6

Dynamical Systems

Controllers such as a thermostat for regulating the temperature in a room or a cruise controller for tracking and maintaining the speed of an automobile interact with the physical world via sensors and actuators. The relevant information about the physical environment corresponds to quantities such as temperature, pressure, and speed that evolve continuously obeying laws of physics. As a result, design and analysis of control systems requires construction of models of the physical system. In this chapter, we focus on continuous-time models of dynamical systems. This mathematically rich area is explored in details in a course on control systems. The purpose of this chapter is to give a brief introduction to the core concepts.

6.1 Continuous-Time Models

6.1.1 Continuously Evolving Inputs and Outputs

The typical architecture of a control system is depicted in figure 6.1. The physical world that is to be controlled is modeled by a component called the *plant*. The evolution of the plant can be influenced by the controller using actuators, but it also depends on uncontrollable factors from the environment, usually called *disturbances*. The controller responds to the commands from the user, called *reference inputs*, and can make its decisions based on measurements of the plant provided by the sensors. For example, in a thermostat design, the plant is the room whose temperature is to be controlled. The sensor is a thermometer that can measure the current temperature. The task of the controller is to regulate the temperature so that it stays *close* to the reference temperature set by the occupant on the thermostat. The controller can influence the temperature by adjusting the heat flow from the furnace. The plant model, in this case, needs to capture how the temperature of the room changes as a function of the heat-flow from the furnace and the difference between the room temperature and the outside temperature, which is the uncontrolled disturbance.

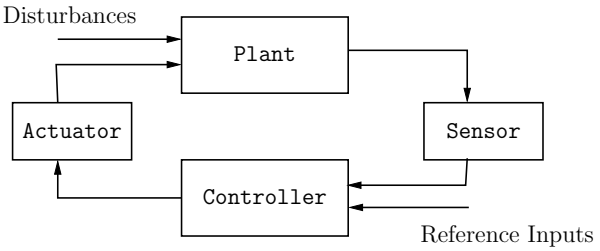

Figure 6.1: Block Diagram of a Control System

Models of dynamical systems can also be conveniently described as components with inputs and outputs that are connected to one another using block diagrams. The underlying model of computation is *synchronous* as in chapter 2 with one essential difference: while the values of the variables of a synchronous reactive component are updated in a sequence of *discrete* logical rounds, the values of the variables of a dynamical system are updated *continuously* with the passage of time. We call such components *continuous-time components*. The variables of a continuous-time component typically range over a compact set such as a closed interval of the set of reals with specified units of measurement. For example, the velocity v of a car can be modeled by a variable that ranges over the real numbers in the interval from 0 to 150 miles per hour. In our specifications of dynamical systems, we assume that every variable has the type `real` without explicitly mentioning the associated interval range.

Signals

The values a variable takes over time can then be described as a function from the time domain to `real`. Such functions from the time domain to the set of reals are called *signals*. Throughout we will assume that the time domain consists of the set of non-negative real numbers and denote this set by `time`. For a variable x, a signal over x is a function that assigns a real value to the variable x for every time t in `time`. We will denote such a function by $\overline{x}$. Thus, $\overline{x}(0)$ denotes the value of the variable x at time 0, and for every time t, $\overline{x}(t)$ denotes its value at time t.

Given a set V of variables, a signal $\overline{V}$ over V assigns values to all the variables in V as a function of time. If the set V contains k variables, then a signal over V is a function from the time domain `time` to k-tuples or vectors over `real`. The number k of variables is called the *dimensionality* of the signal. Alternatively, a k-dimensional signal can be viewed as a k-tuple of single-dimensional signals, one for each of the variables in V.

Since the domain and the range of a signal consists of real numbers or vectors, we can use the standard notion of Euclidean distance over reals, or any other notion of measuring distance that satisfies the classical properties of a metric,

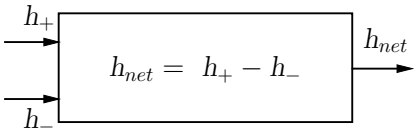

Figure 6.2: Continuous-time Component `NetHeat`

to measure how far apart two quantities are. For two vectors u and v, let $\|u - v\|$ denote the distance between u and v, and we will denote the distance of a vector u from the origin 0 by $\|u\|$. Now the standard mathematical notions of *continuity* and *differentiability* apply to signals. For example, a signal $\overline{V}$ is *continuous* if for all time values $t \in$ `time`, for all $\epsilon > 0$, there exists a $\delta > 0$ such that for all time values $t' \in$ `time`, if $\|t - t'\| < \delta$, then $\|\overline{V}(t) - \overline{V}(t')\| < \epsilon$ holds.

Example: Heat Flow

As a first example, figure 6.2 shows a continuous-time component `NetHeat` that has two input variables h_+ and h_- and a single output variable h_{net}, which denote the heat inflow, heat outflow, and net heatflow, respectively. The component `NetHeat` is a mapping from two input signals to an output signal, and its dynamics is expressed by the equation:

$$h_{net} \;=\; h_+ - h_-,$$

which says that at every time t, the value of the output h_{net} equals the expression $h_+ - h_-$. Given a signal $\overline{h}_+$ for the input variable h_+ and a signal $\overline{h}_-$ for the input variable h_-, the output signal $\overline{h}_{net}$ for the output variable h_{net} is defined by $\overline{h}_{net}(t) = \overline{h}_+(t) - \overline{h}_-(t)$ for all times t in `time`. This unique output signal is called the *response* of the component to the input signals $\overline{h}_+$ and $\overline{h}_-$. If the input signals are continuous, then so is the output signal.

Note that the computation of the output value based on the input values is expressed in a *declarative* style using *(algebraic) equations* instead in an *operational* style using *assignments*. Indeed such a declarative description is the norm in the modeling of control systems since the models are primarily used to express relationships between various signals in a mathematically precise manner so they can be subjected to analysis.

Example: Motion of a Car

The component `NetHeat` is stateless. As an example of a stateful continuous-time component, let us build a model of how the speed of a car changes as a result of the force applied to it by the engine. For the purpose of designing a cruise controller, it typically suffices to make a number of simplifying assumptions. In particular, let us assume that the rotational inertia of the wheels is negligible

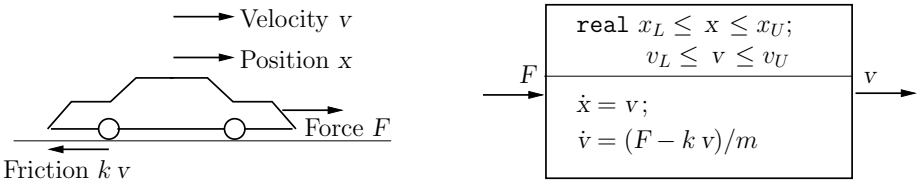

Figure 6.3: Continuous-time Component `Car` Modeling the Car Motion

and that the friction resisting the motion is proportional to the car's speed. The forces acting on the car are shown in figure 6.3. If x denotes the position of the car (measured with respect to an inertial reference) and F denotes the force applied to the car, then using the classical Newton's laws for motion, we can capture the dynamics of the car by the *differential equation*:

$$F - k\,\dot{x} = m\,\ddot{x}.$$

Here k is the coefficient of the frictional force, and m denotes the mass of the car. The quantity $\dot{x}$ denotes the first-order time derivative of the signal assigning values to the position variable x and thus captures the velocity of the car. Similarly, $\ddot{x}$ denotes the second-order derivative of this signal, that is, the acceleration of the car.

This equation of motion is modeled by the continuous-time component `Car` of figure 6.3. It uses two state variables: the variable x modeling the position of the car and the variable v modeling the velocity of the car. For every state variable, the component needs to specify its initial value. In our example, the initial value of the position x is specified by the constraint $x_L \leq x \leq x_U$, where x_L and x_U are constants that give lower and upper bounds on the initial position. This declarative specification of the initialization is equivalent to the nondeterministic assignment using the `choose` construct discussed in chapter 2: `real` x := `choose`$\{z \mid x_L \leq z \leq x_U\}$. Similarly, the initial value of the velocity v is given by the constraint $v_L \leq v \leq v_U$, where v_L and v_U are constants that give lower and upper bounds on the initial velocity.

For every state variable, the dynamics is given by specifying the first-order time derivative of the value of the state variable as a function of the input and state variables. The differential equation $\dot{x} = v$ says that the rate of change of the state variable x at each time t equals the value of the state variable v at time t, and the rate of change of the state variable v at each time t equals the value of the expression $(F - kv)/m$ at time t. The two equations together are equivalent to the original equation $F - k\,\dot{x} = m\,\ddot{x}$. The output of the car is its velocity. For every output variable, the component specifies the value of the output variable as a function of the input variables and the state variables. In this example, the output simply equals the state variable v.

To illustrate the behaviors of this model, let us choose the initial position to be x_0 and initial velocity to be v_0. Now consider the case when the input force F equals the value kv_0 at all times. Then the position x and the velocity v of the car at all times can be obtained by solving the system of differential equations

$$\dot{x} \;=\; v;\; \dot{v} \;=\; k(v_0 - v)/m$$

with the initial condition $\overline{x}(0) = x_0$ and $\overline{v}(0) = v_0$. These equations have a unique solution: the velocity stays constant at the value v_0 at all times, and the distance x increases linearly with time t and is given by the expression $x_0 + t\, v_0$. In other words, given the (constant) input signal $\overline{F}(t) = kv_0$ and the initial state (x_0, v_0), the corresponding signal describing the dynamics of the state/output variable v is given by $\overline{v}(t) = v_0$, and the signal describing the state variable x is given by $\overline{x}(t) = x_0 + t\, v_0$.

Now let us consider the case when the input force F is 0 at all times, the initial position of the car is 0, and the initial velocity is v_0. Then the position x and the velocity v of the car at all times can be obtained by solving the system of differential equations

$$\dot{x} \;=\; v;\; \dot{v} \;=\; -kv/m$$

with the initial condition $\overline{x}(0) = 0$ and $\overline{v}(0) = v_0$. Using rules of differential calculus, we can solve these equations, and the resulting signals corresponding to the position and the velocity of the car are given by:

$$\overline{v}(t) \;=\; v_0\, e^{-kt/m};\;\; \overline{x}(t) \;=\; (mv_0/k)\,[1 - e^{-kt/m}].$$

Note that the velocity decreases exponentially converging to 0, while the position increases converging to the value mv_0/k.

As a third scenario, suppose the initial position is 0, and the initial velocity is 0, and we apply a constant force F_0 to the car. Then the position x and the velocity v of the car can be obtained by solving the system of differential equations

$$\dot{x} \;=\; v;\; \dot{v} \;=\; (F_0 - kv)/m$$

with the initial condition $\overline{x}(0) = 0$ and $\overline{v}(0) = 0$. If we assume that the mass m of the car is 1000 kg, the coefficient k of friction between the tires and the road is 50, and the input force F_0 is $500N$, then the corresponding signal for the velocity is shown in figure 6.4. The velocity increases converging to the value $F_0/k = 10\, m/s$.

Definition of Continuous-Time Components

In general, the initial values for the state variables are specified by a constraint *Init*, which typically specifies an interval of possible values for each variable. As usual, $[\![Init]\!]$ specifies the set of initial states, that is, the set of states that satisfy the constraint specified by *Init*.

The dynamics of the component is specified by (1) a real-valued expression h_y for every output variable y, and (2) a real-valued expression f_x for every state variable x. Each of these expressions is an expression over the input and state variables. The value of the output variable y at time t is obtained by evaluating the expression h_y using the values of the state and input variables at time t, and the signal for a state variable x should be such that its rate of change at each time t equals the value of the expression f_x evaluated using the values of the state and input variables at time t. Thus, the execution of a continuous-time component is similar to the execution of a deterministic synchronous reactive component, except the notion of a round is now infinitesimal: at every time t, the outputs at time t are determined as a function of the inputs at time t and the state of the component at time t, and then the state is updated using the rate of change specified by the derivative evaluated using the inputs and state at time t.

The definition of a continuous-time component is now summarized below:

Continuous-time Component

A continuous-time component H has a finite set I of real-valued input variables, a finite set O of real-valued output variables, a finite set S of real-valued state variables, an initialization *Init* specifying a set $[\![Init]\!]$ of initial states, a real-valued expression h_y over $I \cup S$ for every output variable $y \in O$, and a real-valued expression f_x over $I \cup S$ for every state variable $x \in S$. Given a signal $\overline{I}$ over the input variables I, a corresponding execution of the component H is a differentiable signal $\overline{S}$ over the state variables S and a signal $\overline{O}$ over the output variables O such that

1. $\overline{S}(0) \in [\![Init]\!]$;
2. for every output variable y and time t, $\overline{y}(t)$ equals the value of h_y evaluated using the values $\overline{I}(t)$ for the input variables and $\overline{S}(t)$ for the state variables; and
3. for every state variable x and time t, the time derivative $(d/dt)\,\overline{x}(t)$ equals the value of f_x evaluated using the values $\overline{I}(t)$ for the input variables and $\overline{S}(t)$ for the state variables.

A continuous-time component with no inputs is called *closed*.

Existence and Uniqueness of Response Signals

Determining the signals for the state and output variables corresponding to a given initial state s_0 and a given input signal using mathematical analysis amounts to solving an *initial value problem* for a system of *ordinary differential equations*. We need to impose restrictions on the expressions used to define the state and output responses to ensure uniqueness of such solutions since not all differential equations are well behaved. For example, the *conditional* differential

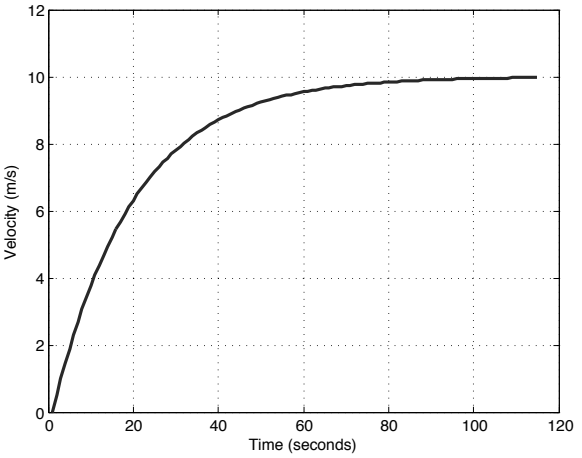

Figure 6.4: Velocity of `Car` in Response to Constant Input Force

equation

$$\dot{x} \;=\; \texttt{if}\ (x = 0)\ \texttt{then}\ 1\ \texttt{else}\ 0$$

states that the derivative of the signal is 1 at time 0 and 0 everywhere else. There is no differentiable signal $\overline{x}$ that can satisfy the given equation. The source of this problem lies in the fact that the right-hand side of this differential equation has a discontinuity at $x = 0$. If the right-hand side of a differential equation is a *continuous* function, then a solution is guaranteed to exist.

To illustrate another potential problem in solving differential equations, consider the differential equation $\dot{x} = x^{1/3}$. For the initial value $\overline{x}(0) = 0$, this differential equation has two solutions: the constant signal $\overline{x}_1(t) = 0$ and the signal $\overline{x}_2(t) = (2t/3)^{3/2}$. A classical way to avoid this problem and ensure *uniqueness* of the solution to a differential equation is to require the right-hand side to be *Lipschitz continuous*. Intuitively, Lipschitz continuity means that there is a constant upper bound on how fast a function changes: a function $f : \texttt{real}^n \mapsto \texttt{real}^n$ is said to be Lipschitz continuous if there exists a constant K such that for all vectors u and v in $\texttt{real}^n$, $\|f(u) - f(v)\| \leq K \|u - v\|$.

In the example component `Car` of figure 6.3, the right-hand side f_x equals v, which when viewed as a function from `real` to `real` is Lipschitz continuous, and the right-hand side f_v equals $(F - kv)/m$, which when viewed as a function from $\texttt{real}^2$ to `real` is Lipschitz continuous. The function $x^{1/3}$ (in the right-hand side of the differential equation we just considered) is *not* Lipschitz continuous since its rate of change grows unboundedly as x approaches 0. The quadratic function x^2 is not Lipschitz continuous as its rate of change grows unboundedly as x increases unboundedly. However, if we know that x ranges over a bounded set D, which is the case in typical modeling of dynamical systems, then the domain of the function x^2 is D, and over this domain, it is Lipschitz continuous.

A classical result in calculus, known as the Cauchy-Lipschitz Theorem, tells us that for the initial value problem, given by $\dot{S} = f(S)$ and $\overline{S}(0) = s_0$, if f is Lipschitz continuous, then the solution signal $\overline{S}$ exists and is unique.

With this motivation, we can require that, in the definition of a continuous-time component, the expression f_x defining the rate of change of each state variable x is a Lipschitz continuous function over the state and input variables. In this case, for a given initial state, assuming that the input signal is continuous, the state signal is uniquely defined. Since the value of an output variable is a function of the state and input variables, in this case, the output signal is also uniquely defined. If the expression h_y defining an output variable is a continuous function, then this output signal is guaranteed to be continuous. Since the output of one component can be connected as an input to another component in a block diagram and can appear in a right-hand side of a differential equation in the definition of that component, to ensure the desired uniqueness and existence of solutions, we can demand that each output expression h_y is also Lipschitz continuous. Components that meet these restrictions, such as the component `Car` of figure 6.3, are said to have Lipschitz-continuous dynamics.

> LIPSCHITZ-CONTINUOUS DYNAMICS
>
> A continuous-time component H with input variables I, output variables O, and state variables S is said to have *Lipschitz-continuous* dynamics if:
>
> 1. for each output variable y, the real-valued expression h_y over $I \cup S$ is a Lipschitz-continuous function; and
> 2. for each state variable x, the real-valued expression f_x over $I \cup S$ is a Lipschitz-continuous function.

It follows that if a continuous-time component H has Lipschitz-continuous dynamics, then for a given initial state and a given continuous input signal $\overline{I}$, the corresponding response of the component as a signal over the state and the output variables exists and is unique.

