Feedback Systems

An Introduction for Scientists and Engineers
SECOND EDITION

Karl Johan Åström
Richard M. Murray

Version v3.0f (30 Aug 2015)

This is the electronic edition of *Feedback Systems* and is available from http://www.cds.caltech.edu/~murray/FBS. Hardcover editions may be purchased from Princeton University Press, http://press.princeton.edu/titles/8701.html.

PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Chapter Four
Examples

> *... Don't apply any model until you understand the simplifying assumptions on which it is based, and you can test their validity. Catch phrase: use only as directed. Don't limit yourself to a single model: More than one model may be useful for understanding different aspects of the same phenomenon. Catch phrase: legalize polygamy."*
>
> Saul Golomb, "Mathematical Models—Uses and Limitations," 1970 [Gol70].

In this chapter we present a collection of examples spanning many different fields of science and engineering. These examples are used throughout the text and in exercises to illustrate different concepts.

4.1 Cruise Control

The cruise control system of a car is a common feedback system encountered in everyday life. The system attempts to maintain a constant velocity in the presence of disturbances primarily caused by changes in the slope of a road. The controller compensates for these unknowns by measuring the speed of the car and adjusting the throttle appropriately.

To model the system we start with the block diagram in Figure 4.1. Let v be the speed of the car and v_r the desired (reference) speed. The controller, which typically is of the proportional-integral (PI) type described briefly in Chapter 1, receives the signals v and v_r and generates a control signal u that is sent to an actuator that controls the throttle position. The throttle in turn controls the torque T delivered by the engine, which is transmitted through the gears and the wheels, generating a force F that moves the car. There are disturbance forces F_d due to variations in the slope of the road, the rolling resistance and aerodynamic forces. The cruise controller also has a human–machine interface that allows the driver to set and modify the desired speed. There are also functions that disconnect the cruise control when the brake is touched.

The system has many individual components—actuator, engine, transmission, wheels and car body—and a detailed model can be very complicated. In spite of this, the model required to design the cruise controller can be quite simple.

To develop a mathematical model we start with a force balance for the car body. Let v be the speed of the car, m the total mass (including passengers), F the force generated by the contact of the wheels with the road, and F_d the disturbance force due to gravity, friction and aerodynamic drag. The equation of motion of the car is

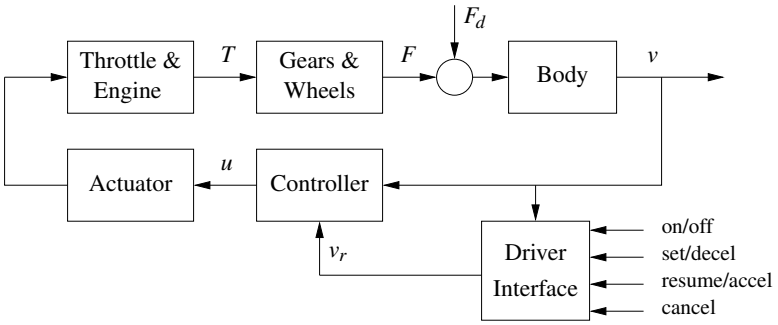


Figure 4.1: Block diagram of a cruise control system for an automobile. The throttle-controlled engine generates a torque T that is transmitted to the ground through the gearbox and wheels. Combined with the external forces from the environment, such as aerodynamic drag and gravitational forces on hills, the net force causes the car to move. The velocity of the car v is measured by a control system that adjusts the throttle through an actuation mechanism. A driver interface allows the system to be turned on and off and the reference speed v_r to be established.

simply

$$m\frac{dv}{dt} = F - F_d. \tag{4.1}$$

The force F is generated by the engine, whose torque is proportional to the rate of fuel injection, which is itself proportional to a control signal $0 \le u \le 1$ that controls the throttle position. The torque also depends on engine speed ω. A simple representation of the torque at full throttle is given by the torque curve

$$T(\omega) = T_m \left(1 - \beta\left(\frac{\omega}{\omega_m} - 1\right)^2\right), \tag{4.2}$$

where the maximum torque T_m is obtained at engine speed ω_m. Typical parameters are $T_m = 190$ Nm, $\omega_m = 420$ rad/s (about 4000 RPM) and $\beta = 0.4$. Let n be the gear

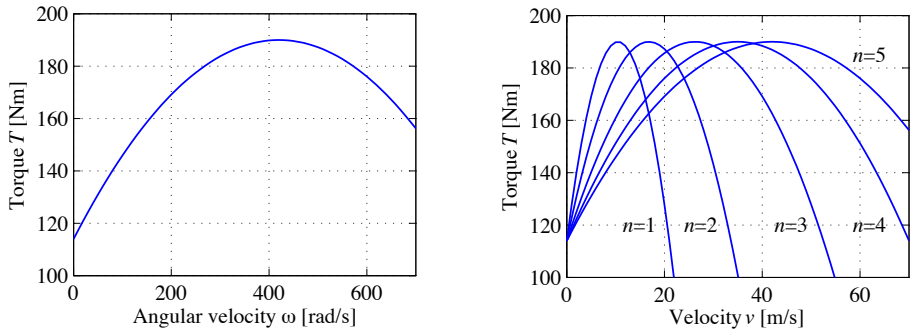
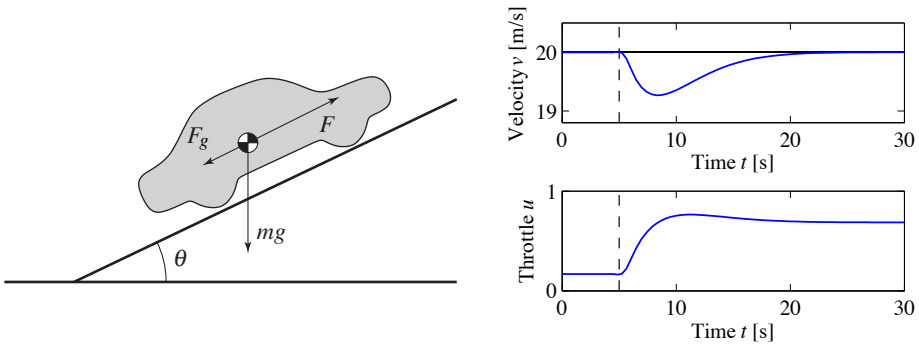


Figure 4.2: Torque curves for typical car engine. The graph on the left shows the torque generated by the engine as a function of the angular velocity of the engine, while the curve on the right shows torque as a function of car speed for different gears.



(a) Effect of gravitational forces (b) Closed loop response

Figure 4.3: Car with cruise control encountering a sloping road. A schematic diagram is shown in (a), and (b) shows the response in speed and throttle when a slope of 4° is encountered. The hill is modeled as a net change of 4° in hill angle θ, with a linear change in the angle between $t = 5$ and $t = 6$. The PI controller has proportional gain is $k_p = 0.5$, and the integral gain is $k_i = 0.1$.

ratio and r the wheel radius. The engine speed is related to the velocity through the expression

$$\omega = \frac{n}{r} v =: \alpha_n v,$$

and the driving force can be written as

$$F = \frac{nu}{r} T(\omega) = \alpha_n u T(\alpha_n v).$$

Typical values of α_n for gears 1 through 5 are $\alpha_1 = 40$, $\alpha_2 = 25$, $\alpha_3 = 16$, $\alpha_4 = 12$ and $\alpha_5 = 10$. The inverse of α_n has a physical interpretation as the *effective wheel radius*. Figure 4.2 shows the torque as a function of engine speed and vehicle speed. The figure shows that the effect of the gear is to "flatten" the torque curve so that an almost full torque can be obtained almost over the whole speed range.

The disturbance force F_d has three major components: F_g, the forces due to gravity; F_r, the forces due to rolling friction; and F_a, the aerodynamic drag. Letting the slope of the road be θ, gravity gives the force $F_g = mg \sin\theta$, as illustrated in Figure 4.3a, where $g = 9.8$ m/s^2 is the gravitational constant. A simple model of rolling friction is

$$F_r = mgC_r \operatorname{sgn}(v),$$

where C_r is the coefficient of rolling friction and $\operatorname{sgn}(v)$ is the sign of v (± 1) or zero if $v = 0$. A typical value for the coefficient of rolling friction is $C_r = 0.01$. Finally, the aerodynamic drag is proportional to the square of the speed:

$$F_a = \frac{1}{2} \rho C_d A v^2,$$

where ρ is the density of air, C_d is the shape-dependent aerodynamic drag coefficient and A is the frontal area of the car. Typical parameters are $\rho = 1.3$ kg/m^3,

$C_d = 0.32$ and $A = 2.4\ \text{m}^2$.

Summarizing, we find that the car can be modeled by

$$m\frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mgC_r \operatorname{sgn}(v) - \frac{1}{2}\rho C_d A v^2 - mg\sin\theta, \tag{4.3}$$

where the function T is given by equation (4.2). The model (4.3) is a dynamical system of first order. The state is the car velocity v, which is also the output. The input is the signal u that controls the throttle position, and the disturbance is the force F_d, which depends on the slope of the road. The system is nonlinear because of the torque curve, the gravity term and the nonlinear character of rolling friction and aerodynamic drag. There can also be variations in the parameters; e.g., the mass of the car depends on the number of passengers and the load being carried in the car.

We add to this model a feedback controller that attempts to regulate the speed of the car in the presence of disturbances. We shall use a proportional-integral controller, which has the form

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau.$$

This controller can itself be realized as an input/output dynamical system by defining a controller state z and implementing the differential equation

$$\frac{dz}{dt} = v_r - v, \qquad u = k_p(v_r - v) + k_i z, \tag{4.4}$$

where v_r is the desired (reference) speed. As discussed briefly in Section 1.5, the integrator (represented by the state z) ensures that in steady state the error will be driven to zero, even when there are disturbances or modeling errors. (The design of PI controllers is the subject of Chapter 11.) Figure 4.3b shows the response of the closed loop system, consisting of equations (4.3) and (4.4), when it encounters a hill. The figure shows that even if the hill is so steep that the throttle changes from 0.17 to almost full throttle, the largest speed error is less than 1 m/s, and the desired velocity is recovered after 20 s.

