,k_2,\theta)\right)$$

Equation 25

$$\cos\left(\frac{2\pi\theta}{N_\theta}\right)(r_x(a_1,k_{1-1})-r_x(a_2,k_2))-\sin\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_y(a_1,k_1-1)-r_y(a_2,k_2)\right)\geq D_H-M\left(1-b_a(a_1,a_2,k_1,k_2,\theta)\right)$$

Equation 26

$$\cos\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_x(a_1,k_1-1)-r_x(a_2,k_2-1)\right)-\sin\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_y(a_1,k_1-1)-r_y(a_2,k_2-1)\right)\geq D_H-M\left(1-b_a(a_1,a_2,k_1,k_2,\theta)\right)$$ Equation 27

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

Finally we must modify Equation 23 to ensure that each pair of aircraft are separated in at least one direction at each time-step while accounting for either aircraft reaching their destination before the end of the planning period:

$$\sum_{\theta=1}^{N_\theta} b_a(a_1,a_2,k_1,k_2,\theta)=1-\sum_{k_3=1}^{k_1-1} b_f(a_1,k_3)-\sum_{k_4=1}^{k_2-1} b_f(a_2,k_4)$$ Equation 28

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

3.2 Conflict Resolution Sense Constraints

In 2-D, it is clear that conflicts between aircraft can be divided into distinct classes of solutions [5], e.g. left or right, or ahead or behind. These classes are referred to here as the "sense" of the resolution. The first mechanism for supervisor input here is the ability to constrain which sense of resolution is adopted without otherwise constraining the path. An initial formulation to achieve this was presented in [1]. The idea of requiring a particular sense translates readily to 3-D, with an MSC requiring ahead, behind, over or under, for example. However, the classes are not as clearly defined in 3-D as in 2-D as there is always a continuum of paths around any conflict. This section develops a 3-D sense constraint approach to overcome this challenge.

3.2.1 Vertical Resolution

In addition to the methods proposed in the SUPEROPT Problem Definition and Literature Review [2], a perhaps more intuitive approach to enforcing a particular sense of resolution is to fix some of the variables in the problem. Forcing resolution in a particular direction can be achieved through fixing the trajectory in the other directions to the RBT. For example, to force a horizontal resolution we would fix the aircraft's altitude:

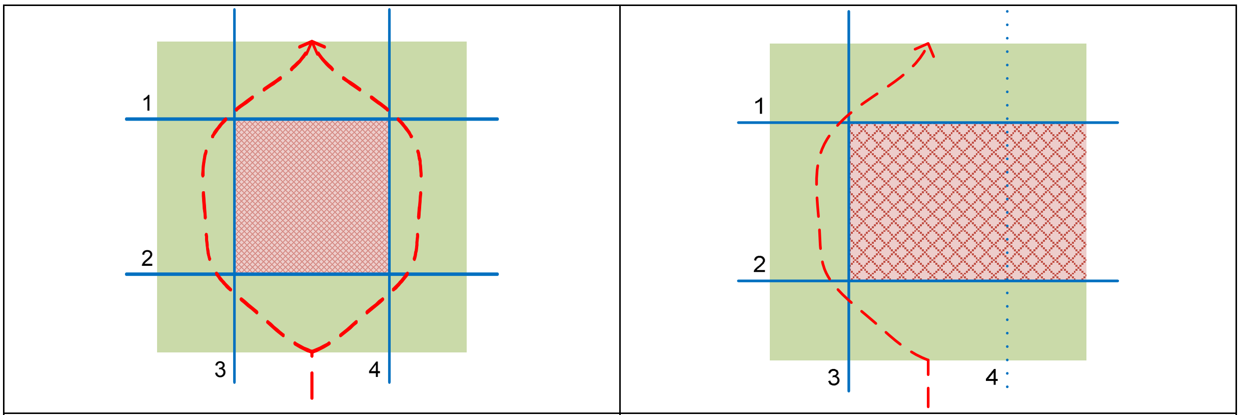
$$r_3(k,a)=r_3^R(k,a)\ \forall \quad k\in\{1,\dots,N_t\}, a\in\{1,\dots,N_a\}$$ Equation 29

Where $r_3(k,a)$ is the altitude of aircraft a at time-step k.

Conversely to enforce vertical resolution we would fix the longitude and latitude of the trajectory

$$r_d(k,a)=r_d^R(k,a)\ \forall \quad d\in\{1,2\}, k\in\{1,\dots,N_t\}, a\in\{1,\dots,N_a\}$$ Equation 30

The simplest way of constraining the sense of a conflict resolution is to fix the binary avoidance variables to prohibit passing on an undesired side of an obstacle. Consider the case illustrated in Figure 7a where there is a square, 2-D obstacle which can be avoided on any side, i.e. with any sense. Now in Figure 7b, the binary allowing the trajectory to pass to the 'right' of the object has been fixed to prohibit this possibility; binaries 1 and 2 have not been fixed as we do not wish to enforce that the path is to the 'left' of the obstacle at all times. In dense traffic situations, sense might become more complicated to constrain, and then the methods presented in [1] must be employed.



(a) Unconstrained resolution

(b) Restricted sense resolution

Figure 7: Constraining a resolution sense through fixing binary variables

Considering again the case of vertical resolution, if we require aircraft 1 to go over aircraft 2 then we constrain the binaries as follows:

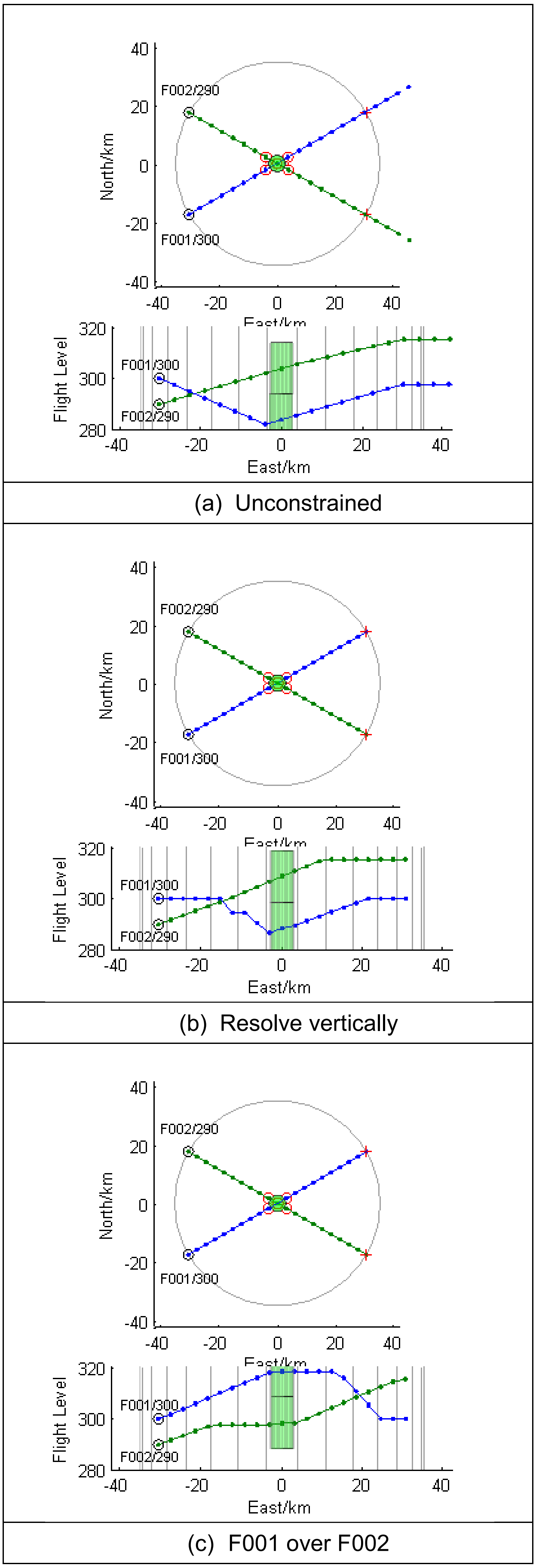
$$b_a(2,1,k,6) = 0 \quad \forall \quad k \in \{1, \dots, N_t\}$$ Equation 31

Figure 8 shows some results using vertical resolution. Each sub-figure shows a pair of trajectories for two aircraft (travelling West to East) projected in the horizontal and vertical planes. The aircraft's closest approach is highlighted with a cylinder representing half of the conflict distance in each direction, i.e. if the cylinders intersect there is a conflict (in which case they would be shaded red). It should be noted that in order to emphasise the different behaviours, the avoidance criteria where changed to 3nm and approximately 2000ft.

Figure 8a shows an unconstrained avoidance scenario with two aircraft travelling West to East, with flight F002 climbing up through F001. In the unconstrained case, separation is achieved through vertical separation and the cost is reduced by increasing the velocity of both aircraft so that they arrive at the destination earlier.

