2024-05-21

Distributed Conflict Detection and Optimal 4D Trajectory Resolution Leveraging Polynomial Based Methods

Michael Klinefelter
Brigham Young University - Provo, mikline@byu.edu

Austin Stone
Brigham Young University, Backflipsciboy@gmail.com

Joshua Miller
Brigham Young University, jmiller627@gatech.edu

Cameron K. Peterson
Brigham Young University, cammy.peterson@byu.edu

John Salmon
Brigham Young University, johnsalmon@byu.edu

BYU ScholarsArchive Citation
Klinefelter, Michael; Stone, Austin; Miller, Joshua; Peterson, Cameron K.; and Salmon, John, "Distributed Conflict Detection and Optimal 4D Trajectory Resolution Leveraging Polynomial Based Methods" (2024). *Student Works*. 388.
https://scholarsarchive.byu.edu/studentpub/388

Distributed Conflict Detection and Optimal 4D Trajectory Resolution Leveraging Polynomial Based Methods

Michael Klinefelter*, Austin Stone†, Joshua Miller‡, Cameron K. Peterson§, and John Salmon¶
Brigham Young University, Provo, UT, 84602

This paper presents a methodology for distributed conflict detection and resolution of aircraft following time-dependent flight paths. We use parametric fifth-order polynomial splines to define the full, time-based paths of vehicles. This representation can be exploited to rapidly detect conflicts and calculate optimal resolution solutions that minimize deviations from the original path. Conflicts are identified using a Sturm sequencing procedure and resolutions are found using gradient-based optimization techniques. Simulations show the locally optimal resolution of complex multi-vehicle conflicts and large-scale scenarios. Also, a method of fitting the flight path model to data sets is presented and flight path trajectories are shown to accurately represent commercial aircraft trajectories from collected automatic dependent surveillance–broadcast (ADS-B) data sets.

I. Introduction

New air traffic management (ATM) concepts are needed to meet the growing trend in air traffic demand. This is illustrated through initiatives such as the Next Generation Air Transportation System (NextGen) and the Single European Sky ATM Research (SESAR), spearheaded by the United States and the European Union, respectively. To address the escalating demand for air travel, these systems are crafted to automate portions of the ATM process. Automation is a crucial component in managing the overall flow of air traffic and has prompted a series of additional initiatives specifically aimed at enhancing the air traffic flow [1, 2]. Specifically, automated tools that help enable the implementation of trajectory-based operations (TBO) have been topics of research to ensure optimal, efficient, and conflict-free flight operations [3]. These initiatives underscore the industry's acknowledgment that more automated, future-oriented traffic management techniques are needed.

A component of managing the air traffic flow during TBOs is the aircraft's trajectory planner, which must detect and deconflict paths with other aircraft. Research on multi-vehicle trajectory planning can be classified into three main categories: (1) strategic, (2) tactical, and (3) emergency. (1) Strategic planning is conducted days to months

[*] Student, Department of Mechanical Engineering, Brigham Young University.
[†] Student, Department of Electrical Engineering, Brigham Young University.
[‡] Student, Department of Mechanical Engineering, Brigham Young University.
[§] Associate Professor, Electrical and Computer Engineering Department.
[¶] Associate Professor, Mechanical Engineering Department.

prior to a flight and focuses on identifying optimal travel corridors, scheduling, and throughput issues. (2) Tactical planning begins hours before a flight and continues throughout the journey, ensuring the flight path is adjusted to avoid potential conflicts and to adapt to in-flight emergencies, airspace restriction changes, or disruptive weather patterns. This planning is typically carried out by trained air traffic controllers with the assistance of software. (3) Emergency path planning is limited to a window of seconds before a catastrophic event and involves employing sense-and-avoid strategies to take rapid evasive action.

Substantial research efforts have been dedicated to enhancing conflict resolution and trajectory formulation of the existing ATM frameworks [4]. While significant research has been conducted on the strategic and emergency regimes, tactical planning has received less attention. However, recent surveys of TBO-based operations involved in the tactical phase can be found in [5]. Ideal automated tactical planning algorithms enable vehicles to safely navigate their surroundings without human intervention. Under these algorithms, vehicles proactively avoid collisions, maintain safe distances, and prevent conflicts with other objects or entities while optimizing performance and minimizing risks. These algorithms utilize a combination of sensors, avoidance algorithms, and communication protocols to detect and respond to obstacles, hazards, or other objects in their environment in real-time [6]. These vehicle behaviors are applicable across a wide range of applications in a variety of industries, including transportation and logistics [7], surveillance [8], and agriculture [9].

Existing tactical planners have an inherent challenge in balancing between the scalability and computational complexity of large-order systems [5]. To account for operations in congested airspaces, the scalability of tactical planners is critical. The trade-offs between scalability and complexity can be seen in the following areas. First, the complexity of mathematical models used to define flight paths makes it difficult to evaluate solution paths quickly. These models become increasingly intricate as airspace dimensions and traffic volumes expand [10]. Second, a significant computational burden results from the need to identify loss of separation (LOS) events across vast areas accurately [11]. Finally, the task of providing optimal, conflict-free flight paths, often relies on gradient-based methods, which are computationally expensive and further compound the computational demands.

An illustration of the challenges inherent in scaling up tactical planners can be seen in how they are implemented within distributed air traffic control (ATC) centers. Small-scale conflict detection and resolution (CD&R) algorithm implementations in ATC centers use trained individuals, with the assistance of software, to adapt to unforeseen events, such as the inclusion of new flight paths, deviations caused by adverse weather conditions, or airspace restrictions. These algorithms often result in similar computational times to those associated with strategic planning, where flight paths are predetermined, a single CD&R algorithm is executed, and a resolved plan is presented [5]. Any modifications to system parameters, such as adding new flight paths or modifying existing ones, may necessitate restarting the entire algorithm. This inflexibility hinders the seamless scalability of ATM systems and underscores the need for innovative approaches to overcome these challenges.

More advanced automated ATM solutions, while highly varied in idea, tend to fall into one of two categories. First, a combined rigorous path planning algorithm with scheduling techniques that define a path along which vehicles may travel while guaranteeing separation between the derived path and other vehicles [12–14]. Second, non-TBO algorithms that alter a vehicle's immediate path to avoid collision without regard for the prior trajectory or distant future intent. Problems arise in these systems when dynamically evolving environments are considered. Path planning algorithms tend to be computationally expensive when operating with a large number of vehicles, which are constraints for each other, and do not allow for real-time operations. Additionally, degradation in safety and efficacy occurs when the aircraft's actual flight path deviates from its proposed trajectory. This second set of algorithms, which may define a local trajectory, fail to look past the local encounter and guarantee that the remaining trajectory is conflict-free with respect to other vehicles further along in the system. While they may offer a best-case resolution in the immediate scenario, prior planning may have avoided the conflict altogether through less drastic resolution maneuvers. Similarly, disruptions to the system caused by a local resolution maneuver are difficult to assess from a whole system health perspective. These types of solutions fall into the categories of strategic and emergency planning. In this paper, we specifically address tactical planning to mitigate the need for computationally complex optimizations (as would be required for strategic planning), while maintaining the original flight path's intent.

In this paper, we propose an automated tactical planning method that scales to large numbers of vehicles while handling simultaneous conflicts. Our trajectories use 4D flight path models with piecewise 5th-order polynomials uniquely defined by enhanced waypoints. Each enhanced waypoint in the flight path is time-based and sets the vehicle's position, velocity, and acceleration at that point and provides a straightforward method to interpolate between them. The flight path between pairs of waypoints is defined using parametric polynomial splines of the fifth order using data that can be generated independently using desired flight characteristics or fit to preexisting flight information such as automatic dependent surveillance–broadcast (ADS-B) messages. This formulation allows for control of the vehicle's position, velocity, and acceleration at each waypoint. We have explored the properties of this family of polynomial splines and exploited them to break up a larger airspace into smaller subspaces and provide rapid conflict detection through Sturm sequencing [15]. This allows us to break down a complex computational problem into localized tasks, which can be solved as a distributed optimization problem. Additionally, the paths are continuously differentiable of class C^3 with respect to time at all points, and C^∞ between enhanced waypoints.

