

Computer Systems Modelling

Course Notes

Computer Science Tripos, Part II
Michaelmas Term 2026

S. Keshav
Department of Computer Science and Technology
University of Cambridge

Chapters 5, 6, and 8 are excerpted, with corrections, from S. Keshav, *Mathematical Foundations of Computer Networking*, Addison-Wesley, 2012. The simulation chapter is adapted from lecture slides by the late Professor R. J. Gibbens.

Contents

About These Notes **4**

Chapter S: Simulation **9**

S.1 Overview 9

S.2 Discrete Event Simulation 11

S.3 Random Number Generation 13

S.4 Generating Random Variables 14

S.5 Simulation Performance Measures 17

S.6 Confidence Intervals for the Mean 18

S.7 Stopping Rules 21

S.8 Variance Reduction 21

S.9 Additional Video Material 22

S.10 Exercises 22

Programming Assignment: Discrete-Event Simulation of a Queue 24

Chapter 6: Stochastic Processes and Queueing Theory **26**

6.1 Overview 26

6.2 Stochastic Processes 29

6.3 Continuous-Time Markov Chains 44

6.4 Birth-Death Processes 46

6.5 The M/M/1 Queue 55

6.6 Two Variations on the M/M/1 Queue 59

6.7 Other Queueing Systems 64

6.8 Further Reading 66

Section 6.9: Exercises 67

Supplementary Note: The M/G/1 Queue 71

Corrections to Chapter 6 74

Chapter 5: Signals, Systems, and Transforms **77**

5.1 Background 77

5.2 Signals 89

5.3 Systems 92

5.4 Analysis of a Linear Time-Invariant System 93

5.5 Transforms 99

5.6 The Fourier Series 100

5.7 The Fourier Transform and Its Properties 104
5.8 The Laplace Transform 113
Sections 8.3 and 8.4: First-Order and Second-Order Systems 121
8.3 A First-Order System 121
8.4 A Second-Order System 123
Section 5.12: Exercises 129
Corrections to Chapter 5 131
Chapter 8: Elements of Control Theory 134
8.1 Overview of a Controlled System 135
8.2 Modeling a System 138
8.5 Basics of Feedback Control 143
8.6 PID Control 148
8.7 Advanced Control Concepts 153
8.8 Stability 157
8.12 Further Reading 167
8.13 Exercises 167
Corrections to Chapter 8 169
Supplementary Note: Model Predictive Control 172
Supplementary Note: From a Differential Equation to Pole Placement 182
Background 191
Sections 3.5 and 3.6: Eigenvalues, Eigenvectors, and Stochastic Matrices 192
3.5 Linear Transformations, Eigenvalues, and Eigenvectors 192
3.6 Stochastic Matrices 205
Corrections to Sections 3.5 and 3.6 211
Section 8.11: Partial Fraction Expansion 213
Answers to Exercises 216
Past Assessment: Simulation and Queueing Theory (2023, with solutions) 240
Past Assessment: Simulation and Queueing Theory (2025, with solutions) 243
Past Assessment: Signals, Systems, and Control (2023, with solutions) 248
Past Assessment: Signals, Systems, and Control (2025, with solutions) 251

About These Notes

These course notes are excerpted from my textbook *Mathematical Foundations of Computer Networking*, published by Addison-Wesley in 2012. I have enhanced them by adding links to YouTube videos as well as hands-on lab material. The simulation chapter is adapted from lecture slides by the late Professor R. J. Gibbens. The notes also include a supplementary, non-examinable note on model predictive control, and background material from prior courses (eigenvalues and eigenvectors, stochastic matrices, and partial fraction expansion) that you may want to refresh.

To use these notes, I suggest the following process:

1. Read the text, which will give you an intuitive overview of each concept, followed by mathematical details.
2. Attend the lecture, ideally after looking through the notes, so you know what material you find more challenging.
3. If you feel that you need more help understanding a concept, click on the video link for a more in-depth explanation. Each video is annotated with its duration.
4. Every concept is illustrated with an example. Read the example problem and attempt to answer it yourself. Then read the answer and compare your answer to the one in the text. This will help you identify where your grasp of the concept may be shaky.
5. Answers to the exercises are at the end of these notes, followed by past examination questions with their solutions.

Interactive labs accompany most sections. A boxed "Lab" link in the margin marks the place where each lab is relevant, and the full list of labs is given after the course outline.

Introductory Videos

- Introduction to computer systems modelling 
- What is a systems model? 

This course notes package has been thoroughly checked for errors, with numerous corrections made in the text. Nevertheless, if you find an error, please write to the course instructor with a correction.

Course Outline

Lecturer:	Prof. S. Keshav
Taken by:	Part II CST
No. of lectures:	16
Prerequisite courses:	Introduction to Probability (IA), Foundations of Data Science (IB)
Teaching style:	In-person lectures
Term:	Michaelmas

Aims

The aims of this course are to introduce the concepts and principles of mathematical modelling and simulation, with particular emphasis on using queuing theory and control theory for understanding the behaviour of computer and communications systems.

Syllabus

1. Overview of computer systems modeling using both analytic techniques and simulation.
2. Introduction to discrete event simulation.
    a. Simulation techniques
    b. Random number generation methods
    c. Statistical aspects: confidence intervals, stopping criteria
    d. Variance reduction techniques.
3. Stochastic processes (builds on starred material in Foundations of Data Science)
    a. Discrete and continuous stochastic processes.
    b. Markov processes and Chapman-Kolmogorov equations.
    c. Discrete time Markov chains.
    d. Ergodicity and the stationary distribution.
    e. Continuous time Markov chains.
    f. Birth-death processes, flow balance equations.
    g. The Poisson process.

4. Queueing theory
    a. The M/M/1 queue in detail.
    b. The equilibrium distribution with conditions for existence and common performance metrics.
    c. Extensions of the M/M/1 queue: the M/M/k queue, the M/M/infinity queue.
    d. Queueing networks. Jacksonian networks.
    e. The M/G/1 queue.
5. Signals, systems, and transforms
    a. Discrete- and continuous-time convolution.
    b. Signals. The complex exponential signal.
    c. Linear Time-Invariant Systems. Modeling practical systems as an LTI system.
    d. Fourier and Laplace transforms.
    e. First-order and second-order systems.
6. Control theory.
    a. Controlled systems. Modeling controlled systems.
    b. State variables. The transfer function model.
    c. Feedback control. PID control.
    d. Stability. BIBO stability. Lyapunov stability.
    e. Introduction to Model Predictive Control.

Objectives

At the end of the course students should

- Be aware of different approaches to modeling a computer system; their pros and cons
- Be aware of the issues in building a simulation of a computer system and analysing the results obtained
- Understand the concept of a stochastic process and how they arise in practice
- Be able to build simple Markov models and understand the critical modelling assumptions
- Be able to solve simple birth-death processes
- Understand and use M/M/1 queues to model computer systems

- Be able to model a computer system as a linear time-invariant system
- Understand the dynamics of a second-order controlled system
- Design a PID control for an LTI system
- Understand what is meant by BIBO and Lyapunov stability

Assessment

Two questions during the finals. There is one optional programming assignment to build a discrete event simulator and use it to simulate an M/M/1 and an M/D/1 queue.

Reference Books

- Keshav, S. (2012). *Mathematical Foundations of Computer Networking*. Addison-Wesley. A customized PDF will be made available to class participants.
- Kleinrock, L. (1975). *Queueing Systems, Vol. 1: Theory*. Wiley.
- Kraniauskas, P. (1992). *Transforms in Signals and Systems*. Addison-Wesley Longman.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis*. Wiley.

Interactive Labs

Each lab runs in a web browser. The full list is also at `https://svr-sk818-web.cl.cam.ac.uk/keshav/teaching/csm-labs.html`.

Lab	Where it is used
Simulation	
Sampling by Inversion	Section S.4.1
Confidence Intervals	Section S.6
Queueing theory	
Random Walk	Section 6.2
Eigenvalues	Background: Section 3.5.2
Markov Chains	Section 6.2
CTMC Rates	Section 6.3
Birth–Death Processes	Section 6.4
M/M/1 Queue	Section 6.5
M/M/1/K Queue	Section 6.6
M/D/1 and M/G/1 Queues	The M/G/1 note
Signals and systems	
Phasors	Section 5.1
Keying	Section 5.1
Convolution	Section 5.1
Complex Exponentials	Section 5.2
System Properties	Section 5.3
LTI Systems	Section 5.4
Fourier Series	Section 5.6
Fourier Transform	Section 5.7
Laplace Transform	Section 5.8
First- and Second-Order Systems	Sections 8.3–8.4
Control theory	
Feedback Control	Section 8.5
PID Control	Section 8.6
Stability	Section 8.8



S

Simulation

This chapter, apart from Section S.6 on confidence intervals for the mean, is adapted from lecture slides by the late Professor R. J. Gibbens, who taught this material in the Department until 2018.

Many systems of practical interest are too complex to analyse with simple mathematical models that we study later in the course. A simulation sidesteps this difficulty: instead of solving equations that describe the system, we build a program that mimics the system's behaviour over time and then measure the program. This chapter describes how such simulations are built and how their outputs should be interpreted. We first describe the types of simulation and the structure of a discrete event simulator, using a car wash as a running example. We then discuss how to generate random numbers and, from them, random variables with any desired distribution. Finally, we study the statistical analysis of simulation outputs: confidence intervals, rules for deciding when to stop a simulation, and techniques for reducing the variance of the estimates.

S.1 Overview

Simulation allows arbitrarily complex systems to be evaluated. It captures the dynamic behaviour of a system: the dynamics of a complex system are obtained by tracking the evolution of the system over time. Examples include the design of communication networks and the modelling of road traffic.


Introduction

S.1.1 Types of Simulation

Simulations can be classified by the nature of their state variables.

- In a *discrete state* (or discrete event) simulation, the state of the system is described by discrete variables, such as the number of jobs at different stages of a production line.
- In a *continuous state* simulation, the state is described by continuous variables, such as the quantities of various chemical reagents.

A similar distinction may be drawn between *discrete time* and *continuous time* simulations, depending on whether the system state is defined only at certain discrete times or at all times.

We will be concerned with discrete event simulation, because it applies most naturally to computer systems, whose state variables are generally discrete. Here are some examples of such state variables: the state of each job in the system; the number of jobs of each kind; and the number or availability of resources or devices.

S.1.2 State Space and Trajectories

State space

Simulation is of particular use when we study systems that are not in steady state. The execution of a simulation model consists of a series of changes in the system's state. We always follow the dynamic evolution of the system, even if we only want a mean value. Each execution of the simulation therefore results in a *trajectory*: a sample path through the state space, such as the one in Figure S.1.

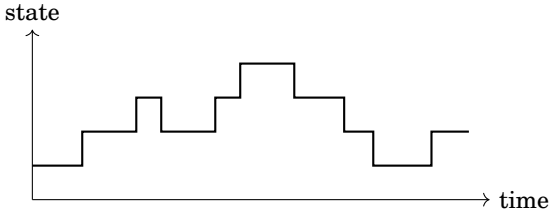


Figure S.1: A trajectory of a discrete event simulation. The state, here the number of jobs in a system, changes only when an event occurs.

S.1.3 Advantages and Disadvantages

Pros and cons

The principal advantage of simulation is its extreme generality. However, simulation has four disadvantages.

1. The design, coding, and debugging of a simulation program are often time consuming, and the program may be difficult to understand. The

task may even approach that of implementing the system and measuring it directly.

2. Generality can lead to complexity that obscures understanding of the model. Fine details may be irrelevant if the workload is poorly approximated.
3. Executing the simulation can be computationally expensive.
4. Statistical analysis of the output can be problematic. For example, how long should we run the simulation before averaging the results?

We address the last of these in Sections S.6 to S.8.

S.2 Discrete Event Simulation

S.2.1 The Simulation Loop


Simulation loop

Each *event* in a simulation contains a time stamp identifying when it occurs, and denotes some change in the state of the system being simulated, such as the arrival of an IP packet. Events are ordered in time in an *event list*. Each entry in the list is a tuple (t, m), where t is the time stamp and m describes the event. The simulator then executes Algorithm 1.

Algorithm 1 The discrete event simulation loop

```
1: Initialize the clock to 0
2: Initialize the event list
3: while the termination criterion is not met do
4:     Remove the earliest tuple (t, m) from the event list
5:     Update the clock to t
6:     Simulate the effect of event m at time t
7: end while
```

It is crucial that the simulator always selects the event with the earliest time stamp. Simulating an event usually generates new events, which are inserted into the event list in time order. Note that much of the running time of a simulation is often spent maintaining the chronological order of the event list, so the choice of data structure for this list matters.

S.2.2 Simulation Variables

A discrete event simulation involves both events and variables, and we keep track of the variables as they change during the simulation. There are three types of variables.

1. A *time variable*, t, records the passage of time during the simulation.

2. *Counter variables* count the number of times that certain events have occurred by time t.
3. *System state variables* define the state of the system at time t.

As simulation events occur, we change these variables and gather any output of interest.

S.2.3 Example: A Car Wash

Car wash

Consider a car wash that can serve one car at a time. Is it viable to install a second car wash? To answer this question, we model the system shown in Figure S.2. It has an entrance (the *source*), an attendant (*att*), two car washes (*cw1* and *cw2*), and an exit (the *sink*). The model works as follows.

- The interarrival times of cars that need to be washed are random.
- If both car washes are busy, an arriving car joins the queue.
- When a car wash becomes idle, the car at the head of the queue is sent to it.

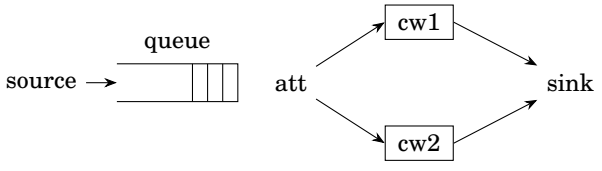


Figure S.2: A car wash with two washing bays. The attendant sends the car at the head of the queue to whichever car wash becomes idle.

The model has four types of events, each a message from one component to another:

- source→att: a car arrives in the system;
- att→cw: a car is sent from the queue to a car wash;
- cw→att: a car wash becomes idle;
- cw→sink: a car departs the system.

Note that the departure of a car is modelled by two events: one represents the car leaving the system, and the other signals to the attendant that the car wash is now idle. This is similar to object-oriented programming, in which objects communicate solely by method invocations.

Suppose that cw1 takes 8 minutes and cw2 takes 10 minutes to wash a car, and that cars arrive at times 6, 11, 12, 17, and 19. Table S.1 shows a possible sequence of events. The event column numbers the messages that move a car; the messages that signal an idle car wash are not numbered. At time 12,

car 3 arrives to find both car washes busy, so it waits in the queue until cw1 becomes idle at time 14.

Table S.1: A possible sequence of events in the car wash simulation

Message	Event	Time	Sender	Receiver	Content
1	–	0	cw1	att	idle
2	–	0	cw2	att	idle
3	1	6	source	att	car 1
4	2	6	att	cw1	car 1
5	3	11	source	att	car 2
6	4	11	att	cw2	car 2
7	5	12	source	att	car 3
8	6	14	cw1	sink	car 1
9	–	14	cw1	att	idle
10	7	14	att	cw1	car 3
11	8	17	source	att	car 4
12	9	19	source	att	car 5
13	10	21	cw2	sink	car 2
14	–	21	cw2	att	idle

S.3 Random Number Generation


Watch this first

A simulation of a system with random interarrival and service times needs a supply of random numbers. It is important not to generate random numbers with an ad hoc method: complex algorithms do not necessarily generate random outputs. Some operating systems can generate random numbers from physical sources, such as key strokes or network interarrival times. However, these mechanisms are not usually suited to generating large volumes of random data.

How, then, can we algorithmically generate a long random sequence? The answer is that the sequence is not random, but appears random as far as can be determined from statistical tests. Such a sequence is termed *pseudo-random*. Here are some important requirements for a pseudo-random number generator.


Requirements

- The algorithm should be fast.
- Its storage requirements should be low.
- The random sequence should only repeat after a very long period.
- The sequence should possess two important statistical properties: uniformity and independence.

For ease of implementation and speed, most random number generators use an integer representation over an interval $[0, m)$. Given a value in $[0, m)$, we obtain a value in the range $[0, 1)$ by dividing by m.

S.4 Generating Random Variables

We assume from now on that we have an efficiently generated supply of random numbers distributed as independent $U(0, 1)$ random variables. How do we generate random variables with other distributions? We will find that with a suitable transformation, the resulting random variables appear to have been drawn from any desired probability distribution.

S.4.1 Discrete Distributions and the Inverse Transform Method

Lab: Sampling by inversion

Suppose we are given a distribution p_i, with $0 \le p_i \le 1$ for $i \in I = \{0, 1, \ldots\}$ and $\sum_{i \in I} p_i = 1$, and that we wish to generate a discrete random variable X whose probability mass function is

$$\mathbb{P}(X = x_i) = p_i \qquad \forall i \in I. \tag{EQ S.1}$$

This may be done by generating a pseudo-random variable U with distribution $U(0, 1)$ and setting

$$X = \begin{cases} x_0 & \text{if } U < p_0, \\ x_1 & \text{if } p_0 \le U < p_0 + p_1, \\ x_2 & \text{if } p_0 + p_1 \le U < p_0 + p_1 + p_2, \\ \quad \vdots & \\ x_i & \text{if } \sum_{j=0}^{i-1} p_j \le U < \sum_{j=0}^{i} p_j, \\ \quad \vdots & \end{cases} \tag{EQ S.2}$$

Intuitively, we divide the unit interval into consecutive pieces of lengths $p_0, p_1, p_2, \ldots$ and set $X = x_i$ if U falls into the ith piece. Because U is $U(0, 1)$, the probability that it falls into a piece equals the length of that piece:

$$\mathbb{P}(X = x_i) = \mathbb{P}\left(\sum_{j=0}^{i-1} p_j \le U < \sum_{j=0}^{i} p_j \right) = p_i. \tag{EQ S.3}$$

If we write $F(x_k) = \sum_{i=0}^{k} p_i$, the process of generating X is as follows.

1. Generate a random number U.
2. If $U < F(x_0)$, set $X = x_0$ and stop.
3. If $U < F(x_1)$, set $X = x_1$ and stop.

4. Continue in this way until X is set.

If the x_i are ordered so that $x_0 < x_1 < \cdots$, then F is the cumulative distribution function of X, and this is equivalent to choosing $X = x_i$ if $F(x_{i-1}) \le U < F(x_i)$. For this reason, the method is known as the *inverse transform method*.

S.4.2 Geometric Random Variables

A geometric random variable counts the number of independent trials, each succeeding with probability p, up to and including the first success. Here,

$$p_i = \mathbb{P}(X = i) = p(1-p)^{i-1}, \qquad i = 1, 2, \ldots \tag{EQ S.4}$$

because the first $i-1$ trials must fail and the ith must succeed. Therefore,

$$\sum_{j=1}^{i-1} p_j = 1 - \mathbb{P}(X > i-1) = 1 - (1-p)^{i-1}, \tag{EQ S.5}$$

since $X > i-1$ exactly when the first $i-1$ trials all fail. Thus, we can use the inverse transform method by setting X to the value of i such that

$$1 - (1-p)^{i-1} \le U < 1 - (1-p)^{i}. \tag{EQ S.6}$$

A little algebra lets us compute i directly. The inequality is equivalent to $(1-p)^i < 1 - U \le (1-p)^{i-1}$. Since $1-U$ is also $U(0,1)$, we may replace it by U. Taking logarithms, and remembering that $\log(1-p) < 0$, so that dividing by it reverses the inequalities, gives $i - 1 \le \log(U)/\log(1-p) < i$. Hence

$$X = \left\lfloor \frac{\log(U)}{\log(1-p)} \right\rfloor + 1. \tag{EQ S.7}$$

S.4.3 Poisson Random Variables

For a Poisson random variable with parameter $\lambda > 0$,

$$p_i = \mathbb{P}(X = i) = e^{-\lambda} \frac{\lambda^i}{i!}, \qquad i = 0, 1, \ldots \tag{EQ S.8}$$

It follows that

$$p_{i+1} = \frac{\lambda}{i+1} p_i, \qquad i = 0, 1, \ldots \tag{EQ S.9}$$

so each probability can be computed cheaply from the previous one. This gives Algorithm 2, in which F holds the running sum $\sum_{j=0}^{i} p_j$. Similar algorithms can be formulated for other discrete distributions.

S.4.4 Continuous Random Variables

Continuous distributions

Algorithm 2 Generating a Poisson(λ) random variable

```
Generate a random number U
Set i = 0, p = e^{-λ}, F = p
while U ≥ F do
    Set p = λp/(i + 1), F = F + p, i = i + 1
end while
Set X = i
```

The inverse transform method extends to continuous random variables. Let U be a random variable with distribution $U(0,1)$, and let F_X be a continuous, strictly increasing distribution function, so that its inverse F_X^{-1} exists. Then

$$X = F_X^{-1}(U) \qquad \textbf{(EQ S.10)}$$

is a random variable with distribution function $F_X(x)$.

Proof, part 1

Proof: Because F_X is strictly increasing, applying it to both sides of an inequality preserves the inequality. Therefore,

$$\begin{aligned}\mathbb{P}(X \le x) &= \mathbb{P}\left(F_X^{-1}(U) \le x\right) \\ &= \mathbb{P}\left(F_X\left(F_X^{-1}(U)\right) \le F_X(x)\right) \\ &= \mathbb{P}\left(U \le F_X(x)\right) \\ &= F_X(x), \qquad \textbf{(EQ S.11)}\end{aligned}$$

where the last step holds because U is $U(0,1)$, so $\mathbb{P}(U \le u) = u$ for $0 \le u \le 1$. □

Proof, part 2

So, the inverse transform method for continuous random variables with distribution function $F_X(x)$ generates $X = F_X^{-1}(U)$, where U is $U(0,1)$. We now apply it to two common distributions.

EXAMPLE S.1: UNIFORM DISTRIBUTION

Consider the uniform random variable on the interval (a,b), with distribution function

$$F_X(x) = \frac{x-a}{b-a}$$

for x in (a,b). Given a pseudo-random uniform value U in the interval $(0,1)$, we set $X = F_X^{-1}(U)$, so that

$$U = F_X(X) = \frac{X-a}{b-a},$$

and therefore $X = (b-a)U + a$.

EXAMPLE S.2: EXPONENTIAL DISTRIBUTION

Exponential

For the exponential distribution with parameter λ,

$$F_X(x) = \begin{cases} 1 - e^{-\lambda x} & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Given U, a pseudo-random variable with distribution $U(0,1)$, we set $X = F_X^{-1}(U)$. Thus $U = F_X(X) = 1 - e^{-\lambda X}$, so that

$$X = -\frac{1}{\lambda}\log(1-U).$$

Note that both U and $1-U$ are distributed as $U(0,1)$, so we might as well set

$$X = -\frac{1}{\lambda}\log(U).$$

S.4.5 Simulating a Poisson Process

Consider the problem of generating the first n event times of a Poisson process of rate λ. Recall that the interarrival times of a Poisson process are independent and exponentially distributed with parameter λ. One way to proceed, therefore, is to generate n independent random numbers $U_1, U_2, \ldots, U_n$, each from a $U(0,1)$ distribution, and set $X_i = -\frac{1}{\lambda}\log(U_i)$. Then the first n event times are $\sum_{i=1}^{j} X_i$ for $j = 1, 2, \ldots, n$.

To generate the events in the first T time units, we could proceed as above and stop when the sum first exceeds T. Algorithm 3 does this, building the sequence of event times in $S(\cdot)$.

Algorithm 3 Generating the events of a Poisson process in $[0, T]$

1: Set $t = 0$, $I = 0$
2: **loop**
3: Generate a random number U
4: Set $t = t - \frac{1}{\lambda}\log(U)$
5: **if** $t > T$ **then** stop
6: **end if**
7: Set $I = I + 1$, $S(I) = t$
8: **end loop**

S.5 Simulation Performance Measures

Because the simulation itself is stochastic, so too are its outputs. It is critical to realize that a simulation can only yield *estimates* of performance measures,

and there will always be some statistical error in these estimates. We can attempt to reduce the error in two ways:

- by running the simulation for longer, until sufficient samples have been taken; or
- by running the same simulation with different pseudo-random number sequences, and combining the results from multiple runs.

Metrics

We will say more about both approaches in Sections S.6 and S.7.

Here are four performance metrics commonly estimated by a simulation of a queueing system.

1. *Utilization*: the proportion of time that a server is busy. An estimate can be obtained by summing the busy times of the server and dividing by T, the simulation length. For a k-server system, the busy times can be summed over all servers and divided by kT.
2. *Throughput*: the mean number of customers receiving service per unit time.
3. *Mean waiting time in the queue*: the mean time that a customer spends in the queue before its service begins.
4. *Mean time in the system*: the mean time that a customer spends in the system, including its service time.

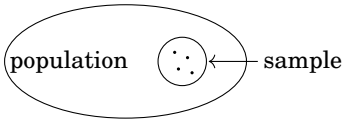
S.6 Confidence Intervals for the Mean

Lab: Confidence intervals

A simulation produces a sample of outputs from which we must estimate a true but unknown quantity, such as the mean waiting time. This section describes how to estimate such a mean and how to quantify the error in the estimate.

S.6.1 Population and Sample

A random variable X describes the population. We draw a sample of n observations $x_1, x_2, \ldots, x_n$ of X, and assume they are i.i.d.



The population has mean and variance

$$\mathbb{E}(X) = \mu, \qquad \mathrm{Var}(X) = \sigma^2. \tag{EQ S.12}$$

Expected values can be computed as sample means, which approach the population mean as $n \to \infty$.

S.6.2 The Sample Mean as a Random Variable

We define the sample mean

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} x_i. \qquad \textbf{(EQ S.13)}$$

The sample mean is a random variable, because it is a function of the random samples $x_1, \ldots, x_n$. Because the samples are i.i.d., its moments are

$$\mathbb{E}(\bar{X}_n) = \mu, \qquad \mathrm{Var}(\bar{X}_n) = \frac{\sigma^2}{n}, \qquad \textbf{(EQ S.14)}$$

but we know neither σ nor μ.

S.6.3 The Sample Variance

We define the sample variance

Sample variance

$$S_n^2 \triangleq \frac{\sum_{i=1}^{n}(x_i - \bar{X}_n)^2}{n-1}. \qquad \textbf{(EQ S.15)}$$

Then $\mathbb{E}(S_n^2) = \sigma^2$, and S_n^2 is computable.

Sample variance (2)

S.6.4 The Central Limit Theorem

Provided σ^2 is finite, standardised sums of i.i.d. variables are asymptotically Gaussian. Equivalently, as $n \to \infty$,

Central limit theorem

$$\frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} Z = N(0,1). \qquad \textbf{(EQ S.16)}$$

Since σ is unknown, we replace it with S_n. For large n we assume

$$\frac{\bar{X}_n - \mu}{S_n/\sqrt{n}} \approx Z. \qquad \textbf{(EQ S.17)}$$

For small n this assumption does not hold. If X is Gaussian, the statistic in Equation S.17 follows Student's t distribution with $n-1$ degrees of freedom, which has heavier tails than Z and approaches Z as $n \to \infty$. For small n, therefore, we replace $z_{\alpha/2}$ below with $t_{n-1,\alpha/2}$; for example, $t_{9,0.025} \approx 2.26$ rather than 1.96.

S.6.5 The Confidence Interval

Confidence intervals

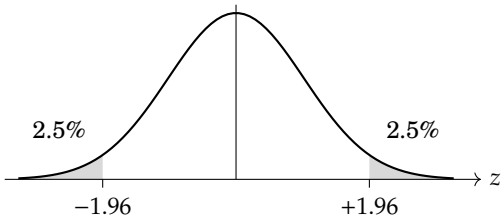


Figure S.3: Under the standard normal, 2.5% of the probability mass lies in each tail beyond ±1.96.

From Figure S.3,

$$\mathbb{P}(\underbrace{-1.96}_{-z_{\alpha/2}} < Z < \underbrace{+1.96}_{+z_{\alpha/2}}) = 1 - \underbrace{0.05}_{\alpha}. \qquad \textbf{(EQ S.18)}$$

Substituting Equation S.17 for Z,

$$\mathbb{P}\left(-z_{\alpha/2} < \sqrt{n}\,\frac{\bar{X}_n - \mu}{S_n} < z_{\alpha/2}\right) \approx 1 - \alpha, \qquad \textbf{(EQ S.19)}$$

and rearranging gives

$$\mathbb{P}\left(\bar{X}_n - z_{\alpha/2}\frac{S_n}{\sqrt{n}} < \mu < \bar{X}_n + z_{\alpha/2}\frac{S_n}{\sqrt{n}}\right) \approx 1 - \alpha. \qquad \textbf{(EQ S.20)}$$

This is the $(1-\alpha)$ confidence interval for μ. Note that μ is fixed and the interval is random, because its endpoints depend on the sample. If we repeated the experiment many times, about $100(1-\alpha)\%$ of the intervals constructed this way would contain μ.

S.6.6 Choosing n and Avoiding Correlation

Two questions remain: what should n be, and how do we avoid correlation between samples? Correlation matters because Equation S.14 assumes independent samples. If successive samples are positively correlated, σ^2/n underestimates $\mathrm{Var}(\bar{X}_n)$, and the interval in Equation S.20 is too narrow. We answer both questions as follows.

1. Group consecutive samples into blocks, and treat the block means as approximately independent samples.
2. Compute the confidence interval for the metric, and stop when its half-width (or its half-width relative to $\bar{X}_n$) is small enough. Stopping as soon as this happens slightly reduces coverage, so we also require a minimum number of samples before stopping.

The next section discusses these two steps in more detail.

S.7 Stopping Rules

How do we know when a simulation has been run long enough for its performance measures to be accurate? One approach is to repeat the simulation several times with different random seeds, to obtain many samples. These multiple runs are called *replications*. Having repeated the experiment n times, we can construct a confidence interval on our measure, say L, using Equation S.20. Although a large number of replications reduces the variance, each replication requires the simulation to re-stabilize, that is, to pass through its initial transient phase again.

Can we avoid this? We may be able to use a single, long run, and break up our samples into n blocks, each of which forms a sample L_i for $i = 1, 2, \ldots, n$. What could go wrong with this? Successive blocks may be correlated. Correlation does not bias the block means, but, as Section S.6.6 explains, it makes S_n^2/n underestimate the variance of $\bar{X}_n$, so the confidence interval is too narrow. If the block size is large, the correlation should be small. Explicit techniques exist to estimate the correlation and to determine a suitable block size.

The simulation can be stopped once the estimate of L becomes stable, for example, once the confidence interval around L becomes sufficiently narrow. Suppose that we wish to run the simulation until the $100(1-\alpha)$ percent confidence interval for the true mean is of width at most ℓ. We can guarantee this by means of the following algorithm.

1. Generate at least 100 data values, say.
2. Repeatedly generate additional data values, stopping when the number of values generated, n, is such that $2z_{\alpha/2}S_n/\sqrt{n} < \ell$.

The initial 100 data values are for illustration; a suitable value depends on the simulation experiment under consideration. The minimum ensures that S_n is a reasonable estimate of σ before the rule is applied, so that the simulation does not stop early because S_n happens to be small. The initial transient phase is a separate issue: it is dealt with by discarding the data collected during a warm-up period before any data values are recorded.

S.8 Variance Reduction

Variance reduction

So far, we have used the sample mean, $\bar{X}_n$, as our estimator for μ, the mean value of our distribution. We know that

$$\mathbb{E}(\bar{X}_n) = \mu \quad \text{and} \quad \text{Var}(\bar{X}_n) = \frac{\sigma^2}{n},$$

where n is the sample size. Might we be able to find an alternative estimator for μ that has a smaller variance? Such *variance reduction techniques* can

sometimes produce significant speed-ups in the simulation, because fewer samples are needed to reach a given accuracy.

S.8.1 Antithetic Variables

Suppose that X_1 and X_2 are two identically distributed random variables with common mean $\mathbb{E}(X_1) = \mathbb{E}(X_2) = \mu$. Then

$$\mathrm{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}\left(\mathrm{Var}(X_1) + \mathrm{Var}(X_2) + 2\,\mathrm{Cov}(X_1, X_2)\right). \qquad \textbf{(EQ S.21)}$$

Hence, we would get a reduced variance by using $(X_1 + X_2)/2$ when X_1 and X_2 are negatively correlated.

Recall that in the inverse transform method, we first generate pseudo-random numbers $U_1, U_2, \ldots$ with a $U(0,1)$ distribution. But then the numbers $1 - U_1, 1 - U_2, \ldots$ are also random numbers with a $U(0,1)$ distribution, and these two series of numbers are negatively correlated. In practice, it often happens that dividing the simulation runs into two groups, and using $1 - U$ for the second group in place of U in the first group, yields two random variables X_1 and X_2 that are negatively correlated. In this case, we say that X_1 and X_2 are *antithetic variables*.

EXAMPLE S.3: ANTITHETIC VARIABLES IN A QUEUEING SIMULATION

Consider a queueing system. Let D_i be the delay in the queue of the ith customer, and suppose that we wish to estimate $\mathbb{E}(D_1 + D_2 + \cdots + D_n)$, the expected sum of the delays of the first n customers. We should expect to need a collection of $2n$ random numbers U_j, one for the interarrival time and one for the service time of each customer. Repeating the simulation using the $2n$ random numbers $1 - U_j$, and averaging the two results, typically gives an improved estimator compared to using a fresh set of $2n$ random numbers.

S.9 Additional Video Material

Two further topics are covered only in video.

- Some issues to watch out for when running simulations 
- Simulating a controlled (closed) system 

S.10 Exercises

1. State and counter variables

Consider a simulator that models the arrival and departure of people from

the King's Cross plaza. We would like to study (a) the mean number of people who are in the plaza as well as the dynamic evolution of this quantity, (b) the mean delay to traverse the plaza and the dynamic evolution of this quantity, and (c) the impact of changing the train schedule on the number of people in the plaza and the plaza traversal delay. What would be the system state variables and counter variables necessary to accomplish this task?

2. Generating a discrete random variable

Suppose that the discrete probability distribution over the 5 values x_i, $i \in [1, 5]$, taken by the random variable X, is given by the vector $[0.1, 0.3, 0.2, 0.1, 0.3]$. Discuss how you could generate values taken on by this variable.

3. Simulating a Poisson process

Suppose that the first four calls to a $U[0, 1]$ random number generator return the values, to three digits of precision, of 0.127, 0.632, 0.829, and 0.243. Use these to compute the times of the first four arrivals to a Poisson process with parameter $\lambda = 5.0$.

4. Stopping rule

Suppose you are using simulations to compute the mean number of people on King's Cross plaza and you would like to have 95% confidence that the true mean lies within ± 10 people of the mean value you estimate from your simulations. What stopping rule should you use to decide that you have run enough simulations?

5. Antithetic variables

Explain one approach you might use to exploit antithetic variables to reduce the variance in your simulation outputs.

6. Inaccurate simulation results

To study the impact of datacenter network capacity on the query latency of a search engine, a researcher builds a simulator using custom code and their own random number generator. However, they notice that the results of this single long simulation run are far from accurate. What might be some reasons for this?

Programming Assignment: Discrete-Event Simulation of a Queue

Optional.

Your task is to implement a discrete event simulator, preferably in Python, to simulate packet arrivals and departures from a queue. Each packet should have a distinct integer sequence number starting at sequence number 0. Choose an appropriate data structure to hold simulation events.

The length of a simulation run is a free parameter that you can choose. Given this length, for each simulation run r, compute the mean waiting time in the queue w_r. You should run as many simulations as necessary so that the half-width of the 95% confidence interval of w (computed over multiple simulation runs) is less than 5% of its mean w_{ave}.

Simulate both an M/M/1 and an M/D/1 queue. Use appropriate statistical libraries as needed.

Once your simulator works, answer the following questions.

1. List the event types in your simulation. Why do you need each of these events?
2. What were the fields you used to describe each event? Why?
3. What data structure did you use to hold simulation events? Why did you choose this data structure?
4. For the M/M/1 simulation, show a trace of 10 packet arrivals and their departures in the format:

   ```
   <timestamp>:  the current queue length, event type, event details //
                                       comment
   ```

   for the case where utilisation = 0.8. Use the comments to explain what is happening in each step of the simulation.
5. You will find that you can achieve the stopping criterion either by choosing a longer simulation length and performing fewer runs or choosing a shorter simulation length and needing to perform more simulation runs. Which approach takes less computation time? Why? Can you suggest a way to find the optimal duration of a simulation run?
6. When the simulation run duration is too short, even when the stopping criterion is met, a problem arises. What is it? Why does it happen? What cautionary lesson can one derive from this?

7. Once you have chosen the simulation run length, compare the simulation results with the corresponding theoretical values by having your code draw a graph for each queue type (M/M/1 and M/D/1). In each graph, the X axis should be the utilisation (ρ) and the Y axis plots both

   a. the mean waiting time in the queue computed by the simulation;

   b. the mean waiting time in the queue that is computed using the appropriate formula.

   For the simulation results, also show the confidence interval.

Note: Tools such as ChatGPT and Co-Pilot will let you automatically generate big chunks of the code. However, they will not improve your understanding of discrete event simulation or queues. For this reason, please don't use these tools.

CHAPTER 6

Stochastic Processes and Queueing Theory[1]

6.1 Overview


1:38

Queues arise naturally when entities demanding service, or **customers,** interact asynchronously with entities providing service, or **servers**. **Service requests** may arrive at a server when it is either unavailable or busy serving other requests. In such cases, service demands must be either queued or dropped. A server that queues demands instead of dropping them can smooth over fluctuations in the rate of service and in the rate of request arrivals, leading to the formation of a queueing system. The study of the probabilistic behaviour of such systems is the subject of queueing theory.

Here are some examples of queueing systems:

1. The arrival and service of packets at the output queue of a switch. Packets may arrive when the output link is busy, in which case the packets, which are implicit service requests, must be buffered (i.e., queued).
2. The arrival of and service of HTTP requests at a Web server. If the Web server is busy serving a request, incoming requests are queued.
3. The arrival of telephone calls to a switch-control processor in the telephone network. The processor may be unable to service the call because the switch is busy. In this case, the call is queued, awaiting the release of network resources.

1. These notes are based on, but differ from the published version of my book. All known errors have been corrected and some additional explanations have been added.

Given a queueing system, we would like to compute certain quantities of interest, such as:

- the **queueing delay:** the expected waiting time for a service request
- the **backlog:** the mean number of service requests that are awaiting service
- the **utilization factor**: the expected fraction of time that the server is busy
- the **drop rate**: the expected fraction of service requests that must be dropped because there is no more space left in the queue
- The mean length of a busy or idle period

Queueing theory allows us to compute these quantities--both for a single queue and for interconnected networks of queues--as long as the incoming traffic and the servers obey certain simplifying conditions. Unfortunately, measurements show that traffic in real networks does *not* obey these conditions. Moreover, we cannot mathematically analyse most networks that are subjected to realistic traffic workloads. Nevertheless, it is worth studying queueing theory for two important reasons. First, it gives us fundamental insights into the behaviour of queueing systems. These insights apply even to systems that are mathematically intractable. Second, the solutions from queueing theory--even from unrealistic traffic models--are a reasonable first approximation to reality. Therefore, as long as we keep in mind that results from queueing theory are only approximate and are primarily meant to give an insight into a real system, we can derive considerable benefit from the mathematical theory of queues.

6.1.1 A general queueing system

We now introduce some standard notation. A **queue** is formed when **customers** present **service requests** or **jobs** to one or more **servers**. Customers arrive at a rate of λ customers/second at times t and the time between arrivals is described by the **interarrival time distribution** $A(t) = P(\text{time between arrivals} \leq t)$. We denote the service time by x which has a **service time distribution** $B(x) = P(\text{service time} \leq x)$ with a mean service rate of μ customers/second. Customers are assumed to wait in the queue with a **mean waiting time** T. Note that in this chapter we will primarily study a single queue in isolation.

6.1.2 Little's theorem

Little's theorem is a fundamental result that holds true for *all* arrival and service processes. The theorem states that the mean number of customers in a queueing system is the product of their mean waiting time and their mean arrival rate. Note that the mean waiting time may be interpreted as including or excluding the service time. If the



4:47

number of customers includes or excludes the customer currently receiving service, respectively, Little's theorem holds for both situations.

Proof of Little's theorem: Suppose that customers arrive to an empty queue[2] at a mean rate of λ customers/second. An average of λt customer arrive in t seconds. Let T denote the mean waiting time in seconds of a newly arriving customer. The total time spent waiting in the queue across all customers during the time interval is therefore λTt customer-seconds.

Erratum 6.1: Read "customers arrive"; below, "or on the way".

Let *N* denote the mean number of customers in the queue. In 1 second, these *N* customers accumulate *N* customer-seconds of total waiting time. Thus, in *t* seconds, they accumulate a total of *Nt* customer-seconds of waiting time. This must equal λ*Tt*, which implies that $N=\lambda T$.

■

Note that this argument is independent of the length of the time interval *t*. Moreover, it does not depend on the order of service of customers, the number of servers, on the way in which customers arrive. Thus, it is a powerful and general law applicable to all queueing systems.

EXAMPLE 6.1: LITTLE'S THEOREM

Suppose that you receive email at the average rate of one message every five minutes. If you read all your incoming mail instantaneously once an hour, what is the average time that a message remains unread?

Solution

The mean message arrival rate is 12 messages/hour. Because you read email once an hour, the mean number of unread messages is 6 (this is the expected number of messages received in half an hour). By Little's theorem, this is the product of the mean arrival rate and the mean waiting time, which immediately tells us that the mean time for a message to be unread is $\frac{6}{12}$ hours =30 minutes.

2. The same reasoning applies when customers arrive to a non-empty queue, but arrivals to an empty queue simplifies the analysis.

■

EXAMPLE 6.2: LITTLE'S THEOREM

Suppose that 10,800 HTTP requests arrive to a web server over the course of the busiest hour of the day. If the mean waiting time for service should be under 6 seconds, what should be the largest allowed queue length?

Solution

The arrival rate $\lambda = 10{,}800/3600 = 3$ requests/second. We want $T \leq 6$. Now, $N = \lambda T$, so $T = N/\lambda$. This means that $N/\lambda \leq 6$ or that $N \leq 6*3 = 18$. So, if the mean queueing delay is to be no larger than 6 seconds, the mean queue length should not exceed 18. In practice, the web server could return a server busy response when the queue exceeds 18 requests. This is conservative because then 18 is the *maximum* queue length rather than its mean.

■

6.2 Stochastic processes


7:34

The foundation of queueing theory is the mathematical study of a **stochastic process**. Such a process is used to model the arrival and service processes in a queue. We will both intuitively and mathematically define a stochastic process and then study some standard stochastic processes.

EXAMPLE 6.3: DETERMINISTIC AND STOCHASTIC PROCESSES

Lab: Random walk

Consider a staircase with 10 steps numbered 1 through 10 and a person standing on the first step, which is numbered 1. Suppose that a clock next to the staircase ticks once a second starting at time 1. Finally, assume that it takes zero time to climb each step.

If the person were to climb one step at each clock tick, then we can predict exactly where the person would be at each time step. At time 0, the person is on step 1, and would stay there until just before time 1. When the clock

ticks and time increments to 1, the person would be at step 2 and would stay there until just before time 2. At time 2 the person would be on step 3, and so on. We therefore call the act of climbing the staircase in this fashion a **deterministic** process.

In contrast, suppose that the person climbs one step or goes down one step or stays on the same step with some (potentially zero) probability. With this change, we lose predictability. That is, we no longer know exactly where the person will be at any moment in time: we can only attach probabilities to the set of places where the person *could* be at that time. The process is no longer deterministic but instead it is **stochastic**.


9:41

We capture this by means of a random variable X_n that denotes the step the person is on immediately after the nth clock tick. The random variable is associated with the probability distribution $\pi(n)$ over the positions where the person could be immediately after that time. For instance, at time 0, the person is at step 1 with probability 1, so the distribution of X_0 over the domain {1, 2,...,10} is given by the discrete probability distribution $\pi(0) = \{1.0, 0,..., 0\}$. Suppose that the probability that the person goes up is p, that the person goes down is q, and that the person stays on the same step is $1\text{-}p\text{-}q$, except at step 1, where the probability of going up is p, and the probability of staying on the same step is $1\text{-}p$. Then, immediately after time 1 (after the first clock tick) $\pi(1) = \{1\text{-}p, p, 0,...,0\}$. Similarly, $\pi(2) = \{(1\text{-}p)^2+pq, p(1\text{-}p\text{-}q) + (1\text{-}p)p, p^2, 0,...,0\}$, and so on.

To compute $\pi(n)$ given $\pi(n-1)$ we determine the different ways that we can reach each particular step summing the probability over all possible ways to reach that step.

Note that we distinguish between the distribution of the random variables at each time step and the actual trajectory taken by a person. For a given trajectory, at each time instant, the person is, of course, only on one step of the staircase. The trajectories are created by sampling from the distributions $\pi(n)$. A trajectory is also therefore called a **sample path**.

■

This example suggests the following definition of a **stochastic process**: it is a family of random variables X_n that are indexed by the time index n. The value of the random variable in a particular trajectory is also called the **state**

of the stochastic process at that point in time. Without loss of generality, we can think of the states as being chosen from the integers from 1 to *N*. Thus, we can imagine the process as moving from the state corresponding to the value taken by random variable X_n in a given trajectory to the state corresponding to the value taken by random variable X_{n+1} at time n+1, just like the person moves from one stair to another. As we have shown, given the probabilities of moving from one step to another, we can, in principle, compute the distribution of each X_n: the distribution of the stochastic process over the state space at that time.

7:10

Time is discrete in this example. In other situations, a system is better modeled when time is continuous. In this case, the family of random variables corresponding to the stochastic process consists of the variables $X(t_1)$, $X(t_2)$,... where the t_i represent the times at which the state transitions occur. Given the probability of moving from step to step, we can compute $\pi(t_{i+1})$, the distribution of $X(t_{i+1})$, from $\pi(t_i)$ the distribution of $X(t_i)$.

In the example, the person's movements were limited to moving up or down one step on each clock tick. We could, instead, allow the person to go from a given step to any other step (not just the steps above and below) with some probability. Indeed, this distribution of probabilities could differ at different steps and even differ at each clock tick! And, finally, the person could be on a ramp, so that the amount of movement could be a real positive or negative quantity, rather than an integer. These variations are all within the scope of definition of a stochastic process, but the analysis of the corresponding processes is progressively more difficult. We will first describe some standard types of stochastic processes and then focus on the simplest ones.

6.2.1 Discrete and continuous stochastic processes

A stochastic process can be classified as a discrete or a continuous process in two ways: (1) the values assumed by the random variables (or the **state space**) can be discrete or continuous, and (2) the index variable, i.e., time, also can be discrete or continuous.

4:33

A **discrete-space process** is one in which the random variables X_k take on discrete values. Without loss of generality, we can think of the state in a discrete-space process as being indexed by an integer in the range 1,2,...*N*.

EXAMPLE 6.4: DISCRETE-SPACE PROCESS

Continuing with Example 6.3, we see that the set of possible states is the set of stairs, which forms a discrete set.

■

A **continuous-space process** is one where the random variables take on values from a finite or infinite continuous interval (or a set of such intervals).

EXAMPLE 6.5: CONTINUOUS-SPACE PROCESS

Continuing with Example 6.3, consider a person walking up and down a ramp rather than a stair. This would allow movements by real amounts. Therefore, the random variable corresponding to the state of the process can take on real values, and the corresponding stochastic process would be a continuous-space process.

■

In a **discrete-time** process, the indices of the random variables are integers. We can think of the stochastic process in a particular trajectory as moving from one state to another at these points in time.

EXAMPLE 6.6: DISCRETE-TIME PROCESS

Continuing with Example 6.3, this corresponds to a person moving from one step to another exactly at each clock tick. Such a stochastic process is also called a **stochastic sequence**.

■

In a **continuous-time** process, the times when the process can move to a new state are chosen from a real interval.

EXAMPLE 6.7: CONTINUOUS-TIME PROCESS

Continuing with Example 6.3, this corresponds to a person moving from stair to stair at will, independent of the clock.

■

Stochastic processes corresponding to all four combinations of {discrete space, continuous space} and {discrete time, continuous time} are well-known. Note that, for simplicity, for the remainder of this chapter we will only study discrete-space stochastic processes.

6.2.2 Markov processes

Lab: Markov chains

An important aspect of a stochastic process is how the probability of transitioning from one state to another is influenced by past history. Continuing with our staircase example (a discrete-time and discrete-space stochastic process), consider a person who is allowed to go from any stair to any other stair. Moreover, we will ask they person to obey the following rules: If he or she arrives at stair 5 from stair 6, move to stair 3. If, however, he or she arrives at stair 5 from stair 3, move to stair 9. In all other cases, move to stair 1. Suppose that at some point, in time we see that the person is on stair 5. What happens next?


5:37

Erratum 6.2: Read "the person".

Erratum 6.3: Read "at some point in time, we see".

The answer is: We don't know. It depends on where the person was in the previous time step. Stated more precisely, the distribution of the random variable X_{n+1} (i.e., $\pi(n+1)$) when $X_n = 5$ depends on the value of X_{n-1}. Generalizing from this example, we can define more complex stochastic processes for whom $\pi(n+1)$ depends not only on X_n, but also on X_{n-1} , X_{n-1}, ..., X_1. Such systems are inherently complex and there is little we can say about them.

Erratum 6.4: Read $X_{n-1}, X_{n-2}, \ldots, X_1$.

In an attempt to curb this complexity, consider the following rule:

$$\pi(n+1) \text{ depends only on the value of } X_n$$

This rule simplifies the situation: if the person is on step 5 at time n, then we know $\pi(n+1)$ *independent of the prior history.* As we will see, this allows us to easily compute many quantities of interest about the process. Moreover, many naturally occurring stochastic processes obey this rule. Due to these two facts, stochastic processes that obey this rule are given a special name: they are called **Markov processes**, in honour of A.N. Markov, who first studied them in 1907.

Erratum 6.5: Markov's first paper on chains was in 1906.

Formally, we state the **Markov property** as (for the case of discrete-time and discrete-space stochastic processes):

$$P(X_{n+1} = j \,|\, X_n = i_n, X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, X_{n-3} = i_{n-3}, \ldots, X_1 = i_1) = P(X_{n+1} = j \,|\, X_n = i_n) \quad \textbf{(EQ 6.1)}$$

The conditional probability $P(X_{n+1} = j \,|\, X_n = i_n)$ is called the **transition probability** to go from state i_n to state j.

EXAMPLE 6.8: MARKOV PROCESS

Consider a discrete-time discrete-space Markov process whose state space is {1,2,3}. Let $P(X_3 = 1 \mid X_2 = 1) = 0.2$; $P(X_3 = 2 \mid X_2 = 1) = 0.4$; $P(X_3 = 3 \mid X_2 = 1) = 0.4$. Suppose that we know that $P(X_2) = 1$. Then, the Markov property allows us to compute $\pi(3)$ no matter which path was taken to reach the state 1 at time 2.

Erratum 6.6: Read $P(X_2 = 1)$.

■

Note that for a discrete-time stochastic process, at time step *n*, we already know the the sequence of prior states. At time *n*, we are usually interested in computing $\pi(n+1)$ given this past history. The Markov property allows us to forget everything about history except the value of the current random variable, which encapsulates all past history. This is similar in spirit to the memorylessness property of the exponential distribution (see Section 1.6.3).

Erratum 6.7: Read "the sequence".

A similar property holds true for continuous time stochastic processes. For simplicity, we will first study discrete-space and discrete-time processes that obey the Markov property, also known as **discrete time Markov chains**, before considering continuous time Markov processes.

Homegeneity 2:32

6.2.3 Homogeneity, state transition diagrams, and the Chapman-Kolmogorov equations

A stochastic process may satisfy the Markov property even if its transition probabilities vary over time. A process whose transition probabilities are time-dependent is called a **non-homogeneous Markov process**. Of course, this greatly complicates the analysis. We can simplify the analysis by decreeing that the transition probabilities should be time-independent. In our example, this means that when the person is on a particular step, say step 4, the probability of going to any other step is always the same, no matter *when* they got to step 4. Such a process is called a **homogeneous Markov process.** For a homogeneous Markov process, we define the *time-independent* transition probability between state *i* and state *j* as $p_{ij} = P(X_n = j | X_{n-1} = i)$ for any *n*.

EXAMPLE 6.9: HOMOGENEOUS MARKOV PROCESS

Consider the discrete-time discrete-space stochastic process in Example 8. If this process were homogeneous, we need not consider exactly one point in time, such as time 2. Instead, if $P(X_{n+1}= 1 \mid X_n = 1) = 0.2$; $P(X_{n+1}=2 \mid X_n = 1) = 0.4$; $P(X_{n+1} = 3 \mid X_n = 1) = 0.4$, we can compute the distribution X_{n+1} given that of X_n for all values of n.

Erratum 6.8: Read "Example 6.8"; below, "Example 6.10".

Erratum 6.9: Read "the distribution of X_{n+1}".

■

The state transition probabilities for a homogeneous Markov chain with N states have two equivalent representations. The first is in the form of an $N \times N$ transition matrix $\mathbf{A}$ whose elements are the probabilities p_{ij}. This representation has the attractive property that the probability of going from any state i to state j in two steps is given by the elements of $\mathbf{A^2}$. The second is as a graph (see Example 10). In this representation, vertices represent states and the annotation on an edge from vertex i to vertex j is p_{ij}. This visually represents a Markov chain. Note that a non-homogeneous Markov chain requires such a state transition diagram for each time step.

EXAMPLE 6.10: REPRESENTING A HOMOGENEOUS MARKOV PROCESS

Continuing with Example 6.9: we have already seen that $p_{11} = 0.2$, $p_{12} = 0.4$, $p_{13} = 0.4$. Suppose that $p_{21} = 1.0$, $p_{22} = 0$, $p_{23} = 0$ and $p_{31} = 0.5$, $p_{32} = 0.25$, $p_{33} = 0.25$. Then, we can represent it in two ways as follows and as shown in Figure 6.1:

$$A = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 1.0 & 0 & 0 \\ 0.5 & 0.25 & 0.25 \end{bmatrix}$$

Representations 9:25

FIGURE 6.15.

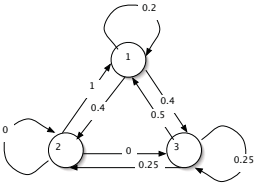


FIGURE 6.1. **State transition diagram for Example 6.10**

Note that ***A*** is a *right stochastic matrix* (that is, each row sums to 1.0).

■

Given the set of transition probabilities p_{ij} for a homogeneous Markov chain, we can define the *m*-step transition probability from state *i* to state *j* denoted $p_{ij}^{(m)}$ by

$$p_{ij}^{(m)} = P(X_{n+m}=j \mid X_n = i) = \sum_k p_{ik}^{(m-1)} p_{kj} \qquad \text{m} = 2,3,... \qquad \textbf{(EQ 6.2)}$$

where the sum of products form comes from summing across independent events (that of going to some intermediate state in *m*-1 steps), and each term is a product, because it is the combination of two independent events (going from state *i* to state *k* and from state *k* to state *m*). These relations exist only because of the Markovian nature of the chain. They are important enough that they are given their own name: the **Chapman-Kolmogorov** equations, which can also be stated as:

Erratum 6.10: Read "mutually exclusive events".

Erratum 6.11: Read "to state *j*".

CK equations 4:18

$$p_{ij}^{(m)} = P(X_{n+m}=j \mid X_n = i) = \sum_k p_{ik} p_{kj}^{(m-1)} \qquad \text{m} = 2,3,... . \qquad \textbf{(EQ 6.3)}$$

Comparing the two, we see that in the first formulation traces the trajectory of the process as it goes from state *i* to state *k* in *m-1* steps, and from state *k* to state *j* in one step, and the second formulation traces the trajectory of the process as it goes from state *i* to state *k* in one step and from state *k* to state *j* in *m-1* steps. Clearly, these are equivalent recursions.

6.2.4 Irreducibility

2:42

If every state of a stochastic process can be reached from every other state in a finite number of steps, then the process is called **irreducible**, otherwise it is **reducible**. Moreover, if states of a stochastic process can be separated into subsets that are mutually unreachable, we call each such set a **separable sub-chain**.

EXAMPLE 6.11: A REDUCIBLE MARKOV CHAIN

Consider the Markov chain in Figure 6.2. Here, the transition probabilities p_{ij} for i even and j odd or i odd and j even are 0. Therefore, if the initial state of the process is an even-numbered state, the trajectory of process is confined to even-numbered states. Alternatively, a process that starts from an odd-numbered state, it will forever stay in odd-numbered states. The even-numbered states are unreachable from the odd-numbered steps states and the chain, therefore, is reducible. Indeed, we could separate the even-numbered and odd-numbered states into separate chains that would equivalently describe the process. We can generalize this idea to construct stochastic processes that can be decomposed into as many sub-chains as we wish.

Erratum 6.12: Delete "it"; below, read "odd-numbered states".

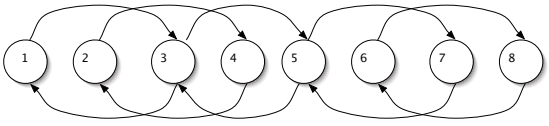


FIGURE 6.2. A reducible Markov chain

■

5:48

6.2.5 Recurrence

For every state j of a stochastic process, one of two conditions must hold: After entering state j, either the probability of re-entering state j after a finite number of steps is 1, or there is some non-zero probability that the state is not reentered after a finite number of steps. In Example 6.3, this is equivalent to saying that after stepping on a stair, say, stair 6, it is either certain that the person will return to stair 6, or there is a nonzero probability that the person will not return to stair 6. If return to a state is certain, we call the state **recurrent**, otherwise we call it **transient**. An example of an irreducible chain with all transient states is an infinite chain whose every state has a probability of p to move to the state to its right and a probability of 1-p to move to state to its left, where $p \neq 0.5$. This is because, when $p \neq 0.5$, the state tends to drift away to the right or the left, and there is a non-zero probability that, after a state is visited, it may never be revisited.

Here is another way to define recurrence formally. Let f_j^n denote the probability that the *first* return to state j is after n steps. State j is recurrent if $\sum_{n=1}^{\infty} f_j^n = 1$ and transient otherwise.

Although a state is recurrent, its expected recurrence period, defined as $\sum_{n=1}^{\infty} n f_j^n$, may be infinite. If so, the state is called **recurrent null**. Otherwise, it is **recurrent non-null**. Recurrent null states are pathological and rarely of practical interest.

6.2.6 Periodicity

1:11

Given a recurrent state *j*, suppose the only way to return to that state is to take *r*, 2*r*, 3*r*... steps, with $r \geq 2$. We then call the state *j* **periodic**, with a period *r*. Periodic states arise when the Markov chain has a cycle. A trivial way to check if a state is periodic is to see if it has a self-loop, that is $p_{jj} > 0$. If so, the state can be re-entered with any desired number of steps, which makes *r* =1, and the state **aperiodic**. For an irreducible Markov chain, if *any* state has a self-loop, all states are aperiodic.

EXAMPLE 6.12: PERIODIC AND APERIODIC MARKOV CHAINS

The Markov chain in Figure 6.2 is periodic with period 2 and the chain in Figure 6.1 is aperiodic.

■

6.2.7 Ergodicity

8:51

The sequence of states visited by a stochastic process is called its **trajectory**. For example, a valid trajectory for the chain in Figure 6.1 is $1 \to 2 \to 1 \to 3 \to \ldots$. Given a trajectory, we can compute a statistic[3] on it, such as the fraction of the trajectory spent in a particular state. This statistic is the limiting ratio of the number of occurrences of that state to the length of the traejectory. Because trajectories can be made as long as we desire, if the limiting statistic exists, it can be approximated as closely as we wish. We call such a limit a **time average**.

Erratum 6.13: Read “trajectory”.

Now, consider a set of instances of the same stochastic process. At each time step, each instance changes state according to the same transition probabilities. Nevertheless, the trajectory associated with any pair of processes in the ensemble may differ owing to their stochastic nature. Suppose that we want to compute a statistic on the

3. A statistic is any parameter computed over a finite sample.

ensemble of trajectories at any time step. For instance, we may wish to compute the fraction of trajectories that are in state 1 at time step 10. If the limiting statistic exists, we can approximate this statistic as closely as we wish by making the ensemble sufficiently large. We call such a limit a **ensemble average**.

Erratum 6.14: Read "an ensemble" (twice).

An interesting question is whether a statistic computed as a time average is the same as a statistic computed as an ensemble average. As the next example shows, this need not necessarily be the case!

EXAMPLE 6.13: TIME AND SPACE AVERAGES

Consider the stochastic process shown in Figure 6.2. Suppose that the initial state is 1 and that the statistic we wish to compute is the fraction of time spent in an odd-numbered state. Clearly, no matter how long the length of the trajectory, this time average will be 1.0. Now, consider an ensemble of instances of this process where the initial state is chosen equally probably as 1 or 2. For this ensemble, the limiting value of the statistic at any time step will be 0.5 and is the ensemble average. For this stochastic process, the time average differs from the ensemble average.

■

Intuitively, a stochastic process is **ergodic** if every statistic computed as a time average over a sufficiently long single trajectory can also be computed as a ensemble average over a sufficiently large number of trajectories. For this to be true, the sequence of states visited by the stochastic process over time should look statistically identical to the set of states occupied by an ensemble of processes at a single time step. We now consider the conditions under which a stochastic process is ergodic.[4]

To begin with, we define a *state j* to be ergodic if it is recurrent non-null and aperiodic. Continuing with Example 6.3, it is a stair that the person will surely return to (recurrent), with a mean recurrence period that is finite (non-null), and such that the returning times do not have a least common divisor larger than 1 (aperiodic). If all the states in a Markov chain are ergodic, other than for a finite set of transient states such that the transient states always lead to an irreducible recurrent chain, the chain itself is ergodic[5]. It can be shown that a finite, aperiodic,

Erratum 6.15: Read "greatest common divisor".

4. Many mutually incompatible definitions of ergodicity exist in the literature. The definition presented here was chosen because it has a simple intuitive basis.

irreducible Markov chain is always ergodic (in other words, all states of a finite, aperiodic, irreducible Markov chain are recurrent non-null).

EXAMPLE 6.14: ERGODIC MARKOV CHAIN

The Markov chain in Figure 6.1 is finite, aperiodic, and irreducible. Therefore, it is also ergodic.

■

Ergodicity is important in practice becayuse, in the long run, an ergodic chain is insensitive to its initial state $\pi(0)$. That is, independent of its initial state, $\pi(n)$, the distribution of X_n (for reasonably large values of n) is the same.

Erratum 6.16: Read "because".

