




Nonlinear Model Predictive Control Applied to Multiple Aircraft Deconflicted Path Planning with Weather Avoidance Constraints

Jessica Pannequin

Department of Electrical Engineering and Computer Science

University of California, Berkeley

Research Advisors: Prof. Alexandre Bayen and Prof. Shankar Sastry.

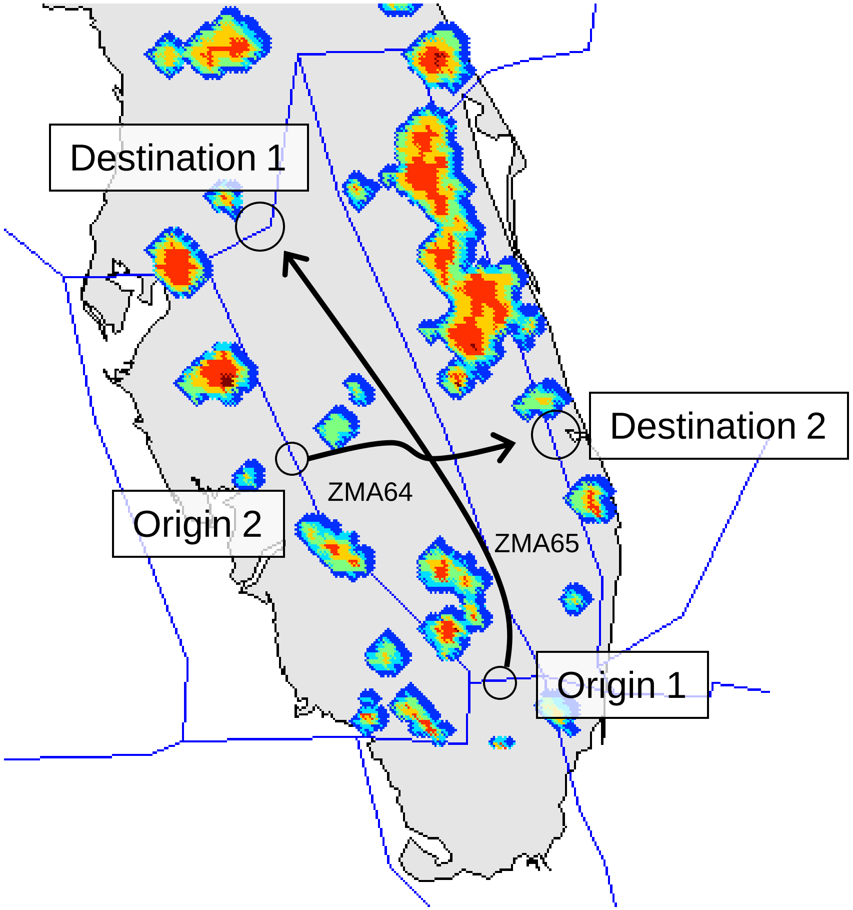
Outline

- Introduction
- Convective Weather Data
- Problem Formulation
- Algorithm Overview
  - Nonlinear Model Predictive Control.
  - Dynamics Discretization.
  - Objective Function.
  - Constraints.
- Results
- Conclusion and Future Work

Introduction

- Aircraft Routing
- Constraints
  - Minimum inter-aircraft separation.
  - Convective Weather Avoidance.
- <u>Goal:</u>

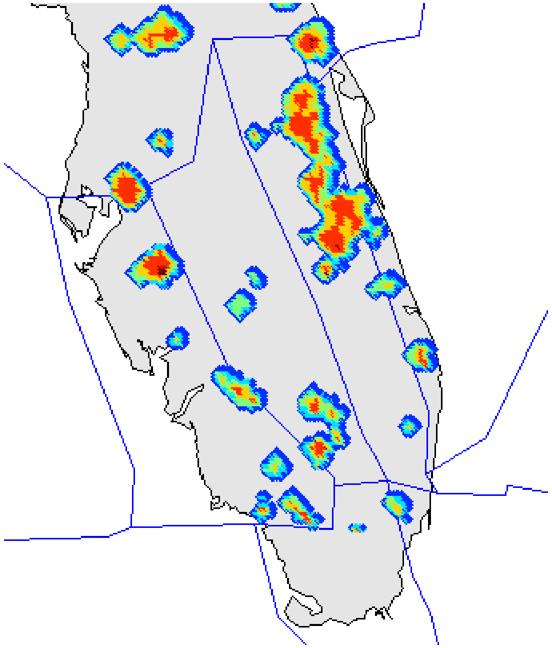
  Compute optimal deconflicted trajectories for multiple aircraft in the presence of convective weather.



Convective Weather Data

- Data obtained from National Center for Atmospheric Research (NCAR)
- VIL (Vertically Integrated Liquid) Algorithm: Weather radar reflectivity data is converted into liquid-water content for a sample volume.
- VIL data matrix, available every 5 minutes:
  - 1830 by 918 elements
  - Each entry covers an area of $16km^2$.
- VIL data can be converted to discrete VIP (Video Integrator and Processor) levels.

VIL (Kg/m^2)	VIP Level	Colormap
0.14	1	Blue
0.7	2	Turquoise
3.5	3	Green
6.9	4	Yellow
12.0	5	Orange
32.0	6	Red



Problem Formulation

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.

Aircraft Dynamics

$$\dot{\bar{x}}_i(t) = v\cos(\bar{\theta}_i(t))$$

$$\dot{\bar{y}}_i(t) = v\sin(\bar{\theta}_i(t))$$

$$\dot{\bar{\theta}}_i(t) = \bar{u}_i(t)$$

Problem Formulation

$$\min_{\bar{\mathbf{u}}} \; J(\mathbf{x_0}, \bar{\mathbf{u}}(\cdot)) = \sum_{i=1}^{N} J_i(\mathbf{x}_{\mathbf{i},0}, \bar{u}_i(\cdot))$$

$$\text{s.t.} \quad -a \le \bar{u}_i(t) \le a \qquad i \in [1,N] \qquad t \in [0,T]$$

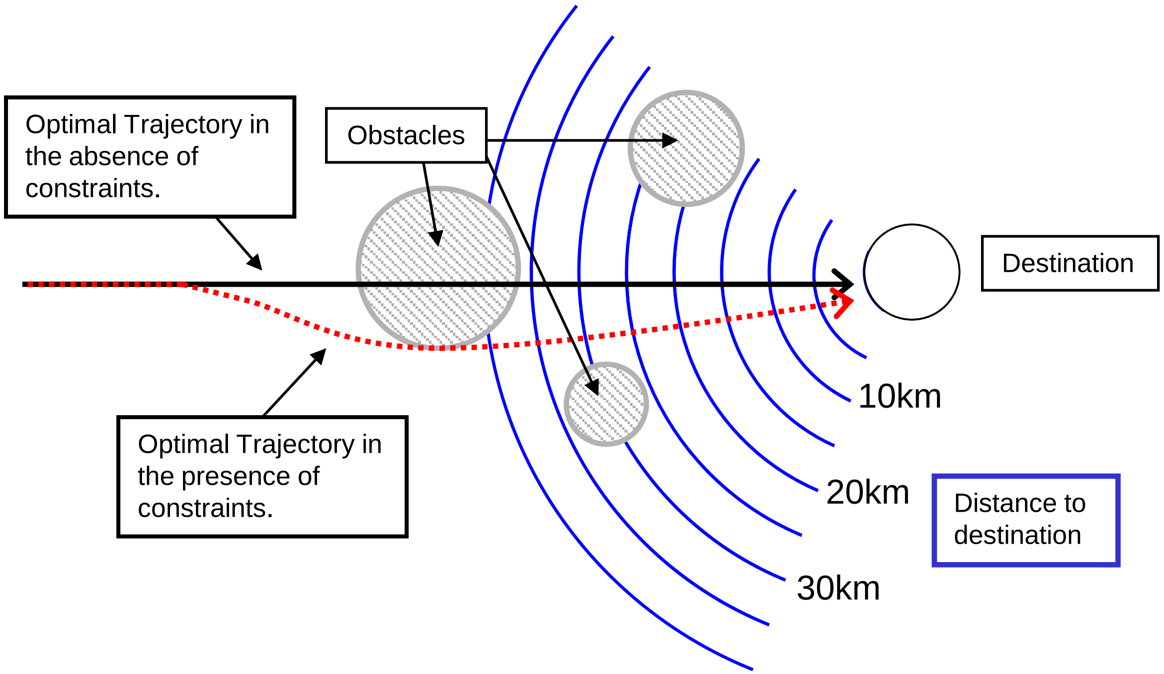
$$(\bar{x}_i(t) - \bar{x}_k(t))^2 + (\bar{y}_i(t) - \bar{y}_k(t))^2 \ge r_{\min}^2 \qquad i \ne k \in [1,N] \qquad t \in [0,T]$$

$$(\bar{x}_i(t), \bar{y}_i(t)) \notin W \qquad i \in [1,N] \qquad t \in [0,T]$$

$$\dot{\bar{x}}_i(t) = v\cos(\theta_i(t)) \qquad i \in [1,N] \qquad t \in [0,T]$$

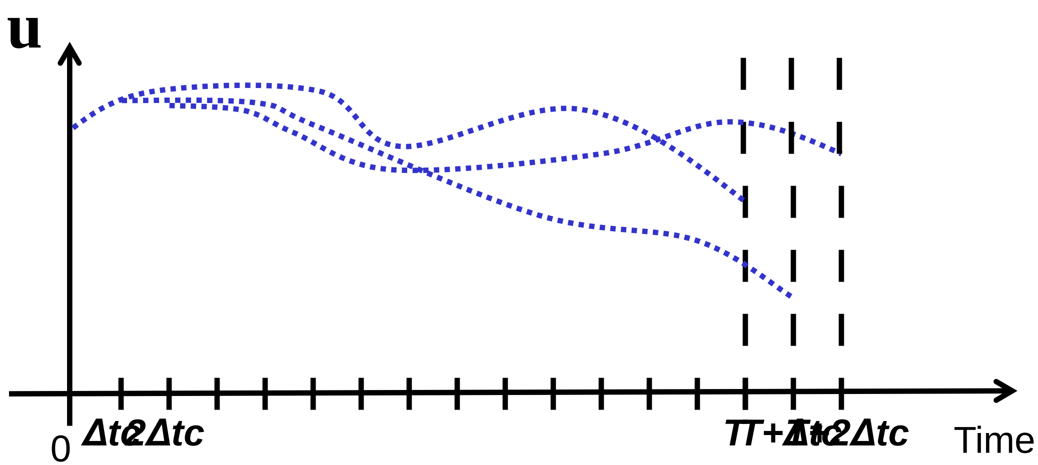
$$\dot{\bar{y}}_i(t) = v\sin(\theta_i(t)) \qquad i \in [1,N] \qquad t \in [0,T]$$

$$\dot{\bar{\theta}}_i(t) = \bar{u}_i(t) \qquad i \in [1,N] \qquad t \in [0,T]$$



Algorithm

- Algorithm is based on nonlinear model predictive control (NMPC)
- NMPC is defined by the computation, at every time step ***Δtc***, of optimal control sequence over a finite horizon ***T***.
- <u>Advantage</u>: capacity to deal with nonlinearity subject to hard constraints.



