Likelihood Functions

The likelihood function answers the question:

What does the sensor tell about the state \( x \) of the object?

(input: sensor data, sensor model)

- **ideal conditions, one object:** \( P_D = 1, \rho_F = 0 \)
  
  at each time one measurement:

  \[
p(z_k | x_k) = \mathcal{N}(z_k; Hx_k, R)
  \]

- **real conditions, one object:** \( P_D < 1, \rho_F > 0 \)
  
  at each time \( n_k \) measurements \( Z_k = \{ z_k^1, \ldots, z_k^{n_k} \} \)

  \[
p(Z_k, n_k | x_k) \propto (1 - P_D)\rho_F + P_D \sum_{j=1}^{n_k} \mathcal{N}(z_k^j; Hx_k, R)
  \]
Bayes Filtering for: $P_D < 1, \rho_F > 0$, well-separated objects

state $x_k$, current data $Z_k = \{z^j_k\}_{j=1}^{m_k}$, accumulated data $Z^k = \{Z_k, Z^{k-1}\}$

interpretation hypotheses $E_k$ for $Z_k$

object not detected, $1 - P_D$
object from object, $P_D$

$m_k + 1$ interpretations

interpretation histories $H_k$ for $Z^k$

• tree structure: $H_k = (E_{H_k}, H_{k-1}) \in \mathcal{H}^k$
• current: $E_{H_k}$, prehistories: $H_{k-i}$

$$p(x_k | Z^k) = \sum_{H_k} p(x_k, H_k | Z^k) = \sum_{H_k} \frac{p(H_k | Z^k)}{\text{weight!}} \cdot \frac{p(x_k | H_k, Z^k)}{\text{given } H_k: \text{unique}}$$

‘mixture’ density
Closer look: \( P_D < 1, \rho_F > 0 \), well-separated targets

filtering (at time \( t_{k-1} \)):

\[
p(x_{k-1}|Z^{k-1}) = \sum_{H_{k-1}} p_{H_{k-1}} \mathcal{N}(x_{k-1}; x_{H_{k-1}}, P_{H_{k-1}})
\]

prediction (for time \( t_k \)):

\[
p(x_k|Z^{k-1}) = \int dx_{k-1} \ p(x_k|x_{k-1}) p(x_{k-1}|Z^{k-1}) \quad \text{(MARKOV model)}
\]

\[
= \sum_{H_{k-1}} p_{H_{k-1}} \mathcal{N}(x_k; Fx_{H_{k-1}}, FP_{H_{k-1}}F^T + D)
\]

measurement likelihood:

\[
p(Z_k, m_k|x_k) = \sum_{j=0}^{m_k} p(Z_k|E_k^j, x_k, m_k) P(E_k^j|x_k, m_k) \quad (E_k^j: \text{interpretations})
\]

\[
\propto (1 - P_D) \rho_F + P_D \sum_{j=1}^{m_k} \mathcal{N}(z_k^j; Hx_k, R) \quad (H, R, P_D, \rho_F)
\]

filtering (at time \( t_k \)):

\[
p(x_k|Z^k) \propto p(Z_k, m_k|x_k) p(x_k|Z^{k-1}) \quad \text{(BAYES’ rule)}
\]

\[
= \sum_{H_k} p_{H_k} \mathcal{N}(x_k; x_{H_k}, P_{H_k}) \quad \text{(Exploit product formula)}
\]
Exercise: Show that $p(x_k|Z^k)$ is given by Kalman updates and weights $p_{H_k}^j$.

$$p(x_k|Z^k) = \sum_{j=0}^{m_k} \sum_{H_{k-1}} p_{H_{k-1}}^j \mathcal{N}(x_{k-1}; x_{H_{k-1}}^j, P_{H_{k-1}}^j)$$

$$p_{H_{k-1}}^j = \frac{p_{H_{k-1}}^{j^*}}{\sum_j p_{H_{k-1}}^{j^*}}$$

$$p_{H_{k-1}}^{j^*} = p_{H_{k-1}} \begin{cases} (1 - P_D) \rho_F & j=0 \\ \frac{P_D}{\sqrt{2\pi S_{H_{k-1}}^j}} e^{-\frac{1}{2}v_{H_k}^j S_{H_{k-1}}^{-1} v_{H_k}^j} & j \neq 0 \end{cases}$$

