B15 Kalman Filtering

Michaelmas Term 2021
University of Oxford

Stephen Duncan
stephen.duncan@eng.ox.ac.uk

Syllabus

These four lectures follow on from the 8 lectures on Linear Dynamic Systems and Optimal Control

Including noise in state space models. Kalman filter in continuous-time. Duality between Kalman filter and Linear Quadratic Regualtor (LQR). Linear Quadratic Gaussian (LQG) control. Loop transfer recovery. Discrete-time state space models. Kalman filter in discrete-time. Implementing the Kalman filter

Learning outcomes:

- Understand how to include process and measurement noise in state space models.
- Derive the Kalman filter as an optimal linear observer for state space systems with noise.
- Recognise the duality between the Kalman filter and the Linear Quadratic Regulator (LQR).
- Understand how a Linear Quadratic Guassian (LQG) controller can be designed by combining a Kalman filter with a Linear Quadratic (LQR) regulator.
- Understand limitations of LQG controller and appreciate how loop transfer recovery can be used to overcome some of these.

- Know how to include integral action in a state space controller.
- Know how to create a discrete-time state space model from a continuous-time model of a system that is operated in sample and hold mode.
- Be able to design a Kalman filter for a discrete-time state space model.
- Appreciate factors to consider when implementing a discrete-time Kalman filter.
- Recognise that an Extended Kalman filter (EKF) can be used to estimate the states of nonlinear state space models.

Lecture notes

Recommended text

- K Astrom & R Murray *Feedback Systems: An Introduction for Scientists and Engineers* Princeton U.P., 2008.
- A Gelb (ed.) *Applied Optimal Estimation* MIT Press, 1974.

Other reading

- T Kailath, A H Sayed & B Hassibi *Linear Estimation* Prentice Hall, 1999.

These lecture notes draw on the notes previously written by Profs Kouvaritakis, Bacic and Reid. However, any mistakes/typos in the notes are down to me; please send any comments or corrections to `stephen.duncan@eng.ox.ac.uk`.

A comment on notation. There is considerable overlap between the Linear Quadratic Regulator (LQR) developed in Lecture 8 and the Kalman filter developed in these lectures - in fact we will show that they are duals of each other. As a result, it is common to use the symbols P, Q and R for different variables in the two approaches, which can cause confusion. Some authors use different symbols for the LQR and the Kalman filter, but we will stick with the more standard notation and use P_{c} for the variable associated with the controller and P_{o} for the variable

associated with the observer when we need both variables in the same analysis (for example when developing the Linear Quadratic Guassian (LQG) controller in Lecture 9).

Contents

9 Kalman Filtering **6**

9.1 Motivation 6

9.2 Review of linear observers 7

9.3 Noise Models 9

9.4 Properties of Noises 11

9.5 Kalman Filter 16

9.6 Duality between Kalman filter and Linear Quadratic Regulator . . . 19

9.7 Infinite horizon Kalman filter 20

9.8 Appendix 23

10 Linear Quadratic Gaussian (LQG) control **25**

10.1 Closed loop response of LQG controller 29

10.2 Loop transfer recovery 32

10.3 Including integral action 33

11 Discrete-time systems **36**

11.1 Discrete-time state space models 36

11.2 Discrete Kalman filter 40

11.3 Combined state and measurement update 46

11.4 Forms of Kalman filter 47

12 Implementing the Kalman filter **48**

12.1 Initialisation 50

12.2 Assessing performance 51

12.3 Testing whiteness of residuals 53
12.4 Choosing Q and R 55
12.5 Outliers in measurement data 56
12.6 Detecting faults and modelling errors 56
12.7 Time-varying systems 58
12.8 Coloured noise 59
12.9 Numerical issues 60
12.10Nonlinear systems 60

9 Kalman Filtering

9.1 Motivation

A Kalman filter is an observer that estimates the states of a system when the system is subject to noise. This observer is *optimal* in the sense that it minimises the covariance of the estimation error. For example, consider the mass, spring, damper system in Figure 1. Then the motion of the mass (relative to steady state) is described by

$$m\ddot{x}(t) = -kx(t) - c\dot{x}(t) + w(t)$$

where $x(t)$ is the position of the mass and $w(t)$ is an unknown force applied to the mass. The system can be described by the state space model

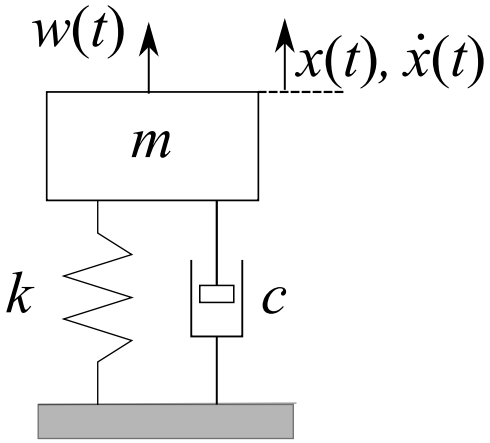

Figure 1: Mass spring damper system.

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m}w(t) \end{bmatrix} \qquad \begin{bmatrix} x(0) \\ \dot{x}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} \tag{9.1}$$

The measurement from the system $y(t)$ is the displacement of the mass, but this measurement is corupted by sensor noise.

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + v(t)$$

where $v(t)$ is the measurement noise. The Kalman filter can provide

- an estimate of $x(t)$, the state that is measured, despite the effect of the sensor noise;

- an estimate of $\dot{x}(t)$, the unmeasured state.

These estimates can be obtained, even when the initial state and the driving force are unknown.

9.2 Review of linear observers

Observers use the measurements of the outputs from a state space system to provide an estimate the states. You have already been introduced to the idea of a *linear state observer* for the case where there are no unknown inputs (e.g.noise) in Lecture 7 of the course on Linear Dynamic Systems and Optimal Control. Assume that we have a Linear Time-invariant (LTI) system that describes the evolution of the state $x(t) \in \mathbb{R}^n$, with m inputs $u(t) \in \mathbb{R}^m$ and p outputs (measurements) $y(t) \in \mathbb{R}^p$

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) & x(0) &= x_0 \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. Given $x(0)$, the state at time $t = 0$, together with the current and future inputs $u(t)$ for $t \geq 0$, then these equations describe the evolution of the states, and hence the outputs, for all future times.

If we want to use state feedback of the form

$$u(t) = Kx(t) + r(t)$$

where $K \in \mathbb{R}^{m\times n}$ is the *feedback gain matrix* and $r(t) \in \mathbb{R}^n$ is an external input vector, which can be considered as the reference signal. To implement this feedback, we require the state $x(t)$, but the problem is that in practice, we do not know the initial state, $x(0)$, so we cannot predict $x(t)$ for future times. In addition, we only have access to the measurements, $y(t)$, rather than the states (unless we have the simple case where $C = I$, so that all states are measured). As we showed, rather than using the state itself in the state feedback, we can use an estimate of the state, $\hat{x}(t) \in \mathbb{R}^n$, which is generated from a (linear) observer

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) & \hat{x}(0) &= \hat{x}_0 \\ \hat{y}(t) &= C\hat{x}(t) + Du(t). \end{aligned} \tag{9.2}$$

where $L \in \mathbb{R}^{n \times m}$ is the observer gain matrix and $\hat{y}(t) = C\hat{x}(t) + Du(t)$. Given $\hat{x}(0)$, an estimate of the state at $t = 0$, together with the measurements $y(t)$ for $t \geq 0$, then the observer provides an estimate of the state for all future times. In addition, if we define the estimation error as $e(t) = x(t) - \hat{x}(t)$ as the difference between the state and the estimate, then we showed that

$$\begin{aligned}
\dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\
&= Ax(t) + \cancel{Bu(t)} - A\hat{x}(t) - \cancel{Bu(t)} - L(y(t) - \hat{y}(t)) \\
&= A(x(t) - \hat{x}(t)) - L(Cx(t) + \cancel{Du(t)} - C\hat{x}(t) - \cancel{Du(t)}) \\
&= (A - LC)(x(t) - \hat{x}(t)) \\
&= (A - LC)e(t).
\end{aligned}$$

This means that provided that the system is *observable*, if we choose the observer gain matrix L so that the eigenvalues of $A - LC$ have negative real parts, the evolution of the estimation error will be asymptotically stable, so that the error will converge to zero and the estimated state $\hat{x}(t)$ will converge to the actual state $x(t)$ for any initial estimate of the state $\hat{x}(0)$. Even if the system has unobservable states, then provided that the system is *detectable*, so that the unobservable states are stable, the estimation error associated with the observer will still asymptotically converge to zero, but the state estimate will not be unique. The estimated state can then be used in the state feedback law, so that

$$u(t) = K\hat{x}(t) + r(t).$$

Linear observers of this form are referred to as *Luenberger* observers. The key challenge in designing a Luenberger observer is choosing the gain matrix L. We have already seen (Lecture 7) how pole placement methods, which were introduced for designing the controller gain K in the state feedback law, can also be used to determine L such that the eigenvalues of $A - LC$ are placed at a specific location in the complex plane and by ensuring that the eigenvalues have negative real parts, the observer will be stable. (In Lecture 7, the pole placement method was applied to a system with a single output, but the approach can be generalised to handle multiple outputs). There are two main limitations of this approach:

- Although the pole placement method will result in the design of an observer that can ensure that the evolution of the error is asymptotically stable, it is

not clear where the eigenvalues of $A - LC$ should be placed for the "best" response. The same problem was encountered when determining K for the state feedback law and we showed that the linear quadratic regulator (LQR) was the optimal feedback law for minimising a quadratic cost function. In a similar manner, we will show how to determine the observer gain matrix L that minimises a specific cost that is a measure of the "size" of the estimation error.

- The development of the linear observer was based on the state space model in (9.4) that assumes perfect knowledge of the evolution of the state. In reality, all systems are subject to disturbances that will affect the state, which will change the evolution of $x(t)$. Also, measurements will be taken by imperfect sensors, so there will be noise that affects these measurements. We want to design an observer that minimises the effects of these disturbances on the estimation error and hence, on the accuracy of the estimate of the state.

The Kalman filter is an observer that addresses both of these issues.

9.3 Noise Models

Before starting the derivation of the Kalman filter, we need to amend the model to include the disturbances or noises that affect the states and the measurements. We will use the state space model

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t) \qquad & x(0) &= x_0 \\ y(t) &= Cx(t) + Du(t) + v(t), & & \end{aligned} \tag{9.3}$$

where $w(t) \in \mathbb{R}^n$ is the *process* or *state* noise that affects the states, while $v(t) \in \mathbb{R}^p$ is the *measurement* noise. It is important to appreciate the following points about the process and measurement noise.

- Because the state equation is a differential equation that describes the *evolution* of the state, if $w(t')$ is the process noise that enters the system at $t = t'$, then this disturbance has an effect on the state $x(t)$ for all $t \geq t'$ (although if the system is stable, the magnitude of this effect will decay over time).

Because the process noise affects the state, its effect will also be seen in the measurement $y(t)$ for $t \geq t'$.

- By contrast, if the measurement is affected by noise $v(t')$ at $t = t'$, then this only has an effect on $y(t')$ at $t = t'$ and does not affect future measurements.
- It is important to recognise the difference between the input signal $u(t)$ and the process and measurement noises, $w(t)$ and $v(t)$. At any time t, the input $u(t)$ is a *known* signal, either because it is decided beforehand (in the case of feed-forward control) or because it is determined by the controller, for example using state feedback. By contrast, $w(t)$ and $v(t)$ are *unknown*. If the noise signals were known, then we could incorporate them into the state space model as an additional input signal.
- Because the disturbances are unknown, we model $w(t)$ and $v(t)$ as *random* (or *stochastic*) processes.
- The aim of a control system is to steer the system so that the state $x(t)$ matches a desired state. If process noise $w(t)$, enters the system, then this will move $x(t)$ away from this desired state, so we want the controller to respond by adjusting $u(t)$ to bring the system back to the desired state. However, if we are using output control, where we are determining the inputs from the measurements via an observer, then we do not want controller to respond to the measurement noise $v(t)$ as this does not affect the state. Hence, the observer needs to distinguish between $w(t)$, which does affect the state, and $v(t)$, which only affects the measurement.

Example 22. *Consider a state space model in (9.3) with*

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad C = [1 \;\; 0] \qquad D = 0$$

This is the same model as used in Examples 14 and 16. It is also the same as the model of the mass spring damper system in (9.1) when $\frac{k}{m} = 1$ and $\frac{c}{m} = 1$ with the inclusion of a known force $u(t)$ in addition to the unknown force $w(t)$. The Simulink model for this system is shown in Figure 2. This is a second-order system and Figure 3 shows the response when a step change

in the input is applied at $t = 1$ for the case where there is no noise and it can be seen that the system is underdamped.

Figure 4 shows the output when the system is subjected to process noise only (i.e. with $u(t) = 0$ and $v(t) = 0$). Because the process noise enters the equation that describes the evolution of the state, the noise is filtered by the system dynamics, which are "low-pass", so the response at the output is dominated by low frequency variations. By contrast, when measurement noise is included in addition to the process noise, the measurement noise enters directly into the output equation without any filtering, so the effect is to add broadband noise, including high frequency components, to the effect of the process noise, as shown in Figure 5.

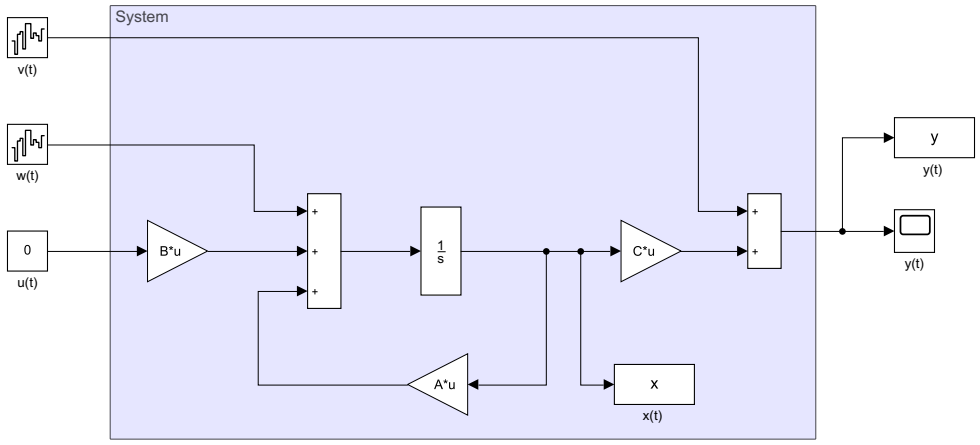

Figure 2: Simulink state space model for Example.

9.4 Properties of Noises

Because $w(t)$ and $v(t)$ are unknown, we model them as vector random variables, which is a vector whose elements are individual random variables. We will assume that the random process is stationary (see A1 course on Time-frequency Analysis for the definition of stationarity for random process). As in the scalar case, the mean of the process noise $\mu_w(t) \in \mathbb{R}^n$ is also a vector, whose elements are the

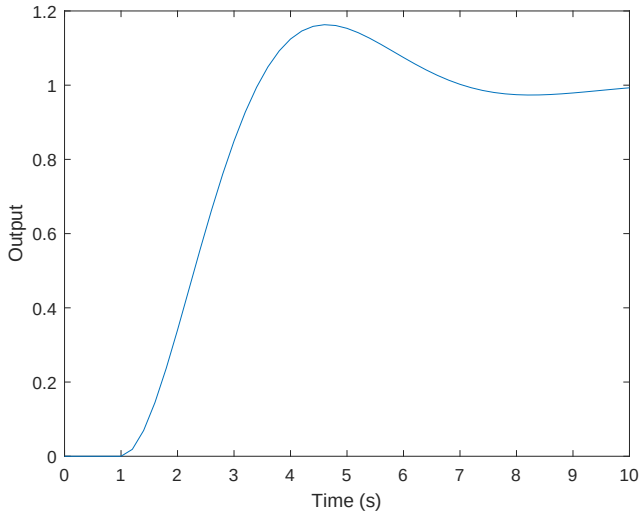

Figure 3: Open loop step response of model with no noise.

