EXAMINATION IN TSRT14 SENSOR FUSION

ROOM: ISY's computer rooms

TIME: 2022-10-20 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, 013-28 22 25,
`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 2022-11-10 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, any incorrect statement will result in 0 p on that question.)

   (a) Consider a setup similar to the one in lab 2, where you were to estimate the orientation of a smartphone based on measurements from the built in IMU. In the lab, an accelerometer outlier step was introduced, to detect and ignore measurements when the phone experienced external accelerations (*i.e.*, was moved).

   Which assumption made in the lab makes this outlier rejection necessary? Also provide a simple test to detect accelerations of the IMU. (2p)

   (b) Assume y from either $\mathcal{H}_0$ or $\mathcal{H}_1$ according to:

$$y \sim \begin{cases} \mathcal{N}(0, R), & \text{under } \mathcal{H}_0 \\ \mathcal{N}(\mu, R), & \text{under } \mathcal{H}_1. \end{cases}$$

   What is the uniformly most powerful test to decide between the two hypothesis given a measurement of y? (2p)

   (c) An extended Kalman filter has been designed to track cars traveling through an intersection. The filter uses a *constant velocity* (CV) motion model, and the measurement model uses the noise specification provided by the sensor manufacturer. Figure 1 shows the result of tracking a car through the intersection.

   You have been asked to have a look at the tuning and suggest changes if you find it necessary. What do you respond? (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 known, and it can be guaranteed that the estimates are independent. 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)

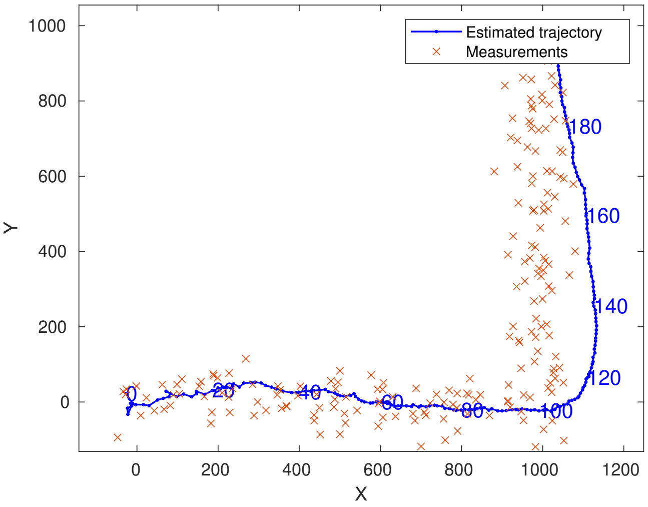

Figure 1: Illustration of measurements and filter result for Exercise 1c.

(e) Which of the following statements are correct?

(i) For linear Gaussian measurements, *weighted least squares* (WLS) is optimal, and reaches the *Cramér-Rao lower bound* (CRLB).

(ii) For linear measurements, the *weighted least squares* (WLS) and the *maximum likelihood* (ML) estimates are identical, but are derived using different principles.

(iii) The *unscented Kalman filter* (UKF) is always to be preferred to the *extended Kalman filter* (EKF).

(iv) *Simultaneous localization and mapping* (SLAM) only provides the relative position with respect to the start position not a global position.

(v) A particle filter gives slightly different results each time it is run on the same data due to its stochastic nature.

(2p)

2. Consider the air traffic control problem of tracking an airplane (in two dimensions) using a radar. The radar is positioned in the origin, measuring range, r, and bearing, ϕ, to the target with the independent standard deviations $\sigma_r = 100$ and $\sigma_\phi = 0.1$, respectively. The file `data20221020.mat` contains a variable `ex2_y` sampled at 2 Hz with measurements of an airplane.

   (a) Create a measurement model in MATLAB for the described radar! (2p)

   (b) Plot the measurements (transformed to Cartesian coordinates) using, *e.g.*,

   ```
   plot(ex2_y(:, 1).*cos(ex2_y(:, 2)),...
        ex2_y(:, 1).*sin(ex2_y(:, 2)), 'x');
   ```

   Based on this, select a suitable motion model to describe the target and create it in MATLAB! (2p)

   (c) Use an unscented Kalman filter to estimate the trajectory of the target! Plot the estimate with 90% confidence intervals. Add the transformed measurements and use them to verify that your solution is reasonable. ***Hint:*** *Use the first measurements to determine how to initialize the filter.* (3p)

   (d) Use a particle filter to estimate the trajectory of the target! Plot the estimate in a new plot with 90% confidence intervals together with the transformed measurements. (3p)

3. The *Kalman filter* (KF) provides an analytic solution to the Bayesian filtering problem given a linear-Gaussian state-space model for the problem. Here, consider a discrete valued *hidden Markov model* (HMM). That is a model where the state x_k assumes discrete values, s^i, $i = 1, \dots, n$. This allows for the pdf of the state to be described by the values p^i_k, $i = 1, \dots, n$, which represents the probability that $x_k = s^i$.

