Dynamic and Closed-Loop Control

Clarence W. Rowley
Princeton University, Princeton, NJ

Belinda A. Batten
Oregon State University, Corvalis, OR

Contents

1 Introduction .................................................. 2

2 Architectures .................................................. 3
   2.1 Fundamental principles of feedback ...................... 3
   2.2 Models of multi-input, multi-output (MIMO) systems .... 4
   2.3 Controllability, Observability ........................... 6
   2.4 State Space vs. Frequency Domain ........................ 8

3 Classical closed-loop control ............................. 8
   3.1 PID feedback .............................................. 9
   3.2 Transfer functions ....................................... 10
   3.3 Closed-loop stability ..................................... 11
   3.4 Gain and phase margins and robustness ................ 12
   3.5 Sensitivity function and fundamental limitations ....... 13

4 State-space design ......................................... 15
   4.1 Full-state Feedback: Linear Quadratic Regulator Problem 15
   4.2 Observers for state estimation .......................... 16
   4.3 Observer-based feedback .................................. 17
   4.4 Robust Controllers: MinMax Control .................... 18
   4.5 Examples .................................................. 20

5 Model reduction ............................................. 23
   5.1 Galerkin projection ....................................... 25
   5.2 Proper Orthogonal Decomposition ....................... 25
   5.3 Balanced truncation ...................................... 27

* Associate Professor, Mechanical and Aerospace Engineering, Senior Member AIAA
† Professor, Mechanical Engineering, Member AIAA
Copyright © 2008 by Rowley and Batten. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.
1 Introduction

In this chapter, we present techniques for feedback control, focusing on those aspects of control theory most relevant for flow control applications.

Feedback control has been used for centuries to regulate engineered systems. In the 17th century, Cornelius Drebbel invented one of the earliest devices to use feedback, an incubator that used a damper controlled by a thermometer to maintain a constant temperature. Feedback devices became more prevalent during the industrial revolution, and James Watt’s speed governor for his steam engine has become one of the most well-known early examples of feedback regulators.

In recent decades, the use of feedback has proliferated, and in particular, it has advanced the aerospace industry in critical ways. The ability of feedback controllers to modify the dynamics of a system, and in particular to stabilize systems that are
naturally unstable, enabled the design of aircraft such as the F-16, whose longitudinal dynamics are unstable, dramatically increasing the performance envelope. This ability of feedback systems to modify a system’s natural dynamics is discussed further in Sections 2.1 and 4.3.

Only recently have closed-loop controllers been used in flow control applications. Our objective here is to outline the main tools of control theory relevant to these applications, and discuss the principal advantages and disadvantages of feedback control, relative to the more common open-loop flow control strategies. We also aim to provide a bridge to the controls literature, using notation and examples that we hope are more accessible to a reader familiar with fluid mechanics, but possibly not with a background in controls. It is obviously not possible to give a detailed explanation of all of control theory in a single chapter, so we merely touch on highlights, and refer the reader to the literature for further details.

2 Architectures

Within this section, we describe the basic architecture of a closed-loop system, discussing reasons for introducing feedback, as opposed to strategies that do not use sensors, or use sensors in an open-loop fashion. The general architecture of a closed-loop control system is shown in Figure 1 where the key idea is that information from the plant is sensed, and used in computing the signal to send to the actuator. In order to describe the behavior of these systems, we will require mathematical models of the system to be controlled, called the plant, as shown in Figure 1. We then design our controller based on a model of the plant, and assumptions about sensors, actuators, and noise or disturbances entering into the system. We will see that some properties such as controllability and observability are fixed by the architecture of the closed-loop system, in particular what we choose as sensors and actuators.

2.1 Fundamental principles of feedback

Why should one use feedback at all? What are the advantages (and disadvantages) of feedback control over other control architectures? In practice, there are many tradeoffs (including weight, added complexity, reliability of sensors, etc), but here we emphasize two fundamental principles of feedback.

Modify dynamics. Feedback can modify the natural dynamics of a system. For instance, using feedback, one can improve the damping of an underdamped system, or stabilize an unstable operating condition, such as balancing an inverted pendulum. Open-loop or feed-forward approaches cannot do this. For flow control applications, an example is keeping a laminar flow stable beyond its usual transition point. We will give specific examples of modifying dynamics through feedback in Section 4.3.

\[^{3}\text{This is true at the linear level—if nonlinearities are present, however, then open-loop forcing can stabilize an unstable operating point, as demonstrated by the classic problem of a vertically-forced pendulum.}\]
Figure 1: Typical block diagram for closed-loop control. Here, $P$ denotes the plant, the system to be controlled, and $C$ denotes the controller, which we design. Sensors and actuators are denoted $y$ and $f$, respectively, and $d$ denotes external disturbances. The $-$ sign in front of $f$ is conventional, for negative feedback.

**Reduce sensitivity.** Feedback can also reduce sensitivity to external disturbances, or to changing parameters in the system itself (the plant). For instance, an automobile with a cruise control that senses the current speed can maintain the set speed in the presence of disturbances such as hills, headwinds, etc. In this example, feedback also compensates for changes in the system itself, such as changing weight of the vehicle as the number of passengers changes. If instead of using feedback, a lookup table were used to select the appropriate throttle setting based only on the desired speed, such an open-loop system would not be able to compensate either for disturbances or changes in the vehicle itself.

### 2.2 Models of multi-input, multi-output (MIMO) systems

A common way to represent a system is using a *state space model*, which is a system of first-order ordinary differential equations (ODEs) or maps. If time is continuous, then a general state-space system is written as

\[
\begin{align*}
\dot{q} &= F(q, f, d), \\
y &= G(q, f, d)
\end{align*}
\]

where $q(t)$ is the state, $f(t)$ is the control, $d(t)$ is a disturbance, and $y(t)$ is the measured output. We often assume that $F$ is a linear function of $q$, $f$, and $d$, for instance obtained by linearization about an equilibrium solution (for fluids, a steady solution of Navier-Stokes). It can also be convenient to define a *controlled output* $z(t)$, which we choose so that the objective of the controller is to keep $z(t)$ as small as possible. For instance, $z(t)$ is often a vector containing weighted versions of the sensed output $y$ and the input $f$, since often the objective is to keep the outputs small, while using little actuator power.

Many of the tools for control design are valid only for linear systems, so the form of equations (1) is often too general to be useful. Instead, we often work with linear approximations. For instance, if the system depends linearly on the actuator $f$, and
if the sensor \( y \) and controlled output \( z \) are also linear functions of the state \( q \), then (1) may be written
\[
\dot{q} = Aq + N(q) + Bf + Bd, \quad q(0) = q_0 \tag{2}
\]
\[
y = C_1q + D_{11}f + D_{12}d \tag{3}
\]
\[
z = C_2q + D_{21}f + D_{22}d. \tag{4}
\]

The term \( Aq \) is the linear part of the function \( F \) above, and \( N(q) \) the nonlinear part (for details of how these are obtained, see Section 5). The term \( C_1q \) represents the part of the state which is sensed and \( C_2q \), the part of the system which is desired to be controlled. The matrices \( D, D_{12}, \) and \( D_{21} \) are various disturbance input operators to the state, measured output, and controlled output; the disturbance terms may have known physics associated with them, or may be considered to be noise. Hence, equation (2) provides a model for the dynamics (physics) of the state, and the influence of the control through the actuators on those dynamics, (3) provides a model of the measured output provided by the sensors, and equation (4), a model of the part of the system where control should be focused.

When the plant is modeled by a system of ordinary differential equations (ODEs), the state will be an element of a vector space, e.g. \( q(t) \in \mathbb{R}^n \) for fixed \( t \). In the case that the plant is modeled by a system of partial differential equations (PDEs), e.g., as with the Navier-Stokes equations, the same notation can be used. In this context, the notation \( q(t) \) is taken to mean \( q(t, \cdot) \), where \( t \) is a fixed time and the other independent variables (such as position) vary. This interpretation of the notation leads to \( q(t) \) representing a “snapshot” of the state space at fixed time. If \( \Omega \) denotes the spatial domain in which the state exists, then \( q(t) \) lies in a state space that is a function space, such as \( L_2(\Omega) \) (the space of functions that are square integrable, in the Lebesgue sense, over the domain \( \Omega \)). The results in this chapter will be presented for systems modeled by ODEs, although there are analogs for nearly every concept defined for systems of PDEs. We do this for two primary reasons. First, the ideas will be applied in practice to computational models that are ODE approximations to the fluid systems, and this will give the reader an idea of how to compute controllers. Second, the PDE analogs require an understanding of functional analysis and are outside the scope of this book. We point out that the fundamental issue that arises in applying these concepts to ODE models that approximate PDE models is one of convergence, that is, whether or not quantities, such as controls, computed for a discretized system converge to the corresponding quantities for the original PDE. We note that it is not enough to require that the ODE model of the system, e.g., one computed from a computational fluid dynamics package, converge to the PDE model. We refer the interested reader to Curtain and Zwart for further discussion on control of PDE systems.

