ME 7247: Advanced Control Systems **Fall 2022–23**

Lecture 11: The Kalman Filter

Friday, October 14, 2022

Lecturer: Laurent Lessard *Scribe: Xavier Hubbard*

In this lecture we discuss how multiple noisy measurements can be combined to estimate the state of a dynamical system. To this end we outline the development of the Kalman Filter from three perspectives while exploring their trade-offs in design and performance. We first explore the Kalman filter as a large estimation problem. We then view it as MAP optimization problem. And finally we develop the Kalman Filter as a recursive problem.

1 Three views of the Kalman Filter

1.1 Problem Definition

The dynamics of the system will be defined as follows:

$$\begin{aligned} x_{t+1} &= Ax_t + w_t, & w_t &\sim \mathcal{N}(0, \Sigma_w) \\ y_t &= Cx_t + v_t, & v_t &\sim \mathcal{N}(0, \Sigma_v) \\ & & x_0 &\sim \mathcal{N}(\mu_x, \Sigma_x) \end{aligned}$$

where we define μ_x and Σ_x as the initial estimates for the mean and covariance of x. We want to develop a way to calculate the expectation and covariance of x at some time t, given a set of observations occurring prior to a time s, such that s is before t. We write this

$$\begin{aligned} \hat{x}_{t|s} &= \mathbf{E}(x_t \mid y_0, y_1, \dots, y_s) \\ \Sigma_{t|s} &= \mathbf{Cov}(x_t \mid y_0, y_1, \dots, y_s) \end{aligned}$$

1.2 Large Scale Estimation

We can develop the formulation of the Kalman Filter as a solution to a large scale estimation problems. To see this, let's write a few iterations of the state equations:

$$\begin{aligned} x_1 &= Ax_0 + w_0 \\ x_2 &= A^2x_0 + Aw_0 + w_1 \\ x_3 &= A^3x_0 + A^2w_0 + Aw_1 + w_2 \\ &\vdots \\ x_t &= A^tx_0 + A^{t-1}w_0 + \cdots + w_{t-1}. \end{aligned}$$

We can capture this problem as a matrix equation:

$$\underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix}}_{\bar{x}} = \underbrace{\begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ A & I & 0 & \cdots & 0 \\ A^2 & A & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A^t & A^{t-1} & \cdots & A & I \end{bmatrix}}_{\bar{A}} \underbrace{\begin{bmatrix} x_0 \\ w_0 \\ w_1 \\ \vdots \\ w_{t-1} \end{bmatrix}}_{\bar{w}}$$

which we will write as

$$\bar{x} = \bar{A}\bar{w} \tag{1}$$

Now let's calculate the measurement equations:

$$\begin{aligned} y_0 &= Cx_0 + v_0 \\ y_1 &= CAx_0 + Cw_0 + v_1 \\ y_2 &= CA^2x_0 + CAw_0 + Cw_1 + v_2 \\ &\vdots \\ x_t &= CA^tx_0 + CA^{t-1}w_0 + \cdots + Cw_{t-1} + v_t \end{aligned}$$

We can also capture this problem as a matrix equation:

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix}}_{\bar{y}} = \underbrace{\begin{bmatrix} C & 0 & 0 & \cdots & 0 \\ CA & C & 0 & \cdots & 0 \\ CA^2 & CA & C & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ CA^t & CA^{t-1} & \cdots & CA & C \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} x_0 \\ w_0 \\ w_1 \\ \vdots \\ w_{t-1} \end{bmatrix}}_{\bar{w}} + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_t \end{bmatrix}}_{\bar{v}}$$

which we will write

$$\bar{y} = \bar{H}\bar{w} + \bar{v}. \tag{2}$$

Since we know w and v are Gaussian we can write:

$$\bar{w} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_x & & & \\ & \Sigma_w & & \\ & & \ddots & \\ & & & \Sigma_w \end{bmatrix}\right) = \mathcal{N}(e_1\mu_x, W) \tag{3a}$$

$$\text{and} \qquad \bar{v} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_v & & \\ & \ddots & \\ & & \Sigma_v \end{bmatrix}\right) = \mathcal{N}(0, V) \tag{3b}$$

This also works in the case where Σ_w or Σ_v varies over time. There, you would have the appropriate Σ_{w_t} and Σ_{v_t} on the block diagonals of the W and V matrices.

