Target Tracking: Lecture 7
Multiple Sensor Tracking Issues

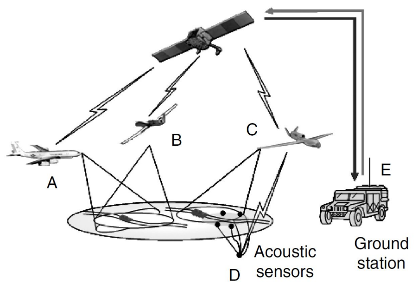
Umut Orguner
umut@metu.edu.tr
room: EZ-12 tel: 4425

Department of Electrical & Electronics Engineering
Middle East Technical University
Ankara, Turkey

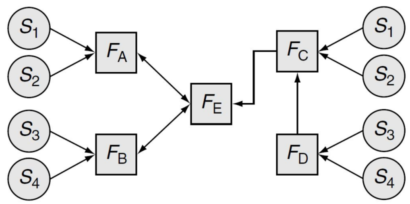
Lecture Outline

- Multiple Sensor Tracking Architectures
- Multiple Sensor Tracking Problems
- Track Association
- Track Fusion

Multi Sensor Architectures



Network-centric operation



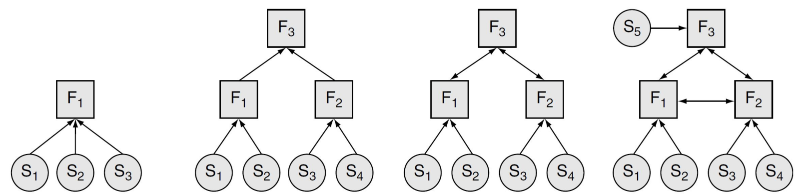
Hierarchical fusion with feedback

Hierarchical fusion without feedback

Corresponding fusion network

Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures



Centralized fusion

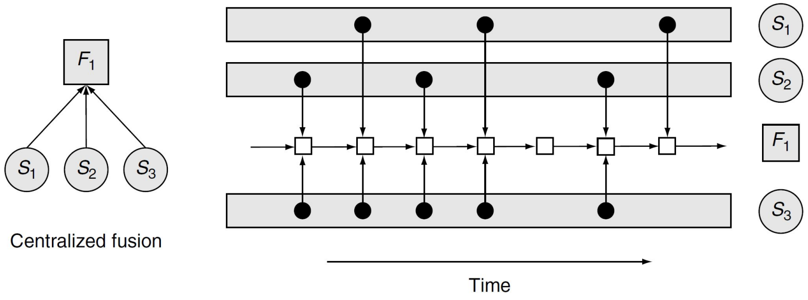
Hierarchical fusion without feedback

Hierarchical fusion with feedback

Distributed fusion

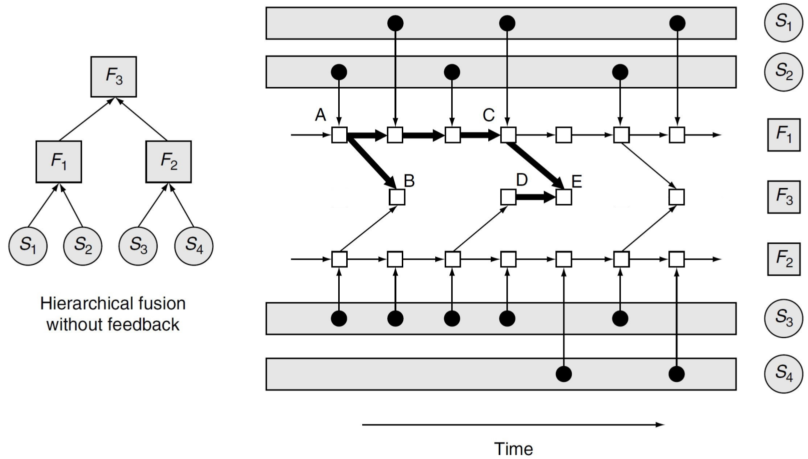
Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Centralized



