Chapter 3

State estimation and observers

3.1 Introduction

Whether it was for stabilization, with or without integrator, tracking a trajectory, we always assumed, when applying linear state-feedback, that the state vector $\mathbf{x}(t)$ was entirely accessible for measurements, and that we were able to measure the *whole* state.

In practice, this might sometimes be the case, if one has enough sensors to measure the whole state, but not always. In some situations, having enough sensors might be too costly, or just plainly impractical. When the state is not available, what one would like to have is an *estimate* of $\mathbf{x}(t)$.

This estimate, which we will note here $\hat{\mathbf{x}}(t)$, can then be used for state-feedback (instead of the actual state), but not only: estimating the state of a system is useful for other applications as well: monitoring and fault-diagnosis. As we will see, state estimation can also be related to filtering.

What we will see in the present chapter are a few techniques to compute an estimate of the state $\mathbf{x}(t)$ using only the available measurements $\mathbf{y}(t)$ as well as the knowledge of the system's state-space representation.

Depending on the structure of this state-space representation, as well as its parameters (ie the matrices of the state-space representation for linear systems), it is not always *possible* to compute an estimate of the whole state. This is related to the notion of *observability*, notion that we will start this chapter with.

3.2 Observability and the Kalman criterion

3.2.1 Observability notions

Given the linear state-space representation of a particular system, we would like to know if it is possible, using the measurements $\mathbf{y}(t)$ (and maybe also the control input signal $\mathbf{u}(t)$) to compute/give an estimate of the state $\mathbf{x}(t)$. When the answer to this question is a resounding yes, the system is said to be *observable*.

In order to give a first intuition of what observability and state estimation are about, a block-diagram perspective can be used as a first approach. Let us consider this with the simple following example.

Example: Vehicle on a straight road with a GPS sensor

Consider the idealized model of a vehicle traveling in a straight line, and whose absolute position is given by a GPS sensor.

Figure 3.1: Vehicle with a GPS moving in a straight line

A state-space representation of an extremely simplified dynamical model of the vehicle could read, in component form:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\dfrac{d}{m}x_2(t) + \dfrac{1}{m}u(t) \end{cases} \tag{3.1}$$

where the output equation is

$$y(t) = x_1(t) \tag{3.2}$$

with $y(t)$ the position as measured by a GPS[1].

Drawing the block diagram of system (3.1)-(3.2), we obtain (see figure 3.2)

[1] In the above expression, the term d represents the combination of all the friction forces having an effect on the car (ie drag force due to the movement of the car in the air, but also the friction due to the wheels, etc.). The control input $u(t)$ simply represents the force generated by the engine (or electrical motor) allowing to move the car forward (even backward if need to park).

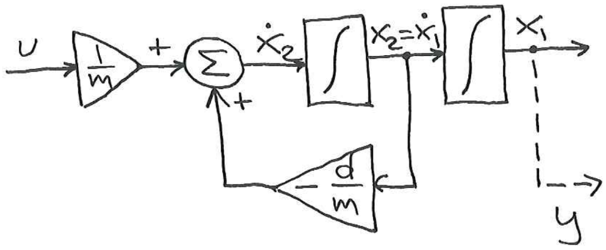

Figure 3.2: Block diagram of the vehicle-with-GPS example

By somewhat thinking in reverse, being able to compute an estimate of $\mathbf{x}(t)$ using the measurements $y(t)$ leads to saying that all the components of the state have an influence, direct or not, on the measurement. Looking at the flow of information in the block diagram figure 3.2, one can see that the state components are either directly connected to the measurement $y(t)$, like $x_1(t)$, or influence other state components, which are themselves connected to $y(t)$, like $x_2(t)$ (which influences $x_1(t)$, itself connected to $y(t)$. With this interpretation, the system is observable. □

Example: changing the sensor with a speedometer

Replacing now our GPS sensor with a speedometer, the output equation becomes

$$y(t) = x_2(t) \tag{3.3}$$

while the rest of the state-space representation remains the same. Rewiring our output, we have the block diagram shown in figure 3.3 below.