Insert mixtures and exploit product formula in the numerator and denominator!
Problem: Growing Memory Disaster:

\[ m \text{ data, } N \text{ hypotheses } \rightarrow N^{m+1} \text{ continuations} \]

radical solution: mono-hypothesis approximation
Problem: Growing Memory Disaster:

\[ m \text{ data}, \ N \text{ hypotheses} \rightarrow N^{m+1} \text{ continuations} \]

radical solution: mono-hypothesis approximation

- **gating:** Exclude competing data with \( ||\nu^i_{k|k-1}|| > \lambda! \)

  \[ \rightarrow \text{KALMAN filter (KF)} \]

  + very simple, − \( \lambda \) too small: loss of target measurement
Problem: Growing Memory Disaster:

\( m \) data, \( N \) hypotheses \( \rightarrow N^{m+1} \) continuations

radical solution: mono-hypothesis approximation

- **gating**: Exclude competing data with \( ||\hat{v}_k^i|k-1|| > \lambda! \)

  \[ \text{KALMAN filter (KF)} \]

  + very simple, – \( \lambda \) too small: loss of target measurement

- Force a **unique interpretation** in case of a conflict!

  look for *smallest statistical distance*: \( \min_i ||\hat{v}_k^i|k-1|| \)

  \[ \text{Nearest-Neighbor filter (NN)} \]
Problem: Growing Memory Disaster:
\[ m \text{ data}, \, N \text{ hypotheses} \rightarrow N^{m+1} \text{ continuations} \]

radical solution: mono-hypothesis approximation

- **gating**: Exclude competing data with \[ \|\nu_{k|k-1}^i\| > \lambda! \]
  
  \[ \text{KALMAN filter (KF)} \]

  + very simple, – \( \lambda \) too small: loss of target measurement

- Force a **unique interpretation** in case of a conflict!

  look for **smallest statistical distance**: \[ \min_i \|\nu_{k|k-1}^i\| \]

  \[ \text{Nearest-Neighbor filter (NN)} \]

  + one hypothesis, – hard decision, – not adaptive

- **global combining**: Merge all hypotheses!

  \[ \text{PDAF, JPDAF filter} \]

  + all data, + adaptive, – reduced applicability
PDAF Filter: formally analog to Kalman Filter

Filtering (scan $k-1$): \[ p(x_{k-1}|\mathcal{Z}^{k-1}) = \mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1}) \] (\(\rightarrow\) initiation)

Prediction (scan $k$): \[ p(x_k|\mathcal{Z}^{k-1}) \approx \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \] (like Kalman)

Filtering (scan $k$): \[ p(x_k|\mathcal{Z}^k) \approx \sum_{j=0}^{m_k} p^j_k \mathcal{N}(x_k; x^j_{k|k}, P^j_{k|k}) \approx \mathcal{N}(x_k; x_{k|k}, P_{k|k}) \]

\[ \nu_k = \sum_{j=0}^{m_k} p^j_k \nu^j_k, \quad \nu^j_k = z^j_k - Hx_{k|k-1} \] combined innovation

\[ W_k = P_{k|k-1}H^\top S_k^{-1}, \quad S_k = HP_{k|k-1}H^\top + R_k \] Kalman gain matrix

\[ p^j_k = \frac{p^*_k}{\sum_j p^*_k}, \quad p^*_k = \left\{ \begin{array}{ll} \frac{1-P_D}{2\pi \sigma_h^2} & (1 - P_D) \rho_F \sigma_h^2 \nu^\top h_k S_{h_k} \nu_h \end{array} \right. \] weighting factors

\[ x_k = x_{k|k-1} + W_k \nu_k \] (Filtering Update: Kalman)

\[ P_k = P_{k|k-1} - \left(1-P_D^0\right) W_k S W_k^\top \] (Kalman part)

\[ + W_k \left\{ \sum_{j=0}^{m_k} p^j_k \nu^j_k \nu_k^\top - \nu_k \nu_k^\top \right\} W_k^\top \] (Spread of Innovations)
The qualitative shape of $p(x_k|z^k)$ is often much simpler than its correct representation: *a few pronounced modes*

**adaptive solution: nearly optimal approximation**
The qualitative shape of $p(x_k | Z^k)$ is often much simpler than its correct representation: a few pronounced modes

