Lab 2: INS and Kalman Filter

TTK5: Kalman Filtering and Navigation

By:
Andreas Nordby Vibeto
andvibeto@gmail.com
(andreanv@stud.ntnu.no)

November, 2016

Task 1

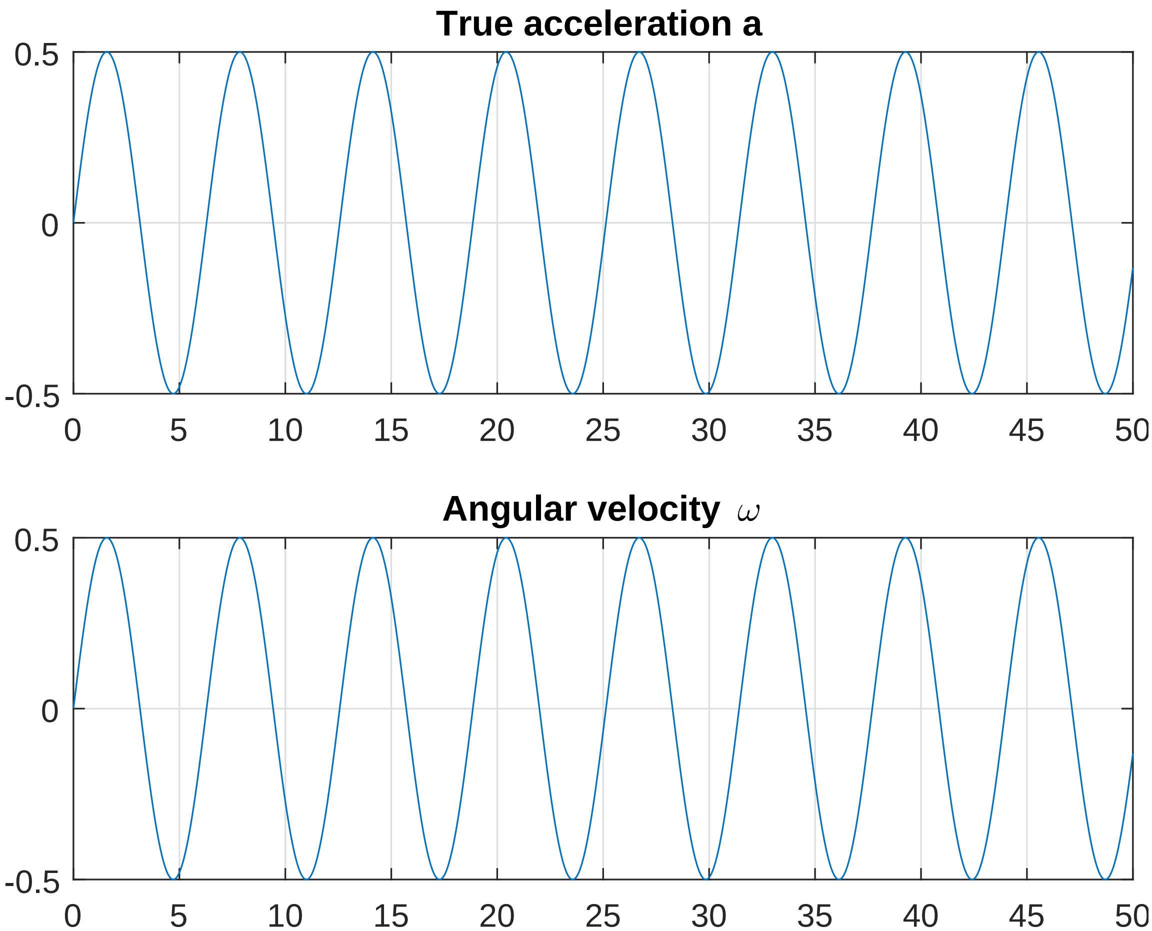

Figure 1: True acceleration and angular velocity for task 1.

Task 2

In order to discretize the system, it must first be written as a state space model. The system

$$\dot{x} = v \tag{1a}$$

$$\dot{v} = a \tag{1b}$$

$$\dot{\theta} = \omega \tag{1c}$$

can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{2a}$$

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ \omega \end{bmatrix}. \tag{2b}$$

By using forward Euler to discretize the system, it can be written on the form

$$\boldsymbol{x}(t_{k+1}) = (\boldsymbol{I} + h\boldsymbol{A}(t_k))\boldsymbol{x}(t_k) + h\boldsymbol{B}(t_k)\boldsymbol{u}(t_k) \tag{3}$$

where h is the step size. The discretized system then becomes

$$\boldsymbol{x}(t_{k+1}) = \begin{bmatrix} 1 & h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(t_k) + \begin{bmatrix} 0 & 0 \\ h & 0 \\ 0 & h \end{bmatrix} \boldsymbol{u}(t_k). \tag{4}$$

Figure 2 shows plots of the states in the discretized system.

Task 3

When white noise is expressed in discrete time it is referred to as a white sequence [1], where the sequence consists of random variables that are uncorrelated [2]. The autocorrelation function for discrete white noise is:

$$R_d(k) = A\delta(k), \quad \delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} . \tag{5}$$

When using Matlab white Gaussian noise can be generated by using `wgn()`, which will generate a sequence of uncorrelated random variables, which can be regarded as a white sequence.

