Chapter 3

Dynamic Behavior

In this chapter we give a broad discussion of the behavior of dynamical systems. We focus on systems modeled by differential equations, but consider the general nonlinear case. This allows us to discuss equilibrium points, stability, limit cycles and other key concepts of dynamical systems. We also introduce some measures of performance for input/output systems that provide additional tools for characterizing dynamic behavior.

3.1 Solving Differential Equations

In the last chapter, we saw that one of the methods of modeling dynamical systems is through the use of ordinary differential equations (ODEs). A state space, input/output system has the form

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= g(x), \end{aligned} \tag{3.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the input, and $y \in \mathbb{R}^q$ is the output. The smooth maps $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^q$ represent the dynamics and measurements for the system. We will focus in this text on single input, single output (SISO) systems, for which $p = q = 1$.

We begin by investigating systems in which the input has been set to a function of the state, $u = \alpha(x)$. This is one of the simplest types of feedback, in which the system regulates its own behavior. The differential equations in this case become

$$\dot{x} = f(x, \alpha(x)) = F(x). \tag{3.2}$$

As before, we write $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ for the state vector.

In order to understand the dynamic behavior of this system, we need to analyze the features of the solutions of this equation. While in some simple situations we can write down the solutions in analytical form, more often we must rely on computational approaches. We begin by describing the class of solutions for this problem.

Initial Value Problems

We say that $x(t)$ is a *solution* of the differential equation (3.2) on the time interval $t_0 \in \mathbb{R}$ to $t_f \in \mathbb{R}$ if

$$\dot{x}(t) = F(x(t)) \quad \text{for all } t_0 \leq t < t_f.$$

A given differential equation may have many solutions. We will most often be interested in the *initial value problem*, where $x(t)$ is prescribed at a given time $t_0 \in \mathbb{R}$ and we wish to find a solution valid for all *future* time, $t > t_0$.

We say that $x(t)$ is a solution of the differential equation (3.2) with initial value $x_0 \in \mathbb{R}^n$ at $t_0 \in \mathbb{R}$ if

$$x(t_0) = x_0 \quad \text{and} \quad \dot{x}(t) = F(x(t)) \quad \text{for all } t_0 \leq t < t_f.$$

For most differential equations we will encounter, there is a *unique* solution that is defined for $t_0 \leq t \leq t_f$. The solution may defined for all time $t > t_0$, in which case we take $t_f = \infty$. Because we will primarily be interested in solutions of the initial value problem for ODEs, we will often refer to this simply as the solution of an ODE.

We will usually assume that t_0 is equal to 0. In the case when F is independent of time (as in equation (3.2)), we can do so without loss of generality by choosing a new independent (time) variable, $\tau = t - t_0$.

Example 3.1 *(Damped oscillator).* Consider a damped, linear oscillator, introduced in Example 2.3. The equations of motion for the system are

$$m\ddot{q} + b\dot{q} + kq = 0,$$

where q is the displacement of the oscillator from its rest position. We assume that $b^2 < 4km$, corresponding to a lightly damped system (the reason for this particular choice will become clear later). We can rewrite this in state space form by setting $x_1 = q$ and $x_2 = \dot{q}$, giving

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{b}{m}x_2.\end{aligned}$$

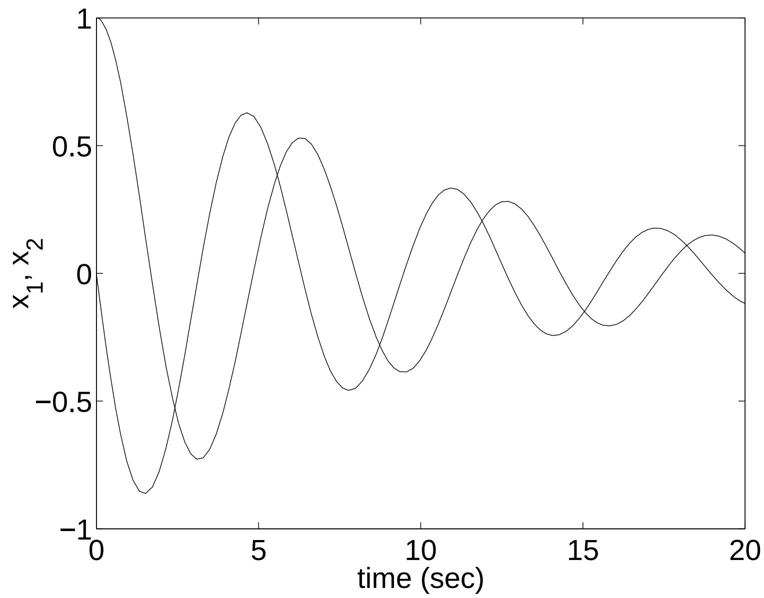


Figure 3.1: Response of the damped oscillator to the initial condition $x_0 = (1, 0)$.

In vector form, the right hand side can be written as

$$F(x) = \begin{bmatrix} x_2 \\ -\frac{k}{m}x_1 - \frac{b}{m}x_2 \end{bmatrix}.$$

The solution to the initial value problem can be written in a number of different ways and will be explored in more detail in Chapter 4. Here we simply assert that the solution can be written as

$$x_1(t) = e^{-\frac{bt}{2m}} \left(x_{10} \cos \omega t + (x_{10} + \frac{x_{20}}{\omega}) \sin \omega t \right)$$
$$x_2(t) = e^{-\frac{bt}{2m}} \left(\left(x_{20} + x_{10}(\omega - \frac{b}{2})\right) \cos \omega t + \left(x_{10}\omega + \frac{b}{2}(x_{10} + \frac{x_{20}}{\omega})\right) \sin \omega t \right)$$

where $x_0 = (x_{10}, x_{20})$ is the initial condition and $\omega = \sqrt{4km - b^2}/2m$. This solution can be verified by substituting it into the differential equation. We see that the solution is explicitly dependent on the initial condition and it can be shown that this solution is unique. A plot of the initial condition response is shown in Figure 3.1. We note that this form of the solution only holds for $b^2 - 4km < 0$, corresponding to an "underdamped" oscillator.

Numerical Solutions

One of the benefits of the computer revolution that we can benefit from is that it is very easy to obtain a numerical solution of the differential equation when the initial condition is given. A nice consequence of this is as soon as we have a model in the form of (3.2), it is straightforward to generate the behavior of x for different initial conditions, as we saw briefly in the previous chapter.

Modern computing environments allow simulation of differential equations as a basic operation. In particular, MATLAB provides several tools for representing, simulating, and analyzing ordinary differential equations of the form in equation (3.2). To define an ODE in MATLAB, we define a function representing the right hand side of equation (3.2):

```
function dxdt = sysname(t, x)
dxdt = [
  F1(x);
  F2(x);
  ...
  Fn(x);
];
```

Each function `Fi(x)` takes a (column) vector `x` and returns the ith element of the differential equation. The first argument to the function `sysname`, `t`, represents the current time and allows for the possibility of time-varying differential equations, in which the right hand side of the ODE in equation (3.2) depends explicitly on time.