Figure 8b shows the same scenario but with the latitude and longitude of the trajectory fixed to force vertically resolution of the conflict. Whilst this may be expected to reach the same solution as the unconstrained problem (since in that case the conflict was also resolved vertically), the trajectories differ significantly, this is due to fixing the horizontal position of the aircraft which prohibits accelerating to reach the destination earlier.

To force a particular sense of vertical resolution we can fix the appropriate avoidance binary with the result shown in Figure 8c where F001 passes above F002, i.e. the sense of the resolution is reversed compared with Figure 8b.



(a) Unconstrained

(b) Resolve vertically

(c) F001 over F002

Figure 8: Vertical resolution results

It is noted that some instances, one aircraft performs additional climb or descent than that required to reach it's destination and is then forced to descend or climb in the other direction. Such behaviour is undesirable due to the increased cost and passenger discomfort. One approach to resolve such behaviour would be to require a monotonic variation in altitude:

$$\mathrm{r_z(i,k) > r_z(i-1,k)\ \forall\, i\ \in \{1,\dots N_a\}, k\ \in \{1,\dots N_t\}:\ r_z(i-1,k) > r_z(i-2,k)} \qquad \text{Equation 32}$$

$$\mathrm{r_z(i,k) < r_z(i-1,k)\ \forall\, i\ \in \{1,\dots N_a\}, k\ \in \{1,\dots N_t\}:\ r_z(i-1,k) < r_z(i-2,k)} \qquad \text{Equation 33}$$

However, it is not always possible to enforce monotonic flight profiles in the example in Figure 8c, as the aircraft begin too close in height to achieve vertical separation without one aircraft having to both climb and descend. Adding the monotonicity constraint makes the optimization infeasible. This motivates the development of constraint prioritization, in which constraints are enforced in some defined ordering up to the point where any more would make the problem infeasible.

3.2.2 Temporal Resolution

An alternative representation of resolution sense is for A to pass *ahead* of B (or vice versa). This equates to a constraint that B cannot occupy any point in space that A currently occupies *or will occupy in the future.* If it did, B could reach the crossing point of the two trajectories before A, violating the sense constraint. Hence, for sense constraints in temporal form, avoidance must be enforced between a pair of vehicles at pairs of different time steps. When combined with the previous method of fixing the vertical or horizontal aspects to the RBT, this provides a powerful formulation to enforce flight A to pass ahead of (or behind) flight B.

Suppose that aircraft a_1 is required to pass ahead of aircraft a_2. We have already defined avoidance binaries between the pair of aircraft at all pairs of time-steps so all that remains is to enforce them at additional time-steps:

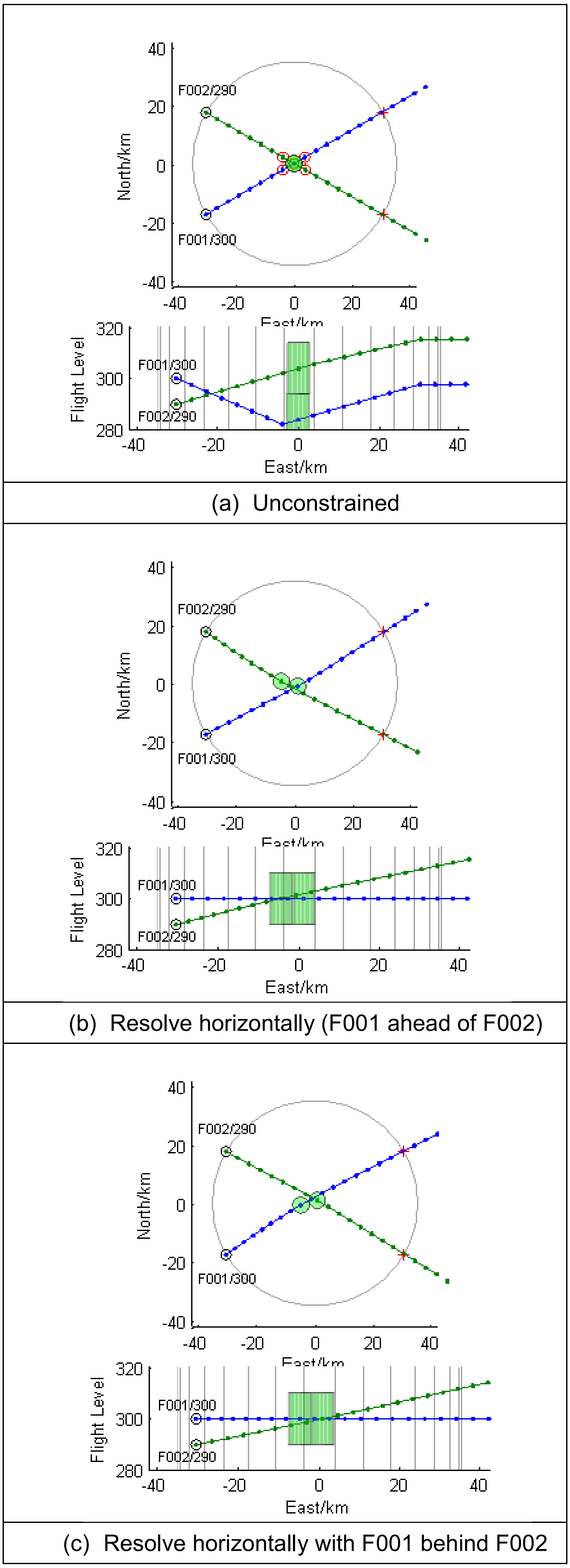
$$\cos\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_x(2,k_1)-r_x(1,k_2)\right)-\sin\left(\frac{2\pi\theta}{N_\theta}\right)\left(r_y(2,k_1)-r_y(1,k_2)\right)\geq D_H-M\left(1-b_a(2,1,k_1,k_2,\theta)\right) \qquad \text{Equation 34}$$

$$\forall\quad k_1\in\{2,\dots,N_t\}, k_2\in\{2,\dots,N_t\}, \theta\ \in\{1,\dots,N_\theta\}:\ (k_1\geq k_2)$$

Where the notation is as before. It should be noted that the principal difference between the above equations and Equation 24 (without the sense constraints) is the extra conditions under which the constraints are applied, i.e. when $k_1 > k_2$, this forces the avoidance of all future positions of a_1 by a_2.

Figure 9 shows an example of temporal resolution using the same scenario as the vertical resolution example in Figure 8. It should be noted that in this case, despite fixing one-dimension of the trajectory, the resulting solution has allowed the aircraft to accelerate and reach the target earlier; this occurs as the vertical distance to the destination at the closest approach is sufficiently small that the cost of arriving earlier is lower than the cost of not achieving the precise altitude of the destination at the arrival time-step.

Finally, should we wish to reverse the sense of the horizontal resolution we can request F001 to pass behind F002 (Figure 9c).



(a) Unconstrained

(b) Resolve horizontally (F001 ahead of F002)

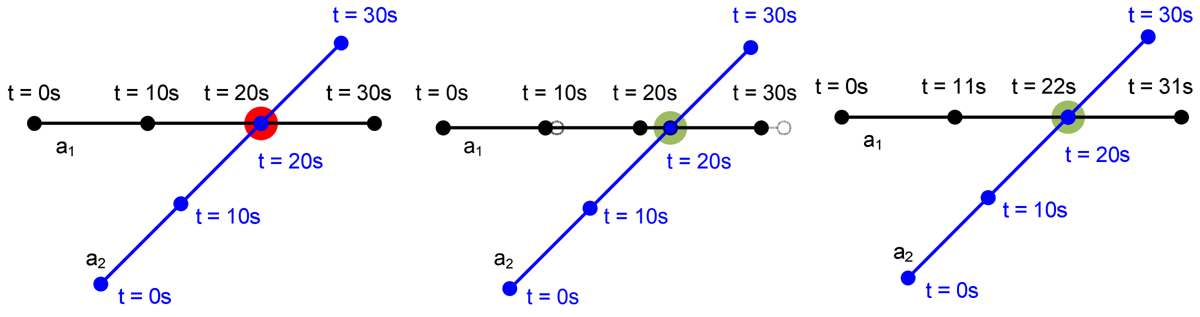
(c) Resolve horizontally with F001 behind F002

Figure 9: Temporal resolution results

3.2.3 Velocity Resolution

Alternatively, conflicts can be avoided through “speed advisories”, i.e. one or more conflicting aircraft are requested to alter their airspeed to ensure safe separation is maintained at all times. Previous results have shown on a larger scale, that this approach can be effective enough to avoid conflicts ever developing [15].