Example: Helicopter Spin

As another example, let us consider the classical problem of controlling a helicopter so as to keep it from spinning. A helicopter has six degrees of motion, three for position and three for rotation. In our simplified version, let us assume that the helicopter position is fixed and the helicopter remains vertical. Then the only freedom of motion is the angular rotation around the vertical, that is, Z-axis. This rotation is called the *yaw* (see figure 6.5). The friction of the main rotor at the top of the helicopter causes the yaw to change. The tail rotor then needs to apply a torque to counteract this rotational force to keep the helicopter from spinning. In this setting, the helicopter model has a continuous-time input signal T, denoting the torque around the Y-axis. The moment of inertia of the helicopter in this simplified setting can be modeled by a single scalar I. The

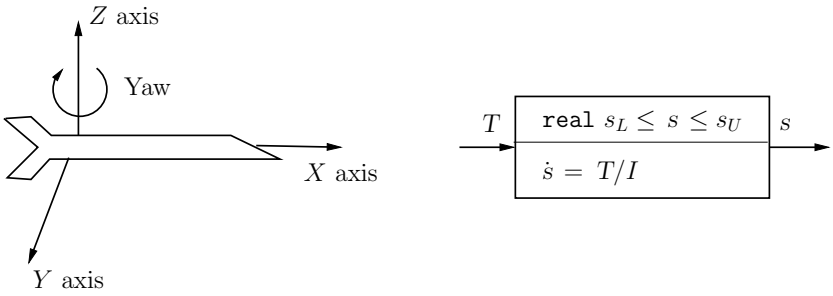

Figure 6.5: Simplified Modeling of Helicopter Motion

output signal of the model is the angular velocity around the vertical axis and is modeled by the spin $s = \dot{\theta}$, where θ gives the yaw. The equation of motion is then given by

$$\ddot{\theta} \;=\; T/I$$

The corresponding continuous-time component is also shown in figure 6.5. There is a single state variable s modeling the spin, a single input variable T corresponding to the torque, and a single output variable, which equals the state s. The initial value of the state is in the interval $[s_L, s_U]$, and its rate of change is given by the expression T/I. Note that this model has Lipschitz-continuous dynamics.

The models expressed as continuous-time components with state variables and differential equations are sometimes called the *state-space representation* of dynamical systems. The dynamics can alternatively be expressed using equations that specify the output signals by integrating over input signals. For instance, for our helicopter model, the value of the output spin at time t equals the sum of its initial value and the integral of the input torque up to time t. This is captured by the *integral equation*

$$\overline{s}(t) \;=\; s_0 \;+\; (1/I) \int_0^t \overline{T}(\tau) d\tau,$$

where s_0 is the initial state. Note that the internal state is implicit in the integral model. Modeling tools such as SIMULINK allow modeling by both state-space representation and integral equations.

Example: Simple Pendulum

Consider a simple pendulum shown in figure 6.6. The pendulum is a rod with a rotational joint at one end and a mass at the other end. For simplicity, we assume that the friction and the weight of the rod are negligible. A motor is placed at the pivot to provide an external torque to control the pendulum.

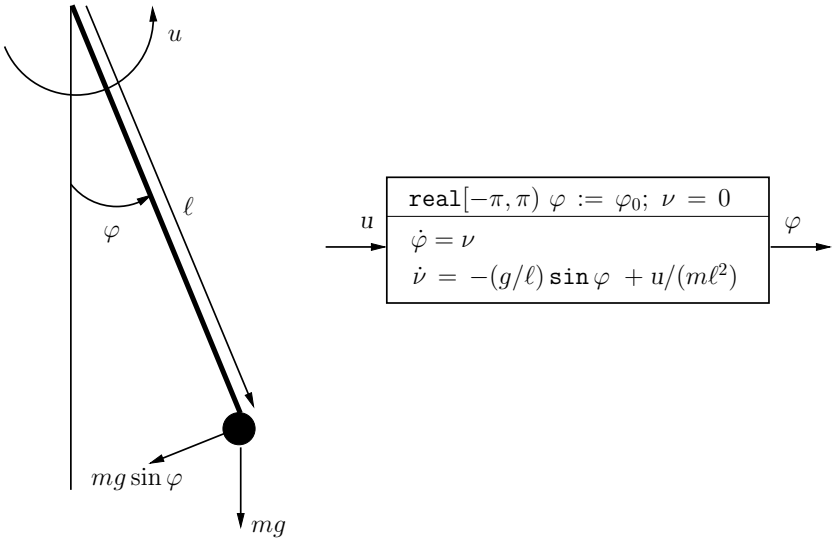

Figure 6.6: Simple Pendulum and Its Model

From Newton's law for rotating objects, the dynamics of the pendulum system is described by the following (nonlinear) differential equation:

$$m\,\ell^2\,\ddot{\varphi} \;=\; u \;-\; m\,g\,\ell\,\sin\varphi$$

where m is the mass, $g = 9.8\,m/s^2$ is the gravitational acceleration, ℓ is the length of the rod, φ is the angle of the pendulum measured counterclockwise from the downward vertical, and u is the external torque applied in the counterclockwise direction around the pivot. Note that the range of values for the angle φ is the interval $[-\pi, \pi)$.

In the continuous-time component modeling the pendulum (see figure 6.6), the second-order differential equation of motion is replaced by a pair of (first-order) differential equations by introducing the state variable ν that captures the angular velocity of the pendulum. The constant φ_0 gives the initial angular position of the pendulum.

To understand the resulting motion, let us assume that the external torque u is set to 0, the length ℓ of the rod is $1m$, and the initial angular displacement is $\pi/4$ radians. Then the motion of the pendulum is described by the equation

$$\ddot{\varphi} \;=\; -9.8\,\sin\varphi;\; \overline{\varphi}(0) \;=\; \pi/4.$$

The resulting oscillatory motion is depicted in figure 6.7, which plots the angular displacement as a function of time.

Exercise 6.1 : Is the dynamics of the continuous-time component of figure 6.6 modeling the pendulum Lipschitz continuous? ■

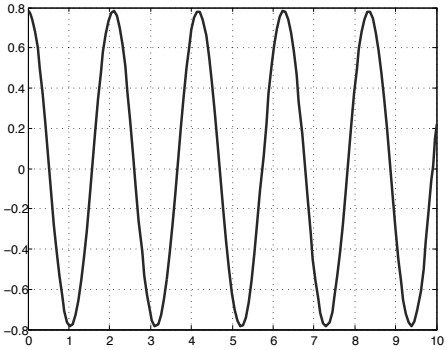

Figure 6.7: Angular Displacement of Simple Pendulum

Exercise 6.2: Consider the model of the simple pendulum from figure 6.6. Suppose the external torque u is set to 0. Analyze the motion of the pendulum if the initial angular displacement is set to $-15\pi/16$ radians, that is, slightly displaced from the vertically upward position. Plot the resulting response signal $\overline{\varphi}$ using MATLAB (use $\ell = 1m$). ■

6.1.2 Models with Disturbance

Let us revisit the model of the motion of a car. The model described in figure 6.3 assumes that the car is moving in a single dimension on a flat road. Suppose we now want to account for the *grade* of the road: on an up hill, the weight of the car works against the force applied by the engine, and on a down hill, the weight of the car adds to this force (see figure 6.8). The cruise controller needs to adjust the force F to keep the net velocity v in the direction along the road constant. We can model the grade of the road by an additional input, denoted θ, that captures the angle of the road with the horizontal: a positive angle indicates an up hill slope, and a negative angle indicates a down hill slope. The weight of the car equals $m\,g$ in the vertically downward direction, where $g = 9.8\,m/s^2$ is the gravitational acceleration. The modified dynamical system model is also shown in figure 6.8. The forces acting on the car in the direction along the road are: F in the forward direction controlled by the engine, $k\,v$ in the backward direction modeling the friction, and $m\,g\,\mathtt{sin}\,\theta$ capturing the gravitational force along the road.

The control design problem for the model of figure 6.8 differs from the corresponding problem for the model of figure 6.3 in a crucial way: the input signal θ modeling the grade of the road is not controlled by the controller and is not known in advance. Such an uncontrolled input is typically called a *disturbance*. The controller should be designed to produce the controlled input signal F so that it works no matter how the input θ varies within a reasonable range of values (for instance, all values in the range $[-\pi/6, +\pi/6]$).

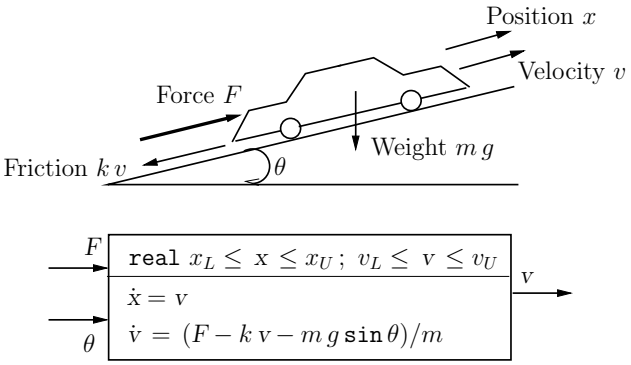

Figure 6.8: Modeling Car Motion on a Graded Road

6.1.3 Composing Components

Continuous-time components can be composed using block diagrams in a way similar to synchronous components. We can define the operations of variable renaming, output hiding, and parallel composition in the same way as in chapter 2. To ensure determinism and well-formed composition, while composing components, we would like to establish absence of cyclic await dependencies. The await dependency of an output variable on input variables is defined the same way as in synchronous components: an output variable y of a continuous-time component H awaits an input variable x if the value of the output y at time t depends on the value of the input x at time t.

For the component `NetHeat` of figure 6.2, the output h_{net} awaits both the input variables h_+ and h_-. For the components of figures 6.3 and 6.8 modeling the motion of a car and the component of figure 6.5 modeling the motion of a helicopter, the output variable at time t equals the value of one of the state variables at time t and, hence, does not await any of the corresponding input variables.

Since the evolution of each output variable y of a continuous-time component is described by an expression h_y over the state and input variables, the await dependencies can be determined by a simple syntactic check: if the set of input variables that occur in the expression h_y is J, then the output y awaits the input variables in J but does not await the remaining input variables.

Exercise 6.3: Consider a continuous-time component H_1 with an input variable u, a state variable x, and an output variable v, as well as a continuous-time component H_2 with the input variable v, a state variable y, and an output variable w. Assuming that both of these components have Lipschitz-continuous dynamics, prove that the parallel composition $H_1 \| H_2$ of the two components also has Lipschitz-continuous dynamics. ■

6.1.4 Stability

In earlier chapters, we have explored safety and liveness requirements for synchronous as well as asynchronous systems. Dynamical control systems also have similar requirements. For example, a safety requirement for a cruise controller can demand that the speed of the car should always be below some maximum speed, and a liveness requirement can demand that the difference between the actual speed and the desired speed should *eventually* be close to zero. A cruise controller, however, has a new kind of requirement, namely, that small perturbations in input values, such as the grade of the road, should not cause disproportionately large changes in the speed of the car. This requirement, which is relevant only for continuous-time systems, is called *stability*. We will first define the notion of Lyapunov stability for dynamical systems and then consider the notion of bounded-input-bounded-output stability for continuous-time components.

Equilibria

To define stability, we first need to understand the notion of an equilibrium state of operation of a dynamical system. For this purpose, we assume that the system is closed: if the original system has inputs, then the stability is analyzed by setting the input signal to a fixed value, say 0, at all times.

Consider a closed continuous-time component whose state S is an n-dimensional vector, with the dynamics given by the Lipschitz-continuous differential equation $\dot{S} = f(S)$. A state s_e of the system is said to be an *equilibrium* state if $f(s_e) = 0$. For an equilibrium state s_e, the constant signal $\overline{S}(t) = s_e$ is a solution to the initial value problem $\dot{S} = f(S)$ and $\overline{S}(0) = s_e$, and since the dynamics is Lipschitz-continuous, this is the only solution. Thus, if the initial state of a system with Lipschitz continuous dynamics is an equilibrium state s_e, then the state of the system is guaranteed to stay at this equilibrium state at all times.

As an example, let us revisit the continuous-time component `Car` of figure 6.3, and let us set the input force F to be the constant signal with value 0. Then the dynamics of the component is described by the equations

$$\dot{x} \;=\; v;\; \dot{v} \;=\; -k\,v/m.$$

A state (x_e, v_e) is an equilibrium state of this system exactly when $v_e = 0$. When the velocity v_e equals 0, the signal that sets the position x always equal to x_e and sets the velocity v always equal to 0 satisfies the above differential equations. In contrast, if the initial velocity v_e is nonzero, the velocity will keep changing at an exponential rate. Thus, for the input signal $\overline{F}(t) = 0$, the system has infinitely many equilibria of the form $(x_e, 0)$.

As another example, consider the pendulum model of figure 6.6, and suppose the input torque u is set to the constant value 0. Then the dynamics of the component is described by the equations

$$\dot{\varphi} \;=\; \nu;\; \dot{\nu} \;=\; -\,g\,\mathtt{sin}\,\varphi/\ell.$$

Recall that the angular displacement φ ranges over the interval $[-\pi, \pi)$, and in this range, $\texttt{sin}\,\varphi$ equals 0 for two values of φ, namely, $\varphi = 0$ and $\varphi = -\pi$. Thus, the system has two equilibria: one where $\varphi = 0$ and $\nu = 0$ (and this corresponds to the pendulum in the vertically downwards position), and one where $\varphi = -\pi$ and $\nu = 0$ (and this corresponds to the pendulum in the vertically inverted position).

Lyapunov Stability

Let us consider a continuous-time component whose state S is an n-dimensional vector, with the dynamics given by the Lipschitz-continuous differential equation $\dot{S} = f(S)$. Consider an equilibrium state s_e of the system. We know that if the initial state of the system is s_e, then the corresponding system evolution is described by the constant signal $\overline{S}_e(t) = s_e$. Now suppose the initial state is perturbed slightly, that is, the initial state is chosen to be s_0 such that the distance $\|s_e - s_0\|$ is small. Consider the signal $\overline{S}_0$ that is the unique response of the system starting from the initial state s_0. If this signal stays *close* to the constant signal $\overline{S}_e$ at all times, then we can conclude that a small perturbation from the equilibrium state causes the system state to stay close to the equilibrium. In such a case, the equilibrium s_e is said to be stable. If, in addition, the state $\overline{S}_0(t)$ *converges* to the equilibrium state s_e as time t advances, then we are guaranteed that after a small perturbation from the equilibrium state, the system state stays close to the equilibrium eventually returning to the equilibrium. When this additional convergence requirement holds, the equilibrium s_e is said to be *asymptotically* stable.

These notions of stability, usually referred to as Lyapunov stability in the literature on dynamical systems, are summarized below.

Lyapunov Stability

Consider a closed continuous-time component H with n state variables S and dynamics given by the equation $\dot{S} = f(S)$, where $f : \texttt{real}^n \mapsto \texttt{real}^n$ is Lipschitz continuous. A state s_e is said to be an equilibrium of the component H if $f(s_e) = 0$. Given an initial state $s_0 \in \texttt{real}^n$, let $\overline{S}_0 : \texttt{time} \mapsto \texttt{real}^n$ be the unique response of the component H from the initial state s_0.

- An equilibrium s_e of H is said to be stable if for every $\epsilon > 0$, there exists a $\delta > 0$ such that for all states s_0, if $\|s_e - s_0\| < \delta$, then $\|\overline{S}_o(t) - s_e\| < \epsilon$ holds at all times $t \geq 0$.
- An equilibrium s_e of H is said to be asymptotically stable if it is stable, and there exists a $\delta > 0$ such that for all states s_0, if $\|s_e - s_0\| < \delta$, then the limit $\texttt{lim}_{t \to \infty}\ \overline{S}_0(t)$ exists and equals s_e.

Let us revisit the continuous-time component `Car` of figure 6.3 again, with the input force F set to the constant value 0. We have already noted that the

state with the position $x = x_e$ and the velocity $v = 0$ is an equilibrium for every choice of x_e. Suppose we perturb this equilibrium; that is, consider the behavior of the component starting at the initial state (x_0, v_0) such that the distance $\|(x_e, 0) - (x_0, v_0)\|$ is small. The car will slow down according to the differential equation $\dot{v} = -k\,v/m$ starting from the initial velocity v_0, with the velocity converging to 0. The position of the car will converge to some value x_f that is a function of the initial position x_0 and the initial velocity v_0: $x_f = x_0 + mv_0/k$. Thus, for the resulting signal, the value $\|(\overline{x}_o(t), \overline{v}_0(t)) - (x_e, 0)\|$ is bounded. Thus, the equilibrium $(x_e, 0)$ is stable. However, it is not asymptotically stable since along this signal, as time advances, the position does not converge to x_e but converges to a different value x_f.

Now, let us consider the pendulum of figure 6.6 with the input signal u set to the constant value 0. We know that the model $\dot{\varphi} = \nu$; $\dot{\nu} = -g\,\mathtt{sin}\,\varphi/\ell$ has two equilibria, namely, $(\varphi = 0, \nu = 0)$ and $(\varphi = -\pi, \nu = 0)$. The latter corresponds to the vertically inverted position and is not stable: if we displace the pendulum slightly from this vertically upward position, the angular velocity ν will keep increasing positively, thereby increasing the displacement φ, pushing the pendulum away from the vertical. In contrast, the equilibrium $(\varphi = 0, \nu = 0)$ corresponding to the vertically downward position is stable. For example, if we set the initial angle to a small positive value, the angular velocity will be negative and will cause the angle to decrease back toward the equilibrium value 0 (see figure 6.7 for a representative behavior). The pendulum will oscillate around the equilibrium position, and if the initial perturbation is (φ_0, ν_0), then the value of $\|(\overline{\varphi}_0(t), \overline{\nu}_0(t)) - (0, 0)\|$ stays bounded, with the bound dependent on the values of φ_0 and ν_0. In this model, this equilibrium is not asymptotically stable: for example, if the initial starting position is ϵ with initial angular velocity 0, then the pendulum will keep swinging forever in the arc from ϵ to $-\epsilon$ around the vertical. In reality, of course, such a pendulum asymptotically converges to the vertical coming to a halt due to the damping effects, which are not captured in our model.

Input-Output Stability

The notion of Lyapunov stability is based on the state-space representation of a dynamical system and concerns the behavior of the system when its state is perturbed from the equilibrium state, with the input signal set to 0. An alternative notion of stability views the dynamical system as a *transformer*, mapping input signals to output signals, and demands that a small change to the input signal should cause only a small change to the output signal. This notion of *input-output stability* is formalized next.

A signal $\overline{x}$ assigning values to the real-valued variable x as a function of time is said to be *bounded* if there exists a constant Δ such that $\|\overline{x}(t)\| \leq \Delta$ for all times t. Here are some typical signals analyzed for their boundedness:

- The constant signal defined by $\overline{x}(t) = a$, for a constant value a, is bounded.

