Assignment 5 - Kalman Filtering in state estimation during reaching

Deadline: 13.45h Friday June 15th 2020

Important: Work in groups of two students (more is not allowed!), where each student hands in the same(!) report through Brightspace before the lecture on the due date. Make a short and concise report and do not forget to include your names. The bulleted questions are for your own understanding and should not be included in the report!

Introduction

The Kalman filter is widely used in many engineering applications for state estimation and prediction. Furthermore, some human motor control studies suggest that our Central Nervous System also improves the estimates of our body state by processes that mimic the working mechanism of a Kalman Filter. The Kalman Filter is therefore part of one of the most popular theories, on how we control our movements, i.e. optimal feedback control. In this assignment, you will learn how to implement and use the Kalman filter. Your task is to simulate a simple movement of the hand, and use the Kalman filter to emulate how the human estimates its position and velocity. After getting acquainted with a simpler version of the Kalman filter, you will extend the algorithm to take into account the time delay, which naturally exists within our nervous system. We will see that our brain might be able to compensate for this delay by utilizing the internal forward model and knowledge about the time delay.

You have to implement the algorithms in the MATLAB code by modifying the files included in this assignment. The files you will need to complete are indicated with an *.

as5_kf_run.m	MATLAB script that steps you through all questions.
as5_kf_plot.m	MATLAB script that plots all data.
[*] as5_kf_dynamics.m	MATLAB script generating the dynamic models you will be using.
[*] as5_kf_kalman.m	MATLAB script implement the Kalman filter.
[*] as5_kf_delay_system.m	MATLAB script generating the matrices that make up the delayed models.
[*] as5_kf_movement.m	MATLAB script generating the force input to move the model.
as5_kf_compare_parameter_levels.m	MATLAB script for iteratively varying parameter levels of the Kalman filter and saving it into a results structure.

Note that each function that you need to complete has predetermined inputs and outputs. You cannot change these inputs and outputs! If you do, these scripts fail to run on our computers, making grading your work impossible. What you do inside the function is entirely up to you.
The main script, *as5_kf_run.m*, consists of ”sections”. The start of each new section is indicated with ”%%”. Each section can easily be run separately by pressing the run section button in the *editor* tab. So, first complete the functions within a section and subsequently try to run the section.
You will also be asked to interpret the data the model gives you by showing plots in your report.

Do not upload the m-files with your report, just paste the parts of the code requested in your report. Do write out mathematical derivations instead of just the answers. Do not repeat the question, or a distilled version of the question, in your report. Answer all questions separately with a clear heading where the answer starts (e.g. 'QUESTION I:'), i.e. do not make the whole assignment into one long story.

1 Implementing the Model

The model we will be using to test our Kalman filter, is that of a simple underdamped system. This is expressed via the continuous state-space equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}\mathbf{u}(t) + \mathbf{w}(t)$$
$$\mathbf{y}(t) = \mathbf{C_c}\mathbf{x}(t) + \mathbf{D_c}\mathbf{u}(t) + \mathbf{v}(t)$$

with x(t) and y(t) representing the system states and output respectively containing hand position and velocity. The continuous system matrices are:

$$\mathbf{A_c} = \begin{bmatrix} 0 & 1 \\ 0 & -\beta/m \end{bmatrix}$$
$$\mathbf{B_c} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$$
$$\mathbf{C_c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$\mathbf{D_c} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The model represents a hand in space with mass m (m = 4kg), and damping coefficient β (β = 3.9N s m^{-1}). The input u = [F], with F being the force that acts on the hand.
The system can be extended to include process and measurement noise. These are added to the input, which then becomes u = [F w v]^T. Thereby, w = [$w1$ $w2$]^T is the process noise and v = [$v1$ $v2$]^T is the measurement noise. Both, w and v are normally-distributed white noise sources with a mean of zero and variances Q and R, for process and measurement noise, respectively:

$$\mathbf{w} \sim N(0, \mathbf{Q})$$
$$\mathbf{v} \sim N(0, \mathbf{R})$$

where:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot 1.47e-07$$
$$\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot 3.3e-4.$$

To incorporate the different noise sources, B_C and D_C have to be extended and thus become $B_{c,Noise}$ = [B_c G_w 0_{2x2}] and $D_{C,Noise}$ = [D_c 0_{2x2} G_v], with G_w = G_v = [1 0; 0 1]. (0_{2x2} = zero matrix).

