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

We present a model predictive control based algorithm for aircraft motion planning that will apply to converging flows of aircraft going through convective weather in the en route airspace. The cost function associated with the model predictive control optimization problem is obtained by solving the Hamilton-Jacobi equation, a non-linear partial differential equation. For a given wind profile, convective weather conditions and aircraft destination, 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. The optimal control sequence is then computed over a fixed horizon by minimizing the aircraft cost function, subject to aircraft separation constraints and bounds on aircraft turning rates. This algorithm results in aircraft trajectories corresponding to a locally optimal solution of the optimization program involving the set of aircraft considered.

Nomenclature

T	Model Predictive Control horizon length, (units: seconds)
Δt	Model Predictive Control time step, (units: seconds)
n	Model Predictive Control number of steps in horizon
N	Total number of aircraft considered
$x_{i,j}$	x position of aircraft i at time j, (units: meters)
$y_{i,j}$	y position of aircraft i at time j, (units: meters)
$\theta_{i,j}$	Heading of aircraft i at time j, (units: radians)
$u_{i,j}$	Turning rate of aircraft i at time j, (units: radians/second)
x	$[x_{1,1}, ..., x_{1,n}, ..., x_{N,1}, ..., x_{N,n}]$ Aircraft x position vector, (units: meters)
y	$[y_{1,1}, ..., y_{1,n}, ..., y_{N,1}, ..., y_{N,n}]$ Aircraft y position vector, (units: meters)
θ	$[\theta_{1,1}, ..., \theta_{1,n}, ..., \theta_{N,1}, ..., \theta_{N,n}]$ Aircraft heading vector, (units: radians)
u	$[u_{1,1}, ..., u_{1,n}, ..., u_{N,1}, ..., u_{N,n}]$ Aircraft turning rate vector, (units: radians/second)
V_i	Value function of aircraft i, encoding the minimum time to reach destination i, (units: seconds)
v	Aircraft velocity, (units: meters/second)
h	Aircraft altitude, (units: meters)
P	Atmospheric pressure, (units: Pascal)
Subscript	
i	Aircraft index
j	Sampling 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, we will however only consider protocols centralized at the sector level, which correspond to the current operational system in place in the NAS.[2]

An input of our problem is a grid, for which we have wind and convective weather data at periodic time intervals. We will approach the problem of deconflicting aircraft paths through severe weather using *Nonlinear Model Predictive Control* (NMPC). Assuming constant velocity, the optimal input sequence, which represents the turning rate for each aircraft, (equal to zero if the aircraft is traveling at fixed heading) will be repeatedly computed over a horizon T by minimizing a cost function for each aircraft subject to non-linear constraints. The cost function is obtained by solving the *Hamilton-Jacobi partial differential equation* (HJ PDE)[10,11,12,13] for a set of wind and turbulence data as well as a destination. The solution to the HJ PDE provides a value function which contains the minimum travel time from any point in the NAS to the specified destination. The shortest travel time incorporates avoidance of convective weather patches as well as effects of the wind flow field on aircraft. Each destination in the NAS thus has a corresponding value function. The set of value functions will serve as a look-up table for our cost function. By penalizing the position at the end of the horizon in our objective function, we aim 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, thus minimizing the remaining travel time, which is encoded in the objective function. The result gives minimum travel time 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

II.A. Aircraft Routing

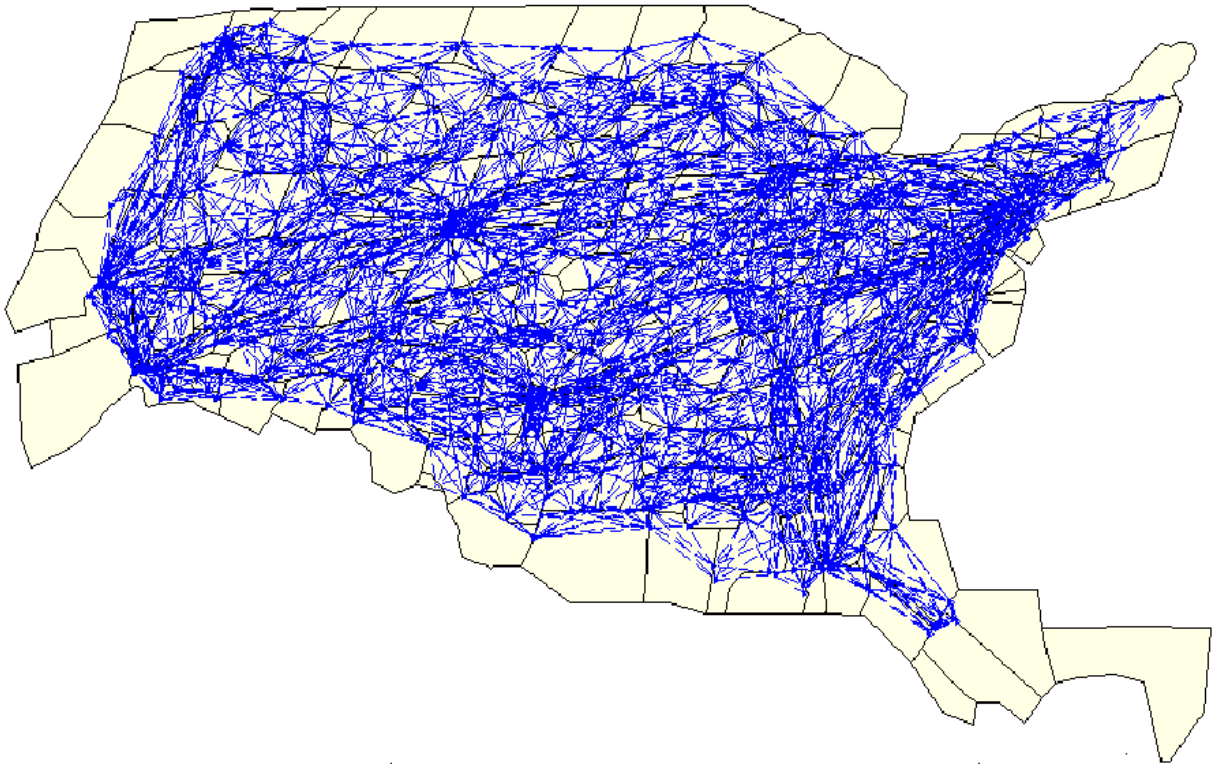


Figure 1. Flow network with origin-destination points for each high altitude sector based on aggregate traffic analysis for the entire NAS.[14]

