8 Linear Control Theory

The focus of this book has largely been on characterizing complex systems through dimensionality reduction, sparse sampling, and dynamical systems modeling. However, an overarching goal for many systems is the ability to actively manipulate their behavior for a given engineering objective. The study and practice of manipulating dynamical systems is broadly known as control theory, and it is one of the most successful fields at the interface of applied mathematics and practical engineering. Control theory is inseparable from data science, as it relies on sensor measurements (data) obtained from a system to achieve a given objective. In fact, control theory deals with living data, as successful application modifies the dynamics of the system, thus changing the characteristics of the measurements. Control theory forces the reader to confront reality, as simplifying assumptions and model approximations are tested.

Control theory has helped shape the modern technological and industrial landscape. Examples abound, including cruise control in automobiles, position control in construction equipment, fly-by-wire autopilots in aircraft, industrial automation, packet routing in the internet, commercial heating ventilation and cooling systems, stabilization of rockets, and PID temperature and pressure control in modern espresso machines, to name only a few of the many applications. In the future, control will be increasingly applied to high-dimensional, strongly nonlinear and multiscale problems, such as turbulence, neuroscience, finance, epidemiology, autonomous robots, and self driving cars. In these future applications, data-driven modeling and control will be vitally important; this is be the subject of Chapters 7 and 10.

This chapter will introduce the key concepts from closed-loop feedback control. The goal is to build intuition for how and when to use feedback control, motivated by practical real-world challenges. Most of the theory will be developed for linear systems, where a wealth of powerful techniques exist [165, 492]. This theory will then be demonstrated on simple and intuitive examples, such as to develop a cruise controller for an automobile or stabilize an inverted pendulum on a moving cart.

Types of Control

There are many ways to manipulate the behavior of a dynamical system, and these control approaches are organized schematically in Fig. 8.1. Passive control does not require input energy, and when sufficient, it is desirable because of its simplicity, reliability, and low cost. For example, stop signs at a traffic intersection regulate the flow of traffic. Active control requires input energy, and these controllers are divided into two broad categories based on whether or not sensors are used to inform the controller. In the first category, open-loop

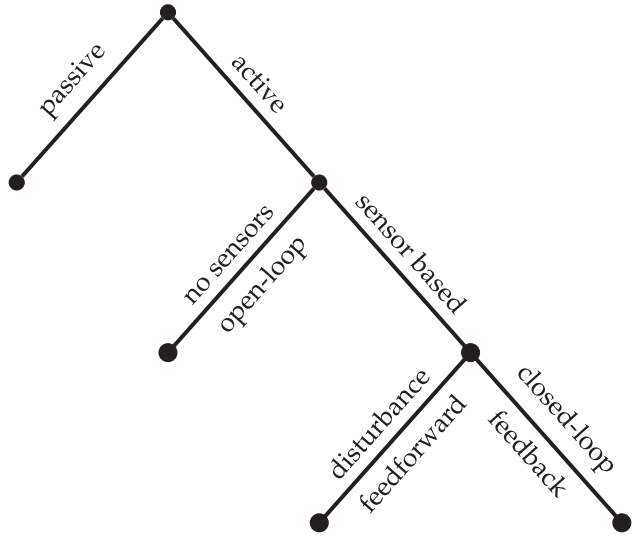

Figure 8.1 Schematic illustrating the various types of control. Most of this chapter will focus on closed-loop feedback control.

control relies on a pre-programmed control sequence; in the traffic example, signals may be pre-programmed to regulate traffic dynamically at different times of day. In the second category, active control uses sensors to inform the control law. Disturbance feedforward control measures exogenous disturbances to the system and then feeds this into an open-loop control law; an example of feedforward control would be to preemptively change the direction of the flow of traffic near a stadium when a large crowd of people are expected to leave. Finally, the last category is closed-loop feedback control, which will be the main focus of this chapter. Closed-loop control uses sensors to measure the system directly and then shapes the control in response to whether the system is actually achieving the desired goal. Many modern traffic systems have smart traffic lights with a control logic informed by inductive sensors in the roadbed that measure traffic density.

8.1 Closed-Loop Feedback Control

The main focus of this chapter is closed-loop feedback control, which is the method of choice for systems with uncertainty, instability, and/or external disturbances. Fig. 8.2 depicts the general feedback control framework, where sensor measurements, $\mathbf{y}$, of a system are fed back into a controller, which then decides on an actuation signal, $\mathbf{u}$, to manipulate the dynamics and provide robust performance despite model uncertainty and exogenous disturbances. In all of the examples discussed in this chapter, the vector of exogenous disturbances may be decomposed as $\mathbf{w} = \begin{bmatrix}\mathbf{w}_d^T & \mathbf{w}_n^T & \mathbf{w}_r^T\end{bmatrix}^T$, where $\mathbf{w}_d$ are disturbances to the state of the system, $\mathbf{w}_n$ is measurement noise, and $\mathbf{w}_r$ is a reference trajectory that should be tracked by the closed-loop system.

Mathematically, the system and measurements are typically described by a dynamical system:

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}_d) \tag{8.1a}$$

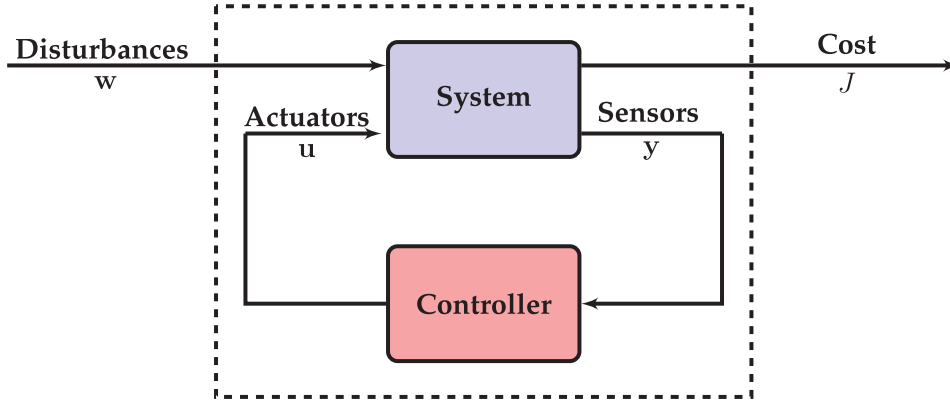

Figure 8.2 Standard framework for feedback control. Measurements of the system, $\mathbf{y}(t)$, are fed back into a controller, which then decides on the appropriate actuation signal $\mathbf{u}(t)$ to control the system. The control law is designed to modify the system dynamics and provide good performance, quantified by the cost J, despite exogenous disturbances and noise in $\mathbf{w}$. The exogenous input $\mathbf{w}$ may also include a reference trajectory $\mathbf{w}_r$ that should be tracked.

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{w}_n). \tag{8.1b}$$

The goal is to construct a control law

$$\mathbf{u} = \mathbf{k}(\mathbf{y}, \mathbf{w}_r) \tag{8.2}$$

that minimizes a cost function

$$J \triangleq J(\mathbf{x}, \mathbf{u}, \mathbf{w}_r). \tag{8.3}$$

Thus, modern control relies heavily on techniques from optimization [74]. In general, the controller in (8.2) will be a dynamical system, rather than a static function of the inputs. For example, the Kalman filter in Section 8.5 dynamically estimates the full state $\mathbf{x}$ from measurements of $\mathbf{u}$ and $\mathbf{y}$. In this case, the control law will become $\mathbf{u} = \mathbf{k}(\mathbf{y}, \hat{\mathbf{x}}, \mathbf{w}_r)$, where $\hat{\mathbf{x}}$ is the full-state estimate.

To motivate the added cost and complexity of sensor-based feedback control, it is helpful to compare with open-loop control. For reference tracking problems, the controller is designed to steer the output of a system towards a desired reference output value $\mathbf{w}_r$, thus minimizing the error $\epsilon = \mathbf{y} - \mathbf{w}_r$. Open-loop control, shown in Fig. 8.3, uses a model of the system to design an actuation signal $\mathbf{u}$ that produces the desired reference output. However, this pre-planned strategy cannot correct for external disturbances to the system and is fundamentally incapable of changing the dynamics. Thus, it is impossible to stabilize an unstable system, such as an inverted pendulum, with open-loop control, since the system model would have to be known perfectly and the system would need to be perfectly isolated from disturbances. Moreover, any model uncertainty will directly contribute to open-loop tracking error.

In contrast, closed-loop feedback control, shown in Fig. 8.4 uses sensor measurements of the system to inform the controller about how the system is actually responding. These sensor measurements provide information about unmodeled dynamics and disturbances that would degrade the performance in open-loop control. Further, with feedback it is

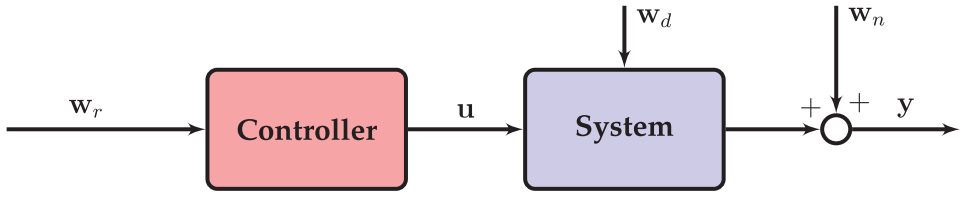

Figure 8.3 Open-loop control diagram. Given a desired reference signal $\mathbf{w}_r$, the open-loop control law constructs a control protocol $\mathbf{u}$ to drive the system based on a model. External disturbances ($\mathbf{w}_d$) and sensor noise ($\mathbf{w}_n$), as well as unmodeled system dynamics and uncertainty, are not accounted for and degrade performance.

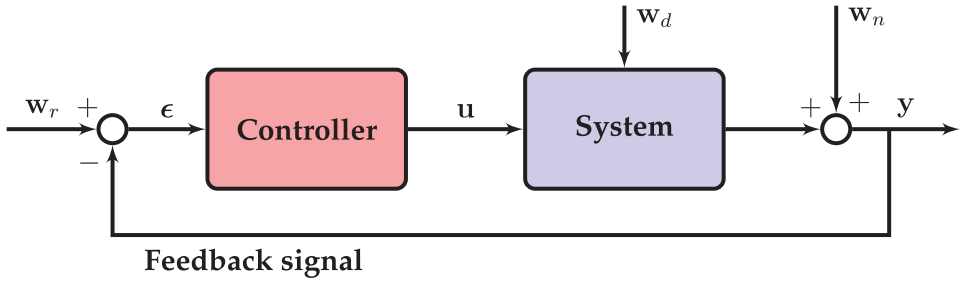

Figure 8.4 Closed-loop feedback control diagram. The sensor signal $\mathbf{y}$ is fed back and subtracted from the reference signal $\mathbf{w}_r$, providing information about how the system is responding to actuation and external disturbances. The controller uses the resulting error $\boldsymbol{\epsilon}$ to determine the correct actuation signal $\mathbf{u}$ for the desired response. Feedback is often able to stabilize unstable dynamics while effectively rejecting disturbances $\mathbf{w}_d$ and attenuating noise $\mathbf{w}_n$.

often possible to modify and stabilize the dynamics of the closed-loop system, something which is not possible with open-loop control. Thus, closed-loop feedback control is often able to maintain high-performance operation for systems with unstable dynamics, model uncertainty, and external disturbances.

Examples of the Benefits of Feedback Control

To summarize, closed-loop feedback control has several benefits over open-loop control:

- It may be possible to stabilize an unstable system;
- It may be possible to compensate for external disturbances;
- It may be possible to correct for unmodeled dynamics and model uncertainty.

These issues are illustrated in the following two simple examples.

Inverted pendulum Consider the unstable inverted pendulum equations, which will be derived later in Section 8.2. The linearized equations are:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ g/L & d \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{8.4}$$

where $x_1 = \theta$, $x_2 = \dot{\theta}$, u is a torque applied to the pendulum arm, g is gravitational acceleration, L is the length of the pendulum arm, and d is damping. We may write this system in standard form as

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}u.$$

If we choose constants so that the natural frequency is $\omega_n = \sqrt{g/L} = 1$ and $d = 0$, then the system has eigenvalues $\lambda = \pm 1$, corresponding to an unstable saddle-type fixed point.

No open-loop control strategy can change the dynamics of the system, given by the eigenvalues of $\mathbf{A}$. However, with full-state feedback control, given by $u = -\mathbf{K}\mathbf{x}$, the closed-loop system becomes

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}u = (\mathbf{A} - \mathbf{B}\mathbf{K})\,\mathbf{x}.$$

Choosing $\mathbf{K} = \begin{bmatrix} 4 & 4 \end{bmatrix}$, corresponding to a control law $u = -4x_1 - 4x_2 = -4\theta - 4\dot{\theta}$, the closed loop system $(\mathbf{A} - \mathbf{B}\mathbf{K})$ has stable eigenvalues $\lambda = -1$ and $\lambda = -3$.

Determining when it is possible to change the eigenvalues of the closed-loop system, and determining the appropriate control law $\mathbf{K}$ to achieve this, will be the subject of future sections.

Cruise control To appreciate the ability of closed-loop control to compensate for unmodeled dynamics and disturbances, we will consider a simple model of cruise control in an automobile. Let u be the rate of gas fed into the engine, and let y be the car's speed. Neglecting transients, a crude model[1] is:

$$y = u. \tag{8.5}$$

Thus, if we double the gas input, we double the automobile's speed.

Based on this model, we may design an open-loop cruise controller to track a reference speed w_r by simply commanding an input of $u = w_r$. However, an incorrect automobile model (i.e., in actuality $y = 2u$), or external disturbances, such as rolling hills (i.e., if $y = u + \sin(t)$), are not accounted for in the simple open-loop design.

In contrast, a closed-loop control law, based on measurements of the speed, is able to compensate for unmodeled dynamics and disturbances. Consider the closed-loop control law $u = K(w_r - y)$, so that gas is increased when the measured velocity is too low, and decreased when it is too high. Then if the dynamics are actually $y = 2u$ instead of $y = u$, the open-loop system will have 50% steady-state tracking error, while the performance of the closed-loop system can be significantly improved for large K:

$$y = 2K(w_r - y) \quad \Longrightarrow \quad (1 + 2K)y = 2Kw_r \quad \Longrightarrow \quad y = \frac{2K}{1 + 2K}w_r. \tag{8.6}$$

For $K = 50$, the closed-loop system only has 1% steady-state tracking error. Similarly, an added disturbance w_d will be attenuated by a factor of $1/(2K + 1)$.

As a concrete example, consider a reference tracking problem with a desired reference speed of 60 mph. The model is $y = u$, and the true system is $y = 0.5\,u$. In addition, there is a disturbance in the form of rolling hills that increase and decrease the speed by ± 10 mph at a frequency of 0.5 Hz. An open-loop controller is compared with a closed-loop proportional controller with $K = 50$ in Fig. 8.5 and Code 8.1. Although the closed-loop controller has significantly better performance, we will see later that a large proportional gain may come at the cost of robustness. Adding an integral term will improve performance.

[1] A more realistic model would have acceleration dynamics, so that $\dot{x} = -x + u$ and $y = x$.

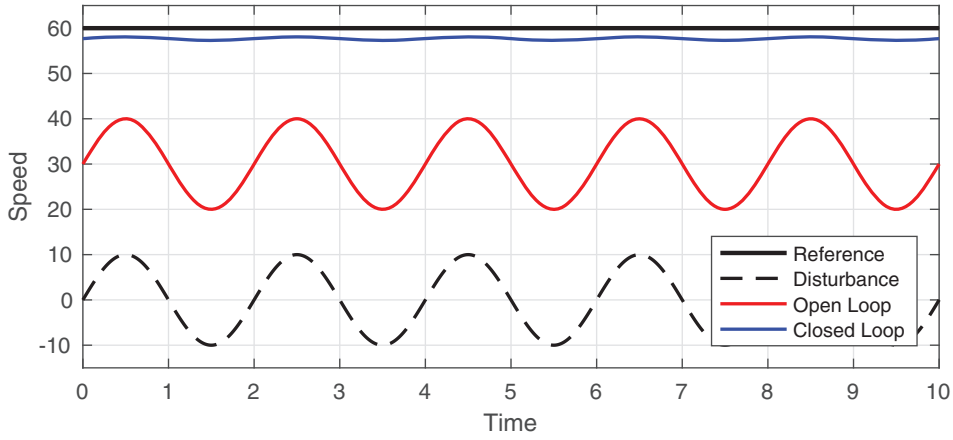

Figure 8.5 Open-loop vs. closed-loop cruise control.

Code 8.1 Compare open-loop and closed-loop cruise control.

```
clear all, close all, clc

t = 0:.01:10;           % time

wr = 60*ones(size(t)); % reference speed
d = 10*sin(pi*t);       % disturbance

aModel = 1;             % y = aModel*u
aTrue = .5;             % y = aTrue*u

uOL = wr/aModel;        % Open-loop u based on model
yOL = aTrue*uOL + d;    % Open-loop response

K = 50;                 % control gain, u=K(wr-y);
yCL = aTrue*K/(1+aTrue*K)*wr + d/(1+aTrue*K);
```

8.2 Linear Time-Invariant Systems

The most complete theory of control has been developed for linear systems [492, 165, 22]. Linear systems are generally obtained by linearizing a nonlinear system about a fixed point or a periodic orbit. However, instability may quickly take a trajectory far away from the fixed point. Fortunately, an effective stabilizing controller will keep the state of the system in a small neighborhood of the fixed point where the linear approximation is valid. For example, in the case of the inverted pendulum, feedback control may keep the pendulum stabilized in the vertical position where the dynamics behave linearly.