The continuous model can be discretized in time, using the time step Δt, via the forward Euler method:

$$\dot{\mathbf{x}}^{(i)}(t) = \frac{\mathbf{x}^{(i)}(t + \Delta t) - \mathbf{x}^{(i)}(t)}{\Delta t}$$

When discretizing, rewrite the state and control input vector in terms of time step k instead of (t), by assuming that $t = k\Delta t$ and $(t + \Delta t) = (k + 1)\Delta t$, hence $x(t)$ as x_k and $x(t + \Delta t)$ as x_{k+1}.

This results in the discrete version of the system:

$$\mathbf{x}_{k+1} = \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{u}_k$$
$$\mathbf{y}_k = \mathbf{C_d}\mathbf{x}_k + \mathbf{D_d}\mathbf{u}_k$$

Question I

In function *as5_kf_dynamics.m*, use the above continuous state space representation to create a model of the ideal hand.

Before you start try to manually derive the matrices of the ideal hand model A_d, B_d, C_d and D_d by using the forward Euler method. Do not include this derivation in your report but use it to better understand the state space model behind the Kalman filter.

Transform the previously created continuous model into a discrete state space model.

a) Implement the discrete state space into *as5_kf_dynamics.m*.

b) Create a discrete forward model, to exactly mimic the ideal hand.

c) Create a model containing noise inputs by extending the model of the ideal hand to incorporate system noise $w_k = [w_{1,k}\ w_{2,k}]^T$ and measurement noise $v_k = [v_{1,k}\ v_{2,k}]^T$. To add the new inputs to the system, you will need to extend the B and D matrices of the ideal hand model.

$$\mathbf{u}_k = \begin{bmatrix} F_k \\ w_{1,k} \\ w_{2,k} \\ v_{1,k} \\ v_{2,k} \end{bmatrix}$$

Implement this into the file: *as5_kf_dynamics.m*

Defining a Movement

Since the system we are working with is underdamped, we have to specifically design an input to move the hand to a new position and actively stop the movement from progressing further.

Question II

Find the input force to initiate a movement and then stop it with an input in the opposite direction by using the model of the real hand without noise. The movement is to be initiated at t = 0s by a constant force of 1.5N for a duration of 1s. Thereafter the movement is to be brought to a halt (zero velocity) by generating a negative force (of −1.5N) of appropriate duration.

a) Create the appropriate force vector needed to generate this movement. Follow the steps provided in *as5_kf_movement.m*. Save the force vector needed to generate the movement in the function output variable *F*.

b) When the correct force is implemented, plot the input force F and the resulting movement (using *as5_kf_run.m* section Q2). Incorporate the subsequently collected plot in your report.

- Looking at the position and velocity profiles, what is the difference between the real hand, compared to the expected hand position given by the forward model. How will this difference change with increasing process noise w?