6.2.8 A fundamental theorem

We now have enough terminology to state (without proof) a fundamental theorem:

Theorem 6.1: The states of an irreducible Markov chain are either all transient, all recurrent null, or all recurrent non-null. If any state is periodic, then all states are periodic with the same period *r*.

■

Intuitively, this categorizes all Markov chains into a few types. The first are those where the process goes from state to state but is not guaranteed to ever return to a state. In this case, all states are transient. In both the second and third type of chain the process has a non-zero probability of returning to at least one of the states. But the chain is irreducible, and so we can go from that state to all other states. Therefore, if the process can return to any one state, by definition it can return to all other states, which makes all states recurrent. The second and third types differ in the expected recurrence period. In the second type, the transition probabilities are such that the expected recurrence period is infinite, so that all states are recurrent null. In the third type, the expected recurrence period for all states is finite. For this type, we have two further subtypes: periodic recurrent non-null chains,

Erratum 6.17: Recurrence needs return with probability 1.

5. We allow a finite number of transient states in the chain because over sufficiently long trajectories or for a sufficiently large ensemble their contribution to any statistic is negligible.

whose states all share the same period, and aperiodic, recurrent non-null (ergodic) chains, for which no such period can be defined.

6.2.9 Stationary (equilibrium) probability of a Markov chain

Recall that for a homogeneous Markov chain the state transition probabilities are time-independent. For a homogeneous chain, we expect that the probability of *being* in any particular state to also be time-independent. (If the probability of going from one state to another does not depend on time, the probability of being in any state shouldn't either).

6:59

Of course, the probability of being in a particular state may depend on the initial state, especially for nonergodic chains, which are sensitive to their initial conditions. We define the **stationary probability distribution** of a Markov chain as follows: Suppose that we start with the initial distribution $\pi(0) = \pi^*$. Then, π^* is also the stationary distribution of the chain, if for all *n*, $\pi(n) = \pi^*$. Intuitively, if we start with the probability of being in each state *j* as defined by the stationary distribution, then the transitions from each state according to the transition probabilities do not change the probability of being in each state.

EXAMPLE 6.15: STATIONARY DISTRIBUTION

Compute the stationary distribution of the Markov chain in Figure 6.3.

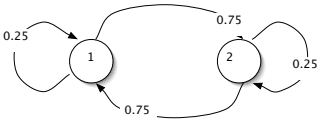


FIGURE 6.3. A simple Markov chain

Solution

Suppose that the initial probability of being in state 1 is 0.5 and of being in state 2 is 0.5, that is $\pi(0) = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$.

After one time step, the probability of being in state 1 is 0.25*0.5 + 0.75*0.5, where the first term is the probabil-

ity of remaining in state 1, and the second term is the probability of going from state 2 to state 1, and we sum these probabilities because these are independent events. As expected, this sums to 0.5, so that the probability of being in state 1 in time step 2 is also 0.5. Symmetrically, if the probability of being in state 2 at time 1 is 0.5, the probability of being in state 2 at time 2 is also 0.5. Therefore, the stationary probability distribution of this chain is $\pi^* = [0.5\ 0.5]$.

Erratum 6.18: Read "mutually exclusive events".

■

7:37

6.2.10 A second fundamental theorem

We now state a second fundamental theorem that allows us to compute stationary probabilities for any Markov chain.

Theorem 6.2: In an irreducible and aperiodic homogeneous Markov chain, the limiting probability distribution

$$\pi^* = \lim_{n \to \infty} \pi(n) \quad \text{(EQ 6.4)}$$

always exists and is independent of the initial state probability distribution $\pi(0)$. Moreover, if all the states are ergodic (being recurrent non-null, in addition to being aperiodic), then π_j^*, the stationary probability of being in state *j*, is non-zero and can be uniquely determined by solving the following set of equations:

$$\sum_j \pi_j^* = 1$$
$$\pi_j^* = \sum_i \pi_i^* p_{ij} \quad \text{(EQ 6.5)}$$

■

This theorem provides us with a simple set of equations to determine the stationary probability that the Markov chain is in any particular state. We only need verify that the set of states is memoryless (i.e., satisfies the Markov property), finite, irreducible (all states can be reached from each other), and aperiodic (for example, because of at least one self-loop). These properties can be verified through simple inspection. Then, we can solve the system of equations above to obtain the stationary probability of being in each state.

EXAMPLE 6.16: STATIONARY PROBABILITY OF A MARKOV CHAIN

Compute the stationary probability for the Markov chain in Figure 6.1.

Solution

3:28

Note that this chain is ergodic, so we can use Theorem 2 to obtain the following equations:

Erratum 6.19: Read "Theorem 6.2".

$$\pi_1^* = 0.2\pi_1^* + 1\pi_2^* + 0.5\pi_3^*$$
$$\pi_2^* = 0.4\pi_1^* + 0\pi_2^* + 0.25\pi_3^*$$
$$\pi_3^* = 0.4\pi_1^* + 0\pi_2^* + 0.25\pi_3^*$$
$$1 = \pi_1^* + \pi_2^* + \pi_3^*$$

We solve this system of equations, using, for example, Gaussian elimination (see Section 3.4.2) to obtain: $\pi_1^* = 15/31$; $\pi_2^* = 8/31$; $\pi_3^* = 8/31$, which is the stationary probability distribution of the chain (Verify this!).

■

6.2.11 Mean residence time in a state

Besides knowing the stationary probability of being in a particular state of a Markov chain, we would also like to know how long the process spends in each state. This duration can be computed by first obtaining the probability P(system stays in state *j* for *m* additional steps given that it just entered state *j*). The probability that the system stays in the same state after one time step is clearly p_{jj}. Moreover, after one time step, being Markovian, the process has no memory that it was in that state earlier. Therefore, the probability of staying in the state for *m* steps is given by $p_{jj}^{m-1}(1-p_{jj})$, which is a geometrically distributed random variable with parameter $(1-p_{jj})$. This allows us to compute the mean of the distribution, that is, the expected residence time in state *j*, as $1/(1-p_{jj})$.

3:04

EXAMPLE 6.17: RESIDENCE TIME

Compute the residence times in each state of the Markov chain shown in Figure 6.1.

Solution

p_{11} = 0.2, so E(residence time in state 1) = 1/(1-0.2) = 1/0.8 = 1.25.

p_{22} = 0, so E(residence time in state 1) = 1/(1-0) = 1.

p_{33} = 0.25, so E(residence time in state 1) = 1/(1-0.25) = 1/0.75 = 1.33.

■

Erratum 6.20: The second and third results are for states 2 and 3.

6.3 Continuous-time Markov Chains

5:49

Lab: CTMC rates

Our discussion so far has focused on discrete-time Markov chains, where state transitions happen every clock tick. We now turn our attention to continuous-time chains, where state transitions can happen independent of clock ticks. Most of the intuitions developed for discrete-time chains carry through to continuous-time chains, with a few modifications. The main point of difference is that we need to consider the time instants t_1, t_2,... when state transitions occur, rather than assuming that a state transition occurs at every clock tick. We briefly state the main results for a continuous-time stochastic process, and then focus on a specific type of continuous-time process: the birth-death process.

6.3.1 Markov property for continuous-time stochastic processes

We first state the Markov property for continuous-time stochastic processes. The stochastic process $X(t)$ forms a continuous-time Markov chain if for all integers n and for any sequence of times t_1, t_2,...,t_{n+1} such that $t_1 < t_2 <...< t_{n+1}$

$$P(X(t_{n+1}) = j \mid X(t_1)= i_1, X(t_2)= i_2,...,X(t_n)= i_n) = P(X(t_{n+1}=j) \mid X(t_n)= i_n) \quad \textbf{(EQ 6.6)}$$

Intuitively, this means that the future ($X(t_{n+1})$) depends on the past only through the current state i_n.

The definitions of homogeneity, irreducibility, recurrence, periodicity, and ergodicity introduced for discrete-time Markov chains in Section 6.2.3 continue to hold for continuous-time chains with essentially no change, so we will not restate them here.

6.3.2 Residence time in a homogenous continuous-time Markov chain

Analogous to the geometric distribution of residence times in a homogenous discrete-time chain, residence times are exponentially distributed for a homogenous continuous-time Markov chain for essentially the same reasons. If we denote the residence time in state j by R_j, the exponential distribution gives us the memorylessness property:

$$P(R_j > s + t | R_j > s) = P(R_j > t) \quad \textbf{(EQ 6.7)}$$

6.0.1 Stationary probability distribution for a homogenous continuous-time Markov chain

Erratum 6.21: Read "6.3.3".

The definition of the stationary probability of a homogenous continuous-time Markov chain closely follows that of a discrete-time Markov chain. Therefore, we omit the intermediate details and directly present the set of equations necessary to compute the stationary probability of a homogeneous continuous-time Markov chain.

4:21

Define the transition probability of going from state i to state j by

$$p_{ij}(t) = P(X(s + t) = j | X(s) = i) \quad \textbf{(EQ 6.8)}$$

Intuitively, this means that if the process is at state i at any time s, then the probability that it will get to state j after a time interval t, is given by $p_{ij}(t)$. This is independent of the value of s because the process is homogeneous.

Define the quantity q_{ij}, which denotes the *rate* at which the process departs from state i to state j (where j and i differ) when it is in state i:

$$q_{ij} = \lim_{\Delta t \to 0} p_{ij}(\Delta t) / \Delta t \quad \textbf{(EQ 6.9)}$$

That is, the probability that the process transitions from i to j during a small interval of length Δt time units, conditional on it already being at state i at time t, is $q_{ij}\Delta t$. We also define the negative quantity q_{ii} by:

$$q_{ii} = -\sum_{j \neq i} q_{ij} \quad \textbf{(EQ 6.10)}$$

Then, $-q_{ii}$ is the rate at which the process does *not* stay in state *i* (i.e. departs to some other state) during an interval of length Δt time units. Because $\sum_j p_{ij}(t) = 1$, (at any time *t*, the chain transitions to *some* state, including the current state) we see that

$$\sum_j q_{ij} = 0 \qquad \textbf{(EQ 6.11)}$$

With these quantities in hand, we can define the time evolution of the probability of being in state *j* at time *t*, defined as $\pi_j(t)$, by:

$$\frac{d\pi_j(t)}{dt} = q_{jj}\pi_j(t) + \sum_{k \neq j} q_{kj}\pi_k(t) \qquad \textbf{(EQ 6.12)}$$

For ergodic continuous-time Markov chains, as $t \to \infty$, these probabilities tend to the stationary probability distributions π_j^* which are implicitly defined by:

$$\begin{aligned} q_{jj}\pi_j^* + \sum_{k \neq j} q_{kj}\pi_k^* &= 0 \\ \sum_j \pi_j^* &= 1 \end{aligned} \qquad \textbf{(EQ 6.13)}$$

Note that this is Equation 6.12 with the rate of change of the probability set to 0--which is what one would expect for a stationary probability-- and with the time-dependent probabilities replaced by their time-independent limiting values.

This ends our brief summary of continuous-time Markov processes. Instead of studying general continuous-time processes, we will instead focus on a smaller but very important subclass: continuous-time birth-death processes.

6.4 Birth-Death processes

Lab: Birth–death processes

5:02

This section discusses a special class of continuous-time homogenous Markov chains having the property that state transitions are permitted from state *j* only to states *j*-1 and *j*+1 (if these states exist). This class is well-suited to describe such processes as the arrival and departure of customers from a queue (the subject of queueing theory, after all!) where the state index corresponds to the number of customers awaiting service. More precisely, if the number of customers in the system is *j*, the Markov chain is considered to be in state *j*. Customer arrivals cause the number of customers in the system to increase by one, which moves the process to state *j*+1, and this happens at a rate $q_{j,j+1}$. Similarly, customer departures (due to service) cause the process to move from state *j* to state *j-1*, and this happens at the rate $q_{j,j-1}$. In keeping with standard terminology, we denote:

$$\begin{aligned} \lambda_j &= q_{j,\,j+1} \\ \mu_j &= q_{j,\,j-1} \end{aligned} \qquad \textbf{(EQ 6.14)}$$

and these are also called the **birth** and **death rates** respectively. Note that these rates can be state-dependent (but cannot be time-dependent due to homogeneity). Also, by definition, at state *i*, the transition rates q_{ij} are 0 for all *j* other than *i*, *i-1*, and *i+1*. Given this fact, Equation 6.10, and Equation 6.14, we find that:

$$q_{jj} = -(\lambda_j + \mu_j) \qquad \textbf{(EQ 6.15)}$$

For a birth-death process, being in state *j* has the intuitive meaning that the population size is *j*, that is, there are *j* customers in the queueing system. Note that when $j = 1$, we have one customer in the system, the customer that is receiving service, and there are *none* in the queue. Generalizing, in state *j*, we have one customer receiving service and *j*-1 in the queue awaiting service.

6.4.1 Time-evolution of a birth-death process

Because a birth-death process is a continuous-time Markov chain, the time evolution of $\pi_j(t)$ is given by Equation 6.12. We substitute Equation 6.14 and Equation 6.15 to find:

4:50

$$\begin{aligned} \frac{d\pi_j(t)}{dt} &= -(\lambda_j + \mu_j)\pi_j(t) + \lambda_{j-1}\pi_{j-1}(t) + \mu_{j+1}\pi_{j+1}(t) && \text{j} \geq 1 \\ \frac{d\pi_0(t)}{dt} &= -\lambda_0\pi_0(t) + \mu_1\pi_1(t) && \text{j} = 0 \end{aligned} \qquad \textbf{(EQ 6.16)}$$

This describes the time-evolution of a birth-death system. In practice, solving these equations is complex, and does not give too many insights into the structure of the system. These are better obtained from the stationary probability distribution, which we study next.

6.4.2 Stationary probability distribution of a birth-death process

The stationary probabilty distribution of an ergodic birth-death process is given by Equation 6.13. Denoting the stationary probability of being in state j by π_j^* and substituting Equation 6.14 and Equation 6.15 into Equation 6.13, we obtain the following equations:

13:14

Erratum 6.22: Read "probability"; $\sum_j \pi_j^* = 1$.

$$\begin{aligned} 0 &= -(\lambda_j + \mu_j)\pi_j^* + \lambda_{j-1}\pi_{j-1}^* + \mu_{j+1}\pi_{j+1}^* \qquad j \geq 1 \\ 0 &= -\lambda_0\pi_0^* + \mu_1\pi_1^* \qquad j = 0 \\ \sum_j \pi_j &= 1 \end{aligned} \tag{EQ 6.17}$$

In matrix form, we can write the first two equations as

$$\boldsymbol{PQ} = \boldsymbol{0} \tag{EQ 6.18}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \pi_0^* & \pi_1^* & \pi_2^* & \dots \end{bmatrix}$$

$$\boldsymbol{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & \dots & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & \dots & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{EQ 6.19}$$

The two matrices are infinite-dimensional if the population size is unbounded. By defining $\boldsymbol{P}(t)$ as

$$\boldsymbol{P}(t) = \begin{bmatrix} \pi_0(t) & \pi_1(t) & \pi_2(t) & \dots \end{bmatrix}$$

we can rewrite Equation 6.16 as

$$d\boldsymbol{P}(t)/dt = \boldsymbol{P}(t)\boldsymbol{Q} \quad \text{(EQ 6.20)}$$

6.4.3 Finding the transition-rate matrix

The $\boldsymbol{Q}$ matrix defined by Equation 6.19 is also called the **transition rate matrix**. It is important because it allows us to derive both the time-dependent evolution of the system (i.e., $P_j(t)$), through Equation 6.20 and the long-term probability of being in state j, through Equation 6.18. Thus, in practice, the first step in studying a birth-death process is to write down its $\boldsymbol{Q}$ matrix.

Consider the following representation of a generic birth-death process in Figure 6.4:

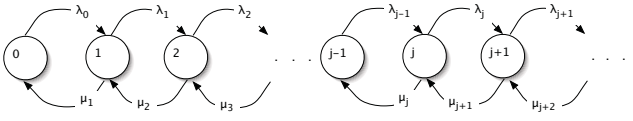


FIGURE 6.4. State-transition-rate diagram for a birth-death process

Here, we represent each state j by a circle, and we label the arc from state j to state k with the transition rate q_{jk}. From this figure, we can determine $\boldsymbol{Q}$ for a birth-death process as follows: notice that the diagonal elements of $\boldsymbol{Q}$, i.e., q_{jj} are the negative of the quantities *leaving* state j. Focussing on the jth column, the $q_{j-1,j}$th elements, immediately above the diagonal (such as element q_{01}), are the rates entering state j from state $j-1$, i.e., λ_{j-1} and the $q_{j+1,j}$th elements, (such as element q_{32}), immediately below the diagonal, are the rates entering state j from state $j+1$. All other elements are 0. In each row, the quantities sum to zero, due to Equation 6.11.

Thus, given the state-transition-rate diagram, it is possible to quickly construct $\boldsymbol{Q}$ and use it to obtain the time-dependent and time-independent (stationary) probabilities of being in each state. We will now use this approach to study some standard birth-death systems. Note that this inspection approach also applies to all continuous time Markov chains, where we can determine the elements of the $\boldsymbol{Q}$ matrix by inspecting the corresponding state-transition-rate diagram, then solving for $\boldsymbol{P}$ and $\boldsymbol{P}(t)$ using Equation 6.19 and Equation 6.20.

Erratum 6.23: Read "Equation 6.18"; $P_j(t)$ is $\pi_j(t)$.

EXAMPLE 6.18: TRANSITION-RATE MATRIX FOR A BIRTH-DEATH PROCESS

Consider the state-rate-transition diagram in Figure 6.5. What are the ***P*** and ***Q*** matrices for this system? What are the equations for its time-evolution and the long-term probability of being in each state?

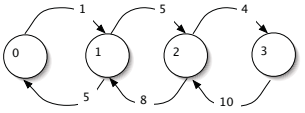


FIGURE 6.5. A simple birth-death process

The ***P*** matrix is $\left[\pi_0^* \ \pi_1^* \ \pi_2^* \ \pi_3^*\right]$ and $\boldsymbol{P}(t) = \left[\pi_0(t) \ \pi_1(t) \ \pi_2(t) \ \pi_3(t)\right]$. By inspection, we can write the ***Q*** matrix as

$$\boldsymbol{Q} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 5 & -10 & 5 & 0 \\ 0 & 8 & -12 & 4 \\ 0 & 0 & 10 & -10 \end{bmatrix}$$

Therefore, the time-evolution of state probabilities is given by:

$$\frac{d}{dt}\left(\left[\pi_0(t) \ \pi_1(t) \ \pi_2(t) \ \pi_3(t)\right]\right) = \left[\pi_0(t) \ \pi_1(t) \ \pi_2(t) \ \pi_3(t)\right]\begin{bmatrix} -1 & 1 & 0 & 0 \\ 5 & -10 & 5 & 0 \\ 0 & 8 & -12 & 4 \\ 0 & 0 & 10 & -10 \end{bmatrix}$$

$$\left[\dot{\pi}_0(t) \ \dot{\pi}_1(t) \ \dot{\pi}_2(t) \ \dot{\pi}_3(t)\right] = \left[\pi_0(t) \ \pi_1(t) \ \pi_2(t) \ \pi_3(t)\right]\begin{bmatrix} -1 & 1 & 0 & 0 \\ 5 & -10 & 5 & 0 \\ 0 & 8 & -12 & 4 \\ 0 & 0 & 10 & -10 \end{bmatrix}$$

which is a system of differential equations that can be solved to give the evolution of the time-varying probabilities $\pi_i(t)$.

The long-term probability of being in each state is given by:

$$\begin{bmatrix} \pi_0^* & \pi_1^* & \pi_2^* & \pi_3^* \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 \\ 5 & -10 & 5 & 0 \\ 0 & 8 & -12 & 4 \\ 0 & 0 & 10 & -10 \end{bmatrix} = 0$$

which is a system of linear equations in four variables that can be solved to obtain the stationary probability distribution of the chain.

■

6.4.4 A pure-birth (Poisson) process

7:31

Consider a system in which $\lambda_j = \lambda$ for all *j* (the departure rate from all states is the same), and $\mu_j = 0$ for all *j* (the death rate is 0). This represents a **Poisson** process whose population grows without bound and whose rate of growth is λ independent of the population size (that is, we expect the population to grow by 1 every $1/\lambda$ seconds independent of the current population size).

We study two properties of this process: the probability of being in state *j* at time *t*, which corresponds to having *j* arrivals in time *t*, and the distribution of inter-arrival times, that is, the expected time between going from any state to the adjacent state.

We can derive the probability of being in any state directly from Equation 6.16. Substituting the values for λ and μ in this equation, we get:

$$\begin{aligned} \frac{d\pi_j(t)}{dt} &= -\lambda\pi_j(t) + \lambda\pi_{j-1}(t) \qquad j \geq 1 \\ \frac{d\pi_0(t)}{dt} &= -\lambda\pi_0(t) \end{aligned} \qquad \text{(EQ 6.21)}$$

The second equation is a trivial differential equation whose solution is given by

$$\pi_0(t) = e^{-\lambda t} \qquad \text{(EQ 6.22)}$$

We substitute this into the first equation to get

$$\frac{d\pi_1(t)}{dt} = -\lambda\pi_1(t) + \lambda e^{-\lambda t} \quad \textbf{(EQ 6.23)}$$

whose solution is

$$\pi_1(t) = \lambda t e^{-\lambda t} \quad \textbf{(EQ 6.24)}$$

By repeatedly substituting this into the first equation, we obtain

$$\pi_j(t) = \frac{(\lambda t)^j}{j!} e^{-\lambda t} \quad \textbf{(EQ 6.25)}$$

This is the density function for the Poisson distribution (see Section 1.5.4) with parameter λt. Thus, for a Poisson process with parameter λ, the probability of j arrivals in time t, which is also the probability of being in state j at time t, is given by a Poisson distribution with parameter λt. Because the mean of the Poisson distribution is also its parameter, the expected number of arrivals in time t is λt. This is intuitively pleasing: the arrival rate is λ so in time t we should see, on average, λt arrivals.

EXAMPLE 6.19: POISSON PROCESS

Consider packets arriving to a link as a Poisson process at a mean rate of 5 packets/second. What is the probability that the link receives (a) two packets after 2 seconds (b) ten packets after 2 seconds?

4:25

Solution

We have $\lambda = 5$ and $t = 2$, so the Poisson parameter is 10. The probability of having two packets arrive to the link after 2 seconds is $\pi_2(2) = (10^2/2!)\, e^{-10} = 50*e^{-10} = 2.26*10^{-3}$. This is a rather unlikely event.

Erratum 6.24: Read 2.27×10^{-3}.

The probability of having 10 packets arrive to the link after 2 seconds is $\pi_{10}(2) = (10^{10}/10!)\, e^{-10} = 0.125$. Note that the expected number of packet arrivals after 2 seconds is 10, yet the probability that the expected number of packet arrivals is actually achieved is only 1 in 8!

■

We now derive the interarrival time distribution for a Poisson process. Let a denote the continuous random variable that represents the time between any two arrivals: we seek the distribution for a. Consider the cumulative density function of a, given by the probability $P(a \le t) = 1 - P(a > t)$. But $P(a > t)$ is just $P(0$ customer arrivals in time $(0, t)) = \pi_0(t) = e^{-\lambda t}$, $t \ge 0$. Hence, the density function of a is given by differentiating $P(a \le t)$ to get:

Erratum 6.25: Read "cumulative distribution function"; below, π_j^*.

$$a(t) = \lambda e^{-\lambda t} \qquad \textbf{(EQ 6.26)}$$

We recognize this as an exponential distribution. This gives us the following important result:

The interarrival times for a Poisson process are drawn from an exponential distribution.

We note that the exponential distribution is memoryless. Thus, for a Poisson process, not only is the rate of transitioning to the next state (the birth rate) independent of the current population size, the *time* at which this transition occurs does not depend on how long the process has been at the current population size.

6.4.5 Stationary probability distribution for a birth-death process


5:39

We now return to computing the stationary probability distribution for a general birth-death process using Equation 6.17. From the second equation, we immediately obtain

$$\pi_1^* = \frac{\lambda_0}{\mu_1}\pi_0^* \qquad \textbf{(EQ 6.27)}$$

Substituting this into the first equation, we find that

$$\pi_2^* = \frac{\lambda_0\lambda_1}{\mu_1\mu_2}\pi_0^* \qquad \textbf{(EQ 6.28)}$$

Repeating this substitution, we find that P_j is given by

$$\pi_j^* = \frac{\lambda_0\lambda_1\cdots\lambda_{j-1}}{\mu_1\mu_2\cdots\mu_j}\pi_0^* = \pi_0^*\prod_{i=0}^{j-1}\frac{\lambda_i}{\mu_{i+1}} \qquad \textbf{(EQ 6.29)}$$

We therefore obtain the long-term probabilities of being in any state *j* as a function of the probability of being in state 0 and the system parameters. Knowing that these probabilities sum to 1, we can determine

$$\pi_0^* = \frac{1}{1 + \sum_{j=1}^{\infty} \prod_{i=0}^{j-1} \frac{\lambda_i}{\mu_{i+1}}} \quad \text{(EQ 6.30)}$$

This can be substituted back into Equation 6.29 to obtain the long-term probability of being in any state *j*. Of course, we need to ensure that the series in the denominator of Equation 6.30 actually converges! Otherwise, π_0^* is undefined, and so are all the other π_i^* s. It can be shown that the condition for convergence (as well as for the chain to be ergodic) is $\sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\mu_n}{\lambda_n} = \infty$. and $\sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n} < \infty$. Intuitively, these conditions ensure that the long-run average departure rate exceeds the long-run average arrival rate.

EXAMPLE 6.20: GENERAL EQUILIBRIUM SOLUTION

Find the equilibrium probabilities of being in each state for the birth-death process shown in Figure 6.5.

Solution

From Equation 6.30, we get

π_0^* = 1/[1 + 1/5 + (1*5)/(5*8) + (1*5*4)/5*8*10)] = 0.73.

This can be substituted into Equation 6.29 obtain

Erratum 6.26: Labels are $\pi_1^*, \pi_2^*, \pi_3^*$; see correction 6.26.

π_1^* = 1/5 π_0^* = 0.2 * 0.73 =0.146.

π_1^* = 1/8 π_0^* = 0.125 * 0.73 =0.09.

π_2^* = 1/20 π_0^* = 0.05 * 0.73 = 0.0365.

As a check, 0.73 + 0.146 + 0.09 + 0.0365 = 1.0025, which is within the rounding error.

■



Introduction to Queueing Theory
4:42

6.5 The M/M/1 queue



M/M/1
11:59

Lab: M/M/1 queue

The M/M/1 queue is the simplest non-trivial queueing system. Here, the a/b/c notation, also called **Kendall notation**, denotes the following:

1. (the 'a' portion in the notation): the arrival process is 'Markovian', i.e, it is a Poisson process with exponentially distributed inter-arrival times
2. (the 'b' portion in the notation): the departure process is 'Markovian', i.e., it is a Poisson process with exponentially-distributed inter-departure times
3. (the 'c' portion in the notation): there is a single server

Erratum 6.27: 'b' describes the service-time distribution.

Extended forms of the notation describe the size of the buffers available (we assume an infinite number), as well as the service discipline (we assume first-come-first-served), and other queueing parameters. However, the three-parameter version of the notation is the one that is commonly used.

The M/M/1 queueing system is a birth-death Markov process with a state-independent arrival rate λ and a state-independent departure rate μ. These rates are therefore also independent of the population size. This is counter-intuitive, in that the rate of departure from a small population is the same as the rate of departure from a large population.

We study the long-term behaviour of the M/M/1 queue by removing state-dependence (the subscript j) in the transition rates in the analysis of Section 6.4.5. From Equation 6.29, we find that:

$$\pi_j^* = \pi_0^* \prod_{i=0}^{j-1} \frac{\lambda}{\mu} = \pi_0^* \left(\frac{\lambda}{\mu}\right)^j \qquad j \geq 0 \qquad \textbf{(EQ 6.31)}$$

To obtain π_0^*, we use Equation 6.30 to obtain

$$\pi_0^* = \frac{1}{\left(1 + \sum_{j=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^j\right)} \quad \textbf{(EQ 6.32)}$$

When $\lambda < \mu$, the infinite sum in the denominator converges, and the denominator reduces to $\left(\frac{1}{1-\frac{\lambda}{\mu}}\right)$, so that

$$\pi_0^* = 1 - \frac{\lambda}{\mu} \quad \textbf{(EQ 6.33)}$$

The ratio λ/μ represents the intensity of the arrival rate, as a fraction of the service rate and can be viewed as the utilization of the system. The value is important enough that it deserves its own symbol, ρ, which allows us to write Equation 6.33 as

$$\pi_0^* = 1 - \rho \quad \textbf{(EQ 6.34)}$$

This equation has the intuitive meaning that the probability that the system is idle (i.e., π_0^*) is (1 - utilization). It can be shown that this relationship is true for *all* queueing systems whose population size is unbounded.

Erratum 6.28: True for single-server queues only.

We now use Equation 6.31 to obtain the stationary probability of being in any state *j* as:

$$\pi_j^* = \rho^j(1-\rho) \quad \textbf{(EQ 6.35)}$$

Note that this is a geometric distribution.

EXAMPLE 6.21: M/M/1 QUEUE

Consider a link to which packets arrive as a Poisson process at a rate of 300 packets/sec such that the time taken to service a packet is exponentially distributed. Suppose that the mean packet length is 500 bytes, and that the link capacity is 1.5 Mbps. What is the probability that the link's queue has 1, 2 and 10 packets respectively?

Solution

The packet length is 500 bytes = 4000 bits, so that the link service rate of 1,500,000 bits/sec = 375 packets/sec. Therefore, the utilization is 300/375 = 0.8. When the link queue has 1 packet, it is in state $j = 2$, because one packet is being served at that time. Thus, we need $\pi_2^* = 0.8^2 * 0.2 = 0.128$. For the queue having two packets, we compute $\pi_3^* = 0.8^3 * 0.2 = 0.1$. For 10 packets in the queue, we compute $\pi_{11}^* = 0.8^{11} * 0.2 = 0.0067$, an fairly small quantity. Thus, even when the utilization is high (80%), the queue size is quite small, rarely exceeding 10 packets.

Erratum 6.29: Read $\pi_{11}^* = 0.0172$; see correction 6.29.

■

Note that the long-term probability that the population size is j depends only on the utilization of the system. As the utilization increases, the probability of reaching larger population sizes increases. To see this analytically, consider the mean number of customers in the system (which is also the mean population size), denoted $\bar{N}$ defined by

$$\bar{N} = \sum_{j=0}^{\infty} j\pi_j^* \qquad \textbf{(EQ 6.36)}$$

It can be shown that this sum converges (i.e., when $\lambda < \mu$) and that

$$\text{Mean number of customers in the system} = \bar{N} = \frac{\rho}{(1-\rho)} \qquad \textbf{(EQ 6.37)}$$

EXAMPLE 6.22: MEAN NUMBER OF CUSTOMERS IN THE SYSTEM

Compute the mean number of packets in the system of Example 6.21.

Solution

The utilization is 0.8, so the mean number of packets in the system is 0.8/(1-0.8) = 0.8/0.2 = 4.

■

It is obvious from Equation 6.37 that as $\rho \to 1$, $\bar{N} \to \infty$. That is, as the arrival rate approaches the service rate, the expected number of customers in the system grows without bound. This is somewhat unexpected: after all, the

arrival rate is smaller than the service rate: why then should the number of customers grow? The reason is that we are dealing with stochastic processes. Even though the arrival rate, on average, is lower than the service rate, there will be time periods when the short-term arrival rate exceeds the service rate. For instance, even if the mean arrival rate is 1 customer per second, there will be short intervals during which two or even three customers may arrive in one second. During this time, the queue builds up, and is drained when the service rate exceeds the arrival rate. In fact, there is an interesting asymmetry in the system: when the short-term arrival rate exceeds the short-term service rate, the queue builds up, but when the service rate exceeds the arrival rate, if the queue is empty, the system does not build up 'service credits.' The server is merely idle. Thus, the system tends to build up queues that are drained only over time. This is reflected in the fact that as the utilization of the system increases, the mean number of customers in the system increases sharply.

It is remarkable that this fundamental insight into the behaviour of a real queueing system can be derived with only elementary queueing theory. Moreover, this insight carries over to all other queueing systems: as the utilization approaches 1, the system becomes **congested**. The behaviour of the mean queue length (which also corresponds to the waiting time, through Little's theorem), is shown in Figure 6.6.

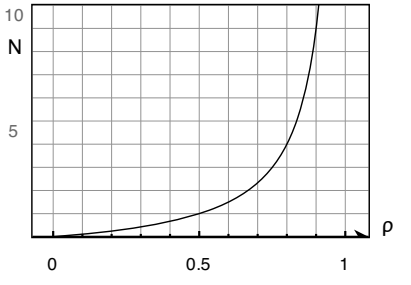


FIGURE 6.6. Mean queue length as a function of utilization

It is clear that the queue length asymptotes to infinity as the utilization approaches 1. In networking terms, this means that as the arrival rate approaches a link's capacity, the queue at the immediately preceding router or switch will grow without bound, causing packet loss. This analysis allows us to derive a practical guideline: we should provision enough service capacity so that the system utilization never exceeds a threshold of around 70%.

Alternatively, if this threshold is exceeded, either service requests should be dropped, or new service capacity should be made available so that the utilization decreases.

Another related quantity of interest for this queue is the mean waiting time in the queue. From Little's theorem, the mean number of customers in the system is the product of their mean waiting time and their mean arrival rate, so $\frac{\rho}{(1-\rho)}$ = mean waiting time *λ, which means that

$$\text{Mean waiting time} = \frac{\frac{\rho}{\lambda}}{(1-\rho)} = \frac{\frac{1}{\mu}}{(1-\rho)} \qquad \text{(EQ 6.38)}$$

Erratum 6.30: EQ 6.38 is the mean time in the system.

This quantity also grows without bound as the utilization approaches 1.

EXAMPLE 6.23: WAITING TIME OF A M/M/1 QUEUE

What is the mean waiting time for a packet in the ~~queue~~ system described in Example 6.21?

Solution

For this queue, μ = 375 and ρ = 0.8. So, the mean waiting time is (1/375)/(1-0.8) = 5/375 seconds = 13.3 ms.

■

6.6 Two variations on the M/M/1 queue

Lab: M/M/1/K queue

We now briefly consider two variations on the M/M/1 queue, essentially to give insight into how one proceeds with the analysis of a queueing system.

6.6.1 The M/M/∞ queue: a responsive server

5:16

Suppose that a provider of service capacity brings on a new server to serve every arriving customer. This is like a private bank where new agents are brought on to provide individual attention to each customer when she or he

arrives. This system can be modelled as a queue with an infinite number of servers (though, at any time, the number of servers is finite).

We can model and analyse this queue using the same techniques as with an M/M/1 queue. We start with the state-transition-rate diagram show in Figure 6.7. Note that μ_j, the rate of departure from the jth queue, is $j\mu$

Erratum 6.31: Read "from state *j*".

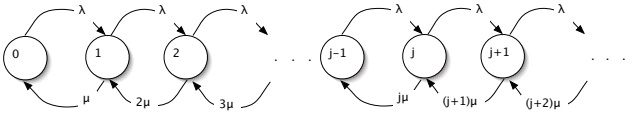


FIGURE 6.7. State-transition-rate diagram for an M/M/∞ queues

which models the fact that when there are j customers, there are j servers. From the diagram, we can directly invoke Equation 6.29 to write down π_j^*, the stationary probability of being in state j, as

$$\pi_j^* = \pi_0^* \prod_{i=0}^{j-1} \frac{\lambda}{(i+1)\mu} = \pi_0^* \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!} \quad \textbf{(EQ 6.39)}$$

We solve for π_0^* using Equation 6.30 as

$$\pi_0^* = \frac{1}{\left[1 + \sum_{j=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}\right]} \quad \textbf{(EQ 6.40)}$$

Recalling the standard expansion $e^x = \sum_{j=0}^{\infty} \frac{x^j}{j!}$, we see that

$$\pi_0^* = e^{-\frac{\lambda}{\mu}} \quad \textbf{(EQ 6.41)}$$

$$\pi_j^* = \pi_0^* \prod_{i=0}^{j-1} \frac{\lambda}{(i+1)\mu} = e^{-\frac{\lambda}{\mu}} \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!} \quad \textbf{(EQ 6.42)}$$

Equation 6.42 shows that the stationary probability of being in state *j* is given by the Poisson distribution with parameter λ/μ. Thus, with 'infinite' servers, the number of customers in the system follows the Poisson distribution. This allows us to compute the expected number of customers in the system as the mean of the Poisson, which is its parameter, i.e., λ/μ. All other parameters of interest for this queueing system can be derived from Equation 6.42.

EXAMPLE 6.24: RESPONSIVE SERVER

Suppose customers arrive at a private bank, modelled as a responsive server, as a Poisson process at the rate of 10 customers/hour. Suppose that a customer's needs can be met on average in 20 minutes, and that the service time distribution is exponentially distributed. What is the probability that there are five customers in the bank at any point in time?

Solution

We have λ = 10 and μ = 3 (i.e. 3 customers can be served an hour, on average, by a server). Thus $\pi_0^* = e^{-10/3} =$ 0.036. We need to find $\pi_5^* = 0.036 * (10/3)^5 * 1/5! = 0.123$. Thus, there is a nearly one in eight chance that there will be 5 customers in the bank at any given time.

■

6.6.2 M/M/1/K: bounded buffers

5:34

Suppose that the queueing system has only *K*-1 buffers. In this case, the population size (including the customer in service) cannot grow beyond *K*, and arrivals when the system is in state *K* are lost (similar to packet loss when arriving to a full queue). To model this, we can simply ignore arrivals to state *K*, which means that we will never enter states *K*+1, *K+2*,... This results in a state-transition-rate diagram shown in Figure 6.8.

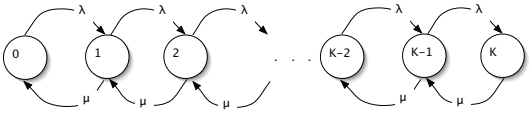


FIGURE 6.8. State-transition-rate diagram for an M/M/1/K queue

The state transition rates are therefore

$$\lambda_j = \begin{cases} \lambda & j < K \\ 0 & j \geq K \end{cases} \qquad \text{(EQ 6.43)}$$

$$\mu_j = \mu \qquad j=1,2,...,K$$

We can therefore use Equation 6.30 to write π_j^* as

Erratum 6.32: Read “Equation 6.29”; below, “Equation 6.30”.

$$\pi_j^* = \begin{cases} \pi_0^* \left(\frac{\lambda}{\mu}\right)^j & j \leq K \\ 0 & j > K \end{cases} \qquad \text{(EQ 6.44)}$$

We use Equation 6.44 to obtain

$$\pi_0^* = \frac{1}{\left[1 + \sum_{j=1}^{K} \left(\frac{\lambda}{\mu}\right)^j\right]} \qquad \text{(EQ 6.45)}$$

Given the standard result $\sum_{k=0}^{n-1} r^k = \frac{1-r^n}{1-r}$, we can simplify this to

$$\pi_0^* = \frac{1-\frac{\lambda}{\mu}}{1-\left(\frac{\lambda}{\mu}\right)^{K+1}} \qquad \text{(EQ 6.46)}$$

So, we can now write Equation 6.44 as

$$\pi_j^* = \begin{cases} \frac{1-\frac{\lambda}{\mu}}{1-\left(\frac{\lambda}{\mu}\right)^{K+1}}\left(\frac{\lambda}{\mu}\right)^j = \frac{1-\rho}{1-\rho^{K+1}}\rho^j & j \leq K \\ 0 & j > K \end{cases} \qquad \text{(EQ 6.47)}$$

As before, given these probabilities, we can compute all quantities of interest about the queueing system, such as the distribution of the queue length, the mean number of customers in the queue, and the mean waiting time. In particular, the intuitive meaning of π_K^* is the probability that the system is 'full' when it has a buffer of size *K*-1. So, π_K^* can be interpreted as the **blocking probability** of a M/M/1/K queue. We can then choose *K* as a sizing parameter to make π_K^* as small as desired.

Erratum 6.33: Read "an M/M/1/K".

Note that in this system, $\pi_0^* \neq 1 - \rho$ because the system size is bounded. Therefore, the number of customers served in a chosen time period may be lower than what the utilization indicates because customer arrivals when the queue is full are lost. Moreover, the system is stable by definition, independent of the utilization, because excess arrivals are automatically dropped.

EXAMPLE 6.25: M/M/1/K QUEUE

Consider the system of Example 6.21, but with the restriction that the queue only has four buffers. What is the probability that three of these are in use? How many buffers should we provision to ensure that the blocking probability is no more than 10^{-6}?

Solution

We have $K = 5$, and $\frac{\lambda}{\mu}$=0.8. From Equation 6.46, $\pi_0^* = (1\text{-}0.8)/(1\text{-}0.8^6) = 0.27$. If three buffers are in use, then the system is in state $j = 4$. From Equation 6.44, we get $\pi_4^* = 0.27(0.8)^4$=0.11.

To size the buffer, we have to choose *K* such that $\pi_K^* < 10^{-6}$. We solve for K^* by using the inequality $10^{-6} > ((.2)(0.8)^K)/(1\text{-}0.8^{K+1})$, to obtain $K^* = 55$. Thus, we need 54 buffers to satisfy this blocking probability.

■

6.7 Other queueing systems

Advanced texts go beyond the Markovian and exponential framework. These queueing systems become much more difficult to analyze, so we will merely state the results for two important results.

6.7.1 M/D/1: deterministic service times

Consider a queueing system in which arrivals are from a Poisson process, but service times are deterministic. That is, as long as the queue is non-empty, the inter-departure time is deterministic (rather than exponentially distributed). Representing the departure rate (a constant) by μ, and the utilization by $\rho = \lambda/\mu$, it can be shown that the system is stable (i.e., the queue length is finite) as long as $\lambda < \mu$. Moreover, the long-term probability that the number of customers in the system is j, i.e., P_j is given by

$$\pi_j^* = \begin{cases} 1-\rho & j=0 \\ (1-\rho)(e^{\rho}-1) & j=1 \\ (1-\rho)\left(\sum_{i=0}^{j} \frac{(-1)^{j-i}(i\rho)^{j-i-1}(i\rho + j - i)e^{i\rho}}{(j-1)!} \right) & j>1 \end{cases} \quad \textbf{(EQ 6.48)}$$

Erratum 6.34: Denominator is $(j-i)!$; see correction 6.34.

This allows us to derive the mean number of customers in the system as

$$\text{Mean customers in the system} = \rho + \frac{\rho^2}{2(1-\rho)} \quad \textbf{(EQ 6.49)}$$

and the mean response time as

$$\text{Mean response time} = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)} \quad \textbf{(EQ 6.50)}$$

Other quantities of interest regarding the M/D/1 queue can be found in standard texts on queueing theory.

6.7.2 G/G/1

Once the arrival and service processes become non-Poisson, the analysis of even a single queue becomes challenging. For such systems few results are available other than Little's theorem, and also that, if the queue size is unbounded, $\pi_0^* = 1-\rho$. See the attached notes for more about this queue.

1:50

6.7.3 Networks of queues

So far, we have studied the behaviour of only a single queue. This is like studying a network with a single router - not very interesting! What happens when we link the output of a queue to the input of another queue, as we do in any computer network? Intuitively, we make the inter-departure process of one queue the inter-arrival process for the second queue. Moreover, we may have more than one inter-departure process mix to form the inter-arrival process. Can this be analysed?

6:07

We represent this composite system, also called a 'tandem of queues' as shown in Figure 6.9.

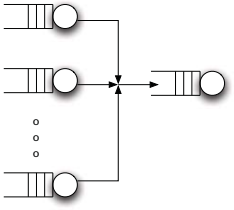


FIGURE 6.9. A network of queues

Here, each queue is shown by a buffer (with customers or jobs in it) and a server (represented by a circle). Jobs served by the servers on the left enter the queue of the server on the right. Each queue and associated server is also called a *node* (drawing on the obvious graph analogy).

If all the queues on the left are M/M/1 queues, recall that their inter-departure processes are Poisson. Moreover, it can be shown that the mixture of Poisson processes is also a Poisson process whose parameter is the sum of the individual processes. Therefore, the input to the queue on the right is a Poisson process that can be analysed as an M/M/1 queue. This leads to the fundamental insight that a tandem of M/M/1 queues is analytically tractable.

Because the departure process of a M/M/*m* queue (i.e. a queue with *m* servers) is also Poisson, this result holds true for tandems of M/M/*m* queues.

We can make things a bit more complicated: we can allow customers to enter *any* queue (node) as a Poisson process and we can also allow customers that leave a node to exit the system altogether with some probability or join any other node in the system with some probability. Note that this can potentially lead to cycles, where customers go through some set of nodes more than once. Nevertheless, Jackson was able to show that these networks behave as if each M/M/m queue was being fed by a single Poisson stream. Such networks are also called **Jacksonian** networks in his honour. For a Jacksonian network, we have a strong result: Let $\pi^*_{k_1 k_2 \ldots k_n}$ denote the long-term probability that there are k_1 customers at the first node, k_2 customers at the second node, and so on. Then:

$$\pi^*_{k_1 k_2 \ldots k_n} = \pi^*_{k_1} \pi^*_{k_2} \ldots \pi^*_{k_n} \quad \text{(EQ 6.51)}$$

That is, the joint probability of having a certain number of customers in each queue is the product of the individual probabilities. We interpret this to mean that each queue in the system acts as if it were independent of the others. This **product form** of the probability distribution greatly simplifies analysis.

Despite the elegance and power of Jacksonian network analysis, it can be rarely applied to study practical computer networks because customers (packets) in real networks rarely arrive as a Poisson process. Thus, the output process is also non-Poisson, which makes subsequent analysis complex. In recent years, the development of network calculus and stochastic network calculus has allowed significant inroads into the study of non-Jacksonian networks.

Additional material: open and closed systems 3:24

6.8 Further reading

The definitive introductory text on queueing theory is the two-volume text: L. Kleinrock, Queueing Systems, Wiley Interscience, 1975. A modern and through treatment of Markov chains can be found in P. Bremaud, Markov Chains, Springer, 1999. Further details on network calculus can be found in J.-Y. Le Boudec and P. Thiran, Network Calculus, Springer, 2001.

Erratum 6.35: Read "thorough".

*

6.9 Exercises

1. Little's theorem

Patients arriving at the Grand River Hospital Emergency Room have a mean waiting time of 3 hours. It has been found that, averaged over the period of a day, patients arrive at the rate of one every 5 minutes.

a. How many patients are awaiting treatment on average at any given point in time?

b. What should the size of the waiting room be so that it can always accommodate arrivals?

2. A stochastic process

Consider that in Example 6.3, a person is on an infinite staircase on stair 10 at time 0 and potentially moves once every clock tick. Suppose that the person

moves from stair i to stair $i + 1$ with probability 0.2, and from stair i to stair $i - 1$ with probability 0.2 (the probability of staying on stair i is 0.6). Compute the probability that the person is on each stair at time 1 (after the first move), time 2, and time 3.

3. Discrete- and continuous-state and time processes

Come up with your own examples for all four combinations of discrete state/discrete time/continuous state/continuous time processes.

4. Markov process

Is the process in Exercise 2 a Markov process? Why or why not?

5. Homogeneity

Is the process in Exercise 2 homogeneous? Why or why not?

6. Representation

a. Represent the process in Exercise 2 by using a transition matrix and a state transition diagram.

b. Do the rows in this matrix have to sum to 1? Do the columns in this matrix have to sum to 1? Why or why not?

c. Now, assume that the staircase has only four steps. Make appropriate assumptions (what are these?) to represent this finite process as a transition matrix and a state-transition diagram.

7. Reducibility

Is the chain in Exercise 2 reducible? Why or why not?

8. Recurrence

Is state 1 in the chain in Exercise 6(c) recurrent? Compute $f_1^{\,1}$, $f_1^{\,2}$, and $f_1^{\,3}$.

9. Periodicity

Is the chain in Exercise 2 periodic? If not, give an example of a chain with period N for arbitrary $N > 1$.

10. Ergodicity

Is any state in the chain of Exercise 6(c) nonergodic? Why or why not?

11. Stationary probability

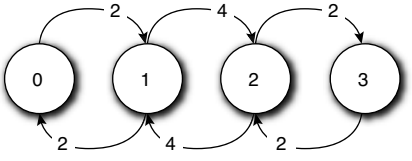
Compute the stationary probability distribution of the chain in Exercise 6(c).

12. Residence times

Compute the residence time in each state of the Markov chain in Exercise 6(c).

13. Stationary probability of a birth-death process

Consider the following state-rate-transition diagram.



a. Compare this with the state-transition probability diagram in Exercise 6(c). What features are the same, and what differ?

b. Write down the $\boldsymbol{Q}$ matrix for this system.

c. Use the $\boldsymbol{Q}$ matrix to compute the stationary-probability distribution of this chain.

14. Poisson process

Prove that the interdeparture time of a pure-death process is exponentially distributed.

15. Stationary probabilities of a birth-death process

Use Equation 6.30 to compute the stationary probability of the birth-death process in Exercise 13.

16. M/M/1 queue

Is the birth-death process in Exercise 13 M/M/1? Why or why not?

17. M/M/1 queue

Consider a link to which packets arrive as a Poisson process at a rate of 450 packets/sec such that the time taken to service a packet is exponentially distributed. Suppose that the mean packet length is 250 bytes and that the link capacity is 1 Mbps.

a. What is the probability that the link's queue has one, two, and ten packets respectively?

b. What is the mean number of packets in the system? What is the mean number in the queue?

c. What is the mean waiting time?

18. Responsive (M/M/∞) server

Compute the ratio of π_j^* for a responsive server to the same value for an M/M/1 queue. How does this ratio behave as a function of j?

19. M/M/1/K server

Assume that the queueing system in Exercise 17 has ten buffers. Compute an upper bound on the probability of packet loss.

Erratum 6.36: Compute the exact loss probability; see correction 6.36.

20. M/D/1 queue

Compute the mean number of customers in an M/D/1 system that has a utilization of 0.9.

a. How does this compare with a similarly loaded M/M/1 system?

b. Compute the ratio of the mean number of customers as a function of ρ.

c. Use this to compare the behavior of an M/D/1 queue with that of an M/M/1 queue under heavy load.



The M/G/1 Queue

A supplementary note to Chapter 6. This note presents an overview of the $M/G/1$ queue.

Lab: M/D/1 and M/G/1 queues

M.1 Introduction

Our discussion so far has assumed that both interarrival times and service times are exponentially distributed (Markov). While Markov arrivals are quite reasonable in practice, especially given that the merger of two Markov arrival streams results also in a Markov stream, service times are usually far from exponential. For example, in the Internet, packets are usually bimodally distributed in length, with large data packets and short acknowledgments. Since these are transmitted over constant-rate links, the service times are also bimodal. Specifically, the service time of a packet is not drawn from an exponential distribution; instead, it is drawn from a bimodal distribution. Thus, if we 'look' at the queue at a particular point in time, the expected remaining service time for the packet currently in service is not memoryless, which is only true for the exponential distribution. This makes the analysis more complex.

In this note, we will not derive the formulae for the M/G/1 queue, and instead only present the results.

M.2 Parameters

The three parameters that are used to describe an M/G/1 queue are:

- λ, the mean arrival rate;
- μ, the mean service rate. Note that if the service time is denoted by S, then $E(S) = 1/\mu$;
- C_S^2, which is the square of the coefficient of variation of the service time: $C_S^2 = \dfrac{E(S^2)}{(E(S))^2} - 1$.

M.3 Expected Time in System

Given these parameters, the expected value of T, the time in the system for a typical job, that is, the sum of the time waiting for service and the service time, is given by:

$$E(T) = \frac{1}{\mu}\left(1 + \frac{\rho(1 + C_S^2)}{2(1 - \rho)}\right). \quad \textbf{(EQ 1)}$$

M.4 Expected Number of Jobs in the System

The expected number of jobs in the system, denoted $E(N)$, is given by:

$$E(N) = \lambda E(T) = \rho + \frac{\rho^2(1 + C_S^2)}{2(1 - \rho)}. \quad \textbf{(EQ 2)}$$

This is also known as the Pollaczek-Khinchine formula. Note that the number of jobs in the queue depends not only on λ and μ, as in the M/M/1 queue, but also on C_S^2.

- If this value is much larger than 1, then the number of jobs in the system is much greater than in an M/M/1 queue with the same arrival and departure rates.
- If $C_S^2 = 1$, which is the case for exponential service times, then $E(N)$ reduces to $\dfrac{\rho}{1 - \rho}$, which is the formula for the number of jobs in the system for the M/M/1 queue.
- If C_S^2 is smaller than 1, then the M/M/1 queue can be used to approximate the system, though the results will be somewhat conservative.

EXAMPLE M.1: TIME IN THE SYSTEM

Suppose that the service time distribution of a queue is described by

$C_S^2 = 5$. Compare the time in the system for this queue with that of a M/M/1 queue with the same mean arrival and departure rates.

Solution:

From Eq. 1, for the M/M/1 queue, where $C_S^2 = 1$,

$$E(T) = \frac{1}{\mu}\left(1 + \frac{\rho(1 + C_S^2)}{2(1-\rho)}\right) = \frac{1}{\mu}\left(1 + \frac{\rho}{1-\rho}\right) = \frac{1}{\mu}\left(\frac{1}{1-\rho}\right).$$

When $C_S^2 = 5$,

$$E(T) = \frac{1}{\mu}\left(1 + \frac{6\rho}{2(1-\rho)}\right) = \frac{1}{\mu}\left(\frac{1+2\rho}{1-\rho}\right).$$

Taking their ratio, we see that for this queue, the mean time in the system exceeds that of a corresponding M/M/1 queue by a factor of $1 + 2\rho$.

M.5 Exercises

1. M/D/1 versus M/M/1

Prove that the mean number of jobs waiting or in service for a M/D/1 queue with mean arrival rate λ and mean service rate μ is always lower than in an M/M/1 queue with the same parameters.

2. Time in the system

Suppose that the service time distribution of a queue whose utilization is 80% is described by $C_S^2 = 3$. Compare the mean time in the system for this queue with that of a M/M/1 queue with the same mean arrival and departure rates.

3. MacroHard Doors

By instrumenting the time when a job starts service in the MacroHard Doors operating system, you discover that, for the first 1000 submitted jobs, the sum of service times is 412 seconds, and the sum of the squared service times is 1425 seconds^2. Assuming that jobs arrive to the system as a Poisson process and the mean utilization is 0.75, what is the mean number of jobs in the system, including the one in service?

Corrections to Chapter 6

The pages of Chapter 6 (Stochastic Processes and Queueing Theory) are reproduced from the book with corrections. The following errata are not yet corrected in the text; each is flagged by a red note in the margin of the page concerned.

6.1 Section 6.1, page 251: "An average of λt customer arrive" should read "customers arrive". Later on the page, "on the number of servers, on the way in which customers arrive" should read "..., or on the way in which customers arrive".

6.2 Page 256: "we will ask they person" should read "we will ask the person".

6.3 Page 256: "Suppose that at some point, in time we see" should read "Suppose that at some point in time, we see".

6.4 Page 256: "depends not only on X_n, but also on $X_{n-1}, X_{n-1}, \ldots, X_1$" should read "... on $X_{n-1}, X_{n-2}, \ldots, X_1$".

6.5 Page 256: Markov's first paper on what are now called Markov chains appeared in 1906, not 1907.

6.6 Example 6.8, page 257: "Suppose that we know that $P(X_2) = 1$" should read "Suppose that we know that $P(X_2 = 1)$".

6.7 Page 257: "we already know the the sequence" should read "we already know the sequence". The video label in the margin should read "Homogeneity".

6.8 Page 258: "in Example 8" should read "in Example 6.8", and "(see Example 10)" should read "(see Example 6.10)".

6.9 Example 6.9, page 258: "compute the distribution X_{n+1} given that of X_n" should read "compute the distribution of X_{n+1} given that of X_n".

6.10 Page 259: the sum in Equation 6.2 comes from summing across mutually exclusive events (the possible intermediate states k), not independent events. Also, "in the first formulation traces" should read "the first formulation traces".

6.11 Page 259: after Equation 6.2, "(going from state i to state k and from state k to state m)" should read "... and from state k to state j)".

6.12 Page 260: "a process that starts from an odd-numbered state, it will forever stay" should read "... state will forever stay", and "unreachable from the odd-numbered steps states" should read "unreachable from the odd-numbered states".

6.13 Page 261: “traejectory” should read “trajectory”.

6.14 Page 262: “a ensemble average” should read “an ensemble average” (twice).

6.15 Page 262: “the returning times do not have a least common divisor larger than 1” should read “. . . a greatest common divisor larger than 1”.

6.16 Page 263: “becayuse” should read “because”.

6.17 Page 263, after Theorem 6.1: “the process has a non-zero probability of returning to at least one of the states” does not make that state recurrent; recurrence needs a return with probability 1. In the second and third types of chain, the process returns to at least one state with probability 1. The conclusion is still correct, because recurrence is a class property: in an irreducible chain, if one state is recurrent, all states are.

6.18 Example 6.15, page 265: the two terms are summed because the events are mutually exclusive, not because they are independent.

6.19 Example 6.16, page 266: “we can use Theorem 2” should read “we can use Theorem 6.2”.

6.20 Example 6.17, page 267: all three results are labelled “E(residence time in state 1)”; the second and third should be labelled states 2 and 3. Also, in Equation 6.6, $P(X(t_{n+1} = j) \mid \ldots)$ should read $P(X(t_{n+1}) = j \mid \ldots)$.

6.21 Page 268: the subsection “Stationary probability distribution for a homogenous continuous-time Markov chain” should be numbered 6.3.3, not 6.0.1.

6.22 Page 271: “probabilty” should read “probability”. In Equation 6.17 the normalization should read $\sum_j \pi_j^* = 1$.

6.23 Page 272, last paragraph of Section 6.4.3: “solving for P and $P(t)$ using Equation 6.19 and Equation 6.20” should read “. . . using Equation 6.18 and Equation 6.20”. $P_j(t)$ here is the $\pi_j(t)$ used in the rest of the section.

6.24 Example 6.19, page 275: $(10^2/2!)e^{-10} = 2.27 \times 10^{-3}$, not 2.26×10^{-3}.

6.25 Page 276: “the cumulative density function” should read “the cumulative distribution function”, and “we find that P_j is given by” should read “we find that π_j^* is given by”.

6.26 Example 6.20, page 277: the three results are labelled π_1^*, π_1^*, π_2^*; they should be π_1^*, π_2^*, π_3^*. The expression for π_0^* should read $(1 \cdot 5 \cdot 4)/(5 \cdot 8 \cdot 10)$ (a parenthesis is missing), and “substituted into Equation 6.29 obtain” should read “. . . to obtain”. The numbers are correct.

6.27 Page 278, Kendall notation, item 2: the second letter describes the service-time distribution ('M' means exponential service times), not the departure process.

6.28 Page 279: $\pi_0^* = 1 - \rho$ holds for single-server queues with an unbounded buffer. It does not hold for multi-server queues; for example, for the M/M/∞ queue $\pi_0^* = e^{-\rho}$.

6.29 Example 6.21, page 280: $\pi_{11}^* = 0.8^{11} \times 0.2 = 0.0172$, not 0.0067. The statement that the queue size rarely exceeds 10 packets should be qualified: more than 10 packets are in the queue with probability $0.8^{12} \approx 0.069$, that is, about 7% of the time. Also, "an fairly small quantity" should read "a fairly small quantity".

6.30 Page 282: Equation 6.38, $(1/\mu)/(1 - \rho)$, is the mean time in the system, including service, not the mean waiting time in the queue. The mean waiting time in the queue is $\rho/(\mu(1 - \rho))$. Example 6.23's 13.3 ms is the mean time in the system. The title of Example 6.23 should read "... of an M/M/1 queue".

6.31 Page 283: "μ_j, the rate of departure from the jth queue, is $j\mu$" should read "... from state j ...". In the caption of Figure 6.7, "an M/M/∞ queues" should read "an M/M/∞ queue".

6.32 Page 285: "We can therefore use Equation 6.30 to write π_j^*" should read "Equation 6.29", and "We use Equation 6.44 to obtain $[\pi_0^*]$" should read "Equation 6.30".

6.33 Page 286: "blocking probability of a M/M/1/K queue" should read "of an M/M/1/K queue".

6.34 Page 287, Equation 6.48 (M/D/1): the denominator in the sum should be $(j - i)!$, not $(j - 1)!$. With $(j - i)!$ the probabilities sum to 1 and give the mean of Equation 6.49.

6.35 Page 289: "A modern and through treatment" should read "A modern and thorough treatment".

6.36 Section 6.9, Exercise 19: with Poisson arrivals, the blocking probability π_K^* is exactly the probability that a packet is lost (arrivals see time averages), so an upper bound is not needed; the exercise can ask for the probability of packet loss itself.

5

Signals, Systems, and Transforms

This chapter focuses on **transforms**: a family of techniques used to study a series of values, or **signals**. Transforms give insight into the nature of signals and help design effective **systems** to manipulate signals. Transforms form the basis for the study of control theory, discussed in more detail in Chapter 8.

We begin by covering background material on complex numbers and Euler's formula. We then discuss types of signals and systems. Subsequently, the bulk of the chapter is devoted to the study of Fourier series, Fourier transforms, Laplace transforms, and the Z transform.

5.1 Background

5.1.1 Sinusoids

A communication network is formed from communication links. A widely used approach to transferring information over a wired or wireless link is to modify an underlying continuous **carrier** signal to represent the symbol values 0 and 1. In nearly all cases of interest, the carrier signal is **sinusoidal**. That is, it is associated with a time period T such that at time t, the amplitude of the signal is described by the function $\cos\left(\frac{2\pi t}{T}\right)$, as shown in Figure 5.1.