High altitude traffic in the en route airspace follows flow patterns, which have been extensively studied in the literature.[15,16,17,18,14] For the purpose of this work, we consider entry and departure points in high altitude sector defined by flows, which characterize high altitude traffic In earlier work,[16] we have explained how to construct the origin-destination points based on a flow analysis, as shown for the entire NAS in Figure 1. Our algorithm computes the optimal trajectory by minimizing an objective function, which represents the minimum time to reach the destination. For each destination, the function representing the minimum time to reach a user-specified target $\mathcal{T}$ (the destination disk on the map of Figure 2 is obtained by solving the HJ PDE. 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.[19,13] We will use this technique in the present work (the computation of the solution to the HJ PDE is detailed in Section III.B). Such trajectories will not be constrained to these flows, but rather will take advantage of favorable winds and also avoid regions of convective weather. Traffic through sectors *ZMA64* and *ZMA65* over Florida will be considered, as shown in Figure 2, given that available weather data provides large regions of convective weather in those sectors, which makes them interesting to study.

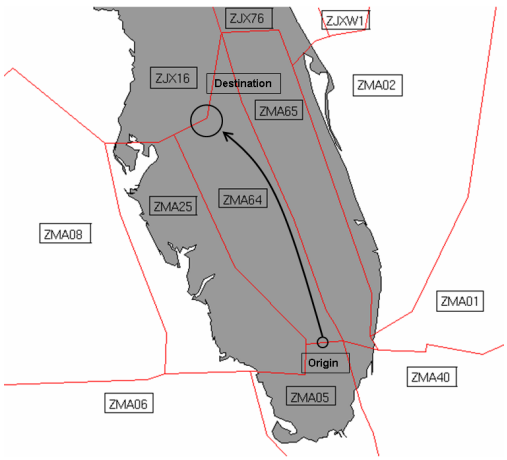


Figure 2. Example of origin-destination pair in Florida Sector ZMA64. The larger circle represents the destination of the aircraft, whereas the smaller circle at the origin represents the protected zone surrounding the aircraft, assuming the aircraft is located at the origin. The same representation will be used in the simulation results of Section IV.

II.B. Constraints

The computation of aircraft minimum time trajectories must take into account constraints that restrict the effective region an aircraft can use.

II.B.1. Protected Zones

One of the key elements 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. These hard constraints will be encoded directly in the formulation of our problem. For the present work, we restrict ourselves to horizontal motion in 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 very relevant for horizontal separation in the en route airspace.[20]

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[21] 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.

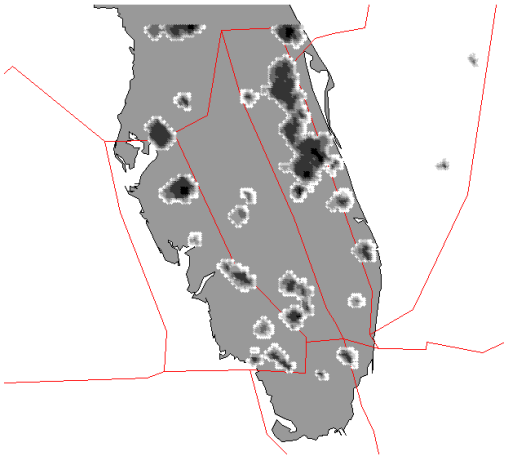


Figure 3. Representation of Convective Weather over Florida on August 1st, 2005 at 8:14pm.

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 3. 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)	VIP Level	Colormap (Figure: 3)
0.14	1	white
0.7	2	pale gray
3.5	3	light gray
6.9	4	medium gray
12.0	5	dark gray
32.0	6	black

II.B.3. Wind Profile

Strong winds can alter the travel time of an aircraft significantly. Therefore knowing the current and forecasted wind profiles can help in optimal wind routing.[22,23,24] Wind data is obtained from the *National Oceanic and Atmospheric Administration* (NOAA) under the name *Rapid Update Cycle* (RUC) Data. Every hour, the current wind data is received as well as a one hour, two hour and three hour forecast. In addition, every three hours, a six hour forecast is available. Each of these data files contain a vector field for thirty-seven isobar levels ranging from 100 to 1000 millibars. For the purpose of our analysis, we approximate an isobar level to one with constant altitude. We use equation (1) to model the relationship between pressure and altitude, where h represents the altitude in meters and P the pressure in Pascal.

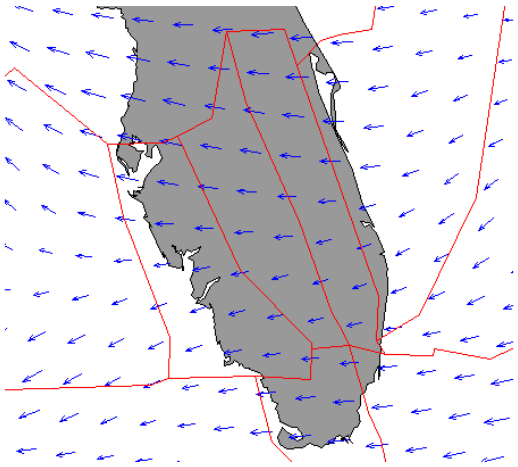


Figure 4. Representation of wind profile over Florida on August 1st, 2006 at 9pm at a pressure of 100mb.

$$\log_{10}(P) \approx 5 - \frac{h}{15500} \tag{1}$$

Using a feature of the NASA software FACET,[1] we can read RUC files and output a text file that can in turn be used in Matlab. For each of the 37 isobar levels, 17063 vectors are described by the following information:

Table 2. Wind Data Structure

Lat	Long	Wind Direction (deg)	Horizontal Velocity (kts)	Vertical Velocity (kts)

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) Dynamic convective weather obstacles.
3) Dynamic wind fields.
Find: for each aircraft, an optimal (minimum-time) trajectory, from the corresponding origin to the destination while avoiding the convective weather and the other aircraft.

