Distributed Multi-Sensor Fusion

Simon Julier
s.julier@ucl.ac.uk
Department of Computer Science
University College London

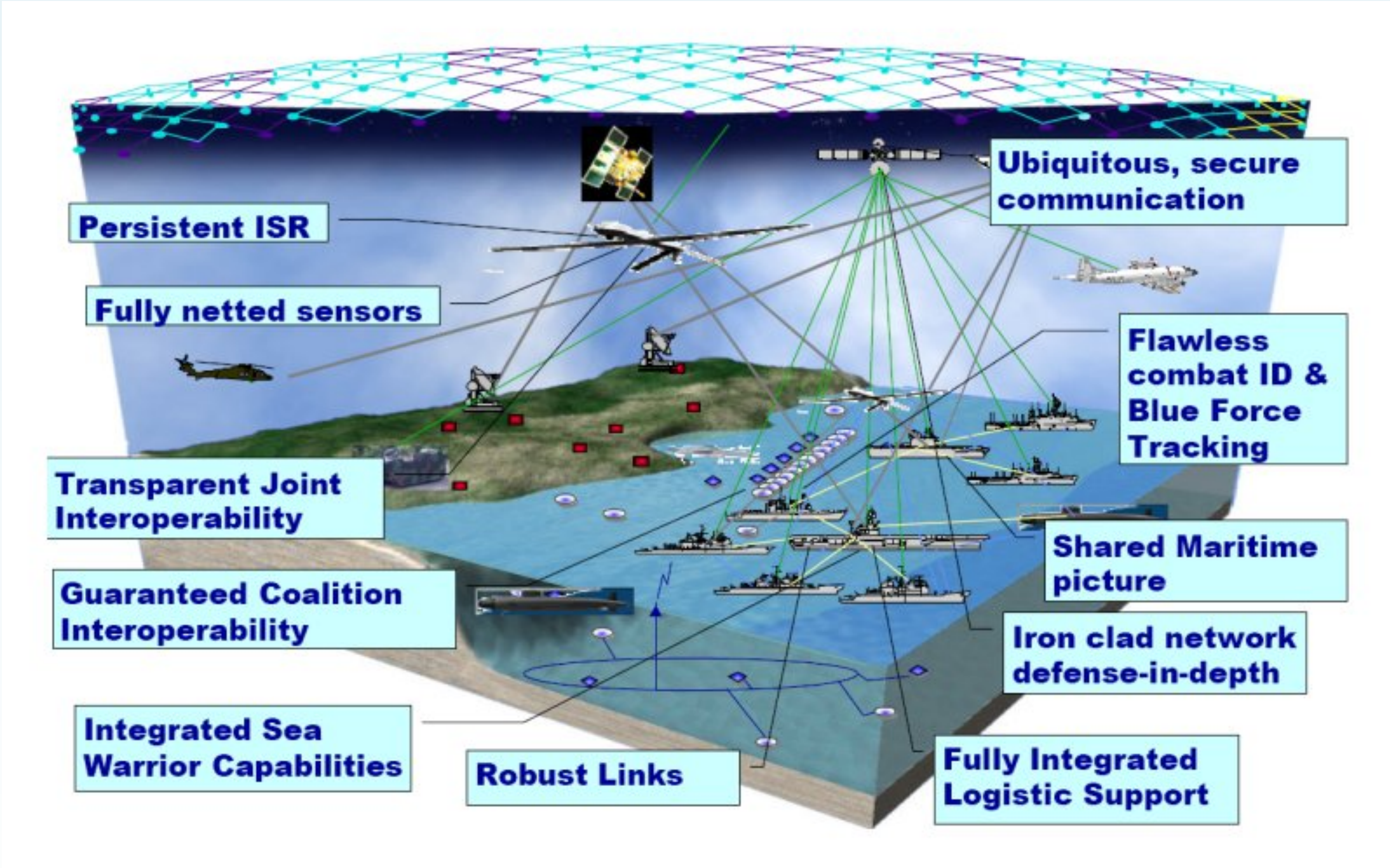
Acknowledgements (Alphabetical Order)

- Gabriel Agamennoni, ETH Zurich
- John Andrews, (formerly BAE Systems)
- Tim Bailey, University of Sydney
- Jordi Barr, Dstl (formerly BAE Systems)
- Daniel Clark, Heriot-Watt
- Amadou Gning, UCL
- Ronald Mahler, (formerly Lockheed Martin)
- David Nicholson, BAE Systems
- Branko Ristic, DSTO
- Paul Thomas, Dstl
- Jeff Uhlmann, University of Missouri-Columbia
- Murat Üney, University of Edinburgh

Structure of Talk

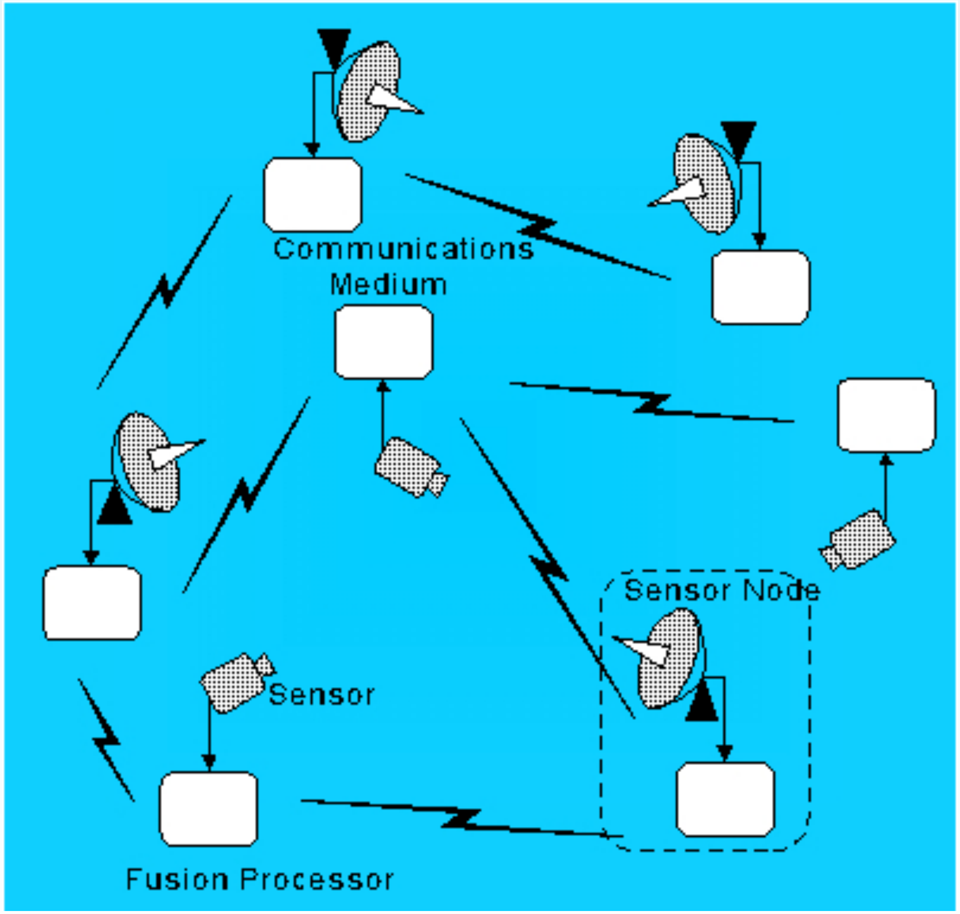
- Motivation
- Distributed data fusion
- Suboptimal distributed data fusion
- Distributed multi-object tracking with PHD filters

Distributed Fusion Architecture



Motivation

- Nodes fuse data from
  - Local observations
  - Local filter predictions
  - Communicated information
- A dynamic network of sensing nodes
  - *No* central processor
  - *No* central communications
  - *No* local knowledge of global network topology
- Scalable, survivable and modular

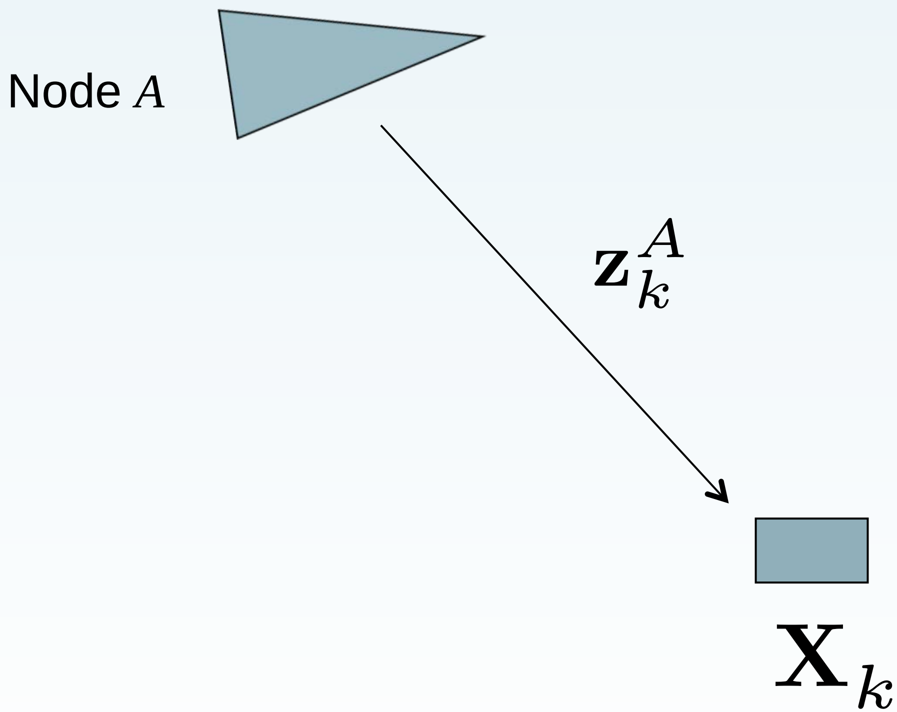


Distributed data fusion

- Motivation
- *Distributed data fusion*
- Suboptimal distributed data fusion
- Distributed multi-object tracking with PHD filters

Single Platform Case

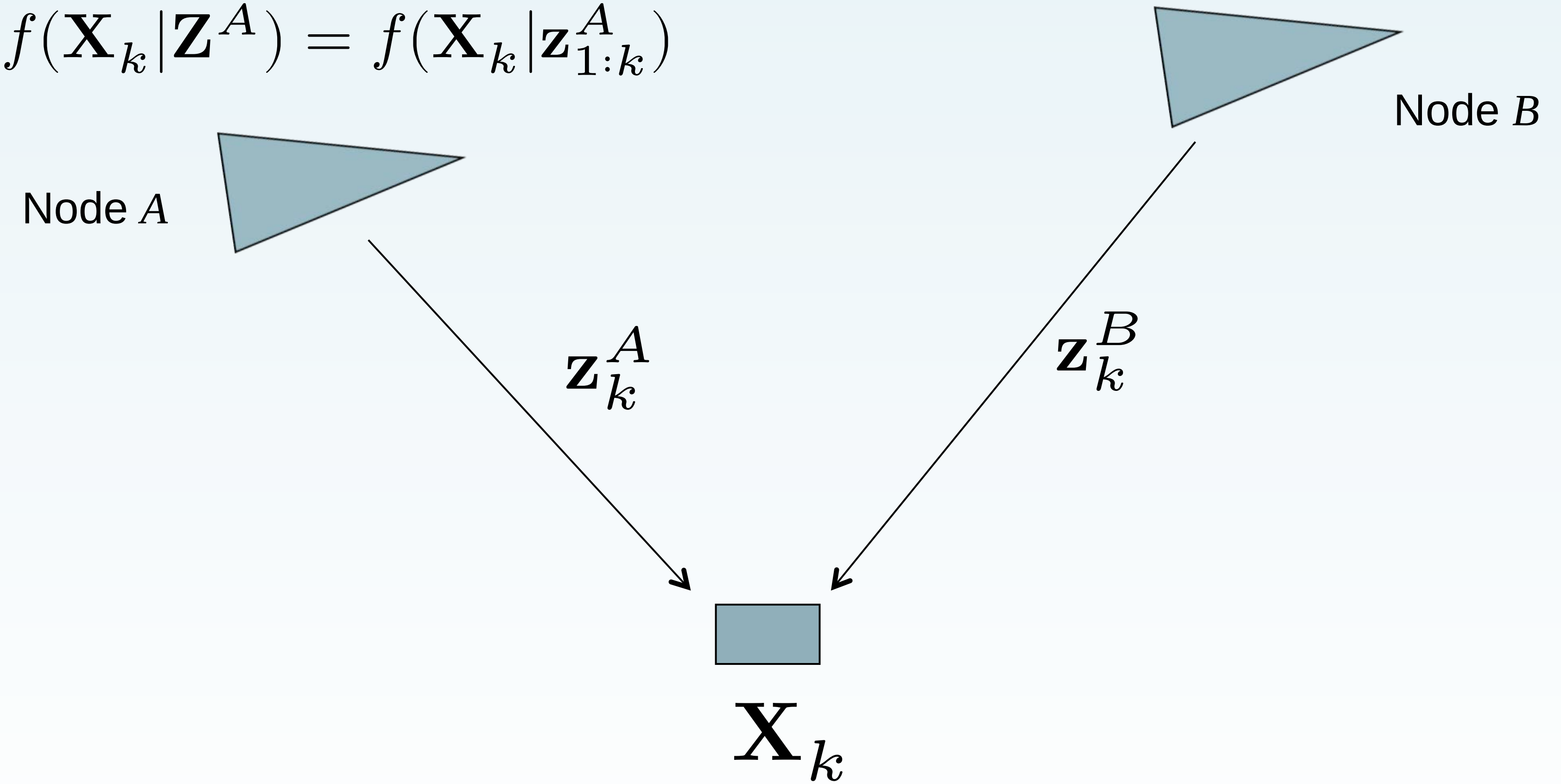
$$f(\mathbf{X}_k|\mathbf{Z}^A) = f(\mathbf{X}_k|\mathbf{z}^A_{1:k})$$



Multiple Independent Platform Case

$$f(\mathbf{X}_k|\mathbf{Z}^B) = f(\mathbf{X}_k|\mathbf{z}^B_{1:k})$$

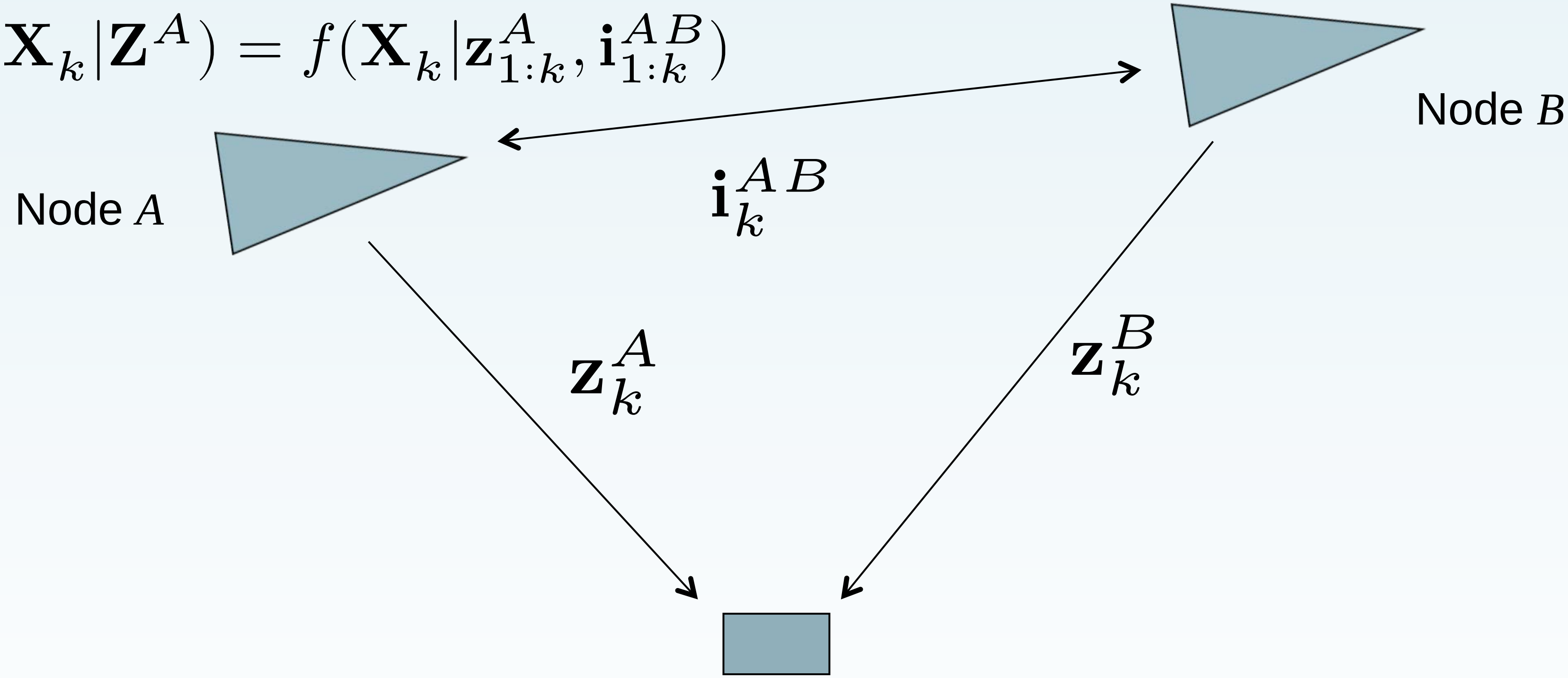
$$f(\mathbf{X}_k|\mathbf{Z}^A) = f(\mathbf{X}_k|\mathbf{z}^A_{1:k})$$



Multiple Distributed Platform Case

$$f(\mathbf{X}_k|\mathbf{Z}^B) = f(\mathbf{X}_k|\mathbf{z}^B_{1:k}, \mathbf{i}^{BA}_{1:k})$$

$$f(\mathbf{X}_k|\mathbf{Z}^A) = f(\mathbf{X}_k|\mathbf{z}^A_{1:k}, \mathbf{i}^{AB}_{1:k})$$



Kalman Filter Formulation

- Each platform maintains its own estimate of the target state,

$$\left\{\hat{\mathbf{x}}_n\left(i \mid j\right), \mathbf{P}_n\left(i \mid j\right)\right\}$$

- Each node runs a Kalman filter locally and fuses locally taken measurements
- The update is distributed to other nodes which fuse with it

Properties of "Ideal DDF"

- The estimate in the network should (eventually) be the same everywhere

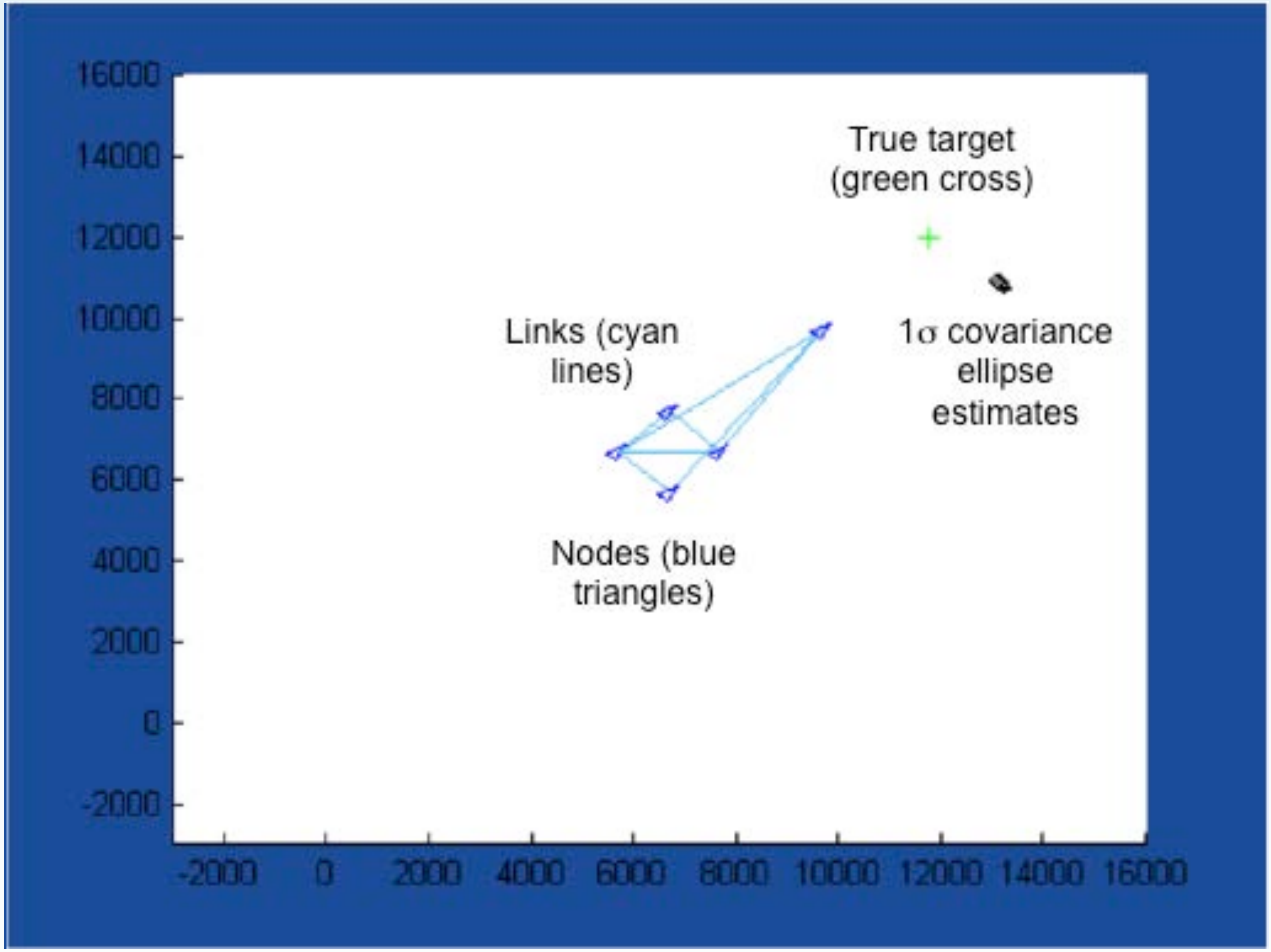
$$f(\mathbf{X}_k|\mathbf{Z}^A) = f(\mathbf{X}_k|\mathbf{Z}^B)$$

- The estimate should be the same as a "super node" which fuses all of the observations centrally

$$f(\mathbf{X}_k|\mathbf{Z}^A) = f(\mathbf{X}_k|\mathbf{z}_{1:k}^A \cup \mathbf{z}_{1:k}^B)$$

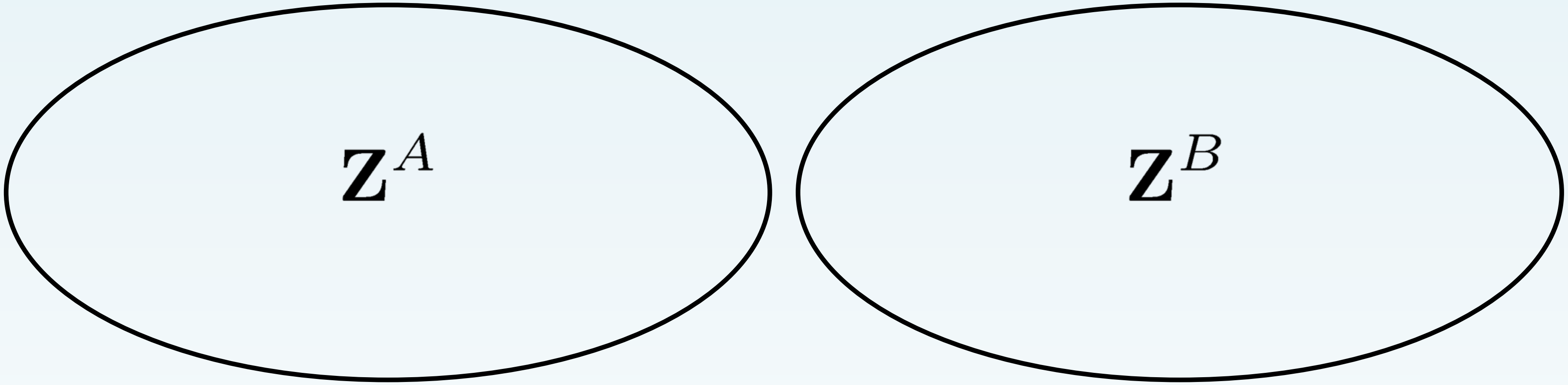
Simple Strategy

- Why don’t we treat the fused estimates from one node to be observations which we can feed directly into the other node?