Throughout much of this chapter, we will restrict our attention to the simplified linear systems
\[
\dot{q}(t) = Aq(t) + Bf(t) \quad q(0) = q_0 \tag{5}
\]
\[
y(t) = Cq(t).
\]
or
\[
\dot{q}(t) = Aq(t) + Bf(t) \quad q(0) = q_0
\]
\[
y(t) = C_1q(t)
\]
\[
z(t) = C_2q(t).
\]

The first of these forms is useful when including sensed measurements with the state, and the second is useful when one wishes to control particular states given by \(z\), different from the sensed quantities. Note that in perhaps the most common case in which feedback is used, controlling about an equilibrium (such as a steady laminar solution), the linearization \([5]\) is always a good approximation of the full nonlinear system \([1]\), as long as we are sufficiently close to the equilibrium \([4]\). Since the objective of control is usually to keep the system close to this equilibrium, often the linear approximation is useful even for systems which naturally have strong nonlinearities.

The fundamental assumption is typically made that the plant is approximately the same as the model. We note that the model is never perfect, and typically contains inaccuracies of various forms: parametric uncertainty, unmodeled dynamics, nonlinearities, and perhaps purposely neglected (or truncated) dynamics. These factors motivate the use of a robust controller as given by MinMax or other \(H_\infty\) controllers that will be discussed in Section 4.3.

### 2.3 Controllability, Observability

Given an input-output system (e.g., a physical system with sensors and actuators), two characteristics that are often analyzed are controllability and observability. For a mathematical model of the form \([5]\), **controllability** describes the ability of the actuator \(f\) to influence the state \(q\). Recall that for a fluid, the state is typically all flow variables, everywhere in space, at a particular time, so controllability addresses the effect of the actuator on the entire flow. Conversely, **observability** describes the ability to reconstruct the full state \(q\) from available sensor measurements \(y\). For instance, in a fluid flow, often we cannot measure the entire flowfield directly. The sensor measurements \(y\) might consist of a few pressure measurements on the surface of a wing, and we may be interested in reconstructing the flow everywhere in the vicinity of the wing. Observability describes whether this is possible (assuming our model of the system is correct).

The following discussion assumes we have a linear model of the form \([5]\). Such a system is said to be **controllable** on the time interval \([0,T]\) if given any initial state \(q_0\), and final state \(q_f\), there exists a control \(f\) that will steer from the initial state to the final state.

Controllability is a stringent requirement: for a fluid system, if the state \(q\) consists of flow variables at all points in space, this definition means that for a system to be controllable, one must be able to drive all flow variables to any desired values in an arbitrarily short time, using the actuator. Models of fluid systems are therefore

\[1\text{and as long as the system is not degenerate: in particular, } A \text{ cannot have eigenvalues on the imaginary axis}\]
often not controllable, but if the uncontrollable modes are not important for the phenomena of interest, they may be removed from the model, for instance using methods discussed in Section 5.

Fortunately, controllability is a property that is easy to test: a necessary and sufficient condition is that the controllability matrix

$$\mathcal{C} = [B \ AB \ A^2B \ \cdots \ A^{n-1}B]$$

have full rank (rank = n). As before, n is the dimension of the state space, so q is a vector of length n and A is an n × n matrix.

The system (5) is said to be observable if for all initial times t₀, the state q(t₀) can be determined from the output y(t) defined over a finite time interval t ∈ [t₀, t₁]. The important point here is that we do not simply consider the output at a single time instant, but rather watch the output over a time interval, so that at least a short time history is available. This idea is what lets us reconstruct the full state (e.g., the full flow field) from what is typically a much smaller number of sensor measurements.

For the system to be observable, it is necessary and sufficient that the observability matrix

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

have full rank. For a derivation of this condition, and of the controllability test (7), see introductory controls texts such as Bélangers.

Although in theory the rank conditions of the controllability and observability matrices are easy to check, these can be poorly conditioned computational operations to perform. A better way to determine these properties is via the controllability and observability Gramians.

An alternative test for controllability of (5) involves checking the rank of an n × n symmetric matrix called the time-$$T$$ controllability Gramian, defined by

$$W_c(T) = \int_0^T e^{At}BB^*e^{A^*\tau} \, d\tau.$$  

This matrix $$W_c(T)$$ is invertible if and only if the system (5) is controllable. Similarly, an alternative test for observability on the interval [0, T] is to check invertibility of the observability Gramian, defined by

$$W_o(T) = \int_0^T e^{A^*\tau}C^*Ce^{A\tau} \, d\tau.$$  

(In these formulas, * denotes the adjoint of a linear operator, which is simply transpose for matrices, or complex-conjugate transpose for complex matrices.) One can show that the ranks of the Gramians in (9, 10) do not depend on T, and for a stable
system we can take \( T \to \infty \). In this case, the (infinite-time) Gramians \( W_c \) and \( W_o \) may be computed as the unique solutions to the **Lyapunov equations**

\[
AW_c + W_c A^* + BB^* = 0 \tag{11}
\]
\[
A^* W_o + W_o A + C^* C = 0. \tag{12}
\]

These are linear equations to be solved for \( W_c \) and \( W_o \), and one can show that as long as \( A \) is stable (all eigenvalues in the open left half of the complex plane), they are uniquely solvable, and the solutions are always symmetric, positive-semidefinite matrices. These Gramians are more useful than the controllability and observability matrices defined in [7,8] for studying systems that are close to uncontrollable or unobservable (i.e., one or more eigenvalues of \( W_c \) or \( W_o \) are almost zero, but not exactly zero). One can in fact construct examples which are arbitrarily close to being uncontrollable (an eigenvalue of \( W_c \) is arbitrarily close to zero), but for which the controllability matrix \( [7] \) is the identity matrix\( [6] \). While the Gramians are harder to compute, they contain more information about the degree of controllability: the eigenvectors corresponding to the largest eigenvalues can be considered the “most controllable” and “most observable” directions, in ways that can be made precise. For derivations and further details, as well as more tests for controllability and observability, and the related concepts of stabilizability and detectability, see Dullerud and Paganini\( [6] \). For further discussion and algorithms for computation, see also Datta\( [7] \).

### 2.4 State Space vs. Frequency Domain

In the frequency domain, the models in Section 2.2 are represented by transfer functions, to be discussed in more detail in Section 3.2. By taking the Laplace transform of the system in \( [5] \), we obtain the representation of the system

\[
\tilde{y}(s) = C(sI - A)^{-1}B \tilde{f}(s) \tag{13}
\]

where \( G(s) = C(sI - A)^{-1}B \) is the transfer function from input to sensed output. For the system in \( [6] \), we have the additional transfer function \( H(s) = C_2(sI - A)^{-1}B \) as the transfer function from input \( f \) to controlled output \( z \). Many control concepts were developed in the frequency domain and have analogs in the state space. The type of approach that is used often depends on the goal of applying the control design. For example, if one is trying to control behavior that has a more obvious interpretation in the time domain, the state space may be the natural way to consider control design. On the other hand, if certain frequency ranges are of interest in control design, the frequency domain approach may be the one that is desired. Many references consider both approaches, and for more information, we refer the reader to standard controls texts\( [11,12,13] \).

### 3 Classical closed-loop control
In this section, we explore some features of feedback control from a classical perspective. Traditionally, classical control refers to techniques that are in the frequency domain (as opposed to state-space representations, which are in the time-domain), and often are valid only for linear, single-input, single-output systems. Thus, in this section, we assume that the input $f$ and output $y$ are scalars, denoted $f$ and $y$, respectively. The corresponding methods are often graphical, as they were developed before digital computers made matrix computations relatively easy, and they involve using tools such as Bode plots, Nyquist plots, and root-locus diagrams to predict behavior of a closed-loop system.

We begin by describing the most common type of classical controller, Proportional-Integral-Derivative (PID) feedback. We then discuss more generally the notion of a transfer function, and Bode and Nyquist methods for predicting closed-loop behavior (e.g., stability, or tracking performance) from open-loop characteristics of a system. For further information on the topics in this section, we refer the reader to standard texts.

3.1 PID feedback

By far the most common type of controller used in practice is Proportional-Integral-Derivative (PID) feedback. In this case, in the feedback loop shown in Figure 4, the controller is chosen such that

$$f(t) = -K_p y(t) - K_i \int_0^\infty y(\tau) d\tau - K_d \frac{dy(t)}{dt}$$

where $K_p$, $K_i$, and $K_d$ are proportional, integral, and derivative gains, respectively.

In general, for a first-order system, PI control (in which $K_d = 0$) is generally effective, while for a second-order system, PD control (in which $K_i = 0$) or full PID control is usually appropriate.