Combining (1) and (2) and (3), we obtain:

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix} = \begin{bmatrix}\bar{A} & 0\\ \bar{H} & I\end{bmatrix}\begin{bmatrix}\bar{w}\\ \bar{v}\end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix}\bar{w}\\ \bar{v}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}e_1\mu_x\\ 0\end{bmatrix}, \begin{bmatrix}W & 0\\ 0 & V\end{bmatrix}\right). \tag{4}$$

This is a linear transformation of Gaussian random variables, which takes the form:

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\bar{A}e_1\mu_x\\ \bar{H}e_1\mu_x\end{bmatrix}, \begin{bmatrix}\bar{A}W\bar{A}^\mathsf{T} & \bar{A}W\bar{H}^\mathsf{T}\\ \bar{H}W\bar{A}^\mathsf{T} & \bar{H}W\bar{H}^\mathsf{T} + V\end{bmatrix}\right). \tag{5}$$

Now recall that we already know how to find the conditional of a joint Gaussian distribution:

$$\begin{bmatrix}x\\ y\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu_x\\ \mu_y\end{bmatrix}, \begin{bmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_y\end{bmatrix}\right) \implies (x \mid y) \sim \mathcal{N}\Big(\mu_x + \Sigma_{xy}\Sigma_y^{-1}(y - \mu_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx}\Big).$$

Which means we can find the final expected value $\mathbf{E}(\bar{x} \mid \bar{y})$ as:

$$\mathbf{E}(\bar{x} \mid \bar{y}) = \bar{A}e_1\mu_x + \bar{A}W\bar{H}^\mathsf{T}(\bar{H}W\bar{H}^\mathsf{T} + V)^{-1}(\bar{y} - \bar{H}e_1\mu_x). \tag{6}$$

One of the interesting things about this problem is that the solution $\mathbf{E}(\bar{x} \mid \bar{y})$ has the form:

$$\mathbf{E}(\bar{x} \mid \bar{y}) = \begin{bmatrix}\hat{x}_{0|t}\\ \hat{x}_{1|t}\\ \hat{x}_{2|t}\\ \vdots\\ \hat{x}_{t|t}\end{bmatrix}.$$

This means that this procedure solves for the entire optimal history of the state even if the user is only interested in the most recent estimates. This is the main drawback of this method; computation and storage complexity scale as the time horizon grows.

1.3 MAP Problem

It is also possible to view the formulation of the Kalman Filter through the lens of Maximum A Posteriori (MAP) estimation. The problem takes the form:

$$\min_{\bar{x}} \begin{bmatrix}\bar{x} - \bar{A}e_1\mu_x\\ \bar{y} - \bar{H}e_1\mu_x\end{bmatrix}^\mathsf{T} \begin{bmatrix}\bar{A}W\bar{A}^\mathsf{T} & \bar{A}W\bar{H}^\mathsf{T}\\ \bar{H}W\bar{A}^\mathsf{T} & \bar{H}W\bar{H}^\mathsf{T} + V\end{bmatrix}^{-1} \begin{bmatrix}\bar{x} - \bar{A}e_1\mu_x\\ \bar{y} - \bar{H}e_1\mu_x\end{bmatrix}. \tag{7}$$

The matrix inverse in the middle poses a bit of a problem, but we remember that we can decompose it into the following:

$$\begin{aligned}\begin{bmatrix}\bar{A}W\bar{A}^\mathsf{T} & \bar{A}W\bar{H}^\mathsf{T}\\ \bar{H}W\bar{A}^\mathsf{T} & \bar{H}W\bar{H}^\mathsf{T} + V\end{bmatrix}^{-1} &= \left[\begin{bmatrix}\bar{A} & 0\\ \bar{H} & I\end{bmatrix}\begin{bmatrix}W & 0\\ 0 & V\end{bmatrix}\begin{bmatrix}\bar{A}^\mathsf{T} & \bar{H}\\ 0 & I\end{bmatrix}\right]^{-1}\\ &= \begin{bmatrix}\bar{A}^\mathsf{T} & \bar{H}\\ 0 & I\end{bmatrix}^{-1}\begin{bmatrix}W^{-1} & 0\\ 0 & V^{-1}\end{bmatrix}\begin{bmatrix}\bar{A} & 0\\ \bar{H} & I\end{bmatrix}^{-1}\end{aligned}$$