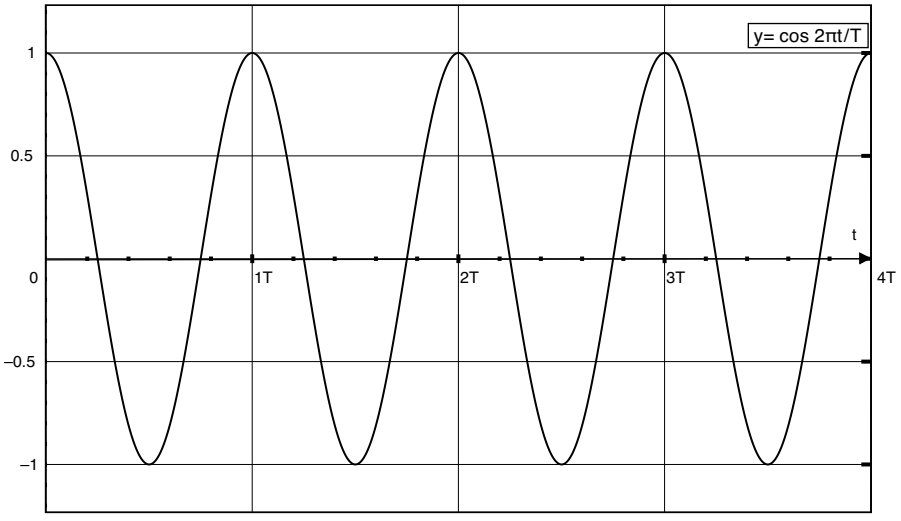


Figure 5.1 A typical carrier wave described by a sinusoidal curve

2:47

How can we generate such a time-varying signal? Consider a disk with a radius of one unit that has an axle at its center. Suppose that it rotates in the counterclockwise direction with a uniform angular velocity and with a period of T seconds. Let us paint a small dot on its rim and draw a vector from the center to this dot. The angle that this rotating vector makes with the X-axis at time t is denoted by θ and given by $\theta = \frac{2\pi t}{T}$ radians (Figure 5.2). At time t, the distance of the dot from the Y-axis is given by $\cos(\theta) = \cos\left(\frac{2\pi t}{T}\right)$, which is precisely the equation describing a sinusoidal carrier. In other words, the unidimensional *time-varying* signal strength of a sinusoidal carrier can be thought of as a projection of a *fixed-length* two-dimensional rotating vector on the X-axis. This warrants a more careful study of the mathematics of rotating two-dimensional vectors.

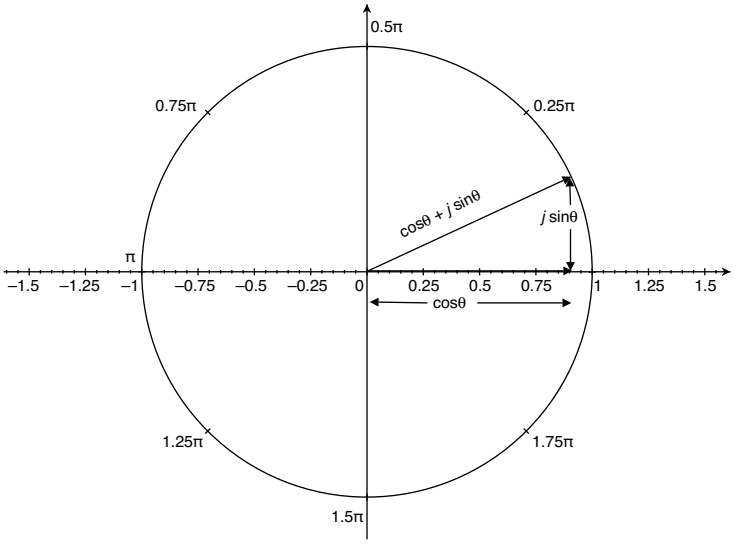
2:47
4:31

5.1.2 Complex Numbers

Lab: Phasors

Recall that any vector on a two-dimensional plane can be written as the vector sum of two *basis* vectors, one lying on the X-axis and one parallel to the Y-axis. Let us consider how to represent these basis vectors. A real value x on the number line naturally corresponds to a vector lying on the X-axis drawn from (0,0) to (0, x). Thus, the first of these basis vectors corresponds to, and can be represented by, a

Erratum 5.1: Read: from $(0,0)$ to $(x,0)$.



4:31

Figure 5.2 Generating a sinusoid from a rotating vector

real number. How can we represent the second vector? Intuitively, we need a way to "raise" a vector off the real axis. Suppose that we denote the operation "rotate a vector counterclockwise by 90 degrees about the origin" by the letter j.[1] Then, the vector $j5$, for example, corresponds to a vector of magnitude 5 lying on the Y-axis. Given this operation, we can denote a vector on the unit disk making an angle θ with the X-axis as the vector sum $\cos(\theta) + j\sin(\theta)$, as shown in Figure 5.2.

We denote the application of the rotation operator j twice in a row like this: $j.j$, or j^2. Note that the application of this combined operator to the vector 1, which is the vector from (0,0) to (0,1), results in the vector –1. We represent this by the equation

Erratum 5.2: Read: from (0, 0) to (1, 0).

$$j.j.1 = j^2.1 = -1 \qquad \textbf{(EQ 5.1)}$$

With a slight abuse of notation, we can take the square root of either side to get

$$j = \sqrt{-1} \qquad \textbf{(EQ 5.2)}$$

revealing j to be the unit imaginary vector. It is unfortunate that most students encounter j as a ghostly, ethereal entity rather than as the far more intuitive 90-degree rotation operator.

1. Mathematicians and engineers differ in the symbol used for this operation. Engineers use j rather than the mathematician's i because by universal engineering convention, the symbol i represents an electric current.

Given the basis vectors 1 and j, it is easy to see that we can represent a vector from the origin to any point (a,b) on the plane as the vector sum $a + jb$. The corresponding tuple (a,b) is called a **complex** number, with a as its **real** part and b as its **imaginary** part. If $z = a+jb$ is a complex number, we denote $a = \text{Re}(z)$; $b = \text{Im}(z)$. We denote by z^* the **complex conjugate** of z, defined as $z^* = a - jb$.

5.1.3 Euler's Formula

Given the importance of rotating vectors, also called **phasors**, it is desirable to compactly represent the current position of the vector on the unit disk. Of course, this can be easily specified by the single value θ. By tradition, and for sound mathematical reasons that we will see later, we instead represent a rotating vector of magnitude c making an angle θ with the X-axis by using the notation $ce^{j\theta}$. This notation is rather counterintuitive! What does it mean to raise e to an imaginary power? The answer is: It does not matter. All that matters is that the following formula, known as **Euler's formula**,[2] has been found to be consistent with the foundations of real analysis:


13:09

$$e^{j\theta} = \cos\theta + j\sin\theta \qquad \textbf{(EQ 5.3)}$$

EXAMPLE 5.1: EULER'S FORMULA

Use Euler's formula to represent $e^{j\frac{\pi}{2}}$ in the form $a + jb$.

Solution:

By definition, $e^{j\frac{\pi}{2}} = \cos\left(\frac{\pi}{2}\right) + j\sin\left(\frac{\pi}{2}\right) = j$.

EXAMPLE 5.2: EULER'S FORMULA

Use Euler's formula to represent $e^{-j\frac{\pi}{10}} + 2e^{-j\frac{2\pi}{10}}$ in the form $a + jb$.

---

2. The Euler formula has been called the most beautiful formula in mathematics, especially when written for the special value of $\theta = \pi$, where, using standard mathematical convention, it reduces to

$$e^{i\pi} + 1 = 0$$

relating five elementary quantities in mathematics: *e, i,* 0, 1, and π.

Solution:

$$
\begin{aligned}
& e^{-j\frac{\pi}{10}} + 2e^{-j\frac{2\pi}{10}} \\
&= \left(\cos\left(-\frac{\pi}{10}\right) + j\sin\left(-\frac{\pi}{10}\right)\right) + 2\left(\cos\left(-\frac{2\pi}{10}\right) + j\sin\left(-\frac{2\pi}{10}\right)\right) \\
&= \left(\cos\left(\frac{\pi}{10}\right) + 2\cos\left(\frac{2\pi}{10}\right)\right) - j\left(\sin\left(\frac{\pi}{10}\right) + 2\sin\left(\frac{2\pi}{10}\right)\right) \\
&= (0.95 + 1.618) - j(0.309 + 1.175) \\
&= 2.568 + j1.484
\end{aligned}
$$

Erratum 5.3: The result is $2.569 - j1.485$.

We have already seen that the projection of a phasor $ce^{j\theta}$ on the X-axis is given by $c\cos(\theta)$, which is the first, or real, component in the expansion of $ce^{j\theta}$ using Euler's formula. We denote this as $\mathrm{Re}(ce^{j\theta})$. Recalling that $\theta = \frac{2\pi t}{T}$, the strength of a carrier signal at time t is therefore given by

$$\mathrm{Re}(ce^{j\left(\frac{2\pi t}{T}\right)}).$$

The factor $\frac{2\pi}{T}$ occurs so frequently in the study of rotating vectors that it has its own symbol, ω, where $\omega = \frac{2\pi}{T}$. The symbol ω has a physical interpretation as the angular frequency of rotation and has units of radians/second. With the notation and concepts built up so far, we can succinctly denote a carrier signal as $\mathrm{Re}(ce^{j\omega t})$.

We have seen two ways of representing a complex number: as a sum $a + jb$, or as a phasor $ce^{j\theta}$. It is straightforward to go from one representation to the other, using the following formulae:

$$
\begin{aligned}
a &= c\cos\theta & \qquad c &= \sqrt{a^2 + b^2} \\
b &= c\sin\theta & \qquad \theta &= \mathrm{atan}\left(\frac{b}{a}\right)
\end{aligned}
\qquad \textbf{(EQ 5.4)}
$$

Erratum 5.4: If $a < 0$, add π (use $\mathrm{atan2}(b, a)$).

Note that we have arbitrarily described the phasor as rotation in the counterclockwise direction. A sinusoidal carrier can equally well be generated by a phasor rotating in the counterclockwise direction. To distinguish between the two, we denote the angular frequency of a phasor rotating in the clockwise direction as a negative frequency. Thus, a phasor with a frequency of -20π radians/second is rotating 10 times a second in the clockwise direction, and the second hand of a clock has a frequency of $-\frac{\pi}{30}$ radians/second.

Erratum 5.5: Read: "in the clockwise direction".

EXAMPLE 5.3: KEYING

Lab: Keying

We now consider how to mathematically represent the transmission of information on a link by modification of the phasor $ce^{j\omega t}$. This is also called **keying**. For simplicity, focus on the problem of sending just one bit: either a 0 or a 1.

In **amplitude shift keying**, we choose two phasor magnitudes c_1 and c_2. To send a 0, the transmitter transmits a signal $c_1\text{Re}(e^{j\omega t})$; to send a 1, it transmits a signal $c_2\text{Re}(e^{j\omega t})$. In practice, this is accomplished by an electronic circuit that sets the degree of amplification of the carrier signal to either c_1 or c_2. If either of the constants c_1 and c_2 is 0, this is also called **on-off keying**.

In **frequency shift keying**, we choose two phasor frequencies ω_1 and ω_2. To send a 0, the transmitter transmits a signal $c\text{Re}(e^{j\omega_1 t})$; to send a 1, it transmits a signal $c\text{Re}(e^{j\omega_2 t})$. In practice, this is accomplished by either coupling the output signal to one of two oscillators tuned to the frequencies ω_1 and ω_2 or sending control signals to a voltage-controlled oscillator.

The **phase** of a sinusoid at time t is the angle it makes with the X-axis at that time. With **phase shift keying**, the transmitter and receiver are assumed to be **phase-locked**; that is, they each have local oscillators that have identical frequencies and phases. The transmitter sends a 0 or a 1 by modifying the phase of the carrier signal. With **binary phase shift keying**, for example, the transmitter transmits the unmodified carrier signal $ce^{j\omega t}$ to send a 0 and $-ce^{j\omega t}$, which is a signal that is exactly 180 degrees out of phase with the receiver's local oscillator, to send a 1. This general idea can be easily extended. For example, with **quadrature phase shift keying**, the transmitter shifts the transmitting phasor's phase by 0, 90, 180, or 270 degrees, using the phasors

$$ce^{j\omega t},\ ce^{j\left(\omega t+\frac{\pi}{2}\right)},\ ce^{j(\omega t+\pi)},\ \text{and}\ ce^{j\left(\omega t+\frac{3\pi}{2}\right)}.$$

These phases allow the receiver to extract *two* bits for each received signal. In practice, phase shift keying is accomplished by slightly delaying the transmitted carrier signal so that it is received out of phase.

One might ask: What is the limit to phase shift keying? Would it be possible to send larger and larger numbers of bits by using more fine-grained phase shifting? Could we send, for example, 32 bits at a time by using 65,536 phases? The answer is that as we increase the number of phase levels, even a slight distortion of a signal along the path from the transmitter to the receiver corrupts the signal. Therefore, the limit to the *capacity* of the path comes from its inherent *noise*. The relationship between the capacity of a channel and its noise characteristics form the basis of *information theory*, which is discussed in Chapter 9.

Erratum 5.6: 2^{16} phases carry 16 bits; 32 bits need 2^{32}.

It is sometimes useful to express sinusoids as complex exponentials. We can do so as follows. First, recall that $\cos(-\theta) = \cos\theta$ and $\sin(-\theta) = -\sin\theta$. Now, because $e^{j\theta} = \cos\theta + j\sin\theta$, it follows that $e^{-(j\theta)} = e^{j(-\theta)} = \cos(-\theta) + j\sin(-\theta) = \cos\theta - j\sin\theta$. Therefore, $e^{j\theta} + e^{-j\theta} = 2\cos\theta$, so that

$$\cos\theta = \frac{1}{2}(e^{j\theta} + e^{-j\theta}) \quad \textbf{(EQ 5.5)}$$

Similarly,

$$\sin\theta = \frac{1}{2j}(e^{j\theta} - e^{-j\theta}) \quad \textbf{(EQ 5.6)}$$

5.1.4 Discrete-Time Convolution and the Impulse Function

Lab: Convolution

Consider the discrete-time function $x(t)$, where t is a non-negative integer, that represents the number of HTTP requests made to a Web server over intervals of 1 second. Suppose that this function, over an interval of 10 seconds, is given by $x(t)$, $0 \le t \le 9$ = 1, 3, 5, 2, 5, 8, 7, 3, 9, 4. That is, in the 0th interval, the Web server receives $x(0) = 1$ request; in the first interval, it receives $x(1) = 3$ requests; in the second interval, it receives $x(2) = 5$ requests; and so on. The histogram of this function is shown in Figure 5.3(a).

10:16

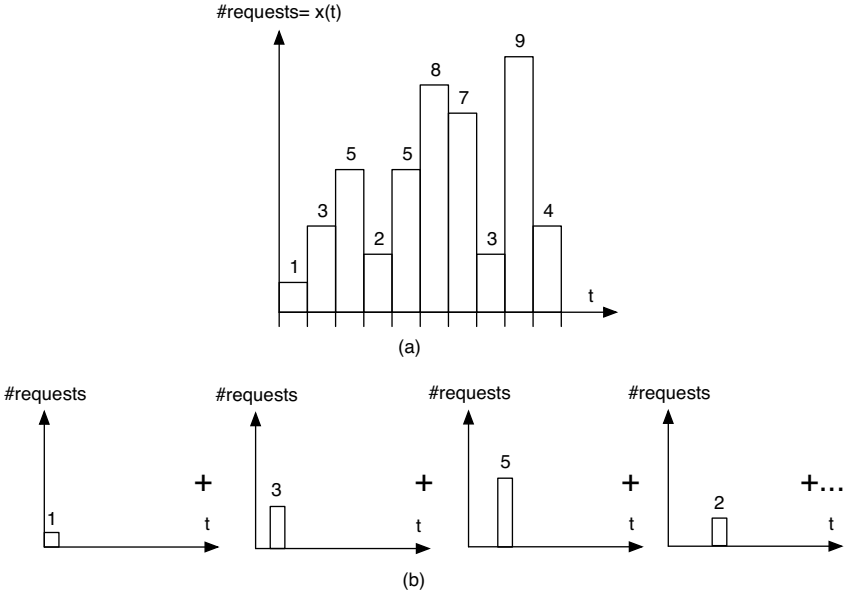


Figure 5.3 Decomposing a discrete-time function

The same histogram can also be represented as the sum of a set of smaller histograms, one per time interval, as shown in Figure 5.3(b). Each smaller histogram is the product of two functions, $x(t)$ and a selector function $sel(t, \tau)$, which is defined as $sel(t, \tau) = 1$ when $t = \tau$ and = 0 otherwise. For instance, the first small histogram corresponds to $x(t)$*$sel(t,0)$ because it is zero everywhere, except when $t = 0$, when $x(t)$ takes the value $x(0)$ and $sel(t,0)$ is 1. Similarly, the second histogram corresponds to $x(t)$*$sel(t,1)$ and so on. Because their sum is $x(t)$, we can write

$$x(t) = \sum_{\tau=0}^{\infty} x(t)sel(t, \tau) \quad \textbf{(EQ 5.7)}$$

We make a small digression to consider the relationship between the function $x(t)$ and the function $x(t - a)$. For concreteness, consider the function $x(t)$ shown in Figure 5.3(a) and let $a = 2$. The function $x(t - a)$ is undefined for times before time 2, and, for convenience, we define it to have the value 0. At time 2, $x(t - a) = x(2 - 2) = x(0) = 1$. At time 3, $x(t - a) = x(1) = 3$ and so on. We plot this in Figure 5.4.

Note that $x(t - 2)$ is just $x(t)$ shifted two steps to the right. A little thought shows that, in general, the time series $x(t - a)$ is the time series $x(t)$ shifted by a time interval a to the right.

Now, consider the function $\delta(t)$, also called the **impulse function**, defined as $\delta(0) = 1$ and $\delta(t) = 0$ for all other values of t. We see that, trivially, $sel(t, 0) = \delta(t)$. From our observation about time-shifting, we can define $sel(t, 1) = \delta(t - 1)$, which is the one-step time-shifted version of $\delta(t)$. Generalizing, we see that

$$sel(t, \tau) = \delta(t - \tau) \quad \textbf{(EQ 5.8)}$$

Therefore, we can rewrite Equation 5.7 as

$$x(t) = \sum_{\tau=0}^{\infty} x(t)\delta(t - \tau) \quad \textbf{(EQ 5.9)}$$

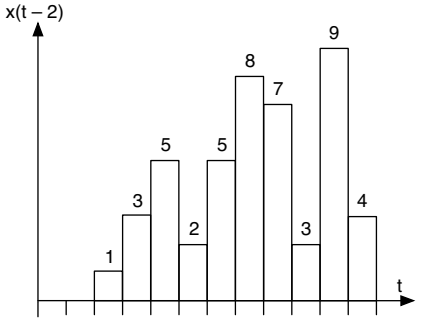


Figure 5.4 Time-shifting a time series

We have already seen that for each term in the summation, the value of the product of $x(t)$ and $\delta(t-\tau)$ is zero except when $\tau = t$, so we can also write the summation as

$$x(t) = \sum_{\tau=0}^{\infty} x(\tau)\delta(t-\tau)$$

Finally, for functions $x(t)$ that are defined over both the positive and negative values of t, we can generalize the summation over the entire number line to get

$$x(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)\delta(t-\tau) \quad \textbf{(EQ 5.10)}$$

This summation is called the **convolution** of $x(t)$ with $\delta(t)$ and is denoted $x(t) \otimes \delta(t)$.[3]

In general, the convolution of two discrete functions $x(t)$ and $y(t)$ is defined as

$$x(t) \otimes y(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)y(t-\tau) \quad \textbf{(EQ 5.11)}$$

Note that *each* value of the convolution of $x(t)$ and $y(t)$ (i.e., at time t) is the result of adding *all* product pairs $x(\tau)$ and $y(t-\tau)$. This is clearly a computationally expensive operation. Moreover, we need to be careful in dealing with values of $x(t)$ and $y(t)$ that lie outside their range of definition.

EXAMPLE 5.4: DISCRETE-TIME CONVOLUTION

Compute the convolution of $x(t)$, $0 \le t \le 9$ = 1, 3, 5, 2, 5, 8, 7, 3, 9, 4 and $y(t)$, $0 \le t \le 9$ = 3, 1, 7, 4, 5, 9, 7, 1, 3, 8.

Solution:

By convention, we assume that both functions are zero outside their range of definition. Let $z(t) = x(t) \otimes y(t)$: $z(0) = \sum_{\tau=-\infty}^{\infty} x(\tau)y(-\tau)$. The only value of τ for which both $x(t)$ and $y(t)$ are nonzero is $\tau = 0$, so the sum reduces to $x(0)*y(0) = 3$. $z(1) = \sum_{\tau=-\infty}^{\infty} x(\tau)y(1-\tau)$. The only values of τ for which both $x(t)$

3. In electrical engineering texts, convolution is represented by *. Unfortunately, this is universally used in computer science to denote multiplication, thus our choice.

and $y(t)$ are nonzero are $\tau = 0, 1$ so the sum reduces to $x(0) * y(1) + x(1) * y(0) = 1 * 1 + 3 * 3 = 10$. It is left as an exercise to the reader to show that $z(2) = 7 + 3 + 15 = 25$, $z(3) = 4 + 21 + 5 + 6 = 36$, $z(4) = 5 + 12 + 35 + 2 + 15 = 69$,....

5.1.5 Continuous-Time Convolution and the Dirac Delta Function

The definition of convolution can be extended to continuous-time functions. We start, as before, with the definition of a selector function with a parameter τ that picks out the value of $x(t)$ at time τ. Thus, we would like to define a function that is zero everywhere except at a single point, where it has a value of 1. Obviously, such a function is discontinuous at that point, which makes it mathematically troublesome. To avoid this problem, we define the selector function as the limiting case of a rectangular function of unit area defined over a support of length T and with height $1/T$ as $T \to 0$. At the limit, this represents an infinitely tall rectangle that nevertheless has a finite area! Nevertheless, this function is continuous everywhere except at the limit, so that its properties at the limit can be extrapolated from those of the limiting series. Setting aside a mathematically rigorous development, which is beyond the scope of this text, we define the continuous-time impulse function by

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$
$$\delta(t) = 0 \quad \text{for } t \neq 0 \qquad \textbf{(EQ 5.12)}$$

Intuitively, a continuous-time impulse function has a unit mass concentrated entirely at $t = 0$. It was first defined by the physicist P. A. M. Dirac and therefore is also called the **Dirac delta** in his honor. The Dirac delta is represented graphically by a vertical arrow at the origin. The integral of the delta function is called the **unit-step function** and written $u(t)$. Note that $u(t)$ is zero for $t < 0$ and 1 for $t > 0$. Its value at zero is undefined (Figure 5.5).

For any function $x(t)$ that is continuous in the neighborhood of the origin, the delta function acts as a selector, so that

$$x(t)\delta(t) = x(0)\delta(t) \qquad \textbf{(EQ 5.13)}$$

We interpret this to mean that the multiplication of $x(t)$ with the delta function results in a delta function whose strength is the value of $x(t)$ at the origin. (Compare this with the discrete-time selector function.) Similarly,

$$x(t)\delta(t-\tau) = x(\tau)\delta(t-\tau) \qquad \textbf{(EQ 5.14)}$$

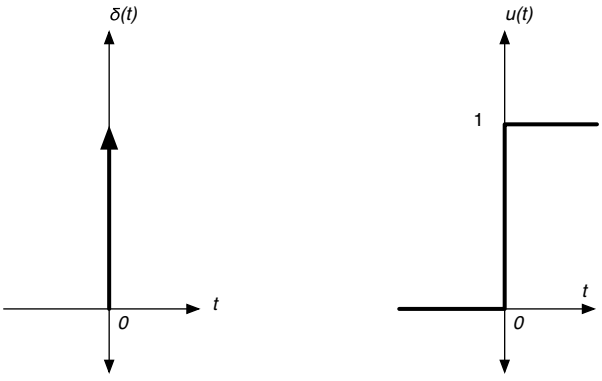


Figure 5.5 The Dirac delta and its integral, the unit-step function

Moreover, analogous to Equation 5.10, we have

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$$
$$x(\tau) = \int_{-\infty}^{\infty} x(t)\delta(t-\tau)dt \qquad \textbf{(EQ 5.15)}$$

where the second equation can be obtained by swapping the variables t and τ.

Finally, we can define the convolution of any two continuous-time functions $x(t)$ and $y(t)$ as

$$x(t) \otimes y(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau \qquad \textbf{(EQ 5.16)}$$

EXAMPLE 5.5: CONTINUOUS-TIME CONVOLUTION

Compute the convolution of the functions $x(t)$ and $y(t)$ defined graphically in Figure 5.6.

Solution:

By convention, we assume that both functions are zero outside their range of definition. Thus,

$$x(\tau) \neq 0 \text{ in the range } -1 \leq \tau \leq 1$$
$$y(t-\tau) \neq 0 \text{ in the range } -1 \leq (t-\tau) \leq 1 \Rightarrow t-1 \leq \tau \leq t+1$$

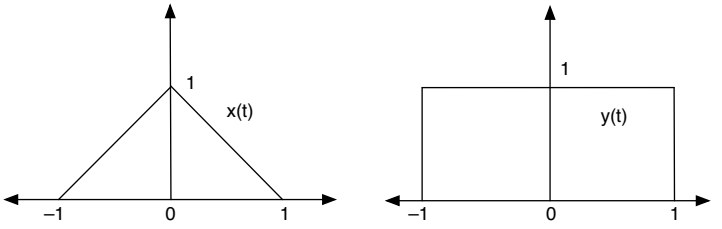


Figure 5.6 Example continuous-time functions for convolution

Therefore, both $x(t)$ and $y(t)$ are simultaneously nonzero only in the range $max(t-1,-1) \le \tau \le min(t+1,1)$, and

$$x(t) \otimes y(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau = \int_{max(t-1,-1)}^{min(t+1,1)} x(\tau)y(t-\tau)d\tau$$

It is necessary to evaluate this integral separately in the ranges $[-\infty,-2]$, $[-2,-1]$, $[-1,0]$, $[0,1]$, $[1,2]$, and $[2,\infty]$. In all the ranges, $y(t) = 1$, but the limits of integration are modified as follows:

$$x(t) \otimes y(t) = \begin{cases} 0 & -\infty \le t \le -2 \\ \int_{-1}^{(t+1)} (1+\tau)d\tau & -2 \le t \le -1 \\ \int_{0}^{(t+1)} (1-\tau)d\tau + \frac{1}{2} & -1 \le t \le 0 \\ \int_{(t-1)}^{0} (1+\tau)d\tau + \frac{1}{2} & 0 \le t \le 1 \\ \int_{t-1}^{1} (1-\tau)d\tau & 1 \le t \le 2 \\ 0 & 2 \le t \le \infty \end{cases}$$

$$x(t) \otimes y(t) = \begin{cases} 0 & -\infty \le t \le -2 \\ \frac{(t+2)^2}{2} & -2 \le t \le -1 \\ 1 - \frac{t^2}{2} & -1 \le t \le 0 \\ 1 - \frac{t^2}{2} & 0 \le t \le 1 \\ \frac{(t-2)^2}{2} & 1 \le t \le 2 \\ 0 & 2 \le t \le \infty \end{cases}$$

Note that, in this case, the convolution changes its form across the entire range. Therefore, it is a tricky task to compute it. As we will see later, the use of transforms greatly reduces the complexity of computing the convolution of two functions.

5.2 Signals

6:19

A signal is a set of data, usually a function of time. We denote a signal as $x(t)$, where t is nominally time. We have already seen two types of signals: **discrete** signals, where t takes integer values, and **continuous** signals, where t takes real values. In contrast, **digital** signals are those where the signal $x(t)$ takes on one of a quantized set of values, typically 0 and 1, and **analog** signals are those where the signal $x(t)$ takes on real values.

Some signals are **periodic**, so that their values repeat after every time period T. The smallest such period is called the **fundamental period** of the signal. Signals that do not have such a period are called **aperiodic**.

Most signals exist for only a limited period in time and are called **time-limited** signals. Others can be modeled as eternal signals and are called **time-unlimited** signals.

Given a signal $x(t)$, we define a **time-shifted** signal that is time-shifted a time units to the right (i.e., it is delayed by a time units as compared to the original signal) to be $x(t - a)$. Similarly, we define a **time-scaled** signal where the signal is expanded or compressed in time by the scalar value a to be $x(at)$.

Here are some examples of signals that arise in computer networks:

- The series of packet round trip times measured at a network source (discrete, analog, aperiodic, time-limited)
- The buffer occupancy over time at a router (continuous, digital, aperiodic, time-unlimited)
- The samples collected at an analog-to-digital transducer during a voice call (discrete, digital, aperiodic, time-limited)
- The physical carrier signal on a wired or wireless link (continuous, analog, periodic, time-unlimited)
- The number of HTTP requests received at a Web server over intervals of 1 second (discrete, digital, aperiodic, time-unlimited)
- The number of routing updates received by a border gateway protocol (BGP) daemon at a router over intervals of 5 minutes (discrete, digital, aperiodic, time-unlimited)

Digital systems, such as computer networks, approximate a continuous signal by a discrete signal whose time instants are sufficiently closely spaced and an analog signal by a digital signal whose quantization levels are sufficiently closely spaced. We will study what is precisely meant by "sufficiently closely spaced" later in this chapter. For the remainder of this chapter, rather than these specific signals, we will consider signals in the abstract.

5.2.1 The Complex Exponential Signal



10:08

Lab: Complex exponentials

A signal that frequently crops up in the study of transforms is the complex exponential signal denoted ke^{st}, where k is a real number and s is a complex quantity that can be written as $s = \sigma + j\omega$. We can expand ke^{st} as $ke^{st} = ke^{(\sigma+j\omega)t} = ke^{\sigma t}e^{j\omega t} = ke^{\sigma t}(\cos\omega t + j\sin\omega t) = ke^{\sigma t}\cos\omega t + jke^{\sigma t}\sin\omega t$. The real and imaginary components of this signal are both exponentially modulated sinusoids having frequency ω. With carefully chosen values of σ, ω, and k, this expression and its conjugate can represent a variety of seemingly unrelated signals:

- When $s = 0$, this reduces to the constant k (Figure 5.7(a)).
- When $\omega = 0$, this reduces to the real monotone exponential $ke^{\sigma t}$ (Figure 5.7(b)).
- When $\sigma = 0$, setting $s = \pm j\omega$ gives us the sum

$$k((\cos\omega t + j\sin\omega t) + (\cos\omega t - j\sin\omega t)) = 2k\cos\omega t,$$

  which is a real sinusoid scaled by a factor k (Figure 5.7(c)).

Erratum 5.7: Read "by a factor $2k$".

- When $\sigma \neq 0$, setting $s = \sigma \pm j\omega$ gives us the sum $2ke^{\sigma t}\cos\omega t$, which is a scaled real sinusoid that is modulated by a monotone exponential (Figure 5.7(d)).
- When $\sigma = 0$, the expression reduces to $k(\cos\omega t + j\sin\omega t)$, which is a helix oriented along the t axis, whose projections on the real-t and imaginary-t planes are sinusoids (Figure 5.8(a)).
- In the general case, the function represents an exponentially modulated helix oriented along the t axis, whose projections on the real-t and imaginary-t planes are exponentially modulated sinusoids (Figure 5.8(b)).

It is worth studying these figures carefully because they provide deep insight into the nature of a complex exponential that will greatly help in understanding the nature of transforms.

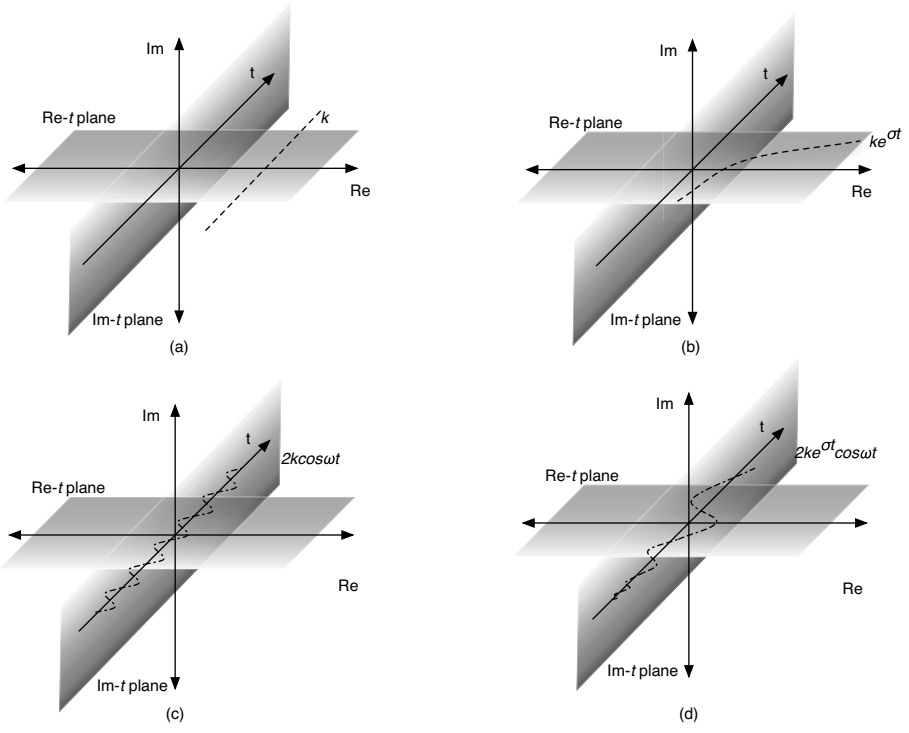


Figure 5.7 Elementary cases of the complex exponential signal

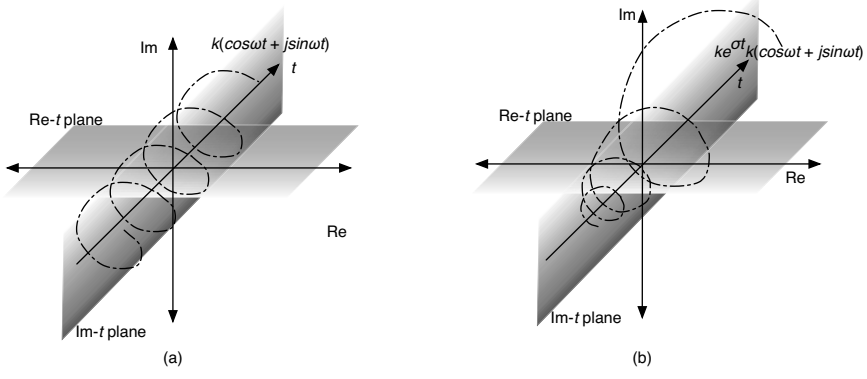


Figure 5.8 More general cases of the complex exponential signal

5.3 Systems

4:55

Lab: System properties

A system is an entity that acts on a set of input signals to produce a set of output signals. Mathematically speaking, a system is a function whose domain and ranges are both sets of functions. We typically denote the input to the system by the vector function $\boldsymbol{x}(t)$ and its output by the vector function $\boldsymbol{y}(t)$. The **transfer function** $\boldsymbol{H}(.)$ converts the input to the output, so that $\boldsymbol{y}(t) = \boldsymbol{H}(\boldsymbol{x}(t))$.[4]

When studying a system, we have one of two goals in mind: (1) to determine the output from an existing system when it receives a specific input (the **analysis** problem) or (2) to design a system that exhibits a desired transformation from a set of input signals to a set of output signals (the **design** problem). In both cases, we use transforms as an essential analytical and design tool. This chapter focuses on the tools of analysis; Chapter 8, on control theory, focuses on design.

As with signals, systems can be categorized in several ways: **continuous time** and **discrete time**, and **analog** and **digital**. Their definitions mirror those for signals. We discuss four categories in more detail.

Causal systems act only on inputs from the past. That is, they cannot know the future. In contrast, **acausal** systems are aware of the future. Although acausal systems sound improbable, they are rather easy to build if they operate on logs or traces of inputs rather than on inputs presented to the system in real time.

10:38

Memoryless systems act only on the inputs presented at any point in time. By contrast, the output of a **dynamic** system may depend on both current and past inputs.

A **time-invariant** system is one whose parameters do not change with time. So, if the system has an output $\boldsymbol{y}(t)$ at time t for an input $\boldsymbol{x}(t)$, it has the output $\boldsymbol{y}(t - T)$ for an input $\boldsymbol{x}(t - T)$. That is, if $\boldsymbol{y}(t) = \boldsymbol{H}(\boldsymbol{x}(t))$, then $\boldsymbol{y}(t - T) = \boldsymbol{H}(\boldsymbol{x}(t - T))$.

Finally, and most important, a **linear** system is one that exhibits the properties of **additivity** and **homogeneity**.

- A system is **additive** if, given that input $\boldsymbol{x}_1$ leads to output $\boldsymbol{y}_1$ and input $\boldsymbol{x}_2$ leads to output $\boldsymbol{y}_2$, the input $(\boldsymbol{x}_1 + \boldsymbol{x}_2)$ leads to the output $(\boldsymbol{y}_1 + \boldsymbol{y}_2)$. That is, if $\boldsymbol{y}_1 = \boldsymbol{H}(\boldsymbol{x}_1)$ and $\boldsymbol{y}_2 = \boldsymbol{H}(\boldsymbol{x}_2)$, then $\boldsymbol{y}_1 + \boldsymbol{y}_2 = \boldsymbol{H}(\boldsymbol{x}_1 + \boldsymbol{x}_2)$.
- A system is **homogeneous (scalar)** if, given that the input $\boldsymbol{x}$ leads to the output $\boldsymbol{y}$, the input $k\boldsymbol{x}$, where k is a scalar, leads to the output $k\boldsymbol{y}$. That is, if $\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{x})$, then $k\boldsymbol{y} = \boldsymbol{H}(k\boldsymbol{x})$.

These two conditions can be combined into the single condition of **superposition**. A system exhibits superposition if, given that the input $\boldsymbol{x}_1$ leads to output $\boldsymbol{y}_1$

4. We are using the term *transfer function* loosely. As discussed in Section 8.2.2, in the context of control theory, the transfer function is more precisely described as the Laplace transform of the impulse response of the system. At this stage, however, this loose but intuitively appealing description of the transfer function suffices.

Erratum 5.8: Linearization is not covered in these notes.

and input $\boldsymbol{x}_2$ leads to output $\boldsymbol{y}_2$, then for all constants k_1 and k_2, the input $(k_1\boldsymbol{x}_1 + k_2\boldsymbol{x}_2)$ leads to the output $(k_1\boldsymbol{y}_1 + k_2\boldsymbol{y}_2)$. That is, if $\boldsymbol{y}_1 = \boldsymbol{H}(\boldsymbol{x}_1)$ and $\boldsymbol{y}_2 = \boldsymbol{H}(\boldsymbol{x}_2)$, then $k_1\boldsymbol{y}_1 + k_2\boldsymbol{y}_2 = \boldsymbol{H}(k_1\boldsymbol{x}_1 + k_2\boldsymbol{x}_2)$.

EXAMPLE 5.6: LINEAR SYSTEM

Is the system defined by the transfer function $H(x) = x^2$ linear?

Solution:

$H(k_1x_1 + k_2x_2) = (k_1x_1 + k_2x_2)^2$, which is not the same as $k_1y_1 + k_2y_2 = k_1^2x_1^2 + k_2^2x_2^2$. Hence, the system is not linear.

Note that any system that limits the output to some maximum or minimum is not linear, because such a system is not homogeneous. But in practice, outputs are almost always bounded—that is, they *saturate* at some limiting value—due to physical limitations. Therefore, most real systems are not linear. However, it is often possible to model the system behavior, around its operating point, as being approximately linear. Specifically, we can use the first-order Taylor expansion of the transfer function in the region of the operating point to approximate the behavior of the system. This technique, called **linearization**, is discussed in Section 8.2.2.

5.4 Analysis of a Linear Time-Invariant System

11:46

Lab: LTI systems

The class of linear time-invariant (**LTI**) systems is important both because it is relatively easy to analyze and because many common systems can be approximated by LTI systems. Here, we study three aspects of an LTI system: the effect of an LTI system on a complex exponential input, the output of an LTI system with a zero input, and the output of an LTI system for an arbitrary input. We will revisit the third topic later to demonstrate the power of using transform domain techniques in analysis.

Note that the discussion here is for continuous time systems. The analysis of discrete time systems is nearly identical, and we defer this discussion to our study of control theory in Chapter 8.

5.4.1 The Effect of an LTI System on a Complex Exponential Input

Consider an LTI system that is described by

$$\boldsymbol{y}(t) = \boldsymbol{H}(\boldsymbol{x}(t))$$

where $\boldsymbol{y}$ and $\boldsymbol{x}$ are, in general, vector functions of time, and $\boldsymbol{H}$ is the transfer function. For simplicity of exposition, in the remainder of this discussion, we will assume that both x and y are scalar functions of time and so will no longer use boldface to represent them. Suppose that the input is a complex exponential function of time, so that $x(t) = ke^{st}$, where s is complex and k is real. Then,

$$y(t) = H(ke^{st}) \quad \textbf{(EQ 5.17)}$$

Now, consider the input $(e^{s\tau})(ke^{st})$, where τ is independent of t. Because the system is LTI and $e^{s\tau}$ is a scalar with respect to t, the corresponding output is $(e^{s\tau})y(t)$. But note that the input $(e^{s\tau})(ke^{st})$ can be rewritten as $ke^{s(t+\tau)}$, which we recognize as a time-shifted input (i.e., t is replaced by $t+\tau$). Therefore, the corresponding output, because of time invariance, must be $y(t+\tau)$. This gives us the relationship

$$y(t+\tau) = e^{s\tau}y(t) \quad \textbf{(EQ 5.18)}$$

This must hold for all t, so it is true for $t = 0$, where

$$y(\tau) = e^{s\tau}y(0) \quad \textbf{(EQ 5.19)}$$

Differentiating both sides with respect to τ, we get

$$\frac{dy(\tau)}{d\tau} = sy(0)e^{s\tau} = sy(\tau) \quad \textbf{(EQ 5.20)}$$

where, for the last step, we invoke Equation 5.19. Using standard techniques for the solution of ordinary differential equations, it is easy to show that the unique solution to this differential equation is given by

$$y(t) = f(s)e^{st} \quad \textbf{(EQ 5.21)}$$

where $f(s)$ is an arbitrary function independent of t. This is an important result! It shows that for *any* LTI system, if the input is a complex exponential, so too is the output. It is impressive that we can characterize all LTI systems so simply: After all, we have absolutely no idea what the system looks like on the inside, yet we are able to state how it will respond to a complex exponential input.

We can gain additional insight into the nature of this phenomenon by comparing Equations 5.21 and 5.17. We see that for the special input $x(t) = ke^{st}$, the effect of the LTI system is to act as a scalar multiplier, with the multiplication factor being $f(s)/k$. Therefore, $x(t) = ke^{st}$ is an **eigenfunction** of an LTI system: that is, an input function such that the effect of the system is simply to scale it. The corresponding **eigenvalue** is $f(s)/k$.

We can summarize this discussion by stating that the input $x(t) = ke^{st}$ is an eigenfunction of any LTI system for any value of s. Therefore, if we can represent an arbitrary input as the sum of such eigenfunctions, the corresponding output will be the

sum of the scaled inputs. From this perspective, the transforms we are about to study can be viewed as a way to represent an arbitrary input as the sum of complex exponentials.

5.4.2 The Output of an LTI System with a Zero Input

11:48

A wide range of LTI systems can be described in the form of a **linear differential equation**, whose general form is

$$\frac{d^N y(t)}{dt^N} + a_1 \frac{d^{N-1} y(t)}{dt^{N-1}} + \ldots + a_{N-1} \frac{dy(t)}{dt} + a_N y(t) = b_{N-M} \frac{d^M x(t)}{dt^M} + b_{N-M+1} \frac{d^{M-1} x(t)}{dt^{M-1}} + \ldots + b_{N-1} \frac{dx(t)}{dt} + b_N x(t) \quad \text{(EQ 5.22)}$$

In nearly all practical systems, $M > N$. Here, we study the output of this class of LTI system when its input $x(t)$ is zero, so that the right-hand side is zero. Using the operator D to denote differentiation, with D^N representing $\frac{d^N}{dt^N}$, we can rewrite the equation, for zero input, as

Erratum 5.9: Read: $M \leq N$.

$$(D^N + a_1 D^{N-1} + \ldots + a_{N-1} D + a_N) y(t) = 0$$

where the polynomial in D on the left-hand side is called the **characteristic polynomial** of the system. Note that this characteristic polynomial is of degree N. Therefore, from the fundamental theorem of algebra, it can be factorized as

$$((D - \lambda_1)(D - \lambda_2) \ldots (D - \lambda_N)) y(t) = 0$$

where the λ_i (in general complex quantities) are the roots of the polynomial, and some of the roots may be repeated. Assume for the moment that the roots are distinct. In this case, each solution of this equation (that is, a value of $y(t)$ for which the equation is true) is given by

$$(D - \lambda_i) y(t) = 0$$

which we can expand as

$$\frac{dy(t)}{dt} = \lambda_i y(t)$$

Comparing this with Equation 5.20, we see that the solution of this equation is given by

$$y(t) = c_i e^{\lambda_i t}$$

where c is a constant independent of t but may be a function of λ_i. Because each root of the equation generates a solution of this form, and the system is linear, by superposition the general solution (assuming distinct roots) is given by their sum:

$$y(t) = \sum_{i=1}^{N} c_i e^{\lambda_i t} \qquad \textbf{(EQ 5.23)}$$

We interpret this as follows: If a linear differential LTI system, irrespective of its internal organization, receives no input and if the roots of its characteristic polynomial are distinct, its output can be expressed as the sum of complex exponentials. Recall that any complex exponential has an intrinsic frequency. Each root of Equation 5.23 corresponds to a **natural frequency** of the LTI system: a frequency at which the system vibrates given zero input. The solution in Equation 5.23 is therefore also called the **natural response** of the system. If a system is subjected to an input that then ceases, such as a burst of packets entering a router's buffer, the subsequent response of the system, because there is no additional input, will be its natural response. The natural response is a combination of its natural frequencies, which are intrinsic to the system.

A system that is given an input at a frequency that coincides with one of its natural frequencies will *resonate*, amplifying the input. This important phenomenon is widely employed in practical systems. For example, the basis of radio reception is to tune the resonant frequency[5] of an electronic circuit so that it selectively amplifies signals corresponding to a desired transmitter, ignoring the rest of the radio spectrum.

EXAMPLE 5.7: REAL NATURAL RESPONSE OF AN LTI SYSTEM

Compute the natural response of the LTI system given by

$$\frac{d^2y(t)}{dt^2} + \frac{5dy(t)}{dt} + 6y(t) = 8x(t).$$

Solution:

The natural response is given by the differential equation $(D^2 + 5D + 6)y(t) = 0$, which can be factored as $((D+3)(D+2))y(t) = 0$. Thus, the natural response is given by $c_1e^{-3t} + c_2e^{-2t}$, where the two constants can be determined from the initial conditions $y(0)$ and $\dot{y}(0)$. The two phasors corresponding to this solution have no complex component, and therefore both correspond to a natural frequency of 0 (i.e., do not have an oscillatory behavior).

---

5. Strictly speaking, we tune the receiver to a range centered on the resonant frequency.

EXAMPLE 5.8: COMPLEX NATURAL RESPONSE OF AN LTI SYSTEM

Compute the natural response of the LTI system given by $\frac{d^2y(t)}{dt^2} + y(t) = 8x(t)$.

Solution:

The natural response is given by the differential equation $(D^2+1)y(t) = 0$, which can be factored as $((D+j)(D-j))y(t) = 0$. Thus, the natural response is given by $c_1e^{-jt} + c_2e^{jt}$, where the two constants can be determined from the initial conditions $y(0)$ and $\dot{y}(0)$. To get more insight into this solution, we use Euler's equation to expand this solution as

$$\begin{aligned} & c_1(\cos(-t) + j\sin(-t)) + c_2(\cos(t) + j\sin(t)) \\ &= c_1(\cos(t) - j\sin(t)) + c_2(\cos(t) + j\sin(t)) \\ &= (c_1 + c_2)\cos(t) + j(c_2 - c_1)\sin(t) \end{aligned}$$

For this solution to be real, it is necessary that $(c_1 + c_2)$ have no imaginary component and that $(c_2 - c_1)$ have no real component. This is possible only if $c_1 = a + jb$ and $c_2 = a - jb$, where a and b are real numbers. Recall that pairs of numbers of this form are called **complex conjugate pairs**. In general, for practical systems, the constants associated with the natural response come in complex conjugate pairs.

Continuing with our solution, we therefore substitute $a \pm jb$ for the constants to get the natural response as $2a\cos(t) + 2b\sin(t)$. We recognize this as an *oscillatory* response, with an angular frequency of 1 radian/second. Interestingly, this natural response does not decay: The system oscillates indefinitely at this frequency of its own accord!

We have thus far assumed that the roots of the characteristic equation are distinct. This need not necessarily be the case. For example, if the characteristic equation is $D^2 + 2D + 1$, the roots are –1 and –1, which are repeated. In the case of repeated roots, the form of the solution is slightly different. For a root λ that is repeated r times, it can be shown that the corresponding solution is given by

$$y(t) = (c_1e^{\lambda t} + c_2te^{\lambda t} + \ldots + c_rt^{r-1}e^{\lambda t}) \qquad \textbf{(EQ 5.24)}$$

5.4.3 The Output of an LTI System for an Arbitrary Input



9:28

We now study how an LTI system responds to a nonzero input. We use the notation $x(t) \rightarrow y(t)$ to denote that if the input to a system is $x(t)$, its output is $y(t)$. Suppose

that we are given $h(t)$, the **impulse response** of the system. This is the response to the system when the input is the Dirac delta $\delta(t)$. Using our notation, we write

$$\delta(t) \rightarrow h(t) \qquad \textbf{(EQ 5.25)}$$

What does the impulse response look like? By definition, because it is a delta function, the input ceases immediately after time 0. Therefore, the response of the system is its natural response (other than at time 0 itself). We have already seen how to compute this response.

Recall from Equation 5.13 that multiplying $\delta(t)$ by $x(t)$ results in scaling the impulse by the value $x(0)$; that is, $x(t)\delta(t) = x(0)\delta(t)$. Because the system is linear, the response of the system to the scaled impulse $x(0)\delta(t)$ will be the scaled output $x(0)h(t)$, so that

$$x(t)\delta(t) \rightarrow x(0)h(t). \qquad \textbf{(EQ 5.26)}$$

Recall from Equation 5.14 that if we time-shift the impulse, we select a different "slice" of $x(t)$. Specifically, $x(t)\delta(t-\tau) = x(\tau)\delta(t-\tau)$. Because the system is both linear and time-invariant, the response to this time-shifted scaled impulse $x(\tau)\delta(t-\tau)$ will be the scaled and time-shifted impulse response $x(\tau)h(t-\tau)$, so that

$$x(\tau)\delta(t-\tau) \rightarrow x(\tau)h(t-\tau). \qquad \textbf{(EQ 5.27)}$$

Finally, Equation 5.15 tells us that we can assemble $x(t)$ by integrating these small "slices" together, so that $x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$. Clearly, this will result in the integration of the corresponding responses together as follows:

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau \rightarrow \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = x(t) \otimes h(t) \qquad \textbf{(EQ 5.28)}$$

This important result tells us that if we know the impulse response $h(t)$ of an LTI system, we can compute its response to *any* input $x(t)$ by convolving the input signal with the impulse response. Therefore, it is important to be able to compute the convolution of any two functions. This, however, is a difficult and complicated operation, as we saw in Section 5.1.5. An important outcome of transform domain techniques is to convert the difficult convolution operation to a simple multiplication.

5.4.4 Stability of an LTI System

We briefly consider the stability of an LTI system. This topic is covered in more detail in Section 8.8. Intuitively, a system is stable if, when all inputs are removed,

the output either eventually dies down or, at most, oscillates with a bounded amplitude. Here, we study how to characterize the stability of an LTI system.

We have seen that the behavior of an LTI system is described by its natural frequencies, which are expressed as complex exponentials. In Section 5.2.1, we saw that a complex exponential can show only one of three behaviors over time: grow without bound, decay to zero, or oscillate with a constant amplitude and frequency. Therefore, an LTI system can be characterized as stable, oscillatory (or marginally stable), or unstable.

Consider the complex exponential $ke^{(\sigma+j\omega)t}$, which represents a root: a solution to an LTI system's characteristic polynomial. If $\sigma > 0$, this exponential grows without bound. On the other hand, if $\sigma < 0$, the exponential decays to zero. Finally, if $\sigma = 0$, the exponential reduces to a pure sinusoidal oscillation.

Given this observation, we can now easily characterize the stability of an LTI system. If *all* its roots have a value of $\sigma < 0$, the system is **asymptotically stable**. If even one root has a value of $\sigma > 0$, it is **unstable**. Finally, if all the values of σ are 0, the behavior of the system depends on whether there are repeated roots. If there are no repeated roots, the system is **purely oscillatory**. On the other hand, if there are repeated roots, the system is unstable.

Erratum 5.10: See correction 5.10.

This observation allows us great insight into the design of stable LTI systems: We must ensure that the system is such that all the roots of the characteristic polynomial (also called the **poles** of the system) lie in the left half of the complex plane. This lies at the heart of control theory and we will consider it in more detail in Chapter 8.

5.5 Transforms

We are finally ready to embark on a study of transforms. Before plunging in, it is worth recapitulating why transforms are important. Transforms allow us to achieve three goals.

1. The convolution of two functions, which arises in the computation of the output of an LTI system, can be computed more easily by transforming the two functions, multiplying the transformed functions, and then computing the inverse transform.
2. Transforms convert a linear differential equation into a simpler algebraic equation.
3. They give insight into the natural response of a system: the natural frequencies at which it oscillates. This allows us to quickly determine whether there are special frequencies at which the system becomes unstable, whether there are special frequencies that cause the system output to become zero, and the frequencies at which most of the output magnitude can be found.

Table 5.1 Summary of Transforms Described in This Chapter

Transform	Signal Type	Transform Description
Fourier series	Continuous, periodic	Sum of sinusoids
Fourier	Continuous, periodic or aperiodic	Integral of sinusoids
Discrete-time Fourier	Discrete, periodic or aperiodic	Sum of sinusoids
Laplace	Continuous, periodic or aperiodic	Integral of complex exponentials
Z	Discrete, periodic or aperiodic	Sum of complex exponentials

We will consider five transforms: the Fourier series, the Fourier transform, the discrete Fourier transform, the Laplace transform, and the Z transform. All the transforms share some common properties. For now, we observe that a transform converts a function of time into either an infinite sum or an integral of exponentials of a complex quantity. The transforms differ in whether the complex quantity is a pure sinusoid or an exponentially modulated sinusoid and whether the input signal is continuous or discrete, as shown in Table 5.1. Although this may appear confusing at first glance, these concepts should become clear by the end of this chapter.

5.6 The Fourier Series

8:29
Lab: Fourier series

We first study **eternal periodic** signals, that is, signals that last for an infinitely long duration and whose values repeat identically after every period of duration T_0. All signals are, of course, time-limited, so this signal cannot be achieved in practice. However, they are still worth studying because the insight gained from the Fourier series allows us to define the Fourier transform for aperiodic signals.

Fourier showed that nearly every periodic signal $x(t)$, other than some pathological signals, can be represented at nearly every value of t as an infinite sum of sinusoids, as follows:

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t) \quad \textbf{(EQ 5.29)}$$

where the **fundamental angular frequency** ω_0 is given by

$$\omega_0 = \frac{2\pi}{T_0} \quad \textbf{(EQ 5.30)}$$

and the constants a_0, a_k, and b_k are real numbers uniquely determined by $x(t)$. This remarkable equation shows that *any* periodic function, not just a function that looks sinusoidal, can be represented as the sum of sinusoids. It is as fundamental as the observation that any natural number can be denoted using only the ten symbols 0–9. More formally, using the appropriate vector space, we can show that the set of pairs of sinusoids in a Fourier series form an infinite orthogonal basis set, so that every vector—corresponding to an eternal periodic function—can be represented as a linear combination of this basis.

Each pair of terms in the series has a frequency that is a multiple of the fundamental frequency and is therefore called a **harmonic** of that frequency. Once the potentially infinite set of values associated with the constants a_0, a_k, and b_k is known, the function itself is completely specified and can be synthesized from them alone. The accuracy of representation quickly improves with the number of terms.[6]

Note that the constant a_0 can be viewed as a degenerate sinusoid with zero frequency. It is called the **DC** component of the signal by analogy to a direct-current electrical system that, unlike an alternating-current, or AC, system, does not have a sinusoidally oscillating voltage or current.

A graphical interpretation of a Fourier series may add some additional insight. Recall from Section 5.1.1 that a sinusoid can be thought of as a being generated by a rotating phasor. We see that each harmonic in the Fourier series corresponds to two rotating phasors that are offset by 90 degrees and with a common rotational frequency of $k\omega_0$ with magnitudes of a_k and b_k, respectively. The sinusoid generated by these phasors add up in just the right way to form $x(t)$.

Erratum 5.11: Read "as being generated".

So far, we have studied only the form of the Fourier series. We do, of course, need a way to determine the constants corresponding to each term in a Fourier series. It can be shown that these are given by the following equations:

$$\begin{aligned} a_0 &= \frac{1}{T_0}\int_0^{T_0} x(t)dt \\ a_k &= \frac{2}{T_0}\int_0^{T_0} x(t)\cos k\omega_0 t dt \\ b_k &= \frac{2}{T_0}\int_0^{T_0} x(t)\sin k\omega_0 t dt \end{aligned} \qquad \textbf{(EQ 5.31)}$$

where the integral is taken over any period of length T_0 (because they are all the same).

6. However, note that due to the **Gibb's phenomenon**, the Fourier series can have an error of about 10%, even with many tens of terms, when representing a sharply changing signal, such as a square wave.

Erratum 5.12: Read: "Gibbs phenomenon".

The form of the Fourier series presented so far is called the **sinusoidal form**. These sinusoids can also be expressed as complex exponentials, as we show next. Note that the kth term of the Fourier series is

$$x_k(t) = a_k \cos k\omega_0 t + b_k \sin k\omega_0 t$$


20:19

Using Equations 5.5 and 5.6, we can rewrite this as

$$x_k(t) = \frac{a_k}{2}(e^{jk\omega_0 t} + e^{-jk\omega_0 t}) + \frac{b_k}{2j}(e^{jk\omega_0 t} - e^{-jk\omega_0 t})$$

Collecting like terms, and defining $c_0 = a_0$, $c_k = \frac{1}{2}(a_k - jb_k)$, $c_{-k} = \frac{1}{2}(a_k + jb_k)$, $k>0$, we can rewrite this as

$$x(t) = c_0 + \sum_{k=1}^{\infty} c_k e^{jk\omega_0 t} + \sum_{k=1}^{\infty} c_{-k} e^{j(-k)\omega_0 t}$$

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \qquad \textbf{(EQ 5.32)}$$

This compact notation shows that we can express $x(t)$ as an infinite sum of complex exponentials. Specifically, it can be viewed as a sum of an infinite number of phasors, each with a real magnitude c_k and a rotational frequency of $k\omega_0$. It can be shown that the constants c_k can be found, if we are given $x(t)$, by the relation

Erratum 5.13: c_k is complex; see correction 5.13.

$$c_k = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jk\omega_0 t}dt \qquad \textbf{(EQ 5.33)}$$

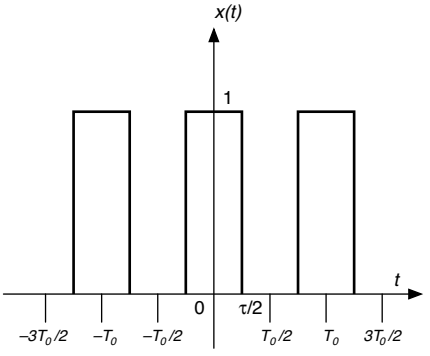
EXAMPLE 5.9: FOURIER SERIES

Find the Fourier series corresponding to the series of rectangular pulses shown in Figure 5.9.

Solution:

The kth coefficient of the Fourier series corresponding to this function is given by

$$c_k = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jk\omega_0 t}dt\,.$$



1:52

Figure 5.9 An infinite series of rectangular pulses

Instead of choosing the limits from 0 to T_0, we will choose the limits from $-T_0/2$ to $T_0/2$ because it is symmetric about 0. Note that in this range, the function is 1 in the range $[-\tau/2, \tau/2]$ and 0 elsewhere. Therefore, the integral reduces to

$$c_k = \frac{1}{T_0}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-jk\omega_0 t}dt = -\frac{1}{jk\omega_0 T_0}e^{-jk\omega_0 t}\Big|_{-\frac{\tau}{2}}^{\tau/2} = \frac{1}{jk\omega_0 T_0}\left(e^{jk\omega_0\frac{\tau}{2}} - e^{-jk\omega_0\frac{\tau}{2}}\right) \quad \textbf{(EQ 5.34)}$$

Using Equation 5.6 and multiplying the numerator and denominator by $\tau/2$, we can rewrite this as

$$c_k = \frac{\tau}{T_0}\left(\frac{\sin\left(\frac{k\omega_0\tau}{2}\right)}{\frac{k\omega_0\tau}{2}}\right) \quad \textbf{(EQ 5.35)}$$

Note that the coefficients c_k are real functions of τ (not t), which is a parameter of the input signal. For a given input signal, we can treat τ as a constant. Observing that $T_0 = \frac{2\pi}{\omega_0}$, the coefficients can be obtained as the values taken by a continuous function of ω, defined as

$$X(\omega) = \frac{\tau\omega_0}{2\pi}\frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}} \quad \textbf{(EQ 5.36)}$$

for the values $\omega = k\omega_0$. That is, $c_k = X(k\omega_0)$. Thus, c_k is a *discrete* function of ω defined at the discrete frequency values $\omega = k\omega_0$, as shown in Figure 5.10.

The function $sin(x)/x$ arises frequently in the study of transforms and is called the **sinc** function. It is a sine wave whose value at 0 is 1 and elsewhere is the sine function linearly modulated by distance from the *Y*-axis. This function forms the envelope of the Fourier coefficients, as shown in Figure 5.10. Note that the function is zero (i.e., has zero-crossings) when a sine function is zero, that is, when $\omega\tau/2 = m\pi$, $m \neq 0$, or $\omega = 2m\pi/\tau$, $m \neq 0$.

Erratum 5.14: See correction 5.14.

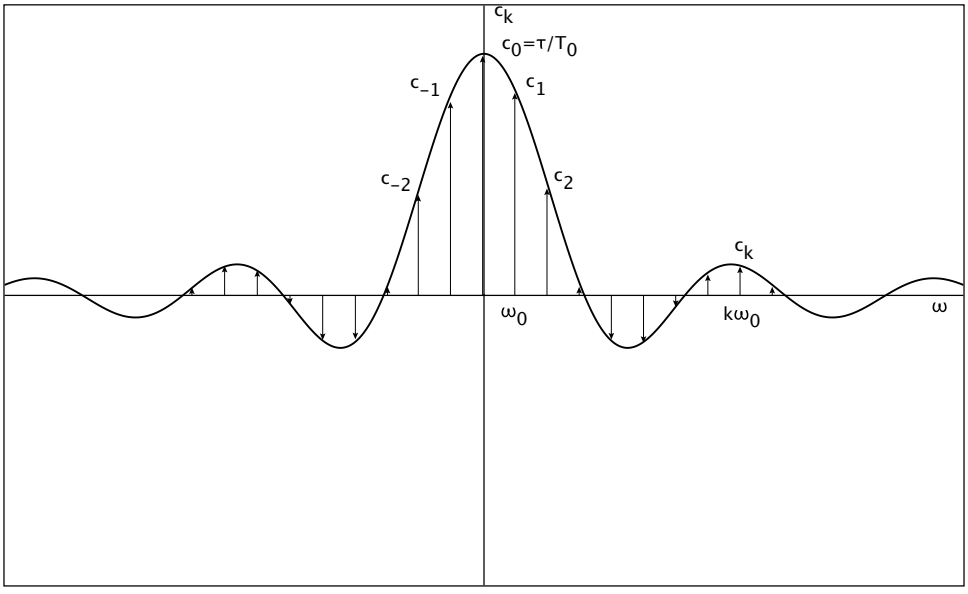


Figure 5.10 The sinc function as a function of ω

5.7 The Fourier Transform and Its Properties

10:55

Lab: Fourier transform

A Fourier series can represent any *eternal* periodic function as an infinite sum of real sinusoids. However, it cannot be used to represent aperiodic functions. These are represented, instead, using a Fourier transform.

