# TMR4515 - Advanced Model Based Design

Project by Magnus Knædal and Simen Øvereng
Norwegian University of Science and Technology

**Abstract - In this report, a time-varying discrete Kalman Filter was used to estimate the angle of two loads that were controlled by enforcing torque from two motor units, connected to the loads by rotational springs. The filter was able to filter the noise with known covariance accurately and the Kalman Gain converged to steady state value. Next, some of the spring coefficients in the system where assumed unknown. To be able to create a practicable controller providing stability and performance for the uncertainties introduced, a dynamic hypothesis testing where designed. By using different time-invariant Kalman Filters in parallel, each one corresponding to a hypothesis, the uncertainty in the system was predicted. The design was implemented in MATLAB and correctly predicted the different hypotheses.**

## 1 Introduction

As a part of TTK4515, a report was to be made on the work in the module "Advanced Model-based Design and Testing of Marine Control System". The goal was to first implement a time-varying discrete Kalman Filter, setting $k_2$ and $k_5$ to be in the middle of the uncertainty region. Then, the parametric uncertainty of $k_2$ and $k_5$ was introduced, and three different static discrete Kalman Filters were developed for three different subsets of the parametric uncertainty. Dynamic Hypothesis Testing (DHT) was used to determine which of the estimators to use based on comparing the probabilities of the different hypothesis to be true.

## 2 System Description

The system to be controlled in this project was a system consisting of two motors with two loads, all connected through four springs, simulating flexible transmission. The controller was set to determine the motor torque in order to control the angles of the loads. The goal of the controller was to regulate the load angles to zero. An illustration of the system is given in Figure 1. The continuous-time state-space model of the plant is given by Equation (1), where $A$ is the plant dynamics, $B$ is the input dynamics, $C$ is the measurement matrix, and $G$ is the modeled process noise dynamics, capturing uncertainties in the dynamic modeling. All continuous-time state-space matrices can be found in Appendix A. $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ represents the time-varying model and measurement noise, respectively. The augmented state vector is chosen as $\boldsymbol{x} = [\theta_{L1}\ \theta_{M1}\ \theta_{M2}\ \theta_{L2}\ \omega_{L1}\ \omega_{M1}\ \omega_{M2}\ \omega_{L2}\ d_1\ d_2]^\top$, where $d_i$ represents the disturbance torques.

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) + G\boldsymbol{w}(t)\\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) + \boldsymbol{v}(t)\end{aligned} \tag{1}$$

The moments of inertia is set to $J_{M1} = J_{M2} = J_{L1} = J_{L2} = 1[\text{Kgm}^2]$, spring coefficients $k_1 = 0.15[\text{N/rad}], k_3 = k_4 = 0.1[\text{N/rad}]$, while the damping terms is $b_1 = b_2 = b_3 = b_4 = b_5 = 0.1[\text{Ns/rad}]$. $k_2$ and $k_5$ are treated as unknown parameters assumed to have values in the intervals $k_2 \in [0.75, 2.5]$ and $k_5 \in [0.9, 2.5]$.

The disturbance torques are modeled as independent stationary first-order stochastic processes generated by driving a low-pass filter with continuous-time white noise with zero-mean and unit intensity, as shown in Equation (2).

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \frac{.2}{s+.2} & 0 \\ 0 & \frac{.2}{s+.2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{2}$$

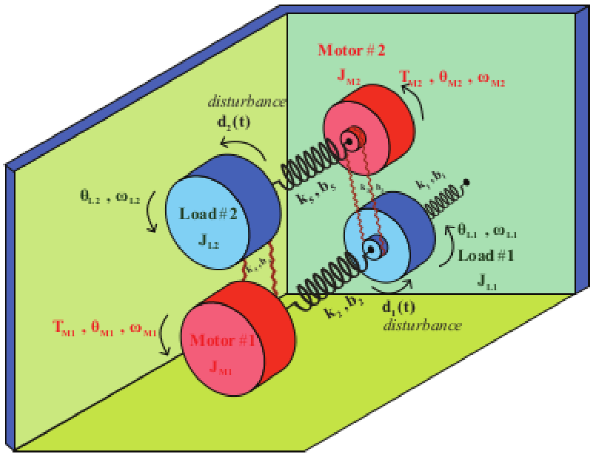

**Figure 1:** Physical System

The measurements of the system were the angles of the loads, $\theta_{L1}$ and $\theta_{L2}$ (in [rad]). The $C$ matrix was modeled as the corresponding measurements, as shown in Appendix A. The measurements was corrupted by an independent zero-mean white noise with intensity of $10^{-6}$, represented by $\boldsymbol{v}(t)$.