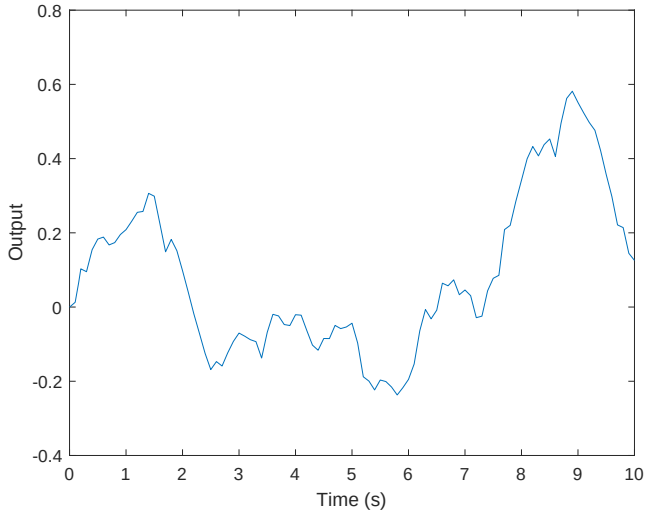

Figure 4: Open loop response of model with process noise, but no input and no measurement noise.

expected values of each of the individual random processes

$$\mu_w = E\left[w(t)\right] = E\left\{\begin{bmatrix} w_1(t) \\ \dots \\ w_n(t) \end{bmatrix}\right\} = \begin{bmatrix} E\left[w_1(t)\right] \\ \dots \\ E\left[w_n(t)\right] \end{bmatrix}$$

where

$$E\left[w_i(t)\right] = \int_{-\infty}^{\infty} w_i p(w_i) \mathrm{d}w_i \qquad \text{for } i = 1, \dots, n$$

with $p(w_i)$ being the probability density function for each element. Because $w(t)$ is taken to be a stationary process, the probability distributions of the noises do not change over time.

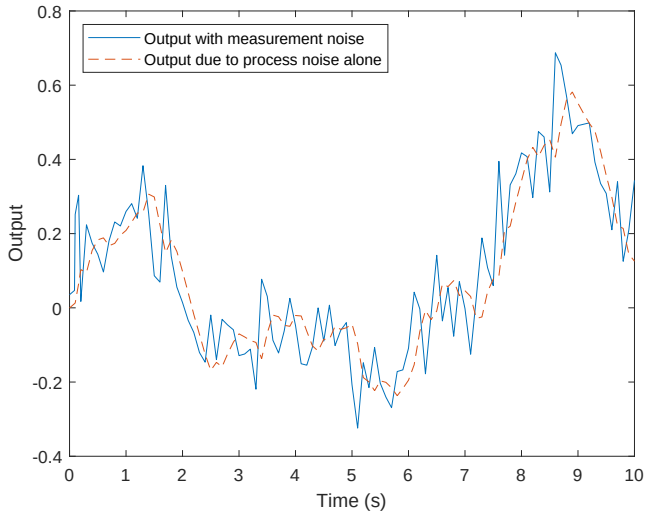

Figure 5: Open loop response of model with process noise and measurement noise, but no input. The effect of the process noise alone is shown by the dashed line.

The autocorrelation of the vector random process $w(t)$ is $R_w(\tau) \in \mathbb{R}^{n \times n}$

$$R_w(\tau) = E\left[w(t)w(t+\tau)^\top\right] = \begin{bmatrix} E\left[w_1(t)w_1(t+\tau)\right] & \dots & E\left[w_1(t)w_n(t+\tau)\right] \\ \vdots & \ddots & \vdots \\ E\left[w_n(t)w_1(t+\tau)\right] & \dots & E\left[w_n(t)w_n(t+\tau)\right] \end{bmatrix}$$

There are similar expressions for the mean $\mu_v \in \mathbb{R}^p$ and autocorrelation $R_v(\tau) \in \mathbb{R}^{p \times p}$ of the measurement noise, $v(t)$.

$$\mu_v = E\left[v(t)\right] = \begin{bmatrix} E\left[v_1(t)\right] \\ \dots \\ E\left[v_p(t)\right] \end{bmatrix}$$

$$R_v(\tau) = E\left[v(t)v(t+\tau)^\top\right] = \begin{bmatrix} E\left[v_1(t)v_1(t+\tau)\right] & \dots & E\left[v_1(t)v_p(t+\tau)\right] \\ \vdots & \ddots & \vdots \\ E\left[v_p(t)v_1(t+\tau)\right] & \dots & E\left[v_p(t)v_p(t+\tau)\right] \end{bmatrix}$$

Because we have assumed that each of the random variables in the vector are *stationary*, their means do not change over time and their autocorrelation functions depend only on the time delay τ (see A1 course on Time-frequency Analysis).

In addition to the process and measurement noise being stationary, we will also assume that they are both zero-mean, Gaussian, white noise processes. Consider each of these properties separately

Gaussian process Each element of random vector process has a Gaussian (Normal) probability density function, so that

$$p(w) = \frac{1}{\sqrt{(2\pi)^n \det(Q)}} \mathrm{e}^{-(w-\mu_w)^\top Q^{-1}(w-\mu_w)}$$

This Gaussian process is characterised by the (vector) mean $\mu_w \in \mathbb{R}^n$ and the correlation matrix $Q \in \mathbb{R}^{n \times n}$

$$Q = E\left[w(t)w(t)^\top\right]$$

Zero-mean The means of both processes are zero, so that $\mu_w = 0$ and $\mu_v = 0$. It is possible to handle noise processes where the means are known, but not zero, either by adding the mean as a (constant) extra input or by applying an offset to the state.

White noise The state noise is white if

$$E[w(t)w(t+\tau)^\top] = Q\delta(\tau)$$

where $Q \in \mathbb{R}^{n \times n}$ is the *correlation* matrix and $\delta(.)$ is the Dirac delta function. This means that $w(t)$ is uncorrelated with $w(t+\tau)$, except when $\tau = 0$, so that the disturbance at time t is independent of both past and future values. When $\tau = 0$, the matrix Q gives the correlation between the individual elements of $w(t)$, which describes the correlation between the noise entering the states. Similarly the noise affecting the measurements will be white if

$$E[v(t)v(t+\tau)^\top] = R\delta(\tau)$$

where $R \in \mathbb{R}^{p \times p}$ is the correlation matrix of the measurement noise. It is possible to handle coloured (i.e. non-white) noise, by considering this as filtered white noise, which is modelled by augmenting the state space model to include the noise filter (see Lecture 12)

The covariance matrix of the state noise is given by

$$E\left[\left(w(t) - \mu_w)\right)\left(w(t) - \mu_w)\right)^\top\right]$$

but because the state noise is assumed to have zero mean, $\mu_w = 0$ and the covariance matrix reduces to $E\left[w(t)w(t)^\top\right]$, which is the correlation matrix. This is also

true for the measurement noise. As a result, Q and R are often referred to as the covariance matrices.

In addition, we will also assume

$$E\left[w(t)v(t+\tau)^\top\right] = 0 \quad \text{for all } \tau$$

so that there is no cross-correlation between the process noise $w(t)$ and the measurement noise $v(t)$. Since the sources of the two noises are usually different, it is reasonable to assume that they are independent. However, the design of the Kalman filter can be modified to handle the case where there is correlation between process and measurement noise, although it does make the analysis more complicated.

Note that because

$$Q = E\left[w(t)w(t)^\top\right]$$

then Q must be symmetric. We will also assume that $Q \geq 0$, so it is positive semi-definite (see Appendix of Linear Systems and Optimal Control notes for definition of semi-definite matrices). This reflects the possibility that the process noise could take the form

$$w(t) = \begin{bmatrix} 0 \\ w_2(t) \end{bmatrix}$$

(this is the form of the noise in the mass spring damper model in (9.3)). Under these circumstances,

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & E\left[w_2(t)w_2(t)^\top\right] \end{bmatrix}$$

which is positive semi-definite.

Since

$$R = E\left[v(t)v(t)^\top\right]$$

then R is also symmetric, but in the derivation of the Kalman filter, we will assume that $R > 0$, so that it is positive definite.

9.5 Kalman Filter

The state space model is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t) \qquad\qquad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) + v(t), \end{aligned}$$

with an initial *unknown* state $x(0) = x_0$. $w(t) \in \mathbb{R}^n$ and $v(t) \in \mathbb{R}^p$ are zero-mean, Gaussian, white noise processes with

$$\begin{aligned} E\left[w(t)w(t+\tau)^\top\right] &= Q\delta(\tau) \\ E\left[v(t)v(t+\tau)^\top\right] &= R\delta(\tau) \\ E\left[w(t)v(t+\tau)^\top\right] &= 0 \end{aligned}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{p\times p}$.

The Kalman filter for this state space system is

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(t)(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t). \end{aligned} \tag{9.4}$$

where $L(t) \in \mathbb{R}^{n\times p}$ is the observer gain matrix and $\hat{y}(t) = C\hat{x}(t) + Du(t)$. Note that although the this observer is linear, because we are allowing $L(t)$ to be time varying, the observer is a time varying state space system. Since $w(t)$ and $v(t)$ are random variables, the states $x(t)$ and the measurement $y(t)$ are random variables. The evolution of $\hat{x}(t)$ depends upon $y(t)$, so $\hat{x}(t)$ is also a random variables. In addition $x(0)$, the initial state of the system, is unknown so the observer is initialised with $\hat{x}(0) = E\left[x(0)\right]$. For a given initial condition, $\hat{x}(0)$, then $\hat{x}(T)$ state estimate at time $t = T$, is determined by $y(t)$ and $u(t)$ over the time period $0 \leq t \leq T$, which are the *past* measurements and inputs.

If we define the state estimation error $e(t) = x(t) - \hat{x}(t)$, then

$$\begin{aligned}
\dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\
&= Ax(t) + \cancel{Bu(t)} + w(t) - A\hat{x}(t) - \cancel{Bu(t)} - L(t)(y(t) - \hat{y}(t)) \\
&= A(x(t) - \hat{x}(t)) + w(t) - L(t)(Cx(t) + \cancel{Du(t)} + v(t) - C\hat{x}(t) - \cancel{Du(t)}) \\
&= A(x(t) - \hat{x}(t)) + w(t) - L(t)C(x(t) - \hat{x}(t)) - L(t)v(t) \qquad (9.5) \\
&= (A - L(t)C)(x(t) - \hat{x}(t)) + w(t) - Lv(t) \\
&= (A - L(t)C)e(t) + w(t) - L(t)v(t) \\
&= \tilde{A}(t)e(t) + w(t) - L(t)v(t) \qquad (9.6)
\end{aligned}$$

where $\tilde{A}(t) = A - L(t)C$. This has a similar form to the expression for the evolution of the error for the state space model without state and measurement noise, apart from the inclusion of $w(t) - L(t)v(t)$. Denote

$$\rho(t) = w(t) - L(t)v(t)$$

Because this term is a random variable, this means that $e(t)$ is also random with

$$E\left[\rho(t)\right] = \cancel{E\left[w(t)\right]} - L(t)\cancel{E\left[v(t)\right]} = 0$$

because both $w(t)$ and $v(t)$ are assumed to have zero-mean. The autocorrelation of $\rho(t)$ is

$$\begin{aligned}
E\left[\rho(t)\rho(t+\tau)^\top\right] &= E\left[\left(w(t) - L(t)v(t)\right)\left(w(t+\tau) - L(t+\tau)v(t+\tau)\right)^\top\right] \\
&= E\left[w(t)w(t+\tau)^\top\right] - L(t)\cancel{E\left[v(t)w(t+\tau)^\top\right]} \\
&\quad - \cancel{E\left[w(t)v(t+\tau)^\top\right]}L(t+\tau)^\top + L(t)E\left[v(t)v(t+\tau)^\top\right]L(t+\tau)^\top \\
&= Q\delta(\tau) + L(t)RL(t+\tau)^\top\delta(\tau) \\
&= Q_\rho(t)\delta(\tau)
\end{aligned}$$

where $Q_\rho(t) = Q + L(t)RL(t)^\top$

The Kalman filter is an observer which uses the gain matrix $L(t)$ that minimises $E[e(t)^\top e(t)]$, the variance of the estimation error. In order to find the optimal $L(t)$, denote $P(t) = E\left[e(t)e(t)^\top\right]$, which is the correlation of the estimation error. Note that the trace of a matrix is the sum of its diagonal elements, so that

$$E\left[e(t)^\top e(t)\right] = \sum_{i=1}^{n} E\left[e_i(t)\right]^2 = \mathrm{tr}\left(P(t)\right)$$

where $\mathrm{tr}(.)$ denotes the trace operator, which means that the variance of the estimation error is the trace of the correlation matrix.

By differentiating $P(t) = E[e(t)e(t)^\top]$ it can be shown (see Appendix) that

$$\dot{P}(t) = \tilde{A}(t)P(t) + P(t)\tilde{A}(t)^\top + Q + L(t)RL(t)^\top \tag{9.7}$$

where the final two terms on the right hand side are $E\left[\rho(t)\rho(t)^\top\right]$. Substituting $\tilde{A}(t) = A - L(t)C$ gives

$$\begin{aligned}
\dot{P}(t) &= \tilde{A}(t)P(t) + P(t)\tilde{A}(t)^\top + Q + L(t)RL(t)^\top \\
&= (A - L(t)C)\,P(t) + P(t)\,(A - L(t)C)^\top + Q + L(t)RL(t)^\top \\
&= AP(t) + P(t)A^\top + Q - L(t)CP(t) - P(t)C^\top L(t)^\top + L(t)RL(t)^\top \\
&= AP(t) + P(t)A^\top + Q + \left(L(t)R - P(t)C^\top\right)R^{-1}\left(L(t)R - P(t)C^\top\right)^\top \\
&\qquad - P(t)C^\top R^{-1}CP(t)
\end{aligned} \tag{9.8}$$

where the last line is obtained by completing the square (to check this, multiply out the bracketed terms in the last line and show that this equals the expression in the third line). We have the assumed that because $R > 0$, then R is invertible and we use the fact that R is symmetric, so that $R = R^\top$.

For the optimal observer, we want to choose the time-varying observer gain $L(t)$ that minimises $E[e(t)^\top e(t)] = \mathrm{tr}(P)$, where $P(t)$ satisfies the (matrix) differential equation in (9.8). We need to find $L(t)$ such that $P(t)$ is as small as possible, which can be done by choosing $L(t)$ so that $\dot{P}(t)$ decreases by the maximum amount possible at each instant in time. This is accomplished by setting

$$L(t) = P(t)C^\top R^{-1}$$

so that

$$\left(L(t)R - P(t)C^\top\right)R^{-1}\left(L(t)R - P(t)C^\top\right)^\top = 0$$

Under these circumstances, the expression for $\dot{P}(t)$ reduces to

$$\dot{P}(t) = AP(t) + P(t)A^\top + Q - P(t)C^\top R^{-1}CP(t)$$

This (matrix) differential equation is initialised with $P(0) = E[x(0)x(0)^\top]$.

