# Analysis of a Model Predictive Control Mixed Integer Linear Program Model for Air Traffic Management

**Logan Wolf**
**Computer Science Dept., William & Mary**
**Williamsburg, VA 23187-8795**
**lbwolf@wm.edu**

**Rex K. Kincaid**
**Mathematics Dept., William & Mary**
**Williamsburg, VA 23187-8795**
**rrkinc@wm.edu**

## ABSTRACT

Air traffic control, to avoid aircraft collisions and reach their destinations efficiently, is a complex problem. Centralized control and distributed control are the two most common strategies for managing air traffic. To fully understand the key differences between the two concepts, we focus on the control problem abstractly, as an optimization problem. Mixed integer linear programs (MILPs) are used to model the control strategies. The performance of the MILPs are compared on simulated data for up to 30 planes needing to resolve conflicts in a 2-dimensional airspace. Results indicate the strengths and weaknesses of each concept rather than an individual algorithmic approach.

## ABOUT

**Rex K. Kincaid** is Chancellor Professor of Mathematics at William & Mary. His B.A. in Mathematics is from DePauw University while his M.S. degree in Applied Mathematics and Ph.D. in Operations Research are from Purdue University. His research interests include network location theory and metaheuristics for discrete optimization problems. He has published more than 90 peer-reviewed research articles with more than 40 undergraduate and M.S. degree students as co-authors.

**Logan Wolf** completed his undergraduate mathematics major at William & Mary in May of 2023. He will graduate with an M.S. in Computer Science with a specialization in Computational Operations Research in May of 2024. He is primarily interested in optimization within the transportation industry.

# Analysis of a Model Predictive Control Mixed Integer Linear Program Model for Air Traffic Management

**Logan Wolf**
**Computer Science Dept., William & Mary**
**Williamsburg, VA 23187-8795**

**Rex K. Kincaid**
**Dept. of Mathematics, William & Mary**
**Williamsburg, VA 23187-8795**

## INTRODUCTION

Air traffic management systems address all airborne and ground-based activities of aircraft (e.g., taxiing, takeoff and landing, air traffic control and airspace flow and capacity decisions). The focus here is on air traffic control and, in particular, collision avoidance and efficient route planning. Two competing strategies, centralized control and distributed control, are key components of air traffic control for air traffic management systems. Fundamentally, centralized control involves a single authority collecting and processing information while distributed control involves each aircraft collecting and processing information and exercising trajectory decisions autonomously. Proponents of centralized control emphasize that it allows strong system stability, e.g., a small change in an aircraft's trajectory due to weather would not cause a cascade of changes throughout the entire system. On the other hand, proponents of distributed control emphasize greater flexibility in trajectory decisions.

Air traffic management requires aircraft path-planning that address multiple aircraft, collision, and obstacle avoidance. Several researchers have developed optimization models (Richards and How, 2002; Cai and Zhang, 2018), as potential solution strategies. Our goal here is to develop an optimization model for both the centralized and distributed control paradigms and evaluate their performance. The results are applicable to both commercial aircraft traffic planning as well as UAV (Unmanned Automated Vehicles) trajectory planning (Xue, 2020).

Trajectory optimization with collision avoidance constraints can be written as a mixed integer linear program (MILP) to minimize a cost function regarding flight time(s) (Cai and Zhang, 2018). Centralized control involves a single authority collecting and processing information (e.g., an air traffic controller), while distributed control involves each plane collecting and processing information and exercising trajectory decisions autonomously (i.e., free flight).

## MILP MODEL 1

A MILP is formulated for an aircraft trajectory optimization problem at a specific altitude in the (2D) airspace. Trajectories are generated randomly and discretized over $t = 1, ..., T$ time steps for each aircraft $p$. That is $(trajx_{p,0}, trajy_{p,0}), \ldots .(trajx_{p,T}, trajy_{p,T})(trajx_{p,0}, trajy_{p,0}), \ldots .(trajx_{p,T}, trajy_{p,T})$ are the coordinates for the p = 1, ..., N planes and t = 1, ..., T time steps. Aircraft must start at their initial trajectory positions and travel within their cone of direction a[p,0]. Each aircraft $p$'s objective is to minimize the deviation from its projected trajectory, subject to velocity, separation, and movement direction constraints. The objective function penalizes aircraft for deviating from their planned trajectories. To remain a linear programming problem, the objective is formulated with the $L_1$ norm. The problem is to minimize the sum of the differences in the X and Y directions for every aircraft in each time step (see Kincaid, Curtis, and Wolf, 2023). The constraints indicated below are for all $p, q \in [1, \ldots, N]$ and for all $t \in [0, \ldots, T-1]$. The subscripts $p$ and $q$ are aircraft indices.