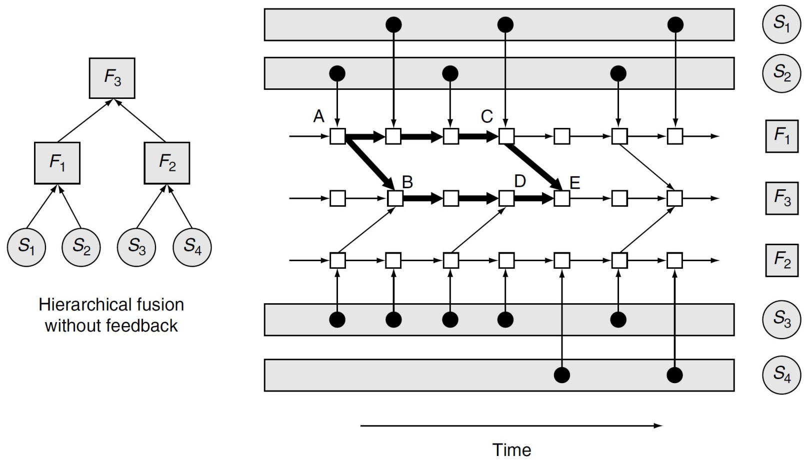
Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Hierarchical without Memory



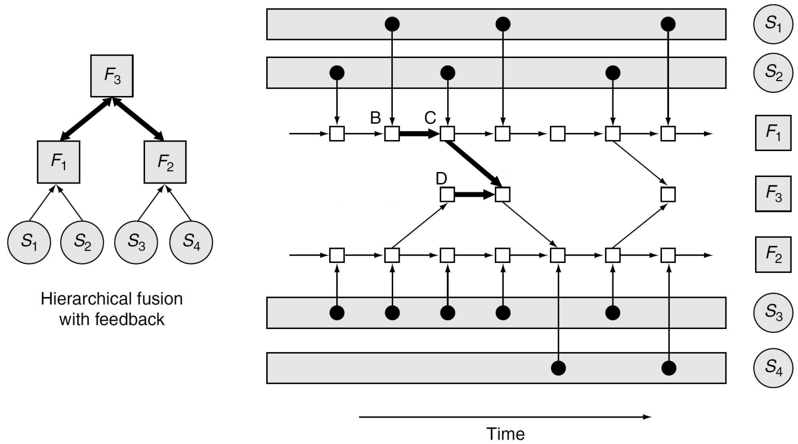
Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Hierarchical with Memory



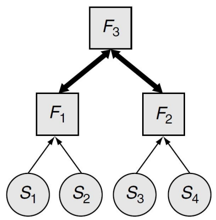
Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Hierarchical with Feedback without Memory

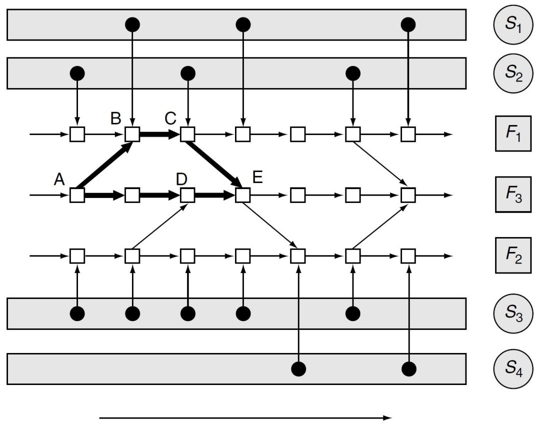


Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Hierarchical with Feedback with Memory



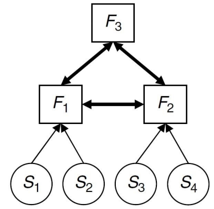
Hierarchical fusion
with feedback



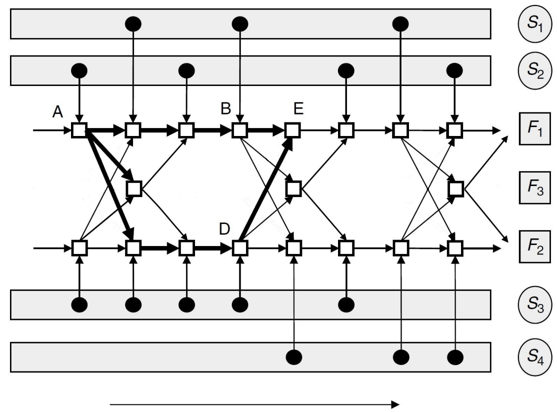
Time

Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Decentralized without Memory