Many approximations were made when deriving the model (4.3). It may seem surprising that such a seemingly complicated system can be described by the simple model (4.3). It is important to make sure that we restrict our use of the model to the uncertainty lemon conceptualized in Figure 3.16b. The model is not valid for very rapid changes of the throttle because we have ignored the details of the engine dynamics, neither is it valid for very slow changes because the properties of the engine will change over the years. Nevertheless the model is very useful for the design of a cruise control system. As we shall see in later chapters, the reason for this is the inherent robustness of feedback systems: even if the model is not perfectly accurate, we can use it to design a controller and make use of the feedback in the controller to manage the uncertainty in the system.

The cruise control system also has a human–machine interface that allows the driver to communicate with the system. There are many different ways to imple-

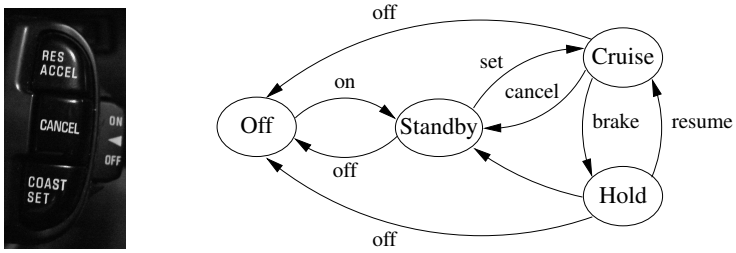


Figure 4.4: Finite state machine for cruise control system. The figure on the left shows some typical buttons used to control the system. The controller can be in one of four modes, corresponding to the nodes in the diagram on the right. Transition between the modes is controlled by pressing one of the five buttons on the cruise control interface: on, off, set, resume or cancel.

ment this system; one version is illustrated in Figure 4.4. The system has four buttons: on-off, set/decelerate, resume/accelerate and cancel. The operation of the system is governed by a finite state machine that controls the modes of the PI controller and the reference generator. Implementation of controllers and reference generators will be discussed more fully in Chapter 11.

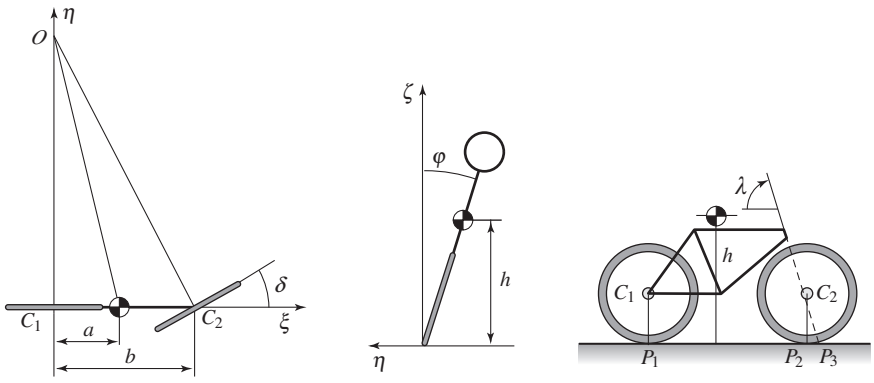
The use of control in automotive systems goes well beyond the simple cruise control system described here. Applications include emissions control, traction control, power control (especially in hybrid vehicles) and adaptive cruise control. Many automotive applications are discussed in detail in the book by Kiencke and Nielsen [KN00] and in the survey papers by Powers et al. [BP96, PN00].

4.2 Bicycle Dynamics

The bicycle is an interesting dynamical system with the feature that one of its key properties is due to a feedback mechanism that is created by the design of the front fork. A detailed model of a bicycle is complex because the system has many degrees of freedom and the geometry is complicated. However, a great deal of insight can be obtained from simple models.

To derive the equations of motion we assume that the bicycle rolls on the horizontal xy plane. Introduce a coordinate system that is fixed to the bicycle with the ξ-axis through the contact points of the wheels with the ground, the η-axis horizontal and the ζ-axis vertical, as shown in Figure 4.5. Let v_0 be the velocity of the bicycle at the rear wheel, b the wheel base, φ the tilt angle and δ the steering angle. The coordinate system rotates around the point O with the angular velocity $\omega = v_0\delta/b$, and an observer fixed to the bicycle experiences forces due to the motion of the coordinate system.

The tilting motion of the bicycle is similar to an inverted pendulum, as shown in the rear view in Figure 4.5b. To model the tilt, consider the rigid body obtained when the wheels, the rider and the front fork assembly are fixed to the bicycle frame. Let m be the total mass of the system, J the moment of inertia of this body



(a) top view (b) rear view (c) side view

Figure 4.5: Schematic views of a bicycle. The steering angle is δ, and the roll angle is φ. The center of mass has height h and distance a from a vertical through the contact point P_1 of the rear wheel. The wheel base b is the distance between P_1 and P_2, and the trail c is the distance between P_2 and P_3.

with respect to the ξ-axis and D the product of inertia with respect to the $\xi\zeta$ axes. Furthermore, let the ξ and ζ coordinates of the center of mass with respect to the rear wheel contact point, P_1, be a and h, respectively. We have $J \approx mh^2$ and $D = mah$. The torques acting on the system are due to gravity and centripetal action. Assuming that the steering angle δ is small, the equation of motion becomes

$$J\frac{d^2\varphi}{dt^2} - \frac{Dv_0}{b}\frac{d\delta}{dt} = mgh\sin\varphi + \frac{mv_0^2h}{b}\delta. \tag{4.5}$$

The term $mgh\sin\varphi$ is the torque generated by gravity. The terms containing δ and its derivative are the torques generated by steering, with the term $(Dv_0/b)\,d\delta/dt$ due to inertial forces and the term $(mv_0^2h/b)\,\delta$ due to centripetal forces.

The steering angle is influenced by the torque the rider applies to the handle bar. Because of the tilt of the steering axis and the shape of the front fork, the contact point of the front wheel with the road P_2 is behind the axis of rotation of the front wheel assembly, as shown in Figure 4.5c. The distance c between the contact point of the front wheel P_2 and the projection of the axis of rotation of the front fork assembly P_3 is called the *trail*. The steering properties of a bicycle depend critically on the trail. A large trail increases stability but makes the steering less agile.

A consequence of the design of the front fork is that the steering angle δ is influenced both by steering torque T and by the tilt of the frame φ. This means that a bicycle with a front fork is a *feedback system* as illustrated by the block diagram in Figure 4.6. The steering angle δ influences the tilt angle φ, and the tilt angle influences the steering angle, giving rise to the circular causality that is characteristic of reasoning about feedback. For a front fork with a positive trail, the bicycle will steer into the lean, creating a centrifugal force that attempts to diminish the lean. Under certain conditions, the feedback can actually stabilize the

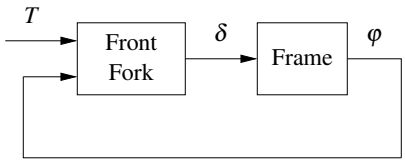


Figure 4.6: Block diagram of a bicycle with a front fork. The steering torque applied to the handlebars is T, the roll angle is φ and the steering angle is δ. Notice that the front fork creates a feedback from the roll angle φ to the steering angle δ that under certain conditions can stabilize the system.

bicycle. A crude empirical model is obtained by assuming that the block B can be modeled as the static system

$$\delta = k_1 T - k_2 \varphi. \tag{4.6}$$

This model neglects the dynamics of the front fork, the tire–road interaction and the fact that the parameters depend on the velocity. A more accurate model, called the *Whipple model*, is obtained using the rigid-body dynamics of the front fork and the frame. Assuming small angles, this model becomes

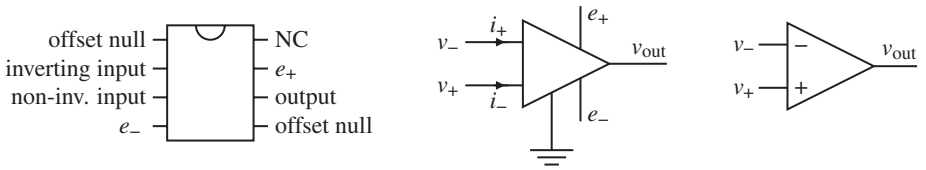
$$M \begin{pmatrix} \ddot{\varphi} \\ \ddot{\delta} \end{pmatrix} + C v_0 \begin{pmatrix} \dot{\varphi} \\ \dot{\delta} \end{pmatrix} + (K_0 + K_2 v_0^2) \begin{pmatrix} \varphi \\ \delta \end{pmatrix} = \begin{pmatrix} 0 \\ T \end{pmatrix}, \tag{4.7}$$

where the elements of the 2×2 matrices M, C, K_0 and K_2 depend on the geometry and the mass distribution of the bicycle. Note that this has a form somewhat similar to that of the spring–mass system introduced in Chapter 3 and the balance system in Example 3.1. Even this more complex model is inaccurate because the interaction between the tire and the road is neglected; taking this into account requires two additional state variables. Again, the uncertainty lemon in Figure 3.16b provides a framework for understanding the validity of the model under these assumptions.

Interesting presentations on the development of the bicycle are given in the books by D. Wilson [Wil04] and Herlihy [Her04]. The model (4.7) was presented in a paper by Whipple in 1899 [Whi99]. More details on bicycle modeling are given in the paper [ÅKL05], which has many references.