Our method is summarized by the flow chart in Figure 1. It shows how the receiving of new flight information begins the CD&R algorithms by grouping of vehicles into cohorts to allow for distributed loss of separation (LoS) detection. If an LoS event is detected then the cohort of vehicles undergoes trajectory optimization to resolve the conflict. Once a conflict free resolution is found, or if there was no conflict to begin with, the new flight information is added to the flight operations system and normal flight operations resume until new information is obtained. The method we provide is simple enough to allow for scaling to large systems while providing rapid detection of conflicts,

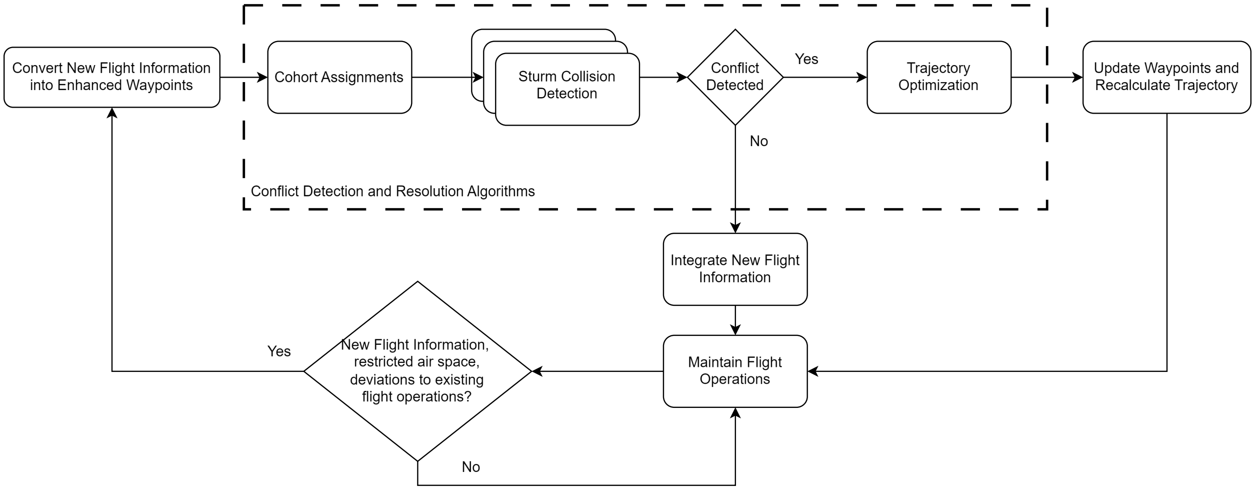

Convert New Flight Information into Enhanced Waypoints
Cohort Assignments
Sturm Collision Detection
Conflict Detected
Yes
No
Trajectory Optimization
Update Waypoints and Recalculate Trajectory
Conflict Detection and Resolution Algorithms
Integrate New Flight Information
Maintain Flight Operations
New Flight Information, restricted air space, deviations to existing flight operations?
Yes
No

Fig. 1 Overview of proposed conflict detection and resolution algorithms.

yet it still maintains enough fidelity to accurately describe realistic flight paths. A procedure for resolving conflicts that exploit these tradeoffs utilizing established gradient-descent constrained optimization methods is given and applied to situational stress tests involving multiple vehicle conflicts.

Figure 2 illustrates an overview of our CD&R methodology using 4D waypoints. In it, two aircraft flying alongside each other with the pre-resolution state of the aircraft indicated by the purple aircraft flying the trajectory given by the red dashed line. Their LOS volumes, gray cylinders, are at each aircraft's center of mass. The LOS volumes of the two aircraft intersect the other aircraft at the moment of time shown by the purple aircraft, indicating an LOS event. To mitigate this conflict, a new trajectory, shown by the black solid line, is calculated which directs the aircraft to fly further away from each other. Note that an LOS volume may intersect with another volume and not be in conflict. Only when an LOS volume intersects with another aircraft is the separation constraint violated.

The CD&R algorithms can be applied to and tested against currently active or historical data sets. Flight path data can be modeled using the provided interpolation formulas through a best-fit algorithm to convert provided real-world data, such as ADS-B information, into trajectories. This formulation can be used to convert existing flight paths into the CD&R scheme presented in this paper for integration.

Application of the CD&R methods in this paper will be applied to commercial airspace. However, the scope of formulations presented has implementation potential for a wide range of path-planning applications, such as aquatic, space, and urban air mobility services. We do not address strategic planning, including optimal scheduling of take-off and landing of flights. An important assumption made in this paper is that there exists a starting waypoint that a vehicle will achieve at a prescribed time.

In summary, multi-vehicle path planning in the tactical regime requires a mathematical model that is simple enough to allow for scaling to large systems, yet contains the necessary information to accurately model real flight paths. We

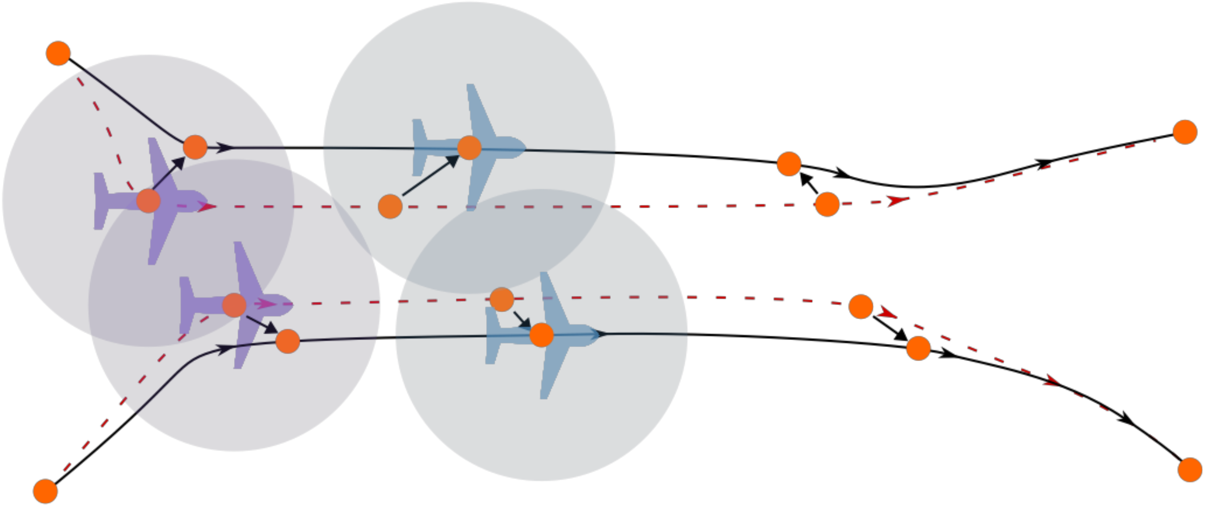


Fig. 2 Modification of 4D flight paths to maintain LOS requirements.

contribute to this field by:

- presenting a scalable and distributable tactical CD&R algorithm that provides time-based guarantees on the flight trajectory,
- using this formulation to generate flight path information for existing aircraft, and
- providing hyper-realistic conflict scenarios under simulation which provide satisfactory results.

The paper will proceed as follows. Section II presents our flight path representation and CD&R. In particular, methods for scaling to large scenarios are added. The procedures for defining flight paths and conducting CD&R work are presented generally, which allows implementation into a variety of vehicle-based systems. Results focus on hypothetical stress test cases of local conflicts with a large number of aircraft and are presented along with modeling of real-world flight data in Section III. Finally, conclusions and a summary are presented in Section IV.

II. Methods

To make tactical multi-vehicle trajectory planning possible, it is necessary to quickly find and resolve conflicts. There exists a natural trade-off between the accuracy of a trajectory formulation used to define a flight path and the computational time required to modify the flight path if another vehicle violates its LOS threshold. We employ fifth-order parametric polynomial splines to define each vehicle's time-based flight path, facilitating precise modeling of real-world aircraft paths. This approach also enables distributing conflict resolutions, lowering the overall computational complexity in scenarios involving multiple LOS violations.

In this section, we will describe our tactical path planning approach and outline the manner in which conflicts are identified and subsequently resolved. We start by defining this formulation using the vehicle's position, velocity, and acceleration at a series of time-based waypoints and provide a method to interpolate the trajectory as a function of time. Sturm sequencing is then introduced as a novel approach to rapidly identify pairwise conflicts. Next, we explore the properties of this family of polynomial splines, which allows us to break up the complex airspace into smaller cohorts. This divides a large computational problem into many local problems that can be computed in a distributed manner. Finally, a procedure for resolving conflicts utilizing established gradient descent-constrained optimization methods is given.

A. Trajectory Formulation

The 5^{th} order polynomial method provides a spline interpolation to give a continuous-time formulation that accurately describes flight paths yet is compact enough to allow for optimization techniques to be applied. We use enhanced waypoints, as adapted from Thompson and Zhang [16], to describe a vehicle's time-based flight path. An enhanced waypoint is a vector representation of the vehicle's spatial, velocity, and acceleration components in the world frame at a specific instance in time. For example, a waypoint at time t_i is defined as

$$W_i = [x(t_i), y(t_i), z(t_i), u(t_i), v(t_i), w(t_i), a(t_i), b(t_i), c(t_i), t_i]^\top,$$

where $[x(t_i), y(t_i), z(t_i)]^\top$, $[u(t_i), v(t_i), w(t_i)]^\top$, and $[a(t_i), b(t_i), c(t_i)]^\top$ are the position, velocity, and acceleration, respectively, of the vehicle at that waypoint in the world frame.

A vehicle's flight path is derived from a linked list of enhanced waypoints. The flight path generated between enhanced waypoints is called a flight segment, which is defined by three 5th order parametric polynomial equations that satisfy the enhanced waypoints' start and end conditions. The resulting flight path is therefore a piece-wise differentiable 5th order polynomial spline.