ODEs define in this fashion can be simulated by using the MATLAB `ode45` command:

```
ode45('file', [0,T], [x10, x20, ..., xn0])
```

The first argument is the name of the file containing the ODE declaration, the second argument gives the time interval over which the simulation should be performed and the final argument gives the vector of initial conditions. The default action of the `ode45` command is to plot the time response of each of the states of the system.

Example 3.2 *(Balance system).* Consider the balance system given in Example 2.1 and reproduced Figure 3.2a. Suppose that a coworker has designed

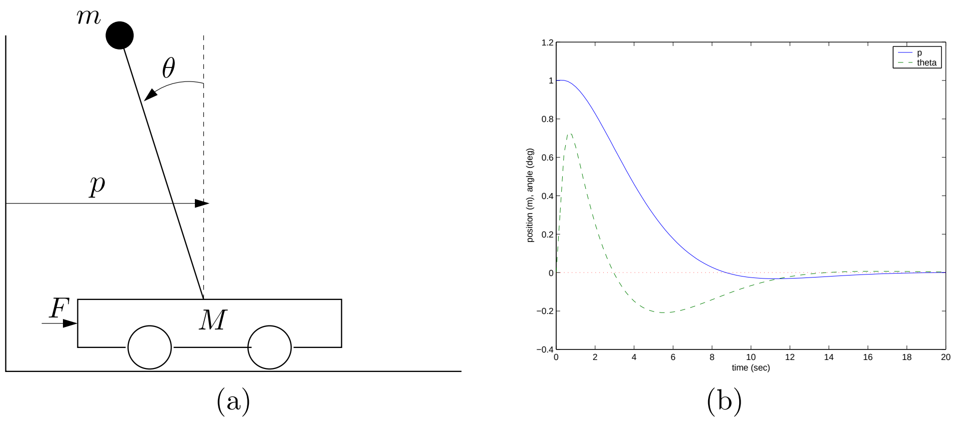


Figure 3.2: Balance system: (a) simplified diagram and (b) initial condition response.

a control law that will hold the position of the system steady in the upright position at $p = 0$. The form of the control law is

$$F = Kx,$$

where $x = (p, \theta, \dot{p}, \dot{\theta}) \in \mathbb{R}^4$ is the state of the system, F is the input, and $K = (k_1, k_2, k_3, k_4)$ is the vector of "gains" for the control law.

The equations of motion for the system, in state space form, are

$$\frac{d}{dt}\begin{bmatrix} p \\ \theta \\ \dot{p} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \dot{\theta} \\ \begin{bmatrix} M+m & ml\cos\theta \\ J+ml^2 & ml\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} -b\dot{x} + ml\sin\theta\,\dot{\theta}^2 + Kx \\ -mgl\sin\theta \end{bmatrix} \end{bmatrix}$$

$$y = \begin{bmatrix} p \\ \theta \end{bmatrix}.$$

We use the following parameters for the system (corresponding roughly to a human being balanced on a stabilizing cart):

$$\begin{aligned} M &= 10 \text{ kg} & m &= 80 \text{ kg} & b &= 0.1 \text{ Ns/m} \\ J &= 100 \text{ kg m}^2/\text{s}^2 & l &= 1 \text{ m} & g &= 9.8 \text{ m/s}^2 \end{aligned}$$

$$K = \begin{bmatrix} -1 & 120 & -4 & 20 \end{bmatrix}$$

This system can now be simulated using MATLAB or a similar numerical tool. The results are shown in Figure 3.2b, with initial condition $x_0 = (1, 0, 0, 0)$. We see from the plot that after an initial transient, the angle and position of the system return to zero (and remain there).

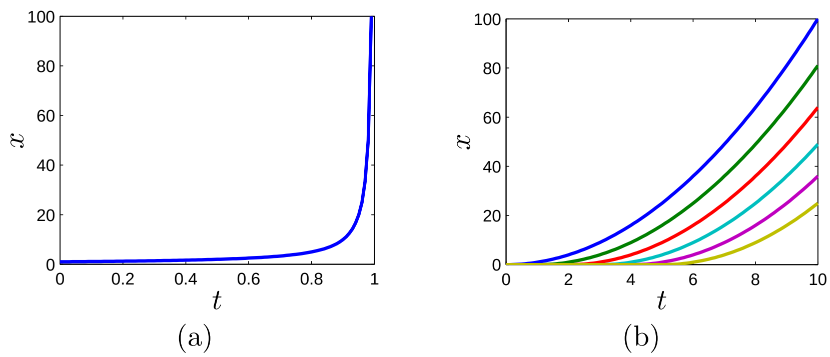


Figure 3.3: Solutions to the differential equation (3.3) (a) and (3.4) (b).

Existence and Uniqueness

Without imposing some conditions on the function F the differential equation (3.2) may not have a solution for all t, and there is no guarantee that the solution is unique. We illustrate this with two examples.

Example 3.3 *(Finite escape time).* Let $x \in \mathbb{R}$ and consider the differential equation

$$\frac{dx}{dt} = x^2 \tag{3.3}$$

with initial condition $x(0) = 1$. By differentiation we can verify that the function

$$x(t) = \frac{1}{1-t} \tag{3.4}$$

satisfies the differential equation and it also satisfies the initial condition. A graph of the solution is given in Figure 3.3a; notice that the solution goes to infinity as t goes to 1. Thus the solution only exists in the time interval $0 \le t < 1$.

Example 3.4 *(No unique solution).* Let $x \in \mathbb{R}$ and consider the differential equation

$$\frac{dx}{dt} = \sqrt{x}$$

with initial condition $x(0) = 0$. By differentiation we can verify that the function

$$x(t) = \begin{cases} 0 & \text{if } t \le a \\ \frac{1}{4}(t-a)^2 & \text{if } t > a \end{cases}$$

satisfies the differential equation for all values of the parameter $a \geq 0$. The function also satisfies the initial condition. A graph of some of the possible solutions is given in Figure 3.3b. Notice that in this case there are many solutions to the differential equations.

These simple examples show clearly that there may be difficulties even with seemingly simple differential equations. Existence and uniqueness can be guaranteed by requiring that the function F has the property that for some fixed $c \in \mathbb{R}$

$$||F(x) - F(y)|| < c||x - y|| \quad \text{for all } x, y,$$

which is called *Lipschitz continuity.* A sufficient condition for a function to be Lipschitz is that the Jacobian, $\partial f/\partial x$, is uniformly bounded for all x. The difficulty in Example 3.3 is that the derivative $\partial f/\partial x$ becomes large for large x and the difficulty in Example 3.4 is that the derivative F_x is infinite at the origin.

