Constraining the Sense of Conflict Resolution: Supervision of Route Optimization

Arthur Richards
Department of Aerospace Engineering
University of Bristol
Bristol, UK
Email: arthur.richards@bristol.ac.uk

***Abstract*—This paper develops a form of constraints for constraining the *sense* of a conflict resolution within a trajectory optimization. The goal is to enable an intuitive but flexible tool for human supervision, enabling the human to request a particular sense of resolution without conservatively constraining the optimizer. The new constraints are based on the total change in angle of the line joining the two aircraft, which can be uniquely related to one aircraft passing ahead of or behind the other. The method has been implemented with Mixed-Integer Linear Programming as the optimizer and demonstrated in simple scenarios of air traffic control within a sector.**

FOREWORD

This paper describes a project that is part of SESAR Workpackage E, which is addressing long-term and innovative research. The project was started early 2011 so this description is limited to an outline of the project objectives augmented by some early findings.

I. INTRODUCTION

Future Air Traffic Management (ATM) under the SESAR concept will be based on trajectories [1]. As part of the drive toward greater efficiency and capacity, it is natural to optimize these trajectories, and numerical trajectory optimization is a well-researched area. However, numerical optimization is naturally sensitive to its inputs and can be difficult to interact with. For example, the EU ERASMUS research project identified inherent conservatism in human approaches to conflict resolution, motivating a subliminal approach that keeps resolution away from the supervisor [2]. Alternatively, this motivates the "Supervision of Route Optimization" (SUPEROPT) research project, whose goal is to provide meaningful interfaces between humans and trajectory optimizers. One part of SUPEROPT, inspired by the "Playbook" approach to automation [3], is to investigate forms of constraints whose effects are intuitive, such that they can be added as "plays".

This paper reports the development of a very simple initial "play" in which the human can specify the sense of resolution of a conflict. Fig. 1 illustrates the two senses for an example conflict, getting both aircraft from origin to destination, respectively, while maintaining minimum separation. We could loosely refer to sense as the choice of "side", but note that both cases involve 'White' spending some time on 'Black's right hand side. Another option would be to consider the direction of turn, but, taking the anticlockwise case as an example, White turns first right, then left, then right again. Instead, this paper observes that the *total perceived angle change* provides a unique differentiator between the two cases. In one case, the line joining the two aircraft moves anticlockwise; in the other clockwise. Note that it is the total change that matters: on the left, the line is instantaneously moving clockwise at the start and the end, but the total change through the manoeuvre is anticlockwise. This idea is also developed in the theory of robot motion planning, in which it is further observed that there are an infinite number of distinct classes of path [4]. The extra paths are achieved by adding multiples of 2π to the angle change, resulting in one aircraft looping around another. Only the two simplest cases, clockwise and anticlockwise, are considered in this paper.

Note that the development of user interfaces is not in the scope of SUPEROPT: our concerns are the mathematical relations between supervisor inputs and optimizer constraints. The WPE C-SHARE project [5] is investigating ideas for representation of 4D trajectory spaces to a human. Furthermore, SUPEROPT is not limited to interactions with executive air traffic controllers. We will exploit the results of the ADAHR project [6] to investigate interactions with a variety of human stakeholders in the SESAR ATM concept.

The contribution of this paper is to show how a trajectory optimizer can be constrained to ensure clockwise or anticlockwise resolution of a conflict in 2-D. Furthermore, we relate the

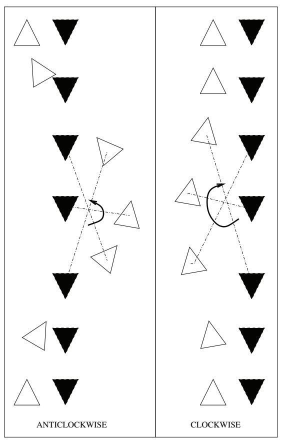


Fig. 1. Illustration of Sense of Conflict Resolution as Angular Change

choice of sense to the more familiar notion of "pass ahead" or "pass behind". For any two aircraft whose paths cross, the first to reach the intersection is said to pass ahead of the other. This decision relates uniquely to the sense of the resolution, and we further provide a tool for determining the sense constraint if the supervisor requests one aircraft to pass ahead of another.