The i^{th} flight segment is the path between enhanced waypoints W_i and W_{i+1}. It is defined by three vector-valued functions: position $[x(\bar{t}), y(\bar{t}), z(\bar{t})]^T$, velocity $[u(\bar{t}), v(\bar{t}), w(\bar{t})]^T$, and acceleration $[a(\bar{t}), b(\bar{t}), c(\bar{t})]^T$, where $\bar{t} \in [t_i, t_{i+1}]$. The total time in the i^{th} segment is $T_i = t_{i+1} - t_i$ and $\tau(t) = (t - t_i)/T_i \in [0, 1]$ is the normalized time. The normalized time is used to simplify later calculations associated with LOS detection.

There are two sets of three constraints that must be satisfied for the flight segment polynomial functions to be valid. First, when $\tau = 0$, the position, velocity, and acceleration equations must match the position, velocity, and acceleration vectors of the first enhanced waypoint (W_i) in the flight segment. The second set of constraints at $\tau = 1$ requires the position, velocity, and acceleration equations to match the corresponding vectors of the end waypoint (W_{i+1}) in the flight segment. In between waypoints, we use a spline fitting algorithm that matches the beginning and end constraints with a unique 5th order polynomial as

$$R_i(\tau) = \Bigg[x(\tau), y(\tau), z(\tau)\Bigg]^\top = \Bigg[x(t_i), y(t_i), z(t_i)\Bigg]^\top + B_i C \Bigg[\tau, \tau^2, \tau^3, \tau^4, \tau^5\Bigg]^\top, \tag{1}$$

where

$$B_i = \begin{bmatrix} x(t_{i+1}) - x(t_i) & u(t_i)T_i & (u(t_{i+1}) + u(t_i))T_i & a(t_i)\frac{T_i^2}{2} & (a(t_{i+1}) - a(t_i))\frac{T_i^2}{2} \\ y(t_{i+1}) - y(t_i) & v(t_i)T_i & (v(t_{i+1}) + v(t_i))T_i & b(t_i)\frac{T_i^2}{2} & (b(t_{i+1}) - b(t_i))\frac{T_i^2}{2} \\ z(t_{i+1}) - z(t_i) & w(t_i)T_i & (w(t_{i+1}) + w(t_i))T_i & c(t_i)\frac{T_i^2}{2} & (c(t_{i+1}) - c(t_i))\frac{T_i^2}{2} \end{bmatrix}$$

and C is a 5×5 real matrix, defined as

$$C = \begin{bmatrix} 0 & 0 & 10 & -15 & 6 \\ 1 & 0 & -2 & 1 & 0 \\ 0 & 0 & -4 & 7 & -3 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}.$$

The first and second time derivatives $V_{j,i}(\tau)$ and $A_{j,i}(\tau)$ of Equation 1 with respect to τ are

$$V_{j,i}(\tau) = B_i C \begin{bmatrix} 1 \\ 2\tau \\ 3\tau^2 \\ 4\tau^3 \\ 5\tau^4 \end{bmatrix} \frac{1}{T_i}, \tag{2}$$

$$\tag{3}$$

and

$$A_{j,i}(\tau) = B_i C \begin{bmatrix} 0 \\ 2 \\ 6\tau \\ 12\tau^2 \\ 20\tau^3 \end{bmatrix} \frac{1}{T_i^2}. \tag{4}$$

The constant values in the C matrix guarantee the position, velocity, and acceleration functions meet the enhanced waypoint conditions at $\tau = 0$ and $\tau = 1$. For example, when $\tau = 0$, $V_{j,i}(0) = B_i C[1\ 0\ 0\ 0\ 0]^\top / T_i = [u(t_i)\ v(t_i)\ w(t_i)]^\top$ and it is the first column of C that is non-zero after the multiplications. Similarly, the second column of C is what remains to give the acceleration constraints. The remaining three columns of C, along with the first two, enforce the unique linear combination of elements of the B_i matrix in the position, velocity, and acceleration functions to enforce the terminal enhanced waypoint W_{i+1} conditions, e.g. $V_{j,i}(1) = [u(t_{i+1})\ v(t_{i+1})\ w(t_{i+1})]$.

B. Fitting Flight Paths to Collections of Enhanced Waypoints

There are two reasons why we might need to fit spline representations to enhanced waypoints or real-world data. The first is because we will want to fit paths so that we can use our algorithms when all that we know are a series of waypoints that we want the vehicle to fly. Second, the fitting of splines to real-world data serves the dual function of validating models through the comparison with actual aircraft flight trajectories and substantiating the efficacy of our work. This is useful in validating the trajectory models used within this architecture for CD&R scenarios or generating 4D flight paths from a collection of desired waypoints. The following paragraphs present an algorithm for fitting ADS-B data to polynomial splines to approximate a flight path.

We show that we can fit spline representations to real-world data using ADS-B data. Raw ADS-B data provides the aircraft's latitude, longitude, altitude, heading, ground speed, and other data, at every second. As is common in GPS-based positions and velocities, data provided by ADS-B is noisy, and applying a polynomial-spline fit every second to the exact ADS-B data could provide unrealistic flight paths. This is the result of our spline method being an exact fitting at each enhanced waypoint. We have found the application of a median filter [17] over position and velocity adequately smooths the data without significant alteration of the waypoints. After smoothing, the data is then coordinate transformed from the Geographic Coordinate System (GCS) frame to the Earth-Centered Earth-Fixed (ECEF) frame.

To calculate the XYZ components of velocity we use the following

$$u = V_g \sin(\chi)$$

$$v = V_g \cos(\chi)$$

$$w = \dot{h},$$

where V_g is the vehicle's ground speed, $\dot{h}$ is the vertical ascent rate, and χ is the aircraft's heading. To obtain the Cartesian components of acceleration from ADS-B data, we use a central finite difference method to calculate the numerical derivative of velocity.

After smoothing and gathering position, velocity, and acceleration vectors at each time step, an appropriate time spacing between waypoints needs to be considered. This decision is largely system dependent which should place consideration on the memory allocation, effects on a number of design variables during conflict resolution optimization, and degradation of flight path accuracy as waypoints are approximated. We present a method for removing intermediate enhanced waypoints while maintaining flight path fidelity to the original data in Algorithm II.1.

Algorithm II.1 takes advantage of the 4D flight path's local control property, where each flight segment can be individually changed without affecting subsequent or prior segments.

The algorithm works by looping over waypoints and, while ignoring the interior waypoints, tests the mean-squared error (MSE) between a flight path inclusive of all waypoints and one matched to just the two exterior points. If the MSE exceeds an upper bound then the previous exterior waypoint is kept and added to our final waypoint list. Otherwise, the index iterates to the next waypoint and checks the MSE of the flight segment that the waypoint generates. This process as described in II.2, continues until all waypoint pairs are evaluated. With this method, the smoothed flight path is guaranteed to be lower than a defined maximum error bound for each flight segment.

C. Conflict Detection Using Sturm Sequencing

The LOS distance is a function that defines the two-dimensional closed manifold around a vehicle. The set of points on this manifold creates a boundary around the vehicle from which the minimum acceptable separation distance between the self-vehicle and another vehicle is defined. An LOS violation event occurs when a vehicle crosses this threshold. For traditional aircraft in Class A airspace, the LOS distance manifold is defined with a vertical dimension of 1000 feet and horizontal displacement of 5 nautical miles centered on the aircraft [18]. This creates a cylinder that moves with the vehicle at its center as seen in Figure 3. Other LOS manifolds, such as spheres and ellipsoids are readily derivable to fit within the context of conflict detection described herein. The LOS distance function can similarly be defined for non-vehicle dynamic or static obstacles, such as ill-conditioned weather patterns and restricted airspace using composite polygon shapes [19]. The distance used for LOS calculations for 4D trajectories is the $\mathcal{L}^2$ difference of

Algorithm II.1: Polynomial Path Smoothing

```
Input: Allowable error($\epsilon$), List of 4D waypoints($W$)
Output: List of reduced 4D Waypoints($S$)
Function PathSmoothing($\epsilon$,$W$) is
    $i \leftarrow 1$
    $S[i] \leftarrow W[1]$
    for $j = 2$ through length($W$) do
        $Intermediates \leftarrow W[i+1:j]$
        $Start \leftarrow W[i]$
        $End \leftarrow W[j+1]$
        $MSE_{new} \leftarrow$ Flight path MSE($Start$, $End$, $Intermediates$)
        if $MSE_{new} > \epsilon$ then
            $MSE_{new} \leftarrow$ Flight path MSE($Start$, $W[j]$, $Intermediates$)
            $S \leftarrow [S, W[j]]$
            $i \leftarrow j$
            $MSE_{total} \leftarrow MSE_{total} + MSE_{new}$
```

Algorithm II.2: Flight Path Mean-Squared Error

```
Input: Waypoints($W$), Number of Discretized Points($N$)
Output: $MSE$
Function Flight path MSE($W$) is
    $MSE \leftarrow 0$
    for $i = 1$ to $|I|$ do
        $R_i \leftarrow I[i]$
        $R \leftarrow x_0 + B_i C \left[\tau, \tau^2, \tau^3, \tau^4, \tau^5\right]^\top$
        $MSE \leftarrow MSE + ||R - R_i||_2$
```

the position vectors between vehicles from an inertial frame. Within this paradigm, an LOS violation occurs when the relative position vector of another vehicle coincides with the LOS distance surface.