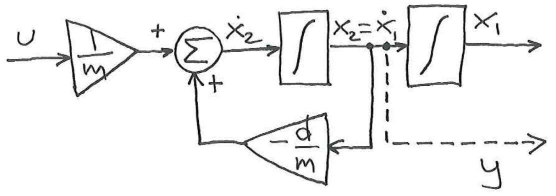

Figure 3.3: Example of a non-observable system

This time, notice that not *all* state components have an effect on the measurement $y(t)$. Indeed, $x_1(t)$ will be totally invisible to $y(t)$. The system is therefore *not* observable. Note that a simple physical explanation of that is that, since we measure only the velocity of the vehicle, it is therefore not possible to know the whole state at time t because we do not know where the vehicle started from. □

3.2.2 A criterion to check observability

Having at our disposal a systematic criterion to test observability of a state-space representation using only its matrices would be certainly more practical than looking at block diagrams each time, like we just did. Especially for systems of order larger than 2.

Before presenting such a criterion, let us give a more formal definition of observability for linear systems.

Definition: The system represented by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \\ y(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{3.4}$$

is said to be (completely) observable if there is a time $T > 0$ such that knowledge of $y(t)$ and $u(t)$[2] on the interval $[0, T]$ is sufficient to determine the whole state $\mathbf{x}(t)$. □

Rudolf Kalman proposed a criterion to systematically verify whether a system represented by a state-space representation is observable or not. Note that, at first glance, and assuming we have only one output, the problem does not seem so obvious since the dimension of $y(t)$ and $u(t)$ is 1 while the dimension of the state which we want to compute is n.

However, the idea for the criterion is pretty simple. Indeed, since $y(t)$ (and $u(t)$) is available on interval $[0, T]$, then we can compute its successive time-derivatives. Differentiating this output signal $n-1$ times successively, we obtain

$$\begin{cases} y(t) = \mathbf{C}\mathbf{x}(t) \\ \dot{y}(t) = \mathbf{C}\dot{\mathbf{x}}(t) = \mathbf{C}\mathbf{A}\mathbf{x}(t) + \mathbf{C}\mathbf{B}u(t) \\ \ddot{y}(t) = \mathbf{C}\ddot{\mathbf{x}}(t) = \mathbf{C}\mathbf{A}^2\mathbf{x}(t) + \mathbf{C}\mathbf{A}\mathbf{B}u(t) + \mathbf{C}\mathbf{B}\dot{u}(t) \\ \vdots \\ y^{(n-1)}(t) = \mathbf{C}\mathbf{x}^{(n-1)}(t) = \mathbf{C}\mathbf{A}^{n-1}\mathbf{x}(t) + \mathbf{C}\mathbf{A}^{n-2}\mathbf{B}u(t) + \cdots + \mathbf{C}\mathbf{B}u^{(n-2)}(t) \end{cases} \tag{3.5}$$

which gives us n equations for n unknowns, the components of the state $\mathbf{x}(t)$. Putting that in vectorial form, we thus get

$$\bar{\mathbf{y}}(t) = \mathbf{W}_o\mathbf{x}(t) + \mathbb{T}\bar{\mathbf{u}}(t), \tag{3.6}$$

where $\bar{\mathbf{y}}(t) = [y(t), \dot{y}(t), \ddot{y}(t), \cdots, y^{(n-1)}(t)]^T$ and $\bar{\mathbf{u}}(t) = [u(t), \dot{u}(t), \ddot{u}(t), \cdots, u^{(n-1)}(t)]^T$, and Toeplitz matrix $\mathbb{T}$ is given by

$$\mathbb{T} = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ \mathbf{CB} & 0 & 0 & ... & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \mathbf{CA}^{n-2}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & 0 \end{bmatrix}, \tag{3.7}$$

[2] In this section, we focus on the single-input-single-output case, ie $y(t), u(t) \in \mathbb{R}$.

while the matrix $\mathbf{W}_o$ written as

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{3.8}$$

is called *observability matrix*. An estimate $\hat{\mathbf{x}}(t)$ of state $\mathbf{x}(t)$ can then be obtained if and only if the row vectors of $\mathbf{W}_o$ are linearly independent, or, in other words, if the observability matrix $\mathbf{W}_o$ is nonsingular (if $\det(\mathbf{W}_o) \neq 0$).