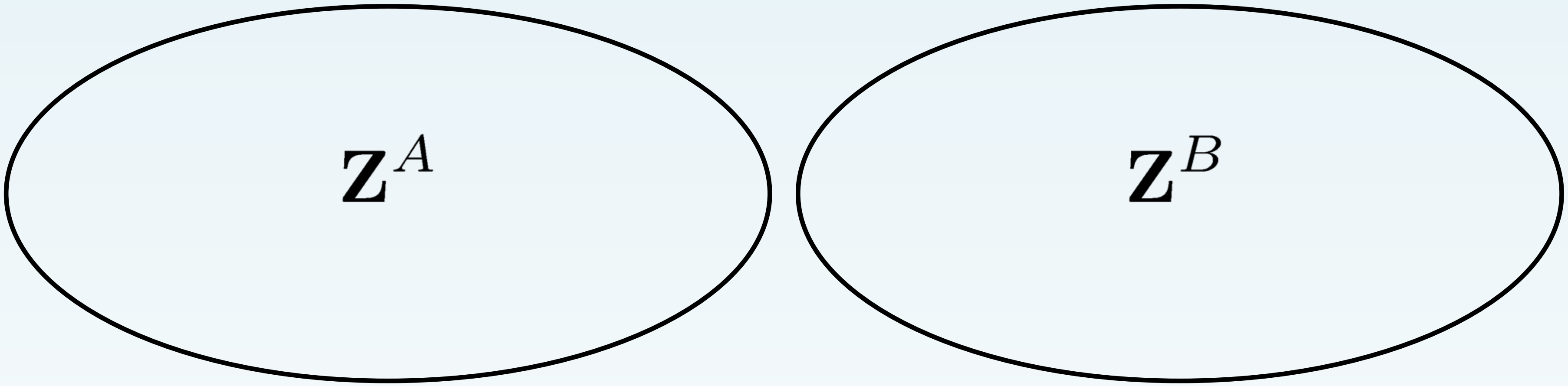
Courtesy D. Nicholson, BAE Systems

Probabilistic / Information Set Representation



$$f\left(\mathbf{X} \mid \mathbf{Z}^A \cup \mathbf{Z}^B\right) = \frac{f\left(\mathbf{Z}^A \cup \mathbf{Z}^B \mid \mathbf{X}\right) f\left(\mathbf{X}\right)}{f\left(\mathbf{Z}^A \cup \mathbf{Z}^B\right)}$$

Conditionally Independent Case



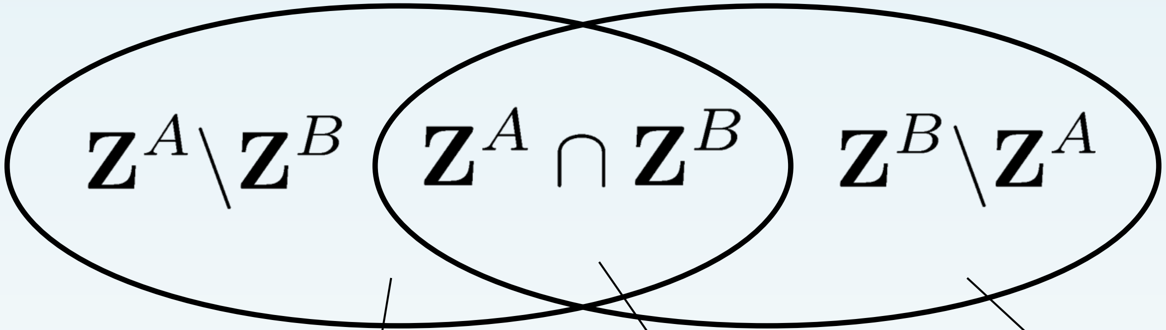
$$f\left(\mathbf{Z}^A \cup \mathbf{Z}^B \mid \mathbf{X}\right) \propto f\left(\mathbf{Z}^A \mid \mathbf{X}\right) \times f\left(\mathbf{Z}^B \mid \mathbf{X}\right)$$

Common Information

- The state information stored in each node is *not* independent of the information in other nodes
  - Common process noise
    - Occurs whether or not nodes have exchanged information
  - Common measurement history
    - Occurs when nodes exchange information

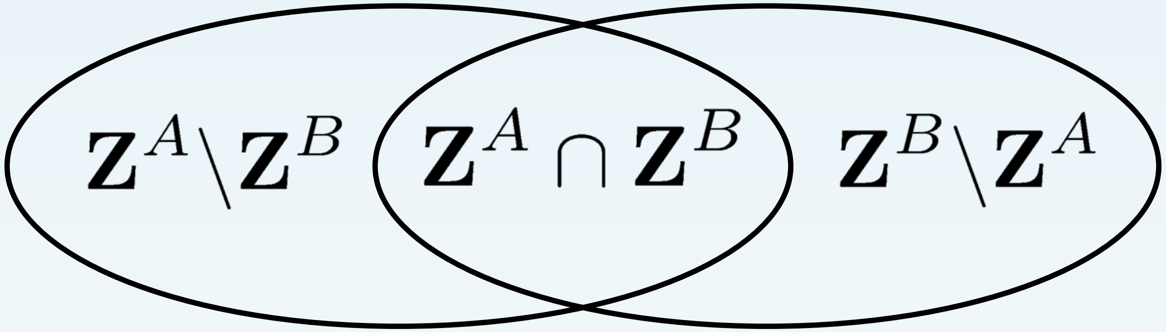
- The effect can be illustrated by considering fusion of data from two different nodes

Fusion of *Dependent* Information Sets



$$f\left(\mathbf{Z}^A\cup\mathbf{Z}^B|\mathbf{X}\right)\propto\underbrace{f\left(\mathbf{Z}^B\backslash\mathbf{Z}^A|\mathbf{X}\right)}_{\text{New information}}\times\underbrace{f\left(\mathbf{Z}^A\cap\mathbf{Z}^B|\mathbf{X}\right)}_{\text{Common information}}\times\underbrace{f\left(\mathbf{Z}^A\backslash\mathbf{Z}^B|\mathbf{X}\right)}_{\text{New information}}$$

Assuming Conditional Independence



$$f\left(\mathbf{Z}^A \cup \mathbf{Z}^B \mid \mathbf{X}\right) \propto f\left(\mathbf{Z}^A \mid \mathbf{X}\right) \times f\left(\mathbf{Z}^B \mid \mathbf{X}\right)$$

$$f\left(\mathbf{Z}^A \cup \mathbf{Z}^B | \mathbf{X}\right) \propto f\left(\mathbf{Z}^A \backslash \mathbf{Z}^B | \mathbf{X}\right) \times \underbrace{f\left(\mathbf{Z}^A \cap \mathbf{Z}^B | \mathbf{X}\right)^2}_{\text{Double counted term}} \times f\left(\mathbf{Z}^B \backslash \mathbf{Z}^A | \mathbf{X}\right)$$

Inconsistency in Kalman Filters

- Consider the state of the *entire* DDF network
- The state vector is

$$\mathbf{x}_{NET}(k) = \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \\ \vdots \\ \mathbf{x}_N(k) \end{bmatrix}$$

- The network estimate is

$$\hat{\mathbf{x}}_{NET}(k \mid k) = \begin{bmatrix} \hat{\mathbf{x}}_1(k \mid k) \\ \hat{\mathbf{x}}_2(k \mid k) \\ \vdots \\ \hat{\mathbf{x}}_N(k \mid k) \end{bmatrix}, \quad \mathbf{P}_{NET}(k \mid k) = \begin{bmatrix} \mathbf{P}_{11}(k \mid k) & \mathbf{P}_{12}(k \mid k) & \dots & \mathbf{P}_{1N}(k \mid k) \\ \mathbf{P}_{21}(k \mid k) & \mathbf{P}_{22}(k \mid k) & \dots & \mathbf{P}_{1N}(k \mid k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{N1}(k \mid k) & \mathbf{P}_{N2}(k \mid k) & \dots & \mathbf{P}_{NN}(k \mid k) \end{bmatrix}$$

Failure Due to Inconsistent Approximation

- However, assuming the estimates are independent is equivalent to using the approximate network estimate

$$\hat{\mathbf{x}}^*_{NET}(k \mid k) = \begin{bmatrix} \hat{\mathbf{x}}_1(k \mid k) \\ \hat{\mathbf{x}}_2(k \mid k) \\ \vdots \\ \hat{\mathbf{x}}_N(k \mid k) \end{bmatrix}, \quad \mathbf{P}^*_{NET}(k \mid k) = \begin{bmatrix} \mathbf{P}_{11}(k \mid k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{22}(k \mid k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_{NN}(k \mid k) \end{bmatrix}$$

- The error in this approximation is

$$\begin{aligned} \Delta\mathbf{P}_{NET}(k \mid k) &= \mathbf{P}^*_{NET}(k \mid k) - \mathbf{P}_{NET}(k \mid k) \\ &= \begin{bmatrix} \mathbf{0} & -\mathbf{P}_{12}(k \mid k) & \dots & -\mathbf{P}_{1N}(k \mid k) \\ -\mathbf{P}_{21}(k \mid k) & \mathbf{0} & \dots & -\mathbf{P}_{1N}(k \mid k) \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{P}_{N1}(k \mid k) & -\mathbf{P}_{N2}(k \mid k) & \dots & \mathbf{0} \end{bmatrix} \end{aligned}$$

Overcoming Double Counting

- Recall that the problematic term is

$$f(\mathbf{Z}^A|\mathbf{X})f(\mathbf{Z}^B|\mathbf{X}) \propto f(\mathbf{Z}^A/\mathbf{Z}^B|\mathbf{X})f(\mathbf{Z}^A \cup \mathbf{Z}^B|\mathbf{X})^2 f(\mathbf{Z}^B \cup \mathbf{Z}^A|\mathbf{X})$$

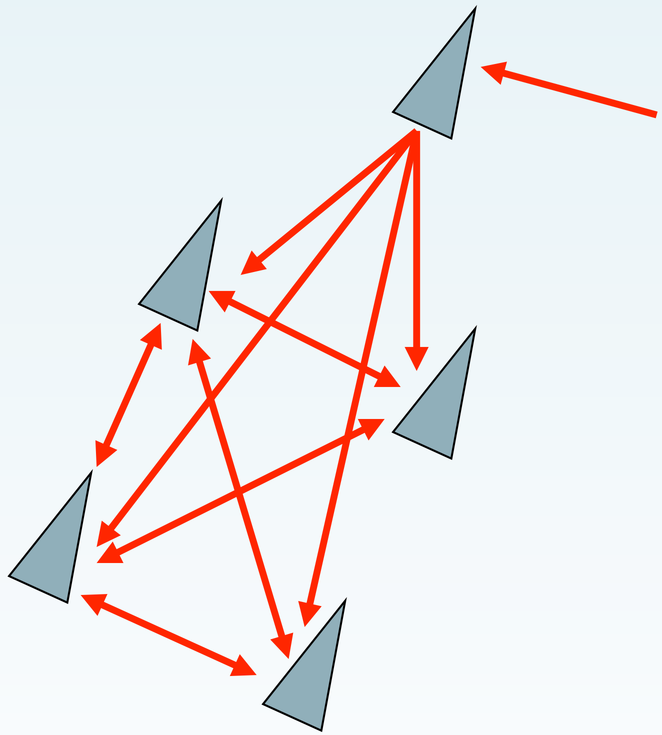
- Chong and Mori showed that right expression "cancels out" the common information

$$f\left(\mathbf{Z}^A \cup \mathbf{Z}^B|\mathbf{X}\right) \propto \frac{f\left(\mathbf{Z}^B|\mathbf{X}\right) f\left(\mathbf{Z}^A|\mathbf{X}\right)}{f\left(\mathbf{Z}^A \cap \mathbf{Z}^B|\mathbf{X}\right)}$$

Cancel out common information

- The common information can only be computed with *special network topologies*

Approach 1: Distribute Observations



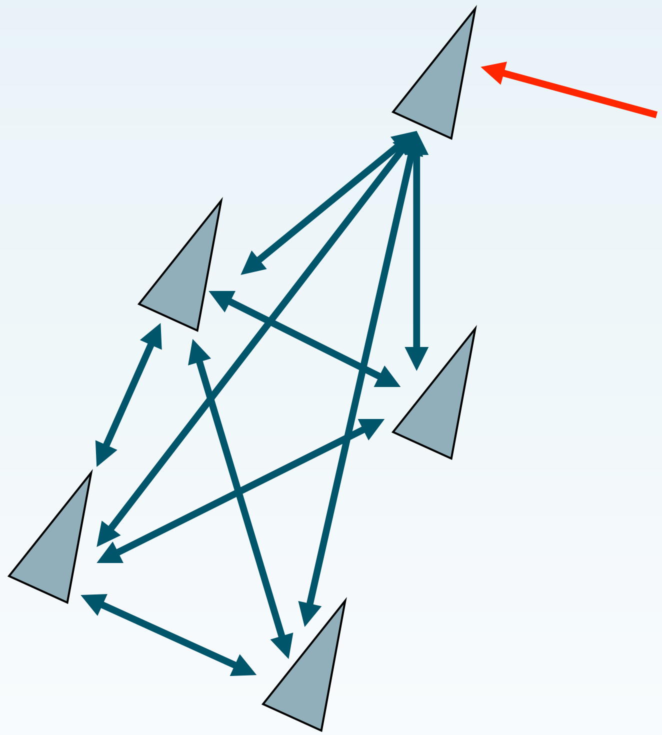
- Broadcast *all* observations to *all* nodes

Pros and Cons

- Advantages:
  - Each node has optimal estimate for all time
  - Distribution provides no additional complexity to fusion algorithm
  - Actually used in practice

- Disadvantages:
  - Requires all nodes to have the same communication and computational abilities
  - Requires extremely large bandwidth
  - Introduces implicit assumption that all nodes have *exactly* the same estimate (=all the links have to work all of the time)

Approach 2: Fully-Connected Network



- Broadcast *all* updated state estimates to *all* nodes

Fully-Connected Networks

- The easiest way to implement a fully connected network is to use the inverse covariance (or information) form of the Kalman Filter
- The state space is replaced by the *information variables*

$$\hat{\mathbf{y}}(k \mid k) = \mathbf{P}^{-1}(k \mid k)\,\hat{\mathbf{x}}(k \mid k)$$

$$\hat{\mathbf{Y}}(k \mid k) = \mathbf{P}^{-1}(k \mid k)$$

Updating in Information Form

- Using information form, the update simplifies to

$$\hat{\mathbf{y}}_n(k \mid k) = \hat{\mathbf{y}}_n(k \mid k-1) + \mathbf{i}_n(k)$$
$$\hat{\mathbf{Y}}_n(k \mid k) = \hat{\mathbf{Y}}_n(k \mid k-1) + \mathbf{I}_n(k)$$

where the information from the observations is

$$\mathbf{i}_n(k) = \mathbf{H}_n^T \mathbf{R}_n^{-1}(k)\, \mathbf{z}_n(k)$$
$$\mathbf{I}_n(k) = \mathbf{H}_n^T \mathbf{R}_n^{-1}(k)\, \mathbf{H}_n$$

Distributed Information Updates

- Since the information from the observations is independent of the state, $\mathbf{i}_n$ and $\mathbf{I}_n$ are independent of previous state estimates and can be safely distributed
- The update rule simply becomes

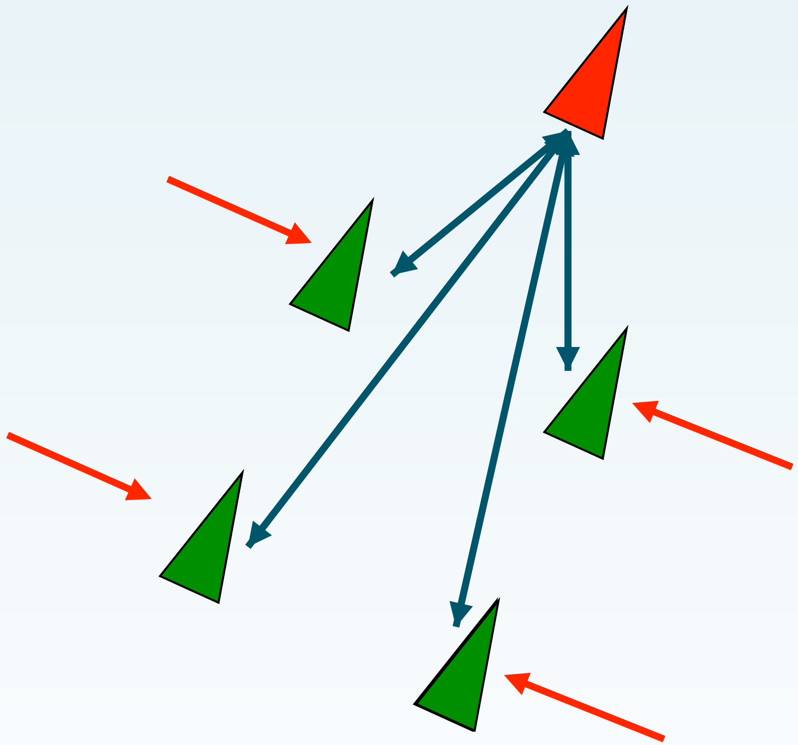
$$\hat{\mathbf{y}}_n(k \mid k) = \hat{\mathbf{y}}_n(k \mid k-1) + \sum_{n=1}^{N} \mathbf{i}_n(k)$$

$$\hat{\mathbf{Y}}_n(k \mid k) = \hat{\mathbf{Y}}_n(k \mid k-1) + \sum_{n=1}^{N} \mathbf{I}_n(k)$$

Fully-Connected Network

- Advantages:
  - Each node has optimal estimate for all time
  - Broadcasting the observation information variables potentially saves bandwidth
- Disadvantages:
  - Requires all nodes to have the same communication and computational abilities
  - Still requires $O(N^2)$ communication links
  - Introduces explicit assumption that all nodes have exactly the same estimate (important if linearising e.g., with an EKF)

Approach 3: Hierarchical Network



- Network has “master” and “slave” nodes
  - Slaves fuse data locally
  - Estimates sent to master which fuses them together
  - Revised estimate broadcast back to slaves

Fusion in the Slave

- The slave updates using the information Kalman filter equations:

$$\hat{\mathbf{y}}_n\left(k \mid k\right) = \hat{\mathbf{y}}_n\left(k \mid k-1\right) + \mathbf{i}_n\left(k\right)$$

$$\hat{\mathbf{Y}}_n\left(k \mid k\right) = \hat{\mathbf{Y}}_n\left(k \mid k-1\right) + \mathbf{I}_n\left(k\right)$$

Fusion in the Master

- The master updates by summing the information from all the slaves
- To compensate for the prediction which was sent out, the master must *subtract out* common information,

$$\hat{\mathbf{y}}_M(k \mid k) = \hat{\mathbf{y}}_M(k \mid k-1) + \sum_{n=1}^{N} \Big(\hat{\mathbf{y}}_n(k \mid k) - \hat{\mathbf{y}}_M(k \mid k-1) \Big)$$

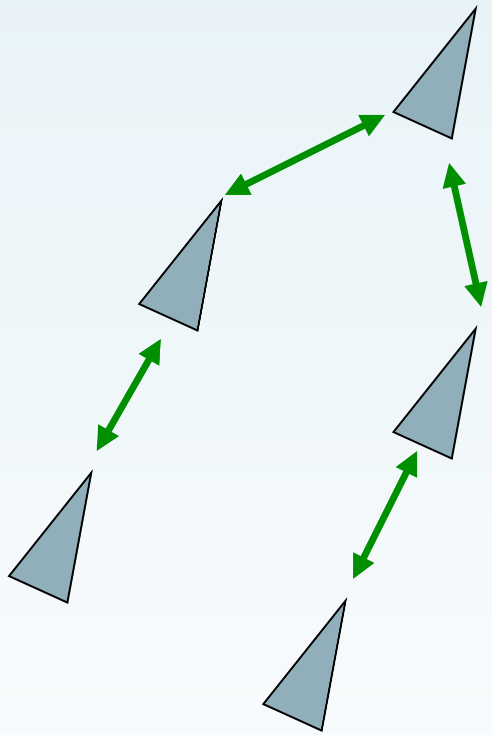
$$\hat{\mathbf{Y}}_M(k \mid k) = \hat{\mathbf{Y}}_M(k \mid k-1) + \sum_{n=1}^{N} \Big(\hat{\mathbf{Y}}_n(k \mid k) - \hat{\mathbf{Y}}_M(k \mid k-1) \Big)$$

Hierarchical Network

- Advantages:
  - Each node has optimal estimate for all time
  - The number of communication links is $O(N)$

- Disadvantages:
  - Additional latency
  - One node is privileged; failure of that node causes the whole network to fail

Approach 4: Channel Filters

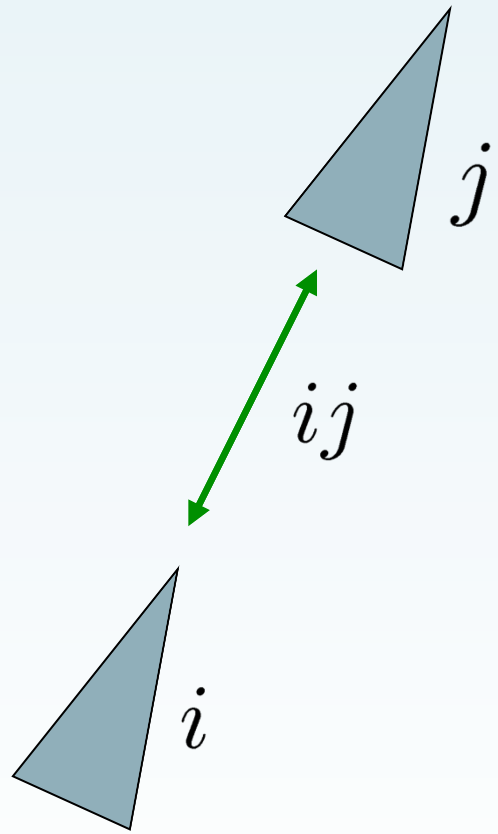


- Constrain the network to be a *tree*
  - Single path between any pair of nodes
- Use “channel filters” to subtract off common information

Estimating Common Information

- Consider a link between a pair of nodes i and j
- The channel filter maintains common information across the link
- It has its own information estimate,

$$\left\{ \hat{\mathbf{y}}_{ij}\left(k \mid k\right),\ \hat{\mathbf{Y}}_{ij}\left(k \mid k\right) \right\}$$



Updating Local Nodes

- The Channel Filter is a regular Kalman Filter but works with the information exchanged between i and j rather than the observation data directly
- First, let the update at filter i using the local sensor observations be written as

$$\tilde{\mathbf{y}}_i(k \mid k) = \hat{\mathbf{y}}_i(k \mid k-1) + \mathbf{i}_n(k)$$

$$\tilde{\mathbf{Y}}_i(k \mid k) = \hat{\mathbf{Y}}_i(k \mid k-1) + \mathbf{I}_n(k)$$

Fusing With Nearby Nodes

- The updated estimate is given by summing all the independent information from a node's neighbours,

$$\hat{\mathbf{y}}_i(k \mid k) = \tilde{\mathbf{y}}_i(k \mid k) + \sum_{j \in N(i)} \{\tilde{\mathbf{y}}_j(k \mid k) - \hat{\mathbf{y}}_{ij}(k \mid k)\}$$

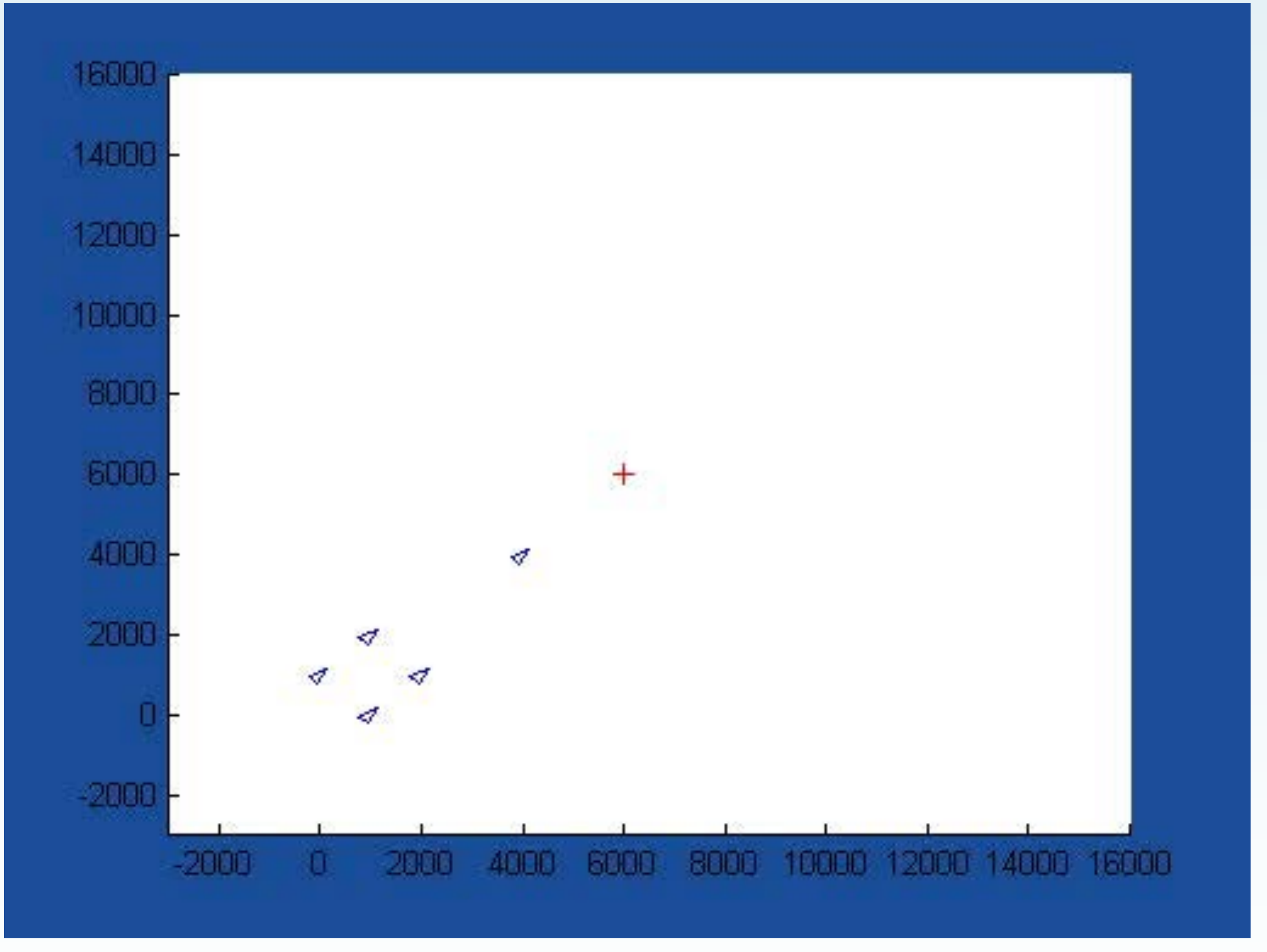
$$\hat{\mathbf{Y}}_i(k \mid k) = \tilde{\mathbf{Y}}_i(k \mid k) + \sum_{j \in N(i)} \left\{\tilde{\mathbf{Y}}_j(k \mid k) - \hat{\mathbf{Y}}_{ij}(k \mid k)\right\}$$

Updating the Channel Filters

- The channel filter update is given by recursively updating with the difference in information variables from the two nodes,

$$\hat{\mathbf{y}}_{ij}(k \mid k) = \tilde{\mathbf{y}}_i(k \mid k) + \tilde{\mathbf{y}}_j(k \mid k) - \hat{\mathbf{y}}_{ij}(k \mid k-1)$$

$$\hat{\mathbf{Y}}_{ij}(k \mid k) = \tilde{\mathbf{Y}}_i(k \mid k) + \tilde{\mathbf{Y}}_j(k \mid k) - \hat{\mathbf{Y}}_{ij}(k \mid k-1)$$



Courtesy D. Nicholson, BAE Systems

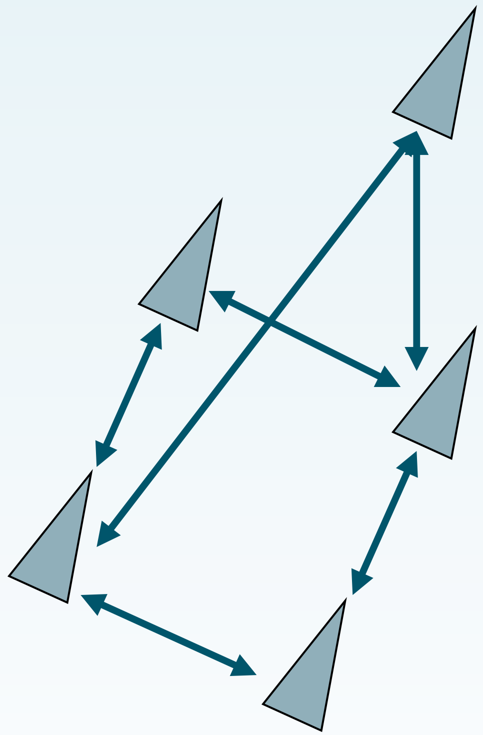
Advantages and Disadvantages

- Advantages:
  - The number of communication links is $O(N)$
  - Optimal in a “time-delayed” sense
- Disadvantages:
  - Estimates at all nodes differ
  - Single path of communication; no redundancy
  - If the network is reconfigured, the channel filters have to be recalculated from scratch

Review of Techniques So Far

- It is possible to develop optimal algorithms for distributed data fusion using local message passing only
- However, these techniques rely on special network topologies:
  - Fully connected
  - Tree-connected
- In general, preserving these topologies can be difficult and undesirable

Adhoc Network



- Arbitrary network with loops and cycles
- Complete flexibility and redundancy

Distributed Data Fusion in Adhoc Networks

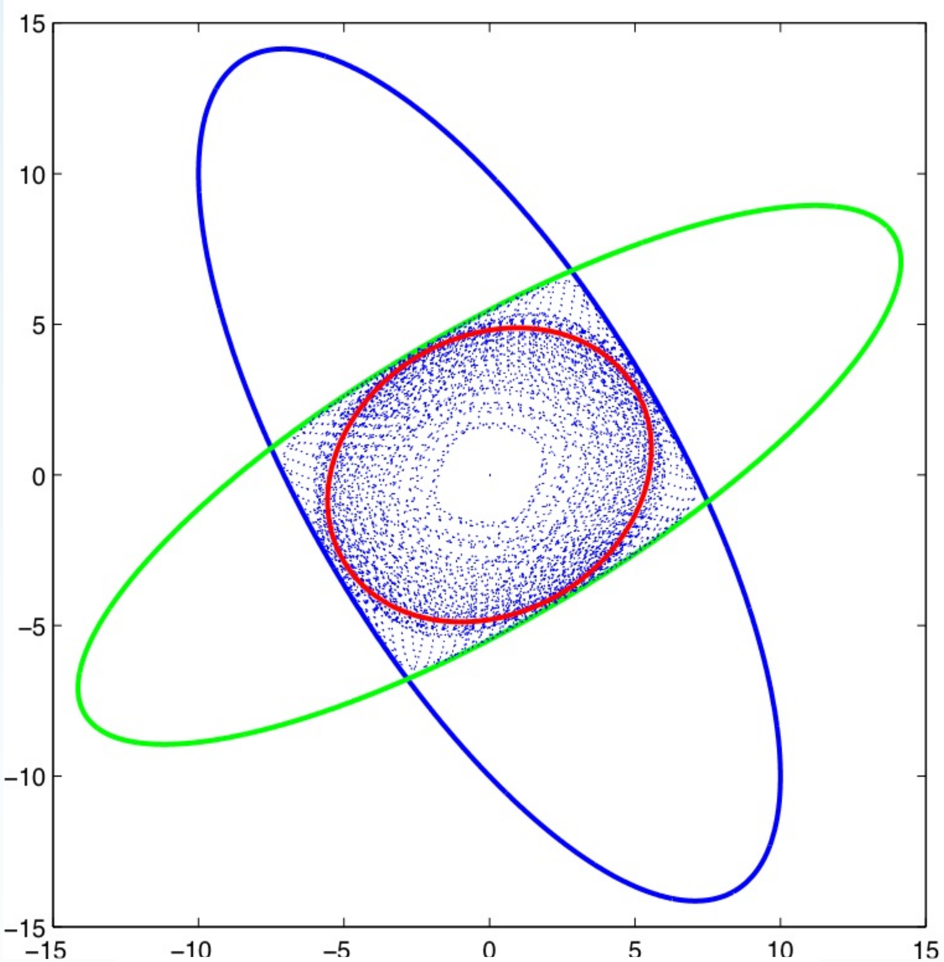
- It has been shown that no local data fusion scheme can be used to develop consistent, *optimal* estimates in this situation
- Therefore, it appears that DDF is strongly limited to the case of very particular data fusion architecture

- If we throw optimality out of the window, can we develop tractable approximations instead?

