Chapter 5
Signals and Systems

Imagine that you are asked to design a system to steer a car straight down the middle of a lane. It seems easy, right? You can figure out some way to sense the position of the car within its lane. Then, if the car is right of center, turn the steering wheel to the left. As the car moves so that it is less to the right, turn the steering wheel less to the left. If it is left of center, turn the steering wheel to the right. This sort of *proportional controller* works well for many applications – but not for steering, as can be seen below.

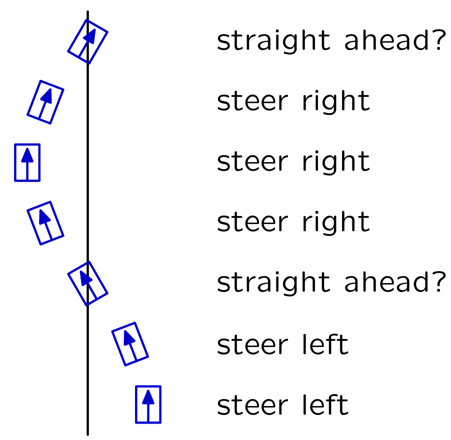


Figure 5.1 Simple (but poor) algorithm for steering: steer to the left in proportion to how far the car is to the right, and vice versa.

It is relatively easy to describe better algorithms in terms that humans would understand: e.g., *Stop turning back and forth!* It is not so easy to specify exactly what one might mean by that, in a way that it could be automated.

In this chapter, we will develop a *Signals and Systems* framework to facilitate reasoning about the dynamic behaviors of systems. This framework will enable construction of simple mathematical models that are useful in both analysis and design of a wide range of systems, including the car-steering system.

5.1 The signals and systems abstraction

To think about dynamic behaviors of systems, we need to think not only about how to describe the system but also about how to describe the *signals* that characterize the inputs and outputs of the system, as illustrated below.

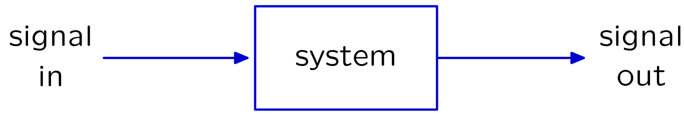


Figure 5.2 Signals and Systems: the system transforms an input signal into an output signal.

This diagram represents a system with one input and one output. Both the input and output are *signals*. A signal is a mathematical function with an independent variable (most often it will be *time* for the problems that we will study) and a dependent variable (that depends on the independent variable). The *system* is described by the way that it transforms the input signal into the output signal. In the simplest case, we might imagine that the input signal is the time sequence of steering-wheel angles (assuming constant speed) and that the output signal is the time sequence of distances between the center of the car and the midline of the lane.

Representing a system with a single input signal and a single output signal seems too simplistic for any real application. For example, the car in the steering example (**figure 5.1**) surely has more than one possible output signal.

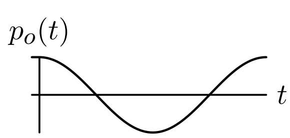
Exercise 5.1. List at least four possible output signals for the car-steering problem.

Possible output signals include

- its three-dimensional position (which could be represented by a 3D vector $\hat{p}(t)$ or by three scalar functions of time),
- its angular position,
- the rotational speeds of the wheels,
- the temperature of the tires, and many other possibilities.

The important point is that the first step in using the signals and systems representation is *abstraction*: we must choose the output(s) that are most relevant to the problem at hand and abstract away the rest.

To understand the steering of a car, one vital output signal is the lateral position $p_o(t)$ within the lane, where $p_o(t)$ represents the distance (in meters) from the center of the lane. That signal alone tells us a great deal about how well we are steering. Consider a plot of $p_o(t)$ that corresponds to **figure 5.1**, as follows.



The oscillations in $p_o(t)$ as a function of time correspond to the oscillations of the car within its lane. Thus, this signal clearly represents an important failure mode of our car steering system.

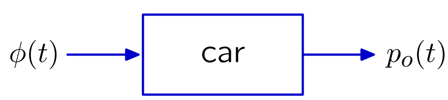
Is $p_o(t)$ the only important output signal from the car-steering system? The answer to this question depends on your goals. Analyzing a system with this single output is likely to give important insights into some systems (e.g., low-speed robotic steering) but not others (e.g., NASCAR). More complicated applications may require more complicated models. But all useful models focus on the most relevant signals and ignore those of lesser significance.[34]

Throughout this chapter, we will focus on systems with one input signal and one output signal (as illustrated figure 5.2). When multiple output signals are important for understanding a problem, we will find that it is possible to generalize the methods and results developed here for single-input and single-output systems to systems with multiple inputs and outputs.

The signals and systems approach has very broad applicability: it can be applied to mechanical systems (such as mass-spring systems), electrical systems (such as circuits and radio transmissions), financial systems (such as markets), and biological systems (such as insulin regulation or population dynamics). The fundamental notion of signals applies no matter what physical substrate supports them: it could be sound or electromagnetic waves or light or water or currency value or blood sugar levels.

5.1.1 Modularity, primitives, and composition

The car-steering system can be analyzed by thinking of it as the combination of car and steering sub-systems. The input to the car is the angle of the steering wheel. Let's call that angle $\phi(t)$. The output of the car is its position in the lane, $p_o(t)$, measured as the lateral distance to the center of the lane.



The steering controller turns the steering wheel to compensate for differences between our desired position in the lane, $p_i(t)$ (which is zero since we would like to be in the center of the lane), and our actual position in the lane $p_o(t)$. Let $e(t) = p_i(t) - p_o(t)$. Thus we can think about the steering controller as having an input $e(t)$ and output $\phi(t)$.

In the composite system (in figure 5.3), the steering controller determines $\phi(t)$, which is the input to the car. The car generates $p_o(t)$, which is subtracted from $p_i(t)$ to get $e(t)$ (which is the input to the steering controller). The triangular component is called a *gain* or *scale* of -1: its output is equal to -1 times its input. More generally, we will use a triangle symbol to indicate that we are multiplying all the values of the signal by a numerical constant, which is shown inside the triangle.

The dashed-red box in figure 5.3 illustrates *modularity* of the signals and systems abstraction. Three single-input, single-output sub-systems (steering controller, car, and inverter) and an adder (two inputs and 1 output) are combined to generate a new single-input ($p_i(t)$), single-output ($p_o(t)$) system. By abstraction, we could treat this new system as a primitive (represented by a single-input single-output box) and combine it with other subsystems to create a new, more

34 There are always unimportant outputs. Think about the number of moving parts in a car. They are not all important for steering!

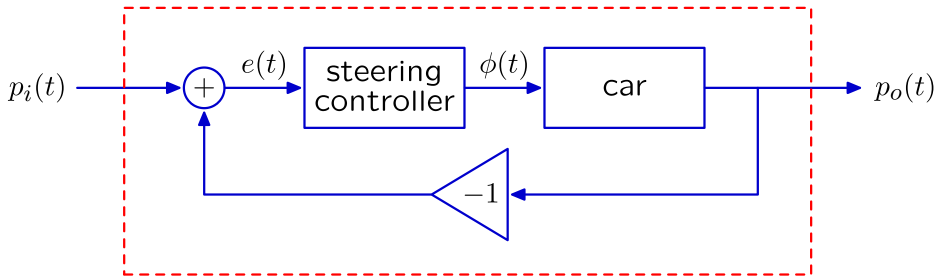


Figure 5.3 Modularity of systems

complex, system. A principal goal of this chapter is to develop methods of analysis for the sub-systems that can be *combined* to analyze the overall system.

5.1.2 Discrete-time signals and systems

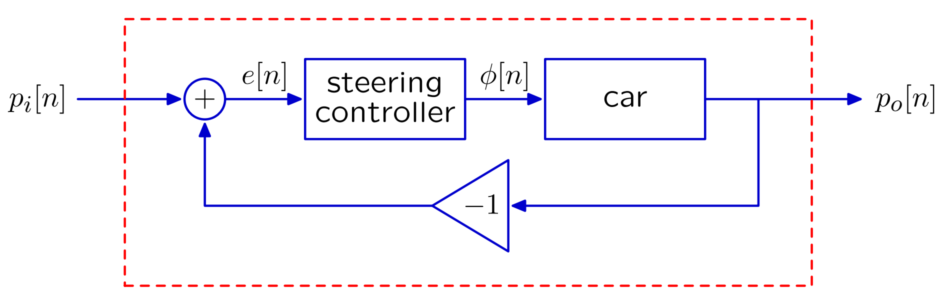
This chapter focuses on signals whose independent variables are discrete (e.g., take on only integer values). Some such signals are found in nature. For example, the primary structure of DNA is described by a *sequence* of base-pairs. However, we are primarily interested in discrete-time signals, not so much because they are found in nature, but because they are found in computers. Even though we focus on interactions with the real world, these interactions will primarily occur at discrete instants of time. For example, the difference between our desired position $p_i(t)$ and our actual position $p_o(t)$ is an error signal $e(t)$, which is a function of continuous time t. If the controller only observes this signal at regular sampling intervals T, then its input could be regarded as a sequence of values $x[n]$ that is indexed by the integer n. The relation between the discrete-time sequence $x[n]$ (note square brackets) and the continuous signal $x(t)$ (note round brackets) is given by

$$x[n] = x(nT) \quad ,$$

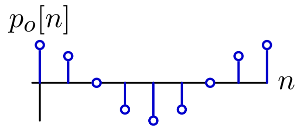
which we call the *sampling* relation. *Sampling* converts a signal of continuous domain to one of discrete domain.

While our primary focus will be on time signals, sampling works equally well in other domains. For example, images are typically represented as arrays of pixels accessed by integer-valued rows and columns, rather than as continuous brightness fields, indexed by real-valued spatial coordinates.

If the car-steering problem in **figure 5.1** were modeled in discrete time, we could describe the system with a diagram that is very similar to the continuous-time diagram in **figure 5.3**. However, only discrete time instants are considered



and the output position is now only defined at discrete times, as shown below.



5.1.3 Linear time-invariant systems

We already have a great way of specifying systems that operate on discrete-time signals: a state machine transduces a discrete-time input signal into a discrete-time output signal. State machines, as we have defined them, allow us to specify *any* discrete-time system whose output is computable from its history of previous inputs.

The representation of systems as state machines allows us to execute a machine on any input we'd like, in order to see what happens. Execution lets us examine the behavior of the system for any particular input for any particular finite amount of time, but it does not let us characterize any general properties of the system or its long-term behavior. Computer programs are such a powerful specification language that we cannot, in general, predict what a program will do (or even whether it will ever stop and return a value) without running it. In the rest of this chapter, we will concentrate on a small but powerful subclass of the whole class of state machines, called discrete-time *linear time-invariant (LTI) systems*, which will allow deeper forms of analysis.

In an LTI system:

- Inputs and outputs are real numbers;
- The state is some fixed number of previous inputs to the system as well as a fixed number of previous outputs of the system; and
- The output is a fixed, linear function of the current input and any of the elements of the state.

In general, each input could be a fixed-length vector of numbers, and each output could also be a fixed-length vector of numbers; we will restrict our attention to the case where the input is a single real number and the output is a single real number.

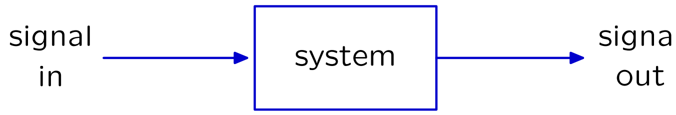
We are particularly interested in LTI systems because they can be analyzed mathematically, in a way that lets us characterize some properties of their output signal for *any* possible input signal. This is a much more powerful kind of insight than can be gained by trying a machine out with several different inputs.

Another important property of LTI systems is that they are compositional: the cascade, parallel, and feedback combinations (introduced in section 4.2) of LTI system are themselves LTI systems.

5.2 Discrete-time signals

In this section, we will work through the PCAP system for discrete time signals, by introducing a primitive and three methods of composition, and the ability to abstract by treating composite signals as if they themselves were primitive.

A *signal* is an infinite sequence of *sample* values at discrete time steps. We will use the following common notational conventions: A capital letter X stands for the whole input signal and $x[n]$ stands for the value of signal X at time step n. It is conventional, if there is a single system under discussion, to use X for the input signal to that system and Y for the output signal.



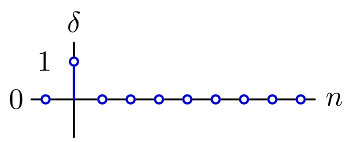
We will say that systems *transduce* input signals into output signals.

5.2.1 Unit sample signal

We will work with a single primitive, called the *unit sample signal*, Δ. It is defined on all positive and negative integer indices as follows[35]:

$$\delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} .$$

That is, it has value 1 at index $n = 0$ and 0 otherwise, as shown below:

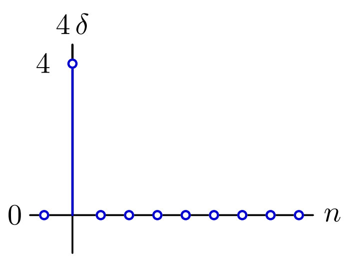


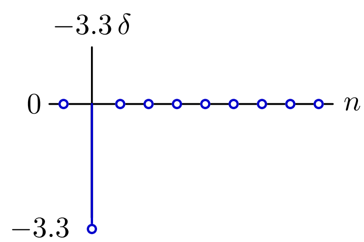
5.2.2 Signal combinators

Our first operation will be *scaling*, or *multiplication by a scalar*. A scalar is any real number. The result of multiplying any signal X by a scalar c is a signal, so that,

$$\text{if } \; Y = c \cdot X \; \text{ then } \; y[n] = c \cdot x[n] \; .$$

That is, the resulting signal has a value at every index n that is c times the value of the original signal at that location. Here are the signals 4Δ and -3.3Δ.



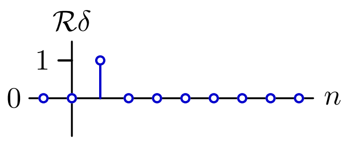


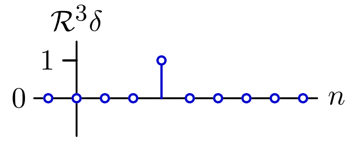
The next operation is the *delay* operation. The result of delaying a signal X is a new signal $\mathcal{R}X$ such that:

$$\text{if } \; Y = \mathcal{R}X \; \text{ then } \; y[n] = x[n-1] \; .$$

That is, the resulting signal has the same values as the original signal, but delayed by one step in time. You can also think of this, graphically, as shifting the signal one step to the $\mathcal{R}$ight. Here is the unit sample delayed by 1 and by 3 steps. We can describe the second signal as $\mathcal{R}\mathcal{R}\mathcal{R}\Delta$, or, using shorthand, as $\mathcal{R}^3\Delta$.

[35] Note that δ is the lowercase version of Δ, both of which are the Greek letter 'delta'.

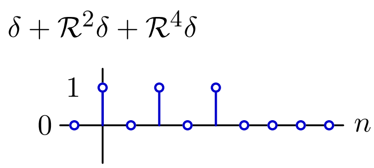


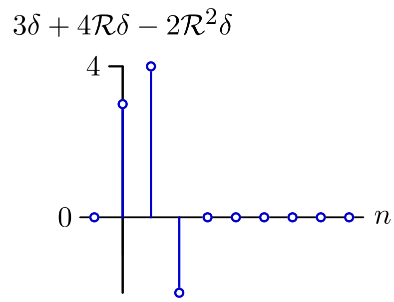


Finally, we can *add* two signals together. Addition of signals is accomplished component-wise, so that

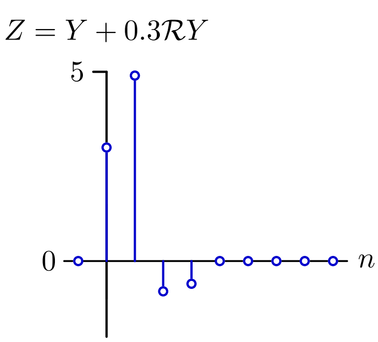
$$\text{if } Y = X_1 + X_2 \text{ then } y[n] = x_1[n] + x_2[n] \ .$$

That is, the value of the composite signal at step n is the sum of the values of the component signals. Here are some new signals constructed by summing, scaling, and delaying the unit sample.





Note that, because each of our operations returns a signal, we can use their results again as elements in new combinations, showing that our system has true compositionality. In addition, we can abstract, by naming signals. So, for example, we might define $Y = 3\Delta + 4\mathcal{R}\Delta - 2\mathcal{R}^2\Delta$, and then make a new signal $Z = Y + 0.3\mathcal{R}Y$, which would look like this:



Be sure you understand how the heights of the spikes are determined by the definition of Z.

Exercise 5.2. Draw a picture of samples -1 through 4 of $Y - \mathcal{R}Y$.

It is important to remember that, because signals are infinite objects, these combination operations are abstract mathematical operations. You could never somehow 'make' a new signal by calculating its value at every index. It is possible, however, to calculate the value at any particular index, as it is required.

Advancing

If we allow ourselves one more operation, that of 'advancing' the signal one step (just like delaying, but in the other direction, written $\mathcal{L}$ for left-shift), then *any* signal can be composed from the unit sample, using a (possibly infinite) number of these operations. We can demonstrate this claim by construction: to define a signal V with value v_n at index n, for any set of integer n, we simply set

$$V = v_0\Delta + \sum_{n=1}^{\infty} v_n \mathcal{R}^n \Delta + \sum_{n=1}^{\infty} v_{-n} \mathcal{L}^n \Delta \;\;,$$

where $\mathcal{R}^n$ and $\mathcal{L}^n$ are shorthand for applying $\mathcal{R}$ and $\mathcal{L}$, respectively, n times.