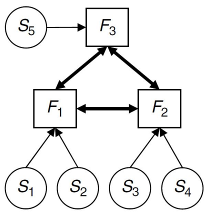
Distributed fusion



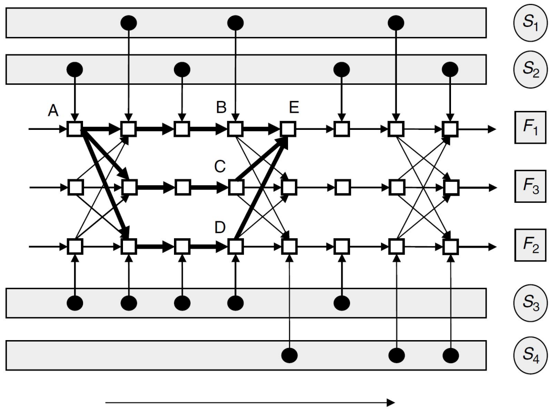
Time

Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Decentralized with Memory



Distributed fusion



Time

Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Multi Sensor Architectures: Pros & Cons

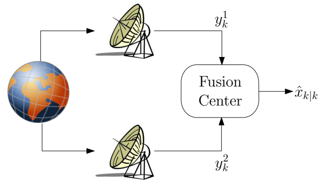
- The traditional centralized architecture gives optimal performance but
  - Requires high bandwidth communications.
  - Requires powerful processing resources at the fusion center.
  - There is a single point of failure and hence reliability is low.
- For distributed architectures
  - Communications can be reduced significantly by communicating tracks less often.
  - Computational resources can be distributed to different nodes
  - Higher survivability.
  - It is a necessity for legacy systems e.g. radars sometimes might not supply raw data.

Problems in Multi Sensor TT

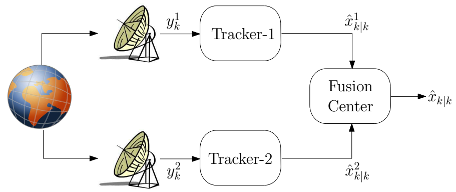
- **Registration:** Coordinates (both time and space) of different sensors or fusion agents must be aligned.
- **Bias:** Even if the coordinate axes are aligned, due to the transformations, biases can result. These have to be compensated.
- **Correlation:** Even if the sensors are independently collecting data, processed information to be fused can be correlated.
- **Rumor propagation:** The same information can travel in loops in the fusion network to produce fake information making the overall system overconfident. This is actually a special case of correlation.
- **Out of sequence measurements:** Due to delayed communications between local agents, sometimes measurements belonging to a target whose more recent measurement has already been processed, might arrive to a fusion center.

Correlation

Centralized Case



Decentralized Case



Suppose the fusion center have the prediction $\hat{x}_{k|k-1}$ in both cases.

Centralized Case

$$\begin{bmatrix} y_k^1 \\ y_k^2 \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x_k + \begin{bmatrix} e_k^1 \\ e_k^2 \end{bmatrix}$$

where e_k^1 and e_k^2 are independent.

Decentralized Case

$$\begin{bmatrix} \hat{x}_{k|k}^1 \\ \hat{x}_{k|k}^2 \end{bmatrix} = \begin{bmatrix} I \\ I \end{bmatrix} x_k - \begin{bmatrix} \tilde{x}_k^1 \\ \tilde{x}_k^2 \end{bmatrix}$$

Correlation

Suppose the target follows the dynamics

$$x_k = Ax_{k-1} + w_k$$

and the ith sensor measurement is given as

$$y_k^i = C_i x_k + e_k^i$$

Then with the KF equations

$$\begin{aligned} \hat{x}_{k|k}^i =& A\hat{x}_{k-1|k-1}^i + K_k^i(y_k^i - C_iA\hat{x}_{k-1|k-1}^i) \\ =& A\hat{x}_{k-1|k-1}^i + K_k^iC_i(x_k - A\hat{x}_{k-1|k-1}^i) + K_k^ie_k^i \\ =& A\hat{x}_{k-1|k-1}^i + K_k^iC_iA(x_{k-1} - \hat{x}_{k-1|k-1}^i) + K_k^iC_iw_k + K_k^ie_k^i \end{aligned}$$

