Improved multi-aircraft ground trajectory prediction through wind forecast error filtering

Ioannis Lymperopoulos & John Lygeros

Automatic Control Laboratory, ETH Zürich

www.control.ethz.ch

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

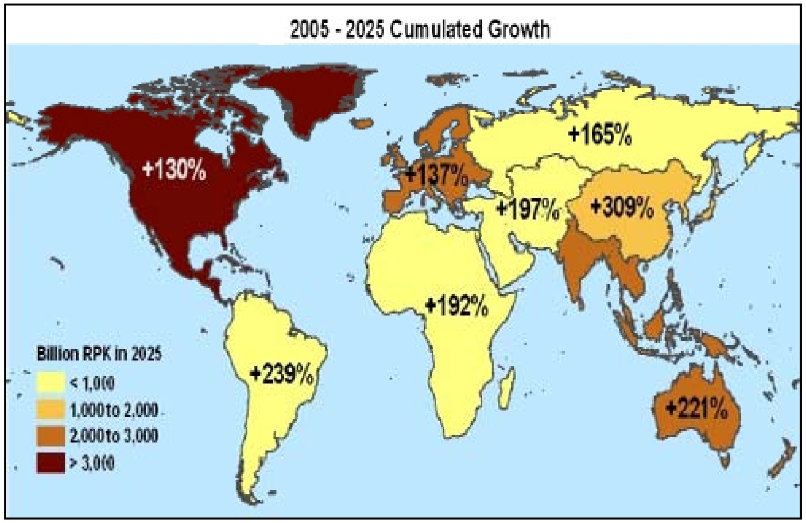
Credits

- In collaboration with
  - I. Lymperopoulos
  - G. Chaloulos, W. Glover, F. Ramponi, A. Stulova
  - A. Lecchini, J. Maciejowski
- Thanks to
  - H. Blom, P. Lezaud
- A word from the sponsors

Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

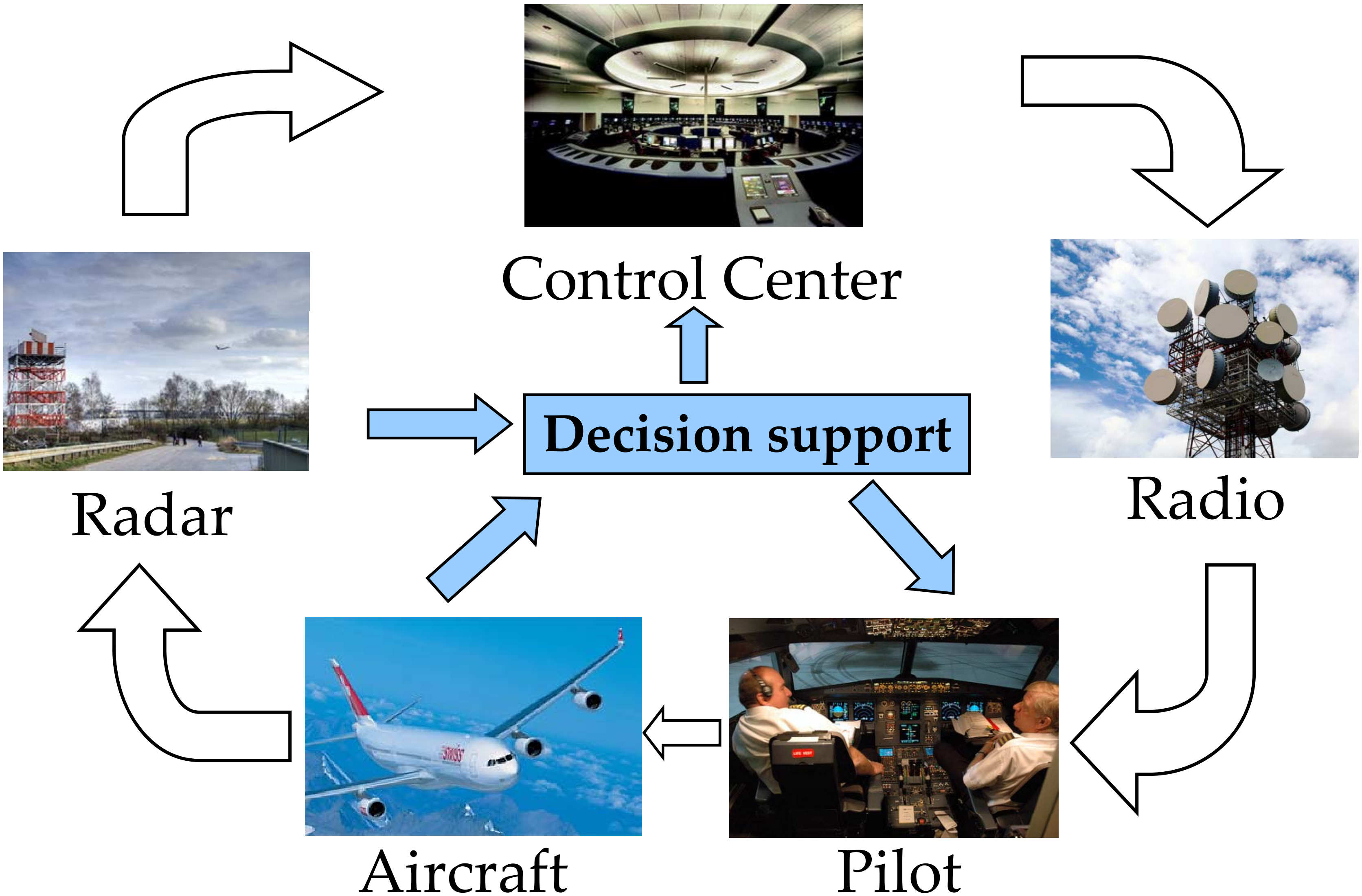
Air Traffic Congestion



- Air traffic demand increases rapidly
- The system reaching its capacity limits
- Air traffic controllers having difficulty handling traffic
- New technologies are coming into the field

- Current Approach
  - Add more air traffic controllers by dividing the airspace into more sectors
- Future Approach
  - Introduce more automation in the system
  - Develop Decision Support Tools for human operators

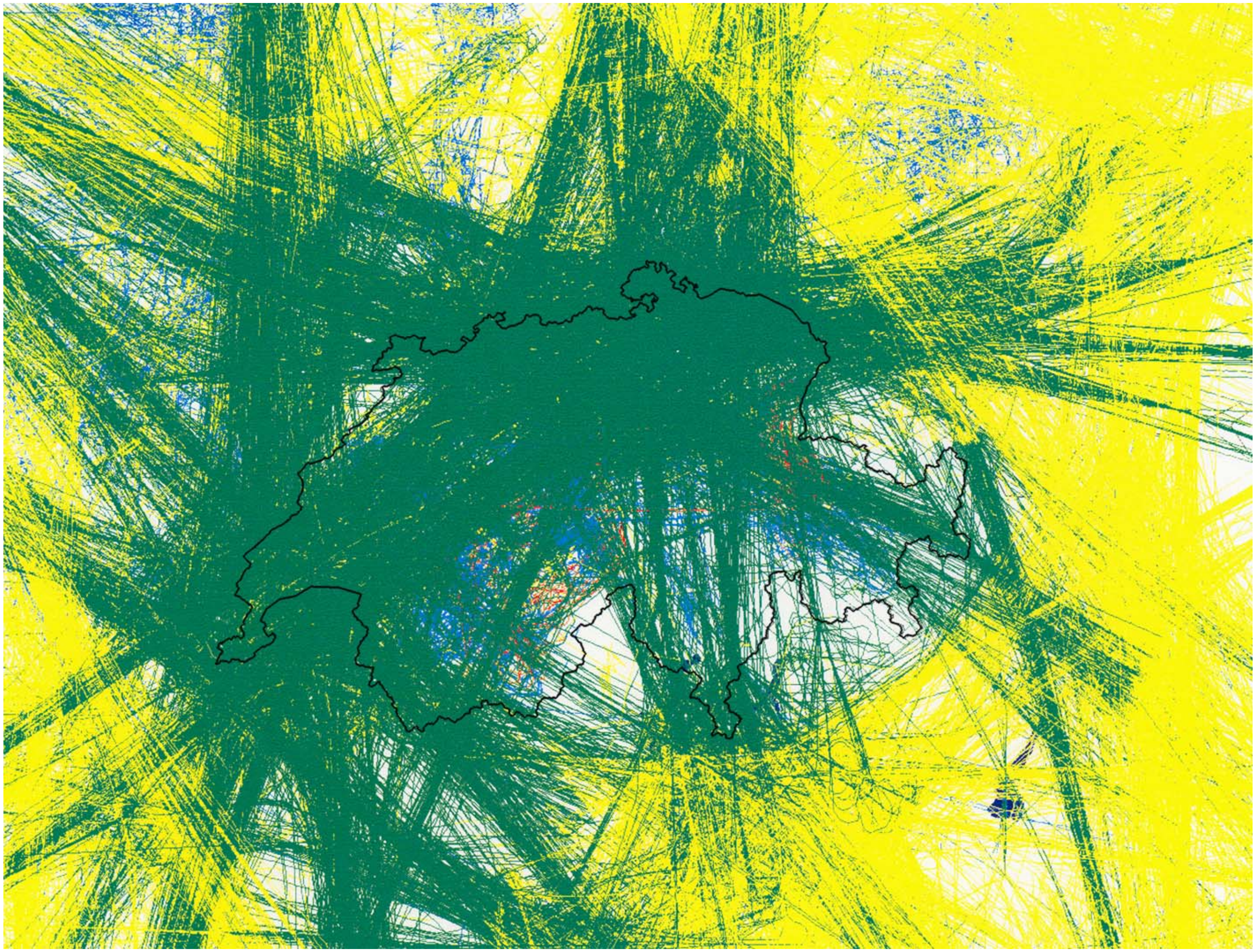
Air Traffic Management



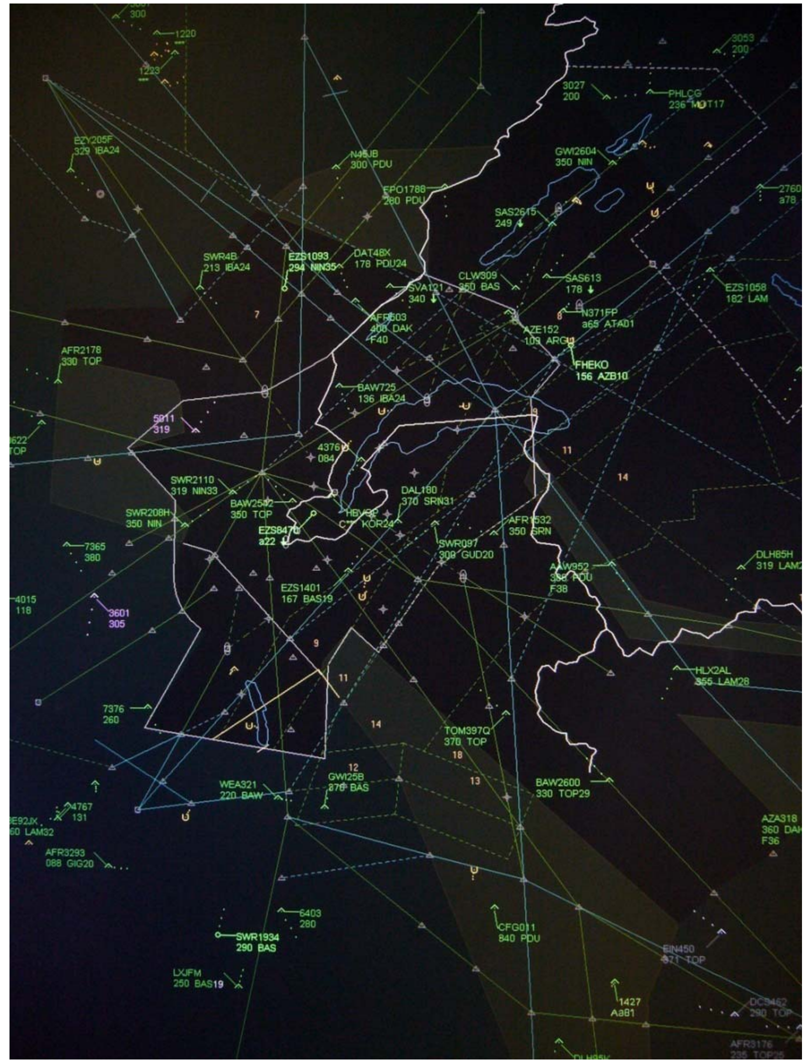
On-line secision support tools

- Predict the future position of aircraft to
  - Identify problems before they materialize
  - Propose solutions to air traffic controllers
  - Futuristic concepts: Automate (some of) the operations
- Key ingredient: Trajectory prediction
- Complicated by uncertainty
  - Actions of air traffic controller and pilot
  - Airspeed settings, aircraft mass
  - Malfunctions, faults, emergencies
  - Weather (primarily wind)
- Exploit information: Secondary radar or datalink

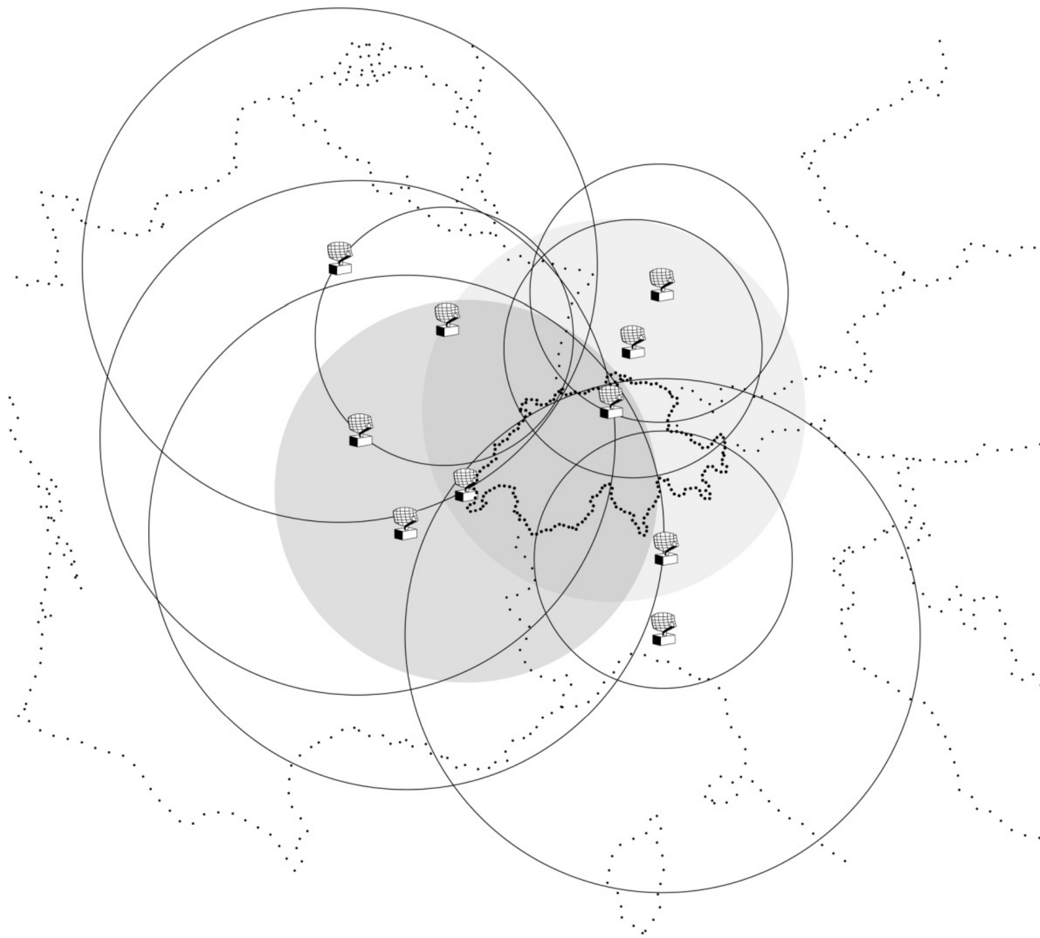
Switzerland – Flight Tracks – *8 Sep 2008*