# 3 Discrete Kalman Filter

## 3.1 Time varying optimal estimator

Initially, no parametric uncertainty was to be assumed. The values for the spring coefficients $k_2$ and $k_5$ was set to the mean value of their uncertainty intervals. The implementation of the Kalman Filter was done by following the guidelines found in [1], chapter 11.3.

The continuous state-space model in Equation (1) was discretized using zero-order hold approximation, with sample time $h = 1\ [ms] = 0.001\ [s]$. The approximations yielded the discretized state-space model is shown in Equation (3):

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{\Phi x}(k) + \boldsymbol{\Delta u}(k) + \boldsymbol{\Gamma w}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{H x}(k) + \boldsymbol{v}(k), \end{aligned} \tag{3}$$

where the parameters is as shown in Equation (4) to Equation (7):

$$\boldsymbol{\Phi} = e^{\boldsymbol{A}h} \approx \boldsymbol{I} + (\boldsymbol{A}h) \tag{4}$$

$$\boldsymbol{\Delta} = \boldsymbol{A}^{-1}(\boldsymbol{\Phi} - \boldsymbol{I})\boldsymbol{B} \tag{5}$$

$$\boldsymbol{\Gamma} = \boldsymbol{A}^{-1}(\boldsymbol{\Phi} - \boldsymbol{I})\boldsymbol{E} \tag{6}$$

$$\boldsymbol{H} = \boldsymbol{C}. \tag{7}$$

In MATLAB, the discretization was easily found by using `[PHI,DELTA] = c2d(A,B,h)` and `[PHI,GAMMA] = c2d(A,E,h)`. The Kalman Filter was implemented in SIMULINK as a MATLAB function block, where it computed both the estimated state prediction step, and the update/correction of the state estimate. The formulas and the iterative structure of the filter can be found in Appendix B. The implementation in SIMULINK can be seen in Figure 2.

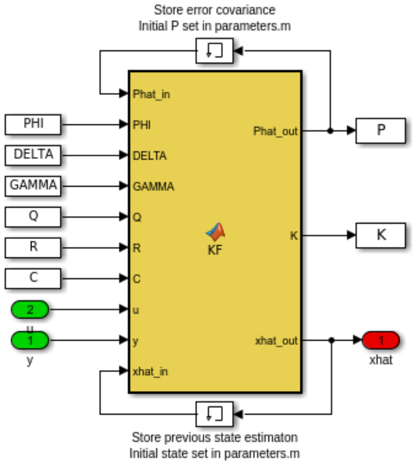

**Figure 2:** Implementation of Kalman filter in Simulink.

The implementation of the Kalman Filter was tested initially by running it in an open-loop to control how the estimates behaved in relation to the real shaft angles and the measured shaft angles with noise (using the real measurements as feedback to the controller). The measurement noise $R$ was set to the intensity of the zero-mean white noise represented by $\boldsymbol{v}(t)$. As for the process noise $Q$, it was set to the zero-mean white noise with the intensity represented by $\boldsymbol{w}(t)$. The continuous-time noise variances are shown in Equation (8). In the discrete filter, R was discretized by dividing on $h$, while Q was discretized by van Loan's Procedure[1]. The resulting estimations are shown in Figures 3 and 4.