Correlation

Define $\tilde{x}_k^i \triangleq x_k - \hat{x}_{k|k}^i$, then

$$\begin{aligned} \tilde{x}_k^i =& x_k - A\hat{x}_{k-1|k-1}^i - K_k^iC_iA(x_{k-1} - \hat{x}_{k-1|k-1}^i) - K_k^iC_iw_k - K_k^ie_k^i \\ =& Ax_{k-1} + w_k - A\hat{x}_{k-1|k-1}^i - K_k^iC_iA(x_{k-1} - \hat{x}_{k-1|k-1}^i) \\ & - K_k^iC_iw_k - K_k^ie_k^i \\ =& (I - K_k^iC_i)A\tilde{x}_{k-1}^i + (I - K_k^iC_i)w_k - K_k^ie_k^i \end{aligned}$$

Hence

$$\begin{aligned} \tilde{x}_k^i =& (I - K_k^iC_i)A\tilde{x}_{k-1}^i + (I - K_k^iC_i)w_k - K_k^ie_k^i \\ \tilde{x}_k^j =& (I - K_k^jC_j)A\tilde{x}_{k-1}^j + (I - K_k^jC_j)w_k - K_k^je_k^j \end{aligned}$$

We can calculate the correlation matrix $\Sigma_k^{ij} \triangleq E(\tilde{x}_k^i\tilde{x}_k^{jT})$ as

$$\Sigma_k^{ij} = (I - K_k^iC_i)A\Sigma_{k-1}^{ij}A^T(I - K_k^jC_j)^T + (I - K_k^iC_i)Q(I - K_k^jC_j)^T$$

Correlation

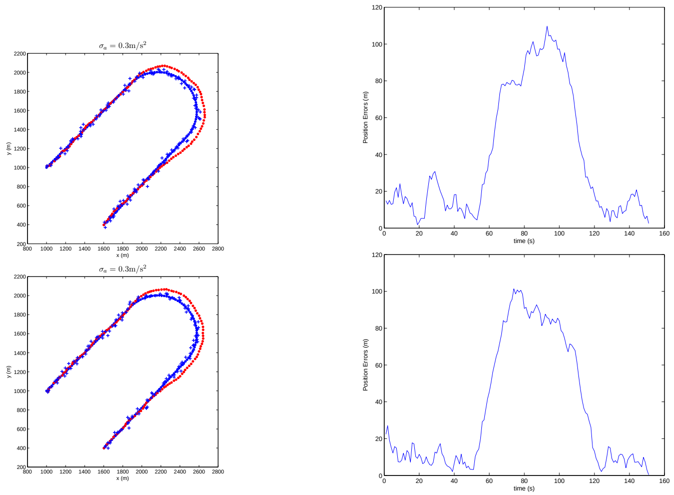
Assuming that $\Sigma_0^{ij} = 0$, we can calculate the correlation between the estimation errors of the local trackers recursively as

$$\Sigma_k^{ij} = (I - K_k^i C_i) A \Sigma_{k-1}^{ij} A^T (I - K_k^j C_j)^T + (I - K_k^i C_i) Q (I - K_k^j C_j)^T$$

- This necessitates that the fusion center knows the individual Kalman gains K_k^i and K_k^j of the local trackers which is not very practical.
- Assuming that the errors are independent is not a good idea either.
- Neglecting the correlation makes the resulting estimates overconfident i.e., very small covariances meaning that too small gates and smaller Kalman gains.
- When $Q = 0$, $\Sigma_k^{ij} = 0$, i.e., no correlation when no process noise.

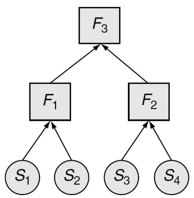
Correlation Illustration

Maneuvers make this problem more dominant and visible.

