EXAMINATION IN TSRT14 SENSOR FUSION

ROOM: ISY's and MAI's computer rooms

TIME: 2022-06-02 at 8:00–12:00

COURSE: TSRT14 Sensor Fusion

PROVKOD: DAT1

DEPARTMENT: ISY

NUMBER OF EXERCISES: 4

RESPONSIBLE TEACHER: Gustaf Hendeby, tel. 013-28 58 15,
`gustaf.hendeby@liu.se`

VISITS: cirka 09:00, 10:00, 11:00

COURSE ADMINISTRATOR: Ninna Stensgård, 013-282225,
ninna.stensgard@liu.se

APPROVED TOOLS: 1. *F. Gustafsson*: "Statistical Sensor Fusion"

PROVIDED MATERIAL:

1. Lecture slides; available from `/courses/TSRT14/`
2. Signal and Systems toolbox manual; available from `/courses/TSRT14/`
3. Current up to date errata for the textbook; available from `/courses/TSRT14/`

MATLAB FILES: The files that are needed for the exam are available at `/courses/TSRT14/`.

SOLUTIONS: Available at the course homepage after the exam.

The exam can be inspected and checked out 2022-06-23 at 12.30–13.00 in Gustaf Hendeby's office, room 2A:503, B-house, entrance 27, A corridor to the right.

PRELIMINARY GRADE LIMITS:	grade 3	15 points
	grade 4	23 points
	grade 5	30 points

NB! Solutions should include code and plots and clear cross references between these. Mark all print-outs with your AID-number, date, course code, and exam code.

Good luck!

STARTING MATLAB (Linux)
Type `matlab &` in a terminal.

PRINTING (Linux):
Printouts of regular files can be sent to a specific printer using the command

```
lp -d printername file.pdf
```

in a terminal. (Exchange `printername` for the actual printer name.) When selecting `File/Print` for a Simulink diagram, select the target printer by adding

```
-Pprintername
```

in the `Device option` box.

ADDING YOUR AID ETC TO PRINTOUTS:
Text can be added in Matlab plots with the commands `title` and `gtext`, and for scope plots in Simulink by right clicking and then change the `Axes properties`. In Simulink diagrams it is possible to double click any empty area and then simply add text by typing it.

FURTHER GUIDELINES:

- Make sure to read all exercises and prioritize before getting started. The level of difficulty is not necessarily increasing.
- Make sure to motivate every step of your solution carefully!
- Comment nontrivial steps in the code; including model choices and tuning.
- Put code for each exercise on a separate printout and keep all related paper (hand written material, code, and plots) together when you hand in your solution.

1. The following questions all require relatively short answers, a few sentences or short calculations should be enough. (Note, an incorrect statement will result in 0 p on that subexercise.)

   (a) In an experiment, similar to the one in lab 1, a ping from a moving object is detected at different times using four microphones. The recorded times are modeled as

   $$y_i = \tfrac{1}{c}\|x - s_i\| + t_0 + e_i, \quad i = 0, \dots, 3$$

   where x and s_i are the position of the object and sensor i, respectively, t_0 the time the ping is broadcast, and $e_i \sim \mathcal{N}(0, R_i)$ measurement noise. It is possible to eliminate t_0 using pairwise differences, which can be interpreted as virtual measurements. How many independent virtual measurement does this method yield? Also provide the measurement equation and noise characteristics for these virtual measurements. (2p)

   (b) Which of the following statements about estimation are correct:

      (i) For linear problems, the *Kalman filter* (KF) can be shown to be unbiased.

      (ii) The KF always reaches the *Cramér-Rao lower bound* (CRLB).

      (iii) The *unscented Kalman filter* (UKF) is always to be preferred to the *extended Kalman filter* (EKF) in the cases it can be applied.

      (iv) The EKF2 requires that the measurement function can be inverted.

      (v) For nonlinear and/or non-Gaussian problems, the *particle filter* (PF) is always the preferred filter.

      (vi) Without the resampling step, a PF always becomes depleted sooner or later.