Secondary Surveillance Radar

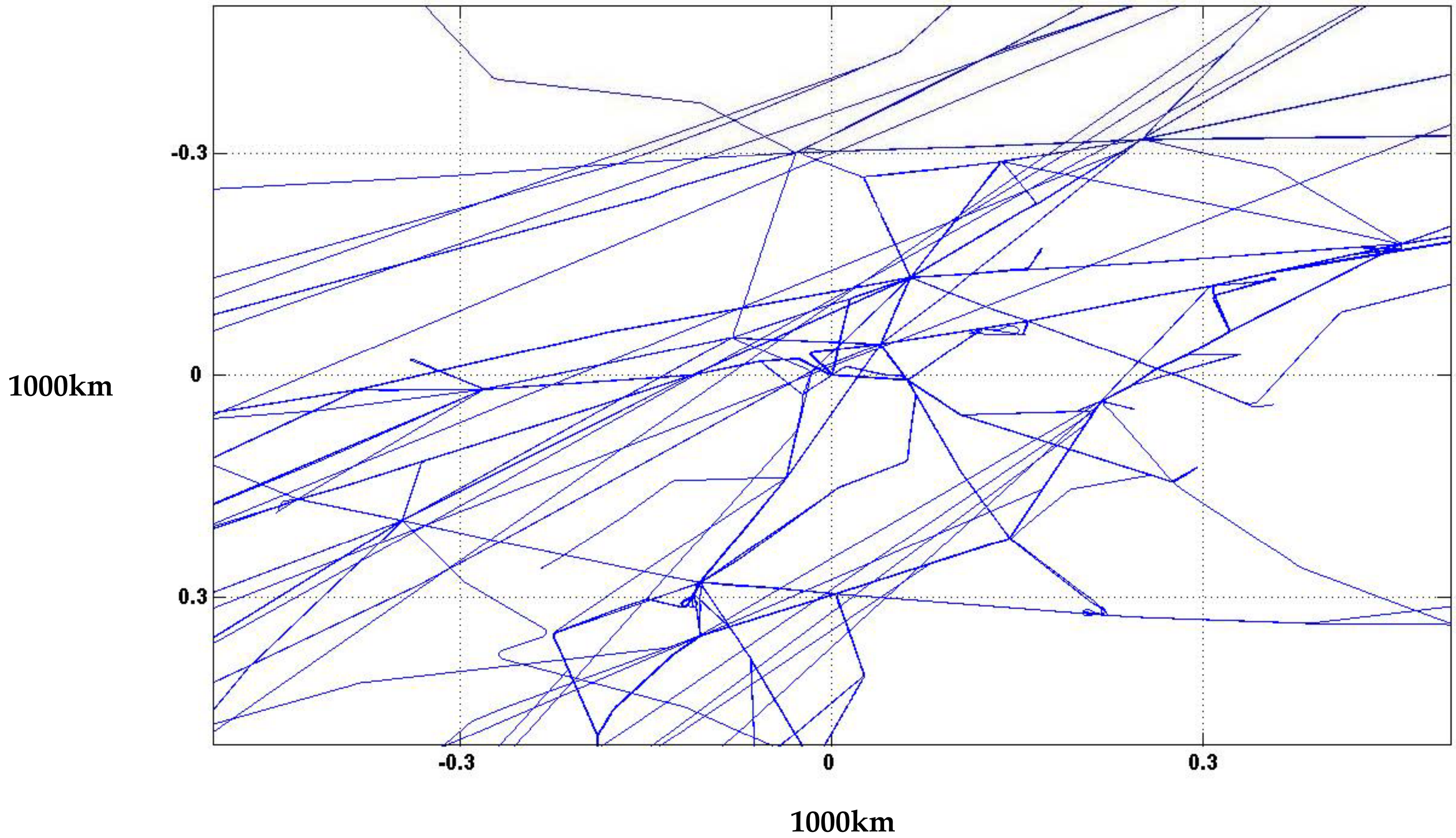


Switzerland Radar Coverage



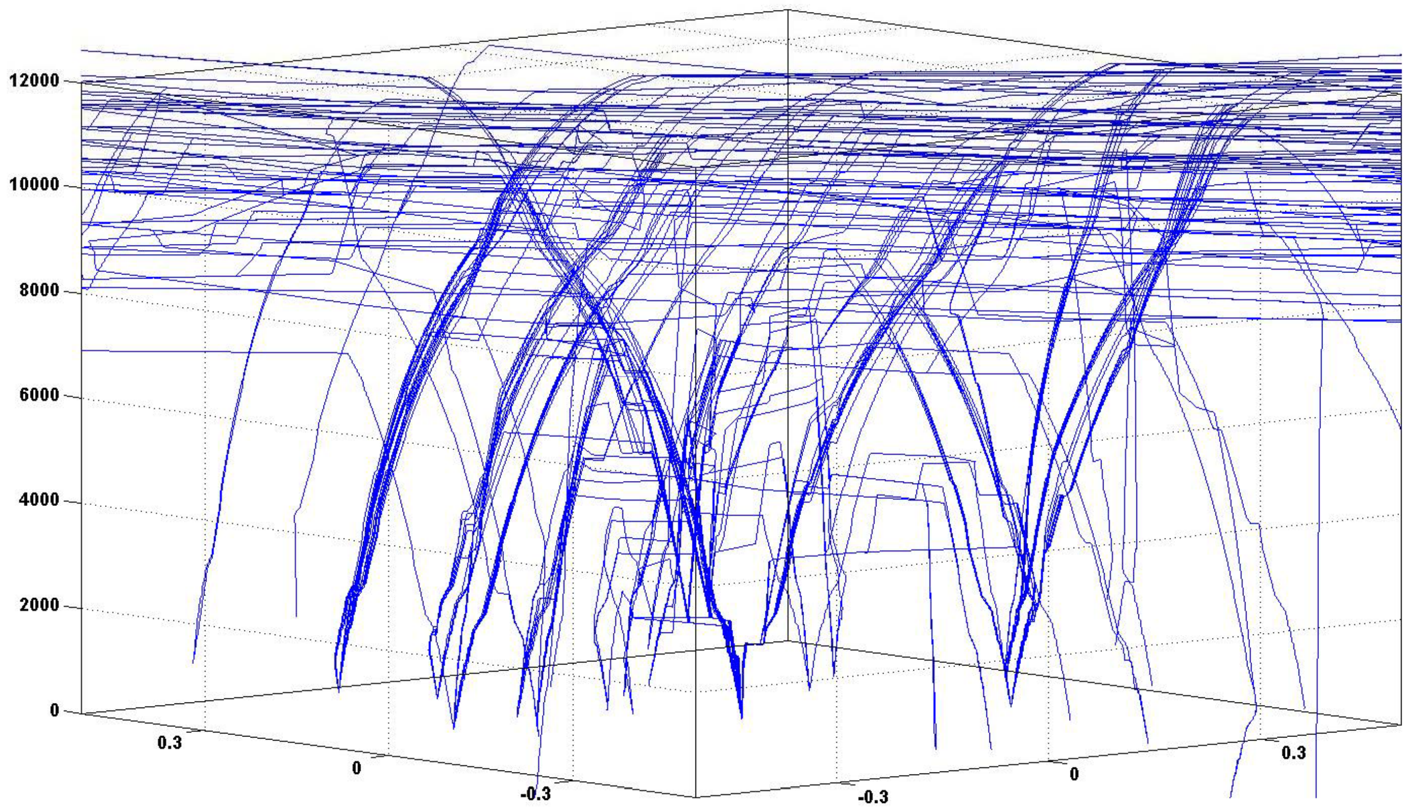
Example: Top view of Zurich airspace

Centered at Zurich, CFMU



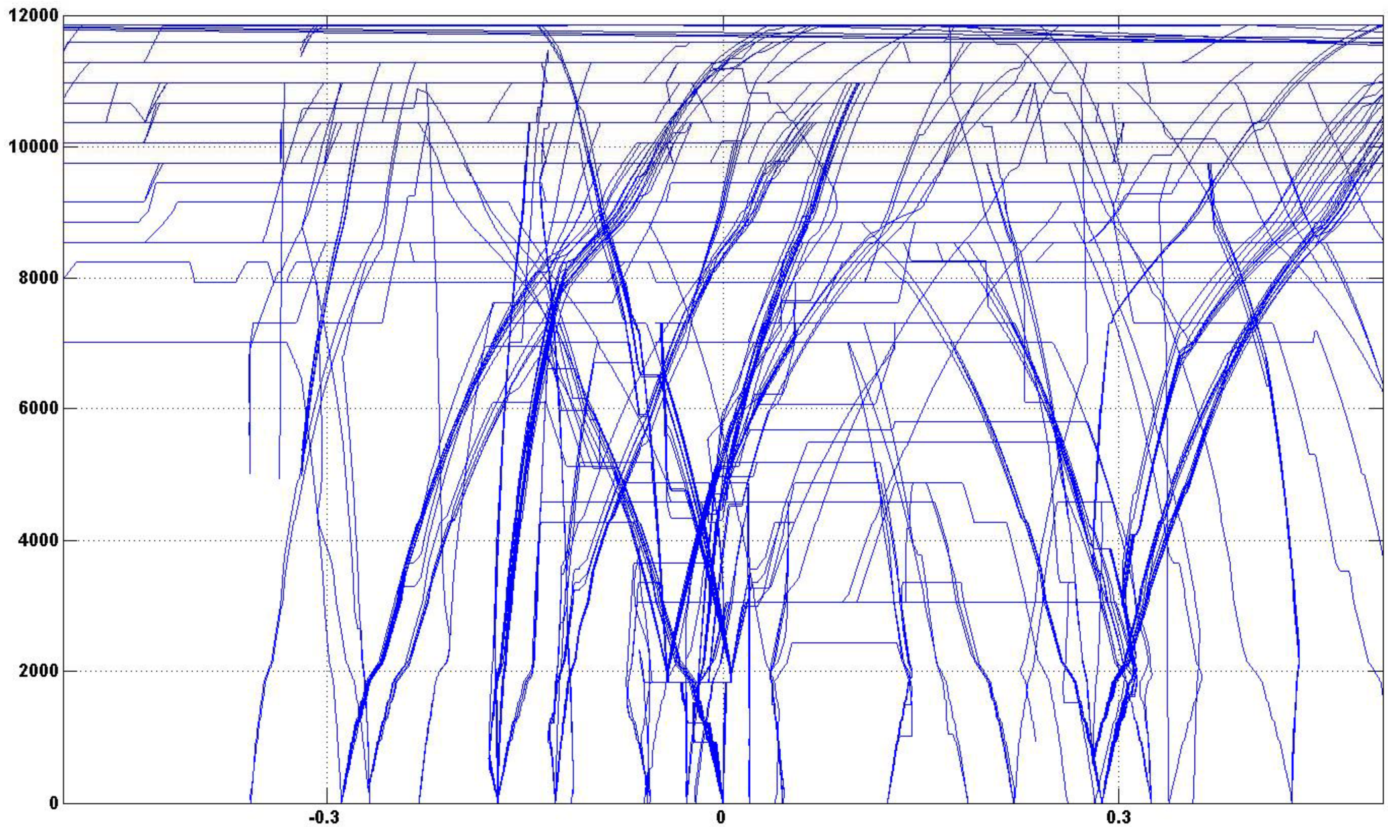
3D view of Zurich airspace

Aircraft **take-off**, **land** and **fly-through**

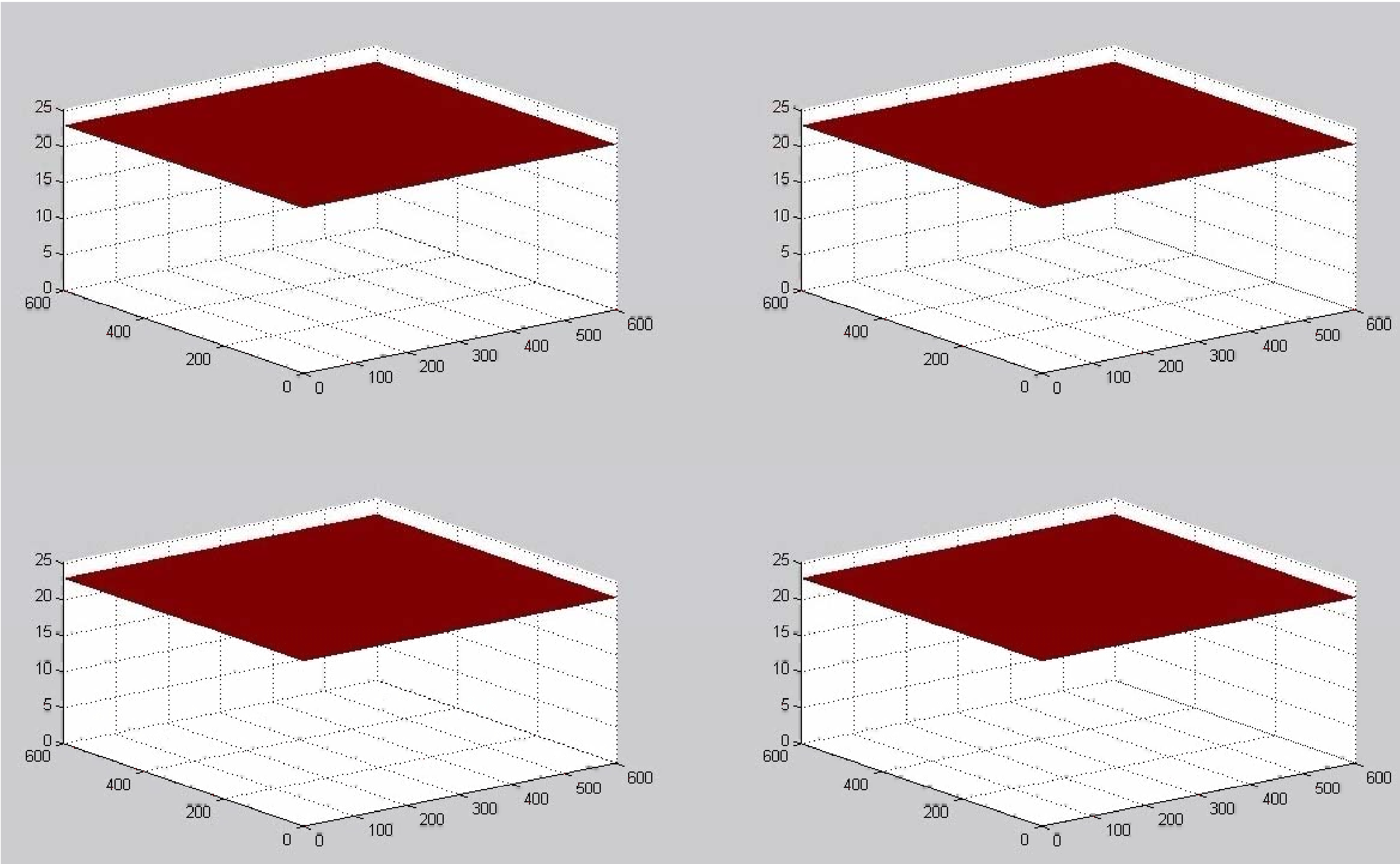


Side view of Zurich airspace

Traffic structured at specific altitude levels



Wind uncertainty– Upper 4 Flight Levels



Main idea

- Exploit information on actual wind conditions
  - Secondary radar
  - Datalink of on-board wind/airspeed measurements
- Local correction to wind forecast
  - Use aircraft as moving wind sensors
  - Spatio-temporal correlation of wind forecast error
- Difficulty:
  - Extract available information from radar tracks
  - Filtering problem
  - Non-linear, non-Gaussian, high dimensional
  - Conventional filtering methods do not work

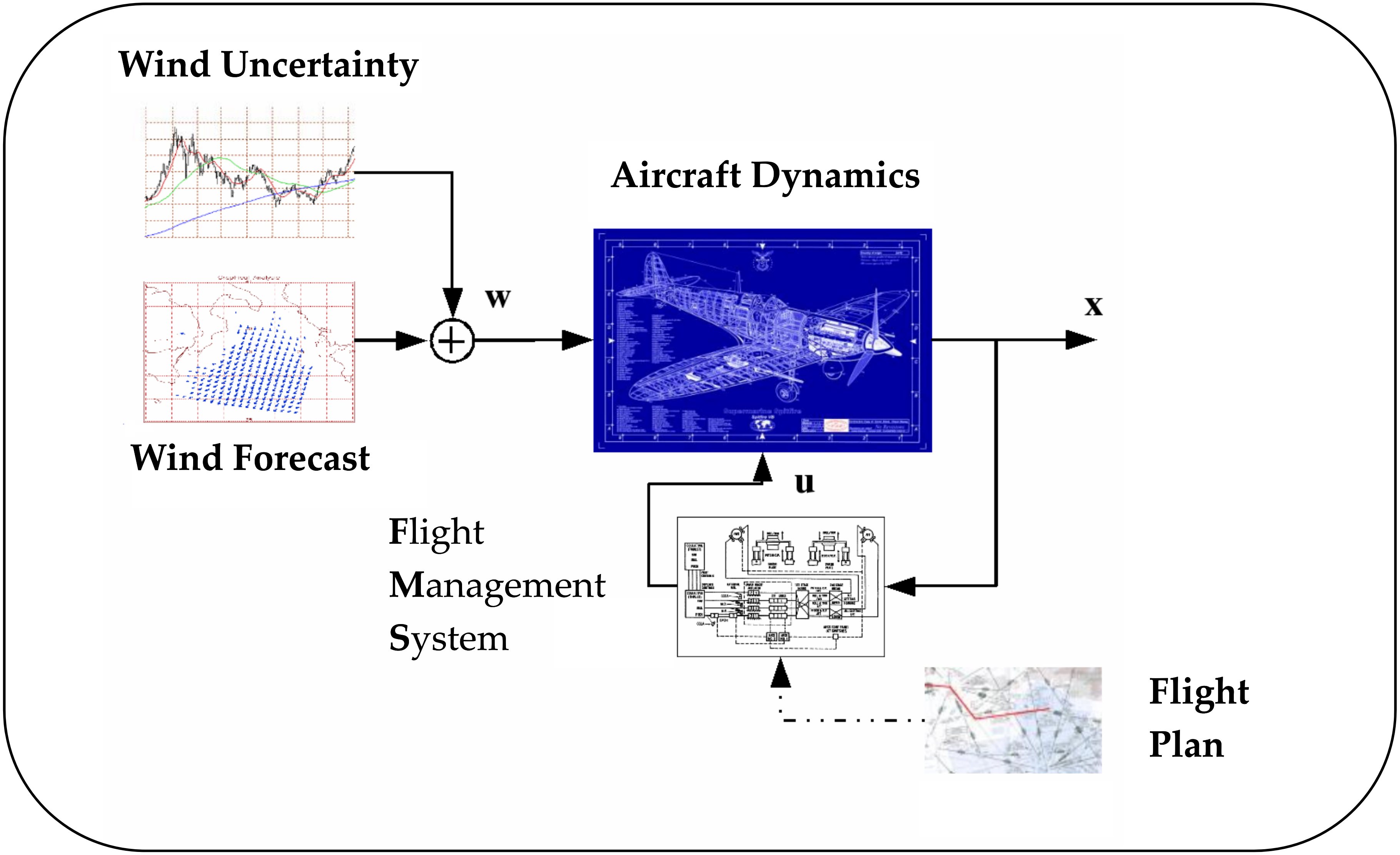
Proposed solution

- Dedicated particle filtering algorithm
- Exploit problem structure to improve computation
  - Forecast error states independent of aircraft states
  - Linear approximation of forecast error dynamics
  - Conditioning on forecast error states makes the nonlinear states of different aircraft independent
- Sequential conditioning particle filter
  - Sequentially condition on measurement of each aircraft
  - Algorithm numerically robust
  - Improvement in wind forecast error
  - Improvement in trajectory prediction error
  - Improvement in conflict probability estimates

Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

Block Diagram of the aircraft model



State of the Aircraft Model

Continuous state

- [Position]
- [True Airspeed]
- [Heading]
- [Mass]