In the current implementation of the method, we consider only static weather. Time varying weather patterns will be included in the final version of this article. This does not alter the formulation of the problem in any way, as will be seen in the algorithm of Section III.A. The problem statement is detailed as follows. The dynamics $(\dot{x}_i, \dot{y}_i, \dot{\theta}_i) = f(x_i, y_i, \theta_i, u_i)$ of aircraft i can be written as:

$$\dot{x}_i(t) = v\cos\theta_i(t) \tag{2}$$
$$\dot{y}_i(t) = v\cos\theta_i(t) \tag{3}$$
$$\dot{\theta}_i(t) = u_i(t) \tag{4}$$

Note that in the current implementation of this method, the wind profile is included in the value function computed by the HJ PDE (Section III.B) but not in the dynamics of the system for simplicity. This will be changed for the final version of this article. We assume the velocity v of each aircraft is constant throughout the portion of the en route airspace of interest. The individual aircraft dynamics are decoupled. We can assemble them into a single system dynamics $\dot{x} = f(x, u)$. We are looking for the optimal input sequence u_i for each aircraft in our system given constraints described below to minimize a cost function, which we will define next. The variable u represents the collection of input sequences u_i for all aircraft with $i \in [1, N]$. The minimization of objective function J can be expressed as shown in equation (5).

$$\begin{array}{ll} \textbf{min:} & J(u(\cdot)) = \sum_{i=1}^{N} J_i(u_i(\cdot)) \\ \textbf{s.t.:} & g(u(\cdot)) \leq 0 \end{array} \tag{5}$$

where $u(\cdot) = [u_1(\cdot), \cdots, u_N(\cdot)]$ is a measurable control input from $[0, T]$ to $\mathbb{R}^n$. We define $(x_i^{u_i(\cdot)}(\cdot), y_i^{u_i(\cdot)}(\cdot))$ as the (x, y) trajectory of aircraft i under input control $u_i(\cdot)$. The optimization problem (5) can be written explicitly as follows with the constraints of the problem.

$$\begin{array}{lllll} \textbf{min:} & \sum_{i=1}^{N} V_i(x_i^{u_i(\cdot)}(T), y_i^{u_i(\cdot)}(T)) & & & \\ \textbf{s.t.:} & -a \leq u_i(t) \leq a & i \in [1, N] & \forall t \in [0, T] \\ & (x_i^{u_i(\cdot)}(t) - x_k^{u_k(\cdot)}(t))^2 + (y_i^{u_i(\cdot)}(t) - y_k^{u_k(\cdot)}(t))^2 \geq r_{\min}^2 & k \neq i \in [1, N] & \forall t \in [0, T] \\ & \dot{x}_i^{u_i(\cdot)}(t) = v \cos \theta_i^{u_i(\cdot)}(t) & i \in [1, N] & \forall t \in [0, T] \\ & \dot{y}_i^{u_i(\cdot)}(t) = v \sin \theta_i^{u_i(\cdot)}(t) & i \in [1, N] & \forall t \in [0, T] \\ & \dot{\theta}_i^{u_i(\cdot)}(t) = u_i(t) & i \in [1, N] & \forall t \in [0, T] \end{array} \tag{6}$$

V_i corresponds to the value function of aircraft i obtained from solving the HJ PDE for the destination of aircraft i (see section III.B). It represents the minimum time function to the destination of aircraft i. 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.[25] The second constraint states that all aircraft must be separated by a distance greater or equal to $r_{\min} = 5$ nautical miles. The last three constraints include the dynamics of the aircraft. This formulation is standard in the path planning literature.[26]

III. Algorithm

The algorithm used to compute the minimum time trajectories combines numerical methods based on level sets methods[27,28,29,12] and an optimization routine built with NMPC.

III.A. Algorithm Outline

The outline presents the algorithm for time varying weather patterns, which will be implemented in the final version of this article. Currently static weather patterns are used with the same algorithm.

Initialization

Set horizon length T and time step $\triangle t$.

For all aircraft ($i = 1 \cdots N$), set origin-destination pair and aircraft speed v_i.

Main Loop

Repeat until all aircraft have reached their destination.

If new weather data is available.

Load VIP and wind data.

For all aircraft ($i = 1 \cdots N$).

Solve HJ PDE (7) to compute value function V_i.

Interpolate V_i in space $\mathbb{R}^2$ to reduce step size.

Compute gradient ∇V_i with respect to x_i and y_i.

For all aircraft ($i = 1 \cdots N$), load value function V_i and gradient ∇V_i.

Solve discrete time non-linear MPC problem (14) on horizon T.

Inputs:

Initial guess u_0 for control sequence u.

Objective function (15) and gradient of objective function (16) with respect to u.

Constraints (21),(22),(25) and gradient of constraints (23),(24),(28) with respect to u.

Ouput:

Optimal control sequence u over time horizon $[0, T]$.

Advance to the next step on the horizon by $\triangle t$ time intervals.

Update initial guess u_0 with new control sequence u.

End of main loop.

Output: Optimal trajectories for set of N aircraft from their origin to their destination.

III.B. Hamilton Jacobi Equation

The value function $V_i(x)$ for aircraft i evaluated at point x provides a cost-to-go penalty corresponding approximately to the time-to-reach from x the prespecified destination depicted in Figure 2 and denoted $\mathcal{T}_i$. In our current static scenario, this value function can be found by solving a static or time-independent HJ PDE in a computational domain Ω representing the airspace.

$$\begin{aligned} \max_u \left(\nabla V_i(x) \cdot f(x_i, u_i)\right) - \ell(x) = 0 \qquad & \text{in } \Omega \setminus \mathcal{T}_i, \\ V_i(x) = h(x) \quad & \text{on } \partial\mathcal{T}_i. \end{aligned} \tag{7}$$

In this work the set $\mathcal{T}_i$ is chosen as a small circle centered around the destination of the origin-destination pair of an aircraft, and $h(x) = 0$ for $x \in \partial\mathcal{T}_i$. More generally, $\mathcal{T}_i$ can also be used to impose hard constraints on the aircraft motion (such as regions forbidden due to large VIP readings); however, we currently impose those constraints sufficiently through the MPC stage.

If we choose the running cost $\ell(x) \equiv 1$, then $V_i(x)$ is just the minimum time to reach $\mathcal{T}_i$ from x, ignoring weather. In order for the value function to discourage travel through regions with convective weather—essentially imposing the weather through soft constraints—we penalize those regions with a higher running cost $\ell(x) > 1$. For the simulations shown, we have chosen $\ell(x) = \min(k^{\mathrm{VIP}(x)}, \ell_{\max})$, where $\mathrm{VIP}(x)$ is the VIP reading for point x from the NCAR data, k is some small constant (the examples use $k = 4$) and $\ell_{\max}$ takes into account the fact that all sufficiently high VIP readings are equivalently forbidden (the examples use $\ell_{\max} = k^3$). Consequently, $V_i(x)$ is a cost to go function related to but not exactly the time to reach the destination. A separate $V_i(x)$ is computed for each destination airport, therefore indexed by i (destination for aircraft i). Note that equation (7) includes the wind in the dynamics, which equations (2),(3),(4) do not. In the final version of this article both will.

