2

Control Crash Course

A single chapter can hardly do justice to the amazing universe of control theory and practice. The textbook [15] gives an accessible introduction to the philosophy and practice of control, and is also full of history. I was fortunate to be at the Simons Institute at Berkeley, California, when one of the authors presented a two-part survey of ideas in the book.[1] These lectures are a great starting point if you are new to control systems and will inspire many old-timers.

2.1 You Have a Control Problem

You surely have encountered control problems in your daily life. If you know how to drive, then you know what it is like to be part of a control system:

$\boldsymbol{y}$ The *observations* (also called the "output") refers to the data you process in order to effectively maneuver the car through traffic: this includes your view of the streets and lights, and the sounds of angry drivers pleading with you to adjust your speed.

$\boldsymbol{u}$ You apply *inputs* to the system: steering wheel, brakes, and gas pedal are continuously adjusted based on your observations.

ϕ This symbol will be used to denote an algorithm that takes in the observations y and produces the response u. This mapping from y to u is known as a *policy*, and sometimes called a *feedback law* (the Greek letter is pronounced "*fee*").[2]

ff You are not simply reacting to horns and lights and the lines on the road. You started off with a plan: get to the farmers' market by 9 a.m. while avoiding the traffic downtown due to the demonstration. This planning is an example of *feedforward* control. Planning is based on forecasts, so inevitably plans will change as you gather information en route: traffic updates, or an invitation from a close friend to park your car and join the protest.

The feedforward component is typically defined with attention to a *reference signal* $\boldsymbol{r}$. The primary control objective is the *tracking problem*: construct a policy so that some object of interest $z(k)$ is approximately equal to $r(k)$ for all $k \geq 0$ (in control courses, it is often assumed that $z = y$).

The yelling and bumps on the road are collectively known as *disturbances*. Along with the reference signal, partial measurements of disturbances and their forecasts are taken into

[1] https://simons.berkeley.edu/talks/murray-control-1.

[2] My apologies to those accustomed to the symbol π. This is reserved for the irrational number.

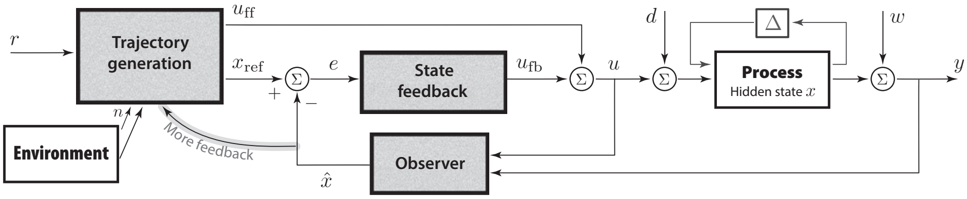


Figure 2.1 Control systems contain purely reactive feedback, as well as planning that is regularly updated. This represents two layers of feedback, differentiated in part by speed of response to new observations. These observations are often limited, so that we require estimates $\widehat{x}$ of a partially "hidden" state process x.

account in both the feedforward and the feedback components of the control system. The final input is often defined as the sum of two components:

$$u(k) = u_{\mathrm{ff}}(k) + u_{\mathrm{fb}}(k), \tag{2.1}$$

where in the shopping problem, u_{ff} quantifies the results of planning before heading to the market (perhaps with updates every 20 minutes), and u_{fb} is the second-by-second operation of the automobile.

The dream of RL is to mimic and surpass the skill using which humans create an internal algorithm ϕ to skillfully navigate through complex and unpredictable environments.

Figure 2.1 shows a block diagram typically used in model-based control design and illustrates a few common design choices: there is a state to be estimated using an **observer**, with state estimates denoted $\widehat{x}$. The block denoted **trajectory generation** constructs two signals: the feedforward component of the control, and also a reference x_{ref} that an internal state should track (the state is associated with the physical process). It is designed so that $x(k) = x_{\mathrm{ref}}(k)$ for all k implies that the tracking problem is solved. The **state feedback** is designed to achieve this ideal.

There is a larger "world state" labeled **environment**, for which partial measurements are available, and forecasts of future events. Forecasts are of course important in the planning process that is part of trajectory generation.

The design of the three gray blocks is based on models of the process, the measurement (or sensor) noise w, the disturbances (such as the "input disturbance" d indicated in the figure), and a model of the environment. The "Δ-feedback loop" is a standard way to represent model uncertainty associated with the process to be controlled. This feature may be motivated by an unfortunate story.

2.1.1 Failure of an Adaptive Control System

Beginning in the 1950s, control theorists in partnership with the US Air Force looked for model-free approaches to flight control. From this came the "MIT rule," which may be regarded as an early attempt at adaptive control or "actor-only" reinforcement learning. Analysis of the MIT rule in [240] is based on techniques similar to the ODE method that is a foundation of this book. See [280] for a more recent study.

Preliminary simulations showed promise, as did field tests on the X-15 airplane. Some quotes from the 1970 report [300] hint at the enthusiasm of scientists and pilots involved:

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

> 1) Nearly invariant response was provided at essentially all aerodynamic flight conditions 2) accurate a priori knowledge of aircraft aerodynamic characteristics was not required to design a satisfactory system 3) aircraft configuration changes were adequately compensated for 4) the dual redundant concept provided a reliable and fail-safe system.

It was also noted that the adaptive control system "inspired confidence and allowed the pilot to spend time cross-checking flight instruments, checking subsystems, and 'sight-seeing.'"

These observations followed 65 test flights.

The control system was not robust enough to provide stable control in all situations encountered. Sadly, a pilot died in a crash attributed to oscillations induced by the adaptive system. The research program was shut down, but the tragedy inspired greater attention to robustness in control design.

It should go without saying that *every control engineer or practicing economist must study failures*. Airplanes and economies inevitably crash. In the long run, it is a greater tragedy if the experts do not bother to learn from disaster.

2.2 What to Do about It?

The vast literature on control solutions is built upon a model of input–output behavior that is used to design the policy ϕ. Modeling and control design are each an art form, with many possible solutions from vast statistics and control tool chests.

When we say *model*, we mean a sequence of mappings from inputs to outputs:

$$y(k) = \mathrm{G}_k(u(0),u(1),u(2),\ldots,u(k)), \qquad k \geq 0. \tag{2.2}$$

Each of the functions G_k may also depend on *exogenous variables* (outside of our control), such as the weather and traffic conditions. And here we come to one of the most vital principles of control design: *the model must capture essential properties of the system to be controlled*, and simultaneously *be simple enough to be useful*.

For example, aerospace engineers will create *absurdly simple models* for the design of flight control systems and from this create a policy ϕ designed to work well for the model. Of course, they do not stop there. The next step is to create an entirely new model for validation and simulate under a range of scenarios in order to answer a range of questions: What happens when the plane is full, empty, or flying through a thunderstorm? How does the control system perform after an engine detaches from a wing? If one of these tests fails, then the control engineer goes back to either improve the model, improve the policy, or *improve the airplane*. That's right, we may require additional sensors to measure pitch angles, or more powerful motors to control ailerons, flaps, or elevators.

I am writing without any knowledge of aerospace engineering. I am describing general principles for anyone interested in control design:

(1) Create a model for control.
(2) Design the policy ϕ based on the model.
(3) Simulate based on a high-fidelity model, and then revisit steps 1 and 2.

The success of this approach has been tremendous, as seen in the history recounted in [15].

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Linear and Time Invariant Model. The most successful class of *absurdly simple models* are linear and time invariant (LTI). The general scalar LTI model is defined by a sequence of scalars $\{b_i\}$ (the *impulse response*), and for a given scalar input sequence $\boldsymbol{u}$, the model defines $y(k)$ as the sum

$$y(k) = \sum_{i=0}^{k} b_i u(k-i) \,, \qquad k \geq 0. \tag{2.3}$$

This is in fact too complex in many situations. A more tractable subclass of LTI models are autoregressive moving-average (ARMA): for scalar coefficients $\{a_i, b_i\}$,

$$y(k) = -\sum_{i=1}^{N} a_i y(k-i) + \sum_{i=0}^{M} b_i u(k-i) \,, \qquad k \geq 0. \tag{2.4}$$

A linear input–output model motivates the design of a policy ϕ that has a similar linear form. A common design technique based on optimization will be described in the following chapters, beginning in Section 3.6.

2.3 State Space Models

2.3.1 Sufficient Statistics and the Nonlinear State Space Model

In statistics, the term *sufficient statistic* is used to denote a quantity that summarizes all past observations. The *state* plays an analogous role in control theory.

A *state space model* requires the following ingredients: the *state space* X on which the state $\boldsymbol{x}$ evolves, and an *input space* (or *action space*) denoted U on which the input $\boldsymbol{u}$ evolves. We may have additional constraints coupling the state and the input, which is modeled via

$$u(k) \in \mathsf{U}(x) \,, \qquad \text{when } x(k) = x \in \mathsf{X}, \tag{2.5}$$

with $\mathsf{U}(x) \subseteq \mathsf{U}$ for each x. We might also want to model an observation process $\boldsymbol{y}$ evolving on a set Y. In the control theory literature, it is common to assume that X, U, and Y are subsets of Euclidean space, while in operations research and reinforcement learning it is more common to assume these are finite sets. Whenever possible, in this book we prefer the control perspective so that we can more easily search for structure of control solutions: For example, is an optimal input a continuous function of the state?

Next we require two functions $\mathrm{F}\colon \mathsf{X}\times\mathsf{U} \to \mathsf{X}$ and $\mathrm{G}\colon \mathsf{X}\times\mathsf{U} \to \mathsf{Y}$ that define the following state equations:

$$x(k+1) = \mathrm{F}(x(k), u(k)), \qquad x(0) = x_0, \tag{2.6a}$$
$$y(k) = \mathrm{G}(x(k), u(k)). \tag{2.6b}$$

An LTI model can often be transformed into a state space model in which the two functions F, G are linear in (x, u).

We might also allow F, G to depend upon the time variable k. It is argued in Section 3.3 that it is often more convenient to simply assume that the state $x(k)$ includes k as one component.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

However, there is one example of a time-dependent model that highlights the role of state as a sufficient statistic. The general input–output model (2.2) always has a state space description, in which the state is the full history of the following inputs:

$$x(k+1) = [u(0), u(1), u(2), \dots, u(k)]^{\intercal}. \tag{2.7}$$

We have $x(k+1) = \mathrm{F}_k(x(k), u(k))$, defined by concatenation, and $y(k) = \mathrm{G}_k(x(k), u(k))$ is a restatement of (2.2). For this deterministic model in which the input fully determines the output, (2.7) is called the (full) *history state*. A practical state space model can be regarded as a compression of the history state.[3]

In many cases, we can construct a good policy via *state feedback*, $u(k) = \phi(x(k))$, for some $\phi\colon \mathbb{R}^n \to \mathbb{R}$; in stochastic control, it is typical to say that ϕ is a *Markov policy* in this case. However, the power of this approach is fully realized only if we are flexible in our definition of the state. We won't be using the full history state because of complexity; what's more, *the "full history" may not be nearly rich enough.*

2.3.2 State Augmentation and Learning

Tracking and disturbance rejection are two of the basic goals in control design. Here we provide a brief glimpse of two tricks used to simultaneously track the reference $\boldsymbol{r}$ while rejecting disturbances:

(i) The definition of state is not sacred; invent a state process that simplifies control design.
(ii) Unknown quantities, including disturbances and even the state space model, can be learned based on input–output measurements.

Let's maintain our simplifying assumption that the input and output are scalar valued, and take $\mathsf{X} = \mathbb{R}^n$. The state evolution is also influenced by a scalar disturbance $\boldsymbol{d}$ that is outside our control, which requires a modification of (2.6a):

$$x(k+1) = \mathrm{F}(x(k), u(k), d(k)). \tag{2.8}$$

The ultimate goal is to achieve these three objectives simultaneously.

(a) Tracking: With $\tilde{y}(k) = y(k) - r(k)$,

$$\limsup_{k\to\infty} |\tilde{y}(k)| = e_\infty\,, \qquad \text{with } e_\infty = 0, \text{ or very small.} \tag{2.9}$$

(b) Disturbance rejection: The error e_∞ is not highly sensitive to the disturbance $\boldsymbol{d}$.
(c) Tuned transient response (you probably know what kind of acceleration "feels right" when driving a car).

A common special case is when the reference and disturbance are assumed independent of time (e.g., driving at constant speed with a steady headwind). In this special case, suppose in addition that the disturbance is known. We might choose $u(k) = \phi(x(k), r(0), d(0))$, where the policy ϕ is designed for success: $e_\infty = 0$. Typically, ϕ is designed so that the state is also convergent: $x(k) \to x(\infty)$ as $k \to \infty$. The limiting values must satisfy

[3] See [337] and its references.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

$$\begin{aligned} x(\infty) &= \mathrm{F}(x(\infty), u(\infty), d(0)), \\ u(\infty) &= \phi(x(\infty), r(0), d(0)). \end{aligned}$$

The outcome $e_\infty = 0$ is expressed as the final constraint:

$$r(0) = y(\infty) = \mathrm{G}(x(\infty), u(\infty)).$$

This approach is thus dependent on an accurate model, as well as direct measurements of $\boldsymbol{d}$.

Suppose that instead of exact measurements of the disturbance, we have a state space model whose output is $y_{\mathrm{m}}(k) = (r(k), d(k))^{\intercal}$:

$$z(k+1) = \mathrm{F}_{\mathrm{m}}(z(k)), \tag{2.10a}$$

$$y_{\mathrm{m}}(k) = \mathrm{G}_{\mathrm{m}}(z(k)), \tag{2.10b}$$

where $\boldsymbol{z}$ evolves on $\mathbb{R}^p$ for some integer $p \geq 1$. The functions $\mathrm{F}_{\mathrm{m}} : \mathbb{R}^p \to \mathbb{R}^p$ and $\mathrm{G}_{\mathrm{m}} : \mathbb{R}^p \to \mathbb{R}$ are assumed known. Part of this state description is $d(k+1) = d(k)$ if the disturbance is static.

Given the larger state space model (2.8, 2.10), we might opt for an *observer*-based solution:

$$u(k) = \phi(x(k), r(k), \hat{d}(k)),$$

where $\{\hat{d}(k)\}$ are estimates of the disturbance, based on input–output measurements up to time k (we might replace $x(k)$ with $\hat{x}(k)$ if we don't directly observe the state). Observer design makes up about 20% of a typical introductory course on state space control systems [7, 29, 76].

A second option, called the Internal Model Principle, is to create a different state augmentation that is entirely observed. For the sake of illustration, consider again the case of constant reference/disturbance. We have (2.10) in this case with $z(k) = y_{\mathrm{m}}(k)$, and F_{m} is the identity function:

$$z(k+1) = z(k).$$

State augmentation is performed based on this model: define for each k,

$$z^I(k+1) = z^I(k) + \tilde{y}(k)\,, \tag{2.11}$$

with error $\tilde{y}(k+1)$ defined above (2.9). We regard $(x(k), z^I(k))$ as the state for the purposes of control, and hence state feedback takes the form

$$u(k) = \phi(x(k), z^I(k)). \tag{2.12}$$

The control design (2.12) is an example of *integral* control, since z^I is the sum of errors (the discrete-time analog of integration).

