DISTRIBUTED SEQUENTIAL LIKELIHOOD RATIO TESTING
FOR TRACK EXISTENCE DECISIONS

2024 AESS Virtual DL Webinar
Dr. Felix Govaers, University of Bonn / Fraunhofer FKIE
Multi Sensor Fusion with limited Communication

Distributed Kalman filtering has significant advantages in a multi sensor scenario:

- Save bandwidth (preprocessing)
- Distributed calculation
- Full information available at arbitrary instants of time
Distributed / Federated / Naïve Kalman Filter
Prediction – Filtering – Cycle

Initialization
\[ p(x_0) \propto \prod_{s=1}^{S} N(x_0; x_0^s, P_0^s) \]

Prediction
\[ p(x_k|z^k) \propto \prod_{s=1}^{S} N(x_k; x_k^s, P_k^s) \]

Filtering
\[ Z_{k}^{1}, \ldots, Z_{k}^{S} \]

\[ x_0^{s} \]
\[ P_0^{s} \]
\[ x_0 \]
\[ x_{0|0} \]
\[ P_{0|0} \]
\[ N \]
Product Representation

The posterior of $S$ **mutually independent** estimates is given by

$$ p(x_k | x_{1:k}, \ldots, x_{S:k}) \propto \prod_{s=1}^{S} N(x_k; x_{s:k}^s, P_{s:k}^s). $$

$$ = N(x_k; x_{k:k}, P_{k:k}) $$

Naïve Fusion (Convex Combination) is **exact** if and only if cross-covariances are zero.

$$ x_{k:k} = P_{k:k} \left( \sum_{s=1}^{S} (P_{k:k}^s)^{-1} x_{s:k}^s \right) $$

$$ P_{k:k} = \left( \sum_{s=1}^{S} (P_{k:k}^s)^{-1} \right)^{-1} $$
Product Representation Prediction w/ Relaxed Evolution Model

The prior is calculated by

\[
p(x_k | Z^{k-1}) = \int dx_{k-1} p(x_k, x_{k-1} | Z^{k-1})
= \int dx_{k-1} p(x_k | x_{k-1}, Z^{k-1}) \cdot p(x_{k-1} | Z^{k-1})
= \int dx_{k-1} p(x_k | x_{k-1}) \cdot p(x_{k-1} | Z^{k-1})
\]

\[
\mathcal{N}(x_k; F_k | x_{k-1}, Q_k) \propto \exp \left\{ -\frac{1}{2} (x_k - F_k x_{k-1})^T Q_k^{-1} (x_k - F_k x_{k-1}) \right\}
\propto \exp \left\{ -\frac{1}{2} (x_k - F_k x_{k-1})^T \frac{1}{S} Q_k^{-1} (x_k - F_k x_{k-1}) \right\}
\propto \exp \left\{ -\frac{1}{2} (x_k - F_k x_{k-1})^T (S Q_k)^{-1} (x_k - F_k x_{k-1}) \right\}^S
\propto \mathcal{N}(x_k; F_k x_{k-1}, S Q_k)^S
\]

\[
p(x_k | Z^{k-1}) \propto \int dx_{k-1} \prod_{s=1}^S \mathcal{N}(x_k; F_k x_{k-1}, S Q_k) \mathcal{N}(x_{k-1}; x_{k-1}^s, P_{k-1}^s)
\]

given in product representation!
Product Formula (1st & 2nd Formulation)

It holds that

\[ N(x; \bar{y}, \bar{P}) \cdot N(z; \bar{z}, S) = N(x; y, P) \cdot N(z; Hx, R) \]

where

\[
\begin{align*}
\bar{P} &= \begin{cases} 
P - WSW^\top \\
(P^{-1} + H^\top R^{-1}H)^{-1} 
\end{cases} & S &= HPH^\top + R \\
\bar{y} &= \begin{cases} 
y + W\nu \\
\bar{P}(P^{-1}y + H^\top R^{-1}z) 
\end{cases} & W &= PH^\top S^{-1} \\
\nu &= z - Hy
\end{align*}
\]
Apply Product Formula (2\textsuperscript{nd} version)