4.3 Operational Amplifier Circuits

An operational amplifier (op amp) is a modern implementation of Black's feedback amplifier. It is a universal component that is widely used for instrumentation, control and communication. It is also a key element in analog computing. Schematic diagrams of the operational amplifier are shown in Figure 4.7. The amplifier has one inverting input (v_-), one noninverting input (v_+) and one output (v_{out}). There are also connections for the supply voltages, e_- and e_+, and a zero adjustment (offset null). A simple model is obtained by assuming that the input currents i_-



(a) Chip pinout (b) Full schematic (c) Simple view

Figure 4.7: An operational amplifier and two schematic diagrams. (a) The amplifier pin connections on an integrated circuit chip. (b) A schematic with all connections. (c) Only the signal connections.

and i_+ are zero and that the output is given by the static relation

$$v_{out} = \text{sat}_{(v_{min},v_{max})}\big(k(v_+ - v_-)\big), \tag{4.8}$$

where sat denotes the saturation function

$$\text{sat}_{(a,b)}(x) = \begin{cases} a & \text{if } x < a \\ x & \text{if } a \le x \le b \\ b & \text{if } x > b. \end{cases} \tag{4.9}$$

We assume that the gain k is large, in the range of 10^6–10^8, and the voltages v_{min} and v_{max} satisfy

$$e_- \le v_{min} < v_{max} \le e_+$$

and hence are in the range of the supply voltages. More accurate models are obtained by replacing the saturation function with a smooth function as shown in Figure 4.8. For small input signals the amplifier characteristic (4.8) is linear:

$$v_{out} = k(v_+ - v_-) =: -kv. \tag{4.10}$$

Since the open loop gain k is very large, the range of input signals where the system is linear is very small.

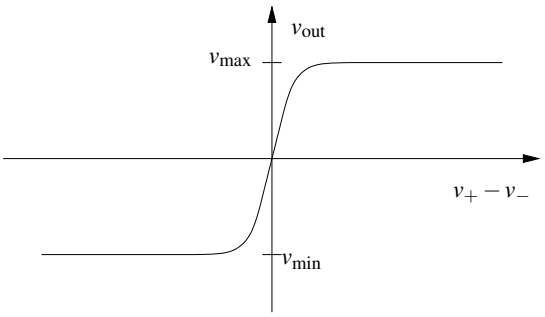
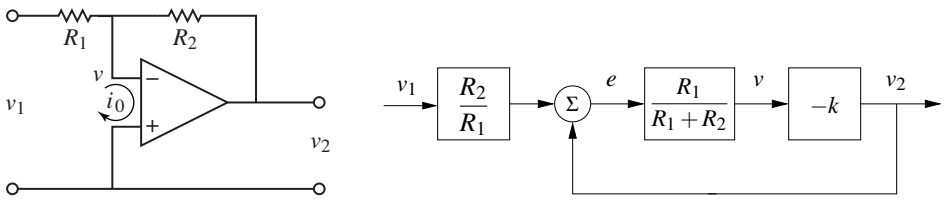


Figure 4.8: Input/output characteristics of an operational amplifier. The differential input is given by $v_+ - v_-$. The output voltage is a linear function of the input in a small range around 0, with saturation at v_{min} and v_{max}. In the linear regime the op amp has high gain.



(a) Amplifier circuit (b) Block diagram

Figure 4.9: Stable amplifier using an op amp. The circuit (a) uses negative feedback around an operational amplifier and has a corresponding block diagram (b). The resistors R_1 and R_2 determine the gain of the amplifier.

A simple amplifier is obtained by arranging feedback around the basic operational amplifier as shown in Figure 4.9a. To model the feedback amplifier in the linear range, we assume that the current $i_0 = i_- + i_+$ is zero and that the gain of the amplifier is so large that the voltage $v = v_- - v_+$ is also zero. It follows from Ohm's law that the currents through resistors R_1 and R_2 are given by

$$\frac{v_1}{R_1} = -\frac{v_2}{R_2},$$

and hence the closed loop gain of the amplifier is

$$\frac{v_2}{v_1} = -k_{\text{cl}}, \qquad \text{where} \quad k_{\text{cl}} = \frac{R_2}{R_1}. \tag{4.11}$$

A more accurate model is obtained by continuing to neglect the current i_0 but assuming that the voltage v is small but not negligible. The current balance is then

$$\frac{v_1 - v}{R_1} = \frac{v - v_2}{R_2}. \tag{4.12}$$

Assuming that the amplifier operates in the linear range and using equation (4.10), the gain of the closed loop system becomes

$$k_{\text{cl}} = -\frac{v_2}{v_1} = \frac{R_2}{R_1}\,\frac{kR_1}{R_1 + R_2 + kR_1} \tag{4.13}$$

If the open loop gain k of the operational amplifier is large, the closed loop gain k_{cl} is the same as in the simple model given by equation (4.11). Notice that the closed loop gain depends only on the passive components and that variations in k have only a marginal effect on the closed loop gain. For example if $k = 10^6$ and $R_2/R_1 = 100$, a variation of k by 100% gives only a variation of 0.01% in the closed loop gain. The drastic reduction in sensitivity is a nice illustration of how feedback can be used to make precise systems from uncertain components. In this particular case, feedback is used to trade high gain and low robustness for low gain and high robustness. Equation (4.13) was the formula that inspired Black when he invented the feedback amplifier [Bla34] (see the quote at the beginning of Chapter 13).

It is instructive to develop a block diagram for the feedback amplifier in Figure 4.9a. To do this we will represent the pure amplifier with input v and output v_2

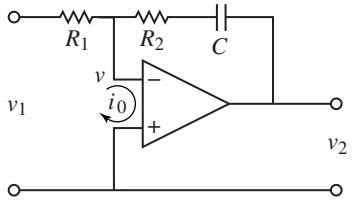


Figure 4.10: Circuit diagram of a PI controller obtained by feedback around an operational amplifier. The capacitor C is used to store charge and represents the integral of the input.

as one block. To complete the block diagram, we must describe how v depends on v_1 and v_2. Solving equation (4.12) for v gives

$$v = \frac{R_2}{R_1+R_2}v_1 + \frac{R_1}{R_1+R_2}v_2 = \frac{R_1}{R_1+R_2}\left(\frac{R_2}{R_1}v_1 + v_2\right),$$

and we obtain the block diagram shown in Figure 4.9b. The diagram clearly shows that the system has feedback and that the gain from v_2 to v is $R_1/(R_1+R_2)$, which can also be read from the circuit diagram in Figure 4.9a. If the loop is stable and the gain of the amplifier is large, it follows that the error e is small, and we find that $v_2 = -(R_2/R_1)v_1$. Notice that the resistor R_1 appears in two blocks in the block diagram. This situation is typical in electrical circuits, and it is one reason why block diagrams are not always well suited for some types of physical modeling.

The simple model of the amplifier given by equation (4.10) provides qualitative insight, but it neglects the fact that the amplifier is a dynamical system. A more realistic model is

$$\frac{dv_{\text{out}}}{dt} = -av_{\text{out}} - bv. \tag{4.14}$$

The parameter b that has dimensions of frequency and is called the *gain-bandwidth product* of the amplifier. Whether a more complicated model is used depends on the questions to be answered and the required size of the uncertainty lemon. The model (4.14) is still not valid for very high or very low frequencies since drift causes deviations at low frequencies and there are additional dynamics that appear at frequencies close to b. The model is also not valid for large signals—an upper limit is given by the voltage of the power supply, typically in the range of 5–10 V—neither is it valid for very low signals because of electrical noise. These effects can be added, if needed, but increase the complexity of the analysis.

The operational amplifier is very versatile, and many different systems can be built by combining it with resistors and capacitors. In fact, any linear system can be implemented by combining operational amplifiers with resistors and capacitors. Exercise 4.5 shows how a second-order oscillator is implemented, and Figure 4.10 shows the circuit diagram for an analog proportional-integral controller. To develop a simple model for the circuit we assume that the current i_0 is zero and that the open loop gain k is so large that the input voltage v is negligible. The current i through the capacitor is $i = Cdv_c/dt$, where v_c is the voltage across the capacitor.

Since the same current goes through the resistor R_1, we get

$$i = \frac{v_1}{R_1} = C\frac{dv_c}{dt},$$

which implies that

$$v_c(t) = \frac{1}{C}\int i(t)\,dt = \frac{1}{R_1 C}\int_0^t v_1(\tau)d\tau.$$

The output voltage is thus given by

$$v_2(t) = -R_2 i - v_c = -\frac{R_2}{R_1}v_1(t) - \frac{1}{R_1 C}\int_0^t v_1(\tau)d\tau,$$

which is the input/output relation for a PI controller.

The development of operational amplifiers was pioneered by Philbrick [Lun05, Phi48], and their usage is described in many textbooks (e.g., [CD75]). Good information is also available from suppliers [Jun02, Man02].

4.4 Computing Systems and Networks

The application of feedback to computing systems follows the same principles as the control of physical systems, but the types of measurements and control inputs that can be used are somewhat different. Measurements (sensors) are typically related to resource utilization in the computing system or network and can include quantities such as the processor load, memory usage or network bandwidth. Control variables (actuators) typically involve setting limits on the resources available to a process. This might be done by controlling the amount of memory, disk space or time that a process can consume, turning on or off processing, delaying availability of a resource or rejecting incoming requests to a server process. Process modeling for networked computing systems is also challenging, and empirical models based on measurements are often used when a first-principles model is not available.

Web Server Control

Web servers respond to requests from the Internet and provide information in the form of web pages. Modern web servers start multiple processes to respond to requests, with each process assigned to a single source until no further requests are received from that source for a predefined period of time. Processes that are idle become part of a pool that can be used to respond to new requests. To provide a fast response to web requests, it is important that the web server processes do not overload the server's computational capabilities or exhaust its memory. Since other processes may be running on the server, the amount of available processing power and memory is uncertain, and feedback can be used to provide good performance in the presence of this uncertainty.