   (2p)

   (c) You and your friend are designing an EKF to track cars in an intersection. Your first filter is based on a constant velocity motion model and radar measurements. You have tuned the process noise, and taken the measurement noise from the sensor specification. As you try out the filter, the estimates are really smooth, but when a car turns, it takes a long time for the filter to notice this.
   What should you change to get a better result? (2p)

   (d) Consider two estimates $x_A \sim \mathcal{N}(\hat{x}_A, P_A)$ and $x_B \sim \mathcal{N}(\hat{x}_B, P_B)$, of two estimates of the same underlying parameter x. The origins of the estimates are unknown. Given

   $$\hat{x}_A = \begin{pmatrix} 0.8 \\ 2 \end{pmatrix} \qquad P_A = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.33 \end{pmatrix}$$
   $$\hat{x}_B = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad P_B = \begin{pmatrix} 0.33 & 0 \\ 0 & 0.5 \end{pmatrix},$$

   calculate the best possible fused estimate, $\mathcal{N}(\hat{x}, P)$. (2p)

(e) Which of the following statements are true of *simultaneous localization and mapping* (SLAM)?

(i) The best way to solve the SLAM problem is to alternate between solving the localization problem and the mapping problem, independently.

(ii) SLAM provides global coordinates for all observed landmarks.

(iii) The computational complexity of EKF SLAM scales favorably with the number of landmarks in the map.

(iv) The computational complexity of FastSLAM 2.0 scales favorably with the number of landmarks in the map.

(v) The SLAM problem is inherently unobservable, unless additional information is introduced.

(2p)

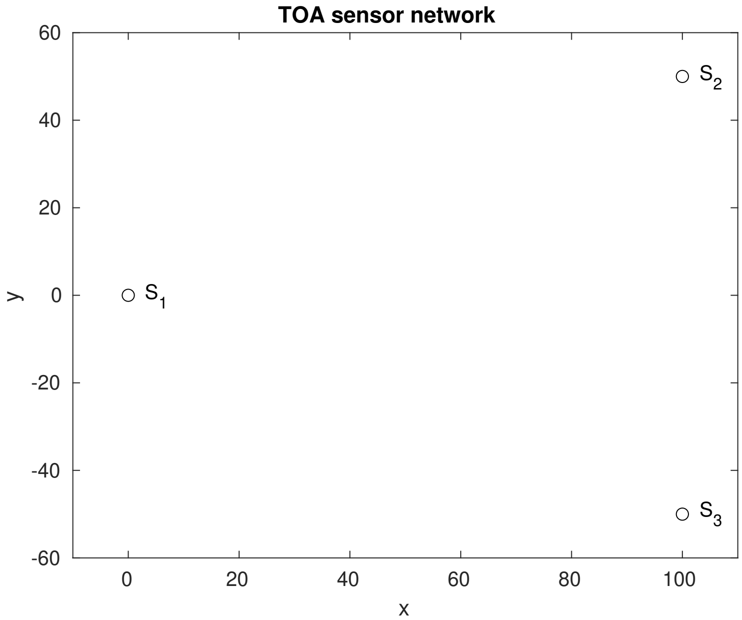

Figure 1: Sensor network considered in Exercise 2.

2. Consider a sensor network consisting of 3 *time of arrival* (TOA) sensors, placed as in Figure 1. The sensors deliver measurements as distance to the object, and each sensor has the measurement noise standard deviation $\sigma_r = 5\,\mathrm{m}$.

   An *all terrain vehicle* (ATV) drives through the surveillance area while emitting pulses at known points in time. The file `data20220602.mat` contains the measurements from the sensors. Each element of `ex2_y1` contains the measurements from S_1, `ex2_y2` and `ex2_y3` similarly for the sensors S_2 and S_3, respectively. The sample times are given in `ex2_t`.

   (a) Estimate the position of the ATV from the measurements at each time independently, *i.e.*, derive snapshot estimates! Plot the resulting position estimates, and connect them over time with a line! Make sure to motivate the choice of models, tuning etc. (3p)

   (b) Construct two different filters to estimate the position of the ATV based on:

      (i) The position estimates obtained in the (a) part!

      (ii) The raw distance measurements as measurements!

      Plot the two new estimated trajectories in the same plot! In both cases, make sure to motivate the choice of models, tuning, etc. (5p)

   (c) Consider what happens if the sensor network does only contain the two sensors S_2 and S_3. Discuss how the results in would change (a) and (b)! Make sure to motivate your answer.