Suppose that $z^I(k)$ converges to some finite limit $z^I(\infty)$, as $k \to \infty$; the value of the limit is irrelevant. This and (2.11) imply perfect tracking:

$$\lim_{k\to\infty} \tilde{y}(k) = \lim_{k\to\infty} [z^I(k+1) - z^I(k)] = 0.$$

This conclusion is remarkable: to obtain perfect tracking, we only require that the policy ϕ is designed so that $z^I(k)$ converges to *some* finite limit. The secret to success is a hidden element of "learning" that comes with integral control.

State augmentation has many other dimensions. If we have forecasts of significant disturbances, then it may be wise to make use of these data: forecasts can be used in the design of the feedforward component $u_{\text{ff}}(k)$ in the decomposition (2.1), or they may be used for state augmentation.

2.3.3 Linear State Space Model

When F and G are linear, we obtain the linear state space model:

$$x(k+1) = Fx(k) + Gu(k), \qquad x(0) = x_0, \tag{2.13a}$$
$$y(k) = Hx(k) + Eu(k), \tag{2.13b}$$

where (F,G,H,E) are matrices of suitable dimension (in particular, F is $n \times n$ for an n-dimensional state space).

The state space model is not unique, in the sense that there are many choices for (F,G,H,E) that result in the same input–output behavior, even though the definition of the state process $\boldsymbol{x}$ will change depending on the model. And never forget, *we may add additional components to $x(k)$ as a means to solve a control problem.*

Linear State Feedback

The linear model (2.13) is often constructed so that the goal is to keep $x(k)$ near the origin – the *regulation problem*; consider, for example, flight control, where we wish to maintain velocity and altitude at some constant values. We first normalize the problem so that these constant values are *zero*. It is then common to apply a linear control law

$$u(k) = -Kx(k), \tag{2.14}$$

where K is called the *gain matrix*. To evaluate choice of gain, we tack on something like a reference signal:

$$u(k) = -Kx(k) + v(k).$$

The *closed-loop behavior* with new “input $\boldsymbol{v}$” has a similar state space description:

$$x(k+1) = (F - GK)x(k) + Gv(k), \qquad x(0) = x_0, \tag{2.15a}$$
$$y(k) = (C - EK)x(k) + Ev(k). \tag{2.15b}$$

The signal $v(k)$ appearing in (2.15a) is viewed as an “input disturbance.” A goal of control is to choose K so that the closed-loop behavior is not very sensitive to this disturbance while simultaneously ensuring good tracking.

Realization Theory

The ARMA model (2.4) admits an infinite number of distinct state space descriptions. Let’s begin with the scalar auto-regressive model:

$$y(k) = -\sum_{i=1}^{N} a_i y(k-i) + u(k), \qquad k \geq 0$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

which is (2.4), with $M = 0$ and $b_0 = 1$. We obtain the state space model (2.13) with $n = N$ by choosing $x(k) = (y(k), \ldots, y(k-N+1))^{\intercal}$, and

$$F = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \cdots & \cdots & -a_N \\ 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.16}$$

$H = [1, 0, 0, \ldots, 0]$, and $E = 0$.

This construction can be generalized: with $M = N - 1$ in (2.4), we first define an intermediate process

$$z(k) = -\sum_{i=1}^{N} a_i z(k-i) + u(k), \qquad k \geq 0. \tag{2.17}$$

So we arrive at a state space model with state space $x(k) = (z(k), \ldots, z(k-N+1))^{\intercal}$ to describe the evolution of $\boldsymbol{z}$. We next use the assumption that $M = N - 1$: setting $u(k) = z(k) = 0$ for $k < 0$, it is possible to show that

$$y(k) = \sum_{i=0}^{N-1} b_i z(k-i) = Hx(k) + Eu(k),$$

where

$$H = [b_0, b_1, \ldots, b_{N-1}], \qquad E = 0. \tag{2.18}$$

The state space description (2.16, 2.18) is known as *controllable canonical form*. There are many other "canonical forms," with special properties you can learn about in a linear systems course [7, 15, 29, 76, 205].

2.3.4 A Nod to Newton and Leibniz

In many engineering applications, it is best to start off in continuous time, with thanks to Newton and Leibnitz for bringing us calculus.

Some notational conventions reserved for continuous time: First, time is denoted using a subscript (such as u_t rather than $u(t)$) as a reminder that time is continuous. Moreover, it is often convenient to suppress time dependency altogether, so that $\frac{d}{dt}u$ represents the derivative at an unspecified time.

The state space model in continuous time has the form

$$\tfrac{d}{dt}x = \mathrm{f}(x, u), \tag{2.19}$$

where x is the state evolving in $\mathbb{R}^n$, and u the input evolving in $\mathbb{R}^m$. The motion of a typical solution to a nonlinear state space model in $\mathbb{R}^2$ is illustrated in Figure 2.2.

When the function f appearing in (2.19) is linear, then we obtain the linear state space model in continuous time. As in (2.13), this is accompanied by an observation process $\boldsymbol{y}$ evolving on $\mathbb{R}^p$:

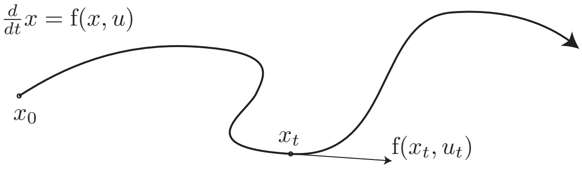


Figure 2.2 Trajectory of a nonlinear state space model in two dimensions: at any time t, the velocity $\frac{d}{dt}x_t$ is a function of the current state x_t and input u_t.

$$\frac{d}{dt}x = Ax + Bu, \tag{2.20a}$$

$$y = Cx + Du, \tag{2.20b}$$

and A, B, C, D are matrices of appropriate dimensions.

The geometry illustrated in Figure 2.2 is sometimes valuable in gaining intuition in control design (note that the vector $\mathrm{f}(x_t, u_t)$ is tangent to the state trajectory). Stability theory and optimal control theory are most attractive in the continuous time domain because of this simple geometry, and the simplicity that comes with calculus.

However, in the end we have to sample time to apply our control and learning algorithms. In this book, we will opt for an Euler approximation. For sampling interval Δ, the discrete time approximation of (2.19) is of the form (2.6a), with $\mathrm{F}(x,u) = x + \Delta\mathrm{f}(x,u)$. For the linear model (2.20a), this leads to $F = I + \Delta A$.

2.4 Stability and Performance

In this section, we consider the state space model (2.6a) in a *closed loop*: a policy ϕ is chosen, so that $u(k) = \phi(x(k))$ for each k. Since the feedback law is fixed, the state then evolves as a state space model without control. With just a slight abuse of notation, we write

$$x(k+1) = \mathrm{F}(x(k)), \qquad k \geq 0. \tag{2.21}$$

Our interest is in the long-run behavior of the state process; in particular, does it converge to an *equilibrium*? We also seek bounds on a particular performance metric called the *total cost*.

The following is assumed throughout:

$$\textit{The state space } \mathsf{X} \textit{ is equal to } \mathbb{R}^n\textit{, or a closed subset.} \tag{2.22}$$

For example, we allow the positive orthant, $\mathsf{X} = \mathbb{R}^n_+$. The restriction on the state space (2.22) is imposed so that any closed and bounded set $S \subset \mathsf{X}$ is necessarily a compact subset of X.

The definition of an equilibrium x^e is straightforward – it is a state at which the system is frozen:

$$x^e = \mathrm{F}(x^e). \tag{2.23}$$

The equilibrium will in fact be a part of the control design. Think of the cruise control in your car, in which "equilibrium" means traveling in a straight line at constant speed. The particular speed is something that you as the driver will choose. The control system then does the best it can to keep $x(k)$ near the desired value x^e.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

2.4.1 Total Cost

This performance metric is based on a function $c\colon \mathsf{X} \to \mathbb{R}_+$, interpreted as the "cost function under policy ϕ," to be considered in greater depth in Chapter 3. Based on this, we arrive at a strange but ubiquitous definition: the total cost is a function of x, known as the (fixed policy) *value function*, and defined as the infinite sum:

$$J(x) = \sum_{k=0}^{\infty} c(x(k)), \qquad x(0) = x \in \mathsf{X}. \tag{2.24}$$

It is assumed that $c(x^e) = 0$, and we seek conditions ensuring that $x(k) \to x^e$ as $k \to \infty$, so there is some hope that J is finite valued. For the cruise control problem, the cost function is designed so that $c(x)$ is large if the state x corresponds to a speed that is far from desired.

Why Is the Controls Community So Excited about Total Cost? This metric for performance is not very intuitive, but there are compelling reasons for using it as a performance metric in control design:

(i) It is "forward looking." One might argue that (2.24) is looking too far foward (who cares about infinity?), but there is implicit "discounting" of the future since for a good policy we have $c(x(k)) \to 0$ quickly as $k \to \infty$.
(ii) Theory for total cost optimal control is often closely related to average cost optimal control – to be seen in Part II of the book.
(iii) If J is finite valued, then stability is typically guaranteed.

Benefit (iii) is the most abstract, but the most valuable aspect of this performance metric. Section 2.4.2 is dedicated to stability theory and its relationship to value functions. A part of this theory is based on the (fixed policy) dynamic programming equation:[4]

$$J(x) = c(x) + J(\mathrm{F}(x)). \tag{2.25}$$

This can be derived from the definition (2.24), written as

$$J(x) = c(x) + \sum_{k=0}^{\infty} c(x^+(k)),$$

where $\boldsymbol{x}^+$ is the solution to (2.21), starting at $x^+(0) = \mathrm{F}(x)$.

2.4.2 Stability of Equilibria

We survey here the most common definitions of stability for a nonlinear state space model. The first and most basic is a form of continuity near the equilibrium x^e. Let $\mathcal{X}(k;x_0)$ denote the state at time k with initial condition x_0: this is simply $x(k)$, obtained recursively from (2.21), starting at $x(0) = x_0$. In particular, the equilibrium property (2.23) implies that $\mathcal{X}(k;x^e) = x^e$ for all k.

[4] *Dynamic programming equation* and *Bellman equation* are used interchangeably, in reverence to [35].

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Stable in the Sense of Lyapunov. The equilibrium x^e is stable in the sense of Lyapunov if for all $\varepsilon > 0$, there exists $\delta > 0$ such that if $\|x_0 - x^e\| < \delta$, then

$$\|\mathcal{X}(k; x_0) - \mathcal{X}(k; x^e)\| < \varepsilon \qquad \text{for all } k \geq 0.$$

In words, if an initial condition is close to the equilibrium, then it will stay close forever. An illustration is provided in Figure 2.3, with $B(r) = \{x \in \mathbb{R}^n : \|x - x^e\| < r\}$ for any $r > 0$.

This is a very weak notion of stability, since there is no guarantee that the state will ever approach the desired equilibrium. The next definitions impose convergence:

Asymptotic Stability. An equilibrium x^e is said to be *asymptotically stable* if x^e is stable in the sense of Lyapunov and for some $\delta_0 > 0$, whenever $\|x_0 - x^e\| < \delta_0$,

$$\lim_{k \to \infty} \mathcal{X}(k; x_0) = x^e. \tag{2.26}$$

The set of x_0 for which the preceding limit holds is called the *region of attraction* for x^e.

The equilibrium is *globally asymptotically stable* if the region of attraction is all of X: that is, $\delta_0 = \infty$, and hence $x(k) \to x^e$ from any initial condition.

It is common to say that the state space model is globally asymptotically stable. That is, it is often stressed that this is a property of the system (2.21) rather than the equilibrium $x^e \in \mathsf{X}$.

Sometimes we obtain a very fast rate of convergence: the state space model is said to be *globally exponentially asymptotically stable* if there are constants $\varrho_0 > 0$ and $B_0 < \infty$ such that for each initial condition and $k \geq 0$,

$$\|\mathcal{X}(k; x_0) - x^e\| \leq B_0 \|x_0 - x^e\| e^{-\varrho_0 k}. \tag{2.27}$$

2.4.3 Lyapunov Functions

The construction of a *Lyapunov function* V is the most common approach to establishing asymptotic stability, as well as bounds on a value function (and more general bounds on the state process). In broad generality, V is a function on X taking nonnegative values, and the crucial property that makes it a Lyapunov function is that $V(x(k))$ is decreasing when $x(k)$ is large: this is formalized as a *drift inequality*. The Lyapunov function V is often regarded as a crude notion of "distance" to the "center of the state space."

For any scalar r, let $S_V(r)$ denote the *sublevel set*:

$$S_V(r) = \{x \in \mathsf{X} : V(x) \leq r\}. \tag{2.28}$$

In addition to the nonnegativity of V, we frequently assume it is *inf-compact*:

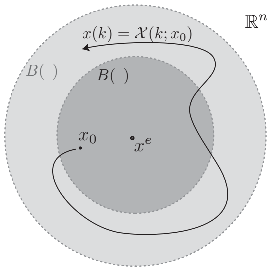


Figure 2.3 If $x_0 \in B(\delta)$, then $\mathcal{X}(k; x_0) \in B(\varepsilon)$ for all $k \geq 0$.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

$$\{x \in \mathsf{X} : V(x) \leq V(x^0)\} \text{ is a bounded set for each } x^0 \in \mathsf{X}.$$

That is, the set $S_V(r)$ takes on one of three forms for any r: the empty set, $S_V(r) = \mathsf{X}$, or $S_V(r) \subset \mathsf{X}$ is bounded.

In most cases, we find that $S_V(r) = \mathsf{X}$ is impossible, so that we arrive at the stronger *coercive* assumption:

$$\lim_{\|x\| \to \infty} V(x) = \infty. \qquad (2.29)$$

In this case, under our standing assumption (2.22), the set $S_V(r)$ is either empty or bounded for each r. Figure 2.4 illustrates the two classes of functions with $\mathsf{X} = \mathbb{R}$ (the function shown on the left is bounded). Here are three numerical examples:

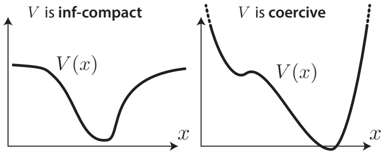


Figure 2.4 Inf-compact and coercive.

(i) $V(x) = x^2$ is coercive since (2.29) holds.
(ii) $V(x) = x^2/(1+x^2)$ is inf-compact but not coercive: $S_V(r) = \mathbb{R}$ for $r \geq 1$, and $S_V(r) = [-a,a]$ (a bounded interval) for $0 \leq r < 1$, with $a = \sqrt{r/(1-r)}$.
(iii) $V(x) = e^x$ is neither: $S_V(r) = (-\infty, \log(r)]$ is not a bounded subset of $\mathbb{R}$ for $r > 0$.