The biases b_1 and b_2 can be discretized with forward Euler using equation 3 from task 2. The bias can be written in state space form as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -\frac{1}{T_1} & 0 \\ 0 & -\frac{1}{T_2} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{w} \tag{6}$$

where

$$\boldsymbol{x} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

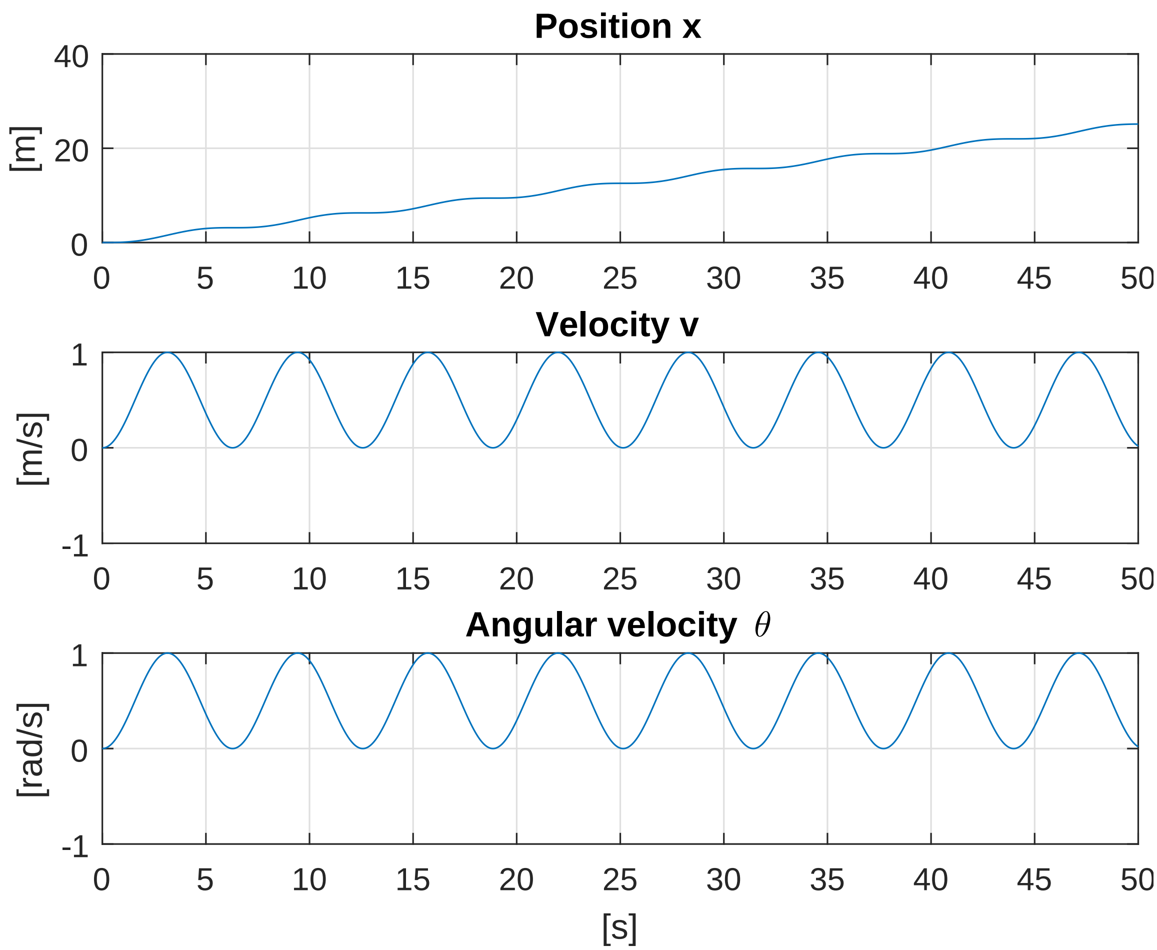

Figure 2: States of the discretized system.

The resulting discretized system is:

$$\boldsymbol{x}(t_{k+1}) = \begin{bmatrix} 1 - \frac{h}{T_1} & 0 \\ 0 & 1 - \frac{h}{T_2} \end{bmatrix} \boldsymbol{x}(t_k) + \begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix} \boldsymbol{w}(t_k). \tag{7}$$

Figure 3 shows the resulting discretized bias, while figures 4 and 5 shows the measured states compared to the true states. Note that for the plots of ω and a the noise is very dominant, but the true states is showes as a sine curve behind all the noise.

