Le 2: estimation theory in nonlinear models

Whiteboard:
- Nonlinear models
- Nonlinear weighted least squares (NWLS)
- NWLS connection to maximum likelihood (ML) estimation
- Nonlinear transform (NLT) and methods

Slides:
- Details on sensor models and methods
- Examples
- Dedicated least squares methods

Summary Lecture 1
- Linear model on batch form:
  \[ y = Hx + e, \quad \text{cov}(e) = R. \]
- WLS minimizes the loss function
  \[ V_{WLS}(x) = (y - Hx)^T R^{-1} (y - Hx). \]
- WLS solution
  \[ \hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} y, \quad P = (H^T R^{-1} H)^{-1}. \]
- LS special case with \( R \propto I \) and gives larger \( P \).
- The fusion formula for two independent estimates is
  \[ E(\hat{x}_1) = E(\hat{x}_2) = x, \quad \text{cov}(\hat{x}_1) = P_1, \quad \text{cov}(\hat{x}_2) = P_2 \Rightarrow \]
  \[ \hat{x} = P_1^{1/2} \hat{x}_1 + P_2^{1/2} \hat{x}_2, \quad P = (P_1^{1/2} + P_2^{1/2})^{-1}. \]
- If the estimates are not independent, \( P \) is larger than indicated.
Chapter 3 Overview

- Model $y_k = h_k(x) + e_k$
- Tool 1: Nonlinear least squares NLS
- NWLS minimization approaches
- Tool 2: Nonlinear transformations NLT
- NLT applied in a direct and an indirect approach
- Ranging sensor example
- Radar sensor example
- Partly linear models

NWLS Theory

Nonlinear model

\[ y_k = h_k(x) + e_k, \quad \text{cov}(e_k) = R_k, \quad k = 1, \ldots, N, \]
\[ y = h(x) + e, \quad \text{cov}(e) = R. \]

The NWLS solution minimizes

\[ \hat{x}_{\text{NWLS}} = \arg\min_x V_{\text{NWLS}}(x) = \arg\min_x \frac{1}{2} \sum_{k=1}^{N} (y_k - h_k(x))^T R_k^{-1} (y_k - h_k(x)). \]

ML for Gaussian noise with parameter dependent covariance $R(x)$

\[ \hat{x}_{\text{ML}} = \arg\min_x \left[ V_{\text{NWLS}}(x) + \frac{1}{2} \sum_k \log \det (R_k(x)) \right]. \]

Minimization Approaches

- **Grid**: evaluate $V(x)$ for a set of grid points $x^{(i)}$ and minimize.
- **Linearization**: first order Taylor expansion

\[ \tilde{y}_k = y_k - h_k(\bar{x}) + h'_k(\bar{x}) \bar{x} = h'_k(x) \bar{x} + e \]

and apply the WLS method to this linear model.
- **Optimization**: basic idea, iterate linearization and WLS. Gauss-Newton falls into this category.
- **Second order Taylor expansion**: compensation for mean and covariance in second order term possible in WLS.

Nonlinear Transformations (NLT)

**Problem**: given a nonlinear mapping

\[ z = g(u) \]

of a Gaussian variable

\[ u \sim \mathcal{N}(\mu_u, P_u), \]

how to approximate the output with a new Gaussian distribution

\[ z \sim \mathcal{N}(..., P_z). \]

Such approximations have two applications:
- **Direct approach**: apply NLT to $x = h^{-1}(y - e)$.
- **Indirect approach**: apply NLT to $y = h(x) + e$. 
NLT: Taylor methods

- **TT1:** first order Taylor transformation (a.k.a. Gauss approximation formula)
  \[ u \sim N(\mu_u, P_u) \rightarrow z \sim N(g(\mu_u), g'(\mu_u)P_u(g'(\mu_u))^T). \]

- **TT2:** second order Taylor transformation
  \[ u \sim N(\mu_u, P_u) \rightarrow z \sim N(g(\mu_u) + \frac{1}{2}[\text{tr}(g''(\mu_u)P_u)]v, \frac{1}{2}[\text{tr}(P_u g''(\mu_u)P_u g''(\mu_u))]) \]

NLT: sample methods

- **MCT:** Monte Carlo transformation
  \[ u^{(i)} \sim p_u(u^{(i)}), \quad i = 1, \ldots, N, \]
  \[ z^{(i)} = g(u^{(i)}), \]
  \[ \mu_z = \frac{1}{N} \sum_{i=1}^{N} z^{(i)}, \]
  \[ P_z = \frac{1}{N-1} \sum_{i=1}^{N} (z^{(i)} - \mu_z)(z^{(i)} - \mu_z)^T. \]

- **UT:** unscented transformation. Similar to MCT, but deterministic samples and other (non-intuitive) weights. Example comes later.