To see the effect of PI and PD feedback on these systems, consider a first-order system of the form

$$\dot{y} + ay = f$$

where $f$ is the input (actuator), $y$ is the output (sensor), and $a$ is a parameter, and suppose that the control objective is to alter the dynamics of (15). Without feedback ($f = 0$), the system has a pole at $-a$, or a dynamic response of $e^{-at}$. We may use feedback to alter the position of this pole. Choosing PI feedback $f = -K_p y - K_i \int y dt$, the closed-loop system becomes

$$\dot{y} + (a + K_p)\dot{y} + K_i y = 0.$$ 

The closed-loop system is now second order, and with this feedback, the poles are the roots of

$$s^2 + (a + K_p)s + K_i = 0$$

Clearly, by appropriate choice of $K_p$ and $K_i$, we may place the two poles anywhere we desire in the complex plane, since we have complete freedom over both coefficients in (17).
Next, consider a second-order system, a spring-mass system with equations of motion
\[ m\ddot{y} + b\dot{y} + ky = f, \tag{18} \]
where \( m \) is the mass, \( b \) is the damping constant, and \( k \) is the spring constant. With PD feedback, \( u = -K_p y - K_d \dot{y} \), the closed-loop system becomes
\[ m\ddot{y} + (b + K_d)\dot{y} + (k + K_p)y = 0 \tag{19} \]
with closed-loop poles satisfying
\[ ms^2 + (b + K_d)s + k + K_p = 0. \tag{20} \]
Again, we may place the poles anywhere desired, since we have freedom to choose both of the coefficients in (20). For tracking problems (where the control objective is for the output to follow a desired reference), often integral feedback is used to drive the steady-state error to zero. However, introducing integral feedback into second-order systems can make the closed loop unstable for high gains, so the techniques in the latter part of this chapter need to be used to ensure stability. Also, note that in practice if the gains \( K_p \) and especially \( K_d \) are increased too much, sensor noise becomes a serious problem. For more information and examples of PID control, we refer the reader to conventional controls texts.

### 3.2 Transfer functions

Linear systems obey the principle of superposition, and hence many useful tools are available for their analysis. One of these tools that is ubiquitous in classical control is the Laplace transform. The **Laplace transform** of a function of time \( y(t) \) is a new function \( \tilde{y}(s) \) of a variable \( s \in \mathbb{C} \), defined as
\[ \tilde{y}(s) = \mathcal{L}\{y(t)\} = \int_0^\infty e^{-st}y(t)\, dt, \tag{21} \]
defined for values of \( s \) for which the integral converges. For properties of the Laplace transform, we refer the reader to standard texts, but the most useful property for our purposes is that
\[ \mathcal{L}\left\{ \frac{dy}{dt} \right\} = s\tilde{y}(s) - y(0). \tag{22} \]
Hence, the Laplace transform converts differential equations into algebraic equations.

For a plant with input \( u \) and output \( y \) and dynamics given by
\[ a_2\ddot{y} + a_1\dot{y} + a_0y = b_1\dot{f} + b_0f, \tag{23} \]
taking the Laplace transform, with zero initial conditions \( y(0) = \dot{y}(0) = f(0) = 0 \), gives
\[ (a_2s^2 + a_1s + a_0)\tilde{y} = (b_1s + b_0)\tilde{f}, \tag{24} \]
or
\[ \frac{\tilde{y}(s)}{\tilde{u}(s)} = \frac{b_1s + b_0}{a_2s^2 + a_1s + a_0} = P(s), \tag{25} \]
where \( P(s) \) is called the transfer function from \( f \) to \( y \). Thus, the transfer function relates the Laplace transform of the input to the Laplace transform of the output. It is often defined as the Laplace transform of the impulse response (a response to an impulsive input, \( f(t) = \delta(t) \)), since the Laplace transform of the Dirac delta function is 1. For a state-space representation as in (2), the transfer function is given by (3).

**Frequency response** The transfer function has a direct physical interpretation in terms of the frequency response. If a linear system is forced by a sinusoidal input at a particular frequency, once transients decay, the output will also be sinusoidal, at the same frequency, but with a change in amplitude and a phase shift. That is,

\[
f(t) = \sin(\omega t) \implies y(t) = A(\omega) \sin(\omega t + \varphi(\omega))
\]

The frequency response determines how these amplitudes \( A(\omega) \) and phase shifts \( \varphi(\omega) \) vary with frequency. These may be obtained directly from the transfer function. For any frequency \( \omega \), \( P(i\omega) \) is a complex number, and one can show that

\[
A(\omega) = |P(i\omega)|, \quad \varphi(\omega) = \angle P(i\omega),
\]

where for a complex number \( z = x + iy \), \( |z| = \sqrt{x^2 + y^2} \) and \( \angle z = \tan^{-1} y/x \). The frequency response is often depicted by a Bode plot, which is a plot of the magnitude and phase as a function of frequency (on a log scale), as Figure 3. The amplitude is normally plotted on a log scale, or in terms of decibels, where \( A(\text{dB}) = 20 \log_{10} A \).

### 3.3 Closed-loop stability

Bode plots tell us a great deal about feedback systems, including information about the performance of the closed-loop system over different ranges of frequencies, and even the stability of the closed loop system, and how close it is to becoming unstable.

For the system in Figure 1, suppose that the transfer function from \( d \) to \( y \) is \( W(s) \), the transfer function from \( f \) to \( y \) is \( P(s) \), and the transfer function of the
controller is \( C(s) \), so that the system has the form shown in Figure 2. Then the closed-loop transfer function from disturbance \( d \) to the output \( y \) is

\[
\frac{\hat{y}(s)}{d(s)} = \frac{W(s)}{1 + P(s)C(s)}.
\]

(28)

Closed-loop poles therefore occur where \( 1 + P(s)C(s) = 0 \), so in order to investigate stability, we study properties of the loop gain \( G(s) = P(s)C(s) \).

One typically determines stability using the Nyquist criterion, which uses another type of plot called a Nyquist plot. The Nyquist plot of a transfer function \( G(s) \) is a plot of \( G(i\omega) \) in the complex plane (\( \text{Im} G(i\omega) \) vs. \( \text{Re} G(i\omega) \)), as \( \omega \) varies from \(-\infty\) to \( \infty \). Note that since the coefficients in the transfer function \( G(s) \) are real, \( G(-i\omega) = G(i\omega) \), so the Nyquist plot is symmetric about the real axis. The Nyquist stability criterion then states that the number of unstable (right-half-plane) closed-loop poles of \( G \) equals the number of unstable (right-half-plane) open-loop poles of \( G \), plus the number of times the Nyquist plot encircles the \(-1\) point in a clockwise direction. (This criterion follows from a result in complex variables called Cauchy’s principle of the argument.)

One can therefore determine the stability of the closed loop just by looking at the Nyquist plot, and counting the number of encirclements. For most systems that arise in practice, this criterion is also straightforward to interpret in terms of a Bode plot. We illustrate this through an example in the following section.

### 3.4 Gain and phase margins and robustness

Suppose that we have designed a controller that stabilizes a given system. One may then ask how much leeway we have before the feedback system becomes unstable. For instance, suppose our control objective is to stabilize a cylinder wake at a Reynolds number at which the natural flow exhibits periodic shedding, and suppose that we have designed a feedback controller that stabilizes the steady laminar solution. How much can we adjust the gain of the amplifier driving the actuator, before the closed-loop system loses stability and shedding reappears? Similarly, how much phase lag can we tolerate, for instance in the form of a time delay, before closed-loop stability is lost? Gain and phase margins address these questions.

As a concrete example, consider the system in Fig. 2 with transfer functions

\[
P(s) = \frac{1}{(s+1)(s+2)(s+3)}, \quad C(s) = 20,
\]

(29)

for which the Nyquist and Bode plots of \( G(s) = P(s)C(s) \) are shown in figure 3. We see that the Nyquist plot does not encircle the \(-1\) point, so for this value of \( C(s) \), the closed-loop system is stable, by the Nyquist stability criterion. However, if the gain were increased enough, the Nyquist plot would encircle the \(-1\) point twice, indicating two unstable closed-loop poles. The amount by which we may increase the gain before instability is called the gain margin, and is indicated GM in the figure. The gain margin may be determined from the Bode plot as well, as also shown in the figure, by noting the value of the gain where the phase crosses \(-180^\circ\).
This value is one measure of how close the closed-loop system is to instability: a small gain margin (close to 1) indicates the system is close to instability.

Another important quantity is the phase margin, also shown in Fig. 3 (denoted PM). The phase margin specifies how much the phase at the crossover frequency (where the magnitude is 1) differs from $-180^\circ$. Clearly, if the phase equals $-180^\circ$ at crossover, the Nyquist plot intersects the $-1$ point, and the system is on the verge of instability. Thus, the phase margin is another measure of how close the system is to instability.

Both of these quantities relate to the overall robustness of the closed-loop system, or how sensitive the closed-loop behavior is to uncertainties in the model of the plant. Of course, our mathematical descriptions of the form \( \frac{P(s)}{C(s)} \) are only approximations of the true dynamics, and these margins give some measure of how much imprecision we may tolerate before the closed-loop system becomes unstable. See Doyle et al.\(^\text{14}\) for a more quantitative (but still introductory) treatment of these robustness ideas.