   The dynamic model is described by the transition matrix Π, where the elements $\Pi_{ij} = \Pr(x_k = s^i | x_{k-1} = s^j)$ describe the probability to transition from state s^j to s^i in a time step.

   The measurements are described by $p(y_k | x_k = s^i)$, $i = 1, \dots, n$, where the likelihood can be computed for all relevant y_k.

   (a) Derive the time update step in a filter working on the HMM based on the Bayesian time update equation

   $$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) \, dx_{k-1}$$

   assuming that the state transition matrix Π is known! Hence, derive all $p^i_{k|k-1}$ given all $p^j_{k-1|k-1}$, the probabilities for the discrete state values before and after the time update.

   For 3 p you may assume $n = 2$, *i.e.*, the number of discrete state values is know to be two. (5p)

   (b) Derive the measurement update step in a filter working on the HMM based on the Bayesian measurement equation

   $$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}$$

   assuming that the likelihood $p(y_k | x_k = s^i)$ is known. Hence, derive all $p^i_{k|k}$ given all $p^j_{k|k-1}$, the probabilities for the discrete state values before and after the measurement update.

   For 3 p you may assume $n = 2$. (5p)

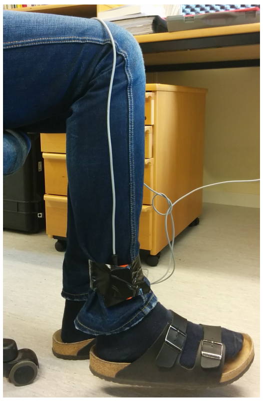

Figure 2: Experimental setup with IMU mounted close to the user's foot.

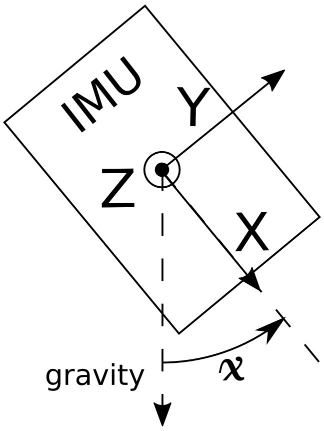

Figure 3: Schematic problem formulation.

4. Capturing the motion of humans can be of interest in many situations, for instance to create animations in movies. One way to accomplish this is to place several *inertial measurement units* (IMUs) on a person's body and use the measurements to estimate his/her motion. In this exercise a simplified version of the problem is considered, where the lower leg is tracked using one IMU placed close to the foot and assuming the subject's knee is not moving. (See Figure 2.)

   The problem is further simplified by assuming the IMU is only rotating directly around its own z-axis, which is aligned with the horizontal plane, as depicted in Figure 3. The task is to estimate the angle denoted x in the figure using measurements from the IMU. The file `data20221020.mat` contains the measurements at 100 Hz (`acc` contains the accelerometer measurements in the x, y, and z direction, respectively, in the columns, and `gyr` contains the gyroscope measurements in a similar way.)

   (a) Suggest a measurement model to estimate x using the accelerometer! Here it can be assumed the accelerometer only measures the gravity. Use this to estimate the orientation and plot it!

   Hint: *(1) The repeated usage of* `estimate` *for each measurement can be a bit slow, start with just a few measurements. (2) This is not a standard* `exsensor` *model.* (4p)

   (b) Suggest a motion model that uses the gyroscope measurements as input! Use this to estimate x assuming the IMU starts with the x-axis pointing down. Plot the new estimate in the same figure as above! (3p)

   (c) Combine the measurement model and the motion model, and apply a filter to estimate x! Plot the result in the same figure as the two previous estimates.

   Hint: *(1) Make sure to specify the frequency as the 4^{th} argument when creating the* `nl` *object to make the model time discrete. (2) To create a* `sig` *object with both measurements and input, use the syntax* `sig(y, fs, u)`*.* (3p)