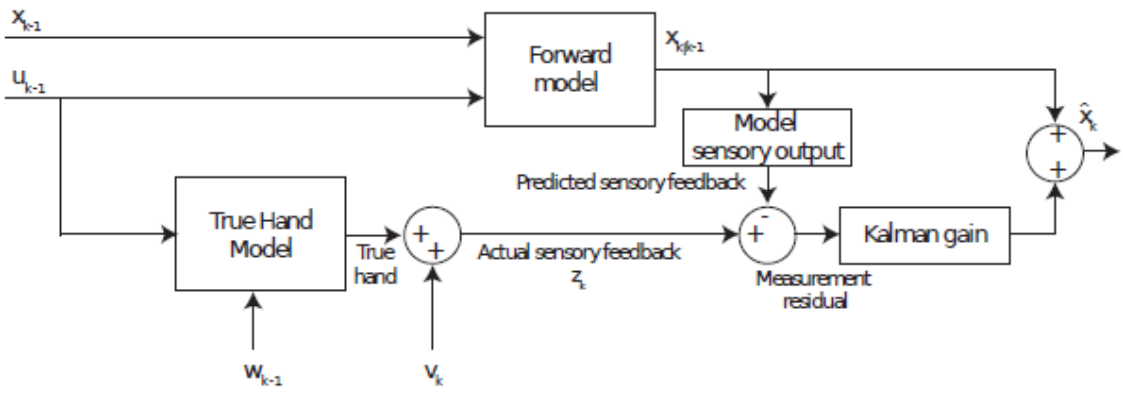

Figure 1: System block diagram of the Kalman filter algorithm, as it is used in this assignment. The a-priori state estimate $x_{k|k-1}$ given by an internal forward model is corrected with a weighted difference between the predicted and actual sensory feedback. This results in the a-posteriori state estimate $\hat{x}_k$. The kalman gain (or 'M' in the equations), specifies the weight with which the correction is made.

The Kalman Filter

During the lecture you have gotten to know the Kalman filter algorithm (Fig. 1). Therefore it is now time to implement your own version of the Kalman filter, which is defined by the following set of equations.

$$\mathbf{x}_{k|k-1} = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

Update Step:

$$\mathbf{M}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R})^{-1}$$
$$\hat{\mathbf{x}}_k = \mathbf{x}_{k|k-1} + \mathbf{M}_k(\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k|k-1})$$
$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{M}_k\mathbf{C}\mathbf{P}_{k|k-1}$$

Thereby, $x_{k|k-1}$ is the a-priori state estimate, $P_{k|k-1}$ the a-priori covariance matrix, P_{k-1} the a-posteriori covariance matrix of the previous iteration whereby $P_{0|0} = 0_{2\times2}$, M_k is the Kalman gain, z_k is the current noisy measurement of the real system, $\hat{x}_k$ is the a-posteriori state estimate and P_k the a-posteriori covariance matrix.

Question III

a) Use the file *as5_kf_kalman.m* to implement the prediction and update equations of the Kalman filter. Include those lines of code in your report that form the prediction and update equations.

b) Thereafter, run section Q3 in the *as5_kf_run* function, which will simulate the movement and run the Kalman filter to estimate the hand position. Implement the resulting plot into your report.

- Inspect the course of the Kalman gains and explain how changes in these gains affect the relative weighting of the different processes in the state estimate.

- At initialization of the Kalman filter, we have set the value of the covariance matrix ($P_{0|0}$) to zero. What does this means for the believe that we have in the correctness of the estimation from our senses.

- Argue what would happen to the Kalman gains throughout the course of the movement, if we had initialized the covariance matrix with its final, steady state value. Why this is the case?

- Once a steady state Kalman gain is reached, will $\mathbf{M}$ and $\mathbf{P}$ change again once more movement input starts coming in?

- Assume a situation where the forward model overestimates the movement of the hand and thus results in a larger velocity and higher steady state position as compared to the real hand. Reason what would happen to the difference between the real hand and the a-posteriori state estimates, for the two scenarios: 1) for increasing values of Q and 2) for increasing values of R.

Investigate the Effect of Different Parameters on the Kalman Filter Estimate

The Kalman filter and its estimates are greatly dependent on its parameters, which will be reflected in the outcome of the estimation. By computing the Kalman filter for a range of the same parameter (e.g. Q or R), the effect of this variable on the outcome can be seen. This will be the topic of this exercise and can be done by editing the function *as5_kf_compare_parameter_levels.m*. The function has already been written for you, however you may vary the 'level' of the parameter change by adjusting the corresponding variable. Do not change how the output is stored in the results structure. Otherwise, the plotting will not work!