If n represents time, then physical systems are always *causal*: inputs that arrive after time n_0 cannot affect the output before time n_0. Such systems cannot advance signals: they can be written without $\mathcal{L}$.

5.2.3 Algebraic properties of operations on signals

Adding and scaling satisfy the familiar algebraic properties of addition and multiplication: addition is commutative and associative, scaling is commutative (in the sense that it doesn't matter whether we pre- or post-multiply) and scaling distributes over addition:

$$c \cdot (X_1 + X_2) = c \cdot X_1 + c \cdot X_2 \;\;,$$

which can be verified by defining $Y = c \cdot (X_1 + X_2)$ and $Z = c \cdot X_1 + c \cdot X_2$ and checking that $y[n] = z[n]$ for all n:

$$\begin{aligned} y[n] &= z[n] \\ c \cdot (x_1[n] + x_2[n]) &= (c \cdot x_1[n]) + (c \cdot x_2[n]) \end{aligned}$$

which clearly holds based on algebraic properties of arithmetic on real numbers.

In addition, $\mathcal{R}$ distributes over addition and scaling, so that:

$$\begin{aligned} \mathcal{R}(X_1 + X_2) &= \mathcal{R}X_1 + \mathcal{R}X_2 \\ \mathcal{R}(c \cdot X) &= c \cdot \mathcal{R}X \;\;. \end{aligned}$$

> *Exercise 5.3.* Verify that $\mathcal{R}$ distributes over addition and multiplication by checking that the appropriate relations hold at some arbitrary step n.

These algebraic relationships mean that we can take any finite expression involving Δ, $\mathcal{R}$, $+$ and $\cdot$ and convert it into the form

$$(a_o + a_1\mathcal{R}^1 + a_2\mathcal{R}^2 + \ldots + a_N\mathcal{R}^N)\Delta \;\;.$$

That is, we can express the entire signal as a *polynomial in* $\mathcal{R}$, applied to the unit sample.

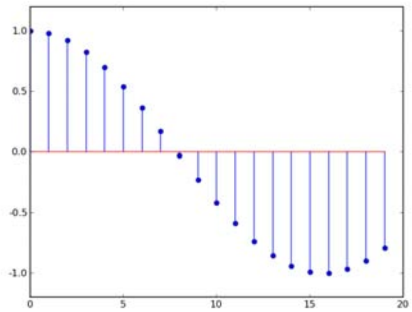
In our previous example, it means that we can rewrite $3\Delta + 4\mathcal{R}\Delta - 2\mathcal{R}^2\Delta$ as $(3 + 4\mathcal{R} - 2\mathcal{R}^2)\Delta$.

5.2.4 Sinusoidal primitives

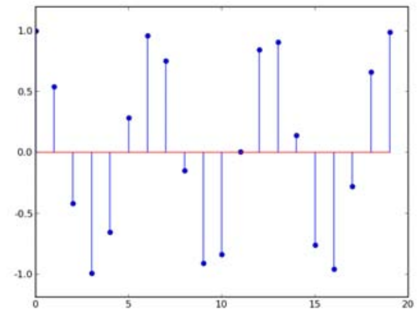
We just saw how to construct complicated signals by summing unit sample signals that are appropriately scaled and shifted. We could similarly start with a family of discretely-sampled sinusoids as our primitives, where

$$x[n] = \cos(\Omega n) \quad .$$

Here are plots of two primitives in this family:



$\cos(0.2n)$



$\cos(1.0n)$

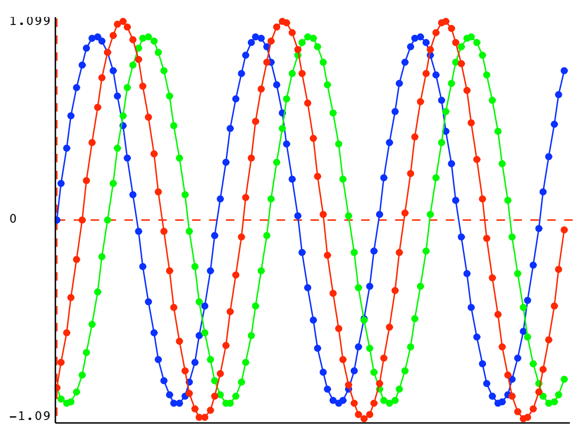
The second plot may seem confusing, but it is just a sparsely sampled sinusoid. Note that signals constructed from even a single sinusoid have non-zero values defined at an infinity of steps; this is in contrast to signals constructed from a finite sum of scaled and shifted unit samples.

Exercise 5.4. If $x[n] = \cos(0.2n)$, what would be the values of $\mathcal{R}X$ at steps -3 and 5?

Here are two sinusoids and their sum, made as follows:

$$s_1[n] = \cos(0.2n - \pi/2)$$
$$S_2 = \mathcal{R}^{10}S_1$$
$$S_3 = S_1 + S_2$$

The blue line is the S_1, the green line is the same signal, delayed by 10, which is S_2, and the red line is their sum.



5.3 Feedforward systems

We will start by looking at a subclass of discrete-time LTI system, which are exactly those that can be described as performing some combination of scaling, delay, and addition operations on the input signal. We will develop several ways of representing such systems, and see how to combine them to get more complex systems in this same class.

5.3.1 Representing systems

We can represent systems using operator equations, difference equations, block diagrams, and Python state machines. Each makes some things clearer and some operations easier. It is important to understand how to convert between the different representations.

Operator equation

An operator equation is a description of how signals are related to one another, using the operations of scaling, delay, and addition on whole signals.

Consider a system that has an input signal X, and whose output signal is $X - \mathcal{R}X$. We can describe that system using the operator equation

$$Y = X - \mathcal{R}X \ .$$

Using the algebraic properties of operators on signals described in section 5.2.3, we can rewrite this as

$$Y = (1 - \mathcal{R})X \ ,$$

which clearly expresses a relationship between input signal X and output signal Y, whatever X may be.

Feedforward systems can always be described using an operator equation of the form

$$Y = \Phi X \ ,$$

where Φ is a polynomial in $\mathcal{R}$.

Difference Equation

An alternative representation of the relationship between signals is a *difference equation*. A difference equation describes a relationship that holds among samples (values at particular times) of signals. We use an index n in the difference equation to refer to a particular time index, but the specification of the corresponding system is that the difference equation hold for *all* values of n.

The operator equation

$$Y = X - \mathcal{R}X \ .$$

can be expressed as this equivalent difference equation:

$$y[n] = x[n] - x[n-1] \ .$$

The operation of delaying a signal can be seen here as referring to a sample of that signal at time step $n - 1$.

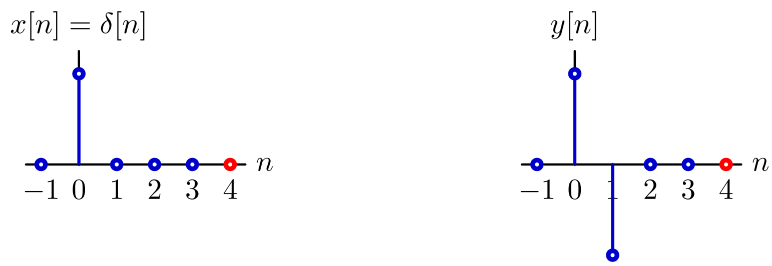
Difference equations are convenient for step-by-step analysis, letting us compute the value of an output signal at any time step, given the values of the input signal.

So, if the input signal X is the unit sample signal,

$$x[n] = \delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad .$$

then using a difference equation, we can compute individual values of the output signal Y:

$$\begin{aligned} y[n] &= x[n] - x[n-1] \\ y[0] &= x[0] - x[-1] &&= 1 - 0 = 1 \\ y[1] &= x[1] - x[0] &&= 0 - 1 = -1 \\ y[2] &= x[2] - x[1] &&= 0 - 0 = 0 \\ y[3] &= x[3] - x[2] &&= 0 - 0 = 0 \\ &\ldots \end{aligned}$$



Block diagrams

Another way of describing a system is by drawing a *block diagram*, which is made up of components, connected by lines with arrows on them. The lines represent signals; all lines that are connected to one another (not going through a round, triangular, or circular component) represent the same signal.

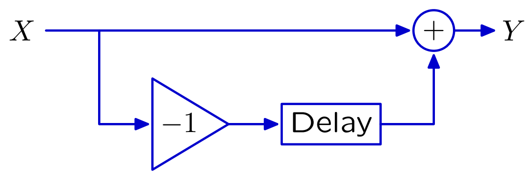
The components represent systems. There are three primitive components corresponding to our operations on signals:

- *Delay* components are drawn as rectangles, labeled Delay, with two lines connected to them, one with an arrow coming in and one going out. If X is the signal on the line coming into the delay, then the signal coming out is $\mathcal{R}X$.
- *Scale* (or *gain*) components are drawn as triangles, labeled with a positive or negative number c, with two lines connected to them, one with an arrow coming in and one going out. If X is the signal on the line coming into the gain component, then the signal coming out is $c \cdot X$.
- *Adder* components are drawn as circles, labeled with +, three lines connected to them, two with arrows coming in and one going out. If X_1 and X_2 are the signals on the lines point into the adder, then the signal coming out is $X_1 + X_2$.

The system

$$Y = X - \mathcal{R}X$$

can be represented with this block diagram.



State machines

Of course, since feedforward LTI systems are a type of state machine, we can make an equivalent definition using our Python state-machine specification language. So, our system

$$Y = X - \mathcal{R}X$$

can be specified in Python as a state machine by:

```
class Diff(sm.SM):
    def __init__(self, previousInput):
        self.startState = previousInput
    def getNextValues(self, state, inp):
        return (inp, inp-state)
```

Here, the `state` is the value of the previous input. One important thing to notice is that, since we have to be able to run a state machine and generate outputs, it has to start with a value for its internal state, which is the input signal's value at time -1. If we were to run:

```
Diff(0).transduce([1, 0, 0, 0])
```

we would get the result

```
[1, -1, 0, 0]
```

This same state machine can also be expressed as a combination of primitive state machines (as defined in sections **4.1.2** and **4.2**).

```
diff = sm.ParallelAdd(sm.Wire(),
                      sm.Cascade(sm.Gain(-1), sm.R(0)))
```

Note that `sm.R` is another name for `sm.Delay` and that the desired initial output value for the system appears as the initialization argument to the `sm.R` machine.

5.3.2 Combinations of systems

To combine LTI systems, we will use the same cascade and parallel-add operations as we had for state machines.

Cascade multiplication

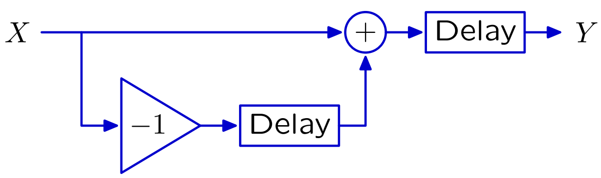
When we make a cascade combination of two systems, we let the output of one system be the input of another. So, if the system M_1 has operator equation $Y = \Phi_1 X$ and system M_2 has operator equation $Z = \Phi_2 W$, and then we compose M_1 and M_2 in cascade by setting $Y = W$, then we have a new system, with input signal X, output signal Z, and operator equation $Z = (\Phi_2 \cdot \Phi_1)X$.

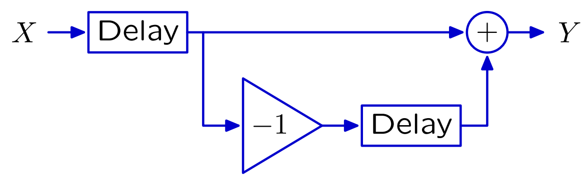
The product of polynomials is another polynomial, so $\Phi_2 \cdot \Phi_1$ is a polynomial in $\mathcal{R}$. Furthermore, because polynomial multiplication is commutative, cascade combination is commutative as well (as long as the systems are *at rest*, which means that their initial states are 0).

So, for example,

$$\mathcal{R}(1 - \mathcal{R})\,X = (1 - \mathcal{R})\mathcal{R}X$$

and these two corresponding block diagrams are equivalent (the algebraic equivalence justifies the diagram equivalence):





Cascade combination, because it results in multiplication, is also associative, which means that any grouping of cascade operations on systems has the same result.

Exercise 5.5. Remembering that the condition on commutativity of cascading is that the systems start at rest, explain why machines `m3` and `m4` do not generate the same output sequence in response to the unit sample signal as input.

```
m1 = sm.ParallelAdd(sm.Wire(), sm.Cascade(sm.Gain(-1), sm.R(2)))
m2 = sm.R(3)
m3 = sm.Cascade(m1, m2)
m4 = sm.Cascade(m2, m1)
```

Parallel addition

When we make a parallel addition combination of two systems, the output signal is the sum of the output signals that would have resulted from the individual systems. So, if the system M_1 has system function $Y = \Phi_1 X$ and system M_2 has system function $Z = \Phi_2 X$, and then we compose M_1 and M_2 with parallel addition by setting output $W = Y + Z$, then we have a new system, with input signal X, output signal W, and operator equation $W = (\Phi_1 + \Phi_2)X$.

Because addition of polynomials is associative and commutative, then so is parallel addition of feed-forward linear systems.

Combining cascade and parallel operations

Finally, the distributive law applies for cascade and parallel combination, *for systems at rest*, in the same way that it applies for multiplication and addition of polynomials, so that if we have three systems, with operator equations:

$$Y = \Phi_1 X$$
$$U = \Phi_2 V$$
$$W = \Phi_3 Z \quad ,$$

and we form a cascade combination of the sum of the first two, with the third, then we have a system describable as:

$$B = (\Phi_3 \cdot (\Phi_1 + \Phi_2))A \quad .$$

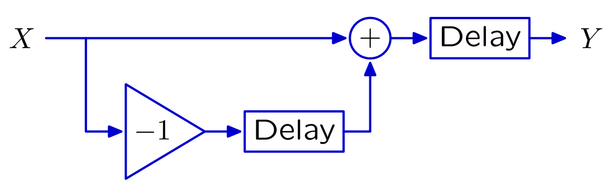
We can rewrite this, using the distributive law, as:

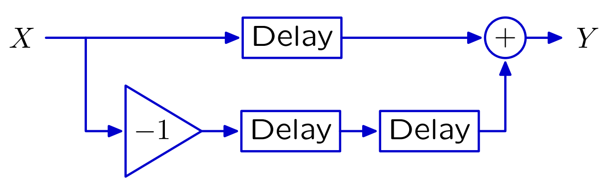
$$B = ((\Phi_3 \cdot \Phi_1) + (\Phi_3 \cdot \Phi_2))A \quad .$$

So, for example,

$$\mathcal{R}(1 - \mathcal{R}) = \mathcal{R} - \mathcal{R}^2 \quad ,$$

and these two corresponding block diagrams are equivalent:





Exercise 5.6. The first machine in the diagram above can be described, for certain initial output values as:

```
m1 = sm.Cascade(sm.ParallelAdd(sm.Wire(),
                               sm.Cascade(sm.Gain(-1), Delay(2))),
                sm.Delay(3))
```

The second machine can be described as:

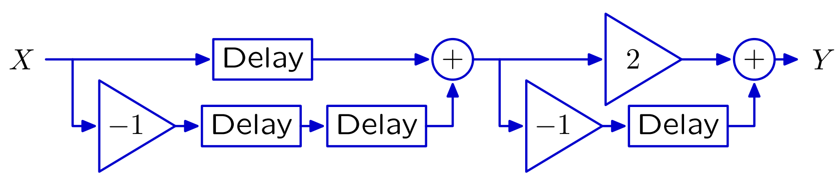
```
m2 = sm.ParallelAdd(sm.Delay(d1),
                    sm.Cascade(sm.Gain(-1),
                               sm.Cascade(sm.Delay(d2), sm.Delay(d3))))
```

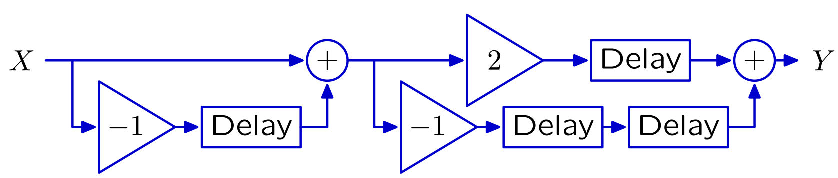
Provide values of `d1`, `d2`, and `d3` that will cause `m2` to generate the same output sequence as `m1` in response to the unit sample signal as input.

Here is another example of two equivalent operator equations

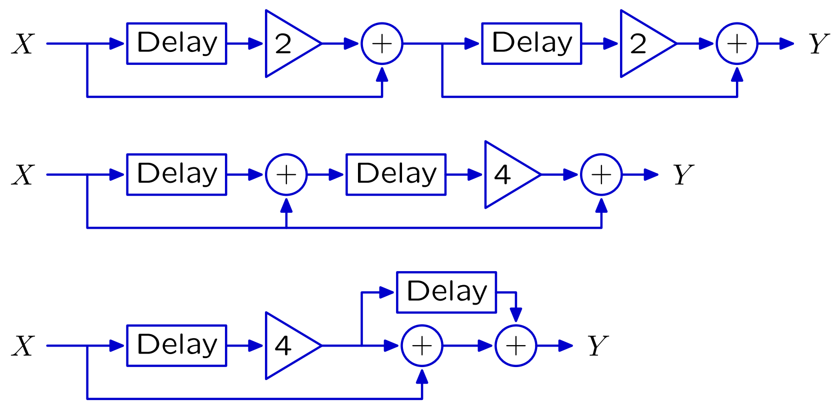
$$(\mathcal{R} - \mathcal{R}^2)(2 - \mathcal{R})X = (1 - \mathcal{R})(2\mathcal{R} - \mathcal{R}^2)X$$

and these two corresponding block diagrams are equivalent if the systems start at rest:





Exercise 5.7. Convince yourself that all of these systems are equivalent. One strategy is to convert them all to operator equation representation.

