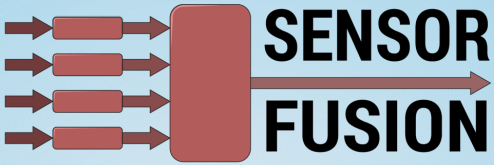


Parameter Estimation using Nonlinear Transformations

Sensor Fusion

Fredrik Gustafsson
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

Background: NLT

Approximate nonlinear transformations (NLTs) of stochastic variables will be used to estimate the parameter x.

Nonlinear transfomation (NLT)

Given the transform

$$z = g(u)$$

and the mean and covariance of the input,

$$\mathrm{E}(u) = \mu_u, \quad \mathrm{Cov}(u) = P_u \quad \text{(often approximated } u \sim \mathcal{N}(\mu_u, P_u)\text{)}$$

find

$$\mathrm{E}(z) = \mu_z \quad \mathrm{Cov}(z) = P_z \quad \text{(often approximated } z \sim \mathcal{N}(\mu_z, P_z)\text{)}.$$

Measurements:

Here, independent nonlinear measurements y_k as a function of the parameter x are considered, $y_k = h(x) + e_k$ with $\mathrm{Cov}(e_k) = R$.

Direct Approach Using NLT

Assuming $x = h^{-1}(y - e)$ is available, the direct method can be applied.

Direct estimation

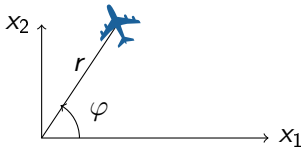
Let $x = z$, $g(u) = h^{-1}(u)$ and $u = y - e$ in the general NLT formulation. Then:

- NLT (TT1, TT2, MCT or UT) gives a Gaussian approximation of the parameter $\mathcal{N}(\hat{x}_k, P_k)$ for each sensor observation $x = g(y_k)$.
- Several estimates can then be combined using the sensor fusion formula to get $\mathcal{N}(\hat{x}, P)$ (without further approximations).

Radar Example: observations

Sensor model:

$$y = (r, \varphi)^T + e = h(x_1, x_2) + e,$$
$$r = \sqrt{x_1^2 + x_2^2} + e_r,$$
$$\varphi = \arctan2(x_1, x_2) + e_\varphi.$$



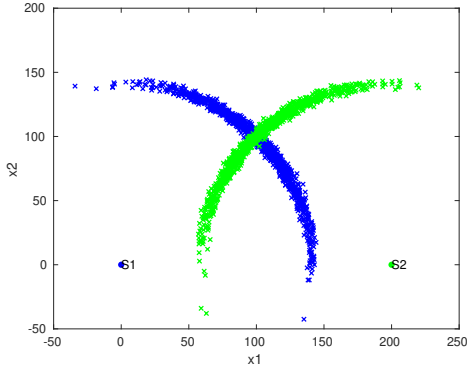
Direct approach, inverting the observation model

$$x = h^{-1}(y - e),$$
$$x_1 = y_1 \cos(y_2) = (r - e_r)\sin(\varphi - e_\varphi),$$
$$x_2 = y_1 \sin(y_2) = (r - e_r)\cos(\varphi - e_\varphi).$$

What is the covariance of $\hat{x} = h^{-1}(y)$?

Radar Example: Monte Carlo samples

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



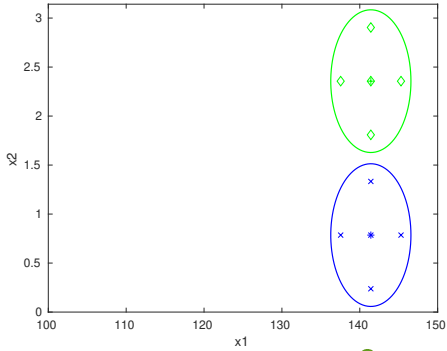
Matlab (SigSys):

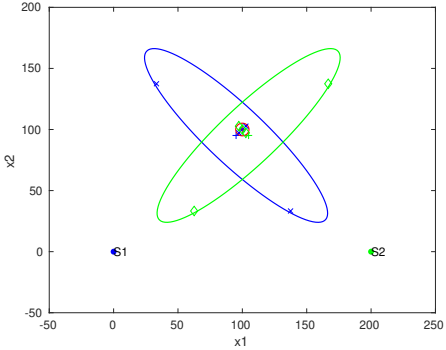
```
hinv= @(R, Phi, p) [p(1) + R * cos(Phi);
                    p(2) + R * sin(Phi)];

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 = hinv(R1, Phi1, p1);
xhat2 = hinv(R2, Phi2, p2);

plot(p1(1), p1(2), '.b',...
     p2(1), p2(2), '.g',...
     'markersize', 15);
hold on;
text(p1(1), p1(2), 'S1');
text(p2(1), p2(2), 'S2');

plot2(xhat1, xhat2, 'legend', 'off');
```

Radar Example: direct approach with UT (1/2)

Left: Distribution of $y = (r, \varphi)$ and sigma points $y^{(i)}$.

Right: Transformed sigma points $x^{(i)} = h^{-1}\left(y^{(i)}\right)$ and fitted Gaussian distribution $\mathcal{N}\left(\hat{x}_k, P_k\right)$.