Implementation of a speed advisory in a MILP implies that we wish the flightpath of the aircraft to follow the RBT but for the location of the aircraft at each time-step to be suitably adjusted. In the case of the vertical and temporal resolution methods outlined above, we fixed the aircraft trajectory in one or two dimensions to enforce resolution in the remaining dimension(s). A speed advisory can be enforced by fixing the aircraft in all 3 spatial dimensions, however, due to the fixed time-step in MILP we can not simply force the location of the new waypoints to be the same as that of the RBT as we would then have effectively fixed the trajectory in all 4 dimensions and thus prevented any form of deviation. Instead, we require that the new way-points occur at some point along the RBT as shown in Figure 10.



(a)RBTs

(b)Resolved trajectories with a fixed time-step (MILP)

(c)Resolved trajectories with a variable time-step

Figure 10: Velocity resolution

It is relatively trivial to force a waypoint to lie on 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* line segment the new point should lie on which is best achieved through another PWA. Equation 34 selects the line segment that aircraft *a*, occupies at time-step *k*.

$$\sum_{j=1}^{N_t-1} b_k(a,k,j) = 1 \qquad \text{Equation 35}$$

$$\forall \quad k \in \{1,\dots,N_t\}, a \in \{1,\dots,N_a\}$$

Where $b_k(k,j,a)$ is a binary variable. Now we must constrain each new way-point to lie on the selected line-segment:

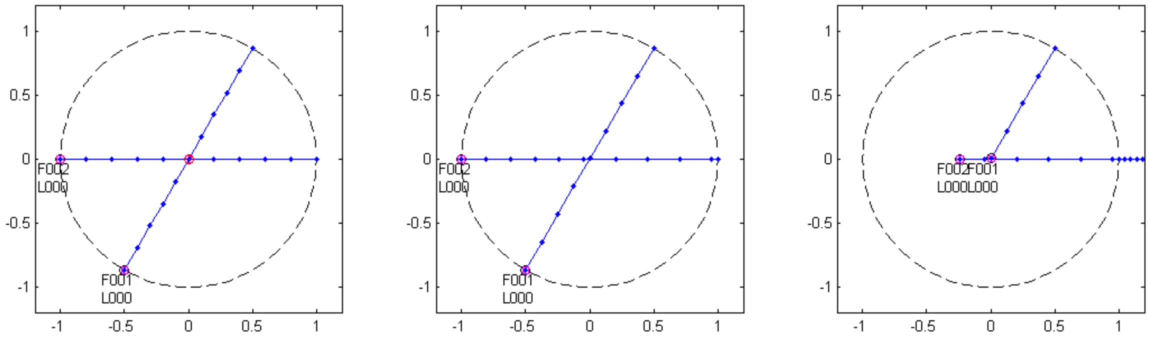
$$\sum_{\mathrm{j}}^{N_t} c_k(a,k,j) \le \begin{cases} b_k(a,k,j), & for\ j = 1 \\ b_k(a,k,j-1) + b_k(a,k,j), & for\ j \in \{2,\dots,\mathrm{N_t}-1\} \\ b_k(a,k,j-1), & for\ j =\ \mathrm{N_t} \end{cases} \qquad \text{Equation 36}$$

$$\boldsymbol{r}(a,k) = \sum_{j=1}^{N_t} \boldsymbol{r}^R(a,j)\, c_k(a,k,j) \pm \partial r \qquad \text{Equation 37}$$

$$\forall \quad k \in \{1,\dots,N_t\}, a \in \{1,\dots,N_a\}$$

Where $c_k(a,k,j)$ is the relative distance of aircraft *a* at time *k* from point *j* of its RBT, $\boldsymbol{r}^R(a,j)$ and ∂r is a small tolerance applied to ensure that the new trajectory remains dynamically feasible.

Figure 11 shows the effect of the above method on a 2D example of two aircraft. From the figure it can be seen that both aircraft remain on their original flightpath but that F001 accelerates and F002 decelerates in order to ensure safe separation. Due to the inter-sample avoidance it can be seen in Figure 11c that the final result is slightly conservative; further refining the constraints could resolve this.



(a)RBTs

(b)Resolved trajectories

(c)Aircraft location at 4th time-step

Figure 11: Velocity resolution results (2D)

4 Collocation MSC Solution

4.1 Nonlinear Model of Aircraft Dynamics

This section presents a collocation model of the aircraft dynamics. Collocation methods approximate the state of an optimal control problem by a basis of polynomials [10] and are an active area of research for problems with 'hard' nonlinear dynamics [11].

The method solves for the coefficients of the Lagrange interpolating polynomial coefficients to the aircraft dynamics. The coefficients can be used to give the planned velocity and position of the vehicle at any time between the start and goal. For the MSC role this is particularly useful as it enables us to derive accurate points for diverting around aircraft/obstacles.

Due to the non-linear nature of the collocation method, the optimizer is unable to guarantee a globally optimal solution. It is noted that the trajectory output from the optimizer is sensitive to the initial guess supplied as an input. The model presented here simply uses a straight line between the aircrafts intial and final positions as the input. Refining the initial guess would decrease solution time and has the potential to offer a lower cost solution. One option would be to use a simple search such as potential fields to generate a feasible initial guess. Alternatively, in some implementations (dependent on planning horizon and timescales) it may be appropriate to initialize the collocation method using a simplified version of the MILP model.

We now develop the collocation model. Initially we consider time as normalized over, $\tau \in \{-1,1\}$, before introducing a global time variable, t, in Equation 46, enabling us to have the duration of the trajectory as a variable in the optimization.

At the highest level, our aim is to minimize some function of our states, $\boldsymbol{x}(\tau)$ and inputs, $\boldsymbol{u}(\tau)$, i.e:

$$\min_{\boldsymbol{u}(t),\boldsymbol{x}(\tau)} \int f(\boldsymbol{x},\boldsymbol{u})d\tau$$ Equation 38

Subject to the system dynamics:

$$\dot{\boldsymbol{x}}(\tau) = g(\boldsymbol{x},\boldsymbol{u})$$ Equation 39

If we consider the aircraft's trajectory at n collocation points then we can represent this as an $n-1^{th}$ order polynomial:

$$x(\tau) = p_1 + p_2\tau + \cdots + p_n\tau^{n-1}$$ Equation 40

To determine the coefficients of the Lagrange interpolating polynomial we define $x(\tau_i) = x_i$ and $u(\tau_i) = u_i$ for $i = 1, \dots, n$. If $\boldsymbol{M}$ is the Vandermonde matrix of order n, then we can write that:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 & \tau_1 & \tau_1^2 & \cdots & \tau_1^{n-1} \\ 1 & \tau_2 & \tau_2^2 & \cdots & \tau_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \tau_n & \tau_n^2 & \cdots & \tau_n^{n-1} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}$$ Equation 41

Or:

$$\underline{x} = \boldsymbol{M}\underline{p}$$ Equation 42

Where $\underline{\boldsymbol{p}}$ is the vector of polynomial coefficients, i.e.:

$$\underline{p} = \boldsymbol{M}^{-1}\underline{x}$$ Equation 43

Differentiating Equation 40 gives:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \tau_1 & \cdots & \tau_1^{n-2} \\ 0 & 1 & \tau_2 & \cdots & \tau_2^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \tau_n & \cdots & \tau_n^{n-2} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}$$

Equation 44

Or:

$$\underline{\dot{x}} = \boldsymbol{N}\underline{p} = \boldsymbol{NM}^{-1}\underline{x} = \boldsymbol{C}\underline{x}$$

Equation 45

That is, we can now express $\dot{\boldsymbol{x}}$ purely in terms of $\boldsymbol{x}$ at the collocation points. For the variable time form of the planning problem we apply the following transformation to map the trajectory onto the desired panning interval:

$$t = \left[\left(t_f - t_0\right)\tau + t_f + t_0\right]/2$$

Equation 46

Where t_0 and t_f are the start and end times of the planning interval, and hence t_f can be variable. Note that for clarity, the index for the aircraft is omitted in this development. Substituting Equation 46 into Equation 45 we have:

$$\dot{x}(t) = \frac{2}{t_f - t_0}\sum_{j=1}^{n} C_{i,j}x(j) = \frac{2}{t_f - t_0}g(x_i, u_i) \quad \forall\, i \in \{1, \dots, n\}$$

Equation 47

$C_{i,j}$ can be calculated offline. Using a simple point mass model the aircraft dynamics can be expressed as follows:

$$x(\tau_i) = \begin{bmatrix} r_x \\ r_y \\ h \\ \theta \\ V_{TAS} \end{bmatrix}$$

Equation 48

$$u(\tau_i) = \begin{bmatrix} \Delta h \\ \Delta\theta \\ \Delta T \end{bmatrix}$$