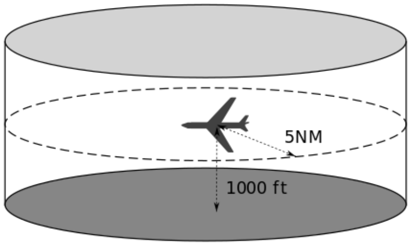


Fig. 3 Loss of Separation Cylinder of an aircraft in Class A airspace.

Sturm sequences are a method for determining the number of real roots of a polynomial function. Various component distance functions between aircraft derived from the 4D trajectories previously defined are themselves a set of polynomial functions formulated from the LOS manifold shape where roots represent an LOS event. We use Sturm sequences as a computationally efficient method for determining when LOS events occur between two vehicles.

Consider a polynomial function $P(\tau)$ and the resulting sequence

$$\begin{aligned} S_0 &= P(\tau) \\ S_1 &= \tfrac{dP(\tau)}{d\tau} \\ S_{j+1} &= -\text{rem}(S(\tau)_{j-1}, S(\tau)_j), \end{aligned}$$

where $\text{rem}(S_{j-1}, S_j)$ is the division remainder obtained when dividing S_{j-1} by S_j. So long as $P(\tau)$ is a polynomial, these sequences are finite. The function $\nu(\tau),\ \tau \in [0,1]$ is then defined as the number of sign changes of the Sturm sequence $\{S_0(\tau), S_1(\tau), S_2(\tau), ..., S_n(\tau)\}$. Sturm's Theorem states that for square-free polynomials, roots without repetition, on the interval $(0,1]$, $\nu(0) - \nu(1)$ gives the number of real roots [15].

In the case of conflict detection, by setting $P(\tau)$ equal to the squared Euclidean distance between vehicles less a square LOS radius, the zeros of $P(\tau)$ represent a vehicle on the LOS spherical manifold and $\nu(0) - \nu(1)$ will be greater than zero. Provided that no vehicles start in violation of LOS rules, which is trivially checked by ensuring $P(0) > 0$, any consecutive sequence of waypoints can be determined to be conflict-free using this method. Similarly, for the typical cylindrical LOS manifold, a set of two Sturm sequences will need to be checked for each flight segment with $P_1(\tau)$ being defined by the vertical LOS requirements and $P_2(\tau)$ defined by the horizontal requirements. In this case, both $\nu_1(0) - \nu_1(1)$ and $\nu_2(0) - \nu_2(1)$ will be greater than zero if a conflict to occurs (i.e. both vertical and horizontal violation occurs).

Significant time is spent in current multi-vehicle path planning algorithms and dynamic obstacle avoidance algorithms determining if and/or when a collision, or equivalently an LOS event, has occurred. The polynomial structure of the flight models outlined in this paper allows for a significant reduction of computational time in determining whether an LOS event has occurred by reducing the algorithm to a fixed length, a deterministic computation which does not require typical root finding gradient descent methods. Furthermore, this method avoids errors associated with finite sampling methods for when LOS events start and end on any given segment. While this method does not give the exact time of an LOS event, standard root-finding techniques can then be employed to find when LOS begins, reaches its minimum, and resolves. However actual roots are not required, only their existence on a flight segment for the CD&R algorithm.

D. Cohorts

Each aircraft in a given airspace needs to guarantee LOS with all other aircraft or objects in that airspace. This can be a computationally expensive endeavor when dealing with large numbers of aircraft. Pairwise conflict detection between m aircraft in a given space requires $m^2 - m/2$ calls to a detection algorithm. This is further complicated in our formulation by each vehicle having multiple flight segments where each flight segment pairing between vehicles with overlapping times needs to be checked for conflicts using a separate Sturm sequence. To ensure the tractability of the detection algorithms in large-scale situations, we use cohorts to break up conflict detection, and subsequent path resolution calculations, into smaller problem groupings. The cohort of a vehicle on a given flight segment is a set of aircraft that occupy the same or neighboring subset of the larger airspace within a given time interval. Cohort lists on a flight segment are thus reflexive, in that if vehicle B is on vehicle A's cohort list of a given segment, then vehicle A will be on a corresponding cohort list of vehicle B.

In the example shown in Figure 4, the cohort of aircraft A2, assuming the timing of aircraft overlap within the space, will include:

- A1 which will not show a collision.
- A3 and A4 which may show an LOS event if the trajectory crossings happen at similar times in both aircraft.
- A6 as it is in the neighboring region and may be affected by conflict resolution maneuvers.

A5 will not be in the cohort of A2 within the Cohort 1 region as it does not cross into a neighboring region.

E. Conflict Resolution

Sturm sequences are checked for all cohort lists for LOS events. All flight segments that contain these events, as well as the preceding and following segments, are sent for conflict resolution. We use a standard interior-point gradient descent algorithm with an objective function that penalizes the summed deviation between the proposed and original flight paths over the set of K affected flight segments of M vehicles. The objective is subject to the constraint in that the LOS distance(s) must be maintained between vehicles along the entirety of the segment. Thus the optimization with a spherical LOS manifold can be written as the minimization

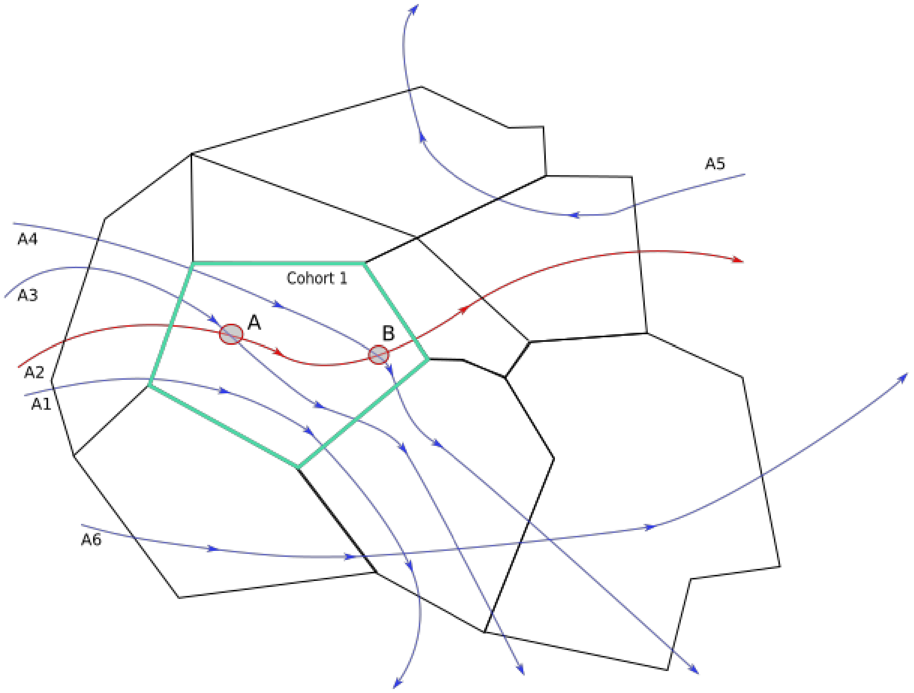


Fig. 4 ***Regional division and cohort inclusions overview.***

$$
\begin{aligned}
\min \quad & \sum_{j \in M} \sum_{i \in K} \int_{t_i}^{t_f} \left\| (R_{j,proposed}(\tau) - R_{j,original}(\tau)) \right\|_2^2 d\tau \\
\text{subject to} \quad & \|(R_{p,i}(\tau) - R_{q,i}(\tau))\|^2 - LOS^2 > 0, \forall p, q \in M, p \neq q.
\end{aligned}
\tag{5}
$$

or similarly with a traditional vertical cylinder as

$$
\begin{aligned}
\text{subject to} \quad & \|(R_{p,i}(\tau)_z - R_{q,i}(\tau)_z)\|^2 - LOS^2_{vertical} > 0, \forall p, q \in M, p \neq q. \\
& \|(R_{p,i}(\tau)_{x,y} - R_{q,i}(\tau)_{x,y})\|^2 - LOS^2_{horizontal} > 0, \forall p, q \in M, p \neq q.
\end{aligned}
\tag{6}
$$

Design variables within the optimization are the start and end enhanced waypoint position, velocity, and acceleration vectors of each affected flight segment. This method gives equal priority to each vehicles with the cost function looking to minimize total deviation across all vehicles. Removing enhanced waypoints from a single vehicle will have the effect of giving that vehicle total priority, causing all other vehicles to ensure the minimum separation constraint is met. Note that while time is stored at each enhanced waypoint, it is not varied. This is to mitigate the risk of potential numerical inaccuracies during integration over near-zero length time intervals. Positions are normalized to the interval [0, 1] against the entire affected airspace, while velocities and accelerations are normalized component-wise to maximum expected (or allowed) velocities and accelerations. Gradients are calculated using complex-step methods [20, 21].