To motivate the Fourier transform, consider the behavior of the pulse series of Example 5.9 when increasing the interpulse spacing T_0 while keeping the pulse width τ constant. Recall that for this signal, the kth Fourier series coefficient c_k is given by

$$X(\omega) = \frac{\tau}{T_0}\frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}} \text{ for } \omega = k\omega_0 \text{ where } \omega_0 = \frac{2\pi}{T_0}.$$

As T_0 increases, ω_0 decreases, which brings the coefficients under the sinc curve, which are spaced ω_0 apart, closer together (see Figure 5.10). In the limit, as $T_0 \to \infty$, the periodic pulse train becomes a single pulse, and the coefficients are spaced infinitely closely together, converging with the sinc curve. Thus, in the limit, the sinc curve corresponds to the Fourier series representing a single pulse and is called its **Fourier transform**.

Arguing along the same lines, we can analyse the limiting behavior of the generic kth Fourier series coefficient

$$c_k = \frac{\omega_0}{2\pi}\int_0^{\frac{2\pi}{\omega_0}} x(t)e^{-jk\omega_0 t}dt$$

as $\omega_0 \to 0$ to define the Fourier transform of a signal $x(t)$ as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad \textbf{(EQ 5.37)}$$

This transforms a signal (a function of time) to a function of the complex quantity $j\omega$. The value of the transform for a specific value of $\omega = w$ can be viewed as the amount of signal energy at the frequency w. The magnitude at $\omega = 0$, for example, is the DC component of the signal. We represent the fact that $X(j\omega)$ is the transform of the signal $x(t)$ as

$$x(t) \overset{\mathcal{F}}{\leftrightarrow} X(j\omega) \qquad \textbf{(EQ 5.38)}$$

The double-headed arrow indicates that $x(t)$ can be recovered from its transform, $X(jw)$, using the *inverse Fourier transform* discussed later.

Erratum 5.15: Read $X(j\omega)$.

EXAMPLE 5.10: FOURIER TRANSFORM

Compute the Fourier transform of a single rectangular pulse of height 1 and width τ centered on the origin (Figure 5.11). This signal is also denoted $rect(t/\tau)$.

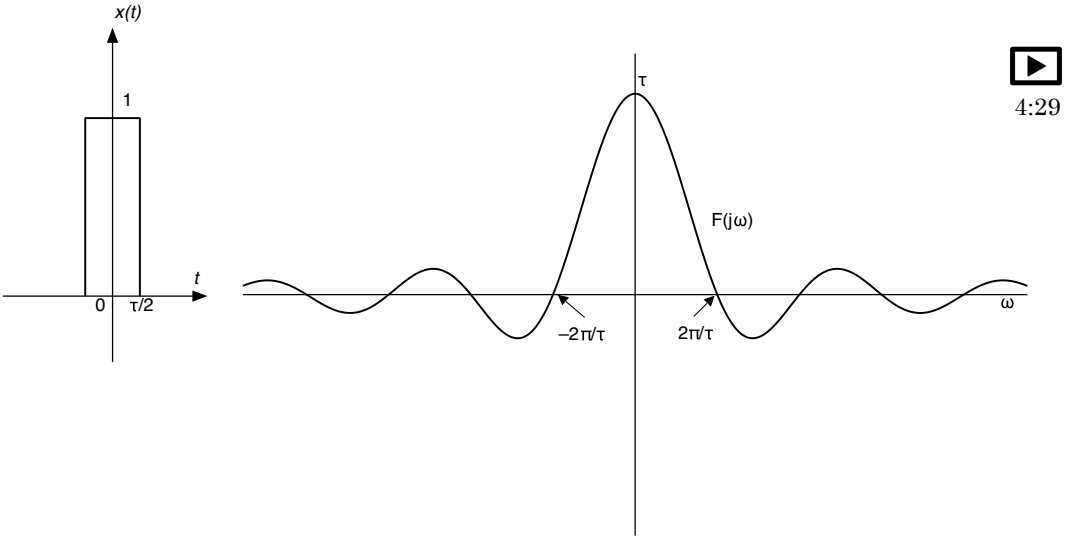


Figure 5.11 A single pulse and its Fourier transform

Erratum 5.16: Read $X(j\omega)$ for $F(j\omega)$.

Solution:

The Fourier transform of this signal $x(t)$ is given by

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt\,.$$

The function is zero outside the interval $[-\tau/2, \tau/2]$ and 1 inside this interval. Therefore, the function reduces to

$$X(j\omega) = \int_{\frac{-\tau}{2}}^{\frac{\tau}{2}} e^{-j\omega t}dt = \frac{-e^{-j\omega t}}{jw}\Big|_{\frac{-\tau}{2}}^{\tau/2} = \frac{e^{j\omega\frac{\tau}{2}} - e^{-j\omega\frac{\tau}{2}}}{j\omega}$$

Erratum 5.17: Read $j\omega$.

Multiplying the numerator and denominator by $\frac{\tau}{2}$ as before, and using Euler's formula, we can rewrite this as

$$X(j\omega) = \tau\frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}} = \tau\text{sinc}\left(\frac{\omega\tau}{2}\right) \qquad \textbf{(EQ 5.39)}$$

Therefore, the Fourier transform of $rect\left(\frac{t}{\tau}\right)$ is a sinc function that has a peak of height τ at the origin and zero crossings when $\omega\frac{\tau}{2} = k\pi$, $k \neq 0$: that is, when $\omega = \frac{2k\pi}{\tau}$, as shown in Figure 5.11. Note that the signal has no energy at frequencies corresponding to its zero crossings. This allows us to introduce a narrow-frequency signal at these frequencies and not interfere with the signal. This is the basis of **orthogonal frequency division multiplexing (OFDM)**, which is widely used in such technologies as 4G and WiFi.

The Fourier transform of a signal may not always exist. The existence of a transform is guaranteed by the **Dirichlet conditions**.



7:06

- The signal $x(t)$ is **absolutely integrable**, that is, $\int_{-\infty}^{\infty} |x(t)|dt < \infty$.
- The signal $x(t)$ has a finite number of maxima and minima and a finite number of discontinuities in any finite interval.

These conditions are satisfied by most signals that arise in practical situations. Note that signals with "jumps," such as rectangular pulses, do have Fourier transforms, as we have already seen. Indeed, we can even compute the Fourier transform of a highly discontinuous function, such as an impulse, as in the next example.

EXAMPLE 5.11: FOURIER TRANSFORM OF AN IMPULSE

Compute the Fourier transform of an impulse.

Solution:

The transform is given by

$$X(j\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}dt\,.$$

From Equation 5.15, setting $\tau = 0$, we see that this integral is simply $X(\tau) = e^{-j\omega\tau}\big|_0 = 1$. Thus, the Fourier transform of an impulse is the constant 1, or $\delta(t) \overset{\mathcal{F}}{\leftrightarrow} 1$.

Given a transformed signal, we can recover it by using the inverse Fourier transform, given by

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega \qquad \textbf{(EQ 5.40)}$$

Compare this with Equation 5.32 to see how the summation in the Fourier series is transformed into its limiting integral.

EXAMPLE 5.12: INVERSE FOURIER TRANSFORM

What is the signal corresponding to the transformed signal $\delta(\omega - \omega_0)$?

Solution:

The signal is given by

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \delta(\omega - \omega_0)e^{j\omega t}d\omega\,.$$

But, from Equation 5.15, the integral reduces to $e^{j\omega_0 t}$. Therefore, we have

$$\frac{e^{j\omega_0 t}}{2\pi} \overset{\mathcal{F}}{\leftrightarrow} \delta(\omega - \omega_0), \text{ or } e^{j\omega_0 t} \overset{\mathcal{F}}{\leftrightarrow} 2\pi\delta(\omega - \omega_0).$$

We now turn to some useful properties of the Fourier transform. Proofs of these properties can be derived from first principles, using the definition of the transform, and can be found in standard textbooks on the subject. These properties, along with the table of Fourier transforms in (Table 5.2) allow us to compute the transforms of most functions that arise in practice without having to compute the integral in Equation 5.37 explicitly.

1. **Linearity:** If $x_1(t) \overset{\mathcal{F}}{\leftrightarrow} X_1(j\omega)$ and $x_2(t) \overset{\mathcal{F}}{\leftrightarrow} X_2(j\omega)$, then for any constants a and b, $ax_1(t) + bx_2(t) \overset{\mathcal{F}}{\leftrightarrow} aX_1(j\omega) + bX_2(j\omega)$. Note that a and b can be complex numbers.


3:36

EXAMPLE 5.13: LINEARITY OF THE FOURIER TRANSFORM

Compute the Fourier transform of the eternal sinusoid $\cos(\omega_0 t)$.

Solution:

We use Equation 5.5 to rewrite $\cos(\omega_0 t) = \frac{1}{2}(e^{j\omega_0 t} + e^{-j\omega_0 t})$. From the previous example and the linearity property, we get $\cos(\omega_0 t) \overset{\mathcal{F}}{\leftrightarrow} \pi(\delta(\omega + \omega_0) + \delta(\omega - \omega_0))$. Thus, a sinusoid in the time domain corresponds to two impulses in the frequency domain. These are at the two frequencies ω_0 and $-\omega_0$ and represent counterrotating phasors with the same frequency. The imaginary components of these phasors are always equal and opposite, and their sum is therefore a real sinusoid.

2. **Time-shifting:** If $x(t) \overset{\mathcal{F}}{\leftrightarrow} X(j\omega)$, then $x_1(t - t_0) \overset{\mathcal{F}}{\leftrightarrow} e^{-j\omega t_0} X_1(j\omega)$. That is, shifting the signal forward in time by t_0 units results in multiplying the transform by $e^{-j\omega t_0}$.

3:01

Erratum 5.18: Read $x(t - t_0) \leftrightarrow e^{-j\omega t_0} X(j\omega)$.

EXAMPLE 5.14: FOURIER TRANSFORM OF A TIME-SHIFTED IMPULSE

Compute the Fourier transform of an impulse that occurs at time t_0.

Solution:

Since $\delta(t) \overset{\mathcal{F}}{\leftrightarrow} 1$, we immediately get $\delta(t - t_0) \overset{\mathcal{F}}{\leftrightarrow} e^{-j\omega t_0}$.

3. **Time-scaling:** If $x(t) \overset{\mathcal{F}}{\leftrightarrow} X(j\omega)$ then $x(at) \overset{\mathcal{F}}{\leftrightarrow} \frac{1}{|a|} X\left(\frac{j\omega}{a}\right)$, where a is an arbitrary constant. Intuitively, $x(at)$ is a times "faster" than $x(t)$. This results in the transform's having a magnitude that is a times smaller and a frequency scale that is a times "slower." For instance, if $a = 5$, what happens at time 10 in the unscaled function happens at time 2 in the scaled function. In contrast, a feature present at 100 radians/second in the unscaled transform is present at 500 radians/second in the scaled transform.

EXAMPLE 5.15: FOURIER TRANSFORM OF A SCALED FUNCTION

Compute the Fourier transform of a rectangular pulse as in Figure 5.11 but with a pulse width of 2τ.

Solution:

The transform of a pulse of width τ is $\tau\text{sinc}\left(\frac{\omega\tau}{2}\right)$. When the pulse is twice as wide, the pulse starts at time $-\tau$ instead of at time $-\frac{\tau}{2}$, expanding the time scale by a factor of 2, so that $a = 1/2$. The transform of the longer pulse is $2\tau\text{sinc}(\omega\tau)$. This has zero crossings when $\omega\tau = k\pi$, that is, when $\omega = \frac{k\pi}{\tau}$, which is twice as fast as before.

Erratum 5.19: Zero crossings are twice as close (spacing π/τ).

---

4. **Duality:** If $x(t) \overset{\mathcal{F}}{\leftrightarrow} X(j\omega)$, then $X(t) \overset{\mathcal{F}}{\leftrightarrow} 2\pi x(-j\omega)$ and $X(jt) \overset{\mathcal{F}}{\leftrightarrow} 2\pi x(-\omega)$. This property allows us to use knowledge of a transform to compute new transforms.

Erratum 5.20: See correction 5.20.

EXAMPLE 5.16: DUALITY

What is the Fourier transform of the constant function $1(t)$?


6:05

Solution:

Since $\delta(t) \overset{\mathcal{F}}{\leftrightarrow} 1$, $1(t) \overset{\mathcal{F}}{\leftrightarrow} 2\pi\delta(-j\omega)$. But $\delta(.)$ is symmetric about the origin, so, we have $1(t) \overset{\mathcal{F}}{\leftrightarrow} 2\pi\delta(j\omega)$, which is an impulse centered on the origin, or at zero frequency. This means that all the energy of the constant function is at a frequency of 0, which is what one would expect.

---

5. **Differentiation:** If $x(t) \overset{\mathcal{F}}{\leftrightarrow} X(j\omega)$, then $\frac{dx^n(t)}{dt^n} \overset{\mathcal{F}}{\leftrightarrow} (j\omega)^n X(j\omega)$.

EXAMPLE 5.17: FOURIER TRANSFORM OF A DERIVATIVE

Compute the Fourier transform of the signal $j\omega_0 e^{j\omega_0 t}$.

Solution:

Since $j\omega_0 e^{j\omega_0 t} = \frac{de^{j\omega_0 t}}{dt}$, then $j\omega_0 e^{j\omega_0 t} \overset{\mathcal{F}}{\leftrightarrow} 2\pi(j\omega)\delta(\omega - \omega_0)$.

Table 5.2 presents some standard transform pairs. These are derived using both first principles and the properties of the Fourier transform discussed earlier.

Table 5.2 Some Standard Fourier Transforms

No.	$x(t)$	$X(j\omega)$	Notes
1	$\delta(t)$	1	
2	1	$2\pi\delta(j\omega)$	
3	$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$	
4	$\cos\omega_0 t$	$\pi(\delta(\omega+\omega_0)+\delta(\omega-\omega_0))$	The Fourier transform of a cosine results in impulses in both the positive and negative frequency axes.
5	$\sin\omega_0 t$	$-j\pi(\delta(\omega-\omega_0)-\delta(\omega+\omega_0))$	See preceding note.
6	$u(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$	
7	$e^{-at}u(t)$	$\frac{1}{a+j\omega}$	$a>0$, and this is an exponential defined only for positive time values.
8	$\cos(\omega_0 t)u(t)$	$\frac{\pi}{2}(\delta(\omega+\omega_0)+\delta(\omega-\omega_0))+\frac{j\omega}{\omega_0^2-\omega^2}$	This is a sinusoid defined only for positive time values.
9	$\sin(\omega_0 t)u(t)$	$\frac{\pi}{2j}(\delta(\omega-\omega_0)-\delta(\omega+\omega_0))+\frac{\omega_0}{\omega_0^2-\omega^2}$	This is a sinusoid defined only for positive time values.
10	$rect\left(\frac{t}{\tau}\right)$	$\tau\text{sinc}\left(\frac{\omega\tau}{2}\right)$	This is a pulse symmetric about the origin with width τ.

Erratum 5.21: Read $2\pi\delta(\omega)$.

6:05

6. **Convolution property:** If

$x_1(t) \overset{\mathcal{F}}{\leftrightarrow} X_1(j\omega)$, $x_2(t) \overset{\mathcal{F}}{\leftrightarrow} X_2(j\omega)$, $y(t) \overset{\mathcal{F}}{\leftrightarrow} Y(j\omega)$, and $y(t) = x_1(t) \otimes x_2(t)$, then $Y(jw) = X_1(jw)X_2(jw)$. That is, convolution in the time domain corresponds to multiplication in the transform domain. For an LTI system with an impulse response of $h(t)$, we know from Equation 5.28 that an input of $x(t)$ leads to an output of $y(t) = x(t) \otimes h(t)$. Therefore, $Y(jw) = X(jw)H(jw)$. We can recover $y(t)$ from this by taking the inverse transform.

Erratum 5.22: Read $j\omega$ for jw.

The symmetric convolution property also holds: Multiplication in the time domain corresponds to convolution in the transform domain. That is, if

$x_1(t) \overset{\mathcal{F}}{\leftrightarrow} X_1(j\omega)$, $x_2(t) \overset{\mathcal{F}}{\leftrightarrow} X_2(j\omega)$, $y(t) \overset{\mathcal{F}}{\leftrightarrow} Y(j\omega)$, and $y(t) = x_1(t)x_2(t)$,

then $Y(jw) = \frac{1}{2\pi} X_1(jw) \otimes X_2(jw)$.

EXAMPLE 5.18: USING CONVOLUTION TO SOLVE A SYSTEM

Erratum 5.23: Read: "of its impulse response $h(t)$".

Consider a system such that the Fourier transform $H(j\omega)$ of its transfer function $h(t)$ is $\frac{1}{j\omega + 2}$. What is its response to the input $e^{-t}u(t)$?

Solution:

Because $x(t) = e^{-t}u(t)$, from row 7 of Table 5.2, $X(j\omega) = \frac{1}{1 + j\omega}$. Therefore, $Y(j\omega) = \frac{1}{(1 + j\omega)(2 + j\omega)}$. Using the method of partial fractions (i.e., writing the expression as $\frac{a}{1 + j\omega} + \frac{b}{2 + j\omega}$ and solving for a and b),[7] we find that $Y(j\omega) = \frac{1}{1 + j\omega} - \frac{1}{2 + j\omega}$. To find the inverse transform, recall that the Fourier transform is linear, so we need to find only the inverse transform of each term in isolation. From row 7 of Table 5.2 again, we get $y(t) = (e^{-t} - e^{-2t})u(t)$.

---

7. More generally, for $F(x) = \frac{P(x)}{(x - \lambda_1)(x - \lambda_2)\ldots(x - \lambda_r)}$, we can write

$F(x) = \frac{k_1}{(x - \lambda_1)} + \ldots + \frac{k_r}{(x - \lambda_r)}$, where $k_r = [(x - \lambda_r)F(x)]\big|_{x = \lambda_r}$. Also see Section 8.11.

It is clear that the Fourier transform greatly simplifies the analysis of LTI systems: Instead of having to deal with the complex convolution operator, we need merely to deal with multiplying two transforms and then taking the inverse. In general, we can always find the inverse of the product from first principles by using Equation 5.40. However, because the Fourier transform may not always exist, system analysis is typically performed using the Laplace transform.

This concludes our discussion of the Fourier transform. We now turn our attention to the more general Laplace transform, which shares most of the properties of the Fourier transform and can also be applied to a wider range of functions.

5.8 The Laplace Transform



8:45

Lab: Laplace transform

The Fourier transform gives us insight into the frequencies in which a signal's energy is concentrated and simplifies the computation of convolutions. Its main problem is that it is not guaranteed for functions that do not satisfy the Dirichlet conditions. We now study the Laplace transform, which shares most of the properties of the Fourier transform and, additionally, is defined for a wider class of functions.

Recall that the Fourier transform is defined by $X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$. The Laplace transform is defined by

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt \qquad \textbf{(EQ 5.41)}$$

where $s = \sigma + j\omega$. Intuitively, instead of taking the limiting sum of a series of pure sinusoids, as with the Fourier transform, the Laplace transform takes the limiting sum of a series of complex exponentials. Alternatively, writing e^{-st} as $e^{-\sigma t}e^{-j\omega t}$, we can write

$$X(s) = \int_{-\infty}^{\infty} (x(t)e^{-\sigma t})e^{-j\omega t}dt \qquad \textbf{(EQ 5.42)}$$

so that the Laplace transform is identical to the Fourier transform of the function $(x(t)e^{-\sigma t})$. Intuitively, given a signal $x(t)$ that does not satisfy the Dirichlet conditions, we multiply it by an exponentially decreasing function $e^{-\sigma t}$, so that the product becomes absolutely integral. Note that, in this process, we must be careful to select a value of σ large enough to ensure convergence. Therefore, unlike the Fourier transform, in specifying the Laplace transform, it is essential to give the **region of convergence**, that is, the values of σ for which the Laplace transform is defined.

Erratum 5.24: Read "absolutely integrable".

As with the Fourier transform, we denote the relationship between a signal $x(t)$ and its transform $X(s)$ by using the notation

$$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s) \qquad \textbf{(EQ 5.43)}$$

Note that the Laplace transform of a signal is, in general, a *complex* function, with both real and imaginary parts, hence its representation as $X(s)$.

The inverse Laplace transform is given by

$$x(t) = \frac{1}{2\pi j} \int_{(\sigma - j\infty)}^{(\sigma + j\infty)} X(s) e^{st} ds \qquad \textbf{(EQ 5.44)}$$

The limits of the integral denote integrating from $-\infty$ to ∞ along the imaginary axis for a value of σ chosen in the transform's region of convergence. In practice, the inverse operation is rarely performed from first principles, because the inverse can be found from a table of standard transforms and the (nice!) properties of the Laplace transform.

Erratum 5.25: Integrate along the line $\mathrm{Re}(s) = \sigma$.

5.8.1 Poles, Zeroes, and the Region of Convergence



11:30

We now examine in greater depth the somewhat mysterious notion of a region of convergence by referring to a concrete example. Observe that the unit step signal $u(t)$ is not absolutely integrable because the area under this signal is infinite. Therefore, it does not satisfy the Dirichlet conditions, and a Fourier transform is not guaranteed. Nevertheless, it can be shown that its Fourier transform is $\pi\delta(\omega) + \frac{1}{j\omega}$. In the next example, we will compute its Laplace transform.

EXAMPLE 5.19: LAPLACE TRANSFORM OF A UNIT STEP

Compute the Laplace transform of the unit step signal.

Solution:

By definition,

$$X(s) = \int_{-\infty}^{\infty} u(t) e^{-st} dt = \int_{0}^{\infty} e^{-st} dt = \left. \frac{e^{-st}}{-s} \right|_{0}^{\infty}.$$

Now, $e^{-s\infty} = 0$ if $Re(s) > 0$. Assuming that this is the case, we can evaluate the integral as

$$\frac{1}{s} \qquad Re(s) > 0\,.$$

Thus,

$$u(t) \overset{\mathscr{L}}{\leftrightarrow} \frac{1}{s} \qquad Re(s) > 0\,.$$

The region of convergence of this transform is the set of all values of s where the condition $Re(s) > 0$ holds. Recall that s is complex, so this is the right-half plane of the complex plane.

The next example reinforces this notion.

EXAMPLE 5.20: LAPLACE TRANSFORM OF A REAL EXPONENTIAL

Compute the Laplace transform of the signal $x(t) = u(t)e^{at}$, where a is a real constant.

Solution:

By definition,

$$X(s) = \int_{-\infty}^{\infty} u(t)e^{at}e^{-st}dt = \int_{0}^{\infty} e^{-(s-a)t}dt = \left.\frac{e^{-(s-a)t}}{-(s-a)}\right|_0^{\infty}.$$

As before, $e^{-(s-a)\infty}$ is 0 if $Re(s-a) > 0$, that is, $Re(s) > a$. If this condition holds, the integral evaluates to $\frac{1}{(s-a)}$. Therefore,

$$u(t)e^{-at} \overset{\mathscr{L}}{\leftrightarrow} \frac{1}{(s-a)} \qquad Re(s) > a.$$

Erratum 5.26: Read $u(t)e^{at} \leftrightarrow 1/(s-a)$.

In this case, the region of convergence is the complex half-plane defined by $Re(s) > a$.

What if $s = a$? In this case, the denominator becomes zero, and the transform's value is infinite. This is called a **pole** of the system. (The pole in the previous example was at 0.)

It is easy to show that the transform pair is valid even for a complex a, where $a = \sigma + j\omega$, as long as $Re(s) > \sigma$.

The values of s for which the transform vanishes are called the **zeroes** of the transform. This is illustrated by the next example.

EXAMPLE 5.21: LAPLACE TRANSFORM OF A SINUSOID

Compute the Laplace transform of the sinusoid $u(t)\cos\omega_1 t$.

Solution:

We use Euler's formula to rewrite the signal as $u(t)\left(\frac{e^{j\omega_1 t}+e^{-j\omega_1 t}}{2}\right)$. By definition,

$$\begin{aligned} X(s) &= \int_{-\infty}^{\infty} u(t)\left(\frac{e^{j\omega_1 t}+e^{-j\omega_1 t}}{2}\right)e^{-st}dt \\ &= \frac{\left(\int_0^{\infty} e^{j\omega_1 t}e^{-st}dt + \int_0^{\infty} e^{-j\omega_1 t}e^{-st}dt\right)}{2} \\ &= \frac{1}{2}\left(\frac{1}{s-j\omega_1}+\frac{1}{s+j\omega_1}\right), \end{aligned}$$

where, in the last step, we used the result from the previous example, and we are assuming that $Re(s) > 0$. This reduces to $\frac{s}{s^2+\omega_1^2}$, with the region of convergence as $Re(s) > 0$. Note that the transform becomes infinite for $s = \pm j\omega_1$—the poles of the transform—and is zero for $s = 0$, which is the zero of the transform.

Note that the zero lies outide of the region of convergence.

It is important to keep track of the region of convergence of the transform. Two time-domain functions that are completely different may have an identical transform and may differ only in their region of convergence. Therefore, the Laplace transform is unique only if the region is also specified.

If the region of convergence of the Laplace transform of a signal includes the imaginary axis, the Fourier transform of the signal is defined and can be obtained by setting $s = j\omega$. Otherwise, the Laplace transform of the signal exists but not its Fourier transform.

5.8.2 Properties of the Laplace Transform

The Fourier transform is a special case of the Laplace transform. Therefore, almost all the properties of the Fourier transform—namely, linearity, time-shifting, time-scaling, differentiation, and the convolution property—also hold for the Laplace transform, though in a slightly different form. We summarize the corresponding properties in Table 5.3.

Note that the differentiation result in the table is for the bilateral Laplace transform. The unilateral transform of dx(t)/dt = sX(s) - x(0). With frequency scaling, the new ROC is α + Re(a) < Re(s) < β + Re (a).

Table 5.3 Properties of the Laplace Transform

Property	Precondition	Postcondition	Notes
Linearity	$x_1(t) \overset{\mathcal{L}}{\leftrightarrow} X_1(s)$ $\quad \alpha_1 < Re(s) < \beta_1$ $x_2(t) \overset{\mathcal{L}}{\leftrightarrow} X_2(s)$ $\quad \alpha_2 < Re(s) < \beta_2$	$ax_1(t) + bx_2(t) \overset{\mathcal{L}}{\leftrightarrow} aX_1(s) + bX_2(s)$ $max(\alpha_1, \alpha_2) < Re(s) < min(\beta_1, \beta_2)$	Both *a* and *b* are arbitrary constants and can be complex.
Time scaling	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s)$ $\quad \alpha < Re(s) < \beta$	$x(at) \overset{\mathcal{L}}{\leftrightarrow} \frac{1}{\lvert a \rvert} X\left(\frac{s}{a}\right)$ $a\alpha < Re(s) < a\beta$	A compression in the time scale expands the frequency scale.
Frequency scaling	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s)$ $\quad \alpha < Re(s) < \beta$	$\frac{1}{\lvert a \rvert} x\left(\frac{t}{a}\right) \overset{\mathcal{L}}{\leftrightarrow} X(as)$ $\frac{\alpha}{a} < Re(s) < \frac{\beta}{a}$	A compression in the frequency scale expands the time scale.
Time-shifting	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s)$ $\quad \alpha < Re(s) < \beta$	$x(t - t_0) \overset{\mathcal{L}}{\leftrightarrow} e^{-st_0} X(s)$ $\alpha < Re(s) < \beta$	Delaying by a time t_0 multiplies the transform by e^{-st_0}.
Frequency-shifting	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s)$ $\quad \alpha < Re(s) < \beta$	$e^{at} x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s - a)$	Note the change in the region of convergence due to frequency shifting. $\alpha + Re(a) < Re(s) < \beta + Re(a)$
Differentiation	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s)$ $\quad \alpha < Re(s) < \beta$	$\frac{d^n x(t)}{dt^n} \overset{\mathcal{L}}{\leftrightarrow} s^n X(s)$ $\alpha < Re(s) < \beta$	Differentiation in the time domain corresponds to multiplication by a factor of *s* in the transform domain.

continues

Note that this differentiation result is for the bilateral Laplace transform. The unilateral transform of dx(t)/dt = sX(s) - x(0).

$\alpha + Re(a) < Re(s) < \beta + Re(a)$

Table 5.3 Properties of the Laplace Transform (*Continued*)

Property	Precondition	Postcondition	Notes
Integration	$x(t) \overset{\mathcal{L}}{\leftrightarrow} X(s) \qquad \alpha < Re(s) < \beta$	$\int_{-\infty}^{t} x(r)dr \overset{\mathcal{L}}{\leftrightarrow} \frac{X(s)}{s}$ $max(\alpha, 0) < Re(s) < \beta$	Integration in the time domain corresponds to division by a factor of s in the transform domain. Note that the region of convergence also changes.
Convolution in time domain	$x_1(t) \overset{\mathcal{L}}{\leftrightarrow} X_1(s) \qquad \alpha_1 < Re(s) < \beta_1$ $x_2(t) \overset{\mathcal{L}}{\leftrightarrow} X_2(s) \qquad \alpha_2 < Re(s) < \beta_2$ $y(t) = x_1(t) \otimes x_2(t)$	$y(t) \overset{\mathcal{L}}{\leftrightarrow} Y(s)$ $Y(s) = X_1(s)X(s)_2$ $max(\alpha_1, \alpha_2) < Re(s) < min(\beta_1, \beta_2)$	The value $y(t)$ is a convolution of two functions $x_1(t)$ and $x_2(t)$. The product of their transforms $X_1(s)$ and $X_2(s)$ determine $Y(s)$, the transform of $y(t)$, and its region of convergence.
Multiplication in time domain	$x_1(t) \overset{\mathcal{L}}{\leftrightarrow} X_1(s) \qquad \alpha_1 < Re(s) < \beta_1$ $x_2(t) \overset{\mathcal{L}}{\leftrightarrow} X_2(s) \qquad \alpha_2 < Re(s) < \beta_2$ $y(t) = x_1(t)x_2(t)$	$y(t) \overset{\mathcal{L}}{\leftrightarrow} Y(s)$ $Y(s) = \frac{1}{2\pi j} X_1(s) \otimes X_2(s)$ $\alpha_1 + \alpha_2 < Re(s) < \beta_1 + \beta_2$	The value $y(t)$ is a product of two functions $x_1(t)$ and $x_2(t)$. The convolution of their transforms $X_1(s)$ and $X_2(s)$ determines $Y(s)$, the transform of $y(t)$, and its region of convergence.
Final value theorem	$X(s)$	$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s)$	The limiting value of $x(t)$ in the time domain as $t \to \infty$ is given by finding the limit of $sX(s)$ as $s \to 0$ in the transform domain.

Erratum 5.27: Convolution row: read $Y(s) = X_1(s)X_2(s)$.

Erratum 5.28: Final value theorem: it holds only if all poles of $sX(s)$ are in the left half-plane.

Table 5.4 Some Standard Laplace Transforms

No.	Signal	$x(t)$	$X(s)$	Region of Convergence
1	Delta or unit impulse	$\delta(t)$	1	All s
2	Unit step	$u(t)$	$1/s$	$Re(s) > 0$
3	Delayed delta	$\delta(t - t_0)$	$e^{-t_0 s}$	All s
4	Ramp	$tu(t)$	$\frac{1}{s^2}$	$Re(s) > 0$
5	Exponential decay	$e^{\alpha t}u(t)$	$\frac{1}{s-\alpha}$	$Re(s) > \alpha$
6	*N*th power decay	$\frac{t^n}{n!}e^{\alpha t}u(t)$	$\frac{1}{(s-\alpha)^{n+1}}$	$Re(s) > \alpha$
7	Sine	$\sin(\omega t)u(t)$	$\frac{\omega}{s^2+\omega^2}$	$Re(s) > 0$
8	Cosine	$\cos(\omega t)u(t)$	$\frac{s}{s^2+\omega^2}$	$Re(s) > 0$
9	Exponentially modulated sine	$e^{\alpha t}\sin(\omega t)u(t)$	$\frac{\omega}{(s-\alpha)^2+\omega^2}$	$Re(s) > \alpha$
10	Exponentially modulated cosine	$e^{\alpha t}\cos(\omega t)u(t)$	$\frac{s-\alpha}{(s-\alpha)^2+\omega^2}$	$Re(s) > \alpha$

These properties, along with the table of common transforms (Table 5.4) allow us to derive the transform of many common signals without having to derive them from first principles. Note that these transforms are defined for functions that exist only for $t > 0$, so that the Laplace integral has limits from 0 to ∞. This is also called the **unilateral** Laplace transform. In a practical system, a *causal* signal is 0 for $t < 0$, so the unilateral transform suffices for all practical systems.

EXAMPLE 5.22: USING THE LAPLACE TRANSFORM TO SOLVE A SYSTEM

Consider a system such that the Laplace transform $H(s)$ of its transfer function $h(t)$ is $\frac{1}{s+2}$. What is its response to the input $e^{-t}u(t)$?

Erratum 5.29: Read "of its impulse response $h(t)$".

Solution:

Since $x(t) = e^{-t}u(t)$, from row 5 of Table 5.4, $X(s) = \frac{1}{s+1}$. Therefore, $Y(s) = \frac{1}{(1+s)(2+s)}$. Using the method of partial fractions, i.e., writing the expression as $\frac{a}{1+s} + \frac{b}{2+s}$ and solving for a and b—also see Section 8.11—we find that $Y(s) = \frac{1}{1+s} - \frac{1}{2+s}$. To find the inverse transform, recall that the Laplace transform is linear, so we need only find the inverse transform of each term in isolation. From row 5 of Table 5.4 again, we get $y(t) = (e^{-t} - e^{-2t})u(t)$.

This analysis assumes that the system is at rest at time zero. If this is not the case, the actual response would be the sum of the natural response, which is the way the system behaves if there is no external input, and the forced response, which is the way the system behaves assuming that its initial state, is at rest. The details of this analysis are beyond the scope of this text.

8.3 A First-Order System



8:29

Lab: First- and second-order systems

One of the simplest systems that can be characterized by a transfer function is a **first-order** system with a transfer function

$$G(s) = \frac{K}{1+\tau s} \qquad \textbf{(EQ 8.7)}$$

where K is called the **gain parameter** and τ is called the **system time constant**. The reason for this nomenclature is best understood by considering the response of this system to a step input $u(t)$, that is, an input that is one unit for all $t > 0$. Recall that the Laplace transform of a step input is $1/s$. Therefore, the Laplace transform of the output, $Y(s)$, is given by

$$Y(s) = \frac{K}{s(1+\tau s)}$$

It is easy to see that the roots of the denominator are s=0 and $s = -\frac{1}{\tau}$. We use this fact to compute the partial fraction expansion as

$$Y(s) = K\left[\frac{1}{s} - \frac{\tau}{1+\tau s}\right]$$

Using Table 5.4 in Section 5.8, we compute

$$y(t) = K\left(1 - e^{\frac{-t}{\tau}}\right) \qquad \textbf{(EQ 8.8)}$$

which is shown in Figure 8.3 for K = 2.5 and $\tau = 2$.

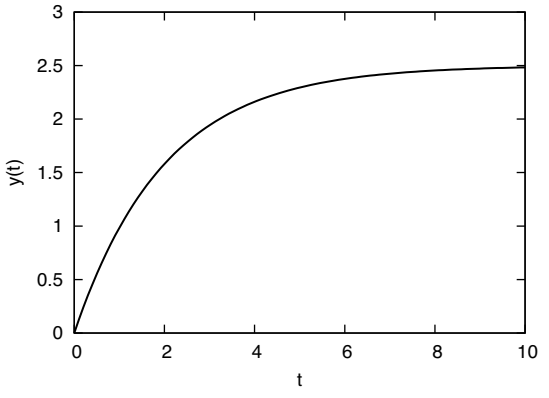


Figure 8.3 Step response of a first-order system with gain 2.5 and time constant 2

A close examination of Equation 8.8 and Figure 8.3 reveals the following.

- In response to a step input, the output rises rapidly, eventually asymptoting to a value that is K times the input value of 1. In other words, in steady state, the system magnifies the input by a factor of K, thus justifying K as the system gain parameter.
- The system reaches $(1 - 1/e)$, or roughly 63% of its final value, when $t = \tau = 2$ and 99.33% of its final value when $t = 5\tau = 10$. For all practical purposes, the system reaches its asymptotic value at time 5τ, justifying τ as the system time constant.
- A step input is like a sharp kick to the system. We can write the system's **step response** to this input as $y(t) = K(1 - e^{-t/\tau}) = K - Ke^{-t/\tau}$. The first term is the **steady state response**, and the second term is the **transient response**. In general, given any sharp change in input, any system will exhibit both forms of response. We are interested in quantifying both. Given the particularly simple form of the Laplace transform of a step function, we often characterize a system by its step response.
- Step inputs are easy to generate. Therefore, if we can model a system as a first-order system, it is possible to numerically obtain the value of K and τ by giving it a step input and plotting its output.

18:01

8.4 A Second-Order System

Intuitively, a second-order system arises when the system responds to the input by accumulating and then releasing energy. A classic example is a violin string excited by a bow. As the bow moves over the string, the string distorts, accumulating energy. At some point, the accumulated energy is sufficient to overcome the motion of the bow, and the string returns and then overshoots its neutral position. The process repeats, resulting in an oscillation, powered by the energy of the bow.

Compared to a first-order system, a second-order system is more complex but still relatively easy to analyze. Its transfer function is given by

$$G(s) = \frac{K}{\frac{s^2}{\omega_n^2} + \frac{2\varsigma s}{\omega_n} + 1} \quad \textbf{(EQ 8.9)}$$

A second-order system is characterized by three factors: its **gain** K, its **natural frequency** ω_n (in radians/second), and its **damping ratio** ς (read *zeta*). As before, to understand the physical meanings of these terms, we consider its unit step response. For a step input, we have

$$Y(s) = \frac{K}{s\left(\frac{s^2}{\omega_n^2} + \frac{2\varsigma s}{\omega_n} + 1\right)}$$

We study four cases corresponding to different values of ς.

8.4.1 Case 1 (Undamped System): $\varsigma = 0$

When $\varsigma = 0$,

$$Y(s) = \frac{K}{s\left(\frac{s^2}{\omega_n^2} + 1\right)} = \frac{K\omega_n^2}{s(s^2 + \omega_n^2)} = K\left(\frac{1}{s} - \frac{s}{s^2 + \omega_n^2}\right)$$

Using Table 5.4 in Section 5.8, we compute

$$y(t) = K(1 - \cos\omega_n t) \quad \textbf{(EQ 8.10)}$$

which is shown in Figure 8.4 for $K = 2.5, \omega_n = 2$.

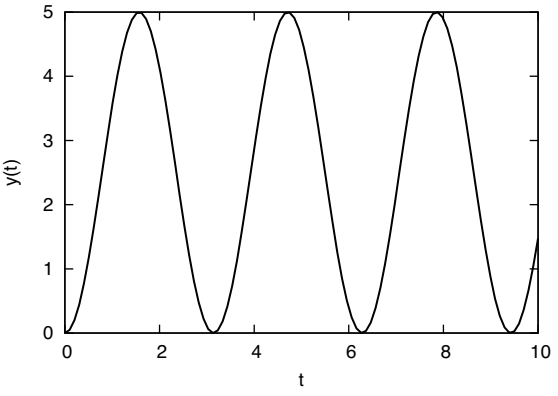


Figure 8.4 Unit step response of an undamped second-order system with $K = 2.5$, $\omega_n = 2$

The output of an undamped second-order system is oscillatory, with a frequency of ω_n. The gain parameter determines the amplitude of the oscillations. These justify naming K and ω_n as the gain and natural frequency, respectively.

8.4.2 Case 2 (Underdamped System): $0 < \varsigma < 1$

When $0 < \varsigma < 1$, the Laplace transform of the step response is given by:

$$Y(s) = \frac{K}{s\left(\frac{s^2}{\omega_n^2} + \frac{2\varsigma s}{\omega_n} + 1\right)} = \frac{K\omega_n^2}{s(s^2 + 2\varsigma s + \omega_n^2)} = K\left(\frac{1}{s} - \frac{(s+\varsigma\omega_n)}{(s+\varsigma\omega_n)^2 + \omega_d^2} - \frac{\varsigma\omega_n}{(s+\varsigma\omega_n)^2 + \omega_d^2}\right)$$

This should be 2 ςωns

where $\omega_d^2 = \omega_n^2(1-\varsigma^2)$.

Using Table 5.4 in Section 5.8, we compute

$$y(t) = K\left(1 - e^{-\varsigma\omega_n t}\left(\cos\omega_d t + \frac{\varsigma}{\sqrt{1-\varsigma^2}}\sin\omega_d t\right)\right)$$

$$y(t) = K\left(1 - \frac{e^{-\varsigma\omega_n t}}{\sqrt{1-\varsigma^2}}(\sqrt{1-\varsigma^2}\cos\omega_d t + \varsigma\sin\omega_d t)\right)$$

Let $\varsigma = \cos\theta$ so that $\theta = \cos^{-1}\varsigma$. Then $\sqrt{1-\varsigma^2} = \sqrt{1-(\cos\theta)^2} = \sin\theta$, so that

Erratum 8.5: See correction 8.5.

$$y(t) = K\left(1 - \frac{e^{-\varsigma\omega_n t}}{\sqrt{1-\varsigma^2}}(\sin\theta\cos\omega_d t + \cos\theta\sin\omega_d t)\right)$$

Erratum 8.6: See correction 8.6 (typesetting).

From the identity $\sin(A + B) = \sin A \cos B + \cos A \sin B$, we get

$$y(t) = K\left(1 - \frac{e^{-\varsigma\omega_n t}}{\sqrt{1-\varsigma^2}}\sin(\omega_d t + \cos^{-1}\zeta)\right);\ t \geq 0 \qquad \textbf{(EQ 8.11)}$$

This should be cos -1 ς

From the form of Equation 8.11, we see that the sinusoidal oscillation is modulated by a decaying exponential of the form $e^{-\varsigma\omega_n t}$. For a fixed value of ω_n, as ς increases, the exponent dies down more rapidly, damping the system. This justifies calling ς the damping ratio. We also see that the phase of the sinusoid is shifted (compared to the undamped system) by a phase angle $arc\tan\frac{\sqrt{1-\varsigma^2}}{\varsigma}$, a function of ς.

Figure 8.5 shows $y(t)$ for different values of ς, while keeping the values of K and ω_n the same as in the case of the undamped system: $K = 2.5, \omega_n = 2$. As ς approaches 1, the transient response of system becomes less oscillatory. For example, when $\varsigma = 0.2$, the system continues to oscillate even after five time periods (i.e., t=10). On the other hand, when $\varsigma > 0.8$, the output steadily approaches the asymptotic value of 2.5.

Erratum 8.7: $t = 10$ is about three periods.

Intuitively, the damping ratio determines the responsiveness of the system. When this ratio is small, the system is "jumpy," responding immediately to a stimulus but taking a long time to reach steady state after receiving a shock. As the ratio increases, the system is more sluggish but reaches its asymptote smoothly.

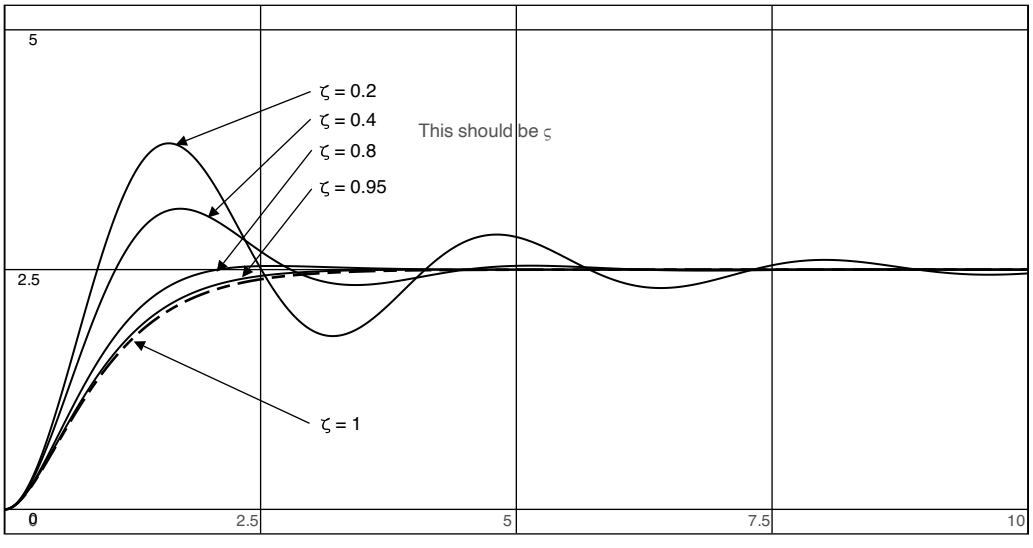


Figure 8.5 Unit step response of an underdamped second-order system with $K = 2.5, \omega_n = 2; \varsigma = 0.2, 0.4, 0.8, 0.95, 1.0$

8.4.3 Case 3 (Critically Damped System): ς = 1

When $\varsigma = 1$, we say that the system is **critically damped**. When subjected to a unit step input, this system is neither too responsive nor too sluggish, rising smoothly to its asymptotic value. Of course, choosing parameters to achieve critical damping for any real system can be a challenge!

To mathematically study such a system, we use the partial fraction expansion of $Y(s)$ as before to find

In this formula, replace ω with ω_n

$$Y(s) = \frac{K\omega^2}{s(s^2 + 2\omega s + \omega^2)} = K\left[\frac{1}{s} - \frac{\omega}{(s+\omega)^2} - \frac{1}{s+\omega}\right]$$

$$y(t) = K[1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}] \quad \textbf{(EQ 8.12)}$$

Note that the output $y(t)$ has no sinusoidal components. The corresponding function is shown in Figure 8.5. Comparing the output of a critically damped system with that of an undamped and underdamped system, it should be clear why a critically damped system is ideal.

8:01

8.4.4 Case 4 (Overdamped System): ς > 1

When the damping ratio exceeds 1, the output of a second-order system becomes less and less responsive to a step input. Such a system is **overdamped**. We can mathematically study the system as follows. We have

$$Y(s) = \frac{K\omega_n^2}{s(s^2 + 2\varsigma\omega_n s + \omega_n^2)}$$

The roots of the second term in the denominator, from the quadratic formula, are

$$\frac{-2\varsigma\omega_n \pm \sqrt{4\omega_n^2\varsigma^2 - 4\omega_n^2}}{2},$$

so that

$$Y(s) = \frac{K\omega_n^2}{s(s + \varsigma\omega_n + \omega_n\sqrt{\varsigma^2 - 1})(s + \varsigma\omega_n - \omega_n\sqrt{\varsigma^2 - 1})}$$

For convenience, let $\gamma = \sqrt{\varsigma^2 - 1}$. From the inverse Laplace transform of the partial fraction expansion, we find that

$$y(t) = K\left[1 + \frac{e^{-(\varsigma+\gamma)\omega_n t}}{2\gamma(\varsigma+\gamma)} - \frac{e^{-(\varsigma-\gamma)\omega_n t}}{2\gamma(\varsigma-\gamma)}\right] \quad \textbf{(EQ 8.13)}$$

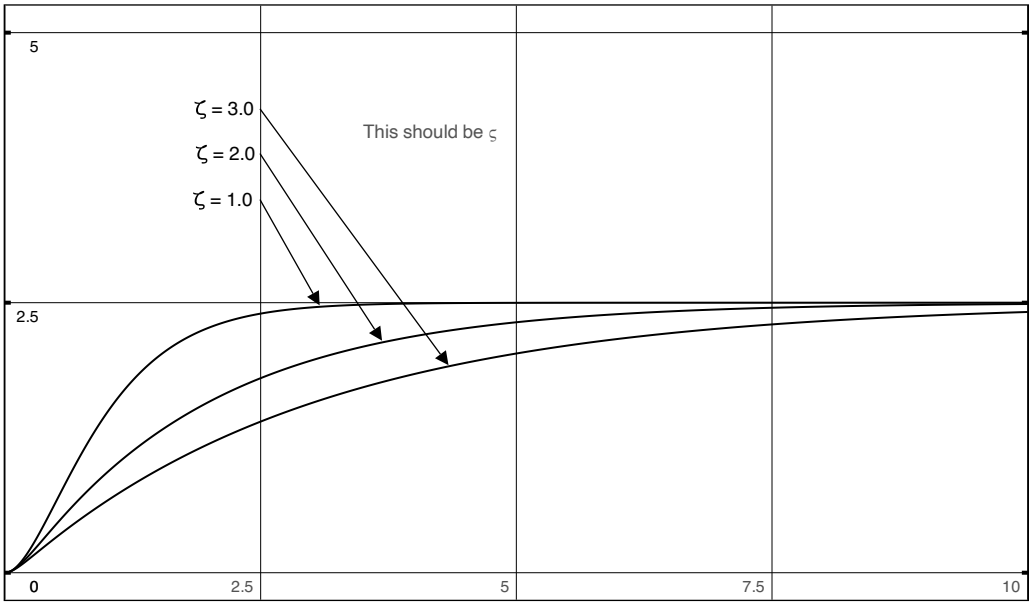


Figure 8.6 Unit step response for an overdamped system with $K = 2.5$, $\omega_n = 2$, $\varsigma = 1.0, 2.0, 3.0$

The step response has no sinusoidal components, being the difference between two exponential curves. The function is plotted in Figure 8.6, with the same values of K and ω_n as before. We see that as the damping ratio increases, the system takes progressively longer to reach its asymptote.

To summarize, the behavior of a second-order system is controlled by its gain, damping ratio, and natural frequency parameters. The gain controls the steady state response to a unit step input. When underdamped, the natural frequency controls the step-response oscillation frequency. Finally, the damping ratio controls the responsiveness of the step response. In designing a control system, it is usually possible to choose arbitrary values for these three parameters. Our goal, therefore, is to tune the system to achieve the desired gain and critical damping.

In practice, no system is truly a second-order system: This is just a convenient approximation. However, the system gives us valuable insights into the behavior of many real systems, as the next example demonstrates.

EXAMPLE 8.5: ROUTE FLAPPING AS A SECOND-ORDER SYSTEM

In a computer network, the dynamic selection of least-delay routes between two nodes can be modeled as a second-order system. Consider two nodes A and B that are connected by two edge-disjoint paths. Suppose that all the traffic

from A to B goes over the first path. This load increases the delay on this path, which we can model as the accumulation of energy. At some point, the energy on the path is so large that the routing algorithm running at node A decides to route traffic destined to B on the second path. This reduces the energy on the first path and increases it on the second. As before, this results in an over-accumulation of energy on the second path, resulting in an oscillatory system exhibiting **route flapping**.

To make this concept more precise, consider a network in which both paths from A and B have a capacity of 1 unit/second. As long as the load from A to B is smaller than 1 unit/second, both paths are uncongested, and the system does not exhibit oscillations. However, suppose that A injects load at the rate of 2 units/second into the first path. This will result in the buffer on the path building up at the rate of 1 unit/second. Suppose that the routing protocol running at A switches paths when the path delay exceeds 5 seconds. This will happen at $t = 5$ seconds. At this point, the load switches to the second path, and the queue there will build up at the rate of 1 unit/second until at time $t = 10$ seconds, traffic will revert to the first path. Clearly, the oscillations have a period of 10 seconds.

EXAMPLE 8.6: DAMPING ROUTE FLAPPING

We saw the oscillatory behavior of a naive threshold-based routing protocol in Example 8.5. We can use the insights from second-order systems to reduce oscillations by using damping. The essence of damping is to reduce system responsiveness. We use this insight as follows: Suppose that the routing algorithm, instead of sending *all* data on the shortest path, distributed load inversely proportional to the measured path delay. Then, the load from A to B would be spread on both paths, increasing overall capacity. Moreover, if there were a sudden increase in delay along one path, load would be proportionally redistributed to another path. By reacting less aggressively to an increase in load, this damped routing algorithm would reduce the oscillatory behavior of the network. The open shortest path first (OSPF) routing protocol in the Internet allows multipath routing for equal-cost paths, so this form of damping is feasible even in practice.

5.12 Exercises

1. **Complex arithmetic**

   Compute $e^{-j\frac{\pi}{2}} + e^{j\frac{\pi}{2}}$.

2. **Phase angle**

   What is the phase angle corresponding to the complex number $1+j$?

3. **Discrete convolution**

   Let $z(t)$ be the convolution of $x(t),\ 0 \le t \le 9\ = 1, 3, 5, 2, 5, 8, 7, 3, 9, 4$ and $y(t)$, $0 \le t \le 9\ = 3, 1, 7, 4, 5, 9, 7, 1, 3, 8$. Compute $z(5)$.

4. **Signals**

   Give an example of a signal that is continuous, digital, aperiodic, and time-unlimited.

5. **Complex exponential**

   Describe the Im-t projection of the curve $5e^t(\cos 3t + j\sin 3t)$.

6. **Linearity**

   Is the system defined by the transfer function $H(x) = 5\frac{dx}{dt} + 1$ linear?

7. **LTI system**

   What is the output of an LTI system when the input is a real sinusoid?

8. **Natural response**

   Compute the natural response of the LTI system given by
   $2\frac{d^2y(t)}{dt^2} + 11\frac{dy(t)}{dt} + 15y(t) = 32x(t)$.

9. **Natural response**

   Determine the exact natural response of the system described by
   $2\frac{d^2y(t)}{dt^2} + y(t) = 32x(t);\ y(0) = 0,\ \dot{y}(0) = 1$. What is its frequency?

10. Stability

Characterize the stability of the system in Exercise 9.

11. Fourier series

Find the Fourier series corresponding to the infinite series of isosceles triangular pulses of base width 2τ and height 1 spaced apart T_0 seconds. Use the fact that $\int xe^{ax}dx = \frac{axe^{ax} - e^{ax}}{a^2} + C$.

12. Fourier series

Compute the third coefficient of the Fourier series representing a unit periodic rectangular pulse of width $1s$ spaced apart by 10s.

13. Fourier transform

Find the Fourier transform of a single left-triangular pulse defined by $1 - t$ in [0,1].

14. Inverse Fourier transform

Compute the inverse Fourier transform of the function $\pi(\delta(\omega + \omega_0) + \delta(\omega - \omega_0))$ from first principles.

15. Computing the Fourier transform

Use the properties of the Fourier transform to compute the transform of the function $\cos(\omega_0(t + t_0)) + \sin(\omega_0(t - t_0))$.

16. Laplace transform

Compute the Laplace transform of the sinusoid $u(t)\sin\omega_0 t$ from first principles. Locate its pole(s) and zero(es).

17. Laplace transform

Find the Laplace transform of the signal $u(t\text{-}t_0)\sin\omega_0(t - t_0)$.

18. Using the Laplace transform to solve a system

Consider a system whose impulse response $h(t)$ is given by $\cos(\omega_0 t)$. What is its response to the input $e^{-t}u(t)$? Assume that the system is causal.

Note: Exercises on first-order and second-order systems (Sections 8.3 and 8.4) are in the exercises for Chapter 8, Section 8.13.

Corrections to Chapter 5

The pages of Chapter 5 (Signals, Systems, and Transforms) are reproduced from the book with corrections. The following errata are not yet corrected in the text; each is flagged by a red note in the margin of the page concerned.

5.1 Section 5.1, page 174: "a vector lying on the X-axis drawn from $(0,0)$ to $(0,x)$" should read "... from $(0,0)$ to $(x,0)$".

5.2 Section 5.1, page 175: "the vector 1, which is the vector from $(0,0)$ to $(0,1)$" should read "... from $(0,0)$ to $(1,0)$".

5.3 Example 5.2, page 177: the last two lines should read $(0.951{+}1.618){-}j(0.309{+}1.176) = 2.569{-}j1.485$. The book has the wrong sign on the imaginary part and small rounding errors.

5.4 Page 177, Equation 5.4: $\theta = \text{atan}(b/a)$ gives an angle in the wrong quadrant when $a < 0$; in that case, add π. (Equivalently, use $\text{atan2}(b,a)$.)

5.5 Page 177: "can equally well be generated by a phasor rotating in the counterclockwise direction" should read "... in the clockwise direction".

5.6 Example 5.3, page 178: 65,536 = 2^{16} phases carry 16 bits per symbol. Sending 32 bits at a time would need 2^{32} phases.

5.7 Page 186: "which is a real sinusoid scaled by a factor k" should read "scaled by a factor $2k$".

5.8 Page 188, footnote 4, and page 189: the transfer-function part of Section 8.2.2 is in these notes, but the state-variable and linearization part is not.

5.9 Section 5.4.2, page 191, just below Equation 5.22: "In nearly all practical systems, $M > N$" should read "In nearly all practical systems, $M \le N$." (If $M > N$, the gain grows without bound at high frequencies, and the coefficient b_{N-M} would have a negative subscript.)

5.10 Page 195: the last case should read "Finally, if no value of σ is positive and at least one is 0, the behavior of the system depends on whether the roots with $\sigma = 0$ are repeated". The other roots may have negative σ.

5.11 Page 197: "can be thought of as a being generated" should read "can be thought of as being generated".

5.12 Section 5.6, page 197, footnote 6: "Gibb's phenomenon" should read "Gibbs phenomenon".

5.13 Section 5.6, page 198, just above Equation 5.33: "... each with a real magnitude c_k" should read "... each with a complex amplitude c_k, that is, a magnitude $|c_k|$ and a starting phase $\arg c_k$". c_k is real only when $x(t)$ is even.

5.14 Section 5.6, page 200: the sinc function is a sine wave whose amplitude falls off in inverse proportion to the distance from the Y-axis, not one that is "linearly modulated" by that distance.

5.15 Page 201: $X(jw)$ should read $X(j\omega)$.

5.16 Figure 5.11, page 202: the transform is labelled $F(j\omega)$; it should be $X(j\omega)$, to match the text.

5.17 Example 5.10, page 202: "jw" in the denominator should read $j\omega$.

5.18 Section 5.7, page 205, time-shifting property: $x_1(t - t_0) \leftrightarrow e^{-j\omega t_0} X_1(j\omega)$ should read $x(t - t_0) \leftrightarrow e^{-j\omega t_0} X(j\omega)$, to match $x(t) \leftrightarrow X(j\omega)$.

5.19 Example 5.15, page 206: the zero crossings of $2\tau\,\mathrm{sinc}(\omega\tau)$ are twice as close together as before (spaced π/τ rather than $2\pi/\tau$); "twice as fast" is ambiguous.

5.20 Section 5.7, page 206, duality property: "$X(t) \leftrightarrow 2\pi x(-j\omega)$" is not well formed; the property is $X(jt) \leftrightarrow 2\pi x(-\omega)$. Similarly, in Example 5.16, $2\pi\delta(-j\omega)$ and $2\pi\delta(j\omega)$ should read $2\pi\delta(-\omega)$ and $2\pi\delta(\omega)$.

5.21 Page 207, Table 5.2, row 2: $2\pi\delta(j\omega)$ should read $2\pi\delta(\omega)$, as in row 6.

5.22 Section 5.7, page 208, convolution property: $Y(jw)$, $X(jw)$ and $H(jw)$ should read $Y(j\omega)$, $X(j\omega)$ and $H(j\omega)$.

5.23 Example 5.18, page 208: "the Fourier transform $H(j\omega)$ of its transfer function $h(t)$" should read "the Fourier transform $H(j\omega)$ of its impulse response $h(t)$". $H(j\omega)$ is the transfer function.

5.24 Page 209: "so that the product becomes absolutely integral" should read "absolutely integrable".

5.25 Page 210, after Equation 5.44: the integral is taken along the vertical line $\mathrm{Re}(s) = \sigma$, which is parallel to the imaginary axis, for a value of σ in the region of convergence.

5.26 Example 5.20, page 211: the result should read $u(t)e^{at} \leftrightarrow 1/(s-a)$, $\mathrm{Re}(s) > a$, to match the signal in the example.

5.27 Table 5.3, page 214, convolution row: $Y(s) = X_1(s)X(s)_2$ should read $Y(s) = X_1(s)X_2(s)$.

5.28 Table 5.3, page 214, final value theorem: the theorem holds only if $x(t)$ has a limit, that is, if all the poles of $sX(s)$ have negative real parts. It does not apply to marginally stable or unstable systems.

5.29 Example 5.22, page 215: "the Laplace transform $H(s)$ of its transfer function $h(t)$" should read "of its impulse response $h(t)$". $H(s)$ is the transfer function.

8

Elements of Control Theory

A computer network is an engineered system designed to meet certain design goals. Two such goals are **responsiveness** and **stability**. Responsiveness allows a system to continue to meet its performance goals despite changes in its operating environment, such as workload fluctuations or component failures. Stability prevents a system from failing or from being sent into uncontrollable oscillations due to certain inputs. Control theory provides powerful tools for building responsive and stable systems. For example, mathematically modeling the behavior of a controlled system demonstrates a fundamental contradiction between responsiveness and stability: The more responsive a system, the greater the chance that it will be unstable.

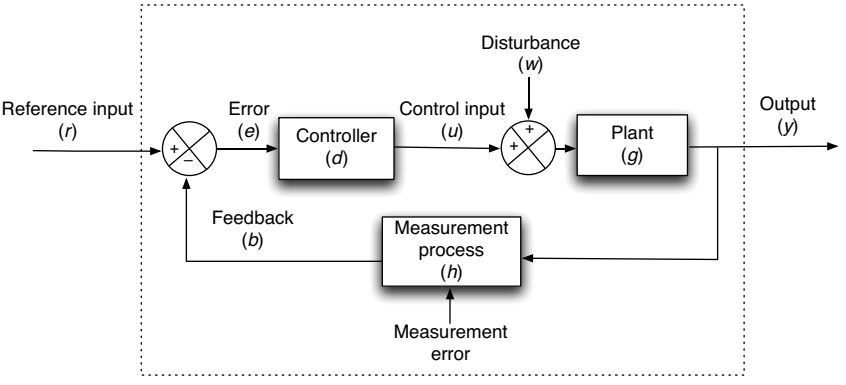
This chapter presents the elements of control theory. For simplicity, we consider primarily continuous-time linear control systems and classical control strategies. The concepts here, however, should provide a strong foundation for the interested student to delve further into this subject. Note that Chapter 5 is an essential prerequisite for understanding the material in this chapter.