Inputs

- Thrust
- Bank angle
- Flight path angle

Discrete state

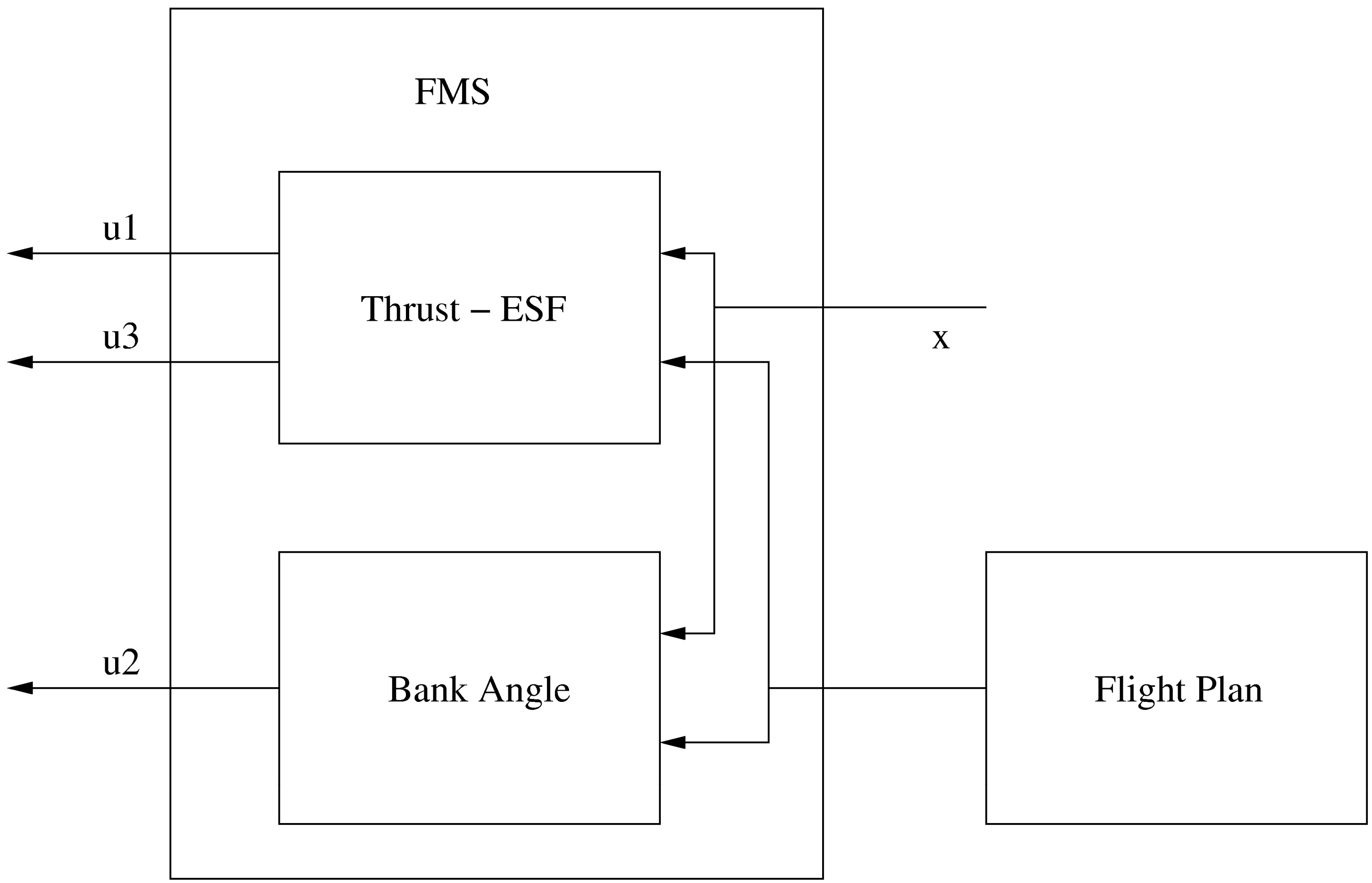
- **FL**: Flight Level
- **WP**: Way point index
- **AM**: Acceleration mode
- **CM**: Climb mode
- **TrM**: Troposphere mode
- **SHM**: Speed hold mode
- **FP**: Flight phase
- **RPM**: Reduced power mode
- **CRM**: Cruise mode

Aircraft dynamics – 3D Flight

$$\dot{x} = \begin{bmatrix} V\cos(\psi)\cos(\gamma) + w_1 \\ V\sin(\psi)\cos(\gamma) + w_2 \\ V\sin(\gamma) + w_3 \\ \frac{T-D}{m}(1-\widehat{u_3}) - W_{agf}\cos(\gamma)V\sin(\gamma) \\ \frac{L\sin(\phi)}{mV\cos(\gamma)} - W_{cgf}\tan(\gamma) \\ -\eta T \end{bmatrix}$$

Where $x = \begin{bmatrix} X & Y & h & V & \psi & m \end{bmatrix}^T$

Flight Management System (FMS)



Hybrid FMS controllers

- Input applied for thrust, bank angle and flight path angle depends on
  - Aircraft state (feedback)
  - Flight plan (reference trajectory)
  - Internal FMS discrete state (logic)
  - Measured wind (disturbance)
- Nonlinear continuous controllers
- Logic based switching

→FMS itself a hybrid system

Stochastic terms

- Wind consists of two parts
  - Nominal (meteorological forecast, assumed available)
  - Stochastic (uncertainty of the forecast)
- Stochastic part correlated in time & space

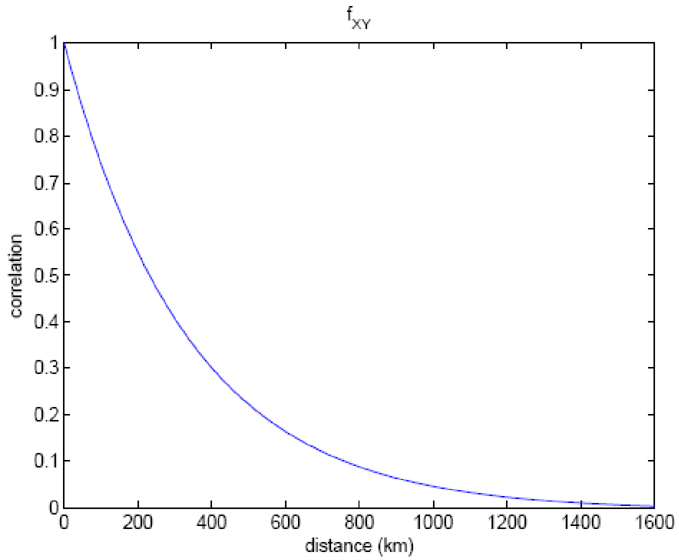
$$r_{xy}\left(t, \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, t', \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}\right) = \sigma(x_3)\sigma(x'_3) f_t(|t-t'|) f_{x_1 x_2}\left(\left\| \begin{matrix} x_1 - x'_1 \\ x_2 - x'_2 \end{matrix} \right\|\right) f_{x_3}\left(|p(x_3) - p(x'_3)|\right).$$

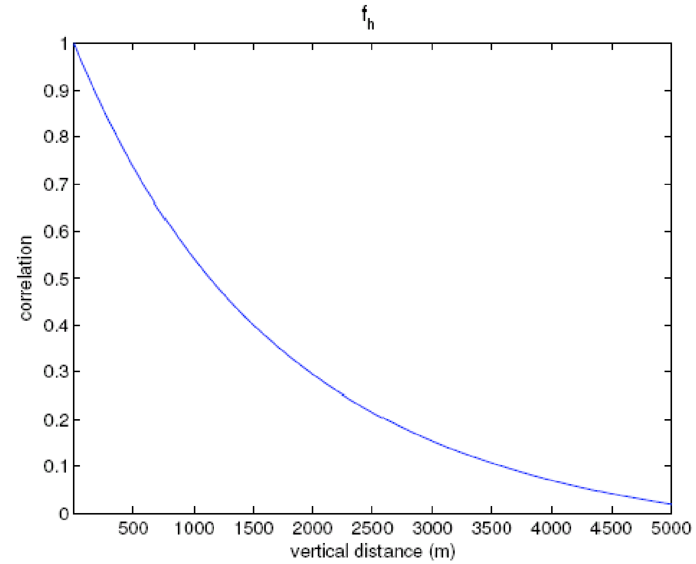
- Time correlation

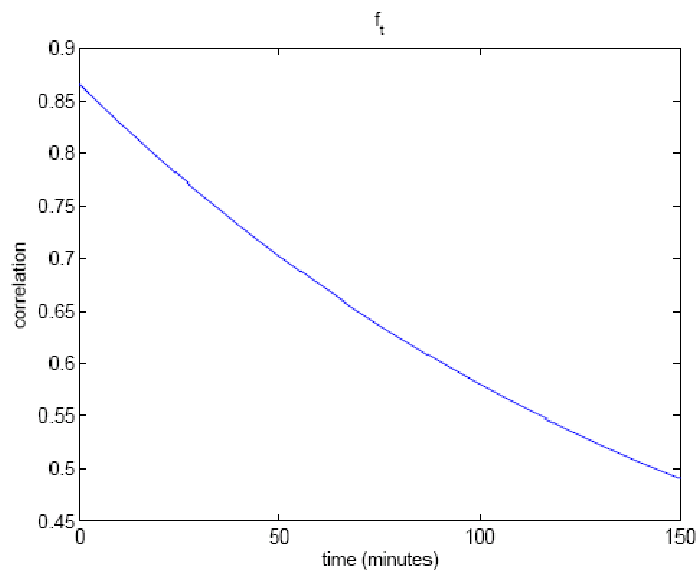
$$f_t(\chi) = c_t + (1 - c_t - d_t) e^{-\frac{\chi}{b_t}} + d_t \cos\left(2\pi \frac{\chi - e_t}{g_t}\right)$$

- Horizontal correlation $f_{x_1 x_2}(\chi) = c_{xy} + (1 - c_{xy}) e^{-\frac{\chi}{b_{xy}}}$
- Altitude correlation $f_{x_3}(\chi) = c_z + (1 - c_z) e^{-\frac{\chi}{b_z}}$

Wind uncertainty correlation

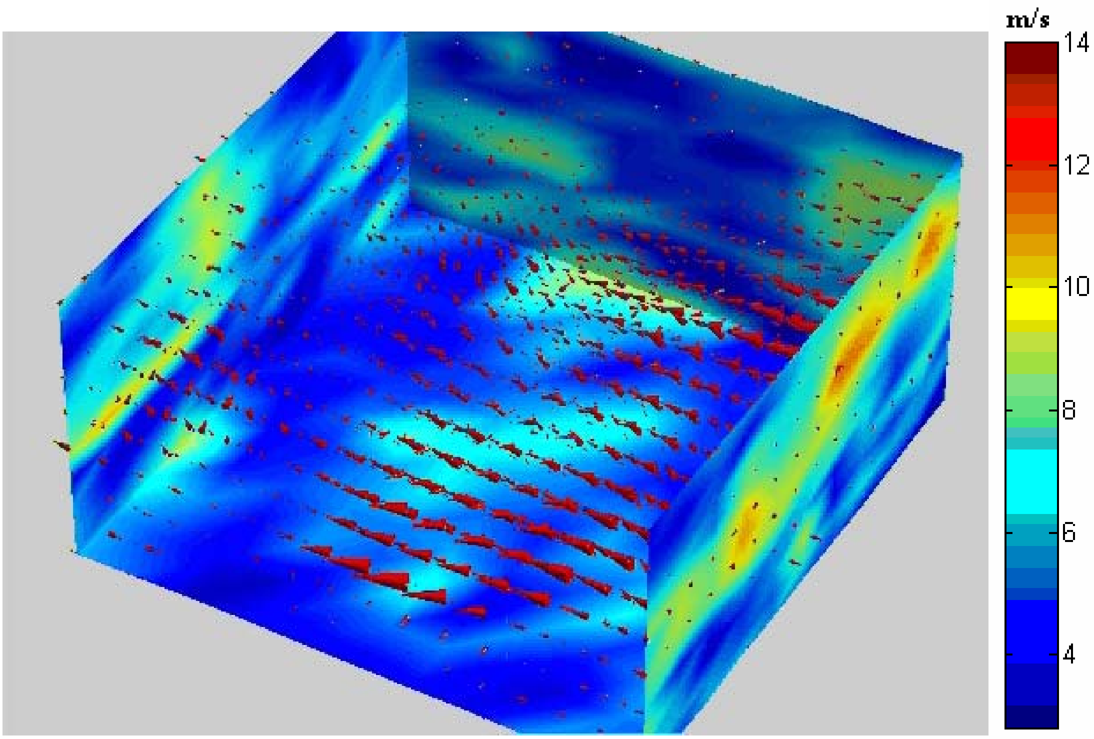






Wind Dynamics

- Wind Forecast
- Wind Forecast errors: Lincoln labs, RUC forecast error

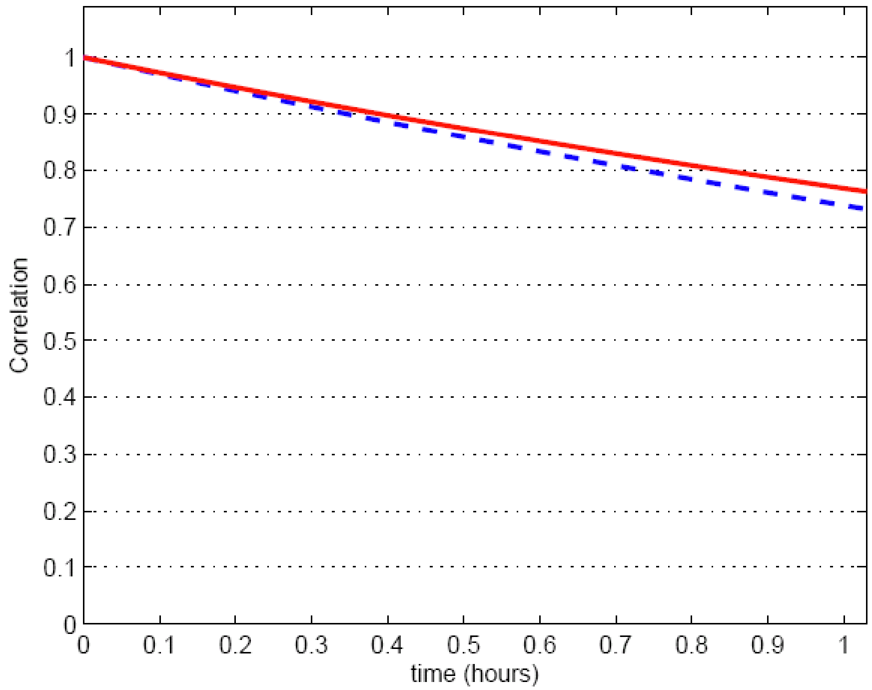


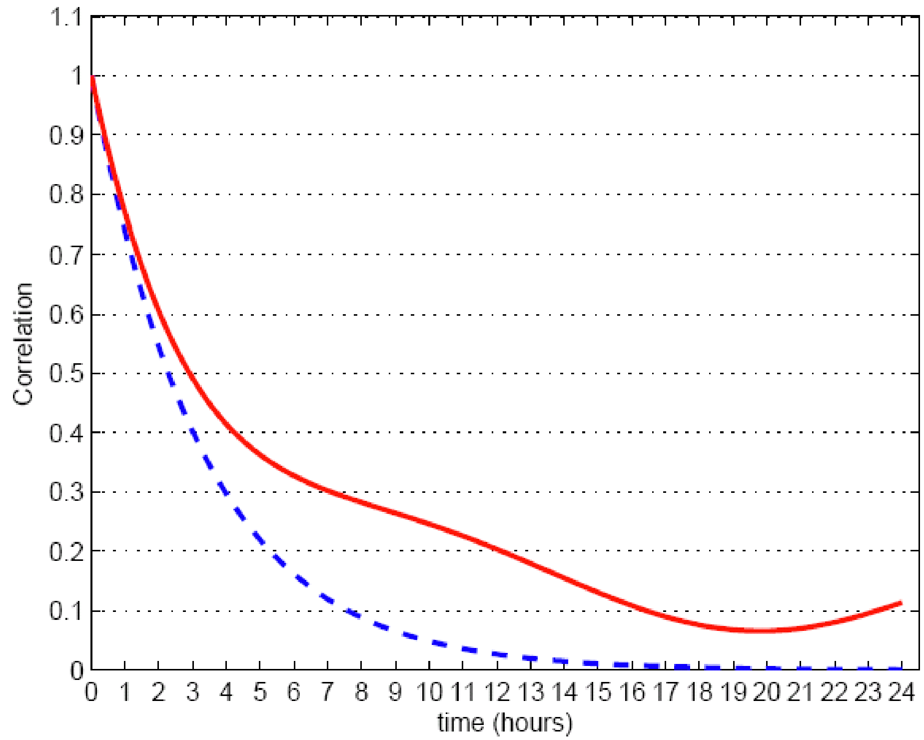
- Wind evaluated at the vertices of a lattice
- Linear interpolation for wind at different locations

$$w_t = Aw_{t-1} + q_t$$

Linear Model with additive Gaussian Noise

Approximation by linear model





Full System State

- Nonlinear states from aircraft
- Linear state of wind forecast error dynamics
- Radar measurements
  - Aircraft position
  - Additive Gaussian noise
  - 80m standard deviation (conservative)

Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

State Space Representation

$$x_t = f_t(x_{t-1}) + v_{t-1} \qquad x_t \in \mathbb{R}^{n_x}$$

$$y_t = h_t(x_t) + e_t \qquad y_t \in \mathbb{R}^{n_y}$$

Where,

$u_t \in \mathbb{R}^{n_x}$: Process Noise

$e_t \in \mathbb{R}^{n_y}$: Measurement Noise

Or alternatively

$$x_t \sim p(x_t | x_{t-1})$$

$$y_t \sim p(y_t | x_t)$$

Estimation Problem

- Given a sequence of real measurements up to time **τ** and the distribution of the state at t=0

$$\mathbb{Y}_\tau = \{y_i\}_{i=1,\dots,\tau}$$

$$x_0 \sim p(x_0)$$

- **Estimate the Probability Density Function**

$$p(\mathbb{X}_t | \mathbb{Y}_\tau)$$

Where, $\mathbb{X}_t = \{x_i\}_{i=0,\dots,t}$: state trajectory up to time t

Three Different Estimation Problems (t, τ)

(t < τ)
Smoothing: refine the estimation about an older part of the state trajectory.

(t = τ)
Filtering: estimate the state trajectory up to a current state.