We can solve for the inverse quite simply:

$$\begin{bmatrix} \bar{A} & 0 \\ \bar{H} & I \end{bmatrix}^{-1} = \begin{bmatrix} \bar{A}^{-1} & 0 \\ -\bar{H}\bar{A}^{-1} & I \end{bmatrix}.$$

This runs into a new matrix inverse. Fortunately, $\bar{A}^{-1}$ has a form that is very easy to invert. Take for example a 4-dimensional version of $\bar{A}$:

$$\bar{A}_4 = \begin{bmatrix} I & 0 & 0 & 0 \\ A & I & 0 & 0 \\ A^2 & A & I & 0 \\ A^3 & A^2 & A & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ -A & I & 0 & 0 \\ 0 & -A & I & 0 \\ 0 & 0 & -A & I \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

Next what is $-\bar{H}\bar{A}^{-1}$? If $\bar{C} = \operatorname{diag}(C, \dots, C)$, we have

$$\bar{H} = \bar{C}\bar{A} \quad \Longrightarrow \quad -\bar{H}\bar{A}^{-1} = -\bar{C}.$$

And so we are able to find a whole solution. Plugging things back in, Eq. (7) becomes:

$$\begin{aligned} &\min_{\bar{x}} \begin{bmatrix} \bar{x} - \bar{A}e_1\mu_x \\ \bar{y} - \bar{H}e_1\mu_x \end{bmatrix}^\mathsf{T} \begin{bmatrix} \bar{A}^{-\mathsf{T}} & -\bar{C}^\mathsf{T} \\ 0 & I \end{bmatrix} \begin{bmatrix} W^{-1} & 0 \\ 0 & V^{-1} \end{bmatrix} \begin{bmatrix} \bar{A}^{-1} & 0 \\ -\bar{C} & I \end{bmatrix} \begin{bmatrix} \bar{x} - \bar{A}e_1\mu_x \\ \bar{y} - \bar{H}e_1\mu_x \end{bmatrix}. \\ &= \min_{\bar{x}} \begin{bmatrix} \bar{A}^{-1}\bar{x} - e_1\mu_x \\ \bar{y} - \bar{C}\bar{x} \end{bmatrix}^\mathsf{T} \begin{bmatrix} W^{-1} & 0 \\ 0 & V^{-1} \end{bmatrix} \begin{bmatrix} \bar{A}^{-1}\bar{x} - e_1\mu_x \\ \bar{y} - \bar{C}\bar{x} \end{bmatrix}. \end{aligned} \tag{8}$$

Here we can simplify further by noticing $\bar{A}^{-1}\bar{x}$:

$$\bar{A}^{-1}\bar{x} = \begin{bmatrix} I & & & & \\ -A & I & & & \\ & -A & I & & \\ & & \ddots & \ddots & \\ & & & -A & I \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 - Ax_0 \\ x_2 - Ax_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} x_0 \\ w_0 \\ w_1 \\ \vdots \end{bmatrix}.$$

Plugging this simplification back into (8) we get:

$$\min_{\bar{x}} \quad \begin{bmatrix} x_0 - \mu_x \\ x_1 - Ax_0 \\ \vdots \end{bmatrix}^\mathsf{T} W^{-1} \begin{bmatrix} x_0 - \mu_x \\ x_1 - Ax_0 \\ \vdots \end{bmatrix} + \begin{bmatrix} y_0 - Cx_0 \\ y_1 - Cx_1 \\ \vdots \end{bmatrix}^\mathsf{T} V^{-1} \begin{bmatrix} y_0 - Cx_0 \\ y_1 - Cx_1 \\ \vdots \end{bmatrix}$$

Which can be simplified to the maximization problem:

$$\min_{\bar{x}} \left(\|x_0 - \mu_x\|^2_{\Sigma_x^{-1}} + \sum_t \|x_{t+1} - Ax_t\|^2_{\Sigma_w^{-1}} + \sum_t \|y_t - Cx_t\|^2_{\Sigma_v^{-1}} \right) \tag{9}$$

We see here an optimization problem with a trade-off. We can think of this has finding the estimates of x such that w_t and v_t are small. The relative sizes of the variances Σ_x, Σ_w, Σ_v determine which terms have a larger weight/importance in the optimization. The glaring issue with this methodology is that the addition of any new information is difficult to deal with due to all the terms being coupled. In other words, we would need to re-solve the entire optimization problem every time new information is included. We will next seek a recursive implementation of the solution.