In this case, we can simply compute an estimate of the state given by

$$\hat{\mathbf{x}}(t) = \mathbf{W}_o^{-1} \left(\bar{\mathbf{y}}(t) - \mathbb{T}\bar{\mathbf{u}}(t) \right) \tag{3.9}$$

Let us summarize the above discussion with the following important result.

Theorem: The system represented by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \qquad y(t) = \mathbf{C}\mathbf{x}(t) \tag{3.10}$$

is observable if and only if the $n \times n$ observability matrix

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{3.11}$$

is invertible. □

Example: Back to the vehicle with speedometer

Setting $d = 1$ and $m = 1$, recalling that we had the state-space representation

$$\begin{cases} \dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(t) \end{cases} \tag{3.12}$$

It is then easy to compute the observability matrix

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} & \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \tag{3.13}$$

and since

$$\det(\mathbf{W}_o) = 0, \tag{3.14}$$

the system is not observable, confirming thus our analysis using both basic thinking and block diagram interpretation (feel free to check that the car example with a GPS *is* observable). □

3.2.3 Observability of linear systems and the role played by inputs

For linear systems, the observability is not change depending on the presence or absence of control inputs. This comes mostly from the linear property. To see this mathematically, recall that

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \tag{3.15}$$

is the solution of the dynamical equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{3.16}$$

Using then the measurement equation of the state-space representation of a linear system, ie using $\mathbf{y}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$, we have

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 + \mathbf{C}\int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \mathbf{D}\mathbf{u}(t) \tag{3.17}$$

Considering for an instant equation (3.17), we see that $\mathbf{y}(t)$ is known since we measure it, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are also known because we assume we know the state-space representation of the considered system. Control signal $\mathbf{u}(t)$ is also assumed to be known, which means that the integral term of (3.17) is known, as well as $\mathbf{D}\mathbf{u}(t)$. In (3.17), the only unknown term is the initial state $\mathbf{x}_0$.

Putting together the know terms on one side, we get

$$\mathbf{y}_h(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 \tag{3.18}$$

where $\mathbf{y}_h(t)$ is defined as

$$\mathbf{y}_h(t) := \mathbf{y}(t) - \mathbf{C}\int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \mathbf{D}\mathbf{u}(t) \tag{3.19}$$

and can be seen as a new output signal containing the history of the effect of the input (ie the convolution term $\int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$). Note that we would have had the same expression as (3.18) had we had no input signal at all, except that $\mathbf{y}_h(t)$ would have been replaced by $\mathbf{y}(t)$. Hence the inputs do not play a role in the possibility or not of recovering the state, ie it does not change anything to the observability property of a linear system. This is why, in many books or tutorials about observability, it is generally assumed that we have a state-space representation of the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \end{cases} \tag{3.20}$$

While we will not use this property in order to obtain a criterion to check stability, we will see how it can come in handy at a later stage.

Note that the above property applies only to linear systems. For nonlinear systems, things get generally more tricky, and there are input signals that can change the observability property and make the system observable/unobservable. For example, think of the nonlinear measurement equation $y = x.u$. If $u = 0$, then it is going to be impossible to estimate the state at all.

3.3 State estimation using observers

As we have seen in the previous section, an estimate of the state can be theoretically obtained directly from the observability analysis, as given by the formula (3.9). However, since any measurement tends to be corrupted by noise, having an estimate based on the successive derivatives of these measurements is not such a good idea considering the usual noise-amplifying character of the derivative operator. Hence, it would be more suitable to have a state estimation algorithm whose description would not need to differentiate the $y(t)$ signals. Among these estimation techniques are the so-called dynamic observers, or simply observers, very widely used in control systems.