Linearization of Nonlinear Dynamics

Given a nonlinear input–output system

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{8.7a}$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \tag{8.7b}$$

it is possible to linearize the dynamics near a fixed point $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ where $\mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \mathbf{0}$. For small $\Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$ and $\Delta\mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$ the dynamics $\mathbf{f}$ may be expanded in a Taylor series about the point $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$:

$$\mathbf{f}(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \underbrace{\left.\frac{\mathbf{df}}{\mathbf{dx}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{A}} \cdot \Delta\mathbf{x} + \underbrace{\left.\frac{\mathbf{df}}{\mathbf{du}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{B}} \cdot \Delta\mathbf{u} + \cdots . \tag{8.8}$$

Similarly, the output equation $\mathbf{g}$ may be expanded as:

$$\mathbf{g}(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) = \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \underbrace{\left.\frac{\mathbf{dg}}{\mathbf{dx}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{C}} \cdot \Delta\mathbf{x} + \underbrace{\left.\frac{\mathbf{dg}}{\mathbf{du}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{D}} \cdot \Delta\mathbf{u} + \cdots . \tag{8.9}$$

For small displacements around the fixed point, the higher order terms are negligibly small. Dropping the Δ and shifting to a coordinate system where $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$, and $\bar{\mathbf{y}}$ are at the origin, the linearized dynamics may be written as:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{8.10a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \tag{8.10b}$$

Note that we have neglected the disturbance and noise inputs, $\mathbf{w}_d$ and $\mathbf{w}_n$, respectively; these will be added back in the discussion on Kalman filtering in Section 8.5.

Unforced Linear System

In the absence of control (i.e., $\mathbf{u} = \mathbf{0}$), and with measurements of the full state (i.e., $\mathbf{y} = \mathbf{x}$), the dynamical system in (8.10) becomes

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x}. \tag{8.11}$$

The solution $\mathbf{x}(t)$ is given by

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0), \tag{8.12}$$

where the matrix exponential is defined by:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2t^2}{2} + \frac{\mathbf{A}^3t^3}{3} + \cdots . \tag{8.13}$$

The solution in (8.12) is determined entirely by the eigenvalues and eigenvectors of the matrix $\mathbf{A}$. Consider the eigendecomposition of $\mathbf{A}$:

$$\mathbf{A}\mathbf{T} = \mathbf{T}\boldsymbol{\Lambda}. \tag{8.14}$$

In the simplest case, $\boldsymbol{\Lambda}$ is a diagonal matrix of distinct eigenvalues and $\mathbf{T}$ is a matrix whose columns are the corresponding linearly independent eigenvectors of $\mathbf{A}$. For repeated eigenvalues, $\boldsymbol{\Lambda}$ may be written in Jordan form, with entries above the diagonal for degenerate eigenvalues of multiplicity ≥ 2; the corresponding columns of $\mathbf{T}$ will be generalized eigenvectors.

In either case, it is easier to compute the matrix exponential $e^{\mathbf{\Lambda} t}$ than $e^{\mathbf{A}t}$. For diagonal $\mathbf{\Lambda}$, the matrix exponential is given by:

$$e^{\mathbf{\Lambda} t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix}. \tag{8.15}$$

In the case of a nontrivial Jordan block in $\mathbf{\Lambda}$ with entries above the diagonal, simple extensions exist related to nilpotent matrices (for details, see Perko [427]).

Rearranging the terms in (8.14), we find that it is simple to represent powers of $\mathbf{A}$ in terms of the eigenvectors and eigenvalues:

$$\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1} \tag{8.16a}$$

$$\mathbf{A}^2 = \left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right)\left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right) = \mathbf{T}\mathbf{\Lambda}^2\mathbf{T}^{-1} \tag{8.16b}$$

$$\cdots$$

$$\mathbf{A}^k = \left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right)\left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right)\cdots\left(\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}\right) = \mathbf{T}\mathbf{\Lambda}^k\mathbf{T}^{-1}. \tag{8.16c}$$

Finally, substituting these expressions into (8.13) yields:

$$e^{\mathbf{A}t} = e^{\mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}t} = \mathbf{T}\mathbf{T}^{-1} + \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}t + \frac{\mathbf{T}\mathbf{\Lambda}^2\mathbf{T}^{-1}t^2}{2} + \frac{\mathbf{T}\mathbf{\Lambda}^3\mathbf{T}^{-1}t^3}{3} + \cdots \tag{8.17a}$$

$$= \mathbf{T}\left[\mathbf{I} + \mathbf{\Lambda}t + \frac{\mathbf{\Lambda}^2t^2}{2} + \frac{\mathbf{\Lambda}^3t^3}{3} + \cdots\right]\mathbf{T}^{-1} \tag{8.17b}$$

$$= \mathbf{T}e^{\mathbf{\Lambda}t}\mathbf{T}^{-1}. \tag{8.17c}$$

Thus, we see that it is possible to compute the matrix exponential efficiently in terms of the eigendecomposition of $\mathbf{A}$. Moreover, the matrix of eigenvectors $\mathbf{T}$ defines a change of coordinates that dramatically simplifies the dynamics:

$$\mathbf{x} = \mathbf{T}\mathbf{z} \quad \Longrightarrow \quad \dot{\mathbf{z}} = \mathbf{T}^{-1}\dot{\mathbf{x}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{x} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z} \quad \Longrightarrow \quad \dot{\mathbf{z}} = \mathbf{\Lambda}\mathbf{z}. \tag{8.18}$$

In other words, changing to eigenvector coordinates, the dynamics become diagonal. Combining (8.12) with (8.17c), it is possible to write the solution $\mathbf{x}(t)$ as

$$\mathbf{x}(t) = \underbrace{\mathbf{T}\underbrace{e^{\mathbf{\Lambda}t}\underbrace{\mathbf{T}^{-1}\mathbf{x}(0)}_{\mathbf{z}(0)}}_{\mathbf{z}(t)}}_{\mathbf{x}(t)}. \tag{8.19}$$

In the first step, $\mathbf{T}^{-1}$ maps the initial condition in physical coordinates, $\mathbf{x}(0)$, into eigenvector coordinates, $\mathbf{z}(0)$. The next step advances these initial conditions using the diagonal update $e^{\mathbf{\Lambda}t}$, which is considerably simpler in eigenvector coordinates $\mathbf{z}$. Finally, multiplying by $\mathbf{T}$ maps $\mathbf{z}(t)$ back to physical coordinates, $\mathbf{x}(t)$.

In addition to making it possible to compute the matrix exponential, and hence the solution $\mathbf{x}(t)$, the eigendecomposition of $\mathbf{A}$ is even more useful to understand the dynamics and stability of the system. We see from (8.19) that the only time-varying portion of the

solution is $e^{\mathbf{A}t}$. In general, these eigenvalues $\lambda = a + ib$ may be complex numbers, so that the solutions are given by $e^{\lambda t} = e^{at}(\cos(bt) + i\sin(bt))$. Thus, if all of the eigenvalues λ_k have negative real part (i.e., $\text{Re}(\lambda) = a < 0$), then the system is stable, and solutions all decay to $\mathbf{x} = \mathbf{0}$ as $t \to \infty$. However, if even a single eigenvalue has positive real part, then the system is unstable and will diverge from the fixed point along the corresponding unstable eigenvector direction. Any random initial condition is likely to have a component in this unstable direction, and moreover, disturbances will likely excite all eigenvectors of the system.

Forced Linear System

With forcing, and for zero initial condition, $\mathbf{x}(0) = \mathbf{0}$, the solution to (8.10a) is

$$\mathbf{x}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \triangleq e^{\mathbf{A}t}\mathbf{B} * \mathbf{u}(t). \tag{8.20}$$

The control input $\mathbf{u}(t)$ is convolved with the kernel $e^{\mathbf{A}t}\mathbf{B}$. With an output $\mathbf{y} = \mathbf{C}\mathbf{x}$, we have $\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B} * \mathbf{u}(t)$. This convolution is illustrated in Fig. 8.6 for a single-input, single-output (SISO) system in terms of the impulse response $g(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B} = \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}\delta(\tau)\,d\tau$ given a Dirac delta input $u(t) = \delta(t)$.

Discrete-Time Systems

In many real-world applications, systems are sampled at discrete instances in time. Thus, digital control systems are typically formulated in terms of discrete-time dynamical systems:

$$\mathbf{x}_{k+1} = \mathbf{A}_d\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k \tag{8.21a}$$

$$\mathbf{y}_k = \mathbf{C}_d\mathbf{x}_k + \mathbf{D}_d\mathbf{u}_k, \tag{8.21b}$$

where $\mathbf{x}_k = \mathbf{x}(k\Delta t)$. The system matrices in (8.21) can be obtained from the continuous-time system in (8.10) as

$$\mathbf{A}_d = e^{\mathbf{A}\Delta t} \tag{8.22a}$$

$$\mathbf{B}_d = \int_0^{\Delta t} e^{\mathbf{A}\tau}\mathbf{B}\,d\tau \tag{8.22b}$$

$$\mathbf{C}_d = \mathbf{C} \tag{8.22c}$$

$$\mathbf{D}_d = \mathbf{D}. \tag{8.22d}$$

The stability of the discrete-time system in (8.21) is still determined by the eigenvalues of $\mathbf{A}_d$, although now a system is stable if and only if all discrete-time eigenvalues are inside the unit circle in the complex plane. Thus, $\exp(\mathbf{A}\Delta t)$ defines a conformal mapping on the complex plane from continuous-time to discrete-time, where eigenvalues in the left-half plane map to eigenvalues inside the unit circle.

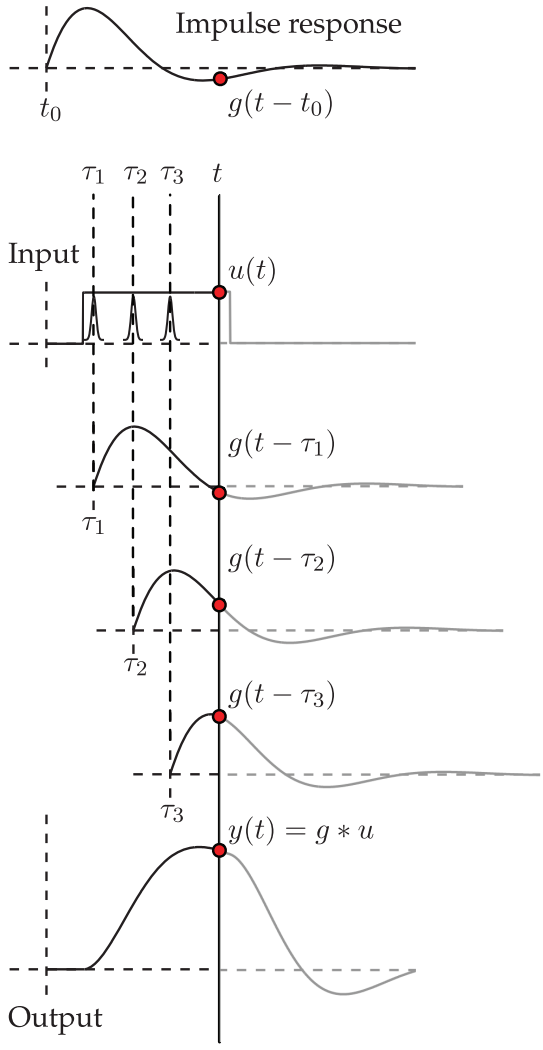

Figure 8.6 Convolution for a single-input, single-output (SISO) system.

Example: Inverted Pendulum

Consider the inverted pendulum in Fig. 8.8 with a torque input u at the base. The equation of motion, derived using the Euler–Lagrange equations[2], is:

$$\ddot{\theta} = -\frac{g}{L}\sin(\theta) + u. \tag{8.23}$$

Introducing the state $\mathbf{x}$, given by the angular position and velocity, we can write this second order differential equation as a system of first order equations:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \qquad \Longrightarrow \qquad \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{L}\sin(x_1) + u \end{bmatrix}. \tag{8.24}$$

[2] The Lagrangian is $\mathcal{L} = \frac{m}{2}L^2\dot{\theta}^2 - mgL\cos(\theta)$, and the Euler–Lagrange equation is $\frac{d}{dt}\partial\mathcal{L}/\partial\dot{\theta} - \partial\mathcal{L}/\partial\theta = \tau$, where τ is the input torque.

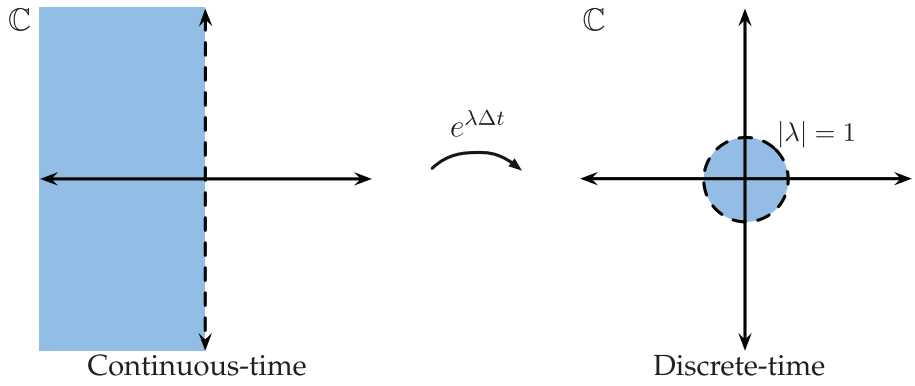

Figure 8.7 The matrix exponential defines a conformal map on the complex plane, mapping stable eigenvalues in the left half plane into eigenvalues inside the unit circle.

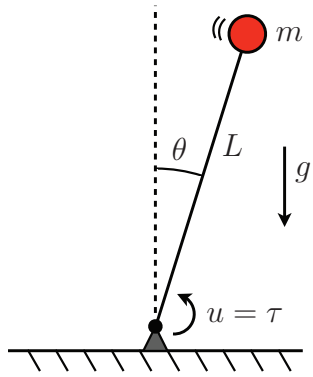

Figure 8.8 Schematic of inverted pendulum system.

Taking the Jacobian of $\mathbf{f}(\mathbf{x}, \mathbf{u})$ yields

$$\frac{\mathbf{df}}{\mathbf{dx}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\cos(x_1) & 0 \end{bmatrix}, \qquad \frac{\mathbf{df}}{\mathbf{du}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{8.25}$$

Linearizing at the pendulum up ($x_1 = \pi$, $x_2 = 0$) and down ($x_1 = 0$, $x_2 = 0$) equilibria gives

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{Pendulum up,} \quad \lambda = \pm\sqrt{g/L}} \qquad \underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{Pendulum down,} \quad \lambda = \pm i\sqrt{g/L}}.$$

Thus, we see that the down position is a stable center with eigenvalues $\lambda = \pm i\sqrt{g/L}$ corresponding to oscillations at a natural frequency of $\sqrt{g/L}$. The pendulum up position is an unstable saddle with eigenvalues $\lambda = \pm\sqrt{g/L}$.

8.3 Controllability and Observability

A natural question arises in linear control theory: To what extent can closed-loop feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$ manipulate the behavior of the system in (8.10a)? We already saw in Section 8.1 that it was possible to modify the eigenvalues of the unstable inverted pendulum system via closed-loop feedback, resulting in a new system matrix $(\mathbf{A} - \mathbf{B}\mathbf{K})$ with stable eigenvalues. This section will provide concrete conditions on when and how the system dynamics may be manipulated through feedback control. The dual question, of when it is possible to estimate the full state $\mathbf{x}$ from measurements $\mathbf{y}$, will also be addressed.

Controllability

The ability to design the eigenvalues of the closed-loop system with the choice of $\mathbf{K}$ relies on the system in (8.10a) being *controllable*. The controllability of a linear system is determined entirely by the column space of the *controllability* matrix $\mathcal{C}$:

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}. \tag{8.26}$$

If the matrix $\mathcal{C}$ has n linearly independent columns, so that it spans all of $\mathbb{R}^n$, then the system in (8.10a) is controllable. The span of the columns of the controllability matrix $\mathcal{C}$ forms a Krylov subspace that determines which state vector directions in $\mathbb{R}^n$ may be manipulated with control. Thus, in addition to controllability implying arbitrary eigenvalue placement, it also implies that any state $\boldsymbol{\xi} \in \mathbb{R}^n$ is reachable in a finite time with some actuation signal $\mathbf{u}(t)$.

The following three conditions are equivalent:

1. *Controllability*. The span of $\mathcal{C}$ is $\mathbb{R}^n$. The matrix $\mathcal{C}$ may be generated by

```
>> ctrb(A,B)
```

and the rank may be tested to see if it is equal to n, by

```
>> rank(ctrb(A,B))
```

2. *Arbitrary eigenvalue placement*. It is possible to design the eigenvalues of the closed-loop system through choice of feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = (\mathbf{A} - \mathbf{B}\mathbf{K})\,\mathbf{x}. \tag{8.27}$$

Given a set of desired eigenvalues, the gain $\mathbf{K}$ can be determined by

```
>> K = place(A,B,neweigs);
```

Designing $\mathbf{K}$ for the best performance will be discussed in Section 8.4.

3. *Reachability of* $\mathbb{R}^n$. It is possible to steer the system to any arbitrary state $\mathbf{x}(t) = \boldsymbol{\xi} \in \mathbb{R}^n$ in a finite time with some actuation signal $\mathbf{u}(t)$.

Note that reachability also applies to open-loop systems. In particular, if a direction $\boldsymbol{\xi}$ is not in the span of $\mathcal{C}$, then it is impossible for control to push in this direction in either open-loop or closed-loop.

Examples The notion of controllability is more easily understood by investigating a few simple examples. First, consider the following system

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad \Longrightarrow \qquad \mathcal{C} = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix}. \tag{8.28}$$

This system is not controllable, because the controllability matrix $\mathcal{C}$ consists of two linearly dependent vectors and does not span $\mathbb{R}^2$. Even before checking the rank of the controllability matrix, it is easy to see that the system won't be controllable since the states x_1 and x_2 are completely decoupled and the actuation input u only effects the second state.

Modifying this example to include two actuation inputs makes the system controllable by increasing the control authority:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \Longrightarrow \qquad \mathcal{C} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}. \tag{8.29}$$

This *fully actuated* system is clearly controllable because x_1 and x_2 may be independently controlled with u_1 and u_2. The controllability of this system is confirmed by checking that the columns of $\mathcal{C}$ do span $\mathbb{R}^2$.

The most interesting cases are less obvious than these two examples. Consider the system

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad \Longrightarrow \qquad \mathcal{C} = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}. \tag{8.30}$$

This two-state system is controllable with a single actuation input because the states x_1 and x_2 are now coupled through the dynamics. Similarly,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \qquad \Longrightarrow \qquad \mathcal{C} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}. \tag{8.31}$$

is controllable even though the dynamics of x_1 and x_2 are decoupled, because the actuator $\mathbf{B} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ is able to simultaneously affect both states and they have different timescales.