$$R = \mathrm{diag}\{10^{-6}, 10^{-6}\},\ Q = \mathrm{diag}\{1, 1\}. \tag{8}$$

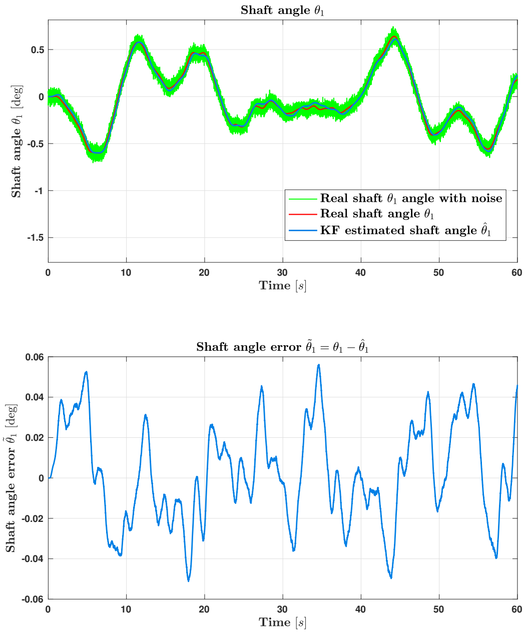

**Figure 3:** Shaft angle $\theta_1$

[1] `https://wolfweb.unr.edu/~fadali/EE782/NumEvalQphi.pdf`

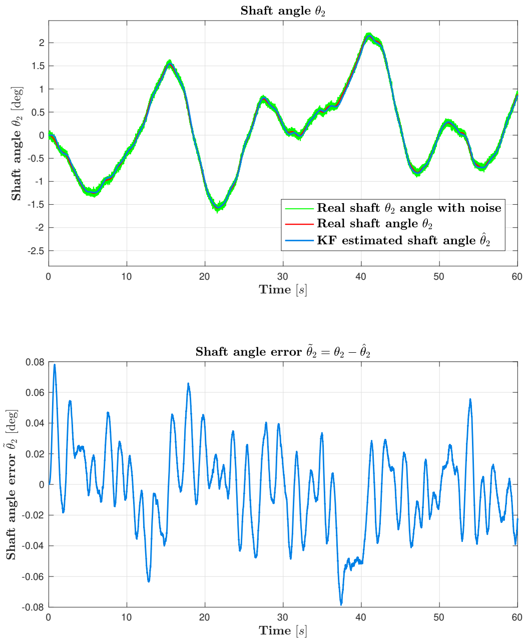

**Figure 4:** Shaft angle $\theta_2$

### 3.2 Steady-state optimal estimator

Since our model is linear and time invariant (i.e. the system matrices are not varying with time), the covariance of estimation error of the corrected predicted estimate $\bar{P}$ and $\hat{P}$, will converge towards steady-state values (see Appendix B). Consequently, the Kalman Filter gain $K$ will converge towards a steady-state Kalman Filter gain value, which can be calculated offline. To confirm this, a time varying Kalman filter was first implemented. The Kalman gains for $\theta_1$ and $\theta_2$ are plotted in Figure 5. It is clear from this that the Kalman gains converges to steady-state values. From hereinafter the steady-state Kalman filter will be used.

A last verification of the implementation was to control the size of the state estimation covariance matrix, P, where it was given that values of magnitude $10^{-6}$ was to be expected when reaching steady-state. The largest steady-state value of the covariance matrix at the last time step of the simulation was $2.837 \cdot 10^{-6}$. By following the observations from the test steps, the Kalman Filter was deemed successfully implemented.

## 4 Hypothesis Testing

In the second part of the project, a Dynamic Hypothesis Test (DHT) was supposed to be carried out. The purpose of such a test is to construct a statistical model, being able to evaluate different hypotheses against each other to determine which of them that are the most likely one to be correct. For a detailed, comprehensive deduction and explanation it is referred to [2].