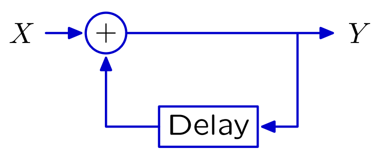


5.4 Feedback Systems

So far, all of our example systems have been *feedforward*: the dependencies have all flowed from the input through to the output, with no dependence of an output on previous output values. In this section, we will extend our representations and analysis to handle the general class of LTI systems in which the output can depend on any finite number of previous input or output values.

5.4.1 Accumulator example

Consider this block diagram, of an *accumulator*:



It's reasonably straightforward to look at this block diagram and see that the associated difference equation is

$$y[n] = x[n] + y[n-1] \ ,$$

because the output on any give step is the sum of the the input on that step and the output from the previous step.

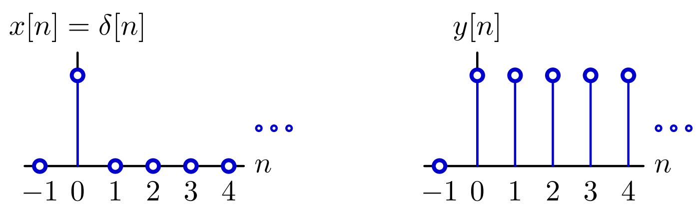
Let's use the difference equation to understand what the output of this system is when the input is the unit sample signal. To compute the output at step n, we need to evaluate

$$y[n] = x[n] + y[n-1] \ .$$

We immediately run up against a question: what is the value of $y[n-1]$? The answer clearly has a profound effect on the output of the system. In our treatment of feedback systems, we will generally assume that they start 'at rest', which means that all values of the inputs and outputs at steps less than 0 are 0. That assumption lets us fill in the following table:

$$
\begin{aligned}
y[n] &= x[n] + y[n-1] \\
y[0] &= x[0] + y[-1] && = 1 + 0 = 1 \\
y[1] &= x[1] + y[0] && = 0 + 1 = 1 \\
y[2] &= x[2] + y[1] && = 0 + 1 = 1 \\
&\ldots
\end{aligned}
$$

Here are plots of the input signal X and the output signal Y:



This result may be somewhat surprising! In feedforward systems, we saw that the output was always a finite sum of scaled and delayed versions of the input signal; so that if the input signal was transient (had a finite number of non-zero samples) then the output signal would be transient as well. But, in this feedback system, we have a transient input with a persistent (infinitely many non-zero samples) output.

We can also look at the operator equation for this system. Again, reading it off of the block diagram, it seems like it should be

$$Y = X + \mathcal{R}Y \quad .$$

It's a well-formed equation, but it isn't immediately clear how to use it to determine Y. Using what we already know about operator algebra, we can rewrite it as:

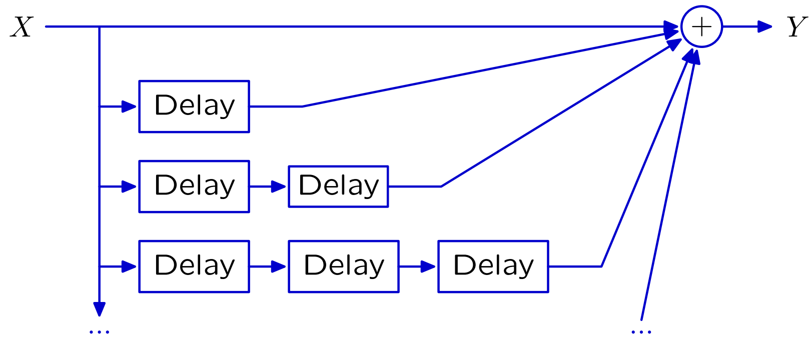
$$Y(1 - \mathcal{R}) = X \quad ,$$

which defines Y to be the signal such that the difference between Y and $\mathcal{R}$Y is X. But how can we find that Y?

We will now show that we can think of the accumulator system as being equivalent to another system, in which the output is the sum of infinitely many feedforward paths, each of which delays the input by a different, fixed value. This system has an operator equation of the form

$$Y = (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\, X$$

and can be represented with a block diagram of the form:



These systems are equivalent in the sense that if each is initially at rest, they will produce identical outputs from the same input. We can see this by taking the original definition and repeatedly substituting in the definition of Y in for its occurrence on the right hand side:

$$Y = X + \mathcal{R}Y$$
$$Y = X + \mathcal{R}(X + \mathcal{R}Y)$$
$$Y = X + \mathcal{R}(X + \mathcal{R}(X + \mathcal{R}Y))$$
$$Y = X + \mathcal{R}(X + \mathcal{R}(X + \mathcal{R}(X + \mathcal{R}Y)))$$
$$Y = (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \ldots)X$$

Now, we can informally derive a 'definition' of the reciprocal of $1 - \mathcal{R}$ (the mathematical details underlying this are subtle and not presented here),

$$\frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots \quad .$$

In the following it will help to remind ourselves of the derivation of the formula for the sum of an infinte geometric series:

$$S = 1 + x + x^2 + x^3 + \cdots$$
$$Sx = \quad x + x^2 + x^3 + x^4 \cdots$$

Subtracting the second equation from the first we get

$$S(1 - x) = 1$$

And so, provided $|x| < 1$,

$$\frac{1}{1-x} = \quad 1 + x + x^2 + x^3 + \cdots$$

Similarly, we can consider the system O, where

$$O = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots$$
$$O\mathcal{R} = \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots$$

So

$$O(1 - \mathcal{R}) = 1$$

And so,

$$\frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots$$

Exercise 5.8. Check this derivation by showing that

$$(1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)(1 - \mathcal{R}) = 1$$

So, we can rewrite the operator equation for the accumulator as

$$Y = \frac{1}{1-\mathcal{R}}X \quad .$$

We don't have an intuitive way to interpret this relationship between X and Y yet, but we will spend most of the rest of this chapter on developing intuition and analysis for systems with feedback.

5.4.2 General form of LTI systems

We can now treat the general case of LTI systems, including feedback. In general, LTI systems can be described by difference equations of the form:

$$y[n] = c_0\, y[n-1] + c_1\, y[n-2] + \ldots + c_{k-1}\, y[n-k] \\ + d_0\, x[n] + d_1\, x[n-1] + \ldots + d_j\, x[n-j] \quad .$$

The state of this system consists of the k previous output values and j previous input values. The output $y[n]$ is a linear combination of the k previous output values, $y[n-1], \ldots, y[n-k]$, j previous input values, $x[n-1], \ldots, x[n-j]$, and the current input, $x[n]$.

This class of state machines can be represented, in generality, in Python, using the `LTISM` class. The state is a tuple, containing a list of the j previous input values and a list of the k previous output values.

```
class LTISM (sm.SM):
    def __init__(self, dCoeffs, cCoeffs):
        j = len(dCoeffs) - 1
        k = len(cCoeffs)

        self.cCoeffs = cCoeffs
        self.dCoeffs = dCoeffs
        self.startState = ([0.0]*j, [0.0]*k)
```

```
def getNextValues(self, state, input):
    (inputs, outputs) = state
    inputs = [input] + inputs

    currentOutput = util.dotProd(outputs, self.cCoeffs) + \
                    util.dotProd(inputs, self.dCoeffs)

    return ((inputs[:-1], ([currentOutput] + outputs)[:-1]),
            currentOutput)
```

The `util.dotProd` method takes two equal-length lists of numbers and returns the sum of their elementwise products (the dot-product of the two vectors). To keep this code easy to read, we do not handle correctly the case where `dCoeffs` is empty, though it is handled properly in our library implementation.

5.4.3 System functions

Now, we are going to engage in a shift of perspective. We started by defining a new signal Y in terms of an old signal X, much as we might, in algebra, define $y = x + 6$. Sometimes, however, we want to speak of the relationship between x and y in the general case, without a specific x or y in mind. We do this by defining a function f: $f(x) = x + 6$. We can do the same thing with LTI systems, by defining *system functions*.

If we take the general form of an LTI system given in the previous section and write it as an operator equation, we have

$$\begin{aligned} Y &= c_0\, \mathcal{R}Y + c_1\, \mathcal{R}^2 Y + \ldots + c_{k-1}\, \mathcal{R}^k Y + d_0\, X + d_1\, \mathcal{R}X + \ldots + d_j\, \mathcal{R}^j X \\ &= (c_0\, \mathcal{R} + c_1\, \mathcal{R}^2 + \ldots + c_{k-1}\, \mathcal{R}^k)\, Y + (d_0 + d_1\, \mathcal{R} + \ldots + d_j\, \mathcal{R}^j)\, X \ . \end{aligned}$$

We can rewrite this as

$$(1 - c_0\, \mathcal{R} - c_1\, \mathcal{R}^2 - \ldots - c_{k-1}\, \mathcal{R}^k)\, Y = (d_0 + d_1\, \mathcal{R} + \ldots + d_j\, \mathcal{R}^j)\, X \ ,$$

so

$$\frac{Y}{X} = \frac{d_0 + d_1\mathcal{R} + d_2\mathcal{R}^2 + d_3\mathcal{R}^3 + \cdots}{1 - c_0\mathcal{R} - c_1\mathcal{R}^2 - c_2\mathcal{R}^3 - \cdots} \ ,$$

which has the form

$$\frac{Y}{X} = \frac{N(\mathcal{R})}{D(\mathcal{R})} \ ,$$

where $N(\mathcal{R})$, the numerator, is a polynomial in $\mathcal{R}$, and $D(\mathcal{R})$, the denominator, is also a polynomial in $\mathcal{R}$. We will refer to Y/X as the *system function*: it characterizes the operation of a system, independent of the particular input and output signals involved.

The system function is most typically written in the form

$$\frac{Y}{X} = \frac{b_0 + b_1\mathcal{R} + b_2\mathcal{R}^2 + b_3\mathcal{R}^3 + \cdots}{a_0 + a_1\mathcal{R} + a_2\mathcal{R}^2 + a_3\mathcal{R}^3 + \cdots} \ ,$$

where $c_i = -a_{i+1}/a_0$ and $d_i = b_i/a_0$. It can be completely characterized by the coefficients of the denominator polynomial, a_i, and the coefficients of the numerator polynomial, b_i. It is always possible to rewrite this in a form in which $a_0 = 1$.

Feedforward systems have no dependence on previous values of Y, so they have $D(\mathcal{R}) = 1$. Feedback systems have persistent behavior, which is determined by $D(\mathcal{R})$. We will study this dependence in detail in section 5.5.

5.4.4 Primitive systems

Just as we had a PCAP system for signals, we have one for LTI system, in terms of system functions, as well. We can specify system functions for each of our system primitives.

A *gain* element is governed by operator equation $Y = kX$, for constant k, so its system function is

$$H = \frac{Y}{X} = k \enspace .$$

A *delay* element is governed by operator equation $Y = \mathcal{R}X$, so its system function is

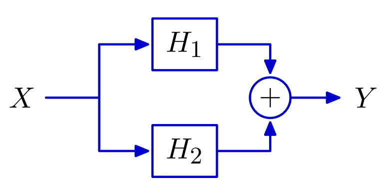
$$H = \frac{Y}{X} = \mathcal{R} \enspace .$$

5.4.5 Combining system functions

We have three basic composition operations: sum, cascade, and feedback. This PCAP system, as our previous ones have been, is *compositional*, in the sense that whenever we make a new system function out of existing ones, it is a system function in its own right, which can be an element in further compositions.

Addition

The system function of the *sum* of two systems is the sum of their system functions. So, given two systems with system functions H_1 and H_2, connected like this:



and letting

$$Y_1 = H_1X \quad \text{and} \quad Y_2 = H_2X \enspace ,$$

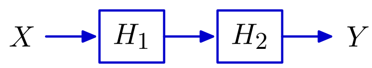
we have

$$\begin{aligned} Y &= Y_1 + Y_2 \\ &= H_1X + H_2X \\ &= (H_1 + H_2)X \\ &= HX \enspace , \end{aligned}$$

where $H = H_1 + H_2$.

Cascade

The system function of the *cascade* of two systems is the product of their system functions. So, given two systems with system functions H_1 and H_2, connected like this:



and letting

$$W = H_1 X \quad \text{and} \quad Y = H_2 W \ ,$$

we have

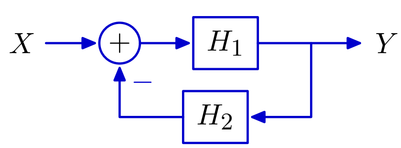
$$\begin{aligned} Y &= H_2 W \\ &= H_2 H_1 X \\ &= HX \ , \end{aligned}$$

where $H = H_2 H_1$. And note that, as was the case with purely feedforward systems, cascade combination is still commutative, so it doesn't matter whether H_1 or H_2 comes first in the cascade. This surprising fact holds because we are only considering LTI systems *starting at rest*; for more general classes of systems, such as the general class of state machines we have worked with before, the ordering of a cascade *does* matter.

Feedback

There are several ways of connecting systems in feedback. Here we study a particular case of *negative feedback* combination, which results in a classical formula called *Black's formula*.

Consider two systems connected like this



and *pay careful attention to the negative sign on the feedback input to the addition*. It is really just shorthand; the negative sign could be replaced with a gain component with value −1. This negative feedback arrangement is frequently used to model a case in which X is a desired value for some signal and Y is its actual value; thus the input to H_1 is the difference between the desired an actual values, often called an *error signal*. We can simply write down the operator equation governing this system and use standard algebraic operations to determine the system function:

$$\begin{aligned}
Y &= H_1(X - H_2Y) \\
Y + H_1H_2Y &= H_1X \\
Y(1 + H_1H_2) &= H_1X \\
Y &= \frac{H_1}{1 + H_1H_2}X \\
Y &= HX \quad ,
\end{aligned}$$

where

$$H = \frac{H_1}{1 + H_1H_2} \quad .$$

Armed with this set of primitives and composition methods, we can specify a large class of machines. Ultimately, we will want to construct systems with multiple inputs and outputs; such systems are specified with a matrix of basic system functions, describing how each output depends on each input.

5.5 Predicting system behavior

We have seen how to construct complex discrete-time LTI systems; in this section we will see how we can use properties of the system function to predict how the system will behave, in the long term, and for any input. We will start by analyzing simple systems and then move to more complex ones.

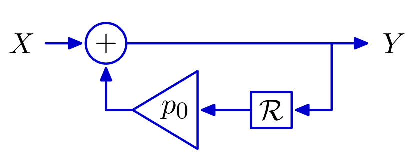
We can provide a general characterization of the long-term behavior of the output, as increasing or decreasing, with constant or alternating sign, for any finite input to the system. We will begin by studying the *unit-sample* response of systems, and then generalize to more general input signals; similarly, we will begin by studying simple systems and generalize to more complex ones.

5.5.1 First-order systems

Systems that only have forward connections can only have a finite response; that means that if we put in a unit sample (or other signal with only a finite number of non-zero samples) then the output signal will only have a finite number of non-zero samples.

Systems with feedback have a surprisingly different character. Finite inputs can result in *persistent response*; that is, in output signals with infinitely many non-zero samples. Furthermore, the qualitative long-term behavior of this output is generally independent of the particular input given to the system, for any finite input. In this section, we will consider the class of *first-order* systems, in which the denominator of the system function is a first-order polynomial (that is, it only involves $\mathcal{R}$, but not $\mathcal{R}^2$ or other higher powers of $\mathcal{R}$.)