The optimization function's halting criteria are largely scenario-dependent but should be based on first-order optimality and a maximum run time with no limit on function evaluations. To meet the goals of a medium-term CD&R regime, optimization occurs in short-time increments based on system goals. If a feasible solution is not obtained within a discrete short-time increment, the waypoints preceding and following the currently included flight segments are included and the optimization is restarted. If at any point the first or last segment of the total flight path is included and a feasible solution cannot be found, then the conflict becomes a flight scheduling issue, which would require delaying or speeding up the start/end of the entire flight path. This scenario is outside the scope of this research.

The CD&R methodology presented here assumes the proposed initial flight paths, including waypoint positions, were chosen for a reason and our conflict resolution should minimize changes made to the original path. We also assume that the positions of all vehicles are known at all times. The use of our objective function (Equation (5)) prevents reshaping of the flight path outside the local conflict area and rewards solutions that bring the vehicle trajectory back to the originally proposed flight path as soon as is feasible, or as dynamic considerations allow. Thus optimization local to the conflict is equivalent to optimization over the entire flight path and is guaranteed to not affect additional aircraft before and after the resolution. The benefit trade-off here is the removal of creating additional conflicts that did not previously exist, where one resolution causes another conflict which cascades throughout a system.

Running the optimization scheme described in this paper without total energy, velocity, or other dynamic vehicle constraints, may lead to scenarios where a vehicle executes a rapid deviation between two waypoints in order to avoid a conflict. While this may produce optimal results in the sense of our objective function, it can lead to high accelerations or other infeasible maneuvers. This case worsens as an enhanced waypoint used to define the polynomial interpolation nears the onset of a conflict (LOS event) relative to time. This is a result of the solutions being localized to the flight segments in which the conflict occurs, as described in the prior paragraph. Flight paths that are dense in waypoints are more likely to experience this problem. In general, it is desirable to use the fewest number of waypoints possible to accurately describe a flight path. Carefully chosen dynamic constraints eliminate this problem by forcing deviations to extend beyond waypoints local to a conflict. Indeed, in a commercial setting, rapid changes in bank angle or high accelerations perpendicular to the vehicle's velocity vector are undesirable. A pilot in such a scenario would begin smooth mitigation procedures long before an LOS violation would occur as opposed to veering away at the last possible second. Such constraints are highly dependent on the scenario and specific vehicles and are not included in this paper, though such constraints can be easily defined from the design variables as the polynomial formulation fully defines the position, velocity, and acceleration of the vehicles at all times.

III. Results

In this section, we apply the methodology described in Section II to various multi-vehicle CD&R scenarios. We first present simulation results that are designed to stress the limits of the optimization algorithm and verify that it is working properly. We then provide an example of fitting real-world ADS-B data points to our polynomial trajectory model. This validates that our flight path model accurately represents actual trajectories from commercial aircraft. Finally, we use real-world flight data gathered from commercial aircraft around the Atlanta, GA airport to demonstrate our method's efficacy in resolving conflicts in a hyper-realistic scenario with many aircraft.

A. Simulations

The simulation results presented here are provided to illustrate the robustness and versatility of our optimization methodology. First, we present a baseline scenario involving two aircraft on the same flight level with an LOS event. Additionally, we resolve two complex local conflicts involving multiple aircraft under simultaneous LOS violations. These scenarios are well beyond what is likely to occur in real-world situations but are used to illustrate the robustness of the optimization methodology.

The optimizations of each simulation were carried out in MATLAB on a 4-core i7-9750H at a clock speed of 3.6 GHz. The code was not optimized for speed and we anticipate the computational time will decrease significantly once implemented in C++ or a similar compiled language. Optimization settings include a final tolerance of 10^{-8} with design variables normalized to the interval [0, 1].

1. Baseline Example

Consider two aircraft approaching each other from opposing directions on a direct collision course. Each aircraft trajectory is modeled with 5 enhanced waypoints. Once Sturm-sequencing has detected the presence of a conflict, the paths are sent off to receive resolution through optimization. Figure 5 shows the initial conflicting trajectories and resolved conflict-free trajectories. As seen here, reciprocal resolutions are possible with both aircraft attempting to deviate minimally from the initially planned trajectory while maintaining required LOS distancing. Figure 5c shows the distance between the two aircraft over time and the total deviation from the initial trajectories. Of note is the ability of the 4D polynomials to hug the LOS requirement, which is 300 meters near the LOS event followed by rapid convergence back to the original flight path trajectory, which is the intended outcome of the resolution algorithm. This ensures that the conflict resolution remains local to the LOS event and does not cause further conflicts in subsequent flight paths.

2. Multiple Aircraft under Simultaneous Conflict

The second example simulation shows eight vehicles with initial positions equally placed around a circle of radius 500 meters. Each vehicle's proposed flight path takes them to the opposite side of the circle through the center, with all aircraft reaching the center at the same time. Each aircraft contains seven enhanced waypoints, including the initial and final enhanced waypoint which will remain unchanged in the optimization. The conflict resolution optimization was solved simultaneously with all eight vehicles receiving equal priority. Figure 6 shows the unresolved and resolved trajectories for these vehicles. The resolution is symmetric with respect to opposing vehicle pairs. Figure 7 shows the

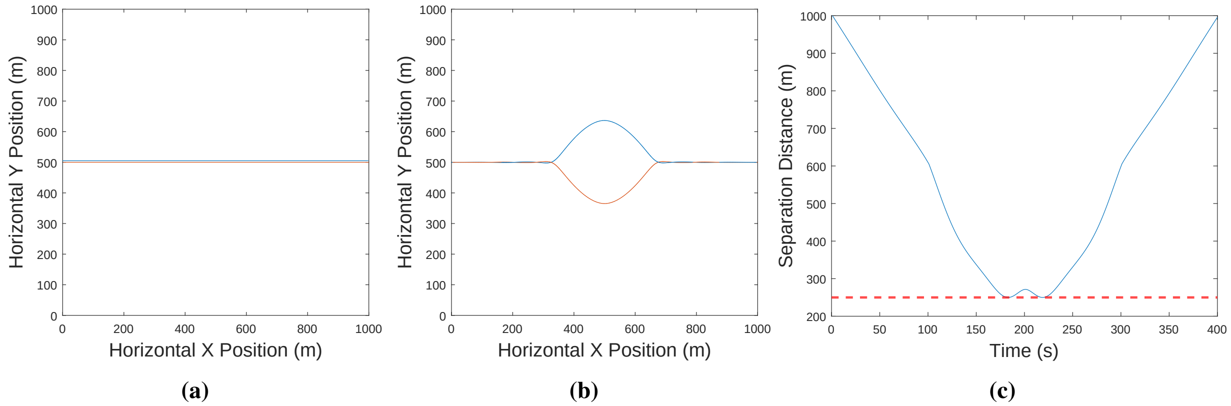
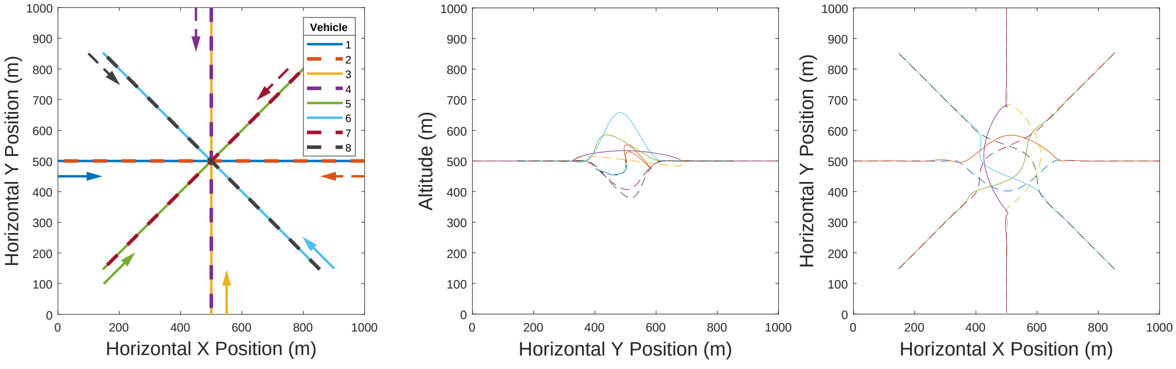


