

A Minimal Book Example

Brandon Stark

2020-08-31

Contents

| | | |
|----------|--|-----------|
| 1 | Prerequisites | 5 |
| 2 | Introduction | 7 |
| 3 | Literature | 9 |
| 4 | Root Locus | 11 |
| 4.1 | Setting up a Root Locus | 11 |
| 4.2 | Properties | 12 |
| 5 | Digital Systems | 13 |
| 5.1 | Discrete Time | 13 |
| 5.2 | Fundamental Discrete Time Signals} | 14 |
| 6 | Final Words | 17 |

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.org/tinytex/>.

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter 4.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))  
plot(pressure, type = 'b', pch = 19)
```

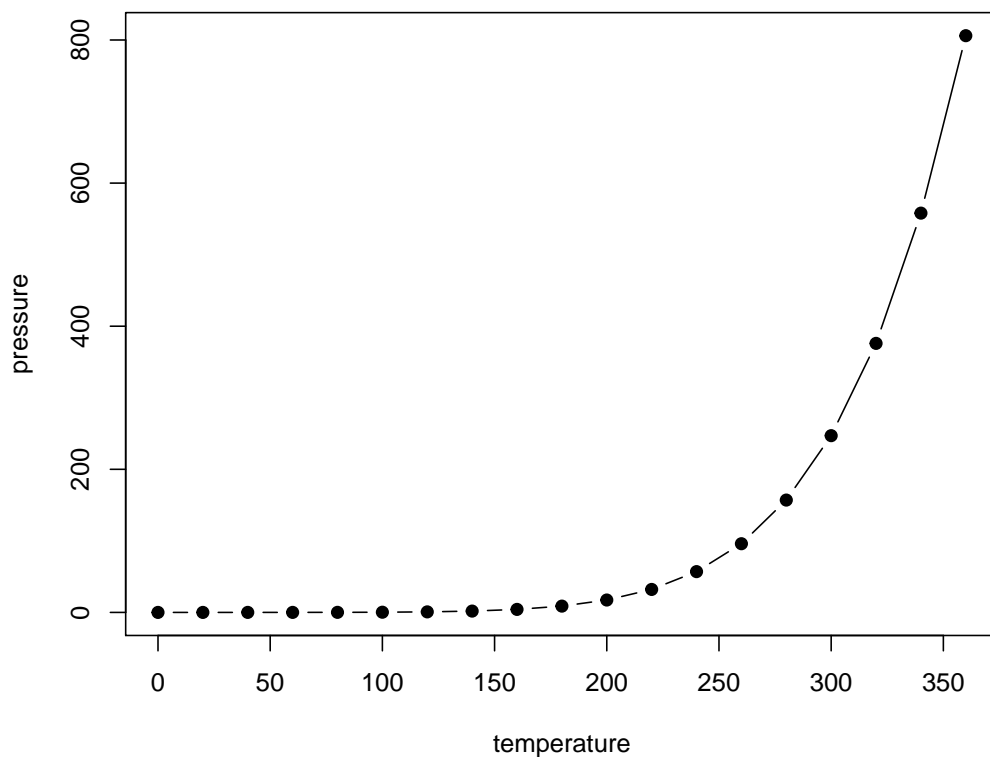


Figure 2.1: Here is a nice figure!

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

Table 2.1: Here is a nice table!

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| 5.7 | 4.4 | 1.5 | 0.4 | setosa |
| 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| 5.1 | 3.8 | 1.5 | 0.3 | setosa |

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2020) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 3

Literature

Here is a review of existing methods.

Chapter 4

Root Locus

The root locus method shows how changes in one of the system parameters modifies the roots of the characteristic equation of a system.

This in turn, shows the changes in the dynamic response of the system.

In simple terms, if I increase the gain of a controller of a closed-loop system, how much will the damping ratio be effected - without having to explicitly find a solution.

Before the advent of software such as Matlab, root locus plots played a significant role in analysing system performance. It is no longer critical to know how to draw a root locus plot, but it is still valuable to understand how to read and understand a root locus plot.

4.1 Setting up a Root Locus

Most commonly, the root locus is used to study the effect of loop gain variation of a closed-loop system.

Given a transfer function of a closed-loop system

$$\frac{Y(s)}{R(s)} = \mathcal{T}(s) = \frac{D_c(s)G(s)}{1 + D_c(s)G(s)} \quad (4.1)$$

The characteristic function is given as

$$0 = 1 + D_c(s)G(s)H(s) \quad (4.2)$$

With careful consideration, this can be rewritten as

$$0 = 1 + K \frac{b(s)}{a(s)} = 1 + KL(s) \quad (4.3)$$

or, even better, as

$$a(s) + Kb(s) = 0 \quad (4.4)$$

As we already know, the characteristic equation sets the placement of the poles of the system

$$a(s) + Kb(s) = 0 \quad (4.5)$$

As we change K, we can change the locations of the poles, and thus design a system to our desired dynamics.

For example, given a plant and a K controller

$$G = \frac{1}{s^2 + 2s + 3} \quad (4.6)$$

The closed loop equation is given as