Suboptimal distributed data fusion

- Motivation
- Distributed data fusion
- *Suboptimal distributed data fusion*
- Distributed multi-object tracking with PHD filters

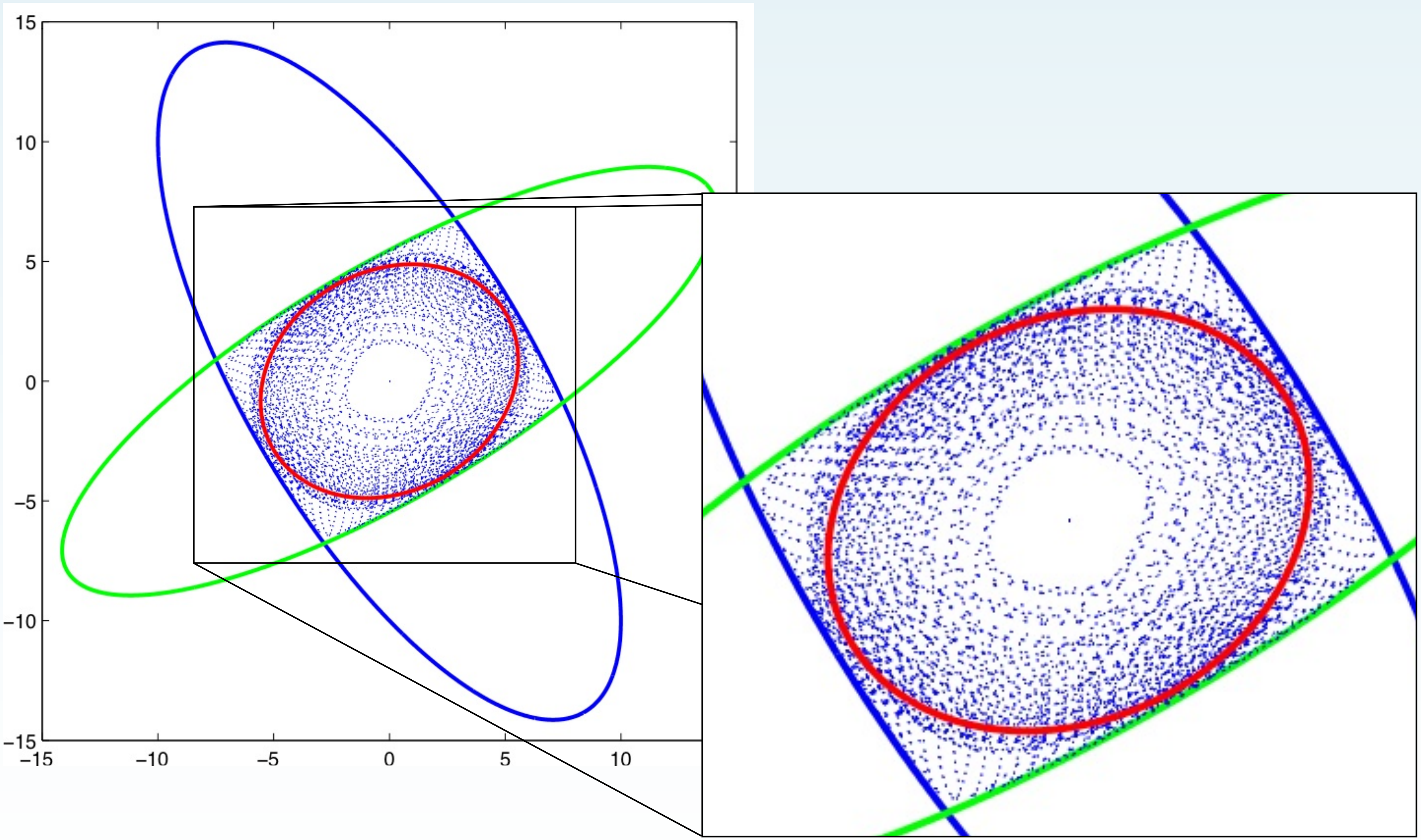
Covariances With *Known* Correlations



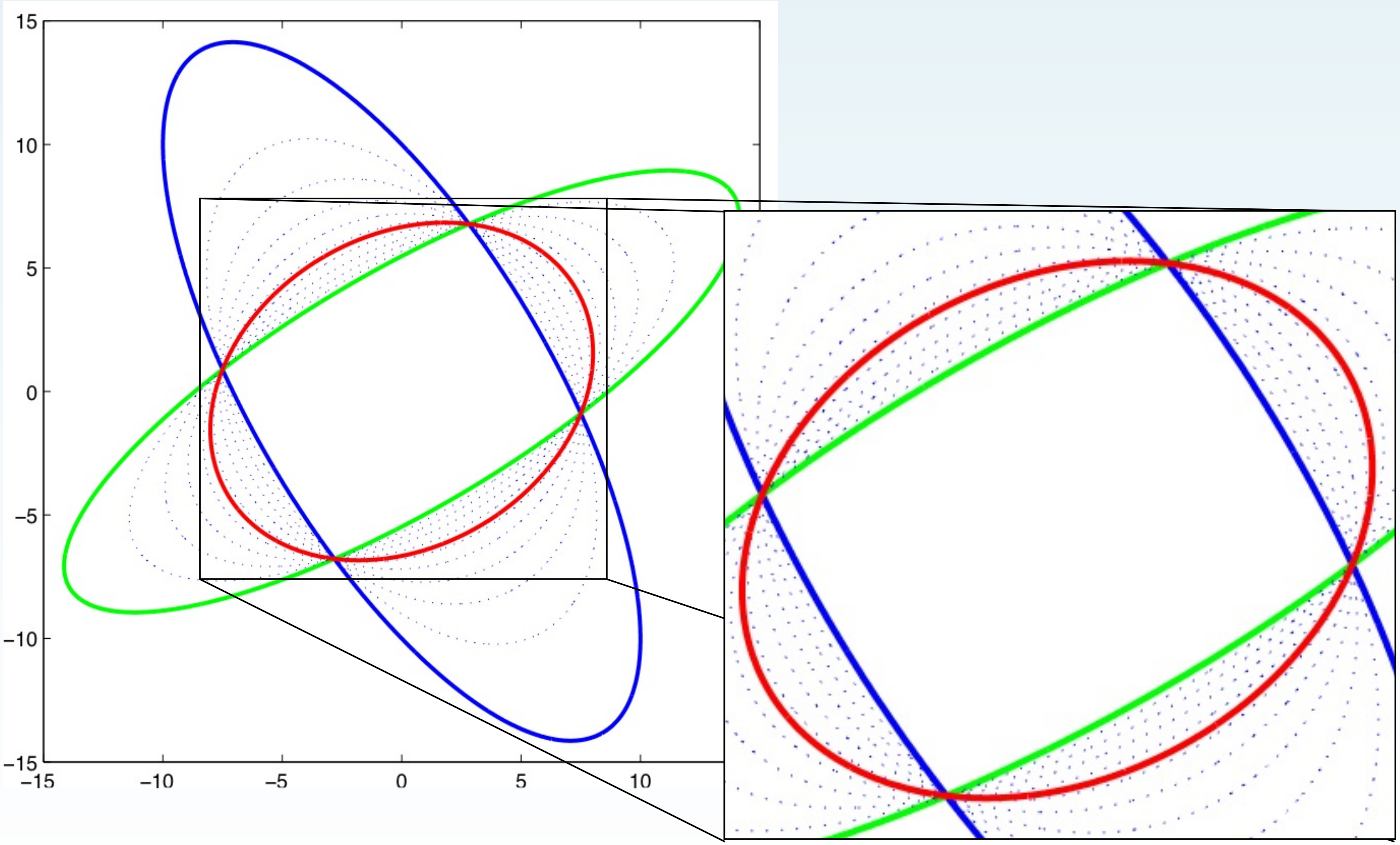
$$\bar{\mathbf{P}}_A^{-1} = \mathbf{P}_A^{-1} + \mathbf{H}_B^T \mathbf{P}_B^{-1} \mathbf{H}_B,$$

$$\bar{\mathbf{P}}_A^{-1} \bar{\mathbf{x}}_A = \mathbf{P}_A^{-1} \hat{\mathbf{x}}_A + \mathbf{H}_B^T \mathbf{P}_B^{-1} \hat{\mathbf{x}}_B$$

Covariances With *Known* Correlations



Covariance Intersection



Parameteristing the Intersection Region

- The update which generates a *family* of ellipses which circumscribe the intersection region is given by

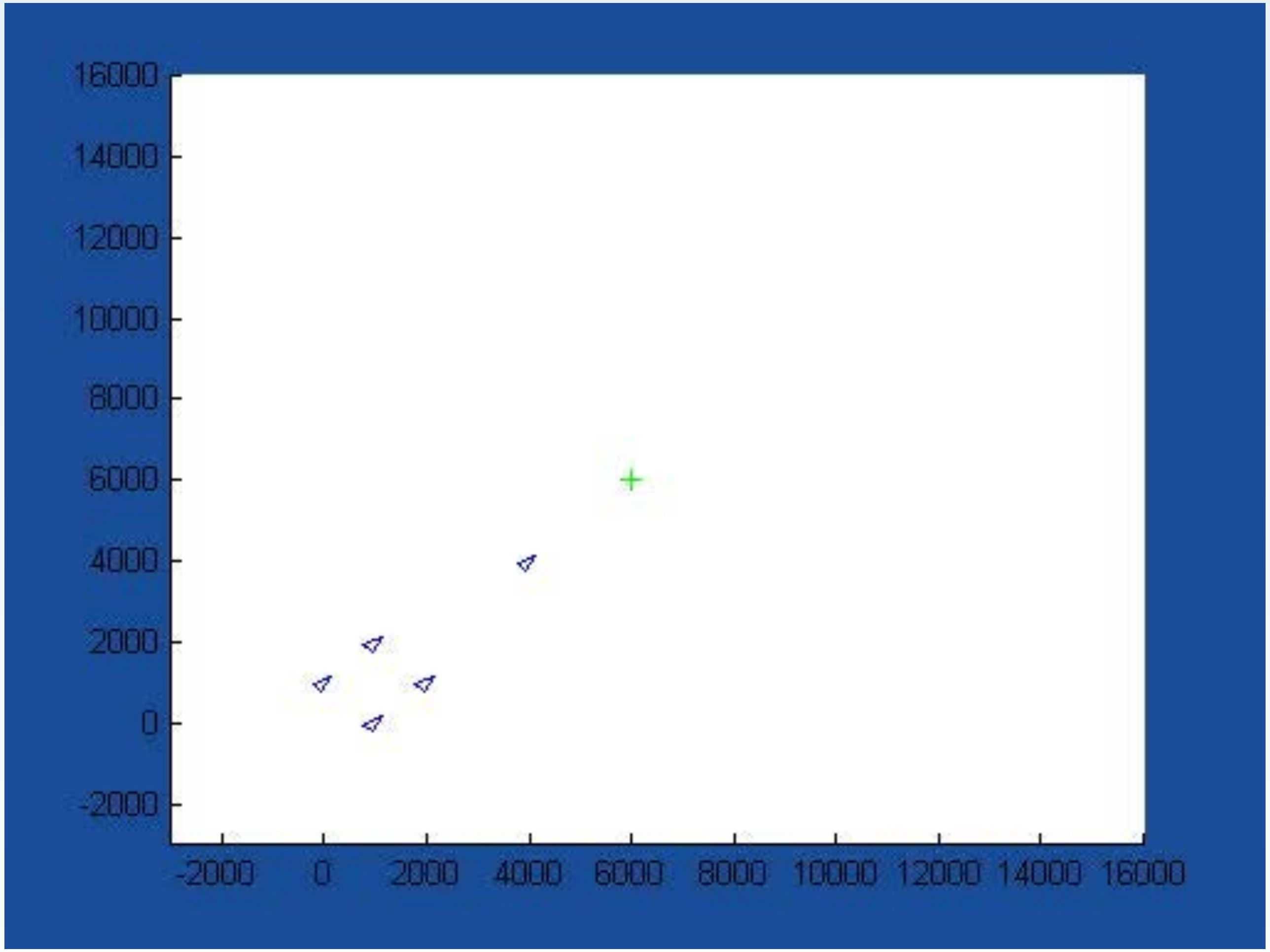
$$\hat{\mathbf{y}}_n(k \mid k) = \omega \hat{\mathbf{y}}_n(k \mid k-1) + (1-\omega)\mathbf{i}_n(k)$$

$$\hat{\mathbf{Y}}_n(k \mid k) = \omega \hat{\mathbf{Y}}_n(k \mid k-1) + (1-\omega)\mathbf{I}_n(k)$$

$$\omega \in [0, 1]$$

- This is the same as a Kalman filter update, but with

$$\left\{ \frac{\mathbf{P}_n(k \mid k-1)}{\omega}, \ \frac{\mathbf{R}_n(k)}{(1-\omega)} \right\}$$



Courtesy D. Nicholson, BAE Systems

Limitations of Covariance Intersection

- CI generates estimate that does not under estimate the mean squared error
- However, the algorithm only understands the first two moments of the distribution
- It cannot exploit other important information (e.g., multimodal, discrete)
- For information of more complicated types, a generalisation of CI is required

Structure of the Fusion Rule

- Suppose we have a fusion rule should be of the form

$$\hat{P}\left(\mathbf{x}|\mathbf{Z}_k^A \cup \mathbf{Z}_k^B\right) \propto \mathcal{F}\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right]$$

where

Function of new information

$$\mathcal{F}\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right] = \mathcal{G}\left[P\left(\mathbf{Z}_k^A \backslash \mathbf{Z}_k^B|\mathbf{x}\right), P\left(\mathbf{Z}_k^B \backslash \mathbf{Z}_k^A|\mathbf{x}\right)\right]$$

$$\times P\left(\mathbf{Z}_k^A \cap \mathbf{Z}_k^B|\mathbf{x}\right) \times P\left(\mathbf{x}\right)$$

Common information single-counted

Robust Fusion Rules

- There are at least *two* classes of rules which satisfy these requirements:
  - *Weighted means* of probability distributions
  - *Weighted geometric means* of probability distributions

- The weighted geometric mean produces better results than the weight mean, and so we focus on it for the rest of the talk

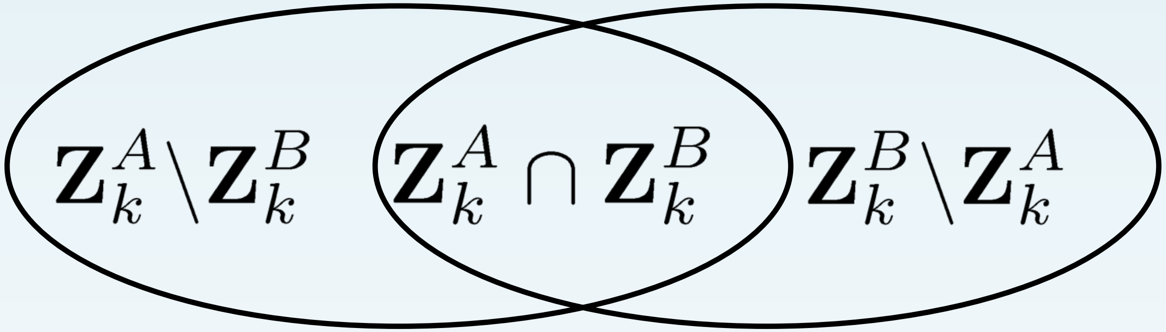
Weighted Geometric Mean (WGM)

- The update is computed from:

$$\hat{P}\left(\mathbf{x}|\mathbf{Z}_k^A \cup \mathbf{Z}_k^B\right) \propto P\left(\mathbf{x}|\mathbf{Z}_k^A\right)^{\omega} P\left(\mathbf{x}|\mathbf{Z}_k^B\right)^{1-\omega}$$

- Despite it’s apparently arbitrary nature, this form crops up in lots of places:
  - *Covariance intersection*, if the distributions are Gaussian
  - *Worst case distributions* to compute upper bounds in binary classifier problems (Chernoff Information)
  - *Logarithmic opinion* pools to fuse opinions of experts and classifiers
  - *Alpha divergences* to approximate message passing in belief networks
  - *Power priors* for combining prior information from earlier studies

WGM Does *Not* Double Count



$$\hat{P}\left(\mathbf{x}|\mathbf{Z}_k^A \cup \mathbf{Z}_k^B\right) \propto P\left(\mathbf{Z}_k^B \backslash \mathbf{Z}_k^A|\mathbf{x}\right)^{\omega} \times \underbrace{P\left(\mathbf{Z}_k^A \cap \mathbf{Z}_k^B|\mathbf{x}\right)}_{\textit{Single}\ \text{counted term}} \times P\left(\mathbf{Z}_k^A \backslash \mathbf{Z}_k^B|\mathbf{x}\right)^{1-\omega} \times P(\mathbf{x})$$

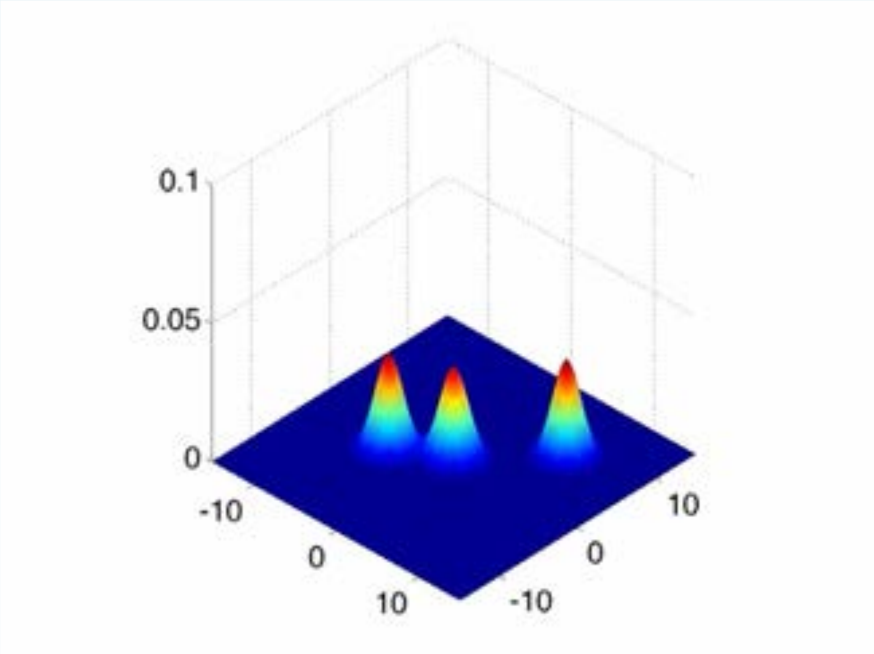
Information Losses and Gains

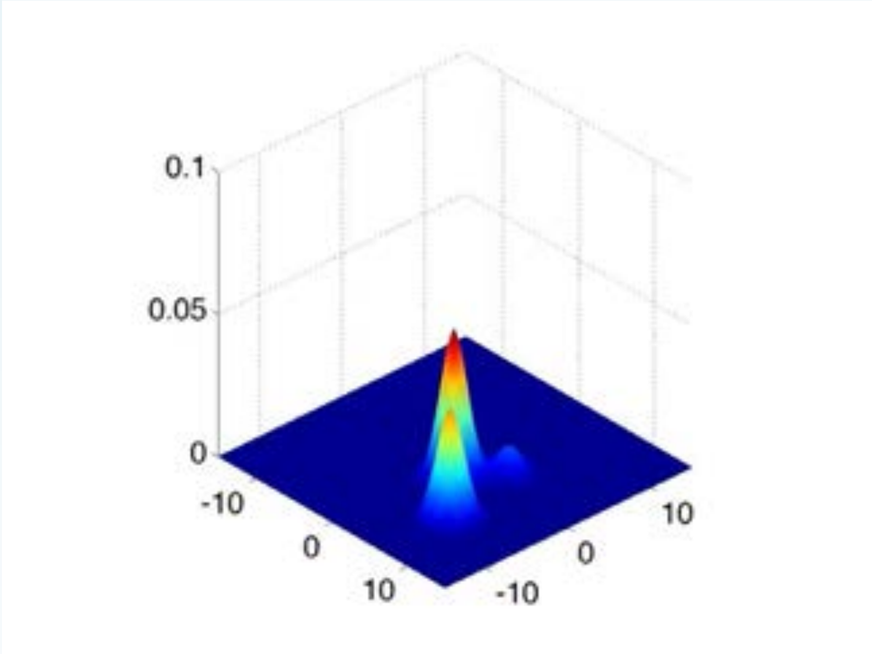
- Therefore, we now need to ask what is the effect of

$$\mathcal{G}\left[P\left(\mathbf{Z}_k^A \backslash \mathbf{Z}_k^B | \mathbf{x}\right), P\left(\mathbf{Z}_k^B \backslash \mathbf{Z}_k^A | \mathbf{x}\right)\right] = P\left(\mathbf{Z}_k^B \backslash \mathbf{Z}_k^A | \mathbf{x}\right)^{\omega} P\left(\mathbf{Z}_k^A \backslash \mathbf{Z}_k^B | \mathbf{x}\right)^{1-\omega}$$

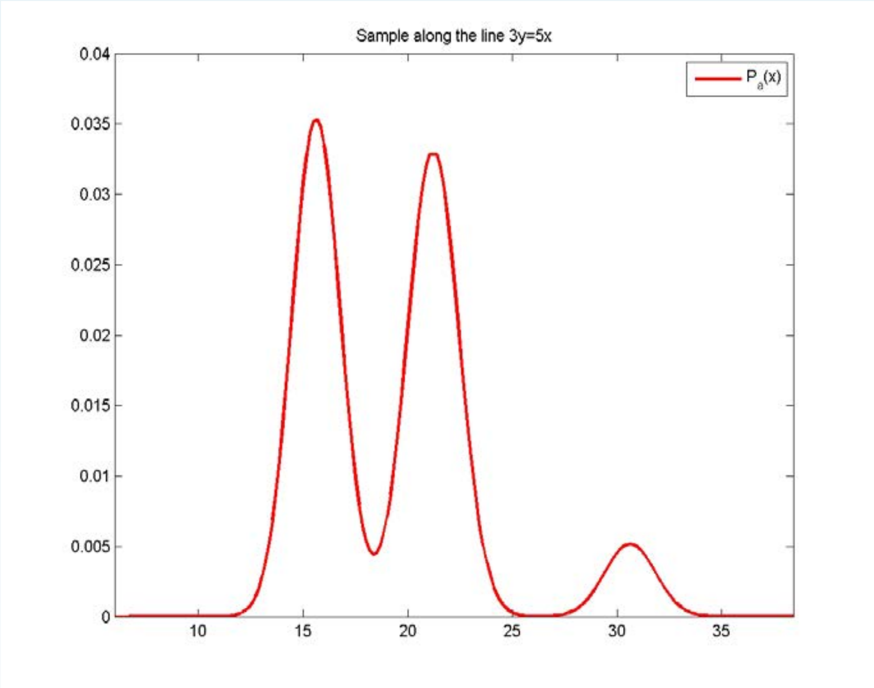
- We can assess this in several ways:
  - By observation
  - Pointwise bounds
  - Information measures
    - Surprisingly *hard*

Example Distributions

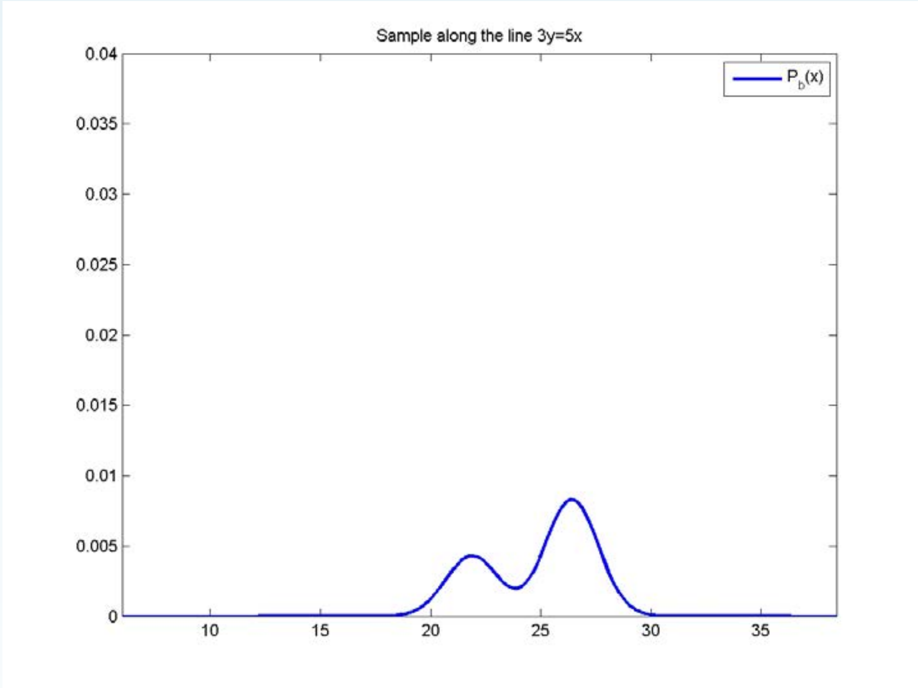




Effect of ω

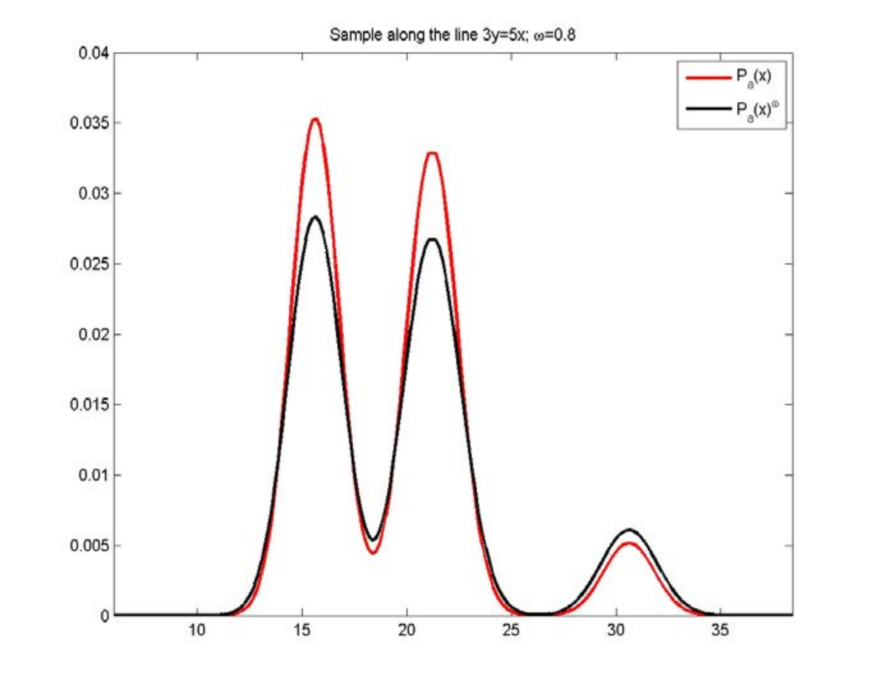


Sample along the line 3y=5x
Pa(x)
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