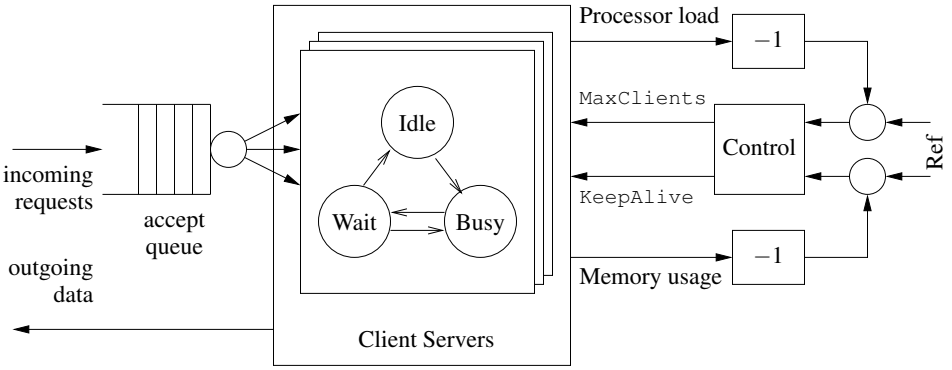


Figure 4.11: Feedback control of a web server. Connection requests arrive on an input queue, where they are sent to a server process. A finite state machine keeps track of the state of the individual server processes and responds to requests. A control algorithm can modify the server's operation by controlling parameters that affect its behavior, such as the maximum number of requests that can be serviced at a single time (`MaxClients`) or the amount of time that a connection can remain idle before it is dropped (`KeepAlive`).

Figure 4.11 illustrates the use of feedback to modulate the operation of an Apache web server. The web server operates by placing incoming connection requests on a queue and then starting a subprocess to handle requests for each accepted connection. This subprocess responds to requests from a given connection as they come in, alternating between a `Busy` state and a `Wait` state. (Keeping the subprocess active between requests is known as the *persistence* of the connection and provides a substantial reduction in latency to requests for multiple pieces of information from a single site.) If no requests are received for a sufficiently long period of time, controlled by the `KeepAlive` parameter, then the connection is dropped and the subprocess enters an `Idle` state, where it can be assigned another connection. A maximum of `MaxClients` simultaneous requests will be served, with the remainder remaining on the incoming request queue.

The parameters that control the server represent a trade-off between performance (how quickly requests receive a response) and resource usage (the amount of processing power and memory used by the server). Increasing the `MaxClients` parameter allows connection requests to be pulled off of the queue more quickly but increases the amount of processing power and memory usage that is required. Increasing the `KeepAlive` timeout means that individual connections can remain idle for a longer period of time, which decreases the processing load on the machine but increases the length of the queue (and hence the amount of time required for a user to initiate a connection). Successful operation of a busy server requires a proper choice of these parameters, often based on trial and error.

To model the dynamics of this system in more detail, we create a discrete-time model with states given by the average processor load x_{cpu} and the percentage memory usage x_{mem}. The inputs to the system are taken as the maximum number of clients u_{mc} and the keep-alive time u_{ka}. If we assume a linear model around the

equilibrium point, the dynamics can be written as

$$\begin{pmatrix} x_{\text{cpu}}[k+1] \\ x_{\text{mem}}[k+1] \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_{\text{cpu}}[k] \\ x_{\text{mem}}[k] \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} u_{\text{ka}}[k] \\ u_{\text{mc}}[k] \end{pmatrix}, \quad (4.15)$$

where the coefficients of the A and B matrices can be determined based on empirical measurements or detailed modeling of the web server's processing and memory usage. Using system identification, Diao et al. [DGH+02, HDPT04] identified the linearized dynamics as

$$A = \begin{pmatrix} 0.54 & -0.11 \\ -0.026 & 0.63 \end{pmatrix}, \qquad B = \begin{pmatrix} -85 & 4.4 \\ -2.5 & 2.8 \end{pmatrix} \times 10^{-4},$$

where the system was linearized about the equilibrium point

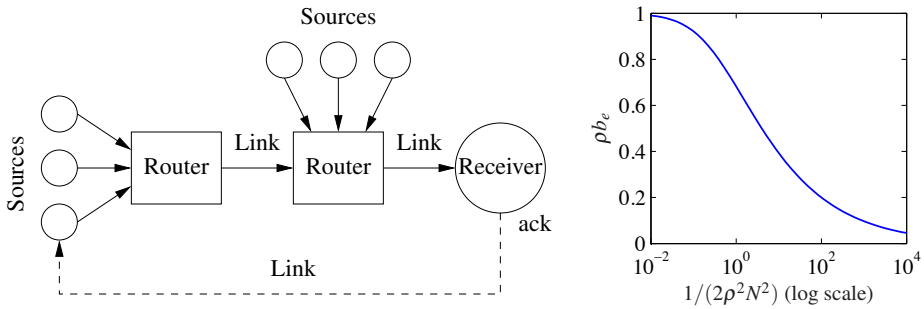
$$x_{\text{cpu}} = 0.58, \qquad u_{\text{ka}} = 11 \text{ s}, \qquad x_{\text{mem}} = 0.55, \qquad u_{\text{mc}} = 600.$$

This model shows the basic characteristics that were described above. Looking first at the B matrix, we see that increasing the `KeepAlive` timeout (first column of the B matrix) decreases both the processor usage and the memory usage since there is more persistence in connections and hence the server spends a longer time waiting for a connection to close rather than taking on a new active connection. The `MaxClients` connection increases both the processing and memory requirements. Note that the largest effect on the processor load is the `KeepAlive` timeout. The A matrix tells us how the processor and memory usage evolve in a region of the state space near the equilibrium point. The diagonal terms describe how the individual resources return to equilibrium after a transient increase or decrease. The off-diagonal terms show that there is coupling between the two resources, so that a change in one could cause a later change in the other.

Although this model is very simple, we will see in later examples that it can be used to modify the parameters controlling the server in real time and provide robustness with respect to uncertainties in the load on the machine. Similar types of mechanisms have been used for other types of servers. It is important to remember the assumptions on the model and their role in determining when the model is valid. In particular, since we have chosen to use average quantities over a given sample time, the model will not provide an accurate representation for high-frequency phenomena.

Congestion Control

The Internet was created to obtain a large, highly decentralized, efficient and expandable communication system. The system consists of a large number of interconnected gateways. A message is split into several packets which are transmitted over different paths in the network, and the packages are rejoined to recover the message at the receiver. An acknowledgment ("ack") message is sent back to the sender when a packet is received. The operation of the system is governed by a simple but powerful decentralized control structure that has evolved over time.



(a) Block diagram (b) Operating point

Figure 4.12: Internet congestion control. (a) Source computers send information to routers, which forward the information to other routers that eventually connect to the receiving computer. When a packet is received, an acknowledgment packet is sent back through the routers (not shown). The routers buffer information received from the sources and send the data across the outgoing link. (b) The equilibrium buffer size b_e for a set of N identical computers sending packets through a single router with drop probability ρ.

The system has two control mechanisms called *protocols*: the Transmission Control Protocol (TCP) for end-to-end network communication and the Internet Protocol (IP) for routing packets and for host-to-gateway or gateway-to-gateway communication. The current protocols evolved after some spectacular congestion collapses occurred in the mid 1980s, when throughput unexpectedly could drop by a factor of 1000 [Jac95]. The control mechanism in TCP is based on conserving the number of packets in the loop from the sender to the receiver and back to the sender. The sending rate is increased exponentially when there is no congestion, and it is dropped to a low level when there is congestion.

To derive an overall model for congestion control, we model three separate elements of the system: the rate at which packets are sent by individual sources (computers), the dynamics of the queues in the links (routers) and the admission control mechanism for the queues. Figure 4.12a is a block diagram of the system.

The current source control mechanism on the Internet is a protocol known as TCP/Reno [LPD02]. This protocol operates by sending packets to a receiver and waiting to receive an acknowledgment from the receiver that the packet has arrived. If no acknowledgment is sent within a certain timeout period, the packet is retransmitted. To avoid waiting for the acknowledgment before sending the next packet, Reno transmits multiple packets up to a fixed *window* around the latest packet that has been acknowledged. If the window length is chosen properly, packets at the beginning of the window will be acknowledged before the source transmits packets at the end of the window, allowing the computer to continuously stream packets at a high rate.

To determine the size of the window to use, TCP/Reno uses a feedback mechanism in which (roughly speaking) the window size is increased by 1 every time a packet is acknowledged and the window size is cut in half when packets are lost. This mechanism allows a dynamic adjustment of the window size in which each

computer acts in a greedy fashion as long as packets are being delivered but backs off quickly when congestion occurs.

A model for the behavior of the source can be developed by describing the dynamics of the window size. Suppose we have N computers and let w_i be the current window size (measured in number of packets) for the ith computer. Let q_i represent the end-to-end probability that a packet will be dropped someplace between the source and the receiver. We can model the dynamics of the window size by the differential equation

$$\frac{dw_i}{dt} = (1-q_i)\frac{r_i(t-\tau_i)}{w_i} + q_i(-\frac{w_i}{2}r_i(t-\tau_i)), \qquad r_i = \frac{w_i}{\tau_i}, \tag{4.16}$$

where τ_i is the end-to-end transmission time for a packet to reach is destination and the acknowledgment to be sent back and r_i is the resulting rate at which packets are cleared from the list of packets that have been received. The first term in the dynamics represents the increase in window size when a packet is received, and the second term represents the decrease in window size when a packet is lost. Notice that r_i is evaluated at time $t-\tau_i$, representing the time required to receive additional acknowledgments.

The link dynamics are controlled by the dynamics of the router queue and the admission control mechanism for the queue. Assume that we have L links in the network and use l to index the individual links. We model the queue in terms of the current number of packets in the router's buffer b_l and assume that the router can contain a maximum of $b_{l,\max}$ packets and transmits packets at a rate c_l, equal to the capacity of the link. The buffer dynamics can then be written as

$$\frac{db_l}{dt} = s_l - c_l, \qquad s_l = \sum_{\{i:\ l\in L_i\}} r_i(t-\tau_{li}^f), \tag{4.17}$$

where L_i is the set of links that are being used by source i, τ_{li}^f is the time it takes a packet from source i to reach link l and s_l is the total rate at which packets arrive at link l.

The admission control mechanism determines whether a given packet is accepted by a router. Since our model is based on the average quantities in the network and not the individual packets, one simple model is to assume that the probability that a packet is dropped depends on how full the buffer is: $p_l = m_l(b_l, b_{\max})$. For simplicity, we will assume for now that $p_l = \rho_l b_l$ (see Exercise 4.6 for a more detailed model). The probability that a packet is dropped at a given link can be used to determine the end-to-end probability that a packet is lost in transmission:

$$q_i = 1 - \prod_{l\in L_i}(1-p_l) \approx \sum_{l\in L_i} p_l(t-\tau_{li}^b), \tag{4.18}$$

where τ_{li}^b is the backward delay from link l to source i and the approximation is valid as long as the individual drop probabilities are small. We use the backward delay since this represents the time required for the acknowledgment packet to be received by the source.