1.4 The Recursive Kalman Filter

The Kalman Filter can be broken into two steps: the measurement update and the time update. They can be thought of in the following way:

$$\begin{bmatrix} \hat{x}_{t|t-1} \\ \Sigma_{t|t-1} \end{bmatrix} \xrightarrow[\text{measurement update}]{y_t} \begin{bmatrix} \hat{x}_{t|t} \\ \Sigma_{t|t} \end{bmatrix} \xrightarrow[\text{time update}]{} \begin{bmatrix} \hat{x}_{t+1|t} \\ \Sigma_{t+1|t} \end{bmatrix}$$

1.4.1 Time Update

For the time update, we wish to see how the state of the system evolves as time progresses given our current model for the state. Let's review our state equations:

$$\begin{aligned} x_t &\sim \mathcal{N}(\hat{x}_{t|t}, \Sigma_{t|t}) \\ x_{t+1} &= Ax_t + w_t. \end{aligned}$$

We can rewrite this as:

$$x_{t+1} = \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x_t \\ w_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} \hat{x}_{t|t} \\ 0 \end{bmatrix}, \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} \Sigma_{t|t} & 0 \\ 0 & \Sigma_w \end{bmatrix} \begin{bmatrix} A^\mathsf{T} \\ I \end{bmatrix} \right) \tag{10a}$$

$$x_{t+1} \sim \mathcal{N}\left(A\hat{x}_{t|t},\ A\Sigma_{t|t}A^\mathsf{T} + \Sigma_w \right) \tag{10b}$$

This tells us that x_{t+1} is Gaussian as described above. The simple form of this equation is as a result of the assumption that w_t and v_t are uncorrelated. If they *are* correlated, then extra care has to be taken when formulating the solution, since w_t will not be independent of y_t.

1.4.2 Measurement Update

For the measurement update, we wish to describe how the model for the state changes as a result of an observation. This can be described as follows:

$$\begin{bmatrix} \hat{x}_{t|t-1} \\ y_{t|t-1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \hat{x}_{t|t-1} \\ v_t \end{bmatrix} \tag{11}$$

where $x_{t|t-1} \sim \mathcal{N}(\hat{x}_{t|t-1}, \Sigma_{t|t-1})$ and $v_t \sim \mathcal{N}(0, \Sigma_v)$. This lets us write the entire system as a Gaussian:

$$\begin{bmatrix} x_{t|t-1} \\ y_{t|t-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{x}_{t|t-1} \\ C\hat{x}_{t|t-1} \end{bmatrix}, \begin{bmatrix} \Sigma_{t|t-1} & \Sigma_{t|t-1}C^\mathsf{T} \\ C\Sigma_{t|t-1} & C\Sigma_{t|t-1}C^\mathsf{T} \end{bmatrix} + \Sigma_v \right). \tag{12}$$

Knowing the conditional form of a Gaussian we can write:

$$x_{t|t-1}|y_{t|t-1} = x_{t|t} \sim \mathcal{N}\left(\hat{x}_{t|t}, \Sigma_{t|t} \right) \tag{13}$$

where

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + \Sigma_{t|t-1}C^\mathsf{T}(C\Sigma - t|t - 1C^\mathsf{T} + \Sigma_v)^{-1}(y_t - C\hat{x}_{t|t-1}) \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{t|t-1}C^\mathsf{T}(C\Sigma_{t|t-1}C^\mathsf{T} + \Sigma_v)^{-1}C\Sigma_{t|t-1}. \end{aligned}$$

We can simplify the notation a bit with the following:

$$\begin{aligned} \hat{x}_t &:= \hat{x}_{t|t-1} \\ \Sigma_t &:= \Sigma_{t|t-1} \end{aligned}$$

giving us the final form of the state and covariance update:

$$\hat{x}_{t+1} = A\hat{x}_t + A\Sigma_t C^{\mathsf{T}}(C\Sigma_t C^{\mathsf{T}} + \Sigma_v)^{-1}(y_t - C\hat{x}_t) \tag{14}$$

$$\Sigma_{t+1} = A\Sigma_t A^{\mathsf{T}} + \Sigma_w - A\Sigma_{t|t-1} C^{\mathsf{T}}(C\Sigma_{t|t-1} C^{\mathsf{T}} + \Sigma_v)^{-1} C\Sigma_t A^{\mathsf{T}} \tag{15}$$

