Started on	Friday, 3 December 2021, 12:00 PM
State	Finished
Completed on	Friday, 3 December 2021, 3:10 PM
Time taken	3 hours 10 mins
Grade	Not yet graded

Question **1**

Marked out of 6.00 Complete

Instructions: Please write your answers to the text fields, you can (but don't need to) use attachments to clarify your answers. Use equations to clarify your answers if necessary. Do not copy-paste equations or other things from say LibreOffice, instead use attachments or properly included figures. Finally submit your answers. Please note that "Finish attempt..." does not yet finish the exam, after pressing that you can still edit. But then finally in the end you need to press "Submit all and finish" -- that can be only done once. If you encounter problems during the exam, email christos.merkatas@aalto.fi and simo.sarkka@aalto.fi, or call +358465442402/Christos or +358407570730/Simo (preferably contact Christos as Simo is traveling then).

Use equations to clarify your answers if necessary.

Describe briefly the following terms and what is their significance in sensor fusion:

a) differential equation (1p)

b) state-space model (1p)

c) Euler method (1p)

d) Euler-Maruyama method (1p)

e) Kalman filter (1p)

f) Extended Kalman filter (1p)

a) differential equation (1p)

Differential equations are used to describe how a system behaves with one or more variables/functions and their derivatives (usually over time in sensor fusion). While the functions represent properties the derivatives represent the rate of change in the system, combining these by describing the relationship between the two gives a differential equation. They are used to model the dynamics of sensor fusion systems.

b) state-space model (1p)

A state-space model represents a system as a set of input and output state variables related by first-order differential equations. In sensor fusion they provide a way to easily represent systems in matrix form so that they can be easily manipulated.

c) Euler method (1p)

The Euler method is a numerical procedure used for solving differential equations with a given initial value. Although derivatives can be solved by hand it is often good to at least check whether the derivatives calculated are correct using a computer. The Euler method is one way of doing this.

d) Euler-Maruyama method (1p)

This is an extension of the Euler method. Where the Euler method could be used to numerically solve ODEs (Ordinary Differential equations) the Euler-Maruyama method is used to numerically solve SDEs(stochastic differential equations). In sensor fusion, this means solving differential equations with unknown inputs.

e) Kalman filter (1p)

A Kalman filter is used to combine predefined information and measurement information to get a more accurate result than just trusting measurement information. The Kalman filter uses two steps: the prediction step and the measurement update step. In the prediction step, the current state of the system is predicted using an ideal dynamic model. Of course in the real world, the system is subjected to random variables and therefore will not be in that exact state given. The measurements are also subjected to noise and therefore won't predict an exact estimate of the system. The measurement update step combines the measurements and the dynamic model in order to get a state of the system with reduced uncertainty.

The Kalman filter is the optimal filter for linear state-space models and an optimal estimator for linear dynamic systems.

f) Extended Kalman filter (1p)

While the Kalman Filter is used for linear models. The extended Kalman filter is used for nonlinear models. It uses linearization to approximate the states of nonlinear models.

Question **2**

Marked out of 6.00 Complete

Consider the scalar differential equation

$$\dot{x}(t) = f\,x(t) + u(t), \quad x(0) = x_0$$

where *f* and x_0 are constants, and *u*(*t*) is a deterministic input.

a) Write down the solution to the differential equation (2p).

b) Form zero-order-hold (ZOH) discretization with a constant time step Δ*t* in the form

$$x_n = f_n\,x_{n-1} + l_n\,u_n,$$

that is, find the expressions for f_n and l_n (4p).

a)

$$\dot{x}(t) = fx(t) + u(t), \quad x(0) = x_0$$

$$\frac{dx}{dt} = fx(t) + u(t)$$

$$x(t) = \frac{f}{2}x(t)^2 + u(t)x(t) + x_0$$

b)

The discretization dynamic model takes the form:

$$x_n = f_n x_{n-1} + l_n u_{n-1}$$

Where f takes the form:

$$\mathbf{f} \triangleq e^{\mathbf{a}(t_n - t_{n-1})}$$

And l takes the form:

$$\mathbf{l} \triangleq \int_{t_{n-1}}^{t_n} e^{\mathbf{a}(t_n - t)} \mathbf{b}_u \, \mathrm{d}t$$

l can be simplified to the form:

$$l \triangleq \left[-\frac{be^{a(t_n - t)}}{a} \right]_{t_n}^{t_n - 1}$$

$$l \triangleq -\frac{be^{a(t_n - t_n)}}{a} + \frac{be^{a(t_n - t_{n-1})}}{a}$$

$$l \triangleq -\frac{b}{a} + \frac{be^{a(t_n - t_{n-1})}}{a}$$

In order to get it in the discretization dynamic model form:

$$x_n = e^{\mathbf{a}(t_n - t_{n-1})}x + u\int_{t_{n-1}}^{t_n} e^{\mathbf{a}(t_n - t)}\mathbf{b}_u \, \mathrm{d}t$$

or

$$x_n = e^{\mathbf{a}(t_n - t_{n-1})}x + u(-\frac{b}{a} + \frac{be^{a(t_n - t_{n-1})}}{a})$$