### 3.5 Sensitivity function and fundamental limitations

One of the main principles of feedback discussed in Section 2.1 is that feedback can reduce a system’s sensitivity to disturbances. For example, without control \((C(s) = 0 \text{ in Fig. 2})\), the transfer function from the disturbance \(d\) to the output \(y\) is \(W(s)\). With feedback, however, the transfer function from disturbance to output is given by \(\frac{1}{1+G(s)}\). That is, the transfer function is modified by the amount

\[
S(s) = \frac{1}{1+G(s)},
\]

called the sensitivity function, where \(G(s) = P(s)C(s)\) is the loop gain, as before. Thus, for frequencies \(\omega\) where \(|S(i\omega)| < 1\), then feedback acts to reduce the effect of disturbances; but if there are frequencies for which \(|S(i\omega)| > 1\), then feedback has an adverse effect, amplifying disturbances more than without control.
Figure 4: Nyquist plot of the loop gain $G(s) = P(s)C(s)$ for the system. For frequencies for which $G(i\omega)$ enters the unit circle centered about the $-1$ point, disturbances are amplified, and for frequencies for which $G(s)$ lies outside this circle, disturbances are attenuated relative to open-loop.

The magnitude of the sensitivity function is easy to see graphically from the Nyquist plot of $G(s)$, as indicated in Figure 4. At each frequency $\omega$, the length of the line segment connecting $-1$ to the point $G(i\omega)$ is $1/|S(i\omega)|$. Thus, for frequencies for which $G(i\omega)$ lies outside the unit circle centered at the $-1$ point, disturbances are attenuated, relative to the open-loop system. However, we see that there are some frequencies for which the Nyquist plot enters this circle, for which $|S(i\omega)| > 1$, indicating that disturbances are amplified, relative to open-loop.

The example plotted in Figure 4 used a particularly simple controller. One might wonder if it is possible to design a more sophisticated controller such that disturbances are attenuated at all frequencies. Unfortunately, one can show that under very mild restrictions, it is not possible to do this, for any linear controller. This fact is a consequence of Bode’s integral formula, also called the area rule (or the “no-free-lunch theorem”), which states that if the loop gain has relative degree $\geq 2$ (i.e., $G(s)$ has at least two more poles than zeros), then the following formula holds:

$$\int_0^\infty \log_{10} |S(i\omega)| \, d\omega = \pi (\log_{10} e) \sum_j \Re p_j,$$

where $p_j$ are the unstable (right-half-plane) poles of $G(s)$ (for a proof, see Doyle et al. [14]). The limitations imposed by this formula are illustrated in Figure 5 which shows $|S(i\omega)|$ for the $G(s)$ used in our example. Here, one sees that the area of attenuation, for which $\log |S(i\omega)| < 0$, is balanced by an equal area of amplification, for which $\log |S(i\omega)| > 0$. These areas must in fact be equal in this log-linear plot. If our plant $P(s)$ had unstable poles, then the area of amplification must be even greater than the area of amplification, as required by (31).
While these performance limitations are unavoidable, typically in physical situations it is not a severe limitation, since the area of amplification may be spread over a large frequency range, with only negligible amplification. However, in some examples, particularly systems with narrow bandwidth actuators, in which these regions must be squeezed into a narrow range of frequencies, this result can lead to large peaks in $S(i\omega)$, leading to strongly amplified disturbances for the closed-loop system. These explain peak-splitting phenomena that have been observed in several flow-control settings, including combustion instabilities\cite{15} and cavity oscillations\cite{16,17}. This peaking phenomenon is further exacerbated by the presence of time delays, as described in the context of a combustor control problem in Cohen and Banaszuk\cite{18}.

\section{State-space design}

We discuss the fundamental approaches to feedback control design in the time domain in the subsections below. We start with the most fundamental problems, and add more practical aspects. By the end of this section, the reader should have an overview of some of the most common approaches to linear control design for state space models.

\subsection{Full-state Feedback: Linear Quadratic Regulator Problem}

In this section, we assume that we have knowledge of all states (i.e., $C_1$ is the identity matrix) and can use that information in a feedback control design. Although this is typically infeasible, it is a starting point for more practical control designs. The linear quadratic regulator (LQR) problem is stated as follows: find the control $f(t)$
to minimize the quadratic objective function

$$J(f) = \int_0^{\infty} \left( q^T(t)Qq(t) + f^T(t)Rf(t) \right) dt$$

subject to the state dynamics

$$\dot{q}(t) = Aq(t) + Bf(t), \quad q(0) = q_0.$$  \hspace{1cm} (33)

The matrices $Q$ and $R$ are state and control weighting operators and may be chosen to obtain desired properties of the closed loop system. In particular, when there is a controlled output as in [\ref{4}], defining $Q = C_2^T C_2$ places the controlled output in the objective function. It is necessary that $Q$ be non-negative definite and that $R$ be positive definite in order for the LQR problem to have a solution. In addition, $(A, B)$ must be controllable (as defined in Section [\ref{2.3}]). This problem is called linear because the dynamic constraints are linear, and quadratic since the objective function is quadratic. The controller is called a regulator because the optimal control will drive the state to zero.

The solution to this problem can be obtained by applying the necessary conditions of the calculus of variations. One finds that under the above assumptions, the solution to this problem is given by the feedback control

$$f(t) = -Kq(t), \quad K = R^{-1}B^T \Pi$$

where the matrix $K$ is called the feedback gain matrix, and the symmetric matrix $\Pi$ is the solution of the algebraic Riccati equation

$$A^T \Pi + \Pi A - \Pi BR^{-1}B^T \Pi + Q = 0.$$  \hspace{1cm} (34)

This is a quadratic matrix equation, and has many solutions $\Pi$, but only one positive-definite solution, which is the one we desire. Equations of this type may be solved easily, for instance using the Matlab commands \texttt{are} or \texttt{lqr}.

Often, one wishes to drive the state to some desired state $q_d$. In that case, a variant of this problem, called a tracking problem can be solved\cite{10}.

\section*{4.2 Observers for state estimation}

Typically, one does not have the full-state available, but rather only a (possibly noisy) sensor measurement of part of the state, as described in Section \[2\]. For these problems, in order to implement a full-state feedback law as described in the previous section, one needs an estimator (or filter or observer) that provides and estimate of the state based upon sensed measurements. The estimator is another dynamic system, identical to the state equation \[5\], but with an added correction term, based on the sensor measurements:

$$\dot{q}_c(t) = Aq_c(t) + Bf(t) + L(y(t) - Cq_c(t)), \quad q_c(0) = q_{c0}.$$  \hspace{1cm} (35)

Without the correction term, this equation is identical to \[5\], and hence, our estimate $q_c(t)$ should match the actual state $q(t)$, as long as we know the initial value $q(0)$,
and as long as our model perfectly describes the actual system. However, in practice, we do not have access to a perfect model, or to an initial value of the entire state, so we add the correction term \( L(y - Cq_c) \), and choose weights \( L \) so that the state estimate \( q_c(t) \) approaches the actual state \( q(t) \) as time goes on. In particular, defining the estimation error \( e = q_c - q \), and combining (5) and (35), we have

\[
\dot{e}(t) = (A - LC)e(t),
\]

so this error converges to zero as long as the eigenvalues of \( A - LC \) are in the open left half plane. One can show (see, for instance, Bélanger) that as long as the pair \((A, C)\) is observable, the gains \( L \) may be chosen to place the eigenvalues of \( A - LC \) anywhere in the complex plane (the pole-placement theorem). Hence, as long as the system is observable, the weights \( L \) can always be chosen so that the estimate converges to the actual state.

The weights \( L \) can also be chosen in an optimal manner, which balances the relative importance of sensor noise and process noise. If one has a noisy or inaccurate sensor, then the measurements \( y \) should not be trusted as much, and the corresponding weights \( L \) should be smaller. Process noise consists of actual disturbances to the system (e.g., \( d \) in (2)), so if these are large, then one will need larger sensor corrections to account for these disturbances, and in general the entries in \( L \) should be larger. The optimal compromise between sensor noise and process noise is achieved by the Kalman filter. There are many types of Kalman filters, for continuous-time, discrete-time, time-varying, and nonlinear systems. Here, we discuss only the simplest form, for linear time-invariant systems, and for more complex cases, we refer the reader to Stengel, Gelb, or other references.

We first augment our model with terms that describe the sensor noise \( n \) and disturbances \( d \):

\[
\begin{align*}
\dot{q} &= Aq + Bf + Dd \\
y &= Cq + n.
\end{align*}
\]

We additionally assume that both \( n \) and \( d \) are zero-mean, Gaussian white noise, with covariances given by

\[
E(dd^T) = Q_e, \quad E(nn^T) = R_e
\]

where \( E(\cdot) \) denotes the expected value. The optimal estimator (that minimizes the covariance of the error \( e = q_c - q \) in steady state) is given by \( L = PC^T R^{-1}_e \), where \( P \) is the unique positive-definite solution of the Riccati equation

\[
AP + PA^T - PC^T R^{-1}_e CP + DQ_e D^T = 0.
\]

This optimal estimation problem is, in a precise sense, a dual of the LQR problem discussed in Section 4.1, and may also be solved easily, using Matlab commands are or 1qe.

### 4.3 Observer-based feedback

Once we have an estimate \( q_c \) of the state, for instance from the optimal observer described above, we can use this estimate in conjunction with the state feedback
controllers described in Section 1.1 using the feedback law
\[ f(t) = -Kq_c(t). \] (40)

Combining the above equation with the observer dynamics \[ (35) \], the resulting feedback controller depends only on the sensed output \( y \), and is called a Linear Quadratic Gaussian (LQG) regulator.