This paper adopts Mixed-Integer Linear Programming (MILP) [7]–[9] to solve the global, non-convex conflict resolution optimization. MILP captures the discrete decision making within the problem – left or right, for example – with binary decision variables. It has been chosen here as it is extensible and, with CPLEX software [10], reliable to solve. Nonlinear optimization has also been proposed for this problem [11]: similar ideas for sense constraints could conceivably be incorporated into such problems, although convergence could be a challenge. Sense constraints could be easily incorporated in a branch-and-bound method [12], although this approach is currently less well-developed for multi-vehicle cases [13]. The method of Hu *et. al.* [14], based on evaluation of different classes of trajectory, could also incorporate sense constraints directly.

The paper begins with a formal problem statement and, as background, the basic MILP formulation in Section II. The main contribution of the sense constraints is developed in Section III, including the relation to passing ahead or behind. Section IV presents results for example scenarios, including investigation of the impact of sense constraints on computation time. Section V draws conclusions and identifies further work.

II. Background and Problem Statement

Define the planned position of aircraft i at time step k as $\mathbf{r}_i(k) = (x_i(k), y_i(k))$. Then for any number of aircraft, the 2-D conflict resolution problem can be expressed as finding trajectories for all aircraft such that, for any pair of aircraft (i, j),

$$\|\mathbf{r}_i(k) - \mathbf{r}_j(k)\| \geq D \ \forall k, \tag{1}$$

where D is the required minimum separation. Since this is a non-convex problem, one approach is to convert it to a choice of linear equations, approximating the separation limit as a square box:

$$\begin{aligned} x_i(k) - x_j(k) &\geq D \text{ or} && (2a)\\ x_j(k) - x_i(k) &\geq D \text{ or} && (2b)\\ y_i(k) - y_j(k) &\geq D \text{ or} && (2c)\\ y_j(k) - y_i(k) &\geq D \ \ \forall k. && (2d) \end{aligned}$$

Then the MILP approach to separation constraints [7], [9] is to use binary variables to encode the discrete decision-making:

$$\begin{aligned} x_i(k) - x_j(k) &\geq D - M b_{ij_1}(k) \text{ and} && (3a)\\ x_j(k) - x_i(k) &\geq D - M b_{ij_2}(k) \text{ and} && (3b)\\ y_i(k) - y_j(k) &\geq D - M b_{ij_3}(k) \text{ and} && (3c)\\ y_j(k) - y_i(k) &\geq D - M b_{ij_4}(k) \text{ and} && (3d)\\ \sum_{p=1}^{4} b_{ij_p} &\leq 3 \ \ \forall k && (3e) \end{aligned}$$

where M is a very large positive value and b_{ij_p}, $p = 1, \ldots, 4$ is a set of binary decision variables. Hence if $b_{ij_p} = 1$, the corresponding separation constraint is effectively removed. The final, logical constraint limits the sum of these binary switches such that at least one of the original constraints is satisfied.

III. Sense Constraints

This section contains the main contribution. The incorporation of sense constraints within MILP recognizes that each "quadrant" ("above", "below", "left", "right") corresponds to a particular binary combination, as illustrated in Figure 2. The angular motion of the relative separation vector is effectively encoded, coarsely, by the changes in the binary settings from step to step. For example, if the binaries change from $(1, 0, 1, 1)$ at one time step to $(1, 1, 0, 1)$ at the next, that represents a clockwise motion.

To constrain the sense of motion, first define a matrix $\mathbf{R}$ of binary settings for each quadrant, such that R_{qp} is the value of binary $b_{ij_p}(k)$ if the vector $\mathbf{r}_{ij}(k) = (x_i(k) - x_j(k), y_i(k) - y_j(k))$ is in quadrant q at time step k. The quadrant numbering can start from anywhere, but must be ordered such that each row repesents the next quadrant around in a clockwise sense from the one above. For example, numbering the quadrants in Figure 2 from left to right gives