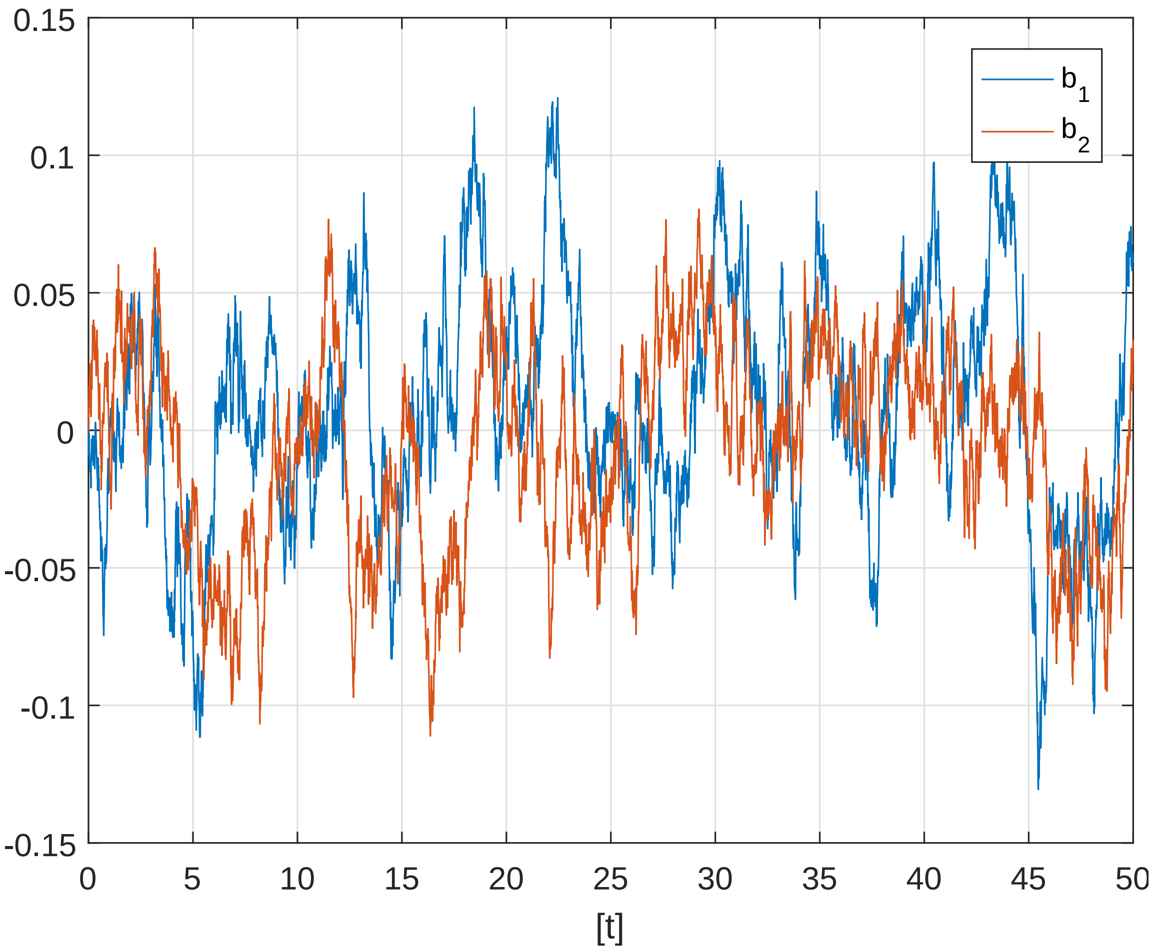

Figure 3: Bias modelled as a Gauss-Markov process for task 3.

Task 4

The Kalman filter consists of several equations, which can be found in table 4.1 in [1].

The states for the Kalman filter are

$$\boldsymbol{x} = \begin{bmatrix} x \\ v \\ b_1 \\ \theta \\ b_2 \end{bmatrix}, \quad u = \begin{bmatrix} a \\ \omega \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{8}$$