Let's consider this very simple system



for which we can write an operator equation

$$\begin{aligned} Y &= X + p_0 \mathcal{R} Y \\ (1 - p_0\mathcal{R})Y &= X \\ Y &= \frac{X}{1 - p_0\mathcal{R}} \end{aligned}$$

and derive a system function

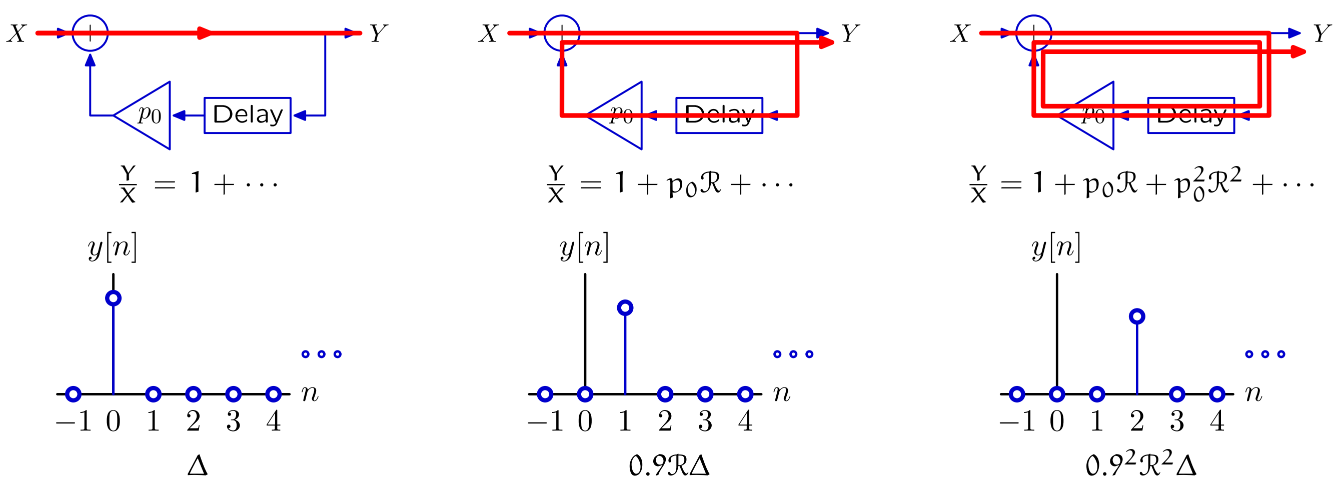
$$H = \frac{Y}{X} = \frac{1}{1 - p_0\mathcal{R}} \quad .$$

Recall the infinite series representation of this system function (derived in section 5.4.1):

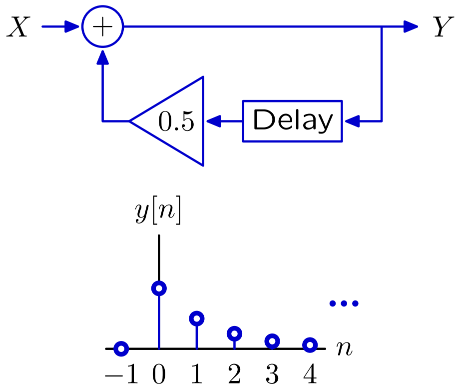
$$\frac{1}{1 - p_0\mathcal{R}} = 1 + p_0\mathcal{R} + p_0^2\mathcal{R}^2 + p_0^3\mathcal{R}^3 + p_0^4\mathcal{R}^4 + \cdots \quad .$$

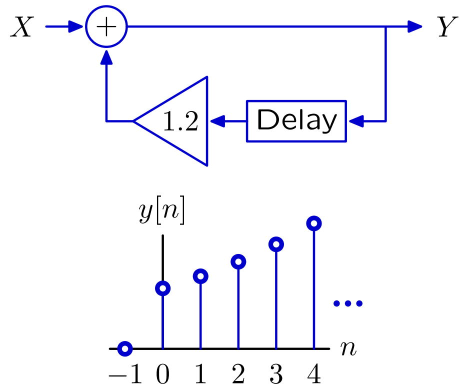
We can make intuitive sense of this by considering how the signal flows through the system. On each step, the output of the system is being fed back into the input. Consider the simple case where the input is the unit sample ($X = \Delta$). Then, after step 0, when the input is 1, there is no further input, and the system continues to respond.

In this table, we see that the whole output signal is a sum of scaled and delayed copies of the input signal; the bottom row of figures shows the first three terms in the infinite sum of signals, for the case where $p_0 = 0.9$.



If traversing the cycle decreases or increases the magnitude of the signal, then the sample values will decay or grow, respectively, as time increases.

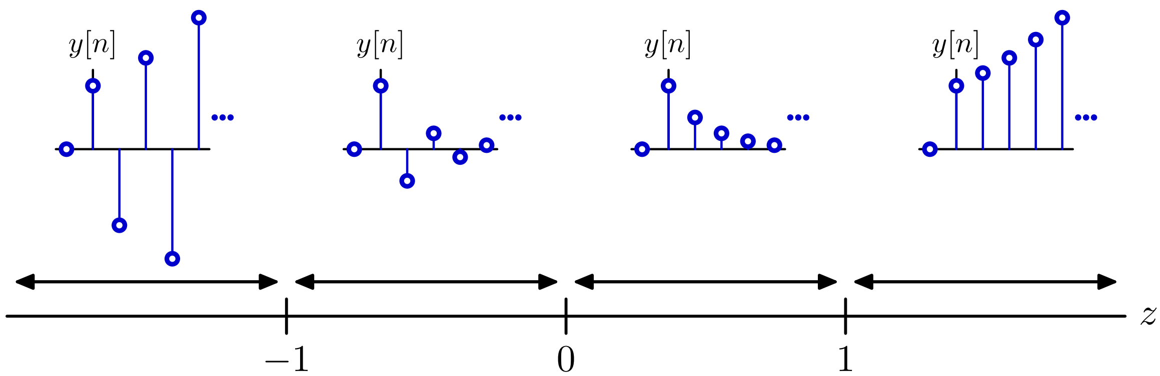




For the first system, the unit sample response is $y[n] = (0.5)^n$; for the second, it's $y[n] = (1.2)^n$.

These system responses can be characterized by a single number, called the *pole*, which is the base of the geometric sequence. The value of the pole, p_0, determines the nature and rate of growth.

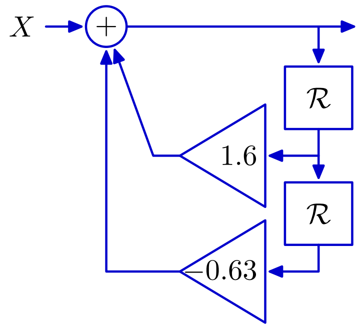
- If $p_0 < -1$, the magnitude increases to infinity and the sign alternates.
- If $-1 < p_0 < 0$, the magnitude decreases and the sign alternates.
- If $0 < p_0 < 1$, the magnitude decreases monotonically.
- If $p_0 > 1$, the magnitude increases monotonically to infinity.



5.5.2 Second-order systems

We will call these persistent long-term behaviors of a signal (and, hence, of the system that generates such signals) *modes*. For a fixed p_0, the first-order system only exhibited one mode (but different values of p_0 resulted in very different modes). As we build more complex systems, they will have multiple modes, which manifest as more complex behavior. Second-order systems are characterized by a system function whose denominator polynomial is second order; they will generally exhibit two modes.

Consider this system



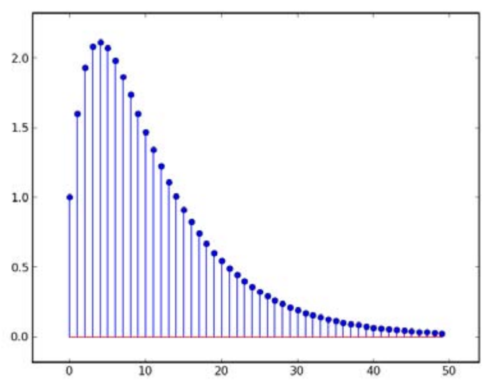
We can describe it with the operator equation

$$Y = 1.6\mathcal{R}Y - 0.63\mathcal{R}^2Y + X \ ,$$

so the system function is

$$H = \frac{1}{1 - 1.6\mathcal{R} + 0.63\mathcal{R}^2} \ .$$

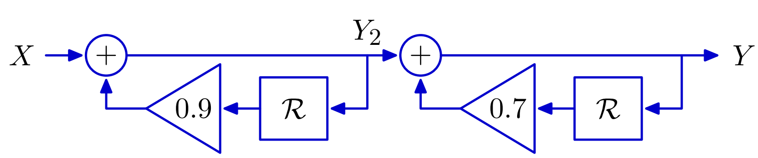
Here is its response to a unit sample signal:

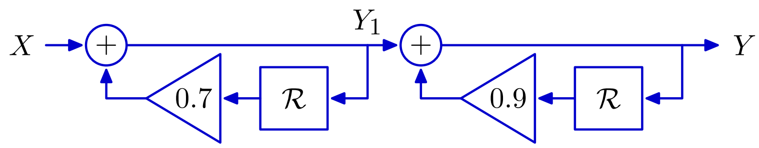


We can try to understand its behavior by decomposing it in different ways. First, let's see if we can see it as a cascade of two systems. To do so, we need to find H_1 and H_2 such that $H_1H_2 = H$. We can do that by factoring H to get

$$H_1 = \frac{1}{1 - 0.7\mathcal{R}} \quad \text{and} \quad H_2 = \frac{1}{1 - 0.9\mathcal{R}} \ .$$

So, we have two equivalent version of this system, describable as cascades of two systems, one with $p_0 = 0.9$ and one with $p_0 = 0.7$:





This decomposition is interesting, but it does not yet let us understand the behavior of the system as the combination of the behaviors of two subsystems.

5.5.2.1 Additive decomposition

Another way to try to decompose the system is as the *sum* of two simpler systems. In this case, we seek H_1 and H_2 such that $H_1 + H_2 = H$. We can do a partial fraction decomposition (don't worry if you don't remember the process for doing this...we won't need to solve problems like this in detail). We start by factoring, as above, and then figure out how to decompose into additive terms:

$$\begin{aligned} H &= \frac{1}{1 - 1.6\mathcal{R} + 0.63\mathcal{R}^2} \\ &= \frac{1}{(1 - 0.9\mathcal{R})(1 - 0.7\mathcal{R})} \\ &= \frac{A}{1 - 0.9\mathcal{R}} + \frac{B}{1 - 0.7\mathcal{R}} \\ &= H_1 + H_2 \ . \end{aligned}$$

To find values for A and B, we start with

$$\frac{1}{(1 - 0.9\mathcal{R})(1 - 0.7\mathcal{R})} = \frac{A}{1 - 0.9\mathcal{R}} + \frac{B}{1 - 0.7\mathcal{R}} \ ,$$

multiply through by $1 - 1.6\mathcal{R} + 0.63\mathcal{R}^2$ to get

$$1 = A(1 - 0.7\mathcal{R}) + B(1 - 0.9\mathcal{R}) \ ,$$

and collect like terms:

$$1 = (A + B) - (0.7A + 0.9B)\mathcal{R} \ .$$

Equating the terms that involve equal powers of $\mathcal{R}$ (including constants as terms that involve $\mathcal{R}^0$), we have:

$$\begin{aligned} 1 &= A + B \\ 0 &= 0.7A + 0.9B \ . \end{aligned}$$

Solving, we find $A = 4.5$ and $B = -3.5$, so

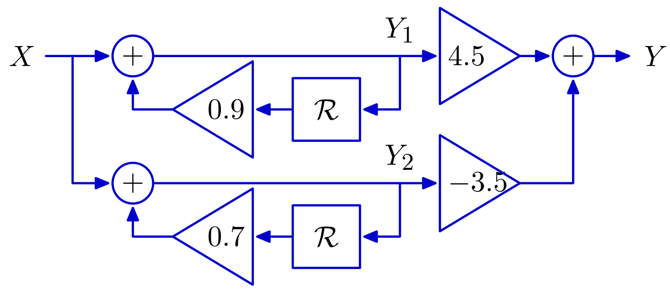
$$\frac{Y}{X} = \frac{4.5}{1 - 0.9\mathcal{R}} + \frac{-3.5}{1 - 0.7\mathcal{R}} \ ,$$

where

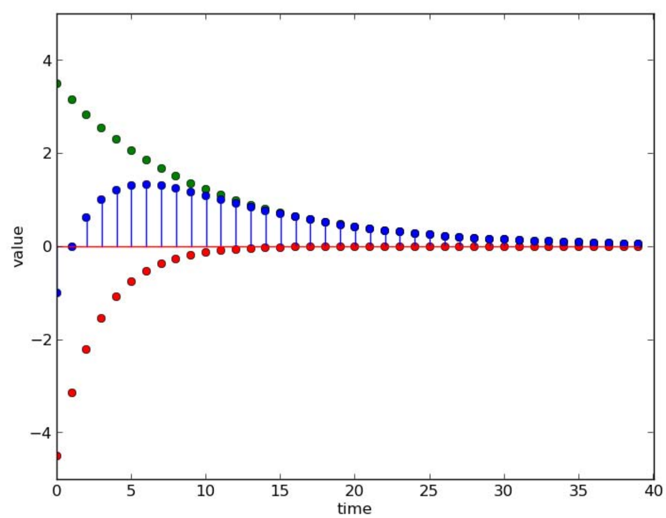
$$H_1 = \frac{4.5}{1 - 0.9\mathcal{R}} \quad \text{and} \quad H_2 = \frac{-3.5}{1 - 0.7\mathcal{R}} .$$

Exercise 5.9. Verify that $H_1 + H_2 = H$.

Here is (yet another) equivalent block diagram for this system, highlighting its decomposition into a sum:



We can understand this by looking at the responses of H_1 and of H_2 to the unit sample, and then summing them to recover the response of H to the unit sample. In the next figure, the blue stem plot is the overall signal, which is the sum of the green and red signals, which correspond to the top and bottom parts of the diagram, respectively.

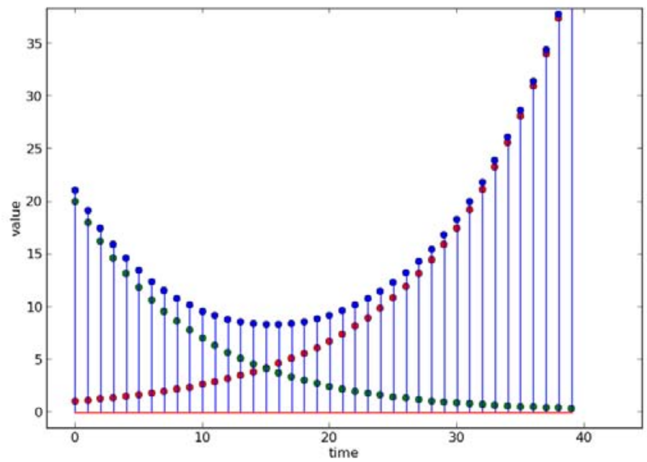


In this case, both of the poles (0.9 and 0.7) are less than 1, so the magnitude of the responses they generate decreases monotonically; their sum does not behave monotonically, but there is a time step at which the dominant pole completely dominates the other one, and the convergence is monotonic after that.

If, instead, we had a system with system function

$$H = \frac{Y}{X} = \frac{1}{1 - 1.1\mathcal{R}} + \frac{20}{1 - 0.9\mathcal{R}} \ ,$$

what would the unit-sample response be? The first mode (first term of the sum) has pole 1.1, which means it will generate monotonically increasing output values. The second mode has pole 0.9, and will decrease monotonically. The plot below illustrates the sum of these two components:

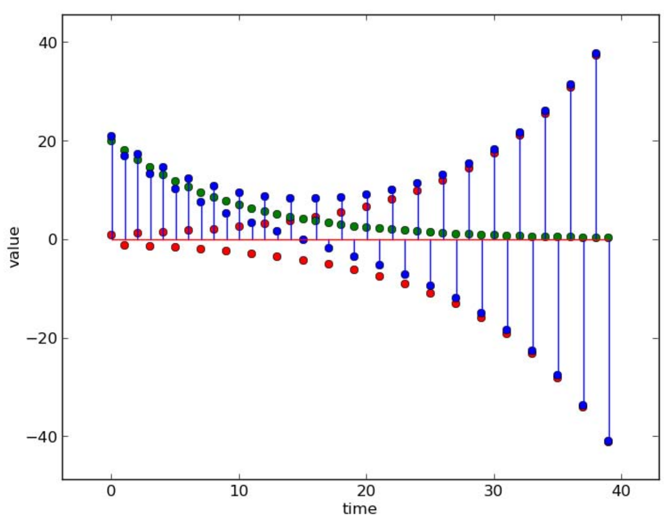


The red points correspond to the output from the mode with pole 1.1; the green points correspond to the output from the mode with pole 0.9; and the blue stem plot shows the sum.

Here is a plot of the output of the system described by

$$\frac{Y}{X} = \frac{21.1\mathcal{R} + 21}{-0.99\mathcal{R}^2 + 0.2\mathcal{R} + 1} \ .$$

It has poles 0.9 and −1.1.



The green dots are generated by the component with pole 0.9, the red dots are generated by the component with pole −1.1 and the blue dots are generated by the sum.

In these examples, as in the general case, the long-term unit-sample response of the entire system is governed by the unit-sample response of the mode whose pole has the larger magnitude. In the long run, the rate of growth of the exponential with the largest exponent will always dominate.

5.5.2.2 Complex poles

Consider a system described by the operator equation:

$$Y = 2X + 2\mathcal{R}X - 2\mathcal{R}Y - 4\mathcal{R}^2 Y \ .$$

It has system function

$$\frac{Y}{X} = \frac{2 + 2\mathcal{R}}{1 + 2\mathcal{R} + 4\mathcal{R}^2} \ . \tag{5.1}$$

But now, if we attempt to perform an additive decomposition on it, we find that the denominator cannot be factored to find real poles. Instead, we find that

$$\begin{aligned}\frac{Y}{X} &= \frac{2 + 2\mathcal{R}}{(1 - (-1 + \sqrt{-3})\mathcal{R})(1 - (-1 - \sqrt{-3})\mathcal{R})} \\ &\approx \frac{2 + 2\mathcal{R}}{(1 - (-1 + 1.732j)\mathcal{R})(1 - (-1 - 1.732j)\mathcal{R})} \ .\end{aligned}$$

So, the poles are $-1 + 1.732j$ and $-1 - 1.732j$. Note that we are using j to signal the imaginary part of a complex number.[36] What does that mean about the behavior of our system? Are the outputs real or complex? Do the output values grow or shrink with time?