**adaptive solution: nearly optimal approximation**

- **individual gating:** Exclude *irrelevant data*! before continuing existing track hypotheses $H_{k-1}$

  $\rightarrow$ *limiting case:* KALMAN filter (KF)
The qualitative shape of \( p(x_k | \mathcal{Z}^k) \) is often much simpler than its correct representation: a few pronounced modes

**adaptive solution: nearly optimal approximation**

- **individual gating:** Exclude *irrelevant data*! before continuing existing track hypotheses \( H_{k-1} \)
  \[ \rightarrow \text{limiting case: KALMAN filter (KF)} \]

- **pruning:** Kill hypotheses of very *small weight*! after calculating the weights \( p_{H_k} \), before filtering
  \[ \rightarrow \text{limiting case: Nearest Neighbor filter (NN)} \]
The qualitative shape of \( p(x_k | \mathcal{Z}^k) \) is often much simpler than its correct representation: a few pronounced modes

**adaptive solution: nearly optimal approximation**

- **individual gating:** Exclude *irrelevant data!*  
  before continuing existing track hypotheses \( H_{k-1} \)  
  \( \rightarrow \) *limiting case:* KALMAN filter (KF)

- **pruning:** Kill hypotheses of very *small weight!*  
  after calculating the weights \( p_{H_k} \), before filtering  
  \( \rightarrow \) *limiting case:* Nearest Neighbor filter (NN)

- **local combining:** Merge *similar hypotheses!*  
  after the complete calculation of the pdfs  
  \( \rightarrow \) *limiting case:* PDAF (global combining)
Successive Local Combining

Partial sums of similar densities $\rightarrow$ moment matching:

$$\sum_{H_k \in \mathcal{H}^{k^*}} p_{H_k} \mathcal{N}(x_k; x_{H_k}, P_{H_k}) \approx p_{H_k^*} \mathcal{N}(x_k; x_{H_k^*}, P_{H_k^*})$$

$\mathcal{H}^{k^*} \subset \mathcal{H}^k \rightarrow H_k^*: \text{effective hypothesis}$
Successive Local Combining

Partial sums of similar densities $\rightarrow$ moment matching:

$$\sum_{H_k \in \mathcal{H}^k} p_{H_k} \mathcal{N}(x_k; x_{H_k}, P_{H_k}) \approx p_{H_k^*} \mathcal{N}(x_k; x_{H_k^*}, P_{H_k^*})$$

$\mathcal{H}^k \subset \mathcal{H}^k \rightarrow H_k^*$: effective hypothesis

similarity: $d(H_1, H_2) < \mu$ mit (z.B.):

$$d(H_1, H_2) = (x_{H_1} - x_{H_2})^\top (P_{H_1} + P_{H_2})^{-1} (x_{H_1} - x_{H_2})$$

Start: Hypothesis of highest weight $H_1 \rightarrow$ search similar hypothesis $(p_{H \backslash \lambda}) \rightarrow$ merge: $(H_1, H) \triangleright H_1^* \rightarrow$ continue search $(p_{H \backslash \lambda}) \ldots$

$\rightarrow$ restart: hypothesis with next to highest weight $H_2 \rightarrow \ldots$
Successive Local Combining

Partial sums of similar densities $\rightarrow$ moment matching:

$$\sum_{H_k \in \mathcal{H}^{k*}} p_{H_k} \mathcal{N}(x_k; x_{H_k}, P_{H_k}) \approx p_{H_k^*} \mathcal{N}(x_k; x_{H_k^*}, P_{H_k^*})$$

$$\mathcal{H}^{k*} \subset \mathcal{H}^{k} \rightarrow H_k^*: \text{effective hypothesis}$$

similarity: $d(H_1, H_2) < \mu$ mit (z.B.):

$$d(H_1, H_2) = (x_{H_1} - x_{H_2})^\top (P_{H_1} + P_{H_2})^{-1} (x_{H_1} - x_{H_2})$$

Start: Hypothesis of highest weight $H_1 \rightarrow$ search similar hypothesis $(p_{H_1} \backslash \mu) \rightarrow$ merge: $(H_1, H) \triangleright H_1^* \rightarrow$ continue search $(p_{H_1} \backslash \mu) \ldots$

$\rightarrow$ restart: hypothesis with next to highest weight $H_2 \rightarrow \ldots$