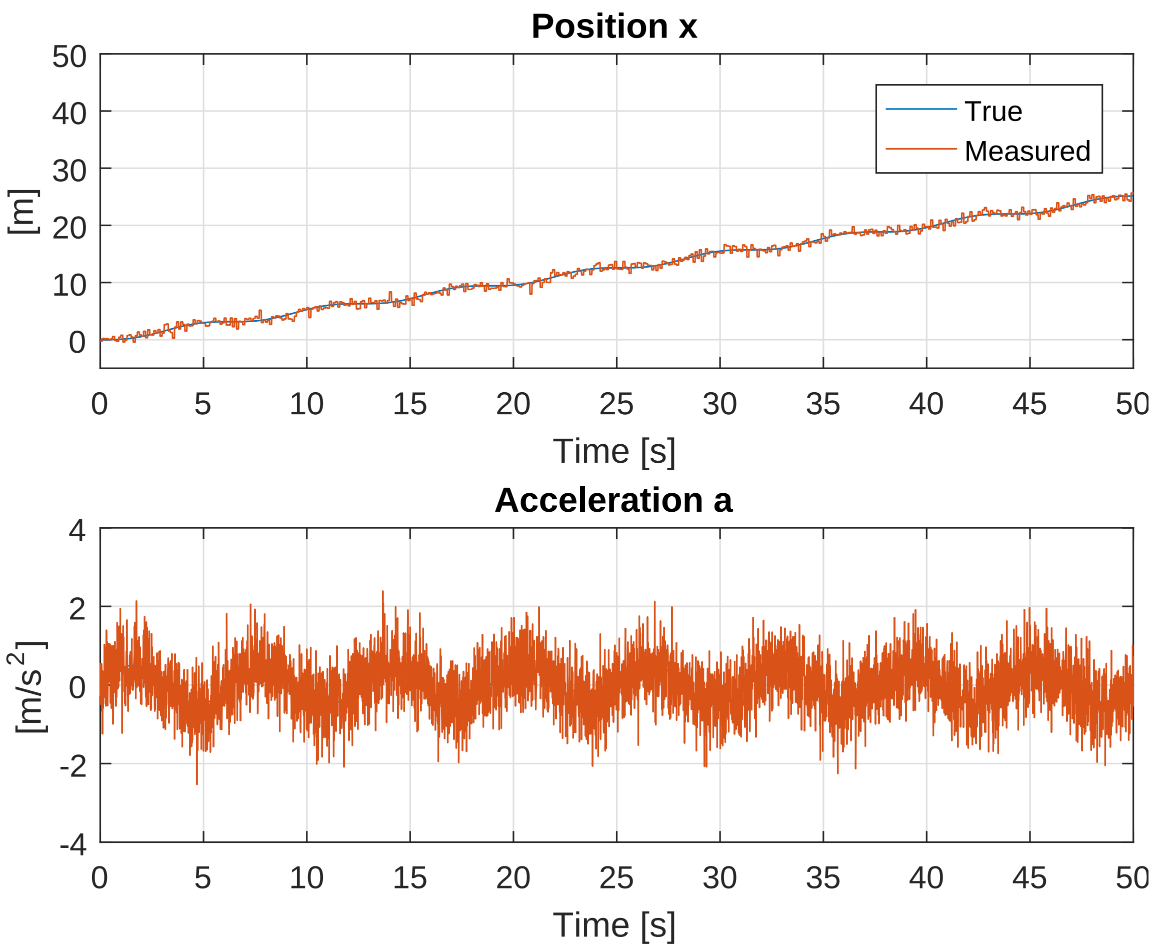

Figure 4: Measured position and acceleration for task 3.

The continous Kalman filter can be written as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{K}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}) \tag{9}$$

where

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{x} \\ \hat{v} \\ \hat{b_1} \\ \hat{\theta} \\ \hat{b_2} \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_2} \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

When the system is discretized it will take the form

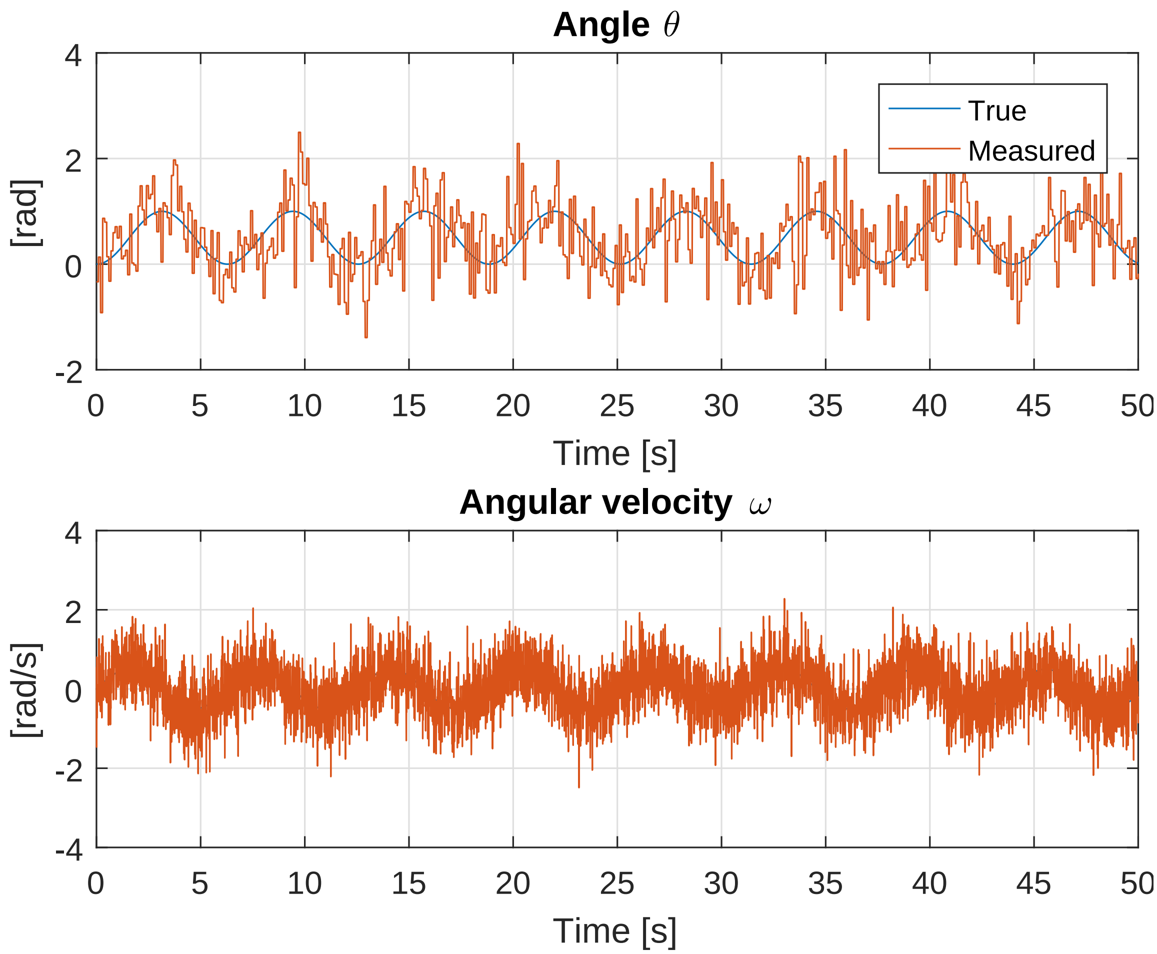

Figure 5: Measured orientation angle and angular velocity for task 3.