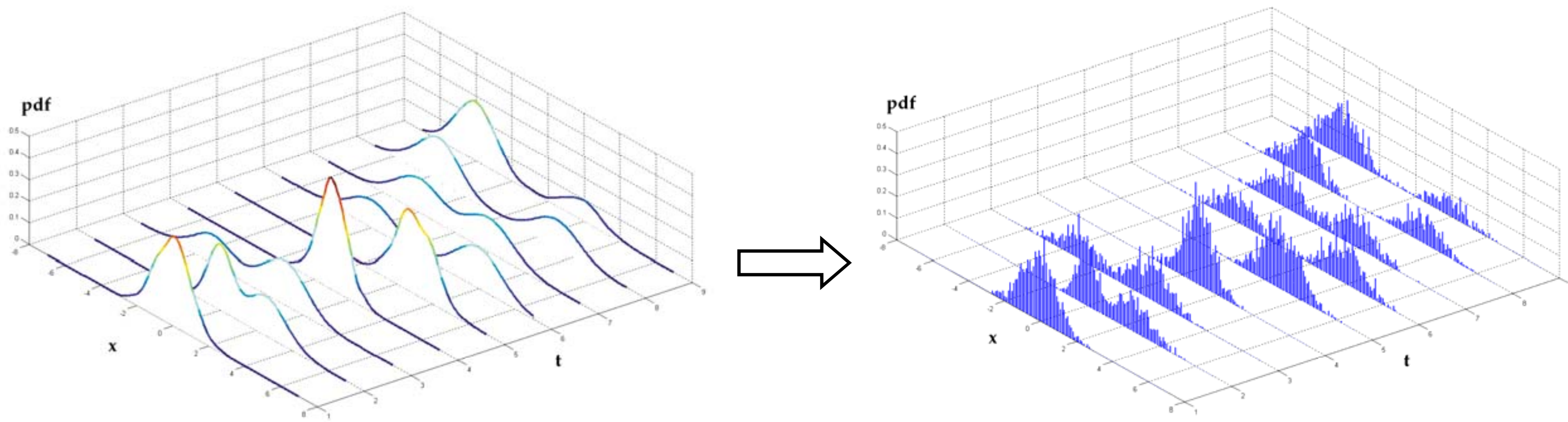
(t > τ)
Prediction: estimate the state trajectory up to a future state

Solutions to the Problem

- **Linear Dynamics, additive Gaussian Noise**
  - Closed Form Solution (Kalman Filter)
  - Optimal Estimator

- **Non-linear Dynamics, non Gaussian Noise**
  - Simulation Based Algorithms (Particle Filter)
  - Convergence for "infinite" computational power
  - Suboptimal for most real cases (but often works)
  - Curse of Dimensionality (does not always work…)

Sequential Monte Carlo Methods



- Concept:
  - Extract underlying information from a sequence of noisy measurements
  - Approximate the evolving pdf of the system state using a discrete – weighted distribution of Monte Carlo samples
- Suitable for:
  - Nonlinear models
  - Non-Gaussian noise
- Difficulties:
  - Degeneracy
  - Sample impoverishment

Particle Filters

A collection of **N** Particles in the State Space

$$\{x_t^i\}_{i=1,\ldots,N}$$

That provide a discrete approximation to the estimation problem

$$p(\mathbb{X}_t|\mathbb{Y}_\tau)$$

Each particle represents a different realization of the process noise

Particle Filter

Step0: Draw **i = 1,…,N** particles $x_0^i \sim p(x_0)$

Step1: (t< t+1) Evolve each particle (**i = 1,…,N**) to the next time step $x_t^i = f_t(x_{t-1}^i) + v_{t-1}^i$

Step2: Use the **measurement** at time t to assign a weight to each of the particles, **i = 1,…,N**, $w_t^i = p(y_t | x_t^i)$

Step3: Resample with replacement – according to the weights, to create a new a discrete approximation $\{\widehat{x}_t^i\}$

Potential problems

- Degeneracy
  - One particle gets all the weight
  - Re-sampling
- Sample impoverishment
  - Particles concentrate in one region
  - Add randomness to keep searching
  - Kernel density estimators
- Curse of dimensionality
  - Number of particles insufficient for many states
  - Exploit structure
  - Marginalized particle filter

Summary

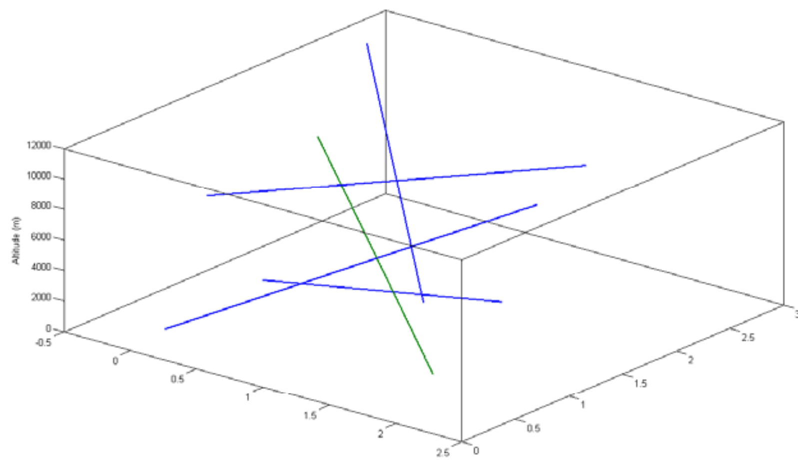
- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

Fusion of Aircraft Tracks

Goal: reduce wind uncertainty by exploiting spatiotemporal wind-field correlation

Facts:

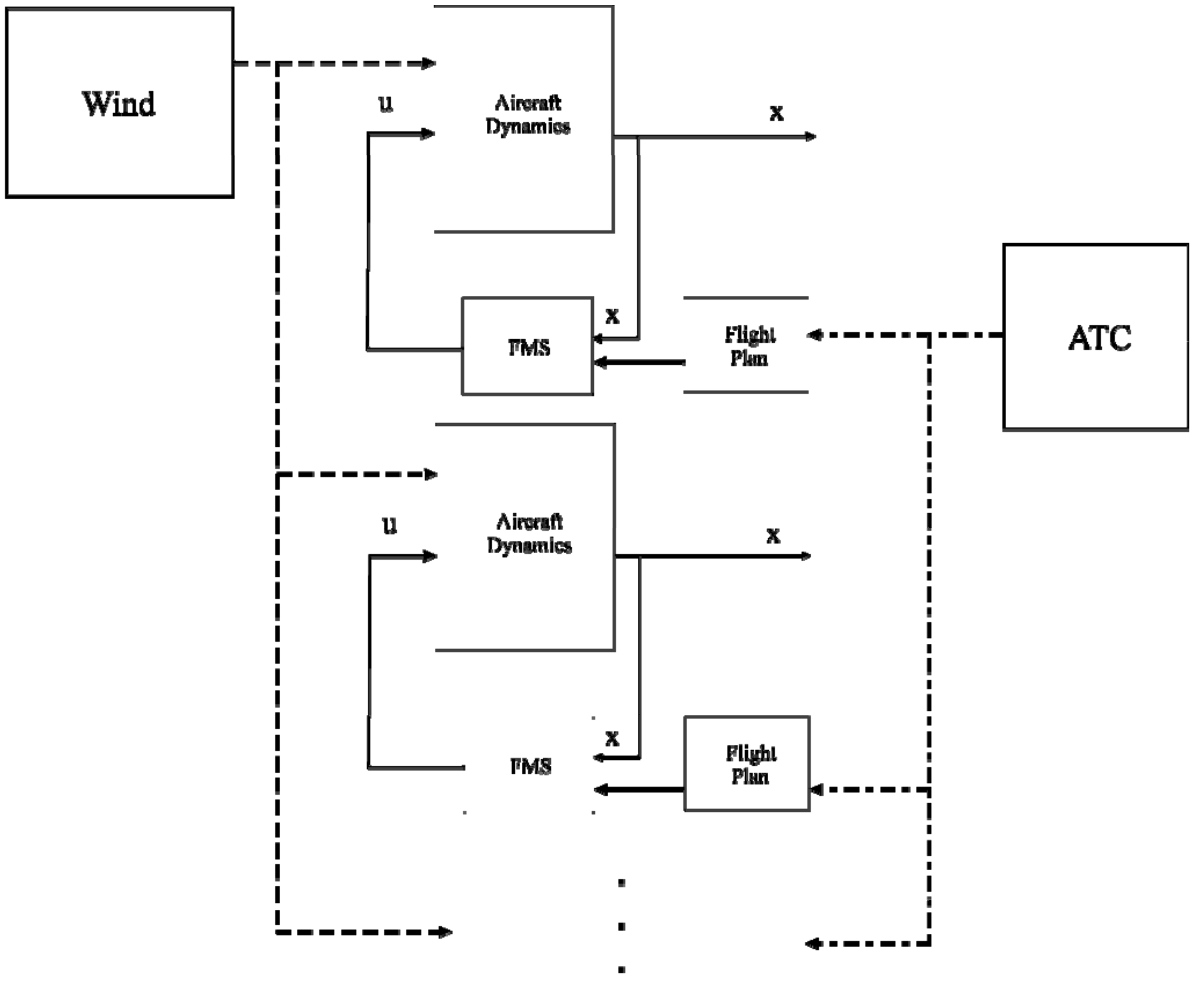
- Aircraft preceding the flight of interest
- Multiple aircraft at different locations in the airspace



Idea: employ aircraft as moving wind sensors

- **Directly** (transmitting through a data-link connection the actual wind-measurements of each aircraft)
- **Indirectly** (filtering the wind using radar measurements of the aircraft position)

Multiple Aircraft Model



Particle Filter Approach

- Curse of dimensionality
  - Large number of non-linear states
- For very few aircraft particle filter diverges
- Variations of the Classical Particle Filter
  - Kernel density estimator
  - Marginalized particle filter

  improve predictions, but break down after max. 10 aircraft
- Extreme computational effort
  - Tens of thousands of particles

Exploit the structure of the system

- Large part of the system state is **linear** (wind forecast error)
- Wind forecast error **not affected by aircraft**
- States of different aircraft **independent** of each other given the wind-field
- Aircraft are **indirectly coupled** through the wind-field
- Particle Filter works well for **individual aircraft** (few non-linear states)

Sequential Conditional Particle Filter

- Each aircraft is assigned a **Particle Filter**
- The results of each filter are fed to the next in a filter **queue**.
- Each particle holds
  - Realization of the **aircraft state**
  - **Mean** and **covariance matrix** of multivariate Gaussian distribution describing the entire wind-field

$$p^i_{t,k} = \{x^i_{t,k}, \mu^i_t, \Sigma^i_t\}$$

Where ***k*=1,…,*K*** is the index of the aircraft

Sequential Conditional Particle Filter

Step0: Draw **i=1,…,N** particles for **k=1,…,K** aircraft from $p(x_{0,k})$ assign to each particle a mean $\mu_0^i = 0$ and covariance $\Sigma_0^i = R$

Step1.1: (k<k+1) Evolve each particle of aircraft **k**, to the next time step, using wind extracted from its own wind distribution

Step1.2: Use the Radar **measurement** of this aircraft at time t to assign a weight to each of the particles of the aircraft PF

Step1.3: Resample with replacement – according to the weights, to create a new a discrete approximation

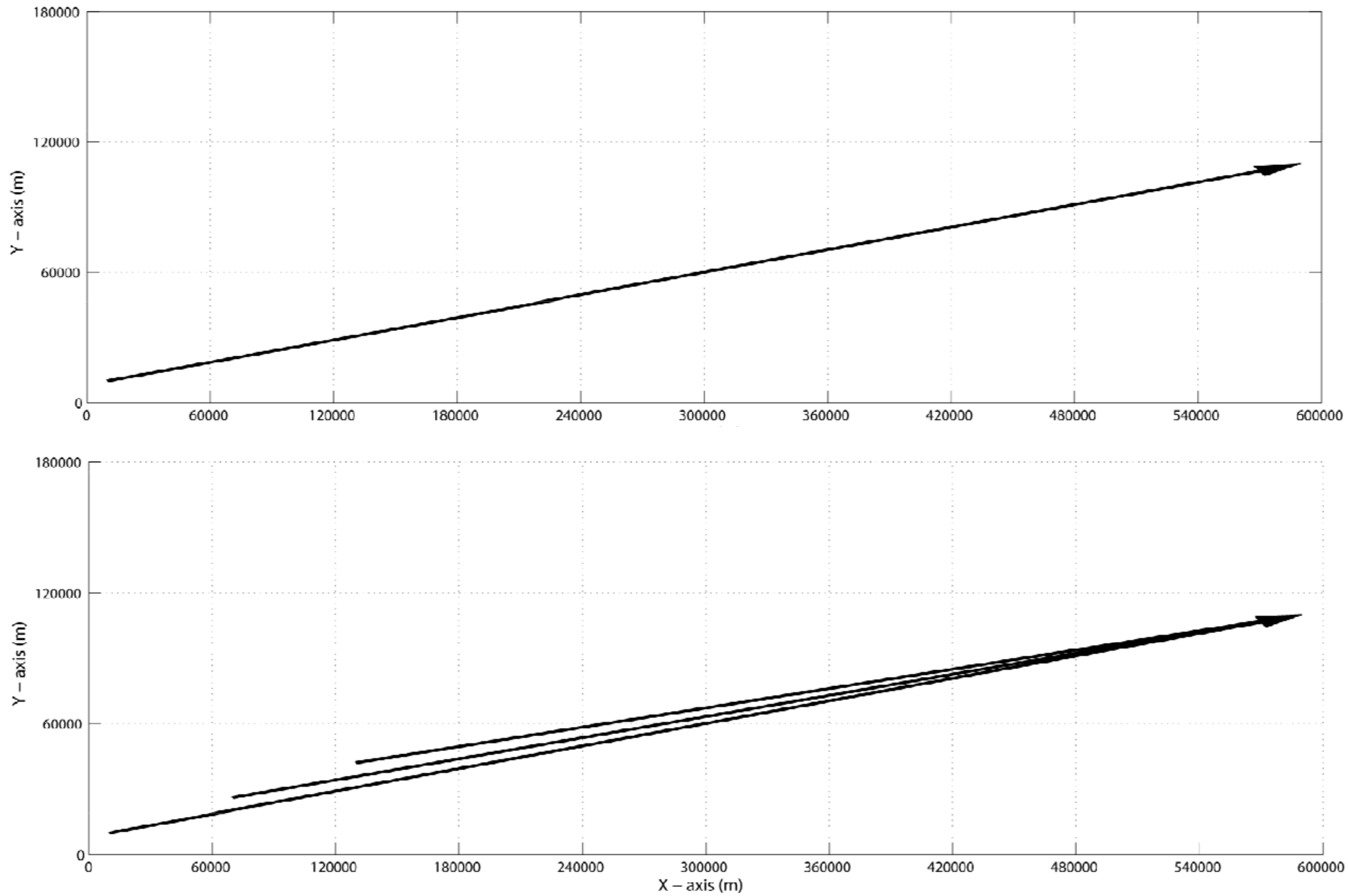
Step3: Condition on the position and wind assumed by each particle and assign new mean and covariance matrices to **next aircraft PF**

Step3: (t<t+1) Wind Evolution to the next time step

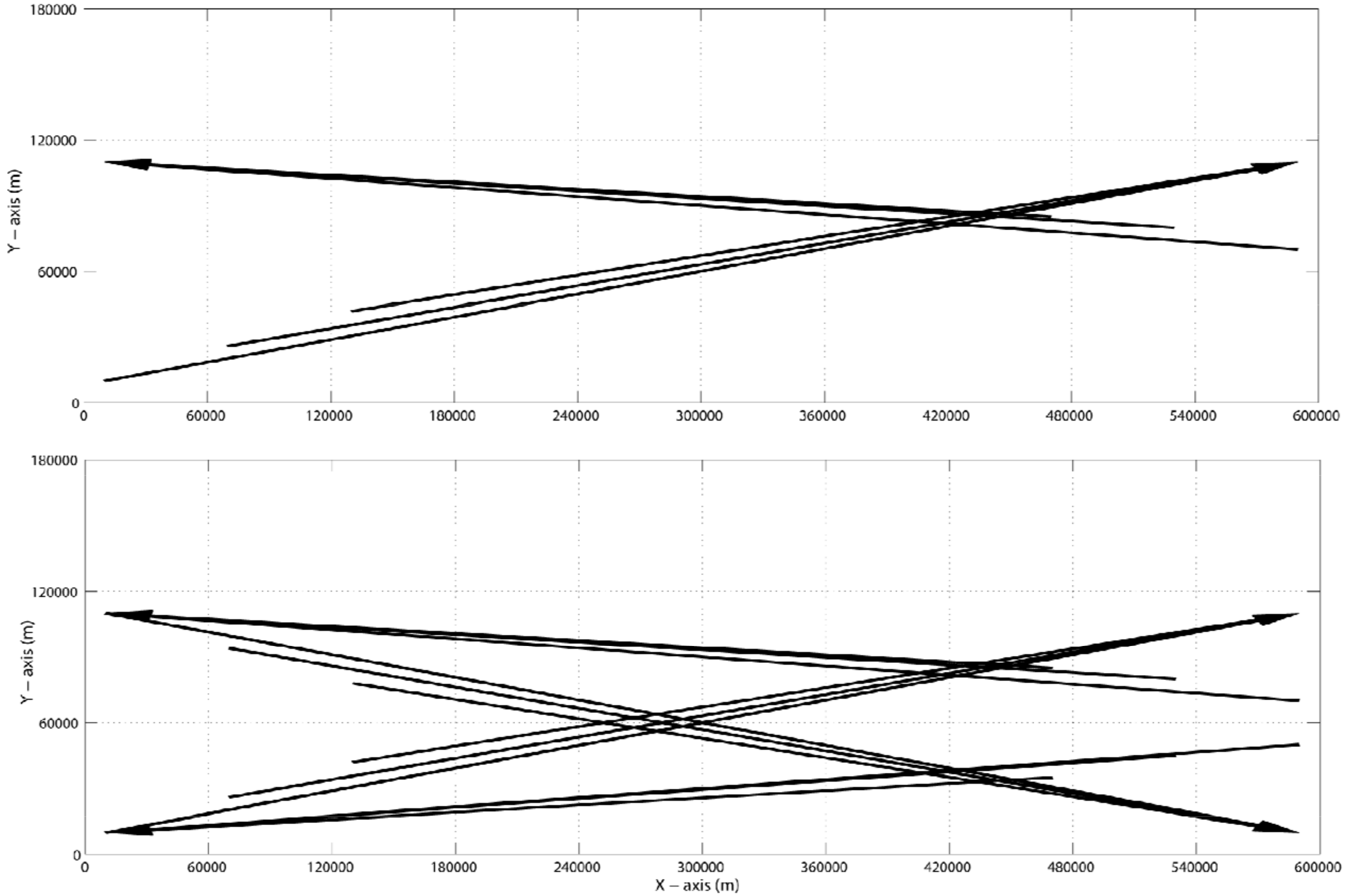
Simulation Setup

- Simulations
  - 600 x 600 km area
  - Wind grid 60 x 60 km, 8 flight levels (up to ~1000 linear states)
  - Different **weather scenarios**
  - Different **aircraft density**
  - 1 to ~250 aircraft (up to 1000 nonlinear states)
  - All aircraft **fly level,** at **constant speed**
  - Radar measurements arrive **every 30 seconds**
- Flights have a **duration of 30 minutes**
  - Process 10 minutes of measurements
  - Predict 20 minutes beyond that
- *Metric*: **Root Mean Square** of Along-Track Error
  - Bias very small in comparison
  - Cross-Track error is small due to the FMS