\[
\mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k|k-1}\mathbf{x}_{k-1}, S\mathbf{Q}_{k|k-1}) \cdot \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{x}^s_{k-1|k-1}, \mathbf{P}^s_{k-1|k-1}) \\
= \mathcal{N}(\mathbf{x}_k; \mathbf{x}^s_{k|k-1}, \mathbf{P}^s_{k|k-1}) \cdot \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{y}^s, \mathbf{Y}^s)
\]

where

\[
\mathbf{x}^s_{k|k-1} = \mathbf{F}_{k|k-1}\mathbf{x}^s_{k-1|k-1}
\]

\[
\mathbf{P}^s_{k|k-1} = \mathbf{F}_{k|k-1}\mathbf{P}^s_{k-1|k-1}\mathbf{F}^\top_{k|k-1} + S\mathbf{Q}_{k|k-1}
\]

\[
\mathbf{y}^s = \mathbf{y}^s((\mathbf{P}^s_{k|k-1})^{-1}\mathbf{x}^s_{k|k-1} + \mathbf{F}^\top_{k|k-1}(S\mathbf{Q}_{k|k-1})^{-1}\mathbf{x}_k)
\]

\[
\mathbf{Y}^s = ((\mathbf{P}^s_{k|k-1})^{-1} + \mathbf{F}^\top_{k|k-1}(S\mathbf{Q}_{k|k-1})^{-1}\mathbf{F}_{k|k-1})^{-1}
\]

\[
p(\mathbf{x}_k|\mathcal{Z}^{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}(\mathbf{x}_k; \mathbf{x}^s_{k|k-1}, \mathbf{P}^s_{k|k-1}) \int d\mathbf{x}_{k-1} \prod_{s=1}^{S} \{ \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{y}^s, \mathbf{Y}^s) \}
\]
Theorem 3 Let \( P_{k|k-1} \) be a tight bound for the fusion of 1
\( Z_1 \),...,
\( Z_s \).

\[
p(x_k|Z^{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x^s_{k|k-1}, P^s_{k|k-1}) \int \text{d}x_{k-1} \prod_{s=1}^{S} \mathcal{N}(x_{k-1}; y^s, Y^s)
\]

\[
\approx \prod_{s=1}^{S} \mathcal{N}(x_k; x^s_{k|k-1}, P^s_{k|k-1})
\]

The **Federated Kalman Filter** ignores the integral:

\[
p(x_k|Z^{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x^s_{k|k-1}, P^s_{k|k-1})
\]
Federated Kalman Filter Filtering

For the filtering step, we use the fact that measurement noise of the sensors is mutually independent:

\[
p(Z_k|x_k) \propto \prod_{s=1}^{S} p(z_k^s|x_k).
\]

Using the linear Gaussian assumption, we obtain for the fused posterior:

\[
p(x_k|Z^k) \propto \prod_{s=1}^{S} \mathcal{N}(z_k^s; H_k^s x_k, R_k^s) \mathcal{N}(x_k; \tilde{x}_{k|k-1}^s, \tilde{P}_{k|k-1}).
\]

And directly obtain

\[
p(x_k|Z^k) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k}^s, P_{k|k}^s),
\]

\[
x_{k|k}^s = \tilde{x}_{k|k-1}^s + W_{k|k-1}^s \left( z_k^s - H_k^s \tilde{x}_{k|k-1}^s \right)
\]

\[
W_{k|k-1}^s = \tilde{P}_{k|k-1} H_k^s \tilde{S}_{k|k-1}^{-1}
\]

\[
S_{k|k-1}^s = H_k^s \tilde{P}_{k|k-1} H_k^s + R_k^s
\]

\[
P_{k|k}^s = \tilde{P}_{k|k-1} - W_{k|k-1}^s S_{k|k-1}^s W_{k|k-1}^s \tilde{P}_{k|k-1}.
\]
DISTRIBUTED KALMAN FILTER
Distributed Kalman Filter

Initialization

\[ p(x_0) \propto \prod_{s=1}^{S} \mathcal{N}(x_0; x_{0|0}^s, P_{0|0}^s) \]