3.2 Qualitative Analysis

The behavior of nonlinear systems will now be discussed qualitatively. We will focus on an important class of systems, already described briefly in the last chapter, known as planar dynamical systems. These systems have state $x \in \mathbb{R}^2$, allowing their solutions to be plotted in the (x_1, x_2) plane. The basic concepts that we describe hold more generally and can be used to understand dynamical behavior in higher dimensions.

Phase Portraits

A convenient way to understand the behavior of dynamical systems with state $x \in \mathbb{R}^2$ is to plot the *phase portrait* of the system, briefly introduced in Chapter 2. We start by introducing the concept of a vector field. For a system of ordinary differential equations

$$\dot{x} = F(x),$$

the right hand side of the differential equation defines at every $x \in \mathbb{R}^n$ a velocity $F(x) \in \mathbb{R}^n$. This velocity tells us how x changes and can be represented as a vector $F(x) \in \mathbb{R}^n$. For planar dynamical systems, we can plot these vectors on a grid of points in the plane and obtain a visual image of the dynamics of the system, as shown in Figure 3.4a.

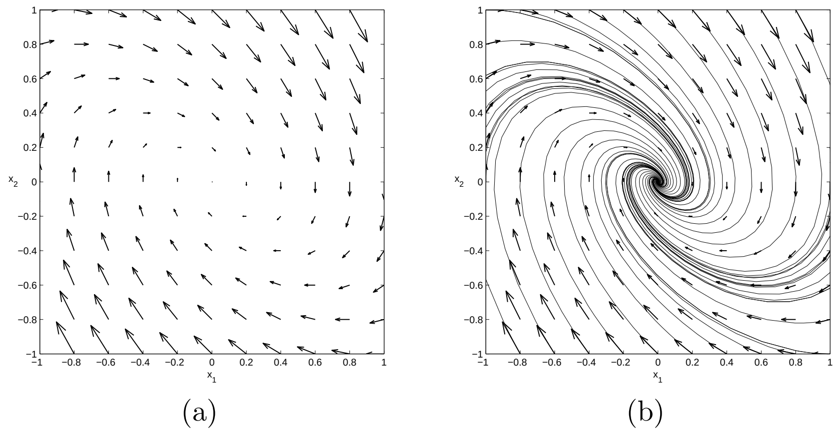


Figure 3.4: Vector field plot (a) and phase portrait (b) for a damped oscillator. This plots were produced using the `phaseplot` command in MATLAB.

A phase portrait is constructed by plotting the flow of the vector field corresponding to the planar dynamical system. That is, for a set of initial conditions $x_0 \in \mathbb{R}^2$, we plot the solution of the differential equation in the plane $\mathbb{R}^2$. This corresponds to following the arrows at each point in the phase plane and drawing the resulting trajectory. By plotting the resulting trajectories for several different initial conditions, we obtain a phase portrait, as show in Figure 3.4b.

Phase portraits give us insight into the dynamics of the system by showing us the trajectories plotted in the (two dimensional) state space of the system. For example, we can see whether all trajectories tend to a single point as time increases or whether there are more complicated behaviors as the system evolves. In the example in Figure 3.4, corresponding to a damped oscillator, we see that for all initial conditions the system approaches the origin. This is consistent with our simulation in Figure 3.1 (also for a damped oscillator), but it allows us to infer the behavior for all initial conditions rather than a single initial condition. However, the phase portrait does not readily tell us the rate of change of the states (although this can be inferred from the length of the arrows in the vector field plot).

Equilibrium Points

An *equilibrium point* of a dynamical system represents a stationary condition for the dynamics. We say that at state x_e is an equilibrium point for a

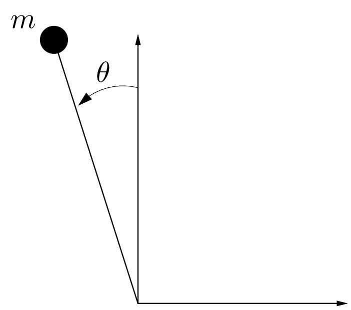


Figure 3.5: An inverted pendulum.

dynamical system

$$\dot{x} = F(x)$$

if $F(x_e) = 0$. If a dynamical system has an initial condition $x(0) = x_e$ then it will stay at the equilibrium point: $x(t) = x_e$ for all $t > 0$.[1]

Equilibrium points are one of the most important features of a dynamical system since they define the states corresponding to constant operating conditions. A dynamical system can have zero, one or more equilibrium points.

Example 3.5 *(Inverted Pendulum).* Consider the inverted pendulum in Figure 3.5. The state variables are the angle $\theta = x_1$ and the angular velocity $d\theta/dt = x_2$, the control variable is the acceleration u of the pivot, and the output is the angle θ.

Newton's law of conservation of angular momentum becomes

$$J\frac{d^2\theta}{dt^2} = mgl\sin\theta + mul\cos\theta$$

Introducing $x_1 = \theta$ and $x_2 = d\theta/dt$ the state equations become

$$\begin{aligned}\frac{dx}{dt} &= \begin{bmatrix} x_2 \\ \frac{mgl}{J}\sin x_1 + \frac{mlu}{J}\cos x_1 \end{bmatrix}\\ y &= x_1.\end{aligned}$$

For simplicity, we assume $mgl/J = ml/J = 1$, so that our equations become

$$\begin{aligned}\frac{dx}{dt} &= \begin{bmatrix} x_2 \\ \sin x_1 + u\cos x_1 \end{bmatrix}\\ y &= x_1.\end{aligned} \tag{3.5}$$

[1] We take $t_0 = 0$ from here on.

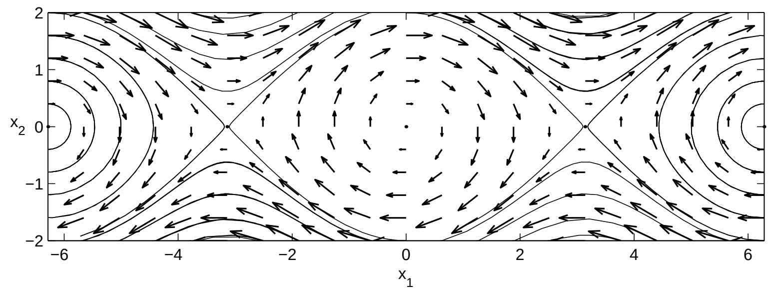


Figure 3.6: Phase portrait for a simple pendulum. The equilibrium points are marked by solid dots along the $x_2 = 0$ line.

This is a nonlinear time-invariant system of second order.

An alternative way to obtain these equations is to rescale time by choosing $t' = \sqrt{J/mgl}\,t$ and scale the input by choosing $u' = gu$. This results in an equation of the same as equation (3.5) and holds for any values of the parameters.

