

TSRT14: Sensor Fusion

Lecture 4

- Detection theory
- Filter theory

Gustaf Hendeby

`gustaf.hendeby@liu.se`

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

- Sensor networks: typical models TOA, TDOA, DOA, and RSS.
- CRLB Theorem: for any unbiased estimator \hat{x} ,

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

Fisher information matrix (FIM) $\mathcal{I}^{-1}(x)$.

- ML estimate *efficient*: asymptotically *unbiased* satisfying CRLB.

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:

$$H_0 : \mathbf{y} = \mathbf{e}$$

$$H_1 : \mathbf{y} = \mathbf{x} + \mathbf{e}$$

$$\mathbf{e} \sim p(\mathbf{e})$$

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

$$H_0 : \mathbf{y} = \mathbf{e}^0 \quad \mathbf{e}^0 \sim p^0(\mathbf{e}^0)$$

$$H_1 : \mathbf{y} = \mathbf{h}(x) + \mathbf{e}^1 \quad \mathbf{e}^1 \sim p^1(\mathbf{e}^1)$$

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

First Example: revisited

Detect if a target is present

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

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

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

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

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

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

Numerical Simulation

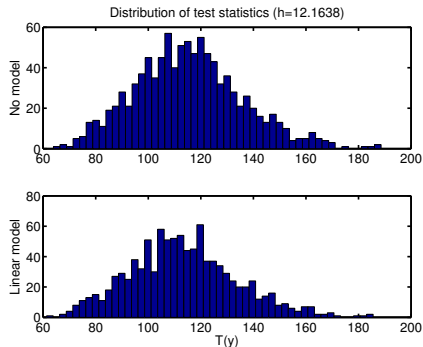
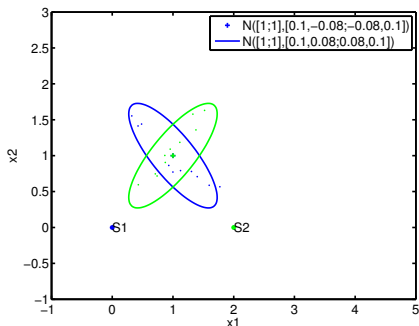
Threshold for the test:

```
% 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 $P_D = 1$.

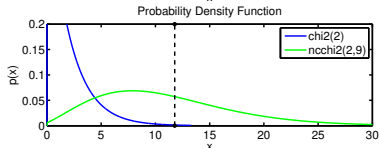
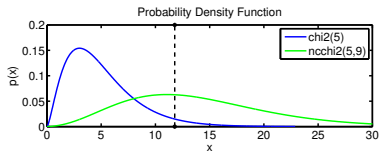
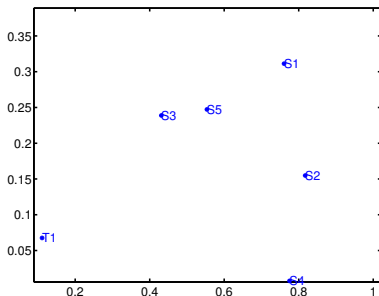


Sensor Network Example

```
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:

$$\begin{aligned}x_{k+1} &= f(x_k, v_k) & \text{or} & & p(x_{k+1}|x_k) \\y_k &= h(x_k, e_k) & \text{or} & & p(y_k|x_k)\end{aligned}$$

Nonlinear model with additive noise:

$$\begin{aligned}x_{k+1} &= f(x_k) + v_k & \text{or} & & p(x_{k+1}|x_k) = p_{v_k}(x_{k+1} - f(x_k)) \\y_k &= h(x_k) + e_k & \text{or} & & p(y_k|x_k) = p_{e_k}(y_k - h(x_k))\end{aligned}$$

Linear model:

$$\begin{aligned}x_{k+1} &= F_k x_k + G_k v_k \\y_k &= H_k x_k + e_k\end{aligned}$$

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:

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

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

Interpretation:

$$\begin{aligned} p_{k+1} &= p_k + T V_k + \frac{T^2}{2} v_k \\ V_{k+1} &= V_k + T v_k \end{aligned}$$

where process noise corresponds to acceleration, $v_k = a_k$.

Linear transformation of independent stochastic vectors implies:

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

Simple and Generic Motion Models (2/2)

Similarly, a nearly constant acceleration model is

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

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

Bayes' Solution: nonlinear model with additive noise

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

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

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

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

$$\nabla_x \log p_{\mathbf{e}}(\mathbf{y} - \mathbf{h}(x)) = \left(\frac{\partial \log p_{\mathbf{e}}(\mathbf{y} - \mathbf{h}(x))}{\partial x_1} \quad \dots \quad \frac{\partial \log p_{\mathbf{e}}(\mathbf{y} - \mathbf{h}(x))}{\partial x_{n_x}} \right)$$

- 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

$$\begin{aligned} \text{RMSE} &= \sqrt{\mathbf{E}((x_1^0 - \hat{x}_1)^2 + (x_2^0 - \hat{x}_2)^2)} = \sqrt{\text{tr}(\text{cov}(\hat{x}))} \\ &\geq \sqrt{\text{tr}(\mathcal{I}^{-1}(x^0))} \end{aligned}$$

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* $x_{1:k}^0$.
 - 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, $\text{cov}(\hat{x}_{k|k}) \succeq P_{k|k}^{\text{CRLB}}$.
- Algorithm identical to the Riccati equation (covariance update) in KF, where the gradients are evaluated along the trajectory $x_{1:k}$:

$$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k Q_k G_k,$$

$$P_{k+1|k+1} = P_{k+1|k} - P_{k+1|k} H_k^T (H_k P_{k+1|k} H_k^T + R_k)^{-1} H_k P_{k+1|k},$$

$$F_k = \nabla_{x_k} f(x_k, v_k),$$

$$G_k = \nabla_{v_k} f(x_k, v_k),$$

$$H_k = \nabla_{x_k} h(x_k, e_k).$$

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:

$$H_0 : \mathbf{y} = \mathbf{e},$$

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

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

- In general case

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

- Bayes optimal filter

$$p(x_k | y_{1:k}) \propto p_{e_k}(y_k - h(x_k)) p(x_k | y_{1:k-1})$$

$$p(x_{k+1} | y_{1:k}) = \int p_{v_k}(x_{k+1} - f(x_k)) p(x_k | y_{1:k}) dx_k.$$

- 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

$$x_{k-1} = z_k + v_k.$$

Gustaf Hendeby

hendeby@isy.liu.se

www.liu.se