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Define new binary variables to capture the choice of move between each time step:

$$m_{ij_q}^{NR}(k) = \begin{cases} 1, & \text{if vector } \mathbf{r}_{ij}(k) \\ & \text{stays in quadrant } q \text{ at step } k \\ 0, & \text{otherwise} \end{cases} \tag{4a}$$

$$m_{ij_q}^{CW}(k) = \begin{cases} 1, & \text{if } \mathbf{r}_{ij}(k) \text{ moves clockwise} \\ & \text{from quadrant } q \text{ at step } k \\ 0, & \text{otherwise} \end{cases} \tag{4b}$$

$$m_{ij_q}^{ACW}(k) = \begin{cases} 1, & \text{if } \mathbf{r}_{ij}(k) \text{ moves anticlockwise} \\ & \text{from quadrant } q \text{ at step } k \\ 0, & \text{otherwise} \end{cases} \tag{4c}$$

and require that at every step $k = 1, \ldots, (N-1)$, exactly one option is chosen,

$$\sum_{q=1}^{4} \left(m_{ij_q}^{NR}(k) + m_{ij_q}^{CW}(k) + m_{ij_q}^{ACW}(k) \right) = 1 \ \forall k. \tag{5}$$

Now the settings of the binary variables at any two adjacent

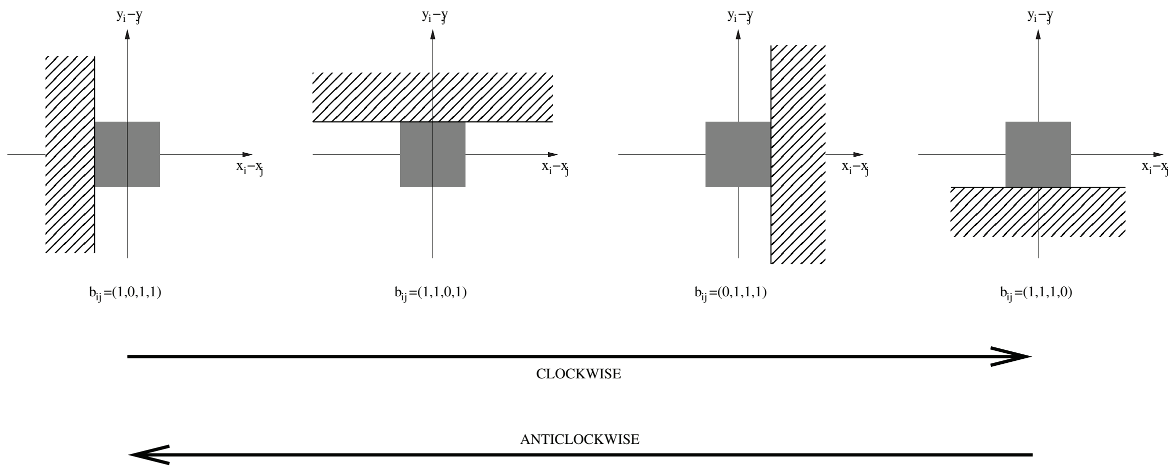


Fig. 2. Relating Sense of Movement to Binary Variables

time steps can be related to the quadrant choices:

$$b_{ij_p}(k) = \sum_{q=1}^{4} R_{qp}\left(m_{ij_q}^{NR}(k) + m_{ij_q}^{CW}(k) + m_{ij_q}^{ACW}(k)\right) \tag{6a}$$

$$b_{ij_p}(k+1) = \sum_{q=1}^{4}\left(R_{qp}m_{ij_q}^{NR}(k) + R_{q^{(+)}p}m_{ij_q}^{CW}(k) + R_{q^{(-)}p}m_{ij_q}^{ACW}(k)\right) \tag{6b}$$

where $q^{(+)} = (q \bmod 4) + 1$ is the next quadrant clockwise from q. Similarly, $q^{(-)} = (q - 2 \bmod 4) + 1$ is the next quadrant anticlockwise from q. So, for example, if at time 10 a clockwise move from quadrant 2 is chosen, then the binaries for time 10 are set to those for quadrant 2 and the binaries for time 11 are set to those for quadrant 3.

Finally, to constrain the sense of the overall motion, a parameter $S_{ij} = \{-1, 1\}$ encodes if the resolution is required to be clockwise (+1) or anticlockwise (-1), by constraining

$$S_{ij}\left(\sum_k \sum_{q=1}^{4}\left(m_{ij_q}^{CW}(k) - m_{ij_q}^{ACW}(k)\right)\right) \geq 1 \tag{7}$$

Hence if $S_{ij} = 1$, this requires $\sum_k \sum_{q=1}^{4} m_{ij_q}^{CW}(k) \geq 1 + \sum_k \sum_{q=1}^{4} m_{ij_q}^{ACW}(k)$, *i.e.* that there has been at least one more clockwise move than there have been anticlockwise moves. This is sufficient to ensure that the sense of the overall movement is clockwise. Similarly, if $S_{ij} = -1$, there must be at least one more anticlockwise move than clockwise.

