TSRT14: Sensor Fusion
Lecture 4

— Detection theory
— Filter theory

Gustaf Hendeby

gustaf.hendeby@liu.se

LINKÖPING UNIVERSITY

Le 4: detection and filter theory

Whiteboard:

- Detection theory
  - Notation overview
  - Neyman-Pearson's lemma
  - Detection tests for no model, linear model, and nonlinear model
- Derivation of Bayesian optimal filter

Slides:

- Detection summary and example
- Filtering model definitions
- General view of nonlinear filtering
- Overview of optimal and approximate filters
- CRLB for filtering

Lecture 3: summary

- CRLB Theorem: for any unbiased estimator $\hat{x}$,

$$\operatorname{cov}(\hat{x}) \succeq \mathcal{I}^{-1}(x^o),$$

  Fisher information matrix (FIM) $\mathcal{I}^{-1}(x)$.
- ML estimate *efficient*: asymptotically *unbiased* satisfying CRLB.
- Sensor networks: typical models TOA, TDOA, DOA, and RSS.

Detection Theory

Chapter 5 Overview

- Detection Theory
  - (Generalized) likelihood ratio ((G)LR) test.
  - Test statistics, $T(\mathbf{y})$.
- Classification
  - Choose between many hypothesis, pick the most likely one.
- Measurement association problem.

Hypothesis Tests

Hypothesis test in statistics:

$$\begin{aligned} H_0 &: \mathbf{y} = \mathbf{e} \\ H_1 &: \mathbf{y} = x + \mathbf{e} \\ & \mathbf{e} \sim p(\mathbf{e}) \end{aligned}$$

General model-based test (sensor clutter versus target present):

$$\begin{aligned} H_0 &: \mathbf{y} = \mathbf{e}^0 & \mathbf{e}^0 \sim p^0(\mathbf{e}^0) \\ H_1 &: \mathbf{y} = \mathbf{h}(x) + \mathbf{e}^1 & \mathbf{e}^1 \sim p^1(\mathbf{e}^1) \end{aligned}$$

Special case: Linear model $\mathbf{h}(x) = \mathbf{H}x$.

First Example: revisited

Detect if a target is present

$$y_1 = x + e_1, \quad \text{cov}(e_1) = R_1$$
$$y_2 = x + e_2, \quad \text{cov}(e_2) = R_2$$

$$\mathbf{y} = \mathbf{H}x + \mathbf{e}, \quad \text{cov}(\mathbf{e}) = \mathbf{R}, \quad \mathbf{H} = \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}$$

$$T(\mathbf{y}) = \mathbf{y}^T \mathbf{R}^{-T/2} \Pi \mathbf{R}^{-1/2} \mathbf{y}$$

$$\Pi = \mathbf{R}^{-T/2}\mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}\mathbf{R}^{-1/2} = \begin{pmatrix} 22.8 & 22.2 & 5.0 & 0.0 \\ 22.2 & 22.8 & 0.0 & 5.0 \\ 5.0 & 0.0 & 22.8 & -22.2 \\ -0.0 & 5.0 & -22.2 & 22.8 \end{pmatrix}$$

Numerical Simulation

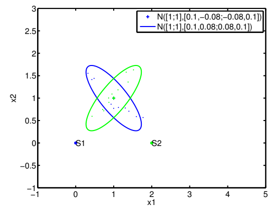
Threshold for the test:

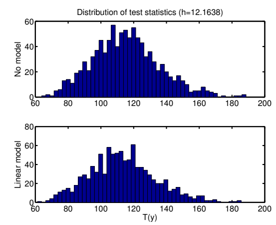
```
% NB! This invokes the chi2dist.erfinv function
% that is only remotely related to the
% mathematical erfinv function!
h = erfinv(chi2dist(2), 0.999)
```

Output:

```
% NB!  This is an approximation!
h =
   12.1638   % Correct value: 13.82
```

Note: For no model (standard statistical test), $\Pi = I$. Both methods perform perfect $P_D = 1$.