Because the weather data is gathered at such low spatial resolution and because the MPC algorithm ensures proper imposition of the vehicle dynamics (2)–(4) over the time horizon T, we use an even simpler model of the vehicles for the HJ PDE

$$\dot{x} = \hat{u}_1 \qquad \dot{y} = \hat{u}_2$$

with $\|\hat{u}\| \leq v$. While the aircraft can instantaneously change direction under these dynamics, they are not unreasonable for the resolutions at which the solution of the HJ PDE is approximated: the node spacing is on the order of tens of kilometers, and aircraft moving under dynamics (2)–(4) are capable of reversing direction in such a region.

Algorithms for directly solving HJ PDEs similar to (7) exist;[27] however, their generalization to time-dependent data and other vehicle dynamics is not trivial and implementations are not publicly available. Instead, we transform (7) into a time-dependent HJ PDE, which can be solved using the publicly released code in the *Toolbox of Level Set Methods*.[28] An advantage of this transformation is that we anticipate few code changes when we move to time-dependent weather data.

III.C. Model Predictive Control

Model predictive control is defined by the computation, at every time step $\triangle t$, of the optimal control sequence over a finite horizon.[30,31] 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[32] which makes it an attractive choice for multi-agent coordination[33,34].

III.C.1. Dynamics Discretization

Since the dynamics $f(x(t), u(t))$ are continuous and Lipzchitz continuous, we can use the *Forward Euler Method* (FEM) to approximate the solution to the differential equations[31] with:

$$x_i(t_0 + \triangle t) = x_i(t_0) + \triangle t \cdot f(x_i(t_0), u_i(t_0)) \quad i \in [1, N] \tag{8}$$

The error resulting from this approximation over n steps is bounded by $nM\triangle t^2/2$ where $M = \max|\partial f/\partial t|$. In our case $M = \sqrt{2a^2(v^2+2)}$. The number of steps that will be used is determined by the relationship between our horizon length T and our time step $\triangle t$. We find that $n = T/\triangle t + 1$. If we apply the FEM to our system dynamics and our initial conditions $x_{1,j}$, $y_{1,j}$, $\theta_{1,j}$, we get the following approximation of the discrete trajectory of aircraft i at time step j.

$$x_{i,2} = x_{i,1} + \triangle t \cdot v\cos(\theta_{i,1}) \tag{9}$$
$$y_{i,2} = y_{i,1} + \triangle t \cdot v\sin(\theta_{i,1}) \tag{10}$$

$$x_{i,j} = x_{i,1} + \triangle t \cdot v\cos(\theta_{i,1}) + \triangle t \cdot v\sum_{k=1}^{j-2}\cos\left[\theta_{i,1} + \triangle t\sum_{r=1}^{k} u_{i,r}\right] \quad \forall i \in [1,N] \quad \forall j \in [3,n] \tag{11}$$

$$y_{i,j} = y_{i,1} + \triangle t \cdot v\sin(\theta_{i,1}) + \triangle t \cdot v\sum_{k=1}^{j-2}\sin\left[\theta_{i,1} + \triangle t\sum_{r=1}^{k} u_{i,r}\right] \quad \forall i \in [1,N] \quad \forall j \in [3,n] \tag{12}$$

$$\theta_{i,j} = \theta_{i,1} + \triangle t\sum_{k=1}^{j-1} u_{i,k} \quad \forall i \in [1,N] \quad \forall j \in [2,n] \tag{13}$$

III.C.2. Discrete Optimization Problem Formulation

Once the system is discretized as detailed in section III.C.1, the problem statement becomes:

$$\begin{array}{lllll} \textbf{min:} & \sum_{i=1}^{N} V_i(x_{i,n}, y_{i,n}) & & & \\ \textbf{s.t.:} & -a \leq u_{i,j} \leq a & i \in [1,N] & \forall j \in [1,n] \\ & (x_{i,j} - x_{k,j})^2 + (y_{i,j} - y_{k,j})^2 \geq r_{\min}^2 & k \neq i \in [1,N] & \forall j \in [1,n] \\ & x_{i,j+1} = x_{i,j} + \triangle t \cdot v\cos(\theta_{i,j}) & i \in [1,N] & \forall j \in [1,n] \\ & y_{i,j+1} = y_{i,j} + \triangle t \cdot v\sin(\theta_{i,j}) & i \in [1,N] & \forall j \in [1,n] \\ & \theta_{i,j+1} = \theta_{i,j} + u_{i,j} & i \in [1,N] & \forall j \in [1,n] \end{array} \tag{14}$$

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 u rather than the states of the

aircraft x, y and θ. Indeed all discrete states $x_{i,j}$, $y_{i,j}$ and $\theta_{i,j}$ for $i \in [1, N]$ and $j \in [1, n]$ are uniquely determined by u. As a result, the objective function and the constraints are all only expressed in terms of u and the initial conditions of the system, as detailed in the following subsections.

III.C.3. Objective Function

We use the value function obtained by solving the HJ PDE as described in section III.B in the objective function of our model predictive control optimization routine, as shown in equation (15). The final point at the end of the horizon is penalized to ensure that deviations due to aircraft conflict resolution and convective weather avoidance are small.

$$\min_u J(u) = \min_u \sum_{i=1}^{N} V_i(x_{i,n}(u_i), y_{i,n}(u_i)) \tag{15}$$

The dependency $x_{i,n}(u_i)$ has been added to signify the implicit dependency of $x_{i,n}$ to the input u_i from equation (11) and similarly for $y_{i,n}$. In later parts of this article, it might be omitted for clarity of notation. We express $x_{i,n}$ and $y_{i,n}$ as a function of the initial conditions and the input sequence u_i of aircraft i as shown in equations (11) and (12). We use the Matlab function *fmincon* to solve our non linear constrained optimization problem. In order to achieve better results, we provide the Matlab optimization function *fmincon* with gradients of the objective function. Without the gradient, the function *fmincon* computes its own numerical approximations to the gradient, which are less accurate and can lead to inaccuracy in the computed optimal solution.