The value function J satisfies the intuitive properties of a Lyapunov function under mild conditions:

Lemma 2.1 *Suppose that the cost function c and the value function J defined in* (2.24) *are nonnegative and finite valued. Then,*

(i) *$J(x(k))$ is nonincreasing, and $\lim_{k\to\infty} J(x(k)) = 0$ for each initial condition.*
(ii) *Suppose in addition that J is continuous, inf-compact, and vanishes only at x^e. Then, for each initial condition, $\lim_{k\to\infty} x(k) = x^e$.*

The proof is postponed to Section 2.4.4, but we note here the first steps: the dynamic programming equation (2.25) implies that for each $k \geq 0$,

$$J(x(k+1)) = J(x(k)) - c(x(k)) \leq J(x(k)). \qquad (2.30)$$

That is, $J(x(k))$ is nonincreasing, so that $x(k) \in S_J(r)$ for each $k \geq 0$, with $r = J(x(0))$. The inf-compact assumption then implies that the state trajectory is "bottled-up" in the bounded set $S_J(r)$.

In the context of total-cost optimal control, the basic drift inequality considered in this book is *Poisson's inequality*: for nonnegative functions $V, c \colon \mathsf{X} \to \mathbb{R}_+$, and a constant $\overline{\eta} \geq 0$,

$$V(\mathrm{F}(x)) \leq V(x) - c(x) + \overline{\eta}. \qquad (2.31)$$

The reference to a French mathematician is explained in the notes. Poisson's inequality is a relaxation of the dynamic programming equation (2.25) through the introduction of $\overline{\eta}$, as well as the inequality.

Poisson's inequality is defined with attention to the dynamics: on combining (2.31) and (2.21), we obtain (similar to (2.30))

$$V(x(k+1)) \leq V(x(k)) - c(x(k)) + \overline{\eta}, \qquad k \geq 0.$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

If $\overline{\eta} = 0$, it follows that the sequence $\{V(x(k) : k \geq 0\}$ is nonincreasing. Under mild assumptions on V, we obtain a weak form of stability:

Proposition 2.2 *Suppose that* (2.31) *holds with* $\overline{\eta} = 0$. *Suppose moreover that* V *is continuous, inf-compact, and has a unique minimum at* x^e. *Then the equilibrium is stable in the sense of Lyapunov.*

Proof From the definition of the sublevel sets, we obtain

$$\bigcap\{S_V(r) : r > V(x^e)\} = S_V(r)\Big|_{r=V(x^e)} = \{x^e\}.$$

The final equality follows from the assumption that x^e is the unique minimizer of V. The inf-compact assumption then implies the following inner and outer approximations: for each $\varepsilon > 0$, we can find $r > V(x^e)$ and $\delta < \varepsilon$ such that[5]

$$\{x \in \mathsf{X} : \|x - x^e\| < \delta\} \subset S_V(r) \subset \{x \in \mathsf{X} : \|x - x^e\| < \varepsilon\}.$$

If $\|x_0 - x^e\| < \delta$, then $x_0 \in S_V(r)$, and hence $x(k) \in S_V(r)$ for all $k \geq 0$ since $V(x(k))$ is nonincreasing. The preceding final inclusion then implies that $\|x(k) - x^e\| < \varepsilon$ for all k. Stability in the sense of Lyapunov follows. □

Bounds on the value function J are obtained by iteration: for example, the two bounds

$$V(x(2)) \leq V(x(1)) - c(x(1)) + \overline{\eta}, \quad V(x(1)) \leq V(x(0)) - c(x(0)) + \overline{\eta}$$

imply that $V(x(2)) \leq V(x(0)) - c(x(0)) - c(x(1)) + 2\overline{\eta}$. We can of course go further:

Proposition 2.3 ((Comparison Theorem)) *Poisson's inequality* (2.31) *implies the following bounds:*

(i) *For each* $\mathcal{N} \geq 1$ *and* $x = x(0)$,

$$V(x(\mathcal{N})) + \sum_{k=0}^{\mathcal{N}-1} c(x(k)) \leq V(x) + \mathcal{N}\overline{\eta}. \tag{2.32}$$

(ii) *If* $\overline{\eta} = 0$, *then* $J(x) \leq V(x)$ *for all* x.

(iii) *Suppose that* $\overline{\eta} = 0$, *and that* V, c *are continuous. Suppose moreover that* c *is inf-compact, and vanishes only at* x^e. *Then the equilibrium is globally asymptotically stable.* □

The proof is found in Section 2.4.4.

Proposition 2.3 raises a question: what if Poisson's inequality is tight, so that the inequality in (2.31) is replaced by equality? Consider this ideal with $\overline{\eta} = 0$, and use the more suggestive notation $V = J^\circ$ for the Lyapunov function:

$$J^\circ(\mathrm{F}(x)) = J^\circ(x) - c(x). \tag{2.33}$$

If J° is nonnegative valued, then we can take $V = J^\circ$ in Proposition 2.3 to obtain the upper bound $J(x) \leq J^\circ(x)$ for all x. Equality requires further assumptions:

[5] This conclusion requires a bit of topology: the characterization of compact sets in terms of "open coverings." If this is new to you, don't worry: topology is not a prerequisite for this book. In the future, you might want to take a first-year mathematical analysis course.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Proposition 2.4 *Suppose that* (2.33) *holds, along with the following assumptions:*

(i) *J is continuous, inf-compact, and vanishes only at x^e.*
(ii) *J° is continuous.*

Then, $J(x) = J^\circ(x) - J^\circ(x^e)$ for each x.

2.4.4 Technical Proofs

To establish Propositions 2.3 and 2.4, we first require Lemma 2.1.

Proof of Lemma 2.1 We begin with the sample path representation of (2.25):

$$J(x(k+1)) - J(x(k)) + c(x(k)) = 0. \tag{2.34}$$

Summing each side from $k = 0$ to $\mathcal{N} - 1$ gives for each $x = x(0)$, and each $\mathcal{N}$,

$$J(x) = J(x(\mathcal{N})) + \sum_{k=0}^{\mathcal{N}-1} c(x(k)).$$

On taking limits, we obtain

$$J(x) = \lim_{\mathcal{N}\to\infty} \Big\{ J(x(\mathcal{N})) + \sum_{k=0}^{\mathcal{N}-1} c(x(k)) \Big\} = \Big\{ \lim_{\mathcal{N}\to\infty} J(x(\mathcal{N})) \Big\} + J(x),$$

which implies (i) under the assumption that $J(x)$ is finite.

The inf-compact assumption in (ii) is imposed to ensure that the state trajectory evolves in a bounded set: (2.30) implies that $x(k) \in S_J(r)$ for the particular value $r = J(x(0))$, and each $k \geq 0$. Suppose that $\{x(k_i) : i \geq 0\}$ is a convergent subsequence of the state trajectory, with limit x^∞. Then $J(x^\infty) = \lim_{i\to\infty} J(x(k_i)) = 0$ follows by continuity of J.

The assumption that J vanishes only at x^e implies that $x^\infty = x^e$. Part (ii) follows, since every convergent subsequence reaches the same value x^e. □

Proof of Proposition 2.3 The bound (2.32) is established following the discussion preceding the proposition. We begin with the sample path representation of (2.31), similar to (2.34):

$$V(x(k+1)) - V(x(k)) + c(x(k)) \leq \overline{\eta}. \tag{2.35}$$

Summing each side from $k = 0$ to $\mathcal{N} - 1$ gives (i):

$$V(x(\mathcal{N})) - V(x(0)) + \sum_{k=0}^{\mathcal{N}-1} c(x(k)) \leq \overline{\eta}\mathcal{N}.$$

Part (ii) follows since $V(x(\mathcal{N})) \geq 0$ for each $\mathcal{N}$, so that when $\overline{\eta} = 0$ we obtain from the preceding bound

$$\sum_{k=0}^{\mathcal{N}-1} c(x(k)) \leq V(x(0)).$$

The proof of (iii) is identical to Lemma 2.1: part (ii) implies that $\lim_{k\to\infty} c(x(k)) = 0$, and the assumptions on c then imply that $x(k) \to x^e$ as $k \to \infty$.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

It remains to show that x^e is stable in the sense of Lyapunov. To see this, first observe that with $\overline{\eta} = 0$, the bound (2.31) implies that $V \geq c$, so that V is also inf-compact. The bound (2.31), and conditions on c, $\overline{\eta}$, also imply that $V(x(k))$ is strictly decreasing when $x(k) \neq x^e$. Continuity of V implies that $V(x(k)) \downarrow V(x^e)$ for each $x(0)$, so that $V(x^e) < V(x(0))$ for all $x(0) \in \mathsf{X}$. Stability in the sense of Lypapunov then follows from Proposition 2.2. □

Proof of Proposition 2.4 The proof begins with iteration, as in Proposition 2.3:

$$J^\circ(x(\mathcal{N})) + \sum_{k=0}^{\mathcal{N}-1} c(x(k)) = J^\circ(x).$$

Lemma 2.1 (ii) and continuity of J° implies that $J^\circ(x(\mathcal{N})) \to J^\circ(x^e)$ as $\mathcal{N} \to \infty$, which implies the desired identity: $J^\circ(x^e) + J(x) = J^\circ(x)$. □

2.4.5 Geometry in Continuous Time

Let's briefly consider an analog of (2.21) in continuous time, with state evolving on $\mathsf{X} = \mathbb{R}^n$:

$$\frac{d}{dt} x_t = \mathrm{f}(x_t), \tag{2.36}$$

where $\mathrm{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ is called the *vector field*. It is common to suppress the time index, writing $\frac{d}{dt} x = \mathrm{f}(x)$.

We let $\mathcal{X}(t; x_0)$ denote the solution to (2.36) at time t, when we need to emphasize dependency on the initial condition x_0. The definition of asymptotic stability of an equilibrium x^e is the same as for the state space model in discrete time (2.21). The equilibrium is globally asymptotically stable if, in addition,

$$\lim_{t\to\infty} \mathcal{X}(t; x_0) = x^e, \qquad \text{for all } x_0 \in \mathsf{X}.$$

Verification of global asymptotic stability invites the following assumptions, generalizing the theory in discrete time. Recall that $V\colon \mathbb{R}^n \to \mathbb{R}$ is *continuously differentiable* (or C^1) if the gradient ∇V exists and is continuous.

Lyapunov Function for Global Asymptotic Stability

- V is nonnegative valued and C^1.
- It is inf-compact (recall the definition that follows (2.28)).
- For any solution $\boldsymbol{x}$, whenever $x_t \neq x^e$,

$$\frac{d}{dt} V(x_t) < 0. \tag{2.37}$$

Naturally, $\frac{d}{dt} V(x_t) = 0$ if $x_t = x^e$: in this case, $V(x_{t+s}) = V(x^e)$ for all $s \geq 0$.

Figure 2.5 illustrates the meaning of the vector field f for the special case $\mathsf{X} = \mathbb{R}^2$, and the figure is intended to emphasize the fact that $V(x_t)$ is nonincreasing when V is a Lyapunov function. The drift condition (2.37) can be expressed in functional form,

$$\langle \nabla V(x), \mathrm{f}(x) \rangle < 0, \qquad x \neq x^e. \tag{2.38}$$

This is illustrated geometrically in Figure 2.6.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

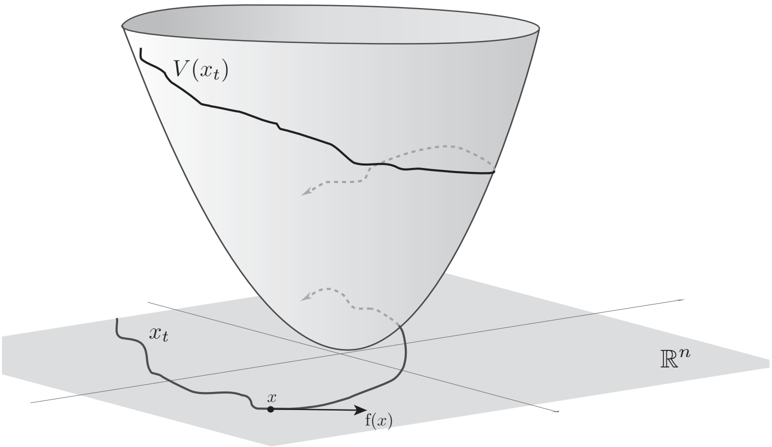


Figure 2.5 If V is a Lyapunov function, then $V(x_t)$ is nonincreasing with time.

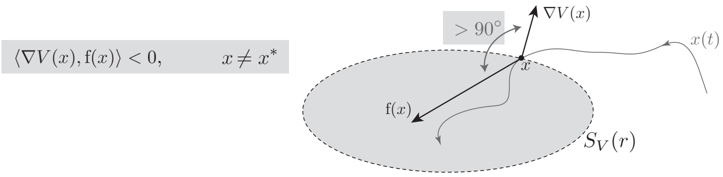

$\langle \nabla V(x), \mathrm{f}(x)\rangle < 0, \qquad x \neq x^*$
$> 90°$
$\nabla V(x)$
$x(t)$
x
f(x)
$S_V(r)$

Figure 2.6 Geometric interpretations of a Lyapunov drift condition: the gradient $\nabla V(x)$ is orthogonal to the level set $\{y : V(y) = V(x)\}$, which is the boundary of the set $S_V(r)$ with $r = V(x)$.

Proposition 2.5 *If there exists a Lyapunov function V satisfying the assumptions for global asymptotic stability, then the equilibrium x^e is globally asymptotically stable.* □

Proposition 2.5 is a partial extension of Proposition 2.3 to the continuous time model. A full extension requires a version of Poisson's inequality. Suppose that $c\colon \mathbb{R}^n \to \mathbb{R}_+$ is continuous, $V\colon \mathbb{R}^n \to \mathbb{R}_+$ is continuously differentiable, and $\overline{\eta} \geq 0$ is a constant, jointly satisfying

$$\langle \nabla V(x), \mathrm{f}(x)\rangle \leq -c(x) + \overline{\eta}, \qquad x \in \mathsf{X}. \tag{2.39}$$

An application of the chain rule implies that this is a continuous time version of (2.31):

$$\frac{d}{dt} V(x_t) \leq -c(x_t) + \overline{\eta}, \qquad t \geq 0.$$

And with a bit more work, we reach the following conclusions:

Proposition 2.6 *If* (2.39) *holds for nonnegative $c, V, \overline{\eta}$, then*

$$V(x_T) + \int_0^T c(x_t)\, dt \leq V(x) + T\,\overline{\eta}, \qquad x_0 = x \in \mathsf{X},\ T > 0.$$

If $\overline{\eta} = 0$, then the total cost is finite:

$$\int_0^\infty c(x_t)\, dt \leq V(x), \qquad x_0 = x \in \mathsf{X}. \tag{2.40}$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Proof For any $T > 0$, we obtain by the fundamental theorem of calculus,

$$-V(x_0) \le V(x_T) - V(x_0) = \int_0^T \Big(\tfrac{d}{dt} V(x_t)\Big)\, dt \le T\overline{\eta} - \int_0^T c(x_t), \qquad T \ge 0.$$

If $\overline{\eta} = 0$, then the bound (2.40) follows on letting $T \to \infty$. □

Converse Theorems We have seen this implication:

Existence of Lyapunov function $\Longrightarrow$ *Stability and/or performance bound*

where the nature of stability depends on the nature of the Lyapunov function bound. What about a converse? That is, if the system is stable, can we infer that a Lyapunov function exists?