Algorithm

- Algorithm is based on nonlinear model predictive control (NMPC)
- NMPC is defined by the computation, at every time step ***Δtc***, of optimal control sequence over a finite horizon ***T***.
- <u>Advantage</u>: capacity to deal with nonlinearity subject to hard constraints.

INITIALIZATION → MAIN LOOP → LOCALLY OPTIMAL TRAJECTORIES

INITIALIZATION

- Horizon Length ***T.***
- Discretization time step ***Δt.***
- MPC time step ***Δtc.***
- Number of Aircraft ***N.***
- Origin-Destination pairs.
- Aircraft velocity ***v.***

Load Weather Data.

Discretize Dynamics.

MAIN LOOP

Inputs :

- Initial control guess u_0.
- Objective function & Gradient.
- Constraints & Jacobian.

TOMLAB SOLVER (SNOPT)

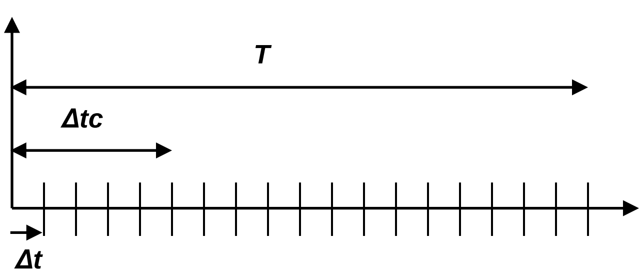
Ouput :

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

- Update Trajectories.
- Remove aircraft when target reached.

Advance to next time step by ***Δtc*** time intervals.

Repeat until all aircraft have reached their destination.



Dynamics Discretization

Algorithm
- NMPC
- **Dynamics Discretization**
- Objective Function
- Constraints

$$\dot{x}_i(t) = v\cos(\overline{\theta}_i(t))$$
$$\dot{y}_i(t) = v\sin(\overline{\theta}_i(t))$$
$$\dot{\overline{\theta}}_i(t) = \overline{u}_i(t)$$

Euler Discretization → at time step j =1.

$$x_{i,1} = x_{i,0} + \Delta t \cdot v\cos(\theta_{i,0})$$
$$y_{i,1} = y_{i,0} + \Delta t \cdot v\sin(\theta_{i,0}) \qquad i \in [1, N]$$
$$\theta_{i,1} = \theta_{i,0} + \Delta t \cdot u_{i,0}$$

Notation:
- $(x_{i,0}, y_{i,0}, \theta_{i,0})$ Initial state of aircraft i.
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $u_i = [u_{i,0} \ldots u_{i,n-1}]$ Control sequence of aircraft i.
- N Total # of aircraft.
- $n = (T/\Delta t) + 1$ # of discrete time steps.

Dynamics Discretization

Algorithm
- NMPC
- **Dynamics Discretization**
- Objective Function
- Constraints

$$\dot{\bar{x}}_i(t) = v\cos(\bar{\theta}_i(t))$$
$$\dot{\bar{y}}_i(t) = v\sin(\bar{\theta}_i(t))$$
$$\dot{\bar{\theta}}_i(t) = \bar{u}_i(t)$$

Euler Discretization → at time step j =1.

$$x_{i,1} = x_{i,0} + \Delta t \cdot v\cos(\theta_{i,0})$$
$$y_{i,1} = y_{i,0} + \Delta t \cdot v\sin(\theta_{i,0}) \qquad i \in [1, N]$$
$$\theta_{i,1} = \theta_{i,0} + \Delta t \cdot u_{i,0}$$

Generalization for all time steps ↓

Aircraft i, Time step j

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

$$y_{i,j} = y_{i,0} + \Delta t \cdot v\sin(\theta_{i,0}) + \Delta t \cdot v\sum_{k=0}^{j-2}\sin\left(\theta_{i,0} + \Delta t\sum_{r=0}^{k} u_{i,r}\right) \qquad i \in [1, N] \quad j \in [2, n-1]$$

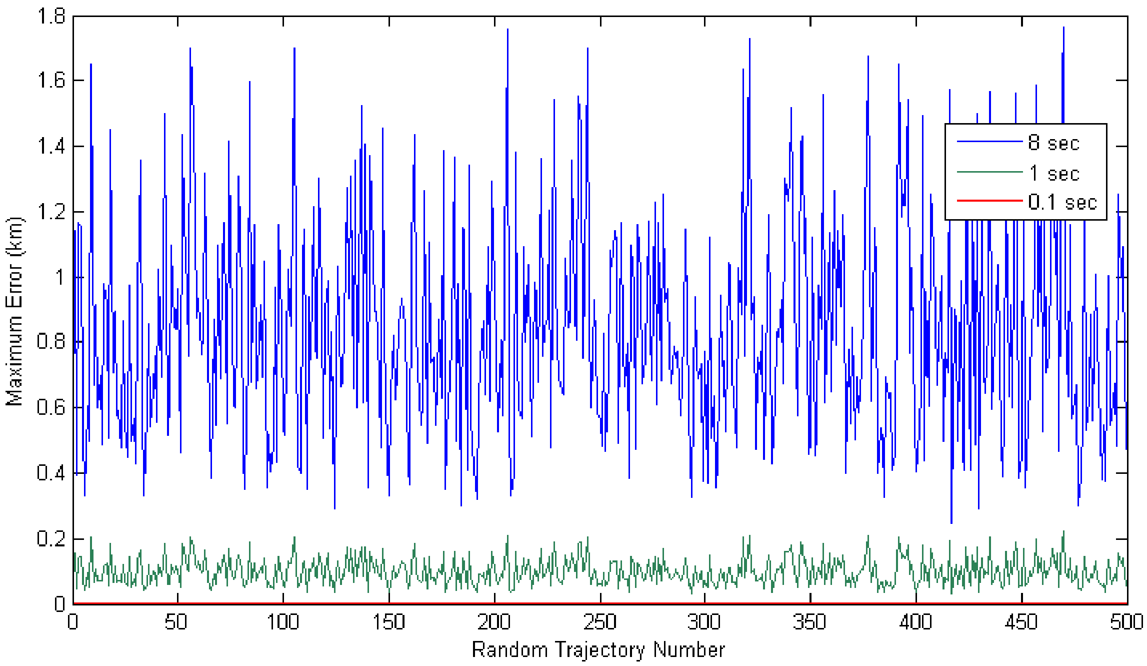
$$\theta_{i,j} = \theta_{i,0} + \Delta t\sum_{r=0}^{j-1} u_{i,r}$$

Notation:
- $(x_{i,0}, y_{i,0}, \theta_{i,0})$ Initial state of aircraft i.
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $u_i = [u_{i,0} \ldots u_{i,n-1}]$ Control sequence of aircraft i.
- N Total # of aircraft.
- $n = (T / \Delta t) + 1$ # of discrete time steps.

Dynamics Discretization

Algorithm
- NMPC
- **Dynamics Discretization**
- Objective Function
- Constraints

- Error Bound Analysis
  - 500 random trajectories

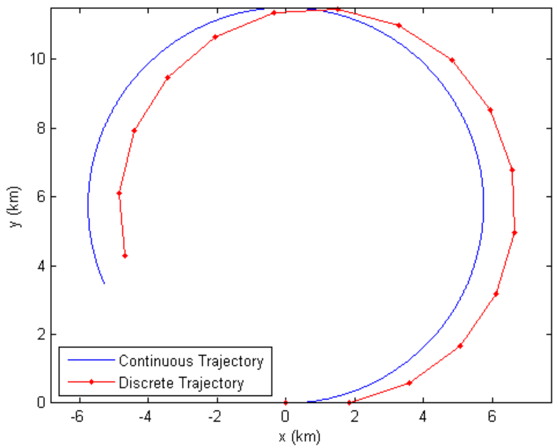


Δt (s)	Max error (km)	Max error (nm)
8	1.78	0.96
1	0.22	0.12
0.1	≈ 0	≈ 0

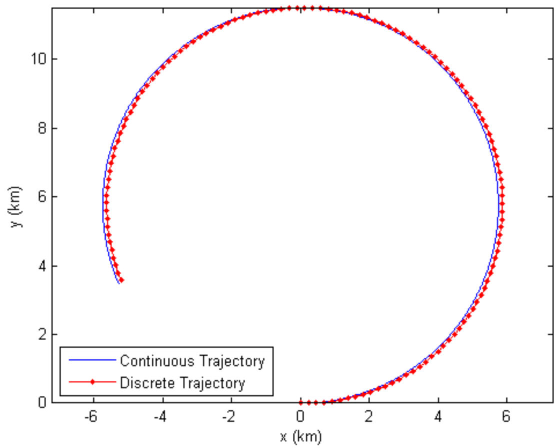
Dynamics Discretization

Algorithm
- NMPC
- **Dynamics Discretization**
- Objective Function
- Constraints