Equation (1) specifies the initial conditions for each aircraft along its projected trajectory. Equations (2) and (3) impose minimum and maximum aircraft velocities, while (4) enforces a minimum separation distance between each aircraft. Equations (5) and (6) enforce a direction cone of 45 degrees ($\pi/4$) for a change in an aircraft's heading. The notation a[p,t] denotes the current direction (angle with respect to horizontal) for plane $p$ at time $t$.

$$min \sum_{t=0}^{T} \sum_{p=1}^{N} \left( \left| x_{p,t} - traj\, x_{p,t} \right| + \left| y_{p,t} - traj\, y_{p,t} \right| \right)$$

$$s.t. \quad x_{p,0} = traj\, x_{p,0} \; and \; y_{p,0} = traj\, y_{p,0} \quad (1)$$

$$\left| x_{p,t+1} - x_{p,t} \right| + \left| y_{p,t+1} - y_{p,t} \right| \geq v_{\min} \quad (2)$$

$$\left| x_{p,t+1} - x_{p,t} \right| + \left| y_{p,t+1} - y_{p,t} \right| \leq v_{max} \quad (3)$$

$$\left| x_{p,t} - x_{q,t} \right| + \left| y_{p,t} - y_{q,t} \right| \geq \min s\, ep \quad (4)$$

$$\cos\left(a[p,t] + \frac{\pi}{4}\right) \cdot \left(x_{p,t+1} - x_{p,t}\right) + \sin\left(a[p,t] + \frac{\pi}{4}\right) \cdot \left(y_{p,t+1} - y_{p,t}\right) \geq 0 \quad (5)$$

$$\cos\left(a[p,t] - \frac{\pi}{4}\right) \cdot \left(x_{p,t+1} - x_{p,t}\right) + \sin\left(a[p,t] - \frac{\pi}{4}\right) \cdot \left(y_{p,t+1} - y_{p,t}\right) \geq 0 \quad (6)$$

## MILP 2: MODEL PREDICTIVE CONTROL MODE

The second MILP model is based on (Richards and How, 2002) who utilize a model predictive control approach (also known as a receding time horizon approach). Unlike the previous model in which each aircraft's shortest route trajectory is known, MILP 2 takes only the start and finish location for each aircraft as input. Each of the N aircraft is represented as a point mass. The position of aircraft $p$ at time step $i$ is given by $\left(x_{ip}, y_{ip}\right)$. Each aircraft is acted upon by control forces $\left(f_{x_{ip}}, f_{y_{ip}}\right)$. The state vector $s_{ip}$ is composed of the aircraft position and velocity. Equation (8) enforces the initial conditions and equation (7) describes the dynamics of the system for each aircraft at each time step. Equations (9) and (10) limit the maximum force and velocity for the point mass $m$ (aircraft). The $M$ in the denominator denotes the number of sides of a polygon used to approximate the Euclidean norm. The final set of equations (11) enforce a rectangular collision avoidance region for each aircraft at each time step. Let $d$ be the safety distance between two aircraft $p$ and $q$. The $c_{ipqk}$ are a set of $k=1 \ldots 4$ binary variables associated with each pair of aircraft for each time step $i$ and $R$ is a large positive number. If $c_{ipqk} = 0$ the inequality holds, otherwise it is relaxed. The last inequality in the set (11) forces at least one of the inequalities in (11) to hold. Additional binary variables $b_{ip}$ are defined such that $b_{ip}$ is 1 if aircraft $p$ arrives at its destination at time $i$. The set (12) enforces that each aircraft arrives at its destination at exactly one time step, after which point that aircraft leaves the system. The objective $\sum_{i=1}^{T} \sum_{p=1}^{N} \left(T_i\, b_{ip}\right)$ is to minimize the sum of the finishing times for each aircraft in each time step. However, as is noted by (Richards and How, (2002)) a small penalty term added to the objective reduces the solve time and prioritizes smoother solutions. The value of $\epsilon > 0$ controls the size of the penalty.