The equilibrium points for the system are given by

$$x_e = \begin{bmatrix} 0 \\ \pm n\pi \end{bmatrix}$$

where $n = 0, 1, 2, \dots$. The equilibrium points for n even correspond to the pendulum hanging down and those for n odd correspond to the pendulum pointing up. A phase portrait for this system is shown in Figure 3.6. The phase plane shown in the figure is $\mathbb{R} \times \mathbb{R}$, which results in our model having an infinite number of equilibrium, corresponding to 0, $\pm\pi$, $\pm 2\pi$, ...

Limit Cycles

Nonlinear systems can exhibit very rich behavior. Consider the differential equation

$$\begin{aligned} \frac{dx_1}{dt} &= -x_2 - x_1(1 - x_1^2 - x_2^2) \\ \frac{dx_2}{dt} &= x_1 - x_2(1 - x_1^2 - x_2^2). \end{aligned} \tag{3.6}$$

The phase portrait and time domain solutions are given in Figure 3.7. The figure shows that the solutions in the phase plane converge to an orbit which is a circle. In the time domain this corresponds to a oscillatory solution.

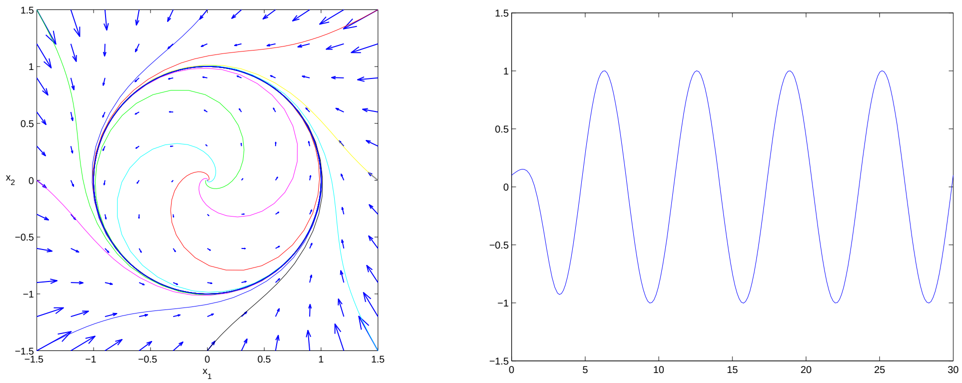


Figure 3.7: Phase portrait and time domain simulation for a system with a limit cycle.

Mathematically the circle is called a *limit cycle*. More formally, we call a solution $x(t)$ a limit cycle of period $T > 0$ if $x(t+T) = x(t)$ for all $t \in \mathbb{R}$.

Example 3.6 *(Predator Prey).* Consider the predator prey example introduced in Example 2.2. We replace the difference equation model used there with a more sophisticated differential equation model. Let $R(t)$ represent the number of rabbits (prey) and $F(t)$ represent the number of foxes (predator). The dynamics of the system are modeled as

$$\begin{aligned} \dot{R} &= r_R R\left(1 - \frac{R}{K}\right) - \frac{aRF}{1 + aRT_h} \qquad R \geq 0 \\ \dot{F} &= r_F F\left(1 - \frac{F}{kR}\right) \qquad\qquad\qquad F \geq 0. \end{aligned}$$

In the first equation, r_R represents the growth rate of the rabbits, K represents the maximum population of rabbits (in the absence of foxes), a represents the interaction term that describes how the rabbits are diminished as a function of the fox population, and T_h depends is a time constant for prey consumption. In the second equation, r_F represents the growth rate of the foxes and k represents the fraction of rabbits versus foxes at equilibrium.

The equilibrium points for this system can be determined by setting the right hand side of the above equations to zero. Letting R_e and F_e represent the equilibrium state, from the second equation we have

$$F_e = kR_e.$$

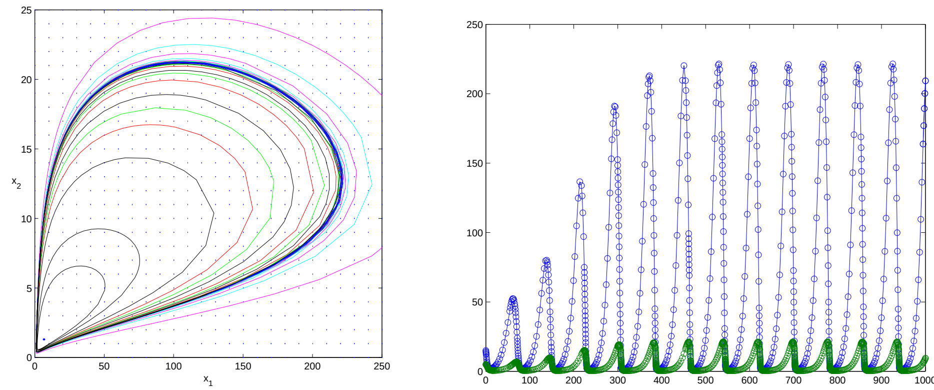


Figure 3.8: Phase portrait and time domain simulation for the predator prey system.

Substituting this into the first equation, we must solve

$$r_R R_e \left(1 - \frac{R_e}{K}\right) - \frac{akR_e^2}{1 + aR_eT_h} = 0.$$

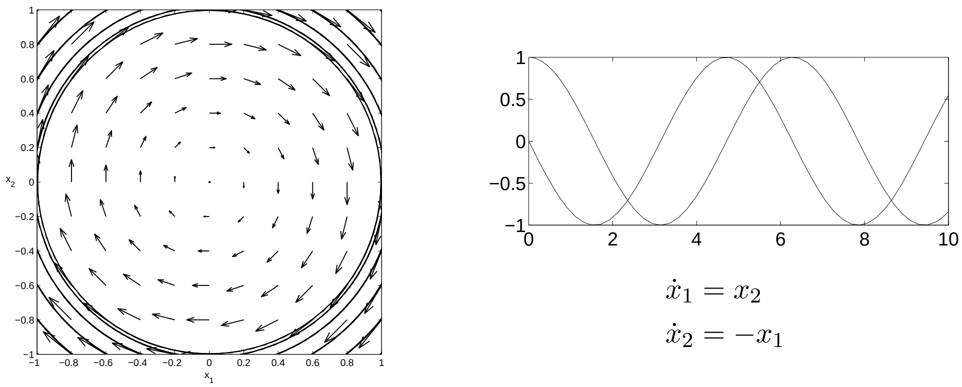
Multiplying through by the denominator, we get

$$\begin{aligned} 0 &= R_e \cdot \left(r_R \left(1 - \frac{R_e}{K}\right)(1 + aR_eT_h) - akR_e \right) \\ &= R_e \cdot \left(\frac{r_R a T_h}{K} R_e^2 + (ak + r_R/K - r_R a T_h) R_e - r_R \right). \end{aligned}$$

This gives one solution at $R_e = 0$ and a second that can be solved analytically or numerically.

The phase portrait for this system is shown in Figure 3.8. In addition to the two equilibrium points, we see a limit cycle in the diagram. This limit cycle is *attracting* or *stable* since initial conditions near the limit cycle approach it as time increases. It divides the phase space into two different regions: one inside the limit cycle in which the size of the population oscillations growth with time (until they rich the limit cycle) and one outside the limit cycle in which they decay.