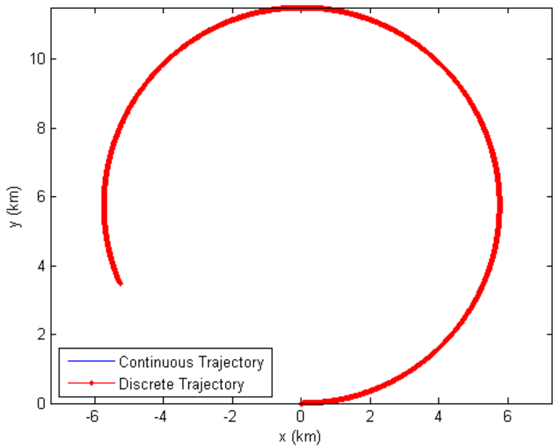
- Error Bound Analysis
  - Experimental Worst Case : Circular Trajectory



Δt = 8 seconds



Δt = 1 seconds



Δt = 0.1 seconds

Δt (s)	8	1	0.1
Max error (km)	1.84	0.23	≈ 0
Max error (nm)	0.99	0.12	≈ 0

Objective Function: First Approach

Quadratic Objective Function:

$$J(\mathbf{x}_0,\mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0},u_i)$$

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

Penalizes remaining distance to target

Penalizes aircraft turning rate

Algorithm

- NMPC
- Dynamics Discretization
- **Objective Function**
  - **1st Approach**
  - 2nd Approach
  - Comparison
- Constraints

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.
- (x_i^t, y_i^t) Target center of aircraft i.

Objective Function: First Approach

Quadratic Objective Function:

$$J(\mathbf{x}_0,\mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0},u_i)$$
$$= \sum_{i=1}^{N} (x_{i,n-1}(\mathbf{x}_{i,0},u_i) - x_i^t)^2 + (y_{i,n-1}(\mathbf{x}_{i,0},u_i) - y_i^t)^2 + \frac{1}{2}\mathbf{u}^T R\mathbf{u}$$

Objective Function Gradient

$$\frac{\partial J(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = 2\begin{bmatrix} \sum_{i=1}^{N} (x_{i,n-1} - x_i^t)\cdot \frac{\partial x_{i,n-1}}{\partial u_{1,0}} + (y_{i,n-1} - y_i^t)\cdot \frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N} (x_{i,n-1} - x_i^t)\cdot \frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + (y_{i,n-1} - y_i^t)\cdot \frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}$$

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.
- (x_i^t, y_i^t) Target center of aircraft i.

Objective Function: First Approach

Algorithm
- NMPC
- Dynamics Discretization
- **Objective Function**
  - **1st Approach**
  - 2nd Approach
  - Comparison
- Constraints

Quadratic Objective Function:

$$J(\mathbf{x}_0,\mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0},u_i)$$

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

Objective Function Gradient:

$$\frac{\partial J(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = 2\begin{bmatrix} \sum_{i=1}^{N}(x_{i,n-1}-x_i^t)\cdot\frac{\partial x_{i,n-1}}{\partial u_{1,0}} + (y_{i,n-1}-y_i^t)\cdot\frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N}(x_{i,n-1}-x_i^t)\cdot\frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + (y_{i,n-1}-y_i^t)\cdot\frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}$$

$$\frac{\partial x_{i,j}}{\partial u_{p,q}} = -\Delta t^2 v \sum_{k=q}^{j-2} \sin(\theta_{i,0} + \Delta t \sum_{r=0}^{k} u_{i,r}) \quad p=i, \quad q \le (j-2)$$

$$\frac{\partial y_{i,j}}{\partial u_{p,q}} = \Delta t^2 v \sum_{k=q}^{j-2} \cos(\theta_{i,0} + \Delta t \sum_{r=0}^{k} u_{i,r}) \quad p=i, \quad q \le (j-2)$$

Notation:
- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.
- (x_i^t, y_i^t) Target center of aircraft i.

Objective Function: Second Approach

Hamilton-Jacobi derived Objective Function:

$$J(\mathbf{x}_0, \mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0}, u_i)$$

$$= \sum_{i=1}^{N} V_i(x_{i,n-1}(\mathbf{x}_{i,0}, u_i), y_{i,n-1}(\mathbf{x}_{i,0}, u_i)) + \frac{1}{2}\mathbf{u}^T R \mathbf{u}$$

Penalizes remaining travel time to target

Penalizes aircraft turning rate

Hamilton-Jacobi Equation:

$$\begin{cases} \max_{u_i}(\nabla V_i(x,y) \cdot f(x,y,u_i) - \ell(x,y)) = 0 & \text{in } \Omega \backslash \mathcal{T}_i \\ V_i(x,y) = 0 & \text{in } \mathcal{T}_i \end{cases}$$

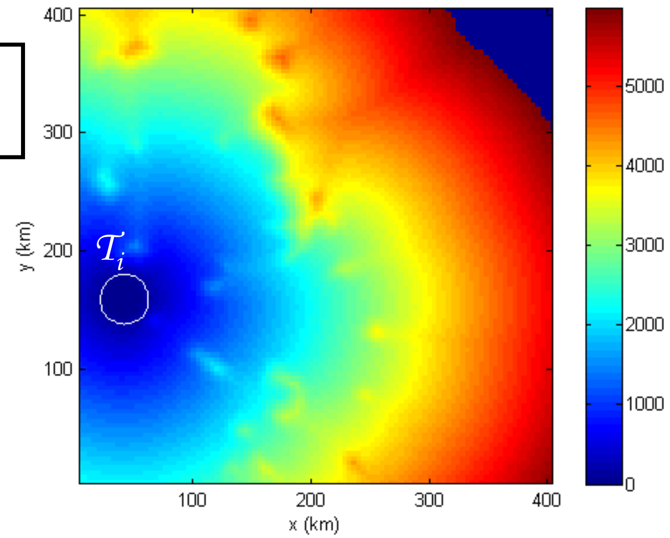
Algorithm
- NMPC
- Dynamics Discretization
- **Objective Function**
  - 1st Approach
  - **2nd Approach**
  - Comparison
- Constraints

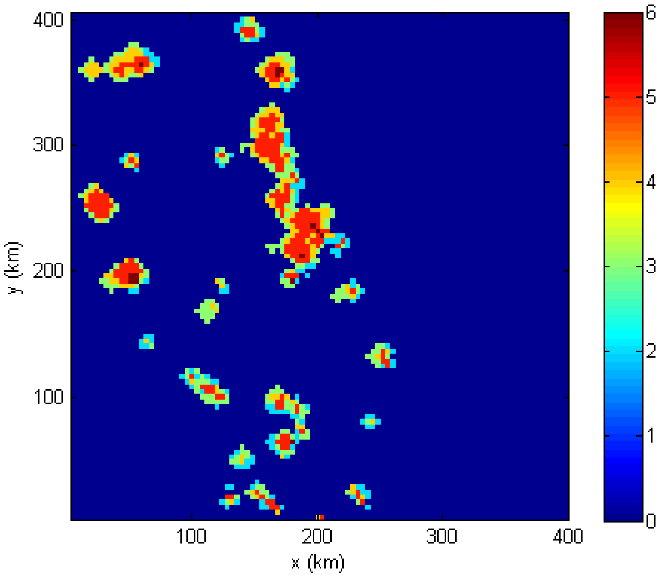
Notation:
- $\mathbf{x}_0 = [\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1, \ldots, u_N]$
- $u_i = [u_{i,0}, \ldots, u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t) + 1$ # of discrete time steps.
- Ω Airspace.
- $\mathcal{T}_i$ Target of aircraft i.

Objective Function: Second Approach

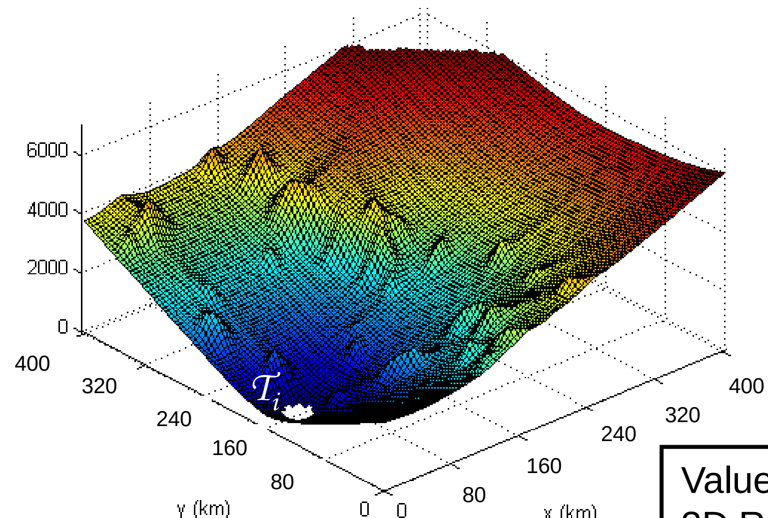
HJ Value Function:

Value function:
2D Representation





VIP Data



Value function:
3D Representation

Algorithm
- NMPC
- Dynamics Discretization
- **Objective Function**
  - 1st Approach
  - **2nd Approach**
  - Comparison
- Constraints

Notation:
$\mathcal{T}_i$ Target of aircraft i.

Objective Function: Second Approach

Hamilton-Jacobi derived Objective Function:

$$J(\mathbf{x}_0,\mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0},u_i)$$

$$= \sum_{i=1}^{N} V_i(x_{i,n-1}(\mathbf{x}_{i,0},u_i), y_{i,n-1}(\mathbf{x}_{i,0},u_i)) + \frac{1}{2}\mathbf{u}^T R\mathbf{u}$$

Hamilton-Jacobi derived Objective Function:

$$\frac{\partial J(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = 2\begin{bmatrix} \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial x_{i,n-1}}\frac{\partial x_{i,n-1}}{\partial u_{1,0}} + \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial y_{i,n-1}}\frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial x_{i,n-1}}\frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial y_{i,n-1}}\frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}$$

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.