Direct Approach Using NLT

**Two step approach:**

Let \( x = z, g(u) = h^{-1}(u) \) and \( u = y - e \) in the general NLT.

1. NLT (TT1, TT2, MCT or UT) gives Gaussian approximation \( N(\hat{x}_k, P_k) \) for each sensor observation \( x = g(y - e) \).
2. The fusion formula gives \( N(\hat{x}, P) \) (with no further approximation).

Indirect Approach Using NLT

**General Bayesian approach to estimation:**

1. Assume a prior of \( x \sim N(\bar{x}, P^{xx}) \).
2. Form the stochastic vector in the NLT \( z = g(u) \) notation
   \[ u = \begin{pmatrix} x \\ e \end{pmatrix} \sim N(\begin{pmatrix} \bar{x} \\ 0 \end{pmatrix}, \begin{pmatrix} P^{xx} & 0 \\ 0 & R \end{pmatrix}). \]
3. Apply a NLT (TT1, TT2, MCT, UT) to the mapping
   \[ z = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ h(x, e) \end{pmatrix} \approx N(\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ P^{yx} & P^{yy} \end{pmatrix}). \]
4. Apply the formula (Lemma 7.1)
   \[ \hat{x} = \bar{x} + P^{xy}(P^{yy})^{-1}(y - \bar{y}), \]
   \[ \text{cov}(\hat{x}) = P^{xx} - P^{xy}(P^{yy})^{-1}P^{yx}. \]
Trilateration

- Two sensors measure range only.
- Typical application: time of arrival (TOA) detection.
- Estimate is the intersection of two circles (trilateration).
- What is the uncertainty/precision/covariance?

NWLS for TOA (1/2)

The likelihood function and the iterations in the NWLS estimate.

```
s = esensor('toa', 2);
s.th = th; s.x0 = x0;
% Change defaults
s.pe = 0.00001 * eye(2);  % Noise variance
y = simulate(s, 1);       % Generate observations
plot(s); hold on
lh2(s, y, 0:0.005:1, 0:0.005:.8);  % Likelihood function plot
plot2(shat.x0 + shat.px0, 'conf', 90,...
    'legend', 'off');  % Estimate and covariance plot
plot(res.TH(1,:), res.TH(2,:), 'r*-');  % Estimate for each iteration
axis([0, 1, 0, .8]);  % Plot network
```

Output:

```
SENSORMOD object: TOA (calibrated from data)
/ sqrt((x(1,1)-th(1))^2+(x(2,1)-th(2))^2) \
 y = sqrt((x(1,1)-th(3))^2+(x(2,1)-th(4))^2) \
th' = [0 0 0 0] x [0 0 0 0]
States: x1 x2
Outputs: y1 y2
```

NWLS for TOA (2/2)

Radar Observations

Sensor model:

\[
y = (r, \phi)^T + e = h(x_1, x_2) + e,
\]

\[
r = \sqrt{x_1^2 + x_2^2} + e_r,
\]

\[
\phi = \arctan2(x_1, x_2) + e_\phi.
\]

Direct approach by inverting the observation model

\[
x = h^{-1}(y - e),
\]

\[
x_1 = y_1 \cos(y_2) = (r - e_r) \sin(\phi - e_\phi),
\]

\[
x_2 = y_1 \sin(y_2) = (r - e_r) \cos(\phi - e_\phi).
\]

What is the covariance of \( \hat{x} = h^{-1}(y)? \)
Radar: Monte Carlo samples from direct method

- Generate measurements of range and bearing.
- Invert \( x = h^{-1}(y) \) for each sample.
- Banana shaped distribution of estimates.

\[
b_{inv} = @(R, \Phi, p) \begin{bmatrix} p(1) + R \cdot \cos(\Phi); \\ p(2) + R \cdot \sin(\Phi) \end{bmatrix};
\]

\[
R1 = ndist(100 * \sqrt{2}, 5); \\
Phi1 = ndist(\pi/4, 0.1); \\
p1 = [0; 0];
\]

\[
R2 = ndist(100 * \sqrt{2}, 5); \\
Phi2 = ndist(3 * \pi/4, 0.1); \\
p2 = [200; 0];
\]

\[
xhat1 = b_{inv}(R1, Phi1, p1); \\
xhat2 = b_{inv}(R2, Phi2, p2);
\]

\[
plot(p1(1), p1(2), '.b', 'markersize', 15); \\
hold on; \\
text(p1(1), p1(2), 'S1'); \\
text(p2(1), p2(2), 'S2'); \\
plot2(xhat1, xhat2, 'legend', 'off');
\]

Radar: analytic direct approach

Analytic approximation of \( \text{cov}(x) \).