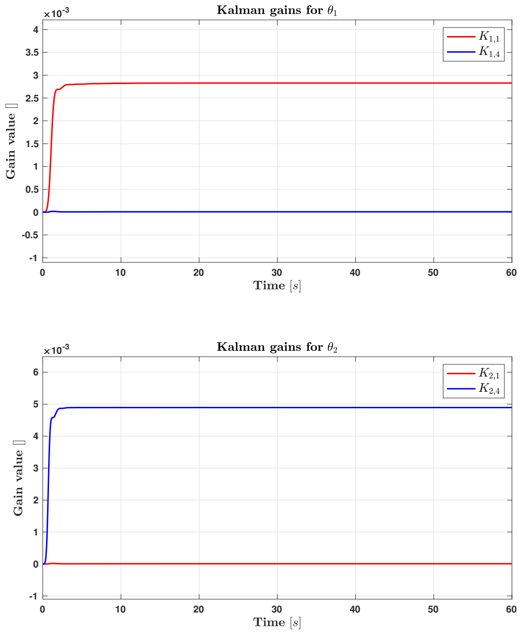

**Figure 5:** Kalman gains over time for the measured states, using initial spring stiffnesses.

In this specific case, an hypothesis $\mathcal{H}_i$ consisted of a set of the unknown spring stiffness coefficients $k_2$ and $k_5$. As previous mentioned, the parametric uncertainty was set to $k_2 \in [0.75, 2.5]$ and $k_5 \in [0.9, 2.5]$. Three hypothesis were to be assumed, so that the parametric uncertainty was divided into three subsets. The subsets are visualized in Figure 6. For simplicity, each subset was represented with their mean value, meaning that the hypothesis consisted of two decimal numbers representing $k_{2,i}$ and $k_{5,i}$. The values for each hypothesis is given in Table 1, found in Appendix C.

Using the Kalman filter developed in Section 3 as a basis, three steady-state discrete Kalman filters were created, one for each hypothesis $\mathcal{H}_i$. The idea is that each Kalman filter will provide the conditional probability of each corresponding hypothesis being true. The probability is based on the updated estimation error $\tilde{y}_{\mathcal{H}_i}(t+1) = y(t) - C\hat{x}_{\mathcal{H}_i}(t+1)$. These conditional probabilities are then used to evaluate which hypothesis is more probable to comply with the collected measurements. Each respective discrete steady-state Kalman filter will

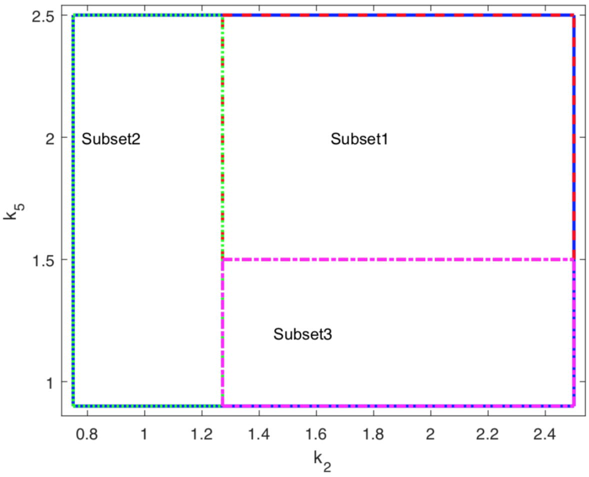

**Figure 6:** The parametric uncertainty set is divided to three subsets

calculate the steps according to Equation (9).

$$\begin{aligned} \hat{x}_{\mathcal{H}_i}(k+1) &= \mathbf{\Phi}_{\mathcal{H}_i}\hat{x}_{\mathcal{H}_i}(k) \\ &+\mathbf{\Delta}_{\mathcal{H}_i}u(k) + K_{\mathcal{H}_i}\tilde{y}_{\mathcal{H}_i}(k), \end{aligned} \tag{9a}$$

$$\tilde{y}_{\mathcal{H}_i}(k) = y(k) - C\hat{x}_{\mathcal{H}_i}(k), \tag{9b}$$

$$K_{\mathcal{H}_i} = P_{\mathcal{H}_i}C^T[CP_{\mathcal{H}_i}C^T + R_d]^{-1} \tag{9c}$$

The Kalman gain $K_{\mathcal{H}_i}$ and the covariance of the predicted estimate $P_{\mathcal{H}_i}$ was found for all three hypotheses off-line, as explained in Section 3.2. Note that the discretized version of the measurement variance matrix R was used here.