Objective Function: Second Approach

Hamilton-Jacobi derived Objective Function:

$$J(\mathbf{x}_0,\mathbf{u}) = \sum_{i=1}^{N} J_i(\mathbf{x}_{i,0},u_i)$$

$$= \sum_{i=1}^{N} V_i(x_{i,n-1}(\mathbf{x}_{i,0},u_i), y_{i,n-1}(\mathbf{x}_{i,0},u_i)) + \frac{1}{2}\mathbf{u}^T R\mathbf{u}$$

Hamilton-Jacobi derived Objective Function:

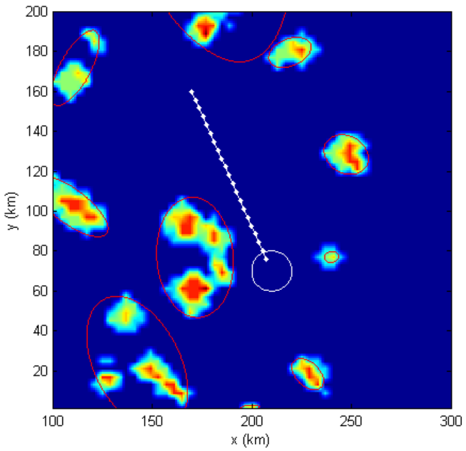
$$\frac{\partial J(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = 2\begin{bmatrix} \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial x_{i,n-1}} \frac{\partial x_{i,n-1}}{\partial u_{1,0}} + \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial y_{i,n-1}} \frac{\partial y_{i,n-1}}{\partial u_{1,0}} \\ \vdots \\ \sum_{i=1}^{N} \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial x_{i,n-1}} \frac{\partial x_{i,n-1}}{\partial u_{N,n-1}} + \frac{\partial V_i(x_{i,n-1},y_{i,n-1})}{\partial y_{i,n-1}} \frac{\partial y_{i,n-1}}{\partial u_{N,n-1}} \end{bmatrix} + R\mathbf{u}$$

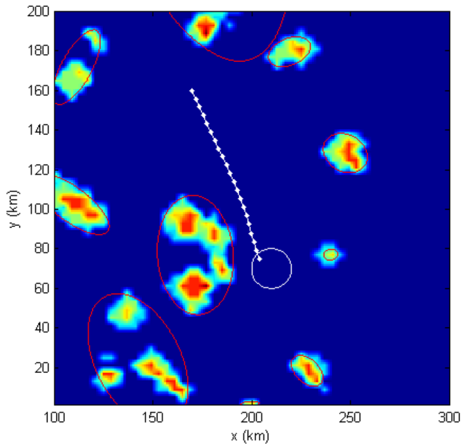
Approximated Numerically

Algorithm
- NMPC
- Dynamics Discretization
- **Objective Function**
  - 1st Approach
  - **2nd Approach**
  - Comparison
- Constraints

Notation:
- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0}, \theta_{i,0})$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.

Objective Function: Comparison



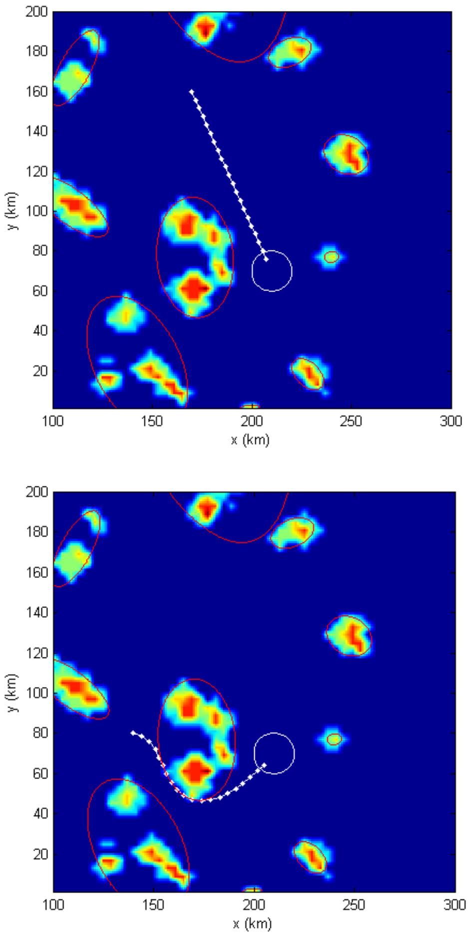


Quadratic Objective Function

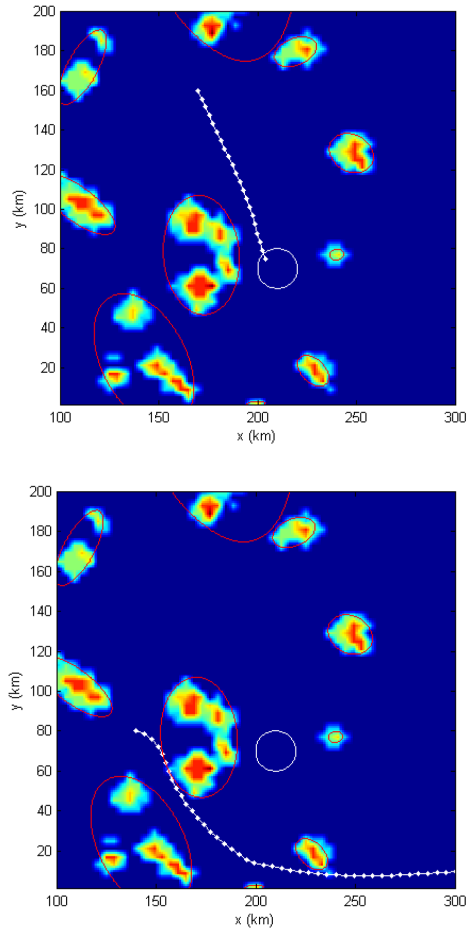
HJ PDE derived Objective Function

Objective Function: Comparison

Algorithm
– NMPC
– Dynamics Discretization
– **Objective Function**
  - 1st Approach
  - 2nd Approach
  - **Comparison**
– Constraints



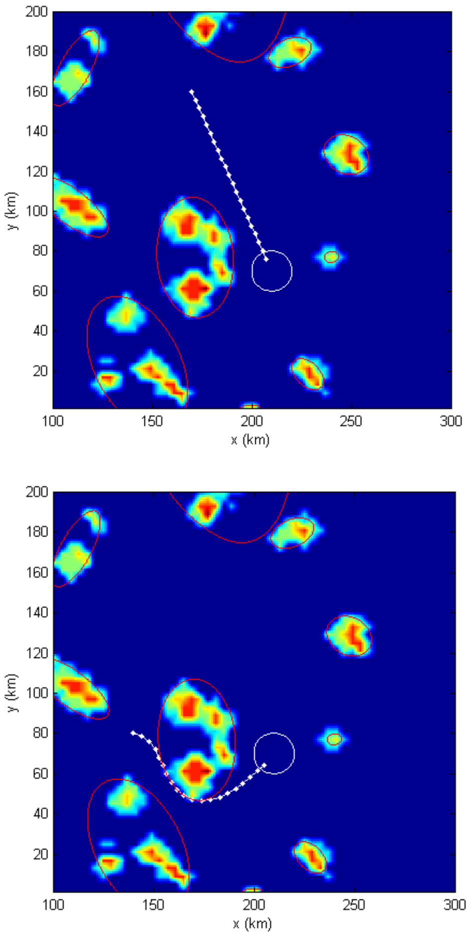
Quadratic Objective Function



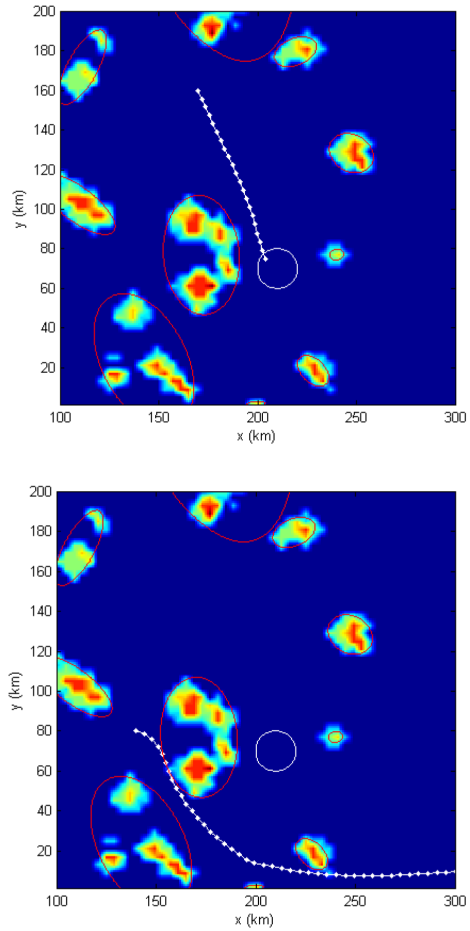
HJ PDE derived Objective Function

Objective Function: Comparison

Algorithm
- NMPC
- Dynamics Discretization
- **Objective Function**
  - 1st Approach
  - 2nd Approach
  - **Comparison**
- Constraints



Quadratic Objective Function



HJ PDE derived Objective Function

	Quadratic Objective Function	HJ PDE derived Objective Function
Advantages	• Well defined gradient. • Shorter computation time. • Scales well.	• Allows for global view of airspace. • Wind easily included.
Drawbacks	• Restricted to local view of airspace.	• Numerically approximated gradient : – Longer computation time. – Sensitivity to changes in aircraft initial position.

Constraints

Algorithm
- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - **Collision**
  - Weather

- All constraints are hard coded.
- Advantages:
  - A feasible solution guarantees a conflict-free path.
  - Avoidance of tedious parameter tuning for repulsive potential fields.
- Disadvantage:
  - Large number of constraints leads to longer computation time.