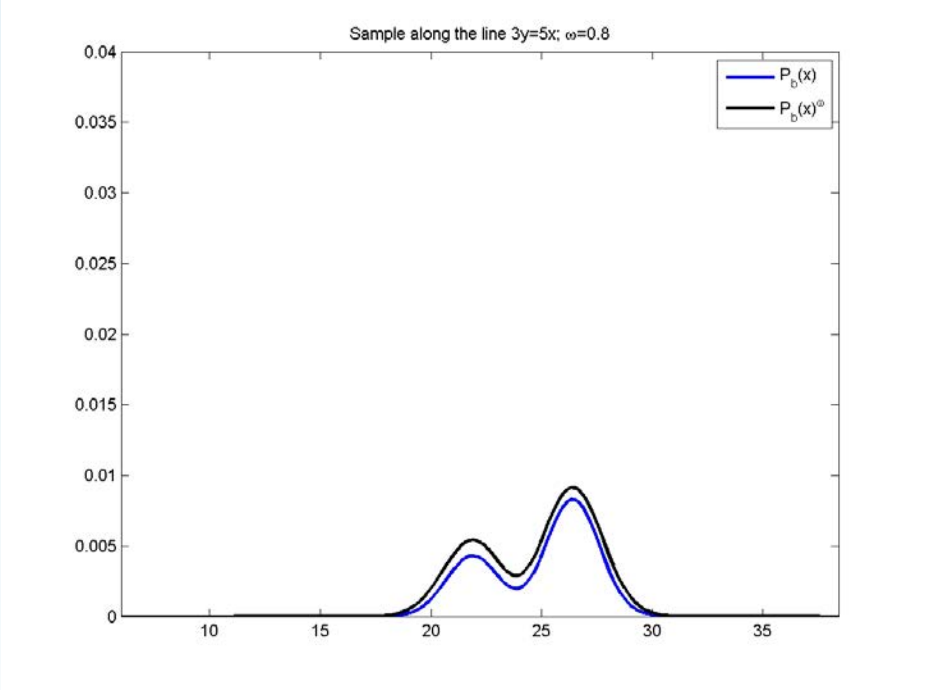


Sample along the line 3y=5x
Pb(x)
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

Effect of ω

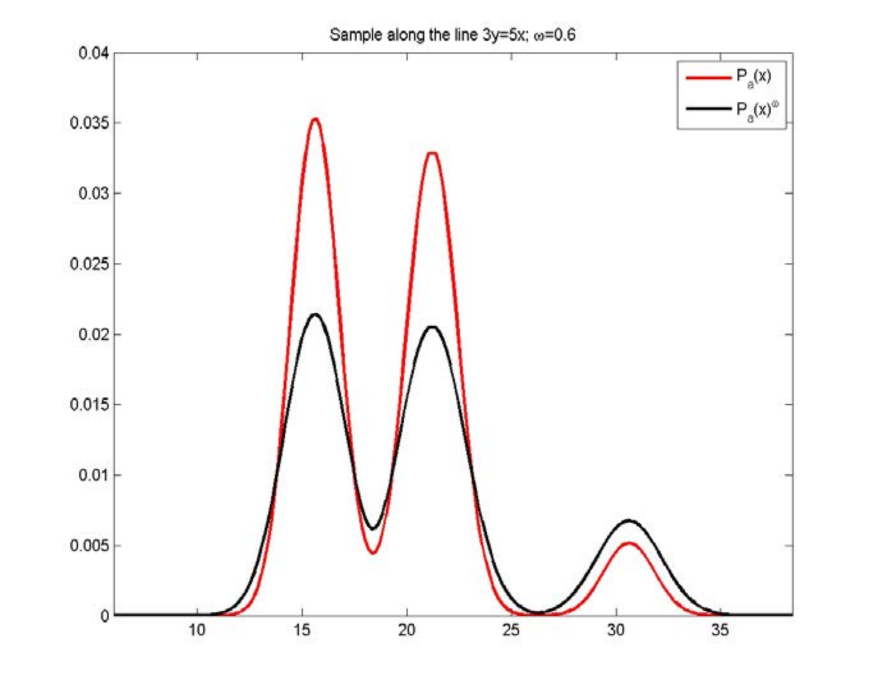


Sample along the line 3y=5x; ω=0.8
Pa(x)
Pa(x)ω
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

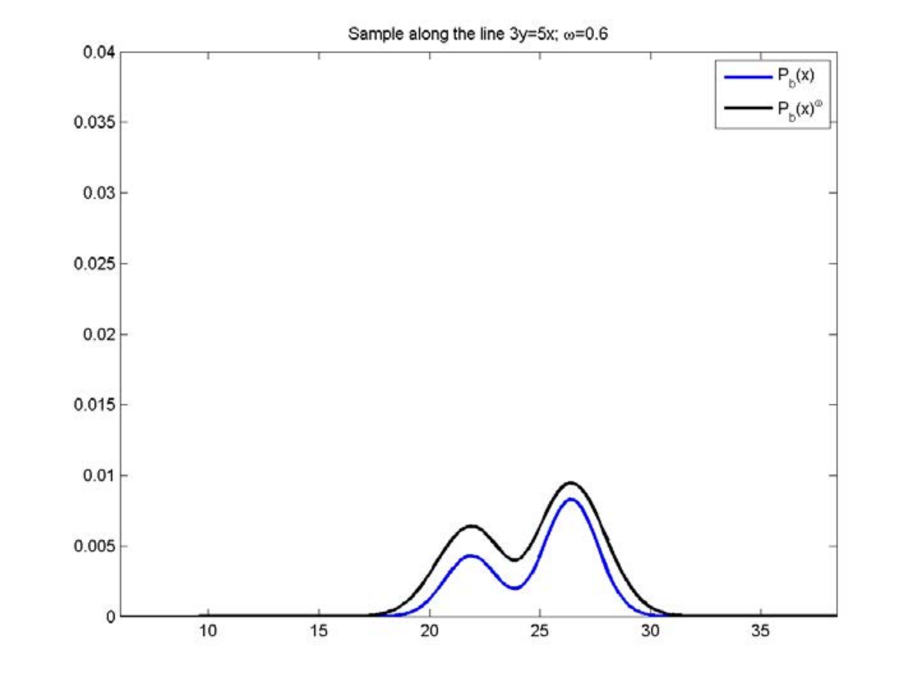


Sample along the line 3y=5x; ω=0.8
Pb(x)
Pb(x)ω
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

Effect of ω

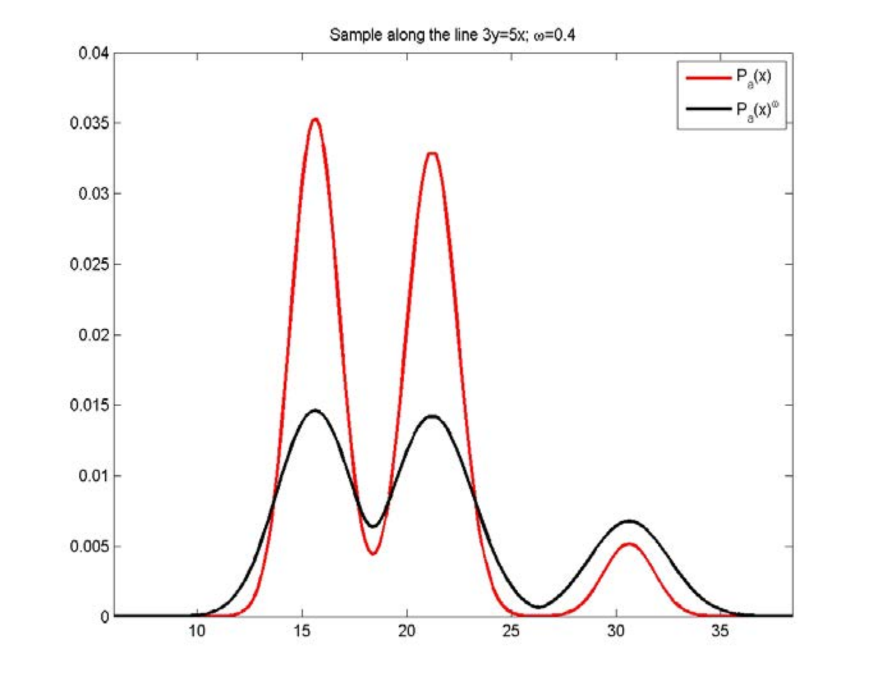


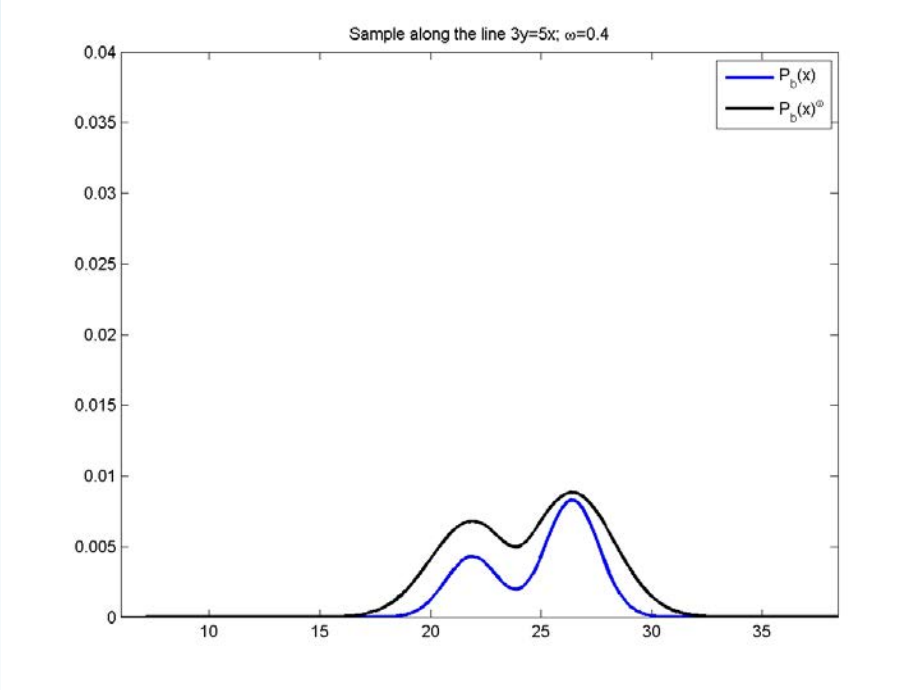
Sample along the line 3y=5x; ω=0.6
$P_a(x)$
$P_a(x)^\omega$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35



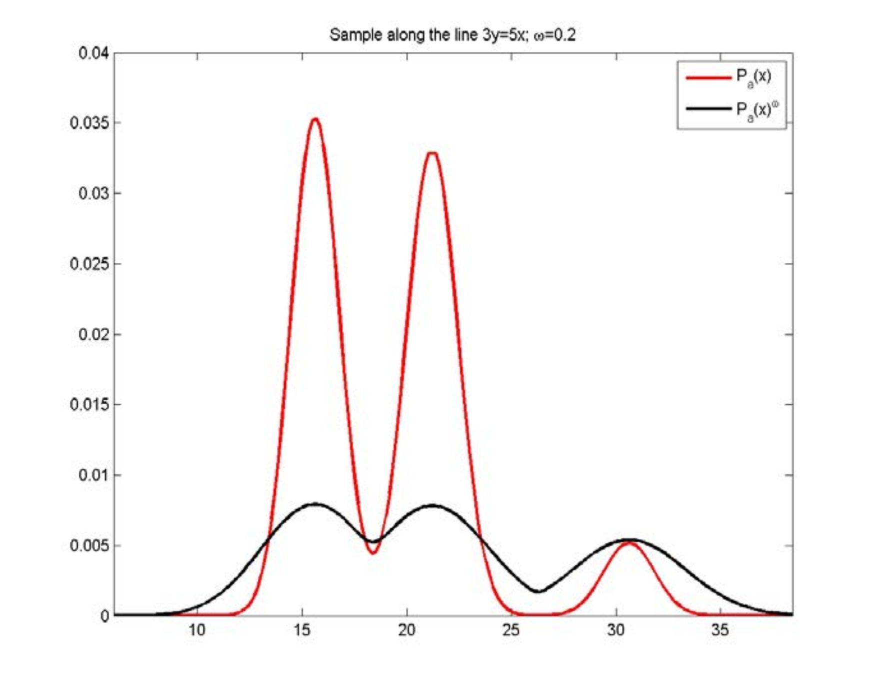
Sample along the line 3y=5x; ω=0.6
$P_b(x)$
$P_b(x)^\omega$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

Effect of ω

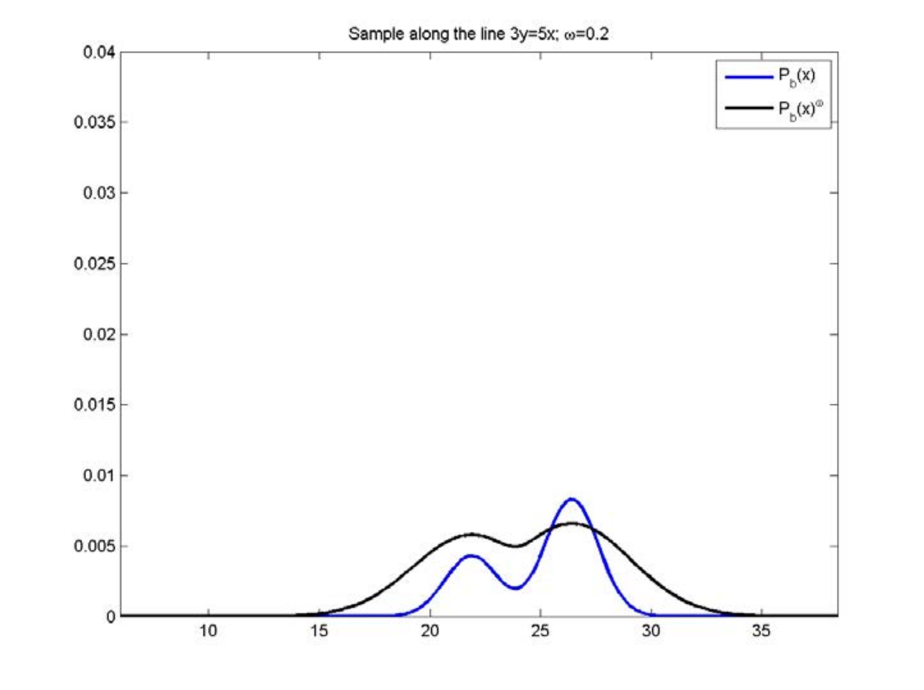

Sample along the line 3y=5x; ω=0.4
$P_a(x)$
$P_a(x)^\omega$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35


Sample along the line 3y=5x; ω=0.4
$P_b(x)$
$P_b(x)^\omega$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

Effect of ω

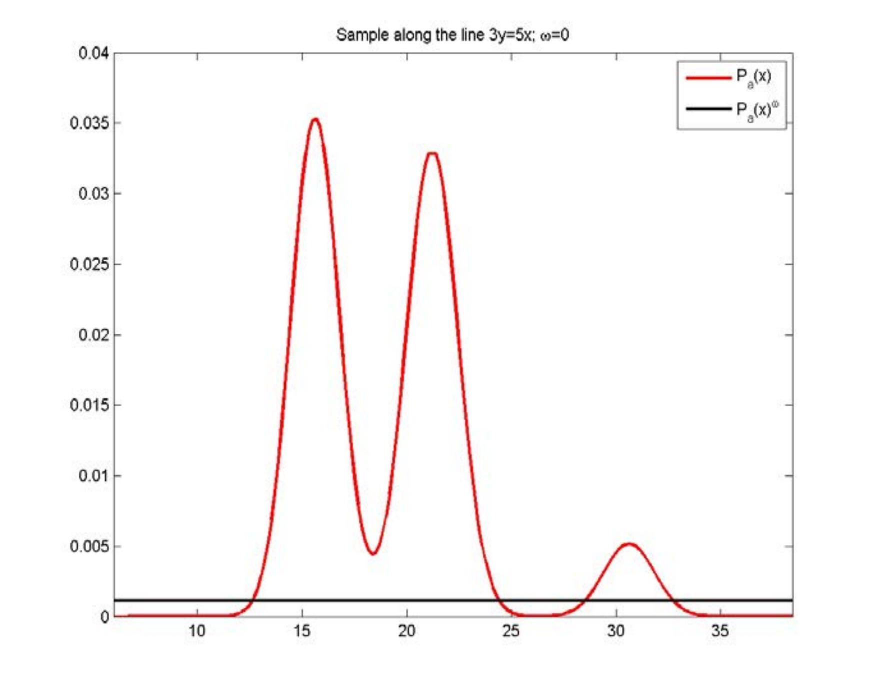


Sample along the line 3y=5x; ω=0.2
P_a(x)
P_a(x)^ω

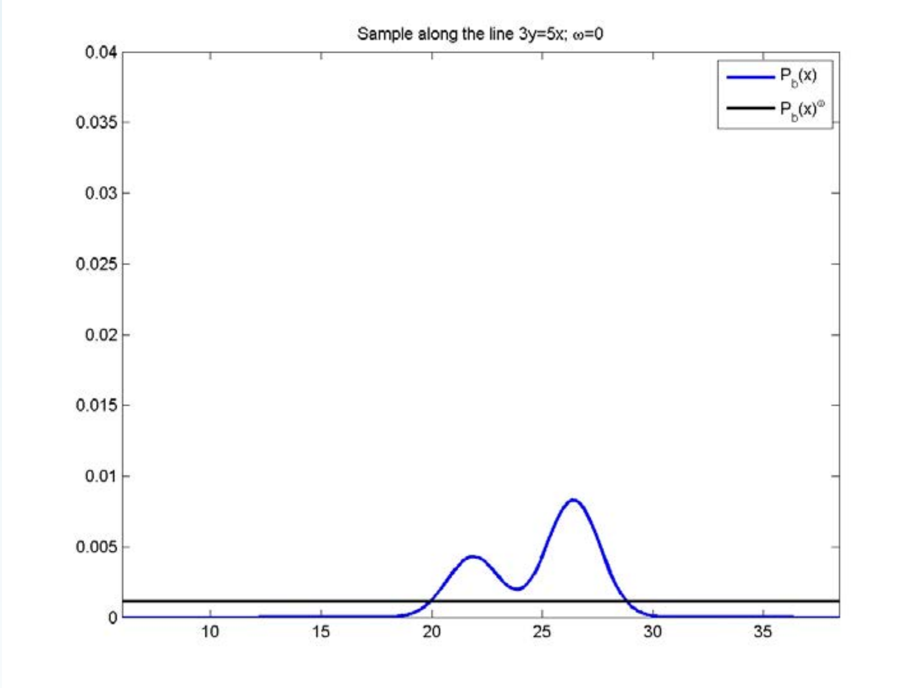


Sample along the line 3y=5x; ω=0.2
P_b(x)
P_b(x)^ω

Effect of ω



Sample along the line 3y=5x; ω=0
$P_a(x)$
$P_a(x)^{\omega}$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35



Sample along the line 3y=5x; ω=0
$P_b(x)$
$P_b(x)^{\omega}$
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
10
15
20
25
30
35

Pointwise Bounds

- It is possible to establish pointwise bounds which apply at each point in the distribution
- Although pointwise bounds play no special role in Bayesian statistics, they provide some insight into the behaviour of the fusion rule

Bounds for the *Unnormalised* Distribution

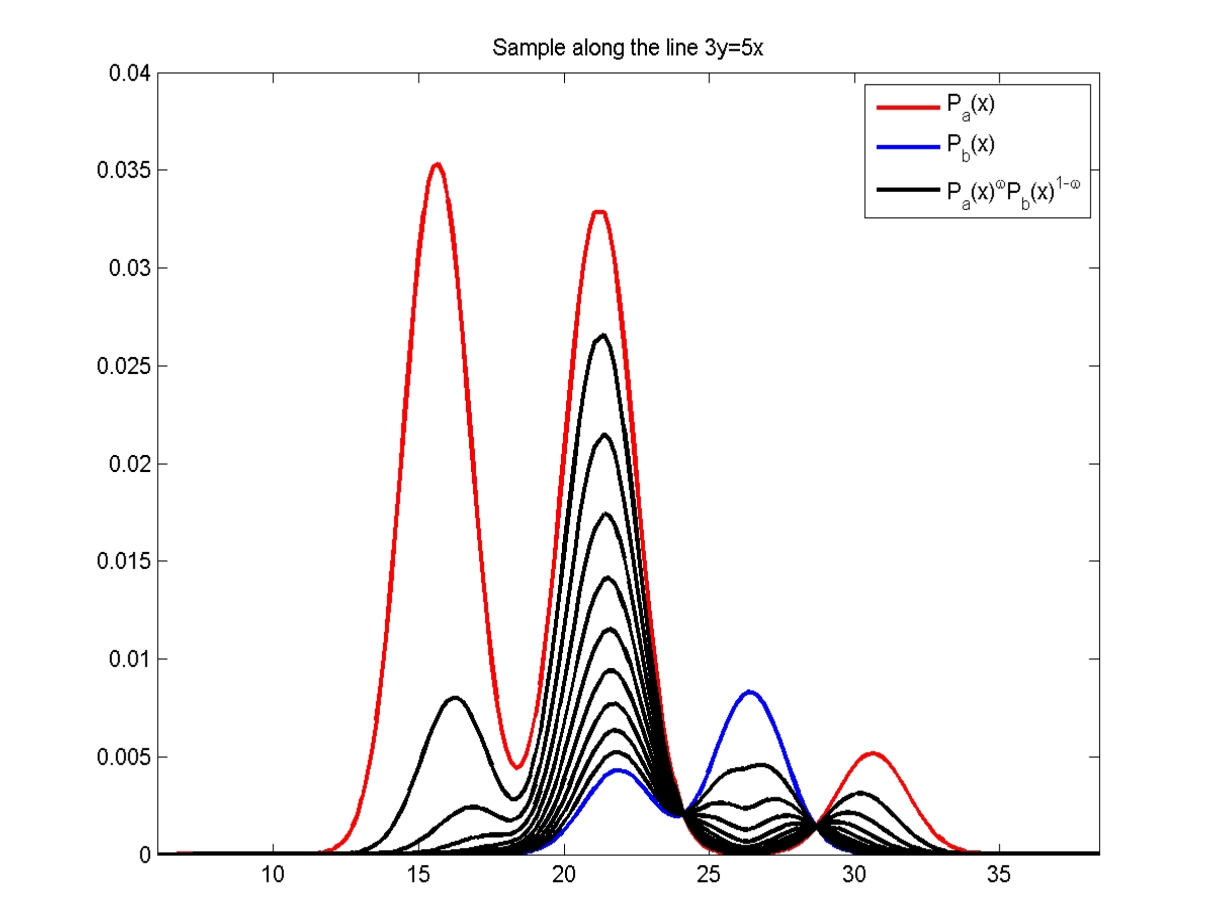
- Let

$$\bar{P} = P\left(\mathbf{x}|\mathbf{Z}_k^A\right)^{\omega} P\left(\mathbf{x}|\mathbf{Z}_k^B\right)^{1-\omega}$$

- This is always “squeezed” between the distributions,

$$\min\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right] \leq \bar{P} \leq \max\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right]$$

Illustration of the *Unnormalised* Bound



Lower Bound

- Consider the distribution

$$\hat{P} = \frac{1}{N} P\left(\mathbf{x}|\mathbf{Z}_k^A\right)^{\omega} P\left(\mathbf{x}|\mathbf{Z}_k^B\right)^{1-\omega}$$

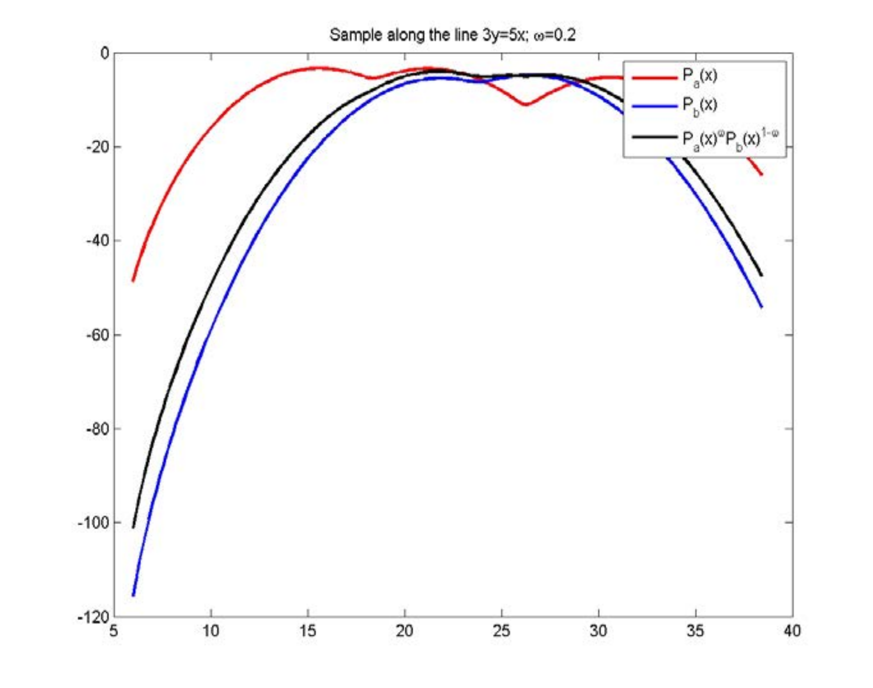
where

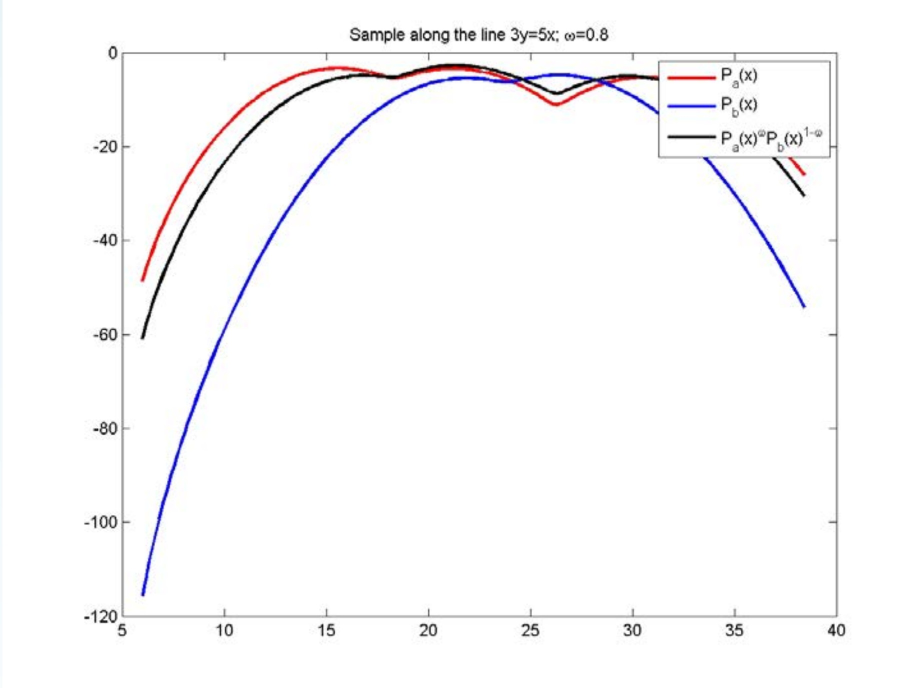
$$N = \int P\left(\mathbf{x}|\mathbf{Z}_k^A\right)^{\omega} P\left(\mathbf{x}|\mathbf{Z}_k^B\right)^{1-\omega} d\mathbf{x}$$

- The WGM obeys the *lower* bound

$$\hat{P} \geq \min\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right] \forall \omega, \mathbf{x}$$

Illustration of the Lower Bound





Interpreting the Lower Bound

- The minimum value of a distribution plays no special role in Bayesian statistics
- However, the bound from below
  - Avoids degenerate cases
    - The support has to contain the intersection of the supports of the prior distributions
- Lower bounds on distributions often play a role in practical filtering algorithms
  - Truncate distributions or modes in MHT if the probability is “too small”