## 4.1 Algorithm

In what follows, we establish the algorithm to calculate, in real time, the conditional probability of each hypothesis based on the observation vector $y$. The method goes as follows: at each sampling time, the sampled measurement signal $y[k]$ is used to generate the conditional probability of each hypothesis being true. These conditional probabilities are then used to evaluate which hypothesis is more probable to comply with the collected measurements. For simplicity, all hypothesis were evaluated to be equally probable at the beginning of simulation, so that $p(\mathcal{H}_i|t=0) = 1/N \; \forall \; i \in [1,2,3]$.

For each time instance, the conditional probabilities were calculated according to:

$$h_{i,t+1} = \frac{\frac{e^{-\frac{1}{2}\tilde{y}^T_{\mathcal{H}_i}(t+1)S^{-1}_{\mathcal{H}_i}\tilde{y}_{\mathcal{H}_i}(t+1)}}{\sqrt{(2\pi)^2\left|S_{\mathcal{H}_i}\right|}}}{\sum_{k=1}^{N} h_{k,t}\frac{e^{-\frac{1}{2}\tilde{y}^T_{\mathcal{H}_k}(t+1)S^{-1}_{\mathcal{H}_k}\tilde{y}_{\mathcal{H}_k}(t+1)}}{\sqrt{(2\pi)^2\left|S_{\mathcal{H}_k}\right|}}} h_{i,t} \tag{10}$$

where $h_{i,t+1}$ represents the conditional probability of hypothesis $\mathcal{H}_i$ being true at time $t+1$. $h_{i,t+1}$ depends on the previous probability $h_{i,t}$ and the current estimated error $\tilde{y}(t+1)$, given by the Kalman filter estimation.

To summarize, the order of the calculations goes as follows:

**Off-line:** Initialize probabilities to be uniformly distributed and the initial state estimations to be zero. Load the steady-state Kalman gains and covariance matrix from the previous off-line calculations. Calculate $S_{\mathcal{H}_i} = C^TPC + R$ (in discrete time) as the covariance of the output estimation vector off-line. Find the corresponding determinants $|S_{\mathcal{H}_i}|$ and inverses $S^{-1}_{\mathcal{H}_i}$.

1. Measure the states $y(t+1)$ of the system.
2. Calculate the new state estimate, $\hat{x}(t+1)$, by using the previous estimation $\hat{x}(t)$, the input $u(t)$, and the previous estimation error $\tilde{y}(t)$.
3. Calculate the new estimation error, $\tilde{y}(t+1) = y(t+1) - C\hat{x}(t+1)$.
4. Calculate the conditional probabilities for each of the hypothesis to be true, following Equation (10).
5. Choose the hypothesis with the largest probability as the most likely one.
6. Repeat throughout the simulation process.

## 4.2 DHT for time varying system

One important implementation detail worth mentioning was that since MATLAB was used, only decimal numbers to a certain precision is possible to store in memory. If one hypothesis is the most likely for a long time, the probabilities of the other hypotheses almost reduces to zero. In MATLAB however, when some of the probabilities gets as low as $10^{-16}$, they becomes identical to zero. This means that, following Equation (10), all the future probabilities will also be identical to zero, even if the hypothesis' parameters becomes the correct ones. This is the result of the implementation not taking care of time varying systems.

From a practical point of view, a lower bound on the probabilities should be implemented. This assures that they do not get so small that their corresponding hypothesis will become “frozen”. A boundary was set by checking if one of the probabilities was larger than `boundary = 0.999999`. If so, it would be bounded, and the other probabilities were set to $(1 - boundary)/(N - 1)$. This gave very satisfying results for time varying systems as well.

# 5 DHT simulations

## 5.1 Test 1

A test was carried out where parameters $k_2$ and $k_5$, of the real system were set to the same values as for the hypotheses', found in Table 1 in Appendix C. The real system was initialized with the values of $\mathcal{H}_3$. After 1/3 of the simulation time, a step change in the parameters where made to those of $\mathcal{H}_2$. Lastly, after 2/3 of the simulation time, they were set equal to the parameters of $\mathcal{H}_1$. Initially, all probabilities where set to 1/3, which corresponds to the uniform initialization, as previously mentioned.