Question IV

a) Investigate the effect of different levels of Q by running the Kalman filter for different levels. A typical range could hereby be between 0% and 300% of the initial value of Q, which is already implemented in function *as5_kf_compare_parameter_levels.m*. However you are free to use other ranges as long as they bring your point across. Insert the resulting figures in your report.

- What happens to the Kalman gains for larger values of Q? Explain, why this happens and what it exactly means.

b) Investigate the effect of different levels of R in the same manner. Again, insert the figure in your report.

- What happens to M for increasing values of R? Why does this occur and what does it mean.

- What is the intuitive interpretation of the covariance matrix P and how does it affect, or depend on, Q, R and M?

Investigate the Effect of an Incorrect Forward Model

A second factor which highly influences Kalman filter performance is the accuracy of the forward model. We will simulate internal model offset by adding a gain g to the forward model, such that $B_{offset} = B \cdot g$. (7). This has already been added to the function *as5_kf_compare_parameter_levels.m*.

Question V

a) Investigate the effect of different settings for the forward model gain factor g and provide the figures of position and velocity profiles for different values of g.

- How does a faulty forward model affect the accuracy of the body state estimation?

- Why does varying forward model accuracy level have no influence on the Kalman gains?

- Why is the estimation offset larger during the movement and decreases when the steady state is reached?

Incorporating a Time Delay

If we want to make our analysis more realistic, we have to incorporate the naturally occurring time delay present in our nervous system. One way to do this, is to extend the state-space matrices in an intelligent way, resulting in a delayed output:

$$A_{delayed} = \begin{bmatrix} A_{N\times N} & 0_{N\times N\cdot(d-1)} & 0_{N\times N} \\ I_{N\times N} & 0_{N\times N\cdot(d-1)} & 0_{N\times N} \\ 0_{N\cdot(d-1)\times N} & I_{N\cdot(d-1)\times N\cdot(d-1)} & 0_{N\cdot(d-1)\times N} \end{bmatrix}$$

$$B_{delayed} = \begin{bmatrix} B_{N\times U} \\ 0_{(N\cdot d)\times U} \end{bmatrix}$$

$$C_{delayed} = [0_{O\times(N\cdot d)} \quad C_{O\times N}]$$

$$D_{delayed} = D$$

with *U* being the number of system inputs (number of rows in u), *N* is the number of system states according to A, *O* the number of outputs according to C and *d* the number of delayed states. The total delay time is obtained via $d \cdot \Delta t$.

Question VI

a) Implement the time delay into your state-space models according to the method described above. Do this by adapting the function *as5_ kf_ delay_system.m*. The delay value is stored in the params structure and initially has been set to 0.1 seconds. However, make sure the exact delay is NOT hard-coded and remains dynamic. Add the resulting plot of the delayed movement with *d* is the delay in number of time steps to the report.

Note: The Kalman filter you have programmed should run with the delayed model. If that is not the case, implement correct dynamic matrix resizing in as5_kf_kalman.m.

- Why is the original *A* matrix inserted in the upper left corner of $A_{delayed}$ and why does it make sense to include the various identity matrices in $A_{delayed}$

- Why is the original C matrix inserted in the right corner of $C_{delayed}$ and what is the effect on how the model output is computed?

b) Run the Kalman filter with the delayed model by executing the corresponding section in *as5_kf_run.m* and report the resulting figure.

- Is the Kalman filter able to correct for the delay?

- How does the Kalman filter correct its estimate when the sensory feedback is being delayed? *Hint: Look closely at what is being corrected and what is used for the correction.*

c) We assume that the Kalman filter is a good estimate of how the brain estimates body state. Come up with an experiment to validate whether the brain uses the same update mechanisms to obtain the a-posteriori state estimate, given delayed feedback of body state. Try to limit your answer to a maximum of 200 words.