AIAA Guidance, Navigation and Control Conference and Exhibit
20 - 23 August 2007, Hilton Head, South Carolina

AIAA 2007-6588

# Multiple Aircraft Deconflicted Path Planning with Weather Avoidance Constraints

Jessica J. Pannequin[*]
*University of California at Berkeley, CA 94720*

Alexandre M. Bayen[†]
*University of California at Berkeley, CA 94720*

Ian M. Mitchell[‡]
*University of British Columbia, Vancouver, BC, Canada V6T 1Z4*

Hoam Chung[§]
*University of California at Berkeley, CA 94720*

Shankar Sastry[¶]
*University of California at Berkeley, CA 94720*

**A nonlinear model predictive control based algorithm is presented for aircraft motion planning that will apply to converging flows of aircraft going through convective weather in the en route airspace. Two approaches for computing the objective function are explored and compared. The first involves a quadratic objective function while the second derives the objective function from the solution to the Hamilton-Jacobi equation, a nonlinear partial differential equation. For given static convective weather conditions and aircraft destinations, the solution to the Hamilton-Jacobi equation provides a value function that corresponds to the minimum travel time from any point in the NAS to the specified destination. In both cases, the optimal control sequence is computed over a fixed horizon by minimizing the aircraft objective function, subject to aircraft separation constraints, bounds on aircraft turning rates and convective weather avoidance constraints. This algorithm results in a set of locally optimal trajectories for the aircraft considered.**

## Nomenclature

| | |
|---|---|
| $T$ | Model Predictive Control horizon length, (units: seconds) |
| $\Delta t$ | Model Predictive Control time step, (units: seconds) |
| $n$ | Model Predictive Control number of steps in horizon |
| $N$ | Total number of aircraft considered |
| $\bar{\mathbf{x}}$ | $[(\bar{x}_1, \bar{y}_1, \bar{\theta}_1), ..., (\bar{x}_N, \bar{y}_N, \bar{\theta}_N)]$ State vector for $N$ aircraft in continuous time. |
| $\bar{\mathbf{u}}$ | $[\bar{u}_1, ..., \bar{u}_N]$ Turning rate vector for $N$ aircraft in continuous time. (units: radians/second). |
| $\mathbf{x}$ | $[(x_1, y_1, \theta_1), ..., (x_N, y_N, \theta_N)]$ States of $N$ aircraft in discrete time. |
| $\mathbf{u}$ | $[u_1, ..., u_N]$ Turning rate for $N$ aircraft in discrete time. (units: radians/second). |
| $x$ | $[x_1, ..., x_N] = [x_{1,0}, ..., x_{1,n-1}, ..., x_{N,0}, ..., x_{N,n-1}]$ $x$ position vector for $N$ aircraft in discrete time (units: kilometers) |
| $y$ | $[y_1, ..., y_N] = [y_{1,0}, ..., y_{1,n-1}, ..., y_{N,0}, ..., y_{N,n-1}]$ $y$ position vector for $N$ aircraft in discrete time (units: kilometers) |
| $\theta$ | $[\theta_1, ..., \theta_N] = [\theta_{1,0}, ..., \theta_{1,n-1}, ..., \theta_{N,0}, ..., \theta_{N,n-1}]$ heading vector for $N$ aircraft in discrete time (units: radians) |
| $x_{i,j}$ | $x$ position of aircraft $i$ at discrete time $j$, (units: kilometers) |
| $y_{i,j}$ | $y$ position of aircraft $i$ at discrete time $j$, (units: kilometers) |
| $\theta_{i,j}$ | Heading of aircraft $i$ at discrete time $j$, (units: radians) |
| $u_{i,j}$ | Turning rate of aircraft $i$ at discrete time $j$, (units: radians/second) |
| $V_i$ | Value function of aircraft $i$, encoding the minimum time to reach destination $i$, (units: seconds) |
| $v$ | Aircraft velocity, (units: kilometers/second) |
| *Subscript* | |
| $i$ | Aircraft index |
| $j$ | Discrete time index |

[*] M.S. Student, Electrical Engineering and Computer Sciences, University of California at Berkeley.

[†] Assistant Professor, Systems Engineering, Civil and Environmental Engineering, University of California at Berkeley. AIAA member.

[‡] Assistant Professor, Department of Computer Science, University of British Columbia, Canada.

[§] Post-Doctoral Researcher, Electrical Engineering and Computer Sciences, University of California at Berkeley. AIAA Member.

[¶] Professor, Electrical Engineering and Computer Sciences, University of California at Berkeley. AIAA Member.

# I. Introduction

In the current Air Traffic Control system, protocols centralized at the sector level are applied by Air Traffic Controllers to aircraft for conflict resolution.[1,2,3] These protocols prevent *Losses of Separation* (LOS), to enforce safety in the *National Airspace System* (NAS). In practice, and under normal weather conditions, these protocols rarely involve more than three or four aircraft at a time, due to flow management techniques applied at the *Air Route Traffic Control Center* (ARTCC) level, to regulate the flow of aircraft in the NAS. However, in the presence of convective weather, aircraft flows are disrupted and are sometimes constrained to converge through narrow windows (in space and time) of the airspace, triggering an increase in potential LOS. Currently, these situations are solved manually by Air Traffic Controllers.[4,5,6,7,8]

The long term goal of this work is to generate fully decentralized algorithms for aircraft conflict resolution protocols that will apply to converging flows of aircraft going through convective weather, inspired by the work of Mao et al.[9] in geometrically unstructured environments. In this version of the work, only protocols centralized at the sector level will be considered. This corresponds to the current operational system in place in the NAS.[2]

An input of the problem is a grid, for which convective weather data is available at periodic time intervals. The problem of deconflicting aircraft paths through severe weather is approached using *Nonlinear Model Predictive Control* (NMPC). Assuming constant aircraft velocity, the optimal input sequence will be repeatedly computed over a horizon $T$ by minimizing an objective function for each aircraft subject to non-linear constraints. The input sequence represents the turning rate for each aircraft (equal to zero if the aircraft is traveling at fixed heading).

Two approaches for computing the objective function are explored and compared. The first one consists of a quadratic cost penalizing the remaining physical distance of all aircraft to their respective targets. The second option incorporates the solution to the *Hamilton-Jacobi partial differential equation* (HJ PDE),[10,11,12,13] for a set of convective weather data as well as a destination. The solution to the HJ PDE provides a value function which represents the approximate minimum travel time from any point in the NAS to the specified destination. The shortest travel time incorporates avoidance of convective weather patches. In both approaches, each destination in the NAS thus has a corresponding value function, which can be either defined by the physical distance to the destination or the minimum travel time to the destination with convective weather constraints. The set of value functions will be integrated in the objective function used in the NMPC routine. By penalizing the position at the end of the horizon in the objective function, the goal is to minimize deviations due to weather avoidance or aircraft conflict resolution. In other words, given that an aircraft deviates from its optimal trajectory to avoid convective weather or another aircraft, the resulting trajectory is such that the point at the end of the horizon is closest to its destination. The result gives locally optimal trajectories, while maintaining aircraft separation constraints and ensuring convective weather avoidance.

This article is organized as follows. In the next section, the formulation of the problem is summarized. Section III presents the algorithm used in this work. Section IV provides results of the simulation and a discussion of the limitations of the method used. Finally, conclusions are presented in section V.

# II. Problem Formulation

The goal of this work is to compute a set of locally optimal deconflicted trajectories for multiple aircraft in the presence of convective weather regions. In the present setting, these constraints will be considered static. The generalization of this technique to dynamic convective weather (with forecast) is one of the potential next steps of this work. These regions of convective weather become obstacles, which aircraft must avoid, as shown for two aircraft in the example of Figure 1.

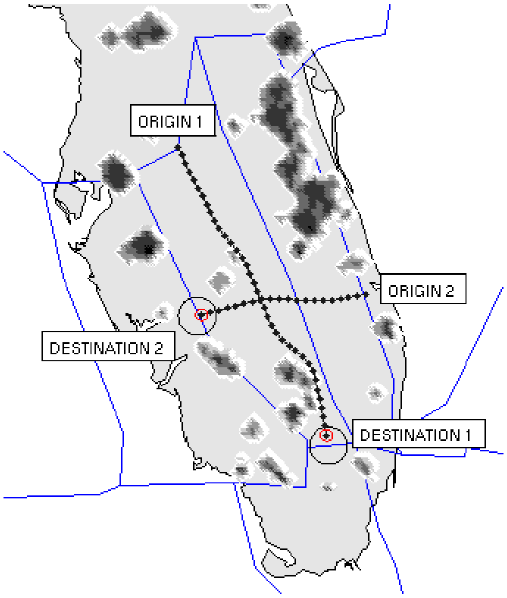
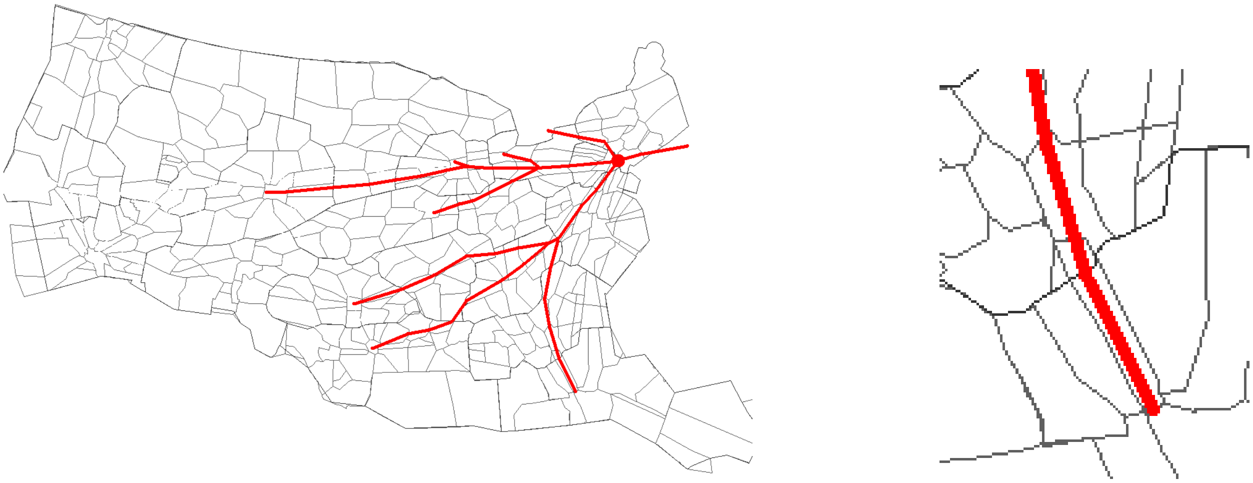


**Figure 1. Example of deconflicted trajectories for two aircraft and their respective origin-destination pairs over Florida.**

### II.A. Aircraft Routing

High altitude traffic in the en route airspace follows flow patterns, which have been extensively studied in the literature.[14,15,16,17,18] For the purpose of this work, we consider entry and departure points defined by flows in high altitude sector, which characterize high altitude traffic. In earlier work,[15,18,19] we have explained how to construct the origin-destination points based on flow analysis, as shown for the entire NAS for flows converging to LaGuardia airport in Figure 2a. Figure 2b represents the flows over Florida which will converge to LaGuardia airport. These graphs were obtained using aggregate modeling, for which each edge represents one aggregate flow with significant traffic in the NAS. In Figure 1, the origin-destination pairs represent the vertices of edges of the graph in Figure 2.