- In many cases: good approximations $\rightarrow$ quasi-optimality
- PDAF, JPDAF: $\mathcal{H}^{k*} = \mathcal{H}^{k} \rightarrow$ limited applicability
- robustness $\rightarrow$ detail mostly irrelevant
Retrodiction for GAUSSian Mixtures

\[
\text{wanted: } p(x_l | Z^k) \leftarrow p(x_{l+1} | Z^k) \text{ for } l < k
\]

\[
p(x_l | Z^k) = \sum_{H_k} p(x_l, H_k | Z_k) = \sum_{H_k} p(x_l | H_k, Z^k) p(H_k | Z^k)
\]

no ambiguities! filtering!

Calculation of \( p(x_l | H_k, Z^k) \) as in case of \( P_D = 1, \rho_F = 0 \):

\[
p(x_l | H_k, Z^k) = \mathcal{N}(x_l; x_{H_k}(l|k), P_{H_k}(l|k))
\]

with parameters given by RAUCH-TUNG-STRIEBEL formulae:

\[
x_{H_k}(l|k) = x_{H_k}(l|l) + W_{H_k}(l|k) (x_{H_k}(l+1|k) - x_{H_k}(l+1|l))
\]

\[
P_{H_k}(l|k) = P_{H_k}(l|l) + W_{H_k}(l|k) (P_{H_k}(l+1|k) - P_{H_k}(l+1|l)) W_{H_k}(l|k)^	op
\]

gain matrix: \( W_{H_k}(l|k) = P_{H_k}(l|l) F_{l+1}^{\top} P_{H_k}(l+1|l)^{-1} \)
Retrodiction of Hypotheses’ Weights

Consider approximation: neglect RTS step!

\[ p(x_l|H_k, Z^k) = \mathcal{N}(x_l; x_{H_k}(l|k), P_{H_k}(l|k)) \approx \mathcal{N}(x_l; x_{H_k}(l|l), P_{H_k}(l|l)) \]

\[ p(x_l|H_k, Z^k) \approx \sum_{H_l} p^*_H \mathcal{N}(x_l; x_{H_k}(l|l), P_{H_k}(l|l)) \]

with recursively defined weights:

\[ p^*_H = p_{H_k}, \quad p^*_H = \sum p^*_{H_{l+1}} \]

summation over all histories \( H_{l+1} \) with equal pre-histories!

- Strong sons strengthen weak fathers.
- Weak sons weaken even strong fathers.
- If all sons die, also the father must die.
Track Extraction: Initiation of the PDF Iteration

**extraction of target tracks:** detection on a higher level of abstraction

**start:** data sets \( Z_k = \{z_{jk}^i\}_{j=1}^{m_k} \) (sensor performance: \( P_D, \rho_F, R \))

**goal:** Detect a target trajectory in a time series: \( \mathcal{Z}^k = \{Z_i\}_{i=1}^k \)

at first simplifying assumptions:

- The targets in the sensors’ field of view (FoV) are well-separated.
- The sensor data in the FoV in scan \( i \) are produced simultaneously.
Track Extraction: Initiation of the PDF Iteration

**extraction of target tracks:** detection on a higher level of abstraction

**start:** data sets $Z_k = \{z^j_k\}_{j=1}^{m_k}$ (sensor performance: $P_D$, $\rho_F$, $R$)

**goal:** Detect a target trajectory in a time series: $Z^k = \{Z_i\}_{i=1}^k$

at first simplifying assumptions:

- The targets in the sensors’ field of view (FoV) are well-separated.
- The sensor data in the FoV in scan $i$ are produced simultaneously.

decision between two competing hypotheses:

$h_1$: Besides false returns $Z^k$ contains also target measurements.