Fig. 5 ***(a) Unresolved baseline example flight path of opposing vehicles. (b) Simultaneous resolution of opposing vehicles. (c) Distance between baseline example vehicles after conflict resolution.***

pairwise separation graphs for all pairs. In each case, minimums are achieved between nearest neighbors around the initial circle. The spacing is such that opposing aircraft never reach their minimum LOS distance as LOS conditions on neighboring aircraft take priority. The overlap of all separation distances as well as the overall system minimum separation between any two vehicles is shown in Figure 8a and 8b, respectively. The overall system maintains the minimum separation within 0.8 meters or 0.64% of the LOS condition. We note the second and sixth enhanced waypoints remain unchanged as the LOS constraints were met through changes to the middle enhanced waypoints three through five. The second waypoints had a separation distance of 125 meters from their nearest neighbor, which is greater than the LOS threshold set for this simulation. This indicates the success of the optimization requirement to return aircraft to their originally planned flight paths when possible. The resolution of the system was achieved in 4 minutes 10 seconds. Additional flight model constraints may be required, such as maximum velocity/acceleration in order to keep the deviations dynamically viable. There is an observation worth going over from Figure 7. From this simulation, it can be seen that polynomial over-fitting occurs in the resolved flight paths. This is a result of enhanced waypoints being fixed in time. The objective function reduces the L^2 norm to the original flight path. Because changes to the flight path are local to adjacent enhanced waypoints, and the derivatives from the preceding enhanced waypoint remain unchanged, the velocity and acceleration vectors coming through the proceeding enhanced waypoint remain unchanged. Thus in order to fit a polynomial that deviates in a singular direction, it is often required to move in the opposite direction first, if only for a short moment. This can be seen in Figure 6 at the start of the divergence of the flight paths from pre-resolution trajectories. To some degree, additional constraints on maximum velocities or penalties on energy consumption will alleviate this and require more enhanced waypoints to be pulled into the resolution, providing a more gradual deviation from the flight path. The effects of doing this are not explored in this paper as they are heavily dependent on the vehicle and dynamic constraints used.



(a)

(b)

(c)

Fig. 6 *(a) Unresolved trajectories for eight converging vehicles (top view). (b) Side-view of resolved vehicle trajectories. (c) Top-view of resolved vehicle trajectories.*

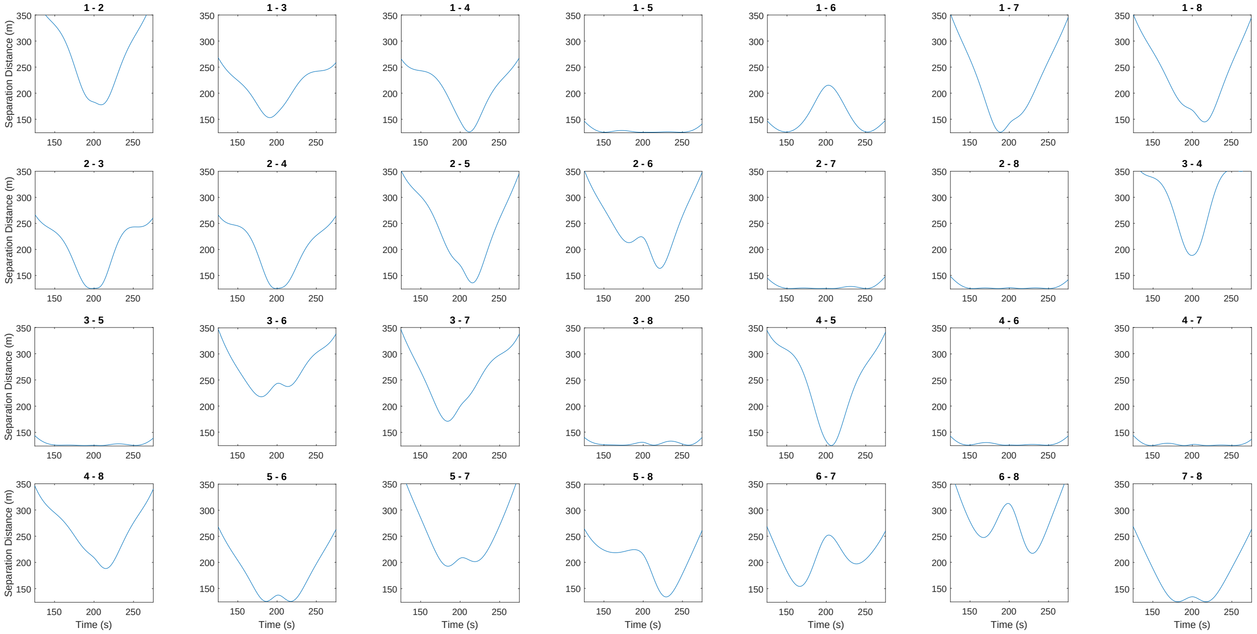
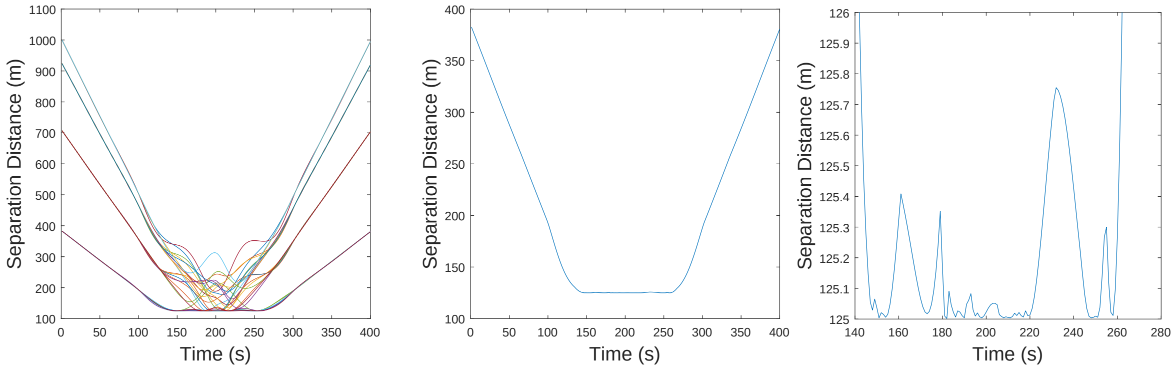


Fig. 7 *Separation distance for all vehicle pairs.*



(a)

(b)

(c)

Fig. 8 *(a) Compiled separation distance for all vehicle pairs. (b,c) System-wide minimum loss of separation across all vehicles.*

Table 1 Error Rates from Reduced Sampling of the original 3599 Waypoints from Flight EZY158T

Maximum MSE per segment [m]	0	0.01	0.100	1.	10	100
Number of Enhanced Waypoints	3599	3316	3072	1557	381	104
Compression Ratio	1.00	1.09	1.17	2.31	9.45	34.61
Total MSE [m]	0	0.452	5.720	231.1	1491.	6354.

3. Large-Scale Cascading Conflicts

Figure 9 contains a 4x4 wall of vehicle separated horizontally and vertically by the minimum LOS distance to neighboring vehicles moving parallel to each other with equal velocities in the y-direction and a single vehicle moving in the opposite direction with identical speed. Each individual aircraft contained five enhanced waypoints. The resolved trajectory results are shown in Figure 9. The 4 wall of vehicles is seen to shift outwards from the center opposing vehicle a they move near each other and resolve back into the expected formation defined by the original trajectories post conflict.

Figure 9 shows the Euclidean distance between the aircraft for all pairs. In each case, minimums are achieved between the nearest neighbors of the aircraft moving in the opposing direction. The effect of resolutions' ability to cascade out trajectories is evident in Figure 9. Resolution of all aircraft occurred in 4 minutes 33 seconds.

