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 \( Z_k = \{ z_k^j \}_{j=1}^{m_k} \), accumulated data \( \mathcal{Z}^k = \{ Z_k, Z^{k-1} \} \)

interpretation hypotheses \( E_k \) for \( Z_k \)

object not detected, \( 1 - P_D \)

\( z_k \in Z_k \) from object, \( P_D \)

\( m_k + 1 \) interpretations

interpretation histories \( H_k \) for \( \mathcal{Z}^k \)

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

\[
p(x_k | \mathcal{Z}^k) = \sum_{H_k} p(x_k, H_k | \mathcal{Z}^k) = \sum_{H_k} \underbrace{p(H_k | \mathcal{Z}^k)}_{\text{weight!}} \underbrace{p(x_k | H_k, \mathcal{Z}^k)}_{\text{given } H_k: \text{ unique}}
\]

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

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

prediction (for time \( t_k \)):
\[
p(x_k | \mathcal{Z}^{k-1}) = \int dx_{k-1} p(x_k | x_{k-1}) p(x_{k-1} | \mathcal{Z}^{k-1}) \quad \text{(MARKOV model)}
\]
\[
= \sum_{H_{k-1}} p_{H_{k-1}} \mathcal{N}(x_k; F_x H_{k-1}, FP_{H_{k-1}} F^T + D) \quad \text{(IMM also possible)}
\]

measurement likelihood:
\[
p(Z_k, m_k | x_k) = \sum_{j=0}^{m_k} p(Z_k | E^j_k, x_k, m_k) P(E^j_k | x_k, m_k) \quad (E^j_k: \text{interpretations})
\]
\[
\propto (1 - P_D) \rho_F + P_D \sum_{j=1}^{m_k} \mathcal{N}(z^j_k; H x_k, R) \quad (H, R, P_D, \rho_F)
\]

filtering (at time \( t_k \)):
\[
p(x_k | \mathcal{Z}^k) \propto p(Z_k, m_k | x_k) p(x_k | \mathcal{Z}^{k-1}) \quad \text{(BAYES' rule)}
\]
\[
= \sum_{H_k} p_{H_k} \mathcal{N}(x_k; x_{H_k}, P_{H_k}) \quad \text{(Exploit product formula)}
\]
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
**Moment Matching:** Approximate an arbitrary pdf $p(x)$ with $\mathbb{E}[x] = x$, $\mathbb{C}[x] = P$ by $p(x) \approx \mathcal{N}(x; x, P)$!

Here especially: 

$$p(x) = \sum_{H} p_H \mathcal{N}(x; x_H, P_H) \quad (\text{normal mixtures})$$

$$x = \sum_{H} p_H x_H$$

$$P = \sum_{H} p_H \left\{ P_H + (x_H - x)(x_H - x)^\top \right\}$$

**spread term**
PDAF Filter: formally analogous to Kalman Filter

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

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

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

\[
x^j_k|k = \begin{cases} x_{k|k-1} & j=0 \\ x_{k|k-1} + W_k \nu^j_k & j \neq 0 \end{cases}
\]

\[
\nu^j_k = z^j_k - H x_k, \quad W_k = P_{k|k-1} H^T S_k^{-1}, \quad S_k = H P_{k|k-1} H^T + R_k
\]

\[
p^j_k = p^*_k \sum_j p^*_j, \quad p^*_k = \begin{cases} (1 - P_D) \rho_F & j=0 \\ \frac{P_D}{\sqrt{2\pi S_h}} e^{-\frac{1}{2} \nu_{h_k}^T S_h^{-1} \nu_{h_k}} & j \neq 0 \end{cases}
\]
PDAF Filter: formally analog to Kalman Filter

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

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

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

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

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

\[ p^j_k = p^j_k^* / \sum_j p^j_k^*, \quad p^j_k^* = \left\{ \frac{(1 - P_D) \rho_F}{\sqrt{2\pi S_{h_k}} e^{-\frac{1}{2} \nu^T_{h_k} S_{h_k} \nu_{h_k}}} \right\} \quad \text{weighting factors} \]