(a) Flow Network

(b) Subset of Flow Network over Florida

**Figure 2. Flow network converging to LaGuardia airport.[19]**

The locally optimal trajectories connecting the origin-destination pairs are computed by minimizing an objective function. In the first approach the objective function includes a quadratic cost which penalizes the remaining physical distance to the target (the destination disks on the map of Figure 1). The second approach incorporates a value function obtained from solving the HJ PDE. For each destination, the value function represents the minimum time to reach the destination disk. This minimum time function is commonly used in path planning and robotics to find an optimal trajectory of the system to the destination, given motion constraints due to the environment.[20,13] In both cases, the resulting locally optimal trajectories will not be constrained to these flows, but rather will take advantage of shorter routes while avoiding regions of convective weather. Traffic through sectors *ZMA64* and *ZMA65* over Florida will be considered, as shown in Figure 3, given that available weather data provides large regions of convective weather in those sectors, which makes them interesting to study.

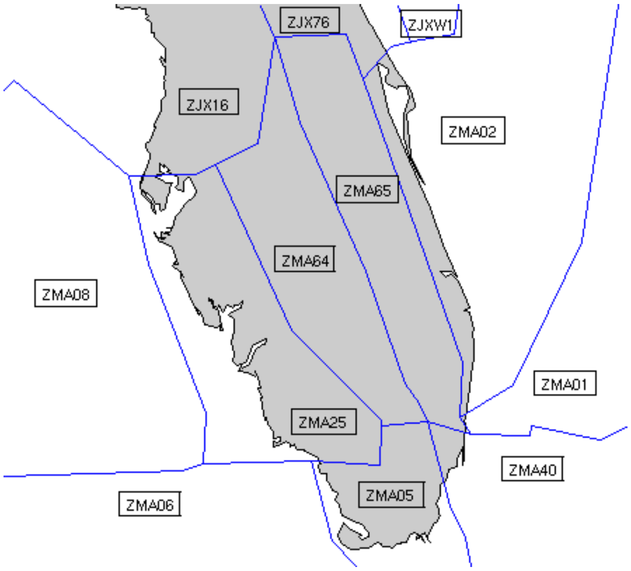


**Figure 3. High Altitude Sectors over Florida.**

## II.B. Constraints

The computation of aircraft optimal trajectories must take into account aircraft separation constraints and convective weather avoidance constraints that restrict their effective airspace. These constraints are encoded directly in the formulation of our problem. Previous work by Shim et al.,[21] modeled separation constraints between multiple autonomous helicopters as repulsive potential fields. The main advantage of hard coding constraints is the guarantee of resulting in a set of conflict-free trajectories if a feasible solution is found. Another advantage of using hard constraints rather than potential fields lies in the fact that fine tuning the repulsive potential fields parameters can be quite tedious. However one drawback of hard coded constraints is the added complexity of the optimization problem and thus an increase in the computation time.

### *II.B.1. Protected Zones*

One of the key requirements in air traffic control is to avoid LOS between aircraft. In high altitude sectors, above $29,000$ feet, this occurs when aircraft are separated by a distance less than 5 nautical miles horizontally or 1000 feet vertically. For the purpose of this project, aircraft motion is restricted to the $(x, y)$ plane. The proposed method can easily be extended to the full $(x, y, z)$ space. However, because of the layered nature of traffic, the present study is relevant for horizontal separation in the en route airspace, which is horizontally stratified.[22]

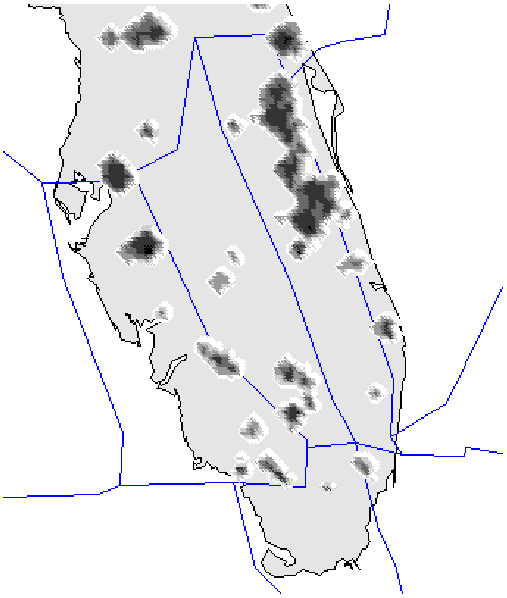


**Figure 4. Representation of Convective Weather over Florida on August 1st, 2005 at 8:14pm.**

### *II.B.2. Convective Weather*

In addition to the aircraft separation constraints, aircraft are prohibited from flying in regions with severe convective weather. Convective weather data is obtained from the *National Center for Atmospheric Research*

(NCAR). The specific data we are using is called *National Convective Weather Detection* (NCWD) which represents archived convective weather data. The NCWD data is obtained using the *Vertically Integrated Liquid* (VIL) algorithm, which converts weather radar reflectivity into a measure of liquid water content in a sample volume. It has been shown[23] that the amount of liquid water correlates well with the level of turbulence in a sample volume.

The data obtained from NCAR is in *Meteorological Data Volume* (MDV) format. In order to read the data into Matlab, it has to be converted to *network Common Data Form* (netCDF) format. The netCDF Matlab toolbox allows us to read and manipulate the convective weather files. Each file contains a matrix of 918 by 1830 elements for the whole continental United States and each entry covers an area of approximately 4 by 4 kilometers. A matrix of this type is available approximately every 5 minutes.

In addition to the VIL representation of the data, there exists corresponding discrete levels called *Video Integrator and Processor* (VIP), which range from 0 to 6. The conversion between the continuous VIL data and the discrete VIP levels is shown in Table 1. The third column shows the colormap for representing the VIP data, as shown in Figure 4. Pilots are prohibited from flying in regions in which the convective weather VIP index is greater than 3. In regions of convective weather, preventing LOS becomes more difficult as the effective airspace is reduced.

**Table 1. Conversion from VIL to VIP Data.**

| VIL ($Kg/m^2$) | 0.14 | 0.7 | 3.5 | 6.9 | 12.0 | 32.0 |
|---|---|---|---|---|---|---|
| VIP Level | 1 | 2 | 3 | 4 | 5 | 6 |
| Colormap (Figure: 4) | white | pale gray | light gray | medium gray | dark gray | black |

### II.C. Problem Statement

The problem solved in this work can be summarized as follows.

**Problem:**
Given the following:
1) $N$ aircraft in a subset of the NAS, with one origin-destination pair per aircraft.
2) Aircraft Dynamics.
3) Static convective weather data.
**Find**: for each aircraft, an optimal trajectory, from the corresponding origin to the destination while avoiding severe convective weather and other aircraft.

At this stage, only static weather is considered. The problem statement is detailed as follows. The continuous time dynamics of aircraft $i$ can be written as:

$$\begin{aligned} \dot{\bar{x}}_i(t) &= v\cos\bar{\theta}_i(t) \\ \dot{\bar{y}}_i(t) &= v\sin\bar{\theta}_i(t) \\ \dot{\bar{\theta}}_i(t) &= \bar{u}_i(t). \end{aligned} \tag{1}$$

The velocity $v$ of each aircraft is assumed to be constant throughout the portion of the en route airspace of interest. We are looking for the optimal input sequence $\bar{u}_i$ for each aircraft in our system given constraints described below to minimize an objective function, which will be defined next. We can assemble the individual aircraft dynamics into a single system dynamics $\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$. The variable $\bar{\mathbf{x}}$ represents the collection of state variables $(\bar{x}_i, \bar{y}_i, \bar{\theta}_i)$ in continuous time for all aircraft ($i \in [1, N]$). The variable $\bar{\mathbf{u}}$ is a measurable control input from $[0, T]$ to $\mathbb{R}^N$ and represents the collection of control sequences $\bar{u}_i$ in continuous time for each aircraft. $T$ represents the time horizon used in the NMPC framework. We also define $\mathbf{x_0}$ as the set of initial states (position and heading) for all aircraft (similarly we define $\mathbf{x_{i,0}}$ as the set of initial states for aircraft $i$). The minimization of objective function $J(\mathbf{x_0}, \bar{\mathbf{u}}(\cdot))$ for the entire system can be expressed as the minimization of the sum of objective functions $J_i(\mathbf{x_{i,0}}, \bar{u}_i(\cdot))$ of each individual aircraft, as shown in equation (2).

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & J(\mathbf{x_0}, \bar{\mathbf{u}}(\cdot)) = \sum_{i=1}^{N} J_i(\mathbf{x_{i,0}}, \bar{u}_i(\cdot)) \\ \text{s.t.} \quad & g(\mathbf{x_0}, \bar{\mathbf{u}}(\cdot)) \leq 0. \end{aligned} \tag{2}$$

We define $(\bar{x}_i^{\bar{u}_i(\cdot)}(\cdot), \bar{y}_i^{\bar{u}_i(\cdot)}(\cdot))$ as the continuous time $(x, y)$ trajectory of aircraft $i$ under input control $\bar{u}_i(\cdot)$ obtained by integration of equation (1). The optimization problem (2) is written explicitly in (3) below with the constraints of the problem.

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & \sum_{i=1}^{N} J_i(\mathbf{x_{i,0}}, \bar{u}_i(\cdot)) & & \\ \text{s.t.} \quad & -a \leq \bar{u}_i(t) \leq a & i \in [1, N] \quad & \forall t \in [0, T] \\ & (\bar{x}_i^{\bar{u}_i(\cdot)}(t) - \bar{x}_k^{\bar{u}_k(\cdot)}(t))^2 + (\bar{y}_i^{\bar{u}_i(\cdot)}(t) - \bar{y}_k^{\bar{u}_k(\cdot)}(t))^2 \geq r_{\min}^2 & (i, k) \in \mathcal{C} \quad & \forall t \in [0, T] \\ & (\bar{x}_i^{\bar{u}_i(\cdot)}(t), \bar{y}_i^{\bar{u}_i(\cdot)}(t)) \notin W & i \in [1, N] \quad & \forall t \in [0, T] \\ & \dot{\bar{x}}_i^{\bar{u}_i(\cdot)}(t) = v \cos \bar{\theta}_i^{\bar{u}_i(\cdot)}(t) & i \in [1, N] \quad & \forall t \in [0, T] \\ & \dot{\bar{y}}_i^{\bar{u}_i(\cdot)}(t) = v \sin \bar{\theta}_i^{\bar{u}_i(\cdot)}(t) & i \in [1, N] \quad & \forall t \in [0, T] \\ & \dot{\bar{\theta}}_i^{\bar{u}_i(\cdot)}(t) = \bar{u}_i(t) & i \in [1, N] \quad & \forall t \in [0, T], \end{aligned} \tag{3}$$

where we define the set $\mathcal{C} = \{(k, i) | i \in [1, N], k \in [1, N], i \neq k\}$. The first constraint incorporates limits on the turning rate of the aircraft. A value of $a = 2.65$ deg.s$^{-1}$ was used based on earlier work.[24] The second constraint states that all aircraft must be separated by a distance greater or equal to $r_{\min} = 5$ nautical miles. The third constraint expresses the fact that the trajectory of each aircraft must not enter regions of severe convective weather, represented by the set $W$. The last three constraints correspond to the dynamics of the aircraft. This formulation is standard in the path planning literature.[25]

# III. Algorithm

The algorithm used to compute the optimal trajectories is based on nonlinear model predictive control. Model predictive control is defined by the computation, at every user-defined time step $\triangle tc$, of the optimal control sequence over a finite horizon.[26] The current state of the system sets the initial conditions at each sampling point. One of the main advantages of this control scheme is its capacity to deal with nonlinearity subject to hard constraints[27] which makes it an attractive choice for multi-agent path planning.[28,29]

The overall algorithm is detailed below. First, the initialization parameters are set (Number of aircraft $N$, horizon length $T$, discretization time step $\triangle t$, MPC time step $\triangle tc$, origin-destinations pairs and aircraft speed $v$). The system dynamics are then discretized with time step $\triangle t$ and an initial guess $\mathbf{u_0}$ for the discrete control sequence $\mathbf{u}$ is provided to the solver. Next, the solver computes the objective function and constraints as well as the objective function gradient and constraints Jacobian repeatedly until it converges to an optimal solution for the control input $\mathbf{u}$ over the time interval $[0, T]$. While a (locally) optimal solution is preferred, a feasible solution is also acceptable because it guarantees a set of conflict free trajectories, which is the main concern of this work. However due to the nonlinearity of the problem, we are not guaranteed to find a feasible point,[30] nor a unique optimal solution because the problem is nonconvex. While in practice, feasible or optimal solutions are most often found, we need to keep this in mind to make the system robust to these numerical issues prior to implementing it on flying agents.