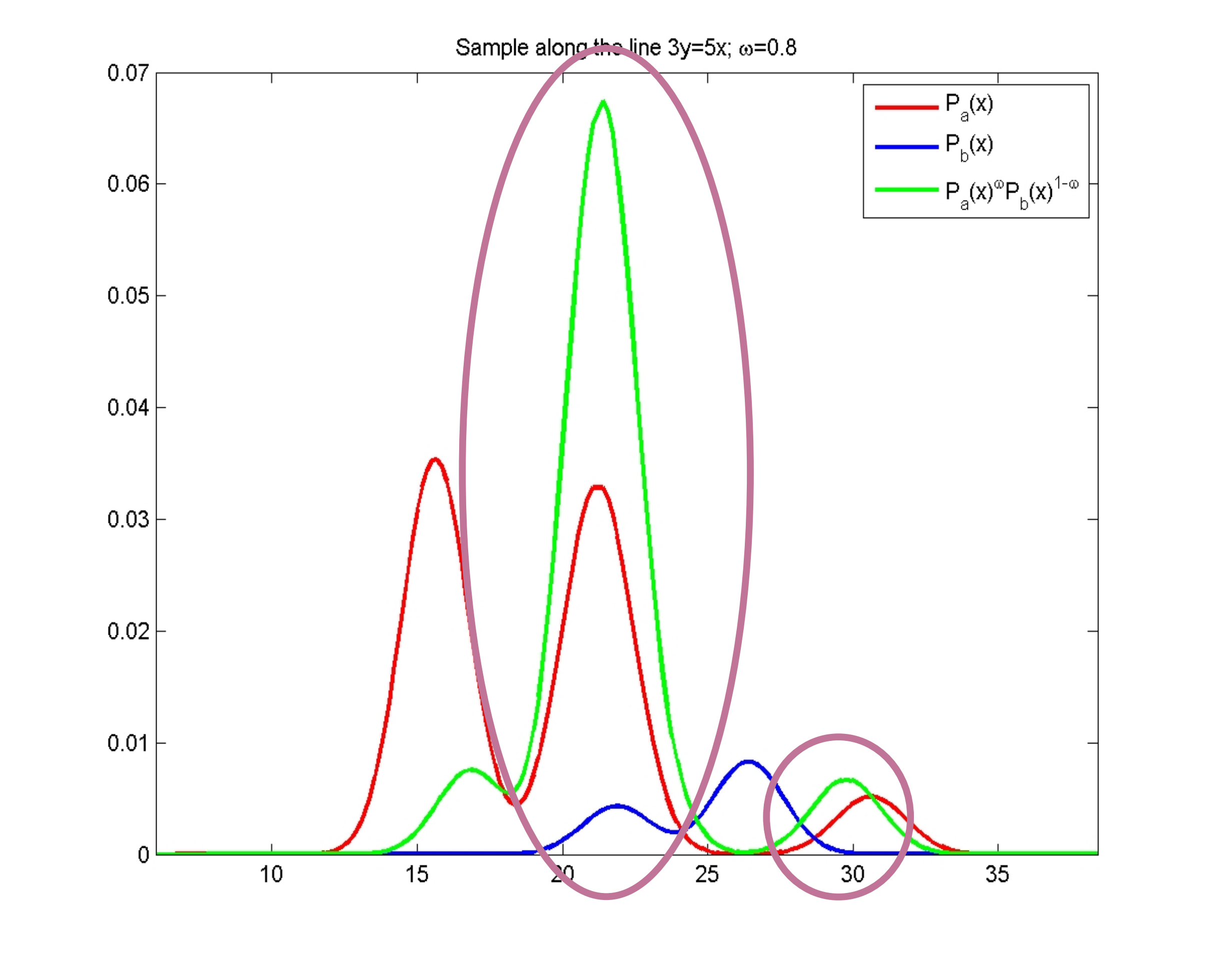
Upper Inequality

- There *can* exist an $\mathbf{x}$ such that

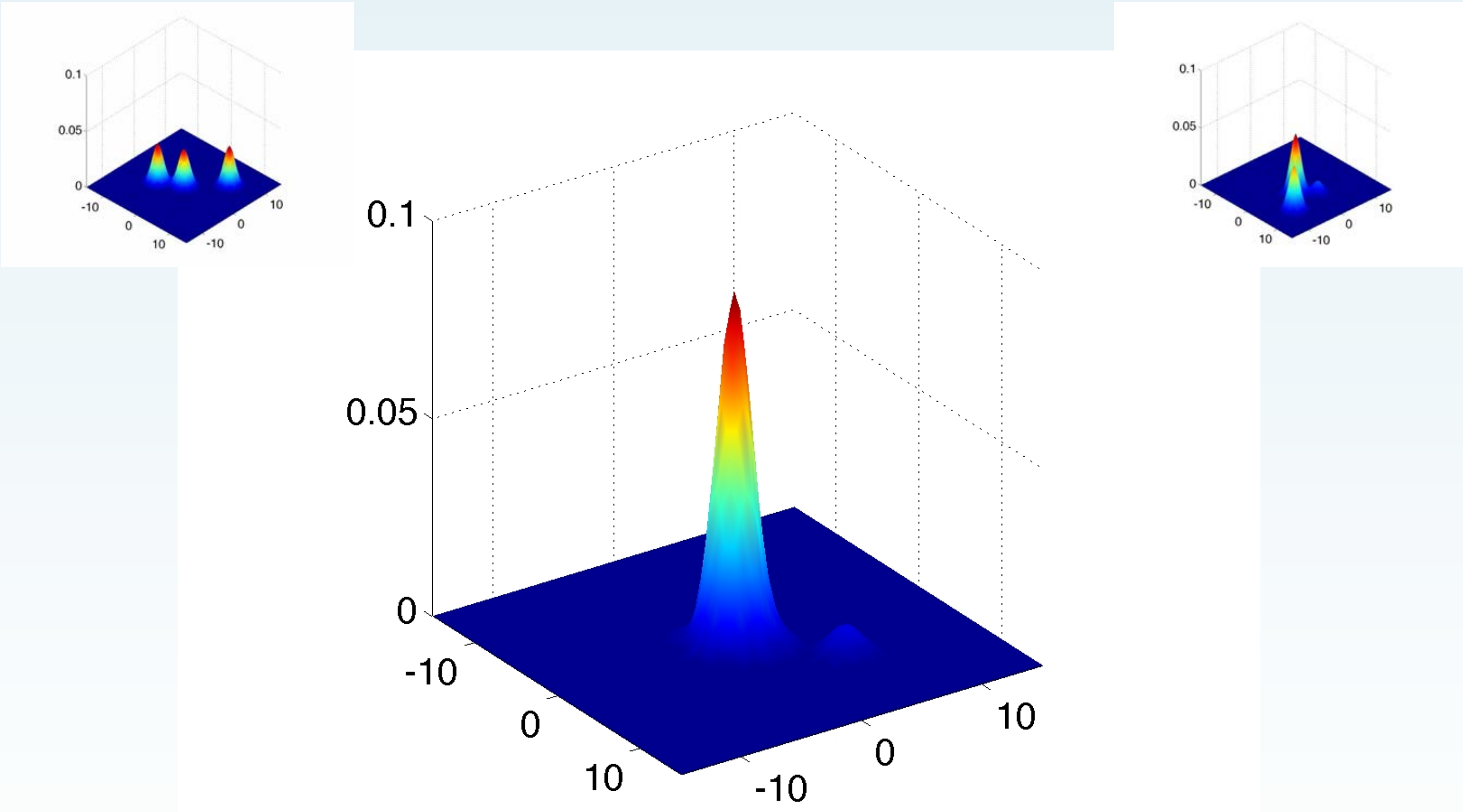
$$\hat{P} \geq \max\left[P\left(\mathbf{x}|\mathbf{Z}_k^A\right), P\left(\mathbf{x}|\mathbf{Z}_k^B\right)\right] \forall\omega$$

- The fact that the distribution can exceed the maximum suggests that *fusion* can occur
  - The distribution becomes “more concentrated”

Illustration of the Upper Inequality

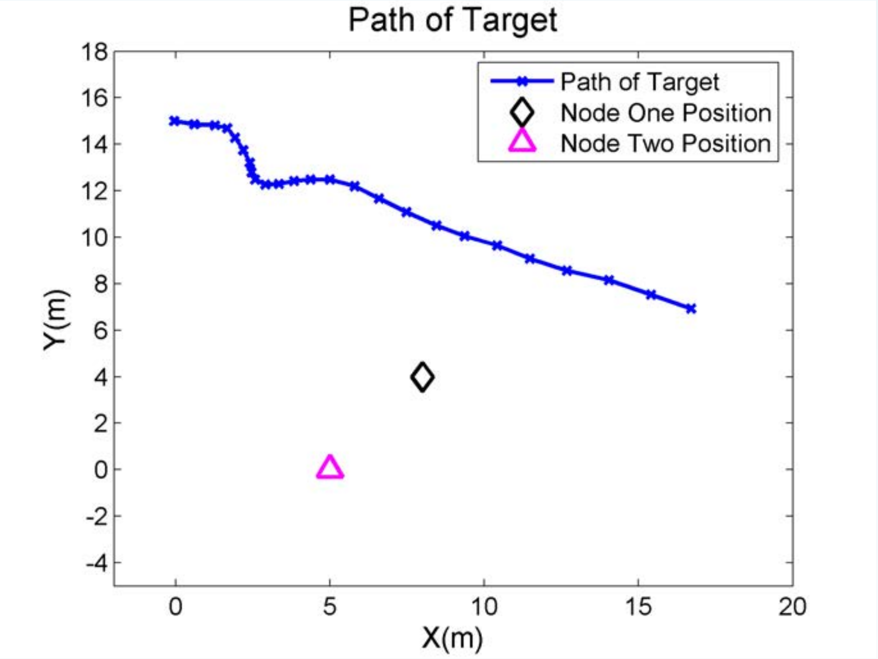


Updated Distribution



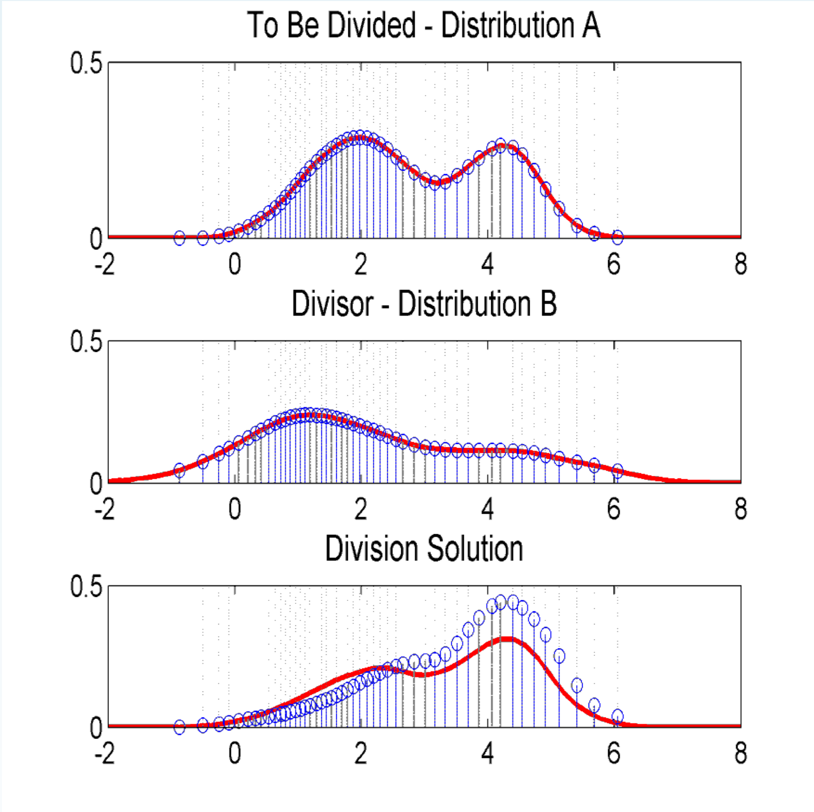
Distributed Target Tracking

- Distributed fusion system
  - 2 nodes
  - Bearings only sensors
- GMMs used to quantify imprecise nature of sensors
  - Bearing-only sensors initialise range-parameterised KFs
- Predictions and updates once per second
- Distribution between nodes once every four seconds



Courtesy of S. Ong, B. Upcroft and T. Bailey, University of Sydney, Australia

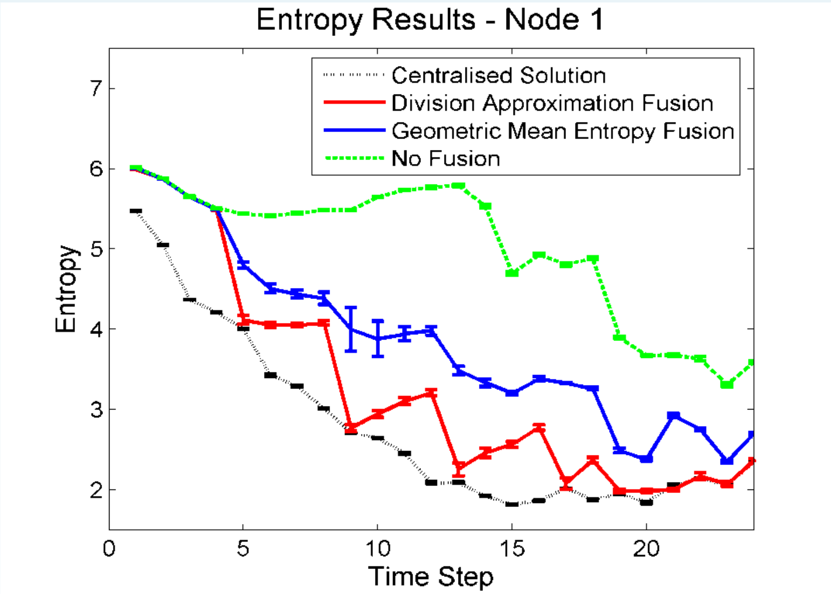
Particle-Based Density Division

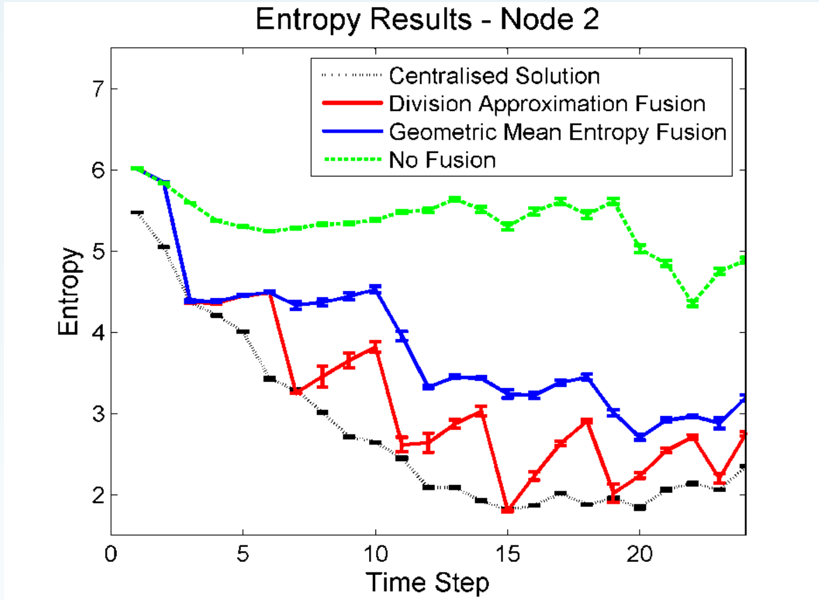


$$P\left(\mathbf{x}|\mathbf{Z}_k^A \cup \mathbf{Z}_k^B\right) \propto \frac{P\left(\mathbf{Z}_k^B|\mathbf{x}\right) P\left(\mathbf{Z}_k^A|\mathbf{x}\right) P(\mathbf{x})}{P\left(\mathbf{Z}_k^A \cap \mathbf{Z}_k^B|\mathbf{x}\right)}$$

Courtesy of S. Ong, B. Upcroft and T. Bailey, University of Sydney, Australia

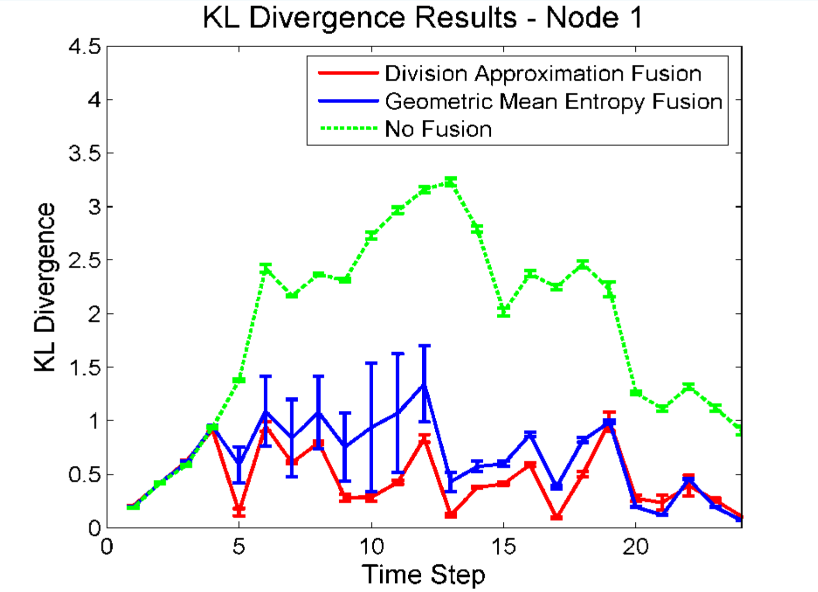
Estimation Results (Entropy)

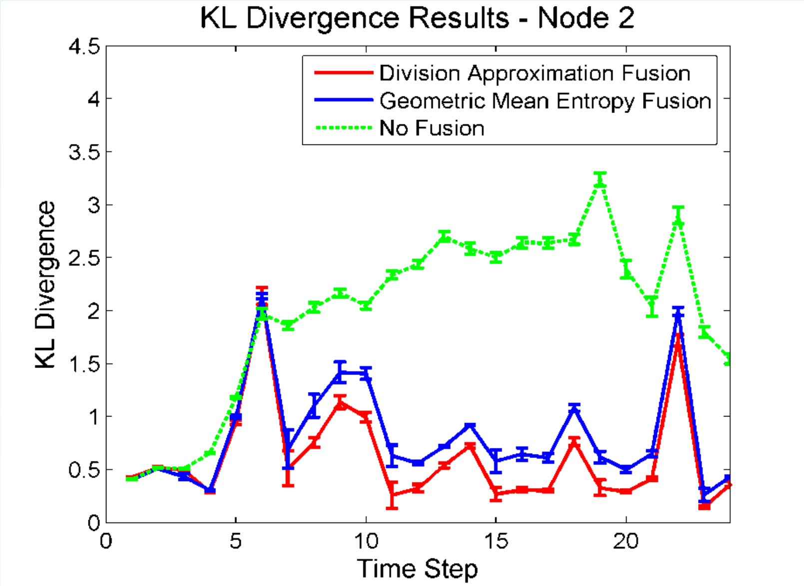




Courtesy of S. Ong, B. Upcroft and T. Bailey, University of Sydney, Australia

KL Divergence from Centralised Solution

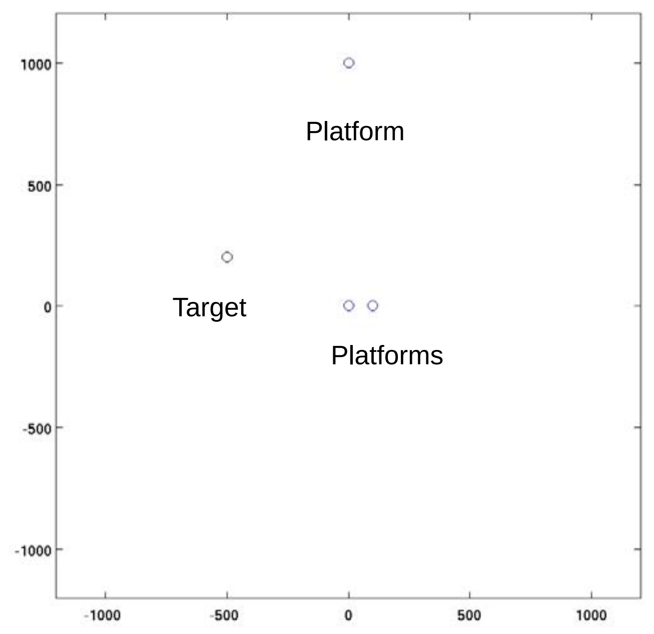




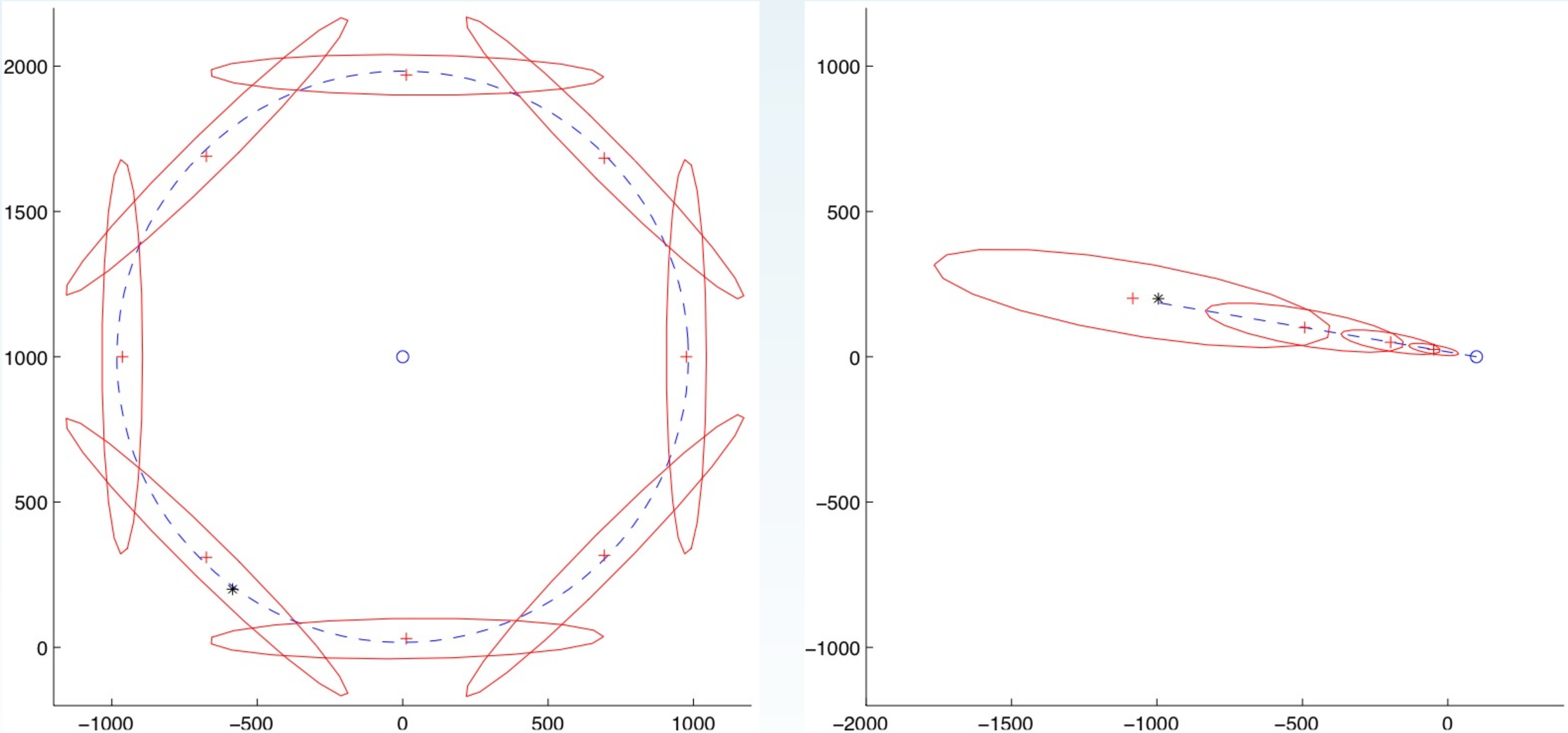
Courtesy of S. Ong, B. Upcroft and T. Bailey, University of Sydney, Australia

Example

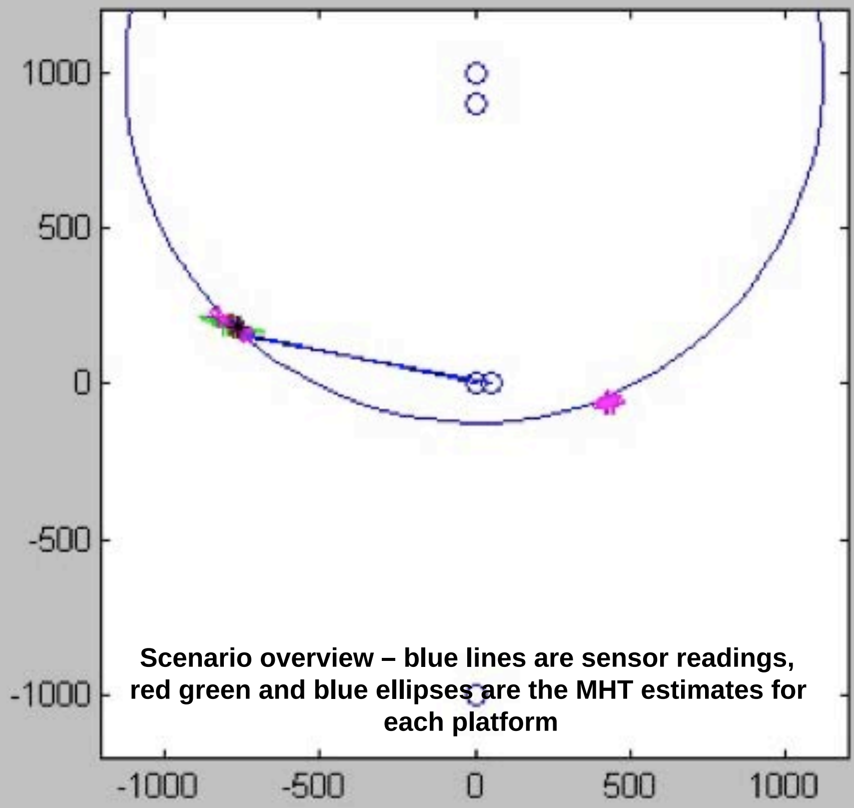
- Distributed fusion system
  - 5 nodes
  - Mix of range and bearing only sensors
- GMMs used to quantify imprecise nature of sensors
  - Bearing-only sensors initialise range-parameterised KFs
  - Range-only sensors initialise angle-parameterised KFs
- Only 70% of communications make it



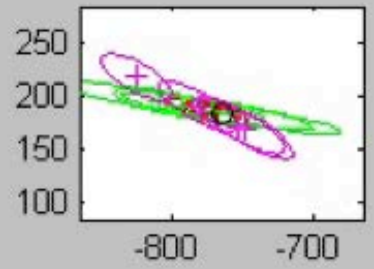
Angle and Range Parameterised KFs



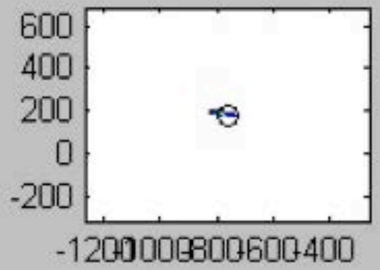
Assumed Independent



Scenario overview – blue lines are sensor readings, red green and blue ellipses are the MHT estimates for each platform