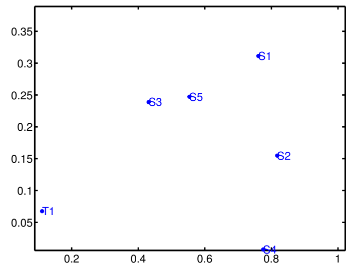
Sensor Network Example

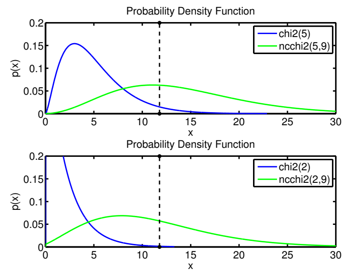
```
ny = 5; nx = 2;
s = exsensor('toa', ny, 1);  % Default network
s.pe = 0.2 * eye(ny);  % Set noise level
plot(s)
y = simulate(s);
T = y.y*(cov(s.pe)\y.y');  % Test statistic
[~, level] = detect(chi2dist(ny), T)
[~, level] = detect(s, y)
```

Output:

```
level =
    0.9626

level =
    0.9985
```





Filter Theory

Chapter 6 overview

- Dynamic state-space models: $x_{k+1} = f(x_k, v_k)$.
- Measurement model: $y_k = h(x_k, e_k)$.
- General Bayesian solution.
- Filtering bounds: parametric and posterior CRLB.

State-Space Models

Nonlinear model:

$$x_{k+1} = f(x_k, v_k) \quad \text{or} \quad x_{k+1}|x_k \sim p(x_{k+1}|x_k)$$
$$y_k = h(x_k, e_k) \quad \text{or} \quad y_k|x_k \sim p(y_k|x_k)$$

Nonlinear model with additive noise:

$$x_{k+1} = f(x_k) + v_k \quad \text{or} \quad x_{k+1}|x_k \sim p(x_{k+1}|x_k) = p_{v_k}\big(x_{k+1} - f(x_k)\big)$$
$$y_k = h(x_k) + e_k \quad \text{or} \quad y_k|x_k \sim p(y_k|x_k) = p_{e_k}\big(y_k - h(x_k)\big)$$

Linear model:

$$x_{k+1} = F_k x_k + G_k v_k$$
$$y_k = H_k x_k + e_k$$

Gaussian model: $v_k \sim \mathcal{N}(0, Q_k)$, $e_k \sim \mathcal{N}(0, R_k)$ and $x_0 \sim \mathcal{N}(0, P_0)$

Simple and Generic Motion Models (1/2)

Newton's force law $F = ma$ gives the "nearly constant velocity model" in n dimensions:

$$x_{k+1} = F_k x_k + G_k v_k = \begin{pmatrix} I_n & TI_n \\ 0_n & I_n \end{pmatrix} x_k + \begin{pmatrix} \frac{T^2}{2} I_n \\ TI_n \end{pmatrix} v_k$$

where $x_k = (p_k^T, V_k^T)^T$.
Interpretation:

$$p_{k+1} = p_k + TV_k + \frac{T^2}{2} v_k$$
$$V_{k+1} = V_k + T v_k$$

where process noise corresponds to acceleration, $v_k = a_k$.
Linear transformation of independent stochastic vectors implies:

$$x_{k+1} = F_k x_k + G_k v_k, \quad x_k \sim \mathcal{N}\big(\hat{x}_{k|k}, P_{k|k}\big), \quad v_k \sim \mathcal{N}\big(0, Q_k\big)$$
$$\implies x_{k+1} \sim \mathcal{N}\big(F\hat{x}_{k|k}, F_k P_{k|k} F_k^T + G_k Q_k G_k^T\big)$$

Simple and Generic Motion Models (2/2)

Similarly, a nearly constant acceleration model is

$$x_{k+1} = F_k x_k + G_k v_k = \begin{pmatrix} I_n & TI_n & \frac{T^2}{2} I_n \\ 0_n & I_n & TI_n \\ 0_n & 0_n & I_n \end{pmatrix} x_k + \begin{pmatrix} \frac{T^3}{6} I_n \\ \frac{T^2}{2} I_n \\ TI_n \end{pmatrix} v_k$$

More motion models in the next lecture.

Nonlinear Filtering: the easy way

Assumption:

There are more measurements than parameters/states.