Constraints Vector:

$$c = [c_1 \ldots c_{Nc}] = [c^{\mathrm{ca}}, c^{\mathrm{w}}]$$

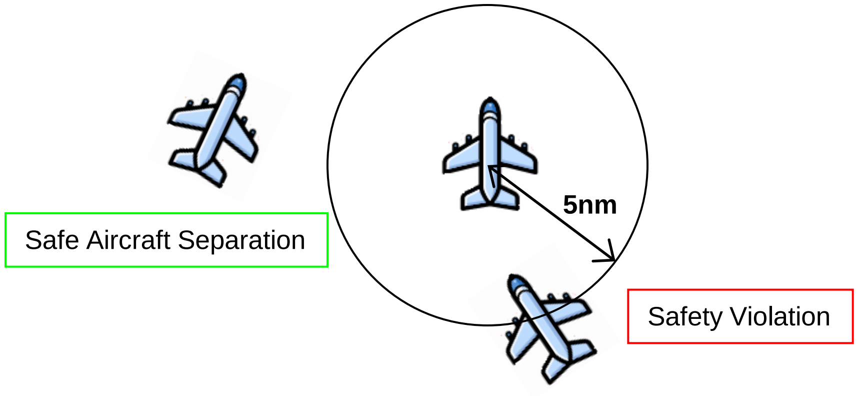
Constraints Jacobian:

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

Notation:
- Nc Total number of constraints
- c^{ca} Collision Avoidance constraints
- c^{w} Weather Avoidance constraints
- N Total # of aircraft
- $n = (T / \Delta t) + 1$ # of discrete time steps

Constraints : Collision Avoidance

Algorithm
- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - **Collision**
  - Weather



Collision Avoidance Constraint between aircraft i and k at time step j.

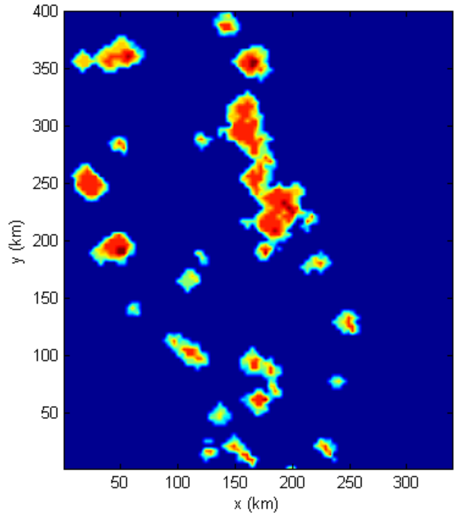
$$c^{\mathrm{ca}}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := r_{\min}^2 - (x_{i,j}-x_{k,j})^2 - (y_{i,j}-y_{k,j})^2 \le 0 \qquad \forall i \ne k \quad \forall j \in [0, n-1]$$

$$\frac{\partial c^{\mathrm{ca}}_{i,j,k}(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = -2 \begin{bmatrix} \left(\frac{\partial x_{i,j}}{\partial u_{1,0}} - \frac{\partial x_{k,j}}{\partial u_{1,0}}\right)(x_{k,j}-x_{i,j}) + \left(\frac{\partial y_{i,j}}{\partial u_{1,0}} - \frac{\partial x_{k,j}}{\partial u_{1,0}}\right)(y_{k,j}-y_{i,j}) \\ \vdots \\ \left(\frac{\partial x_{i,j}}{\partial u_{N,n-1}} - \frac{\partial x_{k,j}}{\partial u_{N,n-1}}\right)(x_{k,j}-x_{i,j}) + \left(\frac{\partial y_{i,j}}{\partial u_{N,n-1}} - \frac{\partial x_{k,j}}{\partial u_{N,n-1}}\right)(y_{k,j}-y_{i,j}) \end{bmatrix}$$

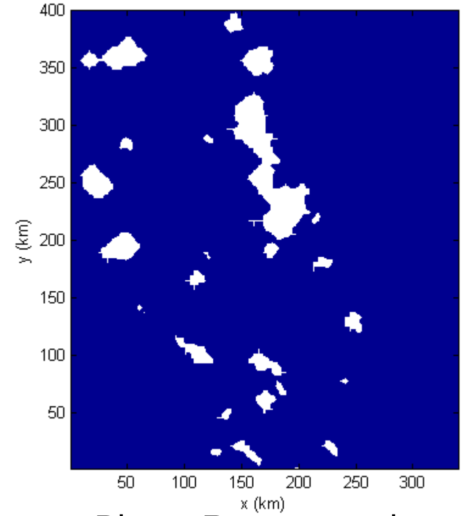
Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1, \ldots, u_N]$
- $u_i = [u_{i,0}, \ldots, u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t) + 1$ # of discrete time steps.

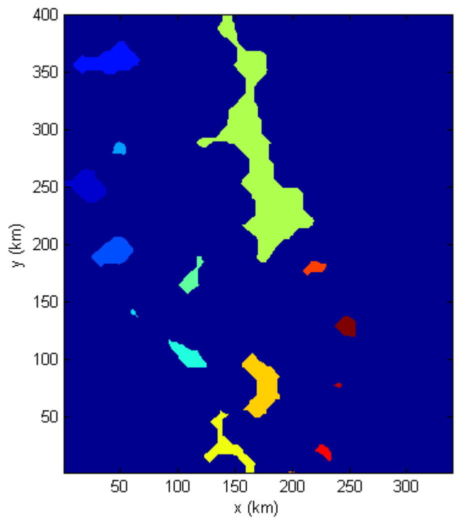
Constraints : Weather Avoidance



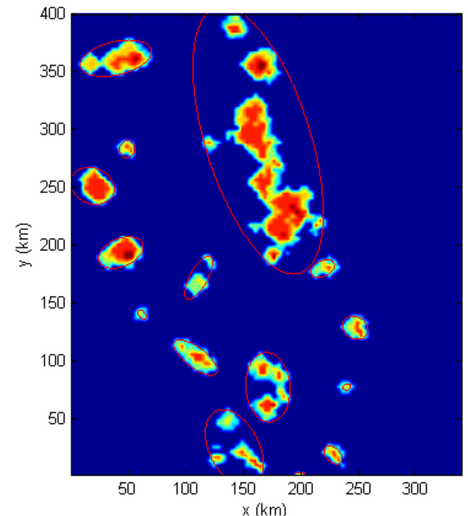
Convective Weather Data



Binary Representation



Weather “Blobs”

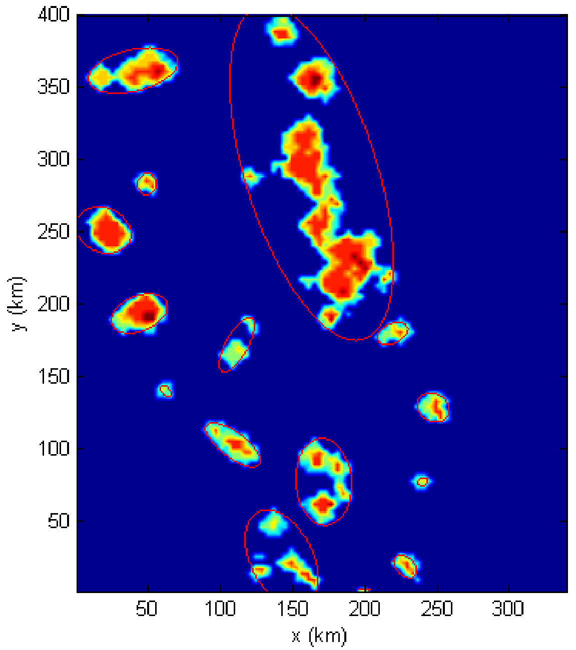


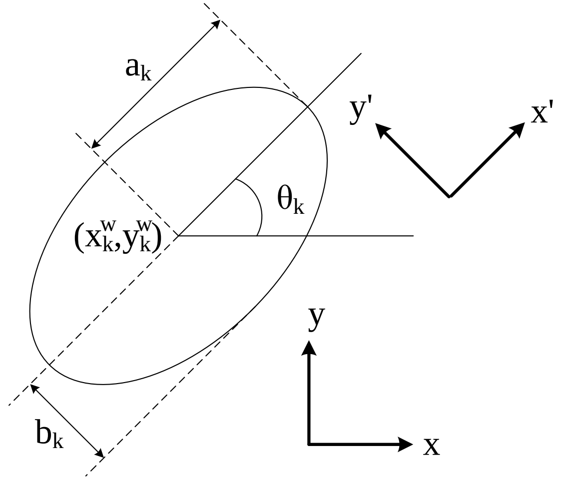
Elliptical Boundaries

Algorithm

- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - Collision
  - **Weather**

Constraints : Weather Avoidance





Constraint enforcing aircraft i to avoid weather cluster k at time step j.

$$c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{(x'_{i,j} - x^{w}_{k})^2}{a_k^2} - \frac{(y'_{i,j} - y^{w}_{k})^2}{b_k^2} \le 0$$

$$i \in [1, N] \quad j \in [0, n-1] \quad \mathrm{k} \in [1, \mathrm{N_k}]$$

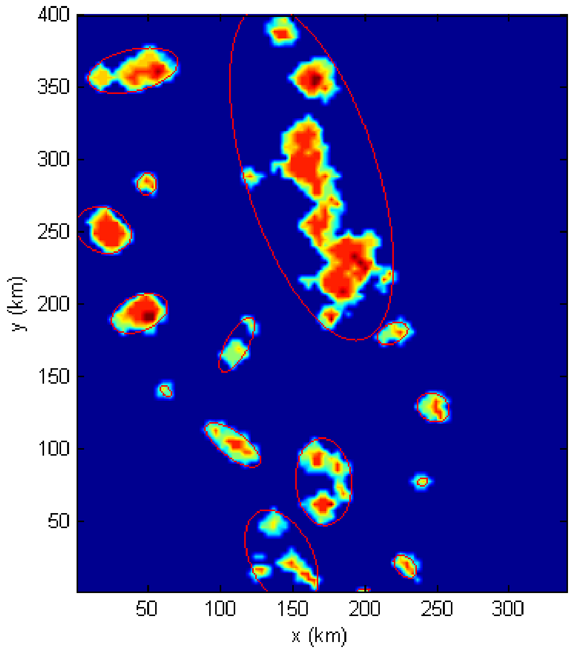
Algorithm

- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - Collision
  - **Weather**

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $(x'_{i,j}, y'_{i,j})$ Position of aircraft i at time step j in ellipse coordinate frame.
- $\mathbf{u} = [u_1, \ldots, u_N]$
- $u_i = [u_{i,0}, \ldots, u_{i,n-1}]$
- N # of aircraft.
- $n = (T / \Delta t) + 1$ # of discrete time steps.
- N_k Total # of convective weather clusters.