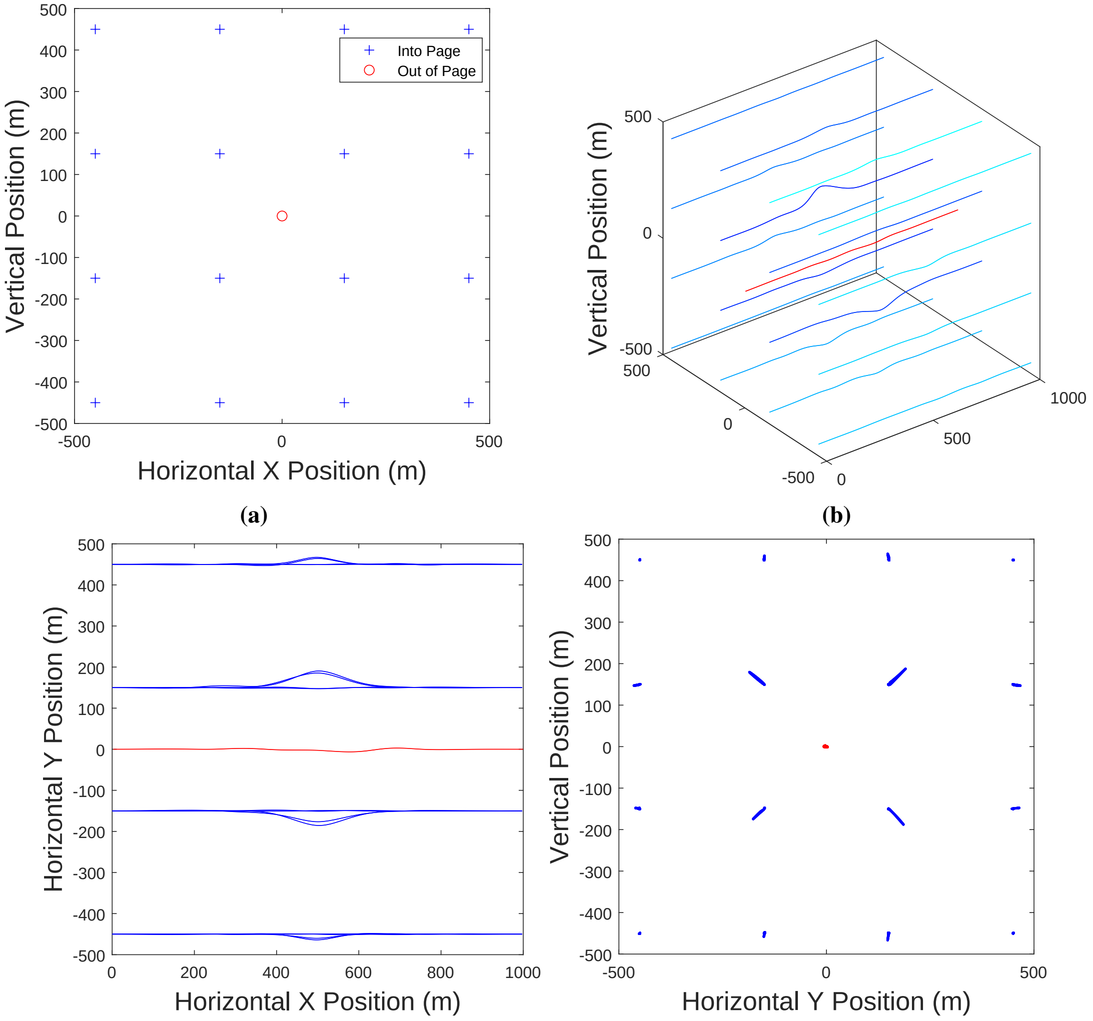
The separation distance between aircraft is shown in Figure 10. The mostly horizontal lines represent the separation distance between vehicle groups traveling in the same direction with divergence occurring around the 200-second mark due to the necessity of providing room for oncoming vehicles.

B. Real-World Flight path Fitting

In this section, we demonstrate the effectiveness of using 5th-order piecewise polynomials to model real-world aircraft trajectories. We do this by fitting 4D flight paths to ADS-B data gathered from the OpenSky Network [22].

To test our trajectory model, we selected a representative flight path (call sign EZY158T) from February 5th, 2017 between 15:45Z and 16:45Z. Using Algorithm II.1 we removed 90% of the enhanced waypoints and kept the MSE of each flight segment less than one kilometer, or an average error of 2.7m per waypoint over a flight path length of approximately 10 km.

From a visual inspection of Figure 11, it is evident that no significant changes to the flight path from the original trajectory are made due to removing enhanced waypoints. The smoothing algorithm provides a way to reduce the number of enhanced waypoints when considering the limitations of the computational complexity of the optimization due to a large design variable space. The reduction in required waypoints using varying MSE requirements of Flight EZY158T is compared to the original flight path in Table 1.



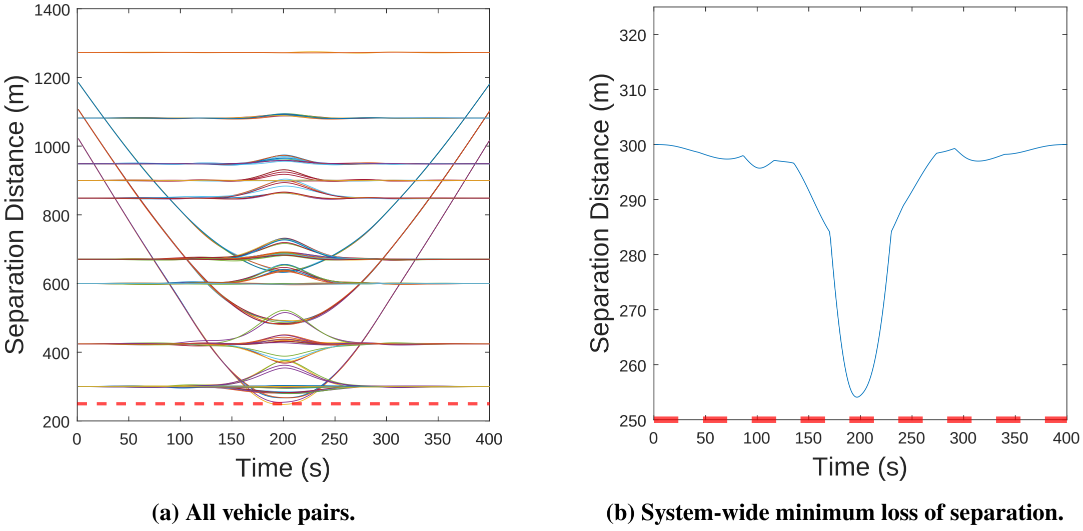
(a)

(b)

(c)

(d)

Fig. 9 ***Large-scale cascading conflict resolution (a) Unresolved vehicle flight paths (side-view). (b-d) Resolved vehicle flight paths***



(a) All vehicle pairs.

(b) System-wide minimum loss of separation.

Fig. 10 ***Large scale Cascading conflict separation distances. (a) Compiled separation distance for all vehicle pairs. (b) Minimum Loss of Separation across all vehicles.***