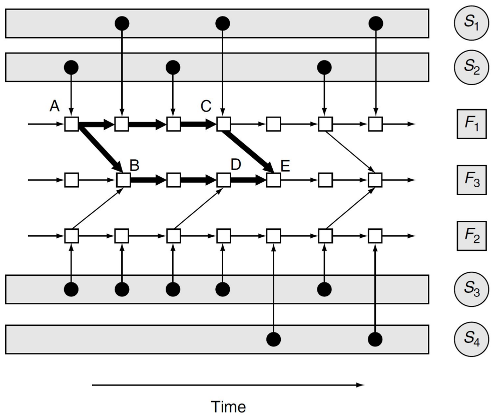


Rumor Propagation

Hierarchical Case: Rumor always flows to the fusion center



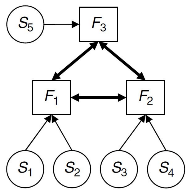
Hierarchical fusion without feedback



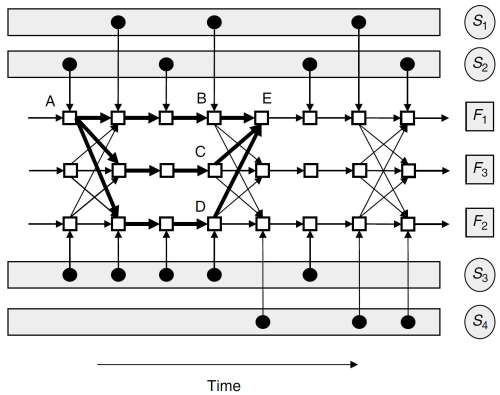
Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Rumor Propagation

Decentralized case: Rumor propagates everywhere.



Distributed fusion



Figures taken from: M.E. Liggins and Kuo-Chu Chang
"Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture,"
Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.

Track Association: Testing

Test for Track Association [Bar-Shalom (1995)]:

- Two estimates $\hat{x}^i_{k|k}$, $\hat{x}^j_{k|k}$ and the covariances $\Sigma^i_{k|k}$, $\Sigma^j_{k|k}$ are given from ith and jth local systems.
- We calculate the difference vector Δ^{ij}_k

$$\Delta^{ij}_k \triangleq \hat{x}^i_{k|k} - \hat{x}^j_{k|k}$$

- Then we calculate covariance $\Gamma^{ij}_k \triangleq E(\Delta^{ij}_k \Delta^{ijT}_k)$ as

$$\Gamma^{ij}_k = \Sigma^i_{k|k} + \Sigma^j_{k|k} - \Sigma^{ij}_k - \Sigma^{ijT}_k$$

- Then test statistics D^{ij}_k calculated as

$$D^{ij}_k = \Delta^{ijT}_k (\Gamma^{ij}_k)^{-1} \Delta^{ij}_k \lessgtr \gamma^{ij}_k$$

  can be used for checking track association.

Track Association

What about the cross covariance Σ^{ij}_k?

- Simple method is to set it $\Sigma^{ij}_k = 0$.
- It can be calculated using Kalman gains if they are transmitted to the fusion center.

Approximation for cross covariance from [Bar-Shalom (1995)]:

- The following cross-covariance approximation was proposed:

$$\Sigma^{ij}_k \approx \rho \left(\Sigma^i_{k|k} \cdot * \Sigma^j_{k|k} \right)^{\cdot \frac{1}{2}}$$

  where multiplication and power operations are to be done element-wise. For negative numbers, square root must be taken on the absolute value and sign must be kept.
- The value of ρ must be adjusted experimentally. $\rho = 0.4$ was suggested for $2D$ tracking.

Track Association

Method proposed by [Blackman (1999)] for two local agents

- Suppose local agent i and j have N^i_T and N^j_T tracks respectively.
- A track association hypothesis θ_k between local agents i and j can be represented as a $N^i_T \times N^j_T$-size binary matrix $Z = [z_{mn} \in \{0, 1\}]$ such that

$$z_{mn} = \begin{cases} 1, & \text{If track } m \text{ of local agent } i \text{ is associated with track } n \text{ of local agent } j \\ 0, & \text{otherwise} \end{cases}$$

- Note that the constraints

$$\sum_{m=1}^{N^i_T} z_{mn} \leq 1 \quad \forall n \quad \text{and} \quad \sum_{n=1}^{N^j_T} z_{mn} \leq 1 \quad \forall m$$

  must be satisfied for a valid track association hypothesis.

Track Association

Method proposed by [Blackman (1999)] for two local agents

- Define the quantities
  - β_T: target density (number of targets/state-space volume)
  - $P_{i \in j}$: probability that local agent i has a track in the common field of view with local agent j given that there is a target there.
  - β^i_{FT}: False track density of the tracker of local agent i (same unit as β_T).
- Then the probability of a track association hypothesis is given by

$$P(\theta_k) \propto (\beta^i_{NA})^{N^i_{NA}} (\beta^j_{NA})^{N^j_{NA}} \prod_{\{m,n|z_{mn}=1\}} \beta_T P_{i \in j} P_{j \in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)$$

  where
  - $\beta^i_{NA} = \beta_T P_{i \in j}(1 - P_{j \in i}) + \beta^i_{FT}$ and $\beta^j_{NA} = \beta_T P_{j \in i}(1 - P_{i \in j}) + \beta^j_{FT}$
  - $N^i_{NA} \triangleq N^i_T - \sum_{m=1}^{N^i_T} \sum_{n=1}^{N^j_T} z_{mn}$ and $N^j_{NA} \triangleq N^j_T - \sum_{m=1}^{N^i_T} \sum_{n=1}^{N^j_T} z_{mn}$