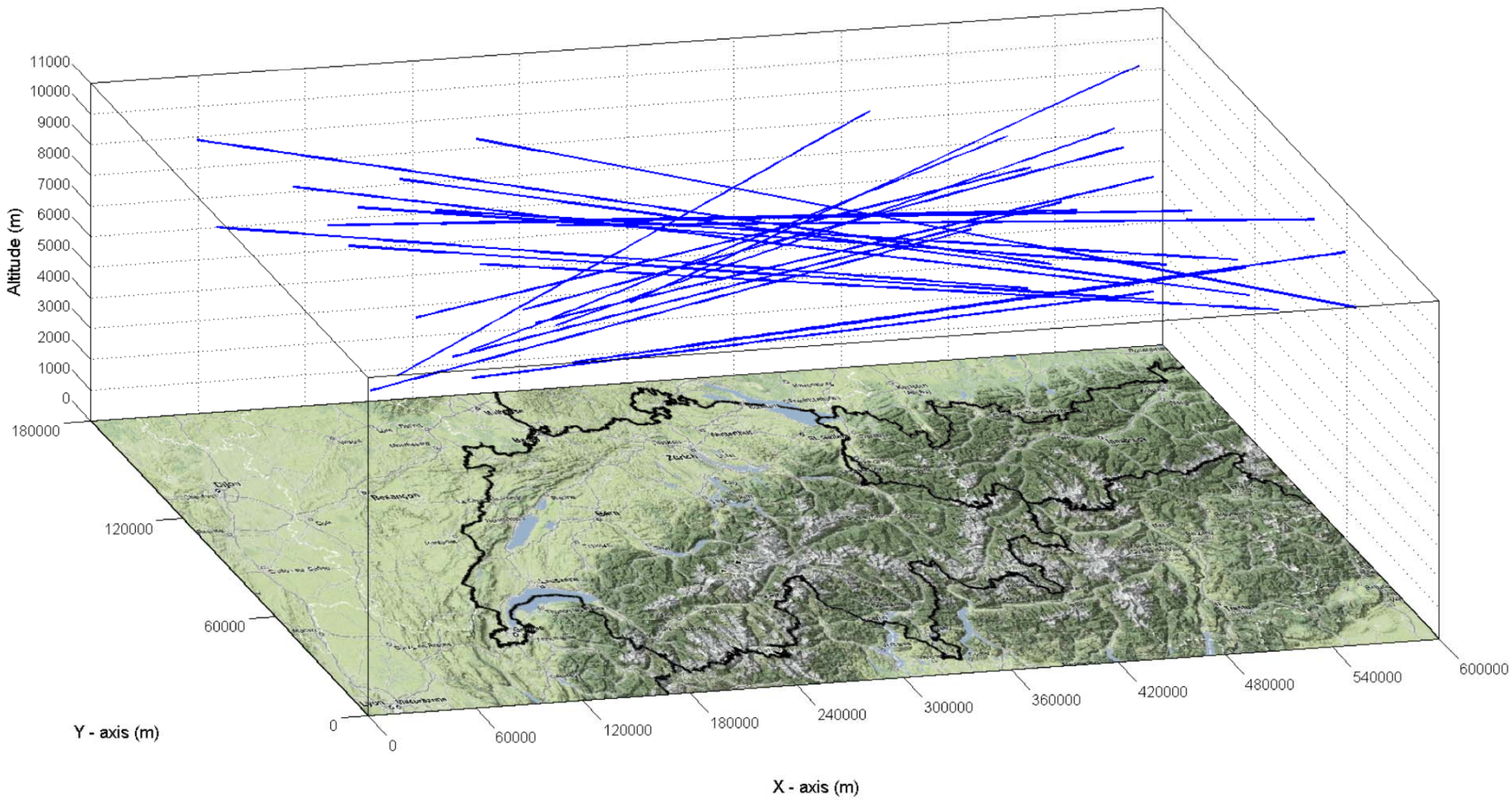
Flight Plans: 1 and 3 aircraft



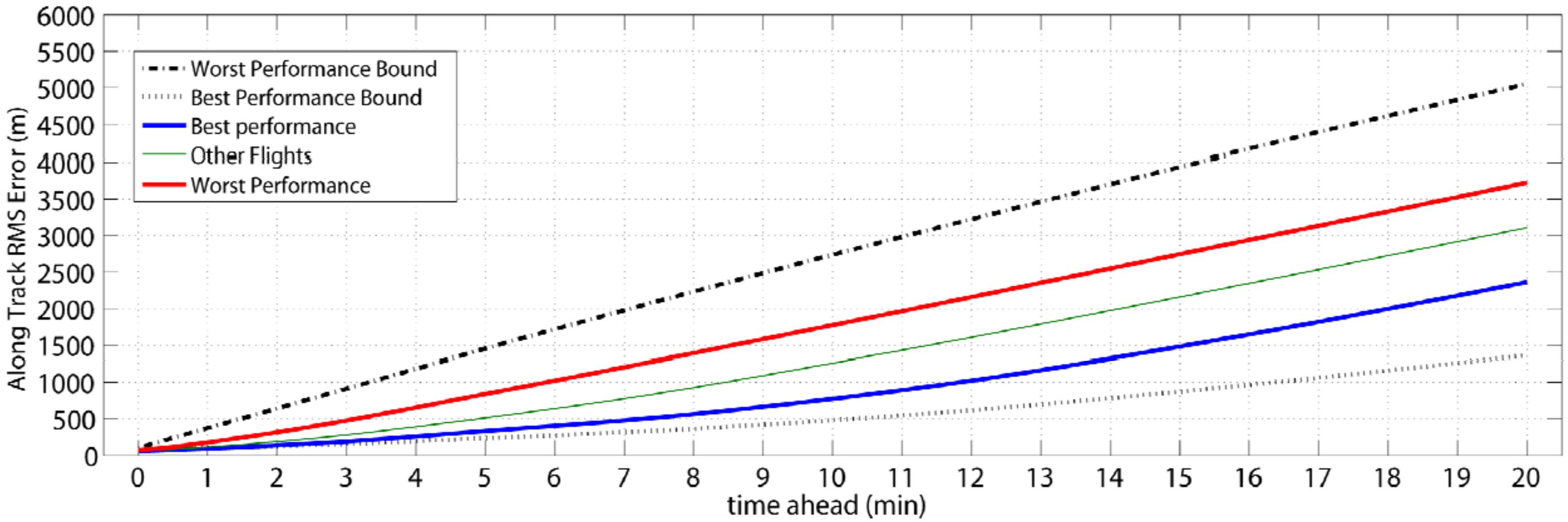
Flight Plans: 6 and 12 aircraft

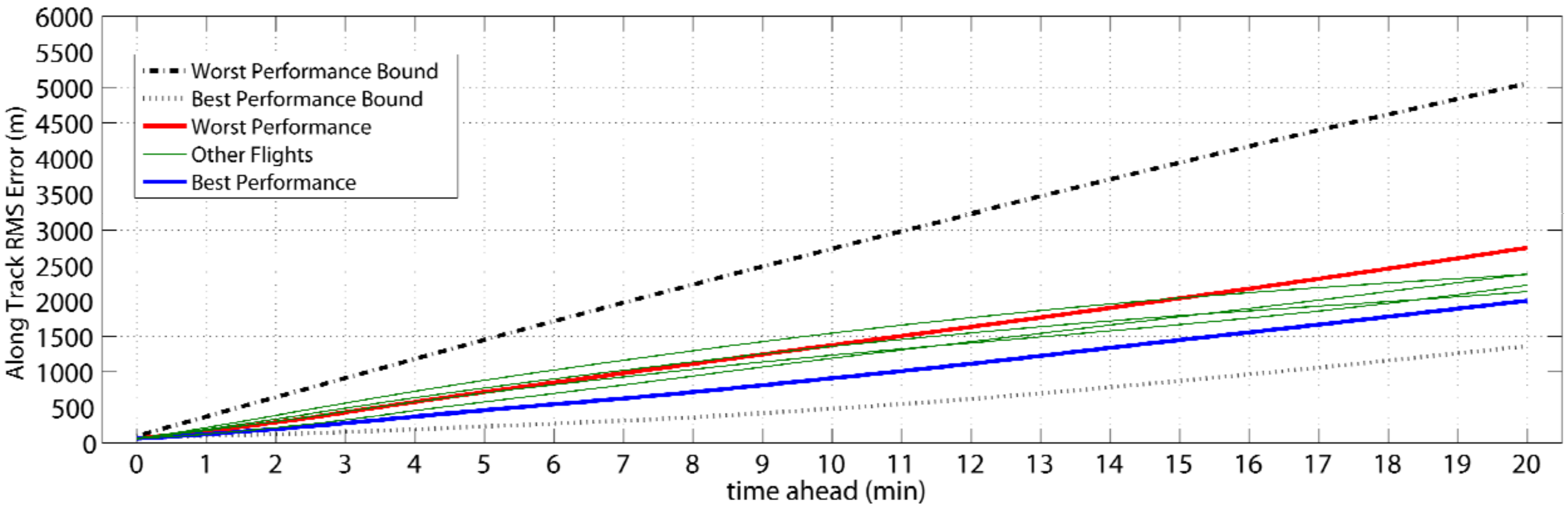


Flight Plans: 24 Aircraft

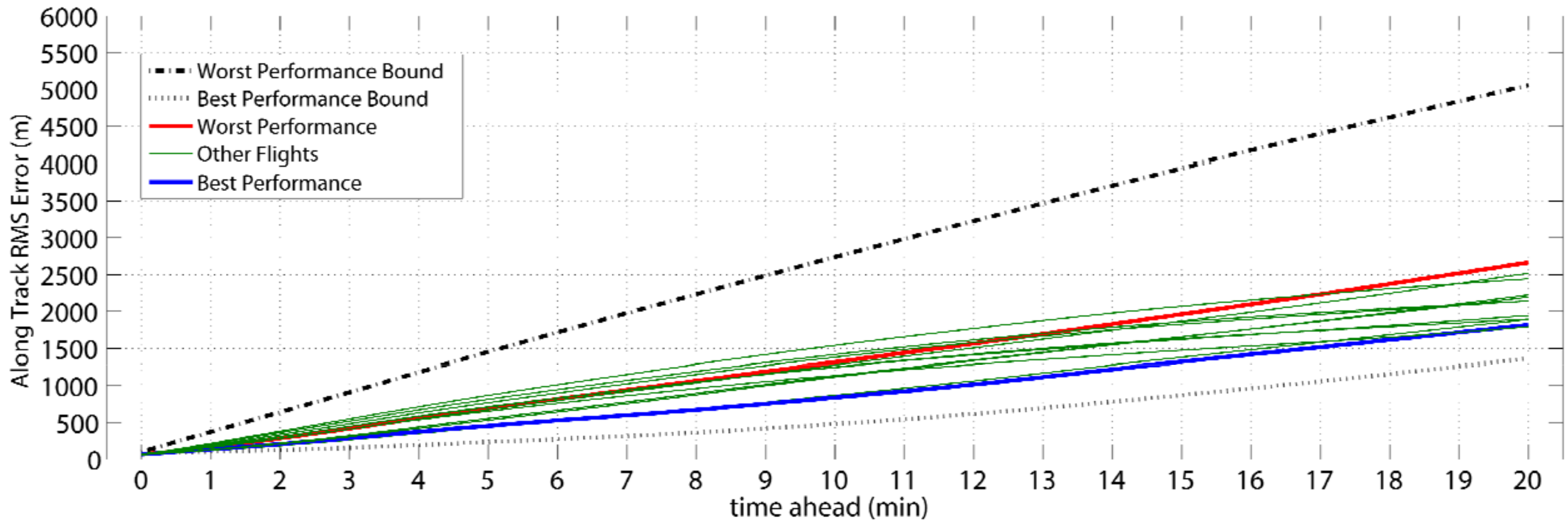


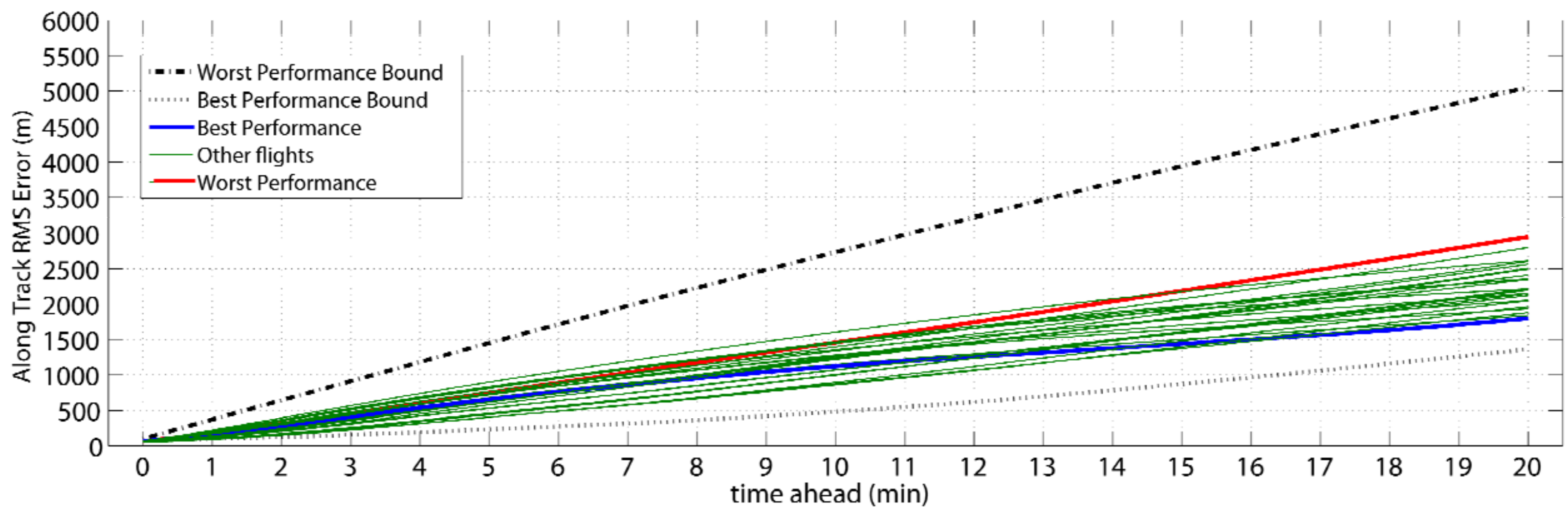
Trajectory Prediction RMSE: 3-6 aircraft





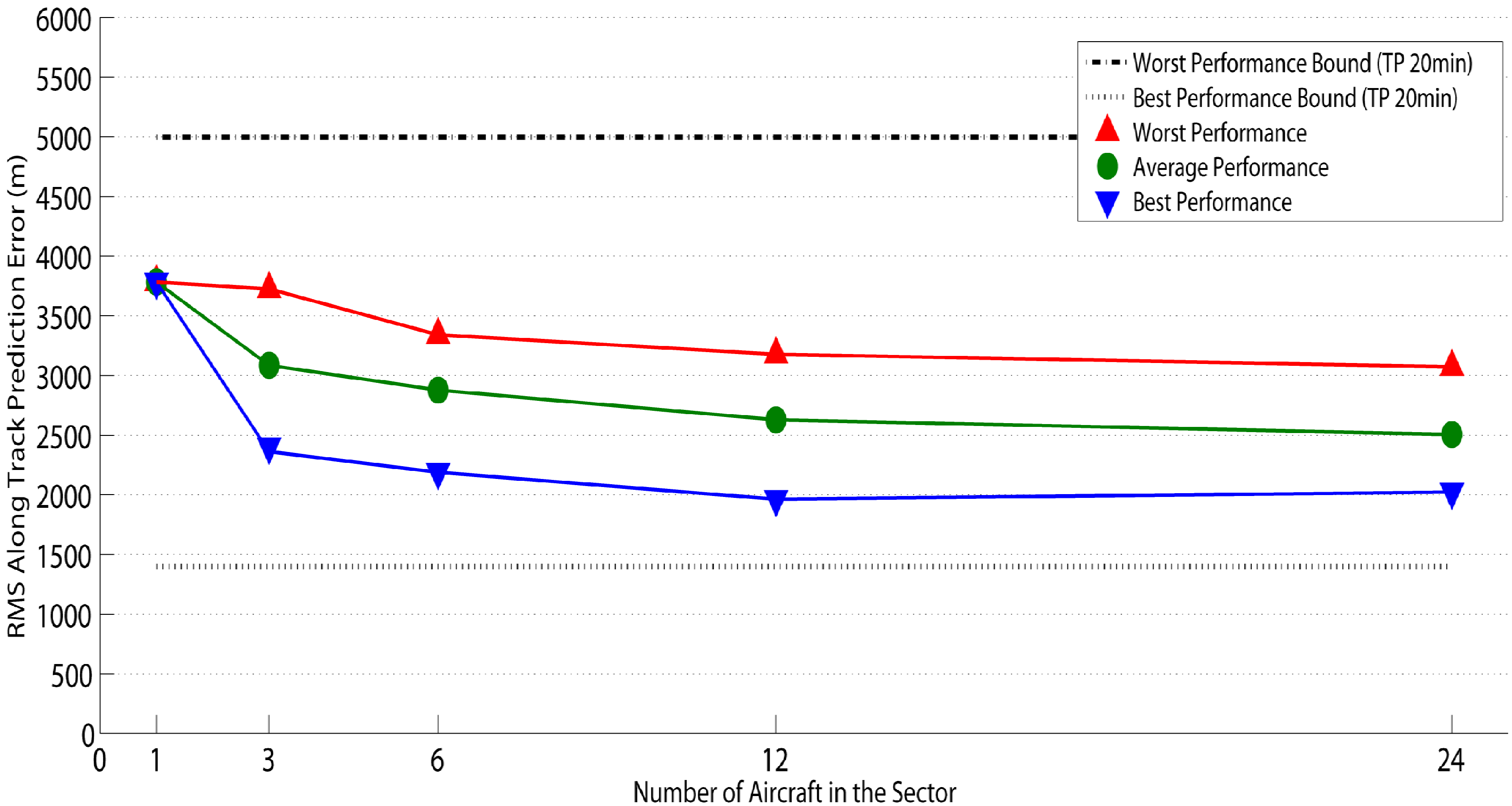
Trajectory Prediction RMSE: 12-24 aircraft



