Solutions for examination in Sensor Fusion, 2024-01-02

1. (a) **Accelerometer: $\boldsymbol{y}^{\text{acc}} = \begin{pmatrix} 0 \\ 0 \\ 9.8 \end{pmatrix}\,\text{ms}^{-2}$. Gyroscope: $\boldsymbol{y}^{\text{gyr}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\,\text{rad}\,\text{s}^{-1}$**
   Magnetometer: The disturbances are expected to dominate the result.

   (b) **(ii) and (vi)**

   (c)
$$\hat{\boldsymbol{x}} = \Big(\sum\nolimits_{i=1}^{N} \boldsymbol{H}_i^T \boldsymbol{R}^{-1} \boldsymbol{H}_i\Big)^{-1} \sum\nolimits_{i=1}^{N} \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_i)$$

   This is the standard WLS estimate, once the measurements are compensated for the mean of the measurement noise.
$$y_i = H_i x + e_i = H_i x + \mu_i + \tilde{e}_i \Leftrightarrow y_i - \mu_i = H_i - \tilde{e}_i,$$
   where $\tilde{e}_i$ is identical to e_i except that $\mathsf{E}\,\tilde{e}_i = 0$.

   (d) **$\hat{\boldsymbol{x}}$ remains the same, $\boldsymbol{P}$ is scaled by a factor $\boldsymbol{k}$**

   The estimates $\hat{x}$ does only depend on the ratio Q/R, whereas P scales with k. A reason for scaling R and Q could be that information about the actual uncertainty of the estimate is needed, *e.g.*, for outlier rejection or when the estimate is used.

   (e) **(ii), (iii), (v)**

2.
```
%% Exercise 2
load data20240102.mat
N = numel(ex2_t);
fs = 1;
%% a
% Create the sensor model and set appropriate noise levels, as described
% in the exercise.
R = 1e-3*eye(3);
S =  [-1,  0, 1;
       0,  0, 0];
sm = exsensor('toa', 3, 1, 4);
sm.th = S(:);
sm.pe = R*eye(3);

% Estimate the first position, eg, using nls (make sure to estimat only x)
xhat_a = nls(sm, sig(ex2_y(:, 1)'), 'thmask', zeros(1, 6));
figure(1); clf;
plot(xhat_a, 'conf', 90)
axis([-1.5 1.5 -.5 6]);

%% b
% Define the given motion model
mm = exmotion('cv2d');
mms = addsensor(mm, sm);
mms.x0 = [xhat_a.x0' 0 0];  % Initialize around the previous estimate
mms.px0 = blkdiag(10*cov(xhat_a.px0), 1*eye(2));
T = 1;
G = [T^2/2 0; 0 T^2/2; T 0; 0 T];
% This is a tuning parameter, in this case the target seems to not be
% affected by much process noise.
Q = 1e-2*eye(2);
mms.pv = G*Q*G';

% Apply a extended Kalman filter
xhat_ekf = ekf(mms, sig(ex2_y'));

% Apply a particle filter.  Here the number of particles is a tuning
% parameter, the proposal another.  If that SIR proposal is used, the
% number of particles must be quite high (~1e6) to produce accaptable
% results, whereas with an optimal proposal, as used here, we get away with
% much less particles.
Np = 1e2;
xhat_pf =  pf(mms, sig(ex2_y'), 'Np', Np, 'proposal', 'opt');
```

```
figure(2); clf; sm.x0 = [];
plot(sm); hold on;
h = xplot2(xhat_ekf, xhat_pf, 'conf', 90);
```