Together, equations (4.16), (4.17) and (4.18) represent a model of congestion control dynamics. We can obtain substantial insight by considering a special case in which we have N identical sources and 1 link. In addition, we assume for the moment that the forward and backward time delays can be ignored, in which case the dynamics can be reduced to the form

$$\frac{dw_i}{dt} = \frac{1}{\tau} - \frac{\rho c(2+w_i^2)}{2}, \qquad \frac{db}{dt} = \sum_{i=1}^{N} \frac{w_i}{\tau} - c, \qquad \tau = \frac{b}{c}, \tag{4.19}$$

where $w_i \in \mathbb{R}$, $i = 1,\ldots,N$, are the window sizes for the sources of data, $b \in \mathbb{R}$ is the current buffer size of the router, ρ controls the rate at which packets are dropped and c is the capacity of the link connecting the router to the computers. The variable τ represents the amount of time required for a packet to be processed by a router, based on the size of the buffer and the capacity of the link. Substituting τ into the equations, we write the state space dynamics as

$$\frac{dw_i}{dt} = \frac{c}{b} - \rho c\left(1 + \frac{w_i^2}{2}\right), \qquad \frac{db}{dt} = \sum_{i=1}^{N} \frac{cw_i}{b} - c. \tag{4.20}$$

More sophisticated models can be found in [HMTG00, LPD02].

The nominal operating point for the system can be found by setting $\dot{w}_i = \dot{b} = 0$:

$$0 = \frac{c}{b} - \rho c\left(1 + \frac{w_i^2}{2}\right), \qquad 0 = \sum_{i=1}^{N} \frac{cw_i}{b} - c.$$

Exploiting the fact that all of the source dynamics are identical, it follows that all of the w_i should be the same, and it can be shown that there is a unique equilibrium satisfying the equations

$$w_{i,e} = \frac{b_e}{N} = \frac{c\tau_e}{N}, \qquad \frac{1}{2\rho^2 N^2}(\rho b_e)^3 + (\rho b_e) - 1 = 0. \tag{4.21}$$

The solution for the second equation is a bit messy but can easily be determined numerically. A plot of its solution as a function of $1/(2\rho^2N^2)$ is shown in Figure 4.12b. We also note that at equilibrium we have the following additional equalities:

$$\tau_e = \frac{b_e}{c} = \frac{Nw_e}{c}, \qquad q_e = Np_e = N\rho b_e, \qquad r_e = \frac{w_e}{\tau_e}. \tag{4.22}$$

Figure 4.13 shows a simulation of 60 sources communicating across a single link, with 20 sources dropping out at $t = 500$ ms and the remaining sources increasing their rates (window sizes) to compensate. Note that the buffer size and window sizes automatically adjust to match the capacity of the link.

A comprehensive treatment of computer networks is given in the textbook by Tannenbaum [Tan96]. A good presentation of the ideas behind the control principles for the Internet is given by one of its designers, Van Jacobson, in [Jac95]. F. Kelly [Kel85] presents an early effort on the analysis of the system. The books by

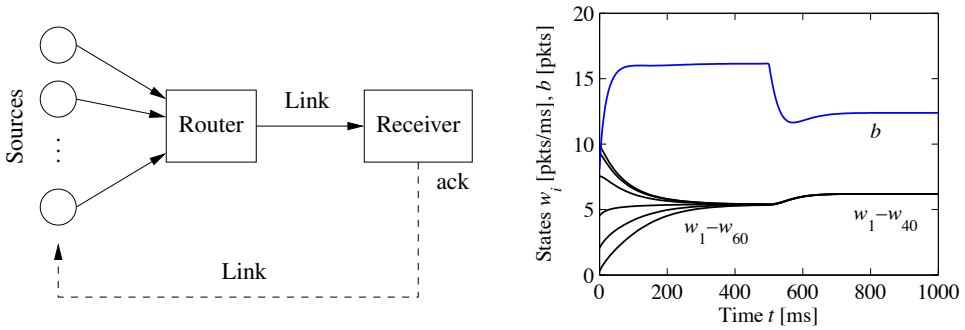


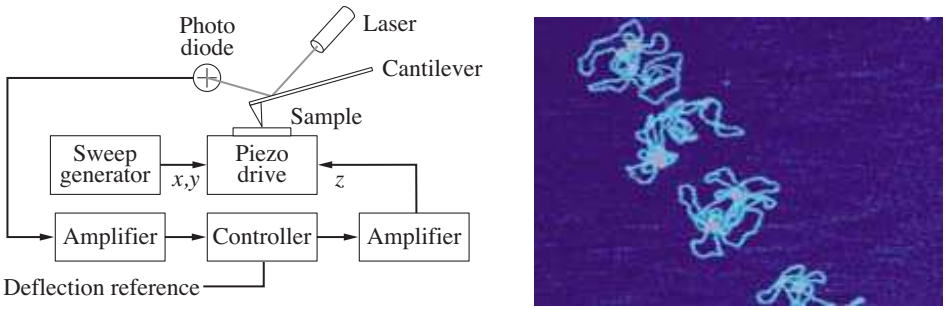
Figure 4.13: Internet congestion control for N identical sources across a single link. As shown on the left, multiple sources attempt to communicate through a router across a single link. An "ack" packet sent by the receiver acknowledges that the message was received; otherwise the message packet is resent and the sending rate is slowed down at the source. The simulation on the right is for 60 sources starting random rates, with 20 sources dropping out at $t = 500$ ms. The buffer size is shown at the top, and the individual source rates for 6 of the sources are shown at the bottom.

Hellerstein et al. [HDPT04] and Janert [Jan14] give many examples of the use of feedback in computer systems.

4.5 Atomic Force Microscopy

The 1986 Nobel Prize in Physics was shared by Gerd Binnig and Heinrich Rohrer for their design of the *scanning tunneling microscope*. The idea of the instrument is to bring an atomically sharp tip so close to a conducting surface that tunneling occurs. An image is obtained by traversing the tip across the sample and measuring the tunneling current as a function of tip position. This invention has stimulated the development of a family of instruments that permit visualization of surface structure at the nanometer scale, including the *atomic force microscope* (AFM), where a sample is probed by a tip on a cantilever. An AFM can operate in two modes. In *tapping mode* the cantilever is vibrated, and the amplitude of vibration is controlled by feedback. In *contact mode* the cantilever is in contact with the sample, and its bending is controlled by feedback. In both cases control is actuated by a piezo element that controls the vertical position of the cantilever base (or the sample). The control system has a direct influence on picture quality and scanning rate.

A schematic picture of an atomic force microscope is shown in Figure 4.14a. A microcantilever with a tip having a radius of the order of 10 nm is placed close to the sample. The tip can be moved vertically and horizontally using a piezoelectric scanner. It is clamped to the sample surface by attractive van der Waals forces and repulsive Pauli forces. The cantilever tilt depends on the topography of the surface and the position of the cantilever base, which is controlled by the piezo element.



(a) Schematic diagram (b) AFM image of DNA

Figure 4.14: Atomic force microscope. (a) A schematic diagram of an atomic force microscope, consisting of a piezo drive that scans the sample under the AFM tip. A laser reflects off of the cantilever and is used to measure the detection of the tip through a feedback controller. (b) An AFM image of strands of DNA. (Image courtesy Veeco Instruments.)

The tilt is measured by sensing the deflection of the laser beam using a photodiode. The signal from the photodiode is amplified and sent to a controller that drives the amplifier for the vertical position of the cantilever. By controlling the piezo element so that the deflection of the cantilever is constant, the signal driving the vertical deflection of the piezo element is a measure of the atomic forces between the cantilever tip and the atoms of the sample. An image of the surface is obtained by scanning the cantilever along the sample. The resolution makes it possible to see the structure of the sample on the atomic scale, as illustrated in Figure 4.14b, which shows an AFM image of DNA.

The horizontal motion of an AFM is typically modeled as a spring–mass system with low damping. The vertical motion is more complicated. To model the system, we start with the block diagram shown in Figure 4.15. Signals that are easily accessible are the input voltage u to the power amplifier that drives the piezo element, the voltage v applied to the piezo element and the output voltage y of the signal amplifier for the photodiode. The controller is a PI controller implemented by a computer, which is connected to the system by analog-to-digital (A/D) and digital-to-analog (D/A) converters. The deflection of the cantilever φ is also shown in the figure. The desired reference value for the deflection is an input to the computer.

There are several different configurations that have different dynamics. Here we will discuss a high-performance system from [SÅD+07] where the cantilever base is positioned vertically using a piezo stack. We begin the modeling with a simple experiment on the system. Figure 4.16a shows a step response of a scanner from the input voltage u to the power amplifier to the output voltage y of the signal amplifier for the photodiode. This experiment captures the dynamics of the chain of blocks from u to y in the block diagram in Figure 4.15. Figure 4.16a shows that the system responds quickly but that there is a poorly damped oscillatory mode with a period of about 35 μs. A primary task of the modeling is to understand the origin of the oscillatory behavior. To do so we will explore the system in more

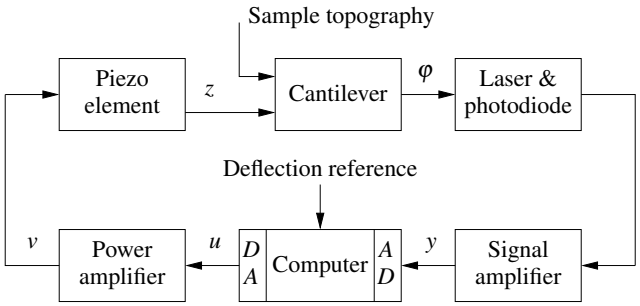


Figure 4.15: Block diagram of the system for vertical positioning of the cantilever for an atomic force microscope in contact mode. The control system attempts to keep the cantilever deflection equal to its reference value. Cantilever deflection is measured, amplified and converted to a digital signal, then compared with its reference value. A correcting signal is generated by the computer, converted to analog form, amplified and sent to the piezo element.

detail.