$$\frac{\partial V_i(x_{i,n}(u_i), y_{i,n}(u_i))}{\partial u_i} = \begin{bmatrix} \frac{\partial V_i(x_{i,n}(u_i), y_{i,n}(u_i))}{\partial u_{i,1}} \\ \vdots \\ \frac{\partial V_i(x_{i,n}(u_i), y_{i,n}(u_i))}{\partial u_{i,n}} \end{bmatrix} \tag{16}$$

$$= \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial x_{i,n}} \frac{\partial x_{i,n}(u_i)}{\partial u_i} + \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial y_{i,n}} \frac{\partial y_{i,n}(u_i)}{\partial u_i} \tag{17}$$

$$= \begin{bmatrix} \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial x_{i,n}} \frac{\partial x_{i,n}(u_i)}{\partial u_{i,1}} + \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial y_{i,n}} \frac{\partial y_{i,n}(u_i)}{\partial u_{i,1}} \\ \vdots \\ \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial x_{i,n}} \frac{\partial x_{i,n}(u_i)}{\partial u_{i,n}} + \frac{\partial V_i(x_{i,n}, y_{i,n})}{\partial y_{i,n}} \frac{\partial y_{i,n}(u_i)}{\partial u_{i,n}} \end{bmatrix} \tag{18}$$

where $\partial V_i(x_{i,n}, y_{i,n})/\partial x_{i,n}$ should be $\frac{\partial V_i(x,y)}{\partial x}|_{(x_{i,n}, y_{i,n})}$ (Similarly for $\partial V_i(x_{i,n}, y_{i,n})/\partial y_{i,n}$). We use this slight abuse of notation for simplicity. From (11) and (12) we obtain:

$$\frac{\partial x_{i,n}}{\partial u_{i,k}} = -\triangle t^2 \cdot v \sum_{m=k}^{n-2} \sin\left[\theta_{i,1} + \triangle t \sum_{r=0}^{m} u_{i,r}\right] \tag{19}$$

$$\frac{\partial y_{i,n}}{\partial u_{i,k}} = \triangle t^2 \cdot v \sum_{m=k}^{n-2} \cos\left[\theta_{i,1} + \triangle t \sum_{r=0}^{m} u_{i,r}\right] \tag{20}$$

The gradients $\partial V_i(x_{i,n}, y_{i,n})/\partial x_{i,n}$ and $\partial V_i(x_{i,n}, y_{i,n})/\partial y_{i,n}$ of the value function with respect to x and y, are computed numerically using the results of the Matlab toolbox.

III.C.4. Constraints

As we did for the objective function and its gradient, we provide the Matlab function *fmincon* with the constraints of the system as well as their gradient with respect to the control variable u. Providing the gradient of the constraints to the function *fmincon* leads to a more accurate solution,[31] than allowing *fmincon* to compute its own numerical approximation to the gradient of the constraints.

INPUT CONSTRAINTS: Let $g_1(u)$ be defined as the vector of $2nN$ entries of input contraint equations that are of the type:

$$u_{i,j} - a \leq 0 \quad \forall i \in [0, N] \quad \forall j \in [0, n] \tag{21}$$
$$-u_{i,j} - a \leq 0 \quad \forall i \in [0, N] \quad \forall j \in [0, n] \tag{22}$$

Taking the gradient of the $g_1(u)$ components with respect to the input u_i gives the vectors:

$$\frac{\partial(u_{i,j} - a)}{\partial u_{i,k}} = [0, \ldots, 1, \ldots, 0]^T \tag{23}$$
$$\frac{\partial(-u_{i,j} - a)}{\partial u_{i,k}} = [0, \ldots, -1, \ldots, 0]^T \tag{24}$$

where the non zero element in both equations corresponds to the entry where $j = k$.

COLLISION AVOIDANCE CONSTRAINTS: Let $g_2(u)$ be defined as the vector of collision avoidance constraints with $nN(N-1)/2$ entries that are expressed by equation (25).

$$(x_{i,j} - x_{k,j})^2 + (y_{i,j} - y_{k,j})^2 \geq r_{\min}^2 \quad \forall i \neq k \in [1, N] \quad \forall j \in [1, n] \tag{25}$$

where $x_{i,j}$ and $y_{i,j}$ are expressed with equation (11) and (12). We need to compute

$$\frac{\partial g_2(u)}{\partial u} \tag{26}$$

If we look at the partial derivative of one element of $g_2(u)$, $r_{\min}^2 - (x_{i,j} - x_{k,j})^2 - (y_{i,j} - y_{k,j})^2 \leq 0$ for example, with respect to one component $u_{p,q}$ with $p \in [1, N]$ and $q \in [1, n]$, we obtain:

$$\begin{aligned} \frac{\partial(r_{min}^2 - (x_{i,j} - x_{k,j})^2 - (y_{i,j} - y_{k,j})^2)}{\partial u_{p,q}} &= 2\left(\frac{\partial x_{i,j}}{\partial u_{p,q}} - \frac{\partial x_{k,j}}{\partial u_{p,q}}\right) \cdot (x_{k,j} - x_{i,j}) \\ &\quad +2\left(\frac{\partial y_{i,j}}{\partial u_{p,q}} - \frac{\partial y_{k,j}}{\partial u_{p,q}}\right) \cdot (y_{k,j} - y_{i,j}) \end{aligned} \tag{27}$$

where the dependency on u of $x_{i,j}$ and $y_{i,j}$ has been omitted for clarity.

$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = -\triangle t^2 \cdot v \sum_{k=q}^{j-2} \sin\left[\theta_{i,1} + \triangle t \sum_{r=1}^{k} u_{i,r}\right] \quad p = i \quad q \leq (j-2) \tag{28}$$
$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = 0 \quad p = i \quad q > (j-2) \tag{29}$$
$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = 0 \quad p \neq i \tag{30}$$

$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = \triangle t^2 \cdot v \sum_{k=q}^{j-2} \cos\left[\theta_{i,1} + \triangle t \sum_{r=1}^{k} u_{i,r}\right] \quad p = i \quad q \leq (j-2) \tag{31}$$
$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = 0 \quad p = i \quad q > (j-2) \tag{32}$$
$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = 0 \quad p \neq i \tag{33}$$

These values can easily be computed using the closed form of $x_{i,j}$ and $y_{i,j}$ as a function of the initial conditions and the input sequence u of equations (11) and (12).