We will see in Section 8.3 that controllability is intimately related to the alignment of the columns of $\mathbf{B}$ with the eigenvector directions of $\mathbf{A}$.

Observability

Mathematically, observability of the system in (8.10) is nearly identical to controllability, although the physical interpretation differs somewhat. A system is *observable* if it is possible to estimate any state $\boldsymbol{\xi} \in \mathbb{R}^n$ from a time-history of the measurements $\mathbf{y}(t)$.

Again, the observability of a system is entirely determined by the row space of the *observability* matrix $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}. \tag{8.32}$$

In particular, if the rows of the matrix $\mathcal{O}$ span $\mathbb{R}^n$, then it is possible to estimate any full-dimensional state $\mathbf{x} \in \mathbb{R}^n$ from the time-history of $\mathbf{y}(t)$. The matrix $\mathcal{O}$ may be generated by

```
>> obsv(A,C);
```

The motivation for full-state estimation is relatively straightforward. We have already seen that with full-state feedback, $\mathbf{u} = -\mathbf{K}\mathbf{x}$, it is possible to modify the behavior of a controllable system. However, if full-state measurements of $\mathbf{x}$ are not available, it is necessary to *estimate* $\mathbf{x}$ from the measurements. This is possible when the system is observable. In Section 8.5, we will see that it is possible to design an observer dynamical system to estimate the full-state from noisy measurements. As in the case of a controllable system, if a system is observable, it is possible to design the eigenvalues of the estimator dynamical system to have desirable characteristics, such as fast estimation and effective noise attenuation.

Interestingly, the observability criterion is mathematically the dual of the controllability criterion. In fact, the observability matrix is the transpose of the controllability matrix for the pair $(\mathbf{A}^T, \mathbf{C}^T)$:

```
>> O = ctrb(A',C')'; % 'obsv' is dual of 'crtb'
```

The PBH Test for Controllability

There are many tests to determine whether or not a system is controllable. One of the most useful and illuminating is the Popov–Belevitch–Hautus (PBH) test. The PBH test states that the pair $(\mathbf{A}, \mathbf{B})$ is controllable if and only if the column rank of the matrix $\begin{bmatrix}(\mathbf{A} - \lambda\mathbf{I}) & \mathbf{B}\end{bmatrix}$ is equal to n for all $\lambda \in \mathbb{C}$. This test is particularly fascinating because it connects controllability[3] to a relationship between the columns of $\mathbf{B}$ and the eigenspace of $\mathbf{A}$.

First, the PBH test only needs to be checked at λ that are eigenvalues of $\mathbf{A}$, since the rank of $\mathbf{A} - \lambda\mathbf{I}$ is equal to n except when λ is an eigenvalue of $\mathbf{A}$. In fact, the characteristic equation $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ is used to determine the eigenvalues of $\mathbf{A}$ as exactly those values where the matrix $\mathbf{A} - \lambda\mathbf{I}$ becomes rank deficient, or degenerate.

Now, given that $(\mathbf{A} - \lambda\mathbf{I})$ is only rank deficient for eigenvalues λ, it also follows that the null-space, or kernel, of $\mathbf{A}-\lambda\mathbf{I}$ is given by the span of the eigenvectors corresponding to that particular eigenvalue. Thus, for $\begin{bmatrix}(\mathbf{A} - \lambda\mathbf{I}) & \mathbf{B}\end{bmatrix}$ to have rank n, the columns in $\mathbf{B}$ must have some component in each of the eigenvector directions associated with $\mathbf{A}$ to complement the null-space of $\mathbf{A} - \lambda\mathbf{I}$.

If $\mathbf{A}$ has n distinct eigenvalues, then the system will be controllable with a single actuation input, since the matrix $\mathbf{A} - \lambda\mathbf{I}$ will have at most one eigenvector direction in the null-space. In particular, we may choose $\mathbf{B}$ as the sum of all of the n linearly independent eigenvectors, and it will be guaranteed to have some component in each direction. It is also interesting to note that if $\mathbf{B}$ is a random vector (>>**B=randn(n,1);**), then $(\mathbf{A}, \mathbf{B})$ will be controllable with high probability, since it will be exceedingly unlikely that $\mathbf{B}$ will be randomly chosen so that it has zero contribution from any given eigenvector.

If there are degenerate eigenvalues with multiplicity ≥ 2, so that the null-space of $\mathbf{A}-\lambda\mathbf{I}$ is multidimensional, then the actuation input must have as many degrees of freedom. In other words, the only time that multiple actuators (columns of $\mathbf{B}$) are strictly required is for

[3] There is an equivalent PBH test for observability that states that $\begin{bmatrix}(\mathbf{A} - \lambda\mathbf{I}) \\ \mathbf{C}\end{bmatrix}$ must have row rank n for all $\lambda \in \mathbb{C}$ for the system to be observable.

systems that have degenerate eigenvalues. However, if a system is highly nonnormal, it may helpful to have multiple actuators in practice for better control authority. Such nonnormal systems are characterized by large transient growth due to destructive interference between nearly parallel eigenvectors, often with similar eigenvalues.

The Cayley–Hamilton Theorem and Reachability

To provide insight into the relationship between the controllability of the pair $(\mathbf{A}, \mathbf{B})$ and the reachability of any vector $\boldsymbol{\xi} \in \mathbb{R}^n$ via the actuation input $\mathbf{u}(t)$, we will leverage the Cayley–Hamilton theorem. This is a gem of linear algebra that provides an elegant way to represent solutions of $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ in terms of a finite sum of powers of $\mathbf{A}$, rather than the infinite sum required for the matrix exponential in (8.13).

The Cayley–Hamilton theorem states that every matrix $\mathbf{A}$ satisfies its own characteristic (eigenvalue) equation, $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{8.33a}$$

$$\implies \quad \mathbf{A}^n + a_{n-1}\mathbf{A}^{n-1} + \cdots + a_2\mathbf{A}^2 + a_1\mathbf{A} + a_0\mathbf{I} = \mathbf{0}. \tag{8.33b}$$

Although this is relatively simple to state, it has profound consequences. In particular, it is possible to express $\mathbf{A}^n$ as a linear combination of smaller powers of $\mathbf{A}$:

$$\mathbf{A}^n = -a_0\mathbf{I} - a_1\mathbf{A} - a_2\mathbf{A}^2 - \cdots - a_{n-1}\mathbf{A}^{n-1}. \tag{8.34}$$

It is straightforward to see that this also implies that any higher power $\mathbf{A}^{k\geq n}$ may also be expressed as a sum of the matrices $\{\mathbf{I}, \mathbf{A}, \cdots, \mathbf{A}^{n-1}\}$:

$$\mathbf{A}^{k\geq n} = \sum_{j=0}^{n-1} \alpha_j \mathbf{A}^j. \tag{8.35}$$

Thus, it is possible to express the infinite sum in the exponential $e^{\mathbf{A}t}$ as:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2t^2}{2} + \cdots \tag{8.36a}$$

$$= \beta_0(t)\mathbf{I} + \beta_1(t)\mathbf{A} + \beta_2(t)\mathbf{A}^2 + \cdots + \beta_{n-1}(t)\mathbf{A}^{n-1}. \tag{8.36b}$$

We are now equipped to see how controllability relates to the reachability of an arbitrary vector $\boldsymbol{\xi} \in \mathbb{R}^n$. From (8.20), we see that a state $\boldsymbol{\xi}$ is reachable if there is some $\mathbf{u}(t)$ so that:

$$\boldsymbol{\xi} = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\, d\tau. \tag{8.37}$$

Expanding the exponential in the right hand side in terms of (8.36b), we have:

$$\begin{aligned}
\boldsymbol{\xi} = &\int_0^t [\beta_0(t-\tau)\mathbf{I}\mathbf{B}\mathbf{u}(\tau) + \beta_1(t-\tau)\mathbf{A}\mathbf{B}\mathbf{u}(\tau) + \cdots \\
&\cdots + \beta_{n-1}(t-\tau)\mathbf{A}^{n-1}\mathbf{B}\mathbf{u}(\tau)]d\tau \\
= &\mathbf{B}\int_0^t \beta_0(t-\tau)\mathbf{u}(\tau)\, d\tau + \mathbf{A}\mathbf{B}\int_0^t \beta_1(t-\tau)\mathbf{u}(\tau)\, d\tau + \cdots \\
&\cdots + \mathbf{A}^{n-1}\mathbf{B}\int_0^t \beta_{n-1}(t-\tau)\mathbf{u}(\tau)\, d\tau
\end{aligned}$$

$$= \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \int_0^t \beta_0(t-\tau)\mathbf{u}(\tau)\,d\tau \\ \int_0^t \beta_1(t-\tau)\mathbf{u}(\tau)\,d\tau \\ \vdots \\ \int_0^t \beta_{n-1}(t-\tau)\mathbf{u}(\tau)\,d\tau \end{bmatrix}.$$

Note that the matrix on the left is the controllability matrix $\mathcal{C}$, and we see that the only way that all of $\mathbb{R}^n$ is reachable is if the column space of $\mathcal{C}$ spans all of $\mathbb{R}^n$. It is somewhat more difficult to see that if $\mathcal{C}$ has rank n then it is possible to design a $\mathbf{u}(t)$ to reach any arbitrary state $\boldsymbol{\xi} \in \mathbb{R}^n$, but this relies on the fact that the n functions $\{\beta_j(t)\}_{j=0}^{n-1}$ are linearly independent functions. It is also the case that there is not a *unique* actuation input $\mathbf{u}(t)$ to reach a given state $\boldsymbol{\xi}$, as there are many different paths one may take.

Gramians and Degrees of Controllability/Observability

The previous tests for controllability and observability are binary, in the sense that the rank of $\mathcal{C}$ (resp. $\mathcal{O}$) is either n, or it isn't. However, there are *degrees* of controllability and observability, as some states $\mathbf{x}$ may be easier to control or estimate than others.

To identify which states are more or less controllable, one must analyze the eigendecomposition of the controllability Gramian:

$$\mathbf{W}_c(t) = \int_0^t e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*\tau}\,d\tau. \tag{8.38}$$

Similarly, the observability Gramian is given by:

$$\mathbf{W}_o(t) = \int_0^t e^{\mathbf{A}^*\tau}\mathbf{C}^*\mathbf{C}e^{\mathbf{A}\tau}\,d\tau. \tag{8.39}$$

These Gramians are often evaluated at infinite time, and unless otherwise stated, we refer to $\mathbf{W}_c = \lim_{t\to\infty} \mathbf{W}_c(t)$ and $\mathbf{W}_o = \lim_{t\to\infty} \mathbf{W}_o(t)$.

The controllability of a state $\mathbf{x}$ is measured by $\mathbf{x}^*\mathbf{W}_c\mathbf{x}$, which will be larger for more controllable states. If the value of $\mathbf{x}^*\mathbf{W}_c\mathbf{x}$ is large, then it is possible to navigate the system far in the $\mathbf{x}$ direction with a unit control input. The observability of a state is similarly measured by $\mathbf{x}^*\mathbf{W}_o\mathbf{x}$. Both Gramians are symmetric and positive semi-definite, having nonnegative eigenvalues. Thus, the eigenvalues and eigenvectors may be ordered hierarchically, with eigenvectors corresponding to large eigenvalues being more easily controllable or observable. In this way, the Gramians induce a new inner-product over state-space in terms of the controllability or observability of the states.

Gramians may be visualized by ellipsoids in state-space, with the principal axes given by directions that are hierarchically ordered in terms of controllability or observability. An example of this visualization is shown in Fig. 9.2 in Chapter 9. In fact, Gramians may be used to design reduced-order models for high-dimensional systems. Through a balancing transformation, a key subspace is identified with the most jointly controllable and observable modes. These modes then define a good projection basis to define a model that captures the dominant input–output dynamics. This form of balanced model reduction will be investigated further in Section 9.2.

Gramians are also useful to determine the minimum-energy control $\mathbf{u}(t)$ required to navigate the system to $\mathbf{x}(t_f)$ at time t_f from $\mathbf{x}(0) = \mathbf{0}$:

$$\mathbf{u}(t) = \mathbf{B}^* \left(e^{\mathbf{A}(t_f - t)} \right)^* \mathbf{W}_c(t_f)^{-1} \mathbf{x}(t_f). \tag{8.40}$$

The total energy expended by this control law is given by

$$\int_0^{t_f} \|\mathbf{u}(\tau)\|^2 \, d\tau = \mathbf{x}^* \mathbf{W}_c(t_f)^{-1} \mathbf{x}. \tag{8.41}$$

It can now be seen that if the controllability matrix is nearly singular, then there are directions that require extreme actuation energy to manipulate. Conversely, if the eigenvalues of $\mathbf{W}_c$ are all large, then the system is easily controlled.

It is generally impractical to compute the Gramians directly using (8.38) and (8.39). Instead, the controllablity Gramian is the solution to the following Lyapunov equation:

$$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathbf{0}, \tag{8.42}$$

while the observability Gramian is the solution to

$$\mathbf{A}^*\mathbf{W}_o + \mathbf{W}_o\mathbf{A} + \mathbf{C}^*\mathbf{C} = \mathbf{0}. \tag{8.43}$$

Obtaining Gramians by solving a Lyapunov equation is typically quite expensive for high-dimensional systems [213, 231, 496, 489, 55]. Instead, Gramians are often approximated empirically using snapshot data from the direct and adjoint systems, as will be discussed in Section 9.2.

Stabilizability and Detectability

In practice, full-state controllability and observability may be too much to expect in high-dimensional systems. For example, in a high-dimensional fluid system, it may be unrealistic to manipulate every minor fluid vortex; instead control authority over the large, energy-containing coherent structures is often enough.

Stabilizability refers to the ability to control all unstable eigenvector directions of $\mathbf{A}$, so that they are in the span of $\mathcal{C}$. In practice, we might relax this definition to include lightly damped eigenvector modes, corresponding to eigenvalues with a small, negative real part. Similarly, if all unstable eigenvectors of $\mathbf{A}$ are in the span of $\mathcal{O}^*$, then the system is detectable.

There may also be states in the model description that are superfluous for control. As an example, consider the control system for a commercial passenger jet. The state of the system may include the passenger seat positions, although this will surely not be controllable by the pilot, nor should it be.

8.4 Optimal Full-State Control: Linear Quadratic Regulator (LQR)

We have seen in the previous sections that if $(\mathbf{A}, \mathbf{B})$ is controllable, then it is possible to arbitrarily manipulate the eigenvalues of the closed-loop system $(\mathbf{A} - \mathbf{B}\mathbf{K})$ through choice of a full-state feedback control law $\mathbf{u} = -\mathbf{K}\mathbf{x}$. This implicitly assumes that full-state measurements are available (i.e., $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{0}$, so that $\mathbf{y} = \mathbf{x}$). Although full-state measurements are not always available, especially for high-dimensional systems, we will

show in the next section that if the system is observable, it is possible to build a full-state estimate from the sensor measurements.

Given a controllable system, and either measurements of the full-state or an observable system with a full-state estimate, there are many choices of stabilizing control laws $\mathbf{u} = -\mathbf{K}\mathbf{x}$. It is possible to make the eigenvalues of the closed-loop system $(\mathbf{A} - \mathbf{B}\mathbf{K})$ arbitrarily stable, placing them as far as desired in the left-half of the complex plane. However, overly stable eigenvalues may require exceedingly expensive control expenditure and might also result in actuation signals that exceed maximum allowable values. Choosing very stable eigenvalues may also cause the control system to over-react to noise and disturbances, much as a new driver will over-react to vibrations in the steering wheel, causing the closed-loop system to jitter. Over stabilization can counterintuitively degrade robustness and may lead to instability if there are small time delays or unmodeled dynamics. Robustness will be discussed in Section 8.8.

Choosing the best gain matrix $\mathbf{K}$ to stabilize the system without expending too much control effort is an important goal in *optimal* control. A balance must be struck between the stability of the closed-loop system and the aggressiveness of control. It is important to take control expenditure into account 1) to prevent the controller from over-reacting to high-frequency noise and disturbances, 2) so that actuation does not exceed maximum allowed amplitudes, and 3) so that control is not prohibitively expensive. In particular, the cost function

$$J(t) = \int_0^t \mathbf{x}(\tau)^* \mathbf{Q}\mathbf{x}(\tau) + \mathbf{u}(\tau)^* \mathbf{R}\mathbf{u}(\tau)\, d\tau \tag{8.44}$$

balances the cost of effective regulation of the state with the cost of control. The matrices $\mathbf{Q}$ and $\mathbf{R}$ weight the cost of deviations of the state from zero and the cost of actuation, respectively. The matrix $\mathbf{Q}$ is positive semi-definite, and $\mathbf{R}$ is positive definite; these matrices are often diagonal, and the diagonal elements may be tuned to change the relative importance of the control objectives.