We first describe a generic control system. We then characterize the behavior of generic first- and second-order systems, culminating with the design and analysis of classical control systems. We end with an introduction to "modern" control systems: state space representation and the principles of observability and controllability, and digital control.

Erratum 8.1: State-space and digital control are not in these notes.

8.1 Overview of a Controlled System

A generic feedback control system is shown in Figure 8.1. The system to be controlled is usually referred to as the **plant**. We assume that the plant and all control elements are continuous, linear, and time invariant. Given a control **input signal** u, the plant responds with an **output signal** y that is also additively influenced by an uncontrollable **disturbance signal** w. The control goal is to maintain the output of the plant at some **reference value** or **command** r despite disturbances. This is done using a **controller**, which compares the reference signal value with the **measured output signal** b, where this measurement itself may be subject to error. The controller chooses a **control signal** u such that the future output of the plant eventually matches the reference input, despite disturbances and measurement errors.



11:05

Figure 8.1 General model of a controlled system

EXAMPLE 8.1: A SIMPLE CONTROL SYSTEM

A thermostat-controlled home-heating furnace is an example of a simple control system. A user can use the control interface to specify a desired home temperature r. The controller compares this temperature with the temperature b measured using a sensor. If the difference (error e) is positive, the controller activates a heating furnace by sending it a control signal u, eventually raising the home's air temperature y to the desired reference value. At this point, the error drops to zero and the furnace is turned off.[1] A disturbance w, such as the

1. This example is highly simplified. Modern furnaces also have more sophisticated control elements, such as a controller that turns on the furnace slightly before it is needed, so that it heats air instead of accidentally cooling it, and a controller that uses measurements to predict how long a furnace needs to be on to heat the home by a certain amount.

inrush of cold outside air due to opening a door or a window, results in a drop of the measured temperature y. The controller compensates for this disturbance by turning on the furnace and eventually causing the output to return to the reference temperature.

Even this simple example demonstrates three lessons that apply to all control systems. First, immediately after the disturbance, a home's temperature could be far from the reference temperature. The presence of the controller does not guarantee that the home temperature will always be at the reference value, only that, in the absence of disturbances, this reference value will eventually be reached. If a series of disturbances come in quick succession, the system may never reach equilibrium.

Second, given that a system is currently not in equilibrium, it is up to the controller to decide how quickly to restore the system to this state. A system is said to have a high **gain**, or be **responsive**, if the controller tries to restore the system rapidly. Although this is usually desirable, a system with too high a gain can become unstable because in such a system, even a short delay in transferring measurements from the plant to the controller causes the controller to overreact to stale information, causing oscillations. In this example, a controller with too high a gain could result in a home whose temperature persistently oscillates from being too hot to being too cold. Choosing the degree of gain in the controller that balances responsiveness and stability is a critical issue in any control system.

Third, note that the controller simply cannot compensate for measurement errors. For example, if the temperature sensor always reads two degrees higher than the true temperature, the home's temperature will always be two degrees lower than the reference value. Therefore, in any feedback control system, it is important to use the best sensors possible.

EXAMPLE 8.2: CONTROLLING WEB SERVER LOAD

Consider a Web server that responds to GET and POST queries from Web browsers. The server uses a buffer to hold pending requests before they are served so that requests arriving to a busy server are not lost. Figure 8.2(a) shows the times at which requests arrive and depart from the buffer. Requests arrive at discrete time instants (shown on the X-axis) and are served after spending some time in the buffer. For example, request 4 arrives at time 5 and is served at time 7. The system is causal, so the departure curve is always at or below the arrival curve.

Although arrivals and departures are discrete events, it is convenient to model them using a continuous "fluid" approximation as shown in Figure 8.2(b).

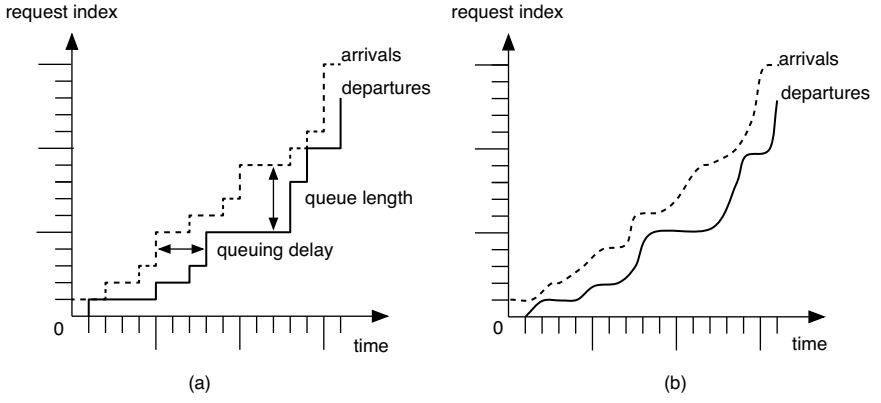


Figure 8.2 (a) Discrete arrivals and departures; (b) fluid approximation

Intuitively, this approximation is obtained by smoothing the arrival and departure curves, which removes abrupt transitions without introducing too much distortion. If queries arrive quickly enough, the fluid approximation has been found in practice to be reasonably accurate and, in any case, greatly simplifies mathematical analysis.

Note that the vertical distance between the arrival and departure curves at any point in time represents the number of queries in the buffer (the queue length) at that time. The control goal is to ensure that this quantity neither exceeds the buffer capacity—causing requests to be lost owing to buffer overflow—nor drops to zero—causing a buffer underflow and idling the Web server, unless, of course, there is no work to be done. Instead, we would like to keep the number of queries in the buffer at some reference value r that is small enough to not greatly impact the server response time but large enough that if the Web server were to suddenly obtain additional capacity, the buffer would not underflow. The greater the value of r, the higher the expected query service rate and the higher the query-processing delay.

The problem is that a client does not know the Web server's service rate, which depends on how many other clients are being served and how long it takes to satisfy a client request. We model this uncertainty by representing the Web server's service rate as a time-varying "disturbance" $w(t)$. The control goal is to manage the client request rate $u(t)$ to prevent buffer underflow and overflow. Specifically, we assume that the controller is given a buffer-occupancy reference value $r(t)$ and that it is instantaneously informed of the current number of bits in the buffer, denoted $y(t)$. The controller uses this feedback information to choose $u(t)$ such that $y(t)$ tracks $r(t)$ despite disturbance $w(t)$.

Note the similarity between this control system and a home thermostat. In both cases, the controller tries to keep output y at the reference level r despite disturbance w by comparing the measured state of the system with the desired reference value.

8.2 Modeling a System

Motivated by Example 8.2, we turn our attention to the process of mathematically modeling a system, which can be viewed either as the plant or as the controlled system shown in the dotted box in Figure 8.1. Both the plant and the controlled system have an input and an output, where the output depends on the input. By using identical approaches to model both the *open* uncontrolled system and the *closed* controlled system, we can recursively control a complex system. That is, we can model and control one aspect of a plant, then represent this controlled system as black box when modeling the remainder of the system. This approach is necessary to control a complex system. We will study some practical problems using this approach in Section 8.7.

Choosing which aspects of a system should be modeled and which aspects can be ignored is an art rather than a science. Developing a system model that is mathematically tractable and yet represents all the relevant aspects of the system will usually require several rounds of refinement. We first outline an approach that has been found to work well in practice and then focus on three mathematically equivalent representations of a system.

8.2.1 Modeling Approach

Start by understanding the working of the system and its component parts. Determine the physical capabilities of each component, its limits of operation, and the operating regimes where it is linear or nonlinear. Components in a control system can usually be described in terms of the concepts of **effort** and **flow**. Effort represents a system input that causes a flow. The product of effort and flow is **power**, and the total power expended over some time duration is called **energy**. For example, in a mechanical system, forces correspond to effort, and velocities correspond to flow. The product of force and velocity is mechanical power, and the integral of power over time is the amount of mechanical energy used. In an electrical system, voltages correspond to effort and currents to flow. Their product is electrical power, whose integral is electrical energy. Similarly, in a computer network, the quantity of data that needs to be sent from a source to a destination represents effort, and the rate at which data is sent corresponds to flow. Although they can be analogously

defined, the concepts of power and energy are not commonly used in computer networks.

System components can be either active or passive. Active elements generate power (not energy!) that flows through the system. The dynamic operation of the system is essentially an attempt to redistribute this power over time. In contrast, passive elements store and dissipate energy but do not generate power. In an electrical system, a voltage source, such as a battery, is an active element that drives the system, whereas capacitors and inductors are passive elements that store and dissipate energy. Similarly, a data source is an active component that generates effort that causes a flow on links and into routers. In contrast, a buffer is a passive component that stores and releases energy, like a capacitor.

The next step is to capture the laws, such as conservation, that constrain the operation of the system. For example, in a mechanical system, Newton's laws form the basic equilibrium constraints. In a computer network, the rate at which a buffer stores data is the difference between the data arrival and departure rates. Similarly, during the slow-start phase of a TCP connection, the rate of transmission grows exponentially. These constitute the mathematical laws of operation of the corresponding components.

Finally, state the intended goal of the control system in terms of the ideal equilibrium operating point (which, by convention, is defined to be zero). If possible, quantify the permissible operating error. This makes it possible to determine the degree to which a chosen control law meets requirements.

At the end of the modeling process, it should be possible to capture the operation of the system in a handful of equations. We will see some concrete examples of the modeling process shortly.

8.2.2 Mathematical Representation

Three common ways to mathematically represent a system (i.e., either the plant or the dotted box in Figure 8.1) are by the **state variable**, the **impulse response**, and the **transfer function** representations. Our discussion assumes that the system is **linear** and **time invariant** (**LTI**). Recall that a time-invariant system is one whose parameters do not change with time: If the system has an output $y(t)$ at time t for an input $u(t)$, it has the output $y(t - T)$ for an input $u(t - T)$. That is, if $y(t) = G(u(t))$, then $y(t - T) = G(u(t - T))$. Recall also that a linear system is one that exhibits **superposition**: If the input u_1 leads to output y_1 and input u_2 leads to output y_2, then for all constants k_1 and k_2, the input $(k_1u_1 + k_2u_2)$ leads to the output $(k_1y_1 + k_2y_2)$. That is, if $y_1 = G(u_1)$ and $y_2 = G(u_2)$, then $k_1y_1 + k_2y_2 = G(k_1u_1 + k_2u_2)$. For an LTI system, it is always possible to choose the initial time to be $t = 0$. We assume the system to be causal, so the input signal is zero for $t < 0$.

Impulse Response Model. Many practical systems of interest have a single scalar input u. In this case, it is possible to represent the system by its **impulse response**. Recall from Section 5.4.3 that if we know the response $g(t)$ of an LTI system to a Dirac delta input, we can compute its response to *any* input $u(t)$ by convolving the input signal $u(t)$ with the impulse response $g(t)$. When using this model, we assume that the system is in equilibrium at time t=0, or $\boldsymbol{x}(0) = \mathbf{0}$. Also, if we are modeling the entire system (not just the plant), the input signal should incorporate the disturbance. The output $y(t)$ is given by

Erratum 8.2: Equivalently, all initial conditions are 0.

$$y(t) = \int_0^t u(\tau)g(t-\tau)d\tau = u(t) \otimes g(t) \qquad \textbf{(EQ 8.5)}$$

The impulse response model is rarely used in practice: Its main purpose is as a step in the derivation of the *transfer function model*, which we describe next.

Transfer Function Model. Consider a system at equilibrium at time 0 that is described by the impulse response function $g(t)$, so that the output $y(t)$ is given by Equation 8.5. By taking the Laplace transform of both sides we get:

9:45

$$Y(s) = U(s)G(s) \qquad \textbf{(EQ 8.6)}$$

In other words, we obtain the Laplace transform of the output by multiplying the Laplace transform of the input (which, if necessary, incorporates a disturbance input) with the Laplace transform of the impulse response, rather than computing a convolution, as is required with the impulse response function. This simple form of the output makes it convenient to represent a single-input single-output causal system by the Laplace transform of its impulse response, which we call its **transfer function**.

The transfer function of many common systems takes the form $N(s)/D(s)$, where $N(s)$ and $D(s)$ are polynomials in s. The highest power of s in $D(s)$ is called the **order** of the transfer function. Consider a denominator polynomial $D(s)$ of order m. The m roots of $D(s)$, that is, the m values of s for which $D(s)$ is zero, denoted $\alpha_1, \alpha_2, \ldots, \alpha_m$, are the values of s for which $Y(s)$ is undefined, and these are called

the **poles** of the system. It turns out that these poles have a deep significance because, using the technique of partial fraction expansion discussed in Section 8.11, $Y(s)$ can be written as a sum of terms of the form $\frac{A}{(s-\alpha)^r}$. Referring to row 6 of Table 5.4 in Section 5.8 and recalling that the Laplace transform is linear, we see that the corresponding term in $y(t)$, obtained by taking the inverse Laplace transform, is $A\frac{t^{r-1}}{(r-1)!}e^{\alpha t}$. Therefore, if the pole is negative, the corresponding term in the output decays with time; if, on the other hand, it is positive, the output grows with time. We like a system to be **stable**, which informally means that for a bounded input, the output also should be bounded. This is possible only if all system poles lie in the left half of the complex s plane. (We discuss this notion in more detail in Section 8.8.) Thus, the nature of $D(s)$ allows us to quickly determine the overall behavior of the system, reinforcing the usefulness of the transfer function system model.

EXAMPLE 8.4: TRANSFER FUNCTION MODEL

We now derive the transfer function model of the Web server system. Recall that the state space representation of the system is

Erratum 8.3: For consistency, read $\dot{x} = u + w$; see correction 8.3.

$$\dot{x} = u - w$$

This is a single-input single-output system that is initially at equilibrium, so that $y(0) = 0$. Therefore, we can use a transfer function to model it. The output y is the same as the state variable x. To incorporate the disturbance into the input signal, we write

$$v = u - w$$
$$\dot{x} = v$$

We obtain the transfer function by taking the unilateral Laplace transform of both sides:

$$sY(s) - y(0) = V(s)$$
$$Y(s) = V(s)\left(\frac{1}{s}\right) + \frac{y(0)}{s}$$

so that

$$G(s) = \frac{1}{s}$$

is the required transfer function model of the system.

"Modern" versus "Classical" Control. The classical approach to control uses transfer function models. Although the results are intuitive, this approach does not permit the modeling of multiple-input multiple-output and nonlinear systems. The "modern" approach, instead, uses the state variable model, which easily generalizes to these domains and can also draw on linear algebra for compact notation and a deep theoretical basis. In keeping with our goal of presenting intuition rather than formalisms, we will use the classical approach, with a brief treatment of the state variable approach in Section 8.8.

Erratum 8.4: Read: Section 8.9.

8.5 Basics of Feedback Control



12:58

Lab: Feedback control

We are now in a position, finally, to investigate the basics of feedback control. We use the transfer function model to study a simple feedback system. We also investigate the goals of a control system and learn how these goals can be achieved using feedback control.

Recall the abstract model of a continuous, linear, time-invariant control system in Figure 8.1. Assume that all inputs and outputs are scalar. Then, we can use transfer functions to model the behavior of each of the control blocks. That is, if the input to a control block G is the signal $u(t)$ and its output is the signal $y(t)$, we can model the control block by using the transfer function $G(s)$, so that the Laplace transform of the output, $Y(s)$, is merely the product of the transfer function and the Laplace transfer of the input $U(s)$:

Erratum 8.8: Read "Laplace transform".

$$Y(s) = G(s)U(s)$$

This allows us to redraw Figure 8.1, replacing each control block by its corresponding transfer function and each signal by its Laplace transform, as shown in Figure 8.7.

The use of the transfer function model allows us to state the system equations as follows (we omit the argument s for clarity):

$$Y = GU + GW \quad \textbf{(EQ 8.14)}$$

$$E = R - B = R - HY \quad \textbf{(EQ 8.15)}$$

$$U = DE = D(R - HY) \quad \textbf{(EQ 8.16)}$$

From Equations 8.14, 8.15, and 8.16, we can write

$$Y = GDR - GDHY + GW$$

$$Y(1 + GDH) = GDR + GW$$

$$Y = \left(\frac{GD}{1 + GDH}\right)R + \left(\frac{G}{1 + GDH}\right)W \quad \textbf{(EQ 8.17)}$$

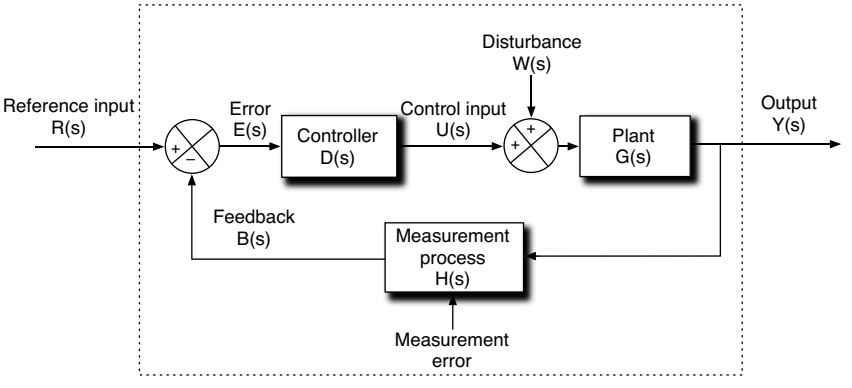
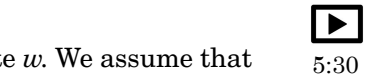


Figure 8.7 Transfer function model of a single-input single-output linear time-invariant feedback control system

Equation 8.17 is the fundamental equation of a feedback control system. It shows that the output is a linear combination of two terms, one arising from the Laplace transform of the reference signal $R(s)$ and the other from the Laplace transform of the disturbance $W(s)$. Note that the only variable that can be modified by a system designer is D, which represents the controller. A common situation is that the measurement process simply feeds the output back to the input, so that $H(s) = 1$: Such a system is called a **unity feedback system**.

We now use this equation to investigate the goals of a control system and how they are achieved by a feedback control system. Essentially, we will compare the controlled system whose output is given by Equation 8.17 with a system with no feedback (i.e., $H = 0$), whose output is given by $Y = G(DR+W)$.

8.5.1 System Goal


5:30

The primary goal of any controller is to ensure that $y = r$ despite w. We assume that we can design the controller, and therefore choose D, more or less as we wish, within some limits, to meet this goal.

It is illuminating to consider how this system goal is achieved in the context of three systems: an uncontrolled system, a controlled system with no feedback, and a system with feedback control. The output of an uncontrolled system is affected by disturbances, and so the control goal is achieved only if $w = 0$. In contrast, a controller that lacks feedback can compensate for a disturbance by predicting it. If it can predict disturbances accurately, it can achieve the system goal. For instance, suppose that an oven temperature controller knows that every time the oven door is opened, a certain amount of heat is lost, and so the heating element needs to be turned on for a certain duration. This controller could maintain oven temperature despite a disturbance created by a door opening by sensing the state of the oven door. In contrast, a feedback control system measures the effect of a disturbance on the output (e.g., the effect of a door opening on the oven temperature) and can use the system-transfer function to determine the appropriate input that removes the effect of the disturbance.

In many situations, the command r is held constant for a reasonably long duration of time. In this case, the goal of a controller is to ensure, first, that the output quickly reaches r and second, that the output stays at r despite disturbances. The time taken for the output to get close to r from its initial state is called the **settling time**. The discrepancy between the actual output and the reference input after the settling time is past is called the **steady state error**.

8.5.2 Constraints

3:46

In achieving the system goal, a controller must take the following constraints into account. The constraints are *stability, disturbance rejection, transient response, linearity, measurement error,* and *robustness.* We consider each in turn.

Stability. Feedback control systems can become unstable if they are improperly designed. Stability essentially means that the output returns to the desired value chosen by a command input despite small variations in the control input or the disturbance. Two widely used stability criteria are **bounded-input**, **bounded-output (BIBO) stability** and **zero-input** stability.

- A system is BIBO stable if, given that the system is initially in equilibrium, any input with bounded magnitude results in an output that also is bounded, although the system may potentially oscillate indefinitely within some bound.
- A system is zero-input stable if, given that the input is zero, all initial states result in the system's eventually reaching the zero equilibrium state.

Erratum 8.9: Delete this clause; see correction 8.9.

EXAMPLE 8.7: STABILITY

The Web server system of Example 8.2 is BIBO stable if, starting with a zero initial queue size, the request queue size does not exceed some bound for all request arrival streams. The system is zero-input stable if, given that the initial queue length is finite, as long as no more requests are accepted into the queue, the queue size eventually drops to zero.

Feedback control systems typically achieve stability by ensuring that the control does not overreact to a disturbance. We study this in more detail in Section 8.8.

Disturbance Rejection. A control system should reject any changes in the output owing to a disturbance. From Equation 8.17, we see that the noise input W is attenuated by a factor of $\frac{G}{1+GDH}$. In contrast, an uncontrolled system or a system without feedback control is subjected to the entire disturbance. This shows the benefit of feedback in rejecting disturbance.

Erratum 8.10: Feedback attenuates W by $1/(1+GDH)$.

Transient Response. When a system in its steady state is subjected to a sudden disturbance, it takes a while for the system to return to equilibrium. This is like a jolt to the handlebar of a bicycle: It takes some time for the bicycle to return to its steady path. The behavior of the system during this time is called its **transient**

6:12

response. If a control system is not properly designed, a sharp disturbance can result in the system's becoming unstable or collapsing.

A feedback control system gives designers a control "knob" to shape a system's dynamic response. To see this, we write Equation 8.17 as

$$Y = \frac{GDR + GW}{1 + GDH}$$

Note that the behavior of $y(t)$ is given by the inverse Laplace transform of this expression, which in turn depends on the roots of the denominator. For instance, if all the roots of the denominator have a strictly negative real component, then $y(t)$ is the sum of decaying exponentials so that the transient response eventually dies down. A designer, therefore, can choose D so that the denominator polynomial has this behavior. This intuition lies at the heart of **pole placement**.

Linearity. A control system designed assuming that the system is LTI will not function correctly when the system is driven into a range where it is nonlinear. Therefore, we must be careful to ensure that the system always stays in the linear range and that, if a disturbance pushes it into a nonlinear regime, the control input brings the system back into a linear regime. In a feedback control system, the system is likely to continue to be linear when the feedback error is small. A designer may need to take special measures to deal with the situation when the feedback error is large. For example, in a computer network, when the network is congested, packet losses from overflowing buffers make the system nonlinear. One reasonable control rule is for sources to drastically reduce their data-transmission rates in the presence of packet loss to restore normal operation. (In practice, a TCP source can be viewed as intentionally placing the system into near-congestion to maximize network utilization.)

EXAMPLE 8.8: LINEARITY

Consider the Web server of Example 8.2. This system is linear as long as the request queue is neither overflowing nor underflowing. If the queue is, for example, overflowing, the error term will be large. In this case, a designer could add a special control rule to set the input rate u to 0, so that the queue drains as quickly as possible and the system reenters the linear regime.

Measurement Error. A feedback system measures the output by using a sensing or measurement device. Importantly, the feedback system *cannot* compensate for an error in this sensor. For instance, if a thermostat sensor is off by 1 degree, the controlled system has a steady state error of 1 degree. Therefore, it is critical that the measurement process be as error-free as possible.

A second form of measurement error arises from delays in the measurement process. Consider a thermostat that takes 5 minutes to measure the room temperature. In such a system, once the setpoint is reached, the furnace would be left on 5 minutes too long, raising the room temperature beyond the setpoint. At this point, the furnace would be turned off, and the room temperature would return to the setpoint, then drop below it for at least 5 minutes before corrective action could be taken. It is clear that the delayed measurement results in a persistent oscillation in the output signal (i.e., the room temperature). In general, feedback delays can cause oscillations or even instability. The general approach to deal with such delays is to be cautious in responding to a measurement, recognizing that it may reflect stale input. This can be achieved by damping the control input. We will consider dealing with feedback delays more fully in Section 8.7.

Robustness. A controller is designed subject to many assumptions: that the system is linear, that the system model is accurate, that the measurement is correct, and that the measurement is not subjected to too much delay. Moreover, we must also assume that the system behavior does not change over time, owing to wear and tear or software upgrades. These are strong assumptions. Therefore, it is necessary to design a controller that does not exhibit poor behavior when one or more of these assumptions are violated. The theory of **robust control** deals with the design of control systems that work well despite bounded variations in the system parameters and is described in more advanced textbooks on control theory.

8.6 PID Control



6:04

Lab: PID control

Proportional-integral-derivative (PID) control, the simplest classical approach to controller design, refers to controllers whose control input is proportional to the error, to the integral of the error, or to the derivative of the error. PID controllers may use two or even three modes simultaneously. We study each type of controller in turn. Note that in all three modes a higher loop gain improves responsiveness.

8.6.1 Proportional-Mode Control

With proportional control, the control signal is a scalar multiple of the feedback error, so that $u = K_p e$ or, in the transform domain, $U = K_p E$, where K_p is called the **loop gain**.[2] The greater the loop gain, the larger the control action corresponding to a disturbance. Therefore, a large loop gain makes the controller more responsive.

2. This is not the same as the system gain. The loop gain is the ratio of the command to the error; the system gain is the ratio of the output to the input, which, in the case of a perfectly controlled system, should be 1.

Erratum 8.11: Loop gain = control input / error.

Note that the control input is proportional to the error, so that, if the system in steady state needs a nonzero control input, there needs to be a nonzero steady state error, also called the **droop**. To keep the steady state error small, we need to make the loop gain large. However, this reduces damping and can make the system unstable. (We will return to this topic when we study system stability in Section 8.8.)

EXAMPLE 8.9: PROPORTIONAL-MODE CONTROL

8:24

Consider the design of a proportional controller for the Web server in Example 8.2. Recall that its transfer function $G(s)$ is given by $\frac{1}{s}$. Suppose that the command r is the desired setpoint of the request buffer. Then, the error is given by $r - y$. For proportional control, the control input is proportional to this error.[3] Specifically, the controller chooses an input u such that, if there is no further disturbance, the system output y returns to the desired setpoint r after a time τ. That is,

$$u = \frac{r-y}{\tau},$$

so that the loop gain is $\frac{1}{\tau}$. Recall that $Y = \frac{1}{s}(U-W) = \frac{1}{s}\left(\left(\frac{R}{\tau} - \frac{Y}{\tau}\right) - W\right)$, which can be rearranged to get

Erratum 8.12: Read $Y = \frac{1}{s}(U + W)$; see correction 8.12.

$$Y = \frac{R}{1+s\tau} - \frac{W\tau}{1+s\tau}$$

To study the step response of this controller, set $W = 0$ and $R = 1/s$. Then, $Y = \frac{1}{s(1+s\tau)}$. Using partial fractions to expand this,

$$Y = \frac{1}{s} - \frac{1}{s+\frac{1}{\tau}},$$

and taking the inverse Laplace transform, we find $y(t) = 1 - e^{-t/\tau}$. That is, when there is no disturbance, if the input command is a step signal, the output is a step signal (i.e., the desired value because $y = r = 1$) along with a transient response that decays exponentially over time. The smaller the value of τ, the larger the loop gain and the faster the decay of the transient response.

3. If the error is negative (that is, the buffer occupancy is greater than the setpoint), the controller must send "negative" packets into the buffer. For simplicity, we interpret this to mean that the controller sends special "cancel" requests that have the effect of removing pending requests from the Web server's request buffer.

We investigate the steady state error in the step response (assuming zero disturbance) by using the final value theorem, which states that

$$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s).$$

Thus,

$$\lim_{t \to \infty} e(t) = \lim_{s \to 0} sE = \lim_{s \to 0} s\left(\frac{1}{s} - Y\right) = \lim_{s \to 0} s\left(\frac{1}{s} - \frac{1}{s} + \frac{1}{s + \frac{1}{\tau}}\right) = 0$$

This shows that, independent of the loop gain, the steady state error is zero.

8.6.2 Integral-Mode Control

6:34

The motivation for integral control is the intuition that the magnitude of the control input should depend not only on the current value of the error but also the total error built up in the past: The greater this overall accumulated error, the greater the correction. Mathematically, with integral control, $u = K_i \int e$ or $U = K_i \frac{E}{s}$, where K_i is the loop gain. Integral control removes the need for a steady state error term but usually makes the control less stable, as the next example demonstrates.

EXAMPLE 8.10: INTEGRAL-MODE CONTROL

Consider the design of an integral controller for the Web server in Example 8.2 with transfer function $G(s) = \frac{1}{s}$. For integral control, the control input u is proportional to the integral of the error: $e = (r - y)$, so $u = K_i \int (r - y)$ and

$$U = \frac{K_i(R - Y)}{s},$$

where the loop gain is K_i. Now, $Y = \frac{1}{s}(U - W) = \frac{1}{s}\left(\left(\frac{K_i(R - Y)}{s}\right) - W\right)$, which can be rearranged to get

Erratum 8.13: Read $Y = \frac{1}{s}(U + W)$; see correction 8.13.

$$Y = \left(\frac{K_i}{s^2 + K_i}\right)R - \left(\frac{s}{s^2 + K_i}\right)W$$

To study the step response of this controller, set $W = 0$ and $R = 1/s$. Then,

1:57

$$Y = \frac{K_i}{s(s^2 + K_i)} = \frac{1}{s} - \frac{s}{s^2 + K_i}.$$

Taking the inverse Laplace transform, we find $y(t) = (1 - \cos\sqrt{K_i}t)$. That is, when there is no disturbance, if the input command is a step signal, the output is the step signal along with a persistent sinusoidal oscillation with magnitude 1 and period $\frac{1}{\sqrt{K_i}}$.

Erratum 8.14: The period is $2\pi/\sqrt{K_i}$.

The transient response neither grows nor decays with time: We call a system with such a response **marginally stable**. This demonstrates our claim that the introduction of integral-mode control diminishes system stability. Note that the larger the loop gain, the faster the oscillation frequency.

8.6.3 Derivative-Mode Control

7:37

Derivative-mode control recognizes the fact that control actions (for the same degree of error) should differ depending on whether the error trend is increasing or decreasing. If the error trend (its derivative) is increasing, the controller needs to take stronger corrective action, but when the error trend is decreasing, the controller must back off the control action to prevent overreaction. Thus, derivative-mode control tends to dampen the actions of a controller.

Derivative-mode control cannot be used in isolation, because a constant steady state error, no matter how large, has a zero derivative. To avoid steady state error, derivative-mode control is combined with either proportional-mode or both integral- and proportional-mode control.

Note also that derivative-mode control is very sensitive to high-frequency noise in the measurement process: Spikes in measurement noise are interpreted as a sharp increase in system error. To avoid this problem in practice, measurements must be filtered using a low-pass filter before using them as the basis of derivative-mode control.

Mathematically, with derivative-mode control, $u = K_d\left(\frac{de}{dt}\right)$ or $U = K_d E s$, where K_d is the loop gain. We demonstrate derivative-mode control through an example.

EXAMPLE 8.11: DERIVATIVE-MODE CONTROL

Consider a derivative controller for the Web server in Example 8.2 with transfer function $G(s) = \frac{1}{s}$. For derivative control, the control input is proportional to the derivative of the error, so that $U = K_d(R - Y)s$, where the loop gain is K_d. Now, $Y = \frac{1}{s}(U - W) = \frac{1}{s}((K_d(R - Y)s) - W)$, which can be rearranged to get

Erratum 8.15: Read $Y = \frac{1}{s}(U + W)$; see correction 8.15.

$$Y = \left(\frac{K_d}{1 + K_d}\right)R - \left(\frac{1}{s(1 + K_d)}\right)W$$

To study the step response of this controller, we set $W = 0$ and $R = 1/s$. Then, $Y = \frac{K_d}{s(1 + K_d)}$. Taking the inverse Laplace transform, we find $y(t) = \frac{K_d}{1 + K_d}$. That is, when there is no disturbance, if the input command is a step signal, the output is an attenuated step signal with no transient response. Note that the larger the loop gain, the smaller the attenuation in the input signal.

We can investigate the steady state error in the step response (assuming zero disturbance) by using the final-value theorem. We have $Y = \frac{K_d}{s(1 + K_d)}$, so from the final-value theorem, the asymptotic value of $y(t)$ is $\frac{K_d}{(1 + K_d)} \neq 1$, which demonstrates our earlier claim that derivative-mode control results in a nonzero steady state error. However, note that the steady state error is a decreasing function of the loop gain, so the steady-state error can be reduced by increasing the loop gain.

8.6.4 Combining Modes

Each of the three basic control modes—proportional, integral, and derivative—has its advantages and disadvantages. Therefore, it is common to combine modes in a single so-called **PID controller**. Such a controller has three control parameters that correspond to loop gains in each mode. That is, for an error E, we have

$$U = \left(K_p + \frac{K_i}{s} + K_d s\right)E$$
$$D = \left(\frac{K_d s^2 + K_p s + K_i}{s}\right) \quad \textbf{(EQ 8.18)}$$

The relative contributions of the three modes can be adjusted using the gain factors. Intuitively, the use of proportional mode corrects for error, the use of integral mode removes steady state error, and the use of derivative mode corrects for instability that may be produced by the integral mode and dampens the control action.

We will not be covering Section 8.7, but I've left it in, in case anyone is interested.

8.7 Advanced Control Concepts

Having studied the basic PID control, we now examine, at an intuitive level, some more advanced control concepts that are applicable to control systems in computer networks.

8.7.1 Cascade Control

When trying to accomplish a complex control objective for a plant or process that exhibits multiple timescales of behavior, it is often useful to layer control. For example, with two layers of control, a lower-layer controller, operating at a faster timescale, controls a portion of the system to achieve a desired setpoint, and the higher-layer controller, operating at a slower timescale, dynamically chooses the setpoint to achieve the top-level control objective. This is called **cascade control**.

EXAMPLE 8.12: CASCADE CONTROL

Consider the use of cascade control to optimize the communication between a Web browser client and a Web server. A Web browser can have multiple connections open to multiple Web servers, where each connection queries for and fetches a set of data objects to be rendered on the screen.[4] Ideally, the connections should make progress at rates proportional to the amount of data being transferred over the connection, so that all the elements of the Web page can be rendered at the same time.

Recall that query-request rate control operates at the tens to hundreds of milliseconds timescale and adjusts the sending rate from a source to ensure that the buffer-occupancy level at the Web server is close to a desired setpoint. Recall from Example 8.2 that the greater this setpoint, the greater the achievable end-to-end throughput. This suggests the use of a higher-layer controller that dynamically modifies the setpoint of the buffer-occupancy level to achieve the overall control objective. Specifically, the higher-layer controller has, as its

4. As a simplification, we assume that all queries are of equal size and take equal amounts of time to process, so that we need to control only the number of queries, independent of their size. Generalizing from this simplified case to the true situation is left as an exercise to the reader.

control goal, the desire to have all connections make equal fractional progress on their queries. The error term for this controller is the difference between the actual fractional progress made by a connection and the mean fractional progress made by all the clients. If this error term is negative, the connection is a lagging connection, and its setpoint is increased using proportional-, integral-, or differential-mode control. On the other hand, if this error term is positive, the connection is a leading connection, and its setpoint is similarly decreased.

To make this example concrete, consider a browser that has three connections to three different Web servers, where the first connection fetches three query responses, the second fetches five query responses, and the third fetches ten responses. Therefore, they should make progress at the relative rates of 3:5:10. Suppose that at some point in time, the higher-layer controller finds that the first connection has retrieved two responses, the second has retrieved three responses, and the third has retrieved five responses. The fractional progress made, therefore, is 2/3:3/5:5/10 = 0.66 : 0.6 : 0.5. The mean progress is 0.59. Therefore, the higher-layer controller decreases the setpoint of the first connection, leaves the second connection's setpoint nearly unchanged, and increases the setpoint of the third connection.

8.7.2 Control Delay

Our discussion so far has assumed that the measurement of the plant output is instantaneously available to the controller. This is, of course, almost never the case. Delays in measurement are particularly challenging for controllers developed for computer networks, where measurement delays can be significant. We first describe the mathematical modeling of measurement delay and then describe the effects of delay on control.

Measurement delays are easy to represent mathematically. Recall from Table 5.3 in Section 5.8 that a delay of τ seconds corresponds to multiplying the Laplace transform of a signal by $e^{-s\tau}$. Therefore, with feedback delay, we find that the error E is given by

$$E = R - e^{-s\tau}Y \quad \textbf{(EQ 8.19)}$$

We can then use this input as the basis for, say, proportional control, where $U = K_pE = K_p(R - e^{-s\tau}Y)$. The subsequent analysis of this system proceeds as before, except that going from the Laplace domain to the time domain is complicated because we cannot use partial fraction expansion, which applies only to ratios of polynomials. Instead, we can use the **Pade approximations** of the exponential function:

$$e^{-s\tau} \approx \frac{1+\frac{(-s\tau)}{2}}{1-\frac{(-s\tau)}{2}}$$

$$\approx \frac{1+\frac{(-s\tau)}{2}+\frac{(-s\tau)^2}{12}}{1-\frac{(-s\tau)}{2}+\frac{(-s\tau)^2}{12}}$$

which replace the exponential with increasingly accurate polynomial ratios. More elaborate Pade approximations can be found in the literature.

Erratum 8.16: Read "Padé".

The general treatment of control systems with delays is complex and beyond the scope of this book. However, two observations are relevant. First, a practical approach to deal with delays is to predict the current system state from past measurements, using an appropriate predictor. Once this is done, the problem reduces to the standard zero-delay control system. Many sophisticated techniques for time-series prediction can be used for this purpose. Second, the general effect of delays is that the controller does not know the response to its prior control action. To prevent oscillations and instability, the controller should choose to be cautious in its actions, delaying its next control action until it has learned the response to its prior action. This leads to a fundamental trade-off between responsiveness and stability. In most practical systems, stability is more important and comes at the cost of decreased responsiveness. For example, in TCP, to dampen the control input, control actions are taken no faster than once per round trip time, and the response to a loss event is artificially delayed by up to half a second. This reduces oscillations, though it also reduces responsiveness.

EXAMPLE 8.13: EFFECT OF CONTROL DELAY

Consider a transport-layer flow-control algorithm that takes control actions in one of two ways (Figure 8.8). If it receives an acknowledgment with the explicit forward congestion notification (EFCN) bit set to zero, indicating that the corresponding data packet did not experience congestion, it increases its sending rate additively; otherwise, it decreases its sending rate multiplicatively. This is a controller with an undamped response. In contrast, consider a flow-control algorithm that collects the EFCN bits set on packets sent during each window of duration one round trip time (measured from the start of the connection), then takes control actions once per round trip time, depending on whether the majority of the bits were set. This controller dampens its response.

Let us do a thought experiment to study the behavior of the two systems. To exaggerate the difference in their behavior, we will assume that the source has

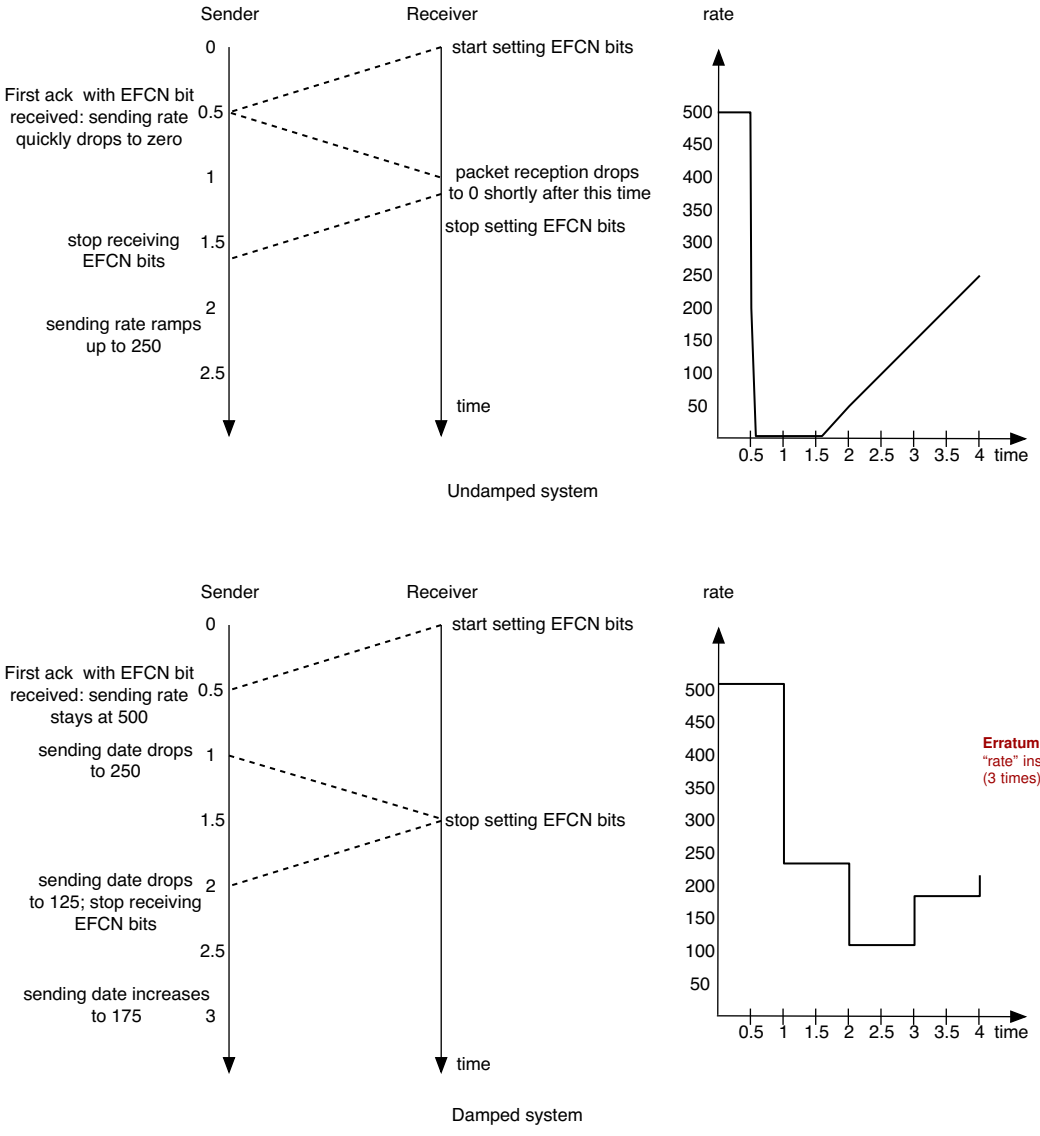


Figure 8.8 Figure for Example 8.13

Erratum 8.17: Read "rate" instead of "date" (3 times).

an infinite number of packets to send and that its fastest sending rate is 1,000 packets per second. Assume also that the delay from the time that an EFCN bit is set to the time that it is received by a sender is 0.5 seconds and that the end-to-end delay is also 0.5 seconds (i.e., the single bottleneck link is next to the receiver), so that the round trip time (RTT) delay is 1 second. Finally, assume that the capacity of the bottleneck link is 500 packets per second, the

additive increase factor is 50 packets/RTT, and that the multiplicative decrease factor is 0.5.

We conveniently define the system to be in equilibrium when the source is sending packets at 500 packets/second and the bottleneck link is serving packets at 500 packets/second. Consider the time evolution of control for the damped and undamped controllers when the system is initially at equilibrium and when, at time 0, the bottleneck service rate decreases abruptly to 250 packets/second, owing to the arrival of another flow.

With both the undamped and the damped controllers, packets with EFCN bits arrive to the sender at time 0.5. With the undamped system, on the arrival of the first acknowledgment packet with an EFCN bit set, the source immediately reduces its sending rate to 250 packets/second, further reducing its rate with every additional acknowledgment. Note that in equilibrium, the source has 500 packets outstanding, all with bits set. So the source reduces its rate to nearly 0 by time 1.5, when the last of the EFCN bits is received. Then, no more bits are set, so the source increases its rate by 50 packets/RTT, reaching its new equilibrium value of 250 at time 6.5. The subsequent time evolution is straightforward. Note that the effect of this immediate response is to quickly drain the bottleneck queue of any packets that are buffered there owing to the drop in the link-service rate.

Erratum 8.18: Time axis of the upper plot in Figure 8.8 is incorrect; see correction 8.18.

With the damped system, the source continues to send at the rate of 500 packets/second until time 1 because it takes control actions only at times 1, 2, 3, and so on. At time 1, it has received 500 acknowledgments, of which 250 have bits set and 250 do not. Assume that, in this borderline case, the source reduces its rate to 250 packets/second at time 1. In the time interval from 1 to 2 seconds, it sends 250 packets but receives 500 packets with EFCN bits set (left over from the first round trip time). So, it reduces its rate once more to 125 packets/second. No more bits are set subsequently, and the source ramps up its rate to 250 over the next 3 seconds. Note that, owing to the delayed response, the bottleneck queue is drained much more slowly than with the undamped system.

Damping decreases responsiveness, so the effect of a disturbance persists longer, but the magnitude of the impulse response diminishes, and the system returns to equilibrium sooner. This is characteristic of any control system with control delays.

8.8 Stability

Lab: Stability

Consider a system that is not subject to a disturbance for some length of time. We expect such a system to eventually settle down so that its output y matches the


12:43

command input r. A system whose output persists at its setpoint is said to be in **equilibrium**, or **relaxed**. By convention, we denote this state—and the corresponding command and output—by a **zero** vector. In many cases, this ideal equilibrium state may never be achieved. Yet it is important because it is the state that a system *strives* to achieve.

Suppose that a system in equilibrium is subjected to a time-limited disturbance. Ideally, if the disturbance is small, the change in the output also should be small. Moreover, the system should rapidly return to equilibrium when the disturbance ceases. We call such an ideal system a **stable** system.

To further grasp the concept of stability, imagine a cone that is lying on its side and another that is balanced on its point. A small disturbance to a cone balanced on its point results in a large change (it falls down); when the disturbance is removed, the cone does not return to its previous balance. In contrast, a small disturbance to a cone lying on its side results in a small change in position, although it may not return to the original equilibrium state when the disturbance is removed.

Besides these two responses to a small disturbance, a third possible response is for the system to oscillate, where the magnitude of the oscillations decays over time. This is akin to a rocking chair or a guitar string subjected to a small impulse disturbance. Depending on the desired output behavior of the system, such a response could be characterized as either stable or unstable.

When designing a controller, it is very desirable that the controlled system be stable. In this section, we study the conditions under which a controlled system can be guaranteed to be stable. Before doing so, it is worth distinguishing between two types of stable responses.

1. A system is said to be **bounded-input**, **bounded-output** (**BIBO**) stable if the output of the system is bounded in magnitude by some value M as long as the disturbance is bounded in magnitude by m.
2. A system is said to be **zero-input** stable if, for all initial states of the system, including nonequilibrium states, when both the command and the disturbance inputs are zero, the system eventually reaches the equilibrium state.

Erratum 8.19: Read: for *every* bounded input; see correction 8.19.

Both BIBO and zero-input stability are relatively easy to characterize for LTI systems. However, for time-varying or nonlinear systems, a system may be stable for some classes of disturbances but not others, or may be stable for small disturbances but not large ones. Such systems may also have multiple equilibrium states and may move from one equilibrium state to another when subject to small disturbance, much like a round pencil lying on its side when subjected to a small torque. Such multiple equilibria arise naturally in computer networks.

EXAMPLE 8.14: METASTABILITY IN THE TELEPHONE NETWORK

The telephone network core is organized as a clique where traffic from any node to any other node follows a one-hop path. This routing is simple but inefficient, in that the one-hop path does not make use of capacity available on two-hop paths. Consider a clique where all links have unit capacity so that the one-hop path from any node A to any other node B has capacity 1. If the clique has n nodes, there are an additional $n - 2$ two-hop paths between them also with unit capacity. So node A could send traffic to node B at a rate as high as $n - 1$. However, using two-hop paths comes at the cost of using twice the network capacity per traffic flow (i.e., two links instead of one). This has the following effect: If the network has a high traffic load so that most one-hop paths are nearly at capacity, the addition of a small additional load on a one-hop path could cause diversion of the excess load to one of the alternative two-hop paths. If these two-hop paths also are close to capacity, additional traffic on either of these hops would cause the use of additional two-hop paths. At high loads, this process continues until most traffic traverses two-hop paths, reducing network efficiency to 50%. This inefficient state turns out to be persistent: Even if traffic volume decreases, most traffic would still traverse two-hop paths. Thus, the system has more than one equilibrium.[5]

Despite the presence of multiple equilibria, note that the system is both BIBO and zero-input stable. To see this, we define the input to the system to be its traffic load and the output as the carried load. The system is in its relaxed state when traffic load is zero, corresponding to a zero carried load. If a small disturbance adds traffic to the system, it is carried on a one-hop path, and the carried load equals the traffic load. Therefore, as long as the input is bounded, so too is the output, showing BIBO stability.

To prove zero-input stability, consider a network in which the traffic load is nonzero at time zero but there is no additional load after time zero. In this case, once the traffic is delivered to its destination, the network always eventually returns to the zero state, as desired.

The instability, or rather, change in state from one equilibrium state to another, is caused by a disturbance when the system load is *high*. We see, therefore, that BIBO and zero-input stability criteria are rather weak. Perhaps a better way to view them is that a system that fails even these two weak stability criteria is not particularly robust.

5. This unstable equilibrium can be avoided by reserving a fraction of each link's capacity for one-hop paths.

EXAMPLE 8.15: COMPUTER NETWORK STABILITY

A proof along the lines of Example 8.14 shows that nearly all computer networks are trivially both BIBO and zero-input stable if we accept the definition that the output of the system is the overall carried load and the zero state is the zero-load state. We could also define the output of the network as the sum of all the router buffer occupancies, so that a congested network would have a large output amplitude. But all buffers, in practice, are bounded in size. So, no matter what the input load, they cannot exceed a predefined constant. From this perspective, again, computer networks are trivially stable. (The most common manifestation of marginal stability is probably the presence of a persistent oscillation, as we saw in Example 8.5.)


7:07

8.8.1 BIBO Stability Analysis of a Linear Time-Invariant System

This section presents the conditions under which a continuous single-input, single-output LTI system is BIBO stable. If the system has input $r(t)$, output $y(t)$, and impulse response $g(t)$, then, from Equation 5.28 in Section 5.4:

$$y(t) = \int_{-\infty}^{\infty} g(\tau)r(t-\tau)d\tau = \int_{0}^{\infty} g(\tau)r(t-\tau)d\tau$$

where the second step follows from the fact that for a causal system, the impulse response is zero when $t < 0$. The magnitude of the output is bounded by

$$\left|\int_{0}^{\infty} g(\tau)r(t-\tau)d\tau\right| \leq \int_{0}^{\infty} |g(\tau)||r(t-\tau)|d\tau \leq r_{max}\int_{0}^{\infty} |g(\tau)|d\tau$$

where r_{max} is the maximum value of the input. The output is bounded as long as the quantity

$$\int_{0}^{\infty} |g(\tau)|d\tau,$$

which is the integral of the absolute value of the impulse response $g(t)$, is bounded. To compute this quantity, we first consider the Laplace transform of the impulse response: the transfer function $G(s)$. It can be shown that the transfer function of any LTI system that can be modeled as a set of differential equations can be written as the ratio of two polynomials in s:

Erratum 8.20: Necessary as well as sufficient; see correction 8.20.

$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_m}{s^n + a_1 s^{n-1+\ldots+a_n}} \qquad \textbf{(EQ 8.20)}$$

The impulse response $g(t)$ of any real system is real, that is, it has no complex coefficients. Therefore, by the definition of the Laplace transform, its Laplace transform $G(s)$ also has real coefficients, and the roots of its denominator polynomial $s^n + a_1 s^{n-1+\ldots+a_n}$ must be either real and distinct, real and repeated, or conjugate complex; and it cannot have nonconjugate complex roots. Using partial fraction expansion (see Section 8.11), such a transfer function can be written in the form

$$G(s) = \left\{ \begin{array}{c} \dfrac{a_1}{(s-\alpha_1)} + \dfrac{a_2}{(s-\alpha_2)} + \ldots + \text{other distinct real roots} \\ \dfrac{a_k + jb_k}{(s-(\alpha_k + j\beta_k))} + \dfrac{a_k - jb_k}{(s-(\alpha_k - j\beta_k))} + \ldots + \text{other complex conjugate roots} \\ \dfrac{a_{m(r)}}{(s-\alpha_m)^r} + \dfrac{a_{m(r-1)}}{(s-\alpha_m)^{r-1}} + \ldots + \dfrac{a_{m(1)}}{(s-\alpha_m)} + \text{other repeated roots} \end{array} \right\} \qquad \textbf{(EQ 8.21)}$$

What does this look like in the time domain? The Laplace transform is linear, so we can take the inverse transform term by term.

- For a term of the form

  $$\frac{a_i}{(s-\alpha_i)},$$

  the corresponding time-domain term is $a_i e^{\alpha_i t}$, which decays exponentially to zero whenever $\alpha_i < 0$; that is, the root lies on the X-axis in the left half of the complex s plane. In this case, the integral of the absolute value of the impulse response is bounded, and the system is stable. If $\alpha_i = 0$, the output is constant, the integral is unbounded, and the system in unstable.
- For complex conjugate terms of the form

  $$\frac{a_k + jb_k}{(s-(\alpha_k + j\beta_k))} + \frac{a_k - jb_k}{(s-(\alpha_k - j\beta_k))},$$

  the corresponding time-domain term is $2e^{\alpha_k t}(a_k \cos\beta t - b_k \sin\beta t)$. Note that the expression in the parentheses is the difference between two sinusoids and therefore is bounded in magnitude. The output corresponding to this pair of terms decays to zero, corresponding to a bounded integral, if and only if α_k, the real part of the complex conjugate root, is negative, so that the pair of roots lie in the left half of the complex s plane. If the real part is zero (i.e., the



5:04

Erratum 8.21: Read "is not BIBO stable" (marginally stable); β is β_k.

root lies on the j axis), the system oscillates, and this oscillation does not decay over time. This system is marginally stable.

- A similar but somewhat more complicated analysis shows that if the system has repeated real or complex roots, then the corresponding time-domain output decays to zero, and the integral of the impulse response is bounded if and only if the real part of the root lies strictly in the left half of the complex s plane: Roots on the j axis lead to marginally stable systems, and repeated roots on the j axis or roots in the right half of the complex s plane lead to unstable systems.

Erratum 8.22: Read: integral of $|g(t)|$; roots *strictly* in the left half plane; see correction 8.22.

Summarizing, we see that an LTI system is BIBO stable if and only if *all* the roots of the denominator polynomial of its transfer function lie in the left half of the complex s plane. If even one root lies on the right half of the plane, the system is unstable. For the special case of conjugate roots on the j axis: (1) if the pairs of conjugate roots on this axis are distinct, the system is oscillatory but bounded; (2) if the pairs of conjugate roots on this axis are repeated, the system is unstable. This simple characterization allows us to quickly determine whether a single-input, single-output LTI system is BIBO stable simply by finding the roots of its denominator polynomial: These are also called the **poles** of the system. This can be determined using numerical algorithms, such as the Jenkins-Traub algorithm.

EXAMPLE 8.16: BIBO STABILITY OF AN LTI SYSTEM

Determine the stability of an LTI system with denominator polynomial $3s^4 + 5s^3 + 7s^2 + 12s + 1$.

Solution:

Using a numerical solver, we find that the roots of this polynomial are –0.08753, –1.6486, $0.03475 \pm j1.51936$. Since the conjugate roots lie in the right-half plane, the system is unstable.

EXAMPLE 8.17: BIBO STABILITY OF AN LTI SYSTEM

Determine the stability of an LTI system with denominator polynomial $s^4 + 50s^2 + 625$.

Solution:

We factor the polynomial as $(s^2 + 25)^2 = ((s + j5)(s - j5))^2$, so that there are two repeated roots at $\pm j5$. The system has repeated roots on the j axis and is therefore unstable.

EXAMPLE 8.18: STABILITY OF A WEB SERVER CONTROLLER

Erratum 8.23: See correction 8.23 (three parts).

Determine the stability of the proportional-, integral-, and derivative-mode controllers of Examples 8.9–8.11.

Solution:

For the proportional-mode controller, we have $Y = \frac{R}{1+s\tau} - \frac{W}{s(1+s\tau)}$. Ignoring the disturbance, the transfer function of the controlled system is given by $G = \frac{1}{1+s\tau}$. The root is at $s = -\frac{1}{\tau}$. Since $\tau > 0$ (it is a time constant), the system's single pole, which corresponds to this root, is always in the left half of the complex s plane and is therefore always BIBO stable.

For the integral-mode controller, we have $Y = \left(\frac{K_i}{s^2+K_i}\right)R$ so that $G = \left(\frac{K_i}{s^2+K_i}\right)$ and the roots are at $\pm jK_i^{1/2}$. The conjugate poles on the j axis indicate that the system is oscillatory and therefore bounded but BIBO marginally stable.

For the derivative mode-controller, we have $Y = \left(\frac{K_d}{1+K_d}\right)R$, so $G = \left(\frac{K_d}{1+K_d}\right)$, which is a constant that is never 0 (i.e., the system has no pole). The effect of the controller is to act as a constant scalar multiplier to the Laplace transform of the control input. If the input is bounded, so too is the output, which implies that the system is BIBO stable.



5:04

Erratum 8.24: Not a corner case; see correction 8.24.

There is one corner case that requires special care. Suppose that $G(s)$ has a single set of complex conjugate roots on the j axis, which corresponds to a marginally stable bounded oscillatory system. Suppose that it is excited by an input $R(s)$ that also has complex conjugate roots at the same locations on the j axis. Then, $Y(s) = G(s)R(s)$ has conjugate roots of multiplicity 2 on the j axis, making it both unbounded and unstable. The physical interpretation of this phenomenon is that an input of a sinusoid at a frequency that exactly matches one of the natural frequencies of the system leads to instability owing to **resonance**.

See the supplementary note on pole placement, page 182.

8.8.3 Placing System Roots

The roots of the denominator of the system-transfer function or of the characteristic equation are critical in determining its stability. If a system designer has one or more free parameters, or "control knobs" to control system behavior, these must be chosen such that these roots for the controlled system lie in the left half of the complex plane. This is called **pole placement**, since each root corresponds to a system pole.

Before the advent of digital computers, finding the roots of the characteristic equation was nearly impossible other than for trivial systems. System engineers, therefore, came up with approximation techniques, such as the root-locus method, Bode plots, and Nyquist plots to roughly determine root locations as a function of the system parameters. Today, sophisticated numerical algorithms have simplified the process of finding the roots of any polynomial. Nevertheless, an appreciation of the influence of the system poles on its behavior is critical in designing a robust and stable control system. We will not explore this complex topic in any detail. However, it should be clear that the farther to the left of the j axis that we place a pole, the faster the decay of impulse response transients, which makes the system more responsive. However, this can also lead to instability owing to overcorrection. The goal of the system designer is to balance responsiveness with stability.

8.8.4 Lyapunov Stability


16:07

Most real systems, such as computer networks, are neither linear nor time invariant. Yet we would like to understand whether they are stable and to design stable control algorithms for them. Here, we outline Lyapunov stability, an important concept that can be used to study such systems.

Erratum 8.25: State vectors are not covered earlier in these notes.

Recall from Section 8.2 that we can represent a system using its state variables in the form of a state vector. The system's state space is the set of all values that can be assumed by its state vector. We declare the ideal system state to be its equilibrium, or relaxed, state when all the system state variables are at their nominal zero value: $\mathbf{x} = \mathbf{0}$. The goal of the controller, then, is to move the system from its current state to this ideal state: The path so taken over time is called the system's **trajectory**. The trajectory is much like the path taken by a marble as it rolls down a slope, except that the system's position at any time is described by a state vector rather than a scalar.

A system is said to be **stable in the sense of Lyapunov** if all system trajectories can be confined to a bounded part of the state space as long as the system was initially confined to some other part of the state space. In other words, if we are allowed to choose the initial conditions, a Lyapunov stable system never wanders too far from equilibrium.

More formally, consider a system described by:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}(t)) \\ \boldsymbol{f}(\boldsymbol{x}(t) = \mathbf{0}) &= \mathbf{0} \end{aligned} \qquad \textbf{(EQ 8.23)}$$

The first equation says that the system-state trajectory can be described in terms of a first-order differential equation. The second equation says that if the system is in equilibrium, then it continues to be in equilibrium (it is a fixed point of **f**). This, of course, ignores disturbances.

Such a system is said to be **stable in the sense of Lyapunov** at $\boldsymbol{x} = \mathbf{0}$ if, for every real number $\varepsilon > 0$, there exists another real number $\delta > 0$ such that if $||\boldsymbol{x}(0)|| < \delta$ then for all $t \geq 0$ $||\boldsymbol{x}(t)|| < \varepsilon$, where $||\boldsymbol{x}|| = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$. So, as long as we can choose the initial conditions to be in a hypersphere centered at the origin and with radius δ, the system's trajectories will be confined to a hypersphere centered at the origin and with radius ε.

The system is **asymptotically stable** at $\boldsymbol{x} = \mathbf{0}$ if it is stable in the sense of Lyapunov, and, moreover, $\boldsymbol{x}(t) \to \mathbf{0}$ as $t \to \infty$. That is, we can choose initial conditions so that further evolution is not only bounded but also eventually converges to the equilibrium state.

Our overall goal is to prove that a nonlinear or time-variant system is either stable in the sense of Lyapunov or asymptotically stable. To do so, we define an auxiliary scalar function called the **Lyapunov function** $V(\boldsymbol{x})$. The Lyapunov function can be nearly arbitrarily defined and corresponds to the amount of energy in the system: The greater the energy, the larger the value of $V(\boldsymbol{x})$. We can think of a disturbance as adding energy to the system that is dissipated by the actions of the controller, so that the system eventually settles down to equilibrium. It can be shown that a system is stable in the sense of Lyapunov if it is possible to define a Lyapunov function for it such that the Lyapunov function decays over time (owing to the actions of the controller) along all system trajectories.

More formally, the Lyapunov stability theorem states that a system described by Equation 8.23 is stable in the sense of Lyapunov at $\boldsymbol{x} = \mathbf{0}$ if it is possible to define a scalar function $V(\boldsymbol{x}(t))$ such that

- $V(\boldsymbol{x}(t)=\mathbf{0}) = 0$
- $V(\boldsymbol{x}(t)) > 0;\ \boldsymbol{x}(t) \neq \mathbf{0}$
- $V(\boldsymbol{x}(t))$ is continuous and has continuous partial derivatives with respect to each component of $\boldsymbol{x}$
- $\frac{dV(\boldsymbol{x})}{dt} \leq 0$ for all system trajectories

Moreover, if $\dot{V}(\boldsymbol{x}) < 0$ (strictly less than zero) along all system trajectories, the system is asymptotically stable. This is also called **Lyapunov 2-stability**, or stability using the second method of Lyapunov.