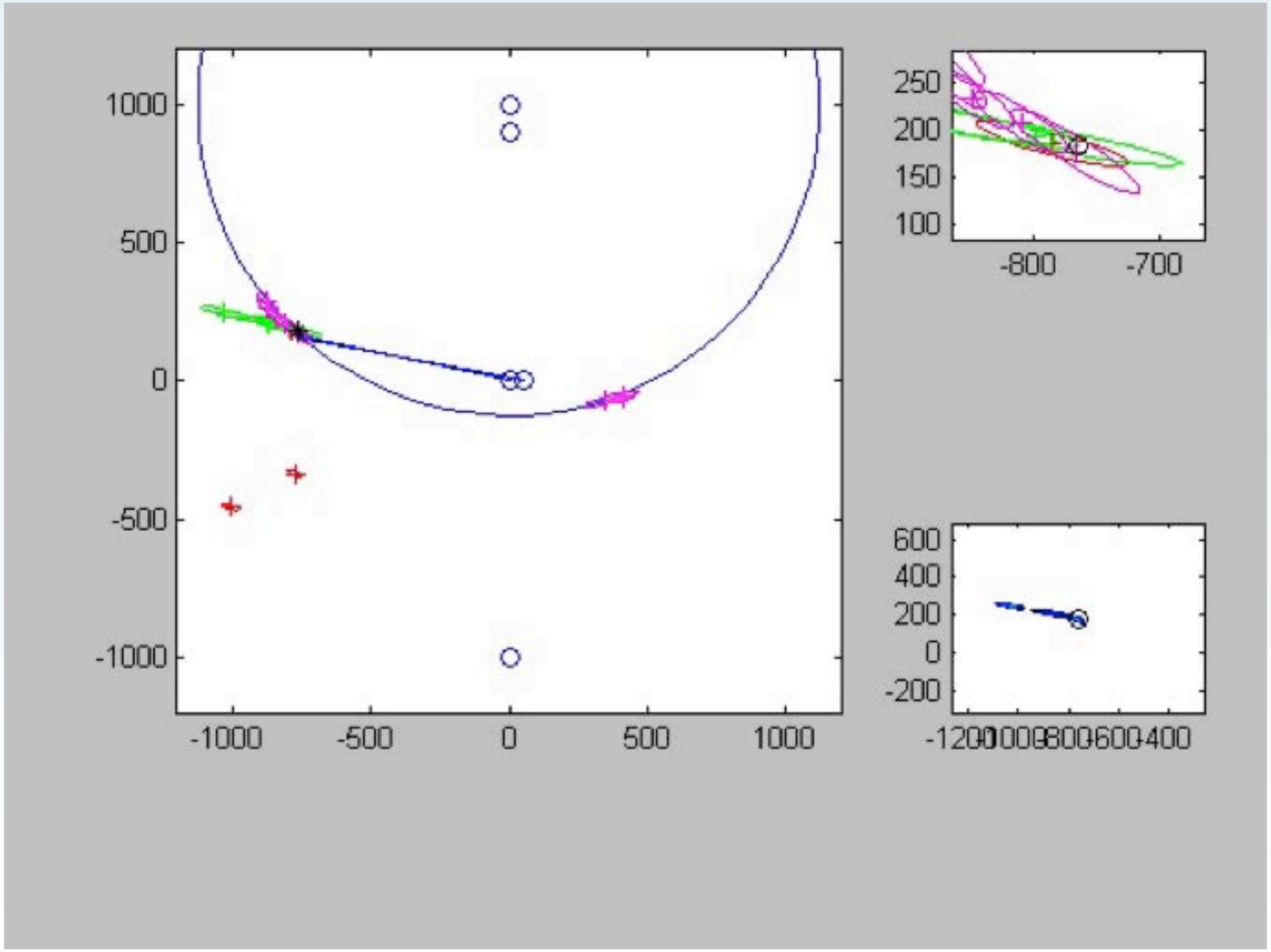


Zoomed in portion of scenario, centered on the target location

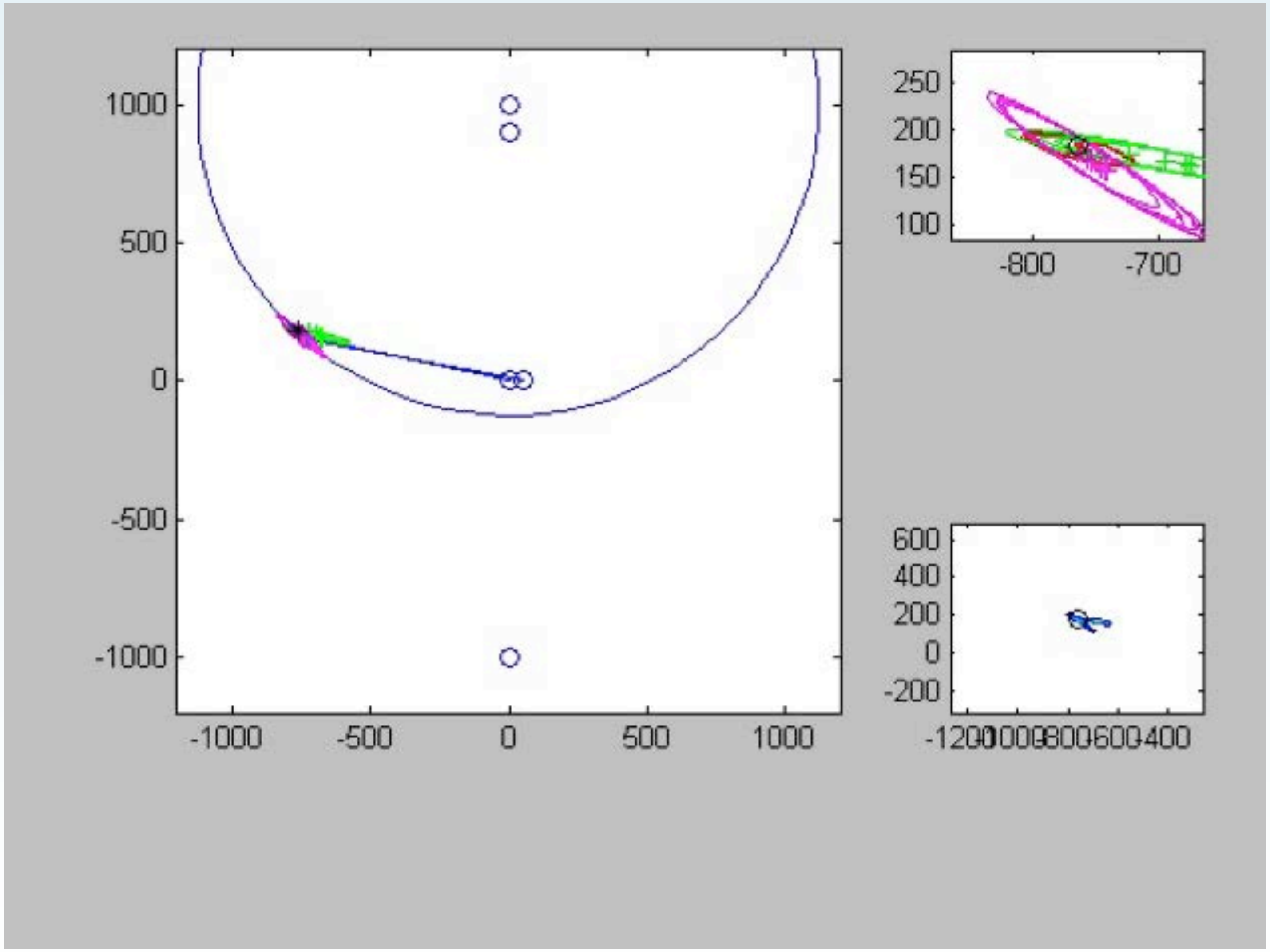


Contours of the pdf, also centered on the target location

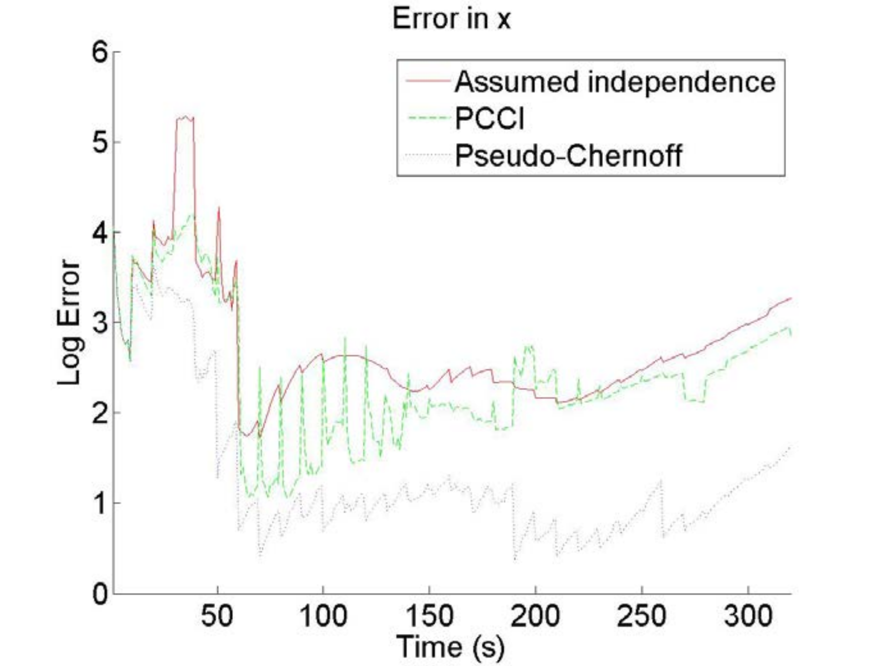
Pairwise Component Covariance Intersection

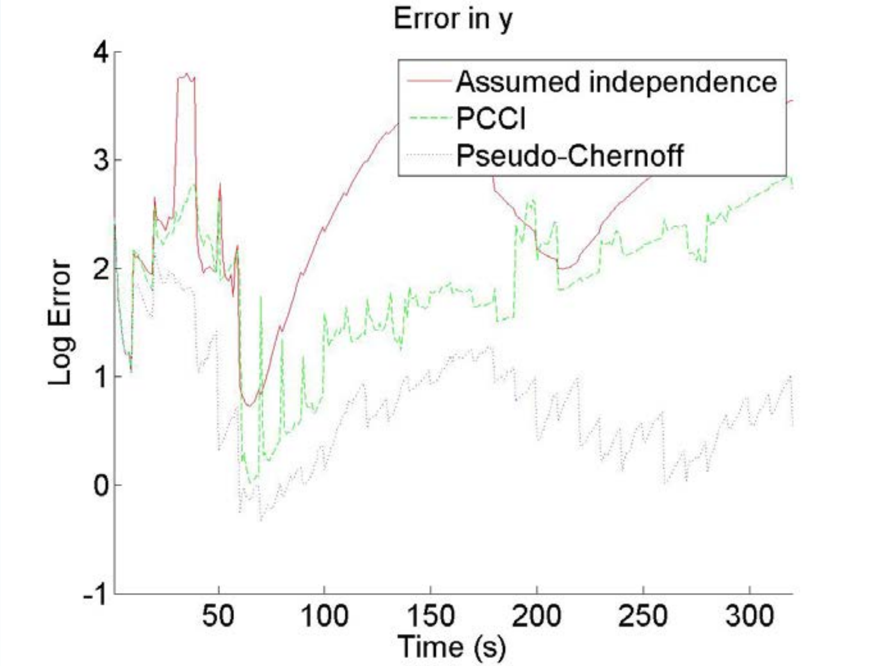


Pseudo-WGM (Crude First Order Approx)

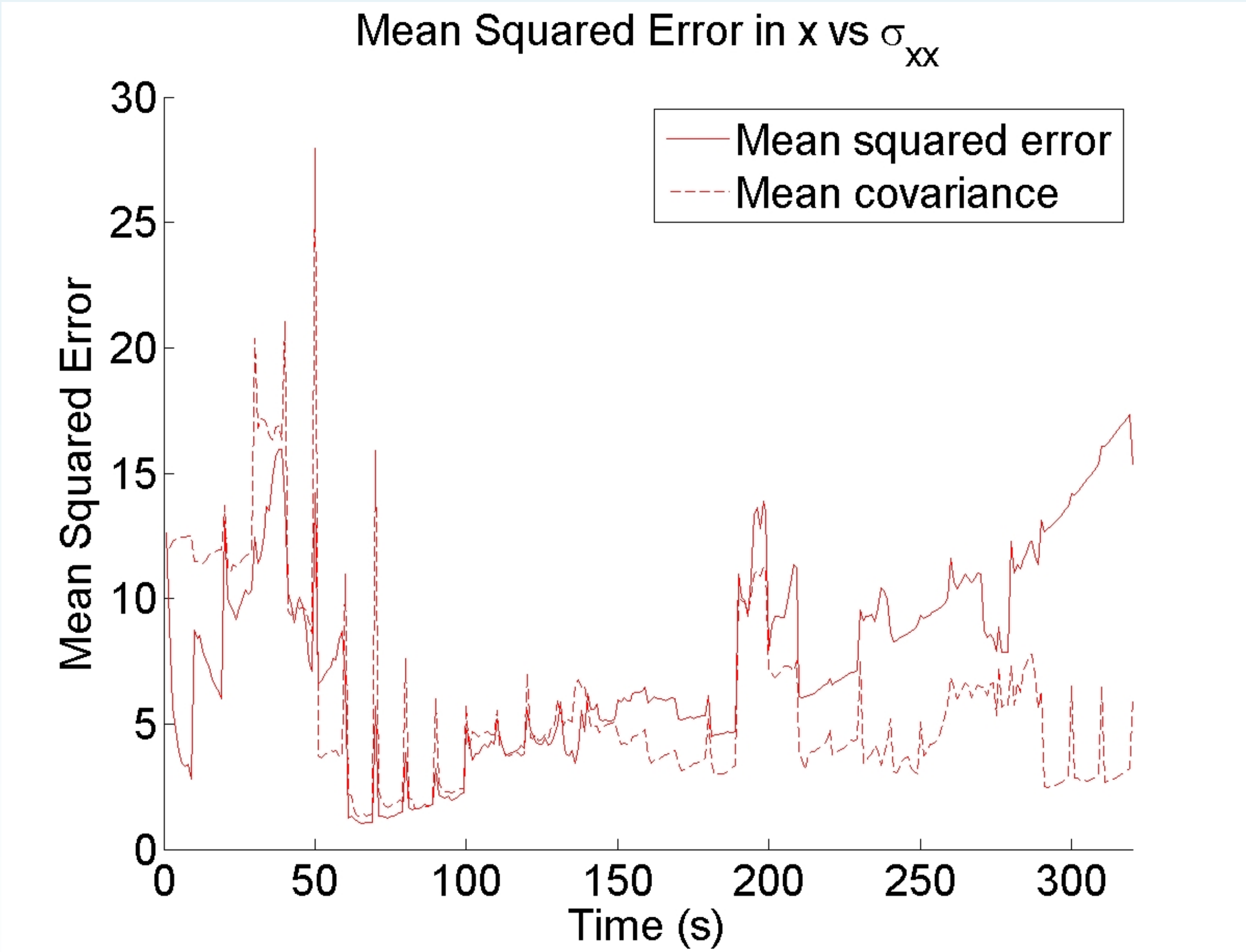


Mean Squared Error in Estimates

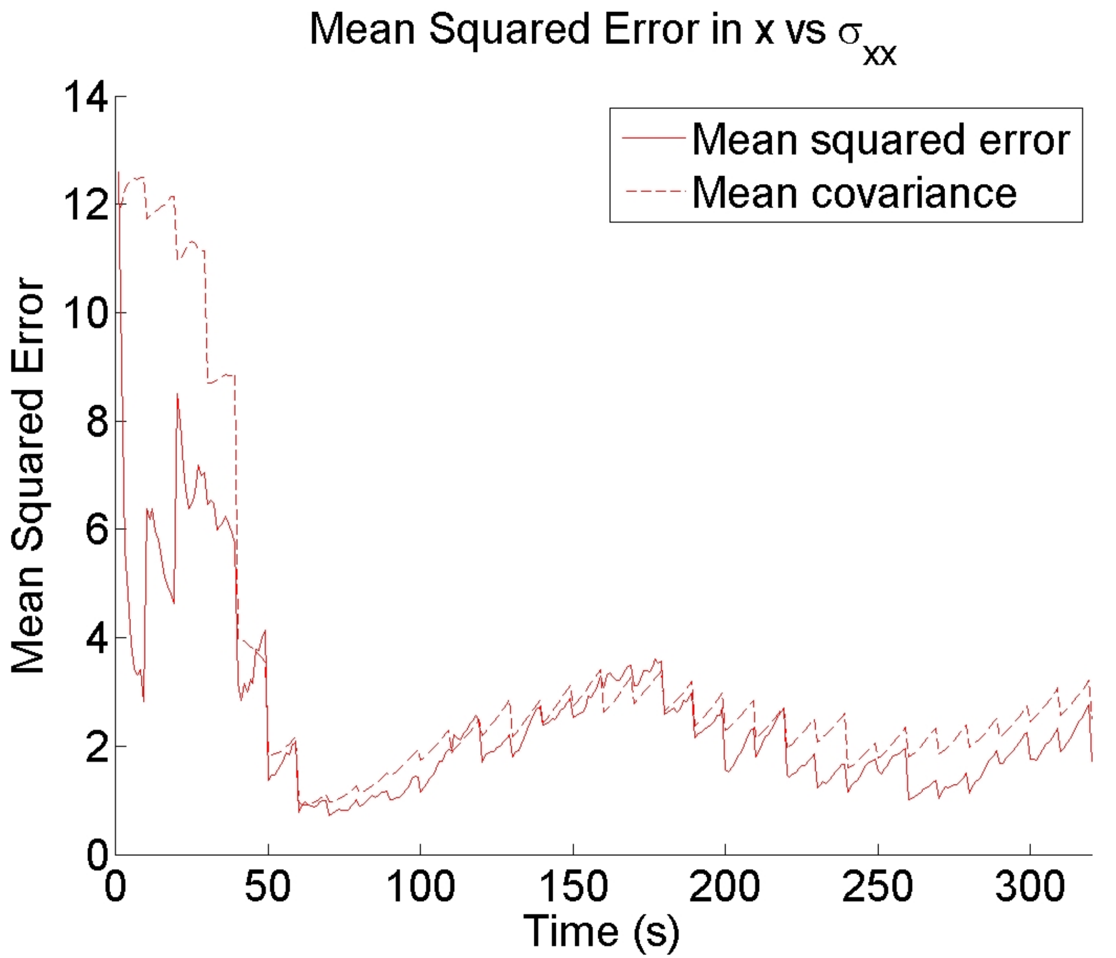




Mean Squared Error vs. Covariance for PCCI



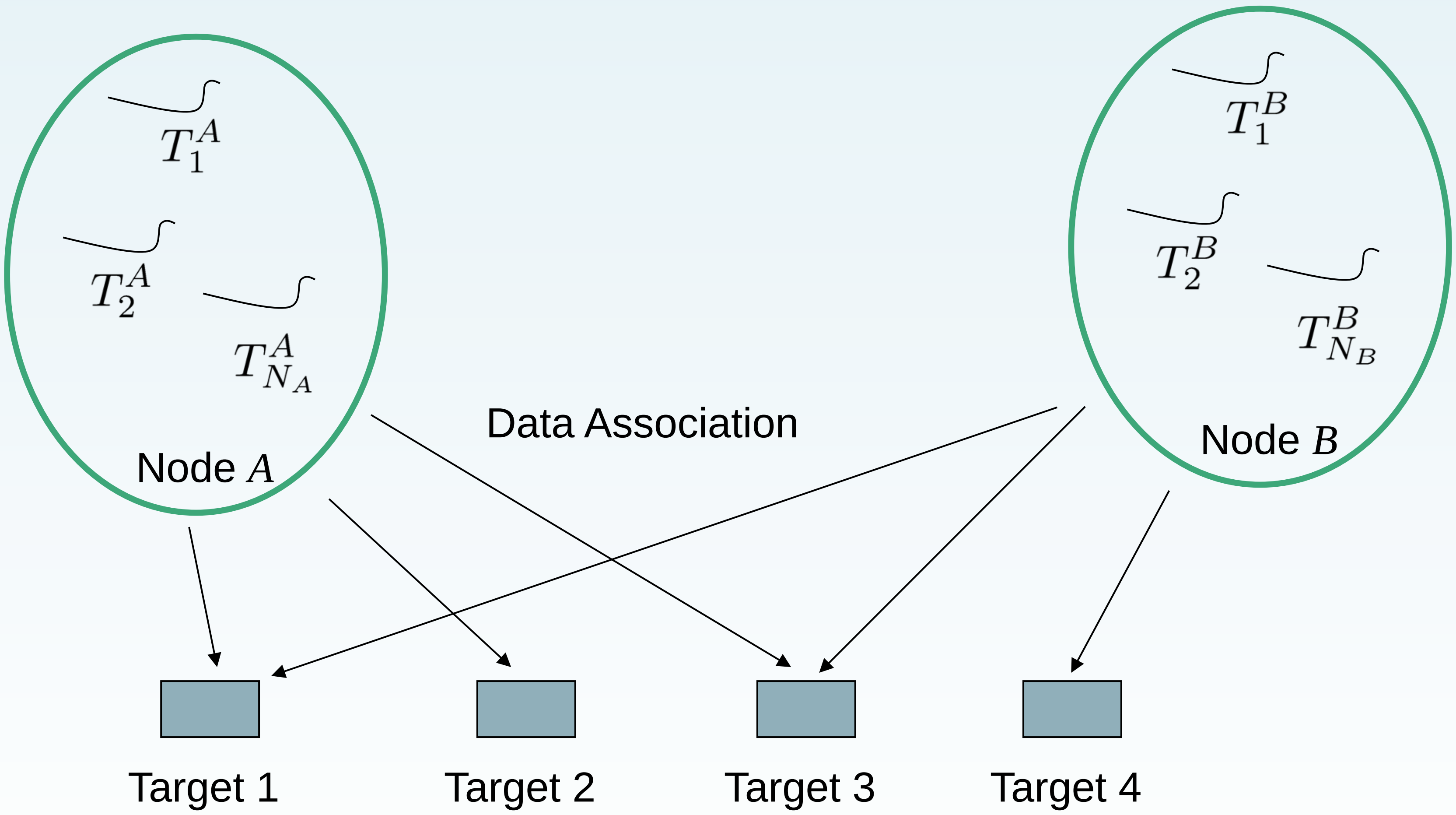
Mean Squared Error vs. Covariance for GMM



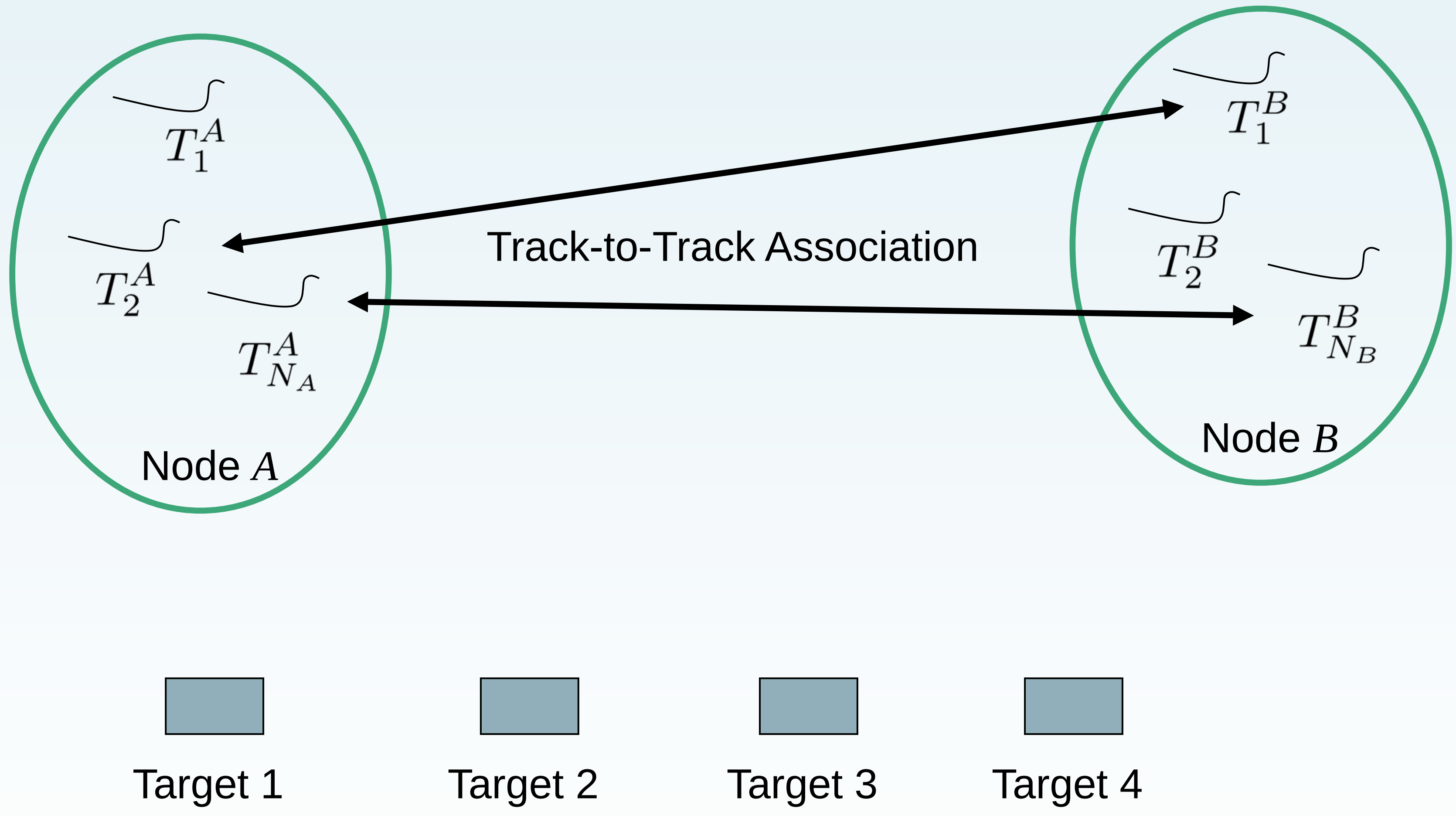
Distributed multi-object tracking

- Motivation
- Distributed data fusion
- Suboptimal distributed data fusion
- *Distributed multi-object tracking with PHD filters*

Track-Based Approaches



Track-Based Approaches

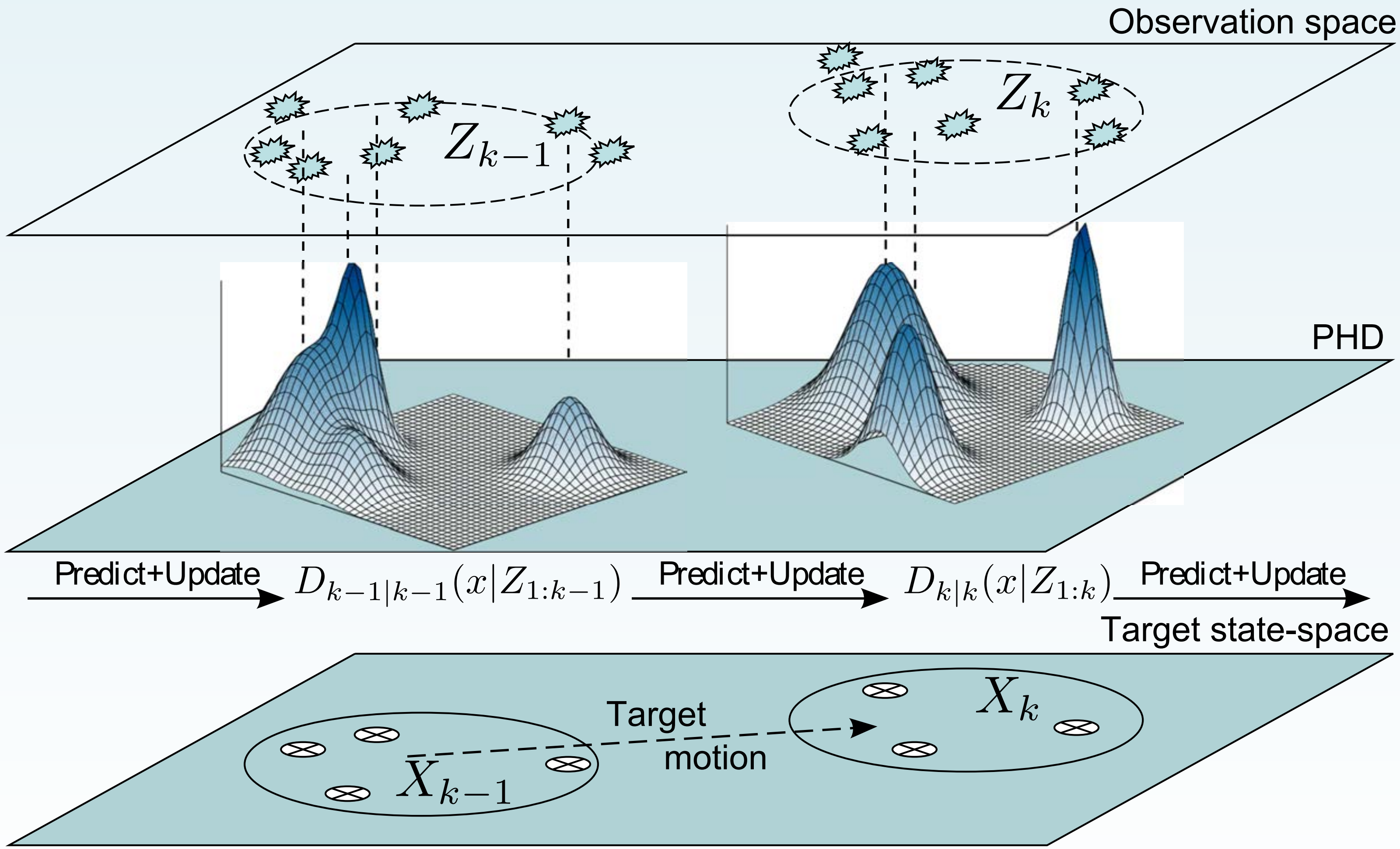


Probabilistic Hypothesis Density Filters

- The idea behind the PHD is to propagate the *intensity function* D($\mathbf{x}$)
- The intensity function specifies the *expected number* of targets in a given region, R

$$\mathrm{E}\left(|X \in \mathcal{R}|\right) = \int_{\mathcal{R}} D(\mathbf{x}) d\mathbf{x}$$