Globalization

\[ p(x_k|z^{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k-1}^s, \hat{P}_{k|k-1}) \]

Prediction

\[ p(x_k|z^k) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k}^s, P_{k|k}^s) \]

Filtering

\( z_{k}, \ldots, z_{k}^{S} \)
Globalized Covariance Solution

Exact solution by ‘globalizing’ the estimate covariance [1]:

\[
p(x_{k-1}|Z_{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}\left(x_{k-1}; x_{k-1|k-1}^{s}, P_{k-1|k-1}^{s}\right)
\]

\[
\propto \mathcal{N}\left(x_{k-1}; P_{k-1|k-1}\right)
\]

\[
\propto \mathcal{N}\left(x_{k-1}; P_{k-1|k-1}\sum_{s=1}^{S} (P_{k-1|k-1}^{s})^{-1} x_{k-1|k-1}^{s}, P_{k-1|k-1}\right)
\]

\[
\propto \mathcal{N}\left(x_{k-1}; \frac{1}{S} \sum_{s=1}^{S} \mathcal{N}\left(x_{k-1}; P_{k-1|k-1}(P_{k-1|k-1}^{s})^{-1} x_{k-1|k-1}^{s}, P_{k-1|k-1}\right)\right)
\]

\[
\propto \prod_{s=1}^{S} \mathcal{N}\left(x_{k-1}; \tilde{x}_{k-1|k-1}^{s}, \tilde{P}_{k-1|k-1}\right)
\]

<table>
<thead>
<tr>
<th>Sensor globalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{k-1}^i = P_{k-1</td>
</tr>
<tr>
<td>$P_{k-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensor prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s,k</td>
</tr>
<tr>
<td>$P_{s,k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensor filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s,k</td>
</tr>
<tr>
<td>$W_{s,k</td>
</tr>
<tr>
<td>$S_{k</td>
</tr>
<tr>
<td>$P_{s,k} = P_{s,k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fusion center</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{k</td>
</tr>
<tr>
<td>$P_{k</td>
</tr>
</tbody>
</table>

**Distributed Kalman Filter**

**Federated Kalman Filter**

**Naïve Fusion**
Target Existence Decision – Distributed

- For automated decision making with respect to track existence, different categories of algorithms exist:
  - **Centralized**: Send all measurements to the Fusion Center (FC) and decide.
  - **Decentralized**: Decide locally on each sensor node, send the decisions to all connected neighbors and fuse the received decisions.
  - **Distributed**: Fuse the local data and compute parameters which are combined in a distinguished fusion center to make a global decision.
Target Existence Decision – Distributed
Likelihood Ratio Test

The Sequential Likelihood Ratio (LR) test is a statistically optimal algorithm to decide between two hypotheses:

- $h_1$: there is a target.
- $h_0$: there is no target.

A decision can be made based on two thresholds $A$ and $B$.

$$LR(k) = \frac{p(h_1|\mathcal{Z}^k)}{p(h_0|\mathcal{Z}^k)}$$

- $LR(k) < A$: accept $h_0$, i.e. delete track
- $LR(k) > B$: accept $h_1$, i.e. confirm track
- $A < LR(k) < B$: continue processing.
Recursive computation of the LR

According to Bayes, one has

\[
\text{LR}(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)}
\]

where

\[
p(Z^k|h_i) = p(Z_k|Z^{k-1}, h_i) p(Z^{k-1}|h_i)
\]

\[
= \int dx_k p(Z_k|x_k, h_1) p(x_k|Z^{k-1}, h_1) p(Z^{k-1}|h_1)
\]

\[
\text{LR}(k) = \text{LR}(k - 1) \cdot \Lambda(k)
\]

\[
\Lambda(k) = \int dx_k p(Z_k|x_k, h_1) p(x_k|Z^{k-1}, h_1) \frac{p(Z^{k-1}|h_1)}{p(Z_k|h_0)}
\]
Distributed Kalman Filter (DKF)