All aircraft trajectories are updated by applying a subsequence of the control $\mathbf{u}$ over the time interval [0, $\triangle tc$]. If an aircraft has reached its target by the end of the horizon, it is removed from the optimization routine to reduce computational complexity. The process then repeats in $\triangle tc$ time intervals until all aircraft have reached their destination. The details of the algorithm are detailed in the subsequent sections.

```
Initialization
  Set number of aircraft N, horizon length $T$, discretization time step $\triangle t$ and MPC time step $\triangle tc$.
  For all aircraft ($i = 1, \cdots, N$), set origin-destination pair and aircraft speed $v$.
  Discretize aircraft dynamics.
  Load VIP weather data.
Main Loop: Repeat until all aircraft have reached their destination.
  Solve discrete time nonlinear optimal control problem (8) on horizon $T$.
    Inputs:
      Initial guess $\mathbf{u_0}$ for control sequence $\mathbf{u}$.
      Objective function (9) and gradient of objective function (10) with respect to $\mathbf{u}$.
      Constraints (20),(24) and constraints Jacobian (21),(26) with respect to $\mathbf{u}$.
    Output:
      Optimal control sequence $\mathbf{u}$ over time horizon $[0, T]$.
  Advance to the next step on the horizon by $\triangle tc$ time intervals and update trajectories.
  Remove aircraft which have reached their target at the end of their horizon.
End of main loop.
Output: Optimal trajectories for set of $N$ aircraft from their origin to their destination.
```

### III.A. Dynamics Discretization

In order to compute the optimal control sequence, the time continuous aircraft dynamics must be discretized. The system dynamics $\mathbf{f}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t))$ shown in equation (1) is continuous and Lipzchitz continuous. Therefore the *Forward Euler Method* (FEM) can be applied to approximate the solution to the differential equations,[31] for $i \in [1, N]$, where aircraft $i$ has initial conditions $\mathbf{x_{i,0}} = (x_{i,0}, y_{i,0}, \theta_{i,0})$. We call $j$ the index corresponding to the discretization in time. Equation (4) shows the approximated position of aircraft $i$ at time step $j = 1$.

$$\begin{aligned} x_{i,1} &= x_{i,0} + \triangle t \cdot v \cos(\theta_{i,0}) \\ y_{i,1} &= y_{i,0} + \triangle t \cdot v \sin(\theta_{i,0}) \\ \theta_{i,1} &= \theta_{i,0} + \triangle t \cdot u_{i,0}. \end{aligned} \tag{4}$$

These results can be generalized to time steps $j \in [2, n-1]$ to find the trajectory of all aircraft ($i \in [1, N]$) in discrete time as shown in equations (5)-(7), which is an approximation of the continuous time solution.

$$x_{i,j} = x_{i,0} + \triangle t \cdot v \cos(\theta_{i,0}) + \triangle t \cdot v \sum_{k=0}^{j-2} \cos\left[\theta_{i,0} + \triangle t \sum_{r=0}^{k} u_{i,r}\right] \tag{5}$$

$$y_{i,j} = y_{i,0} + \triangle t \cdot v \sin(\theta_{i,0}) + \triangle t \cdot v \sum_{k=0}^{j-2} \sin\left[\theta_{i,0} + \triangle t \sum_{r=0}^{k} u_{i,r}\right] \tag{6}$$

$$\theta_{i,j} = \theta_{i,0} + \triangle t \sum_{r=0}^{j-1} u_{i,r}. \tag{7}$$

In discrete time, the entire input control sequence $\mathbf{u}$ is in defined as the vector $[u_1 \dots u_N]^T$, where $u_i$ corresponds to the sequence of control inputs applied to aircraft $i$ over the length of horizon $T$. It is in turn defined as $u_i = [u_{i,0} \dots u_{i,n-1}]^T$, where $u_{i,j}$ corresponds to the control applied to aircraft $i$ at time step $j$. Once the dynamics of all aircraft are discretized, the problem statement becomes the following in discrete time:

$$\begin{array}{lllll}
\min\limits_{\mathbf{u}} & J(\mathbf{x_0}, \mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x_{i,0}}, u_i) & & & \\
\text{s.t.} & -a \le u_{i,j} \le a & i \in [1, N] & j \in [0, n-1] & \\
 & (x_{i,j} - x_{k,j})^2 + (y_{i,j} - y_{k,j})^2 \ge r_{\min}^2 & (i,k) \in \mathcal{C} & j \in [0, n-1] & \\
 & (x_{i,j}, y_{i,j}) \notin W & i \in [1, N] & j \in [0, n-1] & (8) \\
 & x_{i,j+1} = x_{i,j} + \triangle t \cdot v \cos(\theta_{i,j}) & i \in [1, N] & j \in [0, n-1] & \\
 & y_{i,j+1} = y_{i,j} + \triangle t \cdot v \sin(\theta_{i,j}) & i \in [1, N] & j \in [0, n-1] & \\
 & \theta_{i,j+1} = \theta_{i,j} + \triangle t \cdot u_{i,j} & i \in [1, N] & j \in [0, n-1]. &
\end{array}$$

In order to eliminate the equality constraints in the system, which are more difficult to enforce than inequality contraints,[31] the optimization routine is performed over the input $\mathbf{u}$ rather than over the states of the aircraft $\mathbf{x}$. This can easily be done because all discrete states $x_{i,j}$, $y_{i,j}$ and $\theta_{i,j}$ for $i \in [1, N]$ and $j \in [0, n-1]$ are uniquely determined by the input sequence $\mathbf{u}$. As a result, the objective function and the constraints are all expressed only in terms of $\mathbf{u}$ and the initial conditions $\mathbf{x_0} = [\mathbf{x_{1,0}}, \cdots, \mathbf{x_{N,0}}]$ of the system.

## III.B. Objective Function

### *III.B.1. First Approach: Quadratic Objective Function*

The first approach consists of including a quadratic cost, which penalizes the distance between the position at the end of the horizon and the center of the corresponding target destination$(x_i^t, y_i^t)$ for each aircraft, as shown in equation (9). In addition, an input cost of the form $1/2u^T Ru$ is included in the objective function where $R$ is a diagonal matrix whose terms are proportional to the remaining distance of each aircraft to its corresponding destination. Including this input cost serves to reduce unnecessary oscillations in the control, as we will see in section IV.

$$J(\mathbf{x_0}, \mathbf{u}) = \sum_{i=1}^{N} (x_{i,n-1}(\mathbf{x_{i,0}}, u_i) - x_i^t)^2 + (y_{i,n-1}(\mathbf{x_{i,0}}, u_i) - y_i^t)^2 + \frac{1}{2}\mathbf{u}^T R\mathbf{u}. \qquad (9)$$

The point at the end of the horizon is penalized to ensure that deviations due to aircraft conflict resolution and convective weather avoidance are small. The dependency $x_{i,n-1}(\mathbf{x_{i,0}}, \mathbf{u_i})$ has been added to signify the implicit dependency of $x_{i,n-1}$ in the initial states $\mathbf{x_{i,0}} = (x_{i,0}, y_{i,0}, \theta_{i,0})$ and input $u_i$ from equation (5) and similarly for $y_{i,n-1}$ from equation (6). In later parts of the article, it may be omitted for simplicity of notation.

In order to achieve better results,[32,30] the gradient of the objective function, shown in equation (10), is provided to the solver. If the gradient is not provided, the solver computes its own numerical approximation of the gradient, which is less accurate. In addition to increasing the computation time, omitting the gradient can thus lead to inaccuracies in the computed optimal solution.

$$\begin{aligned}
\frac{\partial J(\mathbf{x_0}, \mathbf{u})}{\partial \mathbf{u}} &= 2\left(\sum_{i=1}^{N} (x_{i,n-1} - x_i^t) \cdot \frac{\partial x_{i,n-1}}{\partial \mathbf{u}} + (y_{i,n-1} - y_i^t) \cdot \frac{\partial y_{i,n}}{\partial \mathbf{u}}\right) + R\mathbf{u} \qquad (10) \\
&= 2\begin{bmatrix} \sum_{i=1}^{N} (x_{i,n-1} - x_i^t) \cdot \frac{\partial x_{i,n-1}}{\partial u_{1,0}} + (y_{i,n-1} - y_i^t) \cdot \frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N} (x_{i,n-1} - x_i^t) \cdot \frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + (y_{i,n-1} - y_i^t) \cdot \frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}.
\end{aligned}$$

From (5) and (6) we obtain closed form expressions for the partial derivatives of $(x, y)$ with respect to the control sequence $\mathbf{u}$ shown in equations (11)-(14). The non-zero terms correspond to $p = i \in [1, N]$ and $q \le (j-2)$ where $j \in [0, n-1]$ and $q \in [0, n-1]$. For all other values of $p$ and $q$, the partial derivatives are equal to zero.

$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = -\triangle t^2 \cdot v \sum_{k=q}^{j-2} \sin\left[\theta_{i,1} + \triangle t \sum_{r=0}^{k} u_{i,r}\right] \quad p = i \quad \text{and} \quad q \leq (j-2) \tag{11}$$

$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = \triangle t^2 \cdot v \sum_{k=q}^{j-2} \cos\left[\theta_{i,1} + \triangle t \sum_{r=0}^{k} u_{i,r}\right] \quad p = i \quad \text{and} \quad q \leq (j-2) \tag{12}$$

$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = 0 \qquad p \neq i \quad \text{or} \quad q > (j-2) \tag{13}$$

$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = 0 \qquad p \neq i \quad \text{or} \quad q > (j-2). \tag{14}$$

### *III.B.2. Second Approach: Hamilton-Jacobi Derived Objective Function*

The second approach for computing the objective function consists of replacing the quadratic cost penalizing the remaining distance to the target for each aircraft by a value function $V_i(x_{i,n-1}(\mathbf{x_{i,0}}, u_i), y_{i,n-1}(\mathbf{x_{i,0}}, u_i))$. This value function represents the cost-to-go penalty corresponding approximately to the minimum time to reach the user-specified target of aircraft $i$, as shown in equation (15). In practice, the cost-to-go is commonly used by ATC to provide aircraft with wind optimal routes.[33]

$$J(\mathbf{x_0}, \mathbf{u}) = \sum_{i=1}^{N} V_i(x_{i,n-1}(\mathbf{x_{i,0}}, u_i), y_{i,n-1}(\mathbf{x_{i,0}}, u_i)) + \frac{1}{2}\mathbf{u}^T R \mathbf{u}. \tag{15}$$

This value function is computed by solving a static HJ PDE, (16), in a computational domain $\Omega$ representing the airspace.

$$\begin{aligned} \max\nolimits_{u_i}(\nabla V_i(x,y) \cdot f(x,y,u_i) - \ell(x,y)) = & \ 0 \qquad \text{in } \Omega \setminus \mathcal{T}_i \\ V_i(x,y) = & \ 0 \qquad \text{in } \mathcal{T}_i. \end{aligned} \tag{16}$$

In this work, the target set $\mathcal{T}_i$ is chosen as a small circle centered around the destination of an aircraft (see Figure 1). If $\ell(x,y) = 1$, then $V_i(x,y)$ is simply the minimum time to reach $\mathcal{T}_i$ from $(x,y)$ without taking into account convective weather regions. In order for the value function to discourage travel through regions with convective weather, these regions are penalized with a higher cost. For the present study, $\ell(x,y) = \min(k^{\mathrm{VIP}(x,y)}, \ell_{\max})$, where $\mathrm{VIP}(x,y)$ corresponds to the convective weather index at $(x,y)$ and $k$ is some small constant (the examples use $k = 4$). The value $\ell_{\max}$ is used to express the fact that above a threshold of 3 for VIP readings, all regions are equally penalized because they are forbidden (the examples use $\ell_{\max} = k^3$). As a result, in the present work $V_i(x,y)$ is a cost to go function related but not exactly equal to the time to reach the destination. A separate $V_i(x,y)$ is computed for each destination disk (see Figure 1), therefore indexed by $i$ (destination for aircraft $i$).