3.3.1 The observer as a virtual plant

The essential idea at the core of the observer concept goes like this: assume we make a virtual copy of the plant's model, ie a copy of the plant on the computer, and apply to it *the same control input* as the one applied on the real plant (see figure 3.4).

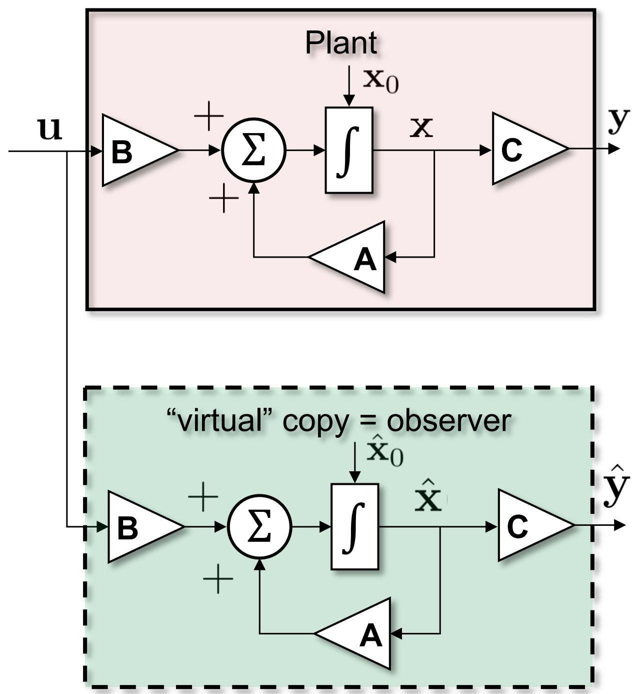

Figure 3.4: The observer as a virtual plant

The reasoning behind this idea is that, since both the plant and the observer have the same models, then if we apply to them the same input, then both plant and observer will exhibit the same dynamical behavior. Hence, the state of the observer, ie the state estimate $\hat{\mathbf{x}}(t)$, should have the same evolution as the state $\mathbf{x}(t)$. Now if the state $\mathbf{x}(t)$ of the plant is not accessible/measurable, the state estimate $\hat{\mathbf{x}}(t)$ certainly is, since the model is built on a computer, which makes every single variable accessible!

To make these few points a bit more rigorous, let us recall the expression of the plant given by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \\ y(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{3.21}$$

and since our observer is (for now) a mere copy of the plant, we have

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t), \qquad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0 \\ \hat{y}(t) = \mathbf{C}\hat{\mathbf{x}}(t) \end{cases} \tag{3.22}$$

Note that, since $\mathbf{x}(t)$ is unknown, there is no reason (unless we are really, really lucky) that both initial conditions are equal. Hence we have $\mathbf{x}_0 \neq \hat{\mathbf{x}}_0$. Saying that the observer is doing a good job means that, ideally, there would be no difference whatsoever between the state and its estimate. Let us therefore define this difference as

$$\tilde{\mathbf{x}}(t) := \hat{\mathbf{x}}(t) - \mathbf{x}(t) \tag{3.23}$$

and call it *estimation error*. Clearly, initially, the estimation error is given by

$$\tilde{\mathbf{x}}_0 = \hat{\mathbf{x}}_0 - \mathbf{x}_0 \neq 0 \tag{3.24}$$

The important question is how this estimation error evolves over time. To answer this, let us compute the so-called *estimation error dynamics*. They are simply obtained by computing the time-derivative of $\tilde{\mathbf{x}}(t)$:

$$\dot{\tilde{\mathbf{x}}}(t) = \dot{\hat{\mathbf{x}}}(t) - \dot{\mathbf{x}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) - \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{u}(t) = \mathbf{A}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)) \tag{3.25}$$

so that the estimation error dynamics are given by

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A}\tilde{\mathbf{x}}(t). \tag{3.26}$$

Thus, if the estimation error dynamics (3.26) is stable, then $\tilde{\mathbf{x}}(t)$ will eventually converge to 0 as $t \to \infty$, which means that $\hat{\mathbf{x}}(t)$ will converge to the true state $\mathbf{x}(t)$!