      (You do **not** have to redo (a) and (b) with only S_1 and S_3.) (2p)

3. (a) Consider a detection problem where the measurement under $\mathcal{H}_0$ is random Gaussian noise $y \sim \mathcal{N}(0,1)$, and χ_2^2 under $\mathcal{H}_1$. Given this setup:

   - Derive a uniformly most powerful detector, *i.e.*, a test statistics and decision rule!
   - Give an expression for how to compute the probability of false alarm, P_{FA}!
   - Give an expression for the probability of detection! (5p)

   (b) Consider range-bearing measurements y_k according to

$$y_k = h(x) + e_k = \begin{pmatrix} \|x_k - x_0\| + e_k^r \\ \operatorname{atan2}(x_k^{\mathsf{y}} - x_0^{\mathsf{y}}, x_k^{\mathsf{x}} - x_0^{\mathsf{x}}) + e_k^{\phi} \end{pmatrix},$$

   where atan2 is the quadrant compensated arctangent function (*cf.* `atan2` in Matlab), x_0 is the sensor location, and x_k the position of the measured object at time k, and $^{\mathsf{x}}$ and $^{\mathsf{y}}$ are used to indicate the x and y component of the position, respectively. You can assume that e_k^r and e_k^{ϕ} are mutually independent and $e_k^r \sim \mathcal{N}(0, (\sigma^r)^2)$ and $e_k^{\phi} \sim \mathcal{N}(0, (\sigma^{\phi})^2)$.

   Derive the CRLB of the position estimate $\hat{x}$ that is based on a single y_k! (5p)

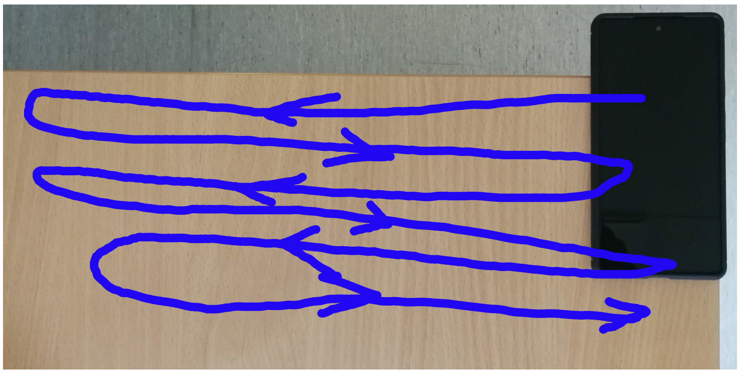

Figure 2: Illustration of phone experiment used in Exercise 4.

4. In this exercise, you will use data from the Sensor Fusion app to estimate the length of a table. An experiment has been conducted where a phone was moved back and forth along the edge of a table three times, with a stop at each edge. The setup is illustrated in Figure 2. The measurements are available in `data20220602.mat` in the variables `ex4_t` (containing the time stamp of the measurements) and `ex4_acc` (where each column contains the measured acceleration in the x-, y-, and z-direction, respectively).

   To simplify, you can consider only the measurement in the direction of motion, *i.e.*, the x-direction. Furthermore, assume that the z-axis is perfectly aligned with gravity at all time.

   To obtain full credits, the exercise should be solved with techniques inspired by the methods discussed in the course, and all steps be well motivated.

   (a) Estimate the position of the phone using dead reckoning, *i.e.*, by double integrating the measured accelerations! Plot the position of the phone, and its velocity! (3p)

   (b) Make use of the fact that the phone is at rest at the extremes, and construct a filter to estimate the position of the phone along the x-axis! Use the norm of the x-accelerations to determine when the phone is at rest, and add virtual measurements for zero speed at the times when phone is indicated to be at rest to your previous solution. Plot the resulting position, velocity, and bias in new plot! (4p)

   (c) You are now given information that the phone is at the initial position at time indices $k < 10$, $1340 < k < 1350$, $2100 < k < 2110$, and $k > 3000$. Use this extra information to further improve the filter estimate! Plot the resulting position, velocity, and bias estimate! (3p)

   Hint: *The Signal and Systems toolbox is not necessarily the best way to solve this problem.*