Adding such a cost function makes choosing the control law a well-posed optimization problem, for which there is a wealth of theoretical and numerical techniques [74]. The linear-quadratic-regulator (LQR) control law $\mathbf{u} = -\mathbf{K}_r\mathbf{x}$ is designed to minimize $J = \lim_{t\to\infty} J(t)$. LQR is so-named because it is a linear control law, designed for a linear system, minimizing a quadratic cost function, that regulates the state of the system to $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{0}$. Because the cost-function in (8.44) is quadratic, there is an analytical solution for the optimal controller gains $\mathbf{K}_r$, given by

$$\mathbf{K}_r = \mathbf{R}^{-1}\mathbf{B}^*\mathbf{X}, \tag{8.45}$$

where $\mathbf{X}$ is the solution to an algebraic Riccati equation:

$$\mathbf{A}^*\mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{X} + \mathbf{Q} = \mathbf{0}. \tag{8.46}$$

Solving the above Riccati equation for $\mathbf{X}$, and hence for $\mathbf{K}_r$, is numerically robust and already implemented in many programming languages [323, 55]. In Matlab, $\mathbf{K}_r$ is obtained via

```
>> Kr = lqr(A,B,Q,R);
```

However, solving the Riccati equation scales as $\mathcal{O}(n^3)$ in the state-dimension n, making it prohibitively expensive for large systems or for online computations for slowly changing

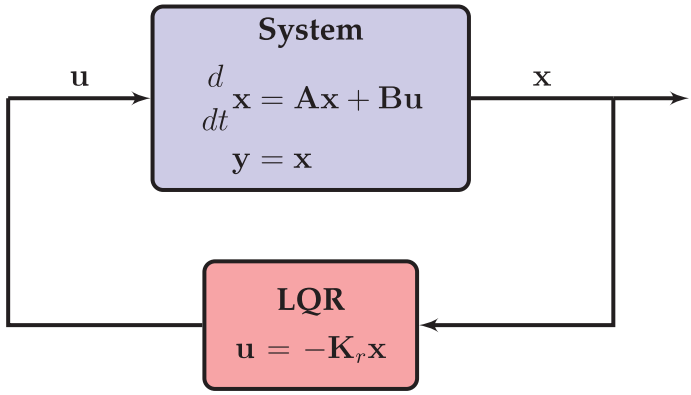

Figure 8.9 Schematic of the linear quadratic regulator (LQR) for optimal full-state feedback. The optimal controller for a linear system given measurements of the full state, $\mathbf{y} = \mathbf{x}$, is given by proportional control $\mathbf{u} = -\mathbf{K}_r\mathbf{x}$ where $\mathbf{K}_r$ is a constant gain matrix obtained by solving an algebraic Riccati equation.

state equations or linear parameter varying (LPV) control. This motivates the development of reduced-order models that capture the same dominant behavior with many fewer states. Control-oriented reduced-order models will be developed more in Chapter 9.

The LQR controller is shown schematically in Fig. 8.9. Out of all possible control laws $\mathbf{u} = \mathbf{K}(\mathbf{x})$, including nonlinear controllers, the LQR controller $\mathbf{u} = -\mathbf{K}_r\mathbf{x}$ is optimal, as we will show in Section 8.4. However, it may be the case that a linearized system is linearly uncontrollable while the full nonlinear system in (8.7) is controllable with a nonlinear control law $\mathbf{u} = \mathbf{K}(\mathbf{x})$.

Derivation of the Riccati Equation for Optimal Control

It is worth taking a theoretical detour here to derive the Riccati equation in (8.46) for the problem of optimal full-state regulation. This derivation will provide an example of how to solve convex optimization problems using the calculus of variations, and it will also provide a template for computing the optimal control solution for *nonlinear* systems. Because of the similarity of optimal control to the formulation of Lagrangian and Hamiltonian classical mechanics in terms of the variational principal, we adopt similar language and notation.

First, we will add a terminal cost to our LQR cost function in (8.44), and also introduce a factor of $1/2$ to simplify computations:

$$J = \int_0^{t_f} \underbrace{\frac{1}{2}\left(\mathbf{x}^*\mathbf{Q}\mathbf{x} + \mathbf{u}^*\mathbf{R}\mathbf{u}\right)}_{\text{Lagrangian, } \mathcal{L}} d\tau + \underbrace{\frac{1}{2}\mathbf{x}(t_f)^*\mathbf{Q}_f\mathbf{x}(t_f)}_{\text{Terminal cost}}. \tag{8.47}$$

The goal is to minimize the quadratic cost function J subject to the dynamical constraint:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{8.48}$$

We may solve this using the calculus of variations by introducing the following augmented cost function

$$J_{\text{aug}} = \int_0^{t_f} \left[\frac{1}{2} \left(\mathbf{x}^* \mathbf{Q} \mathbf{x} + \mathbf{u}^* \mathbf{R} \mathbf{u} \right) + \boldsymbol{\lambda}^* \left(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} - \dot{\mathbf{x}} \right) \right] d\tau + \frac{1}{2} \mathbf{x}(t_f)^* \mathbf{Q}_f \mathbf{x}(t_f). \quad (8.49)$$

The variable $\boldsymbol{\lambda}$ is a Lagrange multiplier, called the *co-state*, that enforces the dynamic constraints. $\boldsymbol{\lambda}$ may take any value and $J_{\text{aug}} = J$ will hold.

Taking the total variation of J_{aug} in (8.49) yields:

$$\delta J_{\text{aug}} = \int_0^{t_f} \left[\frac{\partial \mathcal{L}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathcal{L}}{\partial \mathbf{u}} \delta \mathbf{u} + \boldsymbol{\lambda}^* \mathbf{A} \delta \mathbf{x} + \boldsymbol{\lambda}^* \mathbf{B} \delta \mathbf{u} - \boldsymbol{\lambda}^* \delta \dot{\mathbf{x}} \right] d\tau + \mathbf{Q}_f \mathbf{x}(t_f) \delta \mathbf{x}(t_f). \quad (8.50)$$

The partial derivatives[4] of the Lagrangian are $\partial \mathcal{L} / \partial \mathbf{x} = \mathbf{x}^* \mathbf{Q}$ and $\partial \mathcal{L} / \partial \mathbf{u} = \mathbf{u}^* \mathbf{R}$. The last term in the integral may be modified using integration by parts:

$$-\int_0^{t_f} \boldsymbol{\lambda}^* \delta \dot{\mathbf{x}} \, d\tau = -\boldsymbol{\lambda}^*(t_f) \delta \mathbf{x}(t_f) + \boldsymbol{\lambda}^*(0) \delta \mathbf{x}(0) + \int_0^{t_f} \dot{\boldsymbol{\lambda}}^* \delta \mathbf{x} \, d\tau.$$

The term $\boldsymbol{\lambda}^*(0)\delta\mathbf{x}(0)$ is equal to zero, or else the control system would be non-causal (i.e., then future control could change the initial condition of the system).

Finally, the total variation of the augmented cost function in (8.50) simplifies as follows:

$$\begin{aligned} \delta J_{\text{aug}} = &\int_0^{t_f} \left(\mathbf{x}^* \mathbf{Q} + \boldsymbol{\lambda}^* \mathbf{A} + \dot{\boldsymbol{\lambda}}^* \right) \delta \mathbf{x} \, d\tau + \int_0^{t_f} \left(\mathbf{u}^* \mathbf{R} + \boldsymbol{\lambda}^* \mathbf{B} \right) \delta \mathbf{u} \, d\tau \\ &+ \left(\mathbf{x}(t_f)^* \mathbf{Q}_f - \boldsymbol{\lambda}^*(t_f) \right) \delta \mathbf{x}(t_f). \end{aligned} \quad (8.51)$$

Each variation term in (8.51) must equal zero for an optimal control solution that minimizes J. Thus, we may break this up into three equations:

$$\mathbf{x}^* \mathbf{Q} + \boldsymbol{\lambda}^* \mathbf{A} + \dot{\boldsymbol{\lambda}}^* = \mathbf{0} \quad (8.52a)$$

$$\mathbf{u}^* \mathbf{R} + \boldsymbol{\lambda}^* \mathbf{B} = \mathbf{0} \quad (8.52b)$$

$$\mathbf{x}(t_f)^* \mathbf{Q}_f - \boldsymbol{\lambda}^*(t_f) = \mathbf{0}. \quad (8.52c)$$

Note that the constraint in (8.52c) represents an initial condition for the reverse-time equation for $\boldsymbol{\lambda}$ starting at t_f. Thus, the dynamics in (8.48) with initial condition $\mathbf{x}(0) = \mathbf{x}_0$ and (8.52) with the final-time condition $\boldsymbol{\lambda}(t_f) = \mathbf{Q}_f \mathbf{x}(t_f)$ form a two-point boundary value problem. This may be integrated numerically to find the optimal control solution, even for nonlinear systems.

Because the dynamics are linear, it is possible to *posit* the form $\boldsymbol{\lambda} = \mathbf{P}\mathbf{x}$, and substitute into (8.52) above. The first equation becomes:

$$\left(\dot{\mathbf{P}}\mathbf{x} + \mathbf{P}\dot{\mathbf{x}} \right)^* + \mathbf{x}^* \mathbf{Q} + \boldsymbol{\lambda}^* \mathbf{A} = \mathbf{0}.$$

Taking the transpose, and substituting (8.48) in for $\dot{\mathbf{x}}$, yields:

$$\dot{\mathbf{P}}\mathbf{x} + \mathbf{P}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) + \mathbf{Q}\mathbf{x} + \mathbf{A}^* \mathbf{P}\mathbf{x} = \mathbf{0}.$$

From (8.52b), we have

$$\mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^* \boldsymbol{\lambda} = -\mathbf{R}^{-1} \mathbf{B}^* \mathbf{P}\mathbf{x}.$$

[4] The derivative of a matrix expression $\mathbf{A}\mathbf{x}$ with respect to $\mathbf{x}$ is $\mathbf{A}$, and the derivative of $\mathbf{x}^*\mathbf{A}$ with respect to $\mathbf{x}$ is $\mathbf{A}^*$.

Finally, combining yields:

$$\dot{\mathbf{P}}\mathbf{x} + \mathbf{P}\mathbf{A}\mathbf{x} + \mathbf{A}^*\mathbf{P}\mathbf{x} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{x} = \mathbf{0}. \tag{8.53}$$

This equation must be true for all $\mathbf{x}$, and so it may also be written as a matrix equation. Dropping the terminal cost and letting time go to infinity, the $\dot{\mathbf{P}}$ term disappears, and we recover the algebraic Riccati equation:

$$\mathbf{P}\mathbf{A} + \mathbf{A}\mathbf{P}^* - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^*\mathbf{P} + \mathbf{Q} = \mathbf{0}.$$

Although this procedure is somewhat involved, each step is relatively straightforward. In addition, the dynamics in Eq (8.48) may be replaced with nonlinear dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, and a similar nonlinear two-point boundary value problem may be formulated with $\partial\mathbf{f}/\partial\mathbf{x}$ replacing $\mathbf{A}$ and $\partial\mathbf{f}/\partial\mathbf{u}$ replacing $\mathbf{B}$. This procedure is extremely general, and may be used to numerically obtain nonlinear optimal control trajectories.

Hamiltonian Formulation Similar to the Lagrangian formulation above, it is also possible to solve the optimization problem by introducing the following Hamiltonian:

$$\mathcal{H} = \underbrace{\frac{1}{2}\left(\mathbf{x}^*\mathbf{Q}\mathbf{x} + \mathbf{u}^*\mathbf{R}\mathbf{u}\right)}_{\mathcal{L}} + \boldsymbol{\lambda}^*\left(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\right). \tag{8.54}$$

Then Hamilton's equations become:

$$\dot{\mathbf{x}} = \left(\frac{\partial\mathcal{H}}{\partial\boldsymbol{\lambda}}\right)^* = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad \mathbf{x}(0) = \mathbf{x}_0$$

$$-\dot{\boldsymbol{\lambda}} = \left(\frac{\partial\mathcal{H}}{\partial\mathbf{x}}\right)^* = \mathbf{Q}\mathbf{x} + \mathbf{A}^*\boldsymbol{\lambda} \qquad \boldsymbol{\lambda}(t_f) = \mathbf{Q}_f\mathbf{x}(t_f).$$

Again, this is a two-point boundary value problem in $\mathbf{x}$ and $\boldsymbol{\lambda}$. Plugging in the same expression $\boldsymbol{\lambda} = \mathbf{P}\mathbf{x}$ will result in the same Riccati equation as above.

8.5 Optimal Full-State Estimation: The Kalman Filter

The optimal LQR controller from Section 8.4 relies on full-state measurements of the system. However, full-state measurements may either be prohibitively expensive or technologically infeasible to obtain, especially for high-dimensional systems. The computational burden of collecting and processing full-state measurements may also introduce unacceptable time delays that will limit robust performance.

Instead of measuring the full state $\mathbf{x}$, it may be possible to estimate the state from limited noisy measurements $\mathbf{y}$. In fact, full-state estimation is mathematically possible as long as the pair $(\mathbf{A}, \mathbf{C})$ are observable, although the effectiveness of estimation depends on the degree of observability as quantified by the observability Gramian. The Kalman filter [279, 551, 221] is the most commonly used full-state estimator, as it optimally balances the competing effects of measurement noise, disturbances, and model uncertainty. As will be shown in the next section, it is possible to use the full-state estimate from a Kalman filter in conjunction with the optimal full-state LQR feedback law.

When deriving the optimal full-state estimator, it is necessary to re-introduce disturbances to the state, $\mathbf{w}_d$, and sensor noise, $\mathbf{w}_n$:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w}_d \tag{8.56a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{w}_n. \tag{8.56b}$$

The Kalman filter assumes that both the disturbance and noise are zero-mean Gaussian processes with known covariances:

$$\mathbb{E}\left(\mathbf{w}_d(t)\mathbf{w}_d(\tau)^*\right) = \mathbf{V}_d\delta(t-\tau), \tag{8.57a}$$

$$\mathbb{E}\left(\mathbf{w}_n(t)\mathbf{w}_n(\tau)^*\right) = \mathbf{V}_n\delta(t-\tau). \tag{8.57b}$$

Here $\mathbb{E}$ is the expected value and $\delta(\cdot)$ is the Dirac delta function. The matrices $\mathbf{V}_d$ and $\mathbf{V}_n$ are positive semi-definite with entries containing the covariances of the disturbance and noise terms. Extensions to the Kalman filter exist for correlated, biased, and unknown noise and disturbance terms [498, 372].

It is possible to obtain an estimate $\hat{\mathbf{x}}$ of the full-state $\mathbf{x}$ from measurements of the input $\mathbf{u}$ and output $\mathbf{y}$, via the following estimator dynamical system:

$$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_f\left(\mathbf{y} - \hat{\mathbf{y}}\right) \tag{8.58a}$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}\mathbf{u}. \tag{8.58b}$$

The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are obtained from the system model, and the filter gain $\mathbf{K}_f$ is determined via a similar procedure as in LQR. $\mathbf{K}_f$ is given by

$$\mathbf{K}_f = \mathbf{Y}\mathbf{C}^*\mathbf{V}_n, \tag{8.59}$$

where $\mathbf{y}$ is the solution to another algebraic Riccati equation:

$$\mathbf{Y}\mathbf{A}^* + \mathbf{A}\mathbf{Y} - \mathbf{Y}\mathbf{C}^*\mathbf{V}_n^{-1}\mathbf{C}\mathbf{Y} + \mathbf{V}_d = \mathbf{0}. \tag{8.60}$$

This solution is commonly referred to as the Kalman filter, and it is the optimal full-state estimator with respect to the following cost function:

$$J = \lim_{t\to\infty} \mathbb{E}\left(\left(\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right)^*\left(\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right)\right). \tag{8.61}$$

This cost function implicitly includes the effects of disturbance and noise, which are required to determine the optimal balance between aggressive estimation and noise attenuation. Thus, the Kalman filter is referred to as *linear quadratic estimation* (LQE), and has a dual formulation to the LQR optimization. The cost in (8.61) is computed as an ensemble average over many realizations.

The filter gain $\mathbf{K}_f$ may be determined in Matlab via

```
>> Kf = lqe(A,Vd,C,Vd,Vn); % design Kalman filter gain
```

Optimal control and estimation are mathematical dual problems, as are controllability and observability, so the Kalman filter may also be found using LQR:

```
>> Kf = (lqr(A',C',Vd,Vn))'; % LQR and LQE are dual problems
```

The Kalman filter is shown schematically in Fig. 8.10.

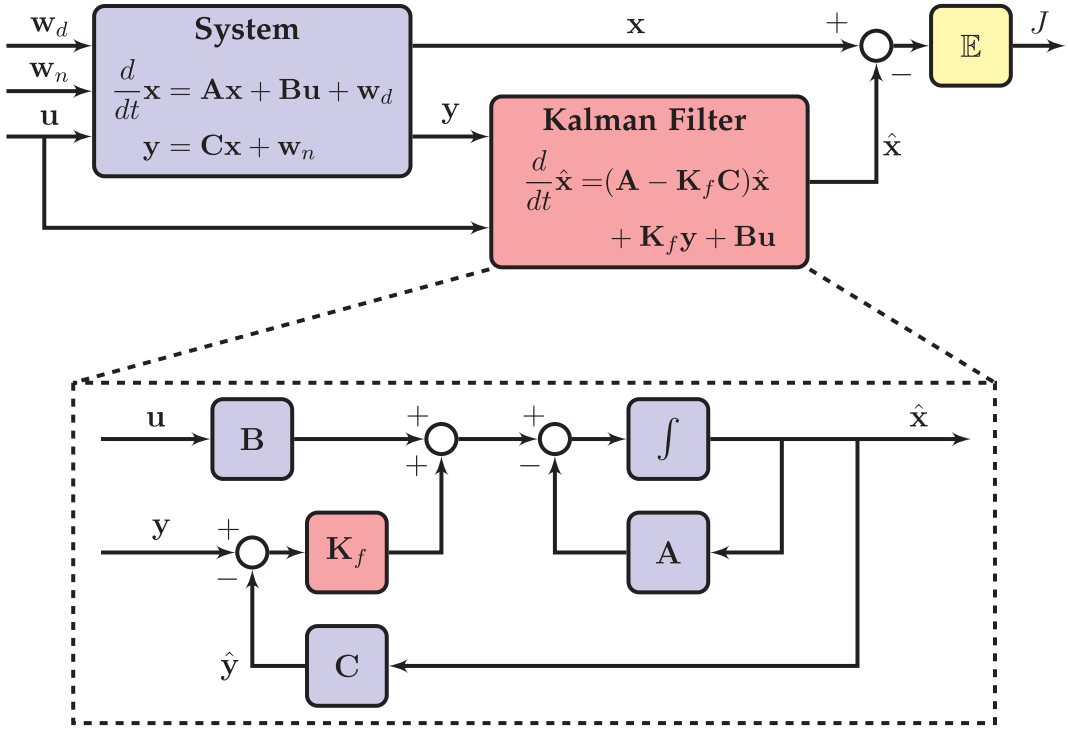

Figure 8.10 Schematic of the Kalman filter for full-state estimation from noisy measurements $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{w}_n$ with process noise (disturbances) $\mathbf{w}_d$. This diagram does not have a feedthrough term $\mathbf{D}$, although it may be included.