IV. Results

IV.A. Conflict Resolution

The collision avoidance portion of the algorithm was first tested in an environment without any convective weather. The results are shown in Figure 5 for different instants in time. One unit on the axis of Figure 5 corresponds to a distance of 10 km. Thus, given that the speed of the aircraft is set to 230 m.s^{-1}, based on previous work,[25] the travel time of each aircraft is approximately 75 minutes. The aircraft are set up in a circular configuration such that they would all potentially collide at the same point. For this particular simulation, a horizon of 1024 seconds and a time-step of 128 seconds was used. Thus the number of discrete points along the horizon is equal to 9. Given that the separation constraint is only enforced at the discrete points along the horizon, decreasing the time step decreases the risk of having LOS between these discrete points. The circle around the current position of each aircraft in Figure 5 corresponds to the protected zone of the aircraft. The actual horizontal separation constraint between aircraft is 5 nautical miles which corresponds to 9.26km. For the purposes of visual representation, this separation constraint was increased to 5 units on the graph, which equals 50km. The physical interpretation of the 50km choice is that the aircraft have to be separated by 50km (rather than 5nm = 9.26km). This test is very standard in the multiple vehicle motion coordination literature.[35,20,36,37,3,38]

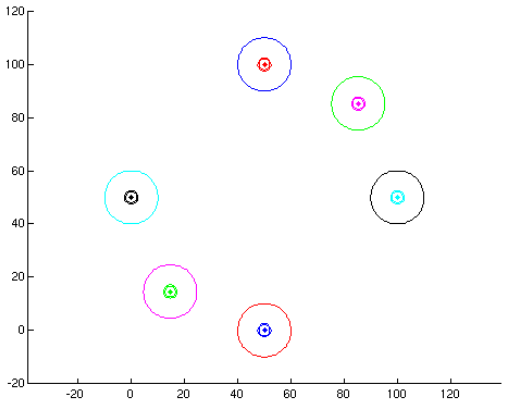
A more realistic configuration is shown in Figure 6 with 8 aircraft. Florida sectors *ZMA64* and *ZMA65* are considered. The origin destination pairs are taken to be the midpoint of opposing sides of a sectors. In this case, a horizon of 64 seconds and time-step of 8 seconds was used in order to be consistent with the simulation shown in Figure 7, where the weather information is included. A shorter time-step was found to be necessary when including the weather information into our optimization routine because of the coarseness of the data. Indeed in order to cover an area of 400 by 400 km for the purpose of including several sectors in Florida, there is a limit to the number of interpolations, which can be performed to reduce the coarseness of the weather data, in order to maintain a reasonably sized matrix. We found that above a time-step of 8 seconds, the optimization routine does not perform as well. In order to avoid increasing the number of discrete points along our horizon and thus increasing the computation time, we decided to reduce the horizon length as well to maintain the number of discrete points at 9. Given that the speed of an aircraft is set to 230 $m.s^{-1}$, it will cover a distance of 14.72km over a horizon of 64 seconds. Thus two aircraft traveling in opposite directions towards each other will first detect a potential conflict when they are 29.44km apart. While this is a relatively small margin, it is still greater than the 9.26km separation they need to maintain between them. As shown in Figure 6, all potential collisions are avoided. For visual clarity, once an aircraft reached its destination, its trajectory is removed from the graph.

IV.B. Conflict Resolution and Convective Weather Avoidance

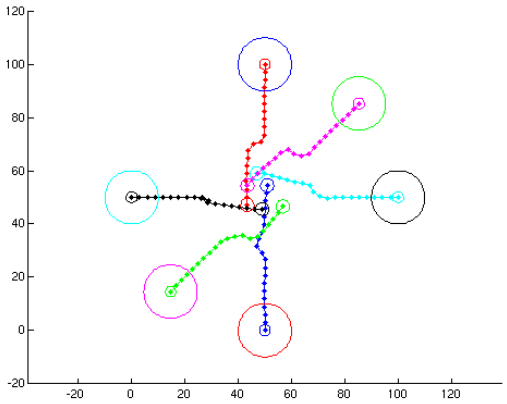
Figure 7 reproduces the same configuration and same separation constraints as the simulation shown in Figure 6 but with 4 aircraft instead of 8. Another difference is that weather information is now included in the optimization routine. The same horizon length (64 seconds) and time-step (8 seconds) was used. The computation time is significantly increased when the weather data is included. Thus for the purpose of this draft only 4 aircraft were included in the simulation. However our goal is to increase the number of aircraft to a more realistic number corresponding to en route constraints for the final version of this paper. As shown on Figure 7, all aircraft successfully avoid convective weather regions as well as other aircraft. Please refer to the following url to view animations and more details concerning these simulations: `http://www.ce.berkeley.edu/ bayen/gnc07/gnc07.htm`

V. Conclusion

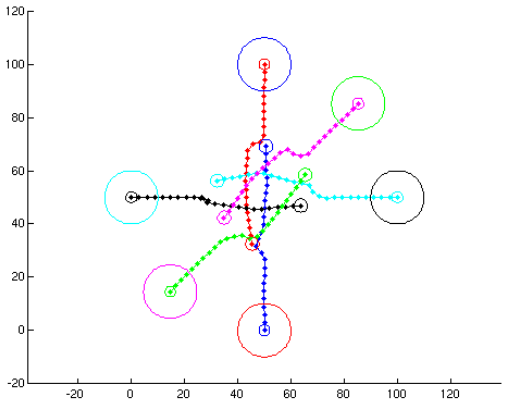
We used a model predictive control framework to solve the minimum time trajectory for multiple aircraft subject to convective weather avoidance constraints. We incorporated a computational technique based on level set methods to determine the objective function of our model predictive control optimization routine. Solving the Hamilton-Jacobi equation for a given destination and weather information (wind profile and convective weather data) provided us with a value function that represents the minimum travel time from any point in the NAS to the specified destination, including necessary deviations due to convective weather avoidance. Minimizing the sum of the value functions of each aircraft subject to collision avoidance constraints led



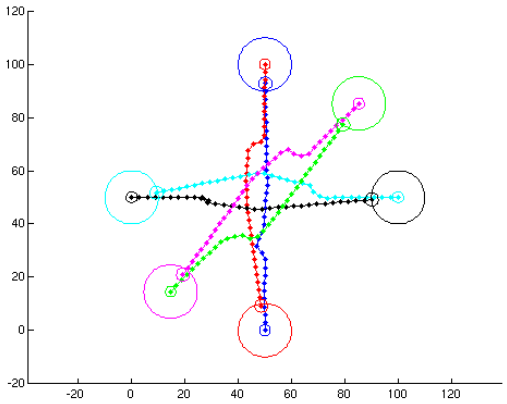
(a) Initial Conditions. All aircraft are located at their respective origins. The larger circles correspond to the destinations of each aircraft. The destination of each aircraft is diametrically opposed to its corresponding origin.