- Suppose the vector x_k changes over time k.
- Here k denotes time, a possible space dimension is covered in y_k.

A general filter framework includes the iterations:

1. **Estimation:** Provides $(\hat{x}_k, P_k)$ by transforming the measurement y_k.
2. **Fusion:** The estimation information $\big(\hat{x}_k, P_k\big)$ is merged with the prior information $\big(\hat{x}_{k|k-1}, P_{k|k-1}\big)$. This gives $\big(\hat{x}_{k|k}, P_{k|k}\big)$.
3. **Transformation:** Propagate information through a motion model (*e.g.*, velocity compensation) $z = f(x_k)$. This gives $\big(\hat{z}, P_z\big)$.
4. **Diffusion:** Adding uncertainty from the motion model. This gives $\big(\hat{x}_{k+1|k}, P_{k+1|k}\big)$.

Bayes' Solution: nonlinear model with additive noise

Bayes law provides the recursion (measurement and time updates):

$$\alpha = \int_{\mathbb{R}^{n_y}} p_{e_k}\big(y_k - h(x_k)\big) p(x_k|y_{1:k-1})\, dx_k,$$
$$p(x_k|y_{1:k}) = \frac{1}{\alpha} p_{e_k}\big(y_k - h(x_k)\big) p(x_k|y_{1:k-1})$$
$$p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^{n_x}} p_{v_k}\big(x_{k+1} - f(x_k)\big) p(x_k|y_{1:k})\, dx_k$$

To get analytical solution, we need a model that keeps the same functional form of the posterior during:

- the nonlinear transformation $f(x_k)$.
- the addition of $f(x_k)$ and v_k.
- the inference of x_k from y_k done in the measurement update.

A General Bayesian Filter Framework

1. **Estimation:** Provides the complete distribution $p(x_k|y_k)$.
2. **Fusion:** Estimated information $p(x_k|y_k)$ is merged with the prior information $p(x_k|y_{1:k-1})$ to obtain $p(x_k|y_{1:k})$.
3. **Transformation:** Propagate information through the dynamics $z = f(x_k, u_k)$. This gives $p(z|y_{1:k})$.
4. **Diffusion:** Add uncertainty from the process noise. This gives $p(x_{k+1}|y_{1:k})$.

Practical Cases with Analytic Solution

Bayes solution can be represented with finite dimensional statistics analytically in the following cases:

- Linear Gaussian model (Kalman filter)
- Hidden Markov model (HMM)
- Linear-Gaussian mixture (Kalman filter filterbank; however exponential complexity in time)

General Approximation Approaches

1. Approximate the model to a case where an optimal algorithm exists.
   i.) *Extended KF* (EKF) which approximates the model with a linear one.
   ii.) *Unscented KF* (UKF) and EKF2 that apply higher order approximations.
2. Approximate the optimal nonlinear filter for the original model.
   i.) *Point-mass filter* (PMF) which uses a *regular* grid of the state space and applies the Bayesian recursion.
   ii.) *Particle filter* (PF) which uses a *random* grid of the state space and applies the Bayesian recursion.

CRLB: estimation

- The Fisher information matrix, $\mathcal{I}(x)$, is defined as

$$\mathcal{I}(x) = \mathsf{E}\Big(\nabla_x^T \log p_{\mathbf{e}}\big(\mathbf{y} - \mathbf{h}(x)\big)\nabla_x \log p_{\mathbf{e}}\big(\mathbf{y} - \mathbf{h}(x)\big)\Big)$$

$$\nabla_x \log p_{\mathbf{e}}\big(\mathbf{y} - \mathbf{h}(x)\big) = \begin{pmatrix} \frac{\partial \log p_{\mathbf{e}}(\mathbf{y}-\mathbf{h}(x))}{\partial x_1} & \dots & \frac{\partial \log p_{\mathbf{e}}(\mathbf{y}-\mathbf{h}(x))}{\partial x_{n_x}} \end{pmatrix}$$

- For Gaussian $\mathbf{e}$, then (compare with WLS covariance!)