$h_0$: There is no target existing in the FoV; all data in $Z^k$ are false.

statistical decision errors:

$$P_1 = \text{Prob(accept } h_1 | h_1)$$ analogous to the sensors’ $P_D$

$$P_0 = \text{Prob(accept } h_1 | h_0)$$ analogous to the sensors’ $P_F$
Practical Approach: Sequential Likelihood Ratio Test

**Goal:** Decide as fast as possible for given decision errors $P_0$, $P_1$!

Consider the ratio of the conditional probabilities $p(h_1|Z^k)$, $p(h_0|Z^k)$ and the likelihood ratio $LR(k) = p(Z^k|h_1)/p(Z^k|h_0)$ as an intuitive decision function:

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

a priori: $p(h_1) = p(h_0)$
Practical Approach: Sequential Likelihood Ratio Test

**Goal:** Decide as fast as possible for given decision errors $P_0$, $P_1$!

Consider the ratio of the conditional probabilities $p(h_1|Z^k)$, $p(h_0|Z^k)$ and the likelihood ratio $LR(k) = p(Z^k|h_1)/p(Z^k|h_0)$ as an intuitive decision function:

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

a priori: $p(h_1) = p(h_0)$

Starting from a time window with length $k = 1$, calculate the test function $LR(k)$ successively and compare it with two thresholds $A$, $B$:

If $LR(k) < A$, accept hypothesis $h_0$ (i.e. no target is existing)!
If $LR(k) > B$, accept hypothesis $h_1$ (i.e. target exists in FoV)!
If $A < LR(k) < B$, wait for new data $Z_{k+1}$, repeat with $LR(k + 1)$!
Sequential LR Test: Some Useful Properties

1. Thresholds and decision errors are approximately related to each other by:

\[ A \approx \frac{1 - P_1}{1 - P_0} \quad \text{and} \quad B \approx \frac{P_1}{P_0} \]

2. The *actual decision length* (number of scans required) is a random variable.

3. On average, the test has a *minimal decision length* for given errors \( P_0, P_1 \).

4. The quantity \( P_0 (P_1) \) affects the *mean decision length* given \( h_1 (h_0) \) holds.

5. Choose the probability \( P_1 \) close to 1 for actually detecting real target tracks.

6. \( P_0 \) should be small for not overloading the tracking system with false tracks.
Iterative Calculation of the Likelihood Ratio

$$\text{LR}(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)} = \frac{\int dx_k p(Z_k, m_k, x_k, Z^{k-1}|h_1)}{p(Z_k, m_k, Z^{k-1}, h_0)}$$
Iterative Calculation of the Likelihood Ratio

\[ \text{LR}(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)} = \frac{\int dx_k p(Z_k, m_k, x_k, Z^{k-1}|h_1)}{p(Z_k, m_k, Z^{k-1}, h_0)} = \frac{\int dx_k p(Z_k, m_k|x_k) p(x_k|Z^{k-1}, h_1) p(Z^{k-1}|h_1)}{|\text{FoV}|^{-m_k} p_F(m_k) p(Z^{k-1}|h_0)} \]
Iterative Calculation of the Likelihood Ratio

$$LR(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)} = \frac{\int dx_k p(Z_k, m_k, x_k, Z^{k-1}|h_1)}{p(Z_k, m_k, Z^{k-1}, h_0)} = \frac{\int dx_k p(Z_k, m_k|x_k) p(x_k|Z^{k-1}, h_1) p(Z^{k-1}|h_1)}{|\text{FoV}|^{-m_k} p_F(m_k) p(Z^{k-1}|h_0)}$$

$$= \frac{\int dx_k p(Z_k, m_k|x_k, h_1) p(x_k|Z^{k-1}, h_1)}{|\text{FoV}|^{-m_k} p_F(m_k)} LR(k-1)$$

basic idea: iterative calculation!
Iterative Calculation of the Likelihood Ratio

\[
\text{LR}(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)} = \frac{\int d x_k p(Z_k, m_k, x_k, Z^{k-1}|h_1)}{p(Z_k, m_k, Z^{k-1}, h_0)} = \frac{\int d x_k p(Z_k, m_k|x_k) \ p(x_k|Z^{k-1}, h_1) p(Z^{k-1}|h_1)}{|\text{FoV}|^{-m_k} \ p_F(m_k) \ p(Z^{k-1}|h_0)} = \frac{\int d x_k p(Z_k, m_k|x_k, h_1) \ p(x_k|Z^{k-1}, h_1)}{|\text{FoV}|^{-m_k} \ p_F(m_k)} \text{LR}(k-1)
\]

basic idea: iterative calculation!

Let \( H_k = \{E_k, H_{k-1}\} \) be an interpretation history of the time series \( Z^k = \{Z_k, Z^{k-1}\} \).