Track Association

Method proposed by [Blackman (1999)] for two local agents

$$P(\theta_k) \propto (\beta^i_{NA})^{N^i_{NA}} (\beta^j_{NA})^{N^j_{NA}} \prod_{\{m,n|z_{mn}=1\}} \beta_T P_{i\in j} P_{j\in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)$$

Divide by the constant $(\beta^j_{NA})^{N^j_T}$

$$P(\theta_k) \propto (\beta^i_{NA})^{N^i_{NA}} \prod_{\{m,n|z_{mn}=1\}} \frac{\beta_T P_{i\in j} P_{j\in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)}{\beta^j_{NA}}$$

Maximizing this probability is equivalent to maximizing $\log$ of it.

$$\log P(\theta_k) = N^i_{NA} \log \beta^i_{NA} + \sum_{\{m,n|z_{mn}=1\}} \log \frac{\beta_T P_{i\in j} P_{j\in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)}{\beta^j_{NA}} + C$$

Track Association: Assignment Problem

Method proposed by [Blackman (1999)] for two local agents

$$\log P(\theta_k) = N^i_{NA} \log \beta^i_{NA} + \sum_{\{m,n|z_{mn}=1\}} \log \frac{\beta_T P_{i\in j} P_{j\in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)}{\beta^j_{NA}} + C$$

Form the assignment matrix:

$\mathcal{A}_{ij}$	T^j_1	T^j_2	T^j_3	T^j_4	NA$_1$	NA$_2$	NA$_3$
T^i_1	ℓ_{11}	ℓ_{12}	ℓ_{13}	ℓ_{14}	$\log \beta^i_{NA}$	$\times$	$\times$
T^i_2	ℓ_{21}	ℓ_{22}	ℓ_{23}	ℓ_{24}	$\times$	$\log \beta^i_{NA}$	$\times$
T^i_3	ℓ_{31}	ℓ_{32}	ℓ_{33}	ℓ_{34}	$\times$	$\times$	$\log \beta^i_{NA}$

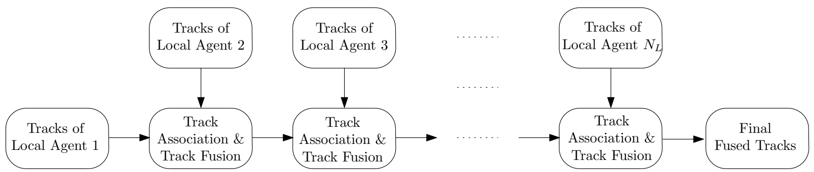
where $\ell_{mn} \triangleq \log \frac{\beta_T P_{i\in j} P_{j\in i} \mathcal{N}(\hat{x}^m_{k|k} - \hat{x}^n_{k|k}; 0, \Gamma^{mn}_k)}{\beta^j_{NA}}$.
Then, use $\texttt{auction}(\mathcal{A}_{ij})$ to get track association decisions.

Track Association

Track association for more than two local agents.

- One way is to solve *multi dimensional assignment problem*.
- The simpler way is to do the so-called **sequential pairwise track association**.

Suppose we have N_L local agents whose tracks need to be fused. Then, we order the local agents according to some criteria e.g. accuracy, priority, etc.



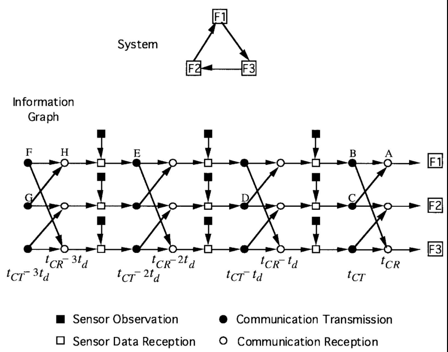
Track Fusion: Independence Assumption

Once we associate two tracks, we have to fuse them to obtain a fused track. This is called as **track fusion**.
Consider the track fusion at point A assuming $t_{CR} = t_{CT} = t$.