$$\min_{s,u,b,c} J = \sum_{i=1}^{T} \sum_{p=1}^{N} \{T_i b_{ip} + \epsilon(|f_{x_{ip}}| + |f_{y_{ip}}|)\}$$

$$\mathrm{s}_{(i+1)p} = \mathbf{A}_p \mathrm{s}_{ip} + \mathbf{B}_p \mathrm{f}_{ip} \quad (7)$$

$$\mathrm{s}_{0_p} = \mathrm{s}_{I_p} \quad (8)$$

$$f_{x_{ip}} \sin\left(\frac{2\pi m}{M}\right) + f_{y_{ip}} \cos\left(\frac{2\pi m}{M}\right) \leq f_{\max} \quad (9)$$

$$v_{x_{ip}} \sin\left(\frac{2\pi m}{M}\right) + v_{y_{ip}} \cos\left(\frac{2\pi m}{M}\right) \leq v_{\max} \quad (10)$$

$$
\begin{aligned}
&x_{ip} - x_{iq} \geq d - Rc_{ipq1} \\
&x_{iq} - x_{ip} \geq d - Rc_{ipq2} \\
&y_{ip} - y_{iq} \geq d - Rc_{ipq3} \\
&y_{iq} - y_{ip} \geq d - Rc_{ipq4} \\
&\text{and } \sum_{k=1}^{4} c_{ipqk} \leq 3
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
&x_{ip} - x_{F_p} \leq R(1 - b_{ip}) \\
&x_{ip} - x_{F_p} \geq -R(1 - b_{ip}) \\
&y_{ip} - y_{F_p} \leq R\left(1 - b_{ip}\right) \\
&y_{ip} - y_{F_p} \geq -R(1 - b_{ip}) \\
&\text{and } \sum_{i=1}^{T} b_{ip} = 1
\end{aligned}
\tag{12}
$$

## SOLVING CENTRALIZED AND DISTRIBUTED MODELS

The above MILP models can be employed with either a centralized or distributed approach. In both cases, it is assumed that all initial trajectories are known at the beginning of the time window. In the centralized approach, the model deconflicts all aircraft simultaneously. The decision variables consist of the (x, y) coordinates (and velocity components in the case of MILP 2) for every aircraft for each of the 80 time steps, as well as all supporting binary variables. (A 20 minute time window was chosen for these computational experiments. The 20 minute window was subdivided into 80 15 second intervals at which route and velocity changes may be made. Other time windows and subdivisions are easily implemented in any of these models.) Alternatively, in the distributed approach, the optimization of each aircraft's trajectory is treated as a separate iteration of the MILP model. For each iteration, one aircraft is selected to be *active*, and this aircraft's trajectory is optimized while holding all other aircraft trajectories fixed. After an *active* aircraft $p$ finds its optimal trajectory, $p$'s default trajectory is updated to the new one and the process repeats until there are no conflicts. In the distributed approach, the decision variables describe the trajectory of just the active aircraft, and the trajectories of all inactive aircraft are instead treated as parameters. The order in which the aircraft are deemed *active* is an exogenous decision. One approach is to identify the aircraft with the most conflicts in the given time window. Outlines for the two approaches are given below.

### Centralized Approach

1. Generate an instance with a Python program.
2. Solve the MILP (trajectory.mod and trajectory.dat) with AMPL/Gurobi.

### Distributed Approach

1. Generate an instance with the Python program.
2. Determine an aircraft, $p$, with the maximum number of conflicts over the 80-step time window.
3. If there are no aircraft conflicts, then STOP.
4. Else, solve the MILP with AMPL/Gurobi for aircraft $p$ (all other aircraft trajectories unchanged).
   Solution deconflicts $p$.
5. Update deconflicted aircraft $p$'s trajectory.
6. Go to step 2.