For second order systems there are methods for determining limit cycles. For general higher order systems there are no general methods so we have to resort to computational analysis. Computer algorithms find limit cycles by searching for periodic trajectories in state space that satisfy the dynamics of the system. In many situations, stable limit cycles can be found by simulating the system with different initial conditions.



$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_1$$

Figure 3.9: Phase portrait and time domain simulation for a system with a single stable equilibrium point.

3.3 Stability

The stability of an equilibrium point determines whether or not solutions nearby the equilibrium point remain nearby, get closer, or get further away.

Definitions

An equilibrium point is *stable* if initial conditions that start near an equilibrium point stay near that equilibrium point. Formally, we say that an equilibrium point x_e is stable if for all $\epsilon > 0$, there exists an $\delta > 0$ such that

$$\|x(0) - x_e\| < \delta \quad \Longrightarrow \quad \|x(t) - x_e\| < \epsilon \quad \text{for all } t > 0.$$

Note that this definition does not imply that $x(t)$ gets closer to x_e as time increases, but just that it stays nearby. Furthermore, the value of δ may depend on ϵ, so that if we wish to stay very close to the equilibrium point, we may have to start very, very close ($\delta \ll \epsilon$). This type of stability is sometimes called stability "in the sense of Lyapunov" (isL for short).

An example of a stable equilibrium point is shown in Figure 3.9. From the phase portrait, we see that if we start near the equilibrium then we stay near the equilibrium. Indeed, for this example, given any ϵ that defines the range of possible initial conditions, we can simply choose $\delta = \epsilon$ to satisfy the definition of stability.

An equilibrium point x_e is (locally) *asymptotically stable* if it is stable in the sense of Lyapunov and also $x(t) \to x_e$ as $t \to \infty$ for $x(t)$ sufficiently close to x_e. This corresponds to the case where all nearby trajectories converge

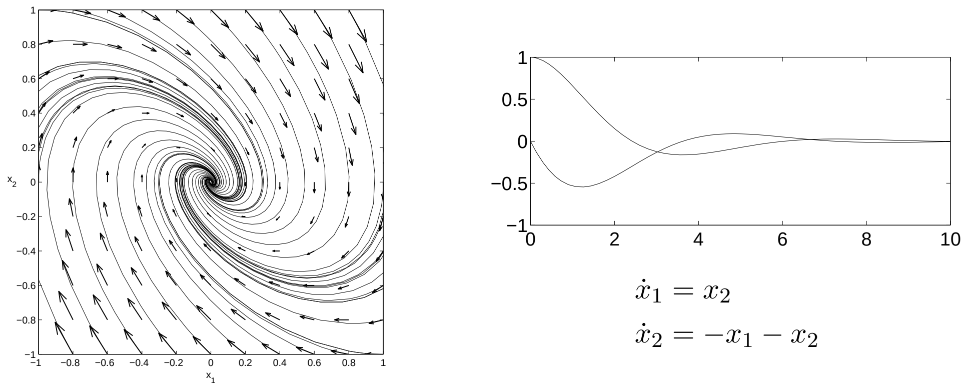


Figure 3.10: Phase portrait and time domain simulation for a system with a single asymptotically stable equilibrium point.

to the equilibrium point for large time. Figure 3.10 shows an example of an asymptotically stable equilibrium point. Note from the phase portraits that not only do all trajectories stay near the equilibrium point at the origin, but they all approach the origin as t gets large (the directions of the arrows on the phase plot show the direction in which the trajectories move).

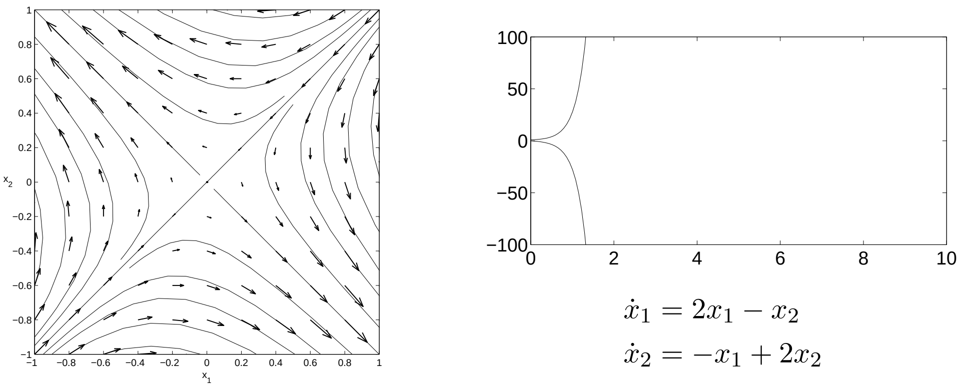
An equilibrium point is *unstable* if it is not stable. More specifically, we say that an equilibrium point is unstable if given some $\epsilon > 0$, there does not exist a $\delta > 0$ such that if $\|x(0) - x_e\| < \delta$ then $\|x(t) - x_e\| < \epsilon$ for all t. An example of an unstable equilibrium point is shown in Figure 3.11.

The definitions above are given without careful description of their domain of applicability. More formally, we define an equilibrium point to be *locally* stable (or asymptotically stable) if it is stable for all initial conditions $x \in B_r(x_e)$ where

$$B_r(x_e) = \{x : \|x - x_e\| < \delta\}$$

is a ball of radius r around x_e and $r > 0$. A system is globally stable if it stable for all $r > 0$. Systems whose equilibrium points are only locally stable can have interesting behavior away from equilibrium points, as we explore in the next section.

For planar dynamical systems, equilibrium points have been assigned names based on their stability type. An asymptotically stable equilibrium point is called a *sink* or sometimes an *attractor*. An unstable equilibrium point can either be a *source*, if all trajectories lead away from the equilibrium point, or a *saddle*, if some trajectories lead to the equilibrium point and others move away (this is the situation pictured in Figure 3.11). Finally, an



$$\dot{x}_1 = 2x_1 - x_2$$
$$\dot{x}_2 = -x_1 + 2x_2$$

Figure 3.11: Phase portrait and time domain simulation for a system with a single unstable equilibrium point.

equilibrium point which is stable but not asymptotically stable (such as the one in Figure 3.9 is called a *center*.

Stability Analysis via Linear Approximation

An important feature of differential equations is that it is often possible to determine the local stability of an equilibrium point by approximating the system by a linear system. We shall explore this concept in more detail later, but the following example illustrates the basic idea.