Note that (15) has no dependence on y meaning the error covariance can be determined ahead of time, before any measurements have been observed. We can simplify our notation one step further by defining the Kalman Gain, L_t:

$$L_t := -A\Sigma_t C^{\mathsf{T}}(C\Sigma_t C^{\mathsf{T}} + \Sigma_v)^{-1}. \tag{16}$$

We can at last define our full Kalman Filter update in its simplified form:

$$\hat{x}_{t+1} = (A + L_t C)\hat{x}_t - L_t y_t \tag{17a}$$

$$\Sigma_{t+1} = (A + L_t C)\Sigma_t A^{\mathsf{T}} + \Sigma_w \tag{17b}$$

where we initialize the filter using the prior distribution on x: $\hat{x}_0 = \mu_x$ and $\Sigma_0 = \Sigma_x$.

1.4.3 Error Dynamics

Lastly we will discuss how the error of our system changes. we define our error as follows:

$$e_t = x_t - \hat{x}_t \implies \mathbf{Cov}(e_t) = \Sigma_t$$

This leads to the *error dynamics*

$$e_{t+1} = (A + L_t C)e_t + (w_t + L_t v_t)$$

These error dynamics reveal that the error does not depend explicitly on the measurements y_t. We can take the covariance of both sides and we obtain

$$\Sigma_{t+1} = (A + L_t C)\Sigma_t (A + L_t C)^{\mathsf{T}} + (\Sigma_w + L_t \Sigma_v L_t^{\mathsf{T}})$$

which is actually the same (after some algebra) as Eq. (17b).

2 Spring-mass-damper example

As a simple example, consider the spring-mass-damper system shown in Fig. 1. This example was borrowed from Engr207b lecture notes on the Kalman filter[1] by Sanjay Lall.

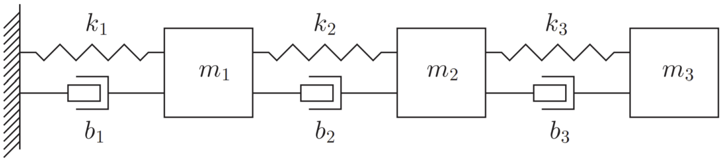


Figure 1: Spring-mass-damper system

We chose $k_i = 2$, $b_i = 0.1$, $m_i = 1$. The system has six states (positions and velocities of the three masses). We have noisy measurements of masses 1 and 2, and our task is to estimate the position of mass 3. Here are the equations describing the dynamics of the system

$$\dot{x}(t) = Ax(t) + Bw(t)$$
$$y(t) = Cx(t) + v(t)$$

Where A, B, C are given by:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4 & 2 & 0 & -0.2 & 0.1 & 0 \\ 2 & -4 & 2 & 0.1 & -0.2 & 0.1 \\ 0 & 2 & -2 & 0 & 0.1 & -0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The first three equations state that the derivative of position is equal to velocity, and this is an exact relationship (so there is no process noise). The last three equations describe force balance on the three masses, and here the process noise enters in as a force disturbance on the masses. We discretized these equations using a timestep of $h = 0.1$ to obtain equations of the form:

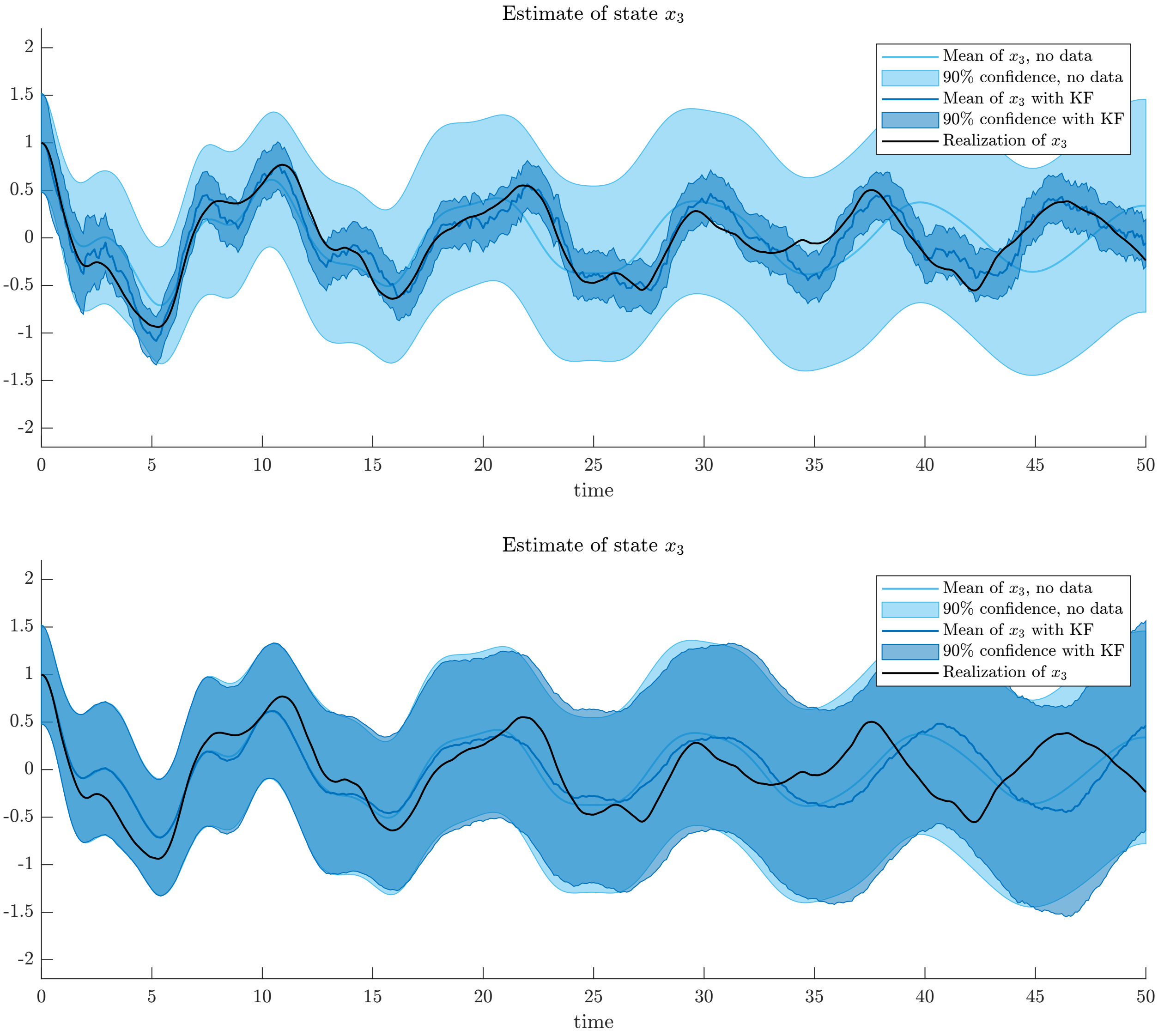
$$x_{t+1} = A_d x_t + B_d w_t$$
$$y_t = C_d x_t + v_t$$

The details of the discretization are not important. You can look at the posted code for more details if you're interested. We used $w_t \sim \mathcal{N}(0, \sigma_w^2 I)$ and $v_t \sim \mathcal{N}(0, \sigma_v^2)$ with $\sigma_w = 0.2$ and $\sigma_v = 0.1$.

In Fig. 2, we simulate one particular realization of a trajectory of the system, and we plot the confidence interval using no measurements (just using the time update $\hat{x}_{t+1} = A_d \hat{x}_t$ and $\Sigma_{t+1} = A_d \Sigma_t A_d^{\mathsf{T}} + W$), and we also plot the confidence interval using the Kalman filter.

The no-measurement confidence interval will be the same no matter what the noise realization looks like. Of course, the KF provides a tighter confidence interval since it is measurement- and therefore realization-dependent. In the bottom panel of Fig. 2, we see what happens when we make the measurement noise very large. Here, the KF estimate is not much better than using no measurements at all.

[1] `http://engr207b.stanford.edu/lectures/kalman_filter_2011_02_22_01.pdf`



Estimate of state x_3

Estimate of state x_3

Figure 2: Simulation of spring-mass-damper system from Fig. 1 using $k_i = 2$, $m_i = 1$, $b_i = 0.1$. We discretized using a timestep of 0.1. Process noise has standard deviation $\sigma_w = 0.2$. We used measurement noise with $\sigma_v = 0.1$ (top panel) and $\sigma_v = 100$ (bottom panel). When there is high measurement noise, the KF estimate tends to the no-measurement estimate.