Algorithms for directly solving HJ PDEs similar to (16) exist,[34,13] however, their generalization to time-dependent data and other vehicle dynamics is not trivial and implementations are not publicly available. Instead, the value functions $V_i(x,y)$ were computed using the publicly released code in the *Toolbox of Level Set Methods*.[35]

As was done for the quadratic objective function, the gradient of the objective function is provided to the solver to increase the accuracy of the results. Equation (17) details the gradient computation.

$$\begin{aligned} \frac{\partial J(\mathbf{x_0}, \mathbf{u})}{\partial \mathbf{u}} &= \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial x_{i,n-1}} \frac{\partial x_{i,n-1}}{\partial \mathbf{u}} + \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial y_{i,n-1}} \frac{\partial y_{i,n-1}}{\partial \mathbf{u}} + R\mathbf{u} \\ &= \begin{bmatrix} \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial x_{i,n-1}} \frac{\partial x_{i,n-1}}{\partial u_{1,0}} + \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial y_{i,n-1}} \frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial x_{i,n-1}} \frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + \frac{\partial V_i(x_{i,n-1}, y_{i,n-1})}{\partial y_{i,n-1}} \frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}, \end{aligned} \tag{17}$$

where $\partial V_i(x_{i,n-1}, y_{i,n-1})/\partial x_{i,n-1}$ should be $\frac{\partial V_i(x,y)}{\partial x}|_{(x_{i,n-1}, y_{i,n-1})}$ (similarly for $\partial V_i(x_{i,n-1}, y_{i,n-1})/\partial y_{i,n-1}$). This slight abuse of notation is used for simplicity. The main difference with the previous case lies in the fact that the value function $V_i(x_i, y_i)$ computed with the *Toolbox of Level Set Methods* cannot be expressed in an analytic form, as is almost always the case when dealing with real world data. Therefore, the partial derivatives $\partial V_i(x_{i,n-1}, y_{i,n-1})/\partial x_{i,n-1}$ and $\partial V_i(x_{i,n-1}, y_{i,n-1})/\partial y_{i,n-1}$ have to be be approximated numerically. The partial derivatives of the positions $(x, y)$ with respect to the control sequence $\mathbf{u}$ are computed using equations (11)-(14). The resulting gradient of the objective function thus contains numerical approximations. This poses a significant problem because the solver computes the Hessian to the Lagrangian using the gradient provided by the user. If these gradients are not accurate, the second order derivatives can be meaningless, which can lead to inaccurate results. The following section details the results of these two methods.

### *III.B.3. Objective Function Implementation: Comparison and Results*

Table 2 summarizes the advantages and drawbacks of both methods. The method using a quadratic objective function exhibits a well defined gradient, which shortens the computation time and allows the system to scale very well. One of the drawbacks of this method is that we do not possess a global view of the environment, which causes the resulting trajectories to be sub-optimal.

In the case when the objective function is derived from the solution the HJ PDE, several complications arise from the necessity to approximate the gradient numerically. These approximations cause the computation time to increase significantly as the solver exhibits difficulties when trying to converge to a solution. In addition, these numerical approximations cause the system to be sensitive to changes in initial aircraft position.

However, another advantage of the HJ PDE method is that winds can easily be encoded in the value function computation, which is crucial because aircraft typically fly wind optimal routes, which can be significantly different from shortest physical paths. But before we can successfully implement the HJ PDE derived objective function and include the wind, we must first find means to compute the gradients more accurately, possibly by using advanced numerical differentiation methods. Another solution could be to use first order optimization algorithms. Both of these options will likely cause the computation time to increase significantly.

**Table 2. Comparison between objective function approaches.**

| | Quadratic Objective Function | Objective Function derived from solution to HJ PDE |
|---|---|---|
| Advantages | • Gradient of objective function is well defined.<br>• Shorter computation time.<br>• Scales well. | • Allows for global view of airspace.<br>• Easily extended to include wind profile. |
| Drawbacks | • Restricted to local view of airspace. | • Numerically approximated gradient, which leads to:<br>- Longer computation time.<br>- Sensitive to changes in aircraft initial position. |

## III.C. Constraints

As was done for the objective function and its gradient, we provide the solver with the constraints $c$ of the system as well as the constraints Jacobian $dc$ with respect to the control variable $\mathbf{u}$. The constraints are supplied in the form of a vector $c = [c_1 \cdots c_{N_c}]$ where $c_i(\mathbf{x_0}, \mathbf{u}) \leq 0, \forall i \in [1, N_c]$ where $N_c$ represents the total number of constraints in the system. The constraints Jacobian $dc$, supplied to the solver is a matrix of the form:

$$dc = \begin{bmatrix} \frac{\partial c_1}{\partial u_{1,0}} & \cdots & \frac{\partial c_{N_c}}{\partial u_{1,0}} \\ & \vdots & \\ \frac{\partial c_1}{\partial u_{N,n-1}} & \cdots & \frac{\partial c_{N_c}}{\partial u_{N,n-1}} \end{bmatrix} \tag{18}$$

### *III.C.1. Input Constraints*

The first constraint in (8) corresponds to a bound on the control variable $\mathbf{u}$. Physically, this represents the maximum feasible turning rate of the aircraft at cruising speed. It can be explicitely written as the following constraint.

$$-a \leq u_{i,j} \leq a \quad \forall i \in [1, N] \quad \forall j \in [0, n-1]. \tag{19}$$

Due to the simplicity of this constraint and the functionality of the solver,[30] it is enforced by simply specifying a lower and upper bound on the decision variable $\mathbf{u}$ when the optimization problem is defined, rather than including it in the constraints vector $c$. As result, it is also not included in the constraints Jacobian $dc$.

### *III.C.2. Collision Avoidance Constraints*

The second constraint in (8) represents the required inter-aircraft minimum horizontal separation. Let $c^{\text{ca}}_{i,j,k}(\mathbf{x_0}, \mathbf{u})$ be defined as the constraint enforcing a safe separation between aircraft $i$ and $k$ at time $j$ as shown in equation (20).

$$c^{\text{ca}}_{i,j,k}(\mathbf{x_0}, \mathbf{u}) \triangleq r^2_{\min} - (x_{i,j} - x_{k,j})^2 + (y_{i,j} - y_{k,j})^2 \leq 0 \quad \forall (i,k) \in \mathcal{C} \quad \forall j \in [0, n-1]. \tag{20}$$

The values $x_{i,j}$ and $y_{i,j}$ are defined by equations (5) and (6). Next, the Jacobian of the constraints vector must be provided to the solver. If we look at the partial derivative of one constraint $c^{\text{ca}}_{i,j,k}$, with respect to the control sequence $\mathbf{u}$, we obtain:

$$\frac{\partial c^{\text{ca}}_{i,j,k}(\mathbf{x_0}, \mathbf{u})}{\partial \mathbf{u}} = -2 \begin{bmatrix} \left(\frac{\partial x_{i,j}}{\partial u_{1,0}} - \frac{\partial x_{k,j}}{\partial u_{1,0}}\right)(x_{k,j} - x_{i,j}) + \left(\frac{\partial y_{i,j}}{\partial u_{1,0}} - \frac{\partial y_{k,j}}{\partial u_{1,0}}\right)(y_{k,j} - y_{i,j}) \\ \vdots \\ \left(\frac{\partial x_{i,j}}{\partial u_{N,n-1}} - \frac{\partial x_{k,j}}{\partial u_{N,n-1}}\right)(x_{k,j} - x_{i,j}) + \left(\frac{\partial y_{i,j}}{\partial u_{N,n-1}} - \frac{\partial y_{k,j}}{\partial u_{N,n-1}}\right)(y_{k,j} - y_{i,j}) \end{bmatrix} \tag{21}$$

The dependency on $(\mathbf{x_{i,0}}, u_i)$ of $x_{i,j}$ and $y_{i,j}$ has been omitted for clarity. The values $\partial x_{i,j}/\partial u_{p,q}$ and $\partial y_{i,j}/\partial u_{p,q}$ are computed using equations (11)-(14). A column vector is obtained for each constraint to construct the Jacobian for collision avoidance constraints.

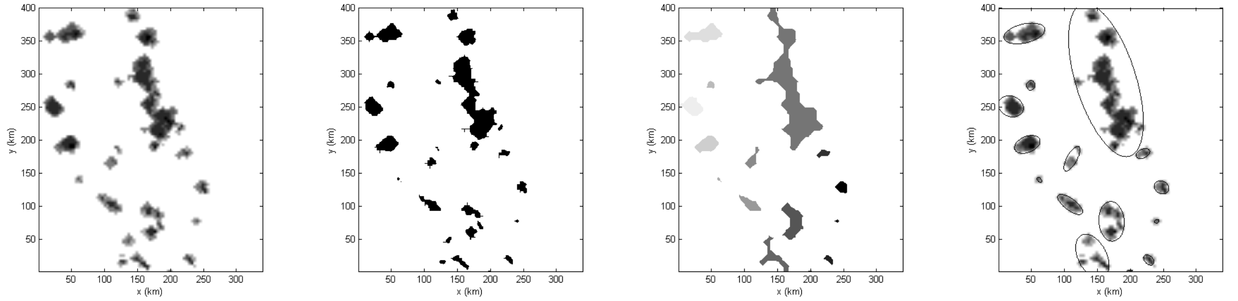
### *III.C.3. Convective Weather Avoidance Constraints:*

Pilots are not allowed to fly through regions where the VIP convective weather index is greater or equal to 3, expressed by the third constraint in (8). In order to enforce this set of constraints, we clustered convective weather regions which are in close proximity to one another in order to enclose them in minimum area ellipses. The process is detailed in Figure 5. Let $c^{\text{w}}_{i,j,k}(\mathbf{x_0}, \mathbf{u})$ be defined as the constraint enforcing aircraft $i$ to avoid weather cluster $k$ at time $j$, as shown in equation (22).

$$c^{\text{w}}_{i,j,k}(\mathbf{x_0}, \mathbf{u}) \triangleq 1 - \frac{(x'_{i,j} - x^{\text{w}}_k)^2}{a_k^2} - \frac{(y'_{i,j} - y^{\text{w}}_k)^2}{b_k^2} \leq 0 \quad \forall i \in [1, N] \quad \forall j \in [0, n-1] \quad \forall k \in [1, N_{\text{w}}]. \tag{22}$$

$N_{\text{w}}$ is defined as the number of clustered convective regions. The pair $(x^{\text{w}}_k, y^{\text{w}}_k)$ corresponds to the center of ellipse $k$ with major axis $a_k$ and minor axis $b_k$. The position $(x'_{i,j}, y'_{i,j})$ is related to $(x_{i,j}, y_{i,j})$ through a rotation matrix determined by the corresponding ellipse orientation, as shown in equation (23). The angle $\theta_k$ gives the orientation angle of the major axis of ellipse $k$ with respect to the $x$ axis.

$$\begin{bmatrix} x_{i,j} \\ y_{i,j} \end{bmatrix} = \begin{bmatrix} x^{\text{w}}_k \\ y^{\text{w}}_k \end{bmatrix} + \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \begin{bmatrix} x'_{i,j} - x^{\text{w}}_k \\ y'_{i,j} - y^{\text{w}}_k \end{bmatrix} \tag{23}$$



(a) Weather data.