Substituting the output estimate $\hat{\mathbf{y}}$ from (8.58b) into (8.58a) yields:

$$\frac{d}{dt}\hat{\mathbf{x}} = \left(\mathbf{A} - \mathbf{K}_f\mathbf{C}\right)\hat{\mathbf{x}} + \mathbf{K}_f\mathbf{y} + \left(\mathbf{B} - \mathbf{K}_f\mathbf{D}\right)\mathbf{u} \tag{8.62a}$$

$$= \left(\mathbf{A} - \mathbf{K}_f\mathbf{C}\right)\hat{\mathbf{x}} + \left[\mathbf{K}_f, \quad \left(\mathbf{B} - \mathbf{K}_f\mathbf{D}\right)\right]\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\end{bmatrix}. \tag{8.62b}$$

The estimator dynamical system is expressed in terms of the estimate $\hat{\mathbf{x}}$ with inputs $\mathbf{y}$ and $\mathbf{u}$. If the system is observable it is possible to place the eigenvalues of $\mathbf{A} - \mathbf{K}_f\mathbf{C}$ arbitrarily with choice of $\mathbf{K}_f$. When the eigenvalues of the estimator are stable, then the state estimate $\hat{\mathbf{x}}$ converges to the full-state $\mathbf{x}$ asymptotically, as long as the model faithfully captures the true system dynamics. To see this convergence, consider the dynamics of the estimation error $\boldsymbol{\epsilon} = \mathbf{x} - \hat{\mathbf{x}}$:

$$\begin{aligned}
\frac{d}{dt}\boldsymbol{\epsilon} &= \frac{d}{dt}\mathbf{x} - \frac{d}{dt}\hat{\mathbf{x}} \\
&= [\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w}_d] - \left[(\mathbf{A} - \mathbf{K}_f\mathbf{C})\hat{\mathbf{x}} + \mathbf{K}_f\mathbf{y} + (\mathbf{B} - \mathbf{K}_f\mathbf{D})\mathbf{u}\right] \\
&= \mathbf{A}\boldsymbol{\epsilon} + \mathbf{w}_d + \mathbf{K}_f\mathbf{C}\hat{\mathbf{x}} - \mathbf{K}_f\mathbf{y} + \mathbf{K}_f\mathbf{D}\mathbf{u} \\
&= \mathbf{A}\boldsymbol{\epsilon} + \mathbf{w}_d + \mathbf{K}_f\mathbf{C}\hat{\mathbf{x}} - \mathbf{K}_f\underbrace{[\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{w}_n]}_{\mathbf{y}} + \mathbf{K}_f\mathbf{D}\mathbf{u} \\
&= (\mathbf{A} - \mathbf{K}_f\mathbf{C})\boldsymbol{\epsilon} + \mathbf{w}_d - \mathbf{K}_f\mathbf{w}_n.
\end{aligned}$$

Therefore, the estimate $\hat{\mathbf{x}}$ will converge to the true full state when $\mathbf{A} - \mathbf{K}_f\mathbf{C}$ has stable eigenvalues. As with LQR, there is a tradeoff between over-stabilization of these eigenvalues and the amplification of sensor noise. This is similar to the behavior of an inexperienced driver who may hold the steering wheel too tightly and will overreact to every minor bump and disturbance on the road.

There are many variants of the Kalman filter for nonlinear systems [274, 275, 538], including the extended and unscented Kalman filters. The ensemble Kalman filter [14] is an extension that works well for high-dimensional systems, such as in geophysical data assimilation [449]. All of these methods still assume Gaussian noise processes, and the particle filter provides a more general, although more computationally intensive, alternative that can handle arbitrary noise distributions [226, 451]. The unscented Kalman filter balances the efficiency of the Kalman filter and accuracy of the particle filter.

8.6 Optimal Sensor-Based Control: Linear Quadratic Gaussian (LQG)

The full-state estimate from the Kalman filter is generally used in conjunction with the full-state feedback control law from LQR, resulting in optimal sensor-based feedback. Remarkably, the LQR gain $\mathbf{K}_r$ and the Kalman filter gain $\mathbf{K}_f$ may be designed separately, and the resulting sensor-based feedback will remain optimal and retain the closed-loop eigenvalues when combined.

Combining the LQR full-state feedback with the Kalman fitler full-state estimator results in the linear-quadratic Gaussian (LQG) controller. The LQG controller is a dynamical system with input $\mathbf{y}$, output $\mathbf{u}$, and internal state $\hat{\mathbf{x}}$:

$$\frac{d}{dt}\hat{\mathbf{x}} = \left(\mathbf{A} - \mathbf{K}_f\mathbf{C} - \mathbf{B}\mathbf{K}_r\right)\hat{\mathbf{x}} + \mathbf{K}_f\mathbf{y} \tag{8.63a}$$

$$\mathbf{u} = -\mathbf{K}_r\hat{\mathbf{x}}. \tag{8.63b}$$

The LQG controller is optimal with respect to the following ensemble-averaged version of the cost function from (8.44):

$$J(t) = \left\langle \int_0^t \left[\mathbf{x}(\tau)^*\mathbf{Q}\mathbf{x}(\tau) + \mathbf{u}(\tau)^*\mathbf{R}\mathbf{u}(\tau)\right] d\tau \right\rangle. \tag{8.64}$$

The controller $\mathbf{u} = -\mathbf{K}_r\hat{\mathbf{x}}$ is in terms of the state estimate, and so this cost function must be averaged over many realizations of the disturbance and noise. Applying LQR to $\hat{\mathbf{x}}$ results in the following state dynamics:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}_r\hat{\mathbf{x}} + \mathbf{w}_d \tag{8.65a}$$

$$= \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}_r\mathbf{x} + \mathbf{B}\mathbf{K}_r\left(\mathbf{x} - \hat{\mathbf{x}}\right) + \mathbf{w}_d \tag{8.65b}$$

$$= \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}_r\mathbf{x} + \mathbf{B}\mathbf{K}_r\boldsymbol{\epsilon} + \mathbf{w}_d. \tag{8.65c}$$

Again $\boldsymbol{\epsilon} = \mathbf{x} - \hat{\mathbf{x}}$ as before. Finally, the closed-loop system may be written as

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \boldsymbol{\epsilon}\end{bmatrix} = \begin{bmatrix}\mathbf{A} - \mathbf{B}\mathbf{K}_r & \mathbf{B}\mathbf{K}_r\\ \mathbf{0} & \mathbf{A} - \mathbf{K}_f\mathbf{C}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \boldsymbol{\epsilon}\end{bmatrix} + \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{I} & -\mathbf{K}_f\end{bmatrix}\begin{bmatrix}\mathbf{w}_d\\ \mathbf{w}_n\end{bmatrix}. \tag{8.66}$$

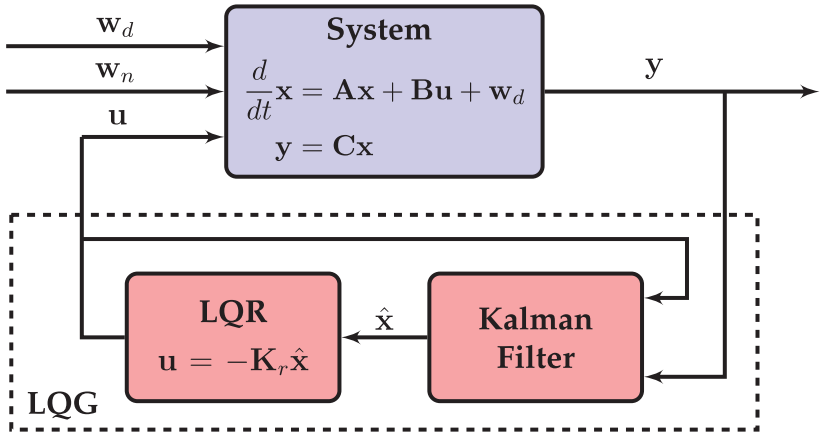

Figure 8.11 Schematic illustrating the linear quadratic Gaussian (LQG) controller for optimal closed-loop feedback based on noisy measurements $\mathbf{y}$. The optimal LQR and Kalman filter gain matrices $\mathbf{K}_r$ and $\mathbf{K}_f$ may be designed independently, based on two different algebraic Riccati equations. When combined, the resulting sensor-based feedback remains optimal.

Thus, the closed-loop eigenvalues of the LQG regulated system are given by the eigenvalues of $\mathbf{A}-\mathbf{B}\mathbf{K}_r$ and $\mathbf{A}-\mathbf{K}_f\mathbf{C}$, which were optimally chosen by the LQR and Kalman filter gain matrices, respectively.

The LQG framework, shown in Fig. 8.11, relies on an accurate model of the system and knowledge of the magnitudes of the disturbances and measurement noise, which are assumed to be Gaussian processes. In real-world systems, each of these assumptions may be invalid, and even small time delays and model uncertainty may destroy the robustness of LQG and result in instability [155]. The lack of robustness of LQG regulators to model uncertainty motivates the introduction of robust control in Section 8.8. For example, it is possible to *robustify* LQG regulators through a process known as loop-transfer recovery. However, despite robustness issues, LQG control is extremely effective for many systems, and is among the most common control paradigms.

In contrast to classical control approaches, such as proportional-integral-derivative (PID) control and designing faster inner-loop control and slow outer-loop control assuming a separation of timescales, LQG is able to handle multiple-input, multiple output (MIMO) systems with overlapping timescales and multi-objective cost functions with no additional complexity in the algorithm or implementation.

8.7 Case Study: Inverted Pendulum on a Cart

To consolidate the concepts of optimal control, we will implement a stabilizing controller for the inverted pendulum on the cart, shown in Fig. 8.12. The full nonlinear dynamics are given by

$$\dot{x} = v \tag{8.67a}$$

$$\dot{v} = \frac{-m^2L^2g\cos(\theta)\sin(\theta) + mL^2(mL\omega^2\sin(\theta) - \delta v) + mL^2u}{mL^2(M + m(1-\cos(\theta)^2))} \tag{8.67b}$$

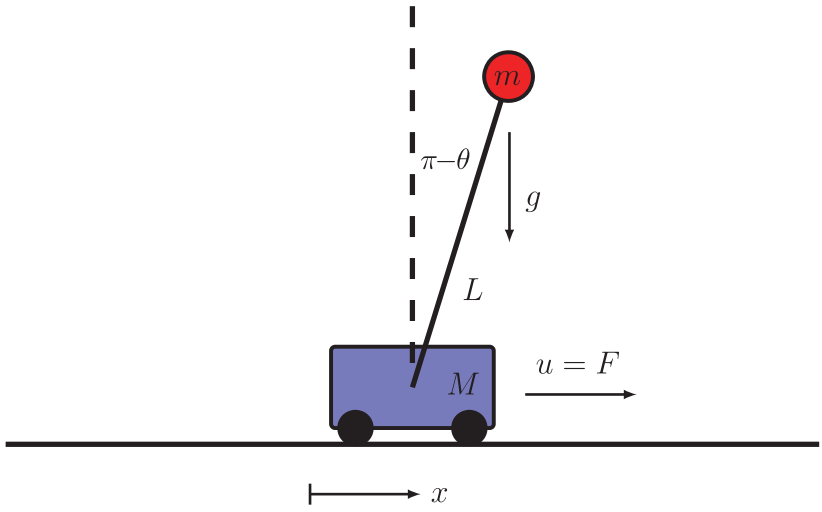

Figure 8.12 Schematic of inverted pendulum on a cart. The control forcing acts to accelerate or decelerate cart. For this example, we assume the following parameter values: pendulum mass ($m = 1$), cart mass ($M = 5$), pendulum length ($L = 2$), gravitational acceleration ($g = -10$), and cart damping ($\delta = 1$).

$$\dot{\theta} = \omega \tag{8.67c}$$

$$\dot{\omega} = \frac{(m+M)mgL\sin(\theta) - mL\cos(\theta)(mL\omega^2\sin(\theta) - \delta v) + mL\cos(\theta)u}{mL^2(M + m(1-\cos(\theta)^2))} \tag{8.67d}$$

where x is the cart position, v is the velocity, θ is the pendulum angle, ω is the angular velocity, m is the pendulum mass, M is the cart mass, L is the pendulum arm, g is the gravitational acceleration, δ is a friction damping on the dart, and u is a control force applied to the cart.

The following Matlab function, **pendcart**, may be used to simulate the full nonlinear system in (8.67):

Code 8.2 Right-hand side function for inverted pendulum on cart.

```
function dx = pendcart(x,m,M,L,g,d,u)

Sx = sin(x(3));
Cx = cos(x(3));
D = m*L*L*(M+m*(1-Cx^2));

dx(1,1) = x(2);
dx(2,1) = (1/D)*(-m^2*L^2*g*Cx*Sx + m*L^2*(m*L*x(4)^2*Sx - d*x
    (2))) + m*L*L*(1/D)*u;
dx(3,1) = x(4);
dx(4,1) = (1/D)*((m+M)*m*g*L*Sx - m*L*Cx*(m*L*x(4)^2*Sx - d*x(2)
    )) - m*L*Cx*(1/D)*u;
```

There are two fixed points, corresponding to either the pendulum down ($\theta = 0$) or pendulum up ($\theta = \pi$) configuration; in both cases, $v = \omega = 0$ for the fixed point, and the cart position x is a free variable, as the equations do not depend explicitly on x. It is possible to linearize the equations in (8.67) about either the up or down solutions, yielding the following linearized dynamics:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\delta}{M} & b\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -b\frac{\delta}{ML} & -b\frac{(m+M)g}{ML} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ b\frac{1}{ML} \end{bmatrix} u, \quad \text{for} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix}, \tag{8.68}$$

where $b = 1$ for the pendulum up fixed point, and $b = -1$ for the pendulum down fixed point. The system matrices **A** and **B** may be entered in Matlab using the values for the constants given in Fig. 8.12:

Code 8.3 Construct system matrices for inverted pendulum on a cart.

```
clear all, close all, clc

m = 1;  M = 5;  L = 2;  g = -10;  d = 1;

b = 1; % Pendulum up (b=1)

A = [0 1 0 0;
     0 -d/M b*m*g/M 0;
     0 0 0 1;
     0 -b*d/(M*L) -b*(m+M)*g/(M*L) 0];
B = [0; 1/M; 0; b*1/(M*L)];
```

We may also confirm that the open-loop system is unstable by checking the eigenvalues of **A**:

```
>> lambda = eig(A)

lambda =
         0
   -2.4311
   -0.2336
    2.4648
```

In the following, we will test for controllability and observability, develop full-state feedback (LQR), full-state estimation (Kalman filter), and sensor-based feedback (LQG) solutions.

Full-state Feedback Control of the Cart-Pendulum

In this section, we will design an LQR controller to stabilize the inverted pendulum configuration ($\theta = \pi$) assuming full-state measurements, $\mathbf{y} = \mathbf{x}$. Before any control design, we must confirm that the system is linearly controllable with the given **A** and **B** matrices:

```
>> rank(ctrb(A,B))

ans =
     4
```

Thus, the pair $(\mathbf{A}, \mathbf{B})$ is controllable, since the controllability matrix has full rank. It is then possible to specify given **Q** and **R** matrices for the cost function and design the LQR controller gain matrix **K**:

Code 8.4 Design LQR controller to stabilize inverted pendulum on a cart.

```
%%  Design LQR controller
Q = eye(4);  % 4x4 identify matrix
```

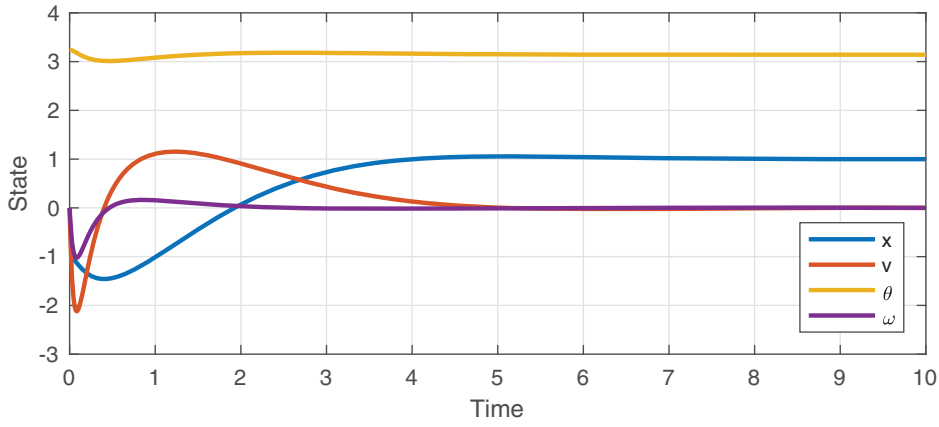

Figure 8.13 Closed-loop system response of inverted pendulum on a cart stabilized with an LQR controller.

```
R = .0001;

K = lqr(A,B,Q,R);
```

We may then simulate the closed-loop system response of the full nonlinear system. We will initialize our simulation slightly off equilibrium, at $\mathbf{x}_0 = \begin{bmatrix} -1 & 0 & \pi + .1 & 0 \end{bmatrix}^T$, and we also impose a desired step change in the reference position of the cart, from $x = -1$ to $x = 1$.

Code 8.5 Simulate closed-loop inverted pendulum on a cart system.

```
%% Simulate closed-loop system
tspan = 0:.001:10;
x0 = [-1; 0; pi+.1; 0];  % initial condition
wr = [1; 0; pi; 0];      % reference position
u=@(x)-K*(x - wr);       % control law
[t,x] = ode45(@(t,x)pendcart(x,m,M,L,g,d,u(x)),tspan,x0);
```

In this code, the actuation is set to:

$$u = -\mathbf{K}\left(\mathbf{x} - \mathbf{w}_r\right), \tag{8.69}$$

where $\mathbf{w}_r = \begin{bmatrix} 1 & 0 & \pi & 0 \end{bmatrix}^T$ is the reference position. The closed-loop response is shown in Fig. 8.13.

In the above procedure, specifying the system dynamics and simulating the closed-loop system response is considerably more involved than actually designing the controller, which amounts to a single function call in Matlab. It is also helpful to compare the LQR response to the response obtained by nonoptimal eigenvalue placement. In particular, Fig. 8.14 shows the system response and cost function for 100 randomly generated sets of stable eigenvalues, chosen in the interval $[-3.5, -.5]$. The LQR controller has the lowest overall cost, as it is chosen to minimize J. The code to plot the pendulum–cart system is provided online.