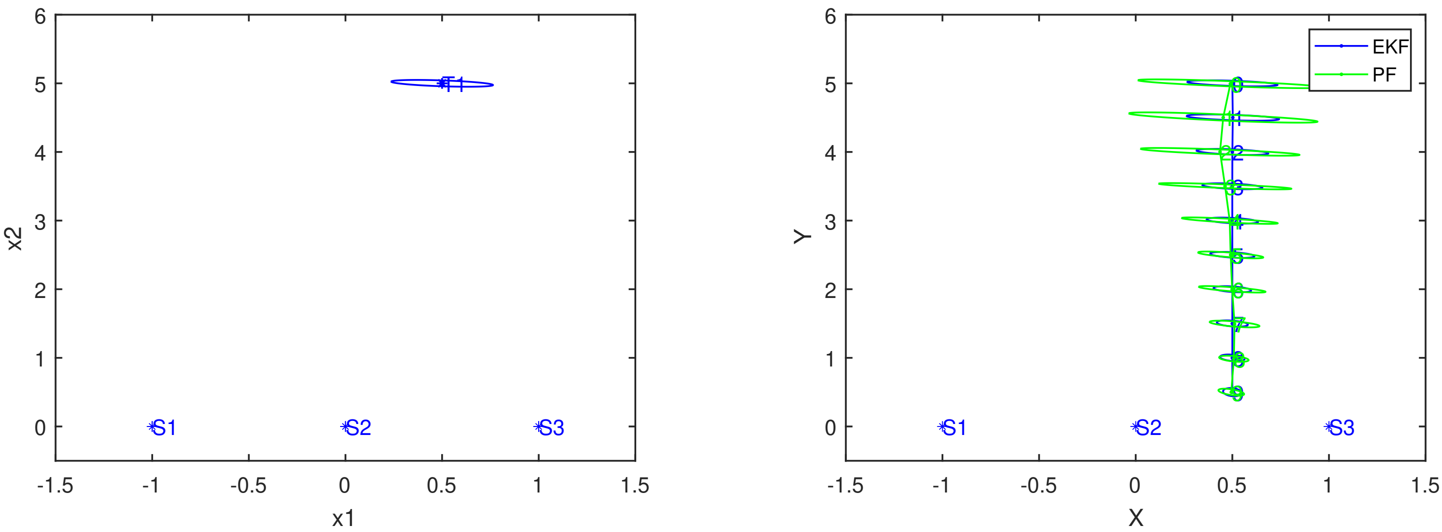


Figure 1: Resulting figures Exercise 2.

3. (a) The likelihood of y_k will have the mean Hx and the variance λ

$$p(y_k|x,\lambda) = \mathcal{N}(y; Hx, \lambda) = \frac{1}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2\lambda}(y_k - Hx)^2}.$$

The log-likelihood is

$$\log p(y_k|x,\lambda) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\lambda - \frac{1}{2\lambda}(y_k - Hx)^2.$$

The derivative with respect to λ can now be computed as

$$\frac{d\log\big(p(y_k|x,\lambda)\big)}{d\lambda} = -\frac{1}{2\lambda} + \frac{1}{2\lambda^2}(y_k - Hx)^2.$$

Notice that it is the same to maximize the likelihood as maximizing the log likelihood since the logarithm is a monotonically increasing function. The ML estimate of λ for a fixed value y_1 and x will then be

$$\arg\max_{\lambda} p(y_1|x,\lambda) = \arg\max_{\lambda} \log\big(p(y_1|x,\lambda)\big).$$

We find the maximum by setting the derivative to zero

$$\begin{aligned} \frac{d\log p(y_1|x,\lambda)}{d\lambda} &= 0 \Rightarrow \\ -\frac{1}{2\lambda} + \frac{1}{2\lambda^2}(y_1 - Hx)^2 &= 0 \Rightarrow \\ \hat{\lambda} &= (y_1 - Hx)^2. \end{aligned}$$

We also notice that this is a maximum since the second derivative is negative.

$$\begin{aligned} \frac{d^2\log p(y_k|x,\lambda)}{d\lambda^2} &= \frac{1}{2\lambda^2} - \frac{1}{\lambda^3}(y_k - Hx)^2 \Rightarrow \\ \frac{d^2\log p(y_k|x,\lambda)}{d\lambda^2}\bigg|_{\lambda=(y_1-Hx)^2} &= -\frac{1}{2(y_1 - Hx)^4} < 0 \end{aligned}$$

We then have that the ML estimate is

$$\hat{\lambda}_{\text{ML}} = (y - Hx)^2$$

(b) The CRLB is the inverse of the FIM. The FIM of λ can be computed in two ways using definitions from the book