- The linearly increasing signal defined by $\overline{x}(t) = a + bt$, for constants a and $b > 0$, is not bounded.
- The exponentially increasing signal defined by $\overline{x}(t) = a + e^{bt}$, for constants a and $b > 0$, is not bounded.
- The exponentially decaying signal defined by $\overline{x}(t) = a + e^{-bt}$, for constants a and $b > 0$, is bounded.
- The step signal defined by $\overline{x}(t) = a$ for $t < t_0$ and $\overline{x}(t) = b$ for $t \geq t_0$, for constants t_0, a, b, is bounded.
- The sinusoidal signal defined by $\overline{x}(t) = a \cos bt$, for constants a and b, is bounded.

A signal over a set V of variables is bounded if the component of the signal corresponding to each variable x in V is bounded.

In a stable system, when started in the initial state $s_0 = 0$, whenever the input signal is bounded, then the output signal produced by the component in response is also bounded. The bound on the output signal can be different from the bound on the input signal. This particular formalization of stability is known as *Bounded Input Bounded Output* (BIBO) stability.

BIBO Stability

A continuous-time component H with Lipschitz-continuous dynamics with input variables I and output variables O is *Bounded-Input-Bounded-Output stable* if, for every bounded input signal $\overline{I}$, the output signal $\overline{O}$ produced by the component H starting in the initial state $s_0 = 0$, in response to the input signal $\overline{I}$, is also bounded.

Let us consider the helicopter model from figure 6.5:

$$\dot{s} \;=\; T/I$$

and assume that the initial spin is 0. Suppose we apply a constant torque T_0 to the system. Then the rate of change of the spin s is constant and the spin keeps increasing linearly. Thus the system is unstable: the input signal is the bounded constant function $\overline{T}(t) = T_0$, and the corresponding output signal is described by the unbounded function $\overline{s}(t) = (T_0/I)\, t$.

Exercise 6.4: Consider the model of the car moving on a graded road shown in figure 6.8. Suppose the input force F is 0 at all times, and the grade θ of the road is constant at 5 degrees uphill. What are the equilibria for the resulting dynamical system? ■

Exercise 6.5: Consider a two-dimensional dynamical system whose dynamics is given by

$$\dot{s}_1 \;=\; 3\, s_1 + 4\, s_2;\;\; \dot{s}_2 \;=\; 2\, s_1 + s_2.$$

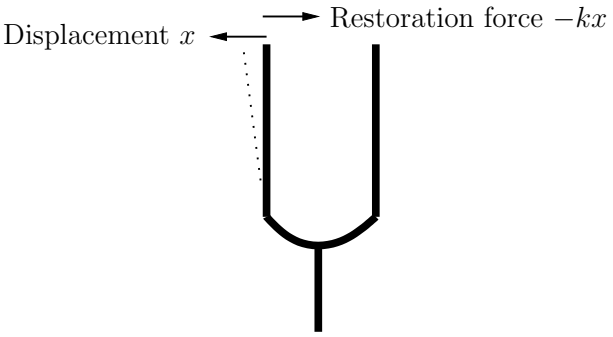

Figure 6.9: Vibrations of a Tuning Fork

Find the equilibria of this system. For each equilibrium, analyze if the equilibrium is asymptotically stable, stable but not asymptotically stable, or unstable. ■

Exercise 6.6: Consider a dynamical system whose dynamics is given by

$$\dot{x} \;=\; x^2 - x.$$

Find the equilibria of this system. For each equilibrium, analyze if the equilibrium is asymptotically stable, stable but not asymptotically stable, or unstable. ■

Exercise 6.7*: A tuning fork consists of a metal tine that can be displaced from its vertical position by striking it with a hammer (see figure 6.9). Once displaced, assuming no friction, the fork vibrates forever, generating a musical tone. Let x denote the displacement of the tine in the horizontal direction. The force that pushes the tine back toward the vertical position is proportional to its displacement at any point in time, and thus the equation of motion is given by

$$m\,\ddot{x} \;=\; -k\,x,$$

where k is a constant depending on the properties of the tuning fork. (a) Design a continuous-time component that captures this dynamics by finding a suitable state-space representation. (b) Assuming initial displacement $\overline{x}(0) = x_0$, find the closed-form solution for the response signal $\overline{x}(t)$ (hint: recall the rules of derivatives of basic trigonometric functions). (c) What are the equilibria of this system? For each equilibrium, analyze if the equilibrium is asymptotically stable, stable but not asymptotically stable, or unstable. ■

6.2 Linear Systems

When the dynamics of a continuous-time component is specified using linear expressions, a number of questions regarding the behavior of the system can be

answered using mathematical analysis.

6.2.1 Linearity

A *linear* expression over a set of variables is formed using the operations of addition and multiplication by a numerical constant. If the variables are $x_1, x_2, \ldots x_n$, then a linear expression has the form $a_1x_1 + a_2x_2 + \cdots + a_nx_n$, where the coefficients $a_1, a_2, \ldots a_n$ are either real, rational, or integer constants. A linear system is a continuous-time component where the expressions used to specify the dynamics of the state and output variables are such linear expressions.

> Linear Component
>
> A continuous-time component H with input variables I, output variables O, and state variables S is said to be a *linear* component if:
>
> 1. for every output variable y, the expression h_y is a linear expression over the variables $I \cup S$; and
> 2. for every state variable x, the expression f_x is a linear expression over the variables $I \cup S$.

In our examples, the components of figures 6.2, 6.3, and 6.5 are linear. The model of a car on a graded road in figure 6.8 with the nonlinear term $m\,g\,\texttt{sin}\,\theta$ in the dynamics is not linear. However, notice that the input θ is not controlled and simply represents disturbance or noise that the controller must handle. As a result, we can replace the input θ by another variable d with the meaning that d captures the value of the expression $m\,g\,\texttt{sin}\,\theta$. Now the dynamics becomes $\dot{v} = (F - k\,v - d)/m$ and is linear. The range of values for the input disturbance $\theta \in [-\pi/6, +\pi/6]$ must be replaced by the range $[m\,g\,\texttt{sin}\,(-\pi/6), m\,g\,\texttt{sin}\,(+\pi/6)]$ for the new variable d. For the model of the simple pendulum in figure 6.6, the dynamics is specified by the differential equation $\ddot{\varphi} = -(g/\ell)\sin\varphi + u/(m\ell^2)$, and thus this component is nonlinear.

A linear expression of n variables defines a function from $\texttt{real}^n$ to $\texttt{real}$, and this function is Lipschitz continuous (see exercise 6.8). As a result, every linear component has Lipschitz-continuous dynamics.

It is worth noting that an expression such as $ax+b$, with $b > 0$, is *not* considered linear, and thus a single-dimensional system with the dynamics $\dot{x} = ax + b$ is not a linear component. In control theory literature, expressions with constants, that is, expressions of the form $a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_{n+1}$, are called *affine* expressions. A particular quantity with affine dynamics is *time* itself as it can be modeled by a variable t with the dynamics $\dot{t} = 1$, and then this variable t can be used in the right-hand sides defining the dynamics of other state variables (for example, the differential equation $\dot{x} = t$ expresses time-dependent evolution of the state variable x). The analysis techniques developed in this chapter focus on linear systems, but some of them can be extended to affine systems also.

Matrix-Based Representation

The conventional form for expressing the dynamics for linear systems uses matrices. Consider a linear component with m input variables $I = \{u_1, \ldots u_m\}$, n state variables $S = \{x_1, \ldots x_n\}$, and k output variables $O = \{y_1, \ldots y_k\}$. In this case, we can view an input as a vector of dimension m, a state as a vector of dimension n, and an output as a vector of dimension k. The dynamics is expressed by four matrices each with real-valued coefficients: a matrix A of dimension $n \times n$, a matrix B of dimension $n \times m$, a matrix C of dimension $k \times n$, and a matrix D of dimension $k \times m$. The dynamics is given by

$$\dot{S} = AS + BI,\ O = CS + DI.$$

That is, for each state variable x_i, the differential equation modeling its rate of change as a function of the state/input variables is given by the linear differential equation:

$$\dot{x_i} \ = \ A_{i,1}\, x_1 + \ \cdots \ + A_{i,n}\, x_n \ + \ B_{i,1}\, u_1 + \ \cdots \ + B_{i,m}\, u_m.$$

For each output variable y_j, its value is defined in terms of the state/input variables by the linear expression:

$$y_j \ = \ C_{j,1}\, x_1 + \ \cdots \ + C_{j,n}\, x_n \ + \ D_{j,1}\, u_1 + \ \cdots \ + D_{j,m}\, u_m.$$

In our example of the car model of figure 6.3, $S = \{x, v\}$, $I = \{F\}$, and $O = \{x\}$. Thus, $n = 2$ and $m = k = 1$. The dynamics can be rewritten as

$$\begin{aligned} \dot{x} &= 0\,x + 1\,v + 0\,F \\ \dot{v} &= 0\,x + (-k/m)\,v + (1/m)\,F \\ v &= 0\,x + 1\,v + 0\,F \end{aligned}$$

The matrices are then given by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -k/m \end{bmatrix};\ \ B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix};\ \ C = [0 \ \ 1];\ \ D = [0].$$

Linear Response

We have defined linearity based on the state-space representation of the component. Alternatively, linearity can be studied based on the properties of the transformation from the space of input signals to the space of output signals induced by the component.

Consider a continuous-time component H with Lipschitz-continuous dynamics with an input variable x and an output variable y. Let us set the initial state of the system to the origin, that is, the state where x equals 0. Given an input signal $\overline{x}$: `time` $\mapsto$ `real`, there is a unique output signal $\overline{y}$: `time` $\mapsto$ `real` corresponding to the execution of the component H on the input signal $\overline{x}$

starting at the origin. Thus, a continuous-time component is a function from the set of input signals to the set of output signals. This transformation is guaranteed to be *linear* for a linear component. Linearity of transformations means the following two properties:

- **Scaling:** If the input signal is scaled by a constant factor, then the output signal also gets scaled by the same factor. Given an input signal $\overline{x}$ and a constant α, let $\alpha\,\overline{x}$ be the input signal whose value at each time t is $\alpha\,\overline{x}(t)$. Then for a linear component H, for all input signals $\overline{x}$ and all scaling factors α, if $\overline{y}$ is the output signal corresponding to the input signal $\overline{x}$, then $\alpha\,\overline{y}$ is the output signal corresponding to the input signal $\alpha\,\overline{x}$.
- **Additivity:** If an input signal can be expressed as a sum of two input signals, then the corresponding output signal is also the sum of the output signals corresponding to the component input signals. That is, if the input signals $\overline{x}$, $\overline{x}_1$, and $\overline{x}_2$ are such that $\overline{x}(t) = \overline{x}_1(t) + \overline{x}_2(t)$ for all times t, and if $\overline{y}$, $\overline{y}_1$, and $\overline{y}_2$ are the output signals produced by the component corresponding to the input signals $\overline{x}$, $\overline{x}_1$, and $\overline{x}_2$, respectively, then it must be the case that $\overline{y}(t) = \overline{y}_1(t) + \overline{y}_2(t)$ for all times t.

In general, the component has multiple input and multiple output variables, and we need to consider signals that are mappings from the time domain to the set of real-valued vectors. In this case, linearity is defined by considering the sum of vectors and scaling of vectors. For two signals $\overline{V}$ and $\overline{V}'$ over a set V of variables and constants $\alpha, \beta \in \texttt{real}$, the signal $\alpha\,\overline{V} + \beta\,\overline{V}'$ is defined to be the signal that assigns, for every time t, the value $\alpha\,\overline{V}(t)(x) + \beta\,\overline{V}'(t)(x)$ to each variable $x \in V$. Linearity of transformations is now captured by the following theorem:

Theorem 6.1 [Linearity of Input-Output Transformation] *Let H be a linear component with input variables I and output variables O. For all input signals $\overline{I}_1$ and $\overline{I}_2$ and constants $\alpha, \beta \in \texttt{real}$, if the output signals generated by the component H from the initial state 0 in response to the input signals $\overline{I}_1$ and $\overline{I}_2$ are $\overline{O}_1$ and $\overline{O}_2$, respectively, then the output signal generated by the component H from the initial state 0 in response to the input signal $\alpha\,\overline{I}_1 + \beta\,\overline{I}_2$ is $\alpha\,\overline{O}_1 + \beta\,\overline{O}_2$.*

Proof. Suppose the dynamics of a linear component H is given by the equations $\dot{S} = AS + BI$ and $O = CS + DI$, where S is the state vector, I is the input vector, and O is the output vector. Suppose the initial state is 0.

For a given input signal $\overline{I}_1$, suppose the state response signal is $\overline{S}_1$ and the output response signal is $\overline{O}_1$. Then we know that the following conditions must hold: $\overline{S}_1(0) = 0$, and for all times t, $(d/dt)\overline{S}_1(t) = A\,\overline{S}_1(t) + B\,\overline{I}_1(t)$ and $\overline{O}_1(t) = C\,\overline{S}_1(t) + D\,\overline{I}_1(t)$.

Similarly, for another input signal $\overline{I}_1$, suppose the state response signal is $\overline{S}_2$ and the output response signal is $\overline{O}_2$. Then $\overline{S}_2(0) = 0$, and for all times t, $(d/dt)\overline{S}_2(t) = A\,\overline{S}_2(t) + B\,\overline{I}_2(t)$ and $\overline{O}_2(t) = C\,\overline{S}_2(t) + D\,\overline{I}_2(t)$.

Given constants $\alpha, \beta \in \texttt{real}$, define the signals $\overline{I} = \alpha\,\overline{I}_1 + \beta\,\overline{I}_2$ and $\overline{S} = \alpha\,\overline{S}_1 + \beta\,\overline{S}_2$ and $\overline{O} = \alpha\,\overline{O}_1 + \beta\,\overline{O}_2$. From basic properties of linear arithmetic and differential calculus, it follows that the following must hold: $\overline{S}(0) = 0$, and for all times t, $(d/dt)\overline{S}(t) = A\,\overline{S}(t) + B\,\overline{I}(t)$ and $\overline{O}(t) = C\,\overline{S}(t) + D\,\overline{I}(t)$. Thus, for the input signal $\overline{I}$, the state response of the component H must be the signal $\overline{S}$ and the output response must be the signal $\overline{O}$. ■

Exercise 6.8: Prove that a linear expression $e = a_1x_1 + a_2x_2 + \cdots + a_nx_n$, viewed as a function from $\texttt{real}^n$ to $\texttt{real}$, is Lipschitz continuous. ■

Exercise 6.9: Recall that the component corresponding to the motion of a car on a graded road (see figure 6.8) can be viewed as a component with two inputs, force F and disturbance d, with the dynamics given by the differential equation $\dot{v} = (F - kv - d)/m$. What are the matrices A, B, C, and D for the standard matrix-based representation of the resulting linear system? ■

Exercise 6.10: Let us revisit the nonlinear model of the pendulum of figure 6.6. A classical approach to designing controllers for nonlinear systems is to linearize the model about an operating point, design a controller for that linearized model, and use it for the original system. For the pendulum example, the operating point of interest is $\varphi = 0$ corresponding to the vertical position of the pendulum. Using the fact that $\sin\varphi \approx \varphi$ for small values of φ, build a corresponding linear component model of the pendulum. ■

Exercise 6.11: Consider a closed linear component H. Prove that the output response of the system is a linear function of the initial state. That is, suppose that $\overline{O}_0$ is the output response of the system starting from the initial state s_0 and $\overline{O}_1$ is the response signal of the system starting from the initial state s_1, and $\alpha, \beta \in \texttt{real}$ are constants. Then prove that the output response of the system starting from the initial state $\alpha\,s_0 + \beta\,s_1$ is the signal $\alpha\,\overline{O}_0 + \beta\,\overline{O}_1$. ■

6.2.2 Solutions of Linear Differential Equations

For linear systems, a number of analysis techniques are available to understand how the output signal is related to the input signal. Let us first consider the linear differential equation $\dot{S} = AS$ and suppose the initial state is given by the vector s_0. To solve this equation, we can construct a sequence of signals $\overline{S}_0, \overline{S}_1, \cdots$ that approximate the desired solution in the following manner. Let $\overline{S}_0$ be the constant signal defined by $\overline{S}_0(t) = s_0$ for all times t. For each $m > 0$, define:

$$\overline{S}_m(t) \;=\; s_0 \;+\; \int_0^t A\,\overline{S}_{m-1}(\tau)\,d\tau.$$

We can use mathematical calculations based on solving integrals to find closed forms for these signals:

$$\overline{S}_1(t) \quad = \quad s_0 + \int_0^t A\,s_0\,d\tau$$

$$\begin{aligned} &= s_0 + A\,t\,s_0 \\ &= [\mathbf{I} + A\,t]\,s_0. \end{aligned}$$

In these calculations, A is an $n \times n$ matrix, s_0 is an $n \times 1$ vector, and t is a scalar. The identity matrix is denoted $\mathbf{I}$ (that is, $\mathbf{I}_{i,j}$ equals 1 if $i = j$ and 0 otherwise). Repeating the calculation for one more step gives:

$$\begin{aligned} \overline{S}_2(t) &= s_0 + \int_0^t A\,([\mathbf{I} + A\,\tau]\,s_0)\,d\tau \\ &= s_0 + A\,t\,s_0 + A^2\,(t^2/2)\,s_0 \\ &= [\mathbf{I} + \sum_{j=1}^{2} A^j\,t^j/j\,!]\,s_0 \end{aligned}$$

After repeating the pattern, we obtain

$$\overline{S}_m(t) = [\mathbf{I} + \sum_{j=1}^{m} A^j\,t^j/j\,!]\,s_0.$$

The sequence of functions $\overline{S}_0, \overline{S}_1, \overline{S}_2, \ldots$ converges to the unique solution of the differential equation given by

$$\overline{S}(t) = [\mathbf{I} + \sum_{j=1}^{\infty} A^j\,t^j/j\,!]\,s_0.$$

Recall that, for a real number a, the quantity e^a is defined as:

$$e^a = 1 + \sum_{j=1}^{\infty} a^j/j!\ .$$

Similarly, the *matrix exponential* e^A for a matrix A is defined by the equation:

$$e^A = \mathbf{I} + \sum_{j=1}^{\infty} A^j/j\,!$$

With this notation, the solution of the differential equation $\dot{S} = AS$, with the initial state s_0, is given by

$$\overline{S}(t) = e^{At}\ s_0.$$

A similar analysis can be performed for the model of a linear component with inputs. Consider the dynamics $\dot{S} = AS + BI$. Suppose the initial state is given by the vector s_0. Given an input signal $\overline{I}$, the resulting state signal $\overline{S}$ is given by the equation:

$$\overline{S}(t) = e^{At}s_0 + \int_0^t e^{A(t-\tau)}\,B\,\overline{I}(\tau)\,d\tau.$$

The response of the system to a given input signal can be computed using this equation: the output value at time t equals $C\,\overline{S}(t) + D\,\overline{I}(t)$.