$$x(k+1) = \boldsymbol{\Phi}(k)\boldsymbol{x}(k) + \boldsymbol{\Delta}(k)\boldsymbol{u}(k) + \boldsymbol{\Gamma}\boldsymbol{w}(k) \tag{10a}$$

$$y(k) = \boldsymbol{H}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{10b}$$

Which gives

$$\boldsymbol{\Phi} = I + h\boldsymbol{A}(t_k) \tag{11a}$$

$$\boldsymbol{\Delta}(k) = h\boldsymbol{B}(t_k) \tag{11b}$$

$$\boldsymbol{\Gamma}(k) = \boldsymbol{E} \tag{11c}$$

The design matrices of the Kalman filter, $\boldsymbol{Q}$ and $\boldsymbol{R}$, are chosen based on knowledge about the noise that is present in the system, namely the variance of the process noise and the measurement noise:

$$\boldsymbol{Q}_d(k) = \begin{bmatrix} Var(w_1) & 0 & 0 & 0 \\ 0 & Var(w_2) & 0 & 0 \\ 0 & 0 & Var(w_3) & 0 \\ 0 & 0 & 0 & Var(w_4) \end{bmatrix}, \ \boldsymbol{R}_d(k) = \begin{bmatrix} Var(v_1) & 0 \\ 0 & Var(v_2) \end{bmatrix}. \tag{12}$$

And they are discretized

$$\boldsymbol{Q}_d = h\boldsymbol{Q}, \ \ \boldsymbol{R}_d = \frac{1}{h} \tag{13}$$

The initial condistions of the filter are chosen as

$$\bar{\boldsymbol{x}}(0) = \boldsymbol{x}_0 \tag{14a}$$

$$\bar{\boldsymbol{P}}(0) = E[(\boldsymbol{x}(0) - \hat{\boldsymbol{x}}(0))(\boldsymbol{x}(0) - \hat{\boldsymbol{x}}(0))^T] = \boldsymbol{P}_0. \tag{14b}$$

The Kalman gain matrix is calculated by

$$\boldsymbol{K}(k) = \bar{\boldsymbol{P}}(k)\boldsymbol{H}^T(k)[\boldsymbol{H}(k)\bar{\boldsymbol{P}}(k)\boldsymbol{H}^T(k) + \boldsymbol{R}_d(k)]^{-1} \tag{15}$$

and the state estimation performed at every timestep is defined as

$$\hat{\boldsymbol{x}}(k) = \bar{\boldsymbol{x}}(k) + \boldsymbol{K}(k)[\boldsymbol{y}(k) - \boldsymbol{H}(k)\bar{\boldsymbol{x}}(k)]. \tag{16}$$

The error covariance update is defined as

$$\hat{\boldsymbol{P}}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k)]\bar{\boldsymbol{P}}(k)[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}(k)]^T + \boldsymbol{K}(k)\boldsymbol{R}_d(k)\boldsymbol{K}^T(k). \tag{17}$$

The propagation of the system is updated in both the state estimation and the error covariance

$$\bar{\boldsymbol{x}}(k+1) = \boldsymbol{\Phi}(k) + \hat{\boldsymbol{x}}(k) + \Delta(k)\boldsymbol{u}(k) \tag{18a}$$

$$\bar{\boldsymbol{P}}(k+1) = \boldsymbol{\Phi}(k)\hat{\boldsymbol{P}}(k)\boldsymbol{\Phi}^T(k) + \boldsymbol{\Gamma}(k)\boldsymbol{Q}_d(k)\boldsymbol{\Gamma}^T(k). \tag{18b}$$