Density of Targets



Probabilistic Hypothesis Density Filters

- The idea behind the PHD is to propagate the *intensity function* $D(\mathbf{x})$
- The intensity function specifies the *average number* of targets in a given region, R

$$\mathrm{E}\left(|X \in \mathcal{R}|\right) = \int_{\mathcal{R}} D(\mathbf{x}) d\mathbf{x}$$

- The complexity of this representation scales with the fidelity of *how* X is represented, not the *number* of targets

Structure of the PHD

- For an iid cluster process, it can be shown that the PHD reduces to

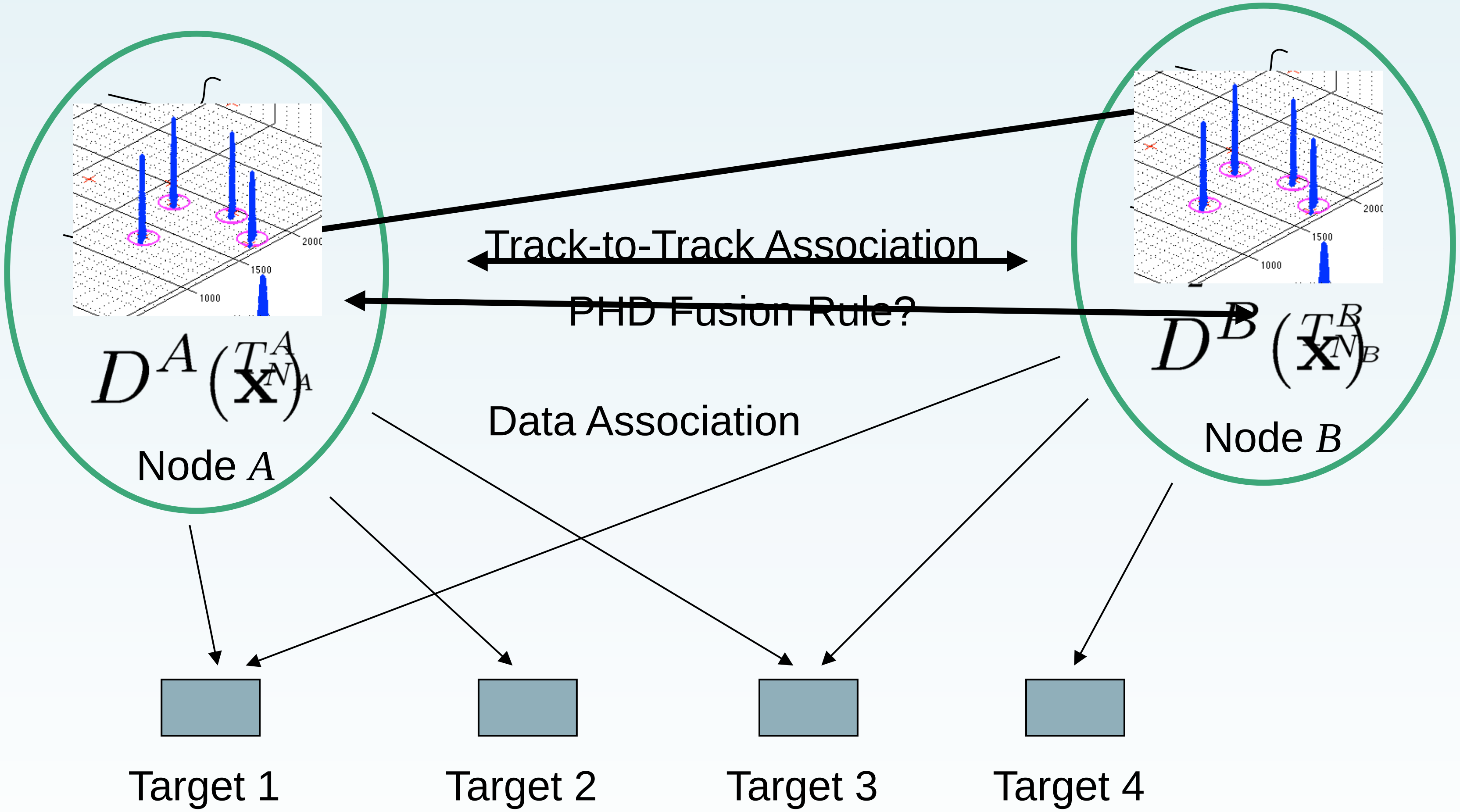
$$D(\mathbf{x}) = s(\mathbf{x}) \sum_{n=0}^{\infty} np(n)$$

Localisation distribution

Cardinality distribution

- Therefore we need only propagate $s(\mathbf{x})$ and $p(n)$
- This can be done in polynomial time

Distributed Multi-target Tracking Redux



Applying EMD to PHD Filters

- Suppose we wish to fuse two nodes A and B
- Each node has its own PHD expressed by its own localisation and cardinality distributions,

$$D^A(\mathbf{x}) = s^A(\mathbf{x}) \sum_{n=0}^{\infty} n p^A(n)$$

$$D^B(\mathbf{x}) = s^B(\mathbf{x}) \sum_{n=0}^{\infty} n p^B(n)$$

Applying EMD to PHD Filters

- By working through all the maths, it can be shown that EMD fusion creates a fused PHD of the form

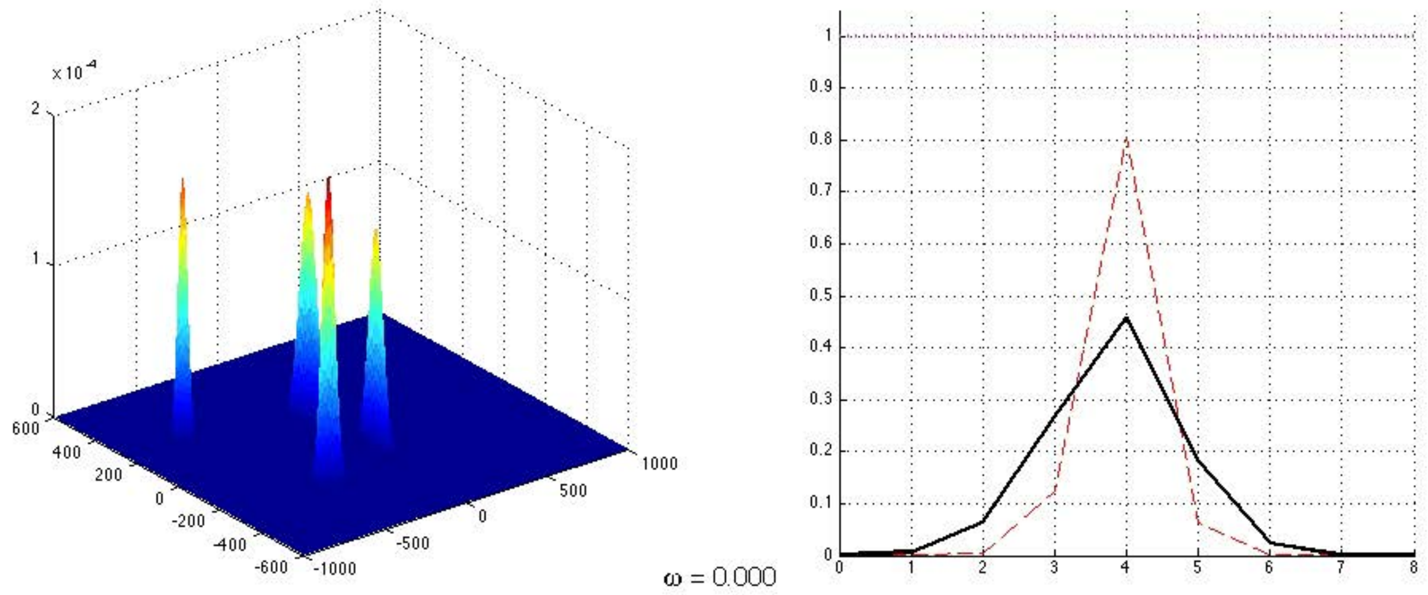
$$\hat{D}_\omega(\mathbf{x}) = \hat{s}_\omega(\mathbf{x}) \sum_{n=0}^{\infty} n \hat{p}_\omega(n)$$

where

$$s^\omega(\mathbf{x}) \propto \left[s^A(\mathbf{x})\right]^\omega \left[s^B(\mathbf{x})\right]^{1-\omega}$$

$$p^\omega(n) \propto \left[p^A(n)\right]^\omega \left[p^B(n)\right]^{1-\omega} \left(\int \left[s^A(\mathbf{x})\right]^\omega \left[s^B(\mathbf{x})\right]^{1-\omega} dx \right)^n$$

Example Fusion of Two PHDs



Properties of the Fusion Equations

- The cardinality distribution *almost* looks like an EMD fusion rule

$$p^{\omega}(n) \propto \underbrace{\left[p^{A}(n)\right]^{\omega}\left[p^{B}(n)\right]^{1-\omega}}_{\text{EMD-Like}} \underbrace{\left(\int \left[s^{A}(\mathbf{x})\right]^{\omega}\left[s^{B}(\mathbf{x})\right]^{1-\omega} dx\right)^{n}}_{\text{Geometric Scaling}}$$

Cardinality Scaling Factor

- It is well-known that the weighted geometric mean is *convex*
- Therefore,

$$[s^A(\mathbf{x})]^\omega [s^B(\mathbf{x})]^{1-\omega} \leq \omega s^A(\mathbf{x}) + (1-\omega) s^B(\mathbf{x})$$

- Since each localisation distribution is normalised,

$$\int \left[s^A(\mathbf{x})\right]^\omega \left[s^B(\mathbf{x})\right]^{1-\omega} d\mathbf{x} \leq \omega \int s^A(\mathbf{x}) d\mathbf{x} + (1-\omega) \int s^B(\mathbf{x}) d\mathbf{x}$$
$$\leq 1$$

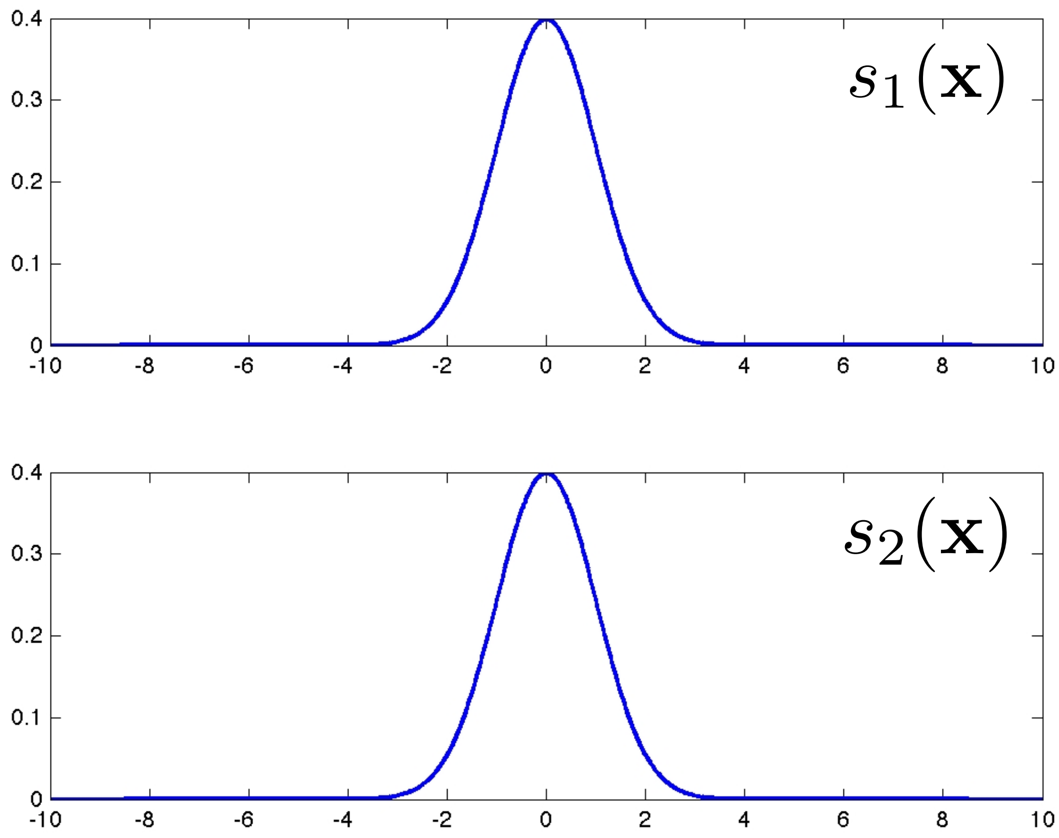
Cardinality Scaling Factor

- Because the scale factor is less than 1, the higher cardinality terms tend to receive a lower weight
- This becomes more marked the smaller the value of

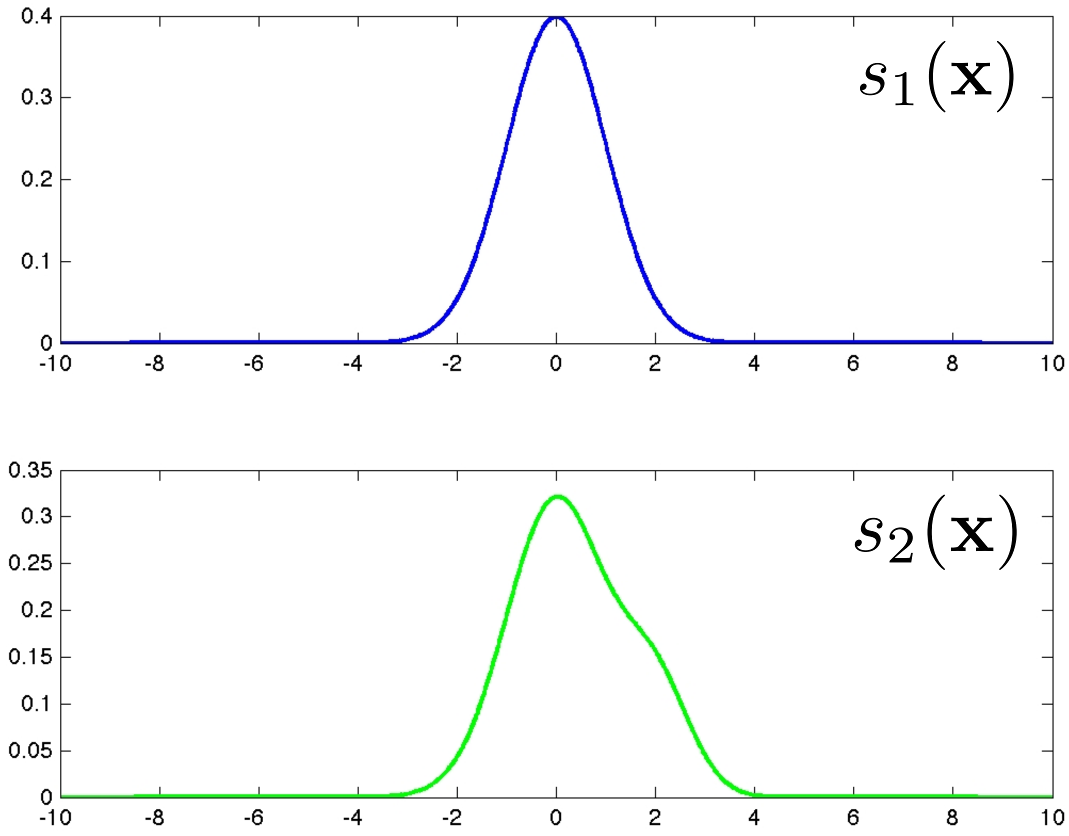
$$\int \left[s^A(\mathbf{x})\right]^{\omega} \left[s^B(\mathbf{x})\right]^{1-\omega} d\mathbf{x}$$

- This integral actually provides some kind of measure of the similarity between the localisation distributions

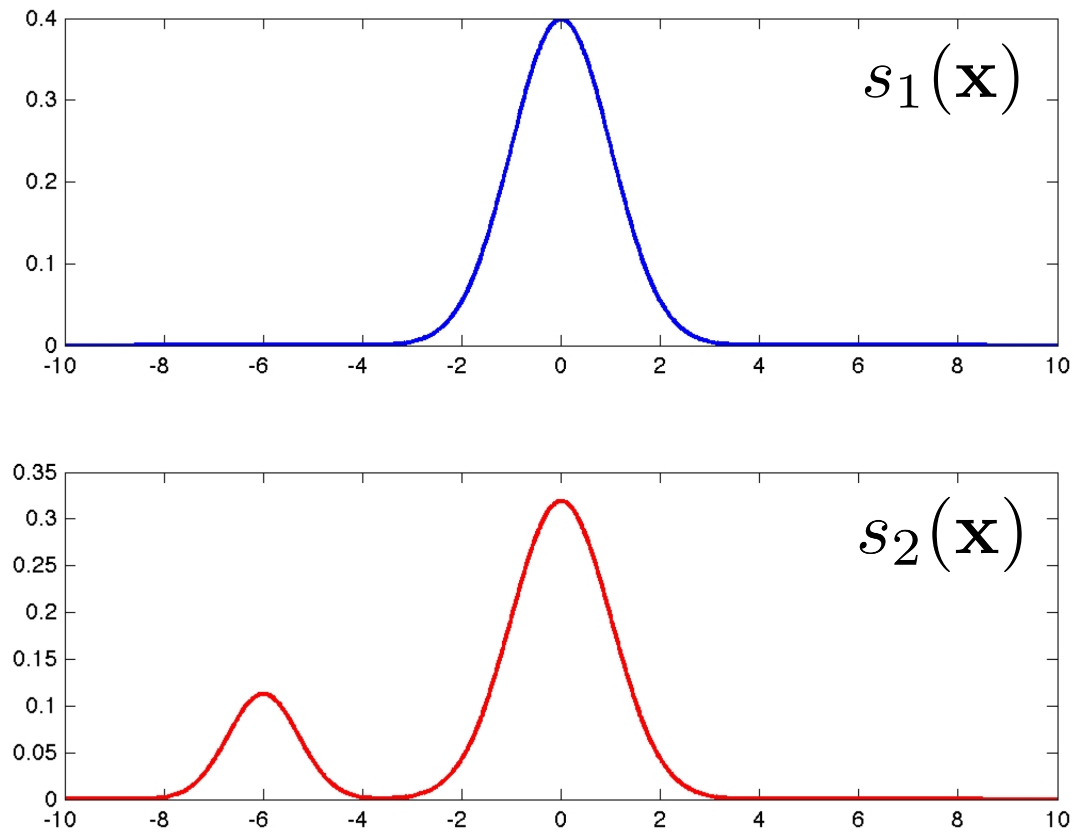
Cardinality Scaling Factor



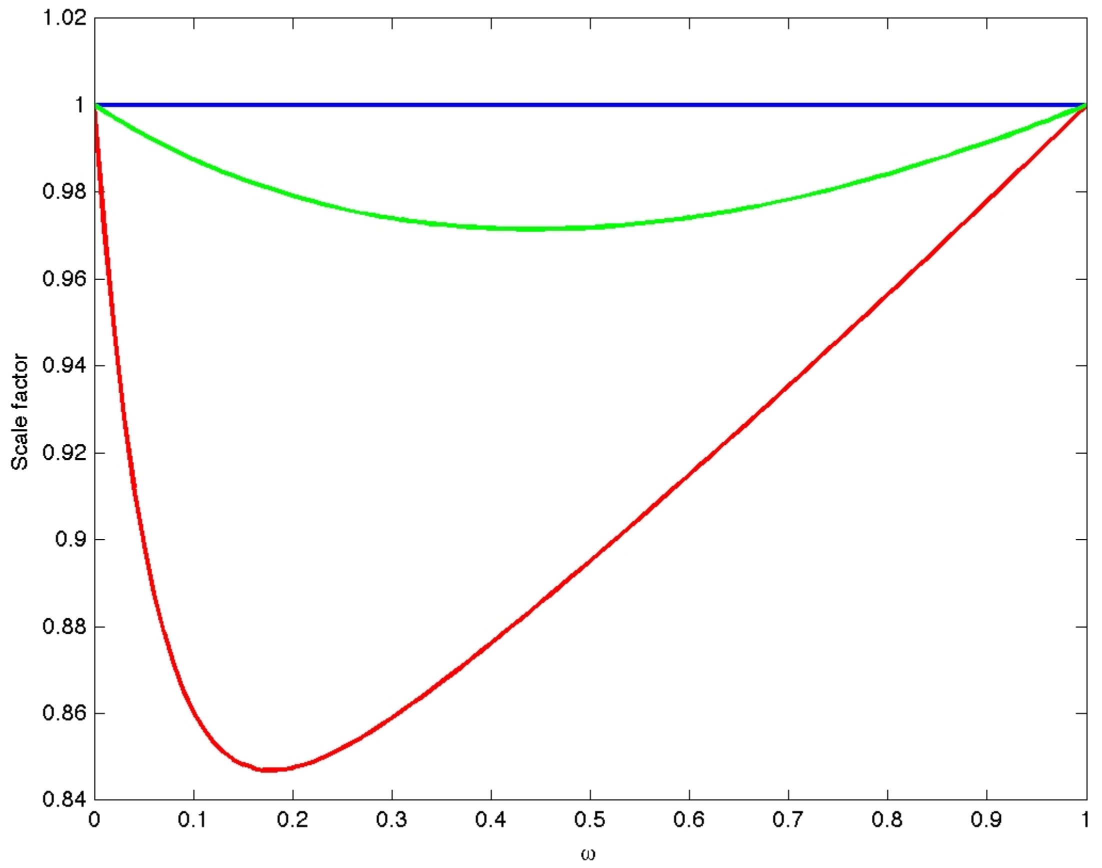
Cardinality Scaling Factor



Cardinality Scaling Factor



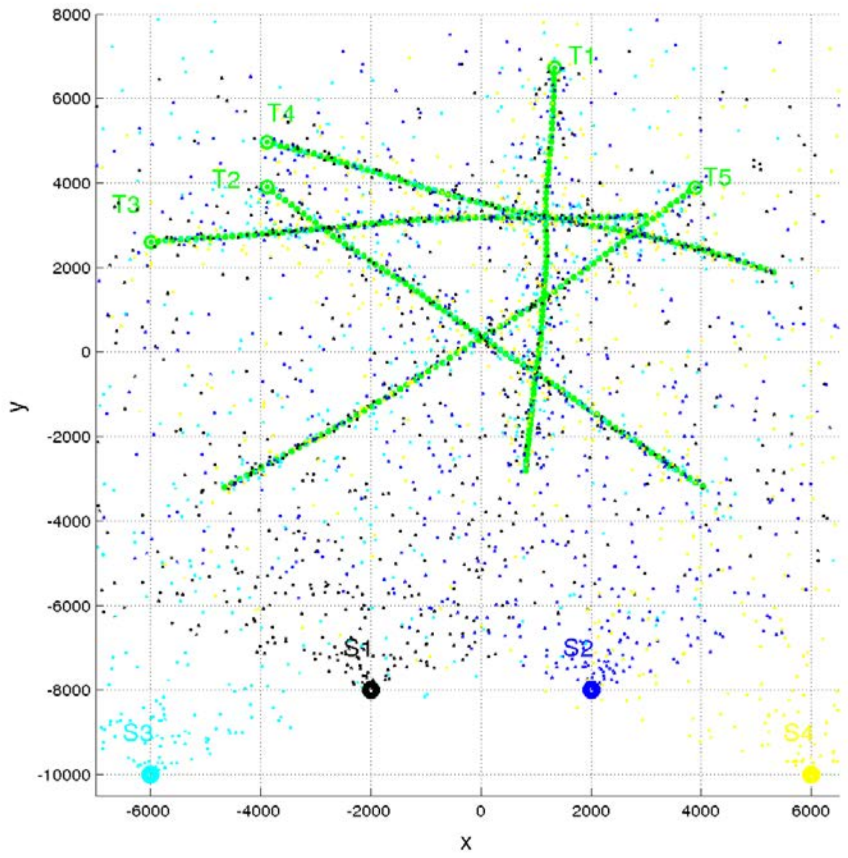
Cardinality Scaling Factor



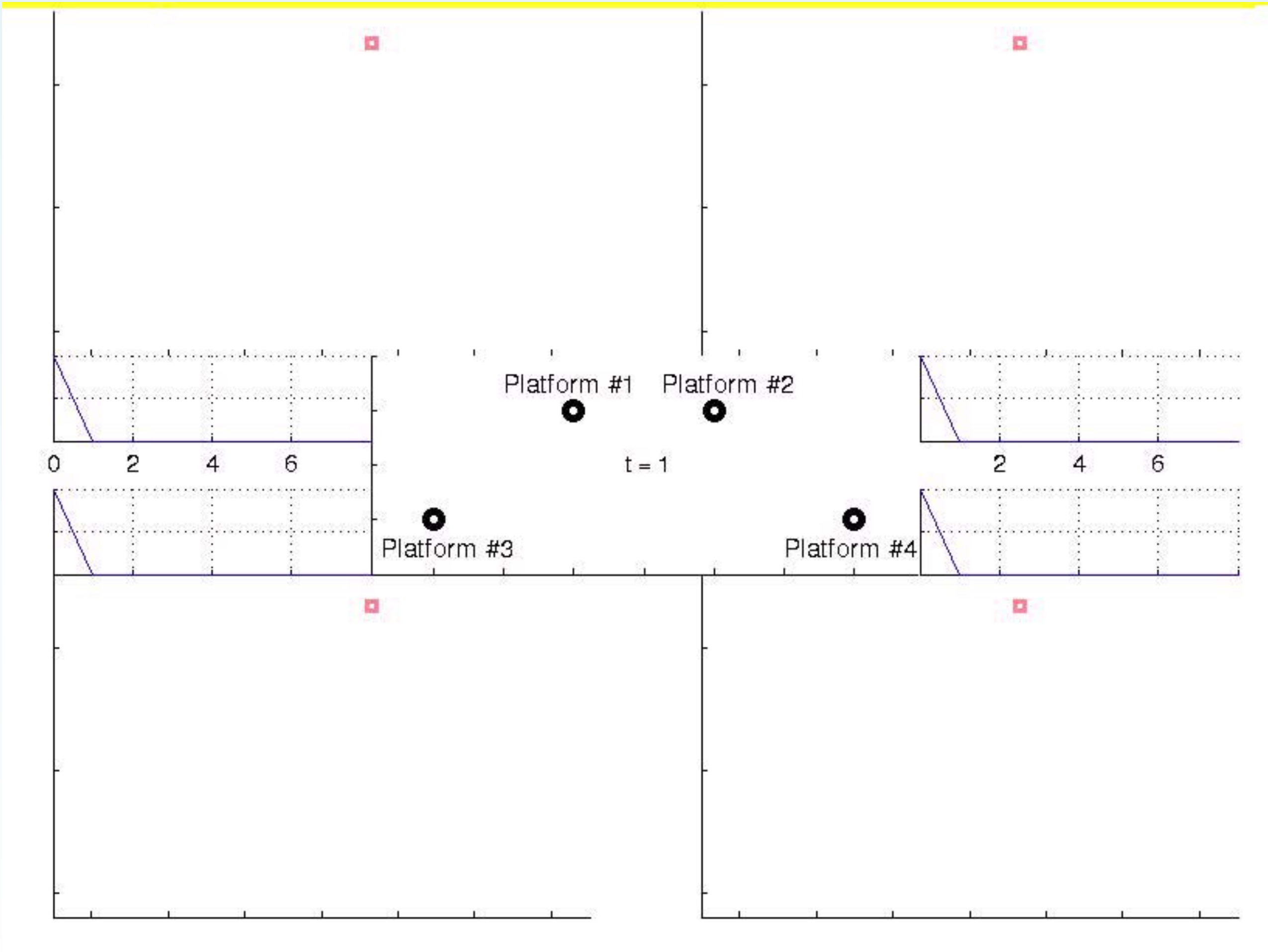
Property of the Scaling Factor

- The integral is 1 when the distributions are the same
- Its value declines as the distributions become “less similar”
- A cause of dissimilarity is clutter
- Clutter tends to create spurious peaks in the PHD
- The clutter should be independent in each node over time
- Therefore this behavior is “kind of” reasonable