For the radar sensor with \( \hat{x} = h^{-1}(y) \), the covariance can be approximated with

\[
\text{cov}(\hat{x}) = \sigma_r^2 - \frac{r^2\sigma_\phi^2}{\sigma_r^2} \left( \frac{b + \cos(2\phi)}{\sin(2\phi)} \right) \left( \frac{b - \cos(2\phi)}{\sin(2\phi)} \right)
\]

\[
b = \frac{\sigma_r^2 + r^2\sigma_\phi^2}{\sigma_r^2 - r^2\sigma_\phi^2}.
\]

Approximation accurate if \( r\sigma_r^2/\sigma_\phi < 0.4 \) and \( \sigma_\phi < 0.4 \).
This is normally the case in radar applications.
It does not hold in the example where \( r\sigma_r^2/\sigma_\phi = 100\sqrt{2} \cdot 0.1/\sqrt{5} \approx 6.3 \).

Radar: direct approach with MC

Fit a Gaussian to the Monte Carlo samples of \( y_k \) and apply the sensor fusion formula to the two Gaussian distributions.

\[
\text{Output:}
\]

\[
\hat{x} = \begin{bmatrix} 100; 100 \end{bmatrix}, \sigma = \begin{bmatrix} 100 & 100; 100 & 100 \end{bmatrix}
\]

Radar: direct approach with TT1

Gauss approximation formula (based on linearizing \( h^{-1}(x) \)) applied to the banana transformation gives too optimistic result (since higher order terms are neglected).

\[
\text{Output:}
\]

\[
\hat{x} = \begin{bmatrix} 100; 100 \end{bmatrix}, \sigma = \begin{bmatrix} 100 & 100; 100 & 100 \end{bmatrix}
\]
Unscented Transformation

Method for transforming mean and covariance of $y$ to $x = g(y)$:
1. Compute the sigma points $y^{(i)}$. These are the mean and symmetric deviations around the mean computed from the covariance matrix of $y$.
2. The sigma points are mapped to $x^{(i)} = h^{-1}(y^{(i)})$.
3. The mean and covariance are fitted to the mapped sigma points

$$
\mu_x = \frac{1}{N} \sum_{i=1}^{N} \omega^i_m x^{(i)},
$$

$$
P_x = \frac{1}{N} \sum_{i=1}^{N} \omega^i c(x^{(i)} - \mu_x)(x^{(i)} - \mu_x)^T.
$$

Tricks and rule of thumbs available to tune the weights. Can be seen as a Monte Carlo method with deterministic sampling.

Radar: direct approach with UT (1/2)

Left: Distribution of $y = (r, \phi)$ and sigma points $y^{(i)}$.
Right: Transformed sigma points $x^{(i)} = h^{-1}(y^{(i)})$ and fitted Gaussian distribution $N(\hat{x}_k, P_k)$.

Blue: left sensor
Green: right sensor
Red: fused estimate

Radar: direct approach with UT (2/2)

Output:

```
[hat1, S1, F1] = uteval(y1, hinv, 'ut1', [], p1);
plot(y1, y2, 'legend', 'off');
hold on;
plot(S1(1,:), S1(2,:), 'xb', 's1');
plot(S2(1,:), S2(2,:), 'dg');
plot(p1(1), p1(2), 'b', 'p1');
hold on;
plot(p2(1), p2(2), 'r', 'p2');
plot2(hat1, hat2, 'legend', 'off');
hold on;
plot(F1(1,:), F1(2,:), 'xb', 'f1');
plot(F2(1,:), F2(2,:), 'dg');
```

Radar: indirect approach with TT1 (1/2)

Gauss approximation formula (based on linearizing $h_k(x)$). The result is overly optimistic as higher order terms are neglected (cf. the direct TT1 approach).