Assume moreover that the total cost is finite:

$$J(x) = \int_0^\infty c(x_t)\, dt, \qquad x_0 = x$$

with arbitrary initial condition. If J is differentiable, then we obtain a solution to (2.37) using $V = J$:

Proposition 2.7 *If J is finite valued, then for each initial condition x_0 and each t,*

$$\frac{d}{dt} J(x_t) = -c(x_t). \tag{2.41}$$

If J is continuously differentiable, the Lyapunov bound (2.37) *follows with equality:*

$$\nabla J(x) \cdot \mathrm{f}(x) = -c(x).$$

Proof We have a simple version of Bellman's principle (a focus of Chapter 3): for any $T > 0$,

$$J(x_0) = \int_0^T c(x_r)\, dr + J(x_T).$$

For $t \ge 0$, $\delta > 0$ given, apply this equation with $T = t + \delta$ and $T = t$:

$$J(x_0) = \int_0^{t+\delta} c(x_r)\, dr + J(x_{t+\delta}),$$
$$J(x_0) = \int_0^{t} c(x_r)\, dr + J(x_t).$$

On subtracting, and then dividing by δ, this gives

$$0 = \frac{1}{\delta} \int_t^{t+\delta} c(x_r)\, dr + \frac{1}{\delta}\big(J(x_{t+\delta}) - J(x_t)\big).$$

Letting $\delta \downarrow 0$, the first term converges to $c(x_t)$ because $c\colon \mathbb{R}^n \to \mathbb{R}$ is continuous, and the second term converges to the derivative of $J(x_t)$ with respect to time, which establishes (2.41). The final conclusion follows from the chain rule. □

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

2.4.6 Linear State Space Models

If the dynamics in (2.21) are linear, with $x(k) \in \mathsf{X} = \mathbb{R}^n$, then

$$x(k+1) = Fx(k), \qquad k \geq 0 \tag{2.42}$$

for an $n \times n$ matrix F, and by iteration

$$x(k) = F^k x, \qquad k \geq 0, \, x(0) = x.$$

This equation is valid with $k = 0$ since we take $F^0 = I$, the $n \times n$ identity matrix.

Suppose that the cost is also quadratic, $c(x) = x^\intercal Sx$, for a symmetric and positive definite matrix S. It follows that $c(x(k))$ is a quadratic function of $x(0)$ for each k:

$$c(x(k)) = (F^k x)^\intercal S F^k x.$$

Hence the value function J defined in (2.24) is also quadratic:

$$J(x) = x^\intercal \Big[\sum_{k=0}^{\infty} (F^k)^\intercal S F^k \Big] x, \qquad x(0) = x \in \mathsf{X}.$$

That is, $J(x) = x^\intercal M x$, where M is the matrix within the brackets. It satisfies a linear fixed point equation, known as the (discrete-time) *Lyapunov equation*:

$$M = S + F^\intercal M F. \tag{2.43}$$

A proof of the following can be obtained based on these calculations:

Proposition 2.8 *The following are equivalent for the linear state space model* (2.42)*:*

(i) *The origin is locally asymptotically stable.*
(ii) *The origin is globally asymptotically stable.*
(iii) *The Lyapunov equation* (2.43) *admits a solution* $M \geq 0$ *for any* $S \geq 0$.
(iv) *Each eigenvalue* λ *of* F *satisfies* $|\lambda| < 1$. □

Controllable Canonical Form. Recall that this state space realization was based on the ARMA model (2.4), with $N = n$. If you have taken a course in signals and systems, you then know that stability of the ARMA model (in an input–output sense called bounded input, bounded output *[BIBO] stability*) is verified by examining the roots $\{p_i : 1 \leq i \leq n\}$ of the rational function

$$a(z) = 1 + \sum_{i=1}^{n} a_i z^{-i} = \prod_{i=1}^{n} (1 - p_i z^{-i}), \qquad z \in \mathbb{C}.$$

The system is BIBO stable if $|p_i| < 1$ for each i. The eigenvalues $\{\lambda_i : 1 \leq i \leq n\}$ of F are obtained as the solution to a root finding problem $\Delta_F(\lambda) = 0$, where

$$\Delta_F(\lambda) = \det(\lambda I - F) = \prod_{i=1}^{n} (\lambda - \lambda_i), \qquad \lambda \in \mathbb{C}.$$

For the state space model in controllable canonical form, it can be shown that $\Delta_F(z) = a(z)z^n$ for any $z \in \mathbb{C}$, and hence $\{p_i : 1 \leq i \leq n\} = \{\lambda_i : 1 \leq i \leq n\}$.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Example 2.4.1 (*Linear Model in Continuous Time*) Consider the linear ODE

$$\frac{d}{dt}x = Ax \tag{2.44}$$

whose solution is the matrix exponential:

$$x_t = e^{At}x(0)\,, \qquad e^{At} = \sum_{m=0}^{\infty} \frac{1}{m!}t^m A^m. \tag{2.45}$$

Consequently, $x_t \to 0$ as $t \to \infty$ from each initial condition if and only if A is *Hurwitz*: each eigenvalue of A has strictly negative real part.

The solution to (2.41) is obtained with a quadratic $J(x) = x^{\intercal}Zx$, where the matrix Z can be found through a bit of linear algebra and calculus. The value function is nonnegative, so we may assume Z is positive semidefinite (hence in particular, symmetric: $Z = Z^{\intercal}$). Symmetry implies,

$$\tfrac{d}{dt}J(x_t) = 2x_t^{\intercal}ZAx_t = x_t^{\intercal}[ZA + A^{\intercal}Z]x_t$$

and from (2.41) this gives

$$x_t^{\intercal}[ZA + A^{\intercal}Z]x_t = -c(x_t) = -x_t^{\intercal}Sx_t.$$

This must hold for each t and each $x(0)$, giving the Lyapunov equation in continuous time:

$$0 = ZA + A^{\intercal}Z + S. \tag{2.46}$$

Euler Approximation If we sample, with constant sampling interval $\Delta > 0$, then from the continuous time model (2.44) we obtain the linear model (2.42): with $t_k = k\Delta$,

$$x(t_{k+1}) = e^{\Delta A}x(t_k)\,, \qquad k \geq 0. \tag{2.47}$$

The Euler approximation of (2.44) also results in the linear model (2.42), but with $F = I + \Delta A$. The matrix F is precisely the first-order Taylor series approximation of the matrix exponential. While only an approximation, it is often good enough for control design.

A particular two-dimensional example is $A = \left(\begin{smallmatrix}-0.2, & 1\\ -1, & -0.2\end{smallmatrix}\right)$. The matrix is Hurwitz, with two eigenvalues $\lambda(A) = -0.2 \pm j$. With sampling interval $\Delta = 0.02$, we find that $F = I + \Delta A$ also has two complex eigenvalues:

$$\lambda(F) = 1 + \Delta\lambda(A) \approx 0.996 \pm 0.02j\,.$$

The eigenvalues satisfy $|\lambda(F)| < 1$, so we see that stability of the discrete-time approximation is inherited from the continuous-time model.

The Matlab command `M = dlyap(F',eye(2))` returns a solution to the Lyapunov equation (2.43) with $S = I$ (the identity matrix):

$$M = \begin{bmatrix} 131.9 & 0.0 \\ 0.0 & 131.9 \end{bmatrix}.$$

The fact that F has complex eigenvalues implies that the state process will exhibit rotational motion. The sample path of $\boldsymbol{x}$ shown on the left-hand side of Figure 2.7 spirals

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

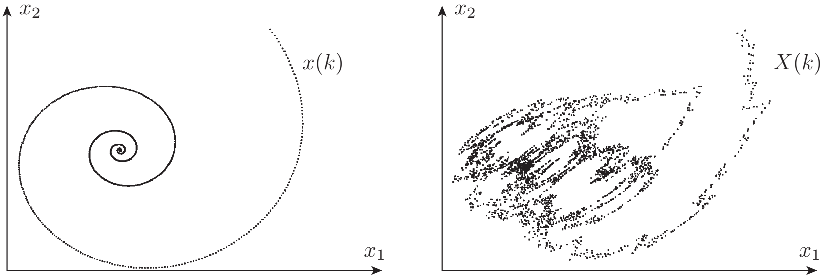


Figure 2.7 At the left is a sample path of the deterministic linear model (2.42). At the right is a sample path from the linear model with disturbance (2.48).

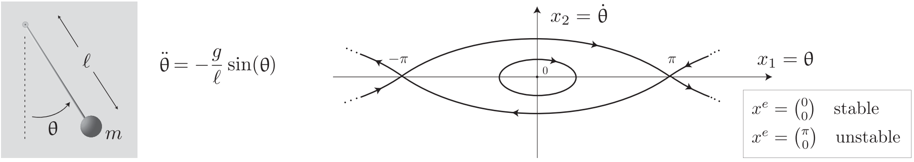

ℓ
θ
m
$\ddot{\theta} = -\frac{g}{\ell}\sin(\theta)$
$x_2 = \dot{\theta}$
$-\pi$
0
π
$x_1 = \theta$
$x^e = \binom{0}{0}$ stable
$x^e = \binom{\pi}{0}$ unstable

Figure 2.8 Frictionless pendulum: stable and unstable equilibria for the state space model.

toward the origin, and is intuitively "stable." The plot on the right is a simulation of the linear model subject to a "white noise" disturbance:

$$X(k+1) = FX(k) + N(k+1), \qquad k \geq 0. \tag{2.48}$$

See the discussion that follows (7.46) for details of the disturbance process $\boldsymbol{N}$. ■

Example 2.4.2 (*Frictionless Pendulum*) The *frictionless pendulum* illustrated on the left-hand side of Figure 2.8 is a favorite example in physics and undergraduate control courses. It is based on several simplifying assumptions:

- There is no friction or air resistance.
- The rod on which the bob swings is rigid and without mass.
- The bob has mass, but zero volume.
- Motion occurs only in two dimensions.
- The gravitational field is uniform.
- "F = MA" (apply classical mechanics, subject to the foregoing).

A nonlinear state space model is obtained in which x_1 is the angular position θ, and x_2 its derivative:

$$\frac{d}{dt}x = \mathrm{f}(x) = \begin{bmatrix} x_2 \\ -\frac{g}{\ell}\sin(x_1) \end{bmatrix}. \tag{2.49}$$

Shown on the right-hand side of Figure 2.8 are sample trajectories of x_t, and two equilibria.

An inspection of state trajectories shown on the right-hand side of Figure 2.8 reveals that the equilibrium $x^e = \binom{\pi}{0}$ is not stable in any sense, which agrees with physical intuition (the pendulum is sitting upright in this case). Trajectories that begin near the equilibrium $x^e = 0$ will remain near this equilibrium thereafter.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

The origin is stable in the sense of Lyapunov. To see this, consider a Lyapunov function defined as the sum of potential and kinetic energy:

$$V(x) = \text{PE} + \text{KE} = mg\ell[1 - \cos(x_1)] + \tfrac{1}{2}m\ell^2 x_2^2.$$

The first term is potential energy relative to the height at the equilibrium $x^e = \mathsf{0}$, and the second is the classical "KE $= \frac{1}{2}mv^2$" formula for kinetic energy. It is not surprising that V is minimized at $x^e = \mathsf{0}$.

We have $\nabla V\,(x) = m\ell^2[(g/\ell)\sin(x_1)\,, x_2]^{\intercal}$, and

$$\nabla V\,(x) \cdot \mathrm{f}(x) = m\ell^2\big\{(g/\ell)\sin(x_1) \cdot x_2 - x_2 \cdot (g/\ell)\sin(x_1)\big\} = 0.$$

This means that $\frac{d}{dt}V(x_t) = 0$, and hence $V(x_t)$ does not depend on time. For example, the periodic orbit shown in Figure 2.8 evolves in a level set of V:

$$\frac{g}{\ell}[1 - \cos(x_1(t))] + \tfrac{1}{2}x_2(t)^2 = \text{const.}$$

From this it follows that the origin is stable in the sense of Lyapunov.

Linearization: Using the first-order Taylor series approximation $\sin(\theta) \approx \theta$, the state space equation for the pendulum can be approximated by the LTI model (2.44): $\frac{d}{dt}x = Ax$, with

$$A = \begin{bmatrix} 0 & 1 \\ -g/\ell & 0 \end{bmatrix}. \tag{2.50}$$

The eigenvalues of A are obtained on solving the quadratic equation $0 = \det(I\lambda - A)$:

$$0 = \det\left(\begin{bmatrix} \lambda & -1 \\ g/\ell & \lambda \end{bmatrix}\right) = \lambda^2 + g/\ell \implies \lambda = \pm\sqrt{g/\ell}\,j.$$

The complex eigenvalues are consistent with the periodic behavior of the pendulum. ■

2.5 A Glance Ahead: From Control Theory to RL

Here is a definition from Wikipedia, as seen on July 2020: "Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward." Here is a translation of some of the key terms:

- ▲ **Machine learning** (ML) refers to prediction/inference based on sampled data.
- ▲ **Take actions** $\equiv$ feedback. That is, the choice of $u(k)$ for each k based on observations.[6]
- ▲ **Software agent** $\equiv$ policy ϕ. This is where the machine learning comes in: the creation of ϕ is based on a large amount of training data collected in "the environment."
- ▲ **Cumulative reward** $\equiv$ negative of the sum of cost, such as (2.24), but with the inclusion of the input:

$$\text{Cumulative reward} = -\sum_k c(x(k), u(k)).$$

[6] The term *features* is a common substitute for the observation process $\boldsymbol{y}$ shown in Figure 2.1.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

An emphasis in the academic community is truly model-free RL, and most of the theory builds on the optimal control concepts reviewed in the next chapter. Some of the main ideas can be exposed right here.

What follows is background on how RL algorithms are currently formulated. Think hard about alternatives – remember, the field remains young!

2.5.1 Actors and Critics

The *actor-critic* algorithm of reinforcement learning is specifically designed within the context of stochastic control, so this is a topic for Part II. The origins of the terms are worth explaining here. We are given a parameterized family of policies $\{\phi^\theta : \theta \in \mathbb{R}^d\}$, which play the role of actors. For each θ, we (or our "software agents") can observe features of the state process $\boldsymbol{x}$ under chosen the policy. The ideal critic then computes exactly the associated value function J_θ, but in realistic situations we have only an estimate.

Since in this book we are minimizing cost rather than maximizing reward, the output of an actor-critic algorithm is the minimum

$$\theta^\star = \arg\min_\theta \langle \nu, J_\theta \rangle, \tag{2.51}$$

where $\nu \geq 0$ serves as a state weighting. This will be defined as a sum

$$\langle \nu, J_\theta \rangle = \sum_i J_\theta(x^i)\nu(x^i),$$

where $\nu(x^i)$ is relatively large for "important states."

Methods to solve the optimization problem (2.51) are explored in Section 4.6, using an approach known as *gradient free optimization*. These algorithms are intended to approximate the true gradient descent algorithms of optimization surveyed in Section 4.4, and are often called "actor-only methods." The meaning of actor-critic methods is explained in Chapter 10.

This is an example of ML: optimizing a complex objective function over a large function class for the purposes of prediction or classification (in this case, we are predicting the best policy). A very short introduction to ML can be found in Section 5.1.