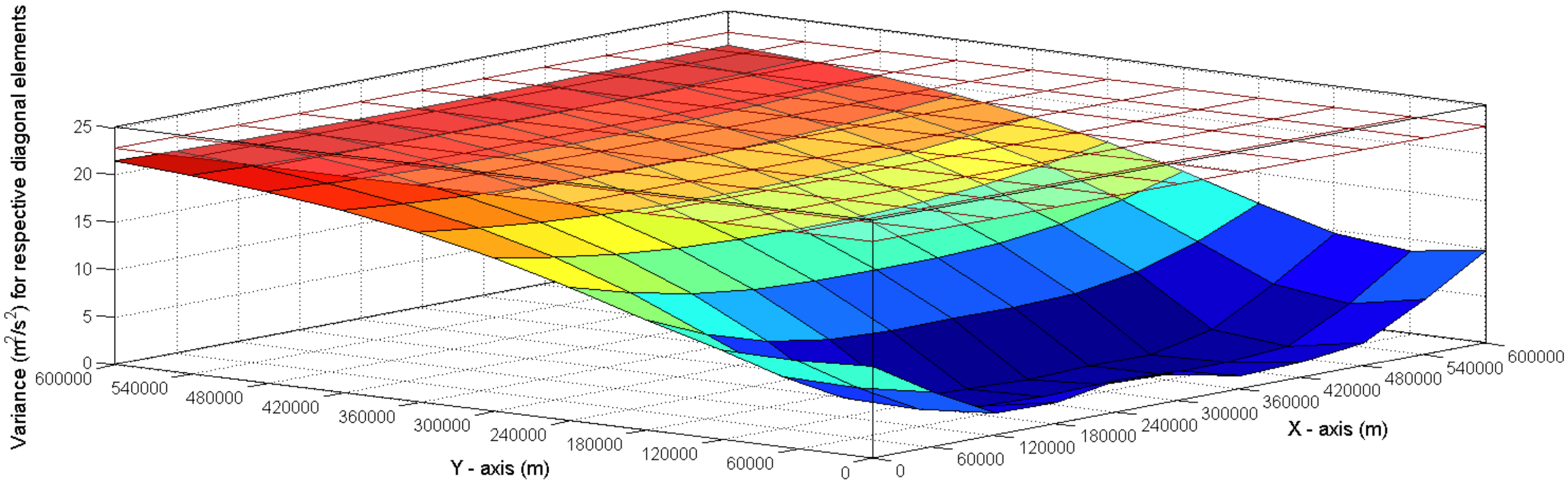


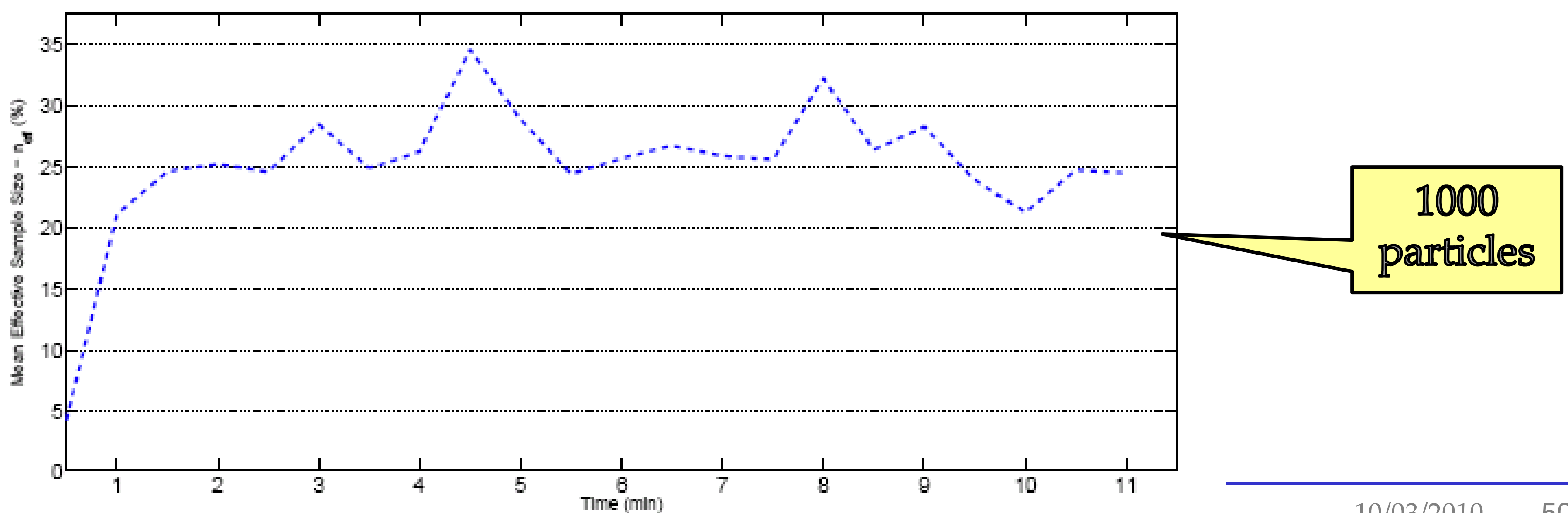
Effect of the number of aircraft

Trajectory Prediction Uncertainty – 20 min ahead

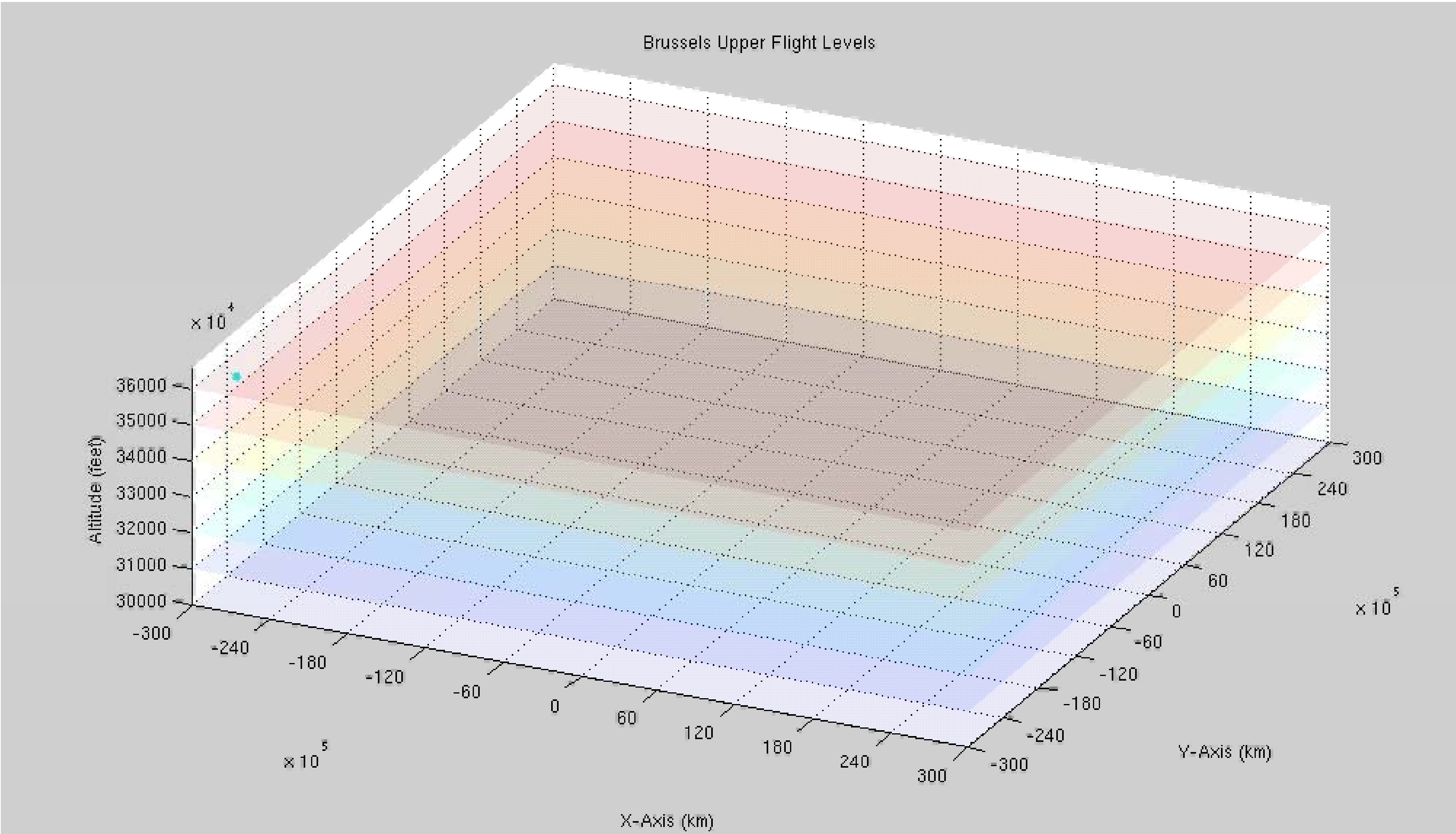


Wind forecast error, effective sample size

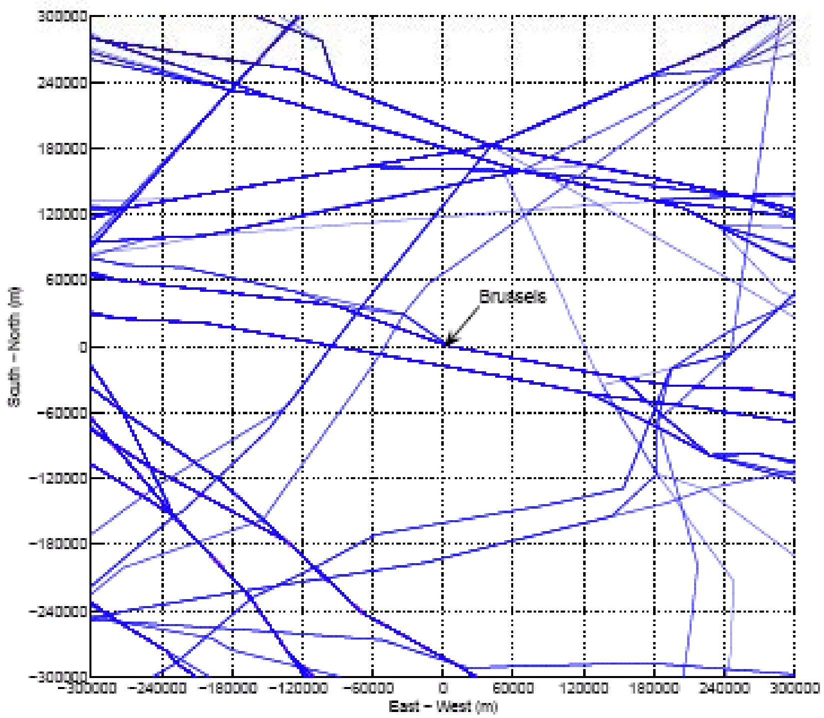


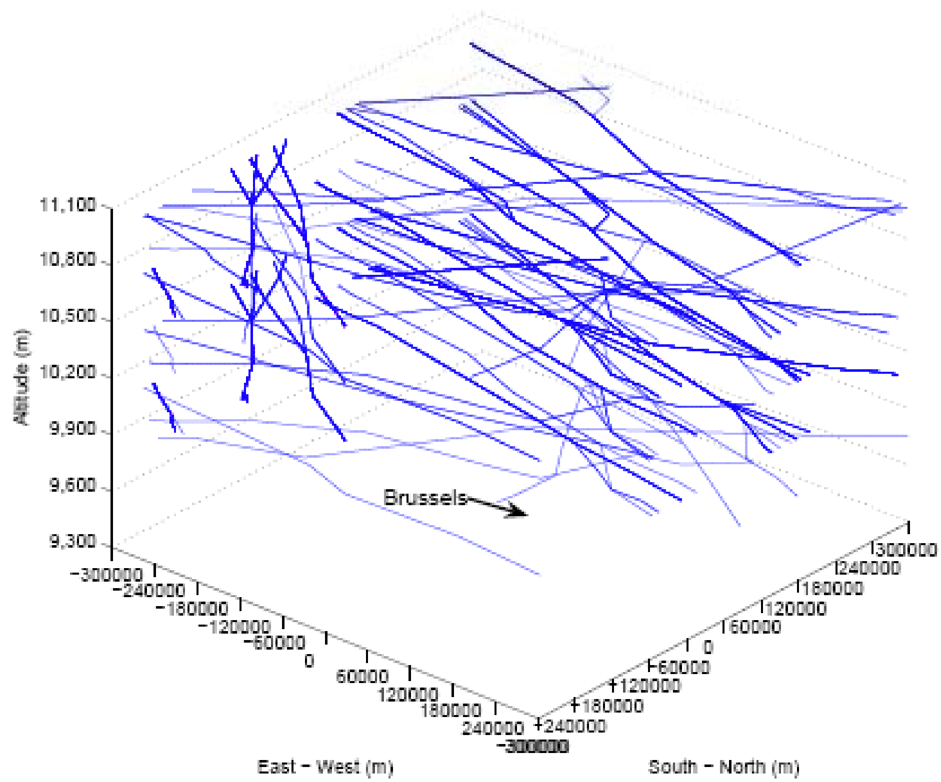


Real Flight Plans - *Brussels*

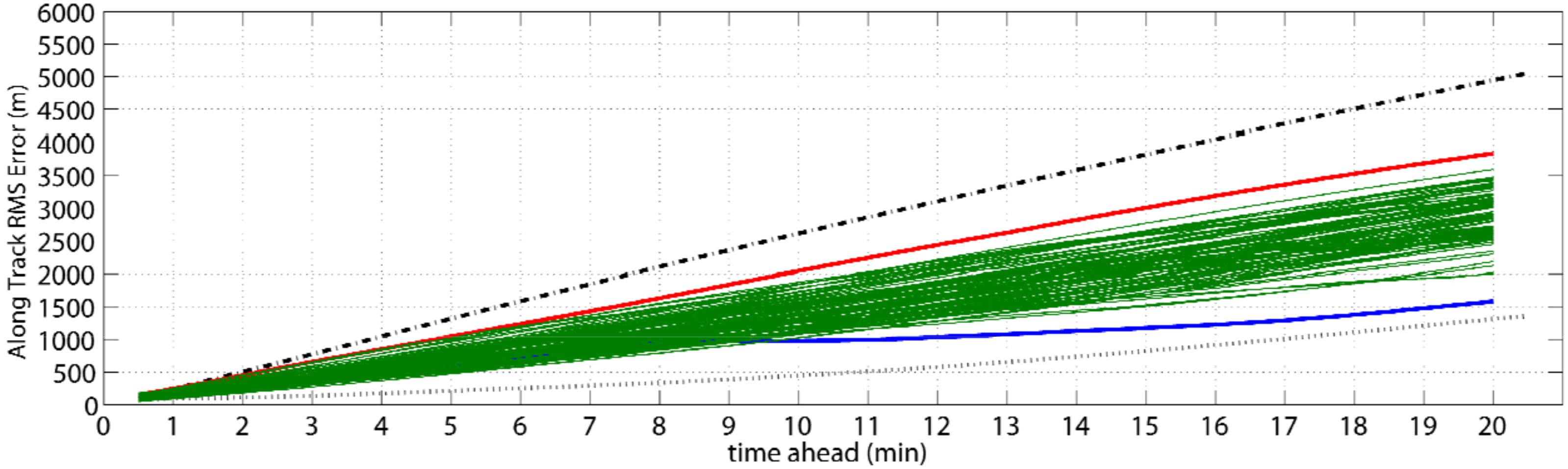


Top & side view

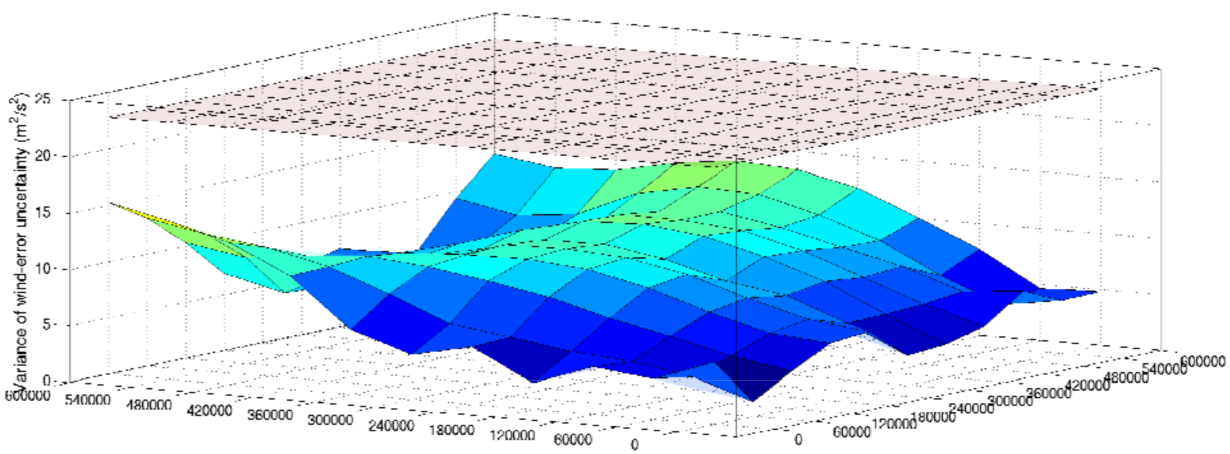




Trajectory Prediction



Wind Forecast Uncertainty



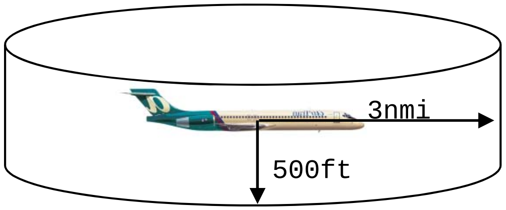
Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

Conflict Detection

Problem:

– Detect future conflicts (conflicts occur when a minimum separation distance between aircraft is violated)



Objective:

Improve Conflict Detection accuracy

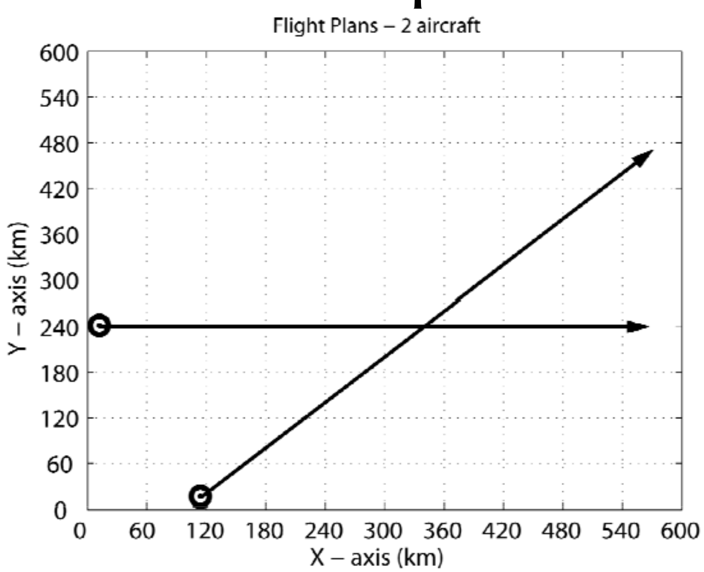
- increase conflict detection rate
- decrease false alarms

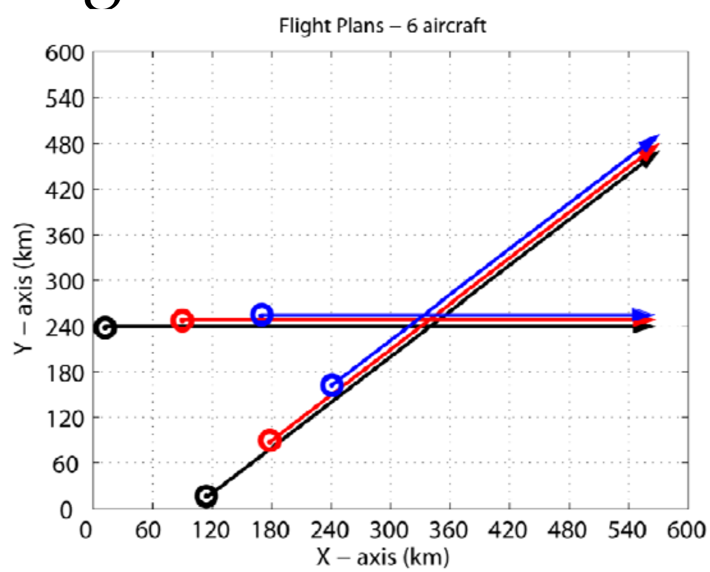
Method:

– Use Radar measurements
– and SCPF to improve Trajectory Prediction accuracy

Simulation Setup

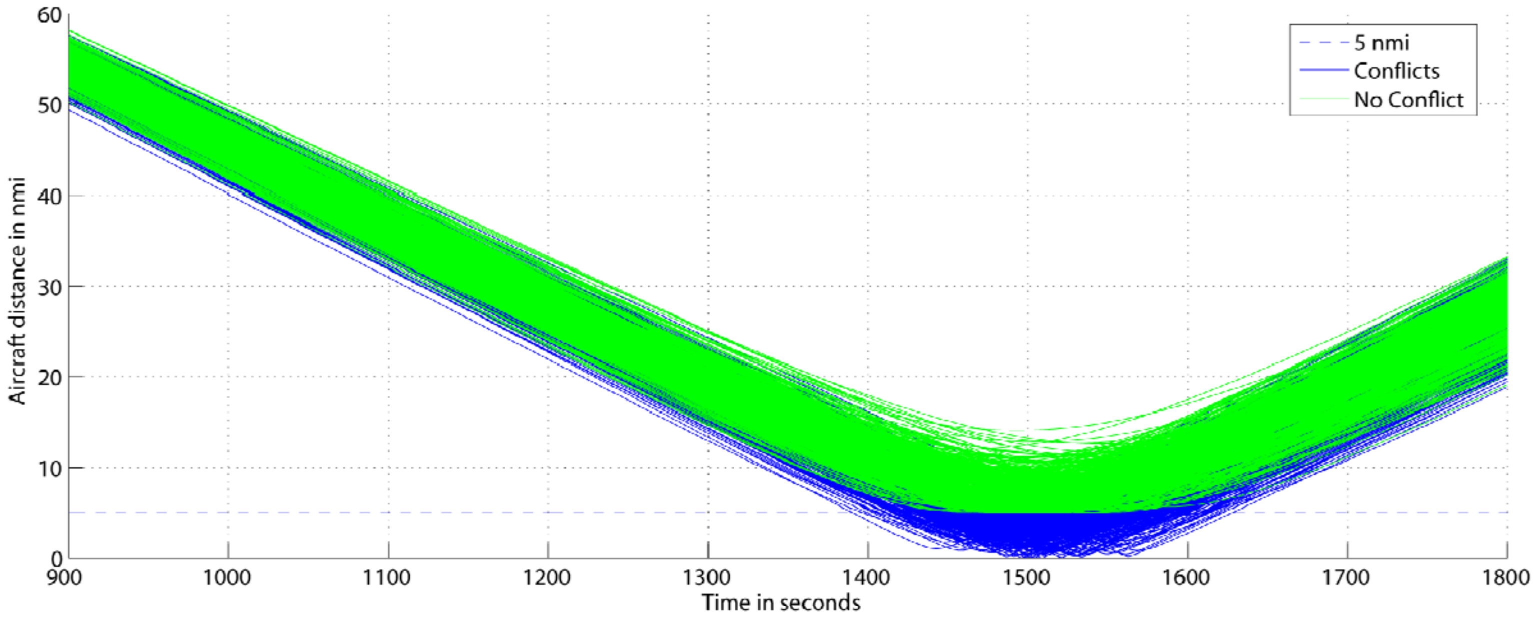
- 2 aircraft approaching at 45^o incidence
- Aircraft fly level at constant speed
- 5mi separation after 25 minutes (for zero forecast error)
- 4 aircraft precede the flights of interest





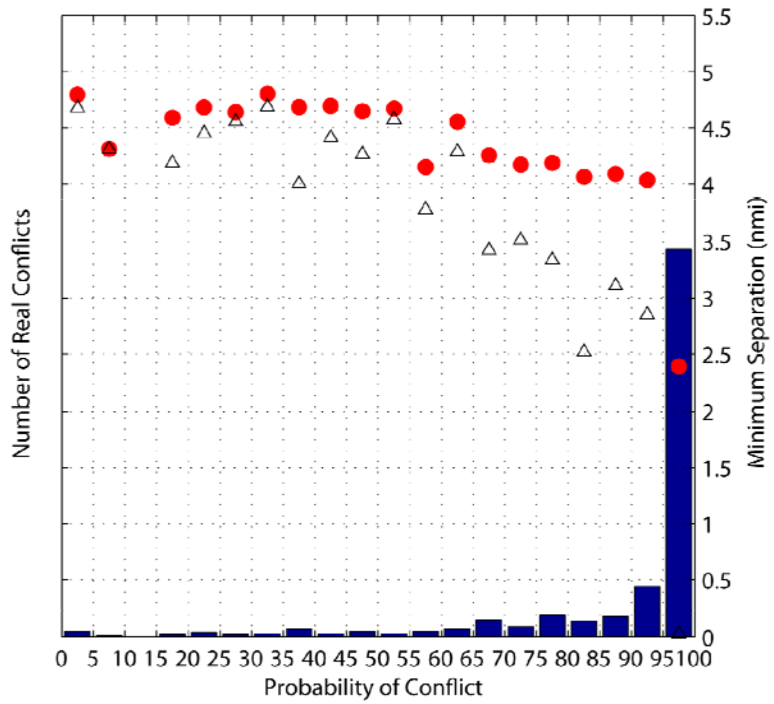
Wind Forecast Errors

- 1000 different wind forecast errors
- **509** lead to conflict
- **491** escape conflict



Simulation Results

- As aircraft approach each other the probability of conflict changes
  - ➢ Filter Radar measurements using SCPF
  - ➢ Identify wind-forecast errors
  - ➢ Evaluate the probability of conflict



Scenarios with conflicts

- Scenarios are placed in different bins according to the calculated probability of conflict

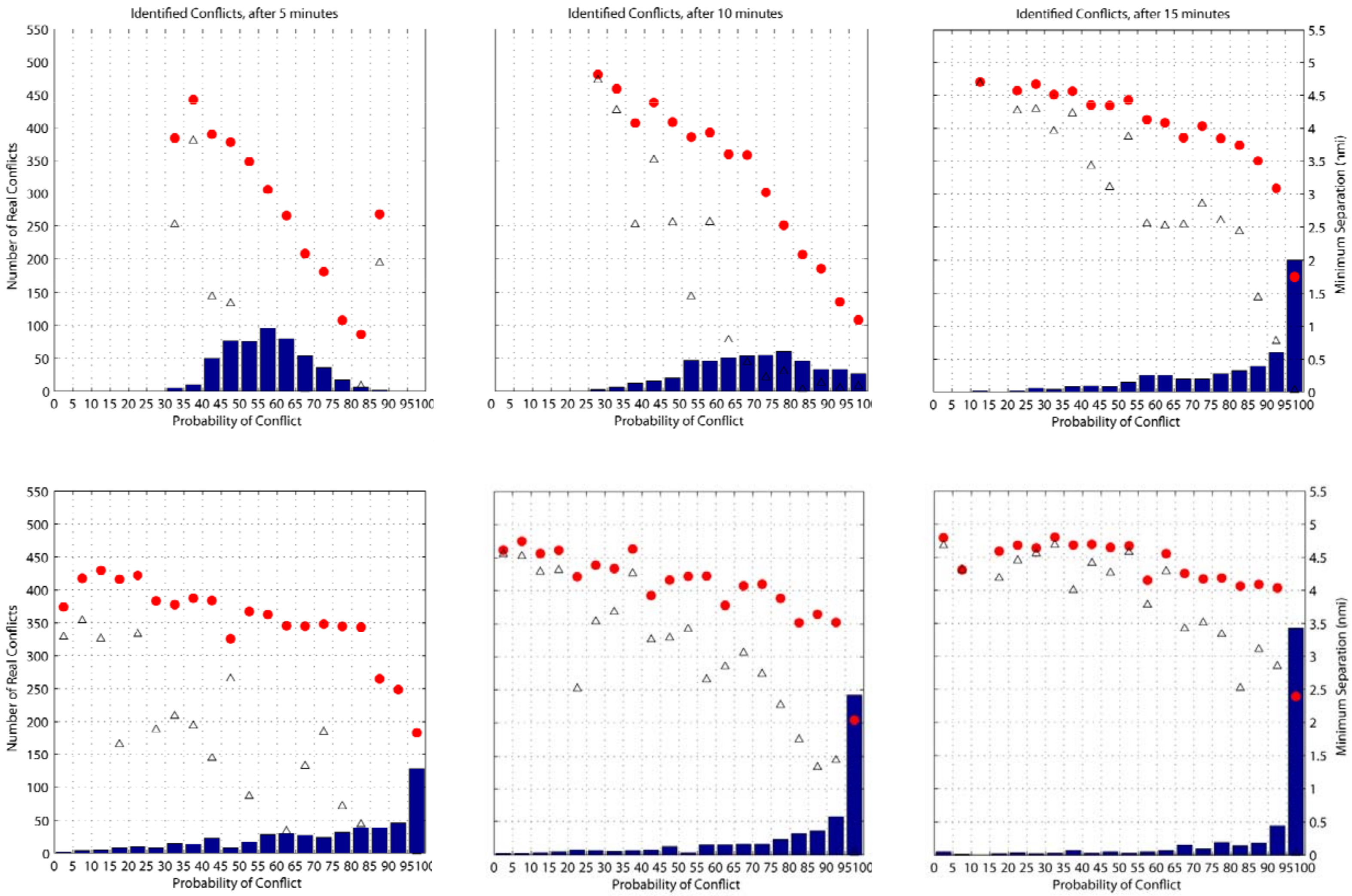
● **Average minimum separation**

△ Smallest minimum separation

Conflict Probability (when a conflict occurred)

Only wind forecasts available

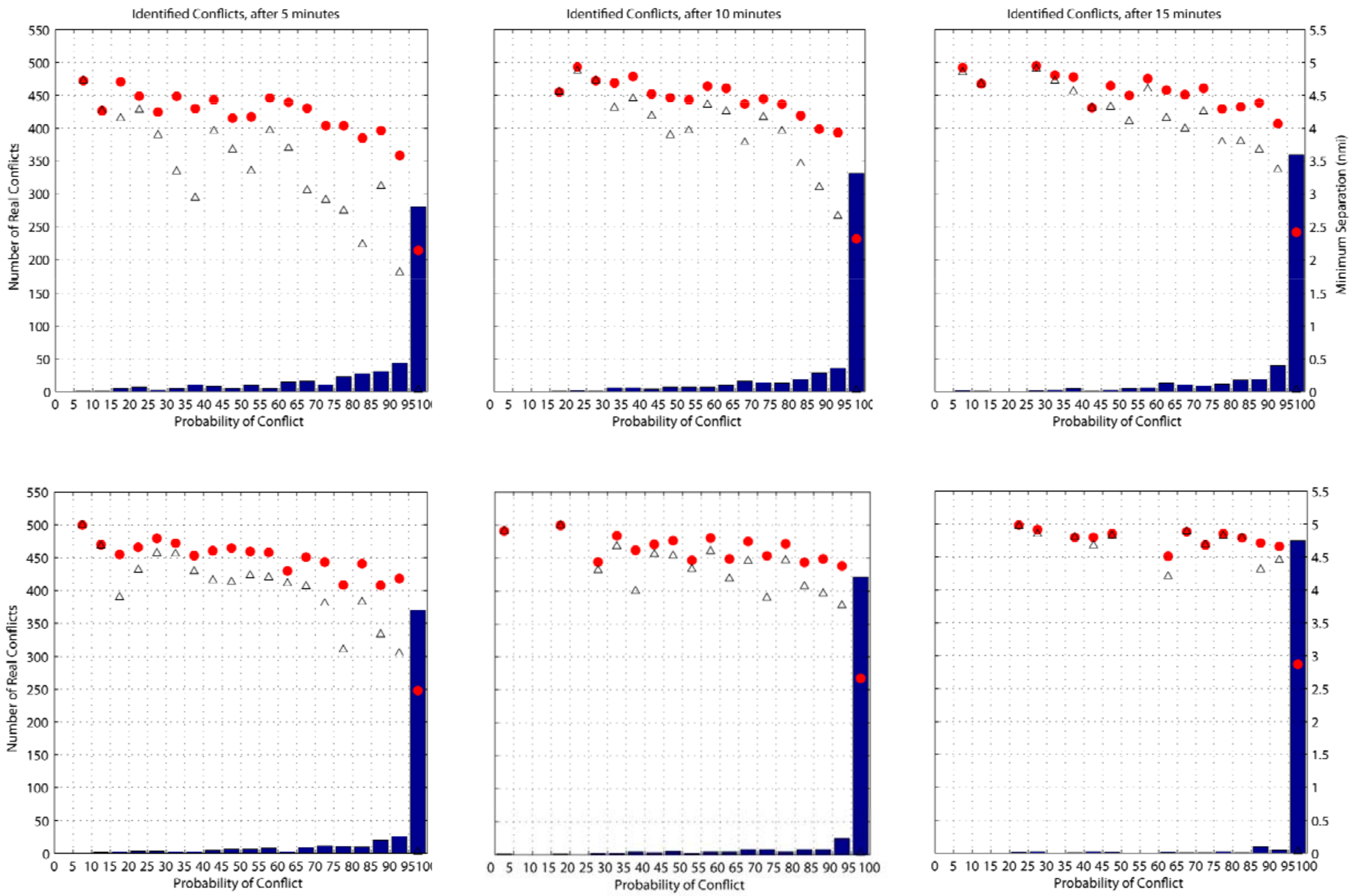
SCPF with 2 aircraft



Conflict Probability (when a conflict occurred)

SCPF with
6 aircraft

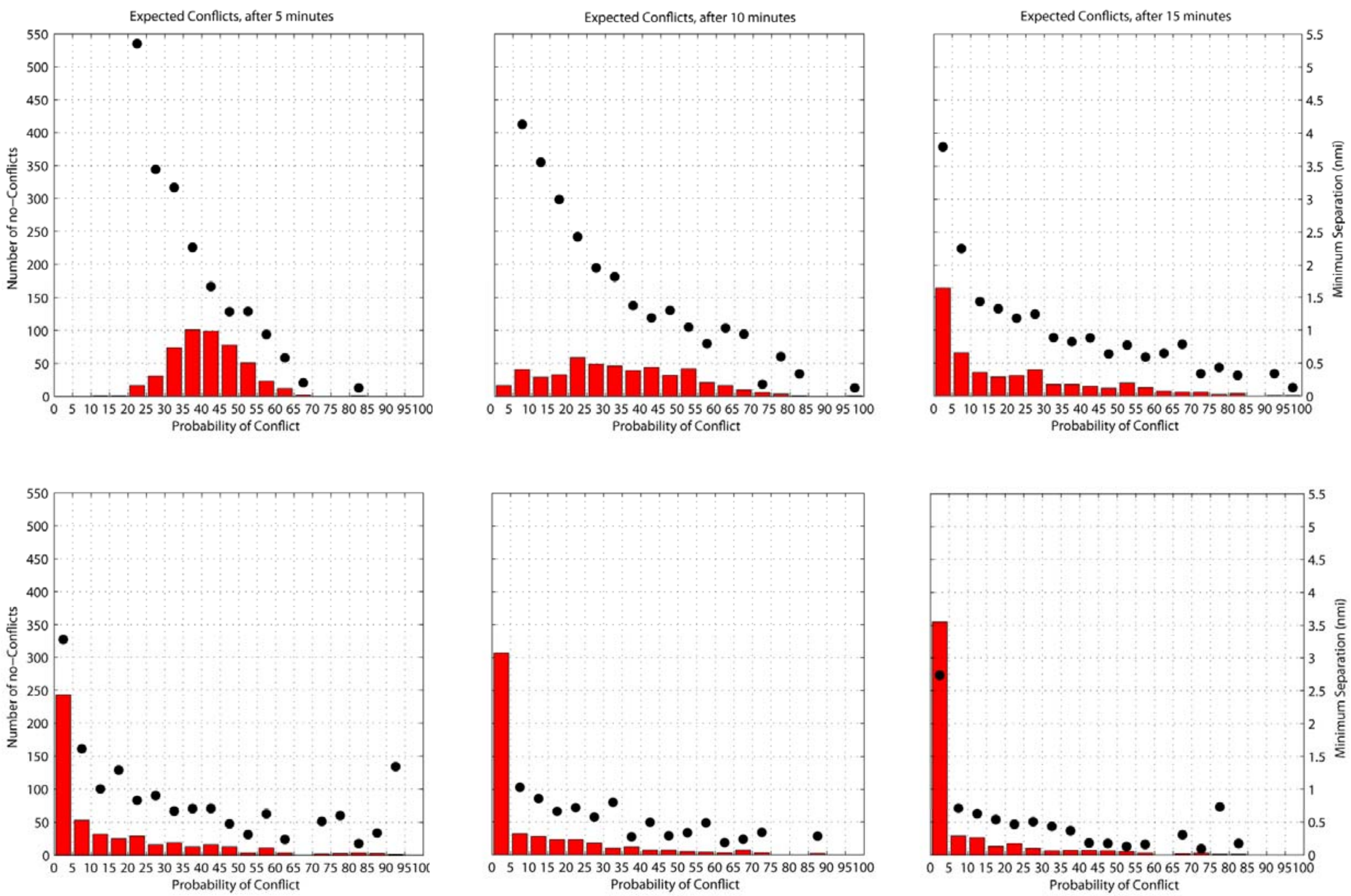
Best
performance
bound



Conflict Probability (for conflict free scenarios)

Only wind forecasts available

SCPF with 6 aircraft



Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

Combined State & Parameter Estimation

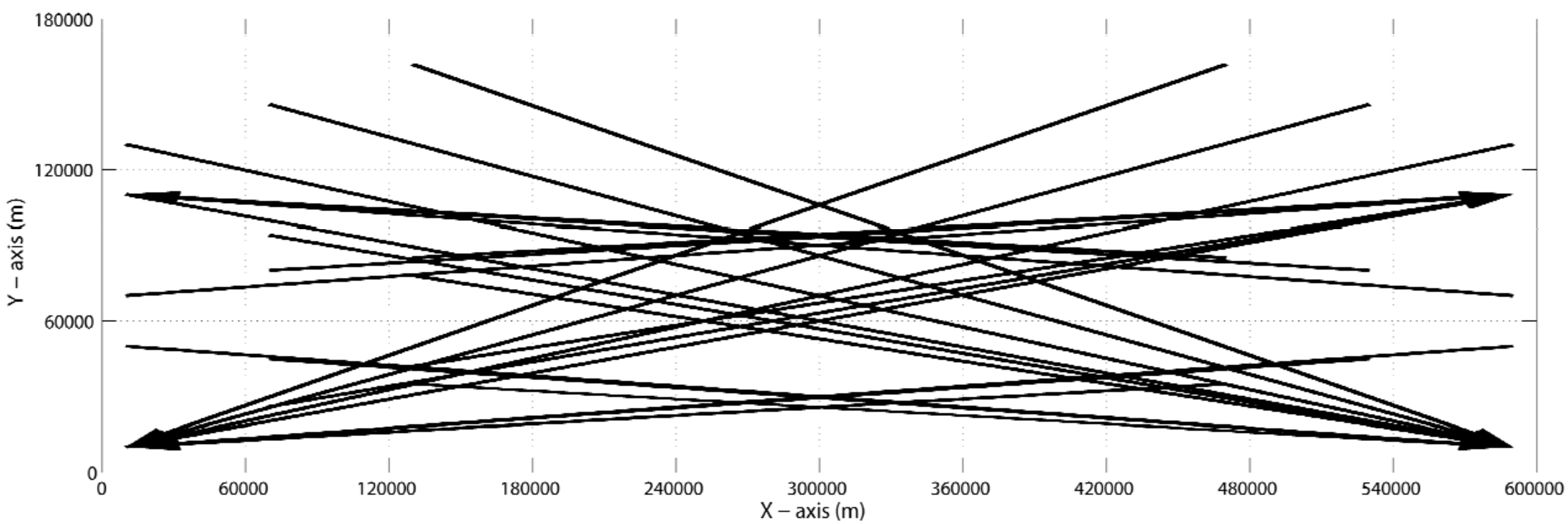
- Assume aircraft fly with constant unknown airspeeds

- Augment state vector to include parameters
- Add artificial noise at each time step
- Use again SCPF to filter radar measurements and estimate both **wind** and **airspeeds**.