Difference equations that represent physical systems have real-valued coefficients. For instance, a bank account with interest ρ might be described with difference equation

$$y[n] = (1 + \rho)y[n-1] + x[n] \ .$$

The position of a robot moving toward a wall might be described with difference equation:

$$d_o[n] = d_o[n-1] + KTd_o[n-2] - KTd_i[n-1] \ .$$

Difference equations with real-valued coefficients generate real-valued outputs from real-valued inputs. But, like the difference equation

$$y[n] = 2x[n] + 2x[n-1] - 2y[n-1] - 4y[n-1] \ ,$$

corresponding to system function 5.1, they might still have complex poles.

Polar representation of complex numbers

Sometimes it's easier to think about a complex number $a + bj$ instead as $re^{j\Omega}$, where

[36] We use j instead of i because, to an electrical engineer, i stands for current, and can't be re-used!

$$a = r\cos\Omega$$
$$b = r\sin\Omega$$

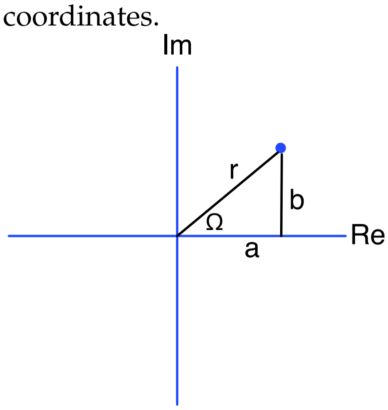
so that the *magnitude*, r, sometimes written as $|a + bj|$, is defined as

$$r = \sqrt{a^2 + b^2}$$

and the *angle*, Ω, is defined as

$$\Omega = \tan^{-1}(b, a) \quad .$$

So, if we think of (a, b) as a point in the complex plane, then (r, Ω) is its representation in polar coordinates.



This representation is justified by Euler's equation

$$e^{xj} = \cos x + j\sin x \quad ,$$

which can be directly derived from series expansions of e^z, $\sin z$ and $\cos z$. To see that this is reasonable, let's take our number, represent it as a complex exponential, and then apply Euler's equation:

$$\begin{aligned} a + bj &= re^{j\Omega} \\ &= r(\cos\Omega + j\sin\Omega) \\ &= \sqrt{a^2 + b^2}(\cos(\tan^{-1}(b, a)) + j\sin(\tan^{-1}(b, a))) \\ &= \sqrt{a^2 + b^2}(\frac{a}{\sqrt{a^2 + b^2}} + j\frac{b}{\sqrt{a^2 + b^2}}) \\ &= a + bj \end{aligned}$$

Why should we bother with this change of representation? There are some operations on complex numbers that are much more straightforwardly done in the exponential representation. In particular, let's consider raising a complex number to a power. In the Cartesian representation, we get complex trigonometric polynomials. In the exponential representation, we get, in the quadratic case,

$$\begin{aligned} \left(re^{j\Omega}\right)^2 &= re^{j\Omega}re^{j\Omega} \\ &= r^2e^{j2\Omega} \end{aligned}$$

More generally, we have that

$$\left(re^{j\Omega}\right)^n = r^n e^{jn\Omega} \quad ,$$

which is much tidier. This is an instance of an important trick in math and engineering: changing representations. We will often find that representing something in a different way will allow us to do some kinds of manipulations more easily. This is why we use diagrams, difference equations, operator equations and system functions all to describe LTI systems. There is no *one* best representation; each has advantages under some circumstances (and disadvantages under others).

Complex modes

Now, we're equipped to understand how complex poles of a system generate behavior: they produce complex-valued modes. Remember that we can characterize the behavior of a mode as

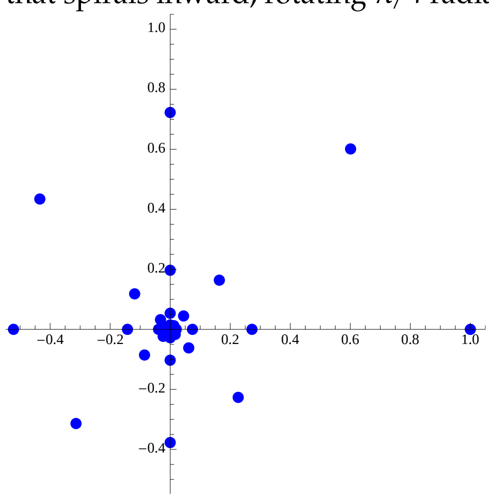
$$\frac{1}{1 - p\mathcal{R}} = 1 + p\mathcal{R} + p^2\mathcal{R}^2 + \cdots + p^n\mathcal{R}^n + \cdots \quad .$$

For a complex pole $p = re^{j\Omega}$, $p^n = r^n e^{j\Omega n}$. So

$$\frac{1}{1 - re^{j\Omega}\mathcal{R}} = 1 + re^{j\Omega}\mathcal{R} + r^2e^{j2\Omega}\mathcal{R}^2 + \cdots$$

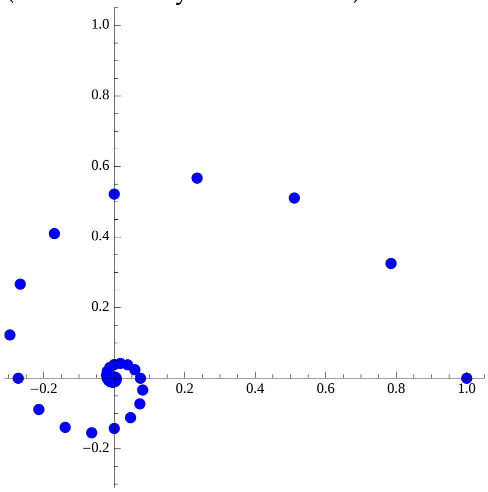
What happens as n tends to infinity when p is complex? Think of p^n as a point in the complex plane with coordinates $(r^n, \Omega n)$. The radius, r^n, will grow or shrink depending on the mode's magnitude, r. And the angle, Ωn, will simply rotate, but will not affect the magnitude of the resulting value. Note that each new point in the sequence p^n will be rotated by Ω from the previous one. We will say that the *period* of the output signal is $2\pi/\Omega$; that is the number of samples required to move through 2π radians (although this need not actually be an integer).

The sequence spirals around the origin of the complex plane. Here is a case for $r = 0.85, \Omega = \pi/4$, that spirals inward, rotating $\pi/4$ radians on each step:

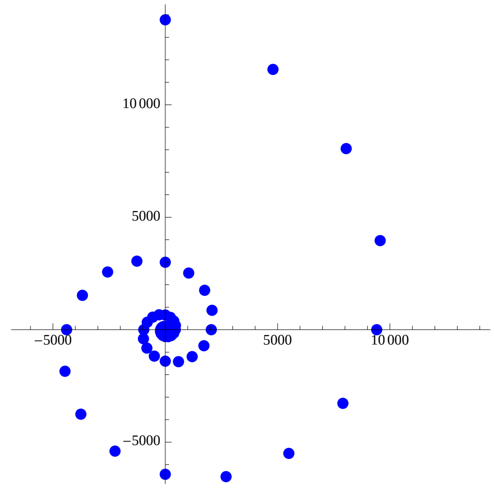


$p = 0.85e^{j\pi/4} \approx 0.601 + 0.601j$

Here, for example, are plots of two other complex modes, with different magnitudes and rates of oscillation. In all cases, we are plotting the unit-sample response, so the first element in the series is the real value 1. The first signal clearly converges in magnitude to 0; the second one diverges (look carefully at the scales).



$p = 0.85e^{j\pi/8} \approx 0.785 + 0.325j$



$p = 1.1e^{j\pi/8} \approx 1.016 + 0.421j$

If complex modes occurred all by themselves, then our signals would have complex numbers in them. But isolated complex pole can result only from a difference equation with complex-valued coefficients. For example, to end up with this system function

$$\frac{Y}{X} = \frac{1}{1 - re^{j\Omega}\mathcal{R}} ,$$

we would need to have this difference equation, which has a complex coefficient.

$$y[n] - re^{j\Omega}y[n-1] = x[n]$$

But, we have difference equations with real parameters, so we can express the value of every sample of the output signal as a linear combination of inputs and previous values with real coefficients, so we know the output signal is real-valued at all samples. The reason this all works out is that, for polynomials with real coefficients, the complex poles always come in *conjugate pairs*: that is, pairs of complex numbers $p = a + bj$ and $p^* = a - bj$. (See section 5.5.4 for a proof of this fact.) In the polar representation, the conjugate pair becomes

$$\begin{aligned} a \pm bj &= \sqrt{a^2 + b^2}\; e^{j\tan^{-1}(\pm b, a)} \\ &= \sqrt{a^2 + b^2}\; e^{\pm j\tan^{-1}(b, a)} \end{aligned}$$

where the only difference is in the sign of the angular part.

If we look at a second-order system with complex-conjugate poles, the resulting polynomial has real-valued coefficients. To see this, consider a system with poles $re^{j\Omega}$ and $re^{-j\Omega}$, so that

$$\frac{Y}{X} = \frac{1}{(1 - re^{j\Omega}\mathcal{R})(1 - re^{-j\Omega}\mathcal{R})}$$

Let's slowly multiply out the denominator:

$$(1 - re^{j\Omega}\mathcal{R})(1 - re^{-j\Omega}\mathcal{R}) = 1 - re^{j\Omega}\mathcal{R} - re^{-j\Omega}\mathcal{R} + r^2 e^{j\Omega}e^{-j\Omega}\mathcal{R}^2$$
$$= 1 - r(e^{j\Omega} + e^{-j\Omega})\mathcal{R} + r^2 e^{j\Omega - j\Omega}\mathcal{R}^2$$

Using the definition of the complex exponential $e^{jx} = \cos x + j \sin x$,

$$= 1 - r(\cos\Omega + j\sin\Omega + \cos(-\Omega) + j\sin(-\Omega))\mathcal{R} + r^2\mathcal{R}^2$$

Using trigonometric identities $\sin -x = -\sin x$ and $\cos -x = \cos x$,

$$= 1 - r(\cos\Omega + j\sin\Omega + \cos\Omega - j\sin\Omega)\mathcal{R} + r^2\mathcal{R}^2$$
$$= 1 - 2r\cos\Omega\mathcal{R} + r^2\mathcal{R}^2$$

So,

$$\frac{Y}{X} = \frac{1}{1 - 2r\cos\Omega\mathcal{R} + r^2\mathcal{R}^2} \quad .$$

This is pretty cool! All of the imaginary parts cancel, and we are left with a system function with only real coefficients, which corresponds to the difference equation

$$y[n] = x[n] + 2r\cos\Omega\, y[n-1] - r^2 y[n-2] \quad .$$

Additive decomposition with complex poles

You can skip this section if you want to.

To really understand these complex modes and how they combine to generate the output signal, we need to do an additive decomposition. That means doing a partial fraction expansion of

$$\frac{Y}{X} = \frac{1}{(1 - re^{j\Omega}\mathcal{R})(1 - re^{-j\Omega}\mathcal{R})} \quad .$$

It's a little trickier than before, because we have complex numbers, but the method is the same. The end result is that we can decompose this system additively to get

$$\frac{Y}{X} = \frac{\frac{1}{2}(1 - j\cot\Omega)}{1 - re^{j\Omega}\mathcal{R}} + \frac{\frac{1}{2}(1 + j\cot\Omega)}{1 - re^{-j\Omega}\mathcal{R}} \quad .$$

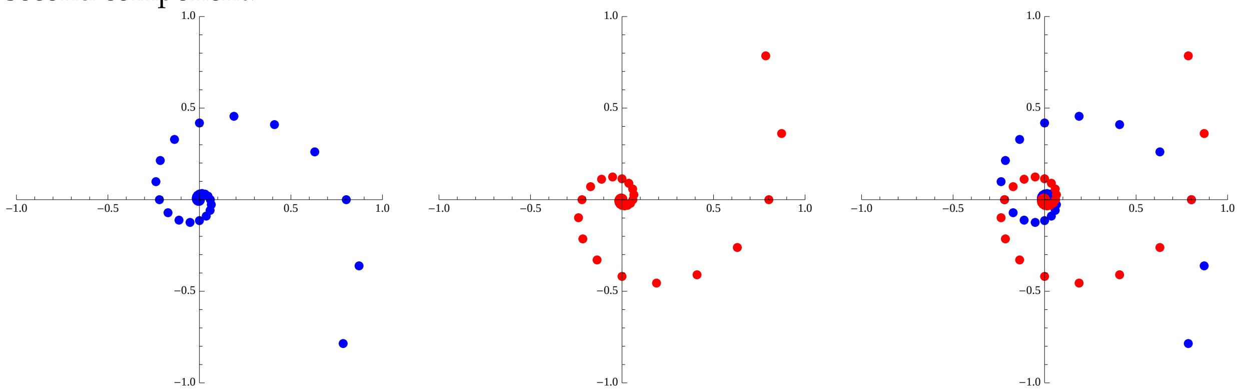
What is the unit-sample response of each of these modes? What is the unit-sample response of their sum? This might be making you nervous...it's hard to see how everything is going to come out to be real in the end.

But, let's examine the response of the additive decomposition; it's the sum of the outputs of the component systems. So, if $x[n] = \delta[n]$,

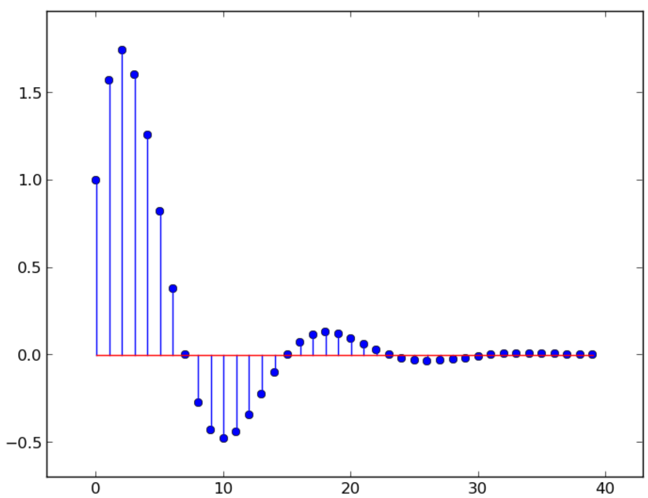
$$y[n] = \frac{1}{2}(1 - j\cot\Omega)r^n e^{jn\Omega} + \frac{1}{2}(1 + j\cot\Omega)r^n e^{-jn\Omega}$$
$$= r^n(\cos n\Omega + \cot\Omega \sin n\Omega) \quad ,$$

which is entirely real.

The figures below show the modes for a system with poles $0.85e^{j\pi/8}$ and $0.85e^{-j\pi/8}$: the blue series starts at $\frac{1}{2}(1 - j\cot(\pi/8))$ and traces out the unit sample response of the first component; the second red series starts at $\frac{1}{2}(1 + j\cot(\pi/8))$ and traces out the unit sample response of the second component.



Note, in the third figure, that the imaginary parts of the contributions of each of the modes cancel out, and that real parts are equal. Thus, the real part of the output is going to be twice the real part of these elements. The figure below shows the unit sample response of the entire system.



In the formula below,

$$y[n] = r^n(\cos n\Omega + \cot\Omega \sin n\Omega) \ ,$$

we know that

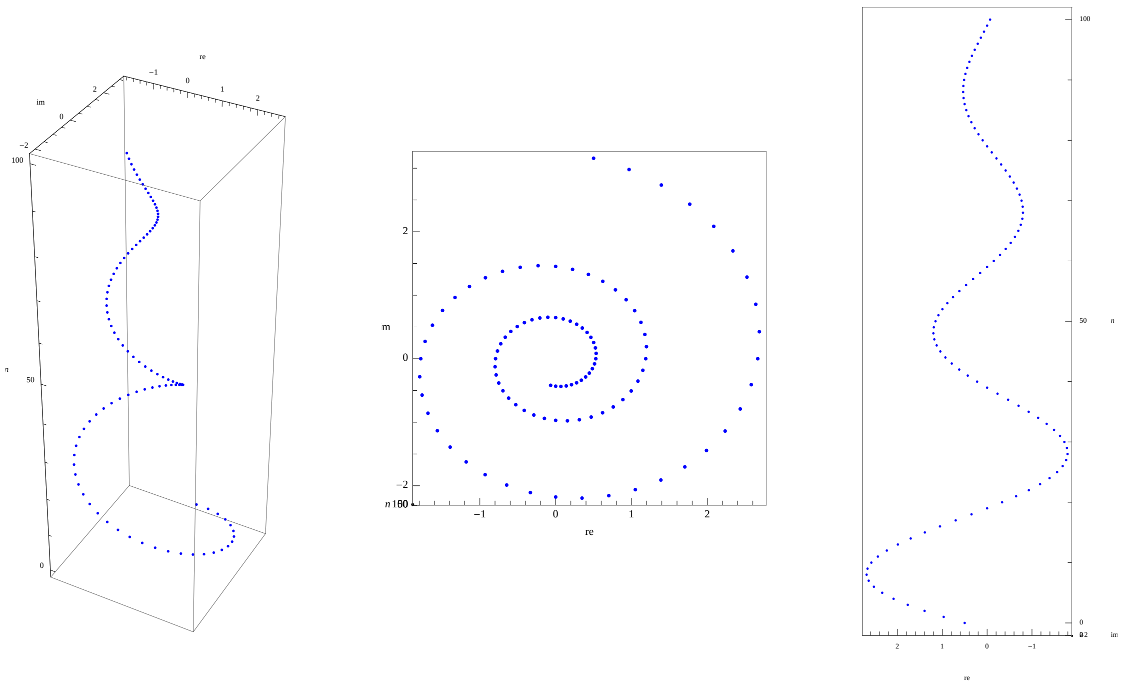
$$-\sqrt{1 + \cot^2\Omega} \leqslant \cos n\Omega + \cot\Omega \sin n\Omega \leqslant \sqrt{1 + \cot^2\Omega} \ ,$$

so

$$-\sqrt{1 + \cot^2\Omega}\, r^n \leqslant y[n] \leqslant \sqrt{1 + \cot^2\Omega}\, r^n$$

$$-\frac{1}{\sin\Omega} r^n \leqslant y[n] \leqslant \frac{1}{\sin\Omega} r^n \ .$$

Just for fun, here is a three-dimensional plot of a single mode of a system with pole $0.98e^{j\pi/20}$. These values were chosen so that it would shrink slowly and also rotate slowly around the complex plane. The first figure shows n growing upward with the complex plane oriented horizontally; the second shows a view looking down onto the complex plane; the third shows a view that projects along the imaginary axis.