## DISTRIBUTED MODEL LOCALIZATION MODIFICATION

Experiments have shown that MILP models following a centralized approach to air traffic management can find better solutions but at a significant computation cost (Kincaid, Curtis and Wolf, 2023). In contrast, a distributed approach generally decreases the computational cost, but with a moderate increase in the performance metric. In a distributed approach, selecting a single (active) aircraft to deconflict and keeping the trajectories of all other aircraft constant reduces the number of decision variables dramatically, allowing for a significantly faster solve time. With the goal of further improving the computational performance of a distributed approach, then, a natural idea is to reduce the number of constraints in each iteration of the MILP. As a result, both the number of time steps and the number of aircraft included in each iteration of the MILP can be reduced. This is accomplished with the introduction of a detection distance parameter for each aircraft. Unlike the centralized and distributed model, in which all aircraft know the trajectory of all other aircraft, this modification imposes a restriction that an aircraft only knows the position and trajectory of aircraft within its detection radius.

Like the distributed approach, a Python script randomly generates position and trajectory data for all the aircraft. Unlike the distributed model, which resolves all conflicts consecutively, this modification functions more like a time-based simulation. Using the detection distance parameter and aircraft trajectory information, the Python program starts at time 0 and loops through all aircraft in an exogenously determined order. For each aircraft, we check the trajectory information of any aircraft currently within the detection radius to determine if there is an upcoming conflict. If such a conflict is detected, an AMPL data file is generated which includes only the currently detected aircraft. This approach is a modification of the distributed model, in which the MILP determines the optimal trajectory of the currently detected aircraft. The Python script then moves to the next aircraft, in order, and when all detected conflicts are resolved at this time step, the process moves on to the next time step. An outline of the algorithm is as follows:

### Modified Distributed Approach

1. Generate an instance with the Python program.
2. Begin at time 0 and select the active aircraft $p$ to be the first aircraft in the pre-determined aircraft list L.
3. Determine if $p$ has any future conflicts with other aircraft currently within the detection radius.
4. If there are no conflicts, go to step 7.
5. Else, solve the MILP for aircraft $p$ (all other aircraft trajectories unchanged). Solution deconflicts $p$.
6. Update deconflicted aircraft $p$'s trajectory.
7. Set $p$ to be the next aircraft in list L. If $p$ is the last in list L, increment the time step.
8. If the final time step is complete, then STOP. Else, go to step 3.

In this version of the model, the assumption that all aircraft trajectories are known by each aircraft or by a centralized entity is not required. This change makes the model sequential, with new trajectories calculated as new information is known or as additional conflicts arise. In the original distributed model, each instance of the MILP included information for aircraft that will never come near the current (active) aircraft, and since the number of constraints grows according to the product of the number of aircraft and the number of time steps, this redundancy quickly becomes computationally expensive. Thus, the detection distance parameter is used to determine which aircraft are relevant in each MILP subproblem. Furthermore, since the AMPL data file is only formulated at the time step where a conflict is detected, the earlier time steps in which no conflicts occur are effectively eliminated from the MILP, further reducing the size of each subproblem.

**RESULTS**

The localization modification when applied to MILP 1 has been shown to decrease solve time at the cost of a larger total trajectory deviation (Kincaid, Curtis, and Wolf, 2023). This section's goals are to evaluate this modification's performance when applied to MILP 2, and then compare the performance of the two modified models. In all cases, aircraft starting locations and route trajectories are randomly generated by a Python program in a 200 n.m. by 200 n.m. (nautical miles) two-dimensional airspace for a twenty-minute time window (80 time steps of 15 seconds) for each aircraft. A minimum speed ($v_{min}$) of 1.75 n.m. and a maximum speed ($v_{max}$) of 2.5 n.m. for each 15 second time step was enforced. (The result is a minimum speed of 420 n.m./hr. and a maximum speed of 600 n.m./hr.) MILP 1 requires a default speed of 2.25 n.m. to calculate discrete points for the initial trajectory of each aircraft. Meanwhile, MILP 2 generates initial trajectories by solving instances of itself for each aircraft independently and without the collision avoidance constraints. These initial trajectories are required for the collision detection logic in the localization modification. A minimum separation value of 5 n.m. between aircraft was required. All aircraft characteristics are identical with an arbitrarily chosen mass of 1 and maximum force ($f_{\max}$) value of 2.5 N.