2.5.2 Temporal Differences

Where do we find a critic? That is, how can we estimate a value function without a model? One answer lies in the sample path representation of the fixed policy dynamic programming equation, previously announced in (2.30). For any θ, we have

$$J_\theta(x(k)) = c(x(k),u(k)) + J_\theta(x(k+1)), \qquad k \geq 0, \quad u(k) = \phi^\theta(x(k)).$$

We might seek an approximation $\widehat{J}$ for which this identity is well approximated. This motivates the *temporal difference* (TD) sequence commonly used in RL algorithms:

$$\mathcal{D}_{k+1}(\widehat{J}) \overset{\text{def}}{=} -\widehat{J}(x(k)) + \widehat{J}(x(k+1)) + c(x(k),u(k)), \qquad k \geq 0, \quad u(k) = \phi^\theta(x(k)). \tag{2.52}$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

After collecting N observations, we obtain the mean-square loss:

$$\Gamma^{\varepsilon}(\widehat{J}) = \frac{1}{N}\sum_{k=0}^{N-1}\big[\mathcal{D}_{k+1}(\widehat{J})\big]^2. \tag{2.53}$$

We are then faced with another machine learning problem: minimize this objective function over all $\widehat{J}$ in a given class (for example, this is where neural networks frequently play a role).

If we can make (2.53) nearly zero, then we have a good estimate of a value function. Beyond its application to actor-critic methods, there are TD- and Q-learning techniques, designed to minimize (2.53) or a surrogate, that are part of a bigger RL toolbox.

2.5.3 Bandits and Exploration

Suppose that our policy is pretty good. Maybe not optimal in any sense, but $x(k) \to x^e$, $u(k) \to u^e$ rapidly as $k \to \infty$, where the limit satisfies $c(x^e, u^e) = 0$. We typically then have continuity:

$$\lim_{k\to\infty}\big[-\widehat{J}(x(k+1)) + \widehat{J}(x(k)) - c(x(k),u(k))\big] = -\widehat{J}(x^e) + \widehat{J}(x^e) - c(x^e,u^e) = 0. \tag{2.54}$$

It follows that we aren't observing very much via the temporal difference (2.52). If N is very large, then $\Gamma^{\varepsilon}(\widehat{J}) \approx 0$. This essentially destroys any hope for a reliable estimate of the value function. Expressed another way: a good policy does not lead to sufficient exploration of the state space.

There are many ways to introduce exploration. We can, for example, adapt our criterion as follows: denote by $\Gamma^{\varepsilon}(\widehat{J}; x)$ the mean-square loss obtained with $x(0) = x$. Rather than take a very long run, perform many shorter runs, from many ($M > 1$) initial conditions. The loss function to be minimized is the average

$$\Gamma(\widehat{J}) = \frac{1}{M}\sum_{i=1}^{M}\Gamma^{\varepsilon}(\widehat{J}; x^i). \tag{2.55}$$

The best way to choose the samples $\{x^i\}$ is a topic of research.

Another approach is to let the input do the exploring. The policy is modified slightly through the introduction of "noise":

$$u(k) = \breve{\phi}(x(k), \xi(k)).$$

For example, $\{\xi(k)\}$ might be a scalar signal, defined as a mixture of sinusoids. The noisy policy is defined so that

(i) $\breve{\phi}(x(k), \xi(k)) \approx \phi^{\theta}(x(k))$ for "most k."
(ii) The state process "explores." In particular, the policy is designed to avoid convergence of $(x(k), u(k))$ to any limiting value.

This is a crude approach, since by changing the input process, the associated value function also changes. More sensible approaches are contained in Chapters 4 and 5, and in the second part of the book: Q-learning and "off-policy SARSA" might be designed around

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

an exploratory policy such as this one, but these algorithms are carefully designed to avoid bias from exploration.

The theory of exploration is mature only within a very special setting: multi-armed bandits. The term “bandit” refers to slot machines: you put money in the machine, pull an arm, and hope that more money pops out. A more rational application is in the advertisement industry, in which an “arm” is an advertisement (which costs money), and the advertiser hopes money will pop out as the ads encourage sales. There is a great history of heuristics and science to create successful algorithms to maximize profit, based only on noisy observations of the performance of candidate ads ([216] is a great reference on the theory of bandits, and a short survey is contained in Section 7.8). It is here that the “exploration/exploitation” trade-off is most clearly seen: you have to accept some loss of revenue through exploration in order to learn the best strategy, and then “exploit” as you gain confidence in your estimates.

The situation is much more complex in control applications: imagine that for each state $x(k)$, there is a multi-armed bandit. “Pulling arm a” at time k means choosing $u(k) = a \in \mathsf{U}$. Concepts from bandit theory have led to heuristics to best balance the exploration/exploitation trade-offs arising in RL. This is an exciting direction for future research [171, 307].

2.6 How Can We Ignore Noise?

It is hard to explain this precisely to a student without a background in probability theory. If you have some exposure to stochastic processes, then you might want to skim Section 7.2: you will learn how to construct a deterministic “fluid model” or “mean-field model” based on a more detailed and complex stochastic state space model, and find justification for control design based on the simpler model.

The pragmatic answer to this question is that we rarely have a reliable model of disturbances, so we leave them unmodeled but not ignored. That is, we attempt to create a control architecture that is not very sensitive to disturbances. There is an elegant theory of robust control for this purpose, though even here “robustness” is only with respect to disturbances within some uncertainty class. The most successful outcomes of this literature lean heavily on frequency domain concepts. For example, it is assumed that disturbances (the d shown in Figure 2.1) are largely limited to lower frequencies, and measurement noise (the w shown on the right-hand side of this figure) is limited to higher frequencies.

Justification for nonlinear control systems is based on Lyapunov function techniques. We establish stability of our control solution through a Lyapunov function V as outlined in Section 2.4.3, and then argue that V will continue to have “negative drift” in the form (2.31) even with error in the model F, or in the presence of the disturbance d.

Finally, the naive “disturbance-free” model obtained through physics, or through techniques surveyed in Section 7.2, often provides a great deal of insight for the structure of control solutions. We might use this insight to build architectures for reinforcement learning.

2.7 Examples

2.7.1 Wall Street

Let’s begin with an example that clearly does not belong in this chapter. Search for “flash crash” on your internet browser to see images of the enormous volatility of stock prices on

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

many time scales. While we have few tools for control design at this stage of the book, there are many interesting modeling questions that will help illustrate control and RL philosophy.

Where Is the Control Problem?

Let's consider the specific problem of stock portfolio management. The goal is to create a computer program that makes decisions second-by-second on which stocks to buy or sell. The goal is to "maximize profit," but there is also the notion of *risk*, which is not easily defined without tools from probability and statistics.

Perhaps more significant is that this control problem is not of the centralized variety. Consider how Figure 2.1 is interpreted for stock trading. The *process* is the global economy and everything that goes along with it! The two blocks *state feedback* and *observer* are the results of thousands of individual decision makers (the "agents") who forecast future prices (and other events), and employ optimization strategies for online decision making. *Trajectory generation* will also be local to each agent: this might represent decisions regarding purchase orders for new computers, new staff, or a new office closer to Wall Street.

In summary: stock trading is a game rather than a classical control problem, but this should not stop us. As an individual (or company designing software for others), we can treat the "process" along with the actions of all other players as a larger process. Reinforcement learning is an appealing approach to control design because the learning (or training) does not require a detailed model (though significant data are required for training).

This is a great example of the value of both measurements and actuation in control. The better your measurements, the more money you can expect to earn in an optimal control solution – there is no better example to illustrate this point. The book *Flash Boys* contains a popular treatment of the role of actuation – in particular, the cost of delay in the feedback loop [223] (see also [30]). It is claimed that millions of dollars can be made by reducing response delay by a millisecond!

State Feedback?

How do we interpret $u(k) = \phi(x(k))$? The input $u(k)$ is easy to understand, given the preceding description of the stock portfolio management problem.

What is the state $x(k)$? I don't know, and I would not trust anyone who claims to have an answer! It is traditional to view prices as a stochastic process that evolves according to the actions of millions of citizens and hundreds of corporations. There is modeling theory based on martingales and changes of measure, so theory from mathematical finance may provide intuition on how to construct a state process. A quick "gut reaction" might be this: $x(k) = x^0(k)$, the vector of all stock prices at time k. Without any knowledge of finance, my gut tells me that this would be a huge mistake. Here are examples of what many would add after further reflection:

(i) Past history of prices. It is important to visit recent performance in terms of both trends and volatility.

(ii) Forecasts of prices. You may have insider knowledge. You may realize that tweets from certain influential people provide insight on the decisions of others, which will then influence stock prices.

(iii) What is the objective? Once you have a formulation of reward and risk, make sure that these essential quantities are functions of your state process.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

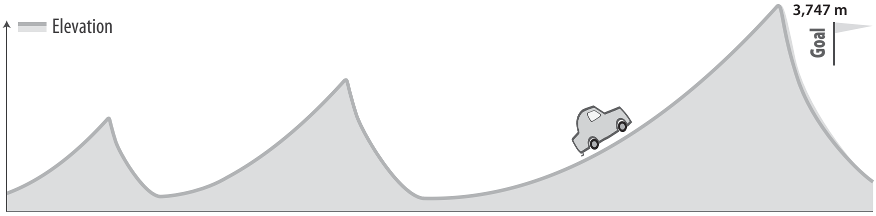


Figure 2.9 Mountain car.

You then have a very high-dimensional vector $x(k)$, and are left to find the feedback law ϕ.

There is no perfect state description. Even if a state space model were available, the full state would not be directly observed (and we would still want to use "side information," such as the tweets of CEOs and politicians). Appendix C contains a summary of *belief states* for partially observed control problems. This is an elegant way to create a fully observed state for the purposes of control, but comes with enormous cost in terms of complexity.

What follows are toy examples that will be useful for applying the methods to be developed over the course of this book. The models are presented in continuous time because of the elegance of calculus and classical mechanics.

2.7.2 Mountain Car

The goal is to drive a car with a very weak engine to the top of a very high mountain, as illustrated in Figure 2.9.

A two-dimensional state space model is obtained using position and velocity $x_t = (z_t, v_t)^\intercal$, and the input u is the throttle position (which is negative when the car is in reverse). In the following, the state space is defined to be a rectangular region,

$$\mathsf{X} = [z^{\text{min}}, z^{\text{goal}}] \times [-\overline{v}, \overline{v}]$$

in which z^{min} is a lower limit for the position z_t, and the target position is z^{goal}. The constraint $x_t \in \mathsf{X}$ means that the velocity v_t is bounded in magnitude by $\overline{v} > 0$.

Within the RL literature, this example was introduced in the dissertation [264], and has since become a favorite basic example [338].

What makes this problem interesting is that the engine is so weak that it is impossible to reach the hill directly from some initial conditions. A successful policy will sometimes put the car in reverse, and travel at maximal speed away from the goal to reach a higher elevation to the left. Several cycles back and forth may be required to reach the goal.

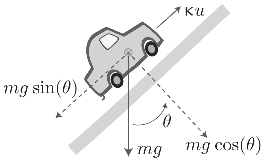


Figure 2.10 Two forces on the mountain car.

A continuous-time model can be constructed based on the two forces on the car, illustrated in Figure 2.10. To obtain a simple model, we need to be careful with our notion of distance: $z^{\text{goal}} - z_t$ denotes the *path distance* along the road to the goal, which is not the same as the distance along the x-*axis* in Figure 2.9. Subject to this convention, Newton's law gives

$$ma = m\frac{d^2}{dt^2}z = -mg\sin(\theta) + \kappa u.$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

With state $x = (z, v)^{\intercal}$, we arrive at the two-dimensional state space model,

$$\begin{aligned} \frac{d}{dt} x_1 &= x_2, \\ \frac{d}{dt} x_2 &= \frac{\kappa}{m} u - g\sin(\theta(x_1)), \end{aligned} \tag{2.56}$$

where $\theta(x_1)$ is the road grade at $z = x_1$.

An examination of the potential energy $\mathcal{U}$ tells us from which states we can reach the goal without control (setting $u = 0$ in (2.56)). The potential energy is proportional to elevation and can be computed by integrating the negative of force, $-F(z)$. For the control-free model, we have $-F(z) = mg\sin(\theta(z))$, and hence

$$\mathcal{U}(z) = \mathcal{U}(0) + mg\int_0^z \sin(\theta(z))\,dz. \tag{2.57}$$

The version of this model adopted in [338, ch. 10] uses these numerical values:

$$\kappa/m = 1\,, \quad g = 2.5\,, \quad \theta(z) = \pi + 3z.$$

In this case, (2.57) gives $\mathcal{U}(z) = \mathcal{U}(0) + mg\sin(3z)/3$. Figure 2.11 shows the potential energy as a function of z on the interval $[z^{\text{min}}, z^{\text{goal}}]$. It has a unique maximum at z^{goal}, which implies that it is necessary to apply external force to reach the goal for any initial condition satisfying $z(0) < z^{\text{goal}}$ and $v(0) \le 0$.

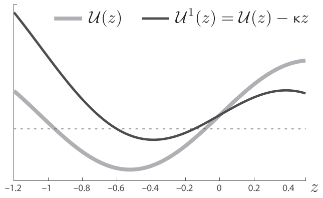


Figure 2.11 Potential energy for the mountain car.

Is the goal reachable? We again examine potential energy. Consider the force as a function of z with $u(k) = 1$ for all k. We obtain $-F(z) = mg\sin(\theta(z)) - \kappa$, and the resulting potential energy is the integral, denoted $\mathcal{U}^1(z) = \mathcal{U}(z) - \kappa z$ and shown in Figure 2.11. We now have $\mathcal{U}(z^{\text{min}}) > \mathcal{U}(z^{\text{goal}})$, so from $z(0) = z^{\text{min}}$ we will reach the goal with this open-loop control law.

Consider the initial position $z^0 = -0.6$, for which $\mathcal{U}^1(z^0)$ is indicated with a dashed line, and let z^1 denote the other value satisfying $z^1 > z^0$ and $\mathcal{U}^1(z^1) = \mathcal{U}^1(z^0)$. If $u(k) = 1$ for all k, then with initial condition $z(0) = -0.6$ and $v(0) = 0$, the car will initially move to the right, and stall at time t_1, for which $z(t_1) = z^1$. It will then reverse direction until it stalls at location z^0, and this process will repeat.

A discrete time model is adopted in [338, ch. 10], based on sampling the ODE with sampling interval $\Delta = 10^{-3}$: using the notation $x(k) = (z(k), v(k))^{\intercal}$,

$$z(k+1) = [\![z(k) + \Delta v(k+1)]\!]_1, \tag{2.58a}$$

$$v(k+1) = [\![v(k) + \Delta[u(k) - 2.5\cos(3z(k))]]\!]_2. \tag{2.58b}$$

This can be expressed in the form (2.6a) by substituting the expression for $v(k+1)$ in (2.58b) into the right-hand side of (2.58a).