Example 3.7 *(Linear approximation of an inverted pendulum).* Consider again the inverted pendulum, whose (simplified) dynamics are given by

$$\frac{dx}{dt} = \begin{bmatrix} x_2 \\ \sin x_1 + u \cos x_1 \end{bmatrix}$$
$$y = x_1$$

If we assume that the angle $x_1 = \theta$ remains small, then we can replace $\sin\theta$ with θ and $\cos\theta$ with 1, which gives the approximate system

$$\frac{dx}{dt} = \begin{bmatrix} x_2 \\ x_1 + u \end{bmatrix}$$
$$y = x_1$$

We see that this system is linear and it can be shown that when x_1 is small, it gives an excellent approximation to the original dynamics.

In particular, it can be shown that if a linear approximation has either asymptotically stable or unstable equilibrium point, then the local stability of the original system must be the same.

The fact that a linear model can sometimes be used to study the behavior of a nonlinear system near an equilibrium point is a powerful one. Indeed, we can take this even further and use local linear approximations of a nonlinear system to design a feedback law that keeps the system near its equilibrium point (design of dynamics). By virtue of the fact that the closed loop dynamics have been chosen to stay near the equilibrium, we can even use the linear approximation to design the feedback that ensures this condition is true!

Lyapunov functions

A powerful tool for determining stability is the use of Lyapunov functions. A *Lyapunov function* $V : \mathbb{R}^n \to \mathbb{R}$ is an energy-like function that can be used to determine stability of a system. Roughly speaking, if we can find a non-negative function that always decreases along trajectories of the system, we can conclude that the minimum of the function is a stable equilibrium point (locally).

To describe this more formally, we start with a few definitions. We say that a continuous function $V(x)$ is *positive definite* if $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$. We will often write this as $V(x) \succ 0$. Similarly, a function is *negative definite* if $V(x) < 0$ for all $x \neq 0$ and $V(0) = 0$. We say that a function $V(x)$ is *positive semidefinite* if $V(x)$ can be zero at points other than $x = 0$ but otherwise $V(x)$ is strictly positive. We write this as $V(x) \succeq 0$ and define negative semi-definite functions analogously.

To see the difference between a positive definite function and a positive semi-definite function, suppose that $x \in \mathbb{R}^2$ and let

$$V_1(x) = x_1^2 \qquad V_2(x) = x_1^2 + x_2^2.$$

Both V_1 and V_2 are always non-negative. However, it is possible for V_1 to be zero even if $x \neq 0$. Specifically, if we set $x = (0, c)$ where $c \in \mathbb{R}$ is any non-zero number, then $V_1(x) = 0$. $V_2(x) \neq 0$ and it is easy to see that $V_2(x) = 0$ if and only if $x = (0, 0)$. Thus $V_1(x) \succeq 0$ and $V_2 \succ 0$.

We can now characterize the stability of a system

$$\dot{x} = F(x) \qquad x \in \mathbb{R}^n.$$

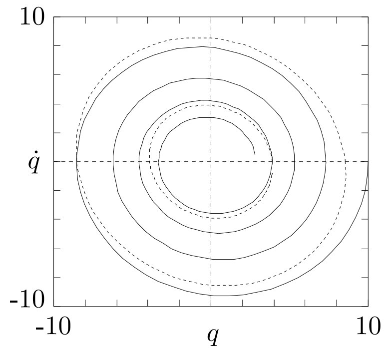


Figure 3.12: Geometric illustration of Lyapunov's stability theorem. The dotted ellipses correspond to level sets of the Lyapunov function; the solid line is a trajectory of the system.

Theorem 3.1. *Let $V(x)$ be a non-negative function on $\mathbb{R}^n$ and let $\dot{V}$ represent the time derivative of V along trajectories of the system dynamics:*

$$\dot{V}(x) = \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x}F(x).$$

Let $B_r = B_r(0)$ be a ball of radius r around the origin. If there exists $r > 0$ such that $\dot{V} \preceq 0$ for all $x \in B_r$, then $x = 0$ is locally stable in the sense of Lyapunov. If $\dot{V} \prec 0$ in B_r, then $x = 0$ is locally asymptotically stable.

If V satisfies one of the conditions above, we say that V is a (local) *Lyapunov function* for the system. These results have a nice geometric interpretation. The level curves for a positive definite function are closed contours as shown in Figure 3.12. The condition that $\dot{V}(x)$ is negative simply means that the vector field points towards lower level curves. This means that the trajectories move to smaller and smaller values of V and, if $\dot{V} \prec 0$, then x must approach 0.

A slightly more complicated situation occurs if $\dot{V}(x) \preceq 0$. In this case it is possible that $\dot{V}(x) = 0$ when $x \neq 0$ and hence x could stop decreasing in value. The following example illustrates these two cases.

Example 3.8. Consider the second order system

$$\begin{aligned}\dot{x}_1 &= -ax_1\\ \dot{x}_2 &= -bx_1 - cx_2.\end{aligned}$$

Suppose first that $a, b, c > 0$ and consider the Lyapunov function candidate

$$V(x) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2.$$

Taking the derivative of V and substituting the dynamics, we have

$$\dot{V}(x) = -ax_1^2 - bx_1x_2 - cx_2^2.$$

To check whether this is negative definite, we complete the square by writing

$$\dot{V} = -a(x_1 + \frac{b}{a}x_2)^2 - (c - \frac{b^2}{a})x_2^2.$$

Clearly $\dot{V} \prec 0$ if $a > 0$ and $(c - \frac{b^2}{a}) > 0$.

Suppose now that $a, b, c > 0$ and $c = b^2/a$. Then the derivative of the Lyapunov function becomes

$$\dot{V} = -a(x_1 + \frac{b}{a}x_2)^2 \leq 0.$$

This function is not negative definite since if $x_1 = -\frac{b}{a}x_2$ then $\dot{V} = 0$ but $x \neq 0$. Hence we cannot include asymptotic stability, but we *can* say the system is stable (in the sense of Lyapunov).

The fact that $\dot{V}$ is not negative definite does not mean that this system is not asymptotically stable. As we shall see in Chapter 4, we can check stability of a linear system by looking at the eigenvalues of the system matrix for the model

$$\dot{x} = \begin{bmatrix} -a & 0 \\ -b & -c \end{bmatrix} x.$$

By inspection (since the system is lower triangular), the eigenvalues are $\lambda_1 = -a < 0$ and $\lambda_2 = -c < 0$, and hence the system can be shown to be asymptotically stable.

To demonstrate asymptotic stability using Lyapunov functions, we must try a different Lyapunov function candidate. Suppose we try

$$V(x) = \frac{1}{2}x_1^2 + \frac{1}{2}(x_2 - \frac{b}{c-a}x_1)^2.$$

It is easy to show that $V(x) \succ 0$ since $V(x) \geq 0$ for all x and $V(x) = 0$ implies that $x_1 = 0$ and $x_2 - \frac{b}{c-a}x_1 = x_2 = 0$. We now check the time derivative of V:

$$\begin{aligned}
\dot{V}(x) &= x_1\dot{x}_1 + (x_2 - \frac{b}{c-a}x_1)(\dot{x}_2 - \frac{b}{c-a}\dot{x}_1) \\
&= -ax_1^2 + (x_2 - \frac{b}{c-a}x_1)(-bx_1 - cx_2 + \frac{b}{c-a}x_1) \\
&= -ax_1^2 - c(x_2 - \frac{b}{c-a}x_1)^2.
\end{aligned}$$

We see that $\dot{V} \prec 0$ as long as $c \neq a$ and hence we can show stability except for this case (explored in more detail in the exercises).