The Distributed Kalman Filter (DKF) achieves a product representation of local estimate parameters also, if process noise is present:

\[
p(x_k|\mathcal{Z}^{k-1}) \propto \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k-1}^s, P_{k|k-1}^s)
\]

- The DKF computes the exact product representation but cannot be applied when the local covariances are data dependent.
- The FKF and Naïve Fusion yield approximately a product representation:

\[
p(x_k|\mathcal{Z}^{k-1}) \approx \frac{1}{c_{k|k-1}} \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k-1}^s, P_{k|k-1}^s)
\]
Distributed Sequential Likelihood Ratio Test

The LR score at time $k$ is given by

$$LR(k) = LR(k - 1) \cdot \Lambda(k)$$

$$\Lambda(k) = \int dx_k \frac{p(Z_k|x_k, h_1) \cdot p(x_k|Z^{k-1}, h_1)}{p(Z_k|h_0)}$$

where

$$p(Z_k|h_0) = |\text{FoV}|^{-m} p_F(m)$$

$$p(Z_k|x_k, h_1) = (|\text{FoV}|^{-m} p_F(m)) (1 - p_D) + \frac{p_D}{\rho_F} \sum_{j=1}^{m} \mathcal{N}(z_j; H_k x_k, R_k)$$

$$p(x_k|Z^{k-1}) = \frac{1}{c_{k|k-1}} \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k-1}^s, P_{k|k-1}^s)$$

Now, the normalization constant is important!
DKF Normalization Constant

The normalization constant is given by

\[ c_{k|k-1} = \int dx_k \prod_{s=1}^{S} \mathcal{N}(x_k; x_{k|k-1}^s, P_{k|k-1}^s). \]

Algebraic manipulations yield

\[ c_{k|k-1} = \prod_{s=1}^{S-1} \mathcal{N}(x_{k|k-1}^{s+1}; x_{k|k-1}^{(1:s)}, P_{k|k-1}^{(1:s)} + P_{k|k-1}^s) \]

\[ x_{k|k-1}^{(1:s)} = P_{k|k-1}^{(1:s)} \sum_{i=1}^{s} (P_{k|k-1}^i)^{-1} x_{k|k-1}^i, \]

\[ P_{k|k-1}^{(1:s)} = \left( \sum_{i=1}^{s} (P_{k|k-1}^i)^{-1} \right)^{-1}. \]
Sequential LR Update for DKF

The updating factor of the LR is given by

$$\Lambda(k) = \frac{1}{c_{k|k-1}} \int d\mathbf{x}_k \prod_{s=1}^{S} \left\{ \left(1 - p_D\right) + \frac{p_D}{\rho F} \sum_{j=1}^{m_s} \mathcal{N}(\mathbf{z}_{k,s}^j; \mathbf{H}_{k,s}^j \mathbf{x}_k, \mathbf{R}_{k,s}^j) \right\} \cdot \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k-1}^s, \mathbf{P}_{k|k-1}^s) \right\}$$

An application of the product formula yields

$$\Lambda(k) = \frac{1}{c_{k|k-1}} \int d\mathbf{x}_k \prod_{s=1}^{S} \sum_{j=0}^{m_s} \mathbf{p}_{j,s}^* \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}^j, \mathbf{P}_{k|k}^j)$$
Normalization and Moment Matching

The normalized weights are given by

\[ p^{j,s} = \frac{p^{* j,s}}{\bar{p}^s} \quad \text{and} \quad \bar{p}^s = \sum_{j=0}^{m_s} p^{* j,s}. \]

Therefore:

\[ \sum_{j=0}^{m_s} p^{* j,s} \mathcal{N}(\mathbf{x}_{k; k|k}^{j,s}, \mathbf{P}_{k|k}^{j,s}) = \bar{p}^s \sum_{j=0}^{m_s} p^{j,s} \mathcal{N}(\mathbf{x}_{k; k|k}^{j,s}, \mathbf{P}_{k|k}^{j,s}) \]

\[ \approx \mathcal{N}(\mathbf{x}_{k; k|k}^{s}, \mathbf{P}_{k|k}^{s}) \]
Computation of Lambda