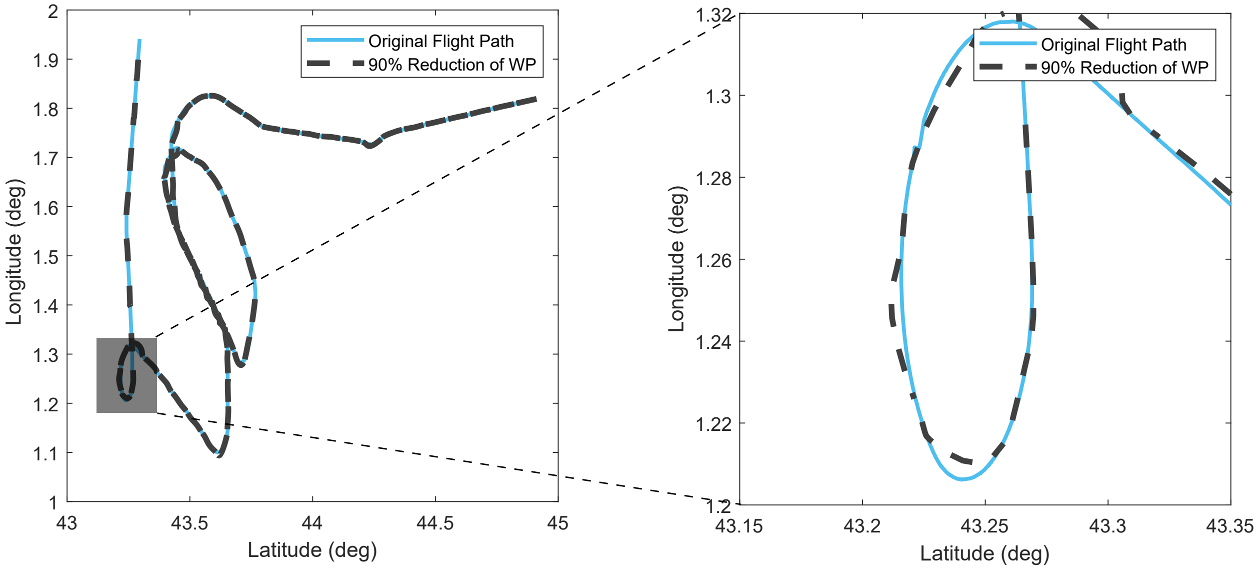


Fig. 11 ***Original flight EZY158T and 90 percent waypoint reduction***

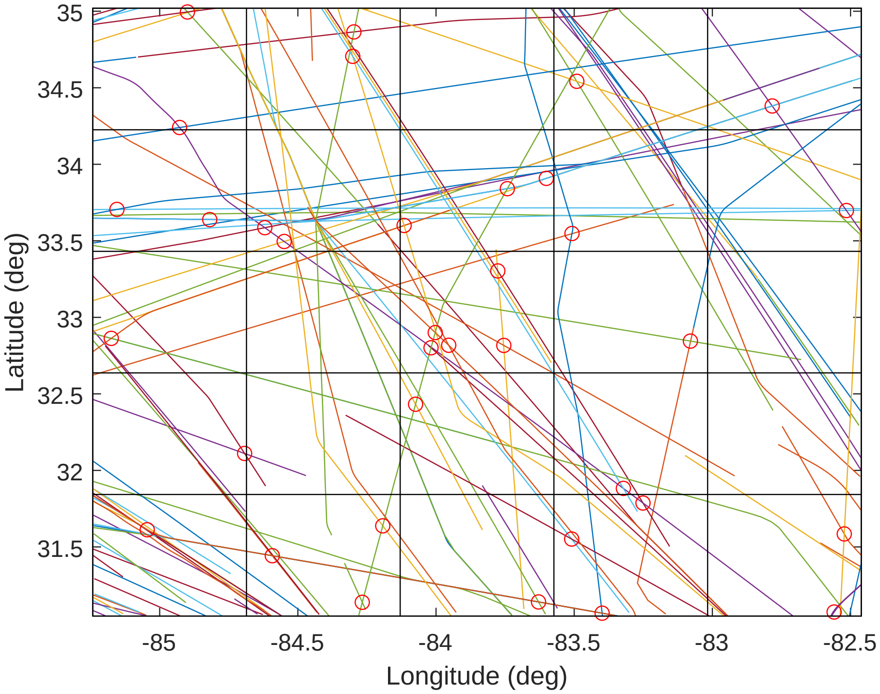


Fig. 12 ***Selected flights near Atlanta area with artificial LoS events***

C. Simulated Conflict Detection and Resolution Environment using Real Aircraft Data

In this section, we demonstrate the ability to perform CD&R on a data set of dense aircraft flying realistic flight trajectories. An ADS-B data set was collected over a square area from (31.04849° N, 85.242115° W) to (35.019862° N, 82.461039° W) over a two-hour period starting at 01:00 UTC on August 24^{th}, 2022. This data set involves 117 aircraft with no LOS conflicts using the standard 5 nm horizontal separation and 1000 ft vertical separation. Because this is a real data set, no collisions or LOS conflicts occurred. We artificially generate a series of LoS events by altering the trajectories of the 117 aircraft. Each trajectory was shifted uniformly to move the average altitude of all aircraft to the same flight level and shifted random flight paths in time to create 35 LoS events. The representative space was divided into a 5×5 uniform grid which represents the cohort mapping used to distribute vehicle trajectories.

Figure 12 shows the unresolved flight paths of the 117 aircraft. Segments of each aircraft trajectory that violate the LOS are shown with red circles indicating the location of minimum separation. As the flight paths are 4D trajectories with a time component, trajectories that cross in the spatial dimensions do not necessarily represent an LoS conflict unless they are also close temporally. Using the same set-up described in Section II, ADS-B data was converted to 4D trajectories using 33 ft maximum MSE per segment, trajectories divided into cohorts and Sturm sequencing carried out to look for LoS events. LoS events were sent to conflict resolution optimization. Conflict resolutions were run serially, without updating of previous resolutions to simulate the distributed capabilities of the algorithm. The total computational time to resolve all 35 conflicts was 25 minutes. The median two-aircraft conflict resolution time took 36 s (n=32), with a maximum of 1:20 min. The median three-aircraft conflict resolution time was 2:01 min (n=3), with a maximum of 4:08 min. The complete system resolution was checked and found to be free of LoS conflicts. As aircraft considered other trajectories within their respective cohorts, there were no post-resolution conflicts created that would need to be further resolved.

IV. Conclusion

In this paper, we present a method for distributed automated conflict detection and resolution using 4D 5th order polynomial splines. Waypoint manipulation exerts local control on the trajectory allowing it to resolve complex conflicts while not creating downstream or upstream conflicts. Separating the airspace into cohorts allows for distributed optimization run by aircraft within the cohort. This ensures reasonable real-time resolution in large-scale systems.

Our results demonstrate appropriate resolutions of hyper-realistic scenarios involving multiple aircraft as well as the resolution of artificially created conflicts using real flight data. Application of polynomial spline models to real-world aircraft data is shown to realistically interpolate between enhanced waypoints maintaining minimal error towards unused data points along the trajectory. The structure of waypoints and optimization algorithms allow for the addition of vehicle-dependant flight constraints such as total velocities and component acceleration. Our approach provides a rapid CD&R algorithm capable of handling multi-vehicle conflicts in large-scale scenarios.

Funding Sources

This project was partially supported by NSF SBIR Phase 1 project 2111827 and 4D Avionic Systems, LLC.

Acknowledgments

The authors would also like to especially thank Dr. Garth Thompson, Gabriel Orndorff, and Joseph Hoskinson for their advice and direction.

References

[1] Federal Aviation Administration, "NextGen - SESAR State of Harmonisation," *FAA State of Harmonisation Initiatives*, 2018, pp. 1–44.

[2] EU Parliment, "New generation European air traffic management system (SESAR)," 2016. URL `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=LEGISSUM:l24459`.

[3] Publications Office of the European Union, "Single European Sky ATM Research 3 Joint Undertaking, SESAR innovation pipeline – Air traffic management research and innovation – 2023 highlights," `https://data.europa.eu/doi/10.2829/966211`, 2024.

[4] Purwananto, Y., Wibisono, W., Fatichah, C., and Santoso, B. J., "A Heuristic Approach for Multi-Objective Aircraft Conflict

Detection and Resolution," *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, 2019, pp. 349–354. https://doi.org/10.1109/ICTS.2019.8850973.

[5] Vivona, R., Cate, K., and Green, S., *Comparison of Aircraft Trajectory Predictor Capabilities and Impacts on Automation Interoperability*, 2011, Chap. 6856. https://doi.org/10.2514/6.2011-6856, URL https://arc.aiaa.org/doi/abs/10.2514/6.2011-6856.

[6] Erzberger, H., "CTAS: Computer Intelligence for Air Traffic Control in the Terminal Area," *NASA TM-103959*, 1992.

[7] Fernández, C., Domínguez, R., Fernández-Llorca, D., Alonso, J., and Sotelo, M. A., "Autonomous Navigation and Obstacle Avoidance of a Micro-Bus," *International Journal of Advanced Robotic Systems*, Vol. 10, No. 4, 2013, p. 212. https://doi.org/10.5772/56125.

[8] Zahínos, R., Abaunza, H., Murillo, J. I., Trujillo, M. A., and Viguria, A., "Cooperative Multi-UAV System for Surveillance and Search & Rescue Operations Over a Mobile 5G Node," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 1016–1024. https://doi.org/10.1109/ICUAS54217.2022.9836167.

[9] Ju, C., and Son, H. I., "Multiple UAV Systems for Agricultural Applications: Control, Implementation, and Evaluation," *Electronics*, Vol. 7, No. 9, 2018. https://doi.org/10.3390/electronics7090162, URL https://www.mdpi.com/2079-9292/7/9/162.

[10] Ryan, H., Chandler, G., Santiago, C., Paglione, M., and Liu, S., "Evaluation of En Route Automation's Trajectory Generation and Strategic Alert Processing," *FAA Technical Note, Report No. DOT/FAA/TC-TN08/10*, 2008.

[11] T., P., Lee, P., Callantine, T., Smith, N., and Palmer, E., "Trajectory-Oriented Time-Based Arrival Operations: Results and Recommendations," *5th USA/Europe ATM R&D Seminar*, 2003.

[12] Li, Y., Cai, K., Yan, S., Tang, Y., and Zhu, Y., "Network-wide flight trajectories planning in China using an improved genetic algorithm," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–7. https://doi.org/10.1109/DASC.2016.7778051.

[13] Johnson, S. C., and Barmore, B., "NextGen Far-Term Concept Exploration for Integrated Gate-to-Gate Trajectory-Based Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 4355. https://doi.org/10.2514/6.2016-4355, URL https://arc.aiaa.org/doi/abs/10.2514/6.2016-4355.

[14] Zhang, W., Kamgarpour, M., Sun, D., and Tomlin, C. J., "Decentralized flight path planning for air traffic management," *Proceedings of the 2011 American Control Conference*, 2011, pp. 2137–2142. https://doi.org/10.1109/ACC.2011.5991156.

[15] Sturm, J. C. F., ""Mémoire sur la résolution des équations numériques"," *Bulletin des Sciences de Férussac*, Vol. 11, 1829, pp. 419–425.

[16] Thompson, J., and Zhang, X., "Coordinated flight control along a complex flight-path," *19th DASC. 19th Digital Avionics Systems Conference. Proceedings (Cat. No.00CH37126)*, Vol. 1, 2000, pp. 2A6/1–2A6/7 vol.1. https://doi.org/10.1109/DASC.2000.886903.

[17] Olive, X., "traffic, a toolbox for processing and analysing air traffic data," *Journal of Open Source Software*, Vol. 4, 2019, p. 1518. https://doi.org/10.21105/joss.01518.

[18] *Air Traffic Control*, Department of Transportation, Washington, DC, USA, 2024.

[19] Kuenz, A., "A global airspace model for 4D-trajectory-based operations," *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 3E3–1–3E3–9. https://doi.org/10.1109/DASC.2011.6096063.

[20] Shampine, L. F., "Accurate Numerical Derivatives in MATLAB." *ACM Trans. Math. Softw.*, Vol. 33, No. 4, 2007, p. 17.

[21] Martins, J., Sturdza, P., , and Alonso, J., "The ComplexStep Derivative Approximation." *ACM Trans. Math. Softw.*, Vol. 29, No. 3, 2003, pp. 245–262.

[22] Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M., "Bringing up OpenSky: A large-scale ADS-B sensor network for research," *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014, pp. 83–94. https://doi.org/10.1109/IPSN.2014.6846743.