Non-minimum phase systems It can be seen from the response that in order to move from $x = -1$ to $x = 1$, the system initially moves in the wrong direction. This behavior

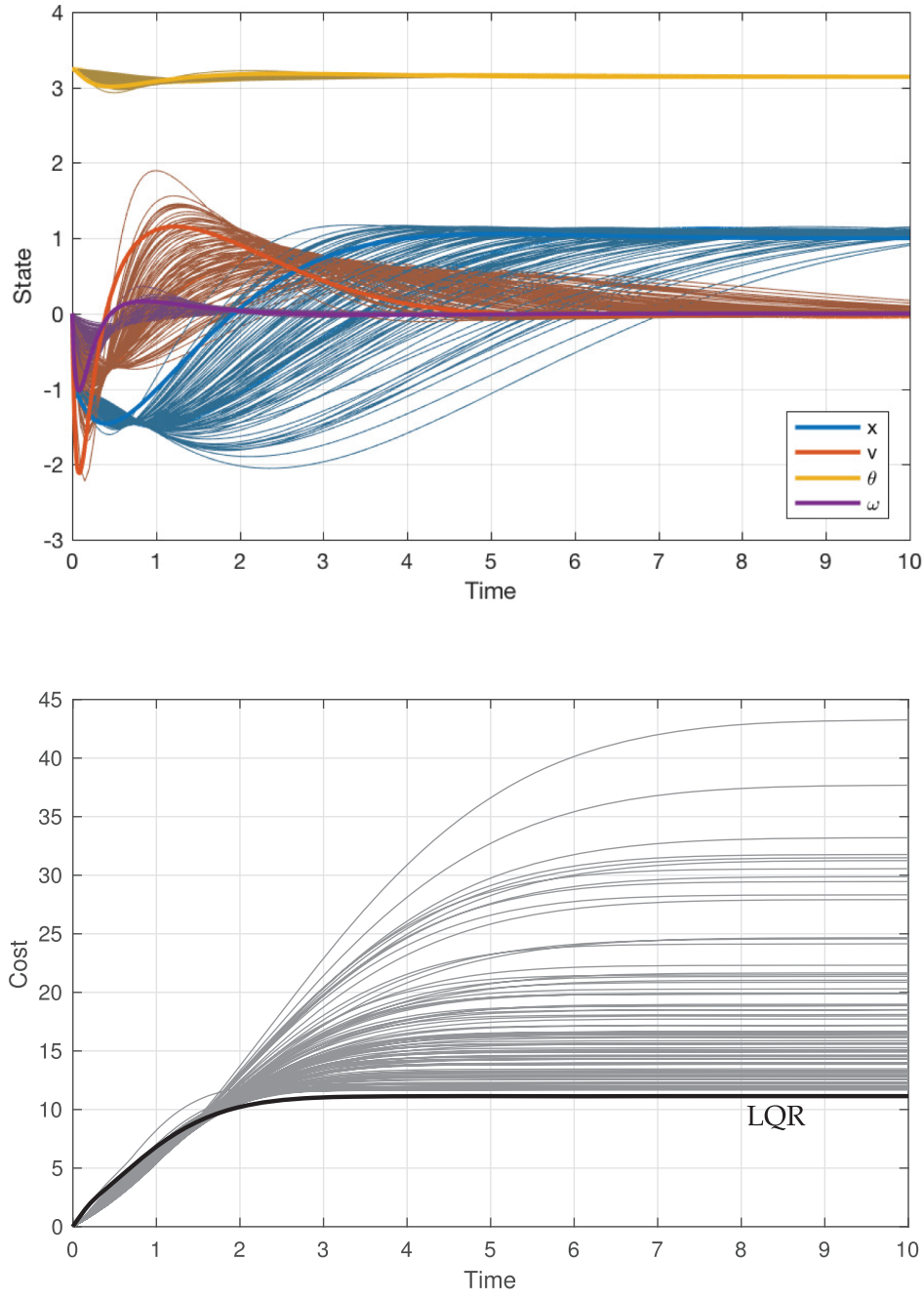

Figure 8.14 Comparison of LQR controller response and cost function with other pole placement locations. Bold lines represent the LQR solutions.

indicates that the system is *non-minimum phase*, which introduces challenges for robust control, as we will soon see. There are many examples of non-minimum phase systems in control. For instance, parallel parking an automobile first involves moving the center of mass of the car away from the curb before it then moves closer. Other examples include

increasing altitude in an aircraft, where the elevators must first move the center of mass down to increase the angle of attack on the main wings before lift increases the altitude. Adding cold fuel to a turbine may also initially drop the temperature before it eventually increases.

Full-State Estimation of the Cart-Pendulum

Now we turn to the full-state estimation problem based on limited noisy measurements $\mathbf{y}$. For this example, we will develop the Kalman filter for the pendulum-down condition ($\theta = 0$), since without feedback the system in the pendulum-up condition will quickly leave the fixed point where the linear model is valid. When we combine the Kalman filter with LQR in the next example, it will be possible to control to the unstable inverted pendulum configuration. Switching to the pendulum-down configuration is simple in the code:

```
b = -1; % pendulum down (b=-1)
```

Before designing a Kalman filter, we must choose a sensor and test for observability. If we measure the cart position, $y = x_1$,

```
C = [1 0 0 0];     % measure cart position, x
```

then the observability matrix has full rank:

```
>> rank(obsv(A,C))

ans =
     4
```

Because the cart position x_1 does not appear explicitly in the dynamics, the system is not fully observable for any measurement that doesn't include x_1. Thus, it is impossible to estimate the cart position with a measurement of the pendulum angle. However, if the cart position is not important for the cost function (i.e., if we only want to stabilize the pendulum, and don't care where the cart is located), then other choices of sensor will be admissible.

Now we design the Kalman filter, specifying disturbance and noise covariances:

```
%%  Specify disturbance and noise magnitude
Vd = eye(4);  % disturbance covariance
Vn = 1;       % noise covariance

%  Build Kalman filter
[Kf,P,E] = lqe(A,eye(4),C,Vd,Vn);  % design Kalman filter
% alternatively, possible to design using "LQR" code
Kf = (lqr(A',C',Vd,Vn))';
```

The Kalman filter gain matrix is given by

```
Kf =
    1.9222
    1.3474
   -0.6182
   -1.8016
```

Finally, to simulate the system and Kalman filter, we must augment the original system to include disturbance and noise inputs:

```
%% Augment system with additional inputs
B_aug = [B eye(4) 0*B];  % [u I*wd 0*wn]
D_aug = [0 0 0 0 0 1];   % D matrix passes noise through

sysC = ss(A,B_aug,C,D_aug);  % single-measurement system

% "true" system w/ full-state output, disturbance, no noise
sysTruth = ss(A,B_aug,eye(4),zeros(4,size(B_aug,2)));

sysKF = ss(A-Kf*C,[B Kf],eye(4),0*[B Kf]);  % Kalman filter
```

We now simulate the system with a single output measurement, including additive disturbances and noise, and we use this as the input to a Kalman filter estimator. At time $t = 1$ and $t = 15$, we give the system a large positive and negative impulse in the actuation, respectively.

```
%%  Estimate linearized system in "down" position
dt = .01;
t = dt:dt:50;

uDIST = sqrt(Vd)*randn(4,size(t,2)); % random disturbance
uNOISE = sqrt(Vn)*randn(size(t));    % random noise
u = 0*t;
u(1/dt) = 20/dt;    % positive impulse
u(15/dt) = -20/dt;  % negative impulse

u_aug = [u; uDIST; uNOISE]; % input w/ disturbance and noise

[y,t] = lsim(sysC,u_aug,t);          % noisy measurement
[xtrue,t] = lsim(sysTruth,u_aug,t); % true state
[xhat,t] = lsim(sysKF,[u; y'],t);  % state estimate
```

Fig. 8.15 shows the noisy measurement signal used by the Kalman filter, and Fig. 8.16 shows the full noiseless state, with disturbances, along with the Kalman filter estimate.

To build intuition, it is recommended that the reader investigate the performance of the Kalman filter when the model is an imperfect representation of the simulated dynamics. When combined with full-state control in the next section, small time delays and changes to the system model may cause fragility.

Sensor-Based Feedback Control of the Cart-Pendulum

To apply an LQG regulator to the inverted pendulum on a cart, we will simulate the full nonlinear system in Simulink, as shown in Fig. 8.17. The nonlinear dynamics are encapsulated in the block '**cartpend_sim**', and the inputs consist of the actuation signal u and disturbance $\mathbf{w}_d$. We record the full state for performance analysis, although only noisy measurements $y = \mathbf{C}\mathbf{x} + w_n$ and the actuation signal u are passed to the Kalman filter. The full-state estimate is then passed to the LQR block, which commands the desired actuation signal. For this example, we use the following LQR and LQE weighting matrices:

```
Q = eye(4);          % state cost
R = .000001;        % actuation cost

Vd = .04*eye(4);   % disturbance covariance
Vn = .0002;        % noise covariance
```

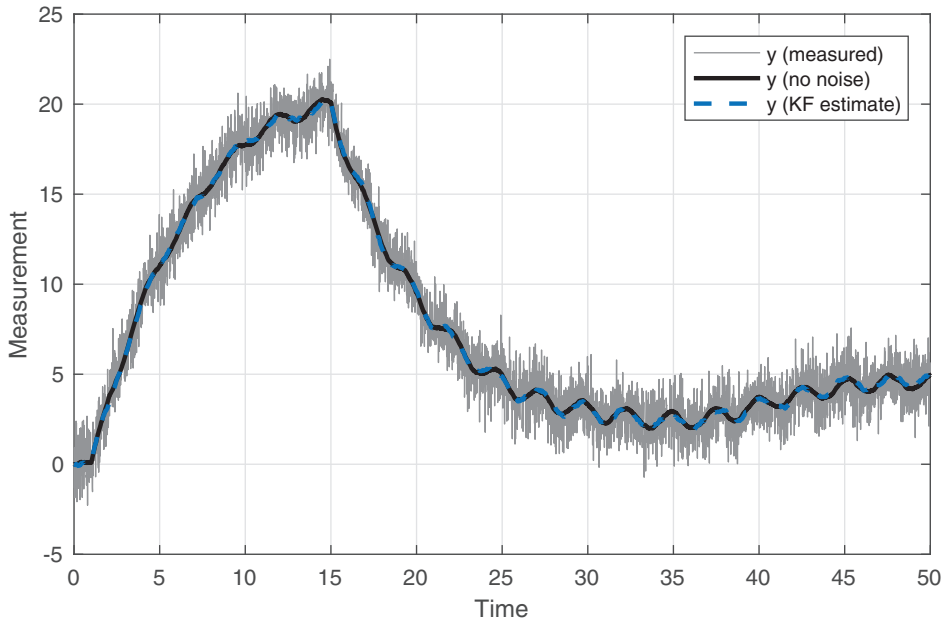

Figure 8.15 Noisy measurement that is used for the Kalman filter, along with the underlying noiseless signal and the Kalman filter estimate.

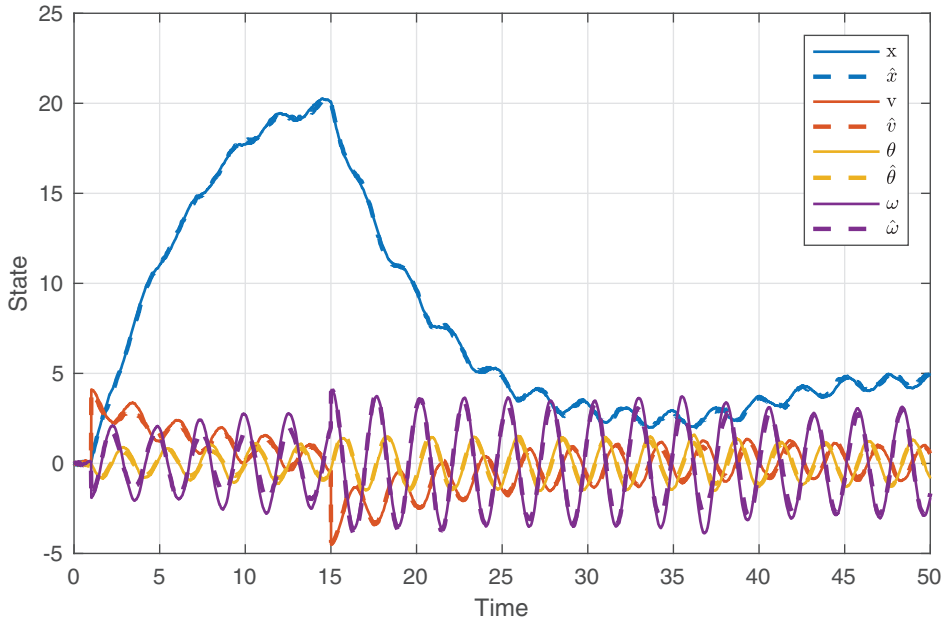

Figure 8.16 The true and Kalman filter estimated states for the pendulum on a cart system.

The system starts near the vertical equilibrium, at $\mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 3.14 & 0 \end{bmatrix}^T$, and we command a step in the cart position from $x = 0$ to $x = 1$ at $t = 10$. The resulting response is shown in Fig. 8.18. Despite noisy measurements (Fig. 8.19) and disturbances (Fig. 8.20), the controller is able to effectively track the reference cart position while stabilizing the inverted pendulum.

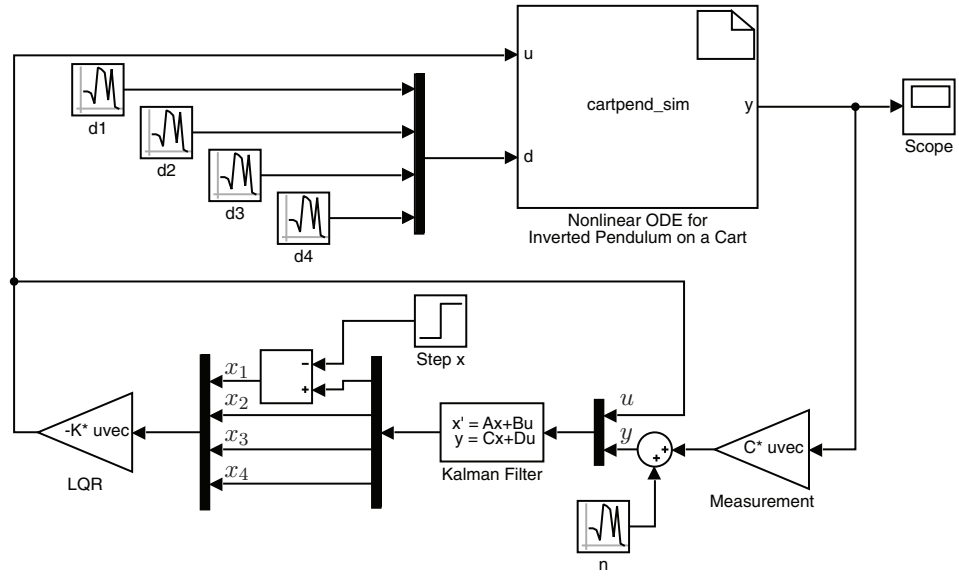

Figure 8.17 Matlab Simulink model for sensor-based LQG feedback control.

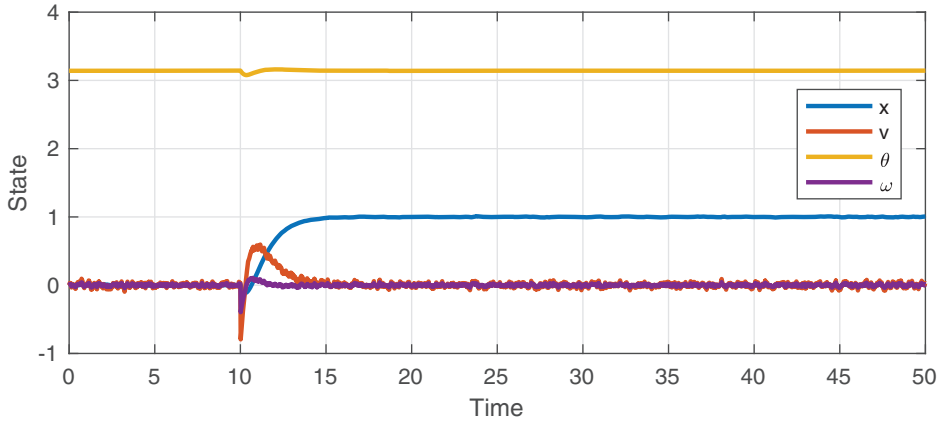

Figure 8.18 Output response using LQG feedback control.

8.8 Robust Control and Frequency Domain Techniques

Until now, we have described control systems in terms of state-space systems of ordinary differential equations. This perspective readily lends itself to stability analysis and design via placement of closed-loop eigenvalues. However, in a seminal paper by John Doyle in 1978 [155][5], it was shown that LQG regulators can have arbitrarily small stability margins, making them *fragile* to model uncertainties, time delays, and other model imperfections.

Fortunately, a short time after Doyle's famous 1978 paper, a rigorous mathematical theory was developed to design controllers that promote robustness. Indeed, this new theory

[5] Title: Guaranteed margins for LQG regulators; Abstract: There are none.

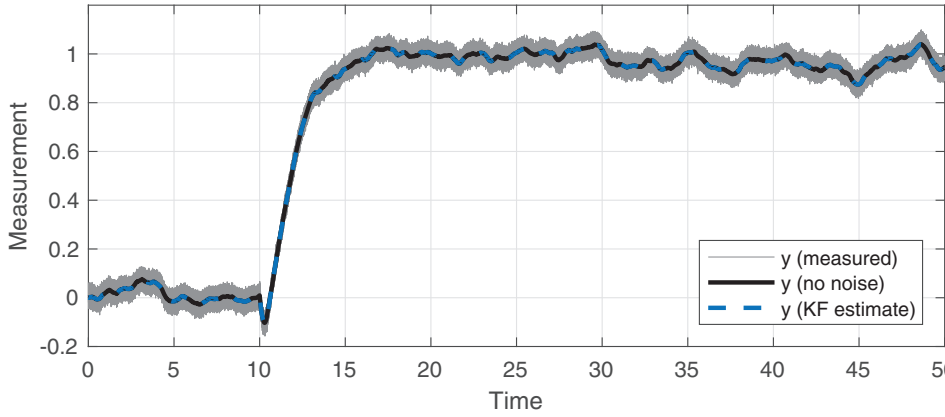

Figure 8.19 Noisy measurement used for the Kalman filter, along with the underlying noiseless signal and the Kalman filter estimate.