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

\[ P_k = P_{k|k-1} - (1-P^0_k) W_k S W_k^T \] (Kalman part)

\[ + W_k \left\{ \sum_{j=0}^{m_k} p^j_k \nu^j_k \nu^T_k - \nu_k \nu_k^T \right\} W_k^T \] (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 $p(x_k|\mathcal{Z}_k)$ 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 $H_{k-1}$
  - → *limiting case*: KALMAN filter (KF)
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 $H_{k-1}$
  
  → *Limiting case:* KALMAN filter (KF)

- **Pruning:** Kill hypotheses of very *small weight*!
  
  after calculating the weights $p_{H_k}$, before filtering
  
  → *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:

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

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

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

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

$\mathcal{H}^* \subset \mathcal{H}^k$ $\rightarrow$ $H_k^*$: 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 $(p_{H \downarrow})$ $\rightarrow$ merge: $(H_1, H) \succ H_1^* \rightarrow$ continue search $(p_{H \downarrow})$ ... $\rightarrow$ **restart:** hypothesis with next to highest weight $H_2 \rightarrow$ ...
Successive Local Combining

Partial sums of similar densities → 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 \(H_1\) → search similar hypothesis \((p_{H \setminus \sigma})\) → merge: \((H_1, H) \succ H_1^*\) → continue search \((p_{H \setminus \sigma})\) . . .

→ restart: hypothesis with next to highest weight \(H_2\) → . . .

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

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

\[
p(x_l|Z^k) = \sum_{H_k} p(x_l, H_k|Z_k) = \sum_{H_k} \left( \frac{p(x_l|H_k, Z^k) \cdot p(H_k|Z^k)}{\text{no ambiguities!}} \right) \text{ filtering!}
\]

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:

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

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

\[
\text{gain matrix: } W_{H_k}(l|k) = P_{H_k}(l|l)F_{l+1}^{T}P_{H_k}(l+1|l)^{-1}
\]
Retrodiction of Hypotheses’ Weights

Consider approximation: neglect RTS step!

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

\[ p(x_l|H_k, Z^k) \approx \sum_{H_{l+1}} p^*_H \cdot \mathcal{N}(x_l; x_{H_k}(l|l), P_{H_k}(l|l)) \]

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.

Sensor Data Fusion - Methods and Applications, 7th Lecture on December 20, 2017
Recapitulation: Detection Process for Sensors

**Detector**: receives signals and decides on object existence

**Processor**: processes detected signals and produces measurements

‘\(D\)’: detector detects an object

\(D\): object actually existent

error of 1. kind: \(P_I = P(\neg 'D'|D)\)

error of 2. kind: \(P_{II} = P('D'|\neg D)\)

measure of detection performance: \(P_D = P('D'|D)\)

detector properties characterized by two parameters:

– detection probability \(P_D = 1 - P_I\)

– false alarm probability \(P_F = P_{II}\)

example (Swerling I model): \(P_D = P_D(P_F, \text{SNR}) = P_F^{1/(1+\text{SNR})}\)

**Detector Design**: Maximize detection probability \(P_D\)

for a given, predefined false alarm probability \(P_F\).
Track Extraction: Initiation of the PDF Iteration

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

**start:** data sets $Z_k = \{z_{kj}^j\}_{j=1}^{mk}$ (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 \( Z_k = \{ z_{kj} \}_{j=1}^{m_k} \) (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.

---

**decision between two competing hypotheses:**

\( h_1 \): Besides false returns \( \mathcal{Z}^k \) contains also target measurements.

\( h_0 \): There is no target existing in the FoV; all data in \( \mathcal{Z}^k \) are false.

---

**statistical decision errors:**

\[
\begin{align*}
P_1 & = \text{Prob}(\text{accept } h_1| h_1) \quad \text{analogous to the sensors’ } P_D \\
P_0 & = \text{Prob}(\text{accept } h_1| h_0) \quad \text{analogous to the sensors’ } P_F
\end{align*}
\]
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:

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

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 $LR(k+1)$!
Iterative Calculation of the Likelihood Ratio

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

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

= \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)} \text{LR}(k - 1)

basic idea: iterative calculation!
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!