(b) Binary representation of severe convective weather regions which must be avoided.

(c) Clustering of convective weather regions in close proximity.

(d) Elliptical boundary enclosing convective weather regions of test weather data.

**Figure 5. Weather Data Clustering.**

Using equation (23), we can write constraint $c^{\mathrm{w}}_{i,j,k}(\mathbf{x_0},\mathbf{u})$ in terms of the position $(x_{i,j}, y_{i,j})$ of aircraft $i$, as shown in equation (24).

$$1 - \frac{[(x_{i,j}-x^{\mathrm{w}}_k)\cos(\theta_k) + (y_{i,j}-y^{\mathrm{w}}_k)\sin(\theta_k)]^2}{a_k^2} + \frac{[(y_{i,j}-y^{\mathrm{w}}_k)\cos(\theta_k) - (x_{i,j}-x^{\mathrm{w}}_k)\sin(\theta_k)]^2}{b_k^2} \leq 0. \tag{24}$$

If we look at the partial derivative of constraint $c^{\mathrm{w}}_{i,j,k}$ with respect to the control $\mathbf{u}$, we obtain:

$$\begin{aligned}\frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial \mathbf{u}} &= \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial \mathbf{u}} + \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial \mathbf{u}} &(25)\\ &= \begin{bmatrix} \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{1,0}} + \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{1,0}} \\ \vdots \\ \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{N,n-1}} + \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{N,n-1}} \end{bmatrix}, &(26)\end{aligned}$$

where

$$\begin{aligned}\frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}} &= -\frac{2\cos(\theta_k)}{a_k^2}\left[(x_{i,j}-x^{\mathrm{w}}_k)\cos(\theta_k) + (y_{i,j}-y^{\mathrm{w}}_k)\sin(\theta_k)\right] \\ &\quad + \frac{2\sin(\theta_k)}{b_k^2}\left[(y_{i,j}-y^{\mathrm{w}}_k)\cos(\theta_k) - (x_{i,j}-x^{\mathrm{w}}_k)\sin(\theta_k)\right] &(27)\\ \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}} &= -\frac{2\sin(\theta_k)}{a_k^2}\left[(x_{i,j}-x^{\mathrm{w}}_k)\cos(\theta_k) + (y_{i,j}-y^{\mathrm{w}}_k)\sin(\theta_k)\right] \\ &\quad - \frac{2\cos(\theta_k)}{b_k^2}\left[(y_{i,j}-y^{\mathrm{w}}_k)\cos(\theta_k) - (x_{i,j}-x^{\mathrm{w}}_k)\sin(\theta_k)\right]. &(28)\end{aligned}$$

The values $\partial x_{i,j}/\partial u_{p,q}$ and $\partial y_{i,j}/\partial u_{p,q}$ are computed using equations (11)-(14). A column vector is obtained for each constraint to construct the weather avoidance constraint Jacobian.

# IV. Results

One of the challenging aspect of this work was to generate an efficient implementations of the proposed algorithms. We decided to use the Tomlab software. Tomlab is an optimization environment which can interface with Matlab.[32] The two main advantages of this package is the reduction of computational time as well as the added access to different types of solvers. This final version using the SNOPT solver,[30] proved to be faster, more robust to changes in initial aircraft positions and provide better scalability. The results detailed in the following sections were obtained using this setup. Moreover, all results shown were computed using the quadratic objective function.

### IV.A. Conflict Resolution

The collision avoidance portion of the algorithm was first tested in an environment without any convective weather. Results with 16 aircraft for different initial conditions and horizon lengths are shown in Figure 6. The simulation parameters for the four cases are shown in Table 3. In all results, one unit on the axis of Figure 6 corresponds to a distance of 1 km. Thus, given that the speed of the aircraft, at cruising altitude, is set to 230 m.s$^{-1}$, based on previous work,[24] the travel time of each aircraft is approximately 1 hour when the effective aircraft considered covers a region of 800 × 800 km. The approximate travel time is shortened to approximately 30 min in the cases when the airspace is reduced to $400 \times 400$km, as shown in Figures 6c-d.

**Table 3. MPC Simulation Parameters for Collision Avoidance Results.**

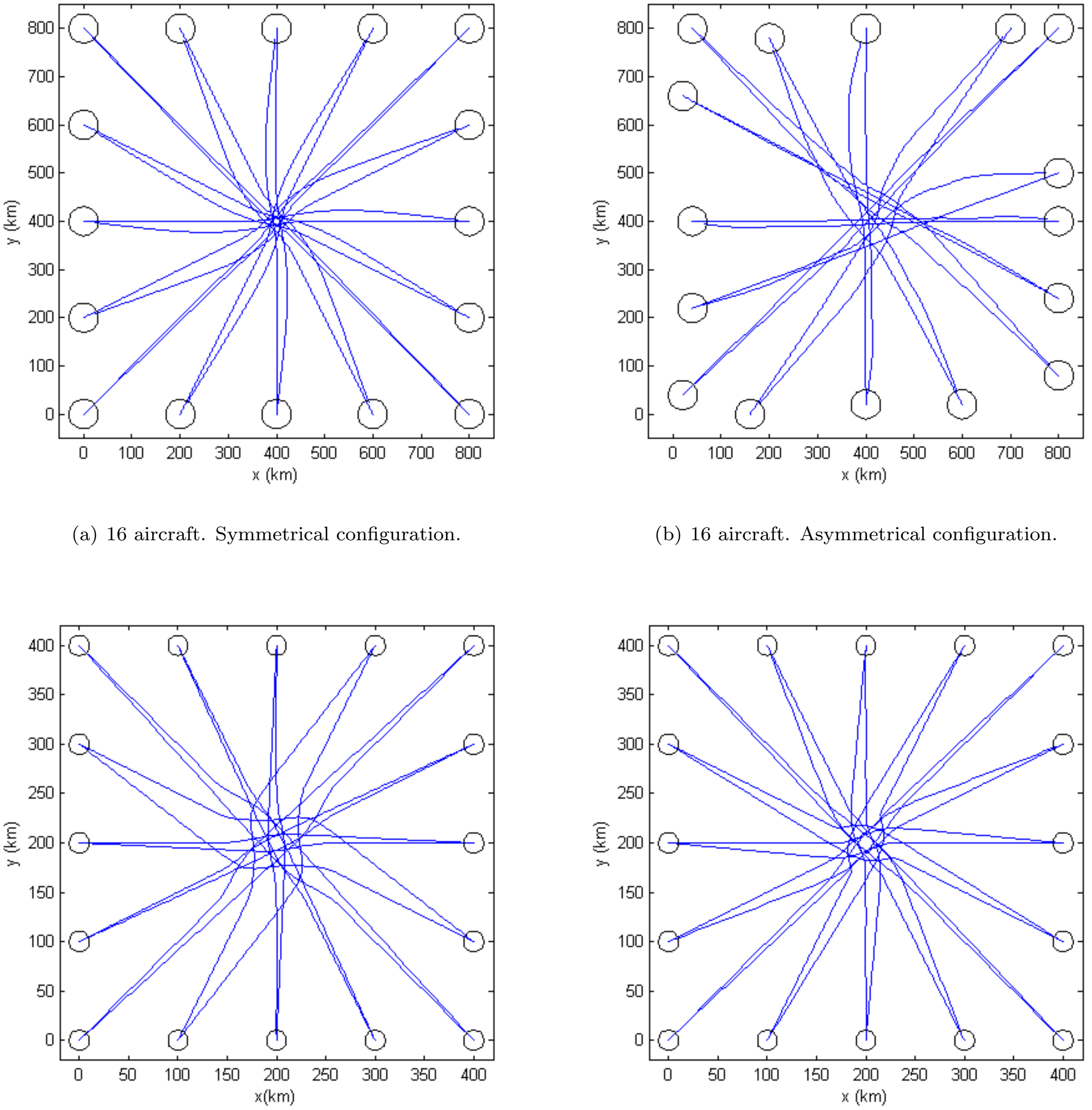
| Figure | $T$ (s) | $\triangle t$ (s) | $v \cdot \triangle t$ (nm) | $\triangle tc$ (s) | Effective Airspace (km$^2$) |
|---|---|---|---|---|---|
| 6a | 256 | 16 | 1.98 | 32 | 800 × 800 |
| 6b | 256 | 16 | 1.98 | 32 | 800 × 800 |
| 6c | 256 | 16 | 1.98 | 32 | 400 × 400 |
| 6d | 128 | 16 | 1.98 | 32 | 400 × 400 |

In the case of Figure 6a, the aircraft are set up in an approximate circular configuration such that they would all potentially collide at the same point. The same simulation was repeated for an asymmetrical configuration as shown in Figure 6b. These two simulations were inspired by the 16 maneuvers results shown in earlier work by Hu et al.[22] When comparing the results of Figures 6a-b with previous results,[22] fewer oscillations in the control parameter as well as less abrupt changes in heading can be observed in the MPC case. Figure 6c displays the same first simulation but with a reduction in the effective airspace to $400 \times 400$. In these first three cases, all MPC iterations exit either because an optimal or a feasible solution has been found. Therefore we are guaranteed that all constraints are respected and no LOS have occurred. While an optimal solution is preferred, a feasible solution is satisfactory because the main concern is to avoid LOS and collisions with severe convective weather.

The simulation shown in Figure 6c was repeated with a horizon reduced by half from 256 seconds to 128 seconds. The result is displayed in Figure 6d. While no constraint violations occurred, it is important to note that in this case, one MPC iteration resulted in the error message *Too many iterations*. All other iterations for the same simulation exited because an optimal or feasible point had been found. Thus while we may have been fortunate to avoid an LOS in this case, such an error message does not guarantee feasibility. The horizon length is the only difference between the simulations shown in Figures 6c and d. This shows that the length of the horizon is an important tuning parameter for the MPC framework. It is necessary to remember that since we are dealing with a nonlinear framework and constraints, we are not guaranteed to find a solution. However most of the time, given the correct MPC tuning parameters, optimal or feasible solutions are found.