Alternative 1:

$$\begin{aligned}\mathcal{I}(\lambda) &= \mathsf{E}\left[\left(\frac{d\log\left(p(y|x,\lambda)\right)}{d\lambda}\right)^2\right] = \mathsf{E}\left[\left(-\frac{1}{2\lambda}+\frac{1}{2\lambda^2}(y-Hx)^2\right)^2\right] \\ &= \mathsf{E}\left[\frac{1}{4\lambda^2}-\frac{1}{2\lambda^3}(y-Hx)^2+\frac{1}{4\lambda^4}(y-Hx)^4\right] = \mathsf{E}\left[\frac{1}{4\lambda^2}-\frac{1}{2\lambda^3}e^2+\frac{1}{4\lambda^4}e^4\right] \\ &= \frac{1}{4\lambda^2}-\frac{\lambda}{2\lambda^3}+\frac{3\lambda^2}{4\lambda^4} = \frac{1}{4\lambda^2}-\frac{2}{4\lambda^2}+\frac{3}{4\lambda^2} = \frac{1}{2\lambda^2}\end{aligned}$$

where the fourth order moment $\mathsf{E}(e^4) = 3\lambda^2$ for a Gaussian variable has been used.

Alternative 2:

$$\begin{aligned}\mathcal{I}(\lambda) &= -\mathsf{E}\left[\frac{d^2\log\left(p(y|x,\lambda)\right)}{d\lambda^2}\right] = -\mathsf{E}\left[\frac{1}{2\lambda^2}-\frac{1}{\lambda^3}(y-Hx)^2\right] \\ &= -\left(\frac{1}{2\lambda^2}-\frac{1}{\lambda^3}\lambda\right) = \frac{1}{2\lambda^2}\end{aligned}$$

The CRLB can now be stated as

$$\operatorname{var}\hat{\lambda} \geq (\mathcal{I}(\lambda))^{-1} = \left(\frac{1}{2\lambda^2}\right)^{-1} = 2\lambda^2$$

(c) Similarly, the FIM of λ in the batch case can be computed in two ways.

Alternative 1: In a similar fashion as before we have

$$\begin{aligned}p(y_{1:N}|x,\lambda) &= \frac{1}{(2\pi\lambda)^{(N/2)}}e^{-\frac{1}{2\lambda}\sum_{k=1}^{N}(y_k-Hx)^2} \Rightarrow \\ \mathcal{I}_{1:N}(\lambda) &= -\mathsf{E}\left[\frac{d^2\log p(y_{1:N}|x,\lambda)}{d\lambda^2}\right] \\ &= -\mathsf{E}\left[\frac{N}{2\lambda^2}-\frac{1}{\lambda^3}\sum_{k=1}^{N}(y_k-Hx)^2\right] \\ &= -\left(\frac{N}{2\lambda^2}-\frac{1}{\lambda^3}N\lambda\right) = \frac{N}{2\lambda^2}\end{aligned}$$

Alternative 2: Since information is additive for independent observations, we have

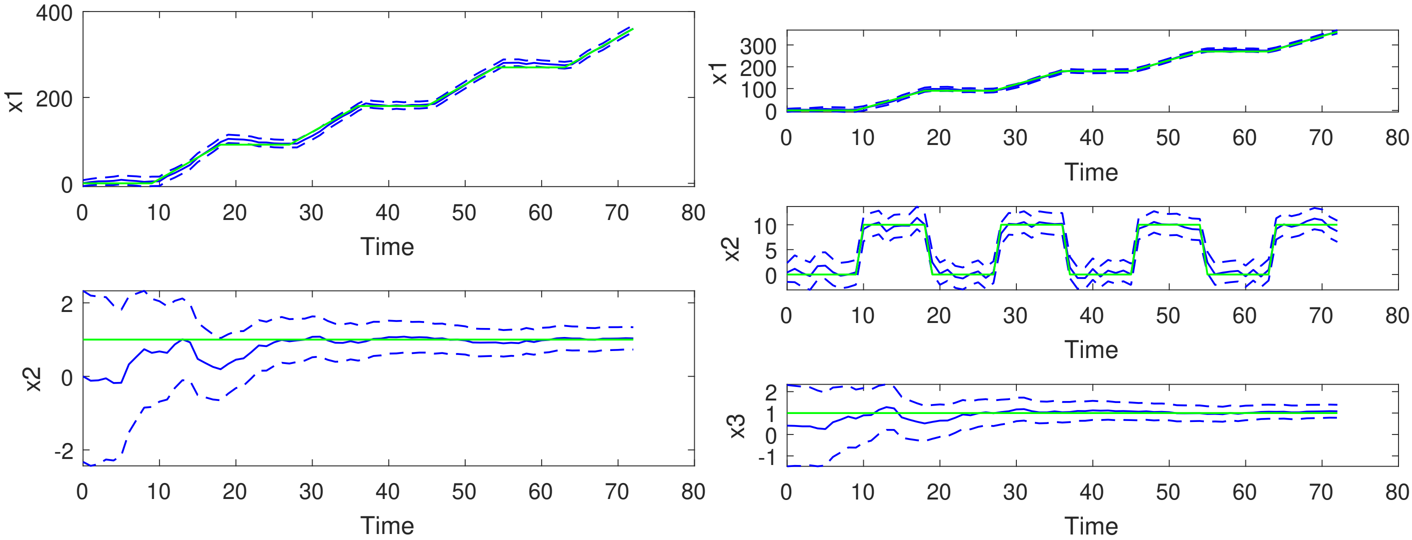
$$\mathcal{I}_{1:N}(\lambda) = \sum_{k=1}^{N}\mathcal{I}_k(\lambda) = \sum_{k=1}^{N}\frac{1}{2\lambda^2} = \frac{N}{2\lambda^2}$$