Since $\mathbf{A}$ is the matrix linked to the model of the plant, then this result means that the observer will work just great and its estimate will converge to the actual state of the plant *as long as this plant is stable*.

However, the downside of this technique would also mean that if the original plant is *not* stable, then neither will the error dynamics be, so that the estimate and the state will gradually drift apart from each other. There is a solution to this issue. We will keep the powerful idea of copying the plant dynamics, but will also add a little extra to make sure the observer does the right thing.

3.3.2 Feedback on the measurement

When we made a pure copy of the plant to design an observer, we used the control input signal $\mathbf{u}(t)$, which we measure, as information to feed to the observer. However, there is another signal we could use: the plant measurements $y(t)$! Hence the idea is to take the available measurements $y(t) = \mathbf{C}\mathbf{x}(t)$ to make corrections on the state estimate $\hat{\mathbf{x}}(t)$. Consider then the following observer equations

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}(y(t) - \hat{y}(t)) \\ \hat{y}(t) = \mathbf{C}\hat{\mathbf{x}}(t) \end{cases} \tag{3.27}$$

Where $\mathbf{L}$ is a column vector of dimension n representing the amount of corrections to be done on $\hat{\mathbf{x}}(t)$ based on the difference between the output $y(t)$ and the estimated output $\hat{y}(t)$. A block diagram of observer (3.27) is given in figure 3.5.

Note that if $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$ in (3.27), then $\hat{y}(t) = y(t)$ which means that $\mathbf{L}(y - \hat{y}) = 0$. This is an important implication, since, in this case, the observer has the same dynamics as the plant.
However, when $\hat{\mathbf{x}}(t) \neq \mathbf{x}(t)$, the feedback term $\mathbf{L}(y - \hat{y})$ corrects the state-estimate $\hat{\mathbf{x}}(t)$ so that it improves over time and gradually converges to $\mathbf{x}(t)$.

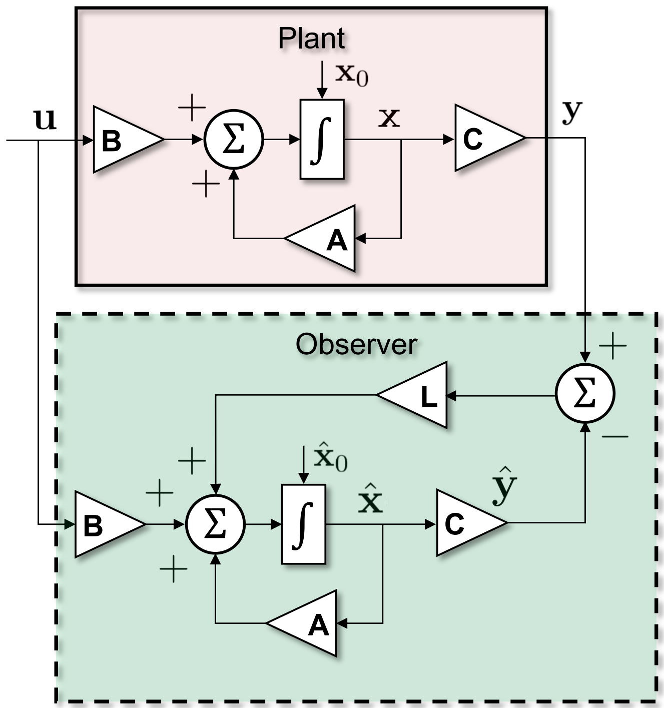

Figure 3.5: Block diagram of an observer

Let us now have a look at the estimation error dynamics corresponding to observer (3.27). Computing the derivative of $\tilde{\mathbf{x}}(t)$, we get