Theorem 8 (Finite horizon Kalman filter)**.** *Consider a state space model*

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + Du(t) + v(t)$$

where $w(t) \in \mathbb{R}^n$ *and* $v(t) \in \mathbb{R}^p$ *are zero-mean, Gaussian, white noise processes, with* $E\left[w(t)w(t+\tau)^\top\right] = Q\,\delta(\tau)$ *and* $E\left[v(t)v(t+\tau)^\top\right] = R\,\delta(\tau)$ *for* $Q = Q^\top \in \mathbb{R}^{n\times n}$ *and* $R = R^\top \in \mathbb{R}^{p\times p}$*, with* R *invertible. The finite horizon Kalman filter is obtained by means of the following steps:*

1. *Solve the Riccati differential equation*

$$\dot{P}(t) = AP(t) + P(t)A^\top + Q - P(t)C^\top R^{-1}CP(t) \qquad P(0) = P_0$$

and denote its (unique) solution by $P(t) \in \mathbb{R}^{n\times n}$*. For any* t*,* $P(t)$ *is symmetric and positive semidefinite.*

2. *The finite horizon optimal Kalman filter can be then constructed as*

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(t)\left(y(t) - \hat{y}(t)\right)$$
$$\hat{y}(t) = C\hat{x}(t) + Du(t)$$

with

$$L(t) = P(t)C^\top R^{-1}$$

where $P(t)$ *is the solution of the Riccati differential equation.*

9.6 Duality between Kalman filter and Linear Quadratic Regulator

In Lecture 8, you were introduced to the Linear Quadratic Regulator (LQR) that determined the state feedback law

$$u(t) = K(t)\hat{x}(t) + r(t)$$

that minimises

$$J = \int_0^\infty x^\top(t)Q_{\mathrm{c}}x(t) + u^\top(t)R_{\mathrm{c}}u(t)\mathrm{dt}$$

$K(t)$ is the feedback gain matrix

$$K(t) = -R^{-1}B^\top P_{\mathrm{c}}(t)$$

where $P_{\mathrm{c}}(t)$ is found by solving the Riccati equation

$$-\dot{P}_{\mathrm{c}}(t) = P_{\mathrm{c}}(t)A + A^{\top}P_{\mathrm{c}}(t) + Q_{\mathrm{c}} - P_{\mathrm{c}}(t)BR_{\mathrm{c}}^{-1}B^{\top}P_{\mathrm{c}}(t) \qquad P_{\mathrm{c}}(T) = Q_{\mathrm{T}}$$

where this equation is solved *backwards* in time from $t = T$ to $t = 0$ with the terminal condition Q_{T}. Note that if $t = -\tau$, then $\frac{\mathrm{d}}{\mathrm{d}\tau} = -\frac{\mathrm{d}}{\mathrm{d}t}$, which is why there is a negative on the left hand side of this equation when solving backwards in time. Also, we are using a subscript on Q_{c}, R_{c} and $P_{\mathrm{c}}(t)$ to distinguish these variables from the corresponding variables in the Kalman filter.

Comparing this Riccati equation with the Riccati equation for the Kalman filter,

$$\dot{P}(t) = AP(t) + P(t)A^{\top} + Q - P(t)C^{\top}R^{-1}CP(t) \qquad P(0) = P_0$$

which is solved *forwards* in time from the initial condition $P(0) = P_0$, we can see that these are *duals* of one another. In other words, since Q, R and $P(t)$ are symmetric (because they are covariance matrices), if we replace A by $A^{\top}$ and B by $C^{\top}$ in the LQR equation, then the right hand sides of the Kalman filter Riccati equation and the LQR Riccati equation are the same.

9.7 Infinite horizon Kalman filter

Because the Riccati equations for the LQR and the Kalman filer are duals of one another, we can apply the convergence results from the LQR to determine when the Riccati equation for the Kalman filter will converge to a steady state solution. Recall from Lecture 8 that the LQR Riccati equation will converge and the resulting LQR controller will be stable if the system is the pairs (A, B) and (A, C) controllable and observable. If the system is not completely controllable or observable, then we require that the system is stabilisable and detectable, so that the uncontrollable and unobservable modes are stable, although under these circumstances, the steady state solution to the Riccati equation will not be unique.

We can apply these results to the Riccati equation for the Kalman filter. Since we are using the Kalman filer to estimate the states, then we need the condition on the observability of (A, C) in order to ensure that the effect of variations in the state can be observed at the output. However, we need to modify the controllability condition because for the LQR, the controllability is expressed in terms of the known

input $u(t)$. However for the Kalman filter, we are interested in the effect of the process noise $w(t)$. We have defined $Q = E\left[w(t)w(t)^\top\right]$, so if we write

$$Q = Q^{\frac{1}{2}}\left(Q^{\frac{1}{2}}\right)^\top$$

where $Q^{\frac{1}{2}}$ is the matrix square root, then we can define

$$\tilde{w}(t) = Q^{\frac{1}{2}}w(t)$$

where $\tilde{w}(t)$ is the "normalised" process noise. For the Riccati equation associated with Kalman filter to converge, we require $(A, Q^{\frac{1}{2}})$ to be controllable. If these conditions hold, then there will be a unique solution to the *algebraic* Riccati equation

$$0 = AP + PA^\top + Q - PC^\top R^{-1}CP$$

and the corresponding Kalman gain will be a constant matrix

$$L = PC^\top R^{-1}$$

If $(A, Q^{\frac{1}{2}})$ is not controllable or (A, C) is not observable, then these conditions can be replaced by the weaker stabilisability and detectability conditions, in which case the Riccati equation will converge to a steady state solution, but it will not be unique.

✎ **Example 23.** *The Simulink model for Kalman filter applied to the second order system in Example 22 is shown in Figure 6 when*

$$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix} \qquad R = 0.001$$

The steady state observer gain for a Kalman filter based on these covariance models is

$$L = \begin{bmatrix} 4.48 \\ 5.05 \end{bmatrix}$$

The estimates for the two states are shown in Figures 7 and 8 and it can be seen that the estimate of $x_1(t)$ is better than the estimate of $x_2(t)$. This is because $C = [1 \quad 0]$, so the output is $x_1(t)$ corrupted by measurement noise (shown by the dashed-dot line in the plot). By contrast, the output does not contain any direct information about $x_2(t)$, so its estimate has to be inferred

from the dynamics of the system.

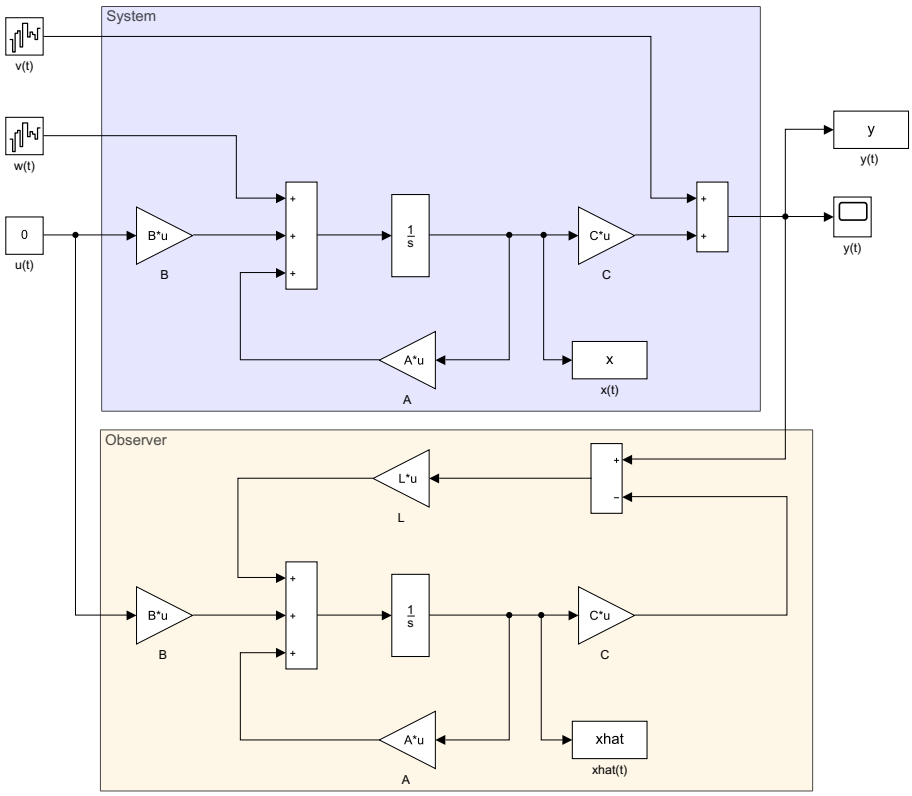

Figure 6: Simulink model of state space model with Kalman filter.

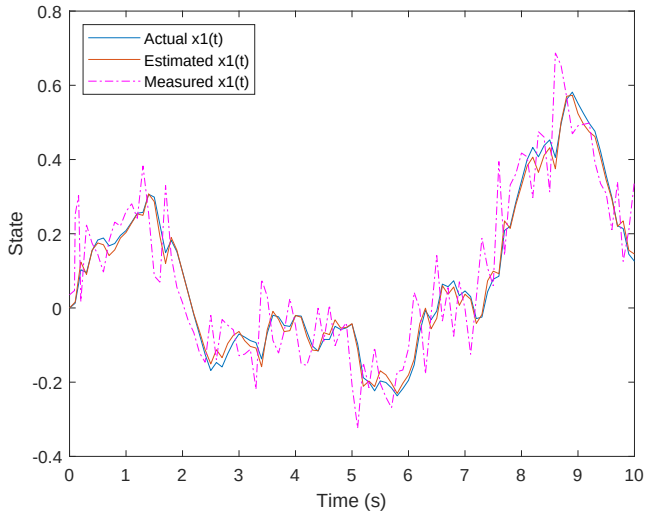

Figure 7: Estimate of $[x(t)]_1$ from Kalman filter (blue) together with actual state (red) and the noisy measurement (dash-dot line).

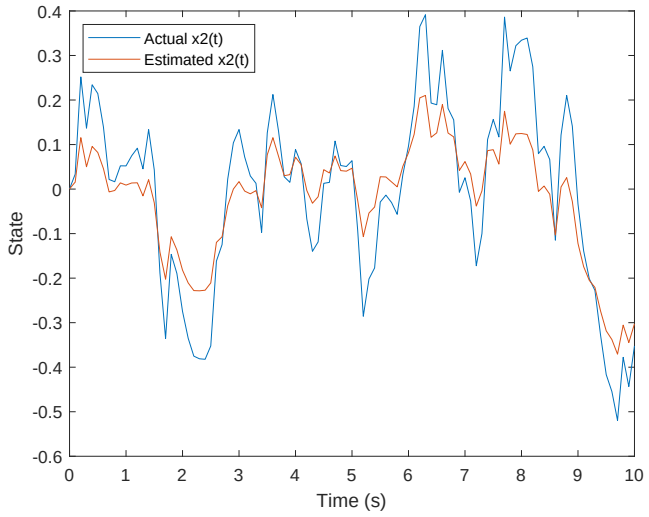

Figure 8: Estimate of $[x(t)]_2$ from Kalman filter (blue) together with actual state (red).

9.8 Appendix

From 9.6

$$\dot{e}(t) = \tilde{A}(t)e(t) + \rho(t)$$

where $\rho(t) = w(t) - L(t)v(t)$, with

$$\begin{aligned} E\left[\rho(t)\rho(\tau)^\top\right] &= Q_\rho \delta(t-\tau) \\ &= \left(Q + L(t)RL(t)^\top\right)\delta(t-\tau) \end{aligned}$$

For $\tau \geq t$, then

$$\begin{aligned} e(t) &= \Phi(t, t_0)e(t_0) + \int_{t_0}^{t} \Phi(t-\xi)\rho(\xi)\mathrm{d}\xi \\ e(\tau) &= \Phi(\tau, t_0)e(t_0) + \int_{t_0}^{\tau} \Phi(\tau-\eta)\rho(\eta)\mathrm{d}\eta \end{aligned}$$

where $\Phi(t, t_0)$ is the state transition matrix associated with $\tilde{A}(t)$ over the interval $t_0 \leq \xi \leq t$. The autocorrelation of $e(t)$ is

$$\begin{aligned}
E\left[e(t)e(\tau)^\top\right] &= \Phi(t,t_0)E\left[e(t_0)e(t_0)^\top\right]\Phi(\tau,t_0)^\top \\
&\quad + \int_{t_0}^{t}\Phi(t,\xi)\cancel{E\left[\rho(\xi)e(t_0)^\top\right]}\mathrm{d}\xi \\
&\quad + \int_{t_0}^{\tau}\cancel{E\left[e(t_0)\rho(\eta)^\top\right]}\Phi(\tau,\eta)^\top\mathrm{d}\eta \\
&\quad + E\left[\left\{\int_{t_0}^{t}\Phi(t,\xi)\rho(\xi)\mathrm{d}\xi\right\}\left\{\int_{t_0}^{\tau}\Phi(\tau,\eta)\rho(\eta)\mathrm{d}\eta\right\}^\top\right] \\
&= \Phi(t,t_0)P(t_0)\Phi(\tau,t_0)^\top \\
&\quad + \int_{t_0}^{t}\left\{\int_{t_0}^{\tau}\Phi(t,\xi)E\left[\rho(\xi)\rho(\eta)^\top\right]\Phi(\tau,\eta)^\top\mathrm{d}\eta\right\}\mathrm{d}\xi \\
&= \Phi(t,t_0)P(t_0)\Phi(\tau,t_0)^\top \\
&\quad + \int_{0}^{t}\Phi(t,\xi)\left\{\int_{t_0}^{\tau}Q_\rho(\xi)\delta(\xi-\eta)\Phi(\tau,\eta)^\top\mathrm{d}\eta\right\}\mathrm{d}\xi \\
&= \Phi(t,t_0)P(0)\Phi(\tau,t_0)^\top + \int_{t_0}^{t}\Phi(t,\xi)Q_\rho(\xi)\Phi(\tau,\xi)^\top\mathrm{d}\xi
\end{aligned}$$

where $E\left[\rho(\xi)e(0)^\top\right] = 0$ and $E\left[e(0)\rho(\eta)^\top\right] = 0$ since $e(0)$ is independent of $\rho(t)$ for all $t > 0$. Differentiating this expression and using Leibniz rule for differentiating integrals (see Lecture 2)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}E\left[e(t)e(\tau)^\top\right] &= \tilde{A}(t)\Phi(t,t_0)P(t_0)\Phi(\tau,t_0)^\top + \Phi(t,t_0)P(t_0)\Phi(\tau,t_0)^\top\tilde{A}(\tau)^\top \\
&\quad + \Phi(t_0,t_0)Q_\rho(t_0)\Phi(t_0,t_0) + \int_{t_0}^{t}\tilde{A}(t)\Phi(t,\xi)Q_\rho(\xi)\Phi(\tau,\xi)^\top\mathrm{d}\xi \\
&\quad + \int_{t_0}^{t}\Phi(t,\xi)Q_\rho(\xi)\Phi(\tau,\xi)^\top\tilde{A}(t)^\top\mathrm{d}\xi
\end{aligned}$$

where we have used $\frac{\mathrm{d}}{\mathrm{d}t}\Phi(t,t_0) = \tilde{A}(t)\Phi(t,t_0)$. In the limits $t \to t_0$ and $\tau \to t_0$, the left hand side becomes $\dot{P}(t_0)$ and the integrals on the right hand side are zero the integration is being performed over an interval of zero. Since $\Phi(t_0,t_0) = I$, then substituting $Q_\rho(t_0) = Q + L(t_0)RL(t_0)^\top$ leads to

$$\dot{P}(t_0) = \tilde{A}(t_0)P(t_0) + \tilde{A}(t_0)P(t_0)^\top + Q + L(t_0)RL(t_0)^\top$$

which has the same form as the expression in (9.7).

10 Linear Quadratic Gaussian (LQG) control

In Lecture 7, we saw that output feedback (rather than state feedback) uses an observer to create an estimate of the state, $\hat{x}(t)$, from current and past measurements $y(t)$ and inputs

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t))$$

where $L \in \mathbb{R}^{n\times p}$ is the gain of an observer, and then using this state estimate in a state feedback law to generate the inputs $u(t)$.

$$u(t) = K\hat{x}(t) + r(t)$$

where $K \in \mathbb{R}^{m\times n}$ is the state feedback matrix. In Lecture 8, we introduced the *infinite* horizon Linear Quadratic Regulator (LQR) for the case where the state is known, which minimises the performance cost

$$J = \int_0^\infty x^\top(t)Q_{\mathrm{c}}x(t) + u^\top(t)R_{\mathrm{c}}u(t)\mathrm{dt} \tag{10.1}$$

by choosing the feedback gain matrix to be

$$K = -R_{\mathrm{c}}^{-1}B^\top P_{\mathrm{c}}$$

where K is the determined from the solution to the algebraic Riccati equation

$$0 = P_{\mathrm{c}}A + A^\top P_{\mathrm{c}} + Q_{\mathrm{c}} - P_{\mathrm{c}}BR_{\mathrm{c}}^{-1}B^\top P_{\mathrm{c}}$$

This derivation was based on a state space model that did not include process and measurement noise, but if these noises are present, then the state space model becomes

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t)\\ y(t) &= Cx(t) + Du(t) + v(t)\end{aligned}$$

where $w(t)$ is the process noise and $v(t)$ is the measurement noise. The Kalman filter minimises

$$E\left[(x(t) - \hat{x}(t))^\top (x(t) - \hat{x}(t))\right]$$

and the (steady state) observer for this system is

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t))$$

where $L \in \mathbb{R}^{n\times p}$ is the steady state Kalman observer gain

$$L = P_{\mathrm{o}} C^{\top} R_{\mathrm{o}}^{-1}$$

with $P_{\mathrm{o}} \in \mathbb{R}^{n\times n}$ being the solution to the algebraic Riccati equation

$$0 = AP_{\mathrm{o}} + P_{\mathrm{o}} A^{\top} + Q_{\mathrm{o}} - P_{\mathrm{o}} C^{\top} R_{\mathrm{o}}^{-1} C P_{\mathrm{o}}$$

Note that we have used the notation P_{c} and P_{o} to distinguish between the solutions of the control algebraic Riccati equation and the observer algebraic Riccati equation.