\( E_k = E_k^0 \): target was not detected, \( E_k = E_k^j \): \( z^j_k \in Z_k \) is a target measurement.

\[
p(x_k|Z^{k-1}, h_1) = \sum_{H_{k-1}} p(x_k|H_{k-1}Z^{k-1}, h_1) p(H_{k-1}|Z^{k-1}, h_1) \quad \text{The standard MHT prediction!}
\]

\[
p(Z_k, m_k|x_k, h_1, h_1) = \sum_{E_k} p(Z_k, E_k|x_k, h_1) \quad \text{The standard MHT likelihood function!}
\]

The calculation of the likelihood ratio is just a by-product of Bayesian MHT tracking.
Iteration Formula for $\text{LR}(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)}$

**initiation:**

$$k = 0, \quad j_0 = 0, \quad \lambda_{j_0} = 1$$

**recursion:**

$$\text{LR}(k+1) = \sum_{j_{k+1}} \lambda_{j_{k+1}} = \sum_{j_{k+1}=0}^{m_{k+1}} \sum_{j_k} \lambda_{j_{k+1}j_k} \lambda_{j_k}$$

with:

$$\lambda_{j_{k+1}j_k} = \begin{cases} 1 - P_D & \text{for} \quad j_{k+1} = 0 \\ \frac{P_D}{\rho_p} \mathcal{N}(\nu_{j_{k+1}j_k}, S_{j_{k+1}j_k}) & \text{for} \quad j_{k+1} \neq 0 \end{cases}$$

**convenient notation:**

with $j_k = (j_k, \ldots, j_1)$ let

$$\sum_{j_k} \lambda_{j_k} = \sum_{j_k=0}^{m_k} \cdots \sum_{j_1=0}^{m_1} \lambda_{j_k \cdots j_1}$$
Iteration Formula for $\text{LR}(k) = p(Z^k|h_1)/p(Z^k|h_0)$

**initiation:**

$k = 0, \quad j_0 = 0, \quad \lambda_{j_0} = 1$

**recursion:**

$\text{LR}(k + 1) = \sum_{j_{k+1}} \lambda_{j_{k+1}} = \sum_{j_{k+1}=0}^{m_{k+1}} \lambda_{j_{k+1}j_k} \lambda_{j_k}$

with:

$\lambda_{j_{k+1}j_k} = \begin{cases} 
1 - P_D & \text{for } j_{k+1} = 0 \\
\frac{P_D}{\rho_F} N(\nu_{j_{k+1}j_k}, S_{j_{k+1}j_k}) & \text{for } j_{k+1} \neq 0
\end{cases}$

**innovation:**

$\nu_{j_{k+1}j_k} = z_{j_{k+1}} - H_{j_{k+1}} x_{j_{k+1}|k}$

**innov. cov.:**

$S_{j_{k+1}j_k} = H_{j_{k+1}} P_{j_{k+1}|k} H_{j_{k+1}}^T + R_{j_{k+1}}$

**state update:**

$x_{j_{k+1}|k} = F_{j_{k+1}} x_{j_k}$

$x_{j_k} = x_{j_k|k-1} + W_{j_k} \nu_{j_k|k-1}$

**covariances:**

$P_{j_{k+1}|k} = F_{j_{k+1}} P_{j_k} F_{j_{k+1}}^T + D_{j_{k+1}}$

$P_{j_k} = P_{j_{k}|k-1} - W_{j_k} S_{j_{k}|k-1} W_{j_{k}|k-1}^T$
Iteration Formula for \( LR(k) = p(Z^k|h_1)/p(Z^k|h_0) \)

**initiation:** \( k = 0, \quad j_0 = 0, \quad \lambda_{j_0} = 1 \)

**recursion:** \( LR(k + 1) = \sum_{j_{k+1}} \lambda_{j_{k+1}} = \sum_{j_{k+1}=0}^{m_{k+1}} \sum_{i_k} \lambda_{j_{k+1}i_k} \lambda_{i_k} \)

with: \( \lambda_{j_{k+1}i_k} = \begin{cases} 1 - P_D & \text{for } j_{k+1} = 0 \\ \frac{P_D}{\rho_F} \mathcal{N}(\nu_{j_{k+1}i_k}, S_{j_{k+1}i_k}) & \text{for } j_{k+1} \neq 0 \end{cases} \)