### 5.1.1 Results and discussion

A simulation of 300 seconds was run. The results are of the test is visualized in Figures 7 and 8. In Figure 7, each hypothesis' stifness parameters are plotted as constant, dashed lines, and the real spring stiffness are shown as varying, solid lines. In Figure 8, the associated probabilities, ranging from 0 to 1, for each hypothesis is depicted.

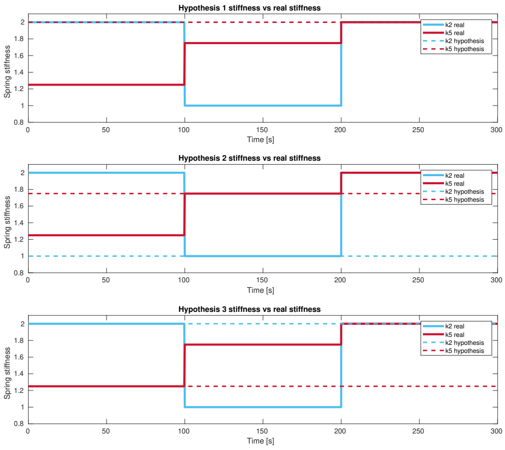

**Figure 7:** Spring stiffness paramteres for simulation 1.

From Figure 8, it can be observed that the most probable hypothesis was found for all three spring stiffness parameters. As the parameters was set to be equal to $\mathcal{H}_3$'s value initially, it also becomes the most probable hypothesis straight away. After the parameters are changed equal to $\mathcal{H}_2$, it takes some time for the relative probabilities to change. The results shows that after approximately 10 seconds following the step change, the most probable hypothesis becomes $\mathcal{H}_2$. Again, at 200 seconds, the spring parameters were switched and set equal to $\mathcal{H}_1$'s values. In addition to a larger delay, it appeared like the probability of $\mathcal{H}_1$ had become so low that it took some time for it to be considered as the most probable one. This is seen as more “zigzag”-shaped probability calculations happening at the two upper plots from approximately 220 to 240 seconds. But again, the correct hypothesis was found to be the most probable one.

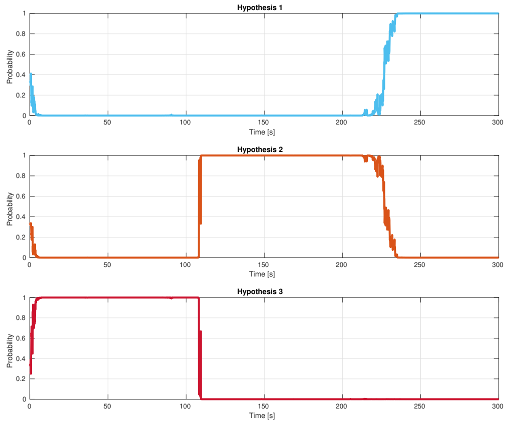

**Figure 8:** DHT probabilities for simulation 1.

## 5.2 Test 2

Next, a test was run using sinusoidal variations of the spring stiffnesses instead of using step changes. The stiffnesses varied according to Equations (11) and (12).

$$k_2(t) = 0.6sin(\frac{2\pi}{150}t - \frac{\pi}{2}) + 1.4 \tag{11}$$

$$k_5(t) = 0.8sin(\frac{2\pi}{150}t - \frac{\pi}{2}) + 1.5 \tag{12}$$

### 5.2.1 Results and discussion

A simulation of 300 seconds was run. The results of the test is visualized in Figure 9 and Figure 10. Figure 9 shows the spring stiffness variations, following the same line formatting as in simulation 1. The sinusoidal variation of both spring stiffnesses becomes close to the ones of $\mathcal{H}_1$ twice, first in the region of 50-100 seconds, and then in the regions of 200-250 seconds. In the other time regions, the parameters are closer to the other two hypotheses' parameters.