The combination of a Linear Quadratic Regulator and a Kalman filter is known as Linear Quadratic Gaussian (LQG) control. This is also referred to as $\mathcal{H}_2$ control (see Controller Performance lectures later in B15 course). We will focus on the LQG controller derived from steady state solutions of the Riccati equations that are the solution to the infinite horizon control problem, but the same approach can be applied to the (time-varying) controller derived from finite horizon problem.

Because of the inclusion of noise in the system, the state $x(t)$ is now a stochastic variable, so that the cost in (10.1) is also a stochastic variable. The LQG controller therefore minimises the *expectation* of the cost

$$E\left[\int_0^{\infty} x^{\top}(t) Q_{\mathrm{c}} x(t) + u^{\top}(t) R_{\mathrm{c}} u(t) \mathrm{dt}\right] \tag{10.2}$$

Summary: Linear Quadratic Gausssian Controller

For an output feedback system, given measurements $y(t)$, then the inputs $u(t)$ that minimise

$$E\left[\int_0^{\infty} x^{\top}(t) Q_{\mathrm{c}} x(t) + u^{\top}(t) R_{\mathrm{c}} u(t) \mathrm{dt}\right]$$

are determined by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t) \quad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) + v(t)\end{aligned} \qquad \rightarrow \textit{LTI system}$$

$$u(t) = K\hat{x}(t) + r(t) \qquad \rightarrow \textit{(estimated) state feedback}$$

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad \hat{x}(0) = E[x(0)] \\ \hat{y}(t) &= C\hat{x}(t) + Du(t).\end{aligned} \qquad \rightarrow \textit{state observer}$$

where

$$\begin{aligned}K &= -R_{\mathrm{c}}^{-1}B^{\top}P_{\mathrm{c}} \\ 0 &= P_{\mathrm{c}}A + A^{\top}P_{\mathrm{c}} + Q_{\mathrm{c}} - P_{\mathrm{c}}BR_{\mathrm{c}}^{-1}B^{\top}P_{\mathrm{c}}\end{aligned}$$

and

$$\begin{aligned}L &= P_{\mathrm{o}}C^{\top}R^{-1} \\ 0 &= AP_{\mathrm{o}} + P_{\mathrm{o}}A^{\top} + Q - P_{\mathrm{o}}C^{\top}R^{-1}CP_{\mathrm{o}}\end{aligned}$$

Example 24. *The Simulink model for LQG controller obtained by appending a state feedback law to the Kalman filter in Example 23 is shown in Figure 9. For the infinite horizon controller that minimises the cost*

$$E\left[\int_0^{\infty} x^{\top}(t)Q_{\mathrm{c}}x(t) + u^{\top}(t)R_{\mathrm{c}}u(t)\mathrm{dt}\right]$$

where

$$Q_{\mathrm{c}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_{\mathrm{c}} = 0.5$$

then the steady state optimal state feedback is

$$K = [-0.73 \quad -1.11]$$

The closed loop response of the system to a step change in the reference signal $r(t)$ at $t = 1$ is shown in Figure 10. It can be seen that the closed loop response has a reasonably short rise time with minimal overshoot. The effect

of including process noise in the system is shown in Figure 11 (the process noise used in the plot is the same as used in Figure 4 in Example 22) and the controller still remains close to the target value despite the effect of the noise. Figure 11 shows that the effect of including measurement noise does not have a major effect on the closed loop response.

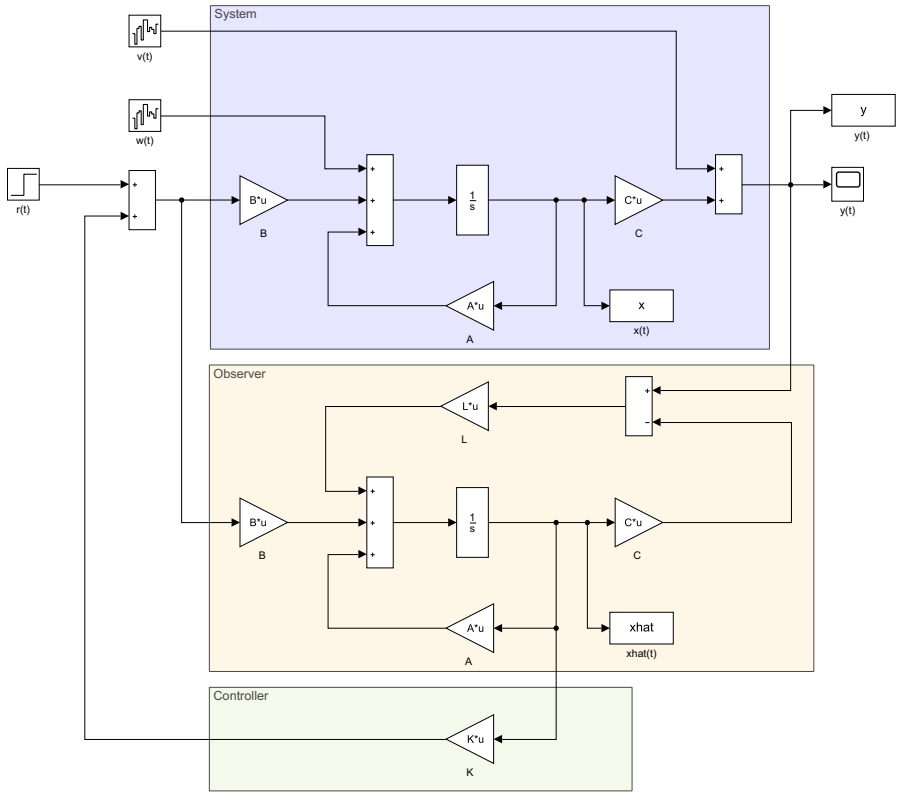

Figure 9: Simulink model of LQG controller.

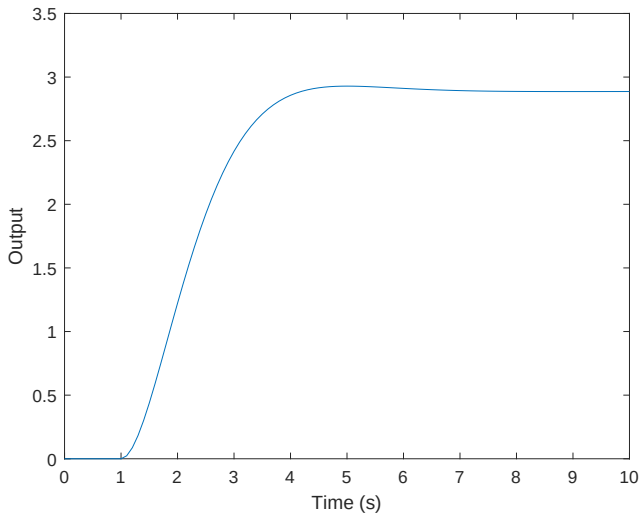

Figure 10: Closed loop step response without any noise.

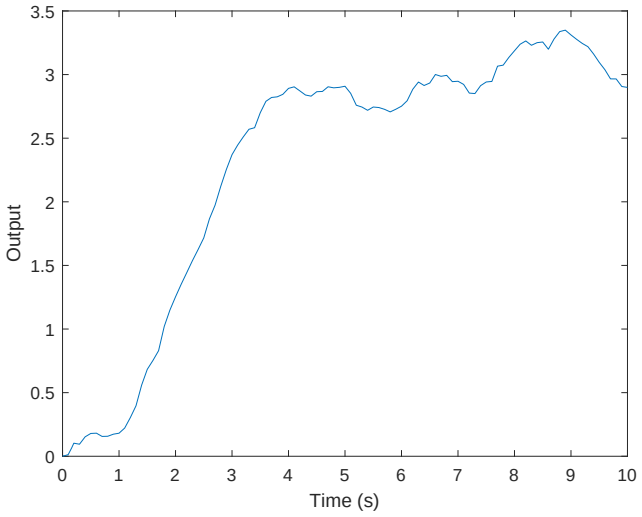

Figure 11: Closed loop step response with process noise only.

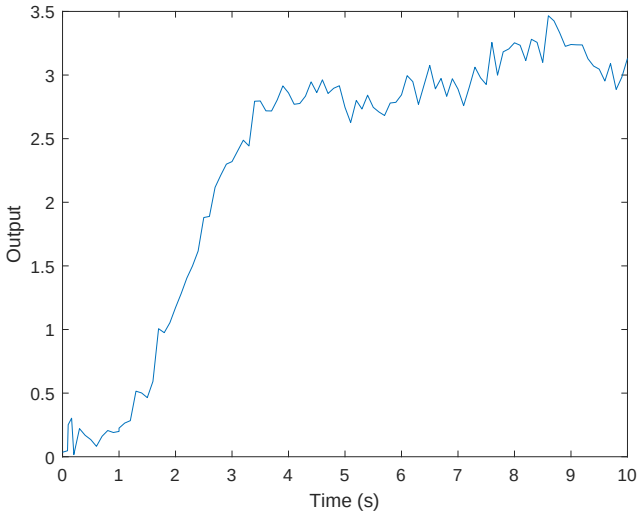

Figure 12: Closed loop step response with both process noise and measurement noise.

10.1 Closed loop response of LQG controller

We have defined the estimation error associated with the Kalman filter as

$$e(t) = x(t) - \hat{x}(t)$$

and we showed previously in (9.6) that

$$\dot{e}(t) = (A - LC)\, e(t) + w(t) - Lv(t)$$

Substituting for the state feedback law in the equation for the evolution of the state and using $u(t) = K\hat{x}(t) + r(t)$ and $\hat{x}(t) = x(t) - e(t)$ gives

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t)\\ &= Ax(t) + BK\hat{x} + Br(t) + w(t)\\ &= Ax(t) + BKx(t) + BKe(t) + Br(t) + w(t)\\ &= (A + BK)\,x(t) + BKe(t) + Br(t) + w(t)\end{aligned}$$

The measurement $y(t)$ is

$$\begin{aligned}y(t) &= Cx(t) + Du(t) + v(t)\\ &= Cx(t) + DK\hat{x}(t) + Dr(t) + v(t)\\ &= Cx(t) + DKx(t) - DKe(t) + Dr(t) + v(t)\\ &= (C + DK)\,x(t) - DKe(t) + Dr(t) + v(t)\end{aligned}$$

Combining these equations leads to

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x(t)\\ e(t)\end{bmatrix} = \begin{bmatrix} (A+BK) & BK\\ 0 & (A-LC)\end{bmatrix}\begin{bmatrix} x(t)\\ e(t)\end{bmatrix} + \begin{bmatrix} B & I & 0\\ 0 & I & -L\end{bmatrix}\begin{bmatrix} r(t)\\ w(t)\\ v(t)\end{bmatrix}$$

$$y(t) = \begin{bmatrix} (C+DK) & -DK\end{bmatrix}\begin{bmatrix} x(t)\\ e(t)\end{bmatrix} + \begin{bmatrix} D & 0 & I\end{bmatrix}\begin{bmatrix} r(t)\\ w(t)\\ v(t)\end{bmatrix}$$

This state space system describes the response of the *closed loop* system from each of the external inputs, $r(t)$, $w(t)$ and $v(t)$, to the output $y(t)$.

The stability of the closed loop is determined by the location of the eigenvalues of the system and following the approach described in Lecture 7, the eigenvalues are found by solving

$$\begin{aligned}0 &= \det\left\{\begin{bmatrix} (sI - A - BK) & BK\\ 0 & (sI - A + LC)\end{bmatrix}\right\}\\ &= \det\{(sI - A - BK)\}\,\det\{(sI - A + LC)\}\end{aligned}$$

so that the eigenvalues of the closed loop system are the union of the eigenvalues of the LQR controller and the Kalman filter. We know that subject to conditions on

the controllability and observability of the system, then each of the LQR controller and the Kalman filter are stable, so that the eigenvalues of $(A+BK)$ and the eigenvalues of $(A-LC)$ all lie in the left half plane. This means that overall closed loop response is also stable.

Summary The LQG controller is designed in two steps

1. Design an LQR for the system as if there were no process noise. Provided that the conditions on the controllability and the observability of the state space model are satisfied, then the closed loop (deterministic) system will be stable and the closed loop poles will be the eigenvalues of $(A+BK)$, which will be in the left half plane.
2. Design a Kalman filter for estimating the states of the system. Again, provided that the conditions on controllability and observability of the system are satisfied, then the observer will be stable and the observer poles will be the eigenvalues of $(A-LC)$, which will be in the left half plane.

Although the closed loop response is stable, this does not automatically imply that LQG *controller* itself is also stable. The controller takes the measurement $y(t)$ as its input (along with the desired value of the state $r(t)$) and generates the input to the system $u(t)$. Since $u(t) = K\hat{x}(t) + r(t)$, then

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L\left(y(t) - C\hat{x}(t)\right) \\ &= A\hat{x}(t) + BK\hat{x}(t) + Br(t) + Ly(t) - LC\hat{x}(t) \\ &= \left(A + BK - LC\right)\hat{x}(t) + Ly(t) + Br(t)\end{aligned}$$

This means that the state space model of the *controller* is

$$\dot{\hat{x}}(t) = \left(A + BK - LC\right)\hat{x}(t) + \begin{bmatrix} L & B \end{bmatrix}\begin{bmatrix} y(t) \\ r(t) \end{bmatrix}$$

$$u(t) = K\hat{x}(t) + \begin{bmatrix} 0 & I \end{bmatrix}\begin{bmatrix} y(t) \\ r(t) \end{bmatrix}$$

and the poles of the controller will occur at the eigenvalues of $(A+BK-LC)$.

Although the eigenvalues of $(A+BK)$ and $(A-LC)$ will both lie in the left half plane, this does not mean that the eigenvalues of $(A+BK-LC)$ will be in the

left half plane, so we cannot guarantee that the LQG *controller* will be stable, even if the underlying system is itself stable (i.e. the eigenvalues of A lie in the left half plane). As we will discuss in the B15 Controller Performance lectures, this means that if the eigenvalues of $(A + BK - LC)$ do not all lie in the left half plane, then the closed loop system is not internally stable and this will lead to the control inputs $u(t)$ growing exponentially. For this reason, once an LQG controller has been designed, then it is important to check the location of the eigenvalues of $(A+BK-LC)$ and if the controller is not stable, then the cost function in (10.2) needs to be modified, usually by reducing Q_{c}, until a stable controller can be found.

10.2 Loop transfer recovery

It can be shown that for a single-input, single-output system, the steady state Linear Quadratic Regulator (LQR) has a phase margin of at least 60°. This means that it is relatively robust to any uncertainties in the underlying state space model. Due to the duality between the LQR and the Kalman filter, the steady state Kalman filter has the same phase margin. However, it was discovered that when the LQR and the Kalman filter are combined in Linear Quadratic Gaussian (LQG) control, the resulting controller does not inherit the good phase margin of the individual components. Intuitively, the reason for this is because the Kalman filter cannot distinguish between the process noise and therefore generates an incorrect estimate of the state, which is then used by the state feedback law to generate the input $u(t)$.