- Other methods
  - Kernel Density Estimation
  - Smoothing

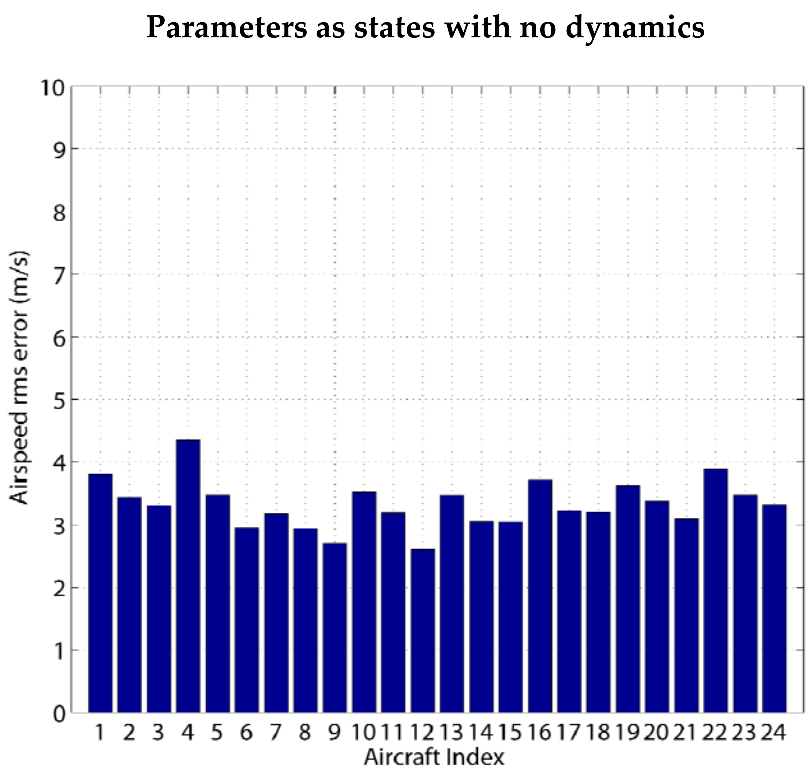
Simulation Setup

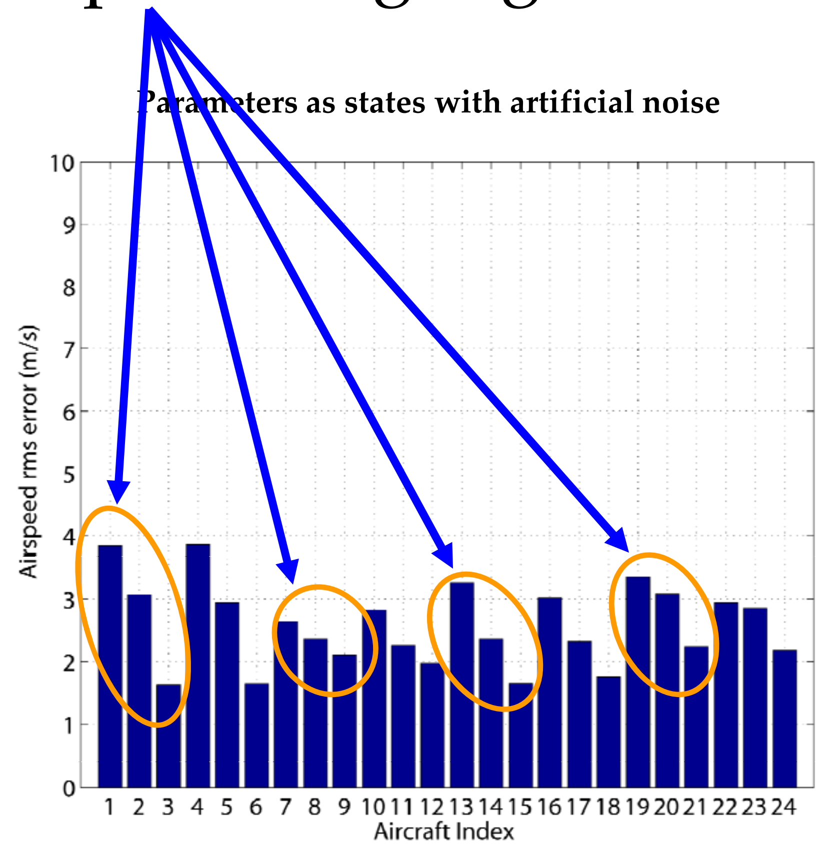
- **24** aircraft
- **400** different weather and airspeed scenarios
- Airspeeds normally distributed with **mean 215 m/s** and **s.d. 10 m/s** (truncated at the operational limits of the aircraft – **187 to 250 m/s**)



Airspeed Estimation using SCPF

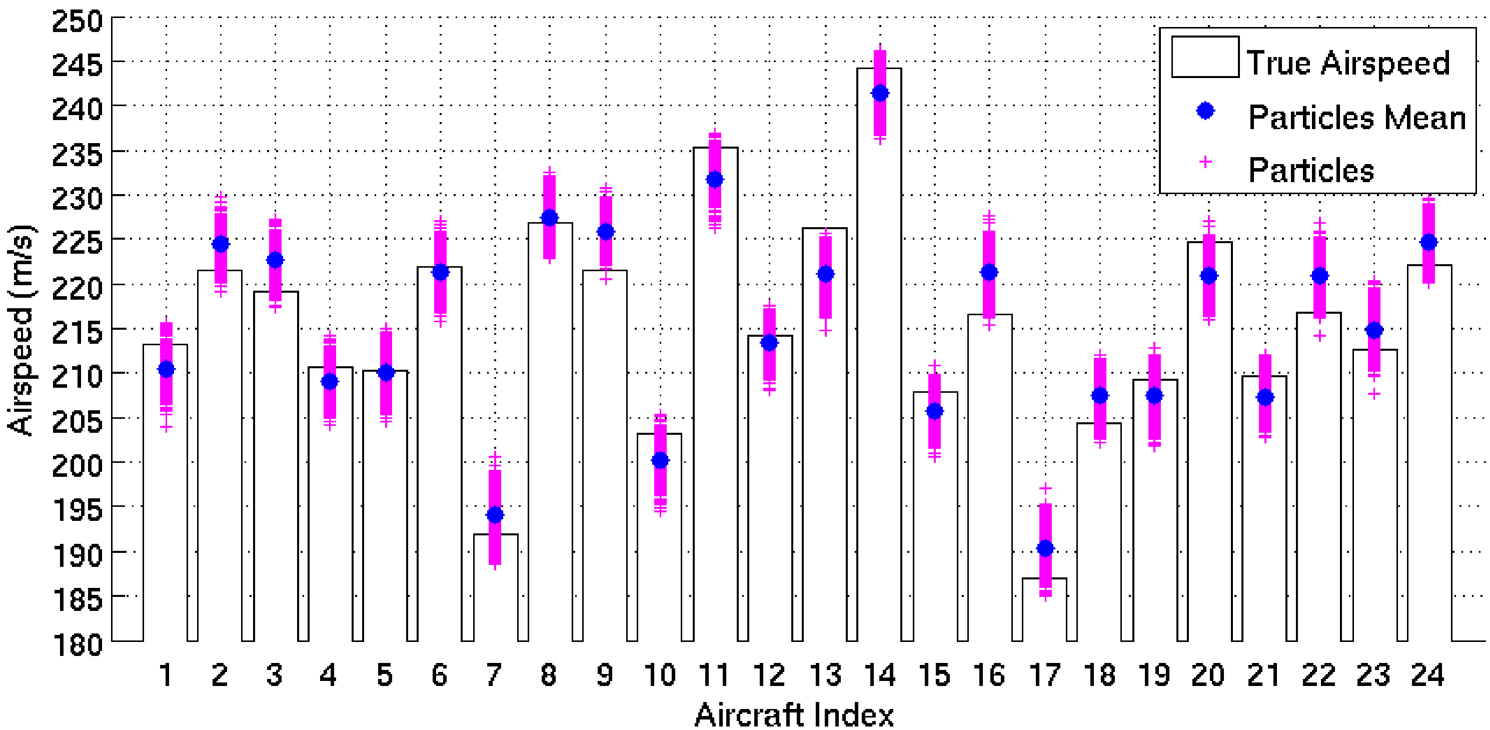
- Airspeed uncertainty decreases
- Aircraft take advantage of preceding flights





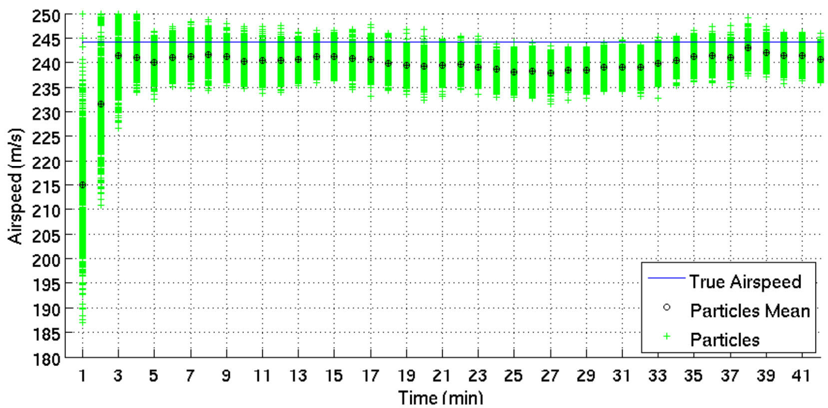
Airspeed Estimation – single scenario

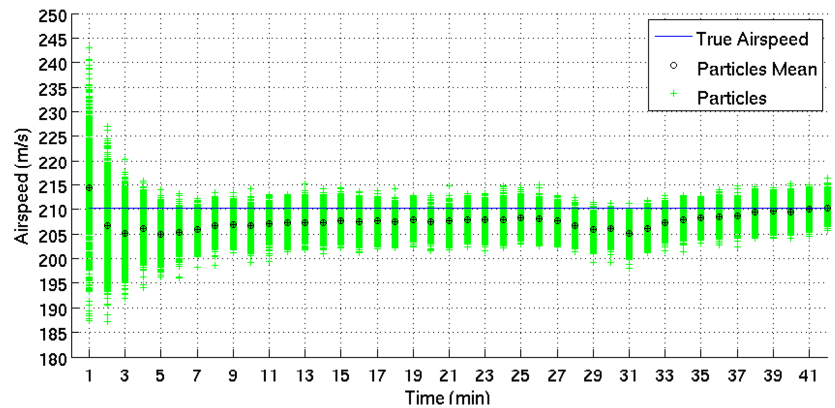
- Airspeed estimation after 41 measurements
- Particles cluster near the real speed of each aircraft

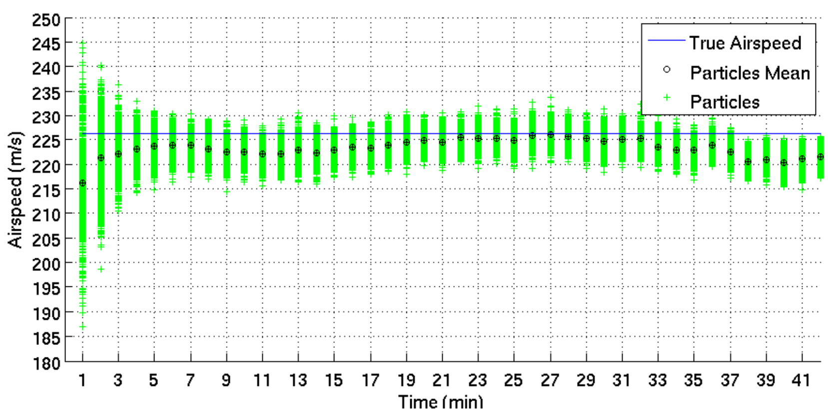


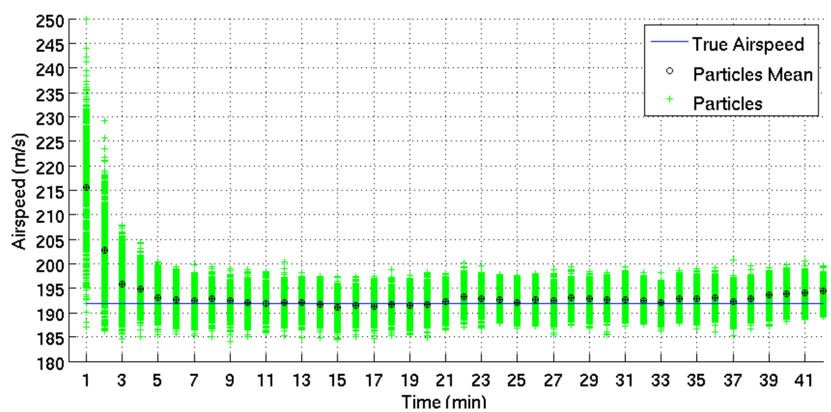
Individual Aircraft

- Particles move fast towards the real airspeed
- Results are good both for medium and extreme speeds



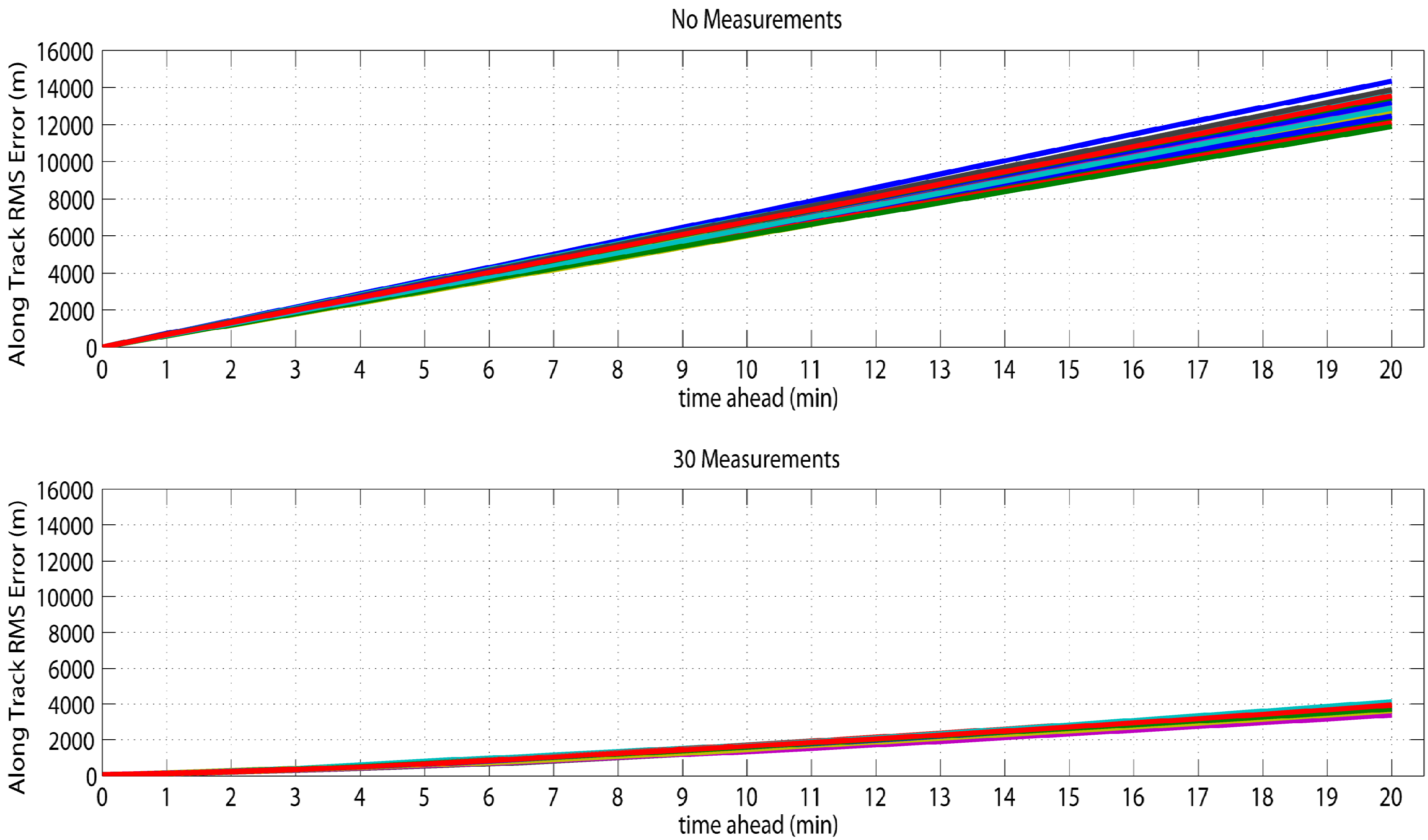






Trajectory Prediction Improvement

- Accuracy improvement due to **wind** and **airspeed** identification



Summary

- INTRODUCTION
- DYNAMICS
- NONLINEAR FILTERING
- TRAJECTORY PREDICTION
- CONFLICT DETECTION
- AIRSPEED ESTIMATION
- CONCLUSION

Conclusion

- Improve aircraft trajectory prediction
  - Filter aircraft measurements
  - Wind forecast error & airspeed estimation
  - Improved trajectory & conflict prediction
- Based on novel particle filtering algorithm
  - Sequential conditioning particle filter
  - Exploit problem structure
- Current work
  - Incorporate on-board datalinked measurements
  - Improved forecast error models
  - Apply to other problems with similar structure