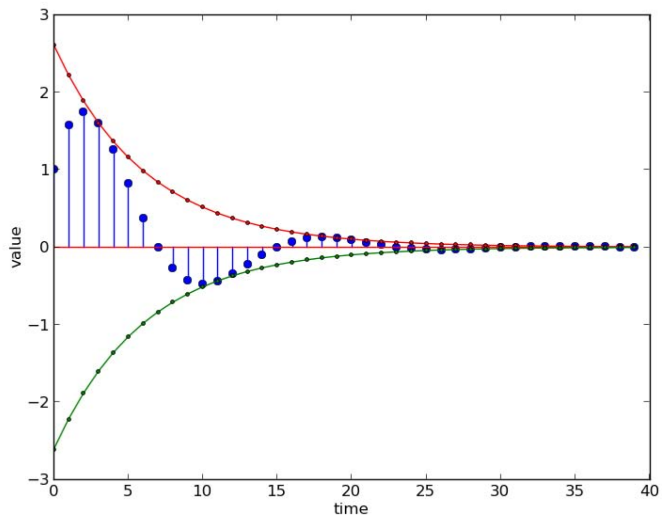
Importance of magnitude and period

Both r and Ω tells us something very useful about the way in which the system behaves. In the previous section, we derived an expression for the samples of the unit sample response for a system with a pair of complex poles. It has the form

$$y[n] = r^n(\cos n\Omega + \alpha \sin n\Omega) \quad ,$$

where α is a constant. We know that $(\cos n\Omega + \alpha \sin n\Omega)$ cannot possibly be less than $-\sqrt{1+\alpha^2}$ or greater than $\sqrt{1+\alpha^2}$.

These bounds provide an *envelope* that constrains where the values of the signal can be as a function of time, and help us understand the rate of growth or decay of the signal, as shown below. The red and green curves are $\pm\sqrt{1+\alpha^2}r^n$, for a system where $r = 0.85, \Omega = \pi/8$, and $\alpha = 2.414$.



The value of r governs the rate of exponential decrease. The value of Ω governs the rate of oscillation of the curve. It will take $2\pi/\Omega$ (the *period* of the oscillation) samples to go from peak to peak of the oscillation.[37] In our example, $\Omega = \pi/8$ so the period is 16; you should be able to count 16 samples from peak to peak.

5.5.2.3 Poles and behavior: summary

In a second-order system, if we let p_0 be the pole with *the largest magnitude*, then there is a time step at which the behavior of the dominant pole begins to dominate; *after that time step*

- If p_0 is real and
  - $p_0 < -1$, the magnitude increases to infinity and the sign alternates.
  - $-1 < p_0 < 0$, the magnitude decreases and the sign alternates.
  - $0 < p_0 < 1$, the magnitude decreases monotonically.
  - $p_0 > 1$, the magnitude increases monotonically to infinity.
- If p_0 is complex
  - and $|p_0| < 1$, the magnitude decreases monotonically.
  - and $|p_0| > 1$, the magnitude increases monotonically to infinity.
- If p_0 is complex and Ω is its angle, then the signal will be periodic, with period $2\pi/\Omega$.

As we have seen in our examples, when we add multiple modes, it is the mode with the largest pole that governs the long-term behavior.

5.5.3 Higher-order systems

Recall that we can describe any system in terms of a system function that is the ratio of two polynomials in $\mathcal{R}$ (and assuming $a_0 = 1$):

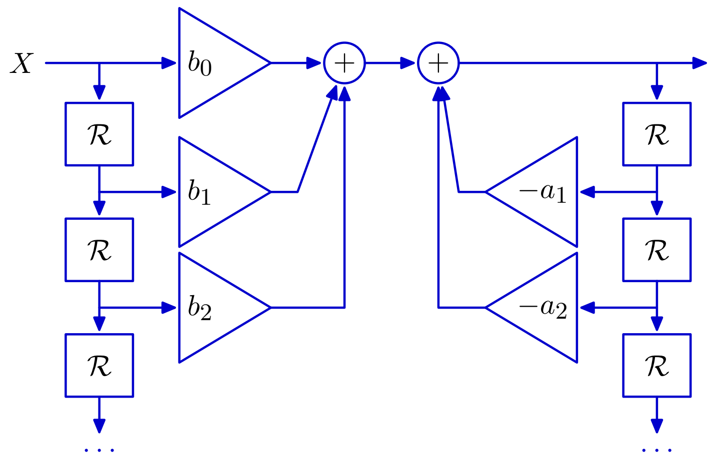
[37] We are being informal here, in two ways. First, the signal does not technically have a period, because unless $r = 1$, it doesn't return to the same point. Second, unless $r = 1$, then distance from peak to peak is not exactly $2\pi/\Omega$, however, for most signals, it will give a good basis for estimating Ω.

$$\frac{Y}{X} = \frac{b_0 + b_1\mathcal{R} + b_2\mathcal{R}^2 + b_3\mathcal{R}^3 + \cdots}{1 + a_1\mathcal{R} + a_2\mathcal{R}^2 + a_3\mathcal{R}^3 + \cdots}$$

Regrouping terms, we can write this as the operator equation:

$$Y = (b_0 + b_1\mathcal{R} + b_2\mathcal{R}^2 + b_3\mathcal{R}^3 + \cdots)\, X - (a_1\mathcal{R} + a_2\mathcal{R}^2 + a_3\mathcal{R}^3 + \cdots)\, Y$$

and construct an equivalent block diagram:



Returning to the general polynomial ratio

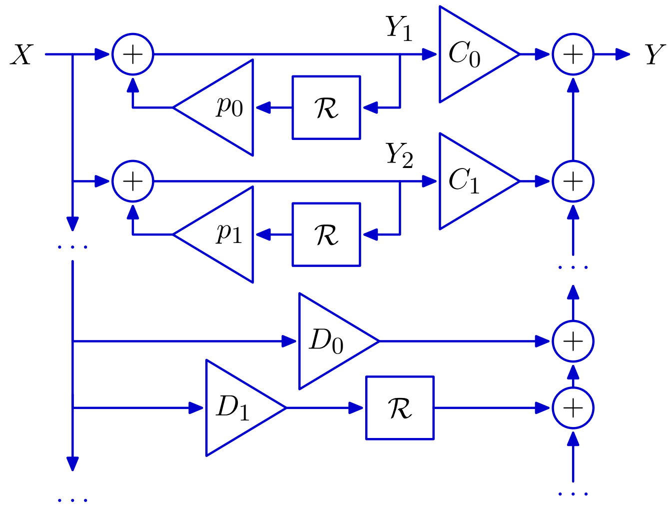
$$\frac{Y}{X} = \frac{b_0 + b_1\mathcal{R} + b_2\mathcal{R}^2 + b_3\mathcal{R}^3 + \cdots}{1 + a_1\mathcal{R} + a_2\mathcal{R}^2 + a_3\mathcal{R}^3 + \cdots}$$

We can factor the denominator of an nth-order polynomial into n factors, and then perform a partial fraction expansion, to turn it into the form of a sum of terms. We won't go over the details here (there are nice tutorials online), but it comes out in the form:

$$\frac{Y}{X} = \frac{C_0}{1 - p_0\mathcal{R}} + \frac{C_1}{1 - p_1\mathcal{R}} + \frac{C_2}{1 - p_2\mathcal{R}} + \cdots + D_0 + D_1\mathcal{R} + D_2\mathcal{R}^2 + \cdots$$

where the C_k and D_k are constants defined in terms of the a_i and b_j from the original polynomial ratio. It's actually a little bit trickier than this: if there are complex poles, then for each conjugate pair of complex poles, we would put in a second-order system with real coefficients that expresses the contribution of the sum of the complex modes.

The constant term D_0 and the terms with $\mathcal{R}^k$ in the numerator occur if the numerator has equal or higher order than the denominator. They do not involve feedback and don't affect the long-term behavior of the system. One mode of the form p_i^n arises from each factor of the denominator. This modal decomposition leads us to an alternative block diagram:



We can fairly easily observe that the behavior is going to be the sum of the behaviors of the individual modes, and that, as in the second-order case, the mode whose pole has the largest magnitude will govern the qualitative long-term nature of the behavior of the system in response to a unit-sample input.

5.5.4 Finding poles

In general, we will find that if the denominator of the system function H is a kth order polynomial, then it can be factored into the form $(1-p_0\mathcal{R})(1-p_1\mathcal{R})\ldots(1-p_{k-1}\mathcal{R})$. We will call the p_i values the *poles* of the system. The entire persistent output of the system can be expressed as a scaled sum of the signals arising from each of these individual poles.

We're doing something interesting here! We are using the PCAP system backwards for analysis. We have a complex thing that is hard to understand monolithically, so we are taking it apart into simpler pieces that we do understand.

It might seem like factoring polynomials in this way is tricky, but there is a straightforward way to find the poles of a system given its denominator polynomial in $\mathcal{R}$.

We'll start with an example. Assume the denominator is $12\mathcal{R}^2 - 7\mathcal{R} + 1$. If we play a quick trick, and introduce a new variable $z = 1/\mathcal{R}$, then our denominator becomes

$$\frac{12}{z^2} - \frac{7}{z} + 1 \ .$$

We'd like to find the roots of this polynomial, which, if you multiply through by z^2, is equivalent to finding the roots of this poyinomial:

$$12 - 7z + z^2 = 0 \ .$$

The roots are 3 and 4. If we go back to our original polynomial in $\mathcal{R}$, we can see that:

$$12\mathcal{R}^2 - 7\mathcal{R} + 1 = (1 - 4\mathcal{R})(1 - 3\mathcal{R}) \ .$$

so that our poles are 4 and 3. So, remember, *the poles* ***are not*** *the roots of the polynomial in* $\mathcal{R}$*, but* ***are*** *the roots of the polynomial in the reciprocal of* $\mathcal{R}$*.*

The roots of a polynomial can be a combination of real and complex numbers, with the requirement that if a complex number p is a root, then so is its complex conjugate, p^*.

Pole-Zero cancellation

For some systems, we may find that it is possible to cancel matching factors from the numerator and denominator of the system function. If the factors represent poles that are not at zero, then although it may be theoretically supportable to cancel them, it is unlikely that they would match *exactly* in a real system. If we were to cancel them in that case, we might end up with a system model that was a particularly poor match for reality.[38]

So, we will only consider canceling poles and zeros at zero. In this example:

$$H = \frac{\mathcal{R}}{\mathcal{R} - 1.6\mathcal{R}^2 + 0.63\mathcal{R}^3}$$

we can cancel $\mathcal{R}$, getting

$$H = \frac{1}{1 - 1.6\mathcal{R} + 0.63\mathcal{R}^2}$$

Now, we find the poles by finding the roots of the equation

$$z^2 - 1.6z + 0.63 = 0$$

Repeated roots

In some cases, the equation in z will have repeated roots. For example, $z^2 - z + 0.25$, which has two roots at 0.5. In this case, the system has a repeated pole; it is still possible to perform an additive decomposition, but it is somewhat trickier. Ultimately, however, it is still the magnitude of the largest root that governs the long-term convergence properties of the system.

5.5.5 Superposition

The principle of superposition states that the response of a LTI system to a sum of input signals is the sum of the responses of that system to the components of the input. So, given a system with system function H, and input $X = X_1 + X_2$,

$$Y = HX = H(X_1 + X_2) = HX_1 + HX_2$$

So, although we have been concentrating on the unit sample response of systems, we can see that, to find the response of a system to any finite signal, we must simply sum the responses to each of the components of that signal; and those responses will simply be scaled, delayed copies of the response to the unit sample.

If Φ is a polynomial in $\mathcal{R}$ and $X = \Phi\Delta$, then we can use what we know about the algebra of polynomials in $\mathcal{R}$ (remembering that H is a ratio of polynomials in $\mathcal{R}$) to determine that

$$Y = HX = H(\Phi\Delta) = \Phi(H\Delta)$$

[38] Don't worry too much if this doesn't make sense to you...take 6.003 to learn more.

So, for example, if $X = (-3\mathcal{R}^2 + 20\mathcal{R}^4)\Delta$, then $Y = HX = H(-3\mathcal{R}^2 + 20\mathcal{R}^4)\Delta = (-3\mathcal{R}^2 + 20\mathcal{R}^4)H\Delta$. From this equation, it is easy to see that once we understand the unit-sample response of a system, we can see how it will respond to any finite input.

We might be interested in understanding how a system H responds to a *step* input signal. Let's just consider the basic step signal, U, defined as

$$u[n] = \begin{cases} 1 & \text{if } n \geqslant 0 \\ 0 & \text{otherwise} \end{cases} \quad .$$

We can express U as an infinite sum of increasingly delayed unit-sample signals:

$$\begin{aligned} U &= \Delta + \mathcal{R}\Delta + \mathcal{R}^2\Delta + \mathcal{R}^3\Delta + \cdots \\ &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\Delta \quad . \end{aligned}$$

The response of a system to U will therefore be an infinite sum of unit-sample responses. Let $Z = H\Delta$ be the unit-sample response of H. Then

$$\begin{aligned} HU &= H(1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\Delta \\ &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)H\Delta \\ &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)Z \end{aligned}$$

Let's consider the case where H is a first-order system with a pole at p. Then,

$$z[n] = p^n$$

If $Y = HU$, then

$$\begin{aligned} y[n] &= z[n] + z[n-1] + z[n-2] + \cdots + z[0] \\ &= \sum_{k=0}^{n} z[k] \\ &= \sum_{k=0}^{n} p^k \end{aligned}$$

It's clear that, if $|p| >= 1$ then $y[n]$ will grow without bound; but if $0 < p < 1$ then, as n goes to infinity, $y[n]$ will converge to $1/(1-p)$.

We won't study this in any further detail, but it's useful to understand that the basis of our analysis of systems applies very broadly across LTI systems and inputs.

5.6 Designing systems

Will eventually include a discussion of root-locus plots.

For now, see **section 5.8.3** for a discussion of picking k for the wall finding robot.

5.7 Summary of system behavior

Here is some terminology that will help us characterize the long-term behavior of systems.

- A signal is *transient* if it has finitely many non-zero samples.
- Otherwise, it is *persistent*.
- A signal is *bounded* if there is exist upper and lower bound values such that the samples of the signal never exceed those bounds; item otherwise it is *unbounded*.

Now, using those terms, here is what we can say about system behavior.

- A transient input to an acyclic (feed-forward) system results in a transient output.
- A transient input to a cyclic (feed-back) system results in a persistent output.
- The poles of a system are the roots of the denominator polynomial of the system function in $1/\mathcal{R}$.
- The dominant pole is the pole with the largest magnitude.
- If the dominant pole has magnitude > 1, then in response to a bounded input, the output signal will be unbounded.
- If the dominant pole has magnitude < 1, then in response to a bounded input, the output signal will be bounded; in response to a transient input, the output signal will converge to 0.
- If the dominant pole has magnitude 1, then in response to a bounded input, the output signal will be bounded; in response to a transient input, it will converge to some constant value.
- If the dominant pole is real and positive, then in response to a transient input, the signal will, after finitely many steps, begin to increase or decrease monotonically.
- If the dominant pole is real and negative, then in response to a transient input, the signal will, after finitely many steps, begin to alternate signs.
- If the dominant pole is complex, then in response to a transient input, the signal will, after finitely many steps, begin to be periodic, with a period of $2\pi/\Omega$, where Ω is the 'angle' of the pole.