The Matrix Exponential

Let us examine the definition of the matrix exponential operation:

$$e^A \;=\; \mathbf{I} \;+\; \sum_{j=1}^{\infty} A^j/j\,!$$

For a square $n \times n$ matrix A, observe that each term $A^j/j\,!$ is a square $n \times n$ matrix, and so is the exponential e^A. If we calculate the matrix e^A, the matrix e^{At} appearing in the solution $\overline{S}(t)$ of the differential equation is easily obtained by multiplying each matrix entry by t.

A number of mathematical tools exist to compute the quantity e^A depending on the structural properties of the matrix A. As an illustrative example, suppose the matrix A is a *diagonal* matrix (that is, each entry $A_{i,j}$, for $i \neq j$, equals 0). Let us denote the ith diagonal entry of the diagonal matrix A by a_i, and the diagonal matrix is denoted $\mathbf{D}(a_1, a_2, \ldots\ a_n)$. In this case, observe that, for every j, the matrix A^j is also a diagonal matrix whose ith diagonal entry is given by a_i^j. Then e^A is also a diagonal matrix and its ith diagonal entry is the sum $1 + \sum_{j=1}^{\infty}\ a_i^j/j!$, which equals e^{a_i}. Thus,

$$e^{\mathbf{D}(a_1,a_2,\cdots a_n)} \;=\; \mathbf{D}(\, e^{a_1}, e^{a_2}, \cdots\ e^{a_n}).$$

As another example, consider the two-dimensional matrix

$$A = \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix}.$$

Observe that for this matrix A, A^2 equals the matrix $\mathbf{0}$ with all entries 0. As a result,

$$e^A \;=\; \mathbf{I} + A \;=\; \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}.$$

In general, whenever $A^k = \mathbf{0}$ for some k, only the first k terms in the infinite series defining the matrix exponential e^A are nonzero, and thus one can obtain an explicit matrix representation of e^A.

Eigenvalues and Eigenvectors

Let us consider a standard tool for calculating the matrix exponential using the *similarity* transformation. This transformation is based on computing the eigenvalues and eigenvectors of a matrix.

For an $(n \times n)$-matrix A, if the equation $A\,x \;=\; \lambda\,x$ holds, for a scalar λ and a non-zero vector x of dimension n, then the value λ is called an *eigenvalue* of the matrix A, and the vector x is called an *eigenvector* of the matrix A corresponding to the eigenvalue λ. An $(n \times n)$-matrix A has at most n distinct eigenvalues, and these correspond to the *characteristic equation* of A given by

$$\mathtt{det}(\, A \;-\; \lambda\,\mathbf{I}) \;=\; 0,$$

where `det` represents the *determinant* of a matrix. Note that a value λ_i is an eigenvalue of the matrix exactly when the term $(\lambda - \lambda_i)$ is a factor of the characteristic polynomial $\texttt{det}(A - \lambda \mathbf{I})$.

As an example, consider the two-dimensional matrix

$$A_1 = \begin{bmatrix} 4 & 6 \\ 1 & 3 \end{bmatrix}.$$

The eigenvalues of this matrix A_1 are the solutions of the equation:

$$\texttt{det}(\begin{bmatrix} 4-\lambda & 6 \\ 1 & 3-\lambda \end{bmatrix}) = 0.$$

Recall that the determinant of a (2×2)-matrix A is given by the expression $A_{1,1}\,A_{2,2} - A_{1,2}\,A_{2,1}$. Thus, the desired eigenvalues are the roots of the polynomial

$$(4-\lambda)\,(3-\lambda) - 6 \;=\; \lambda^2 - 7\lambda + 6 \;=\; (\lambda - 6)\,(\lambda - 1).$$

Thus, the eigenvalues of the matrix A_1 are $\lambda_1 = 6$ and $\lambda_2 = 1$. To obtain the eigenvector x_1 corresponding to the eigenvalue 6, we need to solve the equation $A_1\,x_1 = 6\,x_1$. If the entries of the vector x_1 are x_{11} and x_{12}, then we get the system of linear equations:

$$\begin{bmatrix} 4 & 6 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} \;=\; 6 \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}.$$

This corresponds to

$$4\,x_{11} + 6\,x_{12} \;=\; 6x_{11};\quad x_{11} + 3\,x_{12} \;=\; 6\,x_{12}.$$

These equation are satisfied whenever $x_{11} = 3x_{12}$, and every vector of this form is an eigenvector corresponding to the eigenvalue 6. In particular, let us set $x_1 = [3\ 1]^T$ (note: x_1 is a column vector with two rows and one column and, thus, is the *transpose* of the row vector $[3\ 1]$ and denoted $[3\ 1]^T$). The eigenvector corresponding to the eigenvalue $\lambda_2 = 1$ is obtained by a similar analysis, and in particular, $x_2 = [2\ -1]^T$ is a corresponding eigenvector. Note that the two vectors x_1 and x_2 are *linearly independent*. This is no coincidence: if the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$ are all distinct and the vectors $x_1, x_2 \ldots x_n$ are eigenvectors corresponding to these eigenvalues, respectively, then these n vectors are guaranteed to be linearly independent.

Note that if the matrix A is a diagonal matrix with diagonal entries a_i, then $A - \lambda\mathbf{I}$ is also a diagonal matrix with entries $a_i - \lambda$. The characteristic polynomial of the matrix A then is the product of the terms $(a_i - \lambda)$. Furthermore, for each i, the vector x_i with 1 in the ith entry and 0 s everywhere else satisfies the equation $A\,x_i = a_i\,x_i$ and is thus an eigenvector corresponding to the eigenvalue a_i. For example, for the three-dimensional diagonal matrix $\mathbf{D}(1, 2, 1)$, the characteristic polynomial is $(1 - \lambda)^2(2 - \lambda)$. As a result, this three-dimensional matrix has

two eigenvalues, namely, 1 and 2, and in this case, the (algebraic) multiplicity of the eigenvalue 1 is 2 (since $(\lambda-1)^2$ is a factor of the characteristic polynomial). The vectors $x_1 = [1\ 0\ 0]^T$, $x_2 = [0\ 1\ 0]^T$, and $x_3 = [0\ 0\ 1]^T$ are eigenvectors, corresponding to the eigenvalues 1, 2, and 1, respectively. In this case, all these eigenvectors are linearly independent.

In the examples so far, the matrix has n linearly independent eigenvectors with real-valued entries. However, this need not be the case as indicated by the following two examples.

Consider the following two-dimensional (upper triangular) matrix:

$$A_2 \ = \ \left[\begin{array}{cc} 1 & 2 \\ 0 & 1 \end{array} \right].$$

The characteristic polynomial for this matrix is $(1-\lambda)^2$. Thus, there is only one eigenvalue, namely, 1. The eigenvectors of this matrix are of the form $[a\ 0]^T$ for an arbitrary constant a. All these eigenvectors are linearly dependent on one another, that is, the two-dimensional matrix A_2 has only one linearly independent eigenvector.

Consider the following two-dimensional matrix:

$$A_3 \ = \ \left[\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array} \right].$$

The characteristic polynomial for this matrix is $\lambda^2 + 1$. However, the equation $\lambda^2 + 1 = 0$ has no real-valued solutions. In such a case, we want to interpret matrices as linear transformers over the field of *complex numbers*. With this interpretation, the matrix A_3 has two eigenvalues j and $-j$, both of which are imaginary numbers (note: the imaginary number j is the square-root of -1, and every complex number is of the form $a + bj$, where a, b are real numbers). In this case, the eigenvector corresponding to the eigenvalue j is obtained by solving the equation $A_3\, x \ = \ j\, x$ and is of the form $[1\ \ j]^T$.

If $\lambda_1, \ldots \lambda_p$ are all the (complex) eigenvalues of the matrix A, then

$$\mathtt{det}(\, A \ - \ \lambda\, \mathbf{I}) \ = \ (\lambda - \lambda_1)^{n_1} \ \cdots \ (\lambda - \lambda_p)^{n_p},$$

where n_j is the algebraic multiplicity of the eigenvalue λ_j and $n_1 + \cdots + n_p$ equals n. Note that if a complex number $a + bj$, with $b \neq 0$ is an eigenvalue of the matrix A, then its conjugate, that is, the complex number $a - bj$, must also be an eigenvalue of the matrix A.

Similarity Transformations

Consider the dynamical system H given by

$$\dot{S} \ = \ A\, S;\ \overline{S}(0) \ = \ s_0,$$

where S is the n-dimensional state vector, A is an $(n \times n)$-matrix, and s_0 is the initial state. Suppose P is an *invertible* n-dimensional square matrix with real-valued entries, and P^{-1} is its inverse. Thus, $P^{-1}P = P\,P^{-1} = \mathbf{I}$. Consider the vector S' defined by $S' = P^{-1}\,S$. This defines a linear transformation of the state. Note that the relation $S = P\,S'$ also holds. Let us denote the matrix $P^{-1}A\,P$ by J. When such a relationship holds, the matrices A and J are said to be *similar*.

Now, based on the original dynamical system H with state S, let us specify the dynamical system H' with state S':

$$\dot{S}' \;=\; (d/dt)(P^{-1}\,S) \;=\; P^{-1}\,\dot{S} \;=\; P^{-1}\,A\,S \;=\; P^{-1}A\,P\,S' \;=\; J\,S'.$$

The initial state of this transformed linear system H' is given by

$$\overline{S}'(0) \;=\; P^{-1}\,\overline{S}(0) \;=\; P^{-1}s_0.$$

Such a transformation of the linear system H with state S and dynamics matrix A to obtain another linear system H' with state S' and dynamics matrix J is called a *similarity transformation* (since the matrices A and J are similar). Note that the solution of the system H' is given by:

$$\overline{S}'(t) \;=\; e^{Jt}\,\overline{S}'(0).$$

This implies that the solution of the original system H is given by:

$$\overline{S}(t) \;=\; P\,e^{Jt}\,P^{-1}\,s_0.$$

If the matrix J has properties that make the computation of the matrix exponential e^{Jt} easier, then this can be used to compute the response $\overline{S}(t)$ of the system H.

Suppose the matrix A has n linearly independent eigenvectors $x_1, x_2, \ldots x_n$ with real-valued entries, and let λ_i be the eigenvalue corresponding to the eigenvector x_i. Let us then choose the similarity matrix P to be the matrix whose columns are these n eigenvectors:

$$P \;=\; [\,x_1 \;\; x_2 \;\; \cdots \;\; x_n\,].$$

Since the columns of P are linearly independent, its rank is n, and it is invertible. Note that the ith column of the matrix P is the eigenvector x_i. From the definition of matrix multiplication, the ith column of the matrix product $A\,P$ is the vector $A\,x_i$. Since x_i is an eigenvector corresponding to the eigenvalue λ_i, it follows that the ith column of the matrix product $A\,P$ is the vector $\lambda_i\,x_i$. Hence, the ith column of the matrix product $P^{-1}AP$ is the vector $P^{-1}\lambda_i x_i$ or $\lambda_i P^{-1}x_i$. Recall that the product $P^{-1}P$ is the identity matrix, and the ith column of the product $P^{-1}P$ equals the vector $P^{-1}x_i$ (since x_i is the ith column of the matrix P). It follows that $J = P^{-1}AP$ is the diagonal matrix $\mathbf{D}(\lambda_1, \lambda_2, \ldots \lambda_n)$. It

follows that the matrix exponential e^{Jt} is also a diagonal matrix, and its ith diagonal entry is the scalar $e^{\lambda_i t}$.

The method to compute the response signal of a linear system based on similarity transformation using linearly independent eigenvectors is called *diagonalization* and is summarized in the following theorem:

Theorem 6.2 [Linear System Response by Diagonalization] *Consider an n-dimensional linear system with dynamics given by the differential equation $\dot{S} = A\,S$ with initial state s_0. Suppose the matrix A has n linearly independent real-valued eigenvectors $x_1, x_2, \ldots x_n$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$. Let P be the matrix $[x_1\ x_2\ \cdots\ x_n]$, and let P^{-1} be its inverse. Then the execution of the system is given by the state signal:*

$$\overline{S}(t) \;=\; P\ \mathbf{D}(e^{\lambda_1 t}, e^{\lambda_2 t}, \cdots\ e^{\lambda_n t})\ P^{-1}\ s_0.$$

■

To illustrate this method, let us consider the two-dimensional dynamical system H given by

$$\dot{s}_1 \;=\; 4\,s_1 \,+\, 6\,s_2;\ \dot{s}_2 \;=\; s_1 \,+\, 3\,s_2.$$

The matrix for the dynamics is the matrix A_1 that we used in illustrating the computation of eigenvalues and eigenvectors. As noted earlier, $x_1 = [3\quad 1]^T$ is an eigenvector corresponding to the eigenvalue 6, and $x_2 = [2\quad -1]^T$ is an eigenvector corresponding to the eigenvalue 1. Let us choose the transformation matrix as:

$$P \;=\; [\,x_1\ \ x_2\,] \;=\; \begin{bmatrix} 3 & 2 \\ 1 & -1 \end{bmatrix}.$$

We now need to compute the inverse P^{-1} of this matrix P. This can be done, for instance, by viewing the entries in the desired matrix P^{-1} as unknowns and setting up a system of simultaneous linear equations given by $P\,P^{-1} = \mathbf{I}$:

$$\begin{bmatrix} 3 & 2 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} \;=\; \begin{bmatrix} 3a+2b & 3c+2d \\ a-b & c-d \end{bmatrix} \;=\; \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Solving these equations gives:

$$P^{-1} \;=\; \begin{bmatrix} 1/5 & 2/5 \\ 1/5 & -3/5 \end{bmatrix}.$$

Verify that indeed $P\,P^{-1} = P^{-1}P = \mathbf{I}$. Furthermore, verify that $P^{-1}A_1P$ is the diagonal matrix $\mathbf{D}(6,1)$. What this means is that if we consider the linear system H' with state variables s'_1 and s'_2 defined by

$$s'_1 \;=\; (s_1 + 2s_2)/5;\quad s'_2 \;=\; (s_1 - 3s_2)/5,$$

it has a simpler dynamics given by $\dot{s}'_1 = 6s'_1$ and $\dot{s}'_2 = s'_2$ and thus is easier to analyze. Putting all the pieces together, starting in the initial state s_0, the

state of the system H at time t is described by $P\,\mathbf{D}(e^{6t}, e^t)\,P^{-1}s_0$. If the initial state vector s_0 is $[s_{01}\ s_{02}]^T$, then by calculating the matrix products, we get a closed-form solution for the state of the system H at time t:

$$\begin{aligned}\overline{S}_1(t) &= [(3e^{6t} + 2e^t)s_{01} \ + \ 6(e^{6t} - e^t)s_{02}]/5\\ \overline{S}_2(t) &= [(e^{6t} - e^t)s_{01} \ + \ (2e^{6t} + 3e^t)s_{02}]/5.\end{aligned}$$

As discussed earlier, the matrix A may not have n independent eigenvectors. In this case, it is possible to choose the similarity transformation matrix P in such a way that the matrix $J = P^{-1}AP$ is in *Jordan canonical form*. This is a special form of matrix that is *almost* diagonal, and it is possible to get an explicit representation of the exponential matrix e^{Jt}.

Exercise 6.12: Consider a single-dimensional linear component with one input with dynamics given by $\dot{s} = a\,s \ + \ b\,u$. Suppose we set the input signal to be the constant signal $\overline{u}(t) = c$ for a constant value c. Find a closed-form formula for the system response $\overline{s}(t)$ starting from the initial state s_0 corresponding to this input signal (hint: the integral $\int_0^t e^{-a\tau}\,d\tau$ evaluates to $(1 - e^{-at})/a$). ■

Exercise 6.13: Consider the two-dimensional dynamical system given by

$$\dot{s}_1 \ = \ -s_1 + 2s_2;\ \dot{s}_2 \ = \ s_2.$$

Compute the closed form description of the state signal $\overline{S}(t)$ given the initial state vector s_0 using the method of similarity transformation. ■

Exercise 6.14: Consider the two-dimensional dynamical system given by

$$\dot{s}_1 \ = \ s_2;\ \dot{s}_2 \ = \ -2s_1 - 3s_2.$$

Compute the closed-form description of the state signal $\overline{S}(t)$ given the initial state vector s_0 using the method of similarity transformation. ■

Exercise 6.15: Consider the three-dimensional dynamical system given by

$$\dot{s}_1 \ = \ 3s_1 + 4s_2;\ \dot{s}_2 \ = \ 2s_2;\ \dot{s}_3 \ = 4s_1 + 9s_3.$$

Compute the closed-form description of the state signal $\overline{S}(t)$ given the initial state vector s_0 using the method of similarity transformation. Note that the determinant of a 3×3 matrix A is given by the formula

$$a_{11}(a_{22}a_{33} - a_{23}a_{32}) \ + \ a_{12}(a_{23}a_{31} - a_{21}a_{33}) \ + \ a_{13}(a_{21}a_{32} - a_{22}a_{31}),$$

where a_{ij} denotes the entry of the matrix in ith row and jth column. ■

6.2.3 Stability

Consider the n-dimensional linear system H given by $\dot{S} = AS$. The response of the system starting from the initial state s_0 is described by the signal $\overline{S}(t) = e^{At}s_0$. Our goal is to develop analytical methods to determine stability of the equilibria of this system.

A state s_e is an equilibrium of the system H if the condition $As_e = 0$ holds. To compute the equilibria of the system H, we can view the n elements of the vector s_e as the unknowns and solve the system of n linear equations corresponding to the condition $As_e = 0$. Observe that the state 0, which assigns 0 to all the state variables, is an equilibrium of the system H. If the matrix A is invertible (that is, if the rank of the matrix A is n), then the equation $As_e = 0$ has a unique solution, and 0 is the only equilibrium. If a nonzero state s_e is an equilibrium of the system H, then we can consider a transformed linear system H' with the state vector S' given by $S' = S - s_e$. The dynamics of this transformed system H' is given $\dot{S}' = AS'$, and at each time t, the equation $\overline{S}'(t) = \overline{S}(t) - s_e$ holds. Note that the state 0 is an equilibrium of the system H'. Thus, the behavior of this transformed system around its equilibrium 0 corresponds exactly to the behavior of the original system around its equilibrium s_e. Thus, if we know how to analyze whether the equilibrium 0 is stable, the same analysis technique can be used to determine whether an arbitrary equilibrium is stable: the stability properties of the equilibrium s_e of the system H are exactly the same as the corresponding properties of the equilibrium 0 of the system H'.