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

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

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

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

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

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

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

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

**convenient notation:** with \( j_k = (j_k, \ldots, j_1) \) let \( \sum_{j_k} \lambda_{j_k} = \sum_{j_k=0}^{m_k} \cdots \sum_{j_1=0}^{m_1} \lambda_{j_k \ldots j_1} \)
Iteration Formula for $LR(k) = \frac{p(Z^k|h_1)}{p(Z^k|h_0)}$

**initiation:**

\[ k = 0, \quad j_0 = 0, \quad \lambda_{j_0} = 1 \]

**recursion:**

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

with:

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

**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:**

\[ x_{j_{k+1}|k} = F_{j_{k+1}}x_{j_k} \quad \quad x_j = x_{j|k-1} + W_{j_{k-1}} \nu_{j_k|k-1} \]

**covariances:**

\[ P_{j_{k+1}|k} = F_{j_{k+1}}P_{j_k}F_{j_{k+1}}^T + D_{j_{k+1}} \quad P_{j_k} = P_{j|k-1} - W_{j_{k-1}}S_{j_{k-1}}W_{j_{k-1}}^T \]
Iteration Formula for  \( LR(k) = \frac{p(\mathcal{Z}^k|h_1)}{p(\mathcal{Z}^k|h_0)} \)

initiation:  
\[ k = 0, \quad j_0 = 0, \quad \lambda_{j_0} = 1 \]

recursion:  
\[ LR(k+1) = \sum_{j_{k+1}} \lambda_{j_{k+1}} = \sum_{j_{k+1}=0}^{m_{k+1}} \sum_{j_k} \lambda_{j_{k+1}j_k} \lambda_{j_k} \]

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

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:  
\[ x_{j_{k+1}|k} = F_{j_{k+1}} x_j \]

\[ x_j = x_{j_{k-1}|k} + W_{j_{k-1}} \nu_{j_{k-1}} \]

covariances:  
\[ P_{j_{k+1}|k} = F_{j_{k+1}} P_j F_{j_{k+1}}^T + D_{j_{k+1}} \]

\[ P_j = P_{j_{k-1}} - W_{j_{k-1}} S_{j_{k-1}} W_{j_{k-1}}^T \]

**Exercise 7.1**  
Show that this recursion formulae for calculating the decision function is true.
Sequential Track Extraction: Discussion

- \( \text{LR}(k) \) 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: } \lambda = \sum_i \lambda_i,
\[
\begin{align*}
\lambda_i \times x_i, \quad P & = \frac{1}{\lambda} \sum_i \lambda_i \left[ P_i + (x_i - x)(\ldots)^\top \right].
\end{align*}
$$
Sequential Track Extraction: Discussion

- \( \text{LR}(k) \) 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 \( \text{LR}(k) \):
  - 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:

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

- 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 $\lambda_{jk}$: 

$$p_{jk} = \frac{\lambda_{jk}}{\sum_{j_k} \lambda_{jk}}!$$

$$(\lambda_{jk}, x_{jk}, P_{jk}) \rightarrow p(x_k|Z^k) = \sum_{j_k} p_{jk} \mathcal{N}(x_k; x_{jk}, P_{jk})$$

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 $p(x_k|Z^k)$!

track confirmation: $LR(k) > \frac{P_1}{P_0}$: reset $LR(0) = 1$!

track deletion: $LR(k) < \frac{1-P_1}{1-P_0}$; ev. track re-initiation
DEMONSTRATION (simulated)
DEMONSTRATION (simulated)

Exercise 7.2 (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.
Generalization to Target Cluster (Perfect Resolution)

Scheme directly extendable to clusters consisting of \( n \) targets, if \( n \) is known!