Equation 49

$$\begin{bmatrix} \sum_{j=1}^{n} C_{i,j}r_x(j) \\ \sum_{j=1}^{n} C_{i,j}r_y(j) \\ \sum_{j=1}^{n} C_{i,j}h(j) \\ \sum_{j=1}^{n} C_{i,j}\theta(j) \\ \sum_{j=1}^{n} C_{i,j}V_{TAS}(j) \end{bmatrix} = \frac{t_f - t_0}{2} \begin{bmatrix} V_{TAS}(i)\cos\theta(i) \\ V_{TAS}(i)\sin\theta(i) \\ \Delta h(i) \\ V_{TAS}(i)\Delta\theta(i) \\ \dfrac{\mathrm{T}(i) - D(i)}{m(i)} - \dfrac{g}{V_{TAS}(i)}\Delta h(i) \end{bmatrix}$$

Equation 50

$$\forall\, i \in \{1, \dots, n\}$$

Where: $V_{TAS}(i)$ is the aircraft's true airspeed at point i; θ is its heading; T is the aircrafts thrust acting parallel to the aircraft velocity vector; m is the mass; and D is the drag which is defined as:

$$D = \frac{1}{2}\rho V_{TAS}^2 C_d S \qquad \text{Equation 51}$$

And:

$$C_d = C_{d0} + C_{d2} C_L^2 \qquad \text{Equation 52}$$

$$C_L = \frac{2mg}{\rho V_{TAS}^2 S \cos\phi} \qquad \text{Equation 53}$$

Where ρ is the atmospheric density; S is the wing area; C_{d0} is the parasitic drag coefficient; C_{d2} is the induced drag coefficient and ϕ is the bank angle which is currently assumed to be negligible. This assumption is reasonable for aircraft in cruise but the model could be refined at a later stage. The key feature is that the collocation approach can handle nonlinear dynamics models, enabling more realistic representation of aircraft performance than MILP.

We can now add constraints such as initial position, destination and limits on the states and controls:

$$x(1,k) = x_i(k) \quad \forall \;\; k \,\epsilon \{1, \dots N_x\} \qquad \text{Equation 54}$$

$$x(n,k) = x_f(k) \quad \forall \;\; k \,\epsilon \{1, \dots N_x\} \qquad \text{Equation 55}$$

Where N_x is the number of elements in the state vector; x_i is a vector of the initial states; x_f is a vector of the terminal conditions.

$$x_{min}(k) \le x(i,k) \le x_{max}(k) \quad \forall \;\; i \in \{1, \dots, n\}, k \,\epsilon \{1, \dots N_x\} \qquad \text{Equation 56}$$

$$u_{min}(k) \le u(i,k) \le u_{max}(k) \quad \forall \;\; i \in \{1, \dots, n\}, k \,\epsilon \{1, \dots N_x\} \qquad \text{Equation 57}$$

Where x_{max} and x_{min} are vectors of upper and lower limits respectively on the elements of x; similarly u_{max} and u_{min} limit the control inputs.

4.2 Avoidance of 4-D Obstacles

Section 3 presented a formulation for 3-D obstacle avoidance in the MILP model for aircraft trajectory planning. This section presents an equivalent method for obstacle avoidance using collocation.

In addition to requiring that all aircraft avoid conflicts with each other, there are times where a controller may wish to enforce all aircraft to avoid a region of airspace, e.g. closed sectors or sectors approaching their capacity limits. Closure of airspace is a temporal as well as spatial event which motivates the idea of 4-D obstacle avoidance. Both expected and unforeseen events may cause airspace closures/saturation so this problem is applicable to both the NM and MSC, however this implementation applies to the MSC only.

4.2.1 Review of Polar Sets for Avoidance

Patel and Goulart [4] advanced the idea of using polar sets for obstacle avoidance. The advantage to representing an obstacle in its polar form is that it transforms the constraints into a differentiable function which allows the use of fast gradient based methods to solve the optimization, thus reducing solution times.

A geometric representation of the transformation for a 2-D obstacle is shown in Figure 12 where a convex obstacle in *x-y* is projected into an alternative "reciprocal" space. From the figure we can see that edges of the obstacle are transformed to vertices and that vertices are transformed to edges, this occurs as the polar-obstacle represent the set of hyperplanes that separate the regions within and exterior to the Cartesian obstacle. Thus, there is only one hyperplane that defines an face of the Cartesian obstacle while there are an infinite number of hyperplanes that exist at each vertex.

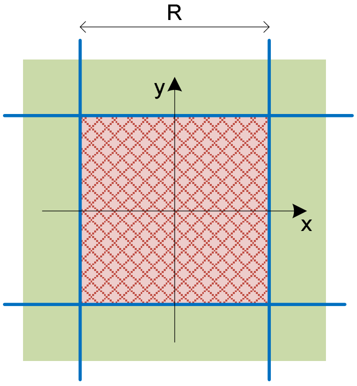
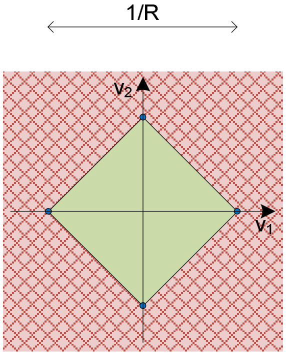
	
(a) Cartesian Obstacle	(b) Polar Obstacle

Figure 12: Polar set representation of an obstacle

Therefore we wish to ensure that a point x is *not* within the obstacle C, and that the equivalent point, y, in the polar-space is within the polar-obstacle C^0, i.e.:

$$x \notin C \Leftrightarrow x^T y \geq 1, y \in C^0 \qquad \text{Equation 58}$$

where we define C^0 to be the polar set of C where $C \subseteq \mathbb{R}^n$ with $0 \in C$:

$$C^0 \triangleq \{v | \langle v, x \rangle \leq 1, \ \ \forall\, x \in C\} \qquad \text{Equation 59}$$

It is important to note that C must contain 0 then if C is closed and convex with $0 \in intC$, then C^0 is compact and convex with $0 \in intC^0$. Furthermore, if the set C is polyhedral with:

$$C = \{x | Hx \leq 1\} \qquad \text{Equation 60}$$

Then:

$$C^0 = \{y | \exists z, y = H^T z, z \geq 0, z^T \mathbf{1} = 1\} \qquad \text{Equation 61}$$

It is important to note that for the above relationship to hold we require the origin to be within the obstacle, C. Therefore, the obstacle avoidance constraints must be formed from the obstacles' perspective.

4.2.2 Polar Sets for 4-D Obstacles

This section advances the novel idea of applying avoidance in four dimensions, i.e. three spatial dimensions plus time. Since the polar set form provides a flexible mechanism for constraining a vector to be outside a convex set, we apply it in four dimensions to avoid an obstacle that occupies a convex region in space for a defined interval in time. It could also be extended to moving obstacles as well, provided the region in 4-D space remained convex. However, since the most likely application in ATM is to handle a fixed spatial volume that closes for a particular time period, that extension has not yet been taken further.

Here we consider obstacle avoidance using polar sets for a single aircraft planning. First we define a set of points, $\boldsymbol{y}_e(t) \in \mathbb{R}^4$, that lies within the polar set of the obstacle defined by the (Cartesian + time) vertices stored in matrix h:

$$\boldsymbol{h}(v)\boldsymbol{y}_e(t) \leq 1\forall \quad v \in \{1..N_v\}, t \in \{1..N_t\}$$ Equation 62

Where N_v is the number of vertices of the obstacle, $\boldsymbol{h}(v)$ is row v of matrix H^T, and N_t is the number of time- points along the trajectory where we enforce the constraints.

Next we ensure that the aircraft remains outside of the obstacle (within the polar set) at each evaluation time point:

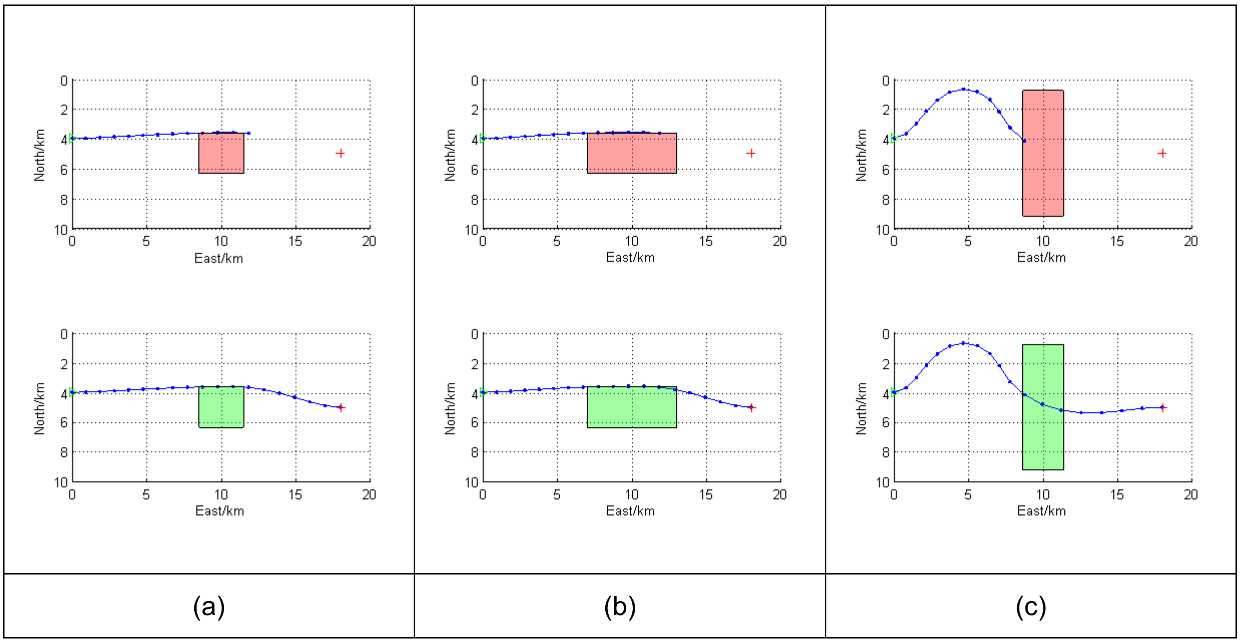
$$\boldsymbol{y}_e(t)^T \begin{pmatrix} \boldsymbol{r}(t) - \boldsymbol{r}_{obs}(t) \\ t - t_{obs} \end{pmatrix} \geq 1 \quad \forall \ t \ \in \{1..N_t\}$$ Equation 63