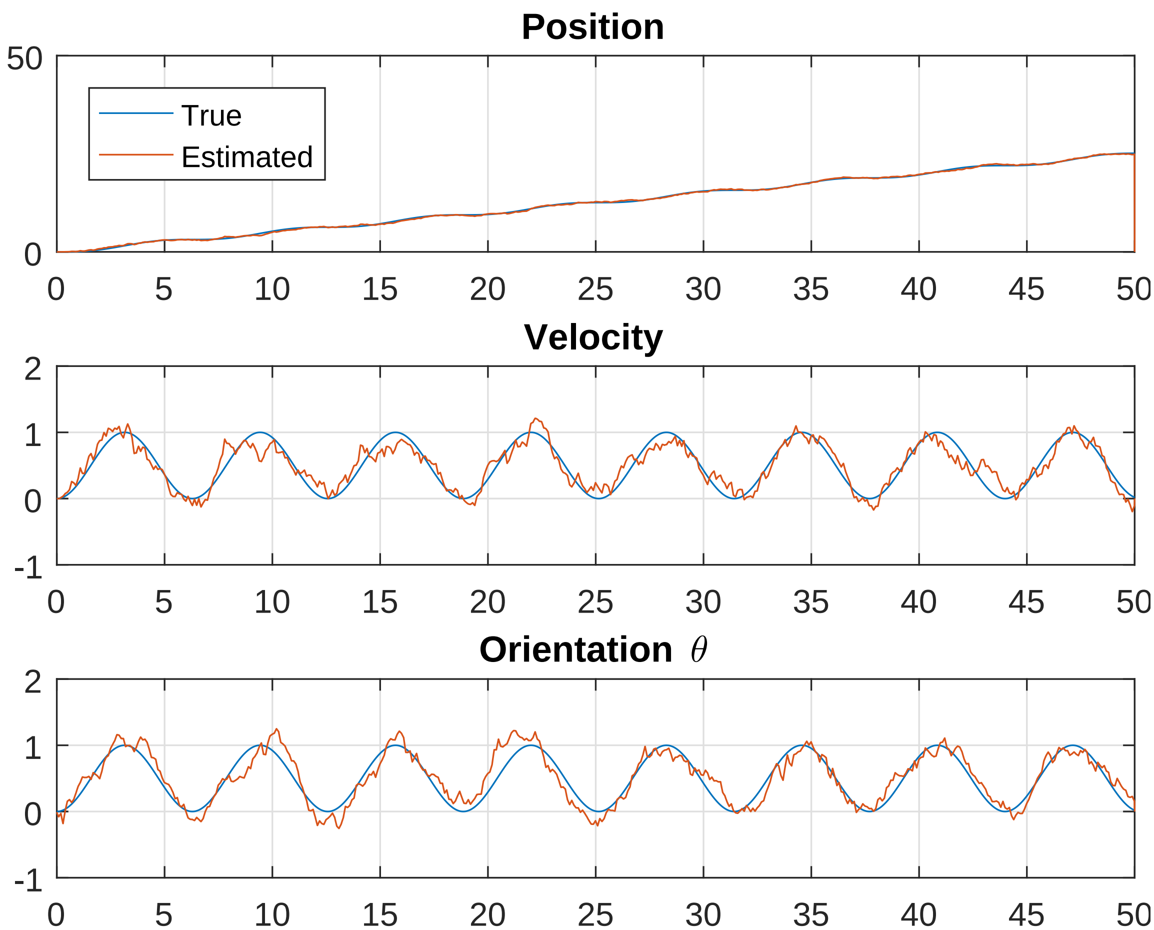

Figure 6: The estimated states for task 4.

Task 5

The difference between a direct and an indirect Kalman filter is that an indirect filter estimates the error of the measurements based on an error model. The error model for the three states x, v and θ will in this case be [1]:

$$\delta\dot{x} = \delta v \tag{19a}$$

$$\delta\dot{v} = b_1 + w_1 \tag{19b}$$

$$\dot{b_1} = -\frac{1}{T_1}b_1 + w_2 \tag{19c}$$

$$\delta\dot{\theta} = b_2 + w_3 \tag{19d}$$

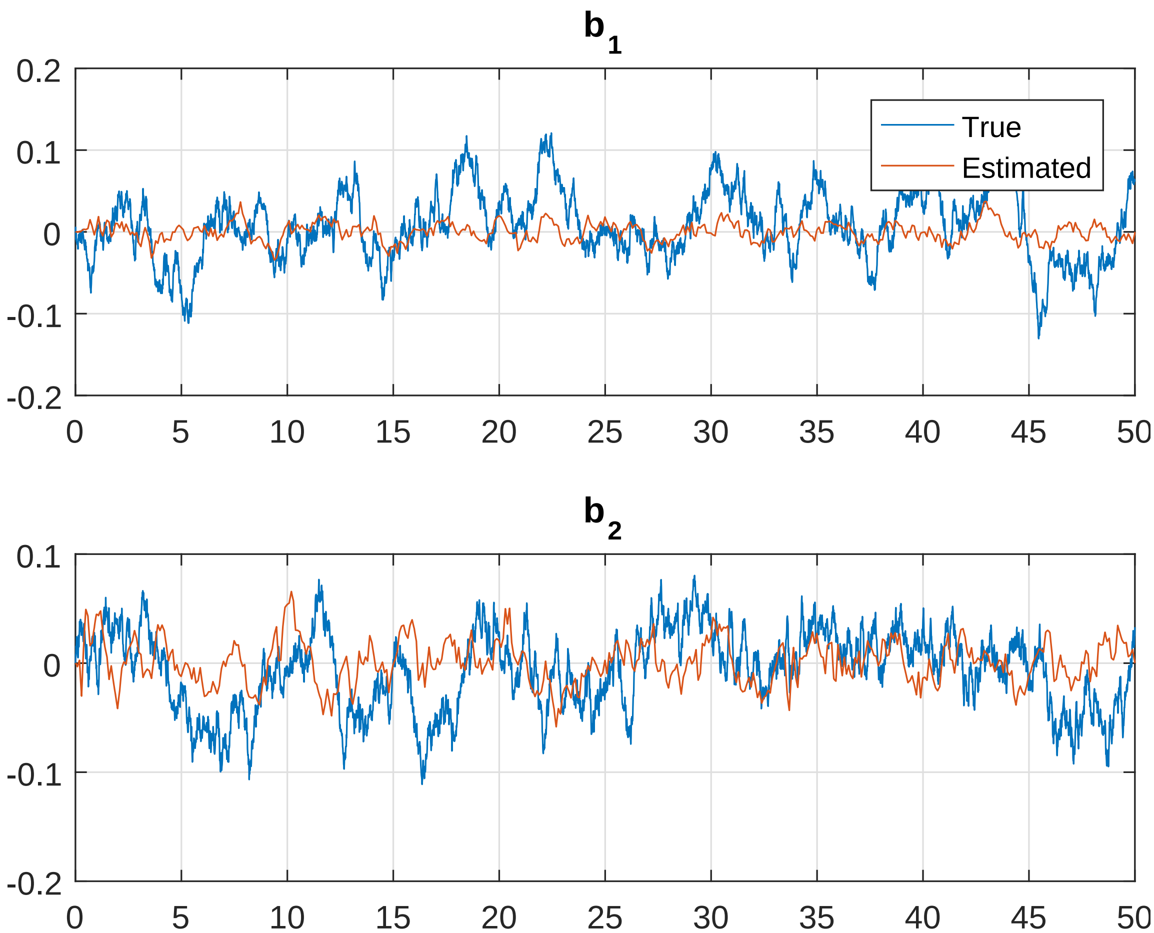

Figure 7: Estimated bias for task 4.

$$\dot{b_2} = -\frac{1}{T_2} b_2 + w_4 \tag{19e}$$

These error models will together form the following states for the Kalman filter

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \delta\hat{x} \\ \delta\hat{v} \\ \hat{b_1} \\ \delta\hat{\theta} \\ \hat{b_2} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} x_{GPS} - x_{INS} + v_1 \\ \theta_{GPS} - \theta_{INS} + v_2 \end{bmatrix} \tag{20}$$

which together will form the Kalman equation

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{\Phi}\hat{\boldsymbol{x}} + \boldsymbol{K}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}) \tag{21}$$

The Φ, $\boldsymbol{\Delta}$, $\boldsymbol{H}$ and $\boldsymbol{\Gamma}$ matrices will stay the same as in the previous task, as will the design matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$.