Blue: left sensor

Green: right sensor

Red: fused estimate

Radar Example: direct approach with UT (2/2)

Matlab (SigSys):

```
[Nhat1, S1, fS1] = uteval(y1, hinv, 'ut1', [], p1)
[Nhat2, S2, fS2] = uteval(y2, hinv, 'ut1', [], p2)

plot2(y1, y2, 'legend', 'off');
hold on;
plot(S1(1, :), S1(2, :), 'xb',...
     S2(1, :), S2(2, :), 'dg');
%%
plot(p1(1), p1(2), '.b',...
     p2(1), p2(2), '.g',...
     'markersize', 15);
hold on;
text(p1(1), p1(2), 'S1');
text(p2(1), p2(2), 'S2');

plot2(Nhat1, Nhat2, xhat,...
      'legend', 'off');
hold on;
plot(fS1(1, :), fS1(2, :), 'xb',...
     fS2(1, :), fS2(2, :), 'dg');
```

Output:

```
Nhat1 =
N([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
fS1 =
  100.0000   102.7386    33.2968    97.2614   137.4457
  100.0000   102.7386   137.4457    97.2614    33.2968
Nhat2 =
N([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
fS2 =
  100.0000    97.2614    62.5543   102.7386   166.7032
  100.0000   102.7386    33.2968    97.2614   137.4457
```

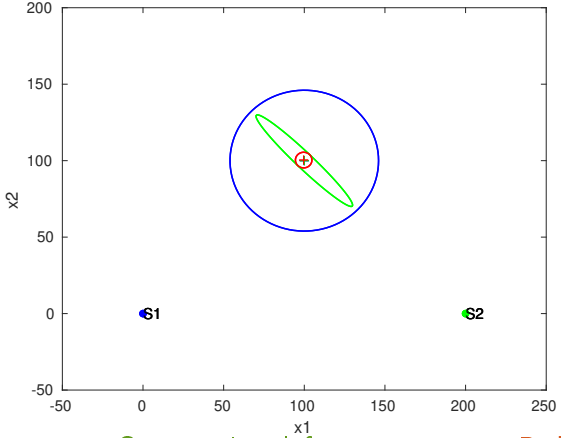
Indirect Approach Using NLT

General Bayesian approach to estimation

1. Assume $y = h(x) + e$, and a prior of $x \sim \mathcal{N}(\hat{x}_0, P^{xx})$.
2. Form the stochastic vector in the NLT $z = g(u)$ notation
$$u = \begin{pmatrix} x \\ e \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_0 \\ 0 \end{pmatrix}, \begin{pmatrix} P^{xx} & 0 \\ 0 & R \end{pmatrix}\right).$$
3. Apply an NLT (TT1, TT2, MCT, UT) to the mapping
$$z = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ h(x,e) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_0 \\ \hat{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ P^{yx} & P^{yy} \end{pmatrix}\right).$$
4. Apply the formula (Lemma 7.1, discussed later)
$$\hat{x}_k = \hat{x}_0 + P^{xy}(P^{yy})^{-1}(y_k - \hat{y})$$
$$\mathrm{Cov}(\hat{x}_k) = P^{xx} - P^{xy}\left(P^{yy}\right)^{-1}P^{yx}.$$

Radar Example: 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.



Blue: prior

Green: prior+left meas.

Red: prior+all meas.

Radar Example: indirect approach with TT1 (2/2)

Matlab (SigSys):

```
g = @(x, p) [ x(1:2);                    % Function for joint distribution
              hypot(x(1)-p(1), x(2)-p(2)) + x(3);
              atan2(x(2)-p(2), x(1)-p(2)) + x(4); ];
Xhat0 = ndist([100; 100], 400*eye(2))   % Prior

e1 = ndist([0; 0], cov(y1));             % Approximate the joint distribution
U1 = tt1eval([Xhat0; e1], g, p1);
[xhat1, P1] = condition(U1, mean(y1), [3, 4]);  % Perfrom conditioning
Xhat1 = ndist(xhat1, P1)

e2 = ndist([0; 0], cov(y2));
U2 = tt1eval([Xhat1; e2], g, p2);        % Approximate the joint distribution
[xhat2, P2] = condition(U2, mean(y2), [3, 4]);  % Perfrom conditioning
Xhat2 = ndist(xhat2, P2)

plot(p1(1), p1(2), '.b',...
plot2(Xhat0, Xhat1, Xhat2, 'legend', 'off');
```

Output:

```
Xhat0 =
N([100;100],[400,0;0,400])
Xhat1 =
N([100;100],[169,-164;-164,169])
Xhat2 =
N([99.6;100],[4.93,0.0121;0.0121,4.93])
```

Summary: estimation using NLT

- *Direct approach*: $x = \mathbf{h}^{-1}(\mathbf{y} - \mathbf{e})$ is approximated.
- *Indirect approach*: The distribution of $\mathbf{y} = \mathbf{h}(x)$ is approximated using a prior of $x \sim \mathcal{N}(\hat{x}_0, P^{xx})$.
  Compute the following transformation using a NLT

$$u = \begin{pmatrix} x \\ e \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_0 \\ 0 \end{pmatrix}, \begin{pmatrix} P^{xx} & 0 \\ 0 & R \end{pmatrix}\right)$$

$$z = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ h(x,e) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_0 \\ \hat{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ P^{yx} & P^{yy} \end{pmatrix}\right)$$

and then

$$\hat{x}_k = \hat{x}_0 + P^{xy}\left(P^{yy}\right)^{-1}(y_k - \hat{y}) \qquad P_x = P^{xx} - P^{xy}\left(P^{yy}\right)^{-1}P^{yx}.$$

Sections 3.4 (first part) and 3.5