The model is consistent with (2.56) using $\theta(z) = \pi + 3z$. The brackets denote projection of the values of $z(k+1)$ to the interval $[z^{\text{min}}, z^{\text{goal}}]$, and $v(k+1)$ to the interval $[-\overline{v}, \overline{v}]$. In addition, the constraint $v(k) \ge 0$ is imposed when $z(k) = z^{\text{min}}$, and $v(k) = 0$ when

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

$z(k) = z^{\text{goal}}$ (the car is parked once it reaches its target). The following values are chosen in numerical experiments:

$$z^{\min} = -1.2, \; z^{\text{goal}} = 0.5, \text{ and } \overline{v} = 70. \tag{2.58c}$$

Here is an aggressive policy that will get you to the top: Whatever direction you are going, accelerate in that direction at maximum rate (provided this is feasible):

$$u(k) = \begin{cases} 0 & z(k) = z^{\text{goal}} \\ \text{sign}(v(k)) & \text{else} \end{cases}. \tag{2.59}$$

If $v(k) = 0$, then $\text{sign}(v(k))$ can be taken to be 1 or -1, subject to the constraint that $v(k+1) \neq 0$.

2.7.3 MagBall

The magnetically suspended metal ball illustrated in Figure 2.12 will be used to illustrate several important modeling concepts. In particular, it shows how to transform a set of nonlinear differential equations into a state space model, and how to approximate this by a linear state space model of the form (2.20). Further details from a control systems perspective may be found in the lecture notes [29].

The input u is the current applied to an electromagnet, and the output y is the distance between the center of the ball and the bottom edge of the magnet. Since positive and negative inputs are indistinguishable at the output of this system, it follows that this cannot be a linear system. The upward force due to the current input is approximately proportional to u^2/y^2, and hence from Newton's law for translational motion we adopt the model

$$ma = m\frac{d^2}{dt^2}y = mg - \kappa\frac{u^2}{y^2},$$

where g is the gravitational constant, and κ is some constant depending on the physical properties of the magnet and ball.

Control design goal: Maintain the distance to the magnet at some reference value r.

We obtain a state space model as a first step to control design. This input–output model can be converted to state space form to obtain something similar to the controllable canonical form description of the ARMA model in (2.16) and (2.18): using $x_1 = y$ and $x_2 = \frac{d}{dt}y$,

$$\frac{d}{dt}x_1 = x_2, \quad \frac{d}{dt}x_2 = g - \frac{\kappa}{m}\frac{u^2}{x_1^2},$$

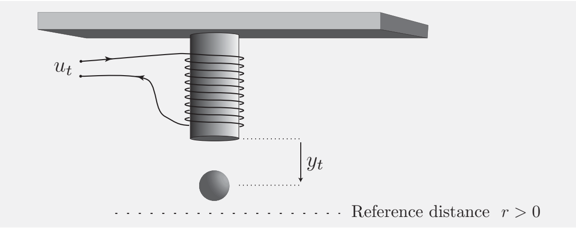


Figure 2.12 Magnetically suspended ball.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

where the latter equation follows from the formula $\frac{d}{dt}x_2 = \frac{d^2}{dt^2}y$. This pair of equations defines a two-dimensional state space model of the form (2.19):

$$\frac{d}{dt}x_1 = x_2 = \mathrm{f}_1(x_1,x_2,u), \tag{2.60a}$$

$$\frac{d}{dt}x_2 = g - \frac{\kappa}{m}\frac{u^2}{(x_1)^2} = \mathrm{f}_2(x_1,x_2,u). \tag{2.60b}$$

It is nonlinear, since f_2 is a nonlinear function of x, and also the state space is constrained: $\mathsf{X} = \{x \in \mathbb{R}^2 : x_1 \geq 0\}$.

Suppose that a fixed current $u^\circ > 0$ is applied, and that the state x° is an equilibrium: $\mathrm{f}(x^\circ,u^\circ) = \mathsf{0}$. From the definition of f_1 in (2.60a), we must have $x_2^\circ = 0$, and setting $\mathrm{f}_2(x^\circ,u^\circ)$ equal to zero in (2.60b) gives

$$x_1^\circ = \sqrt{\frac{\kappa}{mg}}u^\circ > 0. \tag{2.61}$$

If we are *very* successful with our control design, and $x_t = (r,0)^\intercal$ for all t, then we must have

$$u_t = u^\circ\,, \quad t \geq 0\,, \qquad \text{where } u^\circ = r\sqrt{mg/\kappa} : \text{the solution to (2.61) with } x_1^\circ = r.$$

Of course, we don't expect that this "open-loop" approach will be successful. If we are realistically successful, so that $x_t \approx r$ for all t (perhaps after a transient), then we should expect that $u_t \approx u^\circ$ as well. The design of a feedback law to achieve this goal is often obtained through an approximate linear model, called a *linearization*.

Linearization about an Equilibrium State

The linearization is defined exactly as in the frictionless pendulum (2.49). Assume that the signals x_1, x_2, and u remain close to the fixed point $(x_1^\circ,x_2^\circ,u^\circ)$, and write

$$\begin{aligned} x_1 &= x_1^\circ + \tilde{x}_1, \\ x_2 &= x_2^\circ + \tilde{x}_2, \\ u &= u^\circ + \tilde{u}, \end{aligned}$$

where $\tilde{x}_1$, $\tilde{x}_2$, and $\tilde{u}$ are small-amplitude signals. From the state equations (2.60), we then have

$$\begin{aligned} \frac{d}{dt}\tilde{x}_1 &= x_2^\circ + \tilde{x}_2 = \tilde{x}_2, \\ \frac{d}{dt}\tilde{x}_2 &= \mathrm{f}_2(x_1^\circ + \tilde{x}_1, x_2^\circ + \tilde{x}_2, u^\circ + \tilde{u}). \end{aligned}$$

Applying a first-order Taylor series expansion to the right-hand side of the second equation gives

$$\begin{aligned} \frac{d}{dt}\tilde{x}_2 = \mathrm{f}_2(x_1^\circ,x_2^\circ,u^\circ) &+ \frac{\partial \mathrm{f}_2}{\partial x_1}\Big|_{(x_1^\circ,x_2^\circ,u^\circ)}\tilde{x}_1 + \frac{\partial \mathrm{f}_2}{\partial x_2}\Big|_{(x_1^\circ,x_2^\circ,u^\circ)}\tilde{x}_2 \\ &+ \frac{\partial \mathrm{f}_2}{\partial u}\Big|_{(x_1^\circ,x_2^\circ,u^\circ)}\tilde{u} + d. \end{aligned}$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

The final term d represents the error in the Taylor series approximation. After computing partial derivatives, we obtain

$$\frac{d}{dt}\tilde{x}_1 = \tilde{x}_2.$$
$$\frac{d}{dt}\tilde{x}_2 = \alpha\tilde{x}_1 + \beta\tilde{u} + d \qquad \text{with} \qquad \alpha = 2\frac{\kappa}{m}\frac{(u^\circ)^2}{(x_1^\circ)^3}, \qquad \beta = -2\frac{\kappa}{m}\frac{u^\circ}{(x_1^\circ)^2}.$$

This can be represented as a linear state space model with disturbance:

$$\frac{d}{dt}\tilde{x} = \begin{bmatrix} 0 & 1 \\ \alpha & 0 \end{bmatrix}\tilde{x} + \begin{bmatrix} 0 \\ \beta \end{bmatrix}\tilde{u} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d, \qquad \tilde{y} = \tilde{x}_1. \tag{2.62}$$

There is a hidden approximation in (2.62), since d is in fact a nonlinear function of (x,u). In control design, this approximation is taken one step further by setting $d \equiv 0$, to obtain the linear model (2.20). The approximate model is not very useful for simulations, but often leads to effective control solutions.

2.7.4 CartPole

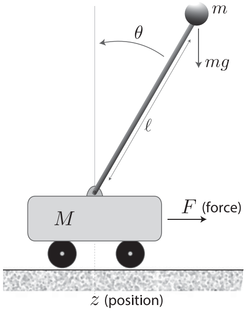


Figure 2.13 CartPole.

The next example has a long history within the control systems literature [14, 258, 331], and was introduced to the RL literature in early research of Barto et al. [26]. It is today a popular test example on openai.com. A history from the perspective of control education can be found in [385], which provides the dynamic equations with state $x = (z,\dot{z},\theta,\dot{\theta})$, where z is the horizontal position of the cart, and the angle θ is as shown in Figure 2.13.

The control design goal is regulation: Keep $\theta = 0$ while the cart is moving at some desired speed, or some desired fixed position. The aforementioned references describe several successful strategies to swing the pole up to a desired position without excessive energy. A normalized model used in [385] is given by

$$\begin{aligned} \tfrac{d}{dt}z &= \tfrac{d}{dt}x_1 = x_2, & \tfrac{d}{dt}x_2 &= u, \\ \tfrac{d}{dt}\theta &= \tfrac{d}{dt}x_3 = x_4, & \tfrac{d}{dt}x_4 &= \sin(x_3) - u\cos(x_3). \end{aligned} \tag{2.63}$$

The state equations are easily linearized near the equilibrium $u^e = 0$ and $x^e = (z^e,0,0,0)^\intercal$ for any z^e: using the first-order Taylor series approximations $\sin(x_3) \approx x_3$ and $\cos(x_3) \approx 1$, we obtain as in the derivation of (2.62)

$$\begin{aligned} \tfrac{d}{dt}\tilde{x}_1 &= \tilde{x}_2, & \tfrac{d}{dt}\tilde{x}_2 &= u, \\ \tfrac{d}{dt}\tilde{x}_3 &= \tilde{x}_4, & \tfrac{d}{dt}\tilde{x}_4 &= \tilde{x}_3 - u + d. \end{aligned} \tag{2.64}$$

Ignoring the "disturbance" (error term) d, the ODE (2.64) is a version of the state space model (2.20) with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}.$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

The matrix A is not Hurwitz, with eigenvalues at ± 1 and repeated eigenvalues at 0. This was anticipated at the start: it is unlikely that the pendulum will remain upright with a constant "open-loop" input, $u_t \equiv 0$. The linear model is of great value for insight, and for designing a linear feedback law to keep the system near the equilibrium:

$$\tilde{u} = -K\tilde{x}.$$

Methods to obtain the 4×1 matrix K through optimal control techniques will be investigated later in the book.

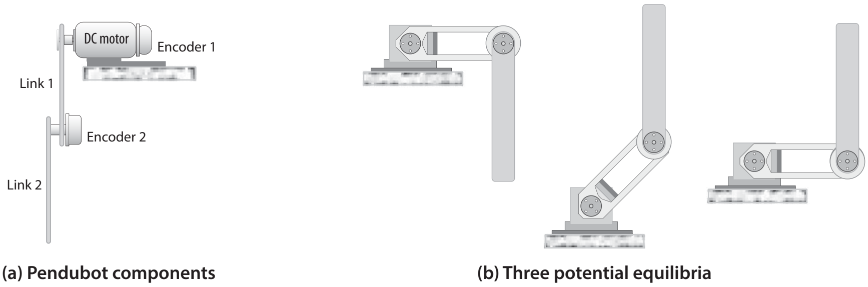
In conclusion, we know what to do locally, but the linearization provides no insight whatsoever on how to swing the pendulum up to the desired vertical position. The robotics community has developed ingenious specialized techniques for classes of nonlinear control problems that include CartPole as a special case (see [14, 82, 331, 385], and Exercise 3.10 for a survey of the approach of [14]). In the near future, we hope to marry existing control approaches with model-free techniques from RL to obtain reliable control designs in more complex settings.

2.7.5 Pendubot and Acrobot

Figure 2.14 shows a photograph of the *Pendubot* as it appeared in the robotics laboratory at the University of Illinois in the 1990s [29, 330] and a sketch indicating its component parts. It is similar to Sutton's Acrobot [341], which is another example that is currently popular on openai.com. The control objective is similar to CartPole: starting from any initial condition, swing the Pendubot up to a desired equilibrium, without excessive energy.

The value of this example is explained in the introduction of [330], where they compare to *CartPole*, and a variation of Furuta [137]:

> The balancing problem for the Pendubot may be solved by linearizing the equations of motion about an operating point and designing a linear state feedback controller, very similar to the classical cart-pole problem ... One very interesting distinction of the Pendubot over both the classical cart-pole system and Furuta's system is the continuum of balancing positions. This feature of the Pendubot is pedagogically useful in several ways, to show students how the Taylor series linearization is operating point dependent and for teaching controller switching and gain scheduling. Students can also easily understand physically how the linearized system becomes uncontrollable at $q_1 = 0, \pm\pi$. [This excerpt refers to the first and third illustrations shown in Figure 2.14b, with q_1, q_2 joint angles shown in Figure 2.15.]

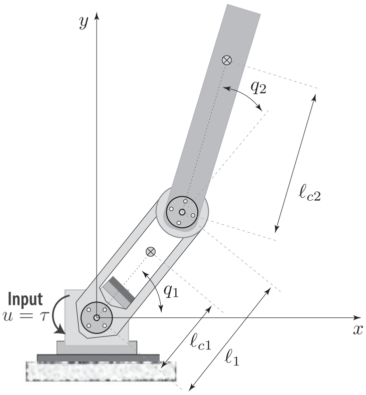


(a) Pendubot components

(b) Three potential equilibria

Figure 2.14 (a) The Illinois Pendubot, showing component parts. (b) A continuum of equilibrium positions.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press



$$
\begin{aligned}
d_{11} &= m_1\ell_{c1}^2 + m_2(\ell_1^2 + \ell_{c2}^2 + 2\ell_1\ell_{c2}\cos(q_2)) + I_1 + I_2\\
d_{22} &= m_2\ell_{c2}^2 + I_2\\
d_{12} &= d_{21} = m_2(\ell_{c2}^2 + \ell_1\ell_{c2}\cos(q_2)) + I_2\\
h_1 &= -m_2\ell_1\ell_{c2}\sin(q_2)\dot q_2^2 - 2m_2\ell_1\ell_{c2}\sin(q_2)\dot q_2\dot q_1\\
h_2 &= m_2\ell_1\ell_{c2}\sin(q_2)\dot q_1^2\\
\phi_1 &= (m_1\ell_{c1} + m_2\ell_1)g\cos(q_1) + m_2\ell_{c2}g\cos(q_1+q_2)\\
\phi_2 &= m_2\ell_{c2}g\cos(q_1+q_2)
\end{aligned}
$$

Figure 2.15 Coordinate description of the Pendubot: ℓ_1 is the length of the first link, and ℓ_{c1}, ℓ_{c2} are the distances to the center of mass of the respective links. The variables q_1, q_2 are joint angles of the respective links, and the input is the torque applied to the lower joint.

The Pendubot consists of two rigid aluminum links: Link 1 is directly coupled to the shaft of a DC motor mounted to the end of a table. Link 1 also includes the bearing housing for the second joint. Two optical encoders provide position measurements: One is attached at the elbow joint, and the other is attached to the motor. Note that no motor is directly connected to link 2 – this makes vertical control of the system, as shown in the photograph, extremely difficult!

The system dynamics can be derived using the so-called Euler–Lagrange equations found in robotics textbooks [332]:

$$d_{11}\ddot q_1 + d_{12}\ddot q_2 + h_1 + \phi_1 = \tau, \tag{2.65a}$$

$$d_{21}\ddot q_1 + d_{22}\ddot q_2 + h_2 + \phi_2 = 0, \tag{2.65b}$$

where the variables can be deduced from Figure 2.15. Consequently, this model may be written in state space form, $\frac{d}{dt}x = \mathrm{f}(x,u)$, where $x = (q_1, q_2, \dot q_1, \dot q_2)^{\intercal}$, and f is defined from the preceding equations.