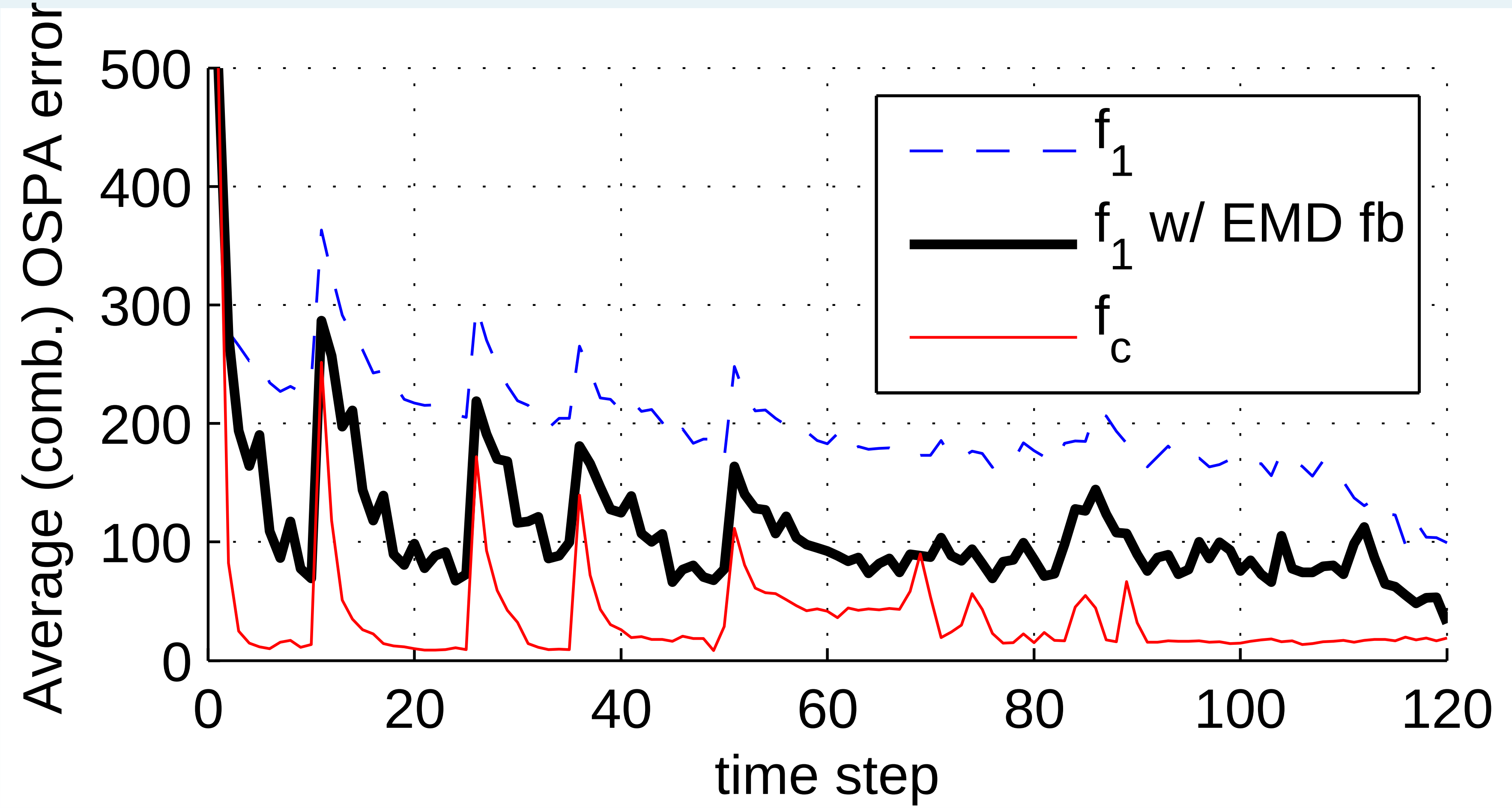
Simulation Scenario



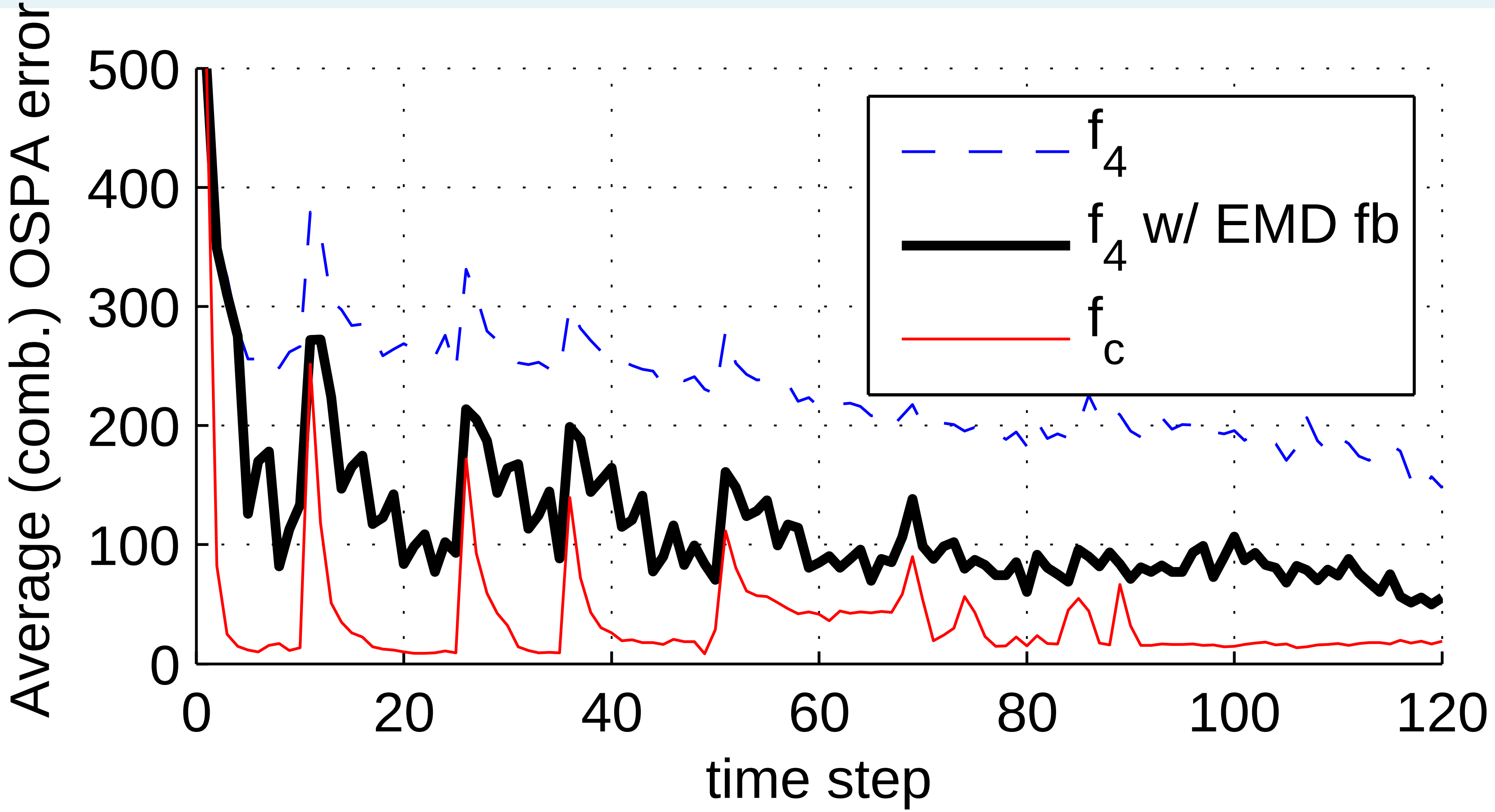
Simulation Scenario



OSPA Results for Sensor 1



OSPA Results for Sensor 4



Tracking Small Ships

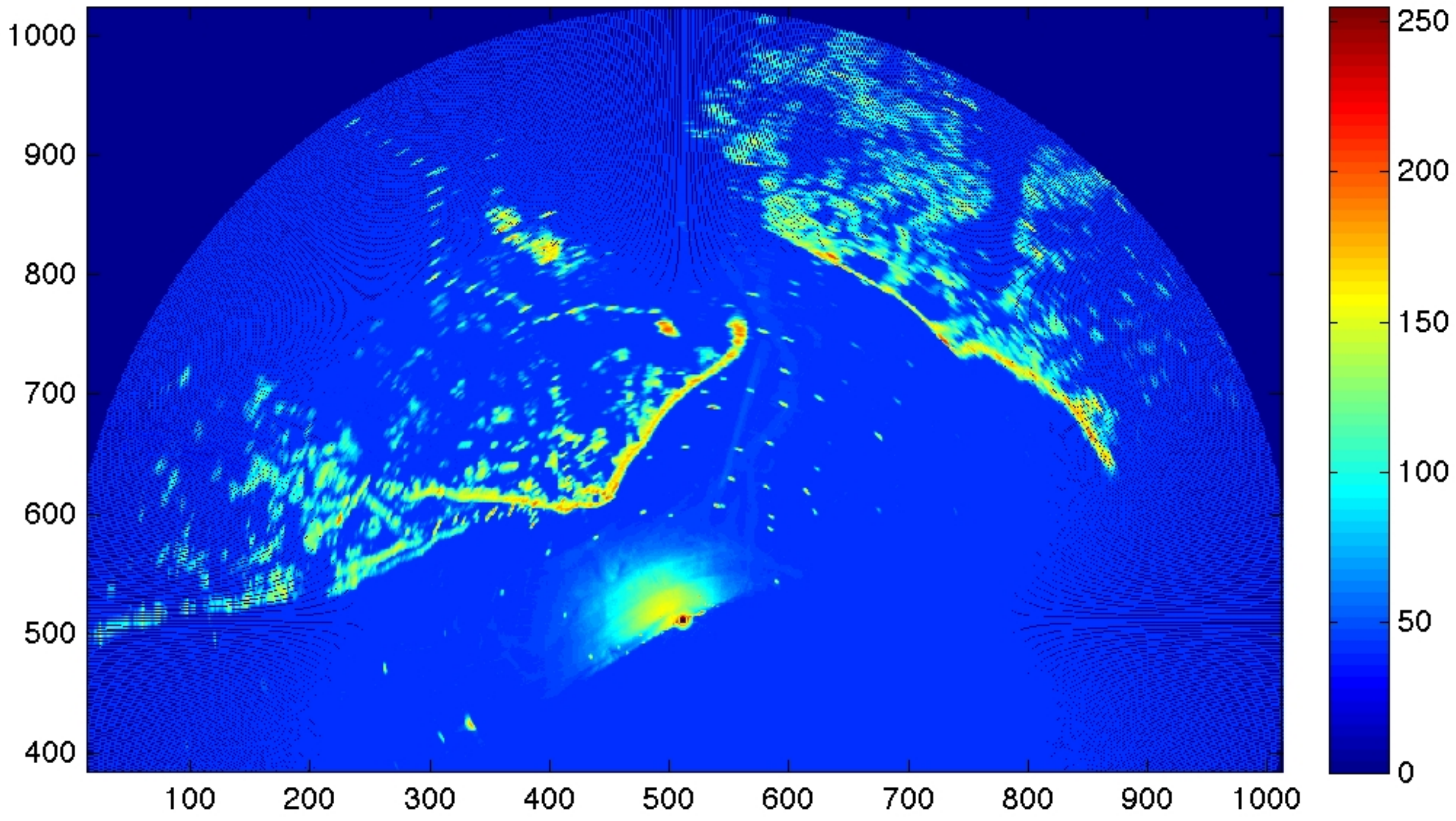
- To test the system in practice, we collaborated with Heriot-Watt and BAE Systems on a CDE-funded project
- The goal was to look at tracking small boats using multiple sensors
  - Cluttered, messy environment
  - Distributed fusion systems

The Location

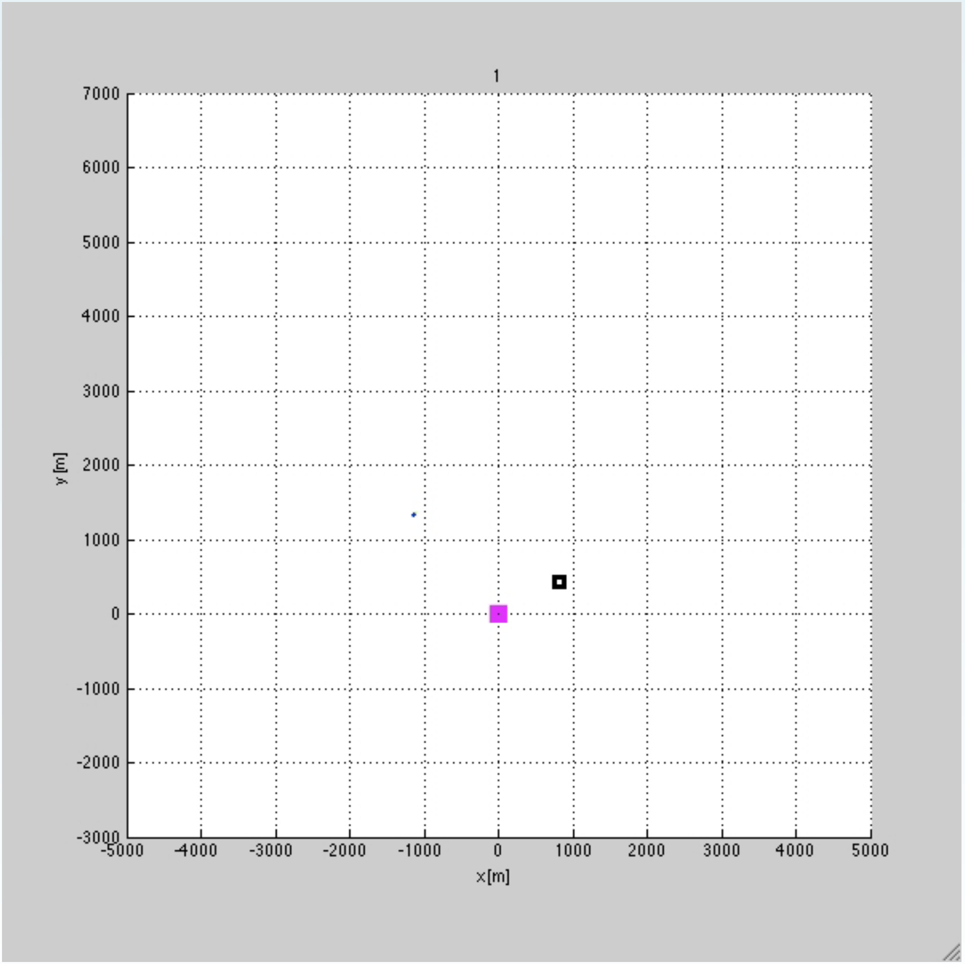
The Targets

The Van

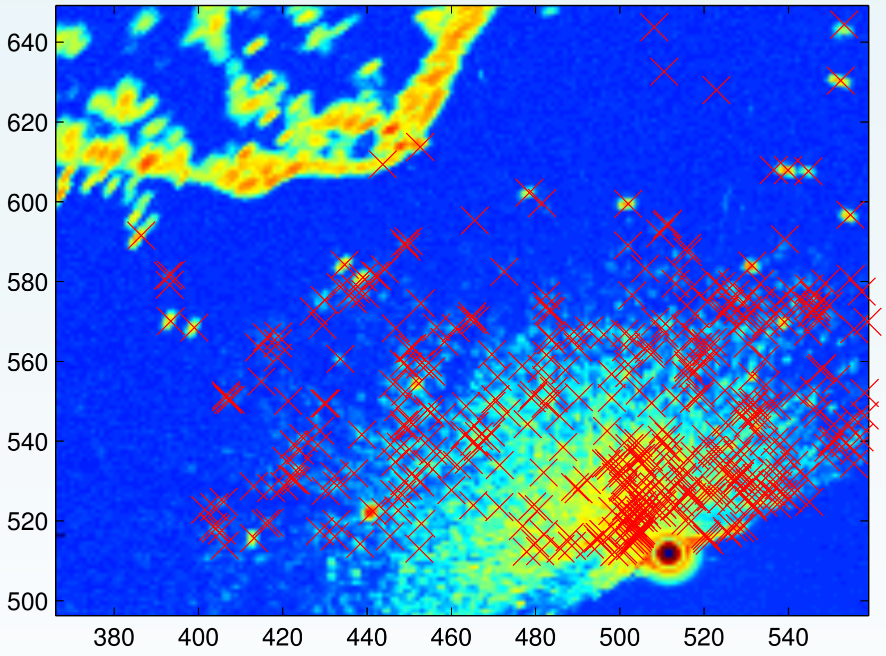
Average Intensity Values



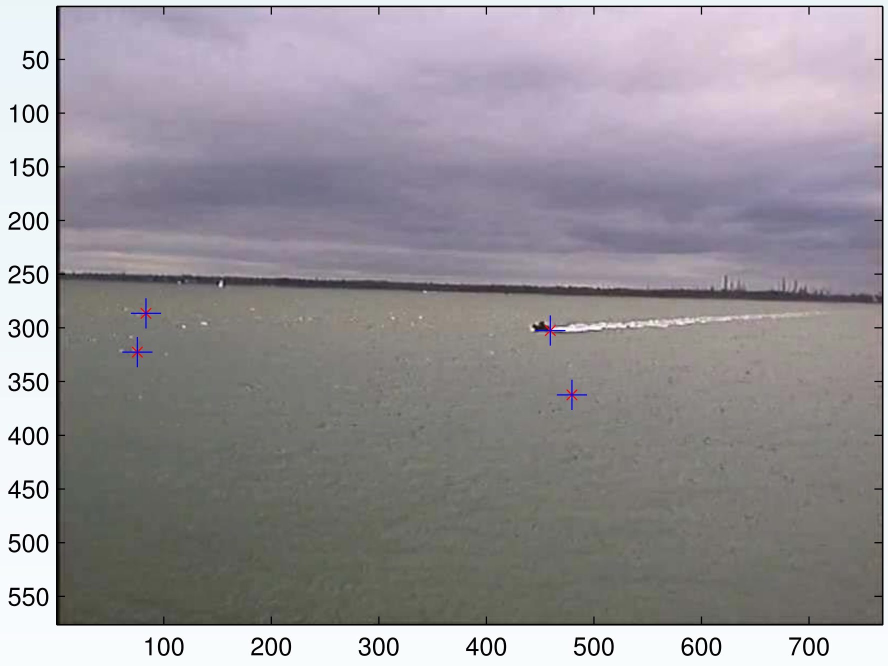
Raw Detections



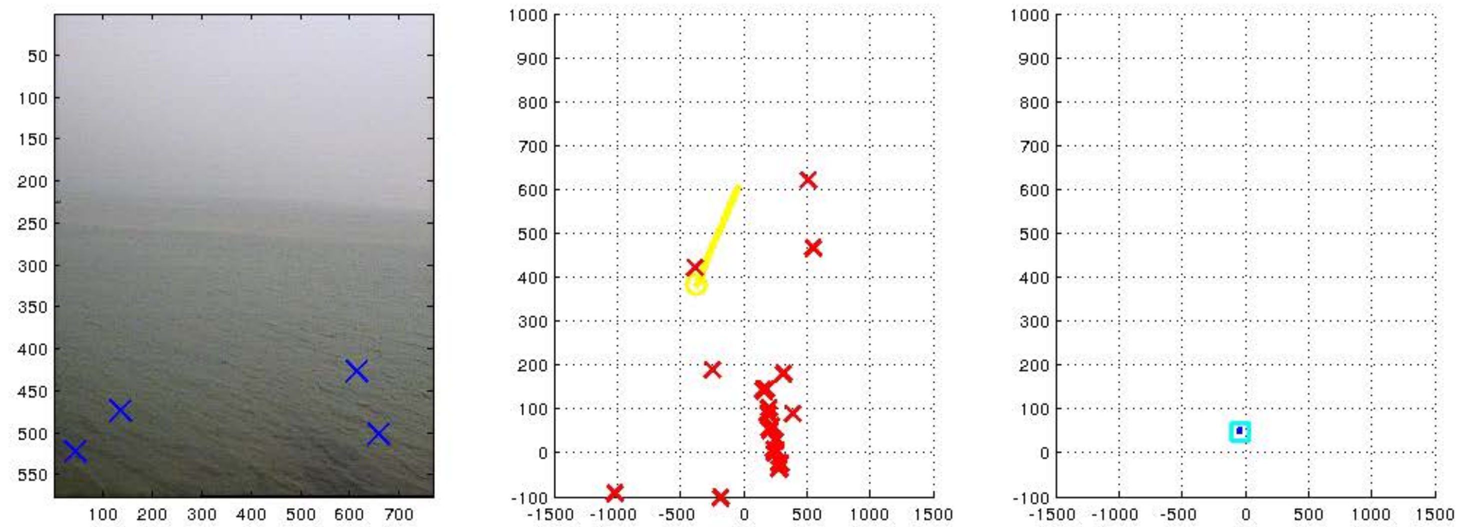
Example Data



Radar Data

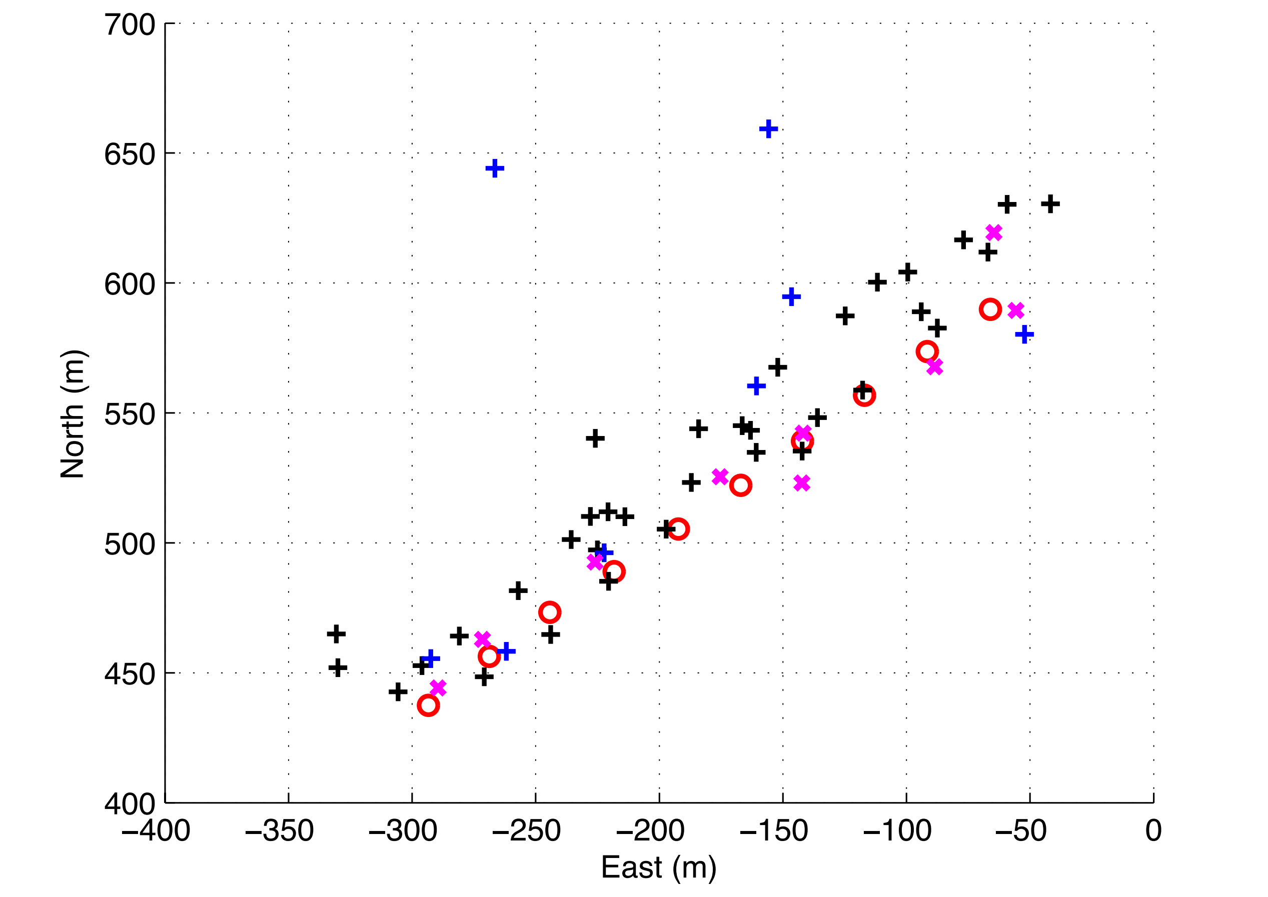


Camera Data

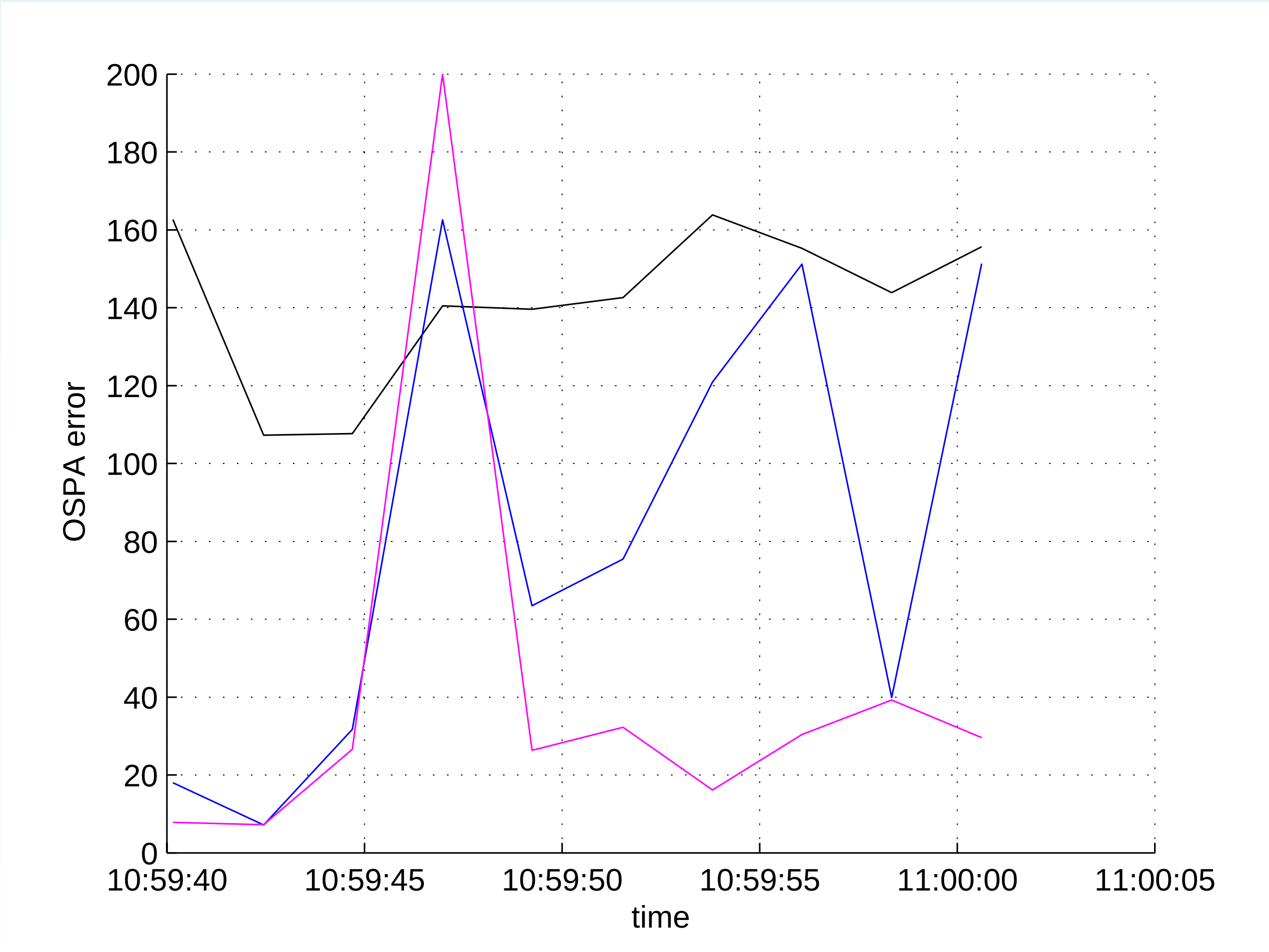


50
100
150
200
250
300
350
400
450
500
550
100 200 300 400 500 600 700
1000
900
800
700
600
500
400
300
200
100
0
-100
-1500 -1000 -500 0 500 1000 1500
1000
900
800
700
600
500
400
300
200
100
0
-100
-1500 -1000 -500 0 500 1000 1500

Ground Truth, RADAR, EO and Fused Results



OSPA Results



Summary and Conclusions

- Distributed data fusion makes it possible to scatter information processing throughout a network in an adaptive manner
- However, book keeping dependencies properly is extremely challenging