The use of a Lyapunov function is both powerful and somewhat open-ended: If such a function can be found, even complex systems can be proved stable. On the other hand, finding such a function is a hit-or-miss process, other than for the simplest cases. In practice, we can use Lyapunov stability as a design tool by defining a plausible energy function and showing that control actions tend to always decrease system energy.

8.12 Further Reading

There are many excellent texts on control theory. A standard reference is B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, Wiley, 2002. Classical control is discussed in detail with numerous examples in M. Gopal, *Control Systems: Principles and Design*, McGraw-Hill, 2008. The standard reference for digital control systems is K. Ogata, *Discrete-Time Control Systems*, 2d ed., Prentice Hall, 1995.

8.13 Exercises

1. A bandwidth management system

Consider the following problem: An ISP wants to ensure that, on a specific congested link, the fraction of P2P traffic is never more than r percent. If this fraction is exceeded, the ISP sends TCP reset packets to P2P connection endpoints, terminating them, and thus reducing their load. For this system, identify the plant, the command, the control input, the disturbance, and the output.

2. Effort and flow

What are the effort and flow variables in the system of Exercise 1?

3. Transfer function

What is the transfer function for the system of Exercise 1 in the regime when the congested link has more than r percent of P2P connections? Assume that the control rule is to reduce the number of P2P connections by u connections over a time period T if the P2P traffic fraction exceeds r percent, and that the system is subject to a random disturbance at an instantaneous rate w.

4. First-order system

When given a step input, a first-order system reaches 63% of its asymptotic value of 4.25 units at time $3s$. What is its transfer function?

5. Second-order system

Prove that for a critically damped second-order system,

$$y(t) = K[1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}] .$$

6. Proportional-mode control

What is the relationship between the loop gain and the system pole for the proportional-mode controller discussed in Example 8.9?

7. Integral-mode control

What is the impulse response of the integral-mode controller in Example 8.10?

8. Stability

Use a numerical solver to determine the stability of an LTI system with denominator polynomial $3s^5 - 4s^3 + 7s^2 + 10s + 1$.

9. Partial fraction expansion

Find the partial fraction expansion of $\frac{s}{(s+3)(s+5)}$.

10. Partial fraction expansion

Find the partial fraction expansion of the polynomial fraction $\frac{1}{s^2 - 4s + 29}$.

Corrections to Chapter 8

The pages of Chapter 8 (Elements of Control Theory) are reproduced from the book with corrections. The following errata are not yet corrected in the text; each is flagged by a red note in the margin of the page concerned.

8.1 Page 319: the introduction to Chapter 8 mentions state-space representation, observability, controllability and digital control. These topics are not included in these notes.

8.2 Page 327: "the system is in equilibrium at time $t = 0$, or $x(0) = 0$" uses the state vector from the omitted part of Section 8.2.2. Read it as: the output and all its derivatives are 0 at $t = 0$.

8.3 Example 8.4, page 328: the state-space representation it recalls is not in these notes. The example writes $\dot{x} = u - w$, subtracting the disturbance, whereas Equation 8.14 and corrections 8.12–8.15 add it. For consistency, read $\dot{x} = u + w$.

8.4 Section 8.2, page 329: the last paragraph of this section should refer to Section 8.9, not 8.8. Section 8.9 (state-space control) is not included in these notes.

8.5 Section 8.4.2, page 332: the alternative form of ζ ($\zeta = \cos\theta$) is used later in this example, in Equation 8.11, and in Figures 8.5 and 8.6.

8.6 Section 8.4.2, page 332: the displaced line should read "Let $\zeta = \cos\theta$, so that $\theta = \cos^{-1}\zeta$. Then $\sqrt{1-\zeta^2} = \sqrt{1-\cos^2\theta} = \sin\theta$."

8.7 Page 333: with $\omega_n = 2$ and $\zeta = 0.2$, the period of the damped oscillation is $2\pi/(\omega_n\sqrt{1-\zeta^2}) \approx 3.2$ s, so $t = 10$ is about three periods of the oscillation (or four time constants, $1/(\zeta\omega_n) = 2.5$ s), not five.

8.8 Page 337: "the Laplace transfer of the input $U(s)$" should read "the Laplace transform of the input $U(s)$".

8.9 Page 339, Stability: the clause "although the system may potentially oscillate indefinitely within some bound" should be deleted. A system whose impulse response oscillates indefinitely, without decaying, is not BIBO stable (see Section 8.8.1). The output of a BIBO stable system can oscillate indefinitely only if its input does.

8.10 Page 339, Disturbance Rejection: without feedback the output due to the disturbance is GW; with feedback it is $GW/(1 + GDH)$. So feedback attenuates the disturbance by a factor of $1/(1 + GDH)$. $G/(1 + GDH)$ is the overall transfer function from W to Y.

8.11 Page 341, footnote 2: the loop gain K_p is the ratio of the control input to the error ($u = K_p e$), not the ratio of the command to the error.

8.12 Example 8.9, page 342: to be consistent with Equation 8.14, in which the disturbance is added to the control input ($Y = GU + GW$), the plant equation should read $Y = (1/s)(U + W)$, not $Y = (1/s)(U - W)$. The equation for Y should then read $Y = R/(1+s\tau)+W\tau/(1+s\tau)$. This does not affect the step response, which is computed with $W = 0$.

8.13 Example 8.10, page 343: for the same reason, the plant equation should read $Y = (1/s)(U + W)$, so that $Y = \Big(K_i/(s^2 + K_i)\Big)R + \Big(s/(s^2 + K_i)\Big)W$.

8.14 Example 8.10, page 344: the oscillation $\cos(\sqrt{K_i}\,t)$ has period $2\pi/\sqrt{K_i}$, not $1/\sqrt{K_i}$.

8.15 Example 8.11, page 345: for the same reason, the plant equation should read $Y = (1/s)(U + W)$, so that $Y = \Big(K_d/(1 + K_d)\Big)R + \Big(1/(s(1 + K_d))\Big)W$.

8.16 Page 348: "Pade approximations" should read "Padé approximations".

8.17 Figure 8.8, page 349: "sending date" should read "sending rate" (three times).

8.18 Example 8.13, page 350: the time axis of the upper (undamped) rate plot in Figure 8.8 is incorrect. At 50 packets per RTT, starting from about 0 at $t = 1.5$, the rate reaches 250 at $t = 6.5$, as the text says, not at about $t = 4$ as the plot shows.

8.19 Page 351: the definition of BIBO stability should read "A system is said to be bounded-input, bounded-output (BIBO) stable if, for every input bounded in magnitude by some value m, the output is bounded in magnitude by some value M, which may depend on m." The key word is "every": it is not enough for the output to be bounded for some bounded inputs.

8.20 Page 353: the condition that $\int_0^\infty |g(\tau)|\,d\tau$ be bounded is necessary as well as sufficient. If the integral is unbounded, then for a fixed t the bounded input $r(t - \tau) = \text{sign}(g(\tau))$ gives $y(t) = \int_0^t |g(\tau)|\,d\tau$, which can be made as large as we like. So an LTI system is BIBO stable if and only if its impulse response is absolutely integrable. The absolute value matters: for $g(t) = \cos(\omega_0 t)$, the integral of $g(t)$ is bounded, but the integral of $|g(t)|$ is not, and the system is not BIBO stable.

8.21 Page 354: "If $\alpha_i = 0$, the output is constant, the integral is unbounded, and the system in unstable" should read "... and the system is not BIBO stable". Like a system with a single pair of conjugate roots on the $j\omega$ axis, such a system is marginally stable. In the term $2e^{\alpha_k t}(a_k \cos\beta t - b_k \sin\beta t)$, β should read β_k.

8.22 Page 355: "the integral of the impulse response is bounded if and only if ..." should read "the integral of the absolute value of the impulse response is bounded if and only if ...". In the summary paragraph, "all the roots of the denominator polynomial of its transfer function lie in the left half of the complex s plane" should read "... lie

strictly in the left half of the complex s plane", and item (1) should read "if the roots on this axis are distinct, the impulse response is a sustained oscillation (or a constant, for a root at the origin). It is bounded, but it is not absolutely integrable, so the system is marginally stable and not BIBO stable: a bounded input at the natural frequency of the system causes resonance and an unbounded output (see page 356)."

8.23 Example 8.18, page 356: (a) Proportional mode: with Example 8.9 as corrected (correction 8.12), $Y = R/(1 + s\tau) + W\tau/(1 + s\tau)$; the term $W/(s(1 + s\tau))$ is wrong. (b) Integral mode: a marginally stable system is not BIBO stable, so "bounded but BIBO marginally stable" should read "oscillatory (marginally stable), and so not BIBO stable". (c) Derivative mode: the system is BIBO stable from R to Y, as stated. Note, however, that the response to the disturbance, $W/(s(1+K_d))$, has a pole at $s = 0$, so a step disturbance produces an output that grows without bound; derivative control alone does not reject a constant disturbance.

8.24 Page 356: the paragraph beginning "There is one corner case that requires special care" is not a corner case. It shows why a system with a single pair of conjugate roots on the $j\omega$ axis is not BIBO stable, as stated in the corrected summary on page 355. In the same paragraph, "making it both unbounded and unstable" should read "so that the output grows without bound".

8.25 Section 8.8.4, page 358: the state-vector representation it recalls from Section 8.2 is not in these notes. Here, the state vector x is simply the list of quantities that describe the system at a given time.

8.26 Section 8.11.3, page 369: "Suppose that $D(s)$ has r repeated roots α_1" should read "Suppose that $D(s)$ has a root α_1 repeated r times", and "where the ith repeated root, $i = 0, 1, \ldots, r-1$, is given by" should read "where the coefficient $a_1(r-i)$, $i = 0, 1, \ldots, r-1$, is given by".

Model Predictive Control

A supplementary note to Chapter 8. This note is not examinable.

This short note presents a conceptual overview of Model Predictive Control.

1 Introduction

MPC overview

The discussion in the text so far has been of classical (PID) control. These techniques work well when the system (or plant) is linear (or can be easily linearized in the neighbourhood of the operating point) and there is little coupling between the outputs, that is, the outputs are relatively independent of each other, depending only on the system state and plant inputs. However, many practical systems are non-linear and with coupled outputs. These systems are better controlled using Model Predictive Control (MPC).

At its heart, an MPC controller uses a model of the system to predict its expected evolution in response to its controlled and uncontrolled inputs. Specifically, the system is assumed to be fully described by its state variables. The model mathematically describes the relationship between the system's inputs and its outputs and the predictions are of the system state for the immediate future. Given these predictions, unlike PID controllers, whose controllers are relatively simple functions of the measured error, an MPC controller uses a potentially complex optimizer to choose control inputs such that:

- system inputs obey operating constraints
- system outputs are as close as possible to their reference values and obey operating constraints
- the cost of the control actions is minimized

EXAMPLE 1: AN AV SPEED CONTROLLER

We use a running example of a controller for an autonomous vehicle (AV). For simplicity, we focus on an MPC-based speed controller, which controls how fast the AV drives using an accelerator and a braking system. The AV's state is its current speed. The model mathematically relates the accelerator setting and brake level (the controlled inputs) as well as environmental factors such as wind direction and speed and road slope (the uncontrolled inputs) to the expected speed of the car (the output). The controller uses this model to predict the future speed of the vehicle in response to both controlled and uncontrolled inputs and uses the prediction to choose a feasible accelerator and braking level such that (a) the vehicle speed matches the desired value and (b) changes in speed do not subject the vehicle to too high a G force. Ideally, this set of control decisions uses the least amount of fuel.

Importantly, to deal with unforeseen disturbances, the optimal trajectory of control inputs (i.e., the set of control inputs for the immediate future) is computed but not entirely carried out. Instead, it is periodically recomputed, and after each re-computation only the first control action is carried out; the remainder of the trajectory is ignored. Thus, if the anticipated system state diverges from the true measured state, previously computed control actions are ignored.

EXAMPLE 2

Consider the actions of the AV speed controller when the AV turns a corner and spots a pedestrian on the road, an unanticipated situation. Prior to spotting the pedestrian, the controller would have chosen control inputs to ensure that the AV's speed matched the reference value. However, once the pedestrian is spotted, the prior control trajectory is set aside and a new control trajectory is computed to bring the AV to rest well before a potential collision.

Of course, this is wasteful in terms of computation overhead. Hence, an MPC designer must carefully decide how often to control the system, how much of the future trajectory to predict, and how best to model the system so that the optimal trajectory is easy to compute. We will discuss these issues later in this note.

2 Model



Let's make this intuitive description more precise. Consider the controller shown in Figure 1. Here, the reference vector r is one input to the MPC controller, the other input being the output of the state estimator SE that

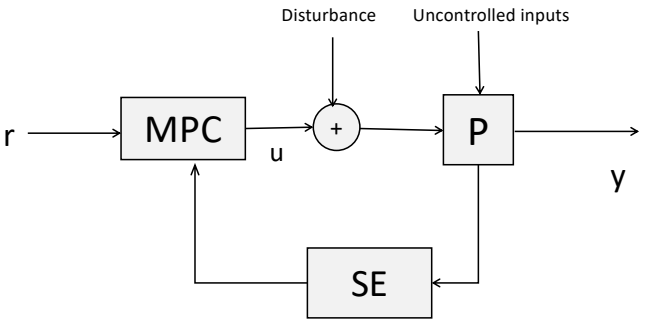


Figure 1: Conceptual model of a Model Predictive Controller (MPC)

observes the plant P and estimates its current state. The output of the controller, u, is subject to disturbances before being given to the plant. The output of the plant is y, and our principal control goal is to ensure that $y = r$.

EXAMPLE 3

For the case of the AV speed controller, the reference value is the scalar desired speed (this is assumed to be set by a higher-level controller). The plant is the AV, and its state corresponds to its speed. The state value is observed by the state estimator and made available to the controller. The output of the system is also its speed. Uncontrolled inputs to the plant include wind, road slope, pedestrians, barriers, and traffic signals. The accelerator and brake positions selected by the controller are the controlled inputs, and may themselves be subject to disturbances; for example, the braking system may apply a braking force that slightly differs from the one intended.

We will assume that the MPC controller operates at discrete time steps, indexed by k, as shown in Figure 2. Each step is called a *sample time*. At the time marked Now, the MPC controller uses a model of the system to predict its anticipated future evolution. This requires a prediction of the future uncontrolled inputs over the *prediction horizon*. This prediction can use an arbitrarily complex algorithm; common techniques include Auto-Regressive Moving Average (ARMA) and its many variants, such as Seasonal ARMA, ARIMA, and ARMAX or predictors based on Deep Neural Networks. The prediction horizon should be long enough to take into account significant future events. In the example of the AV speed controller, this would include stop signs and traffic lights anticipated along its path.

Predictions are used to compute the trajectory of future control actions. We would like these actions to be *feasible* and *optimal*. That is, the actions must

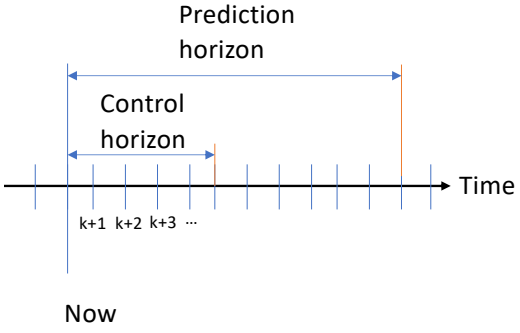


Figure 2: Time diagram

obey operating constraints, and the anticipated outputs must obey operating constraints, minimize control effort, and allow the output to closely track the reference value. To ensure optimality of the control actions, the MPC uses an optimizer to compute the optimal control trajectory over the course of a *control horizon*. The control horizon is typically shorter than the prediction horizon, since it is more difficult to compute, and, in any case, only the first step of the control trajectory is actually implemented.

Note that after the control actions are taken at the present time, time advances by one step, and the predictions and control actions are re-computed over the new prediction and control horizons. For this reason, MPC is also called *receding-horizon control*.

EXAMPLE 4

For the AV speed controller, the MPC would use an optimizer to compute the accelerator and brake positions for the control horizon, taking into account the predictions of significant future events. However, at each time step, the controller would only actuate the first action in the trajectory.

In computing the control actions, the optimizer could be asked to ensure that the controlled input and outputs obey both hard and soft constraints. Hard constraints must always be met, whereas a soft constraint, when violated, only results in a penalty, with the goal being to minimize this penalty.

EXAMPLE 5

For the AV speed controller, a hard constraint is the maximum acceleration and the maximum braking level. These constraints must always be met. A soft constraint might be how often the brake level is changed: if this constraint were violated, the AV's passengers would feel the ride was jerky.

Note that the MPC controller does not require the plant or operating

constraints to be linear. Instead the complexity of non-linear operation is dealt with by the optimizer and the predictor. The actions of the optimizer can be arbitrarily complex. They could, for instance, require the simulation of many possible future evolution paths of the system, with the best of these chosen as the one to realize. This can place a heavy computational burden on the controller.

In case of a highly non-linear system, predictions may be inaccurate and the system behaviour may be far from optimal. Moreover, the cost of computing the optimal trajectory may be quite high. Nevertheless, the overall control paradigm would still be applicable.

3 MPC Parameters

This section discusses how one might choose MPC control parameters such as the sample time, prediction horizon, and control horizon.


MPC parameters

Sample time. The sample time determines how often the MPC controller takes control actions. The shorter this period, the more responsive the control. However, this also increases the cost of computation. A rule of thumb is to first measure the rise time of the uncontrolled system, that is, the time it takes to go from 10% to 90% of its final value, when given a step input. It is best to have the sample time be about 5–10% of the rise time, so that there is ample time to make corrections during the rise time.

EXAMPLE 6

Consider the AV when it is given a step input, that is, the accelerator setting goes from zero to one unit (arbitrarily defined). We would expect that the speed of the AV would rise as shown in Figure 3. Then, we would like the sample time to be about 10–20 times smaller than this rise time.

Prediction horizon. How far should the controller predict into the future? Again, this balances computation time and accuracy of control. In general, the prediction horizon needs to be long enough to cover significant future events, such as an upcoming traffic light for an AV. From a control theoretic perspective, recall that when subjected to a unit step input a system exhibits both a steady state and transient response. The rule of thumb is that the prediction horizon should include about 20–30 sample times during the transient system response.

Control horizon. The control horizon is the time duration for which optimal control actions are computed. One might think that the longer this horizon



Figure 3: Guidance on choosing sample time

the better, but this makes the computation more expensive to carry out. Indeed, although the control horizon can be as long as prediction horizon, only the first step is implemented anyway, so there is no need to make the horizon too long. As a rule of thumb, the control horizon should be about 10%–20% of the prediction horizon.

EXAMPLE 7

Imagine that an on-board map allows the AV to predict that a stop sign is expected, say, 100 time steps in the future. If the prediction horizon was greater than 100 time steps, this significant fact would be available to the MPC controller. Given this information, it should change the desired reference speed trajectory r to slow down the AV from its current speed to zero in 100 time steps. However, the control actions, i.e., the accelerator and brake settings, will need to be computed only for, say, the next 10 steps, which is the control horizon. The control horizon need not be longer since unexpected disturbances, such as wind gusts, may require a re-computation of control settings several times before the stop sign is reached.

Hard and soft constraints. Hard and soft constraints can be set on both inputs and outputs. If a hard constraint cannot be met, the optimizer will be forced to give up, claiming a solution cannot be found. In this situation, there needs to be a fallback plan, otherwise the system will be uncontrolled! Thus, it is best to define most constraints to be soft, reserving hard constraints only to express safety or operational limits.

EXAMPLE 8

The upper and lower limits of operation of the accelerator are hard constraints, as is the maximum braking force. On the other hand, the requirement that the speed not change too abruptly from one time step to another is a soft constraint. We can express the expected absolute change in speed as a penalty variable in the optimization objective, encouraging the optimizer to reduce this penalty to the smallest possible value.

If a pedestrian were to suddenly step in front of the AV, so that it could not stop in time even with the maximum braking force, then the optimizer would declare infeasibility and give up. A fallback plan, in this case, might be to maximally steer the car away from the pedestrian.

Optimization objective. The optimization objective is an expression or function that an optimizer tries to minimize or maximize. As an MPC designer, a careful choice of the objective function allows the optimizer to balance between competing goals. Every objective function must include a term that reflects the error between the reference and actual output values, so that this objective is achieved. In addition, the objective function may have terms reflecting the amount of change in inputs or outputs between two control steps, as well as terms reflecting the cost of control (e.g., the amount of fuel used by the engine). We can balance between competing goals by assigning different weights to each goal, as the next example demonstrates.

EXAMPLE 9

A simple objective function for the AV MPC controller that it would seek to minimize might be $\sum_k \alpha|r(k)-y(k)|+(1-\alpha)|y(k)-y(k-1)|$, where α is a tuning parameter, $r(k)$ is the reference speed, and $y(k)$ is the speed at time step k. When α is close to 1, then the goal is to match the output speed to the reference as fast as possible (even if this results in a jerky ride), whereas when it is close to 0, the goal is to ensure a smooth ride. Intermediate values allow us to balance these goals. In practice, the objective function would need to be far more complex to account, for example, for the cost of fuel.

EXAMPLE 10

For completeness, here is an example of the optimization problem the MPC

controller might solve:

$$\min_{\text{acceleration}(k),\,\text{brake}(k)} \sum_{k=\text{now}}^{\text{now+control horizon}} \alpha|r(k) - y(k)| + (1-\alpha)|y(k) - y(k-1)| \quad \textbf{(EQ 1)}$$

$$\text{subject to } y(k) = f\Big(y(k-1), \text{acceleration}(k), \text{brake}(k)\Big) \quad \textbf{(EQ 2)}$$

$$y(k) \geq 0 \quad \textbf{(EQ 3)}$$

$$\text{acceleration}(k) \geq 0 \quad \textbf{(EQ 4)}$$

$$\text{brake}(k) \geq 0 \quad \textbf{(EQ 5)}$$

where α is a tuning parameter, $r(k)$ is the reference speed, and $y(k)$ is the speed at time step k. f is the mathematical model that relates the prior state and current inputs to current outputs. Note that the objective function includes future time steps, thus requiring a predictor.

4 Some Advanced Concepts

Non-linear MPC

This section discusses some advanced techniques for model predictive control.

4.1 Dealing with Non-Linearity

A significant advantage of MPC-based control over other approaches is that it can deal with non-linear systems. With MPC, any or all of the plant model, the constraints, and the control costs can be non-linear. However, this comes at the expense of additional computation. Moreover, there is no guarantee that the outcomes are optimal.

More specifically, with a linear plant model, linear constraints, and linear or quadratic costs, the optimizer has to solve a convex optimization problem, which is easy to solve, and the outcome is guaranteed to be optimal. However, if any of these are non-linear, then the optimization problem may be non-convex, which can lead to sub-optimal outcomes.

In some cases, the MPC approach can be modified to deal with non-linearities. Two variants of MPC that are widely used in practice to deal with non-linear system models are adaptive MPC and gain-scheduled MPC. In both approaches, the system is linearized around its operating point, using a Taylor expansion. If the system has multiple operating points, the output space is decomposed into a set of sub-spaces, and a linearized model is defined for each subspace. This results in a piecewise-linear approximation of the plant.

With adaptive MPC the piecewise-linear model is computed on the fly as the system changes. Note that with this approach, the structure of the optimization problem stays the same, only its parameters change to reflect

the underlying subspace.

With gain scheduled MPC, the output space is linearized offline, and one controller is designed for each linearized model. As the operating point of the system evolves, a higher-level controller switches between the appropriate controllers. This uses more memory than the adaptive approach, but can be more responsive, since the linearization does not need to be computed on-the-fly.

4.2 Improving MPC Performance


MPC performance

It should be clear by now that the main problem with the MPC approach is its high computational demand. However, there are some approaches to reducing this computational cost, as discussed next.

- The computational cost can be reduced by reducing the length of the prediction and control horizon. Generally speaking, prediction is easier to carry out than optimization. So, reducing the control horizon has the greater impact on reducing computational cost. However, especially with non-linear optimization, a reduction in computation may come at the expense of significant sub-optimality.

- In some cases it is possible to optimize over a longer control horizon without significantly increasing costs by using increasingly larger time steps over the control horizon. For example, each time step can have a duration that multiplicatively increases the duration of the prior step. This allows the controller to respond to significant events quite far in the future, but with only a logarithmic increase in cost.

- A complex plant model is harder to simulate and optimize. If possible, portions of the plant model that are not strictly necessary should be excised. This is the intuition behind model order reduction techniques.

- If floating point operations are computationally expensive, then costs can be reduced by using lower precision or better data representation (as integers, for example, instead of floats).

- With explicit MPC, the optimal output of the controller for a given set of inputs is pre-computed and stored in a table. Then, in operation, instead of running the optimizer, the table can be looked up, reducing computational time. Of course, this requires substantial prior computation, as well as the need to store a potentially large lookup table corresponding to all possible operation states. As a compromise, the table could store (or cache) frequently occurring parameter combinations, reverting to standard operation when the system strays beyond the set of known states.

5 Conclusion

Model predictive control harnesses the power of modern microprocessors to compute optimal control actions based on the measured state. Unlike classical (PID) control, which does not work well when the plant model or constraints are non-linear, the MPC paradigm can deal with complex, coupled outputs and non-linearities. However, this comes at the cost of additional computation. Many variants of the basic approach have been developed to reduce this computational cost.

6 Further Reading

Here are some references to read further about MPC. Rawlings [5] provides a more mathematical tutorial that links MPC to Linear Quadratic Gaussian control. Prediction using ARMA and its variants is discussed in Reference [3] and more recent approaches are surveyed in [2]. Alessio and Bemporad [1] survey approaches used for explicit MPC.

7 Acknowledgements

This note is based substantially on a 5-part tutorial from MATLAB on YouTube [4]. Thanks also to Fiodar Kazhamiaka for his comments on an earlier draft.

References

[1] Alessio, A., and Bemporad, A. A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*. Springer, 2009, pp. 345–369.

[2] Han, Z., Zhao, J., Leung, H., Ma, K. F., and Wang, W. A review of deep learning models for time series prediction. *IEEE Sensors Journal* (2019).

[3] Makhoul, J. Linear prediction: A tutorial review. *Proceedings of the IEEE* 63, 4 (1975), 561–580.

[4] MATLAB: Understanding Model Predictive Control. `https://www.youtube.com/watch?v=8U0xiOkDcmw`. Accessed: 2020-12-22.

[5] Rawlings, J. B. Tutorial overview of model predictive control. *IEEE Control Systems Magazine* 20, 3 (2000), 38–52.

N

From a Differential Equation to Pole Placement

This note links the concepts in Chapters 5 and 8. It does not introduce any additional material and is meant only to clarify these concepts.

N.1 Motivation

Many of the systems we study in this course, from the temperature of a processor to the occupancy of a request buffer, are described by a linear differential equation with constant coefficients. A control engineer, however, rarely works with the differential equation directly. Instead, the engineer reasons about the *poles* of the system: a handful of complex numbers that determine whether the system is stable, how quickly it responds, and whether it oscillates. Designing a controller then amounts to choosing its parameters so that the poles of the controlled system lie where we want them, which is called pole placement.

This note connects these two descriptions. The rest of the note is laid out as follows. First, we write the differential equation in the operator form due to Lathi (2005), which is the form used in Section 5.4.2 of the textbook (Keshav, 2012). Next, we derive the characteristic polynomial and show that its roots determine the response of the system when it receives no input. Then, using the Laplace transform, we show that these roots are exactly the poles of the

transfer function, and explain why the poles govern the response to every input. Finally, we show how feedback changes the characteristic polynomial, which is the basis of pole placement.

N.2 The Lathi form

Consider a system with input $u(t)$ and output $y(t)$ that is described by the linear differential equation

$$\frac{d^N y}{dt^N} + a_1 \frac{d^{N-1} y}{dt^{N-1}} + \cdots + a_{N-1} \frac{dy}{dt} + a_N y = b_{N-M} \frac{d^M u}{dt^M} + \cdots + b_{N-1} \frac{du}{dt} + b_N u, \quad \textbf{(EQ N.1)}$$

where the coefficients a_i and b_i are real constants. This is Equation 5.22 of the textbook, which writes the input as $x(t)$. In this note, we write the input as $u(t)$ instead, to match Chapter 8, where x denotes the state of the system. We require $M \leq N$.[1]

Let D denote the operator d/dt, so that $\mathrm{D}^k y$ is the kth derivative of y. Then Equation N.1 can be written compactly as

$$Q(\mathrm{D})\, y(t) = P(\mathrm{D})\, u(t), \quad \textbf{(EQ N.2)}$$

where

$$Q(\mathrm{D}) = \mathrm{D}^N + a_1 \mathrm{D}^{N-1} + \cdots + a_{N-1}\mathrm{D} + a_N, \quad \textbf{(EQ N.3)}$$

$$P(\mathrm{D}) = b_{N-M}\mathrm{D}^M + b_{N-M+1}\mathrm{D}^{M-1} + \cdots + b_{N-1}\mathrm{D} + b_N. \quad \textbf{(EQ N.4)}$$

We call Equation N.2 the *Lathi form* of the system. Intuitively, the left-hand side describes the internal dynamics of the system, that is, how its output and the derivatives of its output are tied together, and the right-hand side describes how the input drives it. The integer N is the *order* of the system.

Here are two examples that appear in the labs.

(a) **The temperature of a processor.** If a processor dissipates power $p(t)$, has heat capacity C and thermal resistance R to the surrounding air, then its temperature rise $\theta(t)$ above the air satisfies $RC\,\dot{\theta} + \theta = R\,p$. Dividing by RC, the Lathi form is

$$\left(\mathrm{D} + \frac{1}{RC}\right)\theta(t) = \frac{1}{C}\,p(t), \quad \textbf{(EQ N.5)}$$

which is a first-order system with $N = 1$ and $M = 0$.

(b) **A second-order system.** The system with gain K, natural frequency ω_n and damping ratio ζ of Equation 8.9 has the Lathi form

$$\left(\mathrm{D}^2 + 2\zeta\omega_n \mathrm{D} + \omega_n^2\right) y(t) = K\omega_n^2\, u(t). \quad \textbf{(EQ N.6)}$$

[1] If $M > N$, the gain of the system grows without limit as the frequency of the input increases. Such a system would amplify high-frequency noise without bound and cannot be built from physical components.

N.3 The characteristic polynomial and the natural response

We first ask what the system does on its own, when it receives no input. Setting $u(t) = 0$ in Equation N.2 gives

$$Q(\mathrm{D})\, y(t) = 0. \qquad \textbf{(EQ N.7)}$$

To solve this equation, we try $y(t) = c\, e^{\lambda t}$ for some constants $c \neq 0$ and λ. Differentiating an exponential multiplies it by λ, so $\mathrm{D}^k e^{\lambda t} = \lambda^k e^{\lambda t}$, and therefore

$$Q(\mathrm{D})\, c\, e^{\lambda t} = Q(\lambda)\, c\, e^{\lambda t}. \qquad \textbf{(EQ N.8)}$$

Since $c\, e^{\lambda t}$ is never zero, the right-hand side vanishes for all t if and only if

$$Q(\lambda) = \lambda^N + a_1 \lambda^{N-1} + \cdots + a_{N-1}\lambda + a_N = 0. \qquad \textbf{(EQ N.9)}$$

The polynomial $Q(\lambda)$ is the *characteristic polynomial* of the system, and Equation N.9 is its *characteristic equation*. By the fundamental theorem of algebra, it has N roots $\lambda_1, \ldots, \lambda_N$, which may be complex. If the roots are distinct, then, because the system is linear, the general solution of Equation N.7 is the weighted sum

$$y(t) = \sum_{i=1}^{N} c_i\, e^{\lambda_i t}, \qquad \textbf{(EQ N.10)}$$

where the constants c_i are fixed by the initial conditions. This is the *natural response* of the system (Equation 5.23). A root of multiplicity r contributes the terms $e^{\lambda t}, te^{\lambda t}, \ldots, t^{r-1}e^{\lambda t}$ instead (Equation 5.24). Because the coefficients a_i are real, complex roots occur in conjugate pairs, and each pair combines into a real sinusoid multiplied by a real exponential.

Each root $\lambda = \sigma + j\omega$ thus contributes a *mode* $e^{\sigma t}e^{j\omega t}$ to the natural response. Its real part σ determines whether the mode decays ($\sigma < 0$), grows ($\sigma > 0$), or persists with constant amplitude ($\sigma = 0$). Its imaginary part ω determines whether, and how fast, the mode oscillates. For example, the roots of Equation N.5 and Equation N.6 are

$$\lambda = -\frac{1}{RC} \qquad \text{and} \qquad \lambda = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}, \qquad \textbf{(EQ N.11)}$$

respectively. The first decays with time constant RC. The second gives a decaying oscillation when $0 < \zeta < 1$ and two decaying exponentials when $\zeta > 1$, which are the cases studied in Section 8.4. Note that the natural response depends only on the coefficients a_i, that is, only on Q. The input side of the equation, P, plays no part.

N.4 The transfer function and its poles

We now show that the roots of the characteristic equation are exactly the poles of the transfer function. Suppose that the system is at rest at $t = 0$, so that y and its first $N-1$ derivatives are zero, as are u and its derivatives. This is the assumption under which a transfer function is defined (Section 8.2.2). Under this assumption, the Laplace transform of the kth derivative of a signal is s^k times the transform of the signal:

$$\mathcal{L}\left[\frac{d^k y}{dt^k}\right] = s^k\, Y(s). \qquad \textbf{(EQ N.12)}$$

Taking the Laplace transform of both sides of Equation N.2 therefore replaces every D by s:

$$Q(s)\, Y(s) = P(s)\, U(s). \qquad \textbf{(EQ N.13)}$$

Hence the transfer function of the system is

$$G(s) = \frac{Y(s)}{U(s)} = \frac{P(s)}{Q(s)}. \qquad \textbf{(EQ N.14)}$$

The denominator of the transfer function is the characteristic polynomial, with the operator D replaced by the complex variable s. The poles of $G(s)$ are the values of s at which $G(s)$ is infinite, that is, the roots of $Q(s) = 0$. These are the same numbers $\lambda_1, \ldots, \lambda_N$ that we found in Section N.3.[2] The roots of the characteristic equation and the poles of the transfer function are therefore one set of numbers, found in the time domain in the first case and in the s domain in the second.

N.4.1 Why the poles govern the response to every input

The natural response describes the system when it receives no input. The poles matter far more widely than this, because they shape the response to every input. To see why, write the transform of the output for an arbitrary input as

$$Y(s) = \frac{P(s)}{Q(s)}\, U(s). \qquad \textbf{(EQ N.15)}$$

[2]Strictly, this holds when P and Q have no common root. A root λ_c shared by P and Q cancels in $G(s)$ and does not appear as a pole. However, it remains a mode of the natural response. The natural response is the solution of $Q(\mathrm{D})\, y = 0$, which depends only on Q, so $e^{\lambda_c t}$ is still one of its terms. What the cancellation tells us is only that the *input* cannot excite this mode: the factor $(s - \lambda_c)$ in $P(s)$ removes it from $P(s)U(s)/Q(s)$. If the system starts with nonzero initial conditions, these contribute terms to $Y(s)$ whose numerators do not, in general, contain $(s - \lambda_c)$, so $e^{\lambda_c t}$ appears in the output. For example, if $Q(\mathrm{D}) = (\mathrm{D}+1)(\mathrm{D}+2)$ and $P(\mathrm{D}) = \mathrm{D}+1$, then $G(s) = 1/(s+2)$, but a nonzero initial condition still produces a term in e^{-t}. This matters for stability: if the cancelled root lies in the right half-plane, $G(s)$ appears stable while the system itself is not.

When $U(s)$ is a ratio of polynomials, as it is for steps, ramps, exponentials and sinusoids, we can expand $Y(s)$ in partial fractions (Section 8.11). Each root λ_i of $Q(s)$ contributes a term $A_i/(s-\lambda_i)$, whose inverse transform is $A_i e^{\lambda_i t}$: one of the modes of the natural response. The poles of $U(s)$ contribute the remaining terms, which together form the *forced response*.

To see what these terms look like, suppose that $U(s)$ has a simple pole at $s = p$. Its partial fraction expansion then contains a term $R/(s-p)$, where the constant

$$R = \lim_{s \to p} (s-p)\, U(s) \qquad \textbf{(EQ N.16)}$$

is called the *residue* of $U(s)$ at p. It is the coefficient that the partial fraction expansion assigns to this pole (Section 8.11). For example, consider the unit step, whose transform is $U(s) = 1/s$. Its denominator is zero only at $s = 0$, so it has a single simple pole at $p = 0$. Substituting $U(s) = 1/s$ and $p = 0$ into Equation N.16 gives

$$R = \lim_{s \to 0} (s-0) \cdot \frac{1}{s} = \lim_{s \to 0} 1 = 1. \qquad \textbf{(EQ N.17)}$$

The factor $(s-p)$ cancels the pole, which is why the limit is finite. The partial fraction expansion of $U(s)$ therefore consists of the single term $1/(s-0)$, whose inverse transform is $1 \cdot e^{0 \cdot t} = 1$ for $t > 0$. This recovers the step itself, and shows that a step is the special case of an exponential $R\,e^{pt}$ with $p = 0$ and $R = 1$. The term $R/(s-p)$ is the transform of $R\,e^{pt}$, so it represents a component of the input that is a (possibly complex) exponential with amplitude R. If p is not also a root of $Q(s)$, then $Q(p) \neq 0$, and applying Equation N.16 to $Y(s) = \big(P(s)/Q(s)\big)U(s)$ shows that the residue of $Y(s)$ at p is $\big(P(p)/Q(p)\big)R$. The expansion of $Y(s)$ therefore contains the term $\big(P(p)/Q(p)\big)R/(s-p)$. Its inverse transform is

$$y_{\text{forced}}(t) = \frac{P(p)}{Q(p)}\, R\, e^{pt}. \qquad \textbf{(EQ N.18)}$$

In words, the system passes each exponential component of the input through unchanged in shape, and only scales it by the complex number $P(p)/Q(p)$. This is the eigenfunction property of Section 5.4.1, which states that e^{st} is an eigenfunction of every LTI system. We now see that the corresponding eigenvalue is $P(s)/Q(s)$.

Here are two important cases.

(a) **A step input.** A unit step has a single pole at $p = 0$ with residue $R = 1$, so the forced response is the constant $P(0)/Q(0) = b_N/a_N$. This is the steady state value of the output, and b_N/a_N is the steady state gain of the system. For the first-order system of Section 8.3, $P(0)/Q(0) = K$, which is why K is called the gain.

(b) **A sinusoidal input.** The input $\cos\omega_0 t = \frac{1}{2}\left(e^{j\omega_0 t} + e^{-j\omega_0 t}\right)$ has poles at $p = \pm j\omega_0$, each with residue $R = \frac{1}{2}$. Because P and Q have real coefficients, $P(-j\omega_0)/Q(-j\omega_0)$ is the complex conjugate of $P(j\omega_0)/Q(j\omega_0)$, and the two forced terms combine into

$$y_{\text{forced}}(t) = \left|\frac{P(j\omega_0)}{Q(j\omega_0)}\right| \cos(\omega_0 t + \angle P(j\omega_0) - \angle Q(j\omega_0)). \qquad \textbf{(EQ N.19)}$$

The output is a sinusoid at the same frequency as the input. Only its amplitude and phase change, by the magnitude and the angle of $P(j\omega_0)/Q(j\omega_0)$.

Equation N.18 fails when a pole of the input coincides with a pole of the system, since $Q(p)$ is then zero. In that case the two poles merge into a repeated pole of $Y(s)$, which produces a term of the form $t\,e^{pt}$ whose amplitude grows with time. This is the resonance described in Section 5.4.2: an input at one of the natural frequencies of a system is amplified.

Thus every response of the system has two parts. The first is a combination of the natural modes $e^{\lambda_i t}$, which are the same for every input, although the input decides how strongly each is excited through the weights A_i. The second is the forced response, which has the shape of the input.

The step response of the first-order system of Section 8.3 shows this clearly. Its Lathi form is $(\mathrm{D} + 1/\tau)\,y = (K/\tau)\,u$, so $Q(s) = s + 1/\tau$ and $P(s) = K/\tau$. With a unit step input, $U(s) = 1/s$,

$$Y(s) = \frac{K/\tau}{s\,(s + 1/\tau)} = \frac{K}{s} - \frac{K}{s + 1/\tau}, \qquad y(t) = K - Ke^{-t/\tau}. \qquad \textbf{(EQ N.20)}$$

The pole at $s = 0$ comes from the input and gives the steady state response K. The pole at $s = -1/\tau$ comes from the system and gives the transient response $-Ke^{-t/\tau}$. The division of a response into a steady state part and a transient part is therefore the division of the poles of $Y(s)$ into those of the input and those of the system.

Finally, note that nonzero initial conditions do not change this picture. With the unilateral Laplace transform, they add terms to the numerator of Equation N.13, but these terms are still divided by $Q(s)$, so they excite the same modes.

N.5 Feedback and pole placement

A system is asymptotically stable when every root of its characteristic polynomial lies in the left half of the complex plane (Section 5.4.4 and Section 8.8). If a system is unstable, or stable but too slow or too oscillatory, the remedy is feedback. A controller measures the output, compares it with the desired

value, and adjusts the input accordingly. This changes the differential equation that governs the closed loop, and therefore its characteristic polynomial. Because the parameters of the controller appear in the coefficients of the new characteristic polynomial, we can choose them to place its roots. This is *pole placement* (Section 8.8.3).

Example 8.9 is the simplest instance of pole placement. The request buffer of the Web server of Example 8.2 satisfies $\dot{y} = u$, where u is the control input. Its Lathi form is $\mathrm{D}\,y = u$, so $Q(\mathrm{D}) = \mathrm{D}$, with a single root at $\lambda = 0$. The buffer is an integrator: left to itself it neither returns to a setpoint nor drifts away from it, and it is only marginally stable.

With proportional control, the controller chooses $u = (r - y)/\tau$, where r is the desired setpoint and $1/\tau$ is the loop gain. Substituting, $\dot{y} = (r - y)/\tau$, so the closed loop has the Lathi form

$$\left(\mathrm{D} + \frac{1}{\tau}\right) y(t) = \frac{1}{\tau}\, r(t). \qquad \textbf{(EQ N.21)}$$

The characteristic polynomial of the closed loop is $\mathrm{D} + 1/\tau$, with its root at $\lambda = -1/\tau$. Feedback has moved the pole from the origin into the left half-plane, and by choosing the loop gain we can place it anywhere on the negative real axis. The larger the loop gain, the further to the left the pole, and the faster the transient decays.

In this example, the designer has one parameter and can therefore move the pole only along a single curve, here the negative real axis. A controller with more parameters, such as a PID controller, gives the designer more freedom over where the poles go. However, as Section 8.8.3 notes, placing poles far to the left in pursuit of responsiveness risks overcorrection, and the designer must balance responsiveness against stability.

N.6 Summary and exercises

The chain of reasoning in this note is as follows.

1. A linear system with constant coefficients can be written in the Lathi form $Q(\mathrm{D})\, y = P(\mathrm{D})\, u$.
2. Its natural response is a sum of modes $e^{\lambda_i t}$, where the λ_i are the roots of the characteristic equation $Q(\lambda) = 0$.
3. Its transfer function is $G(s) = P(s)/Q(s)$, so its poles are these same roots.
4. Every response of the system, to any input, contains these modes, so the poles determine its stability, speed and tendency to oscillate.
5. Feedback changes the characteristic polynomial, and choosing the parameters of the controller places its roots.

To check your understanding, work through the following exercises.

1. Write the system $\ddot{y} + 3\dot{y} + 2y = \dot{u} + 3u$ in the Lathi form, find its characteristic roots, and write down its transfer function.
2. Using Equation N.21, show that the step response of the closed loop of Example 8.9 is $y(t) = 1 - e^{-t/\tau}$, and identify which term comes from which pole.

References

S. Keshav. *Mathematical Foundations of Computer Networking*. Addison-Wesley, 2012.

B. P. Lathi. *Linear Systems and Signals*. Oxford University Press, New York, 2nd edition, 2005.



Background

This part collects background material that the rest of the notes draw on. Sections 3.5 and 3.6 of the book cover linear transformations, eigenvalues and eigenvectors, and stochastic matrices, which underlie the stationary distribution of a Markov chain in Chapter 6. Section 8.11 covers partial fraction expansion, which is used to invert Laplace transforms in Chapters 5 and 8.

This material is not examinable.

3.5 Linear Transformations, Eigenvalues, and Eigenvectors

This section deals with the important problem of linear transformations and their computation, using eigenvalues and eigenvectors.

3.5.1 A Matrix as a Linear Transformation

Recall that the product of an $n \times n$ matrix with an n dimensional column vector—a matrix of size $n \times 1$—is another column vector of dimension n. We can therefore view the matrix as **transforming** the input vector into the output vector.

Note that the kth element of the output column vector is formed by combining all the elements of the input vector, using the weights found in the kth row of the matrix. This is just a *linear combination* of the elements of the input vector. A square matrix that represents (the weights corresponding to) a set of such linear combinations is said to represent a **linear transformation** of the input vector.

EXAMPLE 3.13: MATRIX AS A LINEAR TRANSFORMATION

Consider the matrix $\begin{bmatrix} 2 & 3 \\ 6 & 2 \end{bmatrix}$ and an input vector $\begin{bmatrix} a \\ b \end{bmatrix}$. The multiplication of this vector with the matrix—$\begin{bmatrix} 2 & 3 \\ 6 & 2 \end{bmatrix} * \begin{bmatrix} a \\ b \end{bmatrix}$ is the output vector $\begin{bmatrix} 2a + 3b \\ 6a + 2b \end{bmatrix}$. The first element of the output vector can be thought of as combining the input elements a and b with weights 2 and 3 (the first row of the matrix), and the second element of the output vector can be thought of as combining the inputs with weights 6 and 2 (the second row of the matrix).

The definition of matrix multiplication allows us to represent the composition of two linear transformations as a matrix product. Specifically, suppose that the matrix $\boldsymbol{A}$ transforms a column vector $\boldsymbol{x}$ to another column vector $\boldsymbol{x'}$ and that the matrix $\boldsymbol{B}$ is now used to transform $\boldsymbol{x'}$ to another vector $\boldsymbol{x''}$. Then, $\boldsymbol{x''} = \boldsymbol{Bx'} = \boldsymbol{B}(\boldsymbol{Ax}) = (\boldsymbol{BA})\boldsymbol{x} = \boldsymbol{Cx}$, where $\boldsymbol{C} = \boldsymbol{BA}$ is the product of the two transformation matrices. That

is, we can represent the composition of the two transformations as the matrix product of the transformation matrices.

EXAMPLE 3.14: COMPOSITION OF LINEAR TRANSFORMATIONS

We have already seen that the matrix $\begin{bmatrix} 2 & 3 \\ 6 & 2 \end{bmatrix}$ transforms a vector $\begin{bmatrix} a \\ b \end{bmatrix}$ to the vector $\begin{bmatrix} 2a+3b \\ 6a+2b \end{bmatrix}$. Suppose that we apply the matrix $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ to this output. The resultant value is $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} * \begin{bmatrix} 2a+3b \\ 6a+2b \end{bmatrix} = \begin{bmatrix} 4a+6b \\ 12a+4b \end{bmatrix}$. Instead, we can compute the product of the two transformation matrices as $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} * \begin{bmatrix} 2 & 3 \\ 6 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}$. Then, applying this product to the initial input gives us $\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix} * \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 4a+6b \\ 12a+4b \end{bmatrix}$, as before.

3.5.2 The Eigenvalue of a Matrix

Lab: Eigenvalues

Consider the matrix $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and an input vector $\begin{bmatrix} a \\ b \end{bmatrix}$. The multiplication of this vector with the matrix is the output vector $\begin{bmatrix} 2a \\ 2b \end{bmatrix} = 2 * \begin{bmatrix} a \\ b \end{bmatrix}$. Therefore, the matrix represents a *doubling* transformation on its input: The result of applying this matrix to any vector is equivalent to multiplying the vector by the scalar 2.

When the result of a matrix multiplication with a *particular* vector is the same as a scalar multiplication with that vector, we call the scalar an **eigenvalue** of the matrix and the corresponding vector an **eigenvector**. More precisely, we define an eigenvalue of a square matrix $\boldsymbol{A}$ to be a scalar λ such that, for *some* nonzero column vector x,

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \qquad \textbf{(EQ 3.14)}$$

The magnitude of an eigenvalue indicates the degree to which the matrix operation scales an eigenvector: The larger the magnitude, the greater the scaling effect.

EXAMPLE 3.15: EIGENVALUES AND EIGENVECTORS

Compute the eigenvalues and corresponding eigenvectors of the matrix $\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}$.

Solution:

Let the eigenvector be $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Then, $\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Expanding the left-hand side, we get a system of two equations:

$$\begin{aligned} 4x_1 + 6x_2 &= \lambda x_1 \\ 12x_1 + 4x_2 &= \lambda x_2 \end{aligned} \qquad \text{(EQ 3.15)}$$

Transposing terms, we get

$$\begin{aligned} (4-\lambda)x_1 + 6x_2 &= 0 \\ 12x_1 + (4-\lambda)x_2 &= 0 \end{aligned}$$

We recognize this as a homogenous system of two linear equations. Recall from Section 3.4.5 that this system has a nonzero solution if and only if the determinant of the coefficient matrix is zero. That is,

$$\begin{vmatrix} (4-\lambda) & 6 \\ 12 & (4-\lambda) \end{vmatrix} = 0$$

We can expand this to get the quadratic:

$$(4-\lambda)^2 - 72 = 0$$

which we solve to find

$$(4-\lambda)^2 = 72$$

$$\lambda = 4 \pm \sqrt{72} = 4 \pm 6\sqrt{2}$$

Given these two eigenvalues, we compute the corresponding eigenvectors as follows. First, we substitute the value $\lambda = 4 \pm 6\sqrt{2}$ in Equation 3.15 to get

$$\begin{aligned} (4-4-6\sqrt{2})x_1 + 6x_2 &= 0 \\ 12x_1 + (4-4-6\sqrt{2})x_2 &= 0 \end{aligned}$$

$$-6\sqrt{2}x_1 + 6x_2 = 0$$
$$12x_1 - 6\sqrt{2}x_2 = 0$$

which both reduce to the equation

$$\sqrt{2}x_1 - x_2 = 0$$

We have two variables and only one equation, which corresponds to an infinite number of solution eigenvectors, parametrized by a single free variable, represented as $\begin{bmatrix} x_1 \\ \sqrt{2}x_1 \end{bmatrix}$. For instance, if we set $x_1 = 1$, one possible eigenvector is $\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$. The set of eigenvectors can also be represented as $a\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$, where a is an arbitrary scalar.

As a check, note that $\boldsymbol{Ax} = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} = \begin{bmatrix} 4+6\sqrt{2} \\ 12+4\sqrt{2} \end{bmatrix}$ and $\lambda\boldsymbol{x} = (4+6\sqrt{7}) * \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} = \begin{bmatrix} 4+6\sqrt{2} \\ 12+4\sqrt{2} \end{bmatrix}$.

Erratum 3.1: Read (4 + 6 √2).

We interpret this geometrically as follows. Suppose that we represent the Cartesian coordinates of a point (x, y) by the vector $\begin{bmatrix} x \\ y \end{bmatrix}$. Then, the eigenvectors parametrized by x_1 as $\begin{bmatrix} x_1 \\ \sqrt{2}x_1 \end{bmatrix}$ are the set of points that lie on the line $y = \sqrt{2}x$. Points that lie on this line are transformed by the matrix to other points *also on the same line*, because the effect of the matrix on its eigenvector is to act like a scalar, which does not change the direction of a vector. Moreover, the *degree* of scaling is the associated eigenvalue of $4 + \sqrt{72}$. That is, a point at a unit distance from the origin and on this line (a unit eigenvector) would be scaled to a point on the line that is a distance of $4 + \sqrt{72}$ from the origin. This is shown in Figure 3.1.

To obtain the other eigenvectors, we substitute the value $\lambda = 4 - \sqrt{72}$ in Equation 3.15 to get

$$(4 - 4 + \sqrt{72})x_1 + 6x_2 = 0$$
$$12x_1 + (4 - 4 + \sqrt{72})x_2 = 0$$

$$\sqrt{72}x_1 + 6x_2 = 0$$
$$12x_1 + \sqrt{72}x_2 = 0$$

This gives us the parametrized eigenvector solution, $\begin{bmatrix} x_1 \\ -\sqrt{2}x_1 \end{bmatrix}$, which can be represented as $a\begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix}$, where a is an arbitrary scalar.

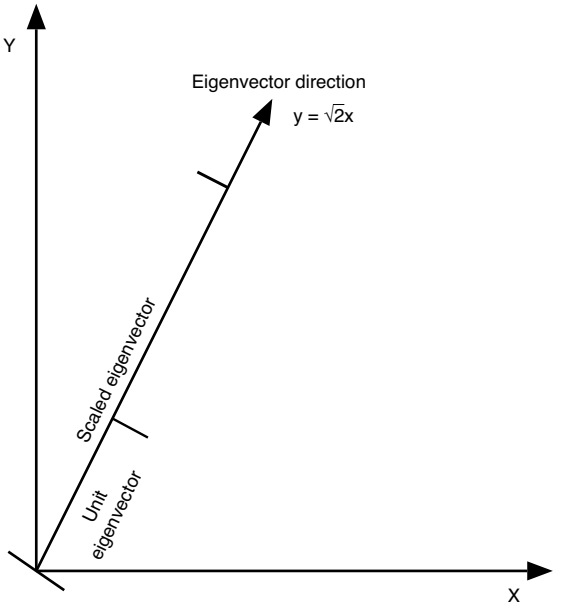


Figure 3.1 The effect of a matrix on an eigenvector is to scale it, preserving its direction.

3.5.3 Computing the Eigenvalues of a Matrix

We can compute the eigenvalues and eigenvectors of a matrix by generalizing the method of the previous example. Consider a matrix $\boldsymbol{A}$ with an eigenvector $\boldsymbol{x}$ and a corresponding eigenvalue λ. We know, by definition, that

$$\boldsymbol{Ax} = \lambda \boldsymbol{x}$$

We rewrite this as

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \qquad \textbf{(EQ 3.16)}$$

This is a homogeneous system of equations, so from Section 3.4.5, it has nontrivial solutions only if the determinant of the coefficient matrix is zero:

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \qquad \textbf{(EQ 3.17)}$$

This determinant is called the **characteristic determinant** of the matrix, and Equation 3.17 is called its **characteristic equation**. Expanding the determinant will result, in general, in obtaining a polynomial of the nth degree in λ, which is the **characteristic polynomial** of the matrix. As we have seen, the eigenvalues of the matrix are the roots of the characteristic polynomial. This important result allows us to compute the eigenvalues of any matrix. Note also that the value of the characteristic determinant of a matrix does not change if the matrix is transposed. Hence, the eigenvalues of a matrix and its transpose are identical.

In general, the roots of a polynomial of degree n can be real or complex. Moreover, the fundamental theorem of algebra tells us that there is at least one root and at most n distinct roots. Therefore, a square matrix of degree n has between one and n distinct eigenvalues, some of which may be complex and will form complex conjugate pairs if the matrix is real. Moreover, some eigenvalues may be repeated. Each eigenvalue corresponds to a family of eigenvectors that are parametrized by at least one free variable.

The set of eigenvalues of a matrix are called its **spectrum**. The largest eigenvalue by magnitude is called the **principal eigenvalue**, or the **spectral radius** of the matrix. Each eigenvalue corresponds to a family of eigenvectors. It can be shown that the set of eigenvectors corresponding to a set of *distinct* eigenvalues are always linearly independent of each other. The set of all vectors that can be expressed as a linear combination of the eigenvectors is called the **eigenspace** of the matrix.

Erratum 3.2: Read "is called".

EXAMPLE 3.16: COMPLEX EIGENVALUES

Let us compute eigenvalues for the matrix $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. We do so by setting the characteristic determinant to 0:

$$\begin{vmatrix} -\lambda & 1 \\ -1 & -\lambda \end{vmatrix} = 0$$

This gives us the characteristic equation

$$\lambda^2 + 1 = 0$$

so that $\lambda = \pm i$. This is the spectrum of the matrix, and the spectral radius is 1.

We find the eigenvector corresponding to $\lambda = i$ by setting

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$x_2 = ix_1$$
$$-x_1 = ix_2$$

which is a set of equations with rank 1; that is, the second equation just restates the first one. One possible vector that satisfies this is $\begin{bmatrix} 1 \\ i \end{bmatrix}$. To check this, note that $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ i \end{bmatrix} = \begin{bmatrix} i \\ -1 \end{bmatrix} = i \begin{bmatrix} 1 \\ i \end{bmatrix}$. The eigenvector corresponding to the eigenvalue $-i$ can be similarly found to be $\begin{bmatrix} x_1 \\ -ix_1 \end{bmatrix}$.

Because both eigenvector families are complex, the matrix never leaves the direction of a real-valued vector unchanged. Indeed, the matrix corresponds to a rotation by 90 degrees, so this is expected. What is unexpected is that the rotation matrix does leave the direction of a complex-valued vector unchanged, which has no obvious intuitive explanation.

EXAMPLE 3.17: EIGENVALUES OF A DIAGONAL MATRIX

Consider a diagonal matrix, such as $\boldsymbol{A} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$. What are its eigenvalues?

The characteristic equation is

$$\begin{vmatrix} a-\lambda & 0 & 0 \\ 0 & b-\lambda & 0 \\ 0 & 0 & c-\lambda \end{vmatrix} = 0$$

A little work shows that this reduces to the equation

$$(a-\lambda)(b-\lambda)(c-\lambda) = 0$$

which shows that the eigenvalues are simply the diagonal elements a, b, and c. This generalizes: We can read off the eigenvalues as the diagonal elements of any diagonal matrix.

3.5.4 The Importance of Eigenvalues

The eigenvalues of a matrix become important when we consider the *repeated* application of a transformation to an input vector. Suppose that we represent the state of a system by a vector, where each element of the vector corresponds to some aspect of the system, such as the buffer occupancy level (discussed at greater length in Section 8.2.2). Suppose further that we can represent the transformation of the state in one time step as being equivalent to the application of a state transformation operator (and the equivalent matrix) on the state vector. Then, the steady-state, or eventual state, of the system can be obtained by the repeated application of the transformation matrix on the initial vector. It turns out that the eigenvalues of the transformation matrix can be used to characterize the steady state.

To see this, first consider the repeated application of a matrix $\boldsymbol{A}$ to its eigenvector $\boldsymbol{x}$ corresponding to an eigenvalue λ. By definition, a single application of $\boldsymbol{A}$ to $\boldsymbol{x}$ = $\boldsymbol{Ax} = \lambda \boldsymbol{x}$. Applying $\boldsymbol{A}$ n times therefore reduces to computing $\lambda^n \boldsymbol{x}$, which is far simpler!

Now, consider an initial state vector v that can be represented as a linear combination of two eigenvectors of $\boldsymbol{A}$—say, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$—like so:

$$\boldsymbol{v} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2$$

Suppose that these eigenvectors correspond to the eigenvalues λ_1 and λ_2. Then, $\boldsymbol{A}^n v$ can be found as follows:

$$\boldsymbol{A}^n\boldsymbol{v} = \boldsymbol{A}^n(c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2) = c_1\boldsymbol{A}^n\boldsymbol{x}_1 + c_2\boldsymbol{A}^n\boldsymbol{x}_2 = c_1\lambda_1^n\boldsymbol{x}_1 + c_2\lambda_2^n\boldsymbol{x}_2$$

We see that the repeated application of $\boldsymbol{A}$ on v, which is a complex operation, is replaced by the far simpler operation of raising a scalar to the nth power.

This intuition is easily generalized. If we can represent the initial vector as the linear combination of the eigenvectors of a matrix, the repeated application of the matrix can be found with little effort, as the next example shows.

EXAMPLE 3.18: COMPUTING $A^N X$ USING EIGENVECTORS

From the previous example, we know that the matrix $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ has two eigenvalues i, and $-i$, with corresponding unit eigenvectors $\begin{bmatrix} 1 \\ i \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -i \end{bmatrix}$. Consider a state vector $\begin{bmatrix} 10 \\ 0 \end{bmatrix}$. We can represent this as $5 * \begin{bmatrix} 1 \\ i \end{bmatrix} + 5 * \begin{bmatrix} 1 \\ -i \end{bmatrix}$. Therefore, $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}^{100}$

Erratum 3.3: See correction 3.3.

$\begin{bmatrix}10\\0\end{bmatrix}$, which is computationally complex, reduces to computing $5*i^{100} * \begin{bmatrix}1\\i\end{bmatrix} + 5*i^{100} * \begin{bmatrix}1\\-i\end{bmatrix} = 5 * \begin{bmatrix}1\\i\end{bmatrix} + 5 * \begin{bmatrix}1\\-i\end{bmatrix} = \begin{bmatrix}10\\0\end{bmatrix}$. Geometrically, this makes sense, because applying $\boldsymbol{A}$ once corresponds to a 90 degree rotation, so applying it for any multiple of four times will leave the initial vector unchanged.

This method of computing the effect of a repeated linear transformation is useful when the initial vector can be written as a linear combination of the eigenvectors. In this context, the following fact is useful: The eigenvectors corresponding to a set of *distinct* eigenvalues form a **linearly independent set**. We have already seen that any set of n linearly independent complex vectors form a basis for the complex hyperspace C^n. Hence, if a square matrix of order n has n distinct eigenvalues, we can express any initial vector as a linear combination of its eigenvectors. This is one reason why knowing the eigenvalues of a matrix is very useful.

What if the initial state vector cannot be written as a linear combination of the eigenvectors? That is, what if the initial state vector is not in the eigenspace of the matrix? It turns out that a somewhat more complex computation, still involving the eigenvalues, allows us to compute $\boldsymbol{A}^n\boldsymbol{x}$.