Output:

```
[hat1, H1] = Hz([95.1; 95.1], [954, -854; -854, 954]);
S1 = [141.4214 145.2943 141.4214 137.5484 141.4214; 0.7854 0.7854 1.3331 0.7854 0.2377];
F1 = [100.0000 102.7386 33.2968 97.2614 137.4457; 100.0000 102.7386 137.4457 97.2614 33.2968];
```

```
[hat2, H2] = Hz([105; 95.1], [954, 854; 854, 954]);
S2 = [141.4214 145.2943 141.4214 137.5484 141.4214; 2.3562 2.3562 2.9039 2.3562 1.8085];
F2 = [100.0000 97.2614 62.5543 102.7386 166.7032; 100.0000 102.7386 33.2968 97.2614 137.4457];
```
Radar: indirect approach with TT1 (2/2)

\[ g = @(x, p) \begin{bmatrix} x(1:2); \\
% Function for joint distribution 
\text{hypot}(x(1) - p(1), x(2) - p(2)) + x(3); \\
\text{atan2}(x(2) - p(2), x(1) - p(2)) + x(4); \\
\end{bmatrix}; \]

\[ X_{hat0} = \text{ndist}([100; 100], 400*\text{eye}(2)) \]

\[ e_1 = \text{ndist}([0; 0], \text{cov}(y_1)); \]

\[ X_{hat1} = \text{ndist}(X_{hat1}, P_1) \]

\[ e_2 = \text{ndist}([0; 0], \text{cov}(y_2)); \]

\[ X_{hat2} = \text{ndist}(X_{hat2}, P_2) \]

Conditioned Linear Models

\[ y_k = h_k^N(x_n) + h_k^l(x_n)x_l + e_k, \quad \text{cov}(e_k) = R_k(x_n), \quad V_{\text{NWLS}}(x_n, x_l) \]

Separable least squares: The WLS solution for \( x_l \) is explicitly given by

\[ \hat{x}_{\text{WLS}}(x_n) = \left( \sum_{k=1}^{N} (h_k^l(x_n))^T R_k^{-1}(x_n) h_k^l(x_n) \right)^{-1} \sum_{k=1}^{N} (h_k^l(x_n))^T R_k^{-1}(x_n) (y_k - h_k^N(x_n)). \]

for each value of \( x_n \).

Which one to choose?

- Almost always utilize the separable least squares principle.
- In some cases, the loss function \( \min_{x_n} V_{\text{WLS}}(x_n, \hat{x}(x_n)) \) might have more local minima than the original formulation.

Summary

Nonlinear model

\[ y = h(x) + e, \quad \text{cov}(e) = R. \]

NWLS minimizes

\[ \hat{x}_{\text{NWLS}} = \arg\min_x V_{\text{NWLS}}(x) = \arg\min_x \frac{1}{2}(y - h(x))^T R^{-1}(y - h(x)) \]

Linearization principle, replace \( y_k = h_k(x) + e_k \) with \( \hat{y}_k = y_k - h_k(\bar{x}) + h_k'(\bar{x})x = h_k'(\bar{x})x + e \) and apply the WLS method to this linear model. Extensions:

- Iterate the linearization process.
- Compensate for the bias and variance of the rest term.

Model with linear sub-structure \( h(x) = h^N(x_n) + h^l(x_n)x_l \); \( x_l \) can be eliminated with WLS, leading to a smaller search space.
Summary Lecture 2 (2/2)

NLT: Approximate $z = g(u)$, $u \sim \mathcal{N}(\hat{u}, P_u)$ with $z \sim \mathcal{N}(\hat{z}, P_z)$.

Variations: TT1, TT2, UT or MCT.

- The direct approach, where $x = h^{-1}(y - e)$ is approximated.
- The indirect approach, where the distribution of $y = h(x)$ is approximated using a prior of $x \sim \mathcal{N}(\hat{x}, P^{xx})$: The trick is to consider the mapping

$$u = \begin{pmatrix} x \\ e \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} \hat{x} \\ 0 \end{pmatrix}, \begin{pmatrix} P^{xx} & 0 \\ 0 & R \end{pmatrix})$$

$$z = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ h(x, e) \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ P^{yx} & P^{yy} \end{pmatrix})$$

and then apply

$$\hat{x} = \hat{x} + P^{xy}(P^{yy})^{-1}(y - \hat{y}),$$

$$\text{cov}(\hat{x}) = P^{xx} - P^{xy}(P^{yy})^{-1}P^{yx}.$$