Constraints : Weather Avoidance

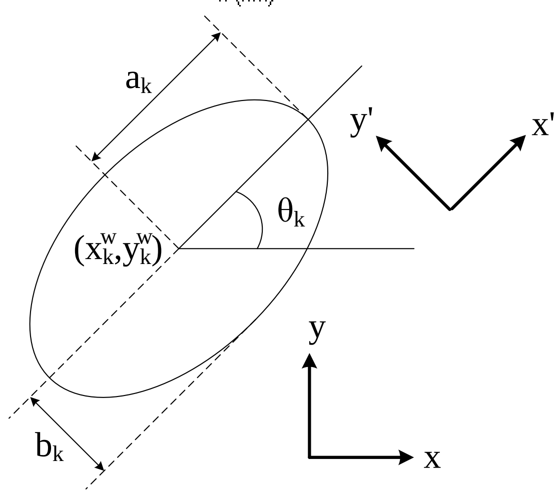


Constraint enforcing aircraft i to avoid weather cluster k at time step j.

$$c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{(x'_{i,j} - x^{w}_{k})^2}{a_k^2} - \frac{(y'_{i,j} - y^{w}_{k})^2}{b_k^2} \le 0$$

$$i \in [1, N] \quad j \in [0, n-1] \quad \mathrm{k} \in [1, \mathrm{N_k}]$$

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

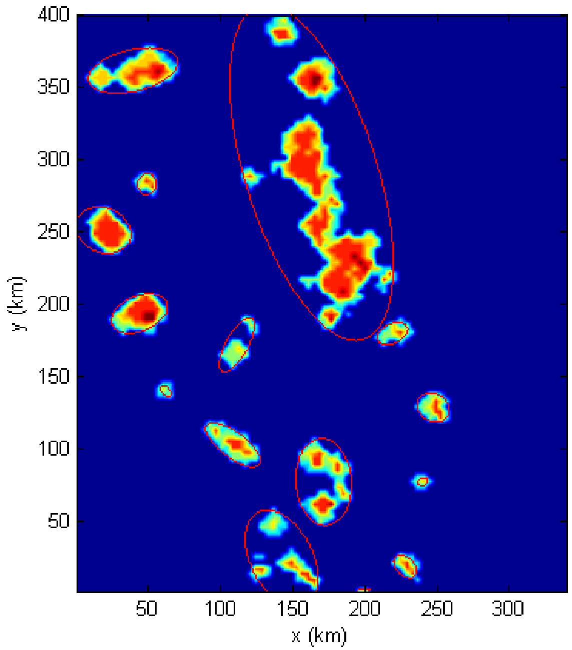


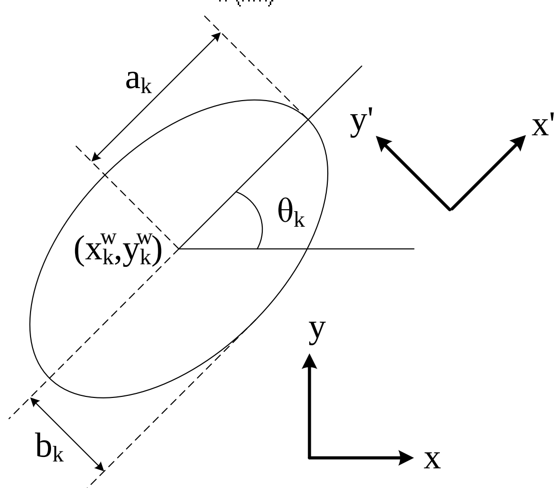
Algorithm
- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - Collision
  - **Weather**

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $(x'_{i,j}, y'_{i,j})$ Position of aircraft i at time step j in ellipse coordinate frame.
- $\mathbf{u} = [u_1, \ldots, u_N]$
- $u_i = [u_{i,0}, \ldots, u_{i,n-1}]$
- N # of aircraft.
- $n = (T / \Delta t) + 1$ # of discrete time steps.
- N_k Total # of convective weather clusters.

Constraints : Weather Avoidance





Constraint enforcing aircraft i to avoid weather cluster k at time step j.

$$c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{(x'_{i,j} - x^{w}_{k})^2}{a_k^2} - \frac{(y'_{i,j} - y^{w}_{k})^2}{b_k^2} \le 0$$

$$i \in [1, N] \quad j \in [0, n-1] \quad \mathrm{k} \in [1, \mathrm{N_k}]$$

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

$$\Downarrow$$

$$c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{\left[(x_{i,j} - x^{w}_{k})\cos(\theta_k) + (y_{i,j} - y^{w}_{k})\sin(\theta_k)\right]^2}{a_k^2} - \frac{\left[(y_{i,j} - y^{w}_{k})\cos(\theta_k) - (x_{i,j} - x^{w}_{k})\sin(\theta_k)\right]^2}{b_k^2} \le 0$$

$$i \in [1, N] \quad j \in [0, n-1] \quad \mathrm{k} \in [1, \mathrm{N_k}]$$

Algorithm
- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - Collision
  - **Weather**

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0}, \ldots, \mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $(x'_{i,j}, y'_{i,j})$ Position of aircraft i at time step j in ellipse coordinate frame.
- $\mathbf{u} = [u_1, \ldots, u_N]$
- $u_i = [u_{i,0}, \ldots, u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t) + 1$ # of discrete time steps.
- N_k Total # of convective weather clusters.

Constraints : Weather Avoidance

Constraint enforcing aircraft i to avoid weather cluster k at time step j.

$$c^{\mathrm{w}}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{\left[(x_{i,j}-x^{\mathrm{w}}_k)\cos(\theta_k)+(y_{i,j}-y^{\mathrm{w}}_k)\sin(\theta_k)\right]^2}{a_k^2} - \frac{\left[(y_{i,j}-y^{\mathrm{w}}_k)\cos(\theta_k)-(x_{i,j}-x^{\mathrm{w}}_k)\sin(\theta_k)\right]^2}{b_k^2} \le 0$$

$$i\in[1,N] \quad j\in[0,n-1] \quad \mathrm{k}\in[1,\mathrm{N_k}]$$

⇓ Gradient Computation

$$\frac{\partial c^{\mathrm{w}}_{i,j,k}(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{1,0}} + \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{1,0}} \\ \vdots \\ \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{N,n-1}} + \frac{\partial c^{\mathrm{w}}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{N,n-1}} \end{bmatrix}$$

Notation:

- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.
- N_k Total # of convective weather clusters.

Constraints : Weather Avoidance

Constraint enforcing aircraft i to avoid weather cluster k at time step j.

$$c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u}) := 1 - \frac{\left[(x_{i,j}-x^{w}_{k})\cos(\theta_k)+(y_{i,j}-y^{w}_{k})\sin(\theta_k)\right]^2}{a_k^2} - \frac{\left[(y_{i,j}-y^{w}_{k})\cos(\theta_k)-(x_{i,j}-x^{w}_{k})\sin(\theta_k)\right]^2}{b_k^2} \le 0$$

$$i\in[1,N] \quad j\in[0,n-1] \quad \mathrm{k}\in[1,\mathrm{N_k}]$$

⇓ Gradient Computation

$$\frac{\partial c^{w}_{i,j,k}(\mathbf{x}_0,\mathbf{u})}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial c^{w}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{1,0}} + \frac{\partial c^{w}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{1,0}} \\ \vdots \\ \frac{\partial c^{w}_{i,j,k}}{\partial x_{i,j}}\frac{\partial x_{i,j}}{\partial u_{N,n-1}} + \frac{\partial c^{w}_{i,j,k}}{\partial y_{i,j}}\frac{\partial y_{i,j}}{\partial u_{N,n-1}} \end{bmatrix}$$

$$\frac{\partial c^{w}_{i,j,k}}{\partial x_{i,j}} = -\frac{2\cos(\theta_k)}{a_k^2}\left[(x_{i,j}-x^{w}_{k})\cos(\theta_k)+(y_{i,j}-y^{w}_{k})\sin(\theta_k)\right] + \frac{2\sin(\theta_k)}{b_k^2}\left[(y_{i,j}-y^{w}_{k})\cos(\theta_k)-(x_{i,j}-x^{w}_{k})\sin(\theta_k)\right]$$