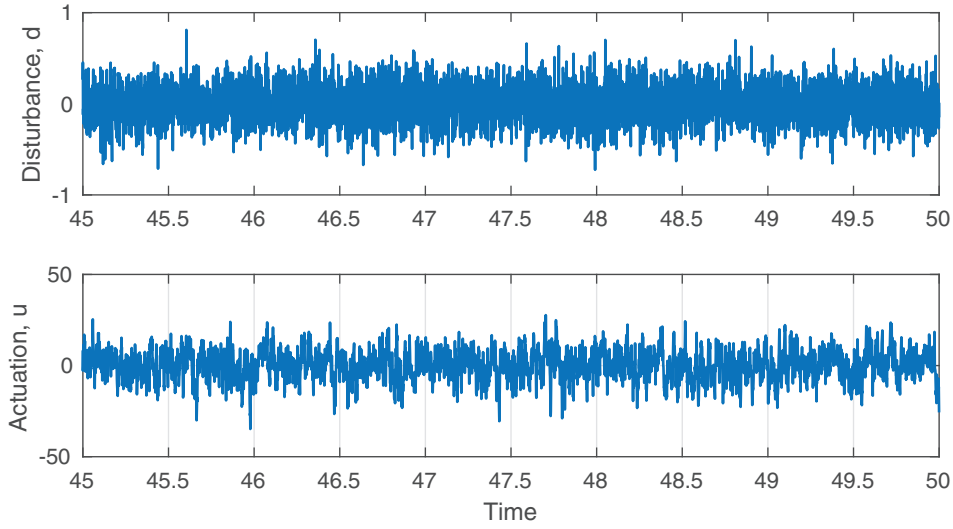

Figure 8.20 Noisy measurement used for the Kalman filter, along with the underlying noiseless signal and the Kalman filter estimate.

of robust control generalizes the optimal control framework used to develop LQR/LQG, by incorporating a different cost function that penalizes *worse-case scenario* performance.

To understand and design controllers for robust performance, it will be helpful to look at frequency domain *transfer functions* of various signals. In particular, we will consider the sensitivity, complementary sensitivity, and loop transfer functions. These enable quantitative and visual approaches to assess robust performance, and they enable intuitive and compact representations of control systems.

Robust control is a natural perspective when considering uncertain models obtained from noisy or incomplete data. Moreover, it may be possible to manage system nonlinearity as a form of structured model uncertainty. Finally, we will discuss known factors that limit robust performance, including time delays and non-minimum phase behavior.

Frequency Domain Techniques

To understand and manage the tradeoffs between robustness and performance in a control system, it is helpful to design and analyze controllers using frequency domain techniques.

The Laplace transform allows us to go between the time-domain (state-space) and frequency domain:

$$\mathcal{L}\{f(t)\} = f(s) = \int_{0^-}^{\infty} f(t)e^{-st}dt. \tag{8.70}$$

Here, s is the complex-valued Laplace variable. The Laplace transform may be thought of as a one-sided generalized Fourier transform that is valid for functions that don't converge to zero as $t \to \infty$. The Laplace transform is particularly useful because it transforms differential equations into algebraic equations, and convolution integrals in the time domain become simple products in the frequency domain. To see how time derivatives pass through the Laplace transform, we use integration by parts:

$$\begin{aligned}\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} &= \int_{0^-}^{\infty} \underbrace{\frac{d}{dt}f(t)}_{dv}\underbrace{e^{-st}}_{u}dt \\ &= \Big[f(t)e^{-st}\Big]_{t=0^-}^{t=\infty} - \int_{0^-}^{\infty} f(t)(-se^{-st})dt \\ &= f(0^-) + s\mathcal{L}\{f(t)\}.\end{aligned}$$

Thus, for zero initial conditions, $\mathcal{L}\{df/dt\} = sf(s)$.

Taking the Laplace transform of the control system in (8.10) yields

$$s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s) \tag{8.71a}$$

$$\mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s). \tag{8.71b}$$

It is possible to solve for $\mathbf{x}(s)$ in the first equation, as

$$(s\mathbf{I} - \mathbf{A})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s) \quad \Longrightarrow \quad \mathbf{x}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}(s). \tag{8.72}$$

Substituting this into the second equation we arrive at a mapping from inputs $\mathbf{u}$ to outputs $\mathbf{y}$:

$$\mathbf{y}(s) = \Big[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{C} + \mathbf{D}\Big]\mathbf{u}(s). \tag{8.73}$$

We define this mapping as the *transfer function*:

$$\mathbf{G}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{C}\,(s\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{B} + \mathbf{D}. \tag{8.74}$$

For linear systems, there are three equivalent representations: 1) time-domain, in terms of the impulse response; 2) frequency domain, in terms of the transfer function; and 3) state-space, in terms of a system of differential equations. These representations are shown schematically in Fig. 8.21. As we will see, there are many benefits to analyzing control systems in the frequency domain.

Frequency Response

The transfer function in (8.74) is particularly useful because it gives rise to the frequency response, which is a graphical representation of the control system in terms of measurable data. To illustrate this, we will consider a single-input, single-output (SISO) system. It is a

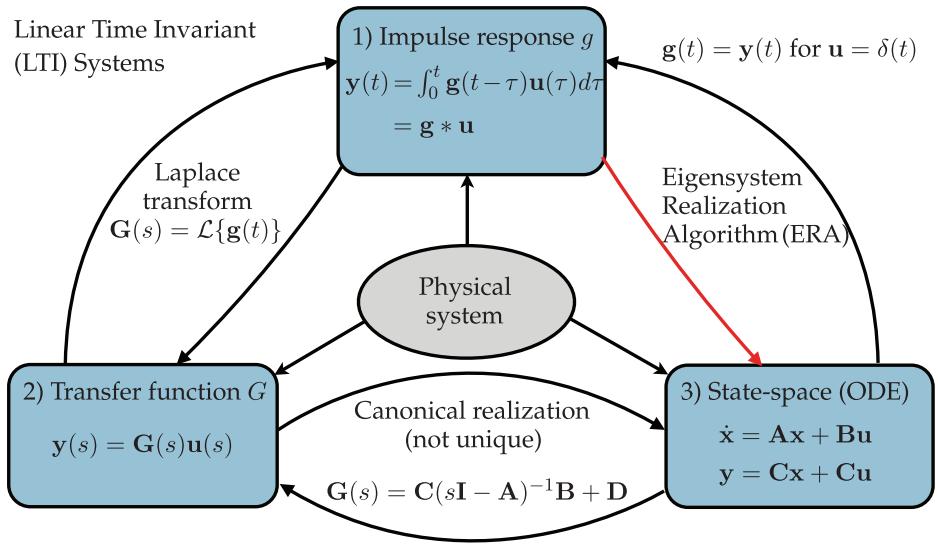

Figure 8.21 Three equivalent representations of linear time invariant systems.

property of linear systems with zero initial conditions, that a sinusoidal input will give rise to a sinusoidal output with the same frequency, perhaps with a different magnitude A and phase ϕ:

$$u(t) = \sin(\omega t) \quad \Longrightarrow \quad y(t) = A\sin(\omega t + \phi). \tag{8.75}$$

This is true for long-times, after initial transients die out. The amplitude A and phase ϕ of the output sinusoid depend on the input frequency ω. These functions $A(\omega)$ and $\phi(\omega)$ may be mapped out by running a number of experiments with sinusoidal input at different frequencies ω. Alternatively, this information is obtained from the complex-valued transfer function $G(s)$:

$$A(\omega) = |G(i\omega)|, \qquad \phi(\omega) = \angle G(i\omega). \tag{8.76}$$

Thus, the amplitude and phase angle for input $\sin(\omega t)$ may be obtained by evaluating the transfer function at $s = i\omega$ (i.e., along the imaginary axis in the complex plane). These quantities may then be plotted, resulting in the *frequency response* or *Bode plot*.

For a concrete example, consider the spring-mass-damper system, shown in Fig. 8.22. The equations of motion are given by:

$$m\ddot{x} = -\delta\dot{x} - kx + u. \tag{8.77}$$

Choosing values $m = 1$, $\delta = 1$, $k = 2$, and taking the Laplace transform yields:

$$G(s) = \frac{1}{s^2 + s + 2}. \tag{8.78}$$

Here we are assuming that the output y is a measurement of the position of the mass, x. Note that the denominator of the transfer function $G(s)$ is the characteristic equation of (8.77), written in state-space form. Thus, the poles of the complex function $G(s)$ are eigenvalues of the state-space system.

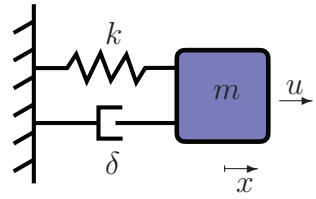

Figure 8.22 Spring-mass-damper system.

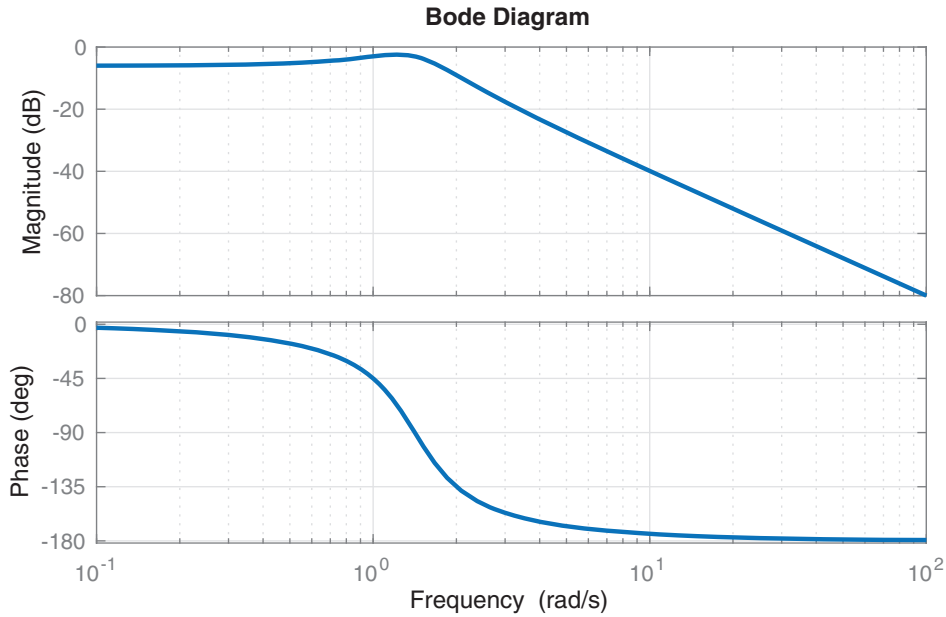

Figure 8.23 Frequency response of spring-mass-damper system. The magnitude is plotted on a logarithmic scale, in units of decibel (dB), and the frequency is likewise on a log-scale.

It is now possible to create this system in Matlab and plot the frequency response, as shown in Fig. 8.23. Note that the frequency response is readily interpretable and provides physical intuition. For example, the zero slope of the magnitude at low frequencies indicates that slow forcing translates directly into motion of the mass, while the roll-off of the magnitude at high frequencies indicates that fast forcing is attenuated and doesn't significantly effect the motion of the mass. Moreover, the resonance frequency is seen as a peak in the magnitude, indicating an amplification of forcing at this frequency.

Code 8.6 Create transfer function and plot frequency response (Bode) plot.

```
s = tf('s');            % Laplace variable
G = 1/(s^2 + s + 2);    % Transfer function

bode(G);                % Frequency response
```

Given a state-space realization,

```
>> A = [0 1; -2 -1];
>> B = [0; 1];
```

```
>> C = [1 0];
>> D = 0;
```

it is simple to obtain a frequency domain representation:

```
>> [num,den] = ss2tf(A,B,C,D); % State space to transf. fun.
>> G = tf(num,den)             % Create transfer function

G =
        1
  -----------
  s^2 + s + 2
```

Similarly, it is possible to obtain a state-space system from a transfer function, although this representation is not unique:

```
>> [A,B,C,D] = tf2ss(G.num{1},G.den{1})

A =
   -1.0000   -2.0000
    1.0000         0
B =
     1
     0
C =
     0     1
D =
     0
```

Notice that this representation has switched the ordering of our variables to $\mathbf{x} = \begin{bmatrix} v & x \end{bmatrix}^T$, although it still has the correct input–output characteristics.

The frequency-domain is also useful because impulsive or step inputs are particularly simple to represent with the Laplace transform. These are also simple in Matlab. The impulse response (Fig. 8.24) is given by

```
>> impulse(G);    % Impulse response
```

and the step response (Fig. 8.25) is given by

```
>> step(G);       % Step response
```

Performance and the Loop Transfer Function: Sensitivity and Complementary Sensitivity

Consider a slightly modified version of Fig. 8.4, where the disturbance has a model, $\mathbf{P}_d$. This new diagram, shown in Fig. 8.26, will be used to derive the important transfer functions relevant for assessing robust performance.

$$\mathbf{y} = \mathbf{GK}(\mathbf{w}_r - \mathbf{y} - \mathbf{w}_n) + \mathbf{G}_d\mathbf{w}_d \tag{8.79a}$$

$$\implies (\mathbf{I} + \mathbf{GK})\mathbf{y} = \mathbf{GK}\mathbf{w}_r - \mathbf{GK}\mathbf{w}_n + \mathbf{G}_d\mathbf{w}_d. \tag{8.79b}$$

$$\implies \mathbf{y} = \underbrace{(\mathbf{I} + \mathbf{GK})^{-1}\mathbf{GK}}_{\mathbf{T}}\mathbf{w}_r - \underbrace{(\mathbf{I} + \mathbf{GK})^{-1}\mathbf{GK}}_{\mathbf{T}}\mathbf{w}_n + \underbrace{(\mathbf{I} + \mathbf{GK})^{-1}}_{\mathbf{S}}\mathbf{G}_d\mathbf{w}_d. \tag{8.79c}$$

Here, $\mathbf{S}$ is the *sensitivity function*, and $\mathbf{T}$ is the *complementary sensitivity function*. We may denote $\mathbf{L} = \mathbf{GK}$ the *loop transfer function*, which is the open-loop transfer function in the

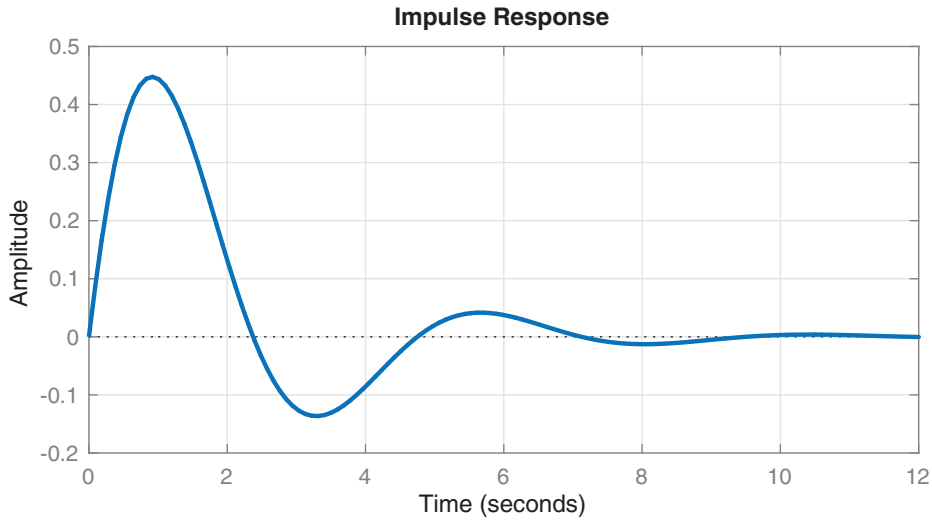

Figure 8.24 Impulse response of spring-mass-damper system.

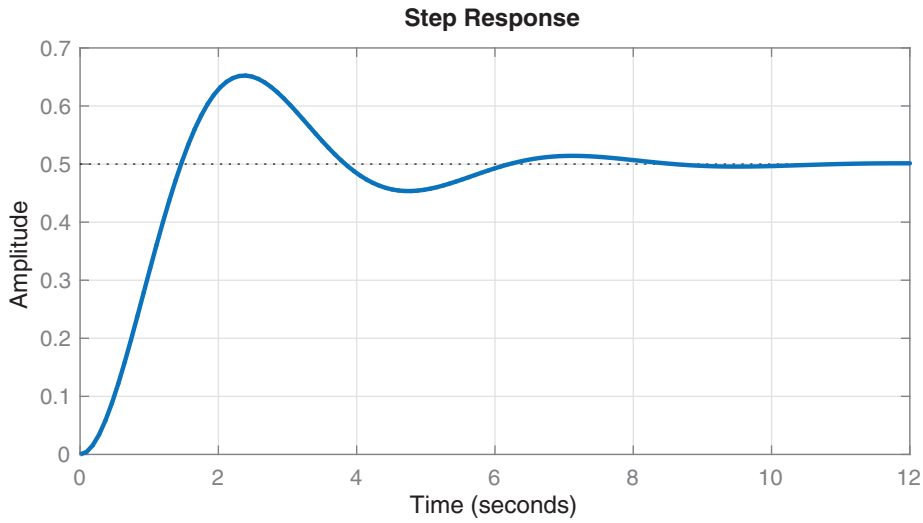

Figure 8.25 Step response of spring-mass-damper system.

absence of feedback. Both $\mathbf{S}$ and $\mathbf{T}$ may be simplified in terms of $\mathbf{L}$:

$$\mathbf{S} = (\mathbf{I} + \mathbf{L})^{-1} \tag{8.80a}$$

$$\mathbf{T} = (\mathbf{I} + \mathbf{L})^{-1}\mathbf{L}. \tag{8.80b}$$

Conveniently, the sensitivity and complementary sensitivity functions must add up to the identity: $\mathbf{S} + \mathbf{T} = \mathbf{I}$.