The natural frequency of the clamped cantilever is typically several hundred kilohertz, which is much higher than the observed oscillation of about 30 kHz. As a first approximation we will model it as a static system. Since the deflections are small, we can assume that the bending φ of the cantilever is proportional to the difference in height between the cantilever tip at the probe and the piezo scanner. A more accurate model can be obtained by modeling the cantilever as a spring–mass system of the type discussed in Chapter 3.

Figure 4.16a also shows that the response of the power amplifier is fast. The photodiode and the signal amplifier also have fast responses and can thus be modeled as static systems. The remaining block is a piezo system with suspension. A schematic mechanical representation of the vertical motion of the scanner is shown in Figure 4.16b. We will model the system as two masses separated by an ideal piezo element. The mass m_1 is half of the piezo system, and the mass m_2 is the other half of the piezo system plus the mass of the support.

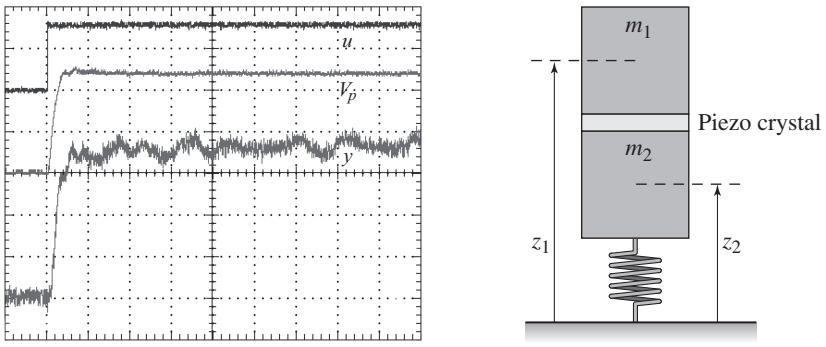
A simple model is obtained by assuming that the piezo crystal generates a force F between the masses and that there is a damping c in the spring. Let the positions of the center of the masses be z_1 and z_2. A momentum balance gives the following model for the system:

$$m_1 \frac{d^2 z_1}{dt^2} = F, \qquad m_2 \frac{d^2 z_2}{dt^2} = -c_2 \frac{dz_2}{dt} - k_2 z_2 - F.$$

Let the elongation of the piezo element $l = z_1 - z_2$ be the control variable and the height z_1 of the cantilever base be the output. Eliminating the variable F in the equations above and substituting $z_1 - l$ for z_2 gives the model

$$(m_1 + m_2)\frac{d^2 z_1}{dt^2} + c_2 \frac{dz_1}{dt} + k_2 z_1 = m_2 \frac{d^2 l}{dt^2} + c_2 \frac{dl}{dt} + k_2 l. \tag{4.23}$$

Summarizing, we find that a simple model of the system is obtained by mod-



(a) Step response (b) Mechanical model

Figure 4.16: Modeling of an atomic force microscope. (a) A measured step response. The top curve shows the voltage u applied to the drive amplifier (50 mV/div), the middle curve is the output V_p of the power amplifier (500 mV/div) and the bottom curve is the output y of the signal amplifier (500 mV/div). The time scale is 25 μs/div. Data have been supplied by Georg Schitter. (b) A simple mechanical model for the vertical positioner and the piezo crystal.

eling the piezo by (4.23) and all the other blocks by static models. Introducing the linear equations $l = k_3 u$ and $y = k_4 z_1$, we now have a complete model relating the output y to the control signal u. A more accurate model can be obtained by introducing the dynamics of the cantilever and the power amplifier. As in the previous examples, the concept of the uncertainty lemon in Figure 3.16b provides a framework for describing the uncertainty: the model will be accurate up to the frequencies of the fastest modeled modes and over a range of motion in which linearized stiffness models can be used.

The experimental results in Figure 4.16a can be explained qualitatively as follows. When a voltage is applied to the piezo, it expands by l_0, the mass m_1 moves up and the mass m_2 moves down instantaneously. The system settles after a poorly damped oscillation.

It is highly desirable to design a control system for the vertical motion so that it responds quickly with little oscillation. The instrument designer has several choices: to accept the oscillation and have a slow response time, to design a control system that can damp the oscillations or to redesign the mechanics to give resonances of higher frequency. The last two alternatives give a faster response and faster imaging.

Since the dynamic behavior of the system changes with the properties of the sample, it is necessary to tune the feedback loop. In simple systems this is currently done manually by adjusting parameters of a PI controller. There are interesting possibilities for making AFM systems easier to use by introducing automatic tuning and adaptation.

The book by Sarid [Sar91] gives a broad coverage of atomic force microscopes. The interaction of atoms close to surfaces is fundamental to solid state physics, see Kittel [Kit95]. The model discussed in this section is based on Schitter [Sch01].

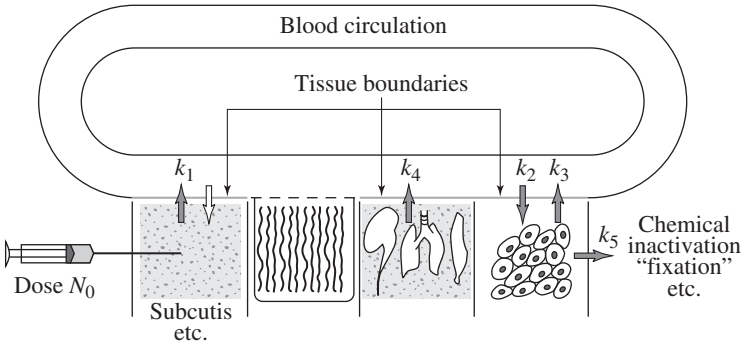


Figure 4.17: Abstraction used to compartmentalize the body for the purpose of describing drug distribution (based on Teorell [Teo37]). The body is abstracted by a number of compartments with perfect mixing, and the complex transport processes are approximated by assuming that the flow is proportional to the concentration differences in the compartments. The constants k_i parameterize the rates of flow between different compartments.

4.6 Drug Administration

The phrase "Take two pills three times a day" is a recommendation with which we are all familiar. Behind this recommendation is a solution of an open loop control problem. The key issue is to make sure that the concentration of a medicine in a part of the body is sufficiently high to be effective but not so high that it will cause undesirable side effects. The control action is quantized, *take two pills*, and sampled, *every 8 hours*. The prescriptions are based on simple models captured in empirical tables, and the dose is based on the age and weight of the patient.

Drug administration is a control problem. To solve it we must understand how a drug spreads in the body after it is administered. This topic, called *pharmacokinetics*, is now a discipline of its own, and the models used are called *compartment models*. They go back to the 1920s when Widmark modeled the propagation of alcohol in the body [WT24]. Compartment models are now important for the screening of all drugs used by humans. The schematic diagram in Figure 4.17 illustrates the idea of a compartment model. The body is viewed as a number of compartments like blood plasma, kidney, liver and tissues that are separated by membranes. It is assumed that there is perfect mixing so that the drug concentration is constant in each compartment. The complex transport processes are approximated by assuming that the flow rates between the compartments are proportional to the concentration differences in the compartments.

To describe the effect of a drug it is necessary to know both its concentration and how it influences the body. The relation between concentration c and its effect e is typically nonlinear. A simple model is

$$e = \frac{c}{c_0 + c} e_{\max}. \tag{4.24}$$

The effect is linear for low concentrations, and it saturates at high concentrations. The relation can also be dynamic, and it is then called *pharmacodynamics*.

Compartment Models

The simplest dynamic model for drug administration is obtained by assuming that the drug is evenly distributed in a single compartment after it has been administered and that the drug is removed at a rate proportional to the concentration. The compartments behave like stirred tanks with perfect mixing. Let c be the concentration, V the volume and q the outflow rate. Converting the description of the system into differential equations gives the model

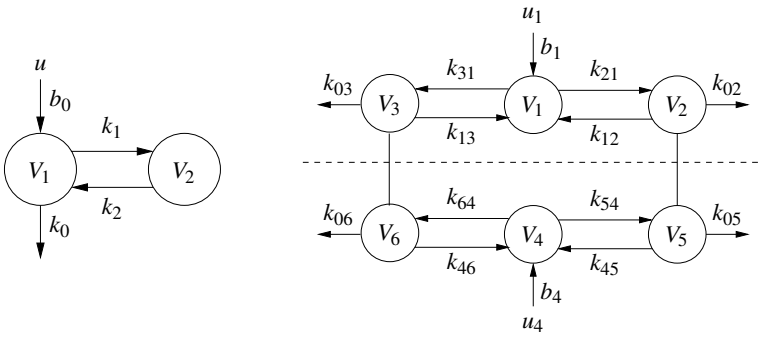
$$V\frac{dc}{dt} = -qc, \quad c \geq 0. \tag{4.25}$$

This equation has the solution $c(t) = c_0 e^{-qt/V} = c_0 e^{-kt}$, which shows that the concentration decays exponentially with the time constant $T = V/q$ after an injection. The input is introduced implicitly as an initial condition in the model (4.25). More generally, the way the input enters the model depends on how the drug is administered. For example, the input can be represented as a mass flow into the compartment where the drug is injected. A pill that is dissolved can also be interpreted as an input in terms of a mass flow rate.

The model (4.25) is called a a *one-compartment model* or a *single-pool model*. The parameter q/V is called the elimination rate constant. This simple model is often used to model the concentration in the blood plasma. By measuring the concentration at a few times, the initial concentration can be obtained by extrapolation. If the total amount of injected substance is known, the volume V can then be determined as $V = m/c_0$; this volume is called the *apparent volume of distribution*. This volume is larger than the real volume if the concentration in the plasma is lower than in other parts of the body. The model (4.25) is very simple, and there are large individual variations in the parameters. The parameters V and q are often normalized by dividing by the weight of the person. Typical parameters for aspirin are $V = 0.2$ L/kg and $q = 0.01$ (L/h)/kg. These numbers can be compared with a blood volume of 0.07 L/kg, a plasma volume of 0.05 L/kg, an intracellular fluid volume of 0.4 L/kg and an outflow of 0.0015 L/min/kg.