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t) = \dot{\hat{\mathbf{x}}}(t) - \dot{\mathbf{x}}(t) &= \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}(y(t) - \hat{y}(t)) - \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{u}(t) \\ &= \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{L}\mathbf{C}(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) - \mathbf{A}\mathbf{x}(t) \\ &= (\mathbf{A} - \mathbf{L}\mathbf{C})(\hat{\mathbf{x}}(t) - \mathbf{x}(t)) \qquad (3.28)\end{aligned}$$

which gives the estimation error dynamics

$$\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{x}}(t) \qquad (3.29)$$

Hence the state estimate $\hat{\mathbf{x}}(t)$ will converge to the state $\mathbf{x}(t)$ of the plant if the real part of each eigenvalue of $\mathbf{A} - \mathbf{L}\mathbf{C}$ is strictly negative.

3.3.3 The observer as a filter

When the measurement $y(t)$ is corrupted by noise, typically high-frequency noise, using an observer can be useful to produce a signal, the estimate, where this noise is dampened. Hence an observer can also be seen under the lens of signal processing and filters. To see this, let us consider the following very simple example.

Example: Observer for a "constant variable"

Assume we want to monitor a variable which, compared with the dynamics of the surrounding system, does not evolve much over a short period of time (take for example, as a very rough approximation, a room or oven whose temperature we monitor over 10 minutes or so). A trivial model of this is to assume that this signal is simply constant, so that we have

$$\dot{x}(t) = 0, \quad x(0) = x_0. \qquad (3.30)$$

We further assume that the sensor measuring this constant variable (which is an oxymoron by the way) is not of very good quality, and that the measurements are corrupted by a noise of sinusoidal form, so that we have

$$y(t) = x + \cos \omega t \qquad (3.31)$$

for the output equation, where we assume that the ω constant is relatively big, so that the noise is considered to be of high-frequency. Considering now model (3.30)-(3.31) without the term $\cos \omega t$, we would simply design an observer as usual, ie we make a copy of the plant complemented by a feedback on the measurements as follows,

$$\begin{cases} \dot{\hat{x}}(t) = 0 + l(y(t) - \hat{y}(t)) \\ \hat{y}(t) = \hat{x}(t) \end{cases}, \qquad (3.32)$$

so that, combining the two lines (3.32), we get

$$\dot{\hat{x}}(t) = -l\hat{x}(t) + ly(t) \qquad (3.33)$$

which is a dynamical system whose variable is $\hat{x}(t)$ and whose input signal is the measurement $y(t)$. Hence, in a sense, system (3.33) is filtering $y(t)$. To see a bit more precisely what it filters out exactly, let us first write the solution of system (3.33) as

$$\hat{x}(t) = e^{-lt}\hat{x}(0) + \int_0^t e^{-l(t-\tau)} l y(\tau) d\tau, \tag{3.34}$$

which, assuming for simplicity that $\hat{x}(0) = 0$ and $l = 1$, and using measurement equation (3.31), gives

$$\begin{aligned} \hat{x}(t) &= \int_0^t e^{-(t-\tau)} \left(x + \cos\omega\tau\right) d\tau \\ &= x\, e^{-t} \int_0^t e^{\tau} d\tau + e^{-t} \int_0^t e^{\tau} \cos\omega\tau d\tau \qquad (3.35) \\ &= x\, e^{-t} \left[e^{\tau}\right]_0^t + e^{-t} \left[e^{\tau} \frac{\omega \sin\omega\tau + \cos\omega\tau}{\omega^2 + 1} \right]_0^t \qquad (3.36) \end{aligned}$$

where the second part of the right handside term is obtained after applying integration by parts twice.