For the remainder of this section, we abbreviate "the state 0 of the linear system H is a stable equilibrium" by "the linear system H is stable" and "the state 0 of the linear system H is an asymptotically stable equilibrium" by "the linear system H is asymptotically stable."

Single-Dimensional System

By definition, the linear system H is stable if, for every $\epsilon > 0$, there exists a $\delta > 0$ such that for all state s_0, if $\|s_0\| < \delta$, then $\|e^{At}s_0\| < \epsilon$ for all times t. The system is asymptotically stable if, in addition, there exists a $\delta > 0$ such that for all states s_0, if $\|s_0\| < \delta$, then the vector $e^{At}s_0$ converges to 0 as t increases. In the latter case, the values of δ for which the condition "for all states s_0, if $\|s_0\| < \delta$, then the signal $e^{At}s_0$ converges to 0" holds is called the *region of attraction*.

First, let us focus on linear systems with dimension 1. Then the dynamics is given by $\dot{x} = ax$. If x_0 denotes the initial state, then the state at time t equals $e^{at}x_0$. Depending on the sign of a, we have three cases:

- If the coefficient a is negative, then the magnitude of the value $e^{at}x_0$ decreases with increasing t and is bounded by the magnitude of the initial value x_0. Thus, the system is stable. Also, observe that no matter what the initial value is, the value of x decays exponentially and will become 0

in the limit. In this case, the system is asymptotically stable, and in fact, the region of attraction includes all the states.

- If the coefficient a is 0, then the value of x stays equal to its initial value x_0. Thus, the magnitude of the state does not change with time, and the system is stable. However, the signal *does not converge* to 0, and thus the system is not asymptotically stable.

- If the coefficient a is positive, then observe that the quantity e^{at} grows exponentially with increasing t. As a result, the magnitude of the value of x increases exponentially and grows in an unbounded manner, and the system is not stable.

Thus, the stability of a one-dimensional linear system depends on the sign of the coefficient of the term capturing dependence of the rate of change on the state: the system is unstable if $a > 0$, stable if $a \leq 0$, and asymptotically stable if $a < 0$.

Diagonal State Dynamics Matrix

Now suppose that the matrix A is an n-dimensional diagonal matrix and equals $\mathbf{D}(a_1, a_2, \ldots a_n)$. In this case, we know that the matrix exponential e^{At} is also a diagonal matrix. If the initial state of the system is given by the vector $[s_{01}\ s_{02}\ \cdots\ s_{0n}]^T$, then the state $\overline{S}(t)$ at time t is given by:

$$\overline{S}(t) \ = \ [\ e^{a_1 t} s_{01} \ \ e^{a_2 t} s_{02} \ \ \cdots \ \ e^{a_n t} s_{0n}\ ]^T.$$

Observe that, for each i, with increasing t, the quantity $e^{a_i t}$ increases exponentially if $a_i > 0$, stays unchanged if $a_i = 0$, and decreases exponentially if $a_i < 0$. The case analysis for a single-dimensional system now generalizes naturally:

- If none of the coefficients a_i s is positive, then for every initial state s_0, for the resulting signal, $\|\overline{S}(t)\| \leq \|s_0\|$ holds at all times, and thus the system is stable.

- If all the coefficients a_i s are negative, then in addition to the stability as in the case above, independent of the initial state s_0, for the resulting signal, $\|\overline{S}(t)\|$ converges to the equilibrium 0, and thus the system is asymptotically stable. The region of attraction consists of the entire state space $\texttt{real}^n$.

- Suppose there exists an index i such that the coefficient a_i is positive. Then if we choose the initial state s_0 such that s_{0i} is positive and $s_{0j} = 0$ for all $j \neq i$, then no matter how small the value s_{0i} is, the ith component of the response signal $\overline{S}(t)$ given by $e^{a_i t} s_{0i}$ will grow unboundedly with increasing t. In this case, the system is unstable.

In summary, when $A = \mathbf{D}(a_1, a_2, \ldots a_n)$, the system is unstable if $a_i > 0$ for some i, stable if $a_i \leq 0$ for all i, and asymptotically stable if $a_i < 0$ for all i.

Diagonalizable State Dynamics Matrix

Now recall the similarity transformation that we used to diagonalize a matrix and compute the matrix exponential. Given a linear system H with the dynamics $\dot{S} = A\,S$, we choose a suitable invertible matrix P and consider the transformed linear system H' with the dynamics $\dot{S}' = J\,S'$, where $J = P^{-1}A\,P$. The state vector S' of H' is obtained from the state vector S of H by linear transformation: $S' = P^{-1}S$. Observe that a linear transformation of a signal preserves the properties of being bounded in magnitude and convergence to 0. This implies that the similarity transformation preserves the stability properties of the equilibrium. This is captured by the following proposition:

Proposition 6.1 [Stability Preservation by Similarity Transformation] *Given an n-dimensional linear system H with state vector S and dynamics $\dot{S} = A\,S$, consider another n-dimensional linear system H' with state vector S' and dynamics $\dot{S}' = J\,S'$, where $J = P^{-1}A\,P$ for some invertible matrix P. Then the system H is stable if and only if the system H' is stable, and the system H is asymptotically stable if and only if the system H' is asymptotically stable.* ■

If we can choose the similarity transformation matrix P so that the matrix J is diagonal, then we can conclude that the system H is unstable if some diagonal entry of J is positive, stable if all entries of J are nonpositive, and asymptotically stable if all diagonal entries of J are negative.

Recall that if all eigenvalues of the matrix A are real valued and distinct, then we can choose the matrix P using the corresponding eigenvectors and the diagonal entries of J correspond to the eigenvalues of A. As a result, if the matrix A has n distinct real-valued eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$, then the equilibrium 0 is unstable if $\lambda_i > 0$ for some i, asymptotically stable if $\lambda_i < 0$ for all i, and stable but not asymptotically stable if $\lambda_i \leq 0$ for all i and $\lambda_j = 0$ for some j.

Let us revisit the model of the car in figure 6.3. As noted earlier, the state dynamics matrix for this system is

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -k/m \end{bmatrix}.$$

The eigenvalues of this matrix are the real numbers 0 and $-k/m$. We can immediately conclude that the system is stable but not asymptotically stable, which coincides with the analysis following the definition of Lyapunov stability in section 6.1.4.

In general, the eigenvalues of a matrix can be complex numbers. It is possible to generalize the above case analysis for real-valued eigenvalues to this case. The system is asymptotically stable exactly when all the eigenvalues have negative real parts.

Theorem 6.3 [Stability Test for Linear Dynamics] *The linear system given by $\dot{S} = A\,S$ is asymptotically stable if and only if every eigenvalue of the matrix A has a negative real part.* ■

If some eigenvalue of the state-dynamics matrix A has a positive real part, then the system is unstable. If some eigenvalues have negative real parts and the remaining are purely imaginary numbers (that is, with real part equal to 0), then the system is not asymptotically stable, and its stability depends on the structure of the Jordan blocks corresponding to each such imaginary eigenvalue.

BIBO Stability

Now let us turn our attention to components with inputs. Consider the system given by

$$\dot{S} = AS + BI;\ O = CS + DI.$$

To check whether the system is BIBO-stable we set the initial state $s_0 = 0$, and consider the behavior of the system for a bounded input signal $\bar{I}$. We note a sufficient condition for checking BIBO-stability in terms of Lyapunov-stability criterion:

Theorem 6.4 [Reducing BIBO-Stability to Lyapunov Stability] *If the linear dynamical system $\dot{S} = AS$ is asymptotically stable, then the continuous-time component with the dynamics $\dot{S} = AS + BI$ and $O = CS + DI$ is BIBO-stable.* ■

The proof of this theorem relies on understanding the dynamics of a continuous-time component using *transfer functions* and is beyond the scope of this textbook.

Note that for the helicopter model of figure 6.5, if we set the input torque to 0, dynamics is given by $\dot{s} = 0$. In this dynamics, if the initial spin is s_0, then it will remain constant at the value s_0, and thus the system, while stable, is not asymptotically stable. As noted earlier, if the system is applied a constant torque as input, the spin increases linearly in an unbounded manner, and the system is not BIBO-stable. Thus, stability of the system $\dot{S} = AS$ does not imply BIBO-stability of the continuous-time component with the dynamics $\dot{S} = AS + BI$. In the reverse direction, BIBO-stability of the continuous-time component with the dynamics $\dot{S} = AS + BI$, in itself, does not ensure that Lyapunov-style stability properties of the system $\dot{S} = AS$ and additional properties of the matrices need to be established for this purpose.

Exercise 6.16*: Prove proposition 6.1. ■

Exercise 6.17: For each of the systems in exercises 6.13, 6.14, and 6.15, determine whether the system is asymptotically stable, stable but not asymptotically stable, or unstable. ■

6.3 Designing Controllers

Given a dynamical system model of the plant, the controller is designed to provide the controlled input signals to maintain the output of the system close to the desired output adjusting to changes in the uncontrolled disturbances. Designing controllers in a principled manner is a well-developed discipline. We will review some basic terminology in control design and get familiar with the most commonly used class of controllers in industrial practice.

6.3.1 Open-Loop vs. Feedback Controller

An *open-loop* controller does not use measurements of the state or outputs of the plant to make its decisions. Such a controller relies on the model of the plant to decide on the controlled input for the plant, and its implementation does not require sensors. For example, consider the first model of the car from figure 6.3. Suppose the controller's objective is to maintain a constant velocity (that is, we want $\dot{v} = 0$ at all times). Then if the initial velocity is v_0, the desired value of the input force F equals $k\,v_0$: the controller can simply apply this constant force to the car to maintain the velocity constant at the value v_0. Such a controller is called *open-loop*: when compared to the architecture shown in figure 6.1, the block for sensors and the flow of information from the plant to the controller is missing. Obviously, such a controller would be less expensive to implement than the one that requires sensors to estimate the state of the plant. However, its design heavily relies on the assumption that the behavior of the plant is entirely predictable and accurately captured by the idealized mathematical model. In practice, operation of such a controller is acceptable, provided there is a possibility of manual intervention. If the driver finds the speed of the car unacceptable, she would simply increment or decrement the desired speed triggering a recalculation of the force applied by the open-loop controller.

A *feedback* controller uses sensors to measure the output, and thus indirectly the current state of the plant, to update the values of the controlled input variables. For example, in the revised model of the car in figure 6.8, the model accounts for the change in the grade of the road. Suppose that the controller is applying the correct amount of force to maintain the velocity of the car at the desired cruising speed. A positive change in the grade θ causes the car to slow down, while a negative change in θ causes the car to speed up. The speed of the car, as measured by the sensors, is an input to the controller. It notices the change in speed and adjusts the force to make the velocity again equal to the desired cruising speed. A feedback controller not only can cope with disturbances (such as the grade) whose variation with time is not predictable in advance, but it can work well even when the mathematical model of the plant is only a rough approximation of the real-world dynamics. Implementation of a feedback controller requires sensors, and its performance is related to the accuracy of measurements by these sensors.

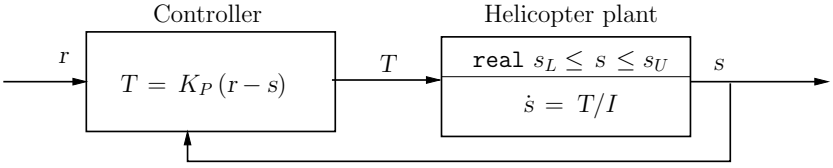

Figure 6.10: Stabilizing Controller for the Helicopter Model

6.3.2 Stabilizing Controller

Stabilizing Helicopter Model

We now describe a simple and typical pattern for designing a controller using the helicopter example of figure 6.5. Recall that the given helicopter model is unstable. The controller is shown in figure 6.10. It takes two input signals: the input variable r represents the *reference signal* that captures the desired spin, and the signal s is the plant output, namely, the (measured) spin of the helicopter. Given two such input variables corresponding to the desired and actual values, we can define the *error signal* e by the equation $e = r - s$. The goal of the controller is to keep the magnitude of the error as small as possible and also ensure that the closed-loop system obtained by composing the helicopter model and the controller is stable. Note that a positive value of e means that the controller should try to increase the actual spin s by applying a positive torque, and a negative value of e means that the controller should try to decrease the actual spin s by applying a negative torque. The controller of figure 6.10 computes the torque by simply scaling the error signal by a positive constant factor K_P. Such a controller is called a *proportional controller*, and the constant K_P is called the *gain* of the controller.

For the closed-loop system consisting of the composition of the controller and the helicopter, the input signal is the reference value r and the output is the spin s. The dynamics of the composite system is given by the equation

$$\dot{s} \;=\; K_P\,(r - s)/I.$$

This is a one-dimensional linear system, and the coefficient capturing the dependence of the rate of change of the state variable s on itself is $-K_P/I$. We know that such a system is asymptotically stable exactly when this coefficient is negative. Thus, the composite system is asymptotically stable as long as the gain K_P is positive.

When the reference input is 0, that is, when the objective of the controller is to keep the helicopter from spinning, the controller applies the torque equal to $-K_P/I$. No matter what the initial spin s_0 is, this causes the spin to decay to 0 exponentially. The higher the value of K_P, the faster the rate of convergence.

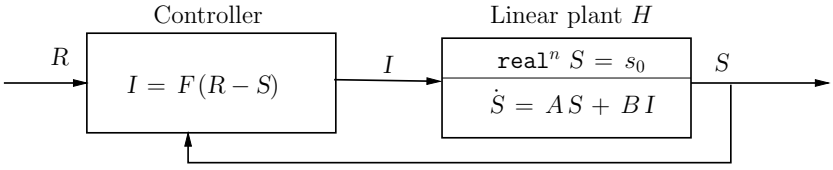

Figure 6.11: State Feedback Controller for Linear Systems

Linear State Feedback

The general architecture of a state feedback controller for a linear system is shown in figure 6.11. The original system is modeled by the linear component H. It has n state variables S and m input variables I, and its dynamics is given by the linear differential equation $\dot{S} = A\,S + B\,I$. In this setup, the assumption is that the controller can observe the state fully, that is, the set of output variables of the plant coincides with its state variables. The controller computes its output I based on the state S of the plant and the *reference* input R. The reference input R has the same dimension n as the state vector S.

The controller is a stateless linear component and is defined by the linear transformation

$$I \;=\; F\,(R \;-\; S).$$

The transformation matrix F has dimension $m \times n$ and is called the *gain matrix*. The closed-loop system has n-dimensional state S and n-dimensional reference input R, and its dynamics is given by

$$\dot{S} \;=\; (A \;-\; BF)\,S \;+\; B\,F\,R.$$

The control design problem is to choose the $(m \times n)$-matrix F such that the composite system is asymptotically stable. Then by theorem 6.4, we are guaranteed that the system is also BIBO-stable, and thus small variations in the reference inputs do not cause large perturbations in the state signal.

Design of the Gain Matrix

Recall that a linear system with dynamics $\dot{S} = A\,S$ is asymptotically stable exactly when every eigenvalue of the matrix A has a negative real part (theorem 6.3). Thus, to design a stabilizing controller for figure 6.11, given the matrices A and B, we need to choose the gain matrix F such that every eigenvalue of the matrix $(A - BF)$ has a negative real part.

To illustrate this computation, let us consider the system with two state variables and one input variable given by:

$$\dot{s}_1 \;=\; 4\,s_1 \;+\; 6\,s_2 \;+\; 2\,u; \quad \dot{s}_2 \;=\; s_1 \;+\; 3\,s_2 \;+\; u.$$

The matrix A for this system is the matrix A_1 for which we computed eigenvalues, eigenvectors, and the matrix exponential in section 6.2.2. Recall that the eigenvalues are 6 and 1, and thus the system is not stable. The matrix B is the column vector $[2 \;\; 1]^T$. The desired gain matrix F is a (1×2)-matrix, and let its entries be f_1 and f_2. We want to select values for these unknowns f_1 and f_2 so that the following matrix has eigenvalues with negative parts:

$$\begin{bmatrix} 4 & 6 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} [f_1 \;\; f_2] = \begin{bmatrix} 4-2f_1 & 6-2f_2 \\ 1-f_1 & 3-f_2 \end{bmatrix}.$$

Control design now corresponds to choosing eigenvalues for this matrix and then solving for the unknown f_1 and f_2. The characteristic polynomial for this matrix is

$$\begin{aligned} P(\lambda, f_1, f_2) &= (4-2f_1-\lambda)(3-f_2-\lambda) - (6-2f_2)(1-f_1); \\ &= \lambda^2 + (2f_1+f_2-7)\lambda + (6-2f_2). \end{aligned}$$

The roots of this characteristic polynomial are λ_1 and λ_2 if the polynomial is of the form $(\lambda-\lambda_1)(\lambda-\lambda_2)$. By matching the coefficients of these quadratic polynomials, we conclude that the eigenvalues of the matrix $(A - BF)$ are λ_1 and λ_2 exactly when the entries of gain matrix F satisfy the following equations:

$$\begin{aligned} 2f_1 + f_2 - 7 &= -\lambda_1 - \lambda_2; \\ 6 - 2f_2 &= \lambda_1\lambda_2. \end{aligned}$$

If we prefer eigenvalues -1 and -2 (and these would ensure asymptotic stability), then we need to solve

$$2f_1 + f_2 - 7 = 3; \;\; 6 - 2f_2 = 2.$$

Solving these equations gives us $f_1 = 4$ and $f_2 = 2$. Thus, the desired matrix F is $[4 \;\; 2]$, that is, the controller should provide the input signal $u = 4(r_1 - s_1) + 2(r_2 - s_2)$ to the system so that the resulting closed-loop system is asymptotically stable with eigenvalues -1 and -2.

We can also choose the eigenvalues to be complex numbers as long as when we choose a complex number, we also choose its conjugate. For example, if we want the eigenvalues to be $-1 + j$ and $-1 - j$, then we need to solve

$$2f_1 + f_2 - 7 = 2; \;\; 6 - 2f_2 = 2.$$

Solving these equations give us $f_1 = 7/2$ and $f_2 = 2$. This means that with the choice of the gain matrix F to be $[7/2 \;\; 2]$, the resulting closed-loop system is asymptotically stable with eigenvalues $-1 + j$ and $-1 - j$.