One might well question whether this procedure should work at all: if the state-feedback controller was designed assuming knowledge of the full state \( q \), and then used with an approximation of the state \( q_c \) is the resulting closed-loop system guaranteed to be stable? The answer, at least for linear systems, is yes, and this major result is known as the separation principle: if the full-state feedback law \( f = -Kq \) stabilizes the system (e.g., \( A - BK \) is stable), and the observer \( (35) \) converges to the actual state (e.g., \( A - LC \) is stable), then the observer-based feedback law \( (40) \) stabilizes the full system.

One can see this result as follows. Combining the controller given by \( (40) \) with the estimator dynamics \( (35) \) and the state dynamics \( (33) \), one obtains the overall closed-loop system
\[ \begin{bmatrix} \dot{q}(t) \\ \dot{q}_c(t) \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix} \begin{bmatrix} q(t) \\ q_c(t) \end{bmatrix}. \] (41)

Changing to the variables \( (q, e) \), where as before, \( e = q_c - q \), \( (41) \) becomes
\[ \begin{bmatrix} \dot{q}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A - BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} q(t) \\ e(t) \end{bmatrix}. \] (42)

Since the matrix on the right-hand side is block upper triangular, its eigenvalues are the union of the eigenvalues of \( A - BK \) and those of \( A - LC \). These are always stable (eigenvalues in the open left-half plane) if the state-feedback law is stable and the observer dynamics are stable, and so the separation principle holds.

This remarkable result demonstrates that one is justified in designing the state feedback gains \( K \) separately from the observer gains \( L \).

4.4 Robust Controllers: MinMax Control

If the state equation has a disturbance term present as
\[ \dot{q}(t) = Aq(t) + Bf(t) + Dd(t), \quad q(0) = q_0, \] (43)
there is a related control design that can be applied, the MinMax design \[ (7,22,23) \]. The objective function to be solved is to find
\[ \min_{f} \max_{d} J(f, d) = \int_{0}^{\infty} (q^T(t)Qq(t) + f^T(t)Rf(t) - \gamma^2 d^T(t)d(t)) \, dt \] (44)
subject to \( (43) \). The solution has the same feedback form as in the LQR problem, but this time, the Riccati matrix \( \Pi \) is the solution to
\[ A^T \Pi + \Pi A - \Pi [BR^{-1}B^T - \theta^2 DD^T]\Pi + Q = 0 \] (45)
where $\theta = 1/\gamma$. The MinMax control is more robust to disturbances than is LQR; the larger the value of $\theta$, the more robust it is. There is a limit to the size of $\theta$, and if it is chosen as large as possible, the resulting controller is a different kind of optimal controller, called the $H_\infty$ controller, which is robust in a sense that can be made precise, but can be overly conservative (see the references for details).

The MinMax estimator can be determined through a similar augmentation of the algebraic Riccati equation

$$\begin{aligned}
AP + PA^T - P \left[ C^T C - \theta^2 Q \right] P + DD^T = 0.
\end{aligned}
$$

is solved for $P$, and $\theta$ is taken to be as large as possible. In particular, $\Pi$ and $P$ are the minimal positive semi-definite solutions to the algebraic Riccati equations (the matrix $[I - \theta^2 P \Pi]$ must also be positive definite).

If one defines $K = R^{-1} B^T \Pi$ as above and

$$
\begin{align*}
L &= [I - \theta^2 P \Pi]^{-1} PC^T, \\
A_c &= A - BK - LC + \theta^2 DD^T \Pi
\end{align*}
$$

then the MinMax controller for the system is given by

$$
\begin{aligned}
f(t) &= -r^{-1} B^T \Pi z_c(t), \\
z_c(t) &= -K z_c(t).
\end{aligned}
$$

Observe that for $\theta = 0$ the resulting controller is the LQG (i.e., Kalman Filter) controller.

We make the following note regarding this theory as applied to PDE systems. The theory for LQG and MinMax control design exists for such systems and requires functional analysis. The application of the methods requires convergent approximation schemes to obtain systems of ODEs for computations. Although this can be done in the obvious way—approximate the PDE by finite differences, finite elements, etc.—convergence of the state equation is not enough to ensure convergence of the controller. This can be seen when one considers the algebraic Riccati equations and notes that for the PDE, the term $A^T$ is replaced by the adjoint of $A$, $A^*$. A numerical approximation scheme that converges for $A$ might not converge for $A^*$. This should not dissuade the interested reader from applying these techniques to fluid control problems. It is noted however as an issue that one must consider, and there are many results in the literature on this topic.

At this stage, the full-order LQG control is impractical for implementation, as for many applications of interest, the state estimate is quite large, making the controller intractable for real-time computation. There are many approaches to obtaining low order controllers for both infinite and finite dimensional systems. Some involve reducing the state space model before computing the controllers, while others involve computing a control for the full-order system—that is known to converge to the controller for the PDE systems—and then reducing the controller in some way. In Section 5 we discuss some reduction techniques that have been explored for low-order control design.
Figure 6: Phase plots of the uncontrolled system for $R = 5$: no disturbance (left), $d = 10^{-3}$ (right).

### 4.5 Examples

Our first example is a nonlinear system of ordinary differential equations given by

\[
\begin{align*}
\dot{x}_1(t) &= \frac{-x_1}{R} + x_2 - x_2 \sqrt{x_1^2 + x_2^2} + d \\
\dot{x}_2(t) &= -\frac{2x_2}{R} + x_1 \sqrt{x_1^2 + x_2^2} + d
\end{align*}
\]

where $R$ is a parameter to be specified. We will assume that $d$ is a small disturbance. Here, the system matrices are given by

\[
A = \begin{bmatrix} -1/R & 1 \\ 0 & -2/R \end{bmatrix} \quad N(x) = \begin{bmatrix} -x_2 \sqrt{x_1^2 + x_2^2} \\ x_1 \sqrt{x_1^2 + x_2^2} \end{bmatrix}
\]

\[
D = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad d(t) = \epsilon \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\]

Depending upon the choice of $R$ and $d$, the system has various equilibria. We will choose $R = 5$ and look at phase plots of the uncontrolled system, as shown in Figure 6. There are five equilibria: three stable, including the origin, and two unstable. When introducing a small disturbance of $d = 10^{-3}$, there is little visible change in the behavior of the system, as shown in the subfigure on the right. The green lines indicate a band of stable trajectories that are drawn to the stable equilibrium at the origin—about which the system was linearized.

If we increase to $R = 6$ and perform the same experiment, we find that the inclusion of the disturbance greatly changes the system, annihilating two of the equilibria, and dramatically decreasing the band of stable trajectories.

To apply the control methodology discussed above, we linearize the system around an equilibrium to design a control. We choose the origin about which to linearize, and design LQR and MinMax controls. The phase plot of the nonlinear system with LQR feedback control included is shown in Figure 8 (the plot is visually...
Figure 7: Phase plot of the uncontrolled system for $R = 6$, $d = 10^{-3}$, showing the annihilation of two of the equilibria seen in Fig. 6.

Figure 8: Phase plot of the LQR controlled system for $R = 5$, $d = 0$ or $d = 10^{-3}$. The goal is to stabilize the origin, and the region of attraction is the s-shaped strip between the two green lines.

identical with and without a disturbance). Note that the LQR control expands of the region that is attracted to the origin, but the sinks at the upper right and lower left regions of the plot still survive, and trajectories outside of the s-shaped region converge to these.

When the MinMax control is applied to the system, a marked change in the phase plot occurs, as shown in Figure[10]. Now, the origin is globally attracting: the sinks at upper left and lower right disappear, and all trajectories are drawn to the origin. Note that this behavior is not necessarily an anomaly of this example. That is, the power of linear feedback control to fundamentally alter—and stabilize—a nonlinear system is often underestimated.

Our second example involves a PDE model with limited sensed measurements and actuation. This is a model for the vibrations in a cable-mass system. The cable is fixed at the left end, and attached to a vertically vibrating mass at the right, as shown in Figure[10]. The mass is then suspended by a spring that has a nonlinear
stiffness term. We assume that the only available measurements are the position and velocity of the mass, and that a force is applied at the mass to control the structure. In addition, there is a periodic disturbance applied at the mass of the form \( d(t) = \alpha \cos(\omega t) \). The equations for this system are given below:

\[
\begin{align*}
\rho \frac{\partial^2 w(t, s)}{\partial t^2} &= \frac{\partial}{\partial s} \left[ \tau \frac{\partial}{\partial s} w(t, s) + \gamma \frac{\partial^2}{\partial t \partial s} w(t, s) \right] \\
m \frac{\partial^2 w(t, \ell)}{\partial t^2} &= -\left[ \tau \frac{\partial}{\partial s} w(t, \ell) + \gamma \frac{\partial^2}{\partial t \partial s} w(t, \ell) \right] - \alpha_1 w(t, \ell) \\
&\quad - \alpha_3 [w(t, \ell)]^3 + d(t) + u(t) \\
\end{align*}
\]

\[ w(t, 0) = 0. \quad (50) \]
\[ w(0, s) = w_0(s), \quad \frac{\partial}{\partial t} w(0, s) = w_1(s). \quad (51) \]

We refer the interested reader to Burns and King\textsuperscript{25} for discussion regarding the approximation scheme applied to the problem for computational purposes.