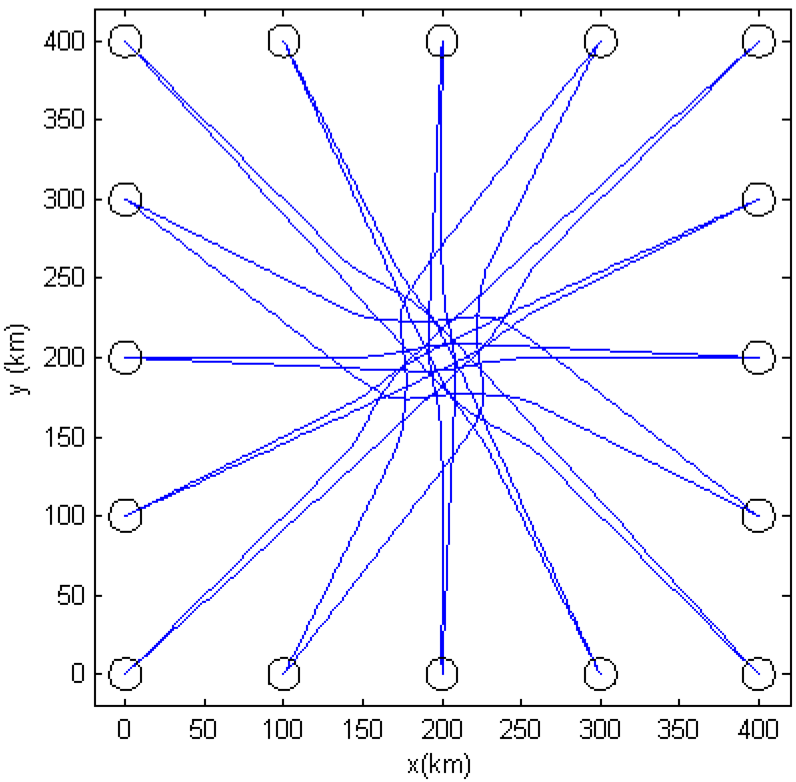
$$\frac{\partial c^{w}_{i,j,k}}{\partial y_{i,j}} = -\frac{2\sin(\theta_k)}{a_k^2}\left[(x_{i,j}-x^{w}_{k})\cos(\theta_k)+(y_{i,j}-y^{w}_{k})\sin(\theta_k)\right] - \frac{2\cos(\theta_k)}{b_k^2}\left[(y_{i,j}-y^{w}_{k})\cos(\theta_k)-(x_{i,j}-x^{w}_{k})\sin(\theta_k)\right]$$

Algorithm
- NMPC
- Dynamics Discretization
- Objective Function
- **Constraints**
  - Collision
  - **Weather**

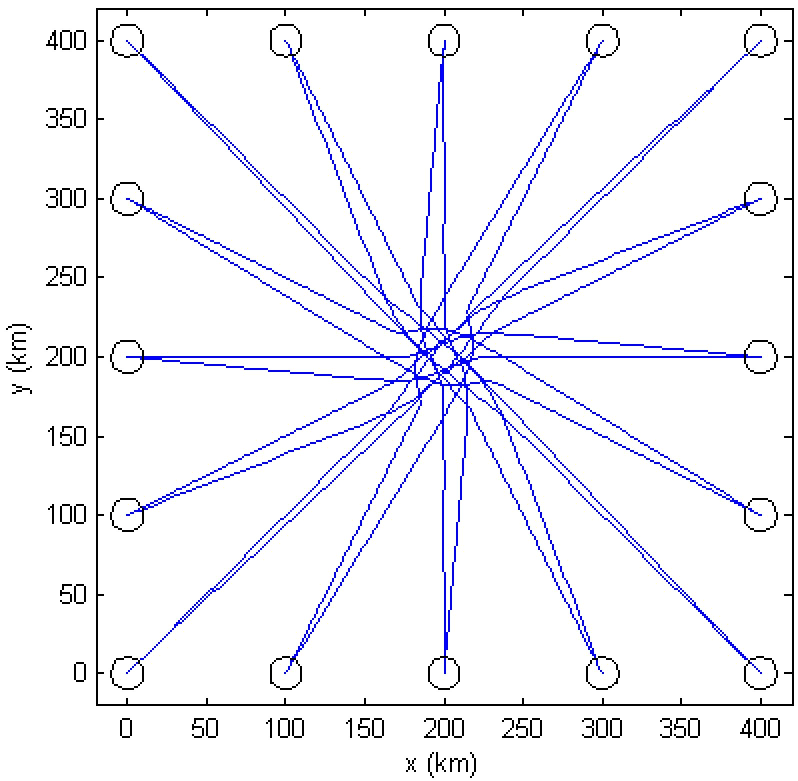
Notation:
- $\mathbf{x}_0 = [\mathbf{x}_{1,0},\ldots,\mathbf{x}_{N,0}]$
- $(x_{i,j}, y_{i,j})$ Position of aircraft i at time step j.
- $\mathbf{u} = [u_1,\ldots,u_N]$
- $u_i = [u_{i,0},\ldots,u_{i,n-1}]$
- N # of aircraft.
- $n = (T/\Delta t)+1$ # of discrete time steps.
- N_k Total # of convective weather clusters.

Results

- Quadratic Objective Function: Aircraft Collision Avoidance
  - 16 aircraft.
  - Time step: Δt = 16 seconds.
  - Control update interval Δtc = 32 seconds.



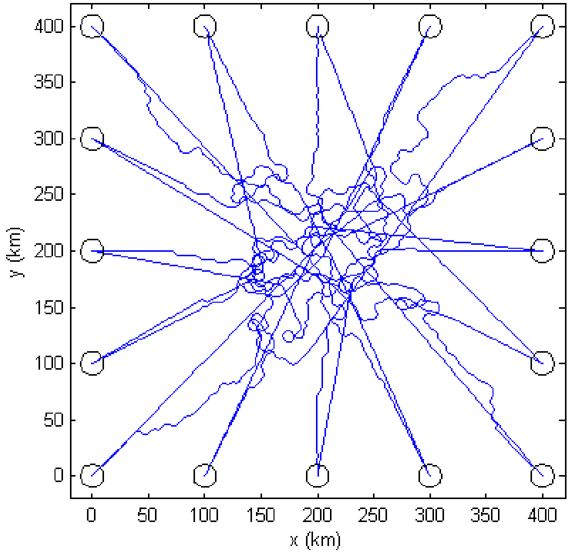
Horizon Length T = 256 seconds



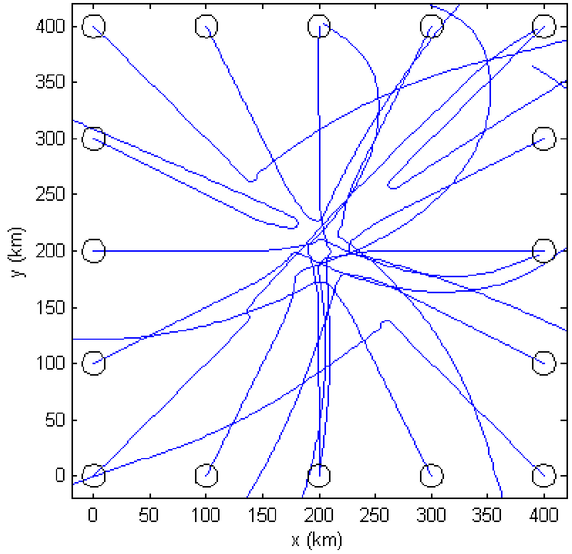
Horizon Length T = 128 seconds

Results

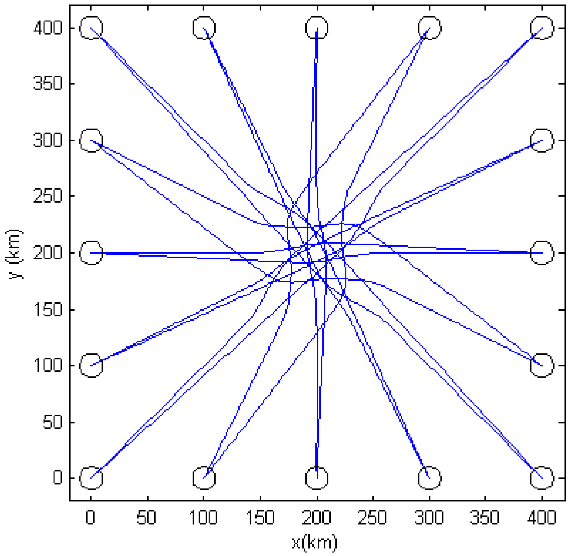
- Input cost matrix R:
  - Penalizes changes in aircraft heading.
  - Requires proper tuning.