Controllability of the Matrix Pair (A, B)

To summarize, to design a stabilizing feedback controller, we choose eigenvalues $\lambda_1, \lambda_2 \ldots \lambda_n$, all with negative real parts, and solve the equation

$$\mathtt{det}\,[\,A - BF - \lambda \mathbf{I}\,] = (\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_n),$$

with mn unknowns corresponding to the entries of the gain matrix F. This approach to stabilization naturally raises the following questions: (1) when is this equation guaranteed to have solutions? and (2) does the existence of a solution depend on the choice of the eigenvalues? It turns out that when the pair of matrices A and B satisfy a certain property, for every choice of the eigenvalues $\lambda_1, \lambda_2 \ldots \lambda_n$, it is possible to choose the entries of the gain matrix F so as to satisfy the above equation.

Given an $(n \times n)$-matrix A and an $(n \times m)$-matrix B, consider the following matrix with n rows and mn columns:

$$\mathbf{C}(A, B) \;=\; [\; B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B \;].$$

That is, the first m columns of the matrix $\mathbf{C}(A, B)$ are the columns of the $(n \times m)$-matrix B, the next m columns are the columns of the $(n \times m)$-matrix AB, and the last m columns are the columns of the $(n \times m)$-matrix $A^{n-1}B$. This matrix $\mathbf{C}(A, B)$ is called the controllability matrix corresponding to the matrix pair (A, B). The pair (A, B) of matrices is called *controllable* if the rank of the controllability matrix is n, that is, if all the rows of the matrix $\mathbf{C}(A, B)$ are linearly independent. In such a case, the linear system with the dynamics $\dot{S} = AS + BI$ is also called controllable.

The following theorem tells us that controllability is a necessary and sufficient condition for the existence of a gain matrix corresponding to the chosen eigenvalues. If the pair (A, B) of matrices is controllable, then it is possible to obtain the desired gain matrix F for any arbitrary choice of eigenvalues as long as we choose only real numbers or when we choose a complex number, we also choose its conjugate. If the pair (A, B) of matrices is not controllable, then not all eigenvalues can be chosen freely.

Theorem 6.5 [Controllability and Eigenvalue Assignment] *Let A be an $(n \times n)$-matrix and B be an $(n \times m)$-matrix. The following two statements are equivalent:*

- *The $(n \times mn)$-controllability matrix $\mathbf{C}(A, B)$ whose columns are the columns of the matrices $B, AB, \ldots A^{n-1}B$, has rank n.*
- *For every choice of (complex) numbers $\lambda_1, \lambda_2, \ldots \lambda_n$ such that a complex number appears in this list exactly when its conjugate also appears in the list, there exists a $(m \times n)$-matrix F such that the eigenvalues of the matrix $(A - BF)$ are $\lambda_1, \lambda_2, \ldots \lambda_n$.*

■

Continuing with our example,

$$A_1 \;=\; \begin{bmatrix} 4 & 6 \\ 1 & 3 \end{bmatrix};\; B \;=\; \begin{bmatrix} 2 \\ 1 \end{bmatrix};\; \mathbf{C}(A_1, B) \;=\; \begin{bmatrix} 2 & 14 \\ 1 & 5 \end{bmatrix}.$$

In this example, $m = 1$ and $n = 2$. The controllability matrix $\mathbf{C}(A_1, B)$ is a (2×2)-matrix, and its rank is 2. Thus, the pair (A_1, B) is controllable. As

we have analyzed already, the eigenvalues of the matrix $(A - BF)$ are λ_1 and λ_2 exactly when the entries of gain matrix F satisfy the following equations: $2f_1 + f_2 = -\lambda_1 - \lambda_2 + 7$ and $2f_2 = 6 - \lambda_1\lambda_2$. This system of linear equations is guaranteed to have a solution for the entries f_1 and f_2 no matter what values we choose for λ_1 and λ_2 (note: if λ_1 is a complex number, then λ_2 must be its conjugate, and this ensures that their sum and product are both real numbers). By theorem 6.5, we know that this is not a coincidence.

For a linear system, controllability entails many other appealing properties. If the linear system is controllable, then for any given initial state and a target state, it is possible to provide an input signal to the system so that the state of the system starting from the initial state becomes equal to the target state in a finite time. More precisely, consider a linear system with dynamics $\dot{S} = AS + BI$ and an initial state s_0, such that the pair (A, B) of matrices is controllable. Then for every state $s \in \texttt{real}^n$, there exists a time $t^* \in \texttt{time}$ and an input signal $\overline{I}$ such that for the unique state signal $\overline{S}$ of the system as a response to the input signal $\overline{I}$ starting from the initial state s_0, $\overline{S}(t^*) = s$.

While we have presented a method for designing a controller to ensure the critical property of stability, it is worth mentioning that there are two key aspects of the control design that we are not addressing here, but for which the theory of linear systems provides well-understood tools:

- **Optimality:** there are many choices for the gain matrix F that ensure stability of the resulting closed-loop system. Theory of optimal control, and in particular the technique for designing the *Linear Quadratic Regulator* (LQR) controller, addresses the question of choosing a matrix that satisfies additional criteria (for example, driving certain state variables to 0 as fast as possible).
- **State Estimation:** We have assumed that the input to the controller is the complete state S of the system. What happens when the controller can observe the state of the system only partially via the output vector O? In this case, the controller needs to *estimate* the state of the system based on the observation of the output signal and the theory of observability and state estimation develops techniques for this purpose.

Exercise 6.18: Consider the linear system with two state variables, one input variable, and the dynamics given by:

$$\dot{s}_1 = s_1/2 + s_2 + u;\ \dot{s}_2 = s_1 + 2s_2 + u.$$

(1) Show that the system is not stable. (2) Show that the system is controllable. (3) Find the gain matrix F so that the eigenvalues for the resulting closed-loop system are $-1 + j$ and $-1 - j$. ■

Exercise 6.19*: Consider the linear system with two state variables, one input variable, and the dynamics given by:

$$\dot{s}_1 = -2s_2 + u;\ \dot{s}_2 = s_1 - 3s_2 + u.$$

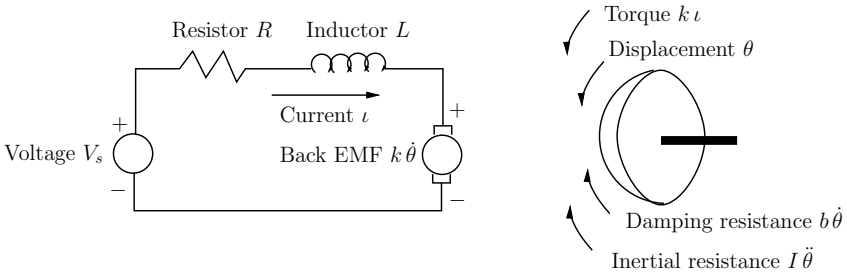

Figure 6.12: Dynamics of a DC Motor

First, show that the matrix-pair (A, B) is not controllable. Then show that it is not possible to choose entries of the gain matrix F so that the matrix $(A - B\,F)$ has arbitrarily chosen eigenvalues: show that one of the eigenvalues of the matrix $(A - B\,F)$ is always -1 no matter what the entries in the gain matrix F are (although the second eigenvalue can be set to an arbitrary value by suitably choosing the entries of F). ■

6.3.3 PID Controllers *

In industrial control systems, the most commonly used design of a controller to correct the discrepancy between the desired reference signal and the measured output signal uses a combination of three terms: a *proportional* term capturing the reaction to the current error, an *integral* term capturing the reaction to the cumulative error, and a *derivative* term capturing the response to the rate of change of error. Such a controller is called a PID controller. Let us illustrate the design of such a controller using a classical example of a control system, namely, a DC motor.

DC Motor

Figure 6.12 shows the design of a DC motor that converts input voltage to rotational motion and is a commonly occurring building block of many electromechanical devices. The electrical circuit is shown on the left and is connected to the rotating shaft shown on the right.

Let us denote the input voltage by V_s, the resistance of the circuit by R, and the inductance of the circuit by L. Let ι denote the electrical current flowing through the circuit and θ denote the angular displacement of the wheel. If k is the electromotive-force (EMF) constant, then the back electromotive-force (EMF) voltage generated by the rotating shaft equals k times its angular velocity. Basic laws of electric circuits tell us that (1) the voltage across a resistor equals the product of the resistance and the electrical current flowing through the circuit, (2) the voltage across an inductor equals the product of the inductance and the

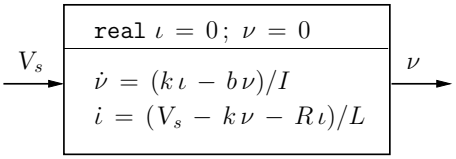

Figure 6.13: Continuous-time Component Modeling DC Motor

rate of change of the electrical current flowing through the circuit, and (3) the sum of voltages around a closed path in an electrical circuit must equal to 0 (Kirchhoff's law). Applying these rules gives us the equation:

$$L\,\dot{\iota} \;+\; R\,\iota \;+\; k\,\dot{\theta} \;=\; V_s.$$

Now let us analyze the motion of the rotating shaft. The torque acting on the shaft equals the EMF constant k times the current flowing through the circuit. If I is the rotational inertia and b is the coefficient of friction corresponding to the damping effects, then Newton's law for rotating bodies gives us:

$$I\,\ddot{\theta} \;+\; b\,\dot{\theta} \;=\; k\,\iota.$$

The output of the DC motor is the rotational velocity of the shaft. The dynamics then can be captured by the continuous-time component shown in figure 6.13. In this model, ν denotes the rotational velocity, and the angular displacement θ is omitted.

Controller for the DC Motor

The controller for the DC motor of figure 6.13 observes the rotational velocity ν and adjusts the source voltage V_s to achieve the desired response. The canonical task for such a controller is to control the voltage so that the rotational velocity changes from its initial value to a desired speed, say r. We already know that a general technique for designing a *proportional* controller consists of scaling the difference between the reference value and the observed output by a suitably chosen proportional gain constant K_P:

$$V_s \;=\; K_p\,(r \;-\; \nu).$$

While we have already analyzed the relationship between the value of the gain constant and the (asymptotic) stability of the resulting closed-loop system, to understand the various requirements other than stability that are of practical interest, let us examine the typical behavior of the DC motor with a proportional controller as shown in figure 6.14. This plot is generated using MATLAB by setting the rotational inertia $I = 0.01$, the damping constant $b = 0.1$, the electromotive force constant $k = 0.01$, the resistance $R = 1$, the inductance $L = 0.5$, the reference input $r = 1$, and the proportional gain constant $K_P = 100$.

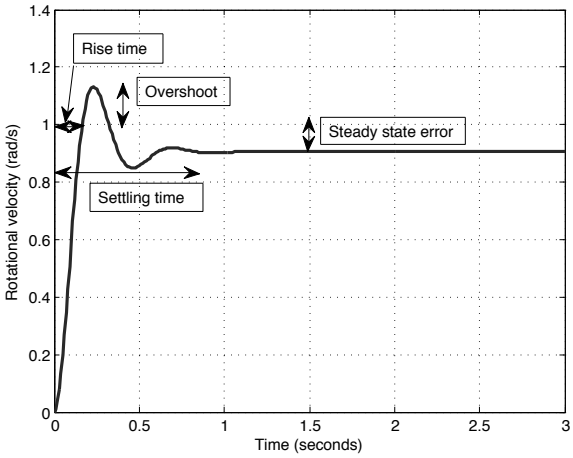

Figure 6.14: Output Response of the DC Motor with a Proportional Controller

The response shown in figure 6.14 is called the *step response* of a continuous-time component: at time t_0 (0 in this example), we want to change the output from some initial value (0 in this example) to a new reference value (1 in this example). The shape of the output response signal shown in figure 6.14 is illustrative of a large number of physical systems. Let us consider intuitively how the output of a system typically changes in response to a change in the reference signal. Let e denote the difference between the reference value and the output of the system. At the initial time instance, the magnitude of this error e is the highest. As e changes, so does the value of the control input supplied by the controller, which impacts the state and the output of the system. The output approaches 1 but overshoots the desired value due to the smoothness of the dynamics of the physical world. The overshoot makes the error negative, causing the controller to change the direction of the derivative of its output. The same phenomenon repeats, causing the output to oscillate for a while before it settles into a steady-state value, which does not change in absence of further external disturbances.

Given such an expected response of the system output, the following metrics capture the performance of the controller:

1. *Overshoot:* The difference between the maximum value of the system output and the desired reference value. For the DC motor response shown in figure 6.14, the maximum rotational velocity is 1.12 rad/s, causing an overshoot of 12%. Ideally, the overshoot should be as small as possible. In particular, a safety requirement can assert that the overshoot should be below some threshold value.

2. *Rise time:* The time difference between the initial time when the reference signal changes and the time at which the output signal crosses the desired

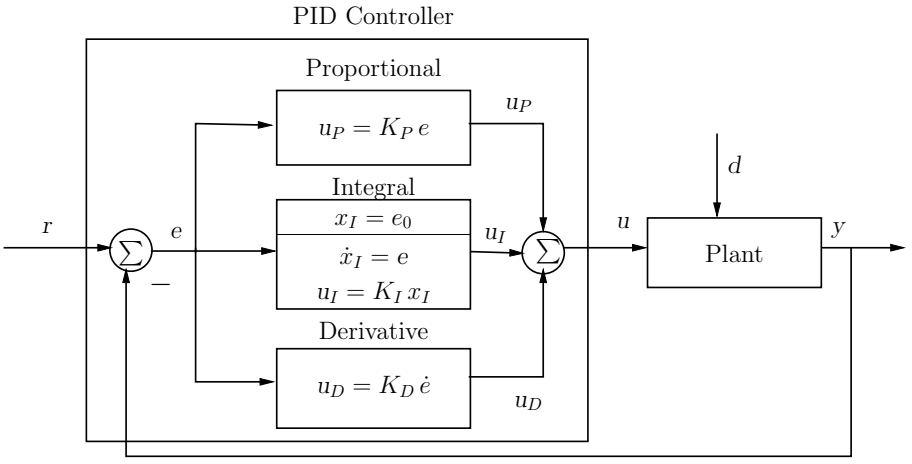

Figure 6.15: A Generic PID Controller

reference value. For the DC motor response shown in figure 6.14, the rise time is 0.15 seconds. Ideally, smaller rise time means better responsiveness of the system, and typically an attempt to reduce the rise time will increase the overshoot.

3. *Steady-state error:* The difference between the steady-state value of the output signal and the value of the reference signal. For the DC motor response shown in figure 6.14, the output stabilizes at 0.9 rad/s, thus leading to 10% steady-state error. Ideally, steady-state error should be 0, but a small error may be acceptable.

4. *Settling time:* The time difference between the initial time when the reference signal changes and the time at which the output signal reaches its steady-state value. For the DC motor response shown in figure 6.14, the settling time is 0.8 seconds. Ideally, settling time should also be small, and the system should reach the desired output value with few oscillations.

For the proportional controller for the DC motor, changing the value of the gain constant K_P affects the rise time and the overshoot, but a purely proportional controller will not get rid of the steady-state error. For optimal performance, we need to incorporate both derivative and integral components.

PID Controller

A generic version of the PID controller is shown in figure 6.15. The controller takes two signals as inputs: the reference signal r and the measured output y of the dynamical system to be controlled. Let the variable e denote the *error signal* capturing the difference between the reference signal r and the measured

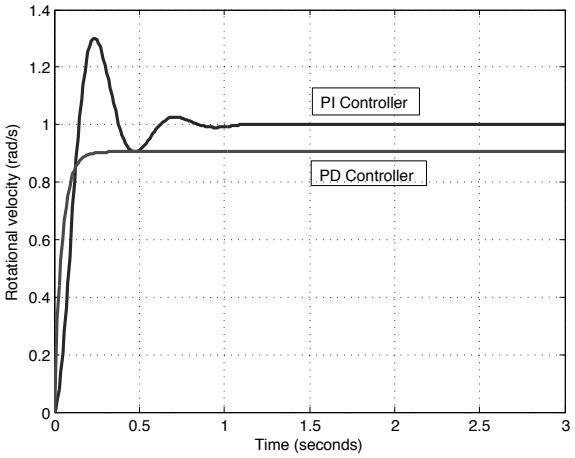

Figure 6.16: Output Response of the DC Motor with PD and PI Controllers

plant output y. Then the controller's output u, which is fed to the plant, is the sum of three terms:

- **Proportional term** $K_P\, e$: The contribution of this term is directly proportional to the current error. The constant K_P is called the *proportional gain*, and the controller scales the error by this factor.

- **Integral term** $K_I \int_0^t e(\tau)\, d\tau$: Note that the integral of the error signal up to time t gives the cumulative error up to time t, and thus the contribution of this term accounts for the cumulative error so far. The constant K_I is called the *integral gain*, and the controller scales the accumulated error by this factor.

- **Derivative term** $K_D\, \dot{e}$: The contribution of this term is correlated to the rate at which the error changes. The constant K_D is called the *derivative gain*, and the controller scales the rate of change of the error by this factor.

Note that the proportional component of the PID controller is stateless. The integral component maintains a state variable x_I that captures the accumulated error, and the rate of change of this state variable equals the error e. The derivative component has a state variable x_D that stores the error e, and the output of the derivative component is the first-order derivative of this state variable. In figure 6.15, using the standard convention, the Sum block is illustrated as a circle labeled with the symbol Σ. Such a component simply outputs the sum of its input signals, where the negative sign on an input signal indicates that the corresponding input should be subtracted. Some of the components may be missing in a specific control design. For instance, a P controller has only the proportional block, and a PI controller has only the proportional and the integral blocks. Both of these are also common in practice.