Figure 7 details the trajectory displayed in Figure 6c. The circle around the current position of each aircraft corresponds to the protected zone of the aircraft. The actual horizontal separation constraint between aircraft is 5 nautical miles which corresponds to 9.26km. This type of test is standard in the multiple vehicle motion coordination literature.[36,22,21,37,3,38]

In addition to choosing the appropriate horizon length, the value of the matrix $R$ in the input cost term $1/2u^T Ru$ is an important tuning parameter for solving the optimization problem. Figure 8 shows the result for three different values of $R$ for the simulation shown in Figure 6c where the system is comprised of 16 aircraft. The horizon length chosen is 256 seconds with a time step $\triangle t = 16$ seconds and a control update interval $\triangle tc = 32$ seconds. Figure 8a displays the resulting trajectories for no input cost. While no violations occur and all aircraft reach their destination, the trajectories appear chaotic with many unnecessary loops. It appears that introducing the input cost penalizes these loops, which are no longer seen when a proper input cost is included, as shown in Figure 8c, where the value of $R$ is chosen to be $10^2$ times the remaining distance to the target. However when the value of $R$ becomes larger, the input cost dominates the cost penalizing the remaining distance of an aircraft to the target. Thus after deviating to avoid other aircraft, each aircraft no longer has an incentive to turn back fast enough towards its target and often misses is completely as shown in Figure 8b, where the value of $R$ is $10^4$ times the remaining distance to target. Thus is it necessary to perform careful parameter tuning prior to implementing the system.



(a) 16 aircraft. Symmetrical configuration.

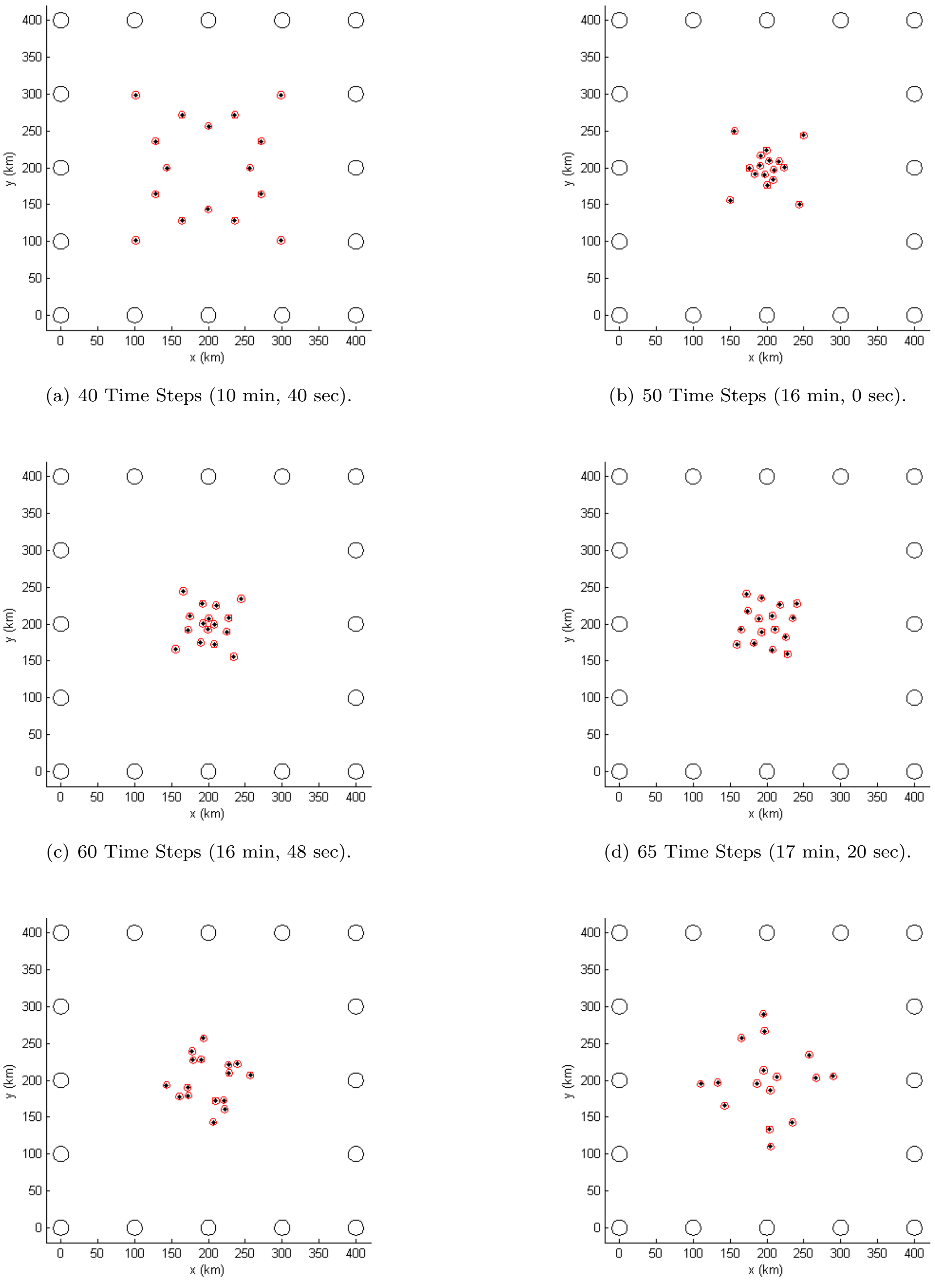
(b) 16 aircraft. Asymmetrical configuration.

(c) 16 aircraft. Symmetrical configuration in smaller environment.

(d) 16 aircraft. Symmetrical configuration in smaller environment and decreased horizon length.

**Figure 6. Trajectories generated for 16 aircraft without convective weather avoidance. The larger circles correspond to the aircraft destinations. (One unit on the $(x, y)$ axes corresponds to 1km).**

One of the main challenges in the implementation of the algorithm was the minimization of the computation time in view of using it in real-time on actual aircraft in the future. All simulations were performed on a Pentium 4 processor with a 3.2GHz CPU and 504MB of RAM. Table 4 shows results for the computation time for different numbers of aircraft in the system, in the absence of weather avoidance constraints. The computation times displayed correspond to the time needed to converge to a solution $\mathbf{u}$ over the horizon $T$. This process repeats until all aircraft have reached their destination. However, we are interested in the time required to converge to a solution because it must be smaller than the control update interval $\triangle tc$ in order to implement the algorithm in real time. Three computation times are shown, the minimum, maximum and average time required to converge to a solution. As can been seen with these results, when the system



(a) 40 Time Steps (10 min, 40 sec).

(b) 50 Time Steps (16 min, 0 sec).

(c) 60 Time Steps (16 min, 48 sec).

(d) 65 Time Steps (17 min, 20 sec).

(e) 70 Time Steps (18 min, 40 sec).

(f) 80 Time Steps (21 min, 20 sec).

**Figure 7. Flight tracks generated for 16 aircraft without convective weather avoidance. (One unit on the $(x, y)$ axis corresponds to 1km). The larger circles correspond to the aircraft destinations while the smaller circles located at the current position of each aircraft refers to the protected zone around an aircraft, which must not be entered by another aircraft. The complete set of trajectories for this simulation is shown in Figure 6c.**

reaches 16 aircraft, the longest iteration took 70.31 seconds which would not allow an implementation in real time with a control update interval of 32 seconds, even though the average time is below this limit at 21.67 seconds. This issue would most likely be solved by using a faster computer. However it is important to consider that a few iterations may take significantly longer than the average, due to a larger number of active contraints. Additional protocol may need to be included in the algorithm prior to real-time implementation to address this issue.

**Table 4. Computation time without weather avoidance.**

| # Aircraft $N$ | min time (s) | max time (s) | av. time (s) | Control update interval $\triangle tc$ (s) |
|---|---|---|---|---|
| 1 | 0.03 | 0.14 | 0.04 | 32.0 |
| 6 | 0.25 | 5.34 | 0.59 | 32.0 |
| 12 | 1.78 | 28.5 | 3.93 | 32.0 |
| 16 | 4.46 | 70.31 | 21.67 | 32.0 |

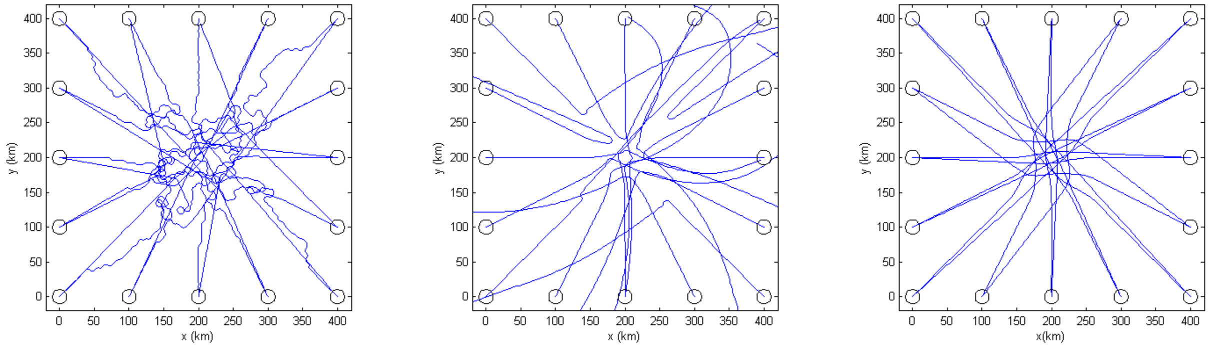
## IV.B. Conflict Resolution and Convective Weather Avoidance

Once the aircraft collision avoidance portion of the code was tested, convective weather avoidance was included in the problem. Figure 9 shows the resulting trajectories for two sets of O-D pairs for 20 aircraft. These configurations were chosen to mimic aircraft entering and leaving sectors as detailed in Figure 10 and 11. For these simulations, a shorter horizon length of 128 seconds was sufficient for aircraft to maintain the minimum separation between all other aircraft as well as to avoid convective weather regions with a VIP index greater or equal to 3. The larger circles of Figure 9 represent the target destination and the ellipses correspond to the boundary enclosing convective weather which aircraft must not cross. Once aircraft have reached their destination, they are removed from the system and no longer displayed on Figures 10 and 11. As an example, notice that aircraft $A9$ reached its target before 50 time steps and it thus not present in Figure 10b.

As shown in Figure 9, all aircraft successfully reach their destination while avoiding convective weather regions. The other difference between the two simulations shown in Figure 9 is the separation required from severe convective weather. In the first case, shown in Figure 9a, aircraft are forbidden to fly through regions where the VIP convective weather index is greater or equal to 3. However, they are allowed to fly right next to these forbidden regions. On the other hand, Figure 9b displays a more conservative constraint for weather avoidance. In this case, not only are convective weather regions with a VIP index greater or equal to 1 enclosed in the bounding ellipse, but in addition a safety zone around these regions is included as well. In these simulations, the level of separation from convective weather regions is enforced at the system level by changing the range of influence during the dilation and erosion steps in the weather clustering process before computing the minimum area ellipses. However a different weather avoidance separation constraint could easily be enforced at the aircraft level by changing the bound on the individual weather constraints.

The simulations shown in Figure 9 are detailed in Figures 10 and 11 respectively, overlapped with the map of Florida and its high altitude sectors. Again the larger black circles correspond to the target disk for each aircraft. The smaller circles placed at the current positions of each aircraft represent the minimum separation required between aircraft. In addition to the current position, the previous 24 time steps are displayed as well, representing the past 3 min and 20 seconds of flight time at cruising altitude. These figures allow us to better verify the aircraft minimum separation constraints are respected.