This model admits various equilibria: For example, when $u^e = \tau^e = 0$, the vertical downward position $x^e = (-\pi/2, 0, 0, 0)$ is an equilibrium, as illustrated on the right-hand side of Figure 2.14. Three other possibilities are shown in Figure 2.14b, each with $\tau^e \neq 0$.

A fifth equilibrium is obtained in the upright vertical position, with $\tau^e = 0$ and $x^e = (+\pi/2, 0, 0, 0)^{\intercal}$. It is clear from the drawing shown on the left-hand side of Figure 2.14 that the upright equilibrium is strongly unstable in the sense that with $\tau = 0$, it is unlikely that the physical system will remain at rest. Nevertheless, the velocity vector vanishes, $\mathrm{f}(x^e, 0) = \mathbf{0}$, so by definition the upright position is an equilibrium when $\tau = 0$.

Although they are complex, we may again linearize these equations about the vertical equilibrium. With the input u equal to the applied torque, and the output y equal to the lower link angle, the resulting state space model is defined by the following set of matrices in the 1990s vintage system described in [330]:

$$A = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 51.9243 & 0 & -13.9700 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -52.8376 & 068.4187 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 15.9549 \\ 0 \\ -29.3596 \end{bmatrix}, \tag{2.66}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \qquad D = 0.$$

Postscripts

For those students who have had a course in undergraduate control systems, the corresponding transfer function has the general form

$$P(s) = k\frac{(s-\gamma)(s+\gamma)}{(s-\alpha)(s+\alpha)(s-\beta)(s+\beta)},$$

with $k > 0$ and $0 < \alpha < \gamma < \beta$. The variable "$s$" corresponds to differentiation. Writing

$$P(s) = k\frac{s^2-\gamma^2}{s^4 - 2(\alpha^2+\beta^2)s^2 + \alpha^2\beta^2},$$

the transfer function notation $Y(s) = P(s)U(s)$ denotes the ODE model:

$$\frac{d^4}{dt^4}\tilde{y} - 2(\alpha^2+\beta^2)\frac{d^2}{dt^2}\tilde{y} + \alpha^2\beta^2\tilde{y} = k[\frac{d^2}{dt^2}u - \gamma^2 u].$$

The roots of the denominator of $P(s)$ are $\{\pm\alpha, \pm\beta\}$, which correspond with the eigenvalues of A. The positive eigenvalues mean that A is not Hurwitz. The fact that $P(s_0) = 0$ for the positive value $s_0 = \gamma$ implies more bad news (a topic far beyond the scope of this book, but the impact of zeros in the right-half plane is worth reading about in basic texts, such as [7, 15, 76, 205]).

2.7.6 Cooperative Rowing

In a sculling boat, each rower has two oars or sculls, one on each side of the boat. The control system discussed here concerns coordination of N *individual scullers* (meaning just one rower per boat) that are part of a single team. You can see five of N teammates on the left-hand side of Figure 2.16. The team objective is to maintain constant velocity toward a target (let's say, the island of Kaua'i), and also maintain "social distance" between boats.

A state space model might be formulated as follows. Let z^i_t denote the distance from the origin, and u^i_t the force exerted by the rower at time t. Taking into account the fact that

Figure 2.16 Cooperative rowing with partial information.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

drag increases with speed, and applying once more Newton's law, $f = ma$, results in the following system equations:

$$\frac{d^2}{dt^2}z^i = -a_i\frac{d}{dt}z^i + b_iu^i - d^i,$$

in which $\{a_i,b_i\}$ are positive scalars, and the disturbance $\{d^i_t\}$ is left unmodeled. If we ignore the disturbance (for the purposes of control design), we can pose the rowing game as a linear-quadratic optimal control problem: a topic covered in Sections 3.1 and 3.6. We will see that this will result in a policy of the form

$$u^i = K^ix + r^i,$$

where x is the $2N$-dimensional vector of positions and velocities for all the rowers, K^i is a $2N$-dimensional row vector, and r^i is a scalar function of time that depends upon the tracking goal. Implementation of this policy requires that each rower know the position and velocity of every other rower at each time. Let's think about how the rowers might cooperate without so much data.

Imagine that each rower only views the nearest neighbors to the left and right. This breaks the team of size N into (overlapping) subteams of size three that coordinate individually. Unfortunately, if N is large, it is known that this distributed control architecture can lead to large oscillations in the positions of the boats with respect to the distant island [130].

The theory of *mean-field games* suggests that a more robust strategy is obtained with just a bit of global information: Assume that at each time t, rower i has access to three scalar observations: her own position and velocity z^i_t,v^i_t, and the average position of all rowers:

$$\bar{z}_t = \frac{1}{N}\sum_{j=1}^{N} z^j_t. \tag{2.67}$$

One possibility is to *pretend* that $x^i_t = (z^i_t,v^i_t,\bar{z}_t)^{\intercal}$ evolves according to a state space model of the form (2.19), in which case it is appropriate to search for a state feedback policy $u^i_t = \phi^i(x^i_t)$.

Before fixing the architecture of the policy, it is essential to consider the goals. Since we have assumed that social distancing is managed through an independent control mechanism, there remain only two:

$$z^i_t \approx \bar{z}_t\,, \qquad v^i_t = \frac{d}{dt}z^i_t \approx v^{\text{ref}} \qquad \text{for all large } t.$$

Based on the discussion in Section 2.3.2, we might obtain better coordination through the introduction of a fourth variable, defined as the integral of the position error

$$z^{I\,i}_t = z^{I\,i}_0 + \int_0^t [z^i_r - \bar{z}_r]\,dr,$$

or the discounted approximation,

$$z^{I\,i}_t = z^{I\,i}_0 + \int_0^t e^{\varrho(t-r)}[z^i_r - \bar{z}_r]\,dr$$

with $\varrho > 0$. Once we have made our choice, we then search for a policy defined as a function of the four variables, $u^i_t = \phi^i(z^i_t,v^i_t,\bar{z}_t,z^{I\,i}_t)$.

However, *do not forget that this is a game* The “best” choice of ϕ^i will depend upon the choice of ϕ^j for all $j \neq i$. We might experiment with “best response” schemes designed to learn a collection of policies $\{\phi^i : 1 \leq i \leq N\}$ that work well for all. Best response is also behind the RL training in AlphaZero [322].

2.8 Exercises

2.1 *Controllable Canonical Form.* Consider the state space model (2.18) with $\mathsf{X} = \mathbb{R}^3$.

(a) If the input is defined by $u(k) = -Kx(k) + v(k)$ for a 1×3 gain matrix K, obtain a state space model in controllable canonical form (with new input $\boldsymbol{v}$).

(b) For the special case $n = 3$ (so that F is a 3×3 matrix), design K so that the eigenvalues of $F - GK$ are each located at $1/2$. Perform this calculation by hand. Your answer will depend upon $\{a_1, a_2, a_3\}$. Based on your effort, explain why this is called controllable canonical form!

(c) Think a bit deeper: Have you solved a control problem? With state $x(k)$ defined via (2.17), would you say you are making good use of your output measurements?

If you are baffled, then seek advice from your professor, fellow students, and a good book on state feedback methods!

2.2 *Controllability and Observability.* Consider the linear state space model

$$\begin{aligned} x(k+1) &= Fx(k) + Gu(k)\,, \quad x(0) = \binom{1}{1}, \\ y(k) &= Hx(k) \qquad \text{with } F = \begin{bmatrix} 0.5 & 1 \\ 0 & 2 \end{bmatrix}, \ G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ H^{\intercal} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{2.68}$$

If you have taken a state space controls course, then you know that this system is not controllable and not observable. If you don’t have this background, then you might be able to guess the definitions of these terms after completing this exercise.

(a) Can you find a feedback law $u(k) = \phi(x(k))$ that results in a bounded output $\boldsymbol{y}$?

(b) Does the situation improve if $H = [1\ 0]$?

(c) How about if $G = [0\ 1]^{\intercal}$?

2.3 *Stabilizability.* The state space model is called *stabilizable* if there is a feedback law $u(k) = \phi(x(k))$ that results in a closed-loop system that is globally asymptotically stable. The example in Exercise 2.2 is not stabilizable.

Perform the following calculations with $F = \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix}$ and $G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$:

(a) Design the gain in $u(k) = -Kx(k)$ so that $F - GK$ has repeated eigenvalues (you will see that you do not have choice in the value). Is K unique?

(b) Solve the Lyapunov equation (2.43) with F replaced by the closed-loop matrix $F - GK$ from (a), and with $S = I$.

(c) Denote $y(k) = x_1(k) = Hx(k)$. Suppose that our goal is to ensure that $y(k) \to r$ as $k \to \infty$, with r a constant. Modify your control design as follows:

$$u(k) = -K_1\tilde{y}(k) - K_2 x_2(k) - K_3 z^{\prime}(k),$$

where $\tilde{y}(k) = y(k) - r$ and $z^{\prime}(k+1) = z^{\prime}(k) + \tilde{y}(k)$ (review discussion surrounding (2.11)). Find $\bar{K}_3 > 0$ sufficiently small so that the system remains stable for $0 \leq K_3 \leq \bar{K}_3$. This is possible because of the inherent robustness of feedback (you verified stability when $K_3 = 0$).

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

(d) Obtain a state space model for the system in a closed loop, with augmented state $x^a = (x_1, x_2, z^I)$:

$$x^a(k+1) = F^a x^a(k) + G^a r,$$

where F^a is 3×3 and G^a is 3×1. Plot the eigenvalues of F^a for a range of values of $K_3 > 0$, and comment on your findings.

Solve the equilibrium equation (for your favorite control design): $x^a(\infty) = F^a x^a(\infty) + G^a r$. Is your equilibrium $x^a(\infty)$ consistent with your control goals?

Obtain a plot of $y(k)$ as a function of k, with initial condition $x_1(0) \gg r$, and verify that it converges to the desired limit and at the predicted rate.

2.4 Consider the scalar state space model, $x(k+1) = x(k) - \alpha x(k)^3$.

(a) Show that the origin is stable in the sense of Lyapunov, and estimate the region of attraction (which will depend upon α).

(b) Explain why this state space model is not globally asymptotically stable.

The state process $\boldsymbol{x}$ is in fact an Euler approximation of the ODE $\frac{d}{dt}x = -x^3$. See Exercise 2.15 for some interesting features of the solution.

Control Systems in Continuous Time

For simulating an ODE, you might try `ode45` in Matlab. There are several Python alternatives.

2.5 *Integral Control Design.* The temperature T in an electric furnace is governed by the linear state equation

$$\frac{d}{dt}T = u + w,$$

where u is the control (voltage) and w is a constant disturbance due to heat losses. It is not directly observed. It is desired to regulate the temperature to a steady-state value prescribed by the set-point $T = T^0$, where T^0 is your comfort temperature. The following should be solved by hand:

(a) Design a state-plus-integral feedback controller to *guarantee* that $T_t \to T^0$ as $t \to \infty$, for any constant w. This can take the form $u = -K_1(T - T^0) - K_2 z^I$ with

$$z^I_t = z^I_0 + \int_0^t [T_r - T^0]\, dr.$$

The closed-loop poles should have natural frequency $\omega_n \approx 1$ (that is, the eigenvalues of the 2×2 matrix that defines the closed-loop state space model should satisfy $|\lambda| \approx 1$.)

(b) To what value does the control u_t converge as $t \to \infty$? Has the controller "learned" w?

2.6 Solve the following based on the linear state space model $\frac{d}{dt}x = Ax$ with $A = \begin{bmatrix} -1 & 4 \\ 0 & -1 \end{bmatrix}$.

(a) Show that $V(x) = \|x\|^2 = x_1^2 + x_2^2$ is *not* a Lyapunov function.

(b) Find a quadratic function V that is.

(c) Consider the Euler approximation $x(k+1) = Fx(k)$ with $F = I + \Delta A$, and $\Delta > 0$. Estimate the range of $\Delta > 0$ for which your function V from part (b) is a Lyapunov function for this discrete time system. Is this range complete? That is, does it include all values for which the eigenvalues of F lie in the open unit disk in $\mathbb{C}$?

2.7 In this exercise, you will consider a particular control architecture for cooperative rowing, using a simplification of the model described in Section 2.7.6. Consider the homogeneous and disturbance-free system

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

$$\frac{d^2}{dt^2} z^i = -a \frac{d}{dt} z^i + u^i \,, \qquad 1 \le i \le N$$

with $a > 0$. The goal is to maintain $v^i_t \overset{\text{def}}{=} \frac{d}{dt} z^i_t \approx v^{\text{ref}}$ for all t, and $z^i_t \approx \bar{z}_t$ for each i,t, with $\bar{z}_t$ the average position (recall (2.67)). We wish to achieve these objectives without requiring that each rower have complete observations.

The following control architecture is of the category studied in [130]:

$$u^i = -K_-[z^i - z^{i-1}] - K_+[z^i - z^{i+1}] - K_v[\tfrac{d}{dt} z^i - v^{\text{ref}}] \,, \qquad 1 \le i \le N,$$

where for notational convenience we interpret $z^0 = z^N$ and $z^{N+1} = z^1$. This architecture is well motivated in terms of goals, and the desire to make decisions based on only local information. Unfortunately, theory predicts problems when N is large.

(a) Describe the closed-loop dynamics as a $2N$-dimensional state space model, with constant input v^{ref}. This will have the form, for some matrix K and vector g,

$$\frac{d}{dt} x = (A - BK)x + g v^{\text{ref}}.$$

The remainder of the exercise is numerical, with $a = v^{\text{ref}} = 1$, $K_- = K_+$, and several values of N (say, 10, 500, 5,000):

(b) Choose nonnegative gains K_+, K_v so that the closed-loop system is stable, in the sense that the key error terms are bounded as functions of time and convergent:

$$e^i_z = \lim_{t \to \infty} (z^i_t - \bar{z}_t) \,, \qquad e^i_v = \lim_{t \to \infty} (v^i_t - v^{\text{ref}}).$$

See if you can obtain gains so that $|e^i_v| \le 0.05$.

Note that you do not yet have any tools to efficiently compute the control gains. Just experiment until you find something that works.

(c) Obtain a plot of the eigenvalues of $A - BK$ for the chosen values of N. Do you find complex eigenvalues? Eigenvalues at zero?

(d) Simulate your control design for various nonideal initial conditions. Think hard about how to plot your results to display the poor behavior of these scullers. Discuss your findings.

2.8 Let's now consider the rowing game in which each rower has access to the average position (2.67), and the control architecture

$$u^i = -K_p[z^i - \bar{z}] - K_I z^{I\,i} - K_v[\tfrac{d}{dt} z^i - v^{\text{ref}}] \,, \qquad 1 \le i \le N, \tag{2.69}$$

where in the notation of Section 2.7.6,

$$z^{I\,i}_t = z^{I\,i}_0 + \int_0^t [z^i_r - \bar{z}_r] \, dr.$$

Repeat (a)–(d) of Exercise 2.7 based on this policy.