5.8 Worked Examples

5.8.1 Specifying difference equations

Here are some examples of LTI systems and the way they would be described as difference equations. It's useful to pay careful attention to the specification of the coefficients. As a reminder, here's the general form.

$$y[n] = c_0\, y[n-1] + c_1\, y[n-2] + \ldots + c_{k-1}\, y[n-k] \\ + d_0\, x[n] + d_1\, x[n-1] + \ldots + d_j\, x[n-j]$$

- Output at step n is 3 times the input at step n:

  $$y[n] = 3x[n]$$

  dCoeffs: 3, cCoeffs: none

- Output at step n is the input at step $n-1$:

  $$y[n] = x[n-1]$$

  dCoeffs: 0, 1, cCoeffs: none

- Output at step n is 2 times the input at step $n-2$:

$$y[n] = 2x[n-2]$$

dCoeffs: 0, 0, 2, cCoeffs: none

- Output at step n is 2 times the output at step $n-1$:

$$y[n] = 2y[n-1]$$

dCoeffs: none, cCoeffs: 2

- Output at step n is the input at step $n-1$ plus the output at step $n-2$:

$$y[n] = x[n-1] + y[n-2]$$

dCoeffs: 0, 1, cCoeffs: 0, 1

5.8.2 Difference equations and block diagrams

Let H represent a system whose input is a signal X and whose output is a signal Y. The system H is defined by the following difference equations:

$$y[n] = x[n] + z[n]$$
$$z[n] = y[n-1] + z[n-1]$$

Start by finding an operator equation.

$$Y = X + Z$$
$$Z = \mathcal{R}Y + \mathcal{R}Z$$

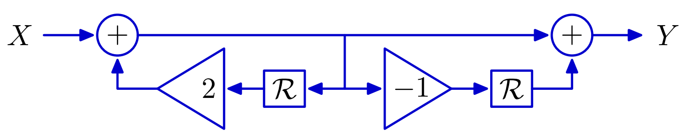
Now, we use the second equation to find an expression for Z:

$$Z = \frac{\mathcal{R}Y}{1-\mathcal{R}}$$

and substitute that into the first equation, and solve for Y:

$$Y = X + \frac{\mathcal{R}Y}{1-\mathcal{R}}$$
$$(1-\mathcal{R})Y = (1-\mathcal{R})X + \mathcal{R}Y$$
$$(1-2\mathcal{R})Y = (1-\mathcal{R})X$$
$$Y = \frac{1-\mathcal{R}}{1-2\mathcal{R}}X$$

Part a. Which of the following systems are valid representations of H? (Remember that there can be multiple "equivalent" representations for a system.)



equivalent to H (yes/no)? **YES**

Find operator equations here; start by naming the signal that's flowing between the two adders. Let's call it W.

$$Y = W - \mathcal{R}W$$
$$W = X + 2\mathcal{R}W$$

Now, rewrite the first equation as

$$Y = (1 - \mathcal{R})W$$

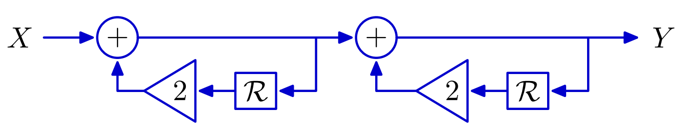
And the second one as

$$W = \frac{X}{1 - 2\mathcal{R}}$$

Then, combine them to get

$$Y = (1 - \mathcal{R})\frac{X}{1 - 2\mathcal{R}}$$
$$= \frac{1 - \mathcal{R}}{1 - 2\mathcal{R}}X$$

Showing that this system is the same as H.



equivalent to H (yes/no)? **NO**

This time, let's name the signal that's flowing between the two adders A. Now, we have equations

$$Y = A + 2\mathcal{R}Y$$
$$A = X + 2\mathcal{R}A$$

Rewrite the first equation as:

$$Y = \frac{A}{1 - 2\mathcal{R}}$$

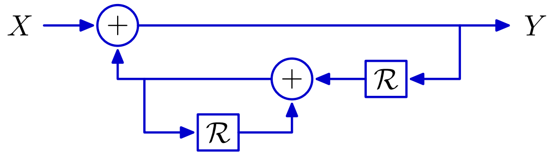
And the second as:

$$A = \frac{X}{1 - 2\mathcal{R}}$$

When we combine them, we get

$$Y = \frac{X}{(1 - 2\mathcal{R})(1 - 2\mathcal{R})}$$

which is not equivalent to the original system.



equivalent to H (yes/no)? **YES**

We'll name the signal flowing between the adders B. We get equations

$$Y = X + B$$

$$B = \mathcal{R}Y + \mathcal{R}B$$

Rewriting the second equation, we have

$$B = \frac{\mathcal{R}Y}{1 - \mathcal{R}}$$

Substituting into the first equation and solving, we get:

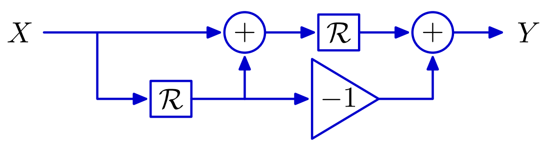
$$Y = X + \frac{\mathcal{R}Y}{1 - \mathcal{R}}$$

$$Y(1 - \mathcal{R}) = (1 - \mathcal{R})X + \mathcal{R}Y$$

$$Y(1 - 2\mathcal{R}) = (1 - \mathcal{R})X$$

$$Y = \frac{1 - \mathcal{R}}{1 - 2\mathcal{R}}X$$

So this system is equivalent to the original one.



equivalent to H (yes/no)? **NO**

Let's call the signal coming out of the first adder C. We get equations

$$Y = \mathcal{R}C - \mathcal{R}X$$

$$C = X + \mathcal{R}X$$

So

$$Y = \mathcal{R}(X + \mathcal{R}X) - \mathcal{R}X$$

$$= \mathcal{R}^2 X$$

which is not equivalent to the original system.

Part b. Assume that the system starts "at rest" and that the input signal X is the unit sample signal. Determine $y[3]$.

$y[3] =$ **4**

n	x[n]	z[n]	y[n]
−1	0	0	0
0	1	0	1
1	0	1	1
2	0	2	2
2	0	4	4

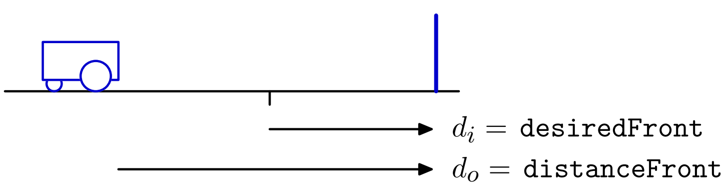
Part c. Let p_o represent the dominant pole of H. Determine p_o.

Enter p_0 or **none** if there are no poles: **2**

The denominator polynomial of the system function is $1 - 2\mathcal{R}$. This is directly in the form that exposes the pole as 2. But, we can also go step by step. We convert this into a polynomial in $z = 1/\mathcal{R}$ to get $1 - 2/z$. The roots of that equation are the same as the roots of $z - 2 = 0$; the single root is 2.

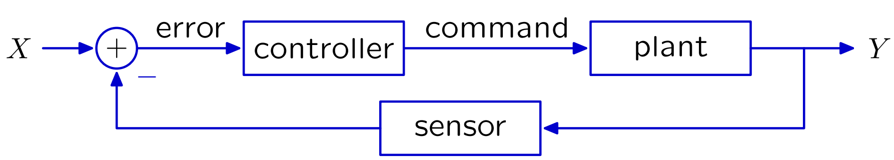
5.8.3 Wall finder

Let's think about a robot driving straight toward a wall. It has a distance sensor that allows it to observe the distance to the wall at time n, $D_s[n]$, and it desires to maintain a distance D_i from the wall.



The robot can execute velocity commands, and we program it to use this rule to set its forward velocity at time n, to be proportional to the error between D_i and its current sensor reading. In this section, we'll develop two different models of the system that includes the robot and the world around it, and try to see how we can use the models to understand how to select a controller.

In this block diagram, we will begin by considering the case in which the sensor box is just a wire; then consider what happens when it is a delay instead.



5.8.3.1 Version 1: No computational delay

In this first model, we'll assume that the sensor responds immediately, so that the robot's commanded velocity $v[n]$ depends *instantaneously* on its actual distance $d_o[n]$ from the wall. Of course this dependence can't truly be instantaneous, but it might be quite small relative to the robot's 0.1-second cycle time and so might justifiably be ignored. So:

$$v[n] = k(d_i[n] - d_o[n]) \enspace ,$$

Although D_i will generally be a constant value, we'll allow for a more general case in which it may vary over time.

We can describe this system with the operator equation:

$$V = k(D_i - D_o) \enspace .$$

Now, we can think about the "plant"; that is, the relationship between the robot and the world. The distance of the robot to the wall changes at each time step depending on the robot's forward velocity and the length of the time steps. Let T be the length of time between velocity commands issued by the robot. Then we can describe the plant with the equation:

$$d_o[n] = d_o[n-1] - Tv[n-1] \enspace .$$

That is, the new distance from the wall is equal to the old distance from the wall, minus the robot's velocity towards the wall times the time interval of a step. In operator algebra terms, we have

$$\begin{aligned} D_o &= \mathcal{R}D_o - T\mathcal{R}V \\ D_o - \mathcal{R}D_o &= -T\mathcal{R}V \\ D_o(1-\mathcal{R}) &= -T\mathcal{R}V \end{aligned}$$

Our overall system is a combination of the plant and the controller; we can combine the two equations to get

$$\begin{aligned} D_o(1-\mathcal{R}) &= -T\mathcal{R}V \\ D_o(1-\mathcal{R}) &= -T\mathcal{R}k(D_i - D_o) \\ D_o(1-\mathcal{R}-Tk\mathcal{R}) &= Tk\mathcal{R}D_i \\ D_o &= \frac{-Tk\mathcal{R}}{1-(1+Tk)\mathcal{R}}D_i \end{aligned}$$

We can solve for the poles analytically after substituting $1/z$ in for $\mathcal{R}$:

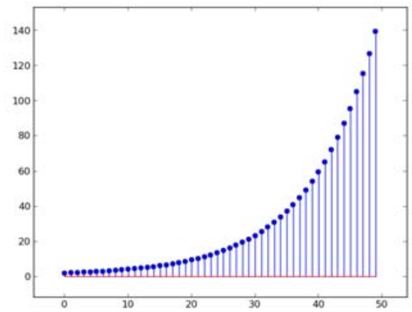
$$z - (1+Tk) = 0 \enspace .$$

There is one pole at $1 + Tk$.

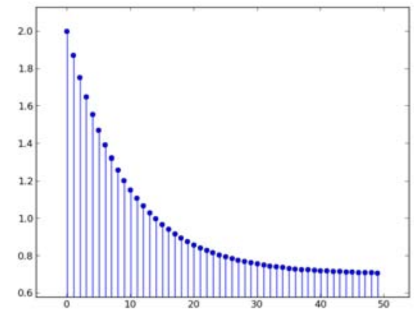
In order for the system to converge, we need

$$\begin{aligned} |\,1+Tk\,| &< 1 \\ -1 < \; 1+Tk &< 1 \\ -2 < \; Tk &< 0 \\ \frac{-2}{T} < \; k &< 0 \end{aligned}$$

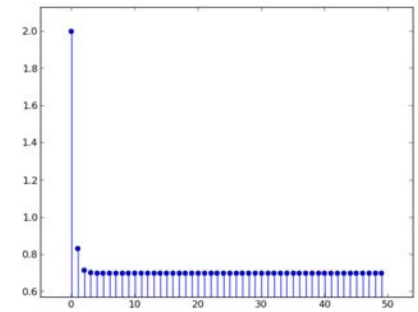
Assuming that $T = 0.1$ (which it is for our robots), then we can use this information to select k. Here are some plots of the evolution of the system, starting at distance 2.0, with an input $D_i = 0.7$.



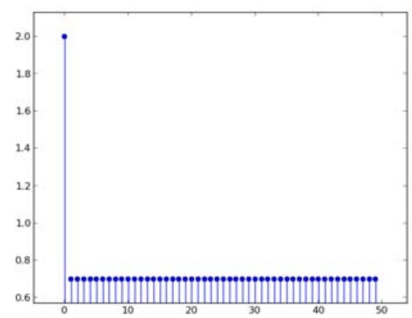
$k = 1$



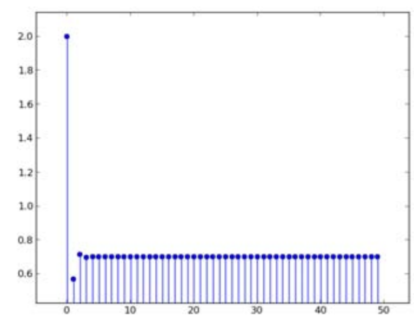
$k = -1$



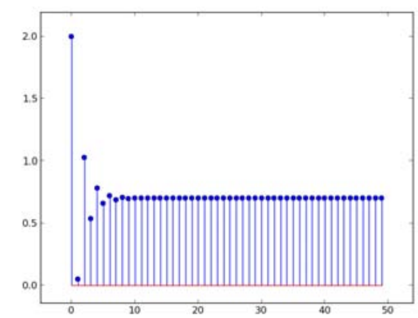
$k = -9$



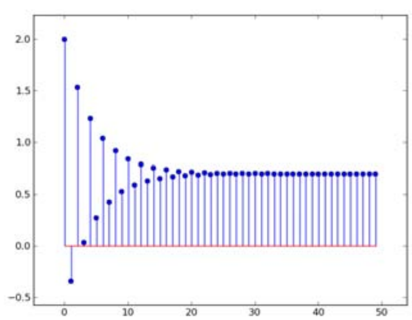
$k = -10$



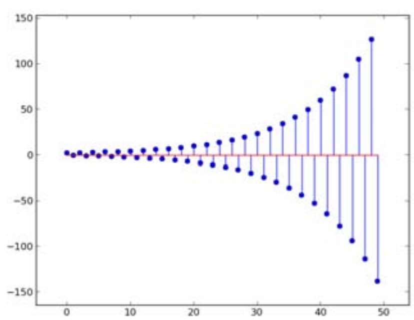
$k = -11$



$k = -15$



$k = -18$

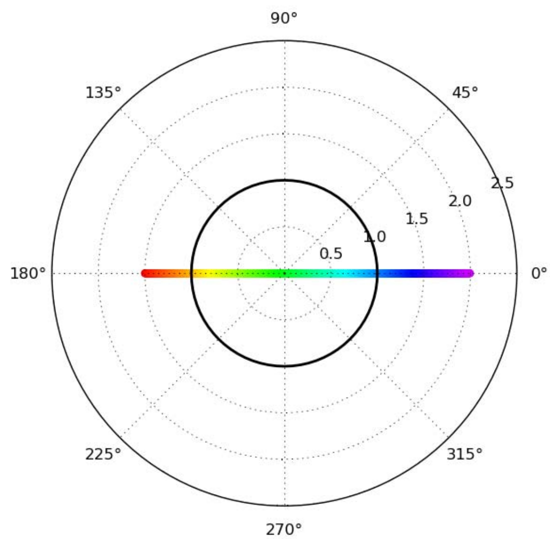


$k = -21$

Generally speaking, the closer the magnitude of the dominant pole is to 0, the faster the system will converge. For this system, $k = -10$ is the 'perfect' gain, which, in a perfect world, would make the robot jump, instantaneously, to the right place. This corresponds to having a pole of 0. (Note that, in this case, the system function degenerates into $\frac{D_o}{D_i} = \mathcal{R}$.)

Of course, in the real world, there will be error, which will cause the robot to overshoot or undershoot, and have to correct, etc. And, in the real world, we can't cause the robot to have an instantaneous increase (or decrease) in velocity, so we couldn't even approach the 'ideal' behavior of moving to the goal all in one step. Note that a positive gain causes the system to diverge as does a gain less than -20. And for gain values between -10 and -20, it converges, but alternates sign.

Here is what is called a *root-locus* plot. It shows how the poles of the system (in this case just one pole) move in the complex plane as we vary parameter k. In this figure, we varied k from -25 to $+10$. The corresponding poles are plotted in different colors, starting with red corresponding to $k = -25$ through violet corresponding to $k = +10$. First, we can easily see that for any value of k, the pole is on the real line. Then, we observe that for the lowest values of k, the pole is outside the unit circle (drawn in dark black, any value inside it has magnitude less than 1), and will cause divergence, and for the highest values of k it is also outside the unit circle and will also diverge.



5.8.3.2 Model 2: A delay in the sensor

Now, we'll consider a version of the problem in which there is a delay in the sensor, so that the commanded velocity $v[n]$ depends on the distance at the *previous* time step, $d_o[n-1]$, rather than on $d_o[n]$.

$$v[n] = k(d_i[n] - d_o[n-1]) \quad ,$$

We can describe this system with the operator equation:

$$V = k(D_i - \mathcal{R}D_o) \quad .$$

We'll leave the model of the plant as it was above (but note that in a real situation, there might be additional delays in the plant, as well as, or instead of, the delay we're modeling in the controller).

$$D_o(1 - \mathcal{R}) = -T\mathcal{R}V \quad .$$

Our overall system is a combination of the plant and the controller; we can combine the two equations to get

$$
\begin{aligned}
D_o(1-\mathcal{R}) &= -T\mathcal{R}V \\
D_o(1-\mathcal{R}) &= -T\mathcal{R}k(D_i-\mathcal{R}D_o) \\
D_o(1-\mathcal{R}-Tk\mathcal{R}^2) &= -Tk\mathcal{R}D_i \\
D_o &= \frac{-Tk\mathcal{R}}{1-\mathcal{R}-Tk\mathcal{R}^2}D_i
\end{aligned}
$$

We can solve for the poles analytically after substituting $1/z$ for $\mathcal{R}$:

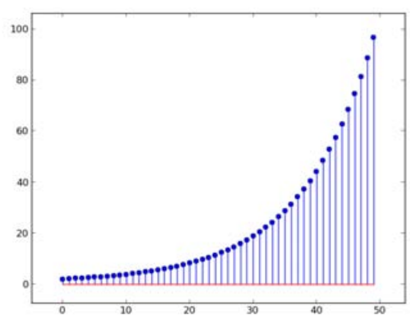
$$z^2 - z - Tk = 0 \quad .$$

We find that the roots of this polynomial are

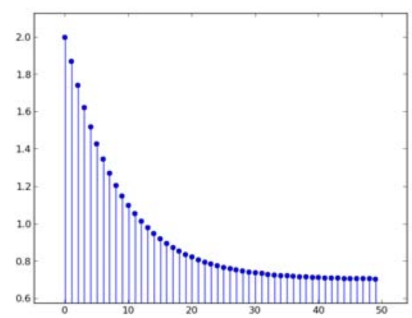
$$\frac{1}{2} \pm \frac{1}{2}\sqrt{1+4kT} \quad .$$

To determine the behavior of the system for some value of k, you can plug it into this formula and see what the values are. Remember that the pole with the largest magnitude will govern the long-term behavior of the system.