As a result one obtains

\[
\Lambda(k) = \frac{c_{k|k}}{c_{k|k-1}} \prod_{s=1}^{S} \bar{p}^s \int d\mathbf{x}_k \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, P_{k|k})
\]

\[
= \frac{c_{k|k}}{c_{k|k-1}} \prod_{s=1}^{S} \bar{p}^s
\]

where the posterior normalization constant is given by

\[
c_{k|k} = \prod_{s=1}^{S-1} \mathcal{N}(\mathbf{x}^{s+1}_{k|k}; \mathbf{x}^{(1:s)}_{k|k}, \mathbf{P}^{(1:s)}_{k|k} + \mathbf{P}^{s}_{k|k})
\]
Conclusion: Distributed Track Existence Decision

Local Sensor Nodes

**Prediction: Relaxed Evolution Model**

\[
x_{k|k-1}^s = F_{k|k-1} x_{k|k-1}^s,
\]

\[
P_{k|k-1}^s = F_{k|k-1} P_{k|k-1}^s F_{k|k-1}^T + S Q_{k|k-1}
\]

**Filtering:**

- Update state parameters with EKF / MHT / PDAD / …
- calculate decision contribution

\[
p^{*j,s} = \begin{cases} (1 - p_D) \\ p_D \frac{p_D}{p_F} \mathcal{N}(z_{k|k-1}, H_k^s x_{k|k-1}^s, S_k^s) \end{cases}
\]

\[
\bar{p}^s = \sum_{j=0}^{m_s} p^{*j,s}.
\]

Fusion Center

**Prediction:** calculate prior constants using the Relaxed Evolution Model and the previous transmission:

\[
c_{k|k-1} = \prod_{s=1}^{S-1} \mathcal{N}(x_{k|k-1}^{s+1}, x_{k|k-1}^{1:s}, p_{k|k-1}^{1:s} + p_{k|k-1}^s).
\]

**Filtering:** calculate posterior constants using the new transmissions.

**Update LR score:**

\[
LR(k) = \Lambda(k) \cdot LR(k - 1)
\]

\[
\Lambda(k) = \frac{c_{k|k-1}}{c_{k|k-1}} \prod_{s=1}^{S} \bar{p}^s
\]
NUMERICAL EXAMPLES
Simulation Setup

For the evaluation, a realistic multi-radar scenario has been chosen:

- 4 radars arranged along a circle of about 13 km
- Poisson distributed FA with mean 5 per sensor per scan
- The target, if present, has a process noise of $\text{psd} = 10$
- Probability of detection is $p_D = 0.2$, 0.5 and $p_D = 0.9$.
- A no target scenario is also considered

We compare against:

- Centralized processing (CKF) for LR calculation (optimal)
- Decentralized mean of all local LR scores (LKF)
Plot of a single scan

Sensor positions

Target position

Radar plots with error covariances
Numerical Results of the LR Scores

\[ p_D = 0.9 \]

Mean LR for \( p_D = 0.90 \)

![Graph showing LR scores for different algorithms over steps with a probability of detection (pD) of 0.9.](image)

- **Fig. 1.** Mean Logarithm of the LR score for 100 Monte Carlo during the first 10 steps of a track existence decision for a probability of detection (\( p_D \)) of 0.9 (a), 0.5 (b), and 0.2 (c). In the simulation run of (d) the target is non-existent.

Deviations in the distributed case while the CKF still has sufficient detections. This effect is particularly strong in non-linear applications, since the linearization depends on the quality of the local estimate. However, one can see that the distributed algorithm outperforms the "naive" approach of the LKF. Figure 1 (d) additionally shows an example of a scenario in which no target is existent. It becomes obvious, that also in this case, the DKF has a performance which is nearly optimal and better than the LKF.