- Independence assumption gives

$$(\Sigma^A_t)^{-1} = (\Sigma^B_t)^{-1} + (\Sigma^C_t)^{-1}$$
$$(\Sigma^A_t)^{-1}\hat{x}^A_t = (\Sigma^B_t)^{-1}\hat{x}^B_t + (\Sigma^C_t)^{-1}\hat{x}^C_t$$

- This is simplistic and expected to give very bad results here.
- This is also called as *naive fusion*.



Track Fusion: Optimal Solution

Consider the track fusion at point A assuming $t_{CR} = t_{CT} = t$.

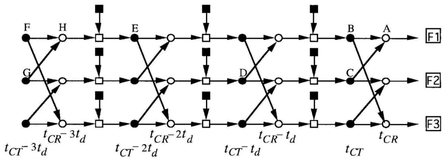
- Optimal solution

$$(\Sigma_t^A)^{-1} = \left(I - (\Sigma_t^C)^{-1}\Sigma_t^{CB}\right)\Delta_B^{-1} + \left(I - (\Sigma_t^B)^{-1}\Sigma_t^{BC}\right)\Delta_C^{-1}$$
$$(\Sigma_t^A)^{-1}\hat{x}_t^A = \left(I - (\Sigma_t^C)^{-1}\Sigma_t^{CB}\right)\Delta_B^{-1}\hat{x}_t^B + \left(I - (\Sigma_t^B)^{-1}\Sigma_t^{BC}\right)\Delta_C^{-1}\hat{x}_t^C$$

where

$$\Delta_B \triangleq \Sigma_t^B - \Sigma_t^{BC}(\Sigma_t^C)^{-1}\Sigma_t^{CB} \qquad \Delta_C \triangleq \Sigma_t^C - \Sigma_t^{CB}(\Sigma_t^B)^{-1}\Sigma_t^{BC}$$

- This is very difficult to compute in a scalable way for variable networks (no fixed-structure).



Track Fusion: Channel Filter

Consider the track fusion at point A assuming $t_{CR} = t_{CT} = t$.

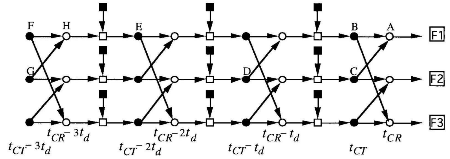
- Channel filter, equivalent measurements, tracklets etc.

$$(\Sigma_t^A)^{-1} = (\Sigma_t^B)^{-1} + (\Sigma_t^C)^{-1} - (\Sigma_{t|t-t_d}^D)^{-1}$$
$$(\Sigma_t^A)^{-1}\hat{x}_t^A = (\Sigma_t^B)^{-1}\hat{x}_t^B + (\Sigma_t^C)^{-1}\hat{x}_t^C - (\Sigma_{t|t-t_d}^D)^{-1}\hat{x}_{t|t-t_d}^D$$

- One can define $\hat{z}_t^C$ and Z_t^C, which are called equivalent measurements or tracklets in the literature, as

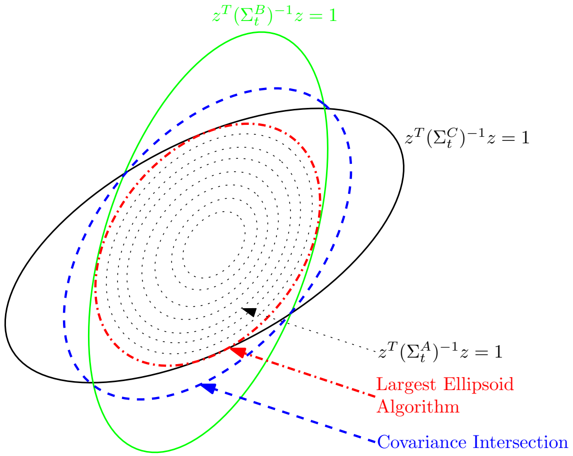
$$(Z_t^C)^{-1} \triangleq (\Sigma_t^C)^{-1} - (\Sigma_{t|t-t_d}^D)^{-1}$$
$$(Z_t^C)^{-1}\hat{z}_t^C \triangleq (\Sigma_t^C)^{-1}\hat{x}_t^C - (\Sigma_{t|t-t_d}^D)^{-1}\hat{x}_{t|t-t_d}^D$$

- Transmitting these quantities instead from a local agent, one can use the independent track fusion formulas.