For this problem, we linearize the system about the origin, and design LQG and MinMax controllers with estimators based on the sensed measurements. In Figure\textsuperscript{[11]} the mass position with respect to time is shown for the two control designs. We see a greater attenuation of the disturbance with MinMax control.
Figure 11: Mass position: LQG controlled (left), MinMax controlled (right)

Figure 12: Mass position: LQG control (left), and MinMax control (right), comparing uncontrolled case (· · ·) with controlled (—).

In Figure 12, we show the phase plot of the controlled mass as compared to the uncontrolled mass. Again, note the greater disturbance attenuation under MinMax control.

It might not be too surprising that the MinMax control does a better job of controlling the mass—the part of the system where the measurements are taken and where actuation occurs. Figure 13 shows behavior of the mid-cable, and we see the great attenuation of the MinMax control once again.

5 Model reduction

The methods of analysis and control synthesis discussed in the previous two sections rely on knowledge of a model of the system, as a system of ordinary differential equations (ODEs), represented either as a transfer function or in state-space form. However, the equations governing fluid motion are not ODEs—they are partial differential equations (PDEs), and even if the resulting PDEs are discretized and written...
in the form \( \mathcal{O} \), the number of states will be very large, equal to the number of gridpoints \( \times \) flow variables, which typically exceeds \( 10^6 \). While it is possible to compute optimal controllers for the Navier-Stokes equations even in full three-dimensional geometries,\(^{16,27}\) in order to be used in practice these approaches would require solving multiple Navier-Stokes simulations faster than real time, which is not possible with today’s computing power.

However, for many fluids problems, one does not need to control every detail of a turbulent flow, and it is sufficient to control the dominant instabilities, or the large-scale coherent structures. For these simpler control problems, it often suffices to use a simplified numerical model, rather than the full Navier-Stokes equations. The process of model reduction involves replacing a large, high-fidelity model with a smaller, more computationally tractable model that closely approximates the original dynamics in some sense.

The techniques described in this section give an overview of some methods of model reduction that are useful for fluid applications. However, we emphasize that obtaining reduced-order models that accurately capture a flow’s behavior even when actuation is introduced remains a serious challenge, and is a topic of ongoing research. In many turbulent flows, there may not even exist a truly low-dimensional description of the flow that will suffice for closed-loop control. Thus, the techniques discussed here are most likely to be successful when one has reason to believe that a low-dimensional description would suffice: for instance, flows dominated by a particular instability, or flows that exhibit a limit cycle behavior (see Sec. 6.2 for a discussion of limit cycles).

In this section, we first discuss Galerkin projection onto basis functions determined by Proper Orthogonal Decomposition (POD), a method used by Lumley,\(^{28}\) Sirovich,\(^{29}\) Aubry et al.\(^{30}\), Holmes et al.\(^{31}\), and others. Next, we discuss balanced truncation,\(^{32}\) a method which has been applied to fluids comparatively recently\(^{33-35}\). There are many other methods for model reduction, such as Hankel norm reduction\(^{36}\), which has even better performance guarantees than balanced truncation. However, most other methods scale as \( n^3 \), and so are not computationally tractable for large fluids simulations where \( n \sim 10^6 \). See Antoulas et al.\(^{37}\) for an overview of several different model reduction procedures for large-scale systems.

Figure 13: Mid-cable phase plot: LQG control (left), MinMax control (right)
5.1 Galerkin projection

We begin with dynamics that are defined on a high-dimensional space, say \( q \in \mathbb{R}^n \) for large \( n \):

\[
\dot{q} = F(q, f), \quad q \in \mathbb{R}^n
\]  

(52)

where as before, \( f \) denotes an input, for instance from an actuator. Typically, trajectories of the system (52) do not explore the whole phase space \( \mathbb{R}^n \), so we may wish to approximate solutions by projections onto some lower-dimensional subspace \( S \subset \mathbb{R}^n \). Denoting the orthogonal projection onto this subspace by \( P_S : \mathbb{R}^n \rightarrow S \subset \mathbb{R}^n \), Galerkin projection defines dynamics on \( S \) simply by projecting:

\[
\dot{q} = P_S F(q, f), \quad q \in S.
\]  

(53)

For instance, if \( S \) is spanned by some basis functions \( \{\varphi_1, \ldots, \varphi_r\} \), then we may write

\[
q(t) = \sum_{j=1}^r a_j(t) \varphi_j,
\]  

(54)

and the reduced equations (52) have the form

\[
\sum_k \langle \varphi_j, \varphi_k \rangle \dot{a}_k(t) = \langle \varphi_j, F(q(t), f(t)) \rangle,
\]  

(55)

which are ODEs for the coefficients \( a_k(t) \). Here, \( \langle \cdot, \cdot \rangle \) denotes an inner product, which is just dot product for vectors, but other inner products may be used. If the basis functions are orthogonal, then the matrix formed by \( \langle \varphi_j, \varphi_k \rangle \) is the identity, so the left-hand side of (55) is simply \( \dot{a}_j \).

Note that this procedure involves two choices: the choice of the subspace \( S \), and the choice of the inner product. For instance, if \( v \) and \( w \) are in \( \mathbb{R}^n \) (regarded as column vectors), and if \( \langle v, w \rangle = v^T w \) denotes the standard inner product, then given any symmetric, positive-definite matrix \( Q \), another inner product is

\[
\langle v, w \rangle_Q = v^T Q w.
\]  

(56)

Different choices of inner product lead to quite different dynamics, and a suitable choice can guarantee certain useful properties. For instance, if one chooses an “energy-based” inner product (such that the “energy” \( \|q\|_Q^2 = \langle q, q \rangle_Q \) is conserved or decreases), then the reduced-order model (55) is guaranteed to preserve stability of an equilibrium point at \( q = 0 \), something that is not guaranteed otherwise. We will see that balanced truncation corresponds to choosing a particularly useful inner product that also satisfies this property.

5.2 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition (POD) is one method of determining basis functions \( \varphi_j \) to use for Galerkin projection as described in the previous section. In particular, POD determines modes that are optimal for capturing the most energetic features in a given data set.
Because we are often interested in partial differential equations, let us assume we are given a set of data \( \mathbf{q}(\mathbf{x}, t) \), a vector of flow variables \( \mathbf{q} \), as a function of space \( \mathbf{x} \) and time \( t \) (here, \( \mathbf{x} \) may be a vector). We wish to approximate \( \mathbf{q} \) as a sum

\[
\tilde{\mathbf{q}}(\mathbf{x}, t) = \sum_{j=1}^{n} a_j(t) \varphi_j(\mathbf{x}),
\]

where the functions \( \varphi_j(\mathbf{x}) \) are fixed, vector-valued basis functions (modes), and we will in addition assume these modes are orthonormal. Note that because the modes \( \varphi_j \) in (57) are fixed, the flow at any time \( t \) is completely specified by the coefficients \( a_j(t) \). These coefficients may be computed from our data \( \mathbf{q}(\mathbf{x}, t) \) using orthonormality:

\[
a_j(t) = \langle \mathbf{q}(\mathbf{x}, t), \varphi_j(\mathbf{x}) \rangle = \int_{\Omega} \mathbf{q}^T(\mathbf{x}, t) \varphi_j(\mathbf{x}) \, d\mathbf{x}
\]

where \( \Omega \) is the spatial domain. We want to find modes that minimize the average error between the projected data \( \tilde{\mathbf{q}} \) and the original data \( \mathbf{q} \).

One can show (e.g., using a variational argument) that the optimal modes are solutions to the infinite-dimensional eigenvalue problem

\[
R \varphi(\mathbf{x}) = \lambda \varphi(\mathbf{x}), \quad R \varphi(\mathbf{x}) = \int_{\Omega} \overline{\mathbf{q}(\mathbf{x}, t) \mathbf{q}^T(\mathbf{x}', t)} \varphi(\mathbf{x}') \, d\mathbf{x}'.
\]

Here, \( R \) is an operator that takes a function of space \( \varphi(\mathbf{x}) \) and creates another function of space, given by the right-hand side of (59), and the overbar represents an appropriate average (e.g., time average). The eigenvalues \( \lambda \) represent the “energy captured” by each POD mode.

In finite dimensions, the integral in (59) becomes a sum, and the POD modes may be found by standard eigenvalue solvers or by singular value decomposition. For instance, suppose the data is a scalar variable \( q \) in a single spatial dimension \( x \), given at certain points in space and time as \( q(x_j, t_k) \), with \( j = 1, \ldots, n \), \( k = 1, \ldots, m \). Then the POD modes are the eigenvectors of the real symmetric matrix

\[
R_{ij} = \frac{1}{m} \sum_{k=1}^{m} q(x_i, t_k) q(x_j, t_k).
\]

This is an \( n \times n \) eigenvalue problem, but if the number of snapshots \( m \) is smaller than the number of gridpoints \( n \), it is more efficient to compute the POD methods using the method of snapshots, described below.