Question **3**

Marked out of 6.00 Complete

a) Explain the principles of 2 methods to discretize a nonlinear stochastic differential equation of the general form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\,\mathbf{w}(t),$$

here **w**(*t*) is a white noise process with spectral density matrix $\mathbf{\Sigma}_\mathbf{w}$ and **B** is a constant matrix (2p).

b) Convert the following second order equation into the standard form above:

$$\ddot{x}(t) = -\sin(x(t)) + w(t),$$

that is, define the corresponding **x**, **f**, and **B** (2p).

c) Discretize the resulting equation by using either of the 2 methods that you described above (2p).

a)

Approximation of the nonlinear model in linear form, followed by discretization.

In this first method the nonlinear system is first approximated into a linear model using taylor series expansion and then discretised using a linear discretization approach.

Euler–Maruyama

This uses the nonlinear dynamic model:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}_w(\mathbf{x}(t))\mathbf{w}(t)$$

And converts it into its discretized form using the following representation:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \Delta t \mathbf{f}(\mathbf{x}_{n-1}) + \mathbf{q}_n$$

$$\mathbf{q}_n \sim \mathcal{N}(0, \mathbf{Q}_n),\ \mathbf{Q}_n \approx \Delta t \mathbf{B}_w(\mathbf{x}_{n-1}) \Sigma_w \mathbf{B}_w(\mathbf{x}_{n-1})^\mathsf{T}$$

b) In the form of:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{w}(t)$$

Where:

$$\dot{\mathbf{x}}(t) = \ddot{x}(t) = -\sin(x(t)) + w(t)$$

$$\mathbf{x} = \dot{x}(t)$$

$$\mathbf{f} = -\sin(x)$$

$$\mathbf{B} = 1$$

c)

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \Delta t \mathbf{f}(\mathbf{x}_{n-1}) + \sqrt{\Delta t} \mathbf{B}_w(\mathbf{x}_{n-1}) \mathbf{q}_n$$

With

$$\mathbf{q}_n \sim \mathcal{N}(0, \Sigma_w)$$

Question **4**

Marked out of 6.00 Complete

Consider the dynamic model

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w(t)$$

where $w(t)$ is a white noise process with spectral density σ_w^2.

a) Assuming that we sample the process at the time steps t_n, n = 1,2,3,... such that $\Delta t = t_n - t_{n-1}$, find the equivalent discrete-time model

$$\mathbf{x}_n = \mathbf{F}_n \mathbf{x}_{n-1} + \mathbf{q}_n,$$

where

$$\mathrm{E}\{\mathbf{q}_n\} = 0$$

and

$$\mathrm{Cov}\{\mathbf{q}_n\} = \mathbf{Q}_n.$$

That is, find $\mathbf{F}_n$ and $\mathbf{Q}_n$ (3p).

b) Assume that we obtain noisy measurements of x_2 (second component of **x**) at the time steps t_n, n = 1,2,3,.... Write the measurement model in the standard form

$$\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \mathbf{r}_k.$$

by defining a suitable **G** (2p).

c) Which algorithm would you choose to estimate the states? Explain the reason for your choice (1p).

a)

Because A^j = I and 0! = 1 and anything to the power of 0 = 1 and A = [0 0; 1 0]:

$$\mathbf{F}_n = e^{\mathbf{A}\Delta t} = \sum_{j=0}^{\infty} \frac{1}{j!}\mathbf{A}^j(\Delta t)^j = \frac{1}{0!}\mathbf{I}(\Delta t)^0 + \frac{1}{1!}\mathbf{A}\Delta t = I + A\Delta t = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix}$$

Where

$$\mathbf{q}_n \triangleq \int_{t_{n-1}}^{t_n} e^{\mathbf{A}(t_n - t)}\mathbf{B}_w\mathbf{w}(t)\mathrm{d}t$$

$$Cov\{\mathbf{q}_n\} = \int_{t_{n-1}}^{t_n} e^{\mathbf{A}(t_n-\tau)}\mathbf{B}_w\Sigma_w\mathbf{B}_w^{\top}e^{\mathbf{A}^{\top}(t_n-\tau)}\mathrm{d}\tau \triangleq \mathbf{Q}_n$$

b)

G is chosen to get only the state variable x2 of the system.

$$\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \mathbf{r}_k$$

$$G = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{y}_n = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_n + \mathbf{r}_k$$

Assuming xn is in the form:

$$\mathbf{x}_n = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

c)

Kalman filter because it is a linear dynamic model. The Kalman filter is an optimal estimator for linear dynamic systems.