One approach to reduce this problem is to modify the covariance of the process noise by including an additional term to "represent" the modelling uncertainty, so that

$$Q_{\mathrm{m}} = Q + q_{\mathrm{m}} I$$

where $q_{\mathrm{m}} \geq 0$ is a (scalar) tuning parameter; increasing q_{m} increases the robustness of the LQG controller to uncertainties in the state space model. This procedure is known as loop transfer recovery as it aims to "recover" the good robustness properties of the LQR controller.

Although loop transfer recovery works well, it does require tuning to find a suit-

able value for q_{m} and it is not straightforward to extend the method to multi-input, multi-output systems. The need for a systematic approach to handling model uncertainties combined with the desire to ensure that the controller can be guaranteed to be stable, led to the development of $\mathcal{H}_\infty$ control which will be introduced in the B15 Controller Performance lectures and covered in more detail in the C20 course in the fourth year.

10.3 Including integral action

The LQR regulator is based on a state feedback law of the form

$$u(t) = Kx(t) + r(t)$$

where $r(t)$ is the external input vector. The LQG controller has the same form except that the input is derived from the state estimate. In both cases, $u(t)$ is obtained from a gain matrix multiplied by the state, which corresponds to proportional control. In most practical control systems, we want to include integral action so that we can make the actual output $y(t)$ match the desired output. We will show how the controllers can be modified to achieve this. To simplify the analysis, we will assume that there is no noise and focus on a LQR controller that is applied to the state space model that takes $D = 0$, so that there is no "straight-through" term in the measurement equation.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t)\end{aligned}$$

We want to include integral action in the controller, so we define

$$z(t) = \int_0^t r_y(\tau) - y(\tau)\,\mathrm{d}\tau$$

which is the integral of the error between the output $y(t)$ and a desired value $r_y(t)$. Note that we are using a the subscript to distinguish the desired output $r_y(t)$ from $r(t)$, which is a reference value for the state. We can write

$$\begin{aligned}\dot{z}(t) &= r_y(t) - y(t)\\ &= r_y(t) - Cx(t)\end{aligned}$$

For a constant r_y, the system will reach steady state when $\dot{x}(t) = 0$ and at this point,

$$0 = A\bar{x} + B\bar{u}$$

where $\bar{x}$ and $\bar{u}$ are the steady state values of the state and the input. The corresponding steady state value of the output will be

$$\bar{y} = C\bar{x}$$

When steady state is reached, then $\dot{z}(t)$ is also zero, so that

$$\begin{aligned} 0 &= r_y - \bar{y} \\ &= r_y - C\bar{x} \end{aligned}$$

Note that the steady state value of $z(t)$ will be zero, so that $\bar{z} = 0$ (otherwise the effect of the integral would keep $z(t)$ changing).

We can express the signals as deviations from their steady state values by defining

$$\begin{aligned} x(t) &= \bar{x} + \delta x(t) \\ u(t) &= \bar{u} + \delta u(t) \\ y(t) &= \bar{y} + \delta y(t) \end{aligned}$$

so that

$$\begin{aligned} \dot{x}(t) &= \dot{\delta x}(t) \\ &= \cancel{A\bar{x}} + A\delta x(t) + \cancel{B\bar{u}} + B\delta u(t) \\ &= A\delta x(t) + B\delta u(t) \end{aligned}$$

$$\begin{aligned} \bar{y} + \delta y(t) &= C\bar{x} + C\delta x \\ \delta y(t) &= C\delta x \end{aligned}$$

$$\begin{aligned} \dot{z}(t) &= \dot{\delta z}(t) \\ &= \cancel{r_y} - \cancel{C\bar{x}} - C\delta x(t) \\ &= -C\delta x(t) \end{aligned}$$

Combining these into a state space model gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix} = \begin{bmatrix} A & 0\\ -C & 0\end{bmatrix}\begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix} + \begin{bmatrix} B\\ 0\end{bmatrix}\delta u(t)$$
$$\delta y(t) = \begin{bmatrix} C & 0\end{bmatrix}\begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix}$$

Define a state feedback law for this system as

$$\delta u(t) = K\begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix}$$

and then find K by solving the infinite time LQR that minimises

$$J = \int_0^\infty \left\{ \begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix}^\top \begin{bmatrix} Q_\mathrm{c} & 0\\ 0 & \eta I\end{bmatrix}\begin{bmatrix}\delta x(t)\\ \delta z(t)\end{bmatrix} + \delta u^\top(t) R_\mathrm{c}\delta u(t)\right\}\mathrm{d}t$$

where $\eta \in \mathbb{R}$ is a positive scaler which adjusts the relative weighting that is applied to integral action in the controller. Partitioning the feedback matrix as

$$K = \begin{bmatrix} K_\mathrm{P} & K_\mathrm{I}\end{bmatrix}$$

then the control signal

$$\begin{aligned} u(t) &= \bar{u} + \delta u(t)\\ &= \bar{u} + K_\mathrm{P}\,\delta x(t) + K_\mathrm{I}\int_0^t \delta z(\tau)\mathrm{d}\tau\end{aligned}$$

which is in the form of a P+I controller. Given controller weightings Q_c, R_c and η, this shows how an optimal P+I can be obtained from an LQR design. The controller is "tuned" by adjusting η, which is the weight on the integral term. You will use this form of controller in the B15 lab.

11 Discrete-time systems

Up to this point, we have considered continuous-time state space models, which express the evolution of the states in terms of differential equations. Although this is convenient for the purpose of designing controllers and observers, control systems are usually implemented on computer systems, where the measurements are samples of the system output that are taken via an A/D converter, while the inputs to the system are generated using a D/A converter, which implements a zero order hold (ZOH). When considering Discrete Systems in the A2 Introduction to Control Systems course, you described the response of these system using transfer functions based on $\mathcal{Z}$-transforms. We will now provide a brief revision of how discrete-time systems can be described using state space models.

11.1 Discrete-time state space models

The state space model of a continuous-time, Linear Time-invariant (LTI) system that describes the evolution of the state $x(t) \in \mathbb{R}^n$, with m inputs $u(t) \in \mathbb{R}^m$ and p outputs (measurements) $y(t) \in \mathbb{R}^p$ is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) & x(0) &= x_0 \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

where $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. We will assume that the measurements from the system are sampled at $t = kT$ for $k = 0, 1, \ldots$, where T is the time between samples. We will denote the measurement sampled at $t = kT$ by $y_k = y(kT)$ and to determine y_k, we require $x_k = x(kT)$, which are the values of the states at the sample times. Given the initial state, $x(0)$, at $t = 0$, the evolution of the state over $t \in [0, T)$ is given by

$$x(T) = \mathrm{e}^{AT}x(0) + \int_0^T \mathrm{e}^{A(T-\tau)}B\,u(\tau)\mathrm{d}\tau$$

If the system is operated in sample and hold mode, then $u(t) = u(0)$ is held constant over $t \in [0, T)$ and provided that A is invertible

$$\begin{aligned} x(T) &= \mathrm{e}^{AT}x(0) + \left\{\int_0^T \mathrm{e}^{A(T-\tau)}B\mathrm{d}\tau\right\}u(0) \\ &= \mathrm{e}^{AT}x(0) + \left[-A^{-1}\mathrm{e}^{A(T-\tau)}B\right]_0^T u(0) \\ &= \mathrm{e}^{AT}x(0) + \left[-A^{-1}B + A^{-1}\mathrm{e}^{AT}B\right]u(0) \\ &= \mathrm{e}^{AT}x(0) + A^{-1}\left(\mathrm{e}^{AT} - I\right)Bu(0) \\ &= A_\mathrm{d}x(0) + B_\mathrm{d}u(0) \end{aligned}$$

where

$$A_\mathrm{d} = \mathrm{e}^{AT} \qquad B_\mathrm{d} = A^{-1}\left(\mathrm{e}^{AT} - I\right)B$$

Note that if A is singular (i.e. when one or more of the eigenvalues of A are zero), then it is not possible to calculate B_d directly. However, one approach to handling this problem is to decompose A as $A = V\Lambda V^{-1}$, where Λ is a diagonal matrix containing the eigenvalues, then

$$B_\mathrm{d} = V\,\mathsf{diag}\left\{\frac{\mathrm{e}^{\lambda_i T} - 1}{\lambda_i}\right\}V^{-1}$$

and B_d can be calculated by considering $\lambda_i \to 0$.

This allows us to directly express the state at $t = T$ in terms of the state at $t = 0$ and the (constant) input applied over the sampling interval $t \in [0, T)$. Applying this repeatedly for each sampling interval leads to the discrete-time model

$$\begin{aligned} x_{k+1} &= A_\mathrm{d}x_k + B_\mathrm{d}u_k \qquad\qquad \text{given } x_0 \\ y_k &= Cx_k + Du_k \end{aligned}$$

where u_k is the constant input applied by the zero order hold over the interval $t \in [kT, (k+1)T)$. Note that while the evolution of the continuous-time model was expressed in terms of a (vector) first-order differential equation, the discrete-time model is a recursion. The recursion is initialised with x_0. Also note that because the output y_k is obtained by sampling the continuous-time evolution of the state at $t = kT$, the C and D matrices in the discrete-time model are the same as in the continuous-time model.

If we take the $\mathcal{Z}$-transform of the signals in the discrete-time model and assume that $x_0 = 0$

$$zX(z) = A_{\mathrm{d}}X(z) + B_{\mathrm{d}}U(z)$$
$$Y(z) = CX(z) + DU(z)$$

and rearranging the first equation gives

$$(zI - A_{\mathrm{d}})\, X(z) = B_{\mathrm{d}}U(z)$$

which leads to

$$Y(z) = G_{\mathrm{d}}(z)U(z)$$

where $G_{\mathrm{d}}(z)$ is the discrete-time transfer function

$$G_{\mathrm{d}}(z) = C\,(zI - A_{\mathrm{d}})^{-1}\, B_{\mathrm{d}} + D$$

The poles of this transfer function are the same as the eigenvalues of A_{d} and the system will be stable if these eigenvalues lie inside the unit circle.

Example 25. *A scalar, continuous-time system described by the transfer function*

$$G(s) = \frac{a}{s + a}$$

is used in a sampled data system, where the sample interval is T and the inputs to the system are generated by a zero-order hold. Determine the discrete-time transfer function, $G_{\mathrm{d}}(z)$ for the sampled system.

Solution:

Method 1: *(Note: this approach was introduced in the A2 Introduction to*

Control Systems course). The discrete-time transfer function is given by

$$\begin{aligned} G_{\mathrm{d}}(z) &= \left(1 - z^{-1}\right) \mathcal{Z}\left\{\mathcal{L}^{-1}\left(\frac{G(s)}{s}\right)\right\} \\ &= \left(1 - z^{-1}\right) \mathcal{Z}\left\{\mathcal{L}^{-1}\left(\frac{1}{s}\,\frac{a}{s+a}\right)\right\} \\ &= \left(1 - z^{-1}\right) \mathcal{Z}\left\{\mathcal{L}^{-1}\left(\frac{1}{s} - \frac{1}{s+a}\right)\right\} \\ &= \left(1 - z^{-1}\right) \mathcal{Z}\left\{\left(1 - \mathrm{e}^{-at}\right)\mathcal{U}(t)\right\} \\ &= \left(1 - z^{-1}\right)\left(\frac{z}{z-1} - \frac{z}{z - \mathrm{e}^{-aT}}\right) \\ &= \frac{1 - \mathrm{e}^{-aT}}{z - \mathrm{e}^{-aT}} \end{aligned}$$

where $\mathcal{U}(t)$ denotes the unit step function.

Method 2: *A state space model for the continuous-time system is*

$$\begin{aligned} \dot{x}(t) &= -ax(t) + au(t) \\ y(t) &= x(t) \end{aligned}$$

so that the "matrices" of the model are $A = -a$, $B = a$, $C = 1$, $D = 0$. The corresponding A_{d} and B_{d} matrices for the discrete-time model are

$$A_{\mathrm{d}} = \mathrm{e}^{-aT} \qquad B_{\mathrm{d}} = \frac{1}{-a}\left(\mathrm{e}^{-aT} - 1\right) a = \left(1 - \mathrm{e}^{-aT}\right)$$

so that

$$G_{\mathrm{d}}(z) = C\left(zI - A_{\mathrm{d}}\right)^{-1} B_{\mathrm{d}} = \frac{1 - \mathrm{e}^{-aT}}{z - \mathrm{e}^{-aT}}$$

In this example, it is easier to obtain the discrete-time transfer function using the state-space model, but this is because there is only a single state, so that it is straightforward to calculate e^{AT}. For systems that have more than one state, then it is necessary to determine the *matrix* exponential, which can be done in MATLAB using the `expm` function. In fact, the MATLAB function `c2d`, which converts the transfer function of a continuous-time system to discrete-time, does this by realising the continuous-time transfer function in terms of a state-space model, generating the A_{d} and B_{d} matrices using the `expm` function to create a discrete-time model and then expressing the state space as a discrete-time transfer function.

11.2 Discrete Kalman filter

As with the continuous-time Kalman filter, the discrete-time version is based on a state-space model that includes both additive process (or state) noise and additive measurement noise

$$\begin{aligned} x_{k+1} &= A_{\mathrm{d}} x_k + B_{\mathrm{d}} u_k + w_k \\ y_k &= C x_k + D u_k + v_k \end{aligned}$$

where $w_k \in \mathbb{R}^n$ is the process noise that enters the system at $t = kT$ and $v_k \in \mathbb{R}^p$ is the noise that affects the measurement at $t = kT$. Note that both sources of noise enter at the sample time. This is a reasonable assumption for the measurement noise, as y_k is sampled at $t = kT$ so each measurement is only affected by the noise at this time instant. However, the state equation is derived from the evolution of the state over the period $t \in ((k-1)T, kT)$, so w_k represents the accumulation of the effect of the process noise over this interval.

We will assume that the process and measurement noises are uncorrelated, zero-mean, white noise, vector random processes, with known covariance matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{p \times p}$

$$\begin{aligned} E\left[w_k w_{k+\ell}^\top\right] &= Q\,\delta_\ell \\ E\left[v_k v_{k+\ell}^\top\right] &= R\,\delta_\ell \end{aligned}$$

where δ_ℓ is the discrete Dirac delta function. The assumption that the noise sequences are white means that the noise at the kth sample is uncorrelated with the noise at any other sample. We will also assume that there is no correlation between the two noise processes so that

$$E\left[w_k v_{k+\ell}^\top\right] = 0$$

The discrete Kalman filter calculates the estimate of the state recursively. Suppose that we have an estimate of the state x_k based on the measurements $\{y_1, y_2, \ldots, y_k\}$, where we will denote the estimate as $\hat{x}_{k|k}$ to indicate that the estimate is *conditional* on the sequence of measurements up to y_k. The aim of the Kalman filter is to obtain an estimate of the state at next time step, $\hat{x}_{k+1|k+1}$, based on the all the

measurements taken up to the $(k+1)$th sample. In order to initialise this recursion, we require an estimate of $\hat{x}_{0|0}$, which is taken to be

$$\hat{x}_{0|0} = E\left[x_0\right]$$

We will denote $e_{k+1|k+1} \in \mathbb{R}^n$ as

$$e_{k+1|k+1} = x_{k+1} - \hat{x}_{k+1|k+1}$$

which is the error in the estimate that is based on the measurements up to the $(k+1)$th sample. As for the continuous-time Kalman filter, define $P_{k+1|k+1} \in \mathbb{R}^{n\times n}$ as the covariance of the estimation error

$$P_{k+1|k+1} = E\left[e_{k+1|k+1}e_{k+1|k+1}^\top\right]$$

The Kalman filter is an *unbiased linear* estimator that minimises the variance of the estimation error

$$E\left[e_{k+1|k+1}^\top e_{k+1|k+1}\right] = \text{trace}\left\{E\left[e_{k+1|k+1}e_{k+1|k+1}^\top\right]\right\} = \text{trace}\left\{P_{k+1|k+1}\right\}$$

The estimate is *unbiased* if

$$E\left[\hat{x}_{k+1|k+1}\right] = E\left[x_{k+1}\right]$$

so that the expected value of the estimate is the same as the expected value of the state. This means that

$$\begin{aligned} E\left[e_{k+1}\right] &= E\left[x_{k+1}\right] - E\left[\hat{x}_{k+1|k+1}\right] \\ &= 0 \end{aligned}$$

The estimator is *linear* because $\hat{x}_{k+1|k+1}$ is formed by taking a linear combination of a prediction of the state $\hat{x}_{k+1|k}$ and the new measurement y_{k+1}. It can be shown that the Kalman filter is the *best* linear unbiased estimator.