After a transient and for t sufficiently big, we get

$$\hat{x}(t) \approx x + \frac{\omega \sin\omega t + \cos\omega t}{\omega^2 + 1} \tag{3.37}$$

That ω is relatively big, in this context, means that we have $\omega \gg 1$ so that the approximation

$$\frac{\omega \sin\omega t + \cos\omega t}{\omega^2 + 1} \approx \frac{1}{\omega} \sin\omega t \tag{3.38}$$

finally leads to

$$\hat{x}(t) \approx x + \frac{1}{\omega} \sin\omega t. \tag{3.39}$$

Hence, comparing $y(t)$ in (3.31) with $\hat{x}(t)$ in (3.39), we see that $\hat{x}(t)$ is just a filtered version of $y(t)$ where the sinusoidal noise was attenuated of a factor of ω. □

The above reasoning can obviously be applied to more complex examples with very similar results. As a sidenote, it is important to remark that models that are nearly as simple as equation (3.30) are quite often used in navigation, where constant acceleration or velocity models replace constant position model (3.30).

3.4 Observer tuning

In one of our computer sessions, we have seen that we could use the so-called Matlab command `place` in order to tune the linear state-feedback gain $\mathbf{K}$. Using the matrices $\mathbf{A}$ and $\mathbf{B}$, we would type

$$\mathbf{K} = \texttt{place}(\mathbf{A}, \mathbf{B}, \mathbf{P}) \tag{3.40}$$

where $\mathbf{P}$ was a vector containing the desired poles of the system in closed-loop. The `place` function computes the values of $\mathbf{K}$ such that

$$\mathbf{P} = \mathrm{eig}(\mathbf{A} - \mathbf{BK}), \tag{3.41}$$

ie *it solves an algebraic equation*[3].
As it turns out, we can use the same reasoning to tune the feedback gain $\mathbf{L}$. To see this, recall that, in the observer design context, the pole placement method means that we want to set the values of $\mathbf{L}$ such that the estimation error dynamics

$$\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{LC})\tilde{\mathbf{x}}(t) \tag{3.42}$$

will have its eigenvalues equal to the desired poles, ie we want

$$\mathbf{P}_{obs} = \mathrm{eig}(\mathbf{A} - \mathbf{LC}) \tag{3.43}$$

with $\mathbf{P}_{obs}$ the vector containing the desired poles of the estimation error dynamics. Hence, we also have to solve an algebraic equation, with this time observer gain $\mathbf{L}$ as the unknown. To do that using the `place` function from Matlab, we first transpose $\mathbf{A} - \mathbf{LC}$ to get

$$(\mathbf{A} - \mathbf{LC})^T = \mathbf{A}^T - \mathbf{C}^T\mathbf{L}^T \tag{3.44}$$

Hence, since transposition does not change eigenvalues, we can rewrite (3.43) as

$$\mathbf{P}_{obs} = \mathrm{eig}(\mathbf{A}^T - \mathbf{C}^T\mathbf{L}^T) \tag{3.45}$$

Comparing (3.45) with (3.41), we see now that the unknown $\mathbf{L}^T$ is at the same spot in equation (3.45) (that needs to be solved) as $\mathbf{K}$ is in (3.41). Hence, using the `place` function, we can write

$$\mathbf{L} = \left(\mathtt{place}(\mathbf{A}^T, \mathbf{C}^T, \mathbf{P})\right)^T. \tag{3.46}$$

3.5 The Kalman Filter as an observer

Since its inception in the 1960s, the so-called Kalman filter has seen many applications, from economics to inertial navigation systems.
In the way is was initially presented, the description of the Kalman filter usually needs a stochastic framework. While this would take us much further than intended in this course, and sometimes needs an entire course in and of itself to see the many interesting theoretical and practical ideas linked with Kalman filtering, it is also possible to quickly put together/hack a Kalman filter (KF) by taking a deterministic view and consider a KF as an observer.

As we have seen, designing an observer basically consists in 3 main things:

- make a copy of the plant model,
- add a correction term based on the measurement $\mathbf{L}(y(t) - \hat{y}(t))$,
- tune the gain vector $\mathbf{L}$ such that the estimation error has the desired dynamics.