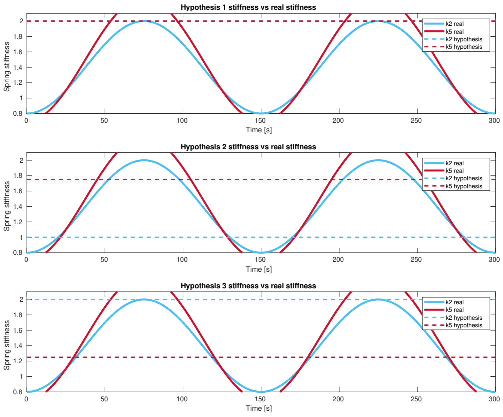

**Figure 9:** Spring stiffness parameters for simulation 2.

From the resulting hypotheses' probabilities in Figure 10, it was observed that $\mathcal{H}_1$ was indeed considered as the most probable approximately in the regions where the parameters were close to being equal to the sinusoidals as discussed above, while $\mathcal{H}_2$ was mostly calculated as the most probable in the regions outside of 50-100 and 200-250 seconds. However, at 150 seconds, which represent the deepest valley of the sine wave, there is some indications on that $\mathcal{H}_3$ could be the most probable. Visually, it was hard to determine the one that should been the most probable one. Mathematically, the values of the time varying spring stiffness parameters in the sinusoidal wave through was $k_{2,min} = 0.8$ and $k_{5,min} = 0.7$. This means that the parameters of $\mathcal{H}_2$ had an error of magnitude $\sqrt{(1.0-0.8)^2+(1.75-0.7)^2} = 1.069$, while $\mathcal{H}_3$ had an error of magnitude $\sqrt{(2.0-0.8)^2+(1.25-0.7)^2} = 1.32$, which indeed suggests that $\mathcal{H}_2$ should be the hypothesis with parameters closer to the real ones. It is clear from the probability plots that the estimated probability for $\mathcal{H}_2$ almost became higher than the probability of $\mathcal{H}_3$ in the wave through, but the probability estimation is not based on absolute error; it rather depends on the state estimation error, which is harder to evaluate in a simple manner.

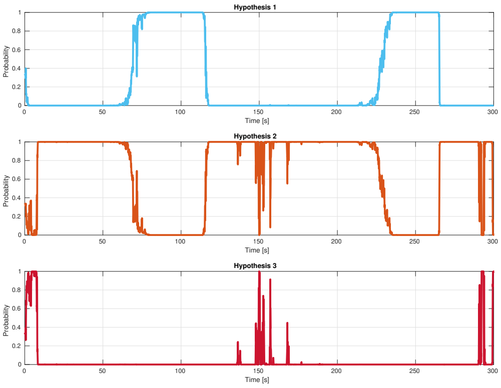

**Figure 10:** DHT probabilities for simulation 2.

# 6 Conclusion

In this project, first a time-varying Kalman Filter where implemented for the benchmark example treated. The implementation was deemed correct and gave satisfying results, especially when Van Loan's method was used for discretizing the Q-matrix. Using the equivalent time-invariant Kalman Filters, a time-varying dynamic hypothesis testing was designed to predict the spring coefficient uncertainty introduced in the system. The design was implemented and tested in MATLAB. Two simulations where done to verify performance, showing satisfying results.

# References

[1] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.

[2] V. Hassani, A. M. Pascoal, and A. J. Sørensen, "Detection of mooring line failures using dynamic hypothesis testing," *Ocean Engineering*, vol. 159, 2018.

All implementations can be found at `https://github.com/magnuok/advanced_model_based_control`