Track Fusion

Illustration of Correlation Independent Schemes.



Track Fusion: LEA Algorithm

Largest ellipsoid algorithm or safe fusion: Suppose we have local estimates $\hat{x}_t^B$, Σ_t^B and $\hat{x}_t^C$, Σ_t^C.

- Find SVD of $\Sigma_t^B = U_1\Lambda_1U_1^T$
- Define the transformation $\mathcal{T}_1 = \Lambda_1^{-1/2}U_1^T$
- Transform Σ_t^C with $\mathcal{T}_1$ and define $P_C = \mathcal{T}_1\Sigma_t^C\mathcal{T}_1^T$.
- Find SVD of $P_C = U_2\Lambda_2U_2^T$.
- Define the transformation $\mathcal{T}_2 = U_2^T\mathcal{T}_1$
- Transform $\hat{x}_t^B$, Σ_t^B and $\hat{x}_t^C$, Σ_t^C with $\mathcal{T}_2$.

$$\hat{z}_t^B = \mathcal{T}_2\hat{x}_t^B \quad \text{and} \quad \hat{z}_t^C = \mathcal{T}_2\hat{x}_t^C$$

$$Z_t^B = \mathcal{T}_2\Sigma_t^B\mathcal{T}_2^T = I_{n_x} \quad \text{and} \quad Z_t^C = \mathcal{T}_2\Sigma_t^C\mathcal{T}_2^T = \Lambda_2$$

- ...

Track Fusion: LEA Algorithm

Largest ellipsoid algorithm continued:

- ...
- Define set of indices $\mathcal{I} = \{i | 1 \leq i \leq n_x, [\Lambda_2]_{ii} < 1\}$
- Find vector $\hat{z}_t^A$ and covariance Z_t^A as

$$[\hat{z}_t^A]_i \triangleq \begin{cases} [\hat{z}_t^B]_i & i \notin \mathcal{I} \\ [\hat{z}_t^C]_i & i \in \mathcal{I} \end{cases}, \quad [Z_t^A]_{ij} \triangleq \begin{cases} [Z_t^B]_{ii} & i = j, i \notin \mathcal{I} \\ [Z_t^C]_{ii} & i = j, i \in \mathcal{I} \\ 0 & i \neq j \end{cases}$$

- Find fused estimate and covariance

$$\hat{x}_t^A = \mathcal{T}_2^{-1} \hat{z}_t^A \qquad \Sigma_t^A = \mathcal{T}_2^{-1} Z_t^A \mathcal{T}_2^{-T}.$$

Track Fusion: CI

Covariance intersection

- Define

$$(\Sigma_t^A(w))^{-1} = w(\Sigma_t^B)^{-1} + (1-w)(\Sigma_t^C)^{-1}$$

- Find

$$w^* = \arg \min_{w \in [0,1]} |\Sigma_t^A(w)|$$

  using optimization.
- Then the fused estimate and covariance are given as

$$(\Sigma_t^A)^{-1} = w^*(\Sigma_t^B)^{-1} + (1-w^*)(\Sigma_t^C)^{-1}$$
$$(\Sigma_t^A)^{-1}\hat{x}_t^A = w^*(\Sigma_t^B)^{-1}\hat{x}_t^B + (1-w^*)(\Sigma_t^C)^{-1}\hat{x}_T^C$$

Track Fusion

According to the recent work (recommended)

> *K. C. Chang, Chee-Yee Chong and S. Mori, "On scalable distributed sensor fusion," Proceedings of 11th International Conference on Information Fusion, Jul. 2008.*

channel filter seems to be the best algorithm for track fusion in terms of

- scalability;
- estimation errors;
- and memory.

References

- M. E. Liggins II, Chee-Yee Chong, I. Kadar, M. G. Alford, V. Vannicola and V. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol.85, no.1, pp. 95-107, Jan. 1997.
- M. E. Liggins and Kuo-Chu Chang, "Distributed fusion architectures, algorithms, and performance within a network-centric architecture," Ch.17, Handbook of Multisensor Data Fusion: Theory and Practice, Taylor & Francis, Second Edition, 2009.
- K. C. Chang, Chee-Yee Chong and S. Mori, "On scalable distributed sensor fusion," *Proceedings of 11th International Conference on Information Fusion*, Jul. 2008.

References

S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles, Techniques*. Storrs, CT: YBS Publishing, 1995.