In summary, the basic separation constraints are those shown in (3). To add a sense constraint between a pair of vehicles, it is necessary to add constraints (5), (6) and (7).

A. Avoiding Corner Cutting

The approach outlined above enforces separation only at discrete time points. This has been observed to cause problems in some cases where the distance traveled in a time step is large compared to the size of the obstacles or separation distances. Maia and Galvão [15] developed a solution to this problem by constraining the binary variables such that it was impossible to go directly from, say $(1, 0, 1, 1)$ to $(1, 1, 0, 1)$ in a single time step. Instead, the transition $(1, 0, 1, 1) \to (1, 0, 0, 1) \to (1, 0, 1, 1)$ would be required, forcing the path to go around the corner. This approach can also be incorporated in the method developed in this paper, by including the intermediate corner states as additional "quadrants" in an enlarged matrix R:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

B. Passing Ahead or Behind

It is potentially more convenient and intuitive to describe the sense of a conflict resolution in terms of one aircraft passing "ahead" or "behind" another aircraft. More precisely, aircraft i passes ahead of j if i reaches the intersection of their paths before j does. If the start and finish points of the two aircraft are known, this decision can be uniquely related to the sense of the conflict resolution, clockwise or anticlockwise.

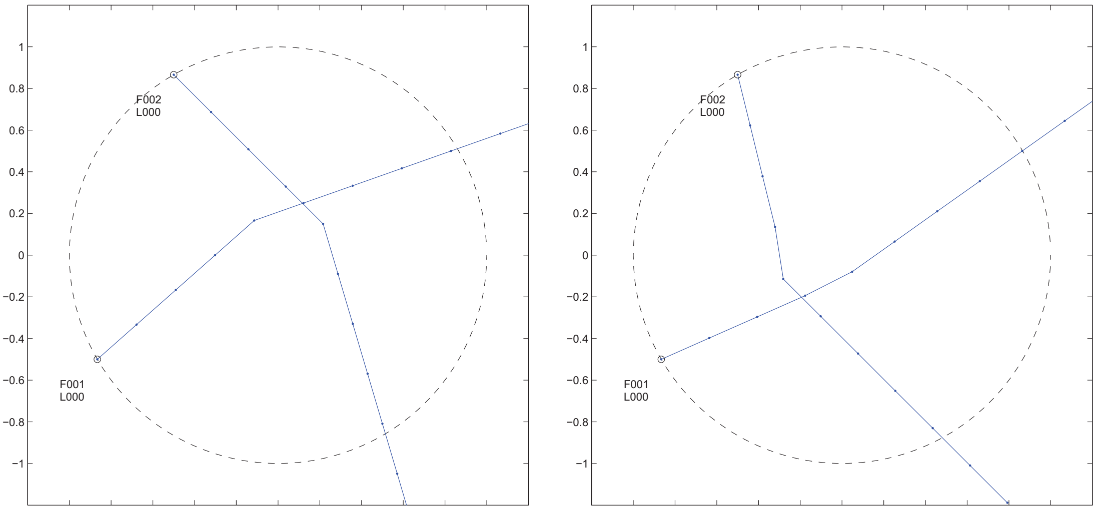
Consider a pair of aircraft i and j whose origins are $\mathbf{s}_i, \mathbf{s}_j$ and destinations are $\mathbf{f}_i, \mathbf{f}_j$, such that the line $(\mathbf{s}_i \to \mathbf{f}_i)$ crosses line $(\mathbf{s}_j \to \mathbf{f}_j)$. Form the 3-D vectors

$$\mathbf{v}_i = \begin{pmatrix} \mathbf{f}_i - \mathbf{s}_i \\ 0 \end{pmatrix}, \quad \mathbf{v}_j = \begin{pmatrix} \mathbf{f}_j - \mathbf{s}_j \\ 0 \end{pmatrix}$$

Then, if i is to pass ahead of j, set $S_{ij} = \text{sign}\left((\mathbf{v}_i \times \mathbf{v}_j)_3\right)$.