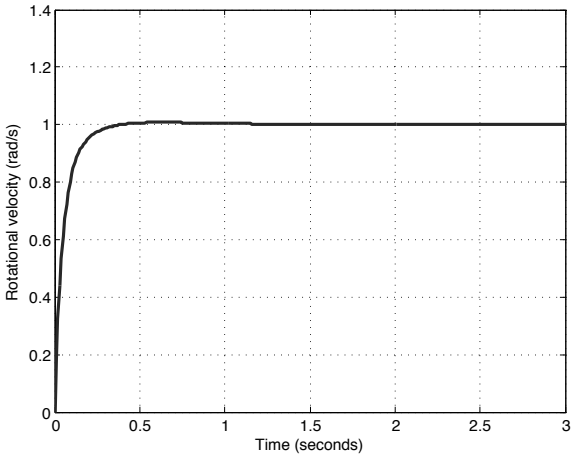

Figure 6.17: Output Response of the DC Motor with a PID Controller

PID Controller for the DC Motor

To understand how the control performance changes with the contributions of various terms, let us revisit the proportional controller for the DC motor (see figure 6.14). Higher values of the proportional gain K_P mean that the rise time will be smaller but with higher overshoot. Too high a value of K_P can cause large oscillations, delaying settling time.

A purely proportional controller has a steady-state error. The effect of the integral term gets rid of the steady-state error. Figure 6.16 shows a PI controller for the DC motor with $K_P = 100$ and $K_I = 200$ (the values of all the other parameters are unchanged). Note that the output now stabilizes close to the desired value 1 (the steady-state error is nonzero but very small), but the overshoot has increased to 30%, which may be unacceptable, and both rise time and settling time have also increased (compared to a purely proportional controller). Higher values of the integral gain K_I lead to better responsiveness but also contribute to higher overshoot.

To reduce the overshoot, we need to use the derivative component. Figure 6.16 also shows response to a PD controller for the DC motor with $K_P = 100$ and $K_D = 10$ (the values of all the other parameters are unchanged). There is no overshoot, but the steady-state error is still large as the output stabilizes at 0.9 rad/s.

Good performance on all metrics is obtained by a judicious use of all three components: figure 6.17 shows a PID controller for the DC motor with $K_P = 100$, $K_D = 10$, and $K_I = 200$. Now the steady-state error is insignificant, there is no overshoot, and the settling time is 0.4 seconds.

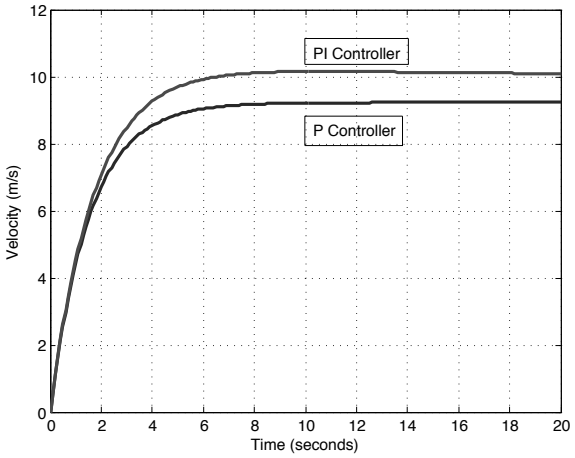

Figure 6.18: Velocity Response with a Cruise Controller

Cruise Controller

Let us revisit the model of a car in figure 6.3. The input to the cruise controller is the car output, namely, the velocity v, and the reference value r, that is, the desired speed set by the driver. The goal of the controller is to supply the input force F so that the velocity becomes equal to the reference value.

We begin the design by first using a proportional controller. The dynamics of the closed loop system is given by:

$$\dot{v} \;=\; (F - kv)/m;\; F \;=\; K_P(r - v).$$

If we set the initial velocity v_0 to 0, the mass m to 100 kg, the coefficient of friction k to 50, the desired velocity r to 10 m/s, and the proportional gain K_P to 600, the resulting velocity response is shown in figure 6.18. There is a significant steady-state error: when the driver wants to increase the speed by 10 m/s, it increases by only 9 m/s. Notice that the settling time is about 10 seconds, which seems reasonable. Increasing the value of K_P will reduce the steady-state error but will also significantly decrease the settling time, which is likely to result in an uncomfortably high acceleration for the passengers.

To remove the steady-state error, we can add an integral component. Note that since there is no overshoot, we don't need a derivative component. For the same values of all the other parameters, the velocity response to the PI controller with $K_P = 600$ and $K_I = 40$ is also shown in figure 6.18. The step response in this case seems ideal: the velocity increases by 10 m/s is about 7 seconds, with no overshoot, and stays stable at the desired value with a small steady-state error.

Note that the basic design of the PI cruise controller did not explicitly depend on the model of the car, but the values of the gain constants K_P and K_I

were obtained by running simulations of the model for different choices. For a different model of the car, for instance, for the car on a graded road or for a car with different values of the parameters m and k, suitable values of the gain constants K_P and K_I need to be obtained.

Let us now revisit the design of the synchronous component `CruiseController` discussed in section 2.4.2. The component `ControlSpeed` can indeed be implemented as a PI controller discussed above. The two inputs *speed* and *cruiseSpeed* correspond to the observed velocity and the desired velocity, and its output corresponds to the force. However, we have a semantic mismatch between the discrete and continuous worlds: the components in section 2.4.2 interact with one another in a discrete manner, and variables range over natural numbers; while the models of the car and the PI controller are continuous-time components that process real-valued signals. This gap is usually bridged informally: an actual implementation of the PI controller samples the velocity only at discrete intervals, and the values it operates on are finite-precision numbers with rounding. This discretization can potentially introduce errors, which means that model-based analysis cannot replace extensive testing of the final system integrating all the components.

Exercise 6.20: Recall the linear pendulum model from exercise 6.10. We will use a *Proportional-Derivative* (PD) controller for the linearized model. The feedback control is given as $u = K_P\,\varphi + K_D\,\dot{\varphi}$, where we need to suitably choose the gain parameters K_P and K_D. Write the equations of the closed-loop system, which consists of the composition of the linearized model of the pendulum and the PD controller. For what values of the parameters K_P and K_D is the closed-loop system stable? Suppose the mass m and the length ℓ are such that $m\ell^2 = 1\,(kg.m^2)$ and $mg\ell = 1\,(N.m)$. Simulate the closed-loop system using MATLAB with different values of the gain parameters K_P and K_D. You can choose some suitable initial position and initial angular velocity. Experiment with different parameter values. Plot and discuss your results. ■

Exercise 6.21*: Figure 6.18 shows the response of the car to a PI controller. Suppose we apply the same controller but now to the model of the car on a graded road as shown in figure 6.8. Consider the input signal $\overline{\theta}(t) = (\texttt{sin}\,(t/5))/3$ (measured in radians) that models a sinusoidal variation in the grade of the road. For this input, plot the velocity of the car in response to the PI controller using the same parameters: the initial velocity v_0 is 0, the mass m is $100\,kg$, the coefficient of friction k is 50, the gravitational acceleration g is $9.8\,m/s^2$, the reference velocity r is $10\,m/s$, the proportional gain K_P is 600, and the integral gain K_I is 40. ■

6.4 Analysis Techniques *

Given a model of a continuous-time component, which may include the plant model, the feedback controller, and constraints on the initial values and disturbances, we want to analyze the behavior of the system. We first discuss the

traditional approach based on *numerical simulation* and then some constraint-based techniques for verifying stability and safety requirements.

6.4.1 Numerical Simulation

Consider a continuous-time component with state variables S and input variables I. The function f gives the rate of change of the state as a function of the state and input variables. Given an input signal $\overline{I}$ that assigns values to inputs as a function of time and an initial state s_0, the evolution of the state of the system, then, can be computed by solving the differential equation $\dot{S} = f(S, I)$ with the initial condition $\overline{S}(0) = s_0$. While for some specific forms of the function f, a closed-form solution for the state response $\overline{S}(t)$ at time t can be computed, a general method for computing this signal is to employ *numerical simulation*. For numerical simulation, the user provides a discretization step parameter Δ, and the simulator attempts to compute the values of the state at times $\Delta, 2\Delta, 3\Delta, \cdots$ that approximate the values of the desired response signal $\overline{S}(t)$ as closely as possible. The simulation algorithm samples the input signal $\overline{I}(t)$ only at times $0, \Delta, 2\Delta, 3\Delta, \cdots$ and the result of the simulation thus depends on the discrete sequence $u_0, u_1, u_2, \ldots$ of input values, where u_i is the value of the input signal $\overline{I}(t)$ at time $t = i\,\Delta$ for each i.

Euler's Method

Euler's method relies on the observation that the rate of change of the desired state signal $\overline{S}$, that is, $d\,\overline{S}/dt$, at time t, is simply the limit of the quantity $(\overline{S}(t+\Delta) - \overline{S}(t))/\Delta$ as the increment Δ goes to 0. As a result, for small values of Δ, it is natural to approximate the value of $\overline{S}(t+\Delta)$ by the following equation

$$\overline{S}(t+\Delta) \;=\; \overline{S}(t) + \Delta\, f(\overline{S}(t), \overline{I}(t)).$$

This approximation assumes that the rate of change of state is constant during the interval $[t, t+\Delta)$ and equals the rate of change at the beginning of the interval. The rate of change at the beginning of the interval is obtained by evaluating the function f using the values of the state and the input at time t. The change in the value of the state is obtained by multiplying this rate by the size Δ of the interval. Thus, given an initial value s_0 for the state and a sequence of values $u_0, u_1, \ldots$ for the input signal, the Euler's method for simulation of the differential equation $\dot{S} = f(S, I)$ computes the following sequence of values: for every $i \geq 0$,

$$s_{i+1} \;=\; s_i + \Delta\, f(s_i, u_i).$$

This sequence of values is linearly extrapolated to give the response signal that defines the state $\overline{S}(t)$ at every time $t \in \texttt{time}$: for every $i \geq 0$ and time value $t \in [i\,\Delta, i\,\Delta + \Delta)$, $\overline{S}(t) = s_i + (t - i\,\Delta)\, f(s_i, u_i)$.

Runge-Kutta Methods

Euler's method estimates the state at the end of an interval by assuming that the rate of change of state stays constant during the interval, and this rate is based only on the state at the beginning of the interval. A better approximation can be obtained if the estimated change in the state at the end of the interval is used to estimate a change in the derivative, and this is used to readjust the state estimate. Runge-Kutta methods comprise a popular class of numerical integration methods based on this idea. In particular, the *second-order Runge-Kutta method* computes the state s_{i+1} by the following calculation:

$$\begin{aligned} k_1 &= f(s_i, u_i), \\ k_2 &= f(s_i + \Delta\, k_1, u_{i+1}), \\ s_{i+1} &= s_i + \Delta\,(k_1 + k_2)/2. \end{aligned}$$

Given the current state s_i and input u_i, the first step computes k_1 to be the current rate of change. However, instead of setting the state s_{i+1} at the end of the interval to be $s_i + \Delta\, k_1$ as in Euler's method, it uses this estimated state to calculate the estimated rate of change k_2 at the end of the interval (the input value u_{i+1} at the end of the interval is used for this estimate). The third step calculates s_{i+1} assuming that the rate of change is constant during the interval but equals the average of the two values k_1 and k_2.

The higher order Runge-Kutta methods use the same basic idea but use estimates of derivatives at the midpoint as well as the endpoints to compute a weighted average. The most commonly used method in practice is the fourth-order Runge-Kutta method. It computes the state s_{i+1} by the following calculation:

$$\begin{aligned} k_1 &= f(s_i, u_i), \\ k_2 &= f(s_i + \Delta\, k_1/2, (u_i + u_{i+1})/2), \\ k_3 &= f(s_i + \Delta\, k_2/2, (u_i + u_{i+1})/2), \\ k_4 &= f(s_i + \Delta\, k_3, u_{i+1}), \\ s_{i+1} &= s_i + \Delta\,(k_1 + 2k_2 + 2k_3 + k_4)/6. \end{aligned}$$

Quality of Approximation

To understand how well these simulation techniques approximate the desired function, let us consider a single-dimensional linear differential equation with no inputs: $\dot{s} = s$ with the initial state $s_0 = 2$. We already know that the solution to this equation is given by the signal $\overline{s}(t) = 2e^t$.

For numerical simulation, let us choose the length Δ of the interval to be 0.1. Over the time interval of $[0, 5]$, the simulation plots resulting from both Euler's method and the second- Runge-Kutta method are shown in figure 6.19. Note that Euler's method introduces significant error that accumulates: the value of

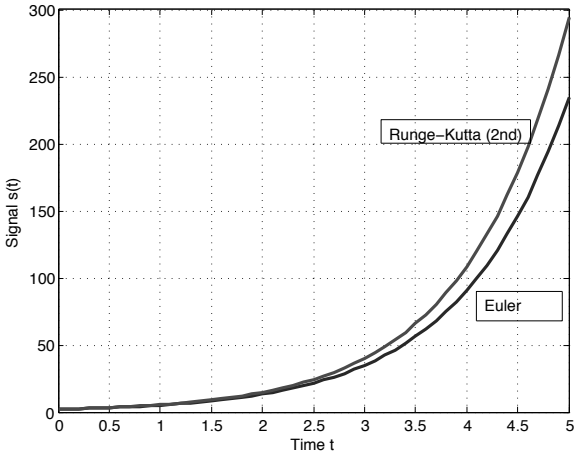

Figure 6.19: Alternative Simulation Algorithms

$\overline{s}(5)$ by Euler's method is 234.78, while the value of $\overline{s}(5)$ by the second-order Runge-Kutta method is 294.54. In this example, switching to the fourth-order Runge-Kutta method improves the accuracy slightly. Since the differences in the plots of second-order and fourth-order methods are not noticeable, figure 6.19 does not show the plot resulting from the fourth-order method, but it turns out that the value of $\overline{s}(5)$ by the fourth-order Runge-Kutta method is 296.83, which coincides with the quantity $2e^5$. Also note that if you were to simulate this differential equation using the standard solvers and plotting routines in MATLAB, the result is identical to the plot given by the fourth-order Runge-Kutta method.

The quality of approximation afforded by numerical simulation can be improved by reducing the step size Δ. Simulation tools automatically tune this parameter to keep the error small. In particular, adaptive techniques are also used to change the value of Δ dynamically, possibly at every step of simulation, to adjust to the current rate of change.

Exercise 6.22: Consider the single-dimensional linear differential equation $\dot{s} = s$. Suppose the initial state is s_0, and let Δ be the length of the interval used for numerical simulation. For Euler's method, find a closed-form formula for the state s_n after n steps of simulation as a function of s_0, Δ, and n. Verify that for $s_0 = 2$, $\Delta = 0.1$, and $n = 50$, s_{50} equals 234.78 (see figure 6.19). ■

Exercise 6.23: Consider the single-dimensional linear differential equation $\dot{s} = s$. Suppose the initial state is s_0, and let Δ be the length of the interval used for numerical simulation. For the second-order Runge-Kutta method of simulation, find a closed-form formula for the state s_n after n steps of simulation as a function of s_0, Δ, and n. Verify that for $s_0 = 2$, $\Delta = 0.1$, and $n = 50$, s_{50} equals 294.54 (see figure 6.19). ■

6.4.2 Barrier Certificates

In chapter 3, we focused on the safety verification problem for (discrete) transition systems: given a transition system and a set of safe states, is it the case that every reachable state of the system is safe? Now let us revisit this problem for continuous-time components.

Safety Verification

Consider a closed continuous-time component H with state variables S, initialization given by *Init*, and dynamics specified by $\dot{S} = f(S)$. For every initial state $s_0 \in [\![Init]\!]$, the system has a unique response signal, and the set of reachable states of the system is the set of states that appear along all these response signals:

$$Reach \;=\; \{\; \overline{S}(t) \;\mid\; \overline{S}(0) \in [\![Init]\!] \text{ and } t \in \texttt{time} \;\}.$$

Given a state property φ, the safety verification problem is to decide whether every state in the set *Reach* satisfies the property φ. For example, the property φ can assert that the magnitude of the error e between the reference signal and the system output is less than some constant δ, and a violation of this property indicates an unacceptable overshoot.

Numerical simulation is an effective technique to understand the behavior of a dynamical system starting from a specific initial state. When we know that the initial state belongs to a set, the simulation tool needs to choose different values for the initial state from the given set and run multiple simulations. Such an approach cannot be exhaustive, and thus we need some alternative techniques.

A natural approach would be to develop a *symbolic simulation* algorithm in the style of the symbolic reachability algorithm of section 3.4. However, this turns out to be challenging even for linear components. To illustrate the difficulty, let us consider a two-dimensional linear system whose dynamics is given by:

$$\dot{s_1} \;=\; -7\,s_1 \,+\, s_2\,; \quad \dot{s_2} \;=\; 8\,s_1 \,-\, 10\,s_2.$$

Suppose the initial set is the rectangle described by the formula

$$Init \;=\; 5 \,\leq\, s_1 \,\leq\, 6 \;\wedge\; -1 \,\leq\, s_2 \,\leq\, 1.$$

Figure 6.20 shows the initial set, and also the system responses when the initial state is chosen to be each of the four corner points. All these system responses converge to the origin, which is no coincidence: verify that both the eigenvalues of the state transition matrix are negative, and thus the system is asymptotically stable (see theorem 6.3).

It is evident from figure 6.20 that the set *Reach* of reachable states of the system has a complex form: it is *not* convex, and although the initial set is described by linear constraints, linear constraints do not suffice to describe the set *Reach*. As a result, it is not possible to develop symbolic methods to compute the set of reachable states exactly.

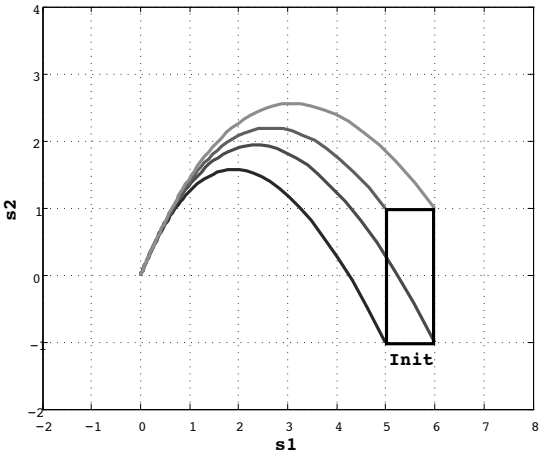

Figure 6.20: Reachable Set of a Continuous-time Component

Inductive Invariants

In chapter 3, we studied the principle of *inductive invariants* to prove safety requirements of (discrete) transition systems. To show that a property φ is an invariant of a transition system T, we find a property ψ such that (1) every state satisfying ψ satisfies φ, (2) the initial states of T satisfy ψ, and (3) if a state s satisfies ψ and (s, t) is a transition of T, then the state t is guaranteed to satisfy ψ. We describe an analogous method for establishing that a given state property is an invariant of a continuous-time system.