Table 5 shows results for the computation time for different numbers of aircraft in the system, in the presence of weather avoidance constraints. As in the conflict resolution case, the computation times displayed correspond to the time needed to converge to a solution $\mathbf{u}$ over the horizon $T$. Three computation times are shown, the minimum, maximum and average time required to converge to a solution. When the system reaches 16 aircraft, the longest iteration took 43.68 seconds. This would not allow an implementation in real time with a control update interval of 32 seconds, even though the average time is below this limit at 29.16 seconds. It it interesting to compare these times with the ones without weather avoidance constraints. We notice that the minimum and average times are smaller in the absence of weather, which is expected because we are dealing with fewer constraints. However the maximum time is greater for pure conflict resolution. This shows that the computation time also depends on the number of active constraints, which is influenced

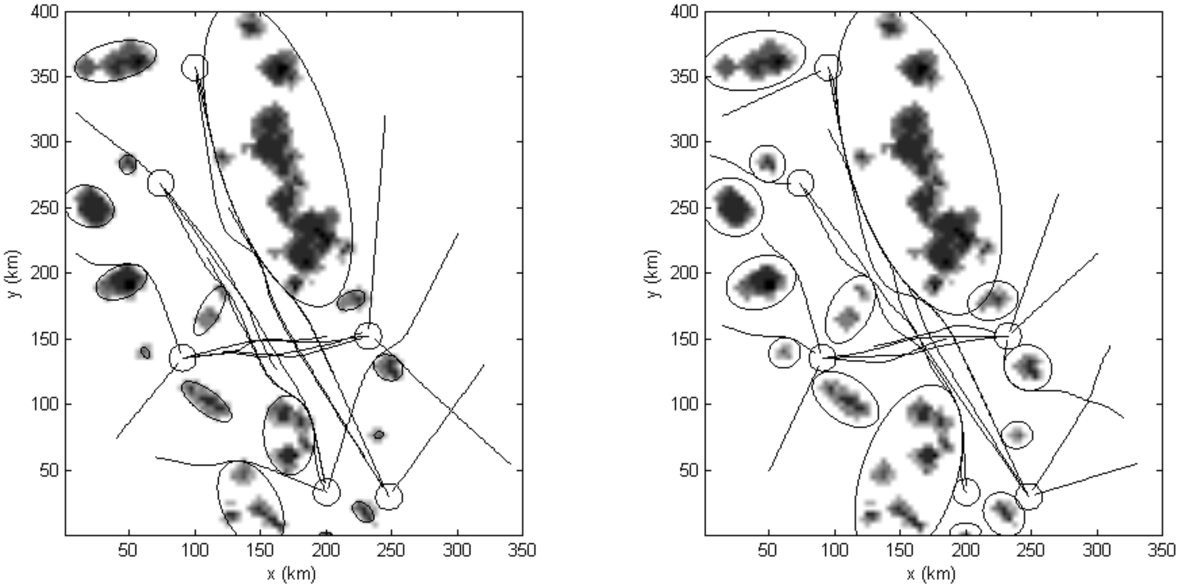


(a) No input cost

(b) The value of $R$ is too large.

(c) Correct tuning of $R$

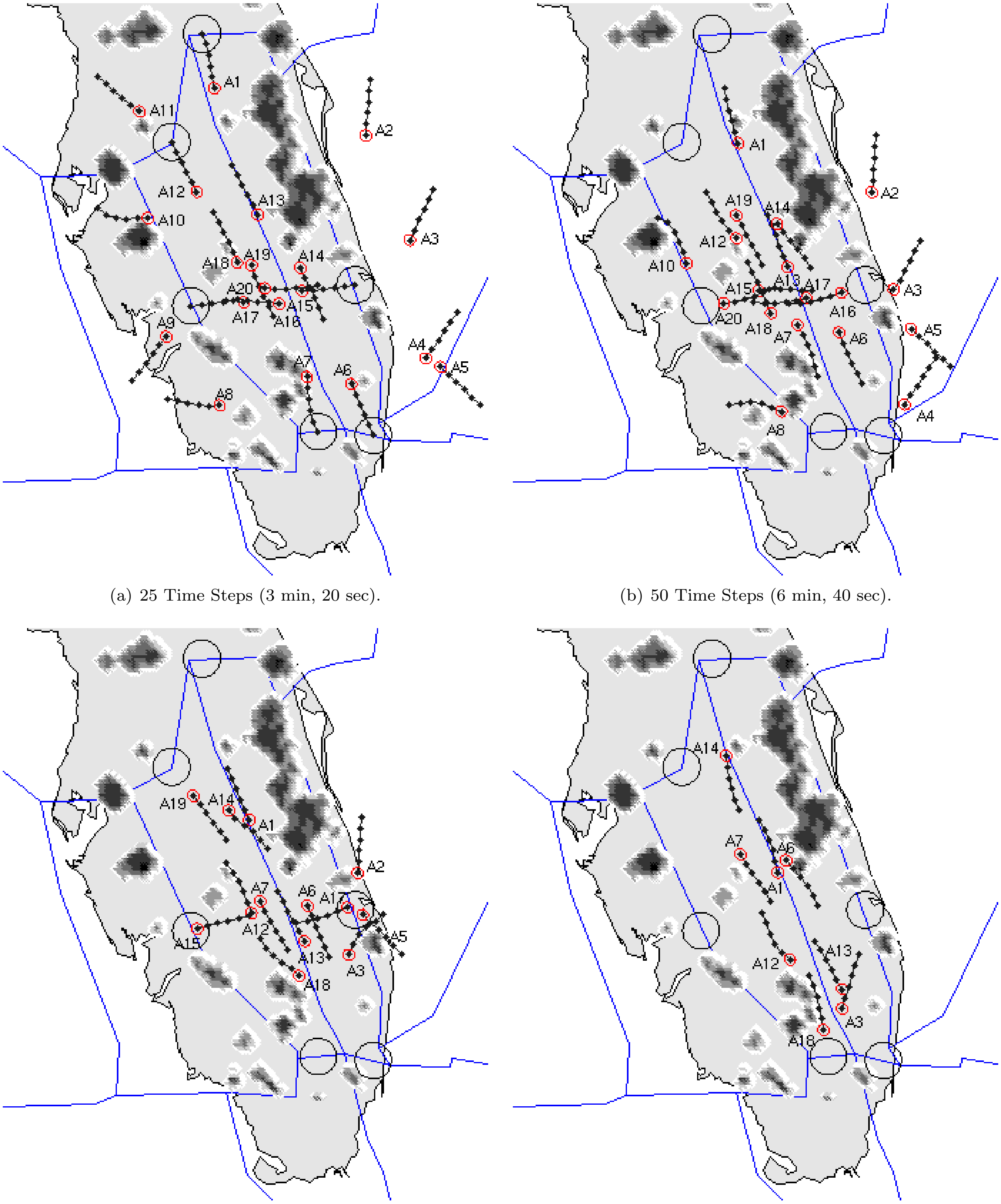
**Figure 8. Affects of the tuning parameter $R$. All simulations were performed with identical origin-destination pairs for 16 aircraft, a horizon length of 256 seconds, a time step of 16 seconds and a control update interval of 32 seconds. (One unit on the $(x, y)$ axes corresponds to 1km).**



(a) Tight bound around convective weather regions.

(b) Conservative bound around convective weather regions.

**Figure 9. Examples of deconflicted trajectories for 20 aircraft in the presence of convective weather for different weather avoidance constraints. (One unit on the $(x, y)$ axes corresponds to 1km).**



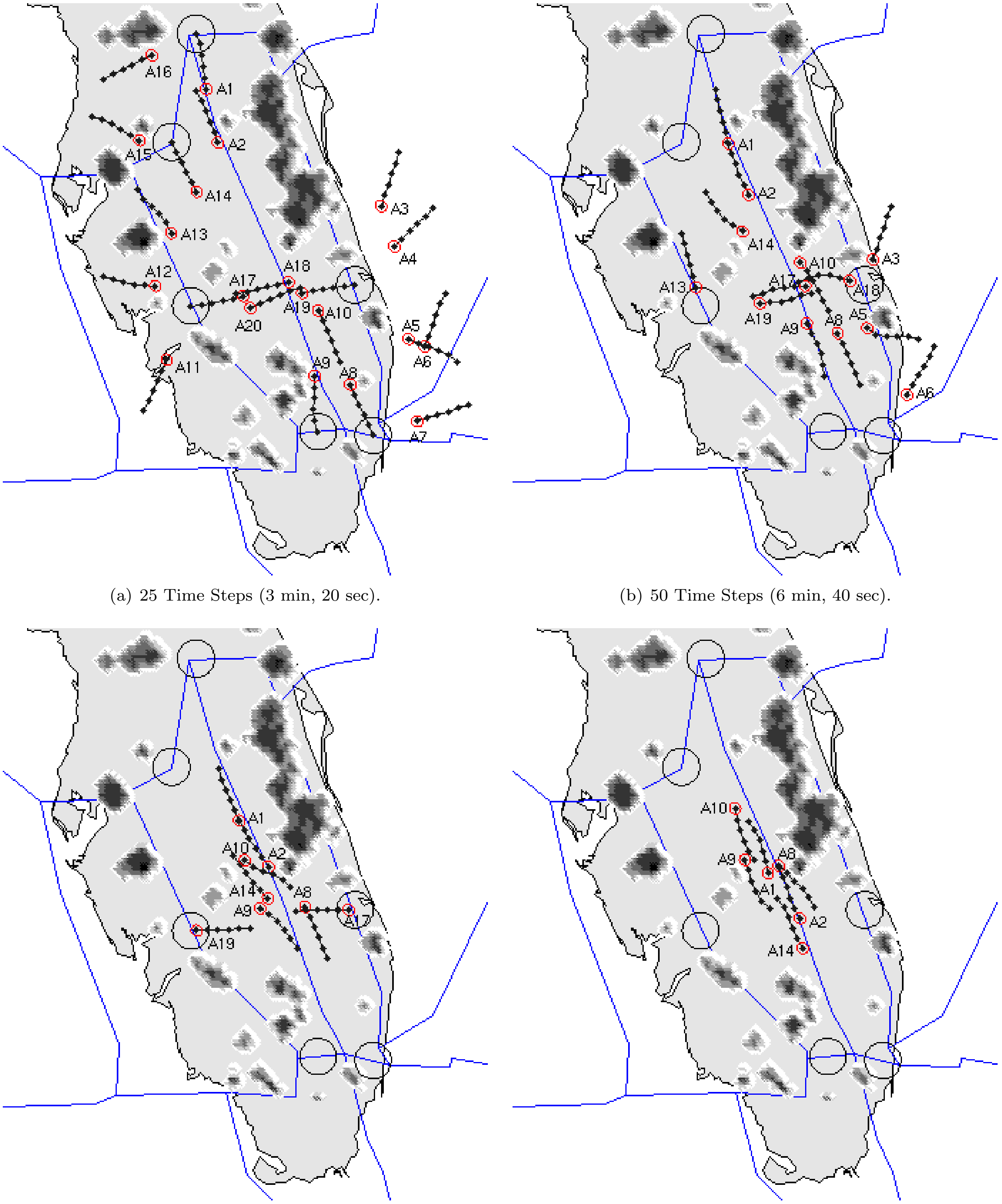
(a) 25 Time Steps (3 min, 20 sec).

(b) 50 Time Steps (6 min, 40 sec).

(c) 75 Time Steps (10 min, 0 sec).

(d) 100 Time Steps (13 min, 20 sec).

**Figure 10. Detailed example of trajectories generated for 20 aircraft over Florida, for a tight bound around convective weather regions.**



(a) 25 Time Steps (3 min, 20 sec).

(b) 50 Time Steps (6 min, 40 sec).

(c) 75 Time Steps (10 min, 0 sec).

(d) 100 Time Steps (13 min, 20 sec).

**Figure 11. Detailed example of trajectories generated for 20 aircraft over Florida, for a conservative bound around convective weather regions.**

by the aircraft origin-destination configuration. For the conflict resolution simulations, aircraft were placed such that they would all potentially collide in the center (Figure 6). This set-up leads to an increase in the number of active constraints. On the other hand, the simulations with weather avoidance dealt with more realistic configurations where 16 aircraft colliding at one point would rarely, if ever, be seen. However in both cases, additional protocol should be included prior to real time implementation to deal with those few iterations which take longer than average to converge to a solution.