IV. Results

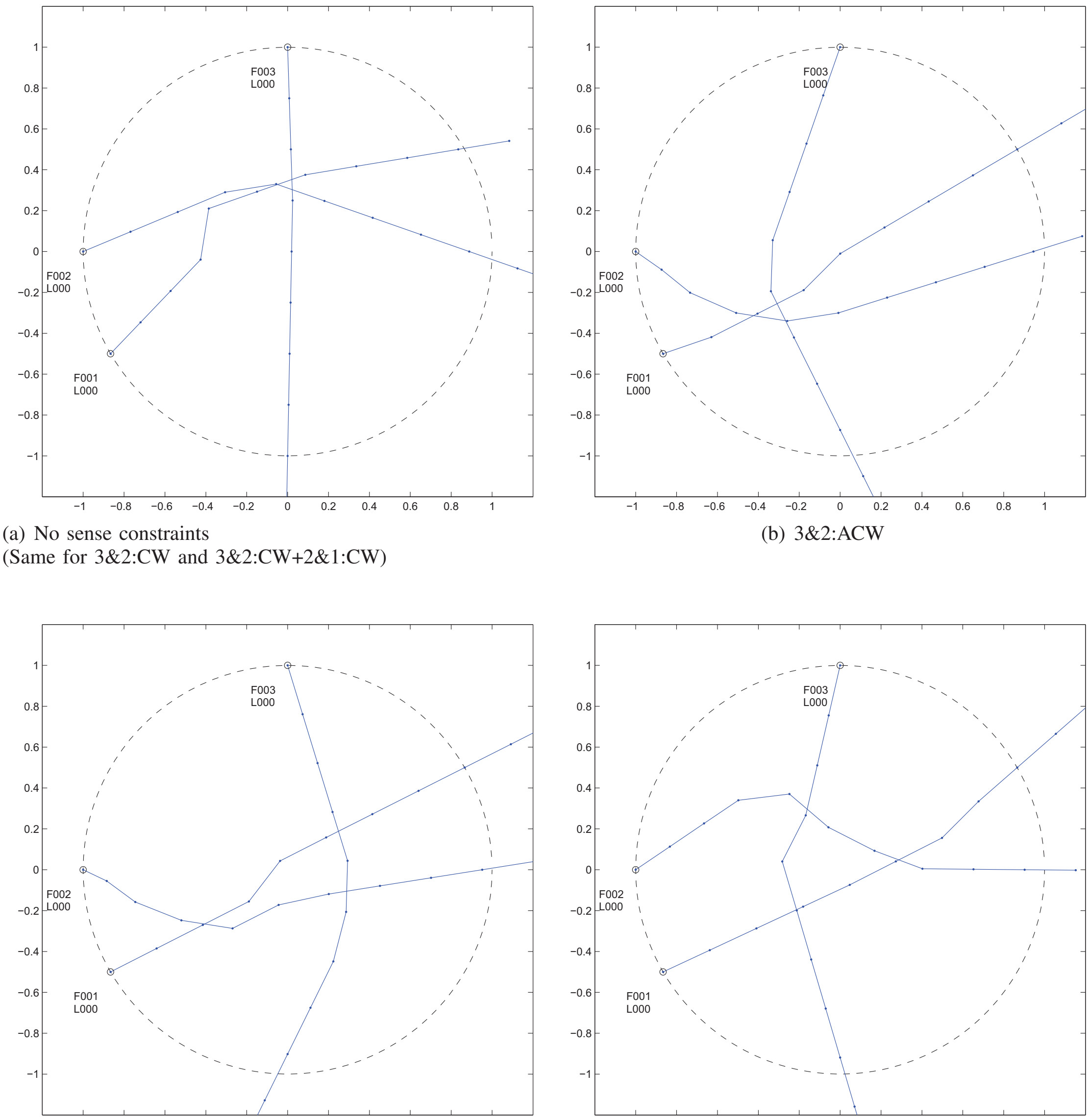
This section illustrates the method in practice, verifying its behaviour in a number of examples. It also includes some investigation of the effect of sense constraints on computation time.