Let us revisit the dynamical system of figure 6.20. Suppose we want to establish that the property φ described by $-4 \leq s_2 \leq 4$ is an invariant of the system. In figure 6.21, we want to show that if the initial state belongs to the rectangle, then the execution stays between the two horizontal lines $s_2 = 4$ and $s_2 = -4$.

As in the case of proofs using inductive invariants, the first step is to identify the "strengthening" ψ. For continuous-time components, the desired strengthening is described by a function $\Psi : \texttt{real}^n \mapsto \texttt{real}$ mapping states to real numbers, that is, a real-valued expression over the state variables S. The set of states satisfying the equation $\Psi(S) = 0$ is called the *barrier*. The set of states satisfying the formula $\Psi(S) \leq 0$, that is, the set of states on or inside the barrier, is the analog of the inductive invariant. The barrier is chosen to satisfy three obligations. First, we need to establish that every state satisfying $\Psi(S) \leq 0$ also satisfies the desired safety property φ. Equivalently, if a state violates φ, then the value of Ψ must be positive. The second obligation is to show that every initial state s_0 satisfies $\Psi(s_0) \leq 0$. These two obligations imply that the barrier should be chosen so that it separates the initial states from the states violating the desired invariant property: all unsafe states are outside the barrier, and all initial states are inside the barrier.

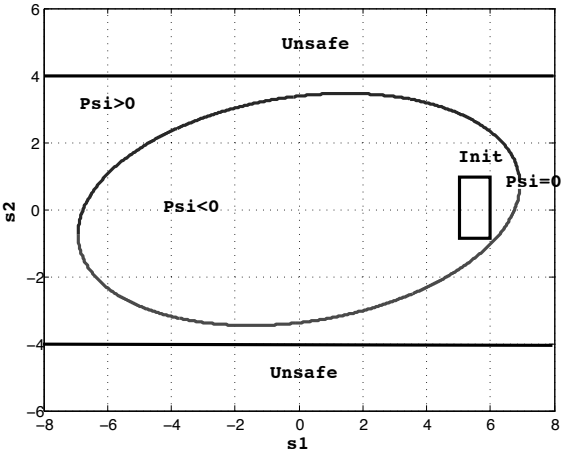

Figure 6.21: Invariant Verification Using a Barrier Certificate

For the example of figure 6.21, the barrier is described by the function

$$\Psi(s_1, s_2) \;=\; 7\,s_1^2 \;-\; 6\,s_1\,s_2 \;+\; 28\,s_2^2 \;-\; 320.$$

The equation $\Psi(s) = 0$ describes the ellipse shown in the figure. Initial states lie inside the ellipse, whereas unsafe states (that is, states above the line $s_2 = 4$ and below the line $s_2 = -4$) are outside the ellipse.

The third obligation in the discrete case corresponds to showing that the inductive invariant is preserved by system transitions. In case of continuous-time components, discrete transitions are replaced by continuous evolution of state over time as described by the differential equation $\dot{S} = f(S)$. A system response $\overline{S}(t)$ demonstrating a violation of the invariant must start in an initial state, which lies inside the barrier, and visit an unsafe state, which lies outside the barrier. The system response is a continuous and differentiable function, and hence it cannot "jump" across the barrier. It suffices to establish that the choice of the barrier with respect to the system dynamics f is such that a crossing of the barrier is impossible.

To understand the required condition, consider the example of figure 6.21. At a given state (s_1, s_2), the value of s_1 changes at the rate $f(s_1) = -7\,s_1 + s_2$, and the value of s_2 changes at the rate $f(s_2) = 8\,s_1 - 10\,s_2$. Thus, the state (s_1, s_2) flows, or evolves, along the vector $(-7\,s_1 + s_2, 8\,s_1 - 10\,s_2)$ given by the dynamics f. A key observation is that the value of Ψ always decreases along these flow directions. In particular, if the state lies on the barrier (that is, the boundary of the ellipse), this vector field points inward. This implies that if the state of the system at time τ lies on the barrier, then the state of the system

at time $\tau + dt$ must lie in the interior. It is clear that system executions cannot cross the barrier from inside to outside.

Lie Derivatives

The appropriate mathematical concept to formalize the requirement that the value of the function Ψ decreases along the flow direction is called the *Lie derivative*. The Lie derivative of a function described by the expression Ψ with respect to the vector field described by the dynamics f is denoted $\mathcal{L}_f\,\Psi$ and gives the rate of change of the value of the expression Ψ as a function of the state variables S as the state evolves according to the dynamics $\dot{S} = f(S)$. In other words, the Lie derivative $\mathcal{L}_f\,\Psi$ is the *directional derivative* of the function Ψ along the vector field f.

The Lie derivative of the function Ψ can be computed from *partial derivatives* of the function Ψ with respect to each coordinate of the state vector:

$$\mathcal{L}_f\,\Psi(S) \;=\; \sum_{s \in S} (\partial\,\Psi/\partial\,s)\,f(s).$$

In our example, the state vector has dimension 2:

$$\begin{aligned}
\mathcal{L}_f\,\Psi(s_1, s_2) &= (\partial\,\Psi/\partial\,s_1)\,f(s_1) \;+\; (\partial\,\Psi/\partial\,s_2)\,f(s_2),\\
&= (14\,s_1 \;-\; 6\,s_2)\,(-7\,s_1 \;+\; s_2) \;+\; (-6\,s_1 \;+\; 56\,s_2)\,(8\,s_1 \;-\; 10\,s_2),\\
&= -146\,s_1^2 \;-\; 566\,s_2^2 \;+\; 564\,s_1\,s_2.
\end{aligned}$$

The third obligation for Ψ to be a barrier certificate is that the value of the function $\mathcal{L}_f\,\Psi$ should always be negative at all points belonging to the barrier. This ensures that for every state s on the barrier, $\Psi(s)$ decreases as the state s evolves according to the dynamics f.

Going back to our example, verify that the expression $-146\,s_1^2 \;-\; 566\,s_2^2 \;+\; 564\,s_1\,s_2$ always has a negative value not just for the states on the barrier.

A technical requirement for the argument outlined above is that the function described by the expression Ψ should be a *smooth* function. A smooth function is a function for which all derivatives, higher order derivatives as well as partial derivatives, exist. This ensures that the Lie derivative is well defined. In the example of figure 6.21, the function Ψ is a quadratic function and is smooth. Every polynomial function is a smooth function. A discontinuous function is not smooth, and this means that the barrier separating the initial states and the bad states cannot be a rectangle (or a polyhedron).

The proof technique for establishing invariants using barrier certificates is summarized below.

Theorem 6.6 [Safety Verification using Barrier Certificate] *Let H be a closed continuous-time component with state variables S, initialization Init, and dynamics given by $\dot{S} = f(S)$. To show that a state property φ is invariant in*

all reachable states of the system H, it suffices to find a smooth real-valued expression Ψ over the state variables, called a barrier certificate*, such that (1) if $Init(S)$ holds, then $\Psi(S) \leq 0$; (2) if $\varphi(S)$ does not hold, then $\Psi(S) > 0$; and (3) the Lie derivative of Ψ with respect to the vector field f is negative for all states belonging to the barrier: if $\Psi(S) = 0$ then $(\mathcal{L}_f\, \Psi)(S) < 0$.*

Proof. Let H be a closed continuous-time component with state variables S, initialization *Init*, and dynamics given by $\dot{S} = f(S)$. Let the real-valued expression Ψ over the state variables be a barrier certificate. That is, the value Ψ is non-positive in all initial states, the value of Ψ is positive in all states that violate φ, and the Lie derivative of Ψ with respect to the vector field f is negative for all states for which Ψ is 0.

We want to prove that every reachable state of the system H satisfies the property φ. The proof is by contradiction. Assume that there is a reachable state that does not satisfy φ. Then there exists a system response signal $\overline{S}$ and a time $t^* \in \texttt{time}$ such that $\overline{S}(0)$ is an initial state and $\overline{S}(t^*)$ does not satisfy the property φ. Let us define the function $\overline{\Psi} : \texttt{time} \mapsto \texttt{real}$ such that $\overline{\Psi}(t) = \Psi(\overline{S}(t))$. The function $\overline{\Psi}$ is a continuous function as it is a composition of two continuous functions.

Since the value of Ψ in every initial state is non-positive, we know that $\overline{\Psi}(0) \leq 0$. Since the value of Ψ in every unsafe state is positive, we know that $\overline{\Psi}(t^*) > 0$. From continuity of $\overline{\Psi}$, it follows that there exists a time τ such that $\overline{\Psi}(\tau) = 0$ and $\overline{\Psi}(t) > 0$ for all times t such that $\tau < t \leq t^*$. By definition, $(d/dt)\overline{\Psi}(\tau)$ equals the Lie derivative $(\mathcal{L}_f\, \Psi)$ evaluated at the state $s_\tau = \overline{S}(\tau)$. Since $\overline{\Psi}(\tau) = 0$, we know that $\Psi(s_\tau) = 0$, and by assumption the value of the Lie derivative $\mathcal{L}_f\, \Psi$ at the state s_τ on the barrier, must be negative.

For the continuous function $\overline{\Psi}$, the conditions (1) $\overline{\Psi}(\tau) = 0$, (2) $\overline{\Psi}(t) > 0$ for all times t such that $\tau < t \leq t^*$, and (3) $(d/dt)\overline{\Psi}(\tau) < 0$, cannot hold all at once, and this leads to contradiction. ■

Recipe for Choosing the Barrier Certificate

Theorem 6.6 gives us a general method for establishing safety of linear as well as non-linear continuous-time systems. The key to use this method effectively is to choose the function Ψ so that it satisfies all the necessary assumptions. We next outline a recipe for choosing this function for linear systems. Similar techniques are also used to establish stability of continuous-time systems and are known as *Lyapunov methods* in the literature.

Given an n-dimensional continuous-time system with dynamics $\dot{S} = A\,S$, we choose a *symmetric* n-dimensional matrix P and a constant k, let the function Ψ be defined as:

$$\Psi(S) \;=\; S^T\, P\, S \;+\; k.$$

Recall that a matrix is called symmetric if it equals its own transpose (that is, lower triangular entries equal upper triangular entries). The function Ψ

defined above is a quadratic expression over the state variables and is a smooth function. In this case, the dynamics of the state is defined by the matrix A. The Lie derivative of the expression Ψ with respect to this dynamics is given by:

$$\begin{aligned}(\mathcal{L}_A\,\Psi)(S) &= (d/dt)\,(S^T\,P\,S\ +\ k),\\ &= \dot{S}^T\,P\,S +\ S^T\,P\,\dot{S},\\ &= S^T\,A^T\,P\,S +\ S^T\,P\,A\,S,\\ &= S^T\,(A^T\,P\ +\ P\,A)\,S.\end{aligned}$$

Let us define the matrix P' to be $A^T\,P\ +\ P\,A$. Observe that the matrix P' is also a symmetric matrix. The Lie derivative of Ψ is given by the quadratic expression $S^T\,P'\,S$. The third requirement for Ψ being a barrier certificate now can be related to a well-understood form of matrices from linear algebra: if the matrix P' is *negative definite*, then for all states S the quantity $S^T\,P'\,S$ is guaranteed to be negative. If all the eigenvalues of the matrix P' are negative, then it is guaranteed to satisfy this condition. Thus, given the dynamics matrix A, the coefficients of the desired symmetric matrix P are chosen so that the resulting matrix P' has negative eigenvalues. Once we fix the matrix P, we need to choose the coefficient k in the definition of the barrier certificate Ψ so that it separates the initial states from the unsafe states.

Let us revisit the two-dimensional continuous-time system of figure 6.20. The dynamics matrix is given by

$$A\ =\ \begin{bmatrix}-7 & 1\\ 8 & -10\end{bmatrix}.$$

Let us choose the symmetric matrix P to be

$$P\ =\ \begin{bmatrix}7 & -3\\ -3 & 28\end{bmatrix}.$$

Observe that

$$S^T\,P\,S\ =\ \begin{bmatrix}s_1 & s_2\end{bmatrix}\begin{bmatrix}7 & -3\\ -3 & 28\end{bmatrix}\begin{bmatrix}s_1\\ s_2\end{bmatrix}\ =\ 7\,s_1^2\ -\ 6\,s_1\,s_2\ +\ 28\,s_2^2.$$

The coefficient k is chosen to be -320 so that the expression $S^T\,P\,S\ +\ k$ is negative in the initial rectangle and is positive in the unsafe states. The Lie derivative is of the form $S^T\,P'\,S$, where the symmetric matrix P' is given by:

$$\begin{bmatrix}-7 & 8\\ 1 & -10\end{bmatrix}\begin{bmatrix}7 & -3\\ -3 & 28\end{bmatrix}+\begin{bmatrix}7 & -3\\ -3 & 28\end{bmatrix}\begin{bmatrix}-7 & 1\\ 8 & -10\end{bmatrix}=\begin{bmatrix}-146 & 282\\ 282 & -566\end{bmatrix}.$$

Observe that the matrix P' is indeed symmetric. If we compute its eigenvalues, we get -5.2984 and -11.7016. This implies that the matrix P' is negative definite. This means that the expression $S^T\,P'\,S$, which equals $-146\,s_1^2\ -\ 566\,s_2^2\ +\ 564\,s_1\,s_2$, is always negative.

Exercise 6.24: For establishing safety of the system shown in figure 6.21, suppose the barrier function is $7\,s_1^2 - 6\,s_1\,s_2 + 28\,s_2^2 + k$, where k is a constant. What are the possible choices of k for which the resulting function is a barrier certificate? ■

Exercise 6.25 *: Suppose we replace the third condition for an expression Ψ to be a barrier certificate by the weaker condition "if $\Psi(S) = 0$ then $(\mathcal{L}_f\,\Psi)(S) \leq 0$" (that is, instead of requiring the Lie derivative to be negative for states on the barrier, it is required to be only non-positive). With this revised definition, from the existence of a barrier certificate, can we conclude that the property φ is an invariant? Prove or disprove your answer. ■

Exercise 6.26 *: This project concerns implementing a tool for symbolic reachability analysis of *one-dimensional* dynamical systems in MATLAB.

A closed interval of the set of real numbers is denoted $[a, b]$, where $a, b \in \texttt{real}$ and $a \leq b$ and corresponds the set of all real numbers between a and b. A state of a one-dimensional dynamical system is a real number. A set of such states can be naturally represented as a *union* of closed intervals. For example, $[0, 1] \cup [4, 5]$ is the set $\{x \in \texttt{real} \mid 0 \leq x \leq 1 \ \vee \ 4 \leq x \leq 5\}$. For the purpose of this project, let us fix the data type for regions to be such unions of closed intervals, each such region is of the form $A = \bigcup_i [a_i, b_i]$.

Part (a): Implement a programming library for computing with regions. The first step is to choose a data structure to represent regions. Your representation should ensure that the intervals are mutually disjoint. Make sure that the empty set is also represented correctly. Explain succinctly and rigorously your implementation of the following operations: (1) union of regions (the operation `Disj`), (2) difference of regions (the operation `Diff`), (3) inclusion test (the function $\texttt{IsSubset}(A, B)$ returns 1 exactly when the region A is a subset of the region B), (4) test for emptiness (the operation `IsEmpty`), (5) sum of regions (the function $\texttt{Sum}(A, B)$ returns the representation for the set $\{x + y \mid x \in A, y \in B\}$), (6) product of a region and a scalar (the function $\texttt{Product}(A, \alpha)$ returns the representation of the set $\{\alpha\, x \mid x \in A\}$), and (7) square of a region (the operation $\texttt{Square}(A)$ returns the representation of the set $\{x^2 \mid x \in A\}$).

Part (b): Consider a discretized representation of the dynamical system of the form $x_{k+1} = f(x_k, u_k)$, $k \geq 0$, where $x_k \in \texttt{real}$ is the state, u_k is the control input that belongs to the region U, and f expresses the dynamics of the system. The set of initial states is *Init*. Let $Reach_k$ be the set of reachable states at step k, that is, $Reach_k$ is the set of all states x such that there exists an execution of the system starting from some state $x_0 \in Init$ under some control inputs $u_0, u_1, ..., u_{k-1}$ from the region U and ending at the state $x_k = x$. The set *Reach* of all reachable states is $Reach = \bigcup_{k \geq 0} Reach_k$. Consider the breadth-first search algorithm to compute successive values of the regions $Reach_k$ up to a maximum number of iterations N. The algorithm should terminate as soon as there is no new state that can be added to *Reach* or when it iterates for N steps.

Implement the algorithm using the library you developed in part (a), provided that the sets *Init* and U are also regions, that is, unions of closed intervals, and the dynamics f can be described using the operations considered in part (a). Experiment your implementation using the following examples. In each case, report how the algorithm terminates and at which step, Output *Reach* (in its minimal form as a union of disjoint closed intervals).

(1) $x_{k+1} = -0.95x_k + u_k$, $Init = [1, 2]$, $U = [-0.1, 0.1]$, $N = 100$;
(2) $x_{k+1} = -0.96x_k + u_k$, $Init = [1, 2]$, $U = [-0.1, 0.1]$, $N = 100$;
(3) $x_{k+1} = -0.95x_k + u_k$, $Init = [1, 2]$, $U = [-0.2, 0.2]$, $N = 100$;
(4) $x_{k+1} = 0.5x_k^2 + u_k$, $Init = [1.8, 1.89]$, $U = [0, 0.1]$, $N = 40$;
(5) $x_{k+1} = 0.5x_k^2 + u_k$, $Init = [1.8, 1.9]$, $U = [0, 0.1]$, $N = 40$.

■

Bibliographic Notes

The core topics discussed in this chapter, namely, models of dynamical systems using differential equations, stability, linear systems, and analysis techniques using concepts from linear algebra, are studied in control theory for decades and appear in numerous textbooks. Our presentation is based on the textbook by Antsaklis and Michel [AM06], and some examples are also borrowed from [FPE02], [LV02], and [LS11].

For a detailed introduction to practical control design using PID controllers, see [AH95] (see also `www.mathworks.com` for tutorials on design of PID controllers using MATLAB).

The use of barrier certificates for verification of safety properties was introduced in [PJ04], and our discussion of this topic is based on the presentation in [Tab09] (see also [Pla10] and [TT09] for discussion on soundness, completeness, and variations of this proof technique).

The numerical calculations and the plots in this chapter are all obtained using the toolkit MATLAB (see [SH92] for an introduction to control design using MATLAB, and the tutorials at `www.mathworks.com` to learn how to use MATLAB).