In practice, the transfer function from the exogenous inputs to the noiseless error $\boldsymbol{\epsilon}$ is more useful for design:

$$\boldsymbol{\epsilon} = \mathbf{w}_r - \mathbf{y} = \mathbf{S}\mathbf{w}_r + \mathbf{T}\mathbf{w}_n - \mathbf{S}\mathbf{G}_d\mathbf{w}_d. \tag{8.81}$$

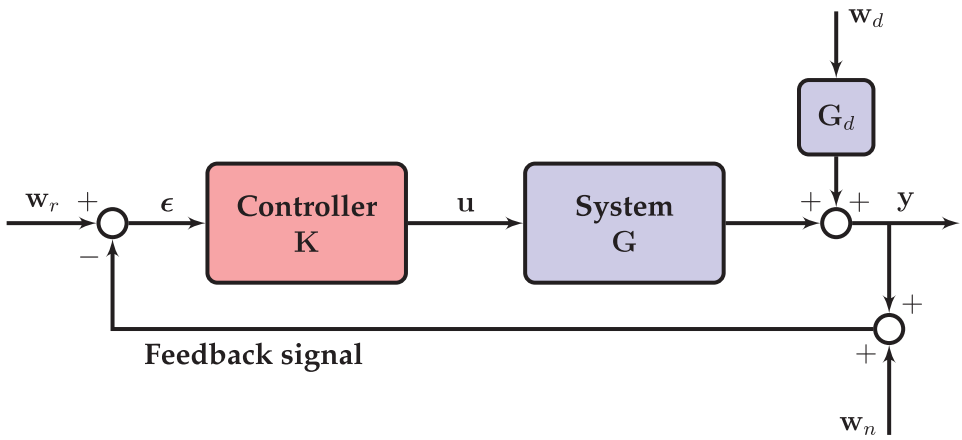

Figure 8.26 Closed-loop feedback control diagram with reference input, noise, and disturbance. We will consider the various transfer functions from exogenous inputs to the error ϵ, thus deriving the loop transfer function, as well as the sensitivity and complementary sensitivity functions.

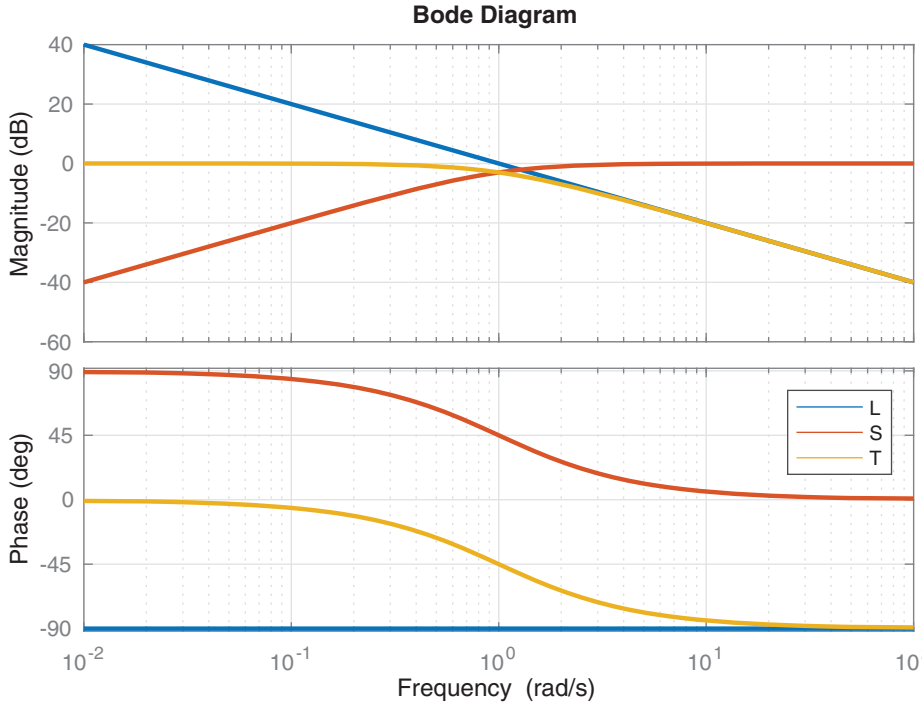

Figure 8.27 Loop transfer function along with sensitivity and complementary sensitivity functions.

Thus, we see that the sensitivity and complementary sensitivity functions provide the maps from reference, disturbance, and noise inputs to the tracking error. Since we desire small tracking error, we may then specify **S** and **T** to have desirable properties, and ideally we will be able to achieve these specifications by designing the loop transfer function **L**. In practice, we will choose the controller **K** with knowledge of the model **G** so that the loop transfer function has beneficial properties in the frequency domain. For example, small

gain at high frequencies will attenuate sensor noise, since this will result in $\mathbf{T}$ being small. Similarly, high gain at low frequencies will provide good reference tracking performance, as $\mathbf{S}$ will be small at low frequencies. However, $\mathbf{S}$ and $\mathbf{T}$ cannot both be small everywhere, since $\mathbf{S} + \mathbf{T} = \mathbf{I}$, from (8.80), and so these design objectives may compete.

For performance and robustness, we want the maximum peak of $\mathbf{S}$, $M_{\mathbf{S}} = \|\mathbf{S}\|_\infty$, to be as small as possible. From (8.81), it is clear that in the absence of noise, feedback control improves performance (i.e. reduces error) for all frequencies where $|\mathbf{S}| < 1$; thus control is effective when $\mathbf{T} \approx \mathbf{1}$. As explained in [492] (pg. 37), all real systems will have a range of frequencies where $|\mathbf{S}| > 1$, in which case performance is degraded. Minimizing the peak $M_{\mathbf{S}}$ mitigates the amount of degradation experienced with feedback at these frequencies, improving performance. In addition, the minimum distance of the loop transfer function $\mathbf{L}$ to the point -1 in the complex plane is given by $M_{\mathbf{S}}^{-1}$. By the Nyquist stability theorem, the larger this distance, the greater the stability margin of the closed-loop system, improving robustness. These are the two major reasons to minimize $M_{\mathbf{S}}$.

The controller *bandwidth* ω_B is the frequency below which feedback control is effective. This is a subjective definition. Often, ω_B is the frequency where $|\mathbf{S}(j\omega)|$ first crosses -3 dB from below. We would ideally like the controller bandwidth to be as large as possible without amplifying sensor noise, which typically has a high frequency. However, there are fundamental bandwidth limitations that are imposed for systems that have time delays or right half plane zeros [492].

Inverting the Dynamics

With a model of the form in (8.10) or (8.73), it may be possible to design an open-loop control law to achieve some desired specification without the use of measurement-based feedback or feedforward control. For instance, if perfect tracking of the reference input $\mathbf{w}_r$ is desired in Fig. 8.3, under certain circumstances it may be possible to design a controller by inverting the system dynamics $\mathbf{G}$: $\mathbf{K}(s) = \mathbf{G}^{-1}(s)$. In this case, the transfer function from reference $\mathbf{w}_r$ to output $\mathbf{s}$ is given by $\mathbf{G}\mathbf{G}^{-1} = \mathbf{1}$, so that the output perfectly matches the reference. However, perfect control is never possible in real-world systems, and this strategy should be used with caution, since it generally relies on a number of significant assumptions on the system $\mathbf{G}$. First, effective control based on inversion requires extremely precise knowledge of $\mathbf{G}$ and well-characterized, predictable disturbances; there is little room for model errors or uncertainties, as there are no sensor measurements to determine if performance is as expected and no corrective feedback mechanisms to modify the actuation strategy to compensate.

For open-loop control using system inversion, $\mathbf{G}$ must also be stable. It is impossible to fundamentally change the dynamics of a linear system through open-loop control, and thus an unstable system cannot be stabilized without feedback. Attempting to stabilize an unstable system by inverting the dynamics will typically have disastrous consequences. For instance, consider the following unstable system with a pole at $s = 5$ and a zero at $s = -10$: $G(s) = (s + 10)/(s - 5)$. Inverting the dynamics would result in a controller $K = (s - 5)/(s + 10)$; however, if there is even the slightest uncertainty in the model, so that the true pole is at $5 - \epsilon$, then the open-loop system will be:

$$G_{\text{true}}(s)K(s) = \frac{s - 5}{s - 5 + \epsilon}.$$

This system is still unstable, despite the attempted pole cancelation. Moreover, the unstable mode is now nearly unobservable.

In addition to stability, **G** must not have any time delays or zeros in the right-half plane, and it must have the same number of poles as zeros. If **G** has any zeros in the right-half plane, then the inverted controller **K** will be unstable, since it will have right-half plane poles. These systems are called *non-minimum phase*, and there have been generalizations to dynamic inversion that provide bounded inverses to these systems [149]. Similarly, time delays are not invertible, and if **G** has more poles than zeros, then the resulting controller will not be realizable and may have extremely large actuation signals **b**. There are also generalizations that provide *regularized* model inversion, where optimization schemes are applied with penalty terms added to keep the resulting actuation signal **b** bounded. These regularized open-loop controllers are often significantly more effective, with improved robustness.

Combined, these restrictions on **G** imply that model-based open-loop control should only be used when the system is well-behaved, accurately characterized by a model, when disturbances are characterized, and when the additional feedback control hardware is unnecessarily expensive. Otherwise, performance goals must be modest. Open-loop model inversion is often used in manufacturing and robotics, where systems are well-characterized and constrained in a standard operating environment.

Robust Control

As discussed previously, LQG controllers are known to have arbitrarily poor robustness margins. This is a serious problem in systems such as turbulence control, neuromechanical systems, and epidemiology, where the dynamics are wrought with uncertainty and time delays.

Fig. 8.2 shows the most general schematic for closed-loop feedback control, encompassing both optimal and robust control strategies. In the generalized theory of modern control, the goal is to minimize the transfer function from exogenous inputs **w** (reference, disturbances, noise, etc.) to a multi-objective cost function **J** (accuracy, actuation cost, time-domain performance, etc.). Optimal control (e.g., LQR, LQE, LQG) is optimal with respect to the $\mathcal{H}_2$ norm, a bounded two-norm on a Hardy space, consisting of stable and strictly proper transfer functions (meaning gain rolls off at high frequency). Robust control is similarly optimal with respect to the $\mathcal{H}_\infty$ bounded infinity-norm, consisting of stable and proper transfer functions (gain does not grow infinite at high frequencies). The infinity norm is defined as:

$$\|\mathbf{G}\|_\infty \triangleq \max_\omega \sigma_1 \left(\mathbf{G}(i\omega)\right). \tag{8.82}$$

Here, σ_1 denotes the maximum singular value. Since the $\|\cdot\|_\infty$ norm is the maximum value of the transfer function at any frequency, it is often called a *worst-case scenario norm*; therefore, minimizing the infinity norm provides robustness to worst-case exogenous inputs. $\mathcal{H}_\infty$ robust controllers are used when robustness is important. There are many connections between $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control, as they exist within the same framework and simply optimize different norms. We refer the reader to the excellent reference books expanding on this theory [492, 165].

If we let $\mathbf{G}_{\mathbf{w}\to\mathbf{J}}$ denote the transfer function from $\mathbf{w}$ to $\mathbf{J}$, then the goal of $\mathcal{H}_\infty$ control is to construct a controller to minimize the infinity norm: $\min \|\mathbf{G}_{\mathbf{w}\to\mathbf{J}}\|_\infty$. This is typically difficult, and no analytic closed-form solution exists for the optimal controller in general. However, there are relatively efficient iterative methods to find a controller such that $\|\mathbf{G}_{\mathbf{w}\to\mathbf{J}}\|_\infty < \gamma$, as described in [156]. There are numerous conditions and caveats that describe when this method can be used. In addition, there are computationally efficient algorithms implemented in both Matlab and Python, and these methods require relatively low overhead from the user.

Selecting the cost function $\mathbf{J}$ to meet design specifications is a critically important part of robust control design. Considerations such as disturbance rejection, noise attenuation, controller bandwidth, and actuation cost may be accounted for by a weighted sum of the transfer functions $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{KS}$. In the *mixed sensitivity* control problem, various weighting transfer function are used to balance the relative importance of these considerations at various frequency ranges. For instance, we may weight $\mathbf{S}$ by a low-pass filter and $\mathbf{KS}$ by a high-pass filter, so that disturbance rejection at low frequency is promoted and control response at high-frequency is discouraged. A general cost function may consist of three weighting filters $\mathbf{F}_k$ multiplying $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{KS}$:

$$\left\| \begin{bmatrix} \mathbf{F}_1\mathbf{S} \\ \mathbf{F}_2\mathbf{T} \\ \mathbf{F}_3\mathbf{KS} \end{bmatrix} \right\|_\infty .$$

Another possible robust control design is called $\mathcal{H}_\infty$ loop-shaping. This procedure may be more straightforward than mixed sensitivity synthesis for many problems. The loop-shaping method consists of two major steps. First, a desired open-loop transfer function is specified based on performance goals and classical control design. Second, the shaped loop is made robust with respect to a large class of model uncertainty. Indeed, the procedure of $\mathcal{H}_\infty$ loop shaping allows the user to design an ideal controller to meet performance specifications, such as rise-time, band-width, settling-time, etc. Typically, a loop shape should have large gain at low frequency to guarantee accurate reference tracking and slow disturbance rejection, low gain at high frequencies to attenuate sensor noise, and a cross-over frequency that ensures desirable bandwidth. The loop transfer function is then robustified so that there are improved gain and phase margins.

$\mathcal{H}_2$ optimal control (e.g., LQR, LQE, LQG) has been an extremely popular control paradigm because of its simple mathematical formulation and its tunability by user input. However, the advantages of $\mathcal{H}_\infty$ control are being increasingly realized. Additionally, there are numerous consumer software solutions that make implementation relatively straightforward. In Matlab, mixed sensitivity is accomplished using the **mixsyn** command in the robust control toolbox. Similarly, loop-shaping is accomplished using the **loopsyn** command in the robust control toolbox.

Fundamental Limitations on Robust Performance

As discussed above, we want to minimize the peaks of $\mathbf{S}$ and $\mathbf{T}$ to improve robustness. Some peakedness is inevitable, and there are certain system characteristics that significantly limit performance and robustness. Most notably, time delays and right-half plane zeros of the open-loop system will limit the effective control bandwidth and will increase the attainable

lower-bound for peaks of **S** and **T**. This contributes to both degrading performance and decreasing robustness.

Similarly, a system will suffer from robust performance limitations if the number of poles exceeds the number of zeros by more than 2. These fundamental limitations are quantified in the *waterbed* integrals, which are so named because if you push a waterbed down in one location, it must rise in a another. Thus, there are limits to how much one can push down peaks in **S** without causing other peaks to pop up.

Time delays are relatively easy to understand, since a time delay τ will introduce an additional phase lag of $\tau\omega$ at the frequency ω, limiting how fast the controller can respond effectively (i.e. bandwidth). Thus, the bandwidth for a controller with acceptable phase margins is typically $\omega_B < 1/\tau$.

Following the discussion in [492], these fundamental limitations may be understood in relation to the limitations of open-loop control based on model inversion. If we consider high-gain feedback $\mathbf{u} = \mathbf{K}(\mathbf{w}_r - \mathbf{y})$ for a system as in Fig. 8.26 and (8.81), but without disturbances or noise, we have

$$\mathbf{u} = \mathbf{K}\boldsymbol{\epsilon} = \mathbf{K}\mathbf{S}\mathbf{w}_r. \tag{8.83}$$

We may write this in terms of the complementary sensitivity **T**, by noting that since $\mathbf{T} = \mathbf{I} - \mathbf{S}$, we have $\mathbf{T} = \mathbf{L}(\mathbf{I} + \mathbf{L})^{-1} = \mathbf{G}\mathbf{K}\mathbf{S}$:

$$\mathbf{u} = \mathbf{G}^{-1}\mathbf{T}\mathbf{w}_r. \tag{8.84}$$

Thus, at frequencies where **T** is nearly the identity **I** and control is effective, the actuation is effectively inverting **G**. Even with sensor-based feedback, perfect control is unattainable. For example, if **G** has right-half plane zeros, then the actuation signal will become unbounded if the gain **K** is too aggressive. Similarly, limitations arise with time delays and when the number of poles of **G** exceed the number of zeros, as in the case of open-loop model-based inversion.

As a final illustration of the limitation of right-half plane zeros, we consider the case of proportional control $u = Ky$ in a single-input, single output system with $G(s) = N(s)/D(s)$. Here, roots of the numerator $N(s)$ are zeros and roots of the denominator $D(s)$ are poles. The closed-loop transfer function from reference w_r to sensors s is given by:

$$\frac{y(s)}{w_r(s)} = \frac{GK}{1 + GK} = \frac{NK/D}{1 + NK/D} = \frac{NK}{D + NK}. \tag{8.85}$$

For small control gain K, the term NK in the denominator is small, and the poles of the closed-loop system are near the poles of G, given by roots of D. As K is increased, the NK term in the denominator begins to dominate, and closed-loop poles are attracted to the roots of N, which are the open-loop zeros of G. Thus, if there are right-half plane zeros of the open-loop system G, then high-gain proportional control will drive the system unstable. These effects are often observed in the root locus plot from classical control theory. In this way, we see that right-half plane zeros will directly impose limitations on the gain margin of the controller.

Suggested Reading

Texts

(1) **Feedback Systems: An Introduction for Scientists and Engineers**, by K. J. Aström and R. M. Murray, 2010 [22].

(2) **Feedback Control Theory**, by J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, 2013 [157].

(3) **Multivariable Feedback Control: Analysis and Design**, by S. Skogestad and I. Postlethwaite, 2005 [492].

(4) **A Course in Robust Control Theory: A Convex Approach**, by G. E. Dullerud and F. Paganini, 2000 [165].

(5) **Optimal Control and Estimation**, by R. F. Stengel, 2012 [501].

Papers and Reviews

(1) **Guaranteed margins for LQG regulators**, by J. C. Doyle, *IEEE Transactions on Automatic Control*, 1978 [155].