(a) No sense constraints

(b) Constrained to anticlockwise resolution

Fig. 3. Results for Two Aircraft



(a) No sense constraints
(Same for 3&2:CW and 3&2:CW+2&1:CW)

(b) 3&2:ACW

(c) 3&2:CW+2&1:ACW
(Same for 3&2:CW+2&1:ACW+3&1:CW)

(d) 3&2:CW+2&1:ACW+3&1ACW

Fig. 4. Results for Three Aircraft

A. Evaluation Software

Figure 6 shows a screenshot of the evaluation software developed for this project. All examples involve a fictional circular sector with aircraft entering on the left and moving to the right. Buttons enable the user to clear the scenario, step through it, introduce new aircraft entering the sector, and print the results. The right hand panel, with room for expansion, enables interaction with the route optimizer. Current functionality enables the user to choose which aircraft can be re-routed, enter sense constraints directly for any pair of aircraft, or generate sense constraints using the ahead/behind method.

The optimizations all use a linearized model of dynamics developed to capture aircraft behaviour in MILP, as described in Ref [8]. The cost function is primarily the time, evaluated in terms of the number of time steps needed to traverse the sector and summed across all aircraft, plus a small weighting on the acceleration. The latter helps the solution process by avoiding a purely discrete cost function and makes results less erratic.

The software runs within Matlab, calling CPLEX 10.1 software [10] for MILP optimization. The optimization constraints are encoded using the AMPL modeling language [16]. All results in this paper were obtained on a 3GHz Core-2 Duo desktop PC running Windows. Computation times were measured simply using Matlab's "tic" and "toc" functions.

B. Two Aircraft Cases

Figure 3 shows results for a two aircraft problem. The circled points with labels denote the entry positions of each aircraft. The result for the problem without sense constraints is shown in Fig. 3(a). It can be seen that in this case, the resolution is clockwise: the line joining the two aircraft moves clockwise. If a sense constraint for a clockwise resolution is added, the solution predictably does not change. When a sense constraint for anticlockwise resolution was added, the new result was that shown in Fig 3(b), which correctly resolves anticlockwise. Table I shows the computation time results for these cases. Solution times are all short, and no significant difference can be seen between the cases.

TABLE I
COMPUTATION TIMES FOR TWO AIRCRAFT CASES

Sense constraint	Computation time(s)
None	0.60
Clockwise	0.62
Anticlockwise	0.65

C. Three Aircraft Cases

Fig 4 shows a selection of results for the same combination of three aircraft entering. The plots verify that in every case, the sense of resolutions in the output match the constraints where they are applied.

Table II shows the computation times for the various three aircraft cases tried. Predictably, for a combinatorial problem, solution times are considerably longer than for two aircraft. Consistently with the previous results, adding constraints that are already satisfied has minimal effect: for example, Case 3 with added clockwise constraints on aircraft 3 and 2 takes very little longer than Case 1 without those constraints, which returns the same answer. However, adding the opposite 3&2 anticlockwise constraint makes the problem much harder, seen in Case 2.

It is interesting to compare the Cases 5 and 6 in the table, corresponding to Figures 4(c) and 4(d). They differ only by the addition of a constraint that 3 and 1 should resolve anticlockwise, changing the result considerably. However, this additional constraint brings the computation time back down nearly to that of the first case without sense constraints, despite the fact that it introduces more binary decision variables. This is consistent with other results identifying the difficulty of predicting MILP solution time [17].

TABLE II
COMPUTATION TIMES FOR THREE AIRCRAFT CASES

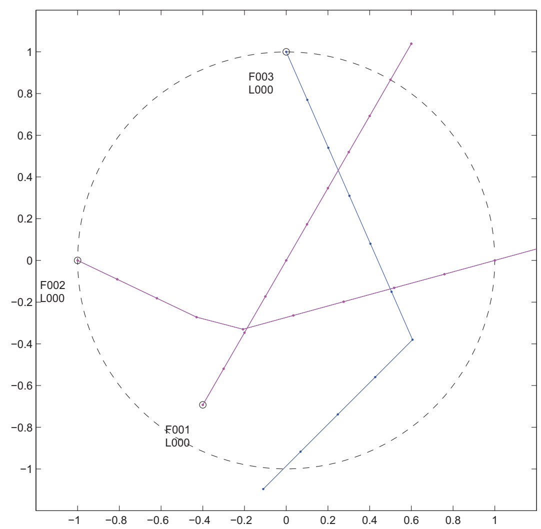
Case	Sense constraints			Computation time(s)
	3 v 2	2 v 1	3 v 1	
1	None	None	None	11.1
2	ACW	None	None	44.9
3	CW	None	None	13.2
4	CW	CW	None	11.7
5	CW	ACW	None	40.6
6	CW	ACW	ACW	12.5
7	CW	ACW	CW	52.2