$$T(s) = \frac{K}{s^2 + 6s + 3 + K} \quad (4.7)$$

The characteristic equation is given as

$$0 = s^2 + 6s + 3 + K \quad (4.8)$$

Using the quadratic formula, we can solve for the poles of this system

$$p = -1 \pm \frac{\sqrt{36 - 4(3 + K)}}{2} \quad (4.9)$$

When $K = 0$, the poles of the system become $p = -1 \pm \frac{5}{2}$. At $K = 6$, the poles of the system become $p = -3$. Above $K > 6$, the poles of the system diverge along the imaginary axis at -3 on the real axis.

The graph of all possible roots of the characteristic equation relative to parameter K is called the root locus and the methods to construct a graph is known as the root locus method of Evans.

4.2 Properties

The root locus is the set of values of s for which $1 + KL(s) = 0$ is satisfied as the real parameter K varies from 0 to $+\infty$. Typically, $1 + KL(s) = 0$ is the characteristic equation of the system and in this case, the roots of the locus are the closed-poles of that system. In otherwords, we do not have to work with the closed-loop system to study the performance of the K controller.

Even more interesting, given $1 + KL = 0$ and assuming that K must be real and positive, then L must be real and negative such that $L = -\frac{1}{K}$.

Of note, the phase of a real, negative system must equal 180° .

Definition 4.1 (Root Locus). The root locus of $L(s)$ is the set of points in the s-plane where the phase of $L(s) = 180^\circ$. We can test whether a point in the s-plane is on the locus by defining the angle to the test point from a zero as ψ_i and the angle to the test point from a pole as ϕ_i . The summation of all the angles of the zeros minus the summation of all the angles of the poles must be 180° .

$$\sum \psi_i - \sum \phi_i = 180^\circ + 360^\circ (l - 1) \quad (4.10)$$

where l is some integer.

Exercise 4.1 (Testing Excercise). Can you solve this equation?

Solution. This is how you solve it.

Chapter 5

Digital Systems

In most modern systems, we've moved away from analog controllers to digital controllers for a number of reasons.

5.1 Discrete Time

The basis for a digital system is in discrete time signals.

Discrete-Time Signals are

1. Composed of samples at discrete instants of time;
2. Often defined at evenly spaced intervals of time;
3. Undefined inbetween samples;

An example can be seen in Figure 5.1. The signal is defined only for the samples and undefined (or rather, not drawn) in between.

5.1.1 Notation

Given an analog signal $y(t)$, we can sample it at every T seconds. The resulting signal can be written as $y(kT)$ where k now represents the sample step index.

If the signal is sampled at regular increments of time T , then T is also known as the Sample Period. The inverse of the sample period is the sample rate.