**innovation:** \( \nu_{j_{k+1}} = z_{j_{k+1}} - H_{j_{k+1}} x_{j_{k+1}|k} \)

**innov. cov.:** \( S_{j_{k+1}} = H_{j_{k+1}} P_{j_{k+1}|k} H^T_{j_{k+1}} + R_{j_{k+1}} \)

**state update:** \( x_{j_{k+1}|k} = F_{j_{k+1}} x_{j_k} \quad x_{j_k} = x_{j_{k-1}|k-1} + W_{j_{k-1}j_k} \nu_{j_{k-1}} \)

**covariances:** \( P_{j_{k+1}|k} = F_{j_{k+1}} P_{j_k} F^T_{j_{k+1}} + D_{j_{k+1}} \quad P_{j_k} = P_{j_{k-1}|k-1} - W_{j_{k-1}j_k} S_{j_{k-1}j_k} W^T_{j_{k-1}j_k} \)

**Exercise 9.1** Show that this recursion formulae for calculating the decision function is true.
Sequential Track Extraction: Discussion

- $LR(k)$ is given by a growing number of summands, each related to a particular interpretation history. The tuple $\{\lambda_{jk}, x_{jk}, P_{jk}\}$ is called a sub-track.
Sequential Track Extraction: Discussion

- $LR(k)$ is given by a growing number of summands, each related to a particular interpretation history. The tuple $\{\lambda_{jk}, x_{jk}, P_{jk}\}$ is called a sub-track.

- For mitigating growing memory problems all approximations discussed for track maintenance can be used if they do not significantly affect $LR(k)$:
  - **individual gating**: Exclude data not likely to be associated.
  - **pruning**: Kill sub-tacks contributing marginally to the test function.
  - **local combining**: Merge similar sub tracks:
    \[
    \{\lambda_i, x_i, P_i\}_i \rightarrow \{\lambda, x, P\} \quad \text{with:} \quad \lambda = \sum_i \lambda_i, \\
    x = \frac{1}{\lambda} \sum_i \lambda_i x_i, \quad P = \frac{1}{\lambda} \sum_i \lambda_i [P_i + (x_i - x)(\ldots)^\top].
    \]
Sequential Track Extraction: Discussion

- $LR(k)$ is given by a growing number of summands, each related to a particular interpretation history. The tuple $\{\lambda_{jk}, x_{jk}, P_{jk}\}$ is called a sub-track.

- For mitigating growing memory problems all approximations discussed for track maintenance can be used if they do not significantly affect $LR(k)$:
  - individual gating: Exclude data not likely to be associated.
  - pruning: Kill sub-tacks contributing marginally to the test function.
  - local combining: Merge similar sub tracks:

$$\{\lambda_i, x_i, P_i\}_i \rightarrow \{\lambda, x, P\} \quad \text{with:} \quad \lambda = \sum_i \lambda_i,$$

$$x = \frac{1}{\lambda} \sum_i \lambda_i x_i, \quad P = \frac{1}{\lambda} \sum_i \lambda_i [P_i + (x_i - x)(\ldots)^{\top}].$$

- The LR test ends with a decision in favor of or against the hypotheses: $h_0$ (no target) or $h_1$ (target existing). Intuitive interpretation of the thresholds!
track extraction at $t_k$: Decide in favor of $h_1$!

initiation of pdf iteration (track maintenance):

Normalize coefficients $\lambda_j$:  

$$p_j = \frac{\lambda_j \lambda_k}{\sum \lambda_j \lambda_k}$$

$$(\lambda_j, x_j, P_j) \rightarrow p(x_k | Z_k^k) = \sum_j p_j \mathcal{N}(x_k; x_j, P_j)$$

Continue track extraction with the remaining sensor data!

sequential LR test for track monitoring:

After deciding in favor of $h_1$ reset $LR(0) = 1!$ Calculate $LR(k)$ from $p(x_k | Z^k)$!

\begin{align*}
\text{track confirmation:} & \quad LR(k) > \frac{P_1}{P_0}: \text{reset } LR(0) = 1! \\
\text{track deletion:} & \quad LR(k) < \frac{1-P_1}{1-P_0}; \text{ ev. track re-initiation}
\end{align*}
DEMONSTRATION (simulated)
DEMONSTRATION (simulated)

Exercise 9.2 (voluntary)

Simulate a detection process with a given $P_D$, target measurements with a given $R$, a detection process with a given $P_D$ and realize the track extraction procedure.