Where t_{obs} is a time in the middle of the interval during which the region is closed. For stationary obstacles the final term is simply $\boldsymbol{r}_{obs}$.

As the constraints are only enforced at the evaluation points, there is a risk of the trajectory cutting into the obstacle between two adjacent time samples. To mitigate this problem, we also constrain the point y_e to satisfy the same avoidance criteria at the previous timestep:

$$\sum_{d=1}^{4} y_e(d,t)(r(d,t-1) - r_{obs}(d,t-1)) \geq 1 \, \forall \, t \in \{2..N_t\}$$ Equation 64

Applying the above formulation for a single aircraft to a single obstacle we can see the possible effects of the obstacles temporal nature on the aircraft trajectories. Figure 13 shows 3 examples of 4-D obstacle avoidance trajectories, each trajectory is shown projected into the X-Y plane at two time points, the first (upper) plot shows the trajectory during the period that the obstacle must be avoided and the second (lower) plot shows the complete trajectory including the period where the obstacle no longer needs to be avoided. Figure 13a the case where the trajectory passes the obstacle during the initial period and is equivalent to the 3-D avoidance trajectory. Figure 13b shows the case where the trajectory avoids the obstacle during the initial time period but then cuts the corner once the obstacle can be ignored. Figure 13c shows the case where the required deviation around the obstacle is so large that it is “cheaper” to wait until such a time that the trajectory can pass straight through the obstacle; the length of the path prior in the first period is due to the weightings in the cost function favouring a minimum time solution, by constantly accelerating whilst waiting for the obstacle to disappear the time required after that point can then be minimized.



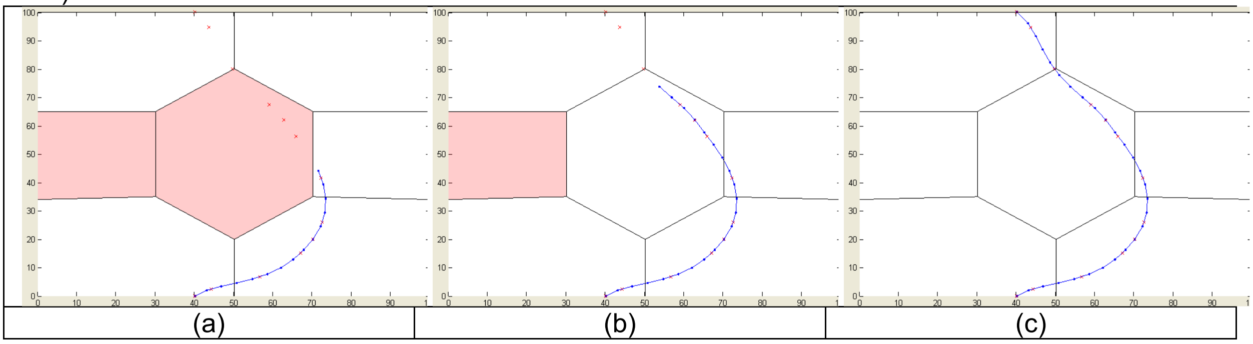
(a)

(b)

(c)

Figure 13: Example 4-D obstacle avoidance trajectories

This concept can be readily extended to multiple obstacles. Defining an arbitrary MSA we can require an aircraft to avoid multiple sectors if, for example, they are closed or reaching their capacity limits. Figure 14 presents an example of multiple 4-D obstacles: Figure 14a shows the initial trajectory as the aircraft avoids two obstacles/sectors, after this point the hexagonal sector is no longer blocked, allowing the trajectory to pass through that sector as shown in Figure 14b. Finally the other sector is also opened (at the point shown in Figure 14b) and the aircraft continues to its destination (Figure 14c).



(a) (b) (c)

Figure 14: Example of multiple 4-D obstacles

4.3 Conflict Resolution Sense Constraints

First we generalize the polar set obstacle avoidance presented in above to multiple vehicles. This is challenging as the collocation form of the problem allows different timescales for different vehicles, i.e. $t_f(a1) \neq t_f(a2)$. The 4-D avoidance form now enables us to enforce avoidance between aircraft without needing to require them all to have a common set of time steps. This leads to a formulation similar to that used for temporal sense constraints using MILP, in which all pairs of samples are required to avoid each other either spatially or temporally. A similar extension to constraint sense of resolution is therefore available and will be investigated later in this section.

The avoidance constraints are developed below, initially just ensuring that all (other) aircraft a_2 avoid the planning aircraft, a_1, at time t_1, i.e. a_1 and t_1 are fixed as illustrated in Figure 15. Fixing a_1 and t_1 is purely to simplify the notation but is easily extended to ensure all aircraft avoid all other aircraft.

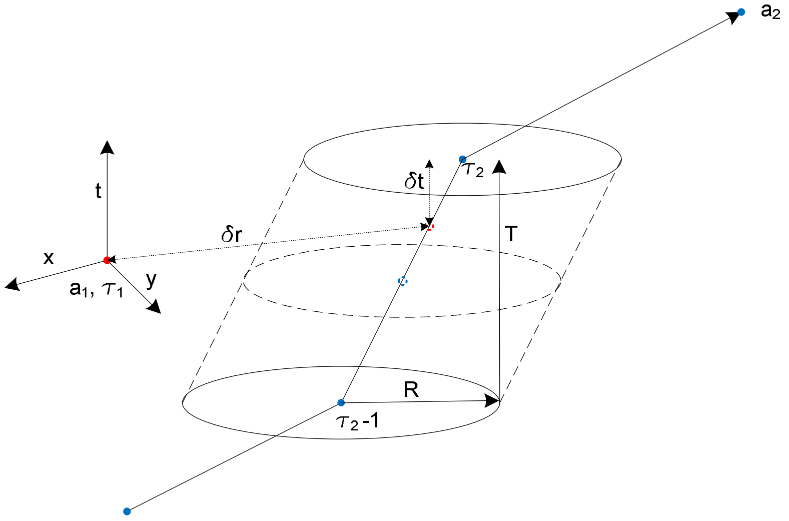


Figure 15: Illustration of vehicular obstacle for multi-vehicle avoidance obstacle

In Figure 15, the origin is shown as the location of a_1, at time τ_1, it should be noted that the vertical direction is time and not a spatial dimension. The trajectory of a_2 is shown as a series of connected (solid blue) nodes and we wish to define an obstacle between τ_2 and $\tau_2 - 1$. The diameter of the cylinder between the two time-points is representative of a 2-D spatial avoidance constraint, i.e. the avoidance distance, R. Similarly, the height of the cylinder, T, represents the period of time that the spatial constraint must be fulfilled. δt is the time of an arbitrary from point τ_1, while T is the total time between τ_2 and $\tau_2 - 1$. Finally, δr is the distance between (a_1, τ_1) and (a_2, τ_1) as projected from τ_2; and δt is the time between τ_1 and τ_2.