All computational experiments were run on William & Mary's high performance computing network. All experiments were executed on a single node. The node is a Dell PowerEdge R630 with 2x10-core Intel Xeon ES-2640 v4 processor with a clock speed of 2.4 GHz and 128 GB of memory.

**Table 1. Results for MILP 2 before and after the localization modification**

| | Unmodified MILP 2 | | | Modified MILP 2 | | |
|---|---|---|---|---|---|---|
| #Aircraft | Route Changes | Objective Value (# time steps) | Runtime (seconds) | Route Changes | Objective Value (# time steps) | Runtime (seconds) |
| 20 | 5 | 2 | 187.1 | 7 | 6 | 38.2 |
| | 5 | 2 | 174.4 | 5 | 4 | 30.7 |
| | 2 | 1 | 47.2 | 2 | 1 | 4.4 |
| | 4 | 2 | 87.1 | 4 | 3 | 22.9 |
| | 2 | 4 | 58.7 | 3 | 2 | 16.2 |
| 30 | 4 | 2 | 246.3 | 6 | 8 | 32.8 |
| | 7 | 3 | 271.5 | 9 | 3 | 48.8 |
| | 7 | 4 | 378.5 | 9 | 9 | 55.4 |
| | 10 | 11 | 412.7 | 12 | 11 | 64.2 |
| | 6 | 5 | 377.0 | 12 | 10 | 38.8 |

Table 1 gives a comparison of the distributed MILP 2 before and after the subproblem modification. Here, "route changes" represents the number of runs of the MILP model required to deconflict all aircraft. The objective value is given as the total delay, meaning the summed difference between the computed finishing times of each aircraft and the best possible finishing times had there been no trajectory conflicts.

By design, the modified MILP 2 solutions have more route changes, as each aircraft can only account for the conflicts it detects, and undetected future conflicts require another route change upon their discovery later in the simulation. Despite the increased number of route changes, the modification allows for a substantial decrease in solve time in every case by decreasing the number of aircraft involved in each iteration of the model. As expected, this leads to a worse objective value in many cases, as the information available to each aircraft is restricted. In essence, the localization modification exhibits the same benefits and drawbacks as the switch from a centralized model to a distributed model: faster solve times at the cost of a worse objective value.