**VI. Conclusion**

In this paper we revisited the approach for a distributed detection algorithm which is able to compute the statistical decision, whether a track exists or not, with close-to-optimum performance. The method is based on the product representation of the Distributed Kalman Filter (DKF), which requires that the measurement models are known at each time step. We extended this approach to circumvent this condition such that it can be applied in non-linear scenarios. To this end, the approximation of the Federated Kalman Filter (FKF) was used. The method computes sufficient statistics, such that only one additional reel number must be transmitted to the fusion center. Moreover, the method can be applied to arbitrary Track-to-Track-Fusion methods, since it is independent from the actual fusion process of the track estimates. In the numerical evaluation, we have shown that the proposed method achieves a good performance and beats the standard naive algorithm in scenarios with high and with low probability of detection.
Numerical Results of the LR Scores

\[ p_D = 0.5 \]

Fig. 1. Mean Logarithm of the LR score for 100 Monte Carlo during the first 10 steps of a track existence decision for a probability of detection \( p_D \) of 0.9 (a), 0.5 (b), and 0.2 (c). In the simulation run of (d) the target is non-existent.

VI. Conclusion

In this paper we revisited the approach for a distributed detection algorithm which is able to compute the statistical decision, whether a track exists or not, with close-to-optimum performance. The method is based on the product representation of the Distributed Kalman Filter (DKF), which requires that the measurement models are known at each time step. We extended this approach to circumvent this condition such that it can be applied in non-linear scenarios. To this end, the approximation of the Federated Kalman Filter (FKF) was used. The method computes sufficient statistics, such that only one additional reel number must be transmitted to the fusion center. Moreover, the method can be applied to arbitrary Track-to-Track-Fusion methods, since it is independent from the actual fusion process of the track estimates. In the numerical evaluation, we have shown that the proposed method achieves a good performance and beats the standard naive algorithm in scenarios with high and with low probability of detection.
Numerical Results of the LR Scores

\[ p_D = 0.2 \]

\[ \log \text{likelihood ratio} \]

Mean LR for \( p_D = 0.20 \)

- CKF
- DKF
- LKF

VI. Conclusion

In this paper we revisited the approach for a distributed detection algorithm which is able to compute the statistical decision, whether a track exists or not, with close-to-optimum performance. The method is based on the product representation of the Distributed Kalman Filter (DKF), which requires that the measurement models are known at each time step. We extended this approach to circumvent this condition such that it can be applied in non-linear scenarios. To this end, the approximation of the Federated Kalman Filter (FKF) was used. The method computes sufficient statistics, such that only one additional reel number must be transmitted to the fusion center. Moreover, the method can be applied to arbitrary Track-to-Track-Fusion methods, since it is independent from the actual fusion process of the track estimates. In the numerical evaluation, we have shown that the proposed method achieves a good performance and beats the standard naive algorithm in scenarios with high and with low probability of detection.
Numerical Results of the LR Scores

No target

Fig. 1. Mean Logarithm of the LR score for 100 Monte Carlo during the first 10 steps of a track existence decision for a probability of detection \( p_D \) of 0.9 (a), 0.5 (b), and 0.2 (c). In the simulation run of (d) the target is non-existent. It becomes obvious, that also in this case, the DKF has a performance which is nearly optimal and better than the LKF.

VI. Conclusion

In this paper we revisited the approach for a distributed detection algorithm which is able to compute the statistical decision, whether a track exists or not, with close-to-optimum performance. The method is based on the product representation of the Distributed Kalman Filter (DKF), which requires that the measurement models are known at each time step. We extended this approach to circumvent this condition such that it can be applied in non-linear scenarios. To this end, the approximation of the Federated Kalman Filter (FKF) was used. The method computes sufficient statistics, such that only one additional reel number must be transmitted to the fusion center. Moreover, the method can be applied to arbitrary Track-to-Track-Fusion methods, since it is independent from the actual fusion process of the track estimates. In the numerical evaluation, we have shown that the proposed method achieves a good performance and beats the standard naive algorithm in scenarios with high and with low probability of detection.
Conclusion

- Distributed Sequential Likelihood Ratio for decision on target detection has been presented.
- Fusion center computes LR score based on single real valued parameter from each sensor.
- The distributed calculation clearly performs better than averaging the local LR scores even with identical sensors parameters.
- The method can well be applied to real world applications.

Contact:
Felix Govaers
Dept. Sensor Data and Information Fusion
felix.govaers@fkie.fraunhofer.de
0228 9435 419