Although there are a number of different ways of writing the discrete-time Kalman filter, the most common approach is based on two separate steps.

- **0. Initialise**

$$\hat{x}_{0|0} = E\left[x_0\right] \qquad P_{0|0} = E\left[\left(x_0 - \hat{x}_{0|0}\right)\left(x_0 - \hat{x}_{0|0}\right)^\top\right]$$

- **1. Prediction**

$$\hat{x}_{k+1|k} = A_{\mathrm{d}}\,\hat{x}_{k|k} + B_{\mathrm{d}}u_k$$
$$P_{k+1|k} = A_{\mathrm{d}}^{\top} P_{k|k} A_{\mathrm{d}} + Q$$

- **2. Measurement Update**

$$L_{k+1} = P_{k+1|k}C^{\top}\left(CP_{k+1|k}C^{\top} + R\right)^{-1}$$
$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + L_{k+1}\left(y_{k+1} - \underbrace{C\hat{x}_{k+1|k} - Du_k}_{\hat{y}_{k+1}}\right)$$
$$P_{k+1|k+1} = \left(I - L_{k+1}C\right)P_{k+1|k}$$

The prediction step creates an (unbiased) estimate of x_{k+1} based upon the measurements up to the kth sample, while the measurement update step uses the new measurement taken at the $(k+1)$th step to improve the estimate. Note that $L_{k+1} \in \mathbb{R}^{n\times p}$ is the (discrete) Kalman gain.

11.2.1 Prediction step error

The *prediction* step uses $\hat{x}_{k|k}$, the estimate of the state at the kth step, to form an *unbiased* estimate of $\hat{x}_{k+1|k}$, the state at the next step. The error associated with this step is

$$\begin{aligned} e_{k+1|k} &= x_{k+1} - \hat{x}_{k+1|k} \\ &= A_{\mathrm{d}}x_k + \cancel{B_{\mathrm{d}}u_k} + w_k - A_{\mathrm{d}}\hat{x}_{k|k} - \cancel{B_{\mathrm{d}}u_k} \\ &= A_{\mathrm{d}}\left(x_k - \hat{x}_{k|k}\right) + w_k \\ &= A_{\mathrm{d}}e_{k|k} + w_k \end{aligned}$$

This estimate is unbiased because

$$E\left[e_{k+1|k}\right] = A_{\mathrm{d}}E\left[e_{k|k}\right] + \cancel{E[w_k]}$$

where $E\left[w_k\right] = 0$ because w_k is assumed to have zero-mean. Hence, if $E\left[e_{k|k}\right] = 0$, then $E\left[e_{k+1|k}\right]$ is also zero. Note that the recursion is initialised with $\hat{x}_{0|0} = E\left[x_0\right]$, so $E\left[e_{0|0}\right] = 0$

The covariance of the error is

$$\begin{aligned} P_{k+1|k} &= E\left[e_{k+1|k}e_{k+1|k}^\top\right] \\ &= E\left[\left(A_\mathrm{d}e_{k|k}+w_k\right)\left(A_\mathrm{d}e_{k|k}+w_k\right)^\top\right] \\ &= A_\mathrm{d}E\left[e_{k|k}\,e_{k|k}^\top\right]A_\mathrm{d}^\top + A_\mathrm{d}\cancel{E\left[e_{k|k}\,w_k^\top\right]} + \cancel{E\left[w_k\,e_{k|k}^\top\right]}A_\mathrm{d}^\top + E\left[w_k\,w_k^\top\right] \\ &= A_\mathrm{d}P_{k|k}A_\mathrm{d}^\top + Q \end{aligned}$$

The terms $E\left[e_{k|k}\,w_k^\top\right] = E\left[w_k\,e_{k|k}^\top\right] = 0$ because

$$\begin{aligned} E\left[e_{k|k}\,w_k^\top\right] &= E\left[\left(x_k-\hat{x}_{k|k}\right)w_k^\top\right] \\ &= E\left[x_k\,w_k^\top\right] - E\left[\hat{x}_{k|k}\,w_k^\top\right] \\ &= 0 \end{aligned} \tag{11.1}$$

To see this, write

$$\begin{aligned} E\left[x_k\,w_k^\top\right] &= E\left[\left(A_\mathrm{d}x_{k-1}+B_\mathrm{d}u_{k-1}+w_{k-1}\right)\,w_k^\top\right] \\ &= A_\mathrm{d}\cancel{E\left[x_{k-1}\,w_k^\top\right]} + B_\mathrm{d}u_{k-1}\,\cancel{E\left[w_k^\top\right]} + \cancel{E\left[w_{k-1}\,w_k^\top\right]} \\ &= 0 \end{aligned}$$

where the first term is zero because x_{k-1} is independent of w_k, the second term is zero because u_k is deterministic, so it can be taken outside the expectation operation and w_k has zero-mean, while the third term is zero because w_k is white noise, so w_{k-1} is independent of w_k.

The second term in (11.1) is also zero because $\hat{x}_{k|k}$ is uncorrelated with w_k.

$$\begin{aligned} E\left[\hat{x}_{k|k}w_k^\top\right] &= E\left[\left\{\hat{x}_{k|k-1}+L_k\left(y_k-C\hat{x}_{k|k-1}-Du_k\right)\right\}w_k^\top\right] \\ &= E\left[\hat{x}_{k|k-1}w_k^\top\right] + L_k\,E\left[y_kw_k^\top\right] - L_kC\,E\left[\hat{x}_{k|k-1}w_k^\top\right] - L_kDu_k\,\cancel{E\left[w_k^\top\right]} \\ &= \left(I-L_kC\right)E\left[\hat{x}_{k|k-1}w_k^\top\right] + L_k\,E\left[\left(Cx_k+Du_k+v_k\right)w_k^\top\right] \\ &= \left(I-L_kC\right)E\left[\hat{x}_{k|k-1}w_k^\top\right] + L_kC\,E\left[x_kw_k^\top\right] + L_kDu_k\,\cancel{E\left[w_k^\top\right]} + \cancel{E\left[v_kw_k^\top\right]} \\ &= 0 \end{aligned}$$

The first term is zero because $\hat{x}_{k|k-1}$ depends upon information that is available at the $(k-1)$th step, which is independent of w_k. The second term is zero because we have just shown that $E\left[x_k\,w_k^\top\right] = 0$.

11.2.2 Measurement step error

The *measurement* step uses the measurement y_{k+1} to obtain $\hat{x}_{k+1|k+1}$ by updating $\hat{x}_{k+1|k}$. We will assume that this update step takes the form of a *linear* combination of the prediction and the new measurement

$$\hat{x}_{k+1|k+1} = L'_{k+1}\hat{x}_{k+1|k} + L_{k+1}y_{k+1}$$

where $L'_{k+1} \in \mathbb{R}^{n\times n}$ and $L_{k+1} \in \mathbb{R}^{n\times n}$. We can then use induction to show that this is in fact the optimal choice.

The error associated with this step is

$$\begin{aligned}
e_{k+1|k+1} &= x_{k+1} - \hat{x}_{k+1|k+1} \\
&= x_{k+1} - \left(L'_{k+1}\hat{x}_{k+1|k} + L_{k+1}y_{k+1}\right) \\
&= x_{k+1} - L'_{k+1}\hat{x}_{k+1|k} - L_{k+1}\left(Cx_{k+1} + v_{k+1}\right) \\
&= x_{k+1} - L'_{k+1}\left(x_{k+1} - e_{k+1|k}\right) - L_{k+1}Cx_{k+1} - L_{k+1}v_{k+1} \\
&= \left(I - L'_{k+1} - L_{k+1}C\right)x_{k+1} + L'_{k+1}e_{k+1|k} - L_{k+1}v_{k+1}
\end{aligned}$$

This means that

$$E\left[e_{k+1|k+1}\right] = \left(I - L'_{k+1} - L_{k+1}C\right)E\left[x_{k+1}\right] + L'_{k+1}E\left[e_{k+1|k}\right] - L_{k+1}\cancel{E\left[v_{k+1}\right]}$$

We have shown that $x_{k+1|k}$ is an unbiased estimate of the state, so $E\left[e_{k+1|k}\right] = 0$. To ensure that the updated estimate is also unbiased, we require $E\left[e_{k+1|k+1}\right] = 0$, we need

$$\begin{aligned}
0 &= I - L'_{k+1} - L_{k+1}C \\
L'_{k+1} &= I - L_{k+1}C
\end{aligned}$$

and the update to obtain $\hat{x}_{k+1|k+1}$ becomes

$$\hat{x}_{k+1|k+1} = \left(I - L_{k+1}C\right)\hat{x}_{k+1|k} + L_{k+1}y_{k+1}$$

while the estimation error is

$$e_{k+1|k+1} = \left(I - L_{k+1}C\right)e_{k+1|k} - L_{k+1}v_{k+1}$$

The covariance of the updated error is

$$\begin{aligned} E\left[e_{k+1|k+1}\, e_{k+1|k+1}^\top\right] &= (I - L_{k+1}C)\, E\left[e_{k+1|k}\, e_{k+1|k}^\top\right] (I - L_{k+1}C)^\top \\ &\quad - (I - L_{k+1}C)\, \cancel{E\left[e_{k+1|k}\, v_{k+1}^\top\right]} \\ &\quad - \cancel{E\left[v_{k+1}\, e_{k+1|k}^\top\right]} (I - L_{k+1}C)^\top \\ &\quad + L_{k+1} E\left[v_{k+1}\, v_{k+1}^\top\right] L_{k+1}^\top \\ &= (I - L_{k+1}C)\, P_{k+1|k}\, (I - L_{k+1}C)^\top + L_{k+1} R L_{k+1}^\top \end{aligned}$$

where $E\left[e_{k+1|k}\, v_{k+1}^\top\right] = E\left[v_{k+1}\, e_{k+1|k}^\top\right] = 0$ since $e_{k+1|k}$ is uncorrelated with v_{k+1} because $e_{k+1|k}$ depends information available at kth step.

We want to choose the Kalman gain L_{k+1} such that we minimise the variance of the estimation error

$$E\left[e_{k+1|k+1}^\top\, e_{k+1|k+1}\right] = \mathsf{tr}\left\{E\left[e_{k+1|k+1}\, e_{k+1|k+1}^\top\right]\right\} = \mathsf{tr}\left\{P_{k+1|k+1}\right\}$$

where $P_{k+1|k+1} = E\left[e_{k+1|k+1}\, e_{k+1|k+1}^\top\right]$. As in the case for the continuous Kalman filter, we find the optimal value by completing the square.

$$\begin{aligned} P_{k+1|k+1} &= (I - L_{k+1}C)\, P_{k+1|k}\, (I - L_{k+1}C)^\top + L_{k+1} R L_{k+1}^\top \\ &= (I - L_{k+1}C)\, P_{k+1|k}\left(I - C^\top L_k^\top\right) + L_{k+1} R L_{k+1}^\top \\ &= \left(P_{k+1|k} - L_{k+1} C P_{k+1|k}\right)\left(I - C^\top L_k^\top\right) + L_{k+1} R L_{k+1}^\top \\ &= P_{k+1|k} - P_{k+1|k} C^\top L_k^\top - L_{k+1} C P_{k+1|k} + L_{k+1} C P_{k+1|k} C^\top L_k^\top + L_{k+1} R L_{k+1}^\top \\ &= P_{k+1|k} - P_{k+1|k} C^\top L_k^\top - L_{k+1} C P_{k+1|k} + L_{k+1}\left(C P_{k+1|k} C^\top + R\right) L_k^\top \\ &= \left[L_{k+1} - P_{k+1|k} C^\top \left(C P_{k+1|k} C^\top + R\right)^{-1}\right]\left(C P_{k+1|k} C^\top + R\right) \\ &\qquad \times \left[L_{k+1} - P_{k+1|k} C^\top \left(C P_{k+1|k} C^\top + R\right)^{-1}\right]^\top \\ &\qquad + P_{k+1|k} - P_{k+1|k} C^\top \left(C P_{k+1|k} C^\top + R\right)^{-1} C P_{k+1|k} \end{aligned}$$

Note that the derivation uses the fact the covariance matrices are symmetric, so that $P_{k+1|k} = P_{k+1|k}^\top$ and $R = R^\top$, which means that $\left(C P_{k+1|k} C^\top + R\right)$ is also symmetric.

The expression for $P_{k+1|k+1}$ is minimised by making

$$L_{k+1} = P_{k+1|k} C^\top \left(C P_{k+1|k} C^\top + R\right)^{-1}$$

so that

$$\begin{aligned} P_{k+1|k+1} &= P_{k+1|k} - P_{k+1|k}C^\top \left(CP_{k+1|k}C^\top + R\right)^{-1} CP_{k+1|k} \\ &= \left(I - L_{k+1}C\right) P_{k+1|k} \end{aligned} \tag{11.2}$$

11.3 Combined state and measurement update

We have presented the discrete-time Kalman filter using separate state and measurement update steps, primarily because that is the form that is most commonly used in practice. However, it is possible to combine the two steps for updating the covariance matrix. Combining the state update equations gives

$$\hat{x}_{k+1|k+1} = A_\mathrm{d}\hat{x}_{k|k} + B_\mathrm{d}u_k + L_{k+1}\left(y_{k+1} - C\hat{x}_{k+1|k} - Du_k\right)$$

Shifting the expression in (11.2) "one step back" gives

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}C^\top \left(CP_{k|k-1}C^\top + R\right)^{-1} CP_{k|k-1}$$

so that

$$\begin{aligned} P_{k+1|k} &= A_\mathrm{d}^\top P_{k|k}A_\mathrm{d} + Q \\ &= A_\mathrm{d}^\top \left(P_{k|k-1} - P_{k|k-1}C^\top \left(CP_{k|k-1}C^\top + R\right)^{-1} CP_{k|k-1}\right) A_\mathrm{d} + Q \\ &= A_\mathrm{d}^\top P_{k|k-1}A_\mathrm{d} - A_\mathrm{d}^\top P_{k|k-1}C^\top \left(CP_{k|k-1}C^\top + R\right)^{-1} CP_{k|k-1}A_\mathrm{d} + Q \end{aligned} \tag{11.3}$$

This is the discrete-time version of the Riccati equation, which is initialised using

$$P_{0|0} = E\left[\left(x_0 - \hat{x}_{0|0}\right)\left(x_0 - \hat{x}_{0|0}\right)\right]$$

As with the continuous-time Riccati equation, if the state space model is controllable and observable, then in the limit $k \to \infty$, the recursion converges to a unique solution

$$\lim_{k\to\infty} P_{k+1|k} = \bar{P}$$

This means that the Kalman gain will converge to a steady state value

$$\bar{L} = \bar{P}C^\top \left(C\bar{P}C^\top + R\right)^{-1}$$

11.4 Forms of Kalman filter

It might seem unnecessarily complicated to derive both the continuous-time and discrete-time versions of the Kalman filter. In general, the continuous-time version is used when the filter is used as an observer within a control system, such as a Linear Quadratic Gaussian (LQG) controller, although the discrete-time form of both the LQR and the Kalman filter will be used in model predictive control (MPC) - see C21 course next year. However, when the Kalman filter is only used to estimate the state of a system, the discrete-time form is almost always used. Historically, Kalman first derived the discrete-time version in 1960 and the continuous-time version was developed by Kalman and Bucy one year later in 1961. In his original paper, Kalman noted the duality with the linear quadratic regulator and he used results from LQR to derive the convergence of the Riccati equation for the Kalman filter. The two-step approach of the discrete-time Kalman filter, based on separate predict and measurement updates, was developed by Schmidt in the 1960's. This form of the Kalman filter was implemented in 2kBytes of memory and used as the main navigation program on the Apollo missions to the moon.