(b) 20 time steps (42 min, 36 sec).Each aircraft has deviated from its optimal trajectory to avoid conflicts with the surrounding aircraft.

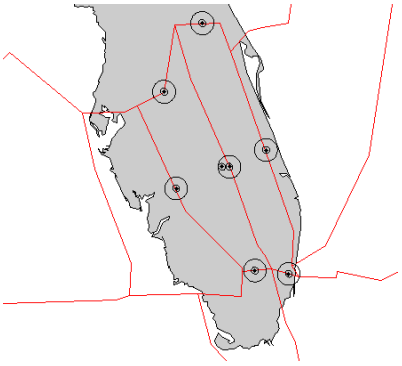


(c) 25 time steps (53 min, 18 sec). Once the conflict is resolved, each aircraft travels towards its destination in a straight line.

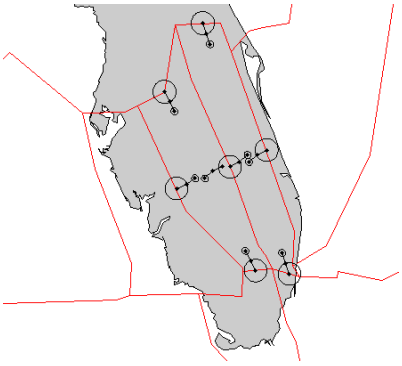


(d) 35 time steps (74 min, 42 sec). All aircraft have reached their target destination.

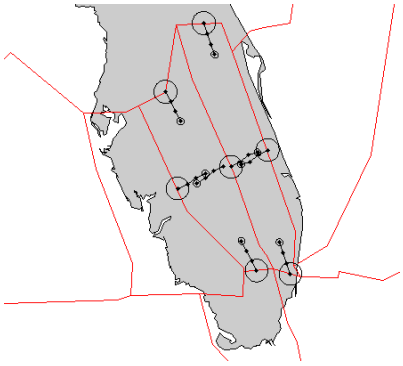
Figure 5. Trajectories generated by the algorithm of section III.A algorithm for 6 aircraft with no weather information. (1 unit = 10km). The larger circles correspond to the aircraft destinations while the smaller circles located at the current position of each aircraft refers to the protected zone around an aircraft, which must not be entered by another aircraft.



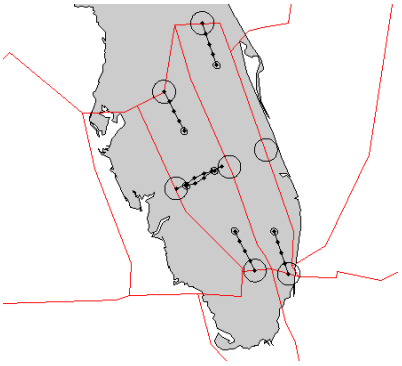
(a) Initial Conditions. All aircraft are located at their origin, at the entry point of a sector. The destination of each aircraft is located on the opposite side of the corresponding sector with respect to the origin.



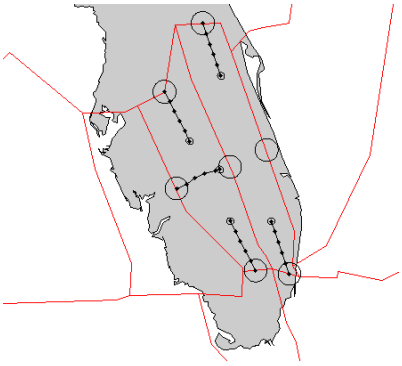
(b) 16 time steps (2 min, 8 sec). The first conflict is resolved by the pair of aircraft traveling east-west in sector ZMA65 (sector on the right) by deviations of the concerned aircraft.



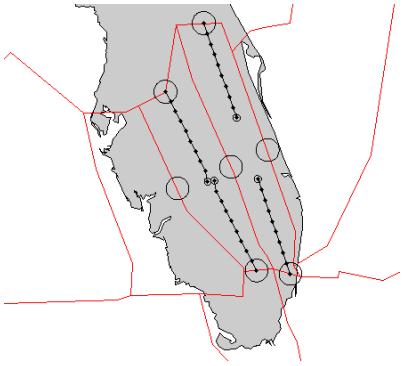
(c) 24 time steps (3 min, 12 sec). The first two aircraft have reached their destination and will therefore be removed from the next figures.



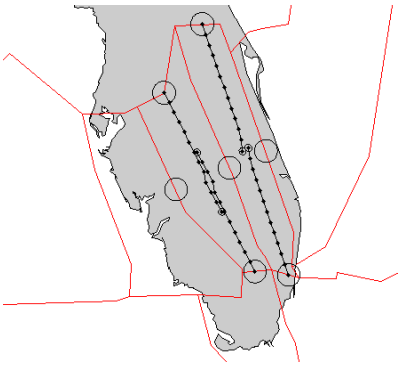
(d) 32 time steps (4 min, 15 sec). The remaining aircraft traveling west has reached its destination while avoiding the aircraft traveling in the opposite direction.



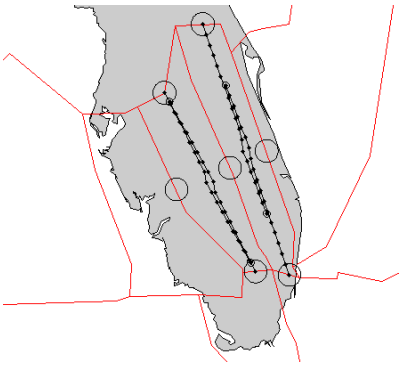
(e) 40 time steps (5 min, 20 sec). The aircraft traveling east has reached its destination.