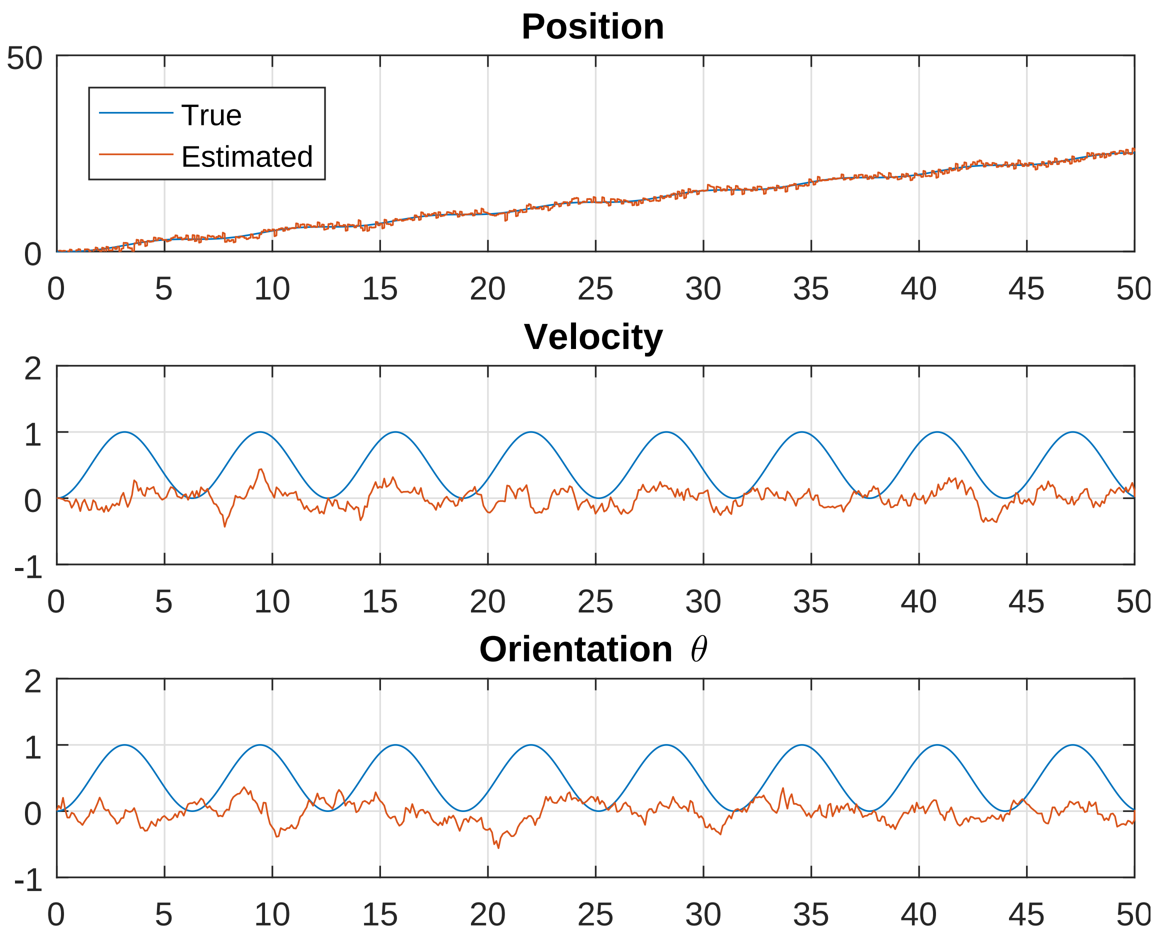

Figure 8: The estimated states for task 5.

When comparing the Kalman filters there are a few differences. Figures 6 and 8 shows that while both filters does a pretty good job estimating the position, the indirect filter has more noise in the estimated position than the direct filter. And when comparing v and θ, the indirect filter is far from as good as the direct filter in estimating the states. However, figures 7 and 9 shows that the indirect filter is better than the direct filter in estimating the bias in the system.

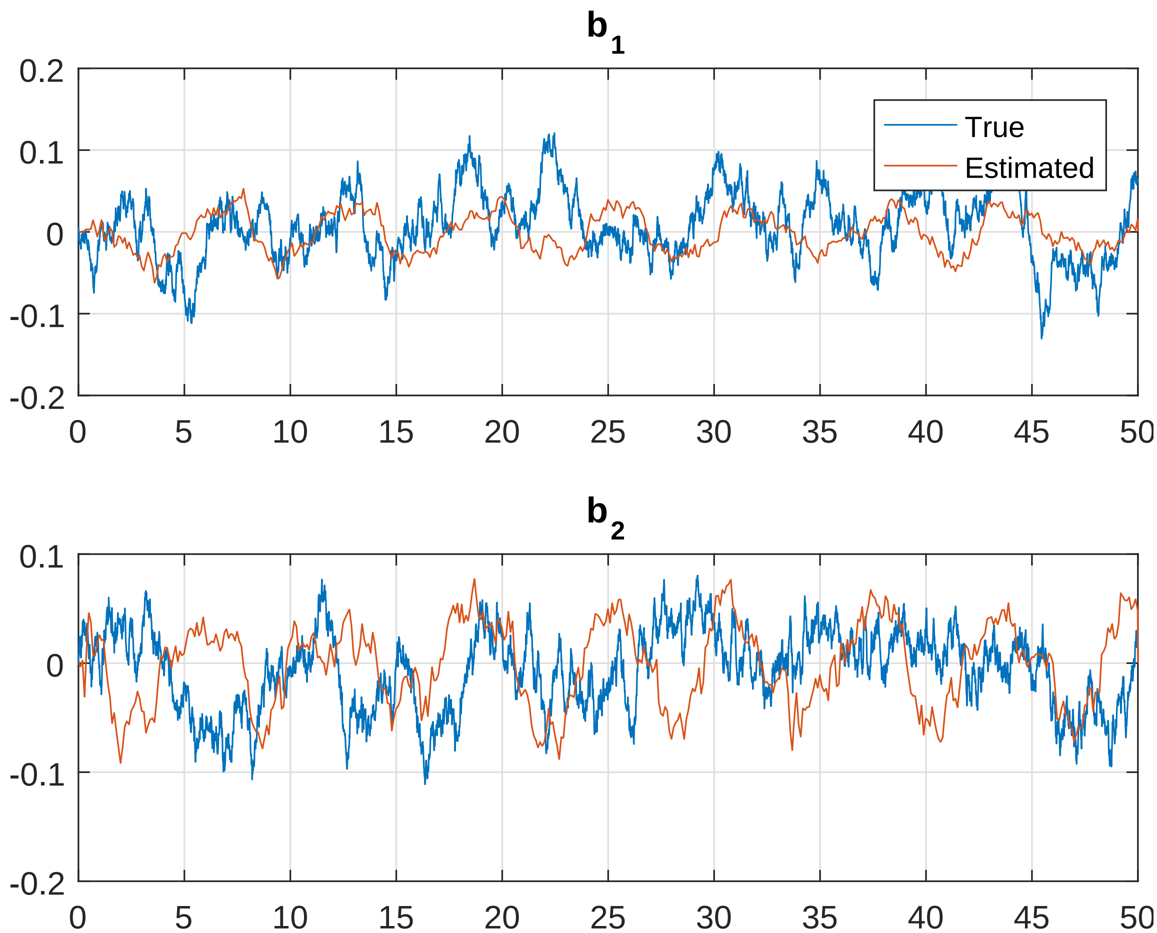

Figure 9: Estimated bias for task 5.

References

[1] Vik, Bjørnar (2014) *"Integrated Satellite and Inertial Navigation Systems"*, Norwegian University of Science and Technology, Department of Engineering Cybernetics, Trondheim

[2] Wikipedia, *https://en.wikipedia.org/wiki/White_noise*, accessed 05.11.2016