12 Implementing the Kalman filter

In this section, we will consider some practical aspects of implementing a Kalman filter. We will illustrate the issues using a model of an object moving in a straight line at a (nominal) constant velocity, which is subjected to random accelerations due to the effect of unmodelled external forces. If $p(t)$ denotes the position of the object, then in continuous time,

$$\ddot{p}(t) = w^{\mathrm{c}}(t) \tag{12.1}$$

where $w^{\mathrm{c}}(t) \in \mathbb{R}$ represents the random acceleration. This could represent the movement of a car along a straight road or a train running along a straight section of track, where the random accelerations could be due to wheel slip or wind buffeting. Although we have assumed that the object is nominally moving at constant velocity, we could also include and input $u(t)$ to describe known accelerations, for example from the engine or from the brakes, but because we are illustrating the response of the Kalman filter to the random variations, we will keep the model simple by not including these.

We will assume that the $w^{\mathrm{c}}(t)$ is a zero-mean, white noise random process, so that

$$E\left[w^{\mathrm{c}}(t)\right] = 0$$
$$E\left[w^{\mathrm{c}}(t)w^{\mathrm{c}}(t+\tau)\right] = Q\delta(\tau)$$

where $Q \in \mathbb{R}$ is the variance of the random acceleration. If $w^{\mathrm{c}}(t) = w_0^{\mathrm{c}}$ is constant over the interval $t \in [0, T)$, then direct integration of (12.1) gives

$$\dot{p}(T) = \dot{p}(0) + Tw_0^{\mathrm{c}}$$
$$p(T) = p(0) + T\dot{p}(0) + \frac{T^2}{2}w_0^{\mathrm{c}}$$

which can be written in state-space form

$$\begin{bmatrix} p(T) \\ \dot{p}(T) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p(0) \\ \dot{p}(0) \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} w_0^{\mathrm{c}}$$

Extending this over each interval $t \in [kT, (k+1)T)$ for $k = 0, 1, 2, \ldots$ leads to the discrete-time state space model

$$x_{k+1} = A_{\mathrm{d}}x_k + Gw_k^{\mathrm{c}}$$

where $x_k = [p(kT)\ \dot{p}(kT)]^\top$ is the state,

$$A_{\rm d} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

and the noise process $w_k \in \mathbb{R}^2$ is

$$w_k = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} w_k^{\rm c} = G w_k^{\rm c}$$

The expected value of this random variable is

$$E\left[w_k\right] = G E\left[w_k^{\rm c}\right] = 0$$

since $w_k^{\rm c}$ has zero mean. The covariance matrix is

$$E\left[w_k w_{k+\ell}^\top\right] = G\, E\left[w_k^{\rm c} w_{k+\ell}^{\rm c}\right] G^\top = G\, Q\, G^\top\, \delta_\ell$$

Remark. The derivation of the both the continuous-time and discrete-time Kalman filters in the previous lectures have been based on the models that uses the state update of the form (for the discrete-time case)

$$x_{k+1} = A_{\rm d} x_k + w_k$$

where w_k has the same dimension as the state and each element of w_k feeds directly into each of the states. We have used this form primarily because it highlights the duality between the Riccati equations for the linear quadratic regulator and the Kalman filter. However, in some books you will see the Kalman filter derived from a model where the state update is described by

$$x_{k+1} = A_{\rm d} x_k + G w_k^{\rm r}$$

where $G \in \mathbb{R}^{n \times n_{\rm w}}$ and $w_k^{\rm r} \in \mathbb{R}^{n_{\rm w}}$ is a "reduced dimension" white-noise process, with $n_{\rm w} \leq n$ so that the number of elements in the noise vector may be less than the number of states. However, is equivalent to using a noise vector $w_k = G w_k^{\rm r}$, where

$$\begin{aligned} E\left[w_k\right] &= G E\left[w_k^{\rm r}\right] \\ E\left[w_k w_{k+\ell}^\top\right] &= G E\left[w_k^{\rm r} (w_{k+\ell}^{\rm r})^\top\right] G^\top = G\, Q^{\rm r}\, G^\top\, \delta_\ell \end{aligned}$$

where $Q^{\mathrm{r}} \in \mathbb{R}^{n_{\mathrm{w}} \times n_{\mathrm{w}}}$ is the covariance matrix for w_k^{r}. The derivation of the Kalman filter then carries through with the covariance matrix of the state noise replaced by $G\,Q^{\mathrm{r}}\,G^\top$, so that update for covariance matrix is modified so that it takes the form

$$P_{k+1|k} = A_{\mathrm{d}}^\top P_{k|k} A_{\mathrm{d}} + G\,Q^{\mathrm{r}}\,G^\top$$

Example 26. *For the model of the position and velocity of an object with $T = 0.05s$ and $Q = 5$ and $R = 1$, Figure 13 shows the estimated position generated by the Kalman filter, along with the true position, along with the measured position. Figure 14 shows the estimated velocity (the nominal velocity is $1m.s^{-1}$), which takes longer to converge to the true value and is subject to more variation compared to the position, as unlike the position, we do not have a direct measurement of velocity.*

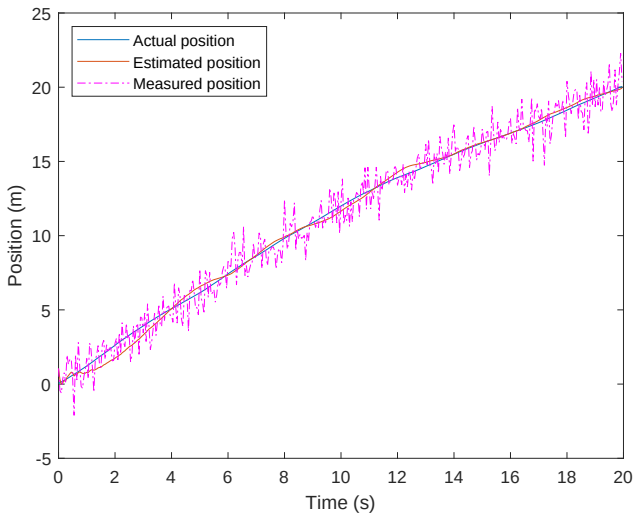

Figure 13: Estimate of position (red) together with actual position (blue) and measured position (dash-dot).

12.1 Initialisation

In order to start the recursion for the Kalman filter, we require $\hat{x}_{0|0} = E[x_0]$ and $P_{0|0} = E[(x_0 - \hat{x}_{0|0})\,(x_0 - \hat{x}_{0|0})^\top$. It is rare to have full knowledge of these values and in practice, the initial values have to be estimated (or sometimes, guessed!). However, provided that the controllability and observability conditions are satisfied, then the Kalman filter will be stable and the expected value of the estimation error

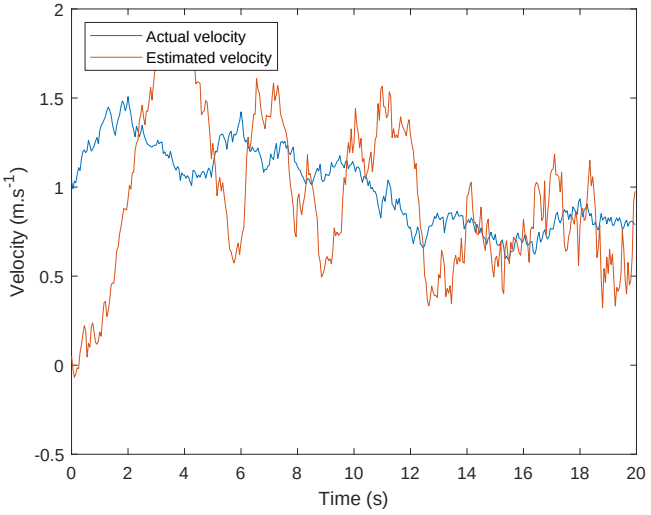

Figure 14: Estimate of velocity (red) together with actual velocity (blue).

will converge to zero. This means that the choice of initial condition is not critical as its effect will decay as the Kalman filter proceeds. However, it is important to ensure that the estimate of $P_{0|0}$ is symmetric and positive definite. For this reason, it is common to choose

$$P_{0|0} = p_0 I$$

where p_0 is a positive scalar.

12.2 Assessing performance

The Kalman filter is designed to minimise $E\left[e_k^\top e_k\right]$ where $e_k = x_k - \hat{x}_k$, the difference between the estimated state and the actual state. The problem is that we do not have access to the actual state (if we did, we would not be going to the trouble of implementing the Kalman filter to estimate it!). However, we do have the measurements, so we can evaluate

$$\nu_k = y_k - C\hat{x}_{k|k}$$

where $\nu_k \in \mathbb{R}^p$ is the difference between our prediction of the measured output and the actual measurement at each time step. The sequence of ν_k for $k = 1, 2, \ldots$ is referred to as the *innovations* because it reflects the new information that is

obtained from each measurement. By writing

$$\begin{aligned}\nu_k &= Cx_k + v_k - C\hat{x}_{k|k} \\ &= C\left(x_k - \hat{x}_{k|k}\right) + v_k \\ &= Ce_{k|k} + v_k\end{aligned}$$

so that

$$E\left[\nu_k\right] = C\cancel{E\left[e_{k|k}\right]} + \cancel{E\left[v_k\right]} = 0$$

and

$$\begin{aligned}E\left[\nu_k\,\nu_{k+\ell}^\top\right] &= E\left[\left(Ce_{k|k} + v_k\right)\left(Ce_{k+\ell|k+\ell} + v_{k+\ell}\right)^\top\right] \\ &= C\,E\left[e_{k|k}\,e_{k+\ell|k+\ell}^\top\right]\,C^\top + C\,\cancel{E\left[e_{k|k}\,v_{k+\ell}^\top\right]} + \cancel{E\left[v_k\,e_{k+\ell|k+\ell}^\top\right]}C^\top + E\left[v_k\,v_{k+\ell}\right] \\ &= \left(C\,P_{k|k}\,C^\top + R\right)\delta_\ell\end{aligned}$$

As a result, the innovations sequence is a zero-mean, Gaussian white-noise process, with covariance matrix $\left(C\,P_{k|k}\,C^\top + R\right)$. We can exploit this fact to assess the performance of the Kalman filter by checking that the hypothesis that the actual innovations sequence has this distribution. There are two main methods for doing this.

1. If the innovations sequence had zero mean and covariance matrix $\left(C\,P_{k|k}\,C^\top + R\right)$, then

   $$\tilde{\nu}_k = \left(C\,P_{k|k}\,C^\top + R\right)^{-1/2}\nu_k \sim \mathcal{N}(0, I)$$

   where $\tilde{\nu}_k \in \mathbb{R}^p$ is called the *normalised* innovation, which is a zero-mean Gaussian distributed process with *unit* variance. As a result, each element of $\tilde{\nu}_k$ will be Normally distributed, so that there is a 95% probability that these elements lie within ± 1.96 (note that 95% of the Normal distribution lies within ± 1.96 standard deviations of the mean). For the case where there is only one measurement, the innovation is scalar and it is straightforward to calculate $\tilde{\nu}_k$ by dividing ν_k by the square root of $\left(C\,P_{k|k}\,C^\top + R\right)$. It is more complicated to calculate $\tilde{\nu}_k$ when it is a vector because it requires the square root of a matrix. However, if R is positive definite and Q is positive semi-definite, so that $P_{k|k}$ is also positive semi-definite, then $\left(C\,P_{k|k}\,C^\top + R\right)$ is positive definite and it is possible to find a unique matrix $X \in \mathbb{R}^{p\times p}$, such that

   $$X\,X^\top = \left(C\,P_{k|k}\,C^\top + R\right)$$

so that $\tilde{\nu}_k = X^{-1}\nu_k$. The MATLAB command `sqrtm` will return this matrix.

In practice, it is often the case that the magnitude of the diagonal elements of $\left(C\,P_{k|k}\,C^\top + R\right)$ are much larger than the magnitude of the off-diagonal elements, so it is reasonable to use the approximation

$$[\tilde{\nu}_k]_i \approx \frac{[\nu_k]_i}{\sqrt{\left[C\,P_{k|k}\,C^\top + R\right]_{i,i}}}$$

2. Because the normalised innovations $\tilde{\nu}_k \sim \mathcal{N}(0, I)$ are normally distributed with unit variance, then

$$\tilde{\nu}_k^\top \tilde{\nu}_k \sim \chi^2(p)$$

so that $\tilde{\nu}_k^\top \tilde{\nu}_k$ is χ^2 distributed with p degrees of freedom, where p is the dimension of $\tilde{\nu}_k$. This means that if we have processed K measurements, then

$$\sum_{k=1}^{K} \tilde{\nu}_k^\top \tilde{\nu}_k \sim \chi^2(Kp)$$

so we test the hypothesis that $\tilde{\nu}_k$ is normally distributed with unit variance by checking whether the value of this statistic lies within the 95% confidence intervals of the χ^2 distribution with Kp degrees of freedom.

Example 27. *For the model of the position and velocity of an object used in Example 26, where $T = 0.05$s, Figure 15 shows the normalised innovations from the Kalman filter estimates, when $Q = 5$ and $R = 1$. It can be seen that almost all of the data points lie within the 95% confidence interval shown by the red dashed lines. For this example, the χ^2 statistic based on 400 data points is 406.6 which lies within the 95% confidence interval $[346.5, 457.3]$.*

12.3 Testing whiteness of residuals

Because

$$E\left[\nu_k\,\nu_{k+\ell}^\top\right] = \left(C\,P_{k|k}\,C^\top + R\right)\delta_\ell$$

then

$$E\left[\tilde{\nu}_k\,\tilde{\nu}_{k+\ell}^\top\right] = I\,\delta_\ell$$

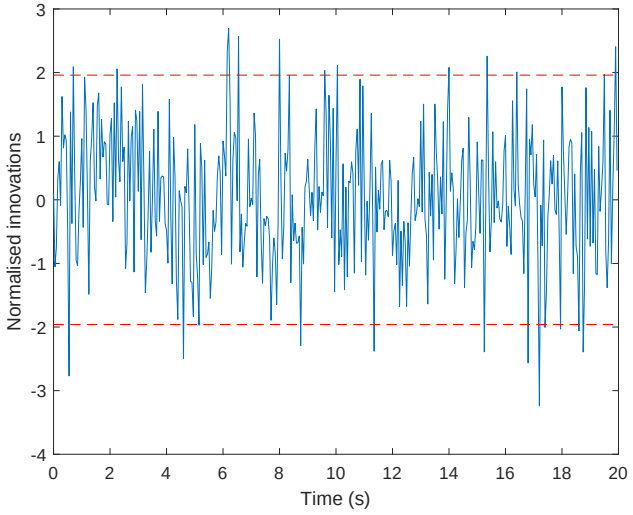

Figure 15: Normalised innovations together with 95% confidence intervals (red dashed lines).

This means that the autocorrelation of each element of $\tilde{\nu}_k$ should be zero except at zero lag (i.e. $\ell = 0$) and that there should no cross-correlation between the different elements in the vector. If we calculate the *estimated* autocorrelation of an individual element as

$$\hat{R}_i(\ell) = \frac{1}{K-\ell} \sum_{k=1}^{K} [\tilde{\nu}_k]_i \, [\tilde{\nu}_{k+\ell}]_i$$

then this estimate is a random variable, such that for $\ell \neq 0$

$$\sqrt{K} \frac{\hat{R}_i(\ell)}{\hat{R}_i(0)} \sim \mathcal{N}(0, 1)$$

This means that the hypothesis that the normalised residual are white can be tested with 95% confidence by checking that this statistic lies within ± 1.96 for $\ell = 1, 2, \ldots$. In practice, this is assessed by plotting $\hat{R}_i(\ell)/\hat{R}_i(0)$ for $\ell = 1, 2, \ldots$ and checking that these values lie within $\pm 1.96/\sqrt{K}$.