As this example illustrates, Lyapunov functions are not unique and hence we can use many different methods to find one. Indeed, one of the main difficulties in using Lyapunov functions is finding them.[2] It turns out that Lyapunov functions can always be found for any stable system (under certain conditions) and hence one knows that if a system is stable, a Lyapunov function exists (and vice versa).

Lasalle's Invariance Principle

Lasalle's theorem enables one to conclude asymptotic stability of an equilibrium point even when one can't find a $V(x)$ such that $\dot{V}(x,t)$ is locally negative definite. However, it applies only to time-invariant or periodic systems. We will deal with the time-invariant case and begin by introducing a few more definitions. We denote the solution trajectories of the time-invariant system

$$\dot{x} = F(x) \tag{3.7}$$

as $s(t, x_0, t_0)$, which is the solution of equation (3.7) at time t starting from x_0 at t_0.

Definition 3.1. The set $S \subset \mathbb{R}^n$ is the ω *limit set* of a trajectory $s(\,\cdot\,, x_0, t_0)$ if for every $y \in S$, there exists a strictly increasing sequence of times t_n such that

$$s(t_n, x_0, t_0) \to y$$

as $t_n \to \infty$.

Definition 3.2. The set $M \subset \mathbb{R}^n$ is said to be an (positively) *invariant set* if for all $y \in M$ and $t_0 \geq 0$, we have

$$s(t, y, t_0) \in M \quad \forall t \geq t_0.$$

It may be proved that the ω limit set of every trajectory is closed and invariant. We may now state Lasalle's principle.

Theorem 3.2 (Lasalle's principle). *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally positive definite function such that on the compact set $\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}$ we have $\dot{V}(x) \leq 0$. Define*

$$S = \{x \in \Omega_c : \dot{V}(x) = 0\}.$$

[2] Fortunately, there are systematic tools available for searching for special classes of Lyapunov functions, such as sums of squares [**?**].

As $t \to \infty$, the trajectory tends to the largest invariant set inside S; i.e., its ω limit set is contained inside the largest invariant set in S. In particular, if S contains no invariant sets other than $x = 0$, then 0 is asymptotically stable.

A global version of the preceding theorem may also be stated. An application of Lasalle's principle is as follows:

Example 3.9 *(Nonlinear spring mass system with damper).* Consider a nonlinear, damped spring mass system with dynamics

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -f(x_2) - g(x_1) \end{aligned}$$

Here f and g are smooth functions modeling the friction in the damper and restoring force of the spring, respectively. We will assume that f, g are both passive; that is,

$$\begin{aligned} \sigma f(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0] \\ \sigma g(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0] \end{aligned}$$

and equality is only achieved when $\sigma = 0$.

Consider the Lyapunov function candidate

$$V(x) = \frac{x_2^2}{2} + \int_0^{x_1} g(\sigma)\, d\sigma,$$

which is positive definite and gives

$$\dot{V}(x) = -x_2 f(x_2).$$

Choosing $c = \min(V(-\sigma_0, 0), V(\sigma_0, 0))$ so as to apply Lasalle's principle, we see that

$$\dot{V}(x) \leq 0 \quad \text{for } x \in \Omega_c := \{x : V(x) \leq c\}.$$

As a consequence of Lasalle's principle, the trajectory enters the largest invariant set in $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\} = \Omega_c \cap \{x_1, 0\}$. To obtain the largest invariant set in this region, note that

$$x_2(t) \equiv 0 \quad \Longrightarrow \quad x_1(t) \equiv x_{10} \quad \Longrightarrow \quad \dot{x}_2(t) = 0 = -f(0) - g(x_{10}),$$

where x_{10} is some constant. Consequently, we have that

$$g(x_{10}) = 0 \quad \Longrightarrow \quad x_{10} = 0.$$

Thus, the largest invariant set inside $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\}$ is the origin and, by Lasalle's principle, the origin is locally asymptotically stable.

3.4 Shaping Dynamic Behavior

From the standpoint of control it is interesting to explore the possibilities of shaping the dynamic behavior of a system.

Forcing Desired Equilibrium Points

The first thing we may be interested in is to find controls that give desired equilibrium points. For that purpose we will consider a controlled system described by (3.1):

$$\begin{aligned}\dot{x} &= f(x,u)\\ y &= g(x),\end{aligned}$$

The equilibrium is given by

$$f(x_e, u_e) = 0$$

Assuming that the control signal can be chosen in the set $\mathcal{U} \subset \mathbb{R}^p$. The set is typically bounded reflecting the fact that external inputs (forces, electrical currents, chemical concentrations) are bounded. The equilibria that can be achieved are in the set

$$X_e = \{x : f(x,u) = 0, u \in \mathcal{U}\}.$$

It follows from this equation that the equilibria lie in a surface of dimension at most p, where p is the number of inputs.

Example 3.10 *(Predator Prey).* Returning to Example 3.6, we illustrate how dynamics can be shaped using the predator prey system. Suppose that we are able to modulate the food supply so that the steady state population of the rabbits is of the form $K = K_0 + u$, where u represents a control input that captures the way in which we modulate the food supply. The dynamics then become:

$$\begin{aligned}\dot{R} &= r_R R\left(1 - \frac{R}{K_0 + u}\right) - \frac{aRF}{1 + aRT_h} & R \geq 0\\ \dot{F} &= r_F F\left(1 - \frac{F}{kR}\right) & F \geq 0.\end{aligned}$$

To change the equilibrium population of the rabbits and foxes, we set $u = K_d - K_0$ where K_d represents the desired rabbit population (when no foxes are present). The resulting equilibrium point satisfies the equations:

$$\begin{aligned}0 &= R_e \cdot \left(\frac{r_R a T_h}{K_d} R_e^2 + \left(ak + \frac{r_R}{K_d} - r_R a T_h\right) R_e - r_R\right)\\ F_e &= kR_e.\end{aligned}$$

We see that we can control the equilibrium point along a one dimensional curve given by the solution to this equation as a function of $K_d > 0$.

Making the Equilibria Stable

Having found controls that give the desired equilibrium points, the next problem is to find feedback that makes the equilibria stable. We will explore this in much more detail in Chapter 5 (for linear systems), so we only illustrate the basic idea here. Let it suffice at this stage to say that only mild conditions on the system are required to do this.

Let x_e be an admissible equilibrium. The controller

$$u = K(x_e - x) \tag{3.8}$$

is called a *proportional controller* because the control action is a linear function of the deviation from the desired equilibrium. In Chapter 5 we will investigate when such a controller will give the desired result and methods for finding the matrix K will also be developed.