First, form the data matrix \( A_{jk} = q(x_j, t_k) \), whose columns are the snapshots:

\[
A = \begin{bmatrix}
q(x_1, t_1) & q(x_1, t_2) & \cdots & q(x_1, t_m) \\
q(x_2, t_1) & q(x_2, t_2) & \cdots & q(x_2, t_m) \\
\vdots & \vdots & \ddots & \vdots \\
q(x_n, t_1) & q(x_n, t_2) & \cdots & q(x_n, t_m)
\end{bmatrix}.
\]

Then the following are equivalent:
1. The POD modes are the eigenvectors of the $n \times n$ matrix $R = \frac{1}{m} AA^T$ (the 
\textit{direct method}).

2. The POD modes are given by $\varphi_j = Ac_j$ where $c_j \in \mathbb{R}^m$ are eigenvectors of the $m \times m$ matrix $M = \frac{1}{m} A^T A$ (the \textit{method of snapshots}).

3. The POD modes are left singular vectors of $A$. That is, writing a singular value decomposition $A = U\Sigma V^*$, the POD modes are the columns of $U$.

For problems in two or three spatial dimensions, one can stack values of $q$ at all gridpoints into a single column vector in each column of $A$ in this case, the order in which the gridpoints are stacked is not important.) If $q$ is a vector, then one may compute separate sets of modes for each flow variable, or a single set of vector-valued modes. The latter approach is always advantageous for incompressible flows, since if the individual snapshots satisfy the continuity equation $\nabla \cdot q(x, t) = 0$, then the individual vector-valued modes will also satisfy continuity. Vector-valued modes have also been shown to behave better in examples in compressible flow as well, although here one must be careful to define a suitable inner product involving both velocities and thermodynamic variables.

For more in-depth tutorials of POD, see Chatterjee or Holmes et al. There are many other extensions to this basic approach as well, including methods for computing modes from incomplete data, traveling POD modes, scaling POD modes for self-similar solutions, and shift modes for better capturing transient behavior.

5.3 Balanced truncation

The POD/Galerkin approach has been successful for a large number of flow problems. However, the method is often quite fragile: the models depend unpredictably on the number of modes kept, and often a large number of modes is required to capture qualitatively reasonable dynamics. One potential problem with the philosophy of POD/Galerkin models is that the choice of modes is based on retaining the most energetic features, and low-energy phenomena (such as acoustic waves) may be important to the overall dynamics. Balanced truncation is another (related) method of model reduction, which has been widely used in the control theory community. It is not based solely on energetic importance, and seems to produce more reliable models than POD/Galerkin in examples. The method also has provable error bounds that are close to the minimum possible for any reduced-order model of a given order.

Balanced truncation applies to linear input-output systems, for instance of the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$. The philosophy of balanced truncation is to truncate the modes that are least controllable (that are the least excited by inputs $u$), and are least observable (that have the smallest effect on future outputs $y$). The degree of controllability and observability is defined quantitatively in terms of the Gramians defined in . In the original coordinates, these two criteria may be contradictory, so balanced truncation proceeds by first transforming to coordinates in which the controllability and
observability Gramians are equal and diagonal, via a balancing transformation. It is always possible to find such a change of coordinates as long as the original system is both controllable and observable. In the transformed coordinates, one then keeps only the first few modes, that are the most controllable and observable.

For algorithms for computing balancing transformations, see Dullerud and Pagani[1, 2], or Datta[3]. The standard approach involves first solving the Lyapunov equations [1, 2], and then computing a transformation that simultaneously diagonalizes the Gramians. Because the Gramians are full (non-sparse) $n \times n$ matrices, this approach becomes computationally intractable for large $n$, as occurs in many fluids problems. For this reason, the snapshot-based approach in Lall et al. [4] can be useful, and an algorithm for computing the balancing transformation directly from snapshots, without first computing the Gramians, is also available[5, 6].

6 Nonlinear systems

While the tools we have studied throughout most of this chapter have addressed linear systems, virtually all real systems are nonlinear. Fortunately, as we have seen in the examples in Section 4.5, linear techniques often work surprisingly well even for nonlinear systems. However, it is helpful to understand characteristics of nonlinear systems, both to understand the limitations of linear control methods, and to better understand the behavior of real systems.

Here, we give only an elementary overview of the most basic aspects of nonlinear systems. For a more detailed introduction, see Hirsch et al. [7], and for a more advanced treatment, see the references [8, 9, 10, 11, 12].

6.1 Multiple equilibria and linearization

Nonlinear systems may be written in the general form

$$\dot{q} = F(q, f, \mu, t),$$

(62)

where $q(t)$ and $f(t)$ are the state vector and input, as in Section 4, and $\mu \in \mathbb{R}^p$ is a vector of parameters (for instance, Reynolds number). In order to apply the control methods discussed in the first part of this chapter, we must rewrite (62) in the form (5). To do this, we first identify equilibrium points of the system (62), which are points where $\dot{q} = 0$, or $F(q, f, \mu, t) = 0$. While linear systems typically have only one equilibrium point (at $q = 0$), nonlinear systems may have multiple equilibria, as illustrated by the example in Section 4.5.

To simplify notation, we will assume that the system does not depend explicitly on time, and for the moment we will suppress the dependence on the parameter $\mu$, writing $\dot{q} = F(q, f)$. Suppose that $(q^*, f^*)$ is an equilibrium point of (62), so that $F(q^*, f^*) = 0$. Expanding $F$ in a Taylor series about $(q^*, f^*)$, we have

$$F(q^* + \delta q, f^* + \delta f) = F(q^*, f^*) + \frac{\partial F}{\partial q}(q^*, f^*) \cdot \delta q + \frac{\partial F}{\partial f}(q^*, f^*) \cdot \delta f + \text{Higher order terms},$$

(63)
where

\[
\frac{\partial F}{\partial q}(q^*, f^*) \equiv \begin{bmatrix}
\frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\
\frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n}
\end{bmatrix}
\]  \quad q = q^*, f = f^*
\]

is the Jacobian matrix of \( F \), evaluated at the point \( (q^*, f^*) \) (also called the derivative at \( (q^*, f^*) \)), and often denoted \( DF(q^*, f^*) \), and \( \partial F/\partial f \) is the matrix of partial derivatives with respect to the components of \( f \). Note that we evaluate these matrices of partial derivatives at the equilibrium point \( (q^*, f^*) \), so these are just constant matrices.

In linearization, we neglect the higher-order terms, so letting

\[
A = \frac{\partial F}{\partial q}(q^*, f^*), \quad B = \frac{\partial F}{\partial f}(q^*, f^*),
\]

the equation \( \dot{q} = F(q, f) \) becomes

\[
\dot{q}^* + \delta q = F(q^*, f^*) + A \cdot \delta q + B \cdot \delta f.
\]

Now, since \( (q^*, f^*) \) is an equilibrium point, \( F(q^*, f^*) = 0 \), and the equation becomes

\[
\dot{\delta q} = A \cdot \delta q + B \cdot \delta f,
\]

which is of the form (5).

If one wishes to include nonlinear terms, as in (2), one defines

\[
N(\delta q, \delta f) = F(q^* + \delta q, f^* + \delta f) - A \delta q - B \delta f.
\]

Note that in general, \( N \) in (2) may depend on \( \delta f \) as well as \( \delta q \), although the control-affine form given in (2) is quite common in practice.

Similarly, if the output \( y \) is a nonlinear function \( y = G(q, f) \), the output equation may be similarly linearized, writing \( y = y^* + \delta y \), with \( y^* = G(q^*, f^*) \), and

\[
\delta y = C \delta q + D \delta f,
\]

where

\[
C = \frac{\partial G}{\partial q}(q^*, f^*), \quad D = \frac{\partial G}{\partial f}(q^*, f^*).
\]

### 6.2 Periodic orbits and Poincaré maps

Nonlinear systems that arise in fluid mechanics often exhibit periodic orbits, which are simply periodic solutions \( q(t) \) of (2). When these periodic orbits are stable (or stable in reverse time), they are called limit cycles. Examples include vortex shedding in the wakes of bluff bodies, or oscillations in aeroelasticity problems.
A common tool for studying periodic orbits is the Poincaré map, which is defined as follows: if the overall phase space has dimension \( n \) \((q \in \mathbb{R}^n)\), one takes an \( n - 1 \)-dimensional cross section \( \Sigma \) of phase space, and defines a map \( \Phi : \Sigma \rightarrow \Sigma \) where \( \Phi(q_0) \) is found by evolving the system \( \dot{q} = F(q) \) forward in time with \( q(0) = q_0 \), until \( q(t) \) once again intersects \( \Sigma \). The point \( \Phi(q_0) \) is then defined as this point of intersection, as illustrated in Figure 14.

The system dynamics may then be represented by the discrete-time system

\[
q_{k+1} = \Phi(q_k),
\]

and periodic orbits in the original system \( \dot{q} = F(q) \) are equilibrium points (where \( \Phi(q) = q \)) of the discrete-time system \( q_{k+1} = \Phi(q_k) \). Approaches are also available for including an input in such systems, and linearizing about the periodic orbit (see, for instance, Farhood et al. 54).