The simple one-compartment model captures the gross behavior of drug distribution, but it is based on many simplifications. Improved models can be obtained by considering the body as composed of several compartments. Examples of such systems are shown in Figure 4.18, where the compartments are represented as circles and the flows by arrows.

Modeling will be illustrated using the two-compartment model in Figure 4.18a. We assume that there is perfect mixing in each compartment and that the transport between the compartments is driven by concentration differences. We further assume that a drug with concentration c_0 is injected in compartment 1 at a volume flow rate of u and that the concentration in compartment 2 is the output. Let c_1 and c_2 be the concentrations of the drug in the compartments, and let V_1 and V_2 be the



(a) Two compartment model (b) Thyroid hormone model

Figure 4.18: Schematic diagrams of compartment models. (a) A simple two-compartment model. Each compartment is labeled by its volume, and arrows indicate the flow of chemical into, out of and between compartments. (b) A system with six compartments used to study the metabolism of thyroid hormone [God83]. The notation k_{ij} denotes the transport from compartment j to compartment i.

volumes of the compartments. The mass balances for the compartments are

$$\begin{aligned} V_1\frac{dc_1}{dt} &= q(c_2-c_1)-q_0c_1+c_0u, \qquad c_1\geq 0,\\ V_2\frac{dc_2}{dt} &= q(c_1-c_2), \qquad c_2\geq 0,\\ y &= c_2, \end{aligned} \tag{4.26}$$

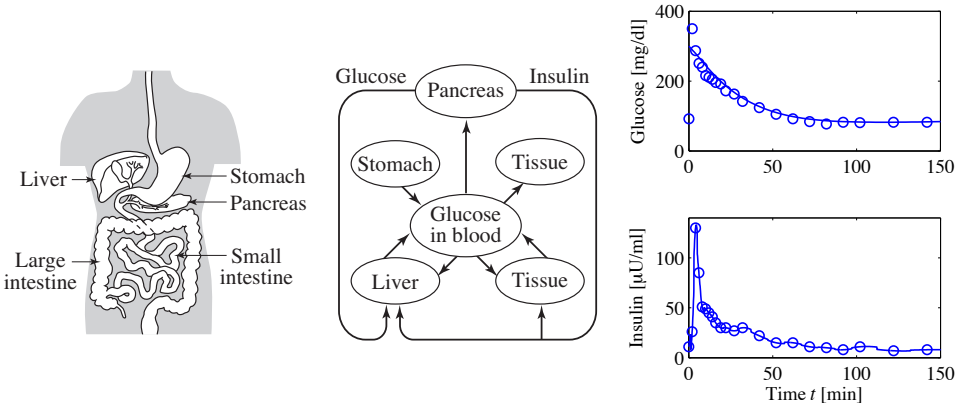
where q represents flow rate between the compartments and q_0 represents the flow rate out of compartment 1 that is not going to compartment 2. Introducing the variables $k_0 = q_0/V_1$, $k_1 = q/V_1$, $k_2 = q/V_2$ and $b_0 = c_0/V_1$ and using matrix notation, the model can be written as

$$\frac{dc}{dt} = \begin{pmatrix} -k_0-k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix} c + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 0 & 1 \end{pmatrix} c. \tag{4.27}$$

Comparing this model with its graphical representation in Figure 4.18a, we find that the mathematical representation (4.27) can be written by inspection.

It should also be emphasized that simple compartment models such as the one in equation (4.27) have a limited range of validity. Low-frequency limits exist because the human body changes with time, and since the compartment model uses average concentrations, they will not accurately represent rapid changes. There are also nonlinear effects that influence transportation between the compartments.

Compartment models are widely used in medicine, engineering and environmental science. An interesting property of these systems is that variables like concentration and mass are always positive. An essential difficulty in compartment modeling is deciding how to divide a complex system into compartments. Compartment models can also be nonlinear, as illustrated in the next section.



(a) Relevant body organs (b) Schematic diagram (c) Intravenous injection

Figure 4.19: Insulin–glucose dynamics. (a) Sketch of body parts involved in the control of glucose. (b) Schematic diagram of the system. (c) Responses of insulin and glucose when glucose in injected intravenously. From [PB86].

Insulin–glucose Dynamics

It is essential that the blood glucose concentration in the body is kept within a narrow range (0.7–1.1 g/L). Glucose concentration is influenced by many factors like food intake, digestion and exercise. A schematic picture of the relevant parts of the body is shown in Figures 4.19a and b.

There is a sophisticated mechanism that regulates glucose concentration. Glucose concentration is maintained by the pancreas, which secretes the hormones insulin and glucagon. Glucagon is released into the bloodstream when the glucose level is low. It acts on cells in the liver that release glucose. Insulin is secreted when the glucose level is high, and the glucose level is lowered by causing the liver and other cells to take up more glucose. In diseases like juvenile diabetes the pancreas is unable to produce insulin and the patient must inject insulin into the body to maintain a proper glucose level.

The mechanisms that regulate glucose and insulin are complicated; dynamics with time scales that range from seconds to hours have been observed. Models of different complexity have been developed. The models are typically tested with data from experiments where glucose is injected intravenously and insulin and glucose concentrations are measured at regular time intervals.

A relatively simple model called the *minimal model* was developed by Bergman and coworkers [Ber89]. This models uses two compartments, one representing the concentration of glucose in the bloodstream and the other representing the concentration of insulin in the interstitial fluid. Insulin in the bloodstream is considered an input. The reaction of glucose to insulin can be modeled by the equations

$$\frac{dx_1}{dt} = -(p_1 + x_2)x_1 + p_1 g_e, \qquad \frac{dx_2}{dt} = -p_2 x_2 + p_3(u - i_e), \tag{4.28}$$

where g_e and i_e represent the equilibrium values of glucose and insulin, x_1 is the concentration of glucose and x_2 is proportional to the concentration of interstitial insulin. Notice the presence of the term x_2x_1 in the first equation. Also notice that the model does not capture the complete feedback loop because it does not describe how the pancreas reacts to the glucose. Figure 4.19c shows a fit of the model to a test on a normal person where glucose was injected intravenously at time $t = 0$. The glucose concentration rises rapidly, and the pancreas responds with a rapid spikelike injection of insulin. The glucose and insulin levels then gradually approach the equilibrium values.

Models of the type in equation (4.28) and more complicated models having many compartments have been developed and fitted to experimental data. A difficulty in modeling is that there are significant variations in model parameters over time and for different patients. For example, the parameter p_1 in equation (4.28) has been reported to vary with an order of magnitude for healthy individuals. The models have been used for diagnosis and to develop schemes for the treatment of persons with diseases. Attempts to develop a fully automatic artificial pancreas have been hampered by the lack of reliable sensors.

The papers by Widmark and Tandberg [WT24] and Teorell [Teo37] are classics in pharmacokinetics, which is now an established discipline with many textbooks [Dos68, Jac72, GP82]. Because of its medical importance, pharmacokinetics is now an essential component of drug development. The book by Riggs [Rig63] is a good source for the modeling of physiological systems, and a more mathematical treatment is given in [KS01]. Compartment models are discussed in [God83]. The problem of determining rate coefficients from experimental data is discussed in [BÅ70] and [God83]. There are many publications on the insulin–glucose model. The minimal model is discussed in [CT84, Ber89] and more recent references are [MLK06, FCF+06].

4.7 Population Dynamics

Population growth is a complex dynamic process that involves the interaction of one or more species with their environment and the larger ecosystem. The dynamics of population groups are interesting and important in many different areas of social and environmental policy. There are examples where new species have been introduced into new habitats, sometimes with disastrous results. There have also been attempts to control population growth both through incentives and through legislation. In this section we describe some of the models that can be used to understand how populations evolve with time and as a function of their environments.

Logistic Growth Model

Let x be the population of a species at time t. A simple model is to assume that the birth rates and mortality rates are proportional to the total population. This gives

the linear model

$$\frac{dx}{dt} = bx - dx = (b-d)x = rx, \qquad x \geq 0, \tag{4.29}$$

where birth rate b and mortality rate d are parameters. The model gives an exponential increase if $b > d$ or an exponential decrease if $b < d$. A more realistic model is to assume that the birth rate decreases when the population is large. The following modification of the model (4.29) has this property:

$$\frac{dx}{dt} = rx(1 - \frac{x}{k}), \qquad x \geq 0, \tag{4.30}$$

where k is the *carrying capacity* of the environment. The model (4.30) is called the *logistic growth model*.

Predator–Prey Models

A more sophisticated model of population dynamics includes the effects of competing populations, where one species may feed on another. This situation, referred to as the *predator–prey problem*, was introduced in Example 3.3, where we developed a discrete-time model that captured some of the features of historical records of lynx and hare populations.

In this section, we replace the difference equation model used there with a more sophisticated differential equation model. Let $H(t)$ represent the number of hares (prey) and let $L(t)$ represent the number of lynxes (predator). The dynamics of the system are modeled as

$$\begin{aligned} \frac{dH}{dt} &= rH\left(1 - \frac{H}{k}\right) - \frac{aHL}{c+H}, & H \geq 0, \\ \frac{dL}{dt} &= b\frac{aHL}{c+H} - dL, & L \geq 0. \end{aligned} \tag{4.31}$$

In the first equation, r represents the growth rate of the hares, k represents the maximum population of the hares (in the absence of lynxes), a represents the interaction term that describes how the hares are diminished as a function of the lynx population and c controls the prey consumption rate for low hare population. In the second equation, b represents the growth coefficient of the lynxes and d represents the mortality rate of the lynxes. Note that the hare dynamics include a term that resembles the logistic growth model (4.30).