The CRLB can now be stated as

$$\operatorname{var}\hat{\lambda} \geq (\mathcal{I}_{1:N}(\lambda))^{-1} = \left(\frac{N}{2\lambda^2}\right)^{-1} = \frac{2\lambda^2}{N}$$

4.

```
clear all; close all
load data20240102

% Measuerment noise levels according to the question.
Rh = 10^2;
Rw = 1^2;
%% Ex 4a
% As in lab 2, use the rate measurement as input to drive the model.  The
% state-space model based on the measurements becomes:
%  x_k = [h_k + T_s*(y_k^w - b)] - e_k^w
%        [b_k                  ]
%  y_k = y_k^h = h_k + e_k^h
%  Which translates into the linear Gaussian model:
Fa = [1 -1; 0 1]; Ga = [1; 0]; Qa = diag([Rw 0]);
Ha = [1 0]; Ra = Rh;
Ma = lss(Fa, Ga, Ha, [], Qa, Ra, 1);
P0a = diag([10, 1]);  % Set the initial uncertainty pretty high
% The measurement object matching this is:
Ya = sig(ex4_yh', 1, ex4_yw');

% Apply a Kalman filter
Xhata = kalman(Ma, Ya, 'alg', 2, 'P0', P0a);

% Plot the result (make new sig-objects to compare with ground truth)
Yaplot = sig(ex4_yh', 1, [], [ex4_h; ex4_b*ones(size(ex4_h))]');
figure(1); clf;
xplot(Xhata, Yaplot, 'conf', 99);

print(1, '-depsc', fullfile('fig', 'ex4a'));

%% Ex 4b
% Now instead use the rate measurement as a regular measurement, this
% yields the state-space model:
%        [h_k + T_s*w_k] + [T_s^2/2]
%  x_k = [w_k          ] + [T_s    ] * v_k
%        [b_k          ] + [0      ]
%  y_k = [y_k^h] = [h_k      ] + e_k^h
%        [y_k^w]   [w_k + b_k] + e_k^w
%  Which translates into the linear Gaussian model:
Fb = [1 1 0; 0 1 0; 0 0 1]; Gb = zeros(3, 0);
q = 2^2;  % Tune this to get a reasonable result.
Qb = q*[.5; 1; 0]*[.5; 1; 0]';
Hb = [1 0 0; 0 1 +1]; Rb = diag([Rh, Rw]);
Mb = lss(Fb, Gb, Hb, [], Qb, Rb, 1);
P0b = diag([10, 1, 1]);  % Set the initial uncertainty pretty high
% The measurement object matching this is:
Yb = sig([ex4_yh; ex4_yw]', 1);

% Apply a Kalman filter
Xhatb = kalman(Mb, Yb, 'alg', 2, 'P0', P0b);

% Plot the result (make new sig-objects to compare with ground truth)
Ybplot = sig([ex4_yh; ex4_w]', 1, [], [ex4_h; ex4_w; ex4_b*ones(size(ex4_h))]');
figure(2); clf;
xplot(Xhatb, Ybplot, 'conf', 99);

print(2, '-depsc', fullfile('fig', 'ex4b'));
```



(a) Exercise 4a

(b) Exercise 4b

Figure 2: Resulting plots for exercise 4.