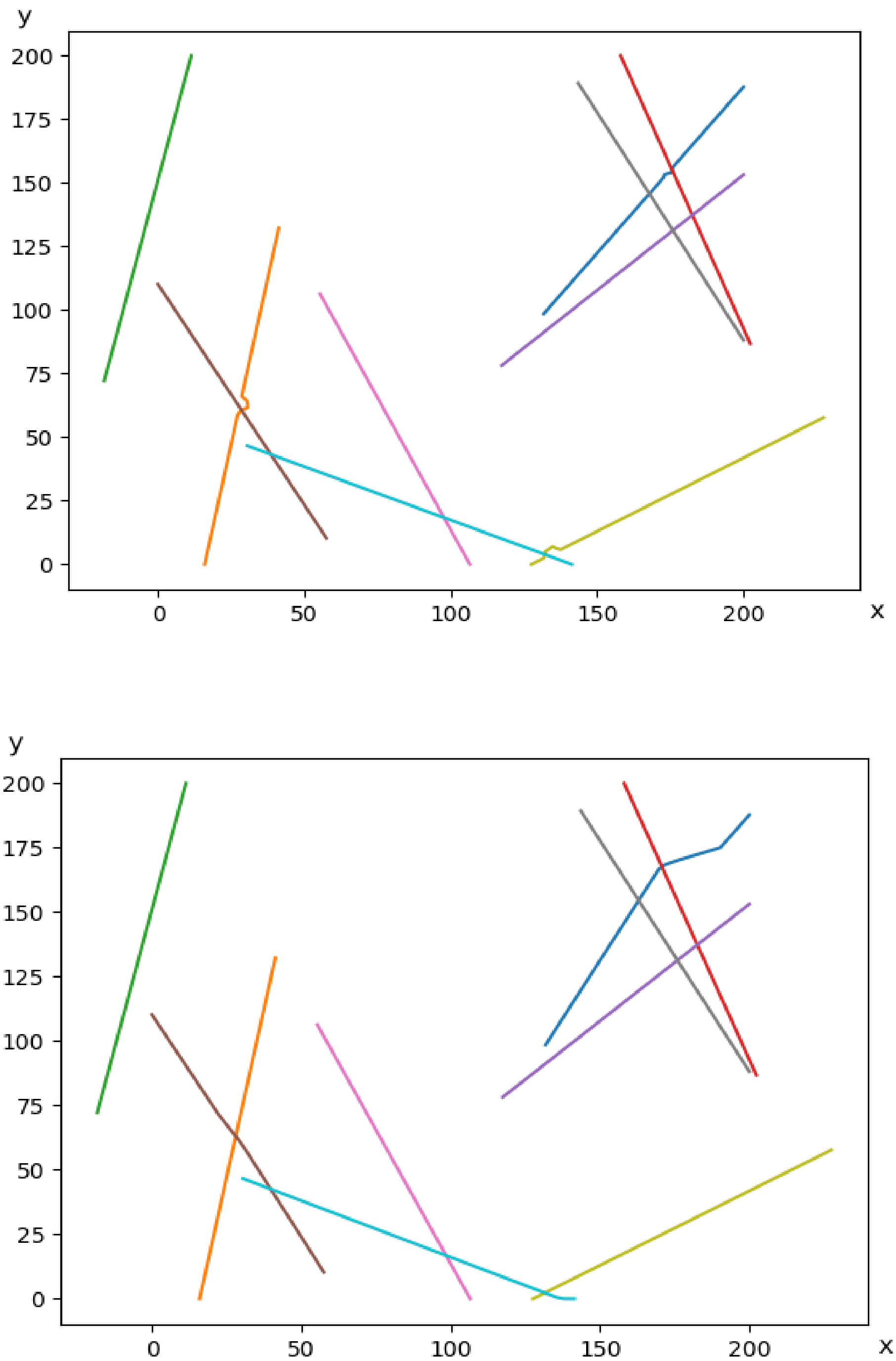
Table 2 gives results for the localization modification when applied to both MILP 1 and MILP 2 on the same test cases. Each test case was generated by first randomly generating initial trajectories for MILP 1 in the usual way, then using those start and end positions for MILP 2 to generate its own initial trajectories. Note that the objective values of the two models cannot be compared directly, with the objective function of MILP 1 being to minimize the total trajectory deviation, and the objective function of MILP 2 being to minimize the summed completion times and fuel penalties for each aircraft. As before, the objective values for MILP 2 are given as the summed difference between the computed completion time and the best possible completion time with no conflicts for each aircraft.

**Table 2. Results for the subproblem modification, MILP 1 and 2**

| | Modified MILP 1 | | | Modified MILP 2 | | |
|---|---|---|---|---|---|---|
| #Aircraft | Route Changes | Objective Value (n.m.) | Runtime (seconds) | Route Changes | Objective Value (# time steps) | Runtime (seconds) |
| 20 | 5 | 28.6 | 5.1 | 2 | 0 | 19.7 |
| | 6 | 27.3 | 1.4 | 4 | 5 | 18.0 |
| | 4 | 86.0 | 7.4 | 4 | 6 | 11.4 |
| | 7 | 32.3 | 4.4 | 7 | 10 | 45.9 |
| | 4 | 24.8 | 2.7 | 4 | 3 | 21.9 |
| 30 | 9 | 77.4 | 11.4 | 11 | 13 | 68.4 |
| | 16 | 69.7 | 7.7 | 12 | 6 | 80.1 |
| | 14 | 69.9 | 5.5 | 13 | 11 | 62.6 |
| | 10 | 51.1 | 6.0 | 10 | 1 | 67.0 |
| | 8 | 41.1 | 4.8 | 7 | 3 | 38.1 |