No input cost



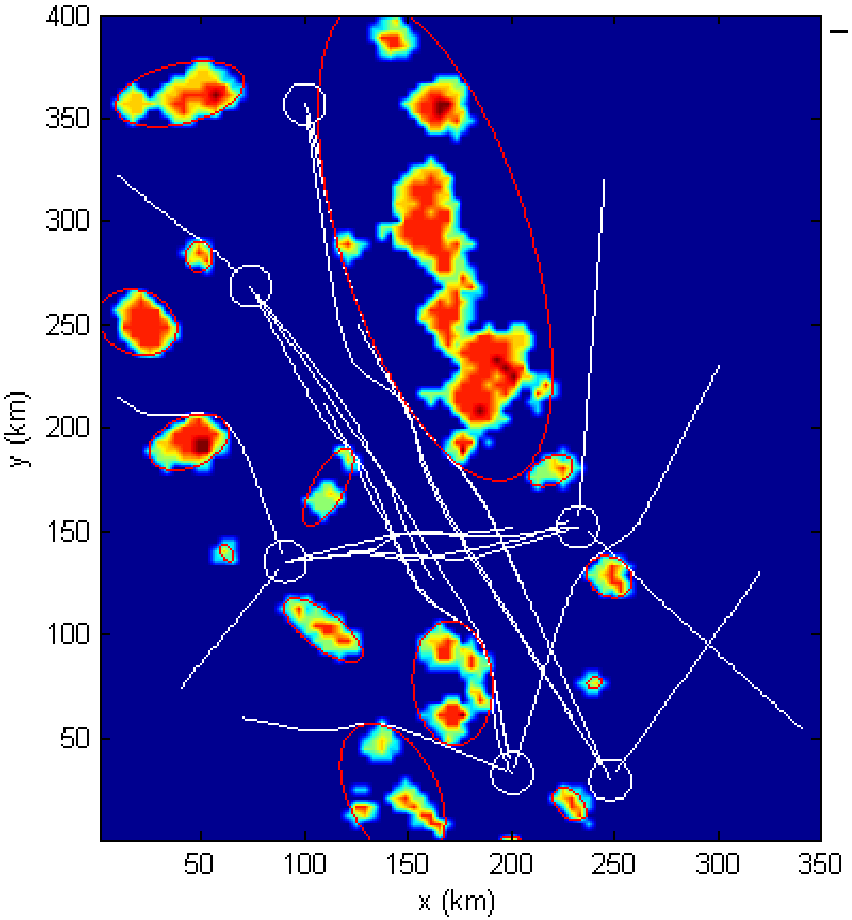
Input cost too large



Correct input cost tuning

Results

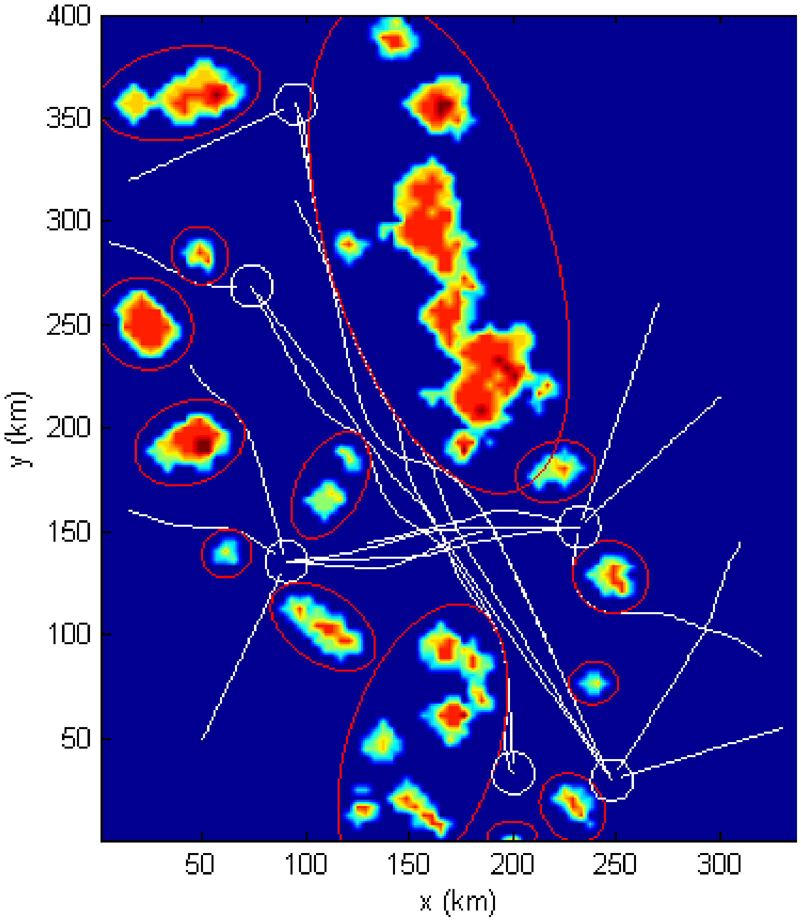
- Quadratic Objective Function: Aircraft collision avoidance and convective weather avoidance.
  - 20 aircraft.
  - Horizon Length: T= 128 seconds.
  - Time step: Δt = 8 seconds.
  - Control update interval Δtc = 32 seconds.



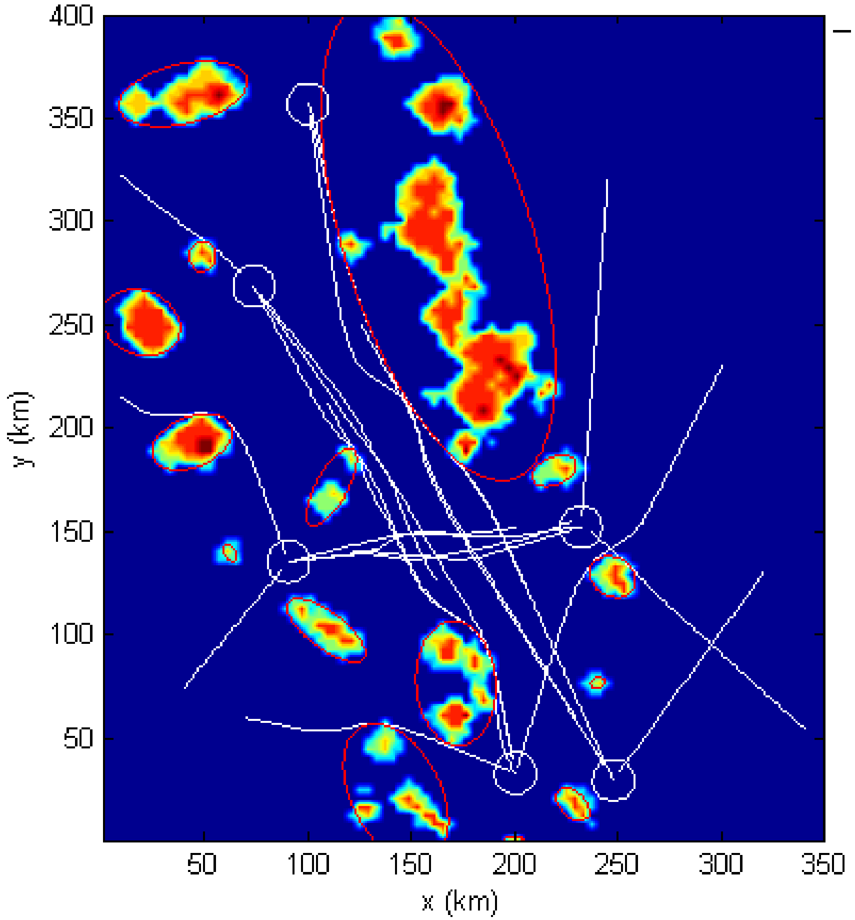
Tight Bound Enclosing Convective Weather

Results

Results
– Collision Avoidance
– **CA & Weather**
  - **Quadratic**
  - HJ PDE



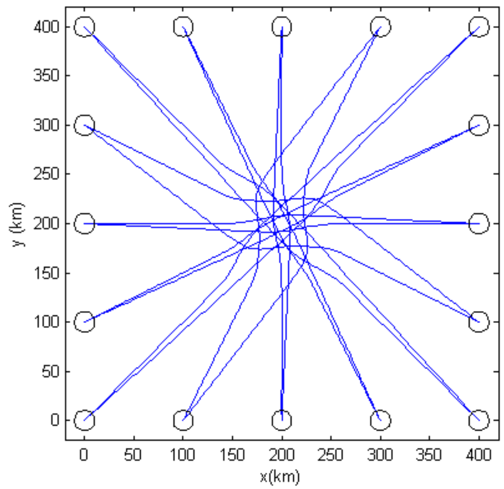
Conservative Bound Enclosing Convective Weather



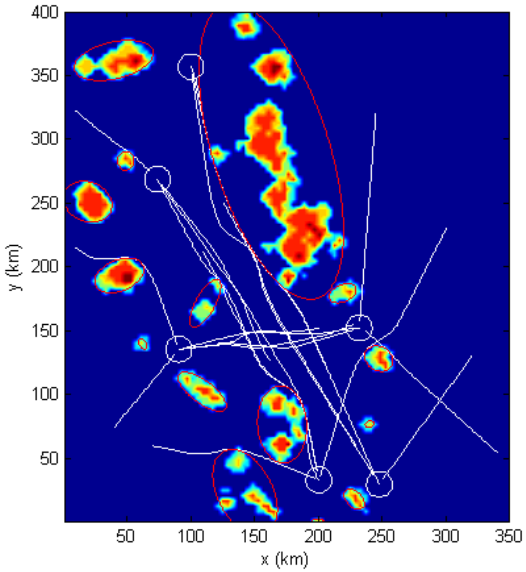
Tight Bound Enclosing Convective Weather

Results

- Timing Analysis:
  - In order to run in real time the convergence time must be smaller than Δtc



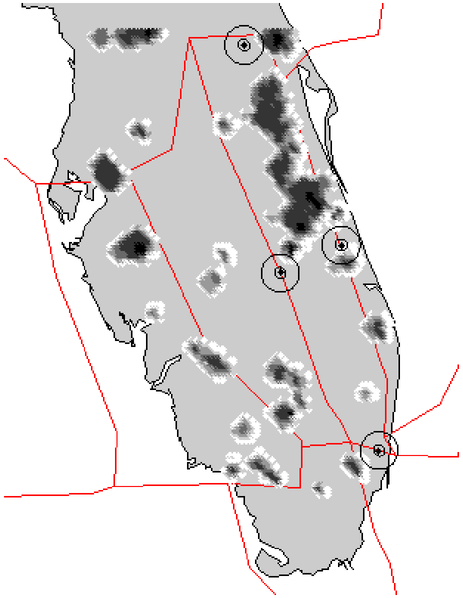
# Aircraft	Min time (s)	Max time (s)	Av. Time (s)	Δtc (s)
1	0.03	0.14	0.04	32.0
6	0.25	5.34	0.59	32.0
12	1.78	28.5	3.93	32.0
16	4.46	**70.31**	21.67	32.0



# Aircraft	Min time (s)	Max time (s)	Av. Time (s)	Δtc (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

Results

- Objective Function constructed with solution to HJE: Aircraft collision avoidance and convective weather avoidance.
  - 4 aircraft.
  - Horizon Length: T= 64 seconds.
  - Time step: Δt = 4 seconds.



Conclusion and Future Work

- Path planning achieved for quadratic cost function. This method scales well and the computing time is short.
- Before we can use HJ PDE objective function, we must find a more accurate way to compute the gradient of the Lagrangian or use first order algorithm for solving the optimal control problem.
- The next goal is to extend this framework to three dimensions and include time varying weather patterns and wind data.

Acknowledgments

- Dr. Banavar Sridhar.
- Dr. Kapil Sheth and Dr. Shon Grabbe.
- Dr. George Meyer.

Acknowledgments

- Dr. Banavar Sridhar.
- Dr. Kapil Sheth and Dr. Shon Grabbe.
- Dr. George Meyer.

- Prof. Alexandre Bayen and Prof. Shankar Sastry.
- Prof. Lucien Polak.
- Dr. Hoam Chung.
- Prof. Mark Hansen and Wanjira Jirajaruporn