There are different ways to tune $\mathbf{L}$, and we saw one of them in the previous section (pole placement/eigenvalue assignment using the `place` command).

[3] Indeed, the unknown $\mathbf{K}$ should be a solution of (algebraic) equation (3.41).

In 1961, Kalman published 2 seminal papers[4] presenting what can be seen as (although there is of course a lot more to it) a method to tune the gain $\mathbf{L}$ for *linear time-varying systems* of the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) \end{cases}. \tag{3.47}$$

for which the observer structure is also a copy of the plant with corrections from the measurements:

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{L}(t)\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right) \\ \hat{\mathbf{y}}(t) = \mathbf{C}(t)\hat{\mathbf{x}}(t) \end{cases} \tag{3.48}$$

Once the copy of the plant is combined with the feedback term $\mathbf{L}(t)\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right)$, the remaining question is obviously how to tune the gain vector $\mathbf{L}(t)$.
To do so, Kalman used a combination of two equations. The first one (or the second one, depending on the order in which these equations are presented) gives the observer gain vector and reads

$$\mathbf{L}(t) = \mathbf{P}(t)\mathbf{C}^T(t)\mathbf{R}^{-1}(t), \tag{3.49}$$

where, for the deterministic interpretation, $\mathbf{R}(t)$ is a positive definite matrix of design parameters (possible time-varying). The $n \times n$ matrix $\mathbf{P}(t)$ is then obtained as a solution of a matrix differential equation expressed as

$$\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T(t) - \mathbf{P}(t)\mathbf{C}^T(t)\mathbf{R}^{-1}(t)\mathbf{C}(t)\mathbf{P}(t) + \mathbf{Q}(t) \tag{3.50}$$

where $\mathbf{P}(t)$ has the initial condition $\mathbf{P}(0) = \mathbf{P}^T(0) > 0$, and where $\mathbf{Q}(t)$ is another positive definite matrix of design parameters. Expression (3.50) is called the Riccati equation and is quite famous in control systems (we have seen a similar Riccati equation in LQR control). Together with the gain expression (3.49), they result in a choice of $\mathbf{L}(t)$ that will, among other things, make sure that the observer estimate $\hat{\mathbf{x}}(t)$ converges to the actual state $\mathbf{x}(t)$.
Equations (3.48), (3.49) and (3.50) are known as the *Kalman filter*.

One of the fundamental results related to the Kalman filter is that its estimate $\hat{\mathbf{x}}(t)$ is the best estimate one can obtain under the assumption of Gaussian noise affecting both the state equation and the output equation. The Kalman filter is said to be *an optimal estimator*.

3.5.1 The Extended Kalman Filter (EKF)

After its inception, the Kalman filter was extended in many directions, and versions/adaptations to different contexts can be found in the literature. One of them, referred to as the *Extended Kalman Filter (EKF)* applies to nonlinear systems described by the dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.51}$$

together with the output equation

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)). \tag{3.52}$$

[4]One in a discrete-time framework, and the other in continuous-time, co-written with Bucy.

For systems put under form (3.51)-(3.52), the observer part of the EKF is given by

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) + \mathbf{L}(t)\,(\mathbf{y}(t) - \mathbf{h}(\hat{\mathbf{x}}(t))) \tag{3.53}$$

where feedback term $\mathbf{L}(t)$ is again calculated from expressions (3.49) and (3.50) but where matrices $\mathbf{A}(t)$ and $\mathbf{C}(t)$ are given by the Jacobian matrices

$$\mathbf{A}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t))\right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)} \tag{3.54}$$

and

$$\mathbf{C}(t) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}(t))\right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)} \tag{3.55}$$

The Extended Kalman Filter found its way in many applications where the system is nonlinear, from induction machines to underwater navigation. However, despite its usefulness, it is also known to be not always easy to tune, and does not have a global optimal property, compared to its linear ancestor.