If there are model errors or disturbances the actual equilibrium may deviate from its desired value. This can be overcome by the controller

$$u(t) = K(x_e - x(t)) + K_i \int_0^\tau (x_e - x(\tau))d\tau. \tag{3.9}$$

This is called a PI (proportional and integral) controller because the control signal is the sum of two terms, one proportional to the error and the other proportional to the integral of the error. A PI controller forces the steady state error to be zero by applying an increasingly large input when $x(t) \neq x_e$ (through the integral term). If the system is asymptotically stable, then x and u will approach constant values and this implied that the integral term is identically zero, which in turn implies that $x(t) = x_e$.

3.5 System Performance Measures

So far, this chapter has focused on the stability characteristics of a systems. While stability is often a desirably feature, stability alone may not be sufficient in many applications. We will want to create feedback systems that quickly react to changes and give high performance in measurable ways. In this section, we consider two measures of performance that were introduced already in the last chapter: step response and frequency response.

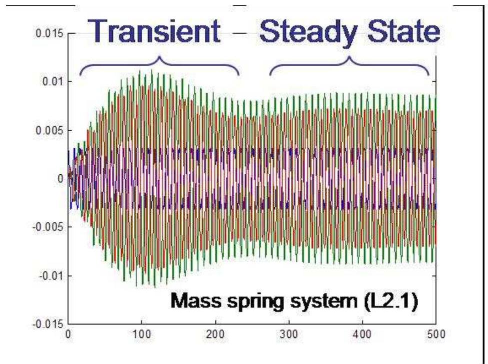


Figure 3.13: Transient versus steady state response

Transient Response versus Steady State Response

We return now to the case of an input/output state space system

$$\begin{aligned}\dot{x} &= f(x,u)\\ y &= h(x)\end{aligned} \tag{3.10}$$

where $x \in \mathbb{R}^n$ is the state and $u, y \in \mathbb{R}$ are the input and output. Generally, the response $y(t)$ to an input $u(t)$ will consist of two components—the transient response and steady state response. The transient response occurs in the first period of time after the input is applied and reflects the effect of the initial condition on the output. This notion is made precise for linear systems and is discussed further in Chapter 4. However, even for nonlinear systems one can often identify an initial portion of the response that is different than the remainder of the response.

The steady state response is the portion of the output response that reflects the long term behavior of the system under the given inputs. For inputs that are periodic, the steady state response will often also eventually be periodic. The steady state response in this case represents the portion of time over which the output is also periodic. An example of the transient and steady state response is shown in Figure 3.13.

Step Response

A particularly common form of input is a *step input*, which represents an abrupt change in input from one value to another. A *unit step* is defined as

$$u = \begin{cases} 0 & t = 0 \\ 1 & t > 0. \end{cases}$$

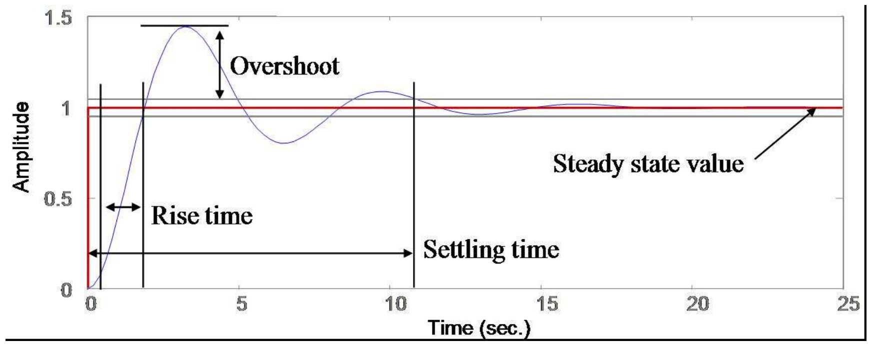


Figure 3.14: Sample step response

The *step response* of the system (3.10) is defined as the output $y(t)$ starting from zero initial condition (or the appropriate equilibrium point) and given a *step input*. We note that the step input is discontinuous and hence is not physically implementable. However, it is a convenient abstraction that is widely used in studying input/output systems.

A sample step response is shown in Figure 3.14. Several terms are used when referring to a step response:

Steady state value The steady state value of a step response is the final level of the output, assuming it converges.

Rise time The rise time is the amount of time required for the signal to go from 5% of its final value to 95% of its final value. It is possible to define other limits as well, but in this book we shall use these percentages unless otherwise indicated.

Overshoot The overshoot is the percentage of the final value by which the signal initially rises above the final value. This usually assumes that future values of the signal do not overshoot the final value by more than this initial transient, otherwise the term can be ambiguous.

Settling time The settling time is the amount of time required for the signal to stay within 5% of its final value for all future times.

For a general nonlinear system, these performance measures can depend on the amplitude of the input step, but for linear system it can be shown that the quantities defined above are independent of the size of the step.

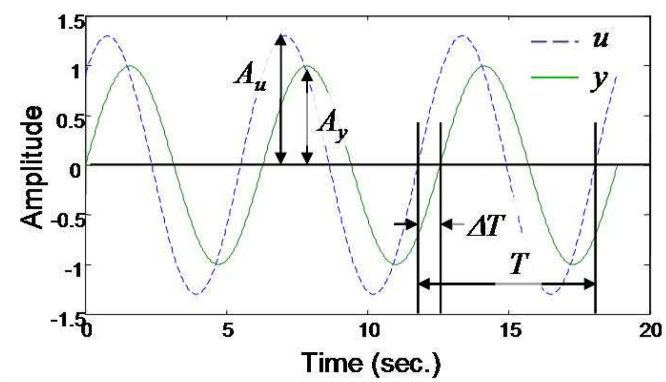


Figure 3.15: Frequency response, showing gain and phase.

Frequency Response

The frequency response of an input/output system measures the way in which the system responds to a sinusoidal excitation on one of its inputs. As we have already seen (and will see in more detail later), for linear systems the particular solution associated with a sinusoidal excitation is itself a sinusoid at the same frequency. Hence we can compare the magnitude and phase of the output sinusoid as compared to the input. More generally, if a system has a sinusoidal output response at the same frequency as the input forcing, we can speak of the frequency response.

Frequency response is typically measured in terms of *gain* and *phase* at a given forcing frequency, as illustrated in Figure 3.15. The gain the system at a given frequency is given by the ratio of the amplitude of the output to that of the input. The phase is given by the the fraction of a period by which the output differs from the input. Thus, if we have an input $u = A_u \sin(\omega t + \psi)$ and output $y = A_y \sin(\omega t + \phi)$, we write

$$\text{gain}(\omega) = \frac{A_y}{A_u} \qquad \text{phase}(\omega) = \phi - \psi.$$

If the phase is positive, we say that the output "leads" the input, otherwise we say it "lags" the input.

3.6 Exercises