**principal approach in case of unknown \( n \):**

1. Start with sensor measurements \( Z_1 \).

2. Assume for a target cluster \( n \leq N \)! A-priorily: \( P(n) = \frac{1}{N} \)

3. hypothesis \( h_n \): there exist \( n \) targets; the data set \( Z_1 \) contains at least one target measurement; \( h_0 \): no target existing at all

4. Consider the following ratio (at least 1, at most \( N \) targets):

\[
\frac{p(h_1 \lor \ldots \lor h_N | Z^k)}{p(h_0 | Z^k)} = \frac{\sum_{n=1}^{N} p(h_n | Z^k)}{p(h_0 | Z^k)} = \frac{\sum_{n=1}^{N} \frac{p(Z^k | h_n) p(h_n)}{p(h_0)}}{p(h_0)}
\]
Generalization to Target Cluster (Perfect Resolution)

Scheme directly extendable to clusters consisting of \( n \) targets, \textit{if \( n \) is known}!

**principal approach in case of unknown \( n \):**

1. Start with sensor measurements \( Z_1 \).

2. Assume for a target cluster \( n \leq N \)! A-priorily: \( P(n) = \frac{1}{N} \)

3. hypothesis \( h_n \): there exist \( n \) targets; the data set \( Z_1 \) contains at least one target measurement; \( h_0 \): no target existing at all

4. generalized LR test function: \( \text{LR}(k) = \frac{1}{N} \sum_{n=1}^{N} \frac{p(Z^k|h_n)}{p(Z^k|h_0)} \)

5. Calculate \( \text{LR}_n(k) = p(Z^k|h_n)/p(Z^k|h_0) \) in analogy to \( n = 1 \).

6. ‘Cardinality’ of having \( n \) objects in the cluster: \( c_k(n) = \frac{\text{LR}_n(k)}{\sum_{n=1}^{N} \text{LR}_n(k)} \)
DEMONSTRATION (simulated)

**Moment Matching:** Approximate an arbitrary pdf $p(x)$ with $\mathbb{E}[x] = x$, $\mathbb{C}[x] = P$ by $p(x) \approx \mathcal{N}(x; x, P)$.

Here especially:

$$p(x) = \sum_i p_i \mathcal{N}(x; x_i, P_i) \quad \text{(Gaussian mixtures)}$$

**Exercise** Show:

$$x = \sum_i p_i x_i$$

$$P = \sum_i p_i \left\{ P_i + (x_i - x)(x_i - x)^\top \right\}$$
2nd Order Approximation:

\[ p(x) = \sum_i p_H \mathcal{N}(x; x_i, P_i) \approx \mathcal{N}(x; \mathbb{E}_p[x], C_p[x]) \]

\[ \mathbb{E}_p[x] = \int dx x p(x) = \sum_i p_i \int dx x \mathcal{N}(x; x_i, P_i) = \sum_i p_i x_i =: \mathbf{x} \]

\[ C_p[x] = \int dx p(x) (x - \mathbb{E}_p[x]) (x - \mathbb{E}_p[x])^\top = \sum_i p_i \int dx (x - \mathbf{x})(x - \mathbf{x})^\top \mathcal{N}(x; x_i, P_i) \]

\[ = \sum_i p_i \int dx \left\{ (x - \mathbf{x})(x - \mathbf{x})^\top - 2(x - x_i)(x_i - x)^\top \right\} \mathcal{N}(x; x_i, P_i) \]

since we have:

\[ \int dx (x - x_i)(x_i - x)^\top \mathcal{N}(x; x_i, P_i) = 0 \]

\[ = \sum_i p_i \int dx \left\{ x x^\top - 2x x_i^\top + x_i x_i^\top + x_i x_i^\top - 2x_i x_i^\top + x x^\top \right\} \mathcal{N}(x; x_i, P_i) \]

\[ = \sum_i p_i \int dx \left\{ (x - x_i)(x - x_i)^\top + (x_i - x)(x_i - x)^\top \right\} \mathcal{N}(x; x_i, P_i) \]

\[ = \sum_i p_i \left\{ P_i + (x_i - x)(x_i - x)^\top \right\} = P \]