2.9 You are given a nonlinear input–output system defined by the nonlinear differential equation:

$$\ddot{y} = y^2(u - y) + 2\dot{u}. \tag{2.70}$$

(a) Obtain a two-dimensional nonlinear state space representation with output y, input u, and states $x_1 = y$ and $x_2 = \dot{y} - 2u$.

(b) Linearize this system of equations around its equilibrium output trajectory when $u \equiv 1$, and write it in state space form.

(c) *For those of you with background in classical control:* Find the transfer function for the linear system obtained in (b), and comment on the implications.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

(d) Obtain a linear compensator $u = -K\tilde{x}$ for the linearization, where $\tilde{x} = (y-1, \dot{y})^\intercal$. To be successful, you want $\tilde{x}_t \to 0$ as $t \to \infty$ for each initial condition for which $\|\tilde{x}_0\|$ is sufficiently small.

2.10 We now consider (2.70) subject to a constant disturbance:

$$\ddot{y}(t) = y^2(u-y) + 2\dot{u} + d,$$

where the value of d is not known in advance. In this case, we cannot expect perfect tracking unless we introduce integral control:

$$u = -K\tilde{x}^a, \qquad \text{where} \quad \tilde{x}^a = (y-1, \dot{y}, z^I)^\intercal, \quad z^I_t = \int_0^t (y-1)\, dr.$$

Find a 1×3 row vector K so that this control design is stabilizing in the sense that $\tilde{x}^a$ is bounded, and $\tilde{x}$ vanishes for "small" initial conditions. Perform simulations to verify that perfect tracking is achieved for initial conditions near the equilibrium value and any fixed value of d satisfying $|d| \le 1$.

2.11 Consider the state space model $\frac{d}{dt}x = Ax + Bu$; $y = Cx$, where A is similar to a diagonal matrix. That is, $\Lambda = V^{-1}AV$ where Λ is a diagonal matrix, with each $\Lambda(i,i)$ an eigenvalue of A, and V is a matrix whose columns are eigenvectors.

(a) Obtain a state space model for $\bar{x} = V^{-1}x$, of the form $\frac{d}{dt}\bar{x} = \bar{A}\bar{x} + \bar{B}u$; $y = \bar{C}\bar{x}$, by finding representations for $(\bar{A}, \bar{B}, \bar{C})$. This state space representation is called *modal form.*

The remainder of the problem is numerical, using

$$A = \begin{bmatrix} 8 & -7 & -2 \\ 8 & -10 & -4 \\ -4 & 5 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad C = [1\ 0\ 0].$$

(b) Find the eigenvalues and eigenvectors of A, and verify that the matrix $\Lambda = V^{-1}AV$ is indeed diagonal when V is the matrix of eigenvectors.

(c) Obtain a state space model in modal form.

2.12 *Foster's Criterion.* Suppose that $\frac{d}{dt}x = \mathrm{f}(x)$ is a nonlinear state space model on $\mathbb{R}^n$. Assume also that there is a C^1 function $V\colon \mathbb{R}^n \to \mathbb{R}_+$, and a set S such that

$$\langle \nabla V(\theta), \mathrm{f}(\theta) \rangle \le -1, \qquad \theta \in S^c. \tag{2.71}$$

Foster introduced a version of this stability criterion for Markov chains in the middle of the last century [135].

(a) Show that $T_K(x) \le V(x)$ for $x \in \mathbb{R}^n$, where

$$T_K(x) = \min\{t \ge 0 : x_t \in K\}, \qquad x_0 = x \in \mathbb{R}^n.$$

(b) In the special case of a stable linear system [$\mathrm{f}(x) = Ax$, with A Hurwitz], show that a solution to (2.71) is given by $V(x) = \log(1 + x^\intercal M x)$ for some matrix $M > 0$, and with $S = \{x : \|x\| \le k\}$ for some scalar k.

(c) Find an explicit V, S for $A = \begin{bmatrix} -1 & 4 \\ 0 & -1 \end{bmatrix}$ (the matrix used in Exercise 2.6).

2.13 Consider the nonlinear state space model on the real line,

$$\frac{d}{dt}x = \mathrm{f}(x) = \frac{1-e^x}{1+e^x} = -\tanh(x/2).$$

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

(a) Sketch f as a function of x, and from this plot explain why $x^e = 0$ is an equilibrium, and this equilibrium is globally asymptotically stable.

(b) Find a solution to the Poisson inequality (2.39): $\langle \nabla V, f \rangle \le -c + \overline{\eta}$, with $c(x) = x^2$ and $\overline{\eta} < \infty$. You might try a polynomial, or a log of a polynomial of $|x|$. See if you can find a solution with $\overline{\eta} = 0$.

(c) Find a solution V to Foster's criterion (2.71), with $S = [-k, k]$ for some $k > 0$. Also, explain why $T_S(x)$ is not finite valued using $S = \{0\}$ (that is, $k = 0$).

2.14 Suppose that one wants to minimize a C^1 function $V : \mathbb{R}^n \to \mathbb{R}_+$. A necessary condition for a point $x^\circ \in \mathbb{R}^n$ to be a minimum is that it be a *stationary point*: $\nabla V(x^\circ) = \mathsf{0}$.

Consider the steepest descent algorithm $\frac{d}{dt}x = -\nabla V(x)$. Find conditions on the function V to ensure that a given stationary point x° will be asymptotically stable for this equation. *One approach*: Find conditions under which the function V is a Lyapunov function for this state space model. We will return to this topic in Section 4.4.

2.15 Consider the nonlinear state space model on the real line,

$$\frac{d}{dt}x = \mathrm{f}(x) = -x^3.$$

(a) Sketch f as a function of x, and from this plot explain why $x^e = 0$ is an equilibrium, and this equilibrium is globally asymptotically stable.

(b) Find a solution to the Poisson inequality (2.39) with $c(x) = x^2$: $\langle \nabla V, f \rangle \le -c + \overline{\eta}$ with $\overline{\eta} < \infty$. You might try a polynomial, or a log of a polynomial in of $|x|$. See if you can find a solution with $\overline{\eta} = 0$.

(c) Find a bounded solution to Foster's criterion (2.71).

2.16 Consider the Van der Pol oscillator, described by the pair of equations

$$\begin{aligned} \frac{d}{dt}x_1 &= x_2, \\ \frac{d}{dt}x_2 &= -(1 - x_1^2)x_2 - x_1. \end{aligned} \tag{2.72}$$

(a) Obtain a linear approximate model $\frac{d}{dt}\tilde{x} = A\tilde{x}$ around the unique equilibrium $x^e = \mathsf{0}$.

(b) Verify that A is Hurwitz, and obtain a quadratic Lyapunov function V for the linear model.

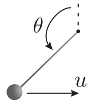
(c) Show that V is also a Lyapunov function for (2.72) on the set $S_V(r)$ defined in (2.28), for some $r > 0$. That is, show that the drift inequality (2.37) holds whenever $x_t \in S_V(r)$.

Conclude that the set $S_V(r) \subset \Omega \equiv$ *the region of attraction for* x^e.

(d) *Can we find the entire region of attraction?* Take a box around the origin $B = \{x : -m \le x_1 \le m, \ -m \le x_2 \le m\}$ for some integer m (definitely larger than 1, but less than 10 will suffice). Choose N values $\{x^i\} \subset B$ (say, $N = 10^3$), and simulate the ODE for each i, with $x_0 = x^i$, to test to see if $x_t \in S_V(r)$ for some $t < \infty$, and hence $x^i \in \Omega$.

Why does entry to $S_V(r)$ guarantee that x_0 is in the region of asymptotic stability?

2.17 *Inverted Pendulum with Friction*. Consider the pendulum with applied force u, and "damping force" $b\dot{\theta}$:



$$\frac{d}{dt}x = \mathrm{f}(x,u) = \begin{bmatrix} x_2 \\ g\sin(x_1) - u\cos(x_1) \end{bmatrix},$$

where $x = (\theta, \dot{\theta})^{\intercal}$, and $a, b > 0$. Note that the location of $\theta = 0$ is now at the top, in contrast to what is shown in Figure 2.8. This is because our goal here is to swing the pendulum up and stabilize in the unstable upward position (corresponding to $\theta = 0$ in this exercise).

Envision the state space as an infinite tube: equate θ and $\theta + 2\pi n$ for any n.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

(a) Obtain a linearized state space model with equilibrium (x^e, u^e) for each possible equilibrium (you will find that $x_2^e = 0$ is required). Comment on the challenge for $x_1^e = \pm\pi/2$.

(b) Obtain a linear feedback law that results in $x^e = 0$ asymptotically stable (locally).

You will obtain a control solution that is globally asymptotically stable in Exercise 3.10 after you learn a few concepts from optimal control in the next chapter.

2.18 *Linear Control Design for MagBall.* Our goal is to maintain the ball at rest at some preassigned distance r from the magnet.

(a) Find u° so that $x^\circ = (r,0)^\intercal$ is an equilibrium: $\mathrm{f}(x^\circ, u^\circ) = 0$. Based on the linearization (2.62), design a linear control law for (2.62), of the form

$$\tilde{u} = -K\tilde{x} = -K_1\tilde{x}_1 - K_2\tilde{x}_2$$

with $\tilde{x}_1 = x - x^\circ$ and $\tilde{x}_2 = x_2$. Make sure that your solution results in $A - BK$ Hurwitz.

(b) A difficulty with this design is that u° depends on c/m, which may not be known. Modify your design as follows:

$$\tilde{u} = -K\tilde{x}^a, \qquad \text{where} \quad \tilde{x}^a = (\tilde{x}_1, \tilde{x}_2, z^I)^\intercal, \quad z_t^I = \int_0^t \tilde{x}_1\, dr \tag{2.73}$$

with $K = [K_1, K_2, K_3]$. This is known as proportional-integral-derivative (PID) control. Obtain a third-order linear state space model, and choose $K_3 > 0$ so that the 3×3 matrix remains Hurwitz, and the transient behavior remains "good" (you decide what that means). Observe that the equilibrium condition $\frac{d}{dt}z^I = 0$ implies that $x_1^e = r$.

(c) Simulate as in Exercise 2.16 to estimate the region of attraction (you may restrict the simulation to initial conditions with zero velocity).

2.19 *Feedback Linearization for MagBall.* For systems with simple nonlinearities, there is a "brute-force" approach to obtain a linear model. For MagBall, we may view $\nu = u^2/x_1^2$ as an input, from which we obtain a linear system via (2.60):

$$\frac{d}{dt}x_1 = x_2,$$
$$\frac{d}{dt}x_2 = g - \frac{\kappa}{m}\nu.$$

(a) As in the previous exercise, obtain a control law $\nu = -K\tilde{x}$, where K_1 and K_2 are parameters chosen for stability and good transient response.

(b) Obtain an expression for the equilibrium x^e for the closed-loop system using the gain K obtained in (a). This is obtained by setting $\frac{d}{dt}x_i = 0$ for $i = 1,2$.

(c) Modify your design as in (2.73): $\nu = -K\tilde{x}^a$. Find $K_3 > 0$ so that the transient behavior remains "good."

(d) You will need to modify your policy in (c) so ν is nonnegative valued, say $\nu = \phi(\tilde{x}^a) = \max(0, K\tilde{x} + K_3 z^I)$. The current applied to the magnet using this policy is then

$$u = x_1\sqrt{\phi(\tilde{x}^a)}. \tag{2.74}$$

Simulate, and estimate the region of attraction. You may restrict the simulation to zero initial velocity.

How does the region of attraction change when κ is doubled? κ divided by 2? Do not change your policy! The point is to check if your solution is robust to an inaccurate model.

Warning: Recall that it is not possible to achieve convergence to x° from any initial condition.

Note: See [199] for a survey on feedback linearization – a topic that has far more depth than is obvious from this example.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

Matrix Algebra

2.20 Let A be an $n \times n$ matrix, and suppose that the following infinite sum exists:

$$U = I + A + A^2 + A^3 + \cdots ,$$

where I denotes the identity matrix. Verify that U is the inverse of the matrix $I - A$.

Note that this coincides with the Taylor series expansion of $f(x) = 1/(1-x)$ when $n = 1$.

2.21 Two square matrices A and $\bar{A}$ are called *similar* if there is an invertible matrix M such that

$$A = M^{-1}\bar{A}M.$$

Obtain the following for two similar matrices A and $\bar{A}$.

(a) Show that A^m is similar to $\bar{A}^m$ for any $m \geq 1$, where the superscript "^{m}" denotes matrix product,

$$A^1 = A, \qquad A^m = A(A^{m-1}), \qquad m \geq 1.$$

(b) Show that v is an eigenvector for A if and only if Mv is an eigenvector for $\bar{A}$.

(c) Suppose that $\bar{A}$ is diagonal ($\bar{A}_{ij} = 0$ if $i \neq j$). Suppose moreover that $|\bar{A}_{ii}| < 1$ for each i. Conclude that $I - A$ admits an inverse by applying Exercise 2.20.

2.22 *Matrix Exponential.* Compute e^{At} for all t for the 2×2 matrix

$$A = aI + bJ\,, \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

The notation is intended to be suggestive: $J^2 = -I$.

It is not difficult to obtain a formula for A^m for each m, as required in the definition (2.45). With $a < 0$ and $b \neq 0$, describe the solution to $\frac{d}{dt}x = Ax$ with a nonzero initial condition.

2.9 Notes

The notion of "state" is flexible in both control theory [15] and reinforcement learning [337, 338]. The motivation is the same in each field: For the purposes of online decision making, replace the full history of observations at time k by some finite-dimensional "sufficient statistic" $x(k)$. One constraint that arises in RL is that the state process must be directly observable; in particular, the belief state that arises in partially observed Markov decision processes (MDPs) requires the (model-based) nonlinear filter, and is hence not directly useful for model-free RL. In practice, the "RL state" is specified as some compression of the full history of observations – see [338, section 17.3] for further discussion.

For more on linear models see [7, 80] and [118] for more advanced and recent material.

Textbook treatments on Lyapunov theory can be found in [45] (nonlinear) and [7, 205] (linear). The Electrical and Computer Engineering (ECE) Department at the University of Illinois had a great course on state space methods – the lecture notes are now available online [29]. The first section of [165] contains a brief crash course on Lyapunov theory, written in the style of this book, and with applications to reinforcement learning.

Poisson's inequality (2.31) is far removed (roughly two centuries) from the celebrated equation introduced by mathematician Siméon Poisson. The motivation back then was

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press

potential theory, as defined in theoretical physics. About one century later, Poisson's equation arose as a central player in studying the evolution of the density of Brownian motion (a particular Markov process). The terminology *Poisson inequality* and *Poisson equation* is today applied to any Markov chain, with *generator* playing the role of the Laplacian. The generator takes any function $h\colon \mathsf{X} \to \mathbb{R}$ to a new function denoted $\mathcal{A}h$. In particular, the deterministic state space model (2.21) can be regarded as a Markov chain [257], and the associated generator is defined as

$$\mathcal{A}h\,(x) = h(\mathrm{F}(x)) - h(x).$$

In this notation, (2.31) becomes $\mathcal{A}V \le -c + \overline{\eta}$.

https://doi.org/10.1017/9781009051873.004 Published online by Cambridge University Press