# Appendices

## A Continous-Time State-space Matrices

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{k_1+k_2+k_3}{J_{L1}} & \frac{k_2}{J_{L1}} & \frac{k_3}{J_{L1}} & 0 & -\frac{b_1+b_2+b_3}{J_{L1}} & \frac{b_2}{J_{L1}} & \frac{b_3}{J_{L1}} & 0 & \frac{1}{J_{L1}} & 0 \\ \frac{k_2}{J_{M1}} & -\frac{k_2+k_4}{J_{M1}} & 0 & \frac{k_4}{J_{M1}} & \frac{b_2}{J_{M1}} & -\frac{b_2+b_4}{J_{M1}} & 0 & \frac{b_4}{J_{M1}} & 0 & 0 \\ \frac{k_3}{J_{M2}} & 0 & -\frac{k_3+k_5}{J_{M2}} & \frac{k_5}{J_{M2}} & \frac{b_3}{J_{M2}} & 0 & -\frac{b_3+b_5}{J_{M2}} & \frac{b_5}{J_{M2}} & 0 & 0 \\ 0 & \frac{k_4}{J_{L2}} & \frac{k_5}{J_{L2}} & -\frac{k_4+k_5}{J_{L2}} & 0 & \frac{b_4}{J_{L2}} & \frac{b_5}{J_{L2}} & -\frac{b_4+b_5}{J_{L2}} & 0 & \frac{1}{J_{L2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/J_{M1} & 0 \\ 0 & 1/J_{M1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},\ G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ .2 & 0 \\ 0 & .2 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## B Kalman Filter Steps

**Initialize:**

$$\begin{aligned} &Q(k) = Q^\top(k) > 0, R(k) = R^\top(k) > 0 \\ &\bar{x}(0) = x_0 \\ &\bar{P}(0) = E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^\top\right] = P_0 \end{aligned}$$

**Predict:**

$$\begin{aligned} \bar{x}(k+1) &= \Phi(k)\hat{x}(k) + \Delta(k)u(k) \\ \bar{P}(k+1) &= \Phi(k)\hat{P}(k)\Phi^\top(k) + \Gamma(k)Q(k)\Gamma^\top(k) \end{aligned}$$

**Update / Correct:**

$$\begin{aligned} K(k) &= \bar{P}(k)H^\top(k)\left[H(k)\bar{P}(k)H^\top(k) + R(k)\right]^{-1} \\ \hat{x}(k) &= \bar{x}(k) + K(k)[y(k) - H(k)\bar{x}(k)] \\ \hat{P}(k) &= [I - K(k)H(k)]\bar{P}(k)[I - K(k)H(k)]^\top \\ &\quad + K(k)R(k)K^\top(k), \quad \hat{P}(k) = \hat{P}(k)^\top > 0 \end{aligned}$$

**Order of steps[2]:**

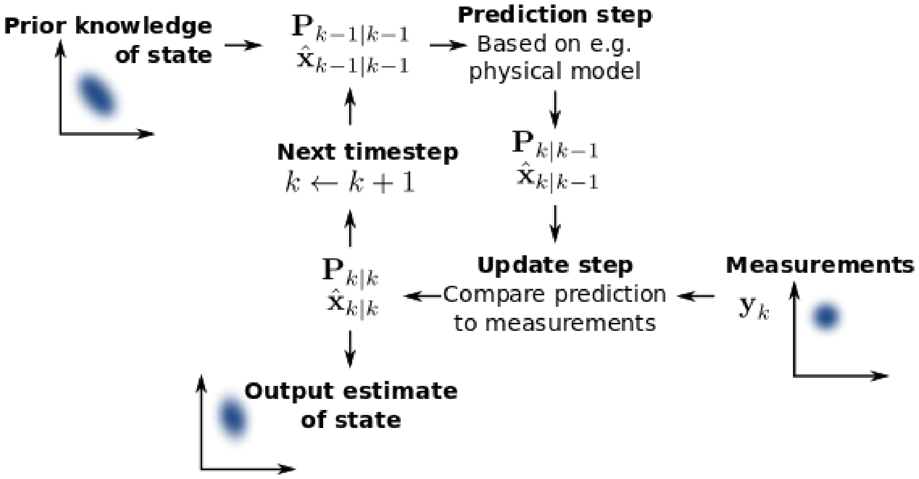

**Figure 11:** Kalman filter iterative steps.

# C Hypothesis parameters

**Table 1:** Spring values for the different hypotheses

| Hypothesis | $\mathcal{H}_1$ | $\mathcal{H}_2$ | $\mathcal{H}_3$ |
|---|---|---|---|
| $k_2$ | 2.0 | 1.0 | 2.0 |
| $k_5$ | 2.0 | 1.75 | 1.25 |

[2] `https://en.wikipedia.org/wiki/Kalman_filter`