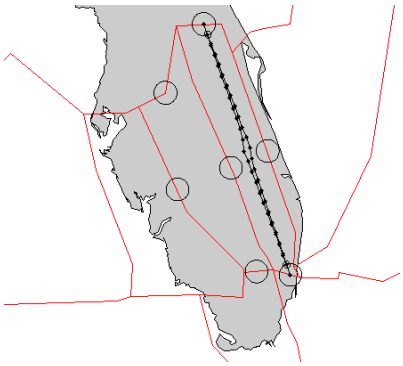
(f) 72 time steps (9 min, 36 sec). A collision is avoided in between two aircraft due to proper deviations.



(g) 96 time steps (12 min, 56 sec). The last conflict between the two aircraft traveling north-south is resolved by aircraft deviations.

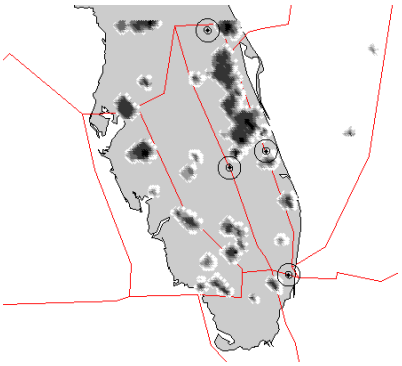


(h) 144 time steps (19 min, 12 sec). The first pair of aircraft traveling north-south has reached their destination.

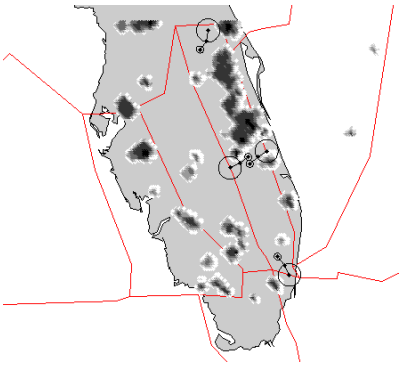


(i) 184 time steps (24 min, 30 sec). Finally the last two aircraft arrive at their destination.

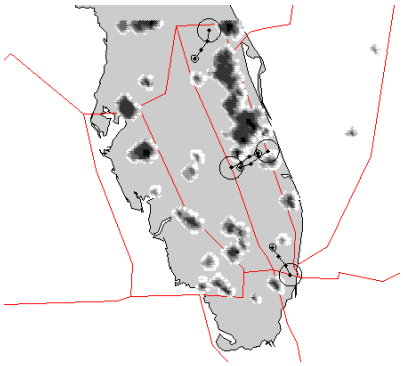
Figure 6. Trajectories generated by the algorithm of section III.A for 8 aircraft with no weather information. The larger circles correspond to the aircraft destinations while the smaller circles located at the current position of each aircraft refers to the protected zone around an aircraft, which must not be entered by another aircraft.



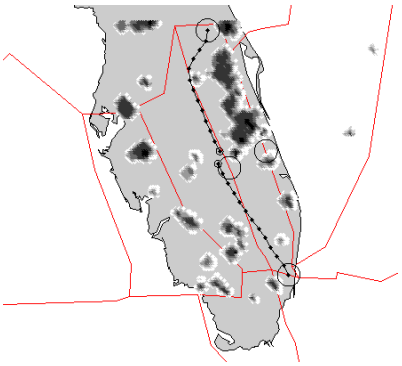
(a) Initial Conditions. All aircraft are located at their origin, at the entry point of a sector. The destination of each aircraft is located on the opposite side of the corresponding sector with respect to the origin.



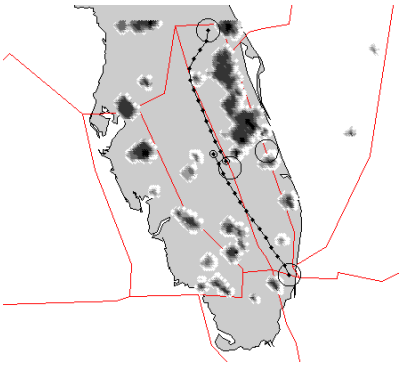
(b) 16 time steps (2 min, 8 sec). The first conflict is resolved by the pair of aircraft traveling east-west in sector ZMA65 (sector on the right) by deviations of the concerned aircraft.



(c) 24 time steps (3 min, 12 sec). The first two aircraft have reached their destination and will therefore be removed from the next graphs.



(d) 96 time steps (12 min, 56 sec). The two aircraft traveling in opposite direction place themselves to such that they avoid each other.



(e) 104 time steps (13 min, 9 sec). The two aircraft have successfully avoided each other.

(f) 192 time steps (25 min, 36 sec). The remaining two aircraft have successfully reached their destination.

Figure 7. Trajectories generated by the algorithm of section III.A for 4 aircraft with weather avoidance. 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.

to a locally optimal trajectories for the set of aircraft considered. Currently the algorithm performs well for a small time-step and relatively short horizon. All aircraft reach their destination without entering regions of convective weather as well as avoiding any LOS. However, this short horizon may present a problem when a larger number of aircraft need to converge through a small area to avoid convective regions. The tradeoff between processing time and horizon length needs to be assessed to determine if a longer horizon is worth the increase in computation time.

The next goal of this work is to incorporate time-varying weather patterns into the path planning optimization routine, which will behave as moving obstacles. As soon as new weather data is available, the Hamilton-Jacobi equation will be solved with the new set of data to obtain an updated value function for each aircraft. These new value functions will then be passed on to the model predictive control routine to compute the objective function.

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, a similar yet decentralized protocol will be designed.

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]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.

[15]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.

[16]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.

[17]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.

[18]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.

[19]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.

[20]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.

[21]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.

[22]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.

[23]Jardin, M. R., *Toward Real-Time En Route Air Traffic Control Optimization*, Ph.D. thesis, Stanford University, 2003.

[24]Lidén, S., "Optimum Cruise Profiles in the Presence of Winds," *IEEE/AIAA* 11^{th} *Digital Avionics Systems Conference*, 1992, pp. 254–261.

[25]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.

[26]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.

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

[28]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.

[29]Sethian, J., *Fast Marching Methods and Level Set Methods*, Cambridge University Press, 1999.

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

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

[32]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.

[33]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.

[34]Kim, H., Shim, D., and Sastry, S., "Flying robots: modeling, control and decision making," *IEEE International Conference on Robotics and Automation*, Vol. 1, 2002, pp. 66–71.

[35]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.

[36]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.

[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.