Of particular interest are the values at which the population values remain constant, called *equilibrium points*. The equilibrium points for this system can be determined by setting the right-hand side of the above equations to zero. Letting H_e and L_e represent the equilibrium state, from the second equation we have

$$L_e = 0 \quad \text{or} \quad H_e^* = \frac{cd}{ab-d}. \tag{4.32}$$

Substituting this into the first equation, we have that for $L_e = 0$ either $H_e = 0$ or

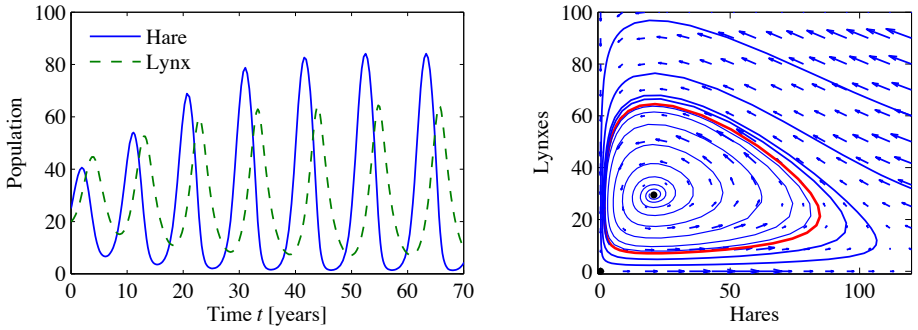


Figure 4.20: Simulation of the predator–prey system. The figure on the left shows a simulation of the two populations as a function of time. The figure on the right shows the populations plotted against each other, starting from different values of the population. The oscillation seen in both figures is an example of a *limit cycle*. The parameter values used for the simulations are $a = 3.2$, $b = 0.6$, $c = 50$, $d = 0.56$, $k = 125$ and $r = 1.6$.

$H_e = k$. For $L_e \neq 0$, we obtain

$$L_e^* = \frac{rH_e(c+H_e)}{aH_e}\left(1 - \frac{H_e}{k}\right) = \frac{bcr(abk - cd - dk)}{(ab-d)^2 k}. \tag{4.33}$$

Thus, we have three possible equilibrium points $x_e = (L_e, H_e)$:

$$x_e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad x_e = \begin{pmatrix} k \\ 0 \end{pmatrix}, \qquad x_e = \begin{pmatrix} H_e^* \\ L_e^* \end{pmatrix},$$

where H_e^* and L_e^* are given in equations (4.32) and (4.33). Note that the equilibrium populations may be negative for some parameter values, corresponding to a nonachievable equilibrium point.

Figure 4.20 shows a simulation of the dynamics starting from a set of population values near the nonzero equilibrium values. We see that for this choice of parameters, the simulation predicts an oscillatory population count for each species, reminiscent of the data shown in Figure 3.6.

Volume I of the two-volume set by J. D. Murray [Mur04] give a broad coverage of population dynamics.

Exercises

4.1 (Cruise control) Consider the cruise control example described in Section 4.1. Build a simulation that re-creates the response to a hill shown in Figure 4.3b and show the effects of increasing and decreasing the mass of the car by 25%. Redesign the controller (using trial and error is fine) so that it returns to within 1% of the desired speed within 3 s of encountering the beginning of the hill.

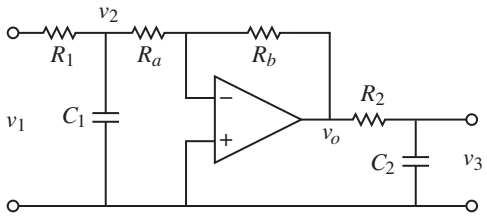
4.2 (Bicycle dynamics) Show that the dynamics of a bicycle frame given by equation (4.5) can be approximated in state space form as

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ mgh/J & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} Dv_0/(bJ) \\ mv_0^2 h/(bJ) \end{pmatrix} u,$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} x,$$

where the input u is the steering angle δ and the output y is the tilt angle φ. What do the states x_1 and x_2 represent?

4.3 (Bicycle steering) Combine the bicycle model given by equation (4.5) and the model for steering kinematics in Example 3.10 to obtain a model that describes the path of the center of mass of the bicycle.

4.4 (Operational amplifier circuit) Consider the op amp circuit shown below.

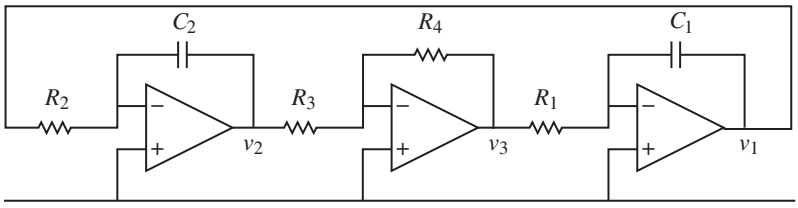


Show that the dynamics can be written in state space form as

$$\frac{dx}{dt} = \begin{pmatrix} -\dfrac{1}{R_1C_1} - \dfrac{1}{R_aC_1} & 0 \\ \dfrac{R_b}{R_a}\dfrac{1}{R_2C_2} & -\dfrac{1}{R_2C_2} \end{pmatrix} x + \begin{pmatrix} \dfrac{1}{R_1C_1} \\ 0 \end{pmatrix} u, \quad y = \begin{pmatrix} 0 & 1 \end{pmatrix} x,$$

where $u = v_1$ and $y = v_3$. (Hint: Use v_2 and v_3 as your state variables.)

4.5 (Operational amplifier oscillator) The op amp circuit shown below is an implementation of an oscillator.



Show that the dynamics can be written in state space form as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & \dfrac{R_4}{R_1R_3C_1} \\ -\dfrac{1}{R_2C_2} & 0 \end{pmatrix} x,$$

where the state variables represent the voltages across the capacitors $x_1 = v_1$ and $x_2 = v_2$.

4.6 (Congestion control using RED [LPW+02]) A number of improvements can be made to the model for Internet congestion control presented in Section 4.4. To ensure that the router's buffer size remains positive, we can modify the buffer dynamics to satisfy

$$\frac{db_l}{dt} = \begin{cases} s_l - c_l & b_l > 0 \\ \text{sat}_{(0,\infty)}(s_l - c_l) & b_l = 0. \end{cases}$$

In addition, we can model the drop probability of a packet based on how close we are to the buffer limits, a mechanism known as random early detection (RED):

$$p_l = m_l(a_l) = \begin{cases} 0 & a_l(t) \le b_l^{\text{lower}} \\ \rho_l r_i(t) - \rho_l b_l^{\text{lower}} & b_l^{\text{lower}} < a_l(t) < b_l^{\text{upper}} \\ \eta_l r_i(t) - (1 - 2b_l^{\text{upper}}) & b_l^{\text{upper}} \le a_l(t) < 2b_l^{\text{upper}} \\ 1 & a_l(t) \ge 2b_l^{\text{upper}}, \end{cases}$$

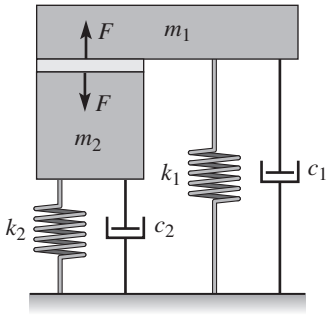
$$\frac{da_l}{dt} = -\alpha_l c_l (a_l - b_l),$$

where α_l, b_l^{upper}, b_l^{lower} and p_l^{upper} are parameters for the RED protocol.

Using the model above, write a simulation for the system and find a set of parameter values for which there is a stable equilibrium point and a set for which the system exhibits oscillatory solutions. The following sets of parameters should be explored:

$$\begin{aligned} &N = 20, 30, \ldots, 60, && b_l^{\text{lower}} = 40 \text{ pkts}, && \rho_l = 0.1, \\ &c = 8, 9, \ldots, 15 \text{ pkts/ms}, && b_l^{\text{upper}} = 540 \text{ pkts}, && \alpha_l = 10^{-4}, \\ &\tau = 55, 60, \ldots, 100 \text{ ms}. \end{aligned}$$

4.7 (Atomic force microscope with piezo tube) A schematic diagram of an AFM where the vertical scanner is a piezo tube with preloading is shown below.



Show that the dynamics can be written as

$$(m_1 + m_2)\frac{d^2 z_1}{dt^2} + (c_1 + c_2)\frac{dz_1}{dt} + (k_1 + k_2) z_1 = m_2 \frac{d^2 l}{dt^2} + c_2 \frac{dl}{dt} + k_2 l.$$

Are there parameter values that make the dynamics particularly simple?

4.8 (Drug administration) The metabolism of alcohol in the body can be modeled by the nonlinear compartment model

$$V_b \frac{dc_b}{dt} = q(c_l - c_b) + q_{iv}, \qquad V_l \frac{dc_l}{dt} = q(c_b - c_l) - q_{\max} \frac{c_l}{c_0 + c_l} + q_{gi},$$

where $V_b = 48$ L and $V_l = 0.6$ L are the apparent volumes of distribution of body water and liver water, c_b and c_l are the concentrations of alcohol in the compartments, q_{iv} and q_{gi} are the injection rates for intravenous and gastrointestinal intake, $q = 1.5$ L/min is the total hepatic blood flow, $q_{\max} = 2.75$ mmol/min and $c_0 = 0.1$ mmol/L. Simulate the system and compute the concentration in the blood for oral and intravenous doses of 12 g and 40 g of alcohol.

4.9 (Population dynamics) Consider the model for logistic growth given by equation (4.30). Show that the maximum growth rate occurs when the size of the population is half of the steady-state value.

4.10 (Fisheries management) The dynamics of a commercial fishery can be described by the following simple model:

$$\frac{dx}{dt} = f(x) - h(x,u), \qquad y = bh(x,u) - cu,$$

where x is the total biomass, $f(x) = rx(1 - x/k)$ is the growth rate and $h(x,u) = axu$ is the harvesting rate. The output y is the rate of revenue, and the parameters a, b and c are constants representing the price of fish and the cost of fishing. Show that there is an equilibrium where the steady-state biomass is $x_e = c/(ab)$. Compare with the situation when the biomass is regulated to a constant value and find the maximum sustainable return in that case.