3.5.5 The Role of the Principal Eigenvalue

Consider a matrix $\boldsymbol{A}$ that has a set of m eigenvalues λ_i, $i = 1,2, ..., m$ with corresponding unit eigenvectors $\boldsymbol{x}_i$. Suppose that we choose a vector $\boldsymbol{v}$ in the eigenspace of the matrix such that it is expressed as a linear combination of *all* the eigenvectors:

$$\boldsymbol{v} = \sum_{i=1}^{m} c_i\boldsymbol{x}_i$$

Then, n applications of the matrix to this vector results in the vector

$$\sum_{i=1}^{m} c_i\lambda_i^n\boldsymbol{x}_i$$

As $n \to \infty$, this sum is dominated by the eigenvalue that has the largest magnitude, which is called the **principal**, or **dominant**, eigenvalue. (If more than one eigenvalue has the same magnitude, they are both considered to be the principal eigenvalues.) To first approximation, we can ignore all the other eigenvalues in computing the limit.

EXAMPLE 3.19: PRINCIPAL EIGENVALUE

Consider the matrix $A = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}$. Recall from Example 3.15 that it has two eigenvalues, $\lambda = 4 \pm \sqrt{72}$, which evaluate to 12.48 and –4.48. The unit eigenvectors of this matrix are $\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$ and $\begin{bmatrix} 1 \\ \frac{-1}{\sqrt{2}} \end{bmatrix}$. Suppose that we start with an initial vector $\begin{bmatrix} 0 \\ 3\sqrt{2} \end{bmatrix}$, which we can express as $2 * \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} - 2 * \begin{bmatrix} 1 \\ \frac{-1}{\sqrt{2}} \end{bmatrix}$. Then, $A^{10} \begin{bmatrix} 0 \\ 3\sqrt{2} \end{bmatrix} = 2 * 12.48^{10} * \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} - 2*(-4.48)^{10} * \begin{bmatrix} 1 \\ \frac{-1}{\sqrt{2}} \end{bmatrix}$. This evaluates to $\begin{bmatrix} 1.83*10^{11} \\ 2.59*10^{11} \end{bmatrix}$. If we ignore the second term (i.e, the contribution due to the eigenvalue –4.48), the resulting value changes only in the third decimal place of precision. It is clear that the dominant eigenvalue is the one that matters.

Erratum 3.4: Second eigenvector is $[1; -\sqrt{2}]$; see correction 3.4.

3.5.6 Finding Eigenvalues and Eigenvectors

Computing the eigenvalues and the eigenvectors of a matrix is important in practice. Here are some facts that help identify the nature of the eigenvalues of a matrix.

- If a matrix is square and diagonal, its eigenvalues are its diagonal elements.
- If a matrix is square and symmetric (that is, $\boldsymbol{A}^T = \boldsymbol{A}$), its eigenvalues are real.
- **Gerschgorin's "circle" theorem** states that all the eigenvalues of a complex matrix lie in the set of disks—on the complex plane—centered on the elements of the diagonal, with a radius equal to the sum of the magnitudes of the off-diagonal elements. Intuitively, if the off-diagonal elements are "not too large," the eigenvalues of the matrix are its diagonal elements.

Erratum 3.5: Radius: off-diagonal sum of that row.

EXAMPLE 3.20: FINDING EIGENVALUES

The matrix $\boldsymbol{A} = \begin{bmatrix} 9.1 & 0.8 & 0.3 \\ 0.8 & 5.8 & 0.2 \\ 0.3 & 0.2 & 6.5 \end{bmatrix}$ is symmetric. Hence, its eigenvalues are real. It has three Gerschgorin disks: (1) center $9.1 + 0i$, radius 1.1, (2) center $5.8 + 0i$ and radius 1.0, and (3) center $6.5 + 0i$ and radius 0.5. Because the eigenvalues are real, they must lie in one of three intervals: [8, 10.2], [4.8, 6.8], and [6, 7]. The second interval overlaps the third, so we know that the eigenvalues lie in either the interval [4.8 7] or the interval [8 10.2].

It is possible to approximately compute the dominant eigenvalue of a matrix[2] by using the **power** method. In this technique, we start with an *arbitrary* initial vector $\boldsymbol{x}_0$ and repeatedly apply $\boldsymbol{A}$ to it. At each step, we compute the **Rayleigh ratio** $\frac{\boldsymbol{x}_k^T \boldsymbol{A} \boldsymbol{x}_k}{\boldsymbol{x}_k^T \boldsymbol{x}_k} = \frac{\boldsymbol{x}^T \boldsymbol{x}_{k+1}}{\boldsymbol{x}_k^T \boldsymbol{x}_k}$, which converges toward the dominant eigenvalue of the matrix. The intuitive idea is that applying $\boldsymbol{A}$ to any vector scales it in multiple dimensions, but the dominant eigenvalue dominates the scaling effect. Repeatedly applying $\boldsymbol{A}$ magnifies the contribution of the dominant eigenvalue, exposing it.

EXAMPLE 3.21: POWER METHOD TO COMPUTE THE DOMINANT EIGENVALUE

Use the power method to compute the dominant eigenvalue of the matrix

$$\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}.$$

Solution:

Suppose that we arbitrarily start with the initial vector $x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Applying the matrix once, we get $\boldsymbol{x}_1 = \begin{bmatrix} 10 \\ 16 \end{bmatrix}$. The Rayleigh ratio evaluates to ([1 1] * $\begin{bmatrix} 10 \\ 16 \end{bmatrix}$)/([1 1] * $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$) = 26/2 = 13. Repeating, we get $x_2 = \begin{bmatrix} 136 \\ 184 \end{bmatrix}$, and the corresponding Rayleigh

2. Usually, but not necessarily, the dominant eigenvalue. It turns out that in nearly all practical cases, this is the one that will be found.

ratio evaluates to 4304/356 = 12.08. After one more iteration, we get $x_3 = \begin{bmatrix} 1648 \\ 2368 \end{bmatrix}$, and the Rayleigh ratio evaluates to 659840/52352 = 12.60. Recall from Example 3.19 that the dominant eigenvalue is 12.48. We obtain an approximation to within 1% of this value in only three iterations.

It can be shown that the speed of convergence of this method depends on the gap between the dominant and the second-largest eigenvalue. The bigger the gap, the faster the convergence. Intuitively, if the second-largest eigenvalue is close in magnitude to the dominant eigenvalue, its scaling effects do not die down easily, requiring more iterations.

The power method can also be used to find the dominant *eigenvector*. To do so, after each iteration, the vector x_i must be *scaled*. That is, we set its largest element to 1 by dividing each element by the largest element, as demonstrated next.

EXAMPLE 3.22: POWER METHOD FOR COMPUTING THE DOMINANT EIGENVECTOR

Compute the dominant eigenvector of the matrix $\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}$ by using the power method.

Solution:

From Example 3.21, we already know that $\boldsymbol{x}_1 = \begin{bmatrix} 10 \\ 16 \end{bmatrix}$. We rescale it by dividing each element by 16 to get the vector $\begin{bmatrix} 0.625 \\ 1 \end{bmatrix}$. Using this as the new value of $\boldsymbol{x}_1$, we get $\boldsymbol{x}_2 = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix} * \begin{bmatrix} 0.625 \\ 1 \end{bmatrix} = \begin{bmatrix} 8.5 \\ 11.5 \end{bmatrix}$. We rescale this again to get $\boldsymbol{x}_2 = \begin{bmatrix} 0.739 \\ 1 \end{bmatrix}$. This allows us to compute $\boldsymbol{x}_3 = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix} * \begin{bmatrix} 0.739 \\ 1 \end{bmatrix} = \begin{bmatrix} 8.956 \\ 12.868 \end{bmatrix}$, which is rescaled to $\begin{bmatrix} 0.696 \\ 1 \end{bmatrix}$. Recall that the eigenvector for this eigenvalue is exactly

$\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$, which we scale to $\begin{bmatrix} 0.707 \\ 1 \end{bmatrix}$. As before, with just three iterations, we are within 1.5% of this final value.

3.5.7 Similarity and Diagonalization

Erratum 3.6: See correction 3.6.

Two matrices are said to be **similar** if they have the same set of eigenvalues. In some cases, given a matrix $\boldsymbol{A}$, it is useful to be able to compute a similar **diagonal** matrix $\boldsymbol{D}$. (We show a use case in Example 3.23.)

A sufficient, but not necessary, condition for a matrix of size n to be diagonalizable is that it has n distinct eigenvalues; to see that this condition is not necessary, consider a diagonal matrix with repeated diagonal elements. In this case, let $\boldsymbol{X}$ denote a matrix whose columns are the eigenvectors of $\boldsymbol{A}$. Then, it can be easily shown, by expansion of the underlying terms, that the matrix

$$\boldsymbol{D} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} \qquad \text{(EQ 3.18)}$$

is a diagonal matrix whose diagonal elements *are the eigenvalues of* $\boldsymbol{A}$.

Knowing the diagonalized version of a matrix makes it trivial to compute its mth power. From Equation 3.18, note that

$$\boldsymbol{D}^2 = (\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X})(\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{X}^{-1}\boldsymbol{A}^2\boldsymbol{X}$$

A simple induction shows that

$$\boldsymbol{D}^m = \boldsymbol{X}^{-1}\boldsymbol{A}^m\boldsymbol{X}$$

so that

$$\boldsymbol{A}^m = \boldsymbol{X}\boldsymbol{D}^m\mathbf{X}^{-1}$$

But the right-hand side is easily computed because $\boldsymbol{D}$ is diagonal. Hence, we can easily compute $\boldsymbol{A}^m$.

EXAMPLE 3.23: DIAGONALIZATION

Consider the matrix $\boldsymbol{A} = \begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}$. Recall from Example 3.15 that it has two eigenvalues, $\lambda = 4 \pm 6\sqrt{2}$, corresponding to the eigenvectors $\begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix}$.

This allows us to write out the matrix $\boldsymbol{X}$ as $\begin{bmatrix} 1 & 1 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix}$. From Equation 3.13, we find that $\boldsymbol{X}^{-1} = \frac{1}{(-\sqrt{2}-\sqrt{2})}\begin{bmatrix} -\sqrt{2} & -1 \\ -\sqrt{2} & 1 \end{bmatrix}$. Therefore, we diagonalize $\boldsymbol{A}$ as

$$\frac{1}{-2\sqrt{2}}\begin{bmatrix} -\sqrt{2} & -1 \\ -\sqrt{2} & 1 \end{bmatrix}\begin{bmatrix} 4 & 6 \\ 12 & 4 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{2} & 1 \\ \sqrt{2} & -1 \end{bmatrix}\begin{bmatrix} 2 & 3 \\ 6 & 2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix} = \begin{bmatrix} 4+6\sqrt{2} & 0 \\ 0 & 4-6\sqrt{2} \end{bmatrix}$$

Note that the diagonal elements of $\boldsymbol{A}$ are its eigenvalues.

Erratum 3.7: Read "elements of D"; "power" for "exponential".

Given this transformation, we compute $\boldsymbol{A}^5$ as

$$\boldsymbol{X}\boldsymbol{D}^m\boldsymbol{X}^{-1} = \begin{bmatrix} 1 & 1 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix}\begin{bmatrix} 4+6\sqrt{2} & 0 \\ 0 & 4-6\sqrt{2} \end{bmatrix}^5 \frac{1}{-2\sqrt{2}}\begin{bmatrix} -\sqrt{2} & -1 \\ -\sqrt{2} & 1 \end{bmatrix}$$

Due to diagonalization, the matrix power computation reduces to computing the exponential of a scalar value. After simplification and maintaining ten digits of precision in the calculations, this reduces to

$$\begin{bmatrix} 150783.99 & 107903.99 \\ 215807.99 & 150783.99 \end{bmatrix}$$

which is within rounding error of the true value of

$$\begin{bmatrix} 150784 & 107904 \\ 215808 & 150784 \end{bmatrix}$$

If the matrix has repeated eigenvalues, it cannot be diagonalized. Instead, the best we can do is to put it in the **Jordan canonical form**, whereby the diagonal elements are the eigenvalues of the matrix, as with a diagonal matrix. However, some elements immediately above the main diagonal may also be 1s. More details on the computation of this form can be found in advanced texts on linear algebra.

Erratum 3.8: See correction 3.8.

3.6 Stochastic Matrices

A **stochastic matrix**, or **Markov matrix**, is a special kind of matrix. A **right stochastic matrix** is a square matrix whose elements are non-negative reals and

each of whose *rows* sums to 1. A **left stochastic matrix** is a square matrix whose elements are non-negative reals and each of whose *columns* sums to 1. Unless otherwise specified, when we refer to a stochastic matrix, we refer to a *right* stochastic matrix.

Stochastic matrices are important in the context of computer networking because each row of such a matrix $\boldsymbol{A}$ corresponds to the state of a finite state machine, or Markov chain, representing a networking protocol or a buffer in a router. Each element a_{ij} can be viewed as the probability of entering state j from state i. The summation criterion expresses the fact that the result of a transition from a state is to either remain in the same state or go to some other state. Stochastic matrices arise frequently in the study of Markov chains, stochastic processes, and queueing theory.

EXAMPLE 3.24: STOCHASTIC MATRIX

The matrix $\boldsymbol{A} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.1 & 0.9 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$ is a stochastic matrix because it is square, the elements are non-negative reals, and each row sums to 1. We interpret row 1 to mean that if the system is in state 1, after one transition, it remains in state 1 with probability 0.25, goes to state 2 with probability 0.5, and goes to state 3 with probability 0.25. Note that if the system enters state 3, it can never leave that state. We call such a state an **absorbing** state.

3.6.1 Computing State Transitions by Using a Stochastic Matrix

Consider a $n \times n$ stochastic matrix $\boldsymbol{A}$ and a column vector $\boldsymbol{p}$ having dimension n with non-negative real elements such that its elements sum to 1. We think of the ith element of $\boldsymbol{p}$ as representing the probability of being in state i at some point in time. Then, $\boldsymbol{p'} = \boldsymbol{A}^T\boldsymbol{p}$ is a vector whose ith element is the probability of being in state i after one transition. (Note the premultiplication not by $\boldsymbol{A}$ but by its transpose.) The reason is that the ith element of $\boldsymbol{p'}$ is given by $p'_i = \sum_{k=1}^{n} p_k a_{ki}$, which, from the law of probability, is the probability of state i after the transition, conditional on the probability of being in each prior state k.

EXAMPLE 3.25: STATE TRANSITIONS

Continuing with the stochastic matrix from Example 3.24, suppose that we start with the system in state 1. What is the probability of being in state 1 after one and two transitions?

Solution:

The initial state vector is $\boldsymbol{p} = [1.0\ 0\ 0]^T$. After one transition, the state vector is given by

$$\boldsymbol{p} = \boldsymbol{A}^T\boldsymbol{p} = \begin{bmatrix} 0.25 & 0.1 & 0 \\ 0.5 & 0.9 & 0 \\ 0.25 & 0 & 1.0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.5 \\ 0.25 \end{bmatrix}$$

and after two transitions,

$$\boldsymbol{p} = \boldsymbol{A}^T(\boldsymbol{A}^T\boldsymbol{p}) = (\boldsymbol{A}^T)^2\boldsymbol{p} = \begin{bmatrix} 0.1125 & 0.115 & 0 \\ 0.575 & 0.86 & 0 \\ 0.3125 & 0 & 1.0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.1125 \\ 0.575 \\ 0.3125 \end{bmatrix}$$

Erratum 3.9: Row 3, column 2 is 0.025; see correction 3.9.

Thus, after two transitions, the system is in state 1 with probability 0.1125. Note that this probability is larger than the simple probability of staying in state 1, which is just 0.25 * 0.25 = 0.0625, because it takes into account the probability of transitioning from state 1 to state 2 and then back from state 2 to state 1, which has an additional probability of 0.5*0.1 = 0.05.

As this example shows, if $\boldsymbol{A}$ is a stochastic matrix, the $[i,j]$th element of $(\boldsymbol{A}^T)^2$ represents the probability of going from state i to state j in two steps. Generalizing, the probability of going from state i to state j in k steps is given by $(\boldsymbol{A}^T)^k$.

3.6.2 Eigenvalues of a Stochastic Matrix

We now present three important results concerning stochastic matrices.

First, *every* stochastic matrix has an eigenvalue of 1. To prove this, consider the $n \times n$ stochastic matrix A and column vector $\boldsymbol{x} = [1\ 1\ \ldots\ 1]^T$. Then, $\boldsymbol{Ax} = \boldsymbol{x}$, because each element of $\boldsymbol{Ax}$ multiplies 1 with the sum of a row of $\boldsymbol{A}$, which, by definition, sums to 1. Because a matrix and its transpose have the same eigenvalues, the transpose of a stochastic matrix also has an eigenvalue of 1. However, the eigenvector of the transposed matrix corresponding to this eigenvalue need not be, and rarely is, the **1** vector.

Second, every possibly complex eigenvalue of a stochastic matrix must have a magnitude no greater than 1. To prove this, consider some diagonal element a_{jj}. Suppose that this element takes the value x. Then, by definition of a stochastic matrix, it must be the case that the sum of the off-diagonal elements is $1 - x$. From Gerschgorin's circle theorem, we know that all the eigenvalues lie within a circle in the complex plane centered at x with radius $1 - x$. The largest-magnitude eigenvalue will be a point on this circle (see Figure 3.2). Although the truth of the proposition is evident by inspection, we now formally prove the result.

Erratum 3.10: Each eigenvalue lies in some row's disk.

Suppose that this point subtends an angle of θ. Then, its coordinates are $(x+(1-x)\cos\theta, (1-x)\sin\theta)$. Therefore, its magnitude is $((x+(1-x)\cos\theta)^2+((1-x)\sin\theta)^2)^{1/2}$, which simplifies to $(x^2 + (1-x)^2+2x(1-x)\cos\theta)^{1/2}$. This quantity is maximized when $\theta = 0$, so that we merely have to maximize the quantity $x^2 + (1-x)^2+2x(1-x)$. Taking the first derivative with respect to x and setting it to zero shows that this expression reaches its maximum of 1 independent of the value of x. So, we can pick x to be a convenient value, such as 1. Substituting $\theta = 0$ and $x = 1$ into $((x+(1-x)\cos\theta)^2+((1-x)\sin\theta)^2)^{1/2}$, we find that the magnitude of the maximum eigenvalue is 1.

Third, it can also be shown, although the proof is beyond the scope of this text, that under some mild assumptions,[3] only *one* of the eigenvalues of a stochastic matrix is 1, which also must therefore be its dominant eigenvalue.

These three facts lead to a remarkable insight. Recall that we compute one step in the state evolution of a stochastic system described by the transition matrix $\boldsymbol{A}$ by multiplying the state vector by $\boldsymbol{A}^T$. We already know that, under some mild assumptions, every stochastic transition matrix $\boldsymbol{A}$ has a unique eigenvalue of 1. We also

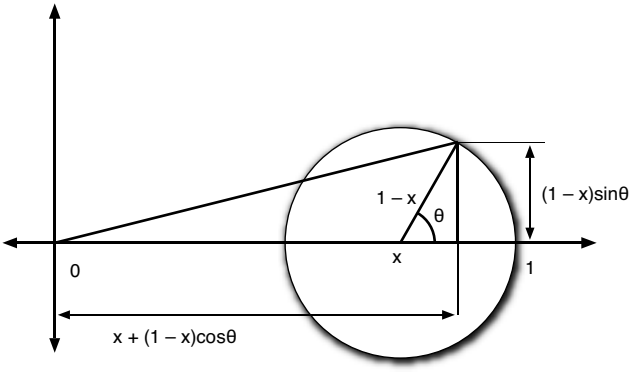


Figure 3.2 Largest-possible eigenvalue of a stochastic matrix

3. These assumptions eliminate such matrices as the identity matrix, which is a degenerate Markov matrix in that all its states are absorbing states.

know that a matrix and its transpose have the same eigenvalues. Therefore, $\boldsymbol{A}^T$ also has a unique eigenvalue of 1. What if the state vector was the eigenvector of $\boldsymbol{A}^T$ corresponding to this eigenvalue? Then, one step in the state evolution would leave the corresponding eigenvector *unchanged*. That is, a system in a state corresponding to that eigenvector would never leave that state. We denote this special state by the vector $\boldsymbol{\pi}$. It is also called the **stationary probability distribution** of the system. The vector $\boldsymbol{\pi}$ can be found by solving the system of linear equations given by

$$A^T\pi = \pi \qquad \textbf{(EQ 3.19)}$$

Because every stochastic matrix has an eigenvalue of 1, every stochastic system must have a unique stationary probability distribution, which is the eigenvector corresponding to the unique eigenvalue of 1. Instead of solving a system of linear equations, we can also compute the stationary probability distribution of $\boldsymbol{A}$ by using the power method, because 1 is also its *dominant* eigenvalue. We put these results together to state that *the power method can be used to compute the stationary probability distribution of a stochastic matrix.* In our study of queueing theory, we will examine the conditions on the matrix $\boldsymbol{A}$ that guarantee that its stationary probability distribution is reached independent of the initial state of the system. Roughly speaking, these are the conditions under which the matrix $\boldsymbol{A}^T$ is said to be **ergodic**.

Erratum 3.11: Uniqueness needs the assumptions on page 141.

EXAMPLE 3.26: GOOGLE PAGERANK ALGORITHM

The power technique of finding the dominant eigenvector of a stochastic matrix can be used to rank a set of Web pages. More precisely, given a set of Web pages, we would like to identify certain pages as being more important than others. A page can be considered to be important by using the recursive definition that (1) many other pages point to it, and (2) the other pages are also important.

The importance of a page can be quantified according to the actions of a random Web surfer who goes from Web page i to a linked Web page j with probability a_{ij}. If a page is important, a random Web surfer will be led to that page more often than to other, less-important pages. That is, if we consider a population of a large number of surfers, then a larger fraction of Web surfers will be at a more important page, compared to a less-important page. Treating the ratio of the number of Web surfers at a page to the total number of surfers as approximating a probability, we see that the importance of a page is just the stationary probability of being at that page.

To make matters more precise, let the matrix $\boldsymbol{A}$ represent the set of all possible transition probabilities. If the probability of the surfer being at page i at

some point is p_i, the probability that the surfer is at page i after one time step is $\boldsymbol{A}^T p_i$. The dominant eigenvector of $\boldsymbol{A}^T$ is then the steady state probability of a surfer being at page i. Given that $\boldsymbol{A}$ is a stochastic matrix, we know that this dominant eigenvector exists and that it can be found by the power method.

Erratum 3.12: Read $(A^T p)_i$; "dominant eigenvector".

What remains is to estimate the quantities a_{ij}. Suppose that page i has links to k pages. Then, we set $a_{ij} = 1/k$ for each page j to which it has a link and set $a_{ij} = 0$ for all other j. This models a surfer going from a page uniformly randomly to one of its linked pages. What if a page has no links? Or if two pages link only to each other? These issues can be approximately modeled by assuming that, with constant probability, the surfer teleports to a randomly chosen page. That is, if there is a link from page i to page j, then $a_{ij} = \alpha/n + (1 - \alpha)/k$, where α is a control parameter; otherwise, $a_{ij} = \alpha/n$. It can be easily shown that these modified a_{ij}s form a stochastic matrix, so that we can extract the dominant eigenvalue, and thus the page rank, by using the power method. A slightly modified version of this algorithm is the publicly described algorithm used by Google to rank Web pages.[4]

---

4. A rather curious fact is that Google's algorithm to rank pages, called the **PageRank** algorithm, was developed, in part, by its cofounder, Larry Page.

Corrections to Sections 3.5 and 3.6

The pages of Sections 3.5 and 3.6 are reproduced from the book. The following errata are not yet corrected in the text; each is flagged by a red note in the margin of the page concerned.

3.1 Example 3.15, page 128: in the check, $\lambda x = (4 + 6\sqrt{7}) * \ldots$ should read $(4 + 6\sqrt{2}) * \ldots$.

3.2 Page 130: "The set of eigenvalues of a matrix are called its spectrum" should read "is called its spectrum".

3.3 Example 3.18, pages 132–133: $[1;\ i]$ and $[1;\ -i]$ are eigenvectors but not unit eigenvectors (each has length $\sqrt{2}$). In the expansion, the second term $5 * i^{100} * [1;\ -i]$ should read $5 * (-i)^{100} * [1;\ -i]$. The value is unchanged because $(-i)^{100} = 1$.

3.4 Example 3.19, page 134: the second eigenvector should be $[1;\ -\sqrt{2}]$, as in Example 3.15, not $[1;\ -1/\sqrt{2}]$. Neither vector has unit length. Then $[0;\ 3\sqrt{2}] = 1.5[1;\ \sqrt{2}] - 1.5[1;\ -\sqrt{2}]$, and $A^{10}[0;\ 3\sqrt{2}] = 1.5\lambda_1^{10}[1;\ \sqrt{2}] - 1.5\lambda_2^{10}[1;\ -\sqrt{2}] \approx [1.381 \times 10^{11};\ 1.953 \times 10^{11}]$, where $\lambda_1 = 4 + \sqrt{72} \approx 12.485$ and $\lambda_2 = 4 - \sqrt{72} \approx -4.485$. Ignoring the second term changes the result only in the fifth significant figure, so the conclusion of the example stands.

3.5 Page 134: each Gerschgorin disk is centred on a diagonal element, and its radius is the sum of the magnitudes of the off-diagonal elements in that row. Every eigenvalue lies in at least one of the disks.

3.6 Section 3.5.7, page 137: the definition of similarity is too informal. Two $n \times n$ matrices A and B are similar if $B = P^{-1}AP$ for some invertible matrix P. Similar matrices have the same eigenvalues, but matrices with the same eigenvalues need not be similar. For example, I and $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ both have the eigenvalue 1, repeated, but the only matrix similar to I is I itself.

3.7 Example 3.23, page 138: "Note that the diagonal elements of A are its eigenvalues" should read "... of D ...". "reduces to computing the exponential of a scalar value" should read "... the power of a scalar value". (Equation 3.13, the inverse of a 2×2 matrix, is not in these notes.)

3.8 Section 3.5.7, page 138: "If the matrix has repeated eigenvalues, it cannot be diagonalized" should read "If the matrix does not have n linearly independent eigenvectors, it cannot be diagonalized." A matrix with repeated eigenvalues may still be

diagonalizable, for example a diagonal matrix with repeated diagonal elements, as noted on page 137.

3.9 Example 3.25, page 140: in $(A^T)^2$, the entry in row 3, column 2 should be 0.025, not 0. After the example, the probability of going from state i to state j in two steps is the $[j, i]$th element of $(A^T)^2$ (equivalently, the $[i, j]$th element of A^2), and similarly for k steps.

3.10 Page 141: one row's Gerschgorin disk does not contain all the eigenvalues. Each eigenvalue lies in at least one disk, and for a stochastic matrix every disk lies inside the unit circle, which proves the result.

3.11 Page 142: "Because every stochastic matrix has an eigenvalue of 1, every stochastic system must have a unique stationary probability distribution" is too strong. Uniqueness needs the mild assumptions of page 141 (for example, an irreducible chain). The power method also needs no other eigenvalue of magnitude 1; a periodic chain has an eigenvalue of -1.

3.12 Example 3.26, page 143: "the probability that the surfer is at page i after one time step is $A^T p_i$" should read $(A^T p)_i$, and "extract the dominant eigenvalue, and thus the page rank" should read "extract the dominant eigenvector, and thus the page rank".

8.11 Partial Fraction Expansion

Partial fraction expansion allows a complex function $f(s)$ that is the ratio of two polynomials in the complex variable s to be represented in an alternative form to which one can easily apply the inverse Laplace transform. To begin with, let

$$f(s) = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_m}{s^n + a_1 s^{n-1+\ldots+a_n}} = \frac{N(s)}{D(s)}$$

We first find the roots of $D(s)$ by solving the equation $D(s) = 0$, usually computed numerically by using such algorithms as the Jenkins-Traub algorithm. Let these roots be $\alpha_1, \alpha_2, \ldots, \alpha_n$. Then, we can write

$$f(s) = \frac{N(s)}{(s-\alpha_1)(s-\alpha_2)\ldots(s-\alpha_n)}$$

The next steps depend on the nature of the roots. There are three cases: distinct roots, complex conjugate roots, and repeated roots.

8.11.1 Distinct Roots

In this case, we write

$$f(s) = \frac{a_1}{(s-\alpha_1)} + \frac{a_2}{(s-\alpha_2)} + \ldots + \frac{a_n}{(s-\alpha_n)}$$

where $a_i \in C$. The coefficient a_i is given by

$$a_i = \lim_{s \to \alpha_i} (s-\alpha_i) f(s)$$

EXAMPLE 8.20: DISTINCT ROOTS

Find the partial fraction expansion of the polynomial fraction $\frac{1}{s^2+3s+2}$.

Solution:

We write the fraction as

$$f(s) = \frac{1}{(s+1)(s+2)} = \frac{a_1}{s+1} + \frac{a_2}{s+2}.$$

Then,

$$a_1 = \lim_{s \to -1} (s+1)\left(\frac{1}{(s+1)(s+2)}\right) = \frac{1}{-1+2} = 1$$

and

$$a_2 = \lim_{s \to -2} (s+2)\left(\frac{1}{(s+1)(s+2)}\right) = \frac{1}{-2+1} = -1,$$

so that

$$f(s) = \frac{1}{s+1} - \frac{1}{s+2}.$$

8.11.2 Complex Conjugate Roots

Suppose $D(s)$ has one pair of complex conjugate roots: They must be conjugate if the coefficients of $D(s)$ are real. Then, we write

$$f(s) = \frac{a_1}{(s-(\alpha_1 + j\beta_1))} + \frac{a_2}{(s-(\alpha_1 - j\beta_1))} + \ldots + \frac{a_n}{(s-\alpha_n)}$$

It can be shown that a_1 and a_2 are the complex conjugates of each other. Moreover,

$$a_1 = \lim_{s \to (\alpha_1 + j\beta_1)} (s-(\alpha_1 + j\beta_1)) f(s).$$

EXAMPLE 8.21: COMPLEX CONJUGATE ROOTS

Find the partial fraction expansion of the polynomial fraction $\frac{1}{s^2 - 6s + 25}$.

Solution:

We write the fraction as

$$f(s) = \frac{1}{(s-(3+j4))(s-(3-j4))} = \frac{a_1}{(s-(3+j4))} + \frac{a_2}{(s-(3-j4))}.$$

Then,

$$a_1 = \lim_{s \to (3+j4)} (s-(3+j4))\left(\frac{1}{(s-(3+j4))(s-(3-j4))}\right) = \frac{1}{(3+j4-(3-j4))} = \frac{1}{j8}$$

and $a_2 = \frac{1}{-j8}$, so that $f(s) = \frac{1}{j8(s-(3+j4))} - \frac{1}{j8(s-(3-j4))}$.

8.11.3 Repeated Roots

If $D(s)$ has repeated roots, the partial fraction expansion is somewhat more complex. Suppose that $D(s)$ has r repeated roots α_1 and that the other roots are distinct. Then,

Erratum 8.26: Read "a root α_1 repeated r times".

$$f(s) = \frac{a_{1(r)}}{(s-\alpha_1)^r} + \frac{a_{1(r-1)}}{(s-\alpha_1)^{r-1}} + \ldots + \frac{a_{1(1)}}{(s-\alpha_1)} + \frac{a_2}{(s-\alpha_2)} + \ldots + \frac{a_n}{(s-\alpha_n)}$$

where the ith repeated root, $i = 0,1,\ldots, r-1$ is given by

$$a_{1(r-i)} = \lim_{s\to\alpha_1}\left(\frac{1}{i!}\frac{d^i}{ds^i}((s-\alpha_1)^r f(s))\right)$$

EXAMPLE 8.22: REPEATED ROOTS

Find the partial fraction expansion of the polynomial fraction $\frac{1}{s^3+7s^2+16s+12}$.

Solution:

We write the fraction as $f(s) = \frac{1}{(s+2)^2(s+3)} = \frac{a_{1(2)}}{(s+2)^2} + \frac{a_{1(1)}}{s+2} + \frac{a_2}{s+3}$. Then,

$$a_{1(2-0)} = \lim_{s\to -2}\left(\frac{1}{0!}\frac{d^0}{ds^0}((s+2)^2 f(s))\right) = \lim_{s\to -2}\left(\frac{1}{s+3}\right) = 1$$

$$a_{1(2-1)} = \lim_{s\to -2}\left(\frac{1}{1!}\frac{d^1}{ds^1}((s+2)^2 f(s))\right) = \lim_{s\to -2}\left(\frac{d}{ds}\left(\frac{1}{(s+3)}\right)\right)$$

$$= \lim_{s\to -2}\left(\frac{s+3\frac{d}{ds}1 - 1\frac{d}{ds}s+3}{s+3^2}\right) = -1$$

$$a_2 = \lim_{s\to -3}(s+3)f(s) = \lim_{s\to -3}\left(\frac{1}{(s+2)^2}\right) = 1$$

$$f(s) = \frac{1}{(s+2)^2} - \frac{1}{s+2} + \frac{1}{s+3}.$$



Answers to Exercises

This chapter gives worked answers to the exercises for simulation (Section S.10, Exercises 1–6), queueing theory (Section 6.9 and the M/G/1 problems), signals, systems, and transforms (Section 5.12, Exercises 1–18), and control theory (Section 8.13, Exercises 1–8). Each answer restates the question, lists the formulas it uses, with their equation numbers in the notes, and then works through the solution.

A.1 Simulation: Section S.10 Exercises

1. State and counter variables

Question. *Consider a simulator that models the arrival and departure of people from the King's Cross plaza. We would like to study (a) the mean number of people who are in the plaza as well as the dynamic evolution of this quantity, (b) the mean delay to traverse the plaza and the dynamic evolution of this quantity, and (c) the impact of changing the train schedule on the number of people in the plaza and the plaza traversal delay. What would be the system state variables and counter variables necessary to accomplish this task?*

Formulas: System state and counter variables (Section S.2.2).

The state of the system is captured by the number of people in the plaza at any time. This number increases when people arrive to the plaza on their way to the train station or when they leave the train station and decreases when they depart the plaza either to enter the train station or the surrounding streets.

To compute the mean delay to traverse the plaza (which we would expect to be the same in both directions), we will need to measure the delay encountered by each person. This requires us to maintain, for each person, the time when they enter the plaza, and the time when they depart. These constitute the counter variables.

Or, to save space, we can maintain one counter for the total number of people traversing the plaza and the total encountered delay, as well as, for each person currently on the plaza, the time when they entered the plaza. When a person leaves, the cumulative counts can be incremented and the individual counts forgotten.

2. Generating a discrete random variable

Question. *Suppose that the discrete probability distribution over the 5 values x_i, $i \in [1, 5]$, taken by the random variable X, is given by the vector* $[0.1, 0.3, 0.2, 0.1, 0.3]$. *Discuss how you could generate values taken on by this variable.*

Formulas: Inverse transform method for a discrete distribution (EQ S.2, Section S.4.1).

Using the inverse transform method, we first compute the cumulative distribution function F as $[0.1, 0.4, 0.6, 0.7, 1.0]$. Generate u, a $U(0, 1)$ random number, and output x_i if $F_{i-1} \leq u < F_i$, where $F_0 = 0$.

3. Simulating a Poisson process

Question. *Suppose that the first four calls to a $U[0, 1]$ random number generator return the values, to three digits of precision, of 0.127, 0.632, 0.829, and 0.243. Use these to compute the times of the first four arrivals to a Poisson process with parameter $\lambda = 5.0$.*

Formulas: Exponential interarrival times by the inverse transform, $X = -\frac{1}{\lambda}\log(U)$; simulating a Poisson process (Section S.4.5, Algorithm 3).

As in Algorithm 3, start the clock at $t_0 = 0$. Let the values drawn from the uniform random distribution be denoted u_i, $i \in [1, 4]$. Each value gives one exponential interarrival time, so the arrival times are given by $t_i = t_{i-1} - \frac{1}{\lambda}\ln(u_i)$. This gives us the times as: $t_1 = 0.413$, $t_2 = 0.413 + 0.092 = 0.505$, $t_3 = 0.505 + 0.037 = 0.542$, $t_4 = 0.542 + 0.283 = 0.825$.

4. Stopping rule

Question. *Suppose you are using simulations to compute the mean number of people on King's Cross plaza and you would like to have 95% confidence that the true mean lies within ± 10 people of the mean value you estimate from your simulations. What stopping rule should you use to decide that you have run enough simulations?*

Formulas: Sample variance $S_n^2 = \sum_{i=1}^{n}(X_i - \bar{X}_n)^2/(n-1)$ (Section S.6.3); confidence interval (EQ S.20); stopping rule $2z_{\alpha/2}S_n/\sqrt{n} < \ell$ (Section S.7).

Let the mean value of the number of people on the plaza computed by each simulation run be denoted by X_i. First run, say, 100 simulations, for the statistics to stabilize. Then,

after n simulations, let the running mean of these mean values be denoted $\bar{X}_n$. After each simulation, we also compute the quantity

$$S_n^2 = \frac{\sum_i (X_i - \bar{X}_n)^2}{n-1}.$$

The confidence level is 95%, so $\alpha = 0.05$, and we need to look up the values $-z_{0.025}$ and $z_{0.025}$, which are -1.96 and 1.96. So, the stopping rule is to compute the value $2 \times 1.96\, S_n / \sqrt{n}$ at the end of each simulation, and stop when this value is smaller than 20.

5. Antithetic variables

Question. *Explain one approach you might use to exploit antithetic variables to reduce the variance in your simulation outputs.*

Formulas: Antithetic variables (Section S.8.1).

Record the uniform random variables u_i used for random number generation in the first simulation run and use their complements $(1 - u_i)$ in the second run, and so on for every pair of runs.

6. Inaccurate simulation results

Question. *To study the impact of datacenter network capacity on the query latency of a search engine, a researcher builds a simulator using custom code and their own random number generator. However, they notice that the results of this single long simulation run are far from accurate. What might be some reasons for this?*

This could be due to many factors including:

- The simulator may be buggy. It is best to validate the simulator by simulating an analytically tractable system, such as an M/M/1 queue and ensuring that the simulator outputs match theoretical predictions.
- The random number generator may be incorrect. It is best to use a standard library than to use your own.
- The output from a single simulation may diverge far from the mean. It is best to rely on the mean of many simulations than on a single run.
- The simulation results depend critically on the accuracy both of the system parameters (e.g., network capacity, buffer sizes) and the parameters of the arrival and service processes (e.g., job size, service time distribution, arrival time distribution). If these parameters are incorrectly set, the results will be incorrect.
- The simulation must correctly model whether the queries constitute an open or closed system. If the queries come from a small population, then the system is closed, and the arrival process should reflect this.

- The simulator may not model certain important aspects of the system. For example, processors may be modeled as constant-rate servers, but due to dynamic voltage scaling, they may be variable-rate. This would change the outputs of the simulation.

- The simulation may not have been run long enough, so the system might still be in its initialization or non-equilibrium state, when the outputs are unreliable.

It should be evident that to accurately model a complex system in a simulation is a complex task!

A.2 Queueing Theory: Chapter 6 Exercises

1. Little's theorem

Question. *Patients arriving at the Grand River Hospital Emergency Room have a mean waiting time of 3 hours. It has been found that, averaged over the period of a day, patients arrive at the rate of one every 5 minutes. (a) How many patients are awaiting treatment on average at any given point in time? (b) What should the size of the waiting room be so that it can always accommodate arrivals?*

> **Formulas:** Little's theorem (Section 6.1.2): $N = \lambda T$.

$\lambda = 1$ patient per 5 minutes = 12 per hour; $T = 3$ hours.

(a) $N = \lambda T = 12 \times 3 = 36$ patients waiting on average.

(b) No finite size can always accommodate arrivals. Little's theorem gives only the mean. Arrivals are random, so the number present exceeds any finite bound with nonzero probability. To size the room we need the distribution of the number present, and then pick a size that is exceeded only with some acceptably small probability. In practice, a small multiple of the mean, say three to four times (108 to 144 places), is a reasonable rule of thumb, though a burst such as a mass food-poisoning incident could still exceed it.

2. A stochastic process

Question. *Consider that in Example 6.3, a person is on an infinite staircase on stair 10 at time 0 and potentially moves once every clock tick. Suppose that the person moves from stair i to stair $i + 1$ with probability 0.2, and from stair i to stair $i - 1$ with probability 0.2 (the probability of staying on stair i is 0.6). Compute the probability that the person is on each stair at time 1 (after the first move), time 2, and time 3.*

> **Formulas:** Chapman–Kolmogorov (EQ 6.2): $p_{ij}^{(m)} = \sum_k p_{ik}^{(m-1)} p_{kj}$.

Start on stair 10. Each tick: up 0.2, down 0.2, stay 0.6.

Stair	7	8	9	10	11	12	13
$t = 1$			0.2	0.6	0.2		
$t = 2$		0.04	0.24	0.44	0.24	0.04	
$t = 3$	0.008	0.072	0.24	0.36	0.24	0.072	0.008

Each entry sums over the stairs one step earlier: for example, $P(X_2 = 10) = 0.2(0.2) + 0.6(0.6) + 0.2(0.2) = 0.44$. Each row sums to 1.

3. Discrete and continuous state and time

Question. *Come up with your own examples for all four combinations of discrete state / discrete time / continuous state / continuous time processes.*

Any sensible examples work. For instance:

- Discrete state, discrete time: packets in a router buffer sampled at each slot boundary; a board-game position after each turn.
- Discrete state, continuous time: packets in a router buffer over time; number of active calls on a trunk.
- Continuous state, discrete time: daily maximum temperature; closing price of a stock each day.
- Continuous state, continuous time: voltage on a wire; water level in a reservoir; unfinished work in a queue.

4. Markov process

Question. *Is the process in Exercise 2 a Markov process? Why or why not?*

Yes. The probability of the next stair depends only on the current stair, not on how the person got there.

5. Homogeneity

Question. *Is the process in Exercise 2 homogeneous? Why or why not?*

Yes. The transition probabilities (0.2, 0.6, 0.2) do not depend on the time step.

6. Representation

Question. *(a) Represent the process in Exercise 2 by using a transition matrix and a state transition diagram. (b) Do the rows in this matrix have to sum to 1? Do the columns in this matrix have to sum to 1? Why or why not? (c) Now, assume that the staircase has only four steps. Make appropriate assumptions (what are these?) to represent this finite process as a transition matrix and a state-transition diagram.*

(a) An infinite tridiagonal matrix with $P_{i,i} = 0.6$, $P_{i,i+1} = P_{i,i-1} = 0.2$, all other entries 0:

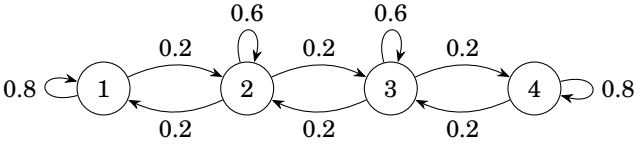
$$P = \begin{pmatrix} \ddots & \ddots & \ddots & & & & \\ & 0.2 & 0.6 & 0.2 & & & \\ & & 0.2 & 0.6 & 0.2 & & \\ & & & 0.2 & 0.6 & 0.2 & \\ & & & & \ddots & \ddots & \ddots \end{pmatrix}$$

The state-transition diagram is an infinite line of states, each with a self-loop of 0.6 and arcs of 0.2 to each neighbour.

(b) Rows must sum to 1, because from any state the process must go somewhere. Columns need not sum to 1 in general. For example, take a star with state 0 at the centre and N outer states: state 0 moves to each outer state with probability $1/N$ and every outer state returns to 0 with probability 1, so column 0 sums to N. In this chain the columns do happen to sum to 1, but that is a property of this chain, not of transition matrices.

(c) With four stairs, states 1–4, we must decide what happens at the ends. We assume reflecting boundaries: a move that would leave the staircase is replaced by staying put.

$$P = \begin{pmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0.2 & 0.8 \end{pmatrix}$$



Other assumptions, such as absorbing ends, are also valid, but they give a reducible chain, and the answers to Exercises 7 to 12 would change.

7. Reducibility

Question. *Is the chain in Exercise 2 reducible? Why or why not?*

Irreducible. Every stair can be reached from every other stair with positive probability in a finite number of steps.

8. Recurrence

Question. *Is state 1 in the chain in Exercise 6(c) recurrent? Compute f_1^1, f_1^2, and f_1^3.*

Formulas: Section 6.2.5: f_j^n = probability that the first return to state j is after n steps; j is recurrent if $\sum_{n=1}^{\infty} f_j^n = 1$.

State 1 is recurrent: the chain is finite and irreducible, so every state is recurrent. f_1^1, f_1^2 and f_1^3 are the probabilities that the first return to state 1 is after 1, 2 and 3 steps:

$$f_1^1 = 0.8, \qquad f_1^2 = 0.2 \times 0.2 = 0.04\ (1 \to 2 \to 1), \qquad f_1^3 = 0.2 \times 0.6 \times 0.2 = 0.024\ (1 \to 2 \to 2 \to 1).$$

A path through state 3 needs at least four steps to return.

9. Periodicity

Question. *Is the chain in Exercise 2 periodic? If not, give an example of a chain with period N for arbitrary $N > 1$.*

> **Formulas:** Section 6.2.6: state j has period $r \geq 2$ if returns can take only $r, 2r, 3r, \ldots$ steps; a self-loop ($p_{jj} > 0$) makes a state aperiodic, and in an irreducible chain one self-loop makes all states aperiodic.

Aperiodic. Every state has a self-loop, so a return is possible in one step and $r = 1$. A chain with period N: N states in a ring, with state i moving to state $(i + 1) \bmod N$ with probability 1. Returns happen only at multiples of N.

10. Ergodicity

Question. *Is any state in the chain of Exercise 6(c) nonergodic? Why or why not?*

No state is nonergodic. The chain is finite, irreducible and aperiodic, so all states are recurrent non-null and aperiodic, which is ergodic.

11. Stationary probability

Question. *Compute the stationary probability distribution of the chain in Exercise 6(c).*

> **Formulas:** Theorem 6.2 (EQ 6.5): $\pi_j^* = \sum_i \pi_i^* p_{ij}$, $\sum_j \pi_j^* = 1$.

The chain is ergodic, so by Theorem 6.2:

$$\pi_1^* = 0.8\pi_1^* + 0.2\pi_2^*, \qquad \pi_2^* = 0.2\pi_1^* + 0.6\pi_2^* + 0.2\pi_3^*,$$
$$\pi_3^* = 0.2\pi_2^* + 0.6\pi_3^* + 0.2\pi_4^*, \qquad \pi_4^* = 0.2\pi_3^* + 0.8\pi_4^*.$$

The first equation gives $\pi_2^* = \pi_1^*$, the second then gives $\pi_3^* = \pi_2^*$, and the last gives $\pi_4^* = \pi_3^*$. With $\sum_j \pi_j^* = 1$: $\pi^* = (0.25,\ 0.25,\ 0.25,\ 0.25)$.

12. Residence times

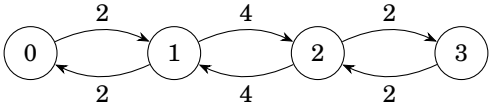
Question. *Compute the residence time in each state of the Markov chain in Exercise 6(c).*

> **Formulas:** Section 6.2.11: the residence time in state j is geometric, with mean $1/(1 - p_{jj})$.

The mean residence time in state j is $1/(1 - p_{jj})$. States 1 and 4: $1/(1 - 0.8) = 5$ ticks. States 2 and 3: $1/(1 - 0.6) = 2.5$ ticks.

13. Stationary probability of a birth–death process

Question. *Consider the state-rate-transition diagram below. (a) Compare this with the state-transition probability diagram in Exercise 6(c). What features are the same, and what differ? (b) Write down the Q matrix for this system. (c) Use the Q matrix to compute the stationary-probability distribution of this chain.*



Formulas: $q_{ii} = -\sum_{j \neq i} q_{ij}$ (EQ 6.10); $PQ = 0$ (EQ 6.18) with $P = [\pi_0^* \ \pi_1^* \ \ldots]$ and $\sum_j \pi_j^* = 1$ (EQ 6.17).

(a) Same: four states, transitions only between neighbours, symmetric structure. Different: labels are rates, not probabilities, so they need not sum to 1; there are no self-loops (time in a state is exponential, not geometric); the process is in continuous time; and the middle transitions are twice as fast as the end ones, whereas in 6(c) all neighbour transitions had equal probability.

(b)

$$Q = \begin{pmatrix} -2 & 2 & 0 & 0 \\ 2 & -6 & 4 & 0 \\ 0 & 4 & -6 & 2 \\ 0 & 0 & 2 & -2 \end{pmatrix}$$

Each row sums to 0; diagonal entries are minus the total outflow rate.

(c) $PQ = 0$ gives

$$-2\pi_0^* + 2\pi_1^* = 0, \qquad 2\pi_0^* - 6\pi_1^* + 4\pi_2^* = 0,$$
$$4\pi_1^* - 6\pi_2^* + 2\pi_3^* = 0, \qquad 2\pi_2^* - 2\pi_3^* = 0.$$

The first gives $\pi_1^* = \pi_0^*$, the second then gives $\pi_2^* = \pi_1^*$, and the last gives $\pi_3^* = \pi_2^*$. With $\sum_j \pi_j^* = 1$: $\pi^* = (0.25,\ 0.25,\ 0.25,\ 0.25)$.

14. Interdeparture time of a pure-death process

Question. *Prove that the interdeparture time of a pure-death process is exponentially distributed.*

Formulas: Section 6.3.2: residence times in a continuous-time Markov chain are exponentially distributed (EQ 6.7).

A pure-death process is a birth–death process whose birth rates are all zero, so it only moves from state j to state $j-1$. The time between two departures is therefore the time the process spends in its current state, which is its residence time in that state. A pure-death process is a continuous-time Markov chain, and residence times in a continuous-time Markov chain

are exponentially distributed (Section 6.3.2). Hence the interdeparture time in state j is exponential, with rate μ_j.

15. Stationary probabilities via Eq. 6.30

Question. *Use Equation 6.30 to compute the stationary probability of the birth-death process in Exercise 13.*

Formulas: EQ 6.29: $\pi_j^* = \pi_0^* \prod_{i=0}^{j-1} \frac{\lambda_i}{\mu_{i+1}}$; EQ 6.30: $\pi_0^* = \left(1 + \sum_{j\geq 1} \prod_{i=0}^{j-1} \frac{\lambda_i}{\mu_{i+1}}\right)^{-1}$.

$\lambda_0/\mu_1 = 2/2 = 1$, $\lambda_1/\mu_2 = 4/4 = 1$, $\lambda_2/\mu_3 = 2/2 = 1$. So $\pi_j^* = \pi_0^*$ for all j, and normalising gives $\pi_0^* = \pi_1^* = \pi_2^* = \pi_3^* = 1/4$, matching Exercise 13(c).

16. Is Exercise 13 M/M/1?

Question. *Is the birth-death process in Exercise 13 $M/M/1$? Why or why not?*

No. M/M/1 needs a constant arrival rate λ and a constant service rate μ in every state, and an infinite state space. Here the rates depend on the state (2, 4, 2) and there are only four states.

17. M/M/1 queue

Question. *Consider a link to which packets arrive as a Poisson process at a rate of 450 packets/sec such that the time taken to service a packet is exponentially distributed. Suppose that the mean packet length is 250 bytes and that the link capacity is 1 Mbps. (a) What is the probability that the link's queue has one, two, and ten packets respectively? (b) What is the mean number of packets in the system? What is the mean number in the queue? (c) What is the mean waiting time?*

Formulas: $\rho = \lambda/\mu$; $\pi_j^* = \rho^j(1-\rho)$ (EQ 6.35); $N = \frac{\rho}{1-\rho}$ (EQ 6.37); mean time in the system $= \frac{1/\mu}{1-\rho}$ (EQ 6.38; see correction 6.30).

$\lambda = 450$ packets/s. Mean service time $= 250 \times 8$ bits$/10^6$ bits/s $= 2$ ms, so $\mu = 500$ packets/s and $\rho = 0.9$. "The link's queue has n packets" means n packets waiting, with one more in service, so the system is in state $n+1$.

(a) One packet queued: $\pi_2 = 0.1 \times 0.9^2 = 0.081$; two: $\pi_3 = 0.1 \times 0.9^3 = 0.0729$; ten: $\pi_{11} = 0.1 \times 0.9^{11} \approx 0.031$.

(b) With $\pi_j^* = \rho^j(1-\rho)$, the mean number in the system is

$$N = \sum_{j=0}^{\infty} j\,\pi_j^* = (1-\rho)\sum_{j=0}^{\infty} j\rho^j = (1-\rho)\,\frac{\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho},$$

using $\sum_{j\ge 0} j\rho^j = \rho \frac{\mathrm{d}}{\mathrm{d}\rho} \sum_{j\ge 0} \rho^j = \rho \frac{\mathrm{d}}{\mathrm{d}\rho} \frac{1}{1-\rho} = \frac{\rho}{(1-\rho)^2}$.

When there are $j \ge 1$ customers in the system, one is in service and $j-1$ are in the queue; when $j = 0$ the queue is empty. So the mean number in the queue is

$$N_q = \sum_{j=1}^{\infty} (j-1)\pi_j^* = \sum_{j=1}^{\infty} j\,\pi_j^* - \sum_{j=1}^{\infty} \pi_j^* = N - (1-\pi_0^*) = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho},$$

since $\pi_0^* = 1-\rho$ (EQ 6.34), so $1-\pi_0^* = \rho$ is the probability that the server is busy. Equivalently, $N_q = N - \rho$: the mean number in service is ρ.

With $\rho = 0.9$: $N = 0.9/0.1 = 9$ and $N_q = 0.81/0.1 = 8.1$.

(c) By EQ 6.38, the mean time in the system is $(1/500)/(1-0.9) = 0.002/0.1 = 0.02$ s $= 20$ ms. This includes the service time. The mean waiting time in the queue alone is $20 - 2 = 18$ ms (equivalently, $N_q/\lambda = 8.1/450$ s). Check with Little's theorem: $N = \lambda T = 450 \times 0.02 = 9$.

18. Responsive (M/M/∞) server

Question. *Compute the ratio of π_j^* for a responsive server to the same value for an $M/M/1$ queue. How does this ratio behave as a function of j?*

Formulas: M/M/∞ (EQ 6.42): $\pi_j^* = e^{-\rho}\rho^j/j!$ with $\rho = \lambda/\mu$; M/M/1 (EQ 6.35): $\pi_j^* = \rho^j(1-\rho)$.

$$\frac{\pi_j^{\mathrm{M/M/\infty}}}{\pi_j^{\mathrm{M/M/1}}} = \frac{e^{-\rho}\rho^j/j!}{(1-\rho)\rho^j} = \frac{e^{-\rho}}{(1-\rho)\,j!}, \qquad \rho < 1.$$

The ρ^j terms cancel. The ratio exceeds 1 at $j = 0$ (since $e^{-\rho} > 1-\rho$) and then falls as $1/j!$, faster than geometrically. So the responsive server is more likely to be nearly empty and far less likely to hold many customers.

19. M/M/1/K with ten buffers

Question. *Assume that the queueing system in Exercise 17 has ten buffers. Compute an upper bound on the probability of packet loss.*

Note: this question has an error (see correction 6.36). There should not be an upper bound. Instead, compute the exact probability of packet loss.

Formulas: M/M/1/K with $K-1$ buffers (EQ 6.47): $\pi_j^* = \frac{1-\rho}{1-\rho^{K+1}}\rho^j$, $0 \le j \le K$; π_K^* is the blocking probability (Section 6.6.2).

With $K-1 = 10$ buffers, $K = 11$. A packet is lost when the system is full, so the loss probability is the blocking probability

$$\pi_{11} = \frac{1-\rho}{1-\rho^{K+1}}\rho^{11} = \frac{0.1}{1-0.9^{12}}\,0.9^{11} \approx 0.0437.$$

20. M/D/1 queue

Question. *Compute the mean number of customers in an $M/D/1$ system that has a utilization of 0.9. (a) How does this compare with a similarly loaded $M/M/1$ system? (b) Compute the ratio of the mean number of customers as a function of ρ. (c) Use this to compare the behavior of an $M/D/1$ queue with that of an $M/M/1$ queue under heavy load.*

Formulas: M/D/1 (EQ 6.49): mean customers in the system $= \rho + \dfrac{\rho^2}{2(1-\rho)}$; M/M/1 (EQ 6.37): $N = \dfrac{\rho}{1-\rho}$.

By EQ 6.49, at $\rho = 0.9$ the mean number in the M/D/1 system is $N = 0.9 + 0.81/0.2 = 4.95$.

(a) By EQ 6.37, M/M/1 at $\rho = 0.9$ has $N = 9$, so M/D/1 has 55% as many.

(b)

$$\frac{N_{\mathrm{M/D/1}}}{N_{\mathrm{M/M/1}}} = \frac{\rho + \rho^2/(2(1-\rho))}{\rho/(1-\rho)} = (1-\rho) + \frac{\rho}{2} = 1 - \frac{\rho}{2}.$$

(c) As $\rho \to 1$ the ratio $\to 1/2$. Both queues blow up as $1/(1-\rho)$, but under heavy load the deterministic server holds about half as many customers. Removing service-time variability halves the queue but does not remove the blow-up.

A.3 Queueing Theory: Additional M/G/1 Problems

Problem 1. M/D/1 versus M/M/1

Question. *Prove that the mean number of jobs waiting or in service for a $M/D/1$ queue with mean arrival rate λ and mean service rate μ is always lower than in an $M/M/1$ queue with the same parameters.*

Formulas: Handout Eq. 2 (Pollaczek–Khinchine): $E(N) = \rho + \dfrac{\rho^2(1+C_S^2)}{2(1-\rho)}$.

Deterministic service times have no variability, so M/D/1 has $C_S^2 = 0$; exponential service times give $C_S^2 = 1$ for M/M/1. Substituting into Eq. 2,

$$E(N)_{\mathrm{M/M/1}} - E(N)_{\mathrm{M/D/1}} = \frac{\rho^2}{2(1-\rho)} > 0 \qquad \text{for all } 0 < \rho < 1.$$

Hence the M/D/1 queue always has fewer jobs in the system. Equivalently, the ratio is $1 - \rho/2 < 1$ (Exercise 20).

Problem 2. Time in the system: M/G/1 versus M/M/1

Question. *Suppose that the service time distribution of a queue whose utilization is 80% is described by $C_S^2 = 3$. Compare the mean time in the system for this queue with that of a $M/M/1$ queue with the same mean arrival and departure rates.*

Formulas: Handout Eq. 1: $E(T) = \frac{1}{\mu}\left(1 + \frac{\rho(1 + C_S^2)}{2(1-\rho)}\right)$, with $C_S^2 = 1$ for M/M/1.

Here $\rho = 0.8$ and $C_S^2 = 3$. From Eq. 1:

$$C_S^2 = 3: \; E(T) = \frac{1}{\mu}\left(1 + \frac{0.8 \times 4}{2 \times 0.2}\right) = \frac{9}{\mu}, \qquad \text{M/M/1 } (C_S^2 = 1): \; E(T) = \frac{1}{\mu}\left(1 + \frac{0.8 \times 2}{2 \times 0.2}\right) = \frac{5}{\mu}.$$

So the mean time in the system is $9/5 = 1.8$ times that of the M/M/1 queue, that is, 80% longer.

Problem 3. MacroHard Doors

Question. *By instrumenting the time when a job starts service in the MacroHard Doors operating system, you discover that, for the first 1000 submitted jobs, the sum of service times is 412 seconds, and the sum of the squared service times is 1425 seconds^2. Assuming that jobs arrive to the system as a Poisson process and the mean utilization is 0.75, what is the mean number of jobs in the system, including the one in service?*

Formulas: Handout Section 2: $C_S^2 = \frac{E(S^2)}{(E(S))^2} - 1$; handout Eq. 2: $E(N) = \rho + \frac{\rho^2(1 + C_S^2)}{2(1-\rho)}$.

$E(S) = 412/1000 = 0.412$ s and $E(S^2) = 1425/1000 = 1.425$ s^2. So

$$C_S^2 = \frac{1.425}{0.412^2} - 1 = \frac{1.425}{0.1697} - 1 \approx 8.39 - 1 = 7.39.$$

With $\rho = 0.75$, Eq. 2 gives

$$E(N) = 0.75 + \frac{0.75^2 \times (1 + 7.39)}{2 \times 0.25} = 0.75 + \frac{0.5625 \times 8.39}{0.5} \approx 0.75 + 9.44 \approx 10.2 \text{ jobs.}$$

Note that the service times are highly variable ($C_S^2 \approx 7.4$), so the queue is much longer than an M/M/1 queue at the same load, which would have $E(N) = 0.75/0.25 = 3$.

A.4 Signals, Systems, and Transforms: Chapter 5 Exercises

1. Complex arithmetic

Question. *Compute $e^{-j\pi/2} + e^{j\pi/2}$.*

Formulas: Euler's formula (EQ 5.3): $e^{j\theta} = \cos\theta + j\sin\theta$; hence (EQ 5.5) $\cos\theta = \dfrac{e^{j\theta} + e^{-j\theta}}{2}$.

By Euler's formula,

$$e^{-j\pi/2} + e^{j\pi/2} = \left(\cos\tfrac{\pi}{2} - j\sin\tfrac{\pi}{2}\right) + \left(\cos\tfrac{\pi}{2} + j\sin\tfrac{\pi}{2}\right) = 2\cos\tfrac{\pi}{2} = 0.$$

Geometrically, $e^{j\pi/2} = j$ and $e^{-j\pi/2} = -j$ are two unit phasors pointing in opposite directions along the imaginary axis, so they cancel.

2. Phase angle

Question. *What is the phase angle corresponding to the complex number* $1 + j$*?*

Formulas: EQ 5.4: $a + jb = c\,e^{j\theta}$ with $c = \sqrt{a^2 + b^2}$ and $\theta = \operatorname{atan}(b/a)$.

Here $a = b = 1$, so $\theta = \operatorname{atan}(1/1) = \pi/4$ radians (45°). The magnitude is $\sqrt{2}$, so $1 + j = \sqrt{2}\,e^{j\pi/4}$.

3. Discrete convolution

Question. *Let* $z(t)$ *be the convolution of* $x(t)$*,* $0 \le t \le 9$*,* $= 1, 3, 5, 2, 5, 8, 7, 3, 9, 4$ *and* $y(t)$*,* $0 \le t \le 9$*,* $= 3, 1, 7, 4, 5, 9, 7, 1, 3, 8$*. Compute* $z(5)$*.*

Formulas: Discrete convolution (EQ 5.11): $z(t) = \displaystyle\sum_{\tau=-\infty}^{\infty} x(\tau)\,y(t-\tau)$.

Both signals are zero outside $0 \le t \le 9$, so only $0 \le \tau \le 5$ contributes to $z(5)$. We need the products $x(\tau)\,y(5-\tau)$, that is, all pairs whose indices sum to 5:

τ	0	1	2	3	4	5
$x(\tau)$	1	3	5	2	5	8
$y(5-\tau)$	9	5	4	7	1	3
product	9	15	20	14	5	24

Summing the products, $z(5) = 9 + 15 + 20 + 14 + 5 + 24 = 87$.