$$\mathcal{I}(x) = \mathbf{H}^T(x)\mathbf{R}^{-1}(x)\mathbf{H}(x),$$
$$\mathbf{H}(x) = \nabla_x \mathbf{h}(x).$$

- Information is *additive*, so if two or more sensors give independent observations $y_k = h_k(x) + e_k$, then $\mathcal{I} = \sum_k \mathcal{I}_k$.
- CRLB provides a lower bound on root mean square error

$$\mathrm{RMSE} = \sqrt{\mathsf{E}\big((x_1^0 - \hat{x}_1)^2 + (x_2^0 - \hat{x}_2)^2\big)} = \sqrt{\mathrm{tr}\big(\mathsf{cov}(\hat{x})\big)} \geq \sqrt{\mathrm{tr}\big(\mathcal{I}^{-1}(x^0)\big)}$$

CRLB: filtering

- CRLB developed for static parameter x, with many measurements $y_{1:k}$.
- The filtering CRLB concerns the case where x is replaced with $x_{1:k}$, with the constraints $x_{n+1} = f(x_n) + v_n$, $n = 1, 2, \dots, k-1$.
- Two cases:
  - Parametric CRLB for filtering: $x_{1:k}$ is seen as a parameter with a *true value* $x^0_{1:k}$.
  - Posterior, or Bayesian, CRLB for filtering: $x_{1:k}$ is seen as a stochastic variable with a *prior* $p(x_{1:k})$.
- Parametric CRLB better in practice: easy to calculate, easy to interpret (given a certain trajectory and model, how well can a nonlinear filter estimate this trajectory?)
- Posterior CRLB useful for theoretical studies.

Parametric CRLB

- The parametric CRLB gives a lower bound on estimation error for a fixed trajectory $x_{1:k}$. That is, $\operatorname{cov}(\hat{x}_{k|k}) \succeq P^{\text{CRLB}}_{k|k}$.
- Algorithm identical to the Riccati equation (covariance update) in KF, where the gradients are evaluated along the trajectory $x_{1:k}$:

$$\begin{aligned}
P_{k+1|k} &= F_k P_{k|k} F_k^T + G_k Q_k G_k, \\
P_{k+1|k+1} &= P_{k+1|k} - P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R_k)^{-1} H_k P_{k+1|k}, \\
F_k &= \nabla_{x_k} f(x_k, v_k), \\
G_k &= \nabla_{v_k} f(x_k, v_k), \\
H_k &= \nabla_{x_k} h(x_k, e_k).
\end{aligned}$$

Posterior CRLB

- Average FIM over all possible trajectories $x_{1:k}$ with respect to v_k.
- Much more complicated expressions.
- For linear system, the parametric and posterior CRLB coincide.

Summary

Summary Lecture 4

- Detection problems as hypothesis tests:

$$H_0: \mathbf{y} = \mathbf{e},$$
$$H_1: \mathbf{y} = \bar{x} + \mathbf{e} = \mathbf{h}(x) + \mathbf{e}.$$

- Neyman-Pearson's lemma: $T(y) = p_{\mathbf{e}}\big(y - \mathbf{h}(x^0)\big)/p_{\mathbf{e}}(y)$ maximizes P_{D} for given P_{FA} (best ROC curve).
- In general case

$$T(y) = 2\log p_{\mathbf{e}}\big(y - \mathbf{h}(x^{\text{ML}})\big) - 2\log p_{\mathbf{e}}(y) \sim \chi^2_{n_x}\big(x^{0,T}\mathcal{I}(x^0)x^0\big).$$

- Bayes optimal filter

$$p(x_k|y_{1:k}) \propto p_{e_k}\big(y_k - h(x_k)\big)p(x_k|y_{1:k-1})$$
$$p(x_{k+1}|y_{1:k}) = \int p_{v_k}\big(x_{k+1} - f(x_k)\big)p(x_k|y_{1:k})\,dx_k.$$

- Intuitive work flow of nonlinear filter:
  - MU: estimation from $y_k = h(x_k) + e_k$ and fusion with $\hat{x}_{k|k-1}$.
  - TU: nonlinear transformation $z = f(x_k)$ and diffusion from $x_{k-1} = z_k + v_k$.