### 6.3 Simple bifurcations

Often, when parameters are varied, one observes qualitative changes in phenomena. In fluid mechanics, these qualitative changes are referred to as transitions; in dynamical systems, these are referred to as bifurcations. Here, we give a brief overview of some common bifurcations. For more information, see standard texts in dynamical systems 3;50;51.

In a precise sense, a bifurcation occurs for the differential equation

\[
\dot{q} = F(q, \mu)
\]

or the map

\[
q_{k+1} = F(q_k, \mu)
\]

for a parameter value \( \mu \) for which the system is not structurally stable. We will not define structural stability in a precise sense here (see Guckenheimer and Holmes 50).
for a definition), but loosely speaking, a system is structurally stable if its phase portrait remains qualitatively unchanged when the system itself (e.g., the right-hand side of (71)) is perturbed. Note that this concept is different from stability of an equilibrium point (or periodic orbit), in which one considers perturbations in the initial conditions, rather than the system itself.

**Types of bifurcations** Two different categories of bifurcations can arise in dynamical systems. **Local bifurcations** of equilibria are the simplest to analyze, and occur when equilibrium points change stability type. An example from fluid mechanics is a laminar flow losing stability when the Reynolds number increases. When this occurs, new phenomena can appear, such as new equilibria, or new periodic orbits. For instance, when the Reynolds number is increased for the flow around a cylinder, the initially stable steady flow (an equilibrium point) transitions to a Karman vortex street (a periodic orbit). These local bifurcations can be studied by analyzing the behavior of the system near the equilibrium point of the differential equation (71) or map (72). Another category of bifurcation is **global bifurcations**, which involve qualitative changes in the phase portrait, without equilibria changing stability type. For instance, periodic orbits may appear or disappear, and invariant manifolds may change the way they intersect with one another. These are usually more difficult to detect and analyze.

Here, we will discuss only the simplest types of local bifurcations, those arising generically when there is a single parameter $\mu \in \mathbb{R}$ (codimension-1 bifurcations). In order for a bifurcation to occur at the point $(q^*, \mu^*)$, two conditions must be met:

1. The linearization $(\partial F/\partial q)(q^*, \mu^*)$ must have an eigenvalue on the imaginary axis (or, for maps, on the unit circle).

2. The eigenvalue must cross the imaginary axis (or unit circle) with nonzero speed.

For differential equations, when a real eigenvalue crosses the imaginary axis, several different types of bifurcations can occur, including saddle-node, transcritical, and pitchfork bifurcations. These can be distinguished by checking degeneracy conditions (conditions on the higher derivatives of $F$; see Guckenheimer and Holmes[21]). When a pair of complex eigenvalues crosses the imaginary axis, a Hopf bifurcation occurs, and in this case, a one-parameter family of periodic orbits is generated in the neighborhood of the bifurcation point. All of these types of bifurcations occur for maps as well. In addition, another type of bifurcation occurs for maps when an eigenvalue crosses the unit circle at the point $-1$: a flip, or period-doubling bifurcation. This type of bifurcation is common in fluids (for instance, arising as bifurcations of a Poincaré map), and an illustration of this type of bifurcation is shown in Fig. 15. Note that after the period doubling bifurcation occurs, a periodic orbit of the original period still exists, but is unstable.

6.4 Characterizing nonlinear oscillations
Figure 15: Period-doubling (flip) bifurcation: The plots at the left show a time history of one of the state variables before (top) and after (bottom) the bifurcation; the center column shows the phase plane, and the Poincaré map before and after bifurcation; the right column shows the location of the corresponding eigenvalue of the Poincaré map crossing the unit circle.
Oscillations occur in a wide variety of fluid systems, and if one wants to control these oscillations, it is important to understand the fundamental mechanisms that produce them. One common mechanism is an equilibrium point (steady solution of Navier-Stokes) becoming unstable for a particular value of a parameter, and producing a stable limit cycle through a Hopf bifurcation, as described in the previous section. Limit cycles are commonly observed in aeroelasticity problems, and occur in cavity flows, cylinder wakes, and many other types of problems.

Another common mechanism is fundamentally linear in nature: an equilibrium may be stable, but lightly damped (with eigenvalues close to the imaginary axis), leading to strong resonant frequencies. If such a system is forced by noise, it can exhibit narrow-band oscillations as well, in much the same way that a tuning fork would continue to ring at a single frequency if it were continually forced at random frequencies. For fluid applications, possible sources of this forcing include acoustic excitation, wall roughness, or boundary-layer turbulence. This mechanism of oscillations in fluids has been discovered in a variety of applications as well, including combustion instabilities, flutter in turbomachinery, and cavity flows.

These two mechanisms of oscillation suggest fundamentally different types of control strategies. In the first scenario, possible control objectives are stabilization of an unstable equilibrium point, or reduction in amplitude of a limit cycle. In the second scenario, the equilibrium point is already stable, and to reduce the amplitude of oscillations, the control strategy should focus on attenuation of disturbances.

It is often difficult to distinguish between these two types of oscillations, purely from spectra: both are characterized by sharp resonant peaks. However, there are methods for distinguishing these based purely on measurements. One first identifies a peak in the spectrum of a time series, and passes the data through a narrow-band filter about the frequency of the peak. One then plots the probability density function (PDF) of the bandpass-filtered data. If the system possesses a limit cycle, more time will be spent at extrema of the limit cycle, so the PDF will exhibit two peaks, near the extrema; conversely, if the system is lightly damped and driven by noise, then more time will be spent near the average value, and the PDF will exhibit a single peak about zero. In this manner, one can distinguish between these two mechanisms of oscillation, and determine the most appropriate control strategy.

6.5 Methods for control of nonlinear systems

The discussions of control design tools in Section 4 have been restricted to applications to linear systems. Although that may seem overly restrictive, that is the starting point for many real control systems. A basic approach to developing nonlinear controllers is to take the nominal design from an LQG orMinMax estimator-based control and to augment it with a nonlinear term. This approach is referred to as forming an extended filter. This extension could be performed for a series of set points, obtaining a family of controls. These controls can then be applied when the system is near the various operating conditions; such an approach is called gain scheduling.

Other control design methodologies are specifically tailored to nonlinear systems,
and for an introduction to these see Khalil. Many of these techniques apply to restricted classes of systems (e.g., Hamiltonian or Lagrangian systems, or systems with particular types of nonlinearities), and so far have not been widely used in the flow control community, where models are typically too large or messy for these techniques to apply, but they may become more widely used as the models for flow control problems improve.

**Receding horizon control**  One approach for nonlinear systems involves solving an optimal control problem, for instance minimizing a cost function similar to \( J \), but using the full nonlinear dynamics. For a nonlinear system, one typically uses a gradient-based search to find a value of \( f(t) \) that gives a local minimum of the cost function, for a particular value of the initial state \( q(0) \). This yields an open-loop control law valid only for this specific initial condition. In practice, one typically evaluates the cost function \( J \) over a finite time horizon, and then recomputes the optimal open-loop control \( f(t) \) as often as possible, using an updated version of the initial state, in order to account for disturbances or model uncertainties. This approach, in which the nonlinear optimal control solution is computed and recomputed in real time, is called *receding horizon control*, and is an effective but computationally intensive approach. For more information on this approach, see Stengel or Åström and Murray.

**Adaptive control and extremum seeking**  Another type of nonlinear control design that can be applied to linear design to augment the performance and robustness of the controller is adaptive control. A strategy that has been particularly successful in recent experiments, especially for situations requiring optimal tuning of a largely open-loop strategy, is extremum seeking. This is an approach where one measures a quantity one wants to maximize or minimize (e.g. minimize drag), and adjusts the value of a control parameter to extremize this quantity. The tuning is performed by superimposing small variations on the control parameter, and slowly adjusting the value of the control parameter depending on whether the quantity to be extremized is in phase or out of phase with the variations of the control parameter. This is an old technique, dating back to the 1950s, but it is only recently that stability of these techniques has been proven for certain classes of systems. This approach has the advantage that it is completely model-free, and so tends to be very robust, though it is not appropriate when control is needed at timescales similar to those of the natural plant dynamics.

There are many good references on other adaptive control strategies, and we refer the interested reader to Haykin.

7 **Summary**

In this chapter, we have given an overview of many topics from control theory, emphasizing the topics that we believe are the most relevant for design of feedback controllers for flow control applications. In particular, the techniques described in
this chapter are useful primarily for understanding closed-loop control systems, as defined in Chapter 3, Section III. While the majority of the flow control work to date has involved open-loop control, closed-loop control is becoming increasingly common, and the ideas in this chapter (for instance, Secs. 2.1 and 3.5) discuss the potential benefits and limitations of feedback control. The next chapter (in particular, section III.B) also discusses sensor requirements specific to closed-loop control designs.

Later chapters of this book contain several examples in which closed-loop control has been used in practice. For instance, Chapter 9, Sec. 5 discusses closed-loop control for turbomachinery applications, and Chapter 10 discusses a number of applications to the control of combustion instabilities.
References


37