Figure 1 gives insight into how the differing objective functions of the two MILP models affect the generated solutions. Note that all aircraft start on one of the four walls, so each aircraft's initial position must have either the x or y coordinate as 0 or 200. The objective function for MILP 1 is to minimize the total deviation from the planned trajectory. Thus, whenever an aircraft veers from its planned route to avoid a collision, it will return to the planned trajectory as quickly as possible, hence the small humps in the graph on the left. Furthermore, aircraft in MILP 1 adhere as closely as possible to the default speed of 2.25 n.m. per 15 second time window, since a faster speed is allowed by the constraints of the model but penalized by the objective function. Meanwhile, aircraft in MILP 2 will almost always travel at the maximum speed of 2.5 n.m. per 15 second time window when possible. Although the initial conditions and the conflict detection logic is the same between the two models, this difference in velocities means that while one aircraft first detects an upcoming conflict and solves for its own trajectory in MILP 1, a different aircraft may be the first to detect the same conflict in MILP 2. This difference can also cause a conflict to appear in one model and not the other, as seen in Table 2, or cause a conflict that would have required only a slight adjustment in trajectory in one model to require a major adjustment in the other.

**Figure 1. Results for the subproblem modification, MILP 1 (top) and MILP 2 (bottom)**



## CONCLUSIONS

Trajectory optimization with collision avoidance constraints written as mixed integer linear programs (MILPs) for centralized and distributed control strategies for a 2-dimensional version of air traffic control were developed, implemented and compared. MILP 2, based on a competing paradigm called model predictive control (Richards and How, 2002), was developed and implemented. As has been previously noted (Kincaid, Curtis, and Wolf, 2023) MILP models for a centralized control strategy may produce better solutions but the computational time needed to produce a solution is a major drawback to this approach. MILPs for the distributed control strategy typically require less computational time but generate solutions that are not as good (with respect to the performance metric) as the centralized control MILP solutions. Even though solutions to the MILPs for the distributed control strategy can be found more quickly the computational time is still long.

A detection distance parameter is introduced in this paper to further reduce the computational effort required for the MILPs that implement the decentralized control strategy. The performance of MILP 1 and MILP 2, with and without the detection distance parameter, are compared on simulated data for up to 30 planes needing to resolve conflicts in a 2-dimensional airspace. The results indicate that the computational effort is least for the modified-MILP 1 which includes the detection distance parameter. Furthermore, the computational cost for MILP 2 is roughly an order of magnitude faster with the detection distance parameter included (modified-MILP 2).

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge Dr. Natalia Alexandrov at NASA Langley Research Center for her help in formulating the problem and feedback on the models presented. Additionally, we acknowledge the contributions of several William & Mary students in 2012-2013 who built and tested an earlier version of the baseline model (Keith Webb, Jennifer Thorne and Matthew Zinn). The authors acknowledge William & Mary Research Computing for providing computational resources and/or technical support that have contributed to the results reported within this paper. URL: https://www.wm.edu/it/rc.

## REFERENCES

Richards, A.G., and How, J.P. (2002). Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming. *Proceedings of American Control Conference.* ACC02-AIAA1057.

Cai, J. & Zhang, N. (2018). Mixed Integer Nonlinear Programming for Three-dimensional Aircraft Conflict Avoidance. *PeerJ Preprints 6:327410v1. https://doi.org/10.7287/peerj.preprints.27410v1*.

Xue, M. (2020). Urban Air Mobility Conflict Resolution: Centralized or Decentralized? NASA Ames Research Center. AIAA AVIATION 2020 FORUM. June 15-19 VIRTUAL EVENT. https://doi.org/10.2514/6.2020-3192.

Fourer, R., Gay, D.M. & Kernighan, B.W. (2003) **AMPL: A Modeling Language for Mathematical Programming**. 2nd Edition. Thompson-Brooks/Cole.

Gurobi Optimization, LLC. (2023) **Gurobi Optimization Reference Manual**. https://www.gurobi.com.

Kincaid, R.K., Curtis, C., and Wolf, L. (2023). Analysis of Centralized and Distributed Air Traffic Management Systems via Mixed Integer Linear Programs. *Proceedings of the 2023 MODSIM World Conference.* Norfolk, VA. May 22-23. No. 3018.