Example 28. *Figure 16 shows the estimated autocorrelation of normalised innovations from the Kalman filter estimates, scaled by the estimated autocorrelation at zero lag, for the estimates from the Kalman filter in Example 26. It can be seen that almost all of the data points lie within the 95% confidence shown by the red dashed lines, so we can accept the hypothesis that the innovations are white.*

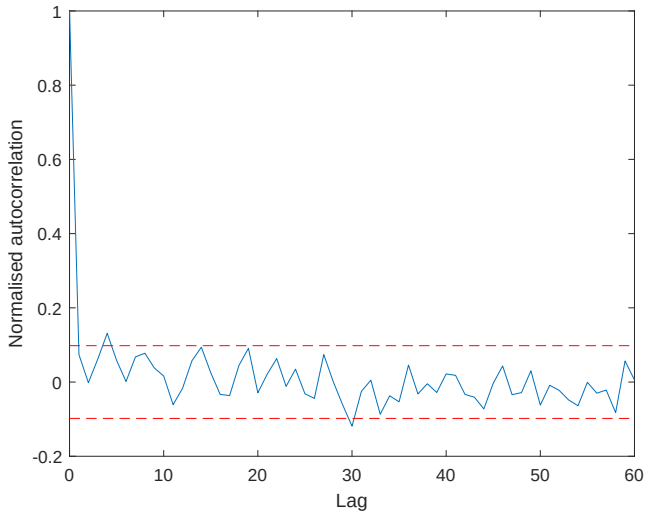

Figure 16: Estimated autocorrelation of normalised residuals divided by value at zero lag, with 95% confidence intervals (red dashed lines).

12.4 Choosing Q and R

One of the most difficult practical aspects of implementing a Kalman filter is the choice of Q and R, the covariance matrices for the process and measurement noises, as it is rare that we have accurate knowledge of the statistics of these noises. However, if we choose values for Q and R, we can run the Kalman filter using these matrices and then check the hypotheses that the normalised residuals $\tilde{\nu}_k$ are both normally distributed and white. If any of the hypotheses are not accepted, then Q and R can be adjusted. For the case where both the process noise and measurement noise are scalar, the following guidelines are useful when trying determine suitable values for Q and R.

1. If the ratio Q/R is too low, then the normalised innovations are non-white, which shows up in the autocorrelation plot.

2. Having found a suitable Q/R ratio, then the individual values of Q and R can be scaled until the test for a Normal distribution and/or the χ^2 test are satisfied

These guidelines are applicable for scalar systems, but choosing Q and R for systems with more than one measurement or more than one noise sources, so that Q and R are matrices, is more complicated. However, examining the normalised innovations

is valuable.

12.5 Outliers in measurement data

We have assumed that the measurements are subject to Gaussian measurement noise v_k, which represents sensor noise. In practice, it is common for measurements to be subject to "outliers" due to sensor errors that cause large deviations in the measurement signal, y_k. Having found suitable Q and R, we can monitor the normalised innovations to identify outliers in the measurement data. At each step of the Kalman filter, we calculate the normalised innovations, $\tilde{\nu}_k$. Because the normalised innovations should be Gaussian distributed with unit variance, then if the magnitude of $\tilde{\nu}_k$ is greater than (say) 2.8, then we can be 99.5% certain that the normalised innovation for this measurement does not belong to a Gaussian distribution with unit variance and can therefore be considered as an outlier. Under these circumstances, the measurement update step in the Kalman filter is not implemented for this measurement.

Example 29. *Figure 17 shows the noisy measurements from the system in Example 26, when there is a measurement outlier at $t = 5$s. The plot of the normalised innovations in Figure fig:normalisedInnovationswithOutlier shows that the corresponding point in the normalised innovations lies outside the 95% confidence bounds, which can be used to detect the outlier.*

12.6 Detecting faults and modelling errors

Monitoring the normalised innovations is also a useful way of detecting changes in the underlying system and in particular, for detecting faults. Faults often occur due to degradation or failure in actuators or sensors. For example, a fault in a sensor can increase the variance of the measurement noise in the measurement, which will increase the variance of the normalised innovations. Actuator faults tend to change the underlying dynamics of the model, which has the effect of causing correlation in the normalised residuals. The same effect will occur if there is an error in the state space model of the system, for example because of unmodelled dynamics. As

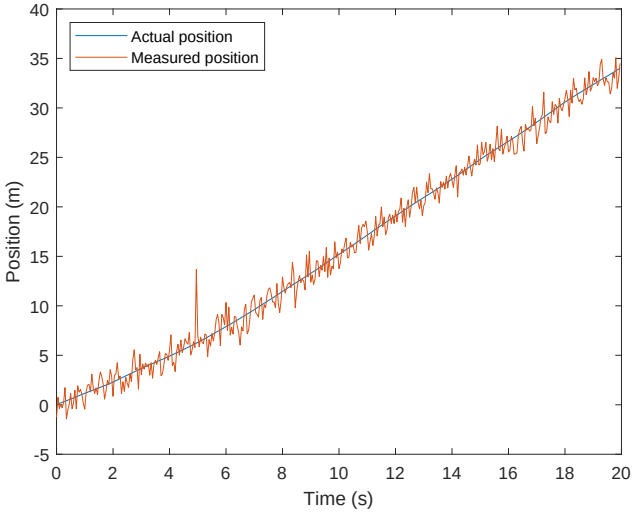

Figure 17: Measurement of position including outlier at $t = 5$s.

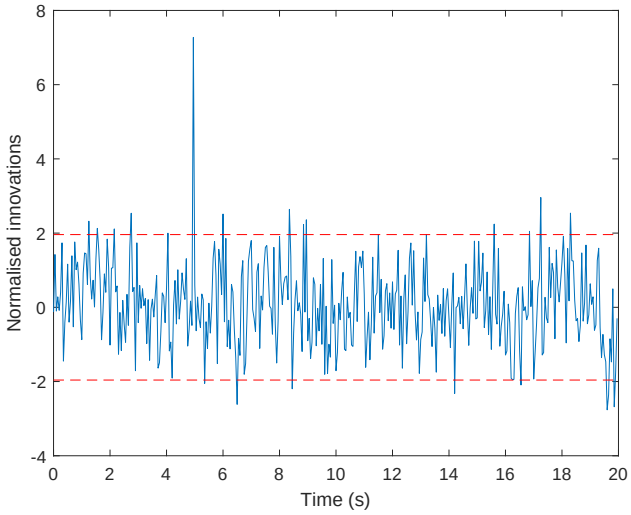

Figure 18: Normalised innovations for measurement data with outlier at $t = 5$s.

a result, self-validating sensors incorporate a Kalman filter so that the innovations can be monitored in order to detect faults.

Example 30. *Figure 19 shows the normalised innovations when there is an increase in the variance of the measurement noise from $R = 1$ to $R = 2$ at $t = 10$s. The plot shows that there is a corresponding increase in the magnitude of the normalised innovations, which can be used to detect the change.*

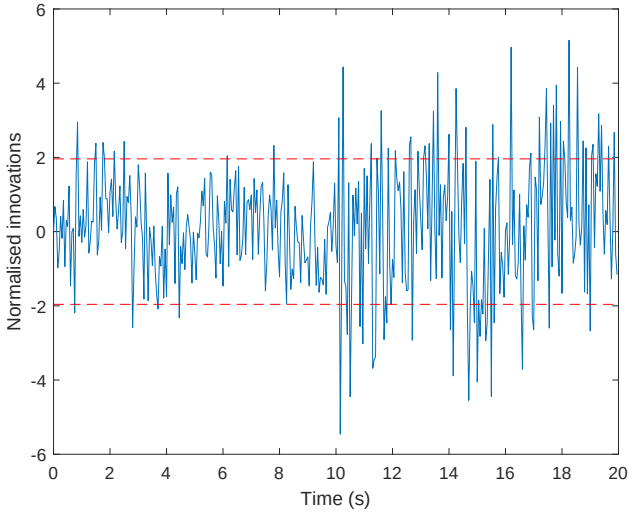

Figure 19: Normalised innovations when there is an increase in the variance of the measurement noise at $t = 10$s, together with 95% confidence bounds (red dashed lines).

12.7 Time-varying systems

Up to now, we have assumed that the matrices in the state space model are constant, however, the derivation of both continuous-time and discrete-time Kalman filters is still valid if the state space model is time varying, so that for the discrete-time case, the model becomes

$$\begin{aligned} x_{k+1} &= A_k^{\mathrm{d}} x_k + B_k^{\mathrm{d}} u_k + G_k^{\mathrm{d}} w_k^{\mathrm{r}} \\ y_k &= C_k x_k + D_k u_k + v_k \end{aligned}$$

with

$$\begin{aligned} E\left[w_k^{\mathrm{r}} (w_{k+\ell}^{\mathrm{r}})^\top\right] &= Q_k^{\mathrm{r}} \delta_\ell \\ E\left[v_k v_{k+\ell}^\top\right] &= R_k \delta_\ell \\ E\left[w_k^{\mathrm{r}} v_{k+\ell}^\top\right] &= 0 \end{aligned}$$

where all of the matrices can change at each time step. Note that we still require the noise sequences to be white, but we are allowing the covariance matrices, which describe the correlation between the individual elements within the noise vectors, to be time-varying. For discrete-time systems that are obtained by discretising an underlying continuous-time model, in addition to describing the case where the underlying model is changing over time, a time-varying model can be used to describe the case where the time between samples is not constant.

The Kalman filter for the time-varying model is

1. Prediction

$$\begin{aligned}\hat{x}_{k+1|k} &= A_k^{\mathrm{d}}\,\hat{x}_{k|k} + B_k^{\mathrm{d}} u_k \\ P_{k+1|k} &= A_k^{\mathrm{d}\top} P_{k|k} A_k^{\mathrm{d}} + G_k^{\mathrm{d}} Q_k^{\mathrm{r}} G_k^{\mathrm{d}\top}\end{aligned}$$

2. Measurement update

$$\begin{aligned}L_{k+1} &= P_{k+1|k} C_k^\top \left(C_k P_{k+1|k} C_k^\top + R_k\right)^{-1} \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + L_{k+1}\left(y_{k+1} - C_k \hat{x}_{k+1|k}\right) \\ P_{k+1|k+1} &= \left(I - L_{k+1} C_k\right) P_{k+1|k}\end{aligned}$$

Because the matrices that describe the model and the noises covariances are time-varying, the error covariance $P_{k+1|k+1}$ will not converge to a steady-state value, so an algebraic Riccati equation does not exist for a time-varying model.

12.8 Coloured noise

The derivation of the Kalman filter is based on the assumption that the process and measurement noises are white. In practice, this is often not reasonable as there may be correlation between the noise at successive time steps, in addition to the correlation between individual elements of the noise vector that is described by Q and R. Noise that has correlation in time is described as "coloured" (because it is no longer white noise). To show this, we will consider the process noise and show how time correlation can be included in the state space model by assuming the noise as being generated by passing white noise through a linear filter, so that for a first order filter described by a (discrete-time) transfer function

$$w_k^{\mathrm{c}} = \frac{g_0}{z + f_1} n_k$$

where $n_k \in \mathbb{R}^m$ is a white-noise process with covariance matrix $E\left[n_k n_{k+\ell}^\top\right] = Q_n$. This noise model can be expressed in state space form

$$w_{k+1}^{\mathrm{c}} = -f_1 w_k^{\mathrm{c}} + g_0 n_k$$

and this can be appended to the state space model for the system

$$\begin{bmatrix} x_{k+1} \\ w^{\mathrm{c}}_{k+1} \end{bmatrix} = \begin{bmatrix} A_k & G^{\mathrm{d}}_k \\ 0 & -f_1 \end{bmatrix} \begin{bmatrix} x_k \\ w^{\mathrm{c}}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ g_0 \end{bmatrix} n_k$$

$$y_k = [C \;\; 0] \begin{bmatrix} x_k \\ w^{\mathrm{c}}_k \end{bmatrix} + v_k$$

The Kalman filter can now be applied to the augmented system. The same approach can be used to model coloured measurement noise, but this results in the measurement noise being part of the state, so that there is no v_k term in the measurement equation. This violates the requirement that the covariance matrix for the measurement noise should be positive definite, so that $R > 0$.

12.9 Numerical issues

The covariance matrix for the estimation errors $E\left[e_k e_k^\top\right] = P_{k|k}$ should be both symmetric and positive definite. However, when Kalman filters were first implemented on a computer, it was discovered that the recursion could become unstable if $P_{k|k}$ was not positive definite as a result of numerical round-off errors. To avoid this problem, $P_{k|k}$ can be expressed in terms of its (matrix) square root

$$P_{k|k} = P^{\frac{1}{2}}_{k|k} \left(P^{\frac{1}{2}}_{k|k}\right)^\top$$

where $P^{\frac{1}{2}}_{k|k}$ is lower triangular. The algorithm can be rearranged so that $P^{\frac{1}{2}}_{k|k}$ is updated, which means that $P_{k|k}$ will be symmetric and positive definite, which is referred to as *square root filtering*. There are efficient methods for implementing this based upon the QR-factorisation (confusingly, this has nothing to do with the Q and R covariance matrices). There is (much!) more information on this in the book by Kalaith *et al.* (1999)

12.10 Nonlinear systems

The Kalman filter is based upon a linear state space model. However, it can be extended to provide an estimate of the state for a nonlinear system. We will start

off by considering a relatively simple time-varying, nonlinear model with *additive* noise (more general models include the noises within the non-linear functions)

$$x_{k+1} = f_k\left(x_k, u_k\right) + w_k$$
$$y_k = h_k\left(x_k, u_k\right) + v_k$$

where as before $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are zero-mean, white noise processes with $E\left[w_k w_{k+\ell}^\top\right] = Q_k\delta_\ell$, $E\left[v_k v_{k+\ell}^\top\right] = R_k\delta_\ell$ and $E\left[w_k v_{k+\ell}^\top\right] = 0$. The function $f_k\left(x_k, u_k\right)$ describes the non-linear evolution of the state, while $h_k\left(x_k, u_k\right)$ describes the non-linear measurement process. Suppose that at time step k, we have an estimate of the state $\hat{x}_{k|k}$ and given the input u_k, the Jacobians of these function around these values are

$$A_k = \left.\frac{\partial f_k}{\partial x}\right|_{\hat{x}_{k|k}, u_k} \qquad C_k = \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}_{k|k}, u_k}$$

The Extended Kalman filter (EKF) for this non-linear model takes the form

1. Prediction

$$\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}, u_k)$$
$$P_{k+1|k} = A_k^\top P_{k|k} A_k + Q_k$$

2. Measurement update

$$L_{k+1} = P_{k+1|k} C_k^\top \left(C_k P_{k+1|k} C_k^\top + R_k\right)^{-1}$$
$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + L_{k+1}\left(y_{k+1} - h_k(\hat{x}_{k+1|k}, u_k)\right)$$
$$P_{k+1|k+1} = \left(I - L_{k+1} C_k\right) P_{k+1|k}$$

Note that even if $f(x_k, u_k)$ and $h(x_k, u_k)$ do not depend on the time step, the EKF will be time-varying because the Jacobians depend upon $\hat{x}_{k|k}$ and u_k, so that they have to be evaluated at each time step, which means that the EKF requires considerably more computational effort compared to the standard Kalman filter. Also, unlike the linear case, which is relatively insensitive to the initial conditions, for the EKF, it is important to choose initial conditions that accurately reflect $x_{0|0} = E\left[x_0\right]$ and $P_{0|0} = \left[\left(x_0 - \hat{x}_{0|0}\right)\left(x_0 - \hat{x}_{0|0}\right)_k^\top\right]$.

In general, for many non-linear systems, the EKF works well. However, because the EKF effectively relies on a Taylor approximation for the update of $P_{k|k}$, it is

not possible to guarantee that the EKF will perform well. The main problem is that even if w_k and v_k are zero-mean, Gaussian white noise, the effect of the nonlinearites mean that the state x_k and the measurement y_k are not necessarily Normally distributed. As a result, we cannot assume that the estimation error $e_{k|k} = x_k - \hat{x}_{k|k}$ also has zero mean and the distribution of the error cannot completely characterised by the covariance matrix $P_{k|k}$. To overcome this, methods have been developed that are based on approximating the distribution of the estimation error by evolving a number of sample points through the state update equation. These include unscented Kalman filtering (UKF) and particle filtering (PF).