Question **5**

Marked out of 6.00 Complete

Consider a 1D Gaussian random walk model with scalar states x_n and measurements y_n:

$$x_n = x_{n-1} + q_n,$$
$$y_n = x_n + r_n,$$

where x_0 ~ N(0,1), q_n ~ N(0,1), and r_n ~ N(0,1).

Explain how you would use Kalman filter to estimate the states x_n from the measurements y_n. Feel free to use equations to clarify your answer (6p).

The Kalman filter algorithm looks like this:

Algorithm 1 Kalman Filter

1: Initialize $\hat{\mathbf{x}}_{0|0} = \mathbf{m}_0$, $\mathbf{P}_{0|0} = \mathbf{P}_0$
2: **for** $n = 1, 2, \ldots$ **do**
3: Prediction (time update):

$$\hat{\mathbf{x}}_{n|n-1} = \mathbf{F}_n \hat{\mathbf{x}}_{n-1|n-1}$$
$$\mathbf{P}_{n|n-1} = \mathbf{F}_n \mathbf{P}_{n-1|n-1} \mathbf{F}_n^{\mathsf{T}} + \mathbf{Q}_n$$

4: Measurement update:

$$\mathbf{K}_n = \mathbf{P}_{n|n-1} \mathbf{G}_n^{\mathsf{T}} (\mathbf{G}_n \mathbf{P}_{n|n-1} \mathbf{G}_n + \mathbf{R}_n)^{-1}$$
$$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n (\mathbf{y}_n - \mathbf{G}_n \hat{\mathbf{x}}_{n|n-1})$$
$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} - \mathbf{K}_n (\mathbf{G}_n \mathbf{P}_{n|n-1} \mathbf{G}_n + \mathbf{R}_n) \mathbf{K}_n^{\mathsf{T}}$$

5: **end for**

Assuming that the measurements yn is given, a kalman filter can be directly implemented using yn. The model is already in the linear dynamic model form:

$$\mathbf{x}_n = \mathbf{F}_n \mathbf{x}_{n-1} + \mathbf{q}_n$$
$$\mathbf{y}_n = \mathbf{G}_n \mathbf{x}_n + \mathbf{r}_n$$

First, define some variables. In this case Fn and Gn are 1 as represented by the equation above. Next Qn and Rn need to be initalised. Qn is the covarience of qn and Rn is the Covariance of rn. In the case of 1d variables, the covariance and variance are the same. Both The qn and rn are defined to be the same:

$$q_k \sim \mathcal{N}(0,1) \text{ and } r_k \sim \mathcal{N}(0,1)$$

Therefore Qn and Rn are both 1.

Next is to initialise the variables xhat0 (initial mean) and P0 (initial covariance). xhat0 will be given the same initial value as x0, which is a normal random variable with a mean of 0 and a variance of 1. P0 will be given a covariance the same as the variance of x0, which is 1. This is because in the case of 1D variables the variance and the covariance are the same.

Now that all of these are defined it is a simple case of following the algorithm. A for loop for a number of steps needs to be defined, this determines how long you predict states for. In the for loop needs two stages: the prediction step and the measurement update step.

First the prediction step (Predict the current state using the dynamic model):

$$\hat{\mathbf{x}}_{n|n-1} = \mathbf{F}_n \hat{\mathbf{x}}_{n-1|n-1}$$

$$\mathbf{P}_{n|n-1} = \mathbf{F}_n \mathbf{P}_{n-1|n-1} \mathbf{F}_n^{\mathsf{T}} + \mathbf{Q}_n$$

The measurement update step (Estimate the current state using the prediction and the new measurement):

$$\mathbf{K}_n = \mathbf{P}_{n|n-1}\mathbf{G}_n^{\mathsf{T}}(\mathbf{G}_n\mathbf{P}_{n|n-1}\mathbf{G}_n + \mathbf{R}_n)^{-1}$$
$$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{G}_n\hat{\mathbf{x}}_{n|n-1})$$
$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} - \mathbf{K}_n(\mathbf{G}_n\mathbf{P}_{n|n-1}\mathbf{G}_n + \mathbf{R}_n)\mathbf{K}_n^{\mathsf{T}}$$

◄ Exam grades

Exam Results ►