**Table 5. Computation Time.**

| # Aircraft $N$ | min time (s) | max time (s) | av. time (s) | Control update interval $\triangle tc$ (s) |
|---|---|---|---|---|
| 6 | 1.26 | 3.62 | 1.93 | 32.0 |
| 12 | 7.12 | 18.64 | 11.41 | 32.0 |
| 16 | 12.12 | 43.68 | 29.16 | 32.0 |
| 20 | 33.89 | 170.62 | 57.09 | 32.0 |

## V. Conclusion

Nonlinear model predictive control was applied to a multi-agent system to compute a set of locally optimal trajectories in the presence of convective weather in a two dimensional space. Two approaches for computing the objective function were explored. The first consisted of a quadratic cost penalizing for each aircraft, the distance between the position at the end of the horizon and the target center. The second method incorporated the solution to the Hamilton-Jacobi nonlinear partial differential equation for a given destination and convective weather data. The solution to this PDE results in a value function which represents approximately the minimum travel time from any point in the NAS to the specified destination. One advantage of this method lies in the fact that the value function derived from the HJ PDE provides a global view of the environment with no local minima. Furthermore, it can incorporate any user defined cost such as minimum time or minimum fuel.

However this value function cannot be expressed in analytical form unlike the quadratic cost function. In order to increase the accuracy of the results, the gradients of the objective function and the constraints are provided to the solver, which in turns uses these gradients to approximate the Hessian of the Lagrangian. In the case of HJ PDE derived objective function, the numerically approximated gradients lead to results, which are not always well behaved. These numerical approximations cause the computation time to increase significantly as the solver exhibits difficulties when trying to converge to a solution. They also cause the system to be sensitive to changes in aircraft initial positions.

The next goal of this work is to incorporate a third dimension in the system as well as including time varying weather patterns and wind profiles. The added dimension will certainly cause a substantial increase in the computation time due to the introduction of a second decision variable. A tradeoff will need to be examined between the added computation time and the usefulness of extending the framework to three dimensions, given the layered structure of high altitude traffic. Introducing time varying weather patterns will present interesting challenges such as choosing proper methods for weather forecasting and dealing with weather avoidance constraints, which change over the optimization horizon.

In the future, including wind data in the current framework will be valuable because aircraft typically chose to fly wind optimal routes. In this case, having a global view of the airspace will become crucial. Also, incorporating winds in this study will enable the use of additional user defined costs such a fuel consumption. One advantage of the HJ PDE method lies in the fact that the system can easily be extended to include wind profiles. However before successfully implementing this method, means for computing accurate numerical gradients must be found or first order optimization algorithms must be explored as an alternative.

This work presents a centralized protocol for multiple aircraft aircraft routine, which is similar to the current operational system in the NAS. This general class of methods usually suffers from limited scalability and poor robustness. In addition, if designed to be applied from the ground, they are subject to equipment failure at the Air Route Traffic Control Center level. One possible solution to this problem is to develop fully decentralized conflict resolution algorithms, which is the motivation for the long term goal of this work. Once the time varying centralized version of the algorithm is built and tested in three dimensions, a decentralized protocol will be designed to be applied in the *Next Generation Air Transportation System* (NGATS).

## Acknowledgments

This work was supported by NASA Ames Research Center under Task Order NNA05CS48G and Task Order TO.048.0.BS.AF and by the National Science Foundation under contract CNS-0615299. We are grateful to Dr. Banavar Sridhar for his support for our work. We are thankful to Dr. Kapil Sheth and Dr. Shon Grabbe for their help with NCWD and RUC data as well as FACET without which this work would not have been possible. We want to acknowledge Dr. George Meyer for his ongoing support for our research. We are thankful to Professor Lucien Polak for fruitful conversations on optimization. We are grateful to Larry Hogle and Bill Berry for helping with the collaboration between UC Berkeley and NASA Ames through the UARC.

## References

[1] Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," *3rd USA/Europe Air Traffic Management R&D Seminar*, ACC, Napoli, Italy, 2000.

[2] Nolan, M., *Fundamentals of Air Traffic Control*, Brooks Cole, 2003.

[3] Hwang, I. and Tomlin, C., "Protocol-based Conflict Resolution for Finite Information Horizon," *American Control Conference*, Vol. 1, Anchorage, Alaska, 2002, pp. 748–753.

[4] Krozel, J., Lee, C., and Mitchell, J., "Estimating time of arrival in heavy weather conditions," *Conference on Guidance, Navigation and Control*, AIAA, Portland, Oregon, 1999, pp. 1481–1495, AIAA-99-4232.

[5] Krozel, J., Penny, S., Prete, J., and Mitchell, J., "Comparison of Algorithms for Synthesizing Weather Avoidance Routes in Transition Airspace," *Conference on Guidance, Navigation and Control*, AIAA, Providence, Rhode Island, 2004, AIAA-2004-4790.

[6] Prete, J. and Mitchell, J., "Safe Routing of Multiple Aircraft Flows in the Presence of Time-Varying Weather Data," *Conference on Guidance, Navigation and Control*, AIAA, Providence, Rhode Island, 2004, AIAA-2004-4791.

[7] Krozel, J., Capozzi, B., and Andre, T., "The Future National Airspace System: Design Requirements Imposed by Weather Constraints," *Conference on Guidance, Navigation and Control*, AIAA, Austin, Texas, 2004, AIAA-2003-5769.

[8] Mitchell, J. and Polishchuk, V., "Airspace Throughput Analysis Considering Stochastic Weather," *Conference on Guidance, Navigation and Control*, AIAA, Keystone, Colorado, 2006, AIAA-2006-6770.

[9] Mao, Z., Feron, E., and Bilimoria, K., "Stability of intersecting aircraft flows under decentralized conflict avoidance rules," *Conference on Guidance, Navigation and Control*, Denver, Colorado, 2000, pp. 748–753, AIAA-2000-4271.

[10] Crandall, M. and Lions, P., "Viscosity Solutions of Hamilton-Jacobi Equations," *Transactions of the American Mathematical Society*, Vol. 227, No. 1, 1983.

[11] Crandall, M. and Lions, P., "Two Approximations of Solutions of Hamilton-Jacobi Equations," *Mathematics of Computation*, Vol. 43, No. 167, 1984, pp. 1–19.

[12] Mitchell, I., Bayen, A., and Tomlin, C., "Computing Reachable sets for continuous dynamic games using level set methods," *Transactions on Automatic Control*, Vol. 5, No. 7, 2005, pp. 947–957.

[13] Cardaliaguet, P., Quincampoix, M., and Saint-Pierre, P., "Set-valued numerical analysis for optimal control and differential games," *Stochastic and Differential Games: Theory and Numerical Methods*, Vol. 4, 1999, pp. 177–247.

[14] Histon, J., Hansman, R., Aigoin, G., Delahaye, D., and Puechmorel, S., "Introducing structural considerations into complexity metrics," *Air Traffic Control Quarterly*, Vol. 10, No. 2, 2002, pp. 115–130.

[15] Robelin, C., Sun, D., Wu, G., and Bayen, A., "MILP control of aggregate Eulerian network airspace models," *American Control Conference*, Minneapolis, Minnesota, 2006, pp. 5257–5262.

[16] Sridhar, B., Soni, T., Sheth, K., and Chatterji, G., "An Aggregate Flow Model for Air Traffic Management," *Conference on Guidance, Navigation and Control*, AIAA, Providence, Rhode Island, 2004, AIAA 2004-5316.

[17] Roy, S. and Verghese, B. S. G., "An Aggregate Dynamic Stochastic Model for Air Traffic Control," *Proceedings of the* $5^{th}$ *USA/Eurpoe ATM R & D Seminar*, ATM, Budapest, Hungary, 2003.

[18] Sun, D., Yang, S., Strub, I., Bayen, A., Sridhar, B., and Sheth, K., "Eulerian Trilogy," *Conference on Guidance, Navigation and Control*, Keystone, Colorado, 2006, AIAA-2006-6227.

[19] Sun, D. and Bayen, A., "A New Eulerian-Lagrangian Large-capacity Cell Transmission Model for Optimization-based Traffic Flow Management." .

[20] Alton, K. and Mitchell, I., "Optimal path planning under different norms in continuous state spaces," *Proceeding of the International Conference on Robotics and Automation*, 2006, pp. 866–872.

[21] Shim, D., Kim, H., and Sastry, S., "Decentralized nonlinear model predictive control of multiple flying robots," *Proceedings of the* $42^{nd}$ *IEEE Conference on Decision and Control*, Vol. 4, Maui, Hawaii, 2003, pp. 3621–3626.

[22] Hu, J., Prandini, M., and Sastry, S., "Optimal Coordinated Motions of Multiple Agents Moving on a Plane," *SIAM Journal on Control and Optimization*, Vol. 42, No. 2, 2003, pp. 637–668.

[23] Robinson, M., Evans, J., and Crowe, B., "En route weather depiction benefits of the NEXRAD vertically integrated liquid water product utilized by the corridor integrated weather system," *Aviation, Range and Aerospace Meteorology*, ARAM, Portland, OR, 2002, pp. 120–123.

[24] Bayen, A., Santhanam, S., Mitchell, I., and Tomlin, C., "A differential game formulation of alert levels in ETMS data for high altitude traffic," *Conference on Guidance, Navigation and Control*, AIAA, Austin,Texas, 2003, AIAA-2003-5341.

[25] Frazzoli, E., Dahleh, M., and Feron, R., "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.

[26] Bemporad, A. and Morari, M., "Robust Model Predictive Control: A Survey," *Robustness in Identification and Control*, Vol. 245, 1999, pp. 207–226.

[27] Mayn, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality," *Automatica*, Vol. 36, 2000, pp. 789–814.

[28] Kim, H., Shim, D., and Sastry, S., "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles," *American Control Conference*, Vol. 5, 2002, pp. 3576–3581.

[29] Kim, H., Shim, D., and Sastry, S., "Flying robots: modeling, control and decision making," *IEEE International Confer-*

*ence on Robotics and Automation*, Vol. 1, 2002, pp. 66–71.

[30] Holmström, K., Göran, A., and Edvall, M., *User's Guide for TOMLAB/SNOPT*, 2006.

[31] Polak, E., *Optimization: Algorithms and Consistent Approximation*, Springer, 1997.

[32] Holmström, K., Göran, A., and Edvall, M., *User's Guide for TOMLAB*, 2006.

[33] Jardin, M. and Bryson, A., "Neighboring Optimal Aircraft Guidance in Winds," *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 4, 2001, pp. 710–715.

[34] Vladimirsky, A., "Static PDEs for Time-Dependent Control Problems," *Interfaces and Free Boundaries*, Vol. 8, No. 3, 2006, pp. 281–300.

[35] Mitchell, I. M. and Templeton, J. A., "A Toolbox of Hamilton-Jacobi Solvers for Analysis of Nondeterministic Continuous and Hybrid Systems," *Hybrid Systems: Computation and Control*, edited by M. Morari and L. Thiele, No. 3414, Springer Verlag, 2005, pp. 480–494.

[36] Tomlin, C., Pappas, G., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509.

[37] Inalhan, G., Stipanovic, D., and Tomlin, C., "Decentralized Optimization with application to Multiple Aircraft Coordination," *Proceedings of the* $41^{st}$ *IEEE Conference on Decision and Control*, Vol. 1, Las Vegas, Nevada, 2002, pp. 1147–1155.

[38] Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *Conference on Guidance, Navigation and Control*, Providence, Rhode Island, 2004, AIAA-2004-5141.