D. Example of "Ahead" or "Behind"

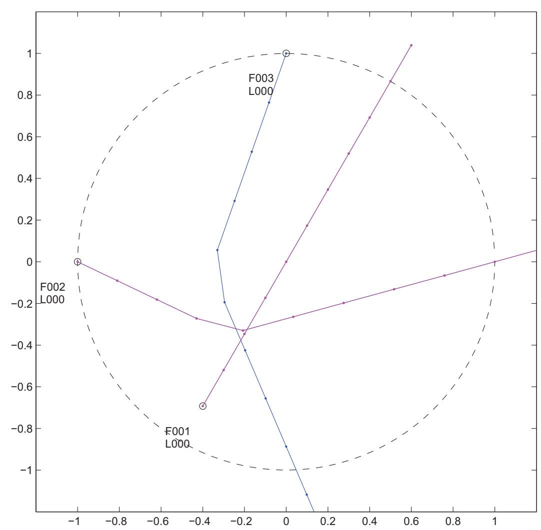
Figure 5 illustrates the use of the "ahead" or "behind" function for a simple three vehicle case. Flights 1 and 2 have already entered the sector and been deconflicted. Their routes have been fixed (indicated by the purple colouring) and will not be altered by the optimizer. Flight 3 enters from the North and is identified as conflicting, as shown in Figure 6. It is first requested to pass ahead of Flight 2. The appropriate sense constraint is generated using the method described in Section III-B and the resulting path is shown in Figure 5(a). Flight 3 can be seen to move to the East, passing ahead of 2 as required, before turning back to its exit point. If the alternative resolution is requested – 3 passes behind 2 – the resulting route is shown in Figure 5(b), and is as expected. Both these examples solved in roughly half a second.

V. Conclusion

This paper has developed a constraint formulation for specifying the sense of a 2D conflict resolution. By using the total angle change as the measurement of the sense, the constraint captures the choice of resolution more effectively than direction of turn or apparent side of the other aircraft. The new formulation enables a supervising human to instruct the optimizer to resolve a conflict in a particular sense, or by specifying one aircraft to pass ahead of another. The new method has been incorporated in MILP optimization and verified by examples in a representative scenario. The effect



(a) 3 to pass ahead of 2



(b) 3 to pass behind 2

Fig. 5. Simulation Results

of adding a sense constraint on computation time has been shown to be highly problem-specific.

Future work will include the development of more "plays" relating to constraint forms, including the extension of the scenario to 3D. Limiting the resolution to small speed changes, inspired by the ERASMUS project's approach [2], is an example of a play to be included. A wider set of supervisor roles, including looking beyond just a single sector, will also be investigated.

Acknowledgment

The work presented in this paper was co-financed by EUROCONTROL on behalf of the SESAR Joint Undertaking in the context of SESAR Work Package E. Any views reflected are the author's own and EUROCONTROL is not liable for any use that may be made of the information contained therein.

The author would like to thank undergraduate student Tom Hacker for deriving and implementing the pass ahead/behind function.

References

[1] SESAR Consortium, "The ATM Target Concept - D3," SESAR Consortium, Tech. Rep., 2007.

[2] R. Weber and E. Cruck, "Subliminal ATC Utilizing 4D Trajectory Negotiation," in *Integrated Communications, Navigation and Surveillance Conference, 2007. ICNS'07*. IEEE, 2007, pp. 1–9.

[3] C. Miller, H. Funk, P. Wu, J. Meisner, and M. Chapman, "The playbook approach to adaptive automation," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 49, no. 1. Human Factors and Ergonomics Society, 2005, pp. 15–19.

[4] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[5] "C-SHARE project blog," visited October 2011. [Online]. Available: http://www.hala-sesar.net/c-share

[6] "ADAHR Project Blog," visited October 2011. [Online]. Available: http://www.hala-sesar.net/adahr

[7] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path-planning," in *European Control Conference*, 2001.

[8] A. G. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of American Control Conference*, 2002.

[9] A. G. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *AIAA Journal of Guidance Control and Dynamics*, vol. 25, no. 4, pp. 755–764, July-August 2002.