- Sample Period - T (seconds)
- Sample Rate - $1/T$ (1/seconds = Hertz), f_s

The notation $y(kT)$ indicates a value of $y(t)$ at $t = kT$. The notation $y[k]$ denotes a discrete time signal that is only defined only for k an integer. Written more succinctly: Parentheses indicate continuous time, brackets indicate discrete time.

It is important to reiterate that a discrete-time signal is only defined for k is an integer. A signal $f[k]$ is defined when k is an integer; $f[1.6]$ does not exist. However, the index for a discrete time signal may be positive or negative. A sequence, written as $f[n]$, may describe ..., $f[-2]$, $f[-1]$, $f[0]$, $f[1]$, $f[2]$,

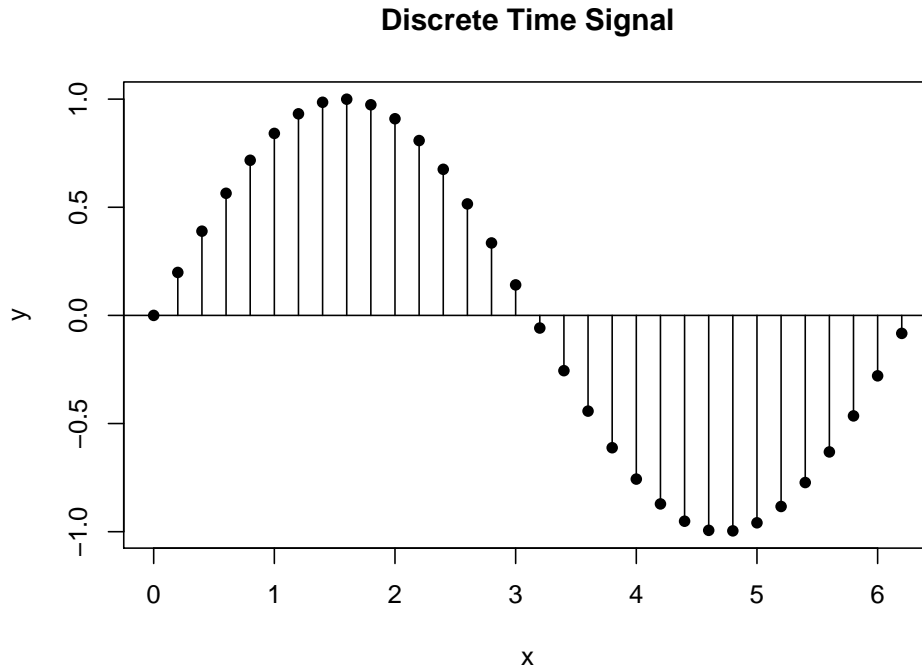


Figure 5.1: Here is a nice figure!

5.1.2 Discrete-Amplitude Signal

Most commonly, a discrete time signal is also a Discrete Amplitude signal. A discrete amplitude signal is a signal for which $y[n]$ has only a finite possible amplitudes, in contrast to a continuous amplitude signal, for which the amplitude can assume any value $-\infty < y[n] < \infty$.

A discrete time, discrete amplitude signal is known as a Digital Signal. An example of a digital signal is the output of an analog-to-digital converter which converts a continuous time, continuous amplitude signal into a signal readable by a computer. The output of the analog-to-digital converter is represented by an eight bit binary number - which can only assume $2^8 = 256$ unique values.

5.2 Fundamental Discrete Time Signals}

5.2.1 Unit Step Function and Unit Impulse Function

The first two fundamental discrete time signals are the Unit Step Function and Unit Impulse Function.

The unit step function is defined by:

$$u[k] = \begin{cases} 0, & k = -1, -2, \dots \\ 1, & k = 0, 1, 2, \dots \end{cases} \quad (5.1)$$

The output of a unit step function $f[k]$ is 0 for all integers of k less than 0, and is 1 for all integers of k equal to or greater than 0.

Similarly, the unit impulse function is defined by:

$$\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

The output of a unit impulse function $\delta[k]$ is 1 for $k = 0$ and is 0 for all other values of k .

5.2.2 Time Shifting, Time Scaling, Time Reversal

Given a discrete time signal, it can be utilized with three time transformations: Discrete Time Shifting, Discrete Time Scaling and Discrete Time Reversal.

Discrete Time Shift Given a signal $y[k]$, the signal can be shifted along the time axis as $y[k - 2]$. This shifts the signal such that it starts 2 steps later - in otherwords, it delays the signal.

Chapter 6

Final Words

We have finished a nice book.

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.20.