To enforce avoidance at time t_1 of the trajectory of aircraft a_1, it is necessary to estimate the position of all other aircraft, $a_2 \epsilon \{1, \ldots N_a\}: a_1 \neq a_2$, at this time, however, as time is discretized separately over each trajectory we do not know which evaluation point in the trajectory of a_2 is closest, forcing us to evaluate the relative position between a_1 at τ_1, and a_2 at $\tau_2 \epsilon \{2, \ldots N_e\}$:

$$\begin{gathered} \delta r(a_2, \tau_2, d) = r(a_2, \tau_1, d) - r_{a_1}(d) \\ \forall\ a_2 \epsilon \{1, \ldots N_a\}, \tau_2 \epsilon \{2, \ldots N_t\}, d \epsilon \{1, \ldots N_d\} \end{gathered}$$ Equation 65

Where $r(a_2, \tau_1, d)$ is the location of a_2 at t_1 in the d^{th} (spatial) dimension and $r_{a_1}(d)$ is the position of a_1 at the same time and dimension. The $r(a_2, \tau_1, d)$ term can be found by interpolating between τ_1, and τ_2:

$$r(a_2, \tau_2, d) = r(a_2, \tau_2 - 1, d) + \Delta r(a_2, \tau_2, d) \Delta t(a_2, \tau_2)$$ Equation 66

Where Δr is the vector between the location of a_2 at $\tau_2$and $\tau_2 - 1$:

$$\Delta r(a_2, \tau_2, d) = r(a_2, \tau_2, d) - r(a_2, \tau_2 - 1, d)$$ Equation 67

And Δt is the normalized time between τ_1 and τ_2:

$$\Delta t(a_2, \tau_2) = \frac{t(\tau_1) - t(\tau_2 - 1, a_2)}{t(\tau_2, a_2) - t(\tau_2 - 1, a_2)}$$ Equation 68

Normalizing time allows us to explicitly include the length of the trajectory (total time) as a variable in the optimization.

If we express the spatial constraint such that we require the normalized spatial distance (as a function of the minimum separation distance) between aircraft to be greater than 1, and similarly that the normalized temporal distance (as a function of the time between τ_2 and τ_2-1) to also be greater than 1, then the constraint is:

$$\begin{bmatrix} \|\delta r\|_2 / R \\ 2\delta t / T \end{bmatrix} \notin [-1, 1]^2$$ Equation 69

And the vertices of our obstacle are:

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}$$ Equation 70

Then using Equation 62 we can define a point, y, that lies in the polar set of H, i.e. $y \epsilon H^0$:

$$H \begin{bmatrix} y(\tau_2, a_2, 1) \\ y(\tau_2, a_2, 2) \end{bmatrix} \leq 1$$ Equation 71

All that remains is to enforce the relationship (from Equation 63) between the point, y, in the polar set and the current state, $\{\delta r, t\}$:

$$\begin{aligned} & y(a_2, \tau_2, 1) \frac{\|\delta r(a_2, \tau_2)\|_2}{R} \\ & + y(a_2, \tau_2, 2) \frac{2\tau_1 - t(\tau_2, a_2) - t(\tau_2 - 1, a_2)}{t(\tau_2, a_2) - t(\tau_2 - 1, a_2)} \end{aligned} \geq 1$$ Equation 72

Finally, we enforce the above conditions for all aircraft relative to a_1 at all time-points along the trajectory of a_2, i.e:

$$\forall\ a_2 \epsilon \{1, \dots N_a\}, \tau_2 \epsilon \{2, \dots N_t\}, d \epsilon \{1, \dots N_d\} : a_1 \neq a_2$$

Equation 73

The above conditions can easily be applied to all aircraft at all time-steps, i.e. a_1 and τ_1 are no longer fixed by:

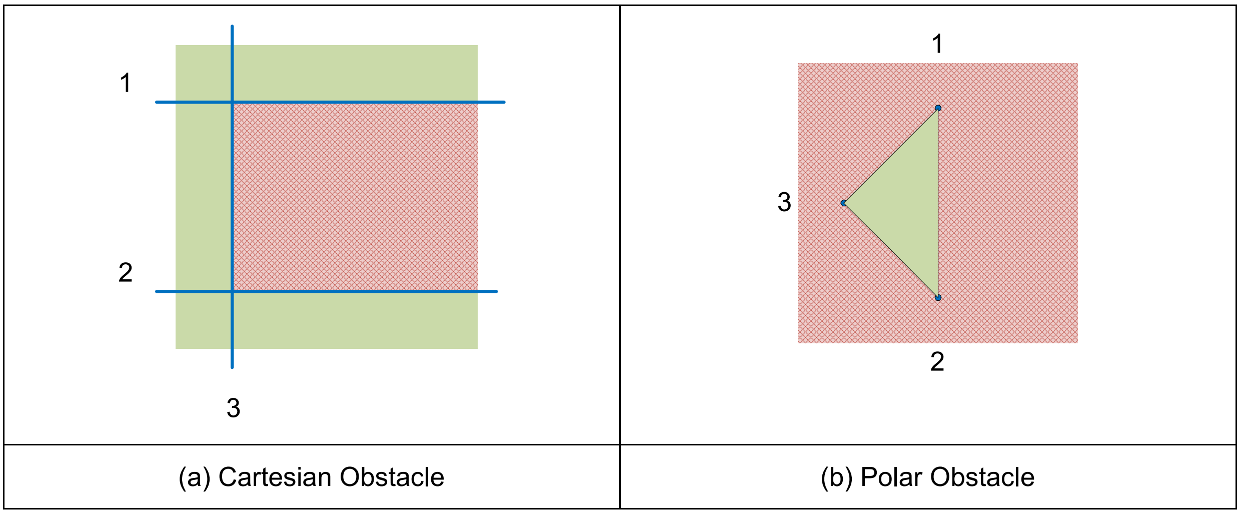
- additionally indexing $r, \delta r, t$ and y over a_1 and τ_1;
- enforcing the constraints over for all a_1 and τ_1 as well as all the previous conditions, i.e.:

$$\forall\ a_1 \epsilon \{1, \dots N_a\}, \tau_1 \epsilon \{2, \dots N_t\}, a_2 \epsilon \{1, \dots N_a\}, \tau_2 \epsilon \{2, \dots N_t\}, d \epsilon \{1, \dots N_d\} : a_1 > a_2$$

Equation 74

4.3.1 Vertical Resolution

Fixing the MILP avoidance binaries is equivalent to removing vertices from the polar set in the collocation formulation. Consider the example in Figure 12 which shows a flat, square obstacle and its polar equivalent. Figure 16 shows that tightening of one constraint, i.e. forcing the trajectory to pass around the remaining 3 sides, is equivalent to removing a vertex from the polar set. This result can be extended to n-dimensions.



(a) Cartesian Obstacle

(b) Polar Obstacle

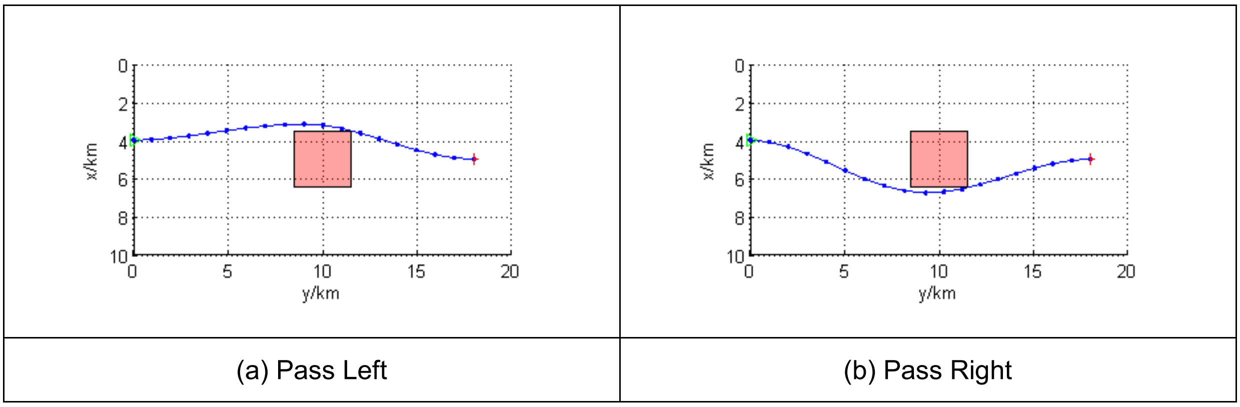
Figure 16: Comparison of MILP and collocation sense constraints

As the polar set of the obstacle is defined as containing the origin, then if we restrict the location of the polar-point of the aircraft's location such that it must lie to one side of the origin, we force the vehicle to pass to that side of the obstacle. An example of constraining the sense of the conflict is presented in Figure 17a and Figure 17b, in this case we have introduced an additional constraint restricting the location of the polar-point to be less than or greater than zero respectively, e.g.:

$$y(\tau_1, 1) \leq 0\ \forall\ \tau_1 \epsilon \{2, \dots N_t\}$$

Equation 75

Where $y(t_1, 1)$ is the coordinate of the point in the polar set that corresponds to the vehicles x-coordinate.