[10] "IBM ILOG CPLEX website," visited October 2011. [Online]. Available: http://www.ibm.com/software/integration/optimization/cplex-optimization-studio/

[11] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L. T. Biegler, "MILP and NLP techniques for centralized trajectory planning of multiple unmanned vehicles," in *Proceedings of the American Control Conference*, 2006.

[12] 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.

[13] ——, "Multi vehicle avoidance using nonlinear branch and bound optimisation," in *Proceedings of the AIAA Guidance Navigation and Control Conference*, no. AIAA-2009-5780. AIAA, August 2009.

[14] J. Hu, M. Pradini, and S. Sastry, "Three dimensional optimal coordinated maneuvers for aircraft conflict avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, pp. 888–900, 2002.

[15] M. H. Maia and R. K. H. Galvao, "On the use of mixed-integer linear programming for predictive control with avoidance constraints," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 7, pp. 822–828, 2009.

[16] AMPL Website, http://www.ampl.com/, visited July 2005. [Online]. Available: http://www.ampl.com/

[17] D. Bertsimas, G. Lulli, and A. Odoni, "An integer optimization approach to large-scale air traffic flow management," *Operations research*, vol. 59, no. 1, p. 211, 2011.

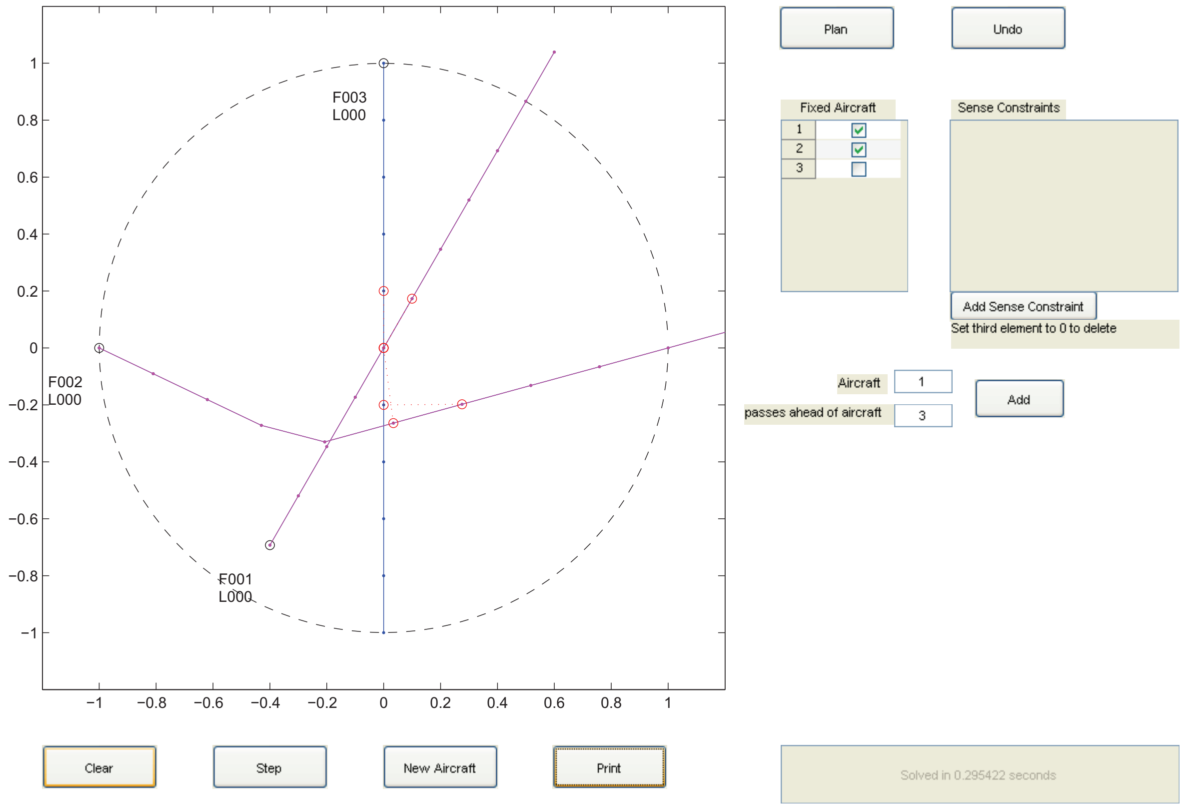


Fig. 6. Evaluation Platform User Interface