A useful way to present this on the board is to write y reversed under x and slide it until index 0 of x lines up with index 5 of y.

4. Signals

Question. *Give an example of a signal that is continuous, digital, aperiodic, and time-unlimited.*

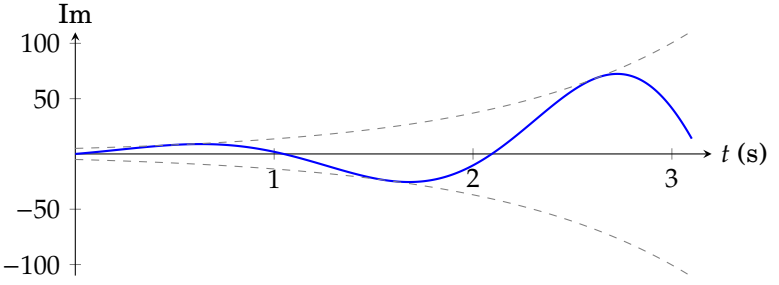
The readings of a digital thermometer. The temperature is defined at every instant, so the signal is continuous in time. The display shows only a finite set of values, so it is digital. Temperature does not repeat exactly, so it is aperiodic. The thermometer has no predefined end time, so the signal is time-unlimited.

5. Complex exponential

Question. *Describe the Im–t projection of the curve* $5e^{t}(\cos 3t + j \sin 3t)$.

> **Formulas:** Euler's formula (EQ 5.3); a complex exponential $e^{(\sigma+j\omega)t}$ grows if $\sigma > 0$, decays if $\sigma < 0$, and has angular frequency ω (Section 5.2.1).

The curve is $5e^{(1+3j)t}$. Its projection on the Im–t plane is obtained by setting the real part to zero, which leaves $j\,5e^{t}\sin 3t$. This is a sinusoid with angular frequency 3 radians/second (period $2\pi/3 \approx 2.09$ s, frequency $3/2\pi \approx 0.48$ Hz) whose amplitude $5e^{t}$ grows exponentially with time. The amplitude is 5 at $t = 0$ and grows by a factor of e every second.



The dashed lines are the envelope $\pm 5e^{t}$.

6. Linearity

Question. *Is the system defined by the transfer function* $H(x) = 5\dfrac{\mathrm{d}x}{\mathrm{d}t} + 1$ *linear?*

> **Formulas:** Linearity (Section 5.3): H is linear if, for all inputs x_1, x_2 and constants k_1, k_2, $H(k_1x_1 + k_2x_2) = k_1H(x_1) + k_2H(x_2)$.

The left-hand side is

$$H(k_1x_1 + k_2x_2) = 5k_1\frac{\mathrm{d}x_1}{\mathrm{d}t} + 5k_2\frac{\mathrm{d}x_2}{\mathrm{d}t} + 1,$$

whereas the right-hand side is

$$k_1H(x_1) + k_2H(x_2) = 5k_1\frac{\mathrm{d}x_1}{\mathrm{d}t} + 5k_2\frac{\mathrm{d}x_2}{\mathrm{d}t} + k_1 + k_2.$$

These differ whenever $k_1 + k_2 \neq 1$, so the system is not linear. The constant term is the culprit: a linear system must produce zero output for zero input, but here $H(0) = 1$. (A system of this form is sometimes called affine.)

7. LTI system

Question. *What is the output of an LTI system when the input is a real sinusoid?*

Formulas: EQ 5.5: $\cos \omega t = \frac{1}{2}\left(e^{j\omega t} + e^{-j\omega t}\right)$; the response of an LTI system to ke^{st} is $f(s)\,e^{st}$ (EQ 5.21), a complex exponential with the same s.

A real sinusoid is the sum of two counter-rotating complex exponentials with equal and opposite frequencies. The response of an LTI system to a complex exponential is the same complex exponential, scaled by a complex constant. By linearity, the output is the sum of the two scaled exponentials. For a real system these constants are complex conjugates, so the imaginary parts cancel and the output is a real sinusoid at the same frequency as the input, in general scaled in amplitude and shifted in phase. An LTI system never creates new frequencies.

8. Natural response

Question. *Compute the natural response of the LTI system given by* $2\dfrac{\mathrm{d}^2y(t)}{\mathrm{d}t^2} + 11\dfrac{\mathrm{d}y(t)}{\mathrm{d}t} + 15y(t) = 32x(t)$.

Formulas: Natural response (EQ 5.23): if the characteristic polynomial has distinct roots λ_i, then $y(t) = \sum_i c_i e^{\lambda_i t}$.

Set the input to zero and write D for $\mathrm{d}/\mathrm{d}t$. The natural response satisfies $(2D^2 + 11D + 15)\,y(t) = 0$, which factors as

$$2(D + 3)(D + 2.5)\,y(t) = 0.$$

The roots are -3 and -2.5, so

$$y(t) = c_1 e^{-3t} + c_2 e^{-2.5t},$$

where c_1 and c_2 are determined by the initial conditions $y(0)$ and $\dot{y}(0)$. Both roots are real, so neither term oscillates, and both are negative, so the response decays to zero. This parallels Example 5.7.

9. Natural response

Question. *Determine the exact natural response of the system described by* $2\dfrac{\mathrm{d}^2y(t)}{\mathrm{d}t^2} + y(t) = 32x(t)$, $y(0) = 0$, $\dot{y}(0) = 1$. *What is its frequency?*

Formulas: Natural response (EQ 5.23); Euler's formula (EQ 5.6): $\sin\theta = \dfrac{e^{j\theta} - e^{-j\theta}}{2j}$.

The natural response satisfies $(2D^2 + 1)\,y(t) = 0$, which factors as

$$2\left(D - \tfrac{j}{\sqrt{2}}\right)\left(D + \tfrac{j}{\sqrt{2}}\right)y(t) = 0,$$

so

$$y(t) = c_1 e^{-jt/\sqrt{2}} + c_2 e^{jt/\sqrt{2}}.$$

Now apply the initial conditions. Setting $y(0) = 0$ gives $c_1 + c_2 = 0$. Differentiating and setting $\dot{y}(0) = 1$ gives

$$-\frac{j}{\sqrt{2}}c_1 + \frac{j}{\sqrt{2}}c_2 = 1, \quad \text{that is,} \quad c_2 - c_1 = -j\sqrt{2}.$$

Solving, $c_1 = j/\sqrt{2}$ and $c_2 = -j/\sqrt{2}$. The two constants are complex conjugates, as Example 5.8 says they must be for a real response. Therefore

$$y(t) = \frac{j}{\sqrt{2}}\left(e^{-jt/\sqrt{2}} - e^{jt/\sqrt{2}}\right) = \frac{j}{\sqrt{2}}\left(-2j\sin\frac{t}{\sqrt{2}}\right) = \sqrt{2}\,\sin\frac{t}{\sqrt{2}}.$$

Check: $y(0) = 0$ and $\dot{y}(0) = \sqrt{2}\cdot\frac{1}{\sqrt{2}} = 1$.

The angular frequency is $1/\sqrt{2} \approx 0.707$ radians/second, which is a frequency of $1/(2\sqrt{2}\,\pi) \approx 0.113$ Hz.

10. Stability

Question. *Characterize the stability of the system in Exercise 9.*

> **Formulas:** Section 5.4.4: write each root as $\sigma + j\omega$. If all $\sigma < 0$ the system is asymptotically stable; if any $\sigma > 0$ it is unstable; if all $\sigma = 0$ and there are no repeated roots it is purely oscillatory (marginally stable).

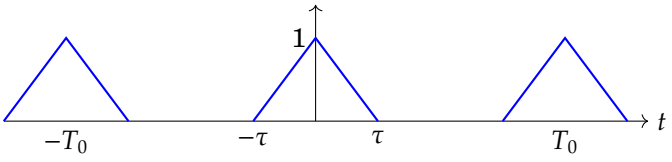
The roots are $\pm j/\sqrt{2}$. Both have $\sigma = 0$ and they are distinct, so the system is oscillatory, or marginally stable. Its natural response, from Exercise 9, is a sinusoid whose amplitude neither grows nor decays.

Note that the system is not asymptotically stable: if it is disturbed it never returns to rest. Moreover, an input at the natural frequency $1/\sqrt{2}$ radians/second makes the system resonate (Section 5.4.2), so a bounded input can produce an output that grows without bound.

11. Fourier series

Question. *Find the Fourier series corresponding to the infinite series of isosceles triangular pulses of base width 2τ and height 1 spaced apart T_0 seconds. Use the fact that* $\displaystyle\int xe^{ax}\,\mathrm{d}x = \frac{axe^{ax} - e^{ax}}{a^2} + C.$

> **Formulas:** Fourier series (EQ 5.32, 5.33): $x(t) = \displaystyle\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$, $c_k = \frac{1}{T_0}\displaystyle\int_{-T_0/2}^{T_0/2} x(t)\,e^{-jk\omega_0 t}\,\mathrm{d}t$, $\omega_0 = \dfrac{2\pi}{T_0}$ (EQ 5.30); EQ 5.5.



As in the rectangular-pulse example (Section 5.6), centre one pulse on the origin and integrate over $[-T_0/2, T_0/2]$. In this range the function is

$$x(t) = 1 - \frac{|t|}{\tau} \quad \text{for } |t| \le \tau, \qquad x(t) = 0 \ \text{ elsewhere.}$$

Write $a = -jk\omega_0$, so that

$$c_k = \frac{1}{T_0}\int_{-\tau}^{\tau}\left(1 - \frac{|t|}{\tau}\right)e^{at}\,\mathrm{d}t = \frac{1}{T_0}\left(\int_{-\tau}^{\tau} e^{at}\,\mathrm{d}t \;-\; \frac{1}{\tau}\int_{0}^{\tau} t\,e^{at}\,\mathrm{d}t \;+\; \frac{1}{\tau}\int_{-\tau}^{0} t\,e^{at}\,\mathrm{d}t\right).$$

The three integrals are

$$\int_{-\tau}^{\tau} e^{at}\,\mathrm{d}t = \frac{e^{a\tau} - e^{-a\tau}}{a}, \qquad \int_{0}^{\tau} t\,e^{at}\,\mathrm{d}t = \frac{(a\tau - 1)e^{a\tau} + 1}{a^2}, \qquad \int_{-\tau}^{0} t\,e^{at}\,\mathrm{d}t = \frac{(a\tau + 1)e^{-a\tau} - 1}{a^2},$$

where the last two use the given integral. Substituting, the terms in $a\tau$ cancel the first integral, and we are left with

$$c_k = \frac{e^{a\tau} + e^{-a\tau} - 2}{a^2 \tau T_0}.$$

Now substitute $a = -jk\omega_0$. By EQ 5.5, $e^{a\tau} + e^{-a\tau} = 2\cos(k\omega_0\tau)$, and $a^2 = -k^2\omega_0^2$, so

$$c_k = \frac{2\,(1 - \cos(k\omega_0\tau))}{k^2\omega_0^2\,\tau\,T_0} = \frac{\tau}{T_0}\left(\frac{\sin(k\omega_0\tau/2)}{k\omega_0\tau/2}\right)^2, \qquad k \neq 0,$$

using $1 - \cos\theta = 2\sin^2(\theta/2)$. For $k = 0$, c_0 is the average value of the signal, τ/T_0, which is also the limit of the expression above as $k \to 0$. The Fourier series is therefore

$$x(t) = \frac{\tau}{T_0}\sum_{k=-\infty}^{\infty}\left(\frac{\sin(k\omega_0\tau/2)}{k\omega_0\tau/2}\right)^2 e^{jk\omega_0 t}.$$

Three points are worth making. First, the coefficients are real, because the signal is even. Second, comparing with EQ 5.35, each coefficient is (up to a constant factor) the square of the coefficient for a rectangular pulse of width τ; this is because a triangle is the convolution of two such rectangles. Third, the coefficients fall off as $1/k^2$ rather than $1/k$, because the triangle has no jumps.

12. Fourier series

Question. *Compute the third coefficient of the Fourier series representing a periodic rectangular pulse of width 1 s spaced apart by 10 s.*

Formulas: Rectangular pulse train of height 1 and width τ (EQ 5.35): $c_k = \dfrac{\tau\omega_0}{2\pi}\,\dfrac{\sin(k\omega_0\tau/2)}{k\omega_0\tau/2}$, $\omega_0 = \dfrac{2\pi}{T_0}$ (EQ 5.30).

Here $T_0 = 10$ s, so $\omega_0 = 2\pi/10$, and $\tau = 1$ s, so $\tau\omega_0/2\pi = 1/10$. For the third coefficient, $k = 3$, so $k\omega_0 = 6\pi/10 = 0.6\pi$ and $k\omega_0\tau/2 = 0.3\pi$. Therefore

$$c_3 = \frac{1}{10}\cdot\frac{\sin(0.3\pi)}{0.3\pi} = \frac{1}{10}\cdot\frac{0.809}{0.942} \approx 0.086.$$

13. Fourier transform

Question. *Find the Fourier transform of a single left-triangular pulse defined by* $1 - t$ *in* $[0, 1]$.

Formulas: Fourier transform (EQ 5.37): $X(j\omega) = \displaystyle\int_{-\infty}^{\infty} x(t)\,e^{-j\omega t}\,\mathrm{d}t$; $\displaystyle\int xe^{ax}\,\mathrm{d}x = \frac{axe^{ax} - e^{ax}}{a^2} + C$ (from Exercise 11).

The function is nonzero only in $[0, 1]$, so

$$X(j\omega) = \int_0^1 (1-t)\,e^{-j\omega t}\,\mathrm{d}t = \int_0^1 e^{-j\omega t}\,\mathrm{d}t - \int_0^1 t\,e^{-j\omega t}\,\mathrm{d}t.$$

The first integral is $\dfrac{1 - e^{-j\omega}}{j\omega}$. For the second, use the given integral with $a = -j\omega$:

$$\int_0^1 t\,e^{-j\omega t}\,\mathrm{d}t = \left[\frac{-j\omega t\,e^{-j\omega t} - e^{-j\omega t}}{-\omega^2}\right]_0^1 = \frac{j\omega e^{-j\omega} + e^{-j\omega} - 1}{\omega^2}.$$

Subtracting and simplifying,

$$X(j\omega) = \frac{1 - j\omega - e^{-j\omega}}{\omega^2}.$$

As a check, expanding $e^{-j\omega} \approx 1 - j\omega - \omega^2/2$ for small ω gives $X(0) = 1/2$, which is the area under the pulse, as it should be.

14. Inverse Fourier transform

Question. *Compute the inverse Fourier transform of the function* $\pi\,(\delta(\omega - \omega_0) + \delta(\omega + \omega_0))$ *from first principles.*

Formulas: Inverse Fourier transform (EQ 5.40): $x(t) = \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{\infty} X(j\omega)\,e^{j\omega t}\,\mathrm{d}\omega$; sifting (EQ 5.15), as used in Example 5.12: $\displaystyle\int_{-\infty}^{\infty} \delta(\omega - \omega_0)\,e^{j\omega t}\,\mathrm{d}\omega = e^{j\omega_0 t}$; EQ 5.5.

By definition,

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \pi\left(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right)e^{j\omega t}\,\mathrm{d}\omega = \frac{1}{2}\int_{-\infty}^{\infty}\delta(\omega-\omega_0)\,e^{j\omega t}\,\mathrm{d}\omega + \frac{1}{2}\int_{-\infty}^{\infty}\delta(\omega+\omega_0)\,e^{j\omega t}\,\mathrm{d}\omega.$$

Applying the sifting property to each integral,

$$x(t) = \frac{e^{j\omega_0 t}+e^{-j\omega_0 t}}{2} = \cos\omega_0 t$$

by EQ 5.5. This confirms row 4 of Table 5.2: an eternal cosine corresponds to two impulses, at $+\omega_0$ and $-\omega_0$.

15. Computing the Fourier transform

Question. *Use the properties of the Fourier transform to compute the transform of the function* $\cos(\omega_0(t+t_0)) + \sin(\omega_0(t-t_0))$.

Formulas: Table 5.2, rows 4 and 5: $\cos\omega_0 t \leftrightarrow \pi\left(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right)$, $\sin\omega_0 t \leftrightarrow j\pi\left(\delta(\omega+\omega_0)-\delta(\omega-\omega_0)\right)$ (sign as corrected in the errata); linearity; time-shifting: $x(t-t_0) \leftrightarrow e^{-j\omega t_0}X(j\omega)$.

By linearity we can transform each term separately. The first term is a cosine shifted earlier by t_0 (that is, a shift of $-t_0$), so its transform is multiplied by $e^{j\omega t_0}$. The second is a sine delayed by t_0, so its transform is multiplied by $e^{-j\omega t_0}$. Therefore

$$X(j\omega) = \pi\left(\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\right)e^{j\omega t_0} + j\pi\left(\delta(\omega+\omega_0)-\delta(\omega-\omega_0)\right)e^{-j\omega t_0}.$$

Note that a time shift leaves the magnitudes of the impulses unchanged and alters only their phases.

16. Laplace transform

Question. *Compute the Laplace transform of the sinusoid* $u(t)\sin\omega_0 t$ *from first principles. Locate its pole(s) and zero(es).*

Formulas: Laplace transform (EQ 5.41): $X(s) = \int_{-\infty}^{\infty} x(t)\,e^{-st}\,\mathrm{d}t$; Example 5.20: $\int_0^{\infty} e^{\alpha t}e^{-st}\,\mathrm{d}t = \frac{1}{s-\alpha}$ for $\mathrm{Re}(s) > \mathrm{Re}(\alpha)$; EQ 5.6.

Use EQ 5.6 to write the signal as $u(t)\dfrac{e^{j\omega_0 t}-e^{-j\omega_0 t}}{2j}$. The step function restricts the integral to $t \geq 0$, so

$$X(s) = \frac{1}{2j}\left(\int_0^{\infty} e^{j\omega_0 t}e^{-st}\,\mathrm{d}t - \int_0^{\infty} e^{-j\omega_0 t}e^{-st}\,\mathrm{d}t\right) = \frac{1}{2j}\left(\frac{1}{s-j\omega_0} - \frac{1}{s+j\omega_0}\right) = \frac{\omega_0}{s^2+\omega_0^2},$$

with region of convergence $\mathrm{Re}(s) > 0$, since both integrals need $\mathrm{Re}(s) > \mathrm{Re}(\pm j\omega_0) = 0$.

The poles are the roots of the denominator, $s = \pm j\omega_0$, which lie on the imaginary axis. The numerator is the constant ω_0, so for $\omega_0 \neq 0$ the transform has no (finite) zeros. Contrast this with the cosine in Example 5.21, whose numerator s gives a zero at $s = 0$.

17. Laplace transform

Question. *Find the Laplace transform of the signal* $u(t - t_0)\sin\omega_0(t - t_0)$.

Formulas: Table 5.3, time-shifting: $x(t - t_0) \leftrightarrow e^{-st_0}X(s)$; result of Exercise 16.

The signal is the switched-on sinusoid of Exercise 16, $x(t) = u(t)\sin\omega_0 t$, delayed by t_0. Both the step and the sinusoid are delayed, so the time-shifting property applies directly:

$$X(s) = e^{-st_0}\,\frac{\omega_0}{s^2 + \omega_0^2}, \qquad \mathrm{Re}(s) > 0.$$

The delay leaves the poles at $\pm j\omega_0$ and the region of convergence unchanged. It only multiplies the transform by e^{-st_0}.

Note that the property needs the step to be delayed as well. The signal $u(t)\sin\omega_0(t - t_0)$, which switches on at $t = 0$ with a phase offset, has a different transform. To find it, expand the sine and use rows 7 and 8 of Table 5.4.

18. Using the Laplace transform to solve a system

Question. *Consider a system whose impulse response* $h(t)$ *is given by* $u(t)\cos\omega_0 t$. *What is its response to the input* $e^{-t}u(t)$*?*

Formulas: Convolution property (Table 5.3; EQ 5.28): $Y(s) = H(s)X(s)$; Table 5.4, row 5: $e^{\alpha t}u(t) \leftrightarrow \dfrac{1}{s - \alpha}$; row 8: $\cos(\omega t)u(t) \leftrightarrow \dfrac{s}{s^2 + \omega^2}$; partial fractions, distinct roots (Section 8.11.1): $a_i = \lim_{s\to\alpha_i}(s - \alpha_i)f(s)$.

From Table 5.4, $H(s) = \dfrac{s}{s^2 + \omega_0^2}$ with $\mathrm{Re}(s) > 0$, and $X(s) = \dfrac{1}{s + 1}$ with $\mathrm{Re}(s) > -1$. The output transform is their product,

$$Y(s) = \frac{s}{(s + 1)(s + j\omega_0)(s - j\omega_0)}, \qquad \mathrm{Re}(s) > 0.$$

The three roots are distinct, so

$$Y(s) = \frac{a_1}{s + 1} + \frac{a_2}{s + j\omega_0} + \frac{a_3}{s - j\omega_0},$$

with

$$a_1 = \frac{-1}{(-1)^2 + \omega_0^2} = \frac{-1}{1 + \omega_0^2},$$

$$a_2 = \frac{-j\omega_0}{(1 - j\omega_0)(-2j\omega_0)} = \frac{1}{2(1 - j\omega_0)}, \qquad a_3 = \frac{j\omega_0}{(1 + j\omega_0)(2j\omega_0)} = \frac{1}{2(1 + j\omega_0)}.$$

As Section 8.11.2 states, a_2 and a_3 are complex conjugates. Inverting term by term with row 5 of Table 5.4,

$$y(t) = \left(\frac{-e^{-t}}{1+\omega_0^2} + \frac{e^{-j\omega_0 t}}{2(1-j\omega_0)} + \frac{e^{j\omega_0 t}}{2(1+j\omega_0)} \right) u(t).$$

The last two terms are conjugates, so their sum is twice the real part of either. Multiplying the numerator and denominator of the last term by $1 - j\omega_0$ and using Euler's formula gives the real form

$$y(t) = \frac{\cos\omega_0 t + \omega_0 \sin\omega_0 t - e^{-t}}{1+\omega_0^2}\, u(t).$$

As a check, $y(0) = 0$, as it must be for the convolution of two signals that start at $t = 0$. The e^{-t} term is a transient. In steady state the output is a sinusoid at the frequency ω_0 of the system's natural response.

A.5 Control Theory: Chapter 8 Exercises

1. A bandwidth management system

Question. *Consider the following problem: An ISP wants to ensure that, on a specific congested link, the fraction of P2P traffic is never more than r percent. If this fraction is exceeded, the ISP sends TCP reset packets to P2P connection endpoints, terminating them, and thus reducing their load. For this system, identify the plant, the command, the control input, the disturbance, and the output.*

Formulas: Section 8.1: the plant is the system being controlled; the command (reference) r is the desired value of the output; the control input u is chosen by the controller; the disturbance w is an uncontrollable input that additively affects the output y.

Plant	The congested link
Command	The desired maximum percentage of P2P traffic on the link, r
Control input	The number of P2P connections reset at a given time
Disturbance	The intrinsic fluctuations in the number of P2P connections
Output	The percentage of P2P traffic on the link

A useful exercise is to map these onto the blocks of Figure 8.1, and to ask what plays the role of the measurement process (here, a traffic classifier that estimates the P2P fraction, which may itself be in error).

2. Effort and flow

Question. *What are the effort and flow variables in the system of Exercise 1?*

Formulas: Section 8.2.1: effort is a system input that causes a flow; in a computer network, the quantity of data to be sent is the effort and the rate at which it is sent is the flow.

The effort is the number of connections that need to be reset at a point in time. The flow is the number of connections actually reset per unit time.

3. Transfer function

Question. *What is the transfer function for the system of Exercise 1 in the regime when the congested link has more than r percent of P2P connections?*

Formulas: Transfer function (EQ 8.6): $Y(s) = U(s)G(s)$ for a system at equilibrium at $t = 0$; Table 5.3, differentiation: $\dot{y} \leftrightarrow sY(s)$ when $y(0) = 0$.

In this regime, the controller removes u connections over a period T, which reduces the P2P fraction at rate u/T, while the disturbance increases it at rate w. Writing y for the current fraction of P2P traffic,

$$\dot{y} = -\frac{u}{T} + w.$$

Taking the Laplace transform of both sides with $y(0) = 0$,

$$sY(s) = -\frac{U(s)}{T} + W(s).$$

Ignoring the disturbance W,

$$G(s) = \frac{Y(s)}{U(s)} = -\frac{1}{sT}.$$

This is an integrator with negative gain, like the Web server of Example 8.4 (where $G(s) = 1/s$). Intuitively, each reset removes traffic permanently, so the output accumulates the effect of the control input over time. The minus sign records that resetting connections reduces the output.

4. First-order system

Question. *When given a step input, a first-order system reaches 63% of its asymptotic value of 4.25 units at time 3 s. What is its transfer function?*

Formulas: First-order system (EQ 8.7): $G(s) = \dfrac{K}{1 + \tau s}$; unit step response (EQ 8.8): $y(t) = K\left(1 - e^{-t/\tau}\right)$, which reaches $1 - 1/e \approx 63\%$ of its final value K at $t = \tau$.

The step response asymptotes to K, so $K = 4.25$. It reaches 63% of this value at $t = \tau$, so $\tau = 3$ s. Therefore

$$G(s) = \frac{4.25}{1 + 3s}.$$

5. Second-order system

Question. *Prove that for a critically damped second-order system,* $y(t) = K\left[1 - e^{-\omega_n t} - \omega_n t\, e^{-\omega_n t}\right]$.

Formulas: Second-order system (EQ 8.9): $G(s) = \dfrac{K}{\dfrac{s^2}{\omega_n^2} + \dfrac{2\zeta s}{\omega_n} + 1}$, critically damped when $\zeta = 1$; partial fractions with repeated roots (Section 8.11.3); Table 5.4, rows 2, 5, and 6: $\dfrac{1}{(s-\alpha)^{n+1}} \leftrightarrow \dfrac{t^n}{n!} e^{\alpha t} u(t)$.

As in Section 8.4, $y(t)$ is the response to a unit step, whose transform is $1/s$. With $\zeta = 1$,

$$Y(s) = \frac{K}{s\left(\dfrac{s^2}{\omega_n^2} + \dfrac{2s}{\omega_n} + 1\right)} = \frac{K}{s\left(\dfrac{s}{\omega_n} + 1\right)^2} = \frac{K\omega_n^2}{s\,(s+\omega_n)^2}.$$

The root $-\omega_n$ is repeated, so we write

$$\frac{\omega_n^2}{s\,(s+\omega_n)^2} = \frac{a_2}{s} + \frac{a_{12}}{(s+\omega_n)^2} + \frac{a_{11}}{s+\omega_n}.$$

Using Section 8.11.3,

$$a_2 = \lim_{s\to 0} \frac{\omega_n^2}{(s+\omega_n)^2} = 1, \qquad a_{12} = \lim_{s\to -\omega_n} \frac{\omega_n^2}{s} = -\omega_n, \qquad a_{11} = \lim_{s\to -\omega_n} \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\omega_n^2}{s}\right) = \lim_{s\to -\omega_n}\left(-\frac{\omega_n^2}{s^2}\right) = -1.$$

So

$$Y(s) = K\left(\frac{1}{s} - \frac{\omega_n}{(s+\omega_n)^2} - \frac{1}{s+\omega_n}\right).$$

Inverting term by term with rows 2, 6 (with $n = 1$), and 5 of Table 5.4,

$$y(t) = K\left[1 - \omega_n t\, e^{-\omega_n t} - e^{-\omega_n t}\right],$$

as required. The output has no sinusoidal component: it rises smoothly to K without overshoot.

6. Proportional-mode control

Question. *What is the relationship between the loop gain and the system pole for the proportional-mode controller discussed in Example 8.9?*

Formulas: Example 8.9: plant $G(s) = 1/s$; control law $u = \dfrac{r-y}{\tau}$, so the loop gain is $K_p = 1/\tau$; closed loop $Y = \dfrac{R}{1+s\tau} + \dfrac{W\tau}{1+s\tau}$ (with the disturbance added to the control input, as in EQ 8.14).

The closed-loop denominator is $1 + s\tau$, so the single pole is at $s = -1/\tau = -K_p$. The pole is the negative of the loop gain. Increasing the gain moves the pole further into the left half-plane, so the transient $e^{-t/\tau} = e^{-K_p t}$ in the step response $y(t) = 1 - e^{-t/\tau}$ decays faster.

7. Integral-mode control

Question. *What is the impulse response of the integral-mode controller in Example 8.10?*

> **Formulas:** Example 8.10: plant $G(s) = 1/s$; control law $u = K_i \int (r - y)$; closed loop $Y = \dfrac{K_i}{s^2 + K_i} R + \dfrac{s}{s^2 + K_i} W$ (disturbance added to the control input, as in EQ 8.14); Table 5.4, row 7: $\sin(\omega t)u(t) \leftrightarrow \dfrac{\omega}{s^2 + \omega^2}$.

For an impulse command, $R(s) = 1$, and with no disturbance,

$$Y(s) = \frac{K_i}{s^2 + K_i} = \sqrt{K_i}\,\frac{\sqrt{K_i}}{s^2 + \left(\sqrt{K_i}\right)^2}.$$

Taking the inverse Laplace transform,

$$y(t) = \sqrt{K_i}\,\sin\!\left(\sqrt{K_i}\,t\right) u(t).$$

The poles are at $\pm j\sqrt{K_i}$, on the imaginary axis, so the system is marginally stable: the response oscillates forever with angular frequency $\sqrt{K_i}$ (period $2\pi/\sqrt{K_i}$). A larger loop gain gives a faster and larger oscillation, but never a decaying one. Compare this with Exercise 6, where proportional control places the pole on the negative real axis.

8. Stability

Question. *Use a numerical solver to determine the stability of an LTI system with denominator polynomial* $3s^5 - 4s^3 + 7s^2 + 10s + 1$.

> **Formulas:** BIBO stability (Section 8.8.1): an LTI system is BIBO stable if and only if all the roots of its denominator polynomial (its poles) have negative real parts.

A numerical root-finder (for example, `numpy.roots([3, 0, -4, 7, 10, 1])` in Python; note the explicit zero coefficient for s^4) gives the five poles

Pole	Real part
-0.10880	< 0
$-1.09346 \pm j\,0.23747$	< 0
$1.14786 \pm j\,1.06275$	> 0

Two poles have a positive real part, so the system is not BIBO stable. Their imaginary parts are nonzero, so its natural response contains a growing oscillation, with angular frequency about 1.06 radians/second and amplitude growing as $e^{1.15t}$.

Computer Systems Modelling: Lent 2022 **Assessment 1**
Instructor: Prof. S. Keshav 10-11 AM, February 23, 2023
Answer all questions Maximum marks 15

Question 1. You wish to use simulations to study the mean delay for customer requests waiting in a buffer to be processed by a video server system. The system examines the customer ID and forwards the request either to a high-priority server (for paying customers) or a low-priority server (for non-paying customers). Once a request enters one of these servers, it is never transferred to the other server. Each server has a single finite queue to buffer requests and it serves requests in FIFO order.

a. How would you represent the state of the system? Why? (0.5 marks)

b. Is the system ergodic or non-ergodic? Why? (1 mark)

c. Relying on the previous answer, sketch a simulation approach you would take to simulate the system. (1 mark)

d. State a stopping rule to ensure that, with 99% confidence, the estimate of the mean waiting time is accurate to ±2 ms? (1 mark) Hint: For the 99% confidence interval $z_{\alpha/2} = 2.58$.

e. Discuss how you might use antithetic variables to reduce the simulation time. (0.5 marks)

(Total: 4 marks)

Solution:

a. As a tuple (x,y) where x and y represent the occupancy in the low/high priority buffer.

b. The system is ergodic, since the system is finite, fully connected, and aperiodic.

c. Since it is ergodic, a single run with means computed for each sub-interval suffices.

d. Let n denote the number of sub-intevals, X_i denote the mean waiting time for the ith run, and $\bar{X}_n = 1/n \sum X_i$. First, run the simulation for a certain number n of sub-intervals so that n is not too small. After each subsequent sub-interval, update n and X_i and recompute $\bar{X}_n$. Then, compute $S_n = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X}_n)^2}{n-1}}$. Stop if $2.58 * \frac{S_n}{\sqrt{n}} < 2$ ms.

e. Generate request arrivals randomly building on a set of uniform random variables $\{u_i\}$ in even-numbered sub-intervals, using $\{1 - u_i\}$ for odd-numbered sub-intervals.

Question 2. Customers arrive to a stable queue with an infinite buffer and are served at a mean rate of 10 customers/s, with service times drawn from a one-sided Gaussian distribution (so all service times are non-negative). If the mean number of customers in the system is 20, what is a tight lower bound on the mean time a customer spends in the system? (2 marks)

Solution: The system is stable, so the arrival rate λ must be lower than the service rate $\mu = 10$ customers/s. From Little's law, $20 = \lambda T$, where T is the mean time in the system, including service. Since $\lambda < \mu = 10$, $20 < 10T$, so $T > 2$ s, and T approaches 2 s as λ approaches 10.

Question 3. A state is called absorbing if the probability of leaving this state is 0.

a. Is a Markov chain with an absorbing state irreducible? (0.5 mark)

b. Can a Markov chain with an absorbing state be ergodic? Why or why not? (1 mark)

Total marks for this question: 1.5

Solution:

a. Since we cannot go from an absorbing state to any other state, the chain is not irreducible.

b. Since return to any non-absorbing state is not certain, these states are non-recurrent (transient), and the chain is therefore non-ergodic. The only exception is a degenerate chain with a single absorbing state. (Potential question): what is the only ergodic Markov chain that has an absorbing state.

Question 4. We model the behaviour of a web surfer going from page to page entirely within Wikipedia as a homogeneous, discrete space, discrete time Markov process. We observe that the mean time spent at any page is 10 time units. What is the probability that the surfer leaves the current page at any time step? **(1 mark)**

Solution: The probability that the surfer stays on page i is p_{ii} and that they leave is 1-p_{ii}. Since the mean residence time in a state is $1/(1\text{-}p_{ii}) = 10$, the probability that they leave at any time step is 0.1.

Question 5. Prove that more than one ergodic Markov chain can have the same stationary probability distribution.**(1 mark)**

Solution: By example: two chains, each with three states such that the transition probabilities are symmetric, so that the stationary probability of being in any state is 1/3, but the probabilities differ.

Question 6. The probability that there can be zero arrivals to an M/M/1 queue over 10 seconds is 0.25. What is the probability of 3 arrivals in the same time period? (A final numeric answer is not necessary: an expression will suffice.) **(1 mark)**

Solution: Arrivals follow a Poisson process, so we have $e^{-10\lambda} = 0.25$, so that $10\lambda = \ln 4$, or $\lambda = \ln 4/10\,\mathrm{s}^{-1}$. The probability of 3 arrivals is $\frac{(10\lambda)^3}{3!}e^{-10\lambda} = \frac{0.25\,(\ln 4)^3}{6} \approx 0.111$.

Question 7. In effort to be responsive yet efficient, a supermarket manager introduces the following rule: normally there will be only one checkout counter open. However, if five customers are waiting for service (including the customer being served), then every customer would get their own checkout counter. The supermarket reverts to a single line whenever the total number of customers waiting for service or in service drops below five.

a. How would you model this system? **Clearly state all assumptions**. (1 mark)

b. What is the longest time someone would need to wait for service (1.5 marks)

Total marks for this question: 2.5

Solution:

a. Assume that customers arrive as a Poisson process and that service durations are exponential. Also assume that it takes negligible time for customers to go from one queue to many. Then, the system can be modeled as an M/M/1 queue as long as the total number of customers waiting for service (including when in line) is less than five, and as an M/M/∞ queue otherwise.

b. There is no bound on the longest wait time, since this is a stochastic process. However, the longest expected waiting time would be for the fourth customer in the single queue. Assuming an M/M/1 queue, PASTA and the memoryless property of exponential service times imply that the remaining service time of the customer in service also has mean $1/\mu$. So this customer waits, on average, for 3 mean service times, an expected delay of $3/\mu$ time units.

Question 8. Jobs arrive to a server as a Poisson process at a mean rate of 2 jobs/s and are served in FIFO order by a server with a mean service rate of 3 jobs/s. What is the largest

variance tolerable in the server if the mean waiting time (including service time) should be less than 5 seconds? **(2 marks)**

Solution: The mean service rate $\mu = 3$ and the mean arrival rate $= \lambda = 2$. So, $\rho = \lambda/\mu = 2/3$. The mean service time is $1/\mu = 1/3$ s. C_S^2, the square of the coefficient of variation, is defined as the ratio of the variance to the square of the mean service time. Let V denote the variance. Then, $C_S^2 = V/(1/3)^2 = 9V$. The expected sum of the time waiting for service and the service time, is given by:

$$\mathbb{E}(T) = \frac{1}{\mu}(1 + \frac{\rho(1 + C_s^2)}{2(1 - \rho)}) \tag{1}$$

Plugging in the values:

$$\mathbb{E}(T) = \frac{1}{3}(1 + \frac{2/3(1 + 9V)}{2(1 - 1/3)}) \tag{2}$$

$$\mathbb{E}(T) = \frac{1}{3}(1 + \frac{2/3(1 + 9V)}{2/3}) \tag{3}$$

$$\mathbb{E}(T) = \frac{1}{3}(1 + 1 + 9V) \tag{4}$$

We want this quantity to be less than 5, and after simplification, we get $V < 13/9$.

Computer Systems Modelling: Lent 2025 **Assessment 1**
Instructor: Prof. S. Keshav 10-11 AM, February 27, 2025
Answer all questions. Please return this sheet at the end of the assessment. Maximum marks 15

Question 1. You are asked to model the behaviour of a web surfer going from website to website as a stochastic process.

a) Is this process (i) discrete or continuous space (ii) discrete or continuous time? Justify your choices. **(1 mark)**

b) What assumption(s) do you need to make to make the process a homogenous Markov process. Give one example of a situation where each of these assumptions may not hold **(2 marks)**

c) Given your assumptions, you observe that the mean time spent at a website is 5 time units. What is the probability that the surfer spends 10 time units at a website? **(1 mark)**

Solution:
(a) This process is:

- **Discrete space** - The state space consists of distinct websites that can be visited, making it countable and discrete.
- **Continuous time** - The time spent on each website is a continuous variable that can take any positive real value.

Justification: While websites are distinct entities (you cannot be on 0.5 of a website), the duration of visits can be any real number (one can spend 2.37 minutes on a site).
(b) To make this a homogeneous Markov process, we need:

1. **Memoryless property:** The future state depends only on the current state, not on the history of previous states.
   Example violation: If the surfer uses browser history to avoid revisiting recent sites, the process depends on history.
2. **Time homogeneity:** Transition probabilities remain constant over time.
   Example violation: If the surfer's behavior changes between day and night (e.g., more likely to visit entertainment sites at night vs. work sites during day).
3. **Independence of time spent:** The time spent on a site doesn't affect transition probabilities.
   Example violation: If a long time spent on one site indicates high interest, making the user more likely to click related links.

(c) The probability of spending *exactly* 10 time units is 0. For the probability of spending *at least* 10 time units, since residence time in a continuous time Markov chain is exponentially distributed:

$$P(T > 10) = e^{-10/5} = e^{-2} = 0.135$$

Question 2.
Consider a continuous-time Markov chain with states {0, 1, 2} and the following transition rate matrix:

$$Q = \begin{bmatrix} -3 & 2 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

a) Calculate the stationary distribution $\pi = [\pi_0, \pi_1, \pi_2]$. **(1 mark)**

b) Calculate the expected time spent in state 0 before transitioning to another state. **(0.5 marks)**

Solution: **Part (a):** To find the stationary distribution π, we need to solve the system of equations $\pi Q = 0$ along with the normalization condition $\sum_{i=0}^{2} \pi_i = 1$.
From $\pi Q = 0$, we get three equations:

$$\begin{aligned} -3\pi_0 + \pi_1 + \pi_2 &= 0 \\ 2\pi_0 - 2\pi_1 + \pi_2 &= 0 \\ \pi_0 + \pi_1 - 2\pi_2 &= 0 \end{aligned}$$

Along with the normalization condition:

$$\pi_0 + \pi_1 + \pi_2 = 1$$

Let's solve this step by step:
From the first equation:

$$\pi_1 + \pi_2 = 3\pi_0 \tag{1}$$

From the second equation:

$$2\pi_0 - 2\pi_1 + \pi_2 = 0$$

Rearranging:

$$\pi_2 = 2\pi_1 - 2\pi_0 \tag{2}$$

Substituting (2) into (1):

$$\begin{aligned} \pi_1 + (2\pi_1 - 2\pi_0) &= 3\pi_0 \\ 3\pi_1 - 2\pi_0 &= 3\pi_0 \\ 3\pi_1 &= 5\pi_0 \\ \pi_1 &= \frac{5}{3}\pi_0 \end{aligned} \tag{3}$$

Substituting (3) back into (2):

$$\begin{aligned} \pi_2 &= 2(\frac{5}{3}\pi_0) - 2\pi_0 \\ \pi_2 &= \frac{10}{3}\pi_0 - 2\pi_0 \\ \pi_2 &= \frac{4}{3}\pi_0 \end{aligned} \tag{4}$$

Using the normalization condition:

$$\begin{aligned} \pi_0 + \frac{5}{3}\pi_0 + \frac{4}{3}\pi_0 &= 1 \\ \pi_0(1 + \frac{5}{3} + \frac{4}{3}) &= 1 \\ \pi_0(\frac{12}{3}) &= 1 \\ \pi_0 &= \frac{1}{4} \end{aligned}$$

Therefore:

$$\pi_0 = \frac{1}{4}$$
$$\pi_1 = \frac{5}{12}$$
$$\pi_2 = \frac{1}{3}$$

Part (b): The expected time spent in state 0 before transitioning to another state is the reciprocal of the total rate of leaving state 0. This rate is given by the negative of the diagonal entry in the first row of Q, which is 3.
Therefore:

$$E[T_0] = \frac{1}{3} \text{ time units}$$

This interpretation makes intuitive sense because the process leaves state 0 at rate 2 to state 1 and rate 1 to state 2, for a total rate of 3, making the expected waiting time $\frac{1}{3}$.

Question 3.
Consider a birth-death process modeling a population where births occur at rate $\lambda = 2$ per unit time and deaths occur at rate $\mu = 1$ per unit time, regardless of the population size.

a) Calculate the approximate probability that the population size increases by exactly 2 in a time interval of length 1. **(2 marks)**

b) What is the long-term average population size? **(0.5 marks)**

Solution: **(a)** To find the probability of exactly 2 net increases in population during 1 time unit, we need to consider all combinations of births and deaths that result in a net increase of 2.
Let B(t) denote births and D(t) denote deaths in time t. For a net increase of 2:

$$B(1) - D(1) = 2$$

The probability is the sum of all possible combinations:

$$P(\text{Net increase} = 2) = \sum_{k \geq 2} P(B(1) = k) \times P(D(1) = k - 2)$$

Using Poisson probabilities:

$$P(B(1) = k) = \frac{e^{-2} \cdot 2^k}{k!}$$

$$P(D(1) = k) = \frac{e^{-1} \cdot 1^k}{k!}$$

Calculating the first few significant terms:

$$k = 2 : P(B = 2)P(D = 0) = \frac{2^2 e^{-2}}{2!} \cdot e^{-1}$$
$$k = 3 : P(B = 3)P(D = 1) = \frac{2^3 e^{-2}}{3!} \cdot \frac{e^{-1}}{1!}$$
$$k = 4 : P(B = 4)P(D = 2) = \frac{2^4 e^{-2}}{4!} \cdot \frac{e^{-1}}{2!}$$

Adding these terms:

$$P(\text{Net increase} = 2) \approx 0.182 \quad \text{(summing all the terms gives 0.185)}$$

(b) For the long-term average population size:
Since $\lambda > \mu$ (2 ¿ 1), this is a transient birth-death process. The expected population size grows linearly with time:

$$E[X(t)] = X(0) + (\lambda - \mu)t \to \infty \text{ as } t \to \infty$$

Therefore, the long-term average population size does not exist, as the population grows without bound at an average rate of $\lambda - \mu = 1$ per time unit.

Question 4. Consider an M/M/1/K queueing system where $\lambda = 50$ customers/hour, $\mu = 60$ customers/hour, and $K = 10$ (including the customer being served).

a) Calculate the probability that the system is full, i.e., the blocking probability. **(1 mark)**

b) Derive an expression for the *effective* arrival rate λ_{eff}, considering that some customers are blocked when the system is full. **(1 mark)**

c) You are given that the mean number of customers in the system is 3.9748. Use this to calculate the average time a customer spends in the system (including service time) for those customers who are not blocked. **(1 mark)**

d) How would you modify the system to reduce the blocking probability to less than 1%? Justify your answer with calculations. **(1.5 marks)**

Solution: a) For an M/M/1/K queue, the blocking probability is given by:

$$P_K = \frac{\rho^K(1-\rho)}{1-\rho^{K+1}}$$

where $\rho = \frac{\lambda}{\mu} = \frac{50}{60} \approx 0.8333$

$$P_{10} = \frac{0.8333^{10}(1-0.8333)}{1-0.8333^{11}} \approx 0.031$$

The blocking probability is approximately 3.1%.
b) The effective arrival rate λ_{eff} is the rate at which customers are actually admitted to the system. It's given by:

$$\lambda_{eff} = \lambda(1 - P_K)$$

$$\lambda_{eff} = 50(1 - 0.031) = 48.445$$

customers/hour
c) The average time a customer spends in the system (for those not blocked) can be calculated using Little's Law:

$$W = \frac{L}{\lambda_{eff}}$$

$$W = \frac{3.9748}{48.445} \approx 0.082\,\text{h}$$

or 4.92 minutes

d) To reduce the blocking probability to less than 1%, we need to increase K. We can find the minimum K that satisfies:

$$\frac{\rho^K(1-\rho)}{1-\rho^{K+1}} < 0.01$$

Solving this numerically (by trying increasing values of K), we find that $K = 16$ gives a blocking probability of approximately 0.0094, which is just below 1%, whereas $K = 15$ gives 0.0114.
To justify: With K = 16,

$$P_{16} = \frac{0.8333^{16}(1-0.8333)}{1-0.8333^{17}} \approx 0.0094$$

Therefore, to reduce the blocking probability to less than 1%, the system should be modified to accommodate at least 16 customers (including the one being served).

Question 5.
Consider an M/G/1 queue where customers arrive according to a Poisson process with rate $\lambda = 10$ per hour. The service time S has the following distribution:

$$P(S = 1) = 0.2, \quad P(S = 2) = 0.3, \quad P(S = 5) = 0.5 \quad \text{(in minutes)}$$

a) Calculate the mean and variance of the service time. **(1 mark)**

b) Using the Pollaczek-Khinchin formula, calculate the average number of customers in the queue.**(1 mark)**

c) Calculate the probability that an arriving customer finds the system empty.**(0.5 marks)**

Solution:

a) $E[S] = 0.2(1) + 0.3(2) + 0.5(5) = 3.3$ minutes $= 0.055$ hours
$E[S^2] = 0.2(1^2) + 0.3(2^2) + 0.5(5^2) = 13.9$ minutes^2
$Var(S) = E[S^2] - (E[S])^2 = 13.9 - 3.3^2 = 3.01$ minutes^2

b) $\rho = \lambda E[S] = 10(0.055) = 0.55$. With $E[S^2] = 13.9$ minutes^2 $= 13.9/3600$ hours^2, the mean number of customers in the queue is $N_q = \frac{\lambda^2 E[S^2]}{2(1-\rho)} = \frac{100 \cdot 13.9/3600}{2(0.45)} \approx 0.43$, and the mean number in the system is $N_q + \rho \approx 0.98$.

c) $P(\text{system empty}) = 1 - \rho = 1 - 0.55 = 0.45$

Computer Systems Modelling: Lent 2023 **Assessment 2**
Instructor: Prof. S. Keshav 10-11 AM, March 16, 2023
Answer all questions Maximum marks 15

Question 1. Use Euler's identity to prove that

$$sin(A+B) = sinAcosB + cosAsinB$$

(1 mark)

Solution: The right hand side is

$$\frac{(e^{jA} - e^{-jA})(e^{jB} + e^{-jB})}{(2)(2j)} + \frac{(e^{jA} + e^{-jA})(e^{jB} - e^{-jB})}{(2)(2j)}$$

which reduces trivially to

$$\frac{2e^{j(A+B)} - 2e^{-j(A+B)}}{4j} = \frac{e^{j(A+B)} - e^{-j(A+B)}}{2j} = sin(A+B)$$

Question 2. Compute the convolution of a function that has the constant value of 1 in the interval [-1, 1] with the function x^2 clipped to interval [0,1].**(3 marks)**

Solution: Let $a(t)$ denote the clipped quadratic and let $b(t)$ denote the constant function. We want to compute $a(t) \otimes b(t)$, which is defined as

$$a(t) \otimes b(t) = \int_0^\infty a(\tau)b(t-\tau)d\tau \tag{1}$$

Note that $b(\tau)$ is symmetric about the Y axis, so this can be rewritten as:

$$a(t) \otimes b(t) = \int_0^\infty a(\tau)b(\tau-t)d\tau \tag{2}$$

which we recognize as time-shifting $b(t)$ by t. Thus, the integral can be reduced to

$$\begin{cases} \int_0^{t+1} \tau^2 d\tau \text{ for } -1 \leq t < 0 \\ \int_0^1 \tau^2 d\tau \text{ for } 0 \leq t < 1 \\ \int_{t-1}^1 \tau^2 d\tau \text{ for } 1 \leq t \leq 2 \end{cases}$$

$$\begin{cases} \frac{\tau^3}{3}\Big|_0^{t+1} & \text{for } -1 \leq t < 0 \\ \frac{\tau^3}{3}\Big|_0^1 & \text{for } 0 \leq t < 1 \\ \frac{\tau^3}{3}\Big|_{t-1}^1 & \text{for } 1 \leq t \leq 2 \end{cases}$$

$$\begin{cases} \frac{(t+1)^3}{3} & \text{for } -1 \leq t < 0 \\ \frac{1}{3} & \text{for } 0 \leq t < 1 \\ \frac{1-(t-1)^3}{3} & \text{for } 1 \leq t \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

Question 3. What is the kth term of the Fourier series corresponding to the eternal signal $(cost + sin2t)$ **(2 marks)**

Solution: Since the second term is a harmonic of the first, the period is $T_0 = 2\pi$ and $\omega_0 = 2\pi/T_0 = 1$. The coefficients are

$$c_k = \frac{1}{2\pi}\int_0^{2\pi} (\cos t + \sin 2t)\, e^{-jkt}\, dt.$$

Using Euler's formula, $\cos t + \sin 2t = \frac{1}{2}e^{jt} + \frac{1}{2}e^{-jt} + \frac{1}{2j}e^{j2t} - \frac{1}{2j}e^{-j2t}$, so the integrand is a sum of terms of the form $e^{j(m-k)t}$. Each such term integrates to 2π over a period if $m = k$, and to 0 otherwise. Therefore

$$c_1 = c_{-1} = \frac{1}{2}, \qquad c_2 = \frac{1}{2j} = -\frac{j}{2}, \qquad c_{-2} = -\frac{1}{2j} = \frac{j}{2},$$

and $c_k = 0$ for all other k. The kth term of the series is $c_k e^{jkt}$.

Question 4. Use the integral rule to find the Laplace transform of the function t^2. **(1 mark)**

Solution: Integrating $u(t)$ twice gives $t^2/2$, so $\mathcal{L}(t^2/2) = \mathcal{L}(u(t))/s^2 = 1/s^3$, and $\mathcal{L}(t^2) = 2/s^3$.

Question 5. Suppose that the throughput of a network flow, T, is related to the TCP buffer size at an endpoint, the control variable B, by the equation:

$$2\frac{dT^2}{dB} + 3\frac{dT}{dB} + T = B$$

What is the transfer function? (1 mark) What is the time evolution the system (1 mark) Is this system stable? Why or why not? (1 mark) **(Total 3 marks)**

Solution: Taking the Laplace transform of both sides

$$2s^2T(s) + 3sT(s) + T(s) = B(s)$$

$$T(s) = \frac{B(s)}{2s^2 + 3s + 1}$$

So, the transfer function is

$$\frac{1}{2s^2 + 3s + 1}$$

$$= \frac{1}{(2s+1)(s+1)} = \frac{2}{2s+1} - \frac{1}{s+1}$$

This immediately tells us that the time evolution of the system is of the form

$$c_1 e^{-1/2t} - c_2 e^{-t}$$

from the linearity of the Laplace transform and the fact that $\frac{K}{s+a} \Longleftrightarrow e^{-at}$.
Both roots of the transfer function (-1 and -1/2) are in the left half plane, so the system is stable.

Question 6. In an underdamped system, we find that the response to a unit step input at time 0 is a decaying sinusoid with a phase shift of $\pi/3$ and a natural frequency of 2π rad/s, asymptoting to a unit step. What is the transfer function? **(2 marks)**

Solution: We have $\omega_d = 2\pi$, $\zeta = \cos(\pi/3) = 0.5$ and $K = 1$. ω_d and ω_n are related by $\omega_d^2 = \omega_n^2(1 - \zeta^2)$, so

$$\omega_n = \frac{4\pi}{\sqrt{3}}$$

the transfer function is

$$\frac{1}{(\frac{3s^2}{16\pi^2}) + \frac{\sqrt{3}s}{4\pi} + 1}$$

Question 7.

a. What is the output $y(t)$ of a controlled plant with transfer function $G(s) = \frac{1}{s+a}$ when the controller is defined by $U(s) = bs(R–Y)$, the command R is the unit step function, and $W = 0$. (2 marks)

b. What is the steady state error given a step input to this controller? (1 mark)

(Total 3 marks)

Solution: $Y = G(s)U(s) = \frac{bs(R-Y)}{s+a}$, so $Y = \frac{bs}{(1+b)s+a}R$. For a step input $R(s) = \frac{1}{s}$, and $Y = \frac{b}{(1+b)s+a}$ $= (\frac{b}{1+b})(\frac{1}{s+\frac{a}{1+b}})$, which is of the form $\frac{K}{s+a}$, so the inverse transform is

$$y(t) = \frac{b}{1+b}e^{-\frac{at}{1+b}}$$

For a step input where $R(s) = 1/s$, we have already shown that $Y = \frac{b}{(1+b)s+a}$. By the final value theorem,

$$y(\infty) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} \frac{bs}{(1+b)s+a} = 0.$$

So the output decays to zero, and the steady state error is $1 - y(\infty) = 1$.

Computer Systems Modelling: Lent 2025 **Assessment 3**
Instructor: Prof. S. Keshav 10-11 AM, March 20, 2025
Answer all questions Maximum marks 15

Question 1.
Consider the periodic square wave function:

$$f(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & 1 \leq t < 2 \end{cases}$$

where $f(t+2) = f(t)$ for all t. Derive the Fourier series expansion of $f(t)$ as a sum of sines and cosines. **(2 marks)**

Solution:
For a periodic function with period $T_0 = 2$, $\omega_0 = \frac{2\pi}{T_0} = \pi$. So the Fourier series is given by:

$$f(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t)$$

$$= a_0 + \sum_{k=1}^{\infty} (a_k \cos k\pi t + b_k \sin k\pi t)$$

First, we calculate a_0:

$$a_0 = \frac{1}{T_0} \int_{T_0} f(t)dt = \frac{1}{2} \int_0^2 f(t)dt = \frac{1}{2} \left[\int_0^1 1\, dt + \int_1^2 0\, dt \right] = \frac{1}{2} \cdot 1 = \frac{1}{2}$$

For the cosine coefficients a_k:

$$\begin{aligned} a_k &= \frac{2}{T_0} \int_{T_0} f(t) \cos k\omega_0 t dt \\ &= \frac{2}{2} \int_{T_0} f(t) \cos k\pi t dt \\ &= \int_0^1 1 \cdot \cos(k\pi t)\, dt + \int_1^2 0 \cdot \cos(k\pi t)\, dt \\ &= \frac{1}{k\pi} \sin(k\pi t) \Big|_0^1 \\ &= \frac{1}{k\pi} [\sin(k\pi) - \sin(0)] \\ &= \frac{1}{k\pi} \cdot 0 = 0 \quad \text{for all } k \geq 1 \end{aligned}$$

For the sine coefficients b_k:

$$\begin{aligned}
b_k &= \frac{2}{T_0}\int_{T_0} f(t)\sin k\omega_0 t dt \\
&= \frac{2}{2}\int_0^2 f(t)\sin k\pi t dt \\
&= \int_0^1 1\cdot\sin(k\pi t)\,dt + \int_1^2 0\cdot\sin(k\pi t)\,dt \\
&= -\frac{1}{k\pi}\cos(k\pi t)\Big|_0^1 \\
&= -\frac{1}{k\pi}[\cos(k\pi)-\cos(0)] \\
&= -\frac{1}{k\pi}[(-1)^k - 1] \\
&= \frac{1}{k\pi}[1-(-1)^k]
\end{aligned}$$

Therefore:

$$b_k = \begin{cases} \frac{2}{k\pi}, & \text{for } k \text{ odd} \\ 0, & \text{for } k \text{ even} \end{cases}$$

The Fourier series expansion is:

$$f(t) = \frac{1}{2} + \sum_{k=1}^{\infty}\frac{1}{k\pi}[1-(-1)^k]\sin(k\pi t)$$

$$f(t) = \frac{1}{2} + \sum_{k=1,3,5,\ldots}^{\infty}\frac{2}{k\pi}\sin(k\pi t)$$

Question 2. A website owner measures the number of hits that it receives every second for several years. Treating this as a signal, if you did a Fourier transform, at what frequencies would you expect to find the most energy? Justify your answer. **(1 mark)**

Solution: 1/day, 1/week, 1/year since this is are natural periods at which there is cyclicity.

Question 3. Use the integral rule to find the Laplace transform of the function t^2. **(1 mark)**

Solution: We have

$$t^2 = 2\int t dt$$

So

$$L(t^2) = 2L(\int t dt)$$

But

$$t \leftrightarrow 1/s^2$$

So

$$L(\int t dt) = 1/sL(t) = 1/s\cdot 1/s^2 = 1/s^3$$

$$L(t^2) = 2/s^3$$

Question 4. With dynamic voltage scaling, a CPU increases or decreases its cycle rate as a function of a voltage control signal, allowing server power to be reduced when the CPU load is

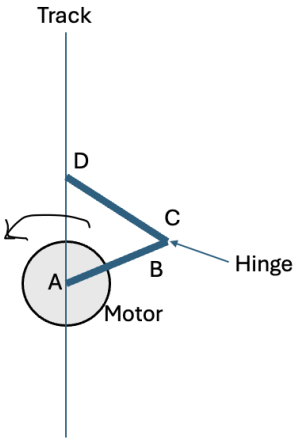


Figure 1: Hinged rotating rods

low. Suppose that the cycle rate $C(t)$ at time t is proportional to the weighted sum of the control voltage at that time $V(t)$ as well as the first derivative of this control voltage, and that this rate is an offset from a fixed base rate C_{min}.

a. Represent the system with an appropriate equation in the time domain (0.5 marks)

b. Is it linear? (0.5 marks)

c. What is the system equation in the Laplace domain? (1 mark)

(Total: 2 marks)

Solution: The time-domain relationship is

$$C(t) = \alpha V(t) + \beta V'(t) + C_{min}u(t)$$

Because of the constant offset term, the system is not linear.
Taking the Laplace transform

$$C(s) = \alpha V(s) + \beta s V(s) + C_{min}/s$$

Question 5. Consider the system in Figure 1. Here, a unit length rigid rod AB is coupled to the shaft of an electrically driven motor at one end A, and the other end B is attached by a hinge to end C of another unit length rigid rod CD so that the rod CD can freely and completely rotate around rod AB. End D of the rod CD is restricted to move vertically along a track as shown: it *must* touch the track at all times. Assume that the two rods are in the same vertical plane at all times and that the motor makes one complete rotation in 2π seconds.

a. If the motion of the rod AB is modelled as a phasor, how would you model the movement of end D of rod CD? **(0.5 marks)**

b. Give the equation of motion of point D *relative to point C* in the form of a complex exponential. Justify your answer. **(1 mark)**

c. Give the equation of motion of point D *relative to point A*. Justify your answer. **(1 mark)**

d. Now, assume that the motor is non-ideal, so cannot be relied to maintain the desired speed of rotation. However, its rotation speed is directly proportional to its input voltage. Describe a control system that can be used to maintain the desired rate of one rotation per 2π seconds. Specifically, write down a table with six rows that identifies the following six features: the plant, the measurement process, the disturbance, the reference input, the feedback, and the control input. **(1.5 marks)**

e. Assume that the motor can be modelled as a first-order system. If, at time 0, a unit voltage is applied to the motor, we find that a rotation speed of .63 radians/s is achieved at time 5 and that the final rate of rotation is 1 radian/s. What is the equation of motion of the motor when given a step input? **(1 mark)**

f. Describe how you could use proportional-mode control to achieve the desired rate of rotation. Identify a problem with the use of proportional mode control in this specific system.**(1 mark)**

Solution:

a. As the sum of two phasors.

b. $-e^{-jt} = e^{j(\pi - t)}$: it moves clockwise at 1 rad/s around point C, starting from angle π.

c. 2 sin t (2j sin t is also acceptable), since this is the sum of two phasors with opposite directions of rotation.

d.

Plant	the motor, two rods, the groove, rotation sensor
Measurement process	sensor that measures motor rotation rate
Disturbance	anything that causes the motor to not run at the desired rate
Reference input	1 rad/s
Feedback	actual rotation rate
Control input	control voltage

e. 63% achieved at τ, so equation is $y(t) = 1 - e^{-t/5}$

f. Voltage changes in proportion to error. Since there is need for non-zero control at steady state, there will be some droop.

Question 6. Suppose that the output of a system T, is related to the control variable B, by the equation:

$$2\frac{dT^2}{d^2B} + 3\frac{dT}{dB} + T = B$$

a. What is the transfer function? **(1 mark)**

b. What is the output as a function of time when the system is given a unit step input at time 0? **(1 mark)**

c. Is this system stable? Why or why not? **(1 mark)**

Solution: Taking the Laplace transform of both sides

$$2s^2T(s) + 3sT(s) + T(s) = B(s)$$

$$T(s) = \frac{B(s)}{2s^2 + 3s + 1}$$

So, the transfer function is

$$\frac{1}{2s^2+3s+1}$$
$$= \frac{1}{(2s+1)(s+1)}$$

Given a unit step input, whose Laplace transform is $1/s$, the Laplace transform of the output is

$$= \frac{1}{s(2s+1)(s+1)} = \frac{1}{s} - \frac{2}{s+1/2} + \frac{1}{s+1}$$

and, in the time domain

$$y(t) = 1 - 2e^{-t/2} + e^{-t}$$

Both roots of the transfer function (-1 and -1/2) are in the left half plane, so the system is stable.