(a) Pass Left

(b) Pass Right

Figure 17: Examples of collocation sense constraints

4.3.2 Temporal Resolution

Constraining one flight to pass ahead or behind another has been implemented in 2-D + time. In addition to the multi-vehicle avoidance criteria defined in Section 4.3, ahead/behind can be implemented by restricting the time-dimension of the point, y, in the polar set in the same manner as used to enforce spatial sense constraints in Section 4.3.1. Equation 76 forces a_1 to pass ahead of a_2.

$$y(\tau_1, a_1, \tau_2, a_2, 2) \geq 0$$

Equation 76

Implying, as before, that ,a_2 cannot resolve its conflict with ,a_1 by occupying the same space but earlier than a1.

It should be noted that this has currently only been implemented in 2D, the extension to 3D would follow an identical approach to that described for MILP in Section 3.2.2.

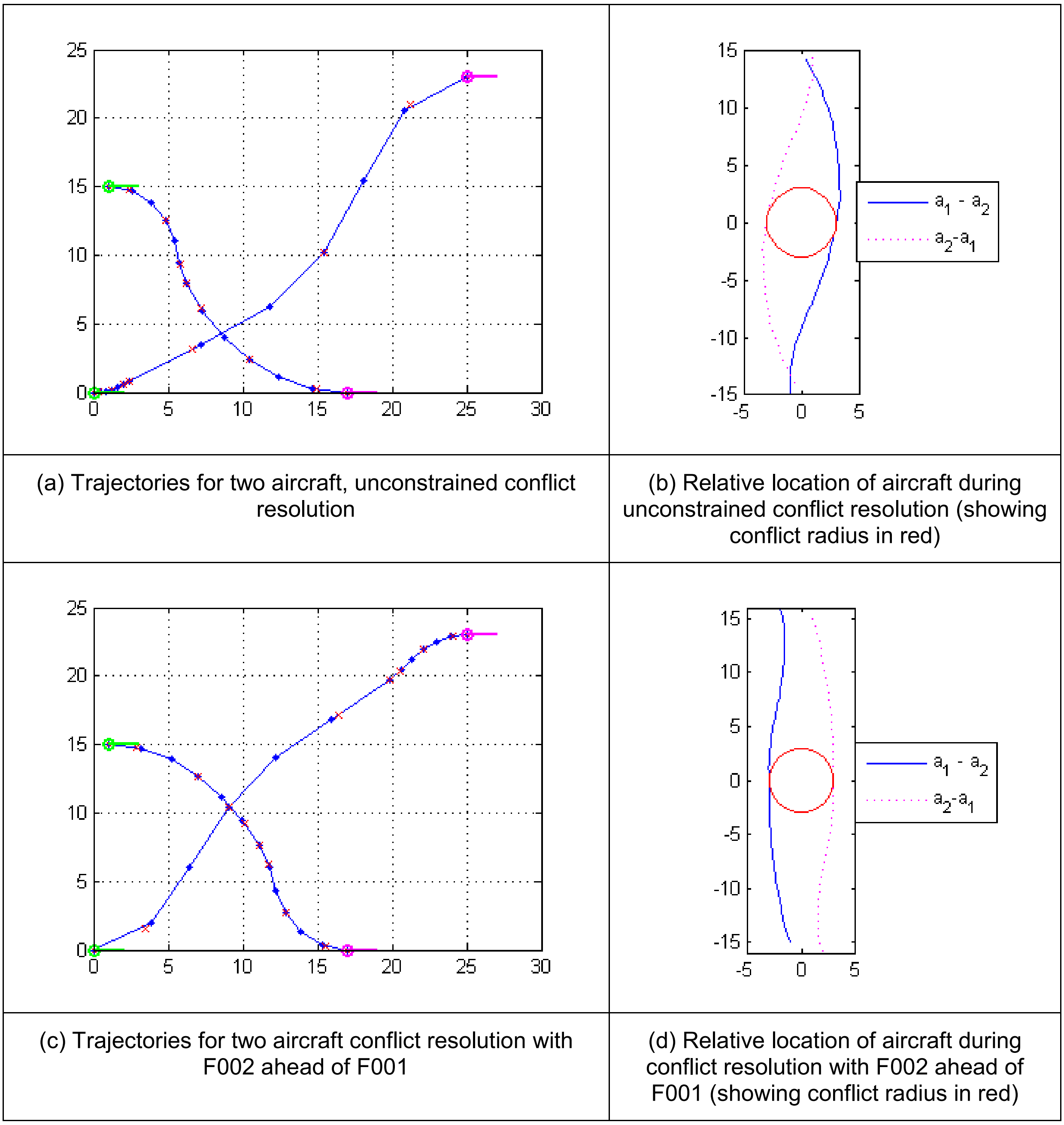
$$y(\tau_1, a_1, \tau_2, a_2, 2) \geq 0$$

Equation 77

Figure 18 shows the effect of sense constraints on a multi-vehicle problem. Figure 18a and Figure 18c show two pairs of trajectories: in the first instance the constraints enforce F001 to pass ahead of F002 and vice versa in the second figure. Figure 18b and Figure 18d show the relative position of each aircraft at the time-steps of the other vehicle, i.e. after the interpolation to the other frame of reference has been performed, as well as indicating the conflict region (red circle). The relative location plots show that, with the exception of the instants discussed below, the trajectories remain conflict free at all times.

It is apparent from Figure 18d that in some instances, as the constraints are only enforced at discrete points, a trajectory is not guaranteed to remain conflict free between time-steps. This was previously solved (in the single aircraft, single obstacle case) by enforcing the avoidance constraints at both the current time-step *and* the previous step in order to prevent corner-cutting. The current multi-vehicle formulation does not permit such a scheme due to the transformation of (x, y) into δr, however this transformation is not essential so a scheme to avoid corner-cutting could be implemented in the future.

The constraints are only enforced relative to one aircraft $(a_1 < a_2)$ in order to reduce the number of variables in the optimization, however, we can see from Figure 18d that F002 plotted relative to F001, does not remain conflict free at all time-steps. This could be resolved by enforcing the constraints in both directions.



(a) Trajectories for two aircraft, unconstrained conflict resolution

(b) Relative location of aircraft during unconstrained conflict resolution (showing conflict radius in red)

(c) Trajectories for two aircraft conflict resolution with F002 ahead of F001

(d) Relative location of aircraft during conflict resolution with F002 ahead of F001 (showing conflict radius in red)

Figure 18: Example collocation sense constraints

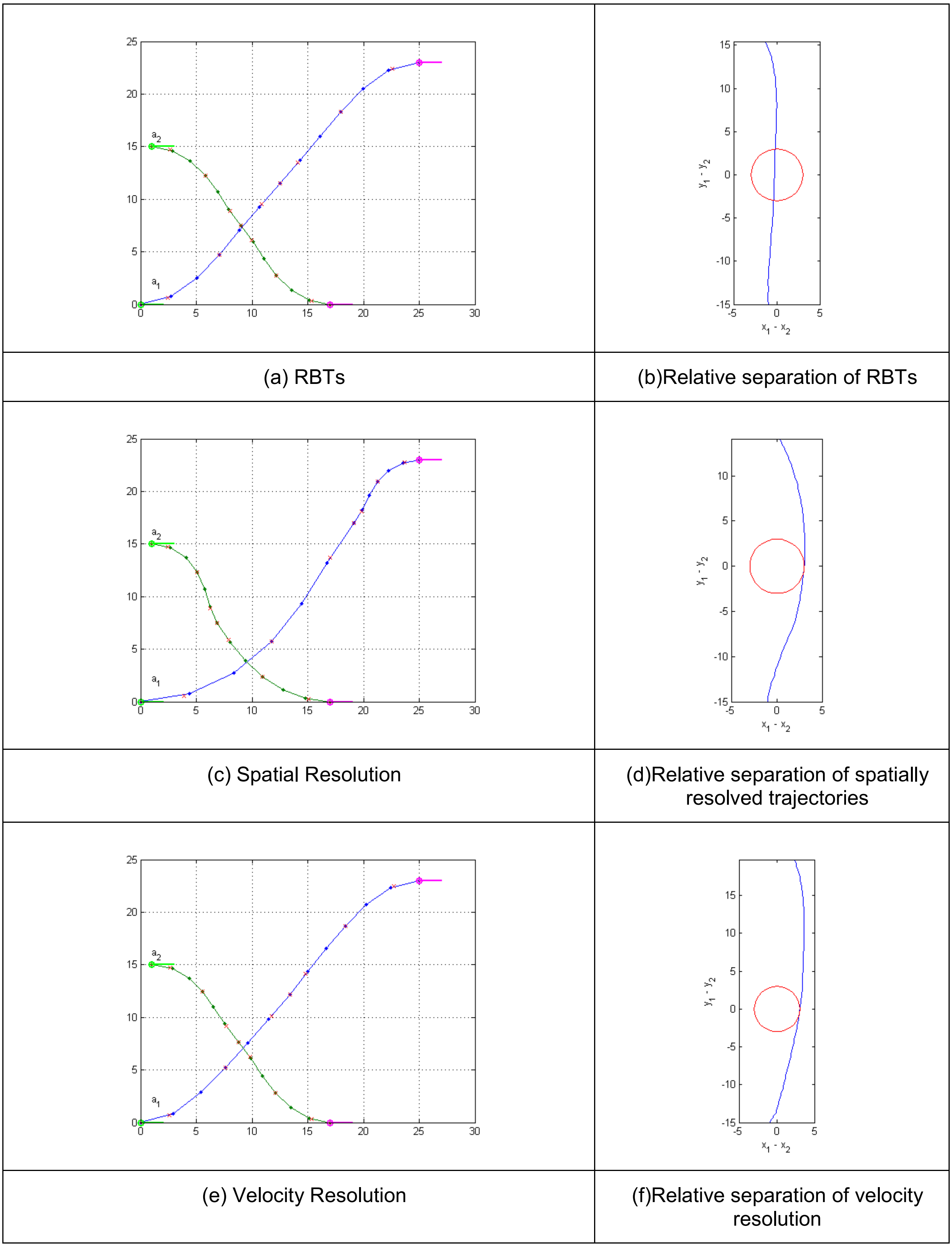
4.3.3 Velocity Resolution

Due to the explicit inclusion of time as a decision variable, implementation of speed advisories in collocation can be achieved simply by fixing the location of the collocation points in all 3 spatial dimensions to force resolution in time (achieved by adjusting the velocities accordingly):

$$x(\tau_i, d) = \boldsymbol{r}^R(a, \tau_i, d) \pm \partial r \qquad \forall\ a \in \{1, \dots, N_a\}, i \in \{1, \dots, N_c\}, d \in \{1, \dots, 3\} \qquad \text{Equation 78}$$

Where the ∂r term is a small tolerance included to ensure the trajectory remains dynamically feasible.

Figure 19 shows a 2D example of velocity resolution using collocation. Figure 19a shows the original RBTs, i.e. the inputs to the optimization and Figure 19b shows the relative position of the two aircraft and it is clear that they pass closer that the avoidance distance indicated by the red circle. Figure 19c and Figure 19d show the conflict resolved with the trajectories resolved with the collocation model of Section 4.3 where it is clear that both aircraft have deviated from their RBTs in order to avoid one another. Figure 19e and Figure 19f show the same conflict resolution but including the constraint in Equation 78 which causes both aircraft to remain on their RBTs but to adjust their velocities to avoid a collision which is as expected.



(a) RBTs

(b)Relative separation of RBTs

(c) Spatial Resolution

(d)Relative separation of spatially resolved trajectories

(e) Velocity Resolution

(f)Relative separation of velocity resolution

Figure 19: Collocation velocity resolution (2D)

A detailed comparison at Figure 19a and Figure 19e shows that the location of the evaluation points have moved in the resolved trajectory and the relative position plots clearly show that the two aircraft now avoid each other. As further evidence of the effect of the additional constraint we can consider the duration of the aircraft trajectories, these are shown in Table 2. Looking at the trajectory durations, it is clear that:

- the spatial resolution results in only a very small increase compared to the RBTs;
- the velocity resolution increases the trajectory durations to a greater extent but that the total increase is less than 10% of the original, conflicting trajectories.

	Trajectory Duration / s		
Conflict resolution	**a_1**	**a_2**	**Total**
None (Figure 19a)	17.90	11.58	29.48
Spatial (Figure 19c)	17.93	11.75	29.68
Velocity (Figure 19e)	17.91	14.46	32.37

Table 2: Trajectory lengths and optimization solution times for results in Figure 19

5 Computational Complexity

It should be noted that an initial assessment of the additional computational complexity imposed by each algorithm is included in the document. However, the complexity comparisons made are based on very limited test runs and a more comprehensive study of the relative performance of the different algorithms remains to be performed. All models were run on an Intel quad-core P4, 3.4GHz PC with 3GB RAM running windows XP; MATLAB® was used for data processing with AMPL as an interface to CPLEX and IPOPT solvers for the MILP and collocation models, respectively.

5.1 Sense Constraints

Table 3 presents some typical solution times for some example two aircraft avoidance problems, with and without sense constraints.

The MILP results show an unusual pattern: by adding additional constraints, the solution time improves. This is due to the way they are implemented by fixing some of the variables thus reducing the size of the problem space. It should also be noted that the addition of the sense constraints (the last two lines of the table) increase the solution time again, due to the extra variables needed for avoidance between different time steps.

There are too few collocation results to draw any firm conclusions. The spread of the times is again likely to be a function of how close the initial guess was to the final solution.

Comparing the MILP and Collocation results it appears that the MILP optimizer is faster, although it should be noted that in the unconstrained case this is not true but also that the collocation method has not yet been extended to 3D.

	Solution Time (s)	
Scenario	MILP (3D)	Collocation (2D + time)
Unconstrained	13.09	9.72
Pass Left (single vehicle)	N/A	2.45
Pass Right (single vehicle)	N/A	1.14
Vertical Resolution	0.34	N/A
F001 over F002	0.39	N/A
F002 over F001	0.19	N/A
Horizontal Resolution	1.19	N/A
Resolve Horizontally with F001 ahead of F002	8.65	1.70
Resolve Horizontally with F002 ahead of F001	2.20	2.46
Velocity Resolution	0.95 (2D)	2.72

Table 3: Example solution times for MILP and Collocation sense constraints

5.2 Avoidance of 4-D Obstacles

The table below presents an initial comparison of the typical solution times for a variety of 4-D obstacle scenarios. As the scenarios are fairly similar (single aircraft, single obstacle) there is little variation in the solution time. The variation that is present is likely to be a function of how close the initial guess was to the final solution.

Scenario	Solution Time (s)
Figure 13a	2.17
Figure 13b	2.30
Figure 13c	1.47

Table 4: Comparison of computational complexity of various 4-D obstacle scenarios

6 Conclusions

A variety of algorithms for supervisor interaction with optimizations have already been demonstrated. The main body of work to date has focussed on enabling MSC (as the supervisor) input as this was the area identified as having the most richness to explore different solutions.

Two different implementations of the MSC model have been implemented. High-level sense constraints to allow either vertical (above/below), temporal (ahead/behind) or a velocity resolution to a conflict have been developed for *both* models. Constraints have been developed for both models to ensure the trajectories remain conflict free between time-steps even when the time-step is large compared to the avoidance distance.

As the collocation method uses an interior point optimizer to solve the problem, there is no guarantee of a globally optimal solution and it has been observed that the method is sensitive to the initial guess and also the number of collocation and evaluation points. Alternative methods, such as potential fields or RRTs, to generate an initial feasible solution could therefore improve the optimizer performance.

7 References

[1] **A. G. Richards,“***Constrainting the sense of Conflict Resolution: Supervision of Route Optimization*”,First SESAR Innovation Days, 2011.

[2] **SUPEROPT,** *“Problem Definition and Literature Review”***,** E.02.01-D1, March 2012.

[3] **BADA,** “*User Manual for the Base of Aircraft Data*”, Revision 3.8, 2010.

[4] **Patel and Goulart,** *“Trajectory Generation for Aircraft Avoidance Manoeuvres Using Online Optimization”*, Journal of Guidance, Control and Dynamics, 2011.

[5] **J. Latombe,** *“Robot Motion Planning”,* Kluwer Academic Publishers, 1991.

[6] **EAD,** http://www.ead.eurocontrol.int, accessed June 2012.

[7] **PHARE**, *“Project Website – Concepts – Multi-Sector Planning”*, visited March 2012, [Online], Available: http://www.eurocontrol.int/phare/public/standard_page/MSP.html

[8] **ADAHR E.02.09**, *“ADAHR: Roles and Responsibilities”*, Edition 0.0.1

[9] **A. J. Eele and A. G. Richards,** *“Path-planning with avoidance using nonlinear branch-and-bound optimization”*, AIAA Journal of Guidance, Control and Dynamics, vol. 32, no. 2, pp. 384–394, March 2009

[10]**B Fornberg,** *“A Practical Guide to Pseudospectral Methods”*, Cambridge University Press, 1996

[11]***Q Gong, F Fahroo, and I Ross,*** “Spectral algorithm for pseudospectral methods in optimal control”, Journal of Guidance, Control and Dynamics, 31(2):460-471, 2008

[12]**N.M.C. de Oliveira & L.T. Biegler,** “Constraint handling and stability properties of model predictive control”, *American Institute of Chemical Engineers' Journal*, 1994, 40, pp.1138-1155

[13]**A. Bemporad & M. Morari**, “Control of Systems integrating Logic Dynamics and Constraints”, *Automatica, 1999, 35(3), pp.407-427*

[14]**Cook, A.** *European air traffic management: principles, practice, and research* Ashgate Pub Co, **2007**

[15]**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

- END OF DOCUMENT -