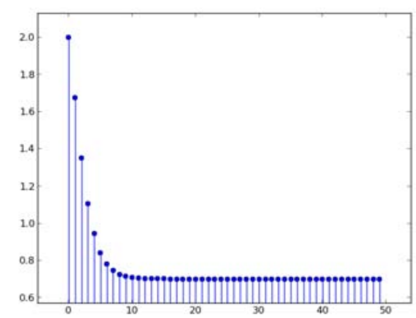
Here are some plots of the evolution of the system, starting at distance 2.0, with an input $D_i = 0.7$.



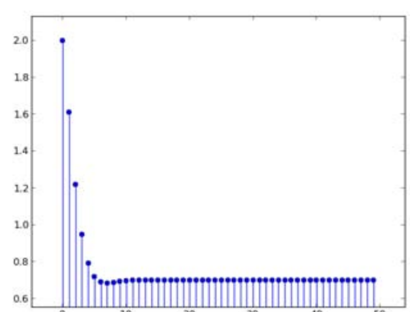
$k = 1$



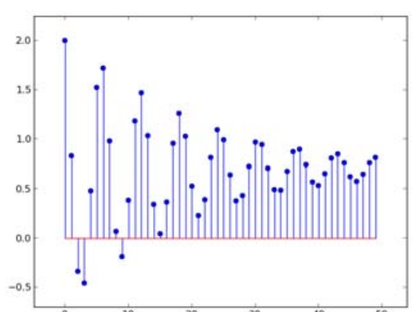
$k = -1$



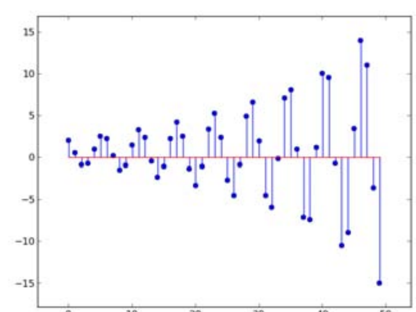
$k = -2.5$



$k = -3$



$k = -9$



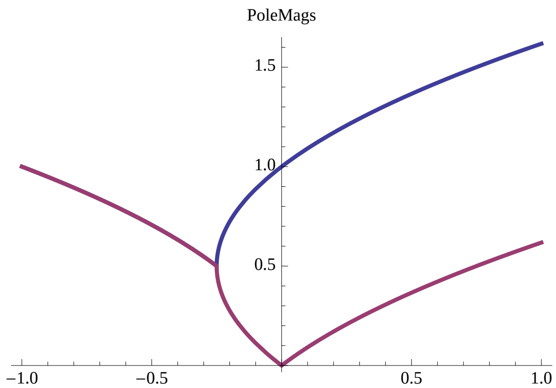
$k = -11$

$T = 0.1$, the system is monotonically divergent for $k > 0$, monotonically convergent for $-2.5 < k < 0$, converges but oscillates for $-10 < k < -2.5$, and diverges while oscillating for $k < -10$.

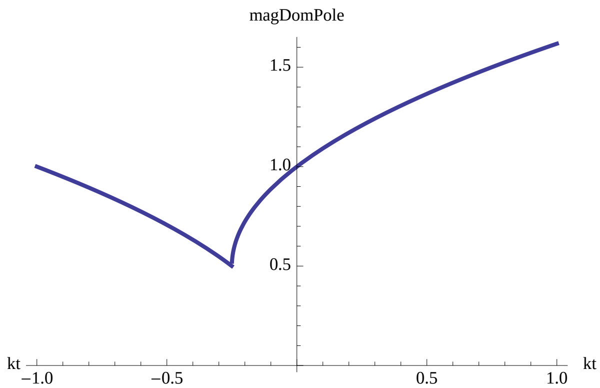
Below, on the left is

$$|\frac{1}{2} \pm \frac{1}{2}\sqrt{1+4kT}|$$

that is, the magnitudes of the two poles as a function of kT. Note that, on the left branch, there are still two poles, but they are a complex conjugate pair with the same magnitude. On the right is the maximum of the pole magnitudes. We can see that it is minimized at $kT = -0.25$.

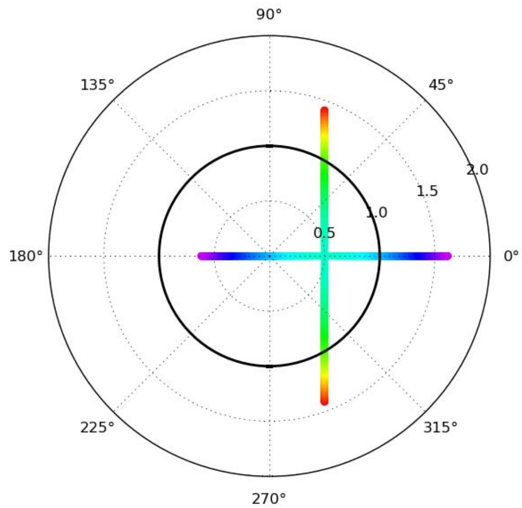


Magnitudes of both poles



Magnitude of dominant pole

Here is a *root-locus* plot for this system. It shows how the poles of the system move in the complex plane as we vary parameter k. In this figure, we varied k from −20 to +10. The corresponding poles are plotted in different colors, starting with red corresponding to $k = -20$ through violet corresponding to $k = +10$. There are two red points, corresponding to the conjugate pair of complex poles arising when $k = -20$. This system is unstable, because the magnitudes of those complex poles are greater than 1 (outside the unit circle). As we increase k, these poles move down; that is, their real part stays constant and the imaginary part decreases, until we reach complex poles (greenish on the plot) that are stable. Finally, these two poles meet on the real line: one 'turns' right as k increases and one 'turns' left. Now, we have two real poles, one of which is closer to the unit circle than the other. As soon as the positive real pole is equal to 1 then the system will begin to turn unstable again (even though the other pole is still inside the unit circle. This, for $k = 10$, the system is unstable (eventually diverging monotonically) because the largest pole is positive and outside the unit circle.



5.8.4 Cool difference equations

Newton's law of cooling states that: the rate of change of the temperature of an object is proportional to the difference between its own temperature and the temperature of its surroundings.

We can model this process in discrete time, by assuming that the change in an object's temperature from one time step to the next is proportional to the difference (on the earlier step) between the temperature of the object and the temperature of the environment, as well as to the length of the time step.

Let

- $o[n]$ be temperature of object
- $s[n]$ be temperature of environment
- T be the duration of a time step
- K be the constant of proportionality

Part a. Write a difference equation for Newton's law of cooling. Be sure the signs are such that the temperature of the object will eventually equilibrate with that of the environment.

$$o[n] = o[n-1] + TK(s[n-1] - o[n-1])$$

Part b. Write the system function corresponding to this equation (show your work):

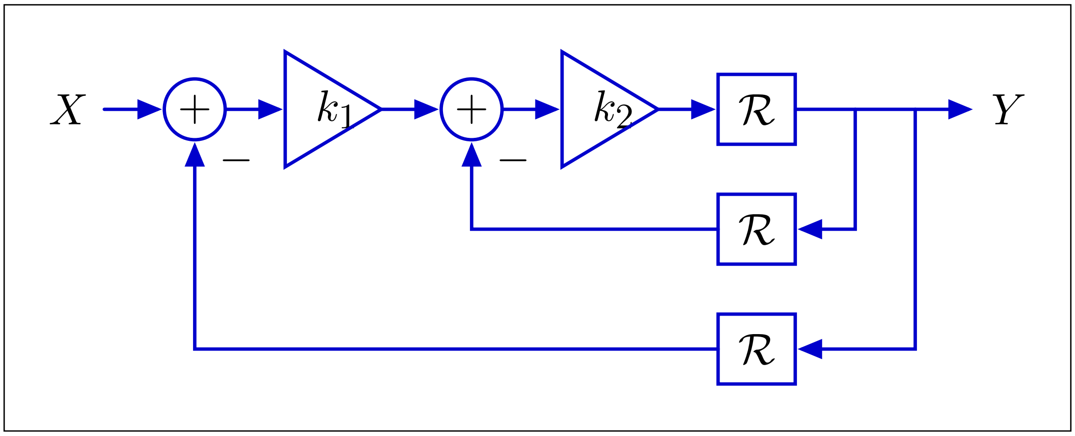
$$H = \frac{O}{S} = \frac{KT\mathcal{R}}{1-(1-KT)\mathcal{R}}$$

First, convert the difference equation to an operator equation, then solve for O in terms of S.

$$O = \mathcal{R}O + KT(\mathcal{R}S - \mathcal{R}O)$$
$$O - \mathcal{R}O + KT\mathcal{R}O = KT\mathcal{R}S$$
$$O(1 - (1 - KT)\mathcal{R}) = KT\mathcal{R}S$$
$$\frac{O}{S} = \frac{KT\mathcal{R}}{1 - (1 - KT)\mathcal{R}}$$

5.8.5 Modeling

Consider the following system:



Part a. Write the system function:

$$H = \frac{Y}{X} = \frac{k_1 k_2 \mathcal{R}}{1 + k_2 \mathcal{R}^2 (1 + k_1)}$$

We start by naming some of the internal signals. Let Z be the output of the gain of k_1 and W be the output of the gain of k_2. Then we can write the following set of operator equations:

$$Z = k_1(X - \mathcal{R}Y)$$
$$W = k_2(Z - \mathcal{R}Y)$$
$$Y = \mathcal{R}W$$

Eliminating Z and W, we have:

$$Y = \mathcal{R}k_2(Z - \mathcal{R}Y)$$
$$= \mathcal{R}k_2(k_1(X - \mathcal{R}Y) - \mathcal{R}Y)$$
$$= k_1 k_2 \mathcal{R}X - k_1 k_2 \mathcal{R}^2 Y - k_2 \mathcal{R}^2 Y$$

Reorganizing terms, we have

$$Y + k_2(1 + k_1)\mathcal{R}^2 Y = k_1 k_2 \mathcal{R}X$$

which leads us to the answer.

Part b.

Let $k_1 = 1$ and $k_2 = -2$. Assume that the system starts "at rest" (all signals are zero) and that the input signal X is the unit sample signal. Determine $y[0]$ through $y[3]$.

$y[0] =$ **0**

$y[1] =$ **-2**

$y[2] =$ **0**

$y[3] =$ **-8**

First, we write the difference equation:

$$y[n] = -2x[n-1] + 4y[n-2]$$

Then we can calculate the values step by step.

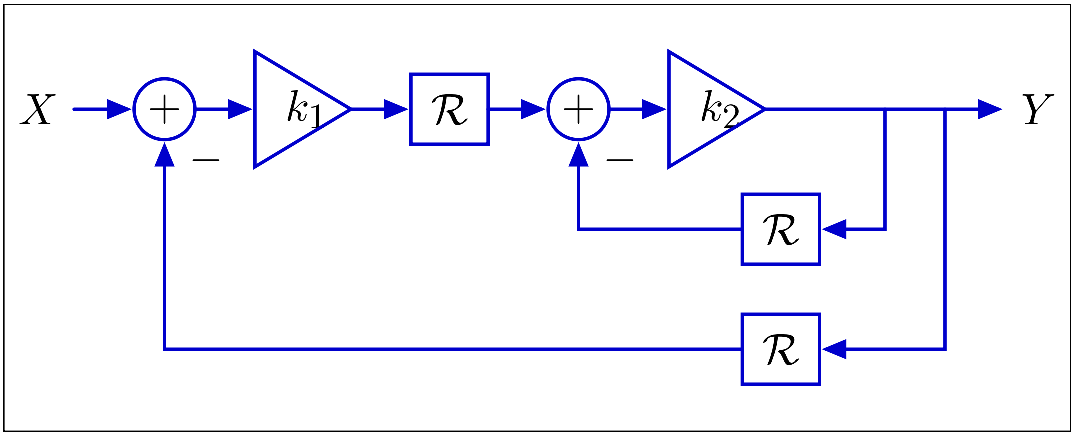
- $y[0] = -2x[-1] + 4y[-1] = 0 + 0 = 0$
- $y[1] = -2x[0] + 4y[-1] = -2 + 0 = -2$
- $y[2] = -2x[1] + 4y[0] = 0 + 0 = 0$
- $y[1] = -2x[2] + 4y[1] = 0 + 4 \cdot (-2) = -8$

Part c. Let $k_1 = 1$ and $k_2 = -2$, determine the poles of H.

Enter poles or **none** if there are no poles: 2, -2

For these k values, the denominator polynomial is $1 - 4\mathcal{R}^2$. So, we need to find the roots of the polynomial $z^2 - 4$, which are ± 2.

Part d. For each of the systems below indicate whether the system is equivalent to this one:



Equivalent to H (yes/no)? **No**

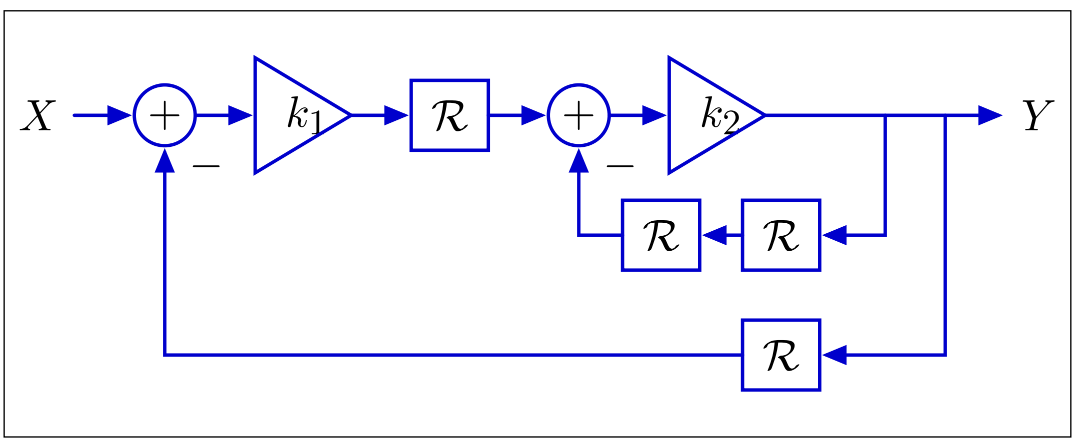
Let the output of the gain of k_1 be W. Then we can write the following set of operator equations:

$$W = k_1(X - \mathcal{R}Y)$$
$$Y = k_2(\mathcal{R}W - \mathcal{R}Y)$$

Eliminating W, we have:

$$
\begin{aligned}
Y &= k_2(\mathcal{R}W - \mathcal{R}Y) \\
&= k_2(\mathcal{R}k_1(X - \mathcal{R}Y) - \mathcal{R}Y) \\
&= k_1k_2\mathcal{R}X - k_1k_2\mathcal{R}^2Y - k_2\mathcal{R}Y
\end{aligned}
$$

which is not equal to the operator equation for the original system.



Equivalent to H (yes/no)? **Yes**

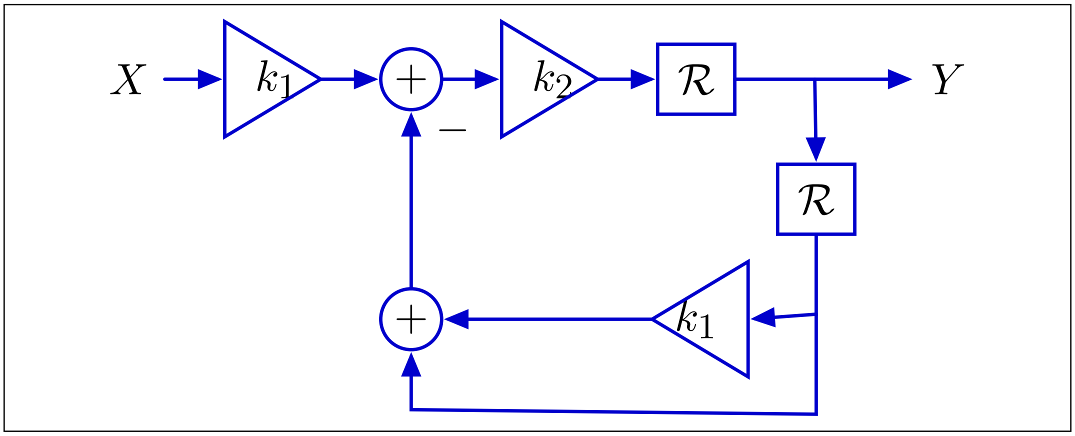
Let the output of the gain of k_1 be W. Then we can write the following set of operator equations:

$$W = k_1(X - \mathcal{R}Y)$$
$$Y = k_2(\mathcal{R}W - \mathcal{R}^2Y)$$

Eliminating W, we have:

$$
\begin{aligned}
Y &= k_2(\mathcal{R}W - \mathcal{R}^2Y) \\
&= k_2(\mathcal{R}k_1(X - \mathcal{R}Y) - \mathcal{R}^2Y) \\
&= k_1k_2\mathcal{R}X - k_1k_2\mathcal{R}^2Y - k_2\mathcal{R}^2Y
\end{aligned}
$$

which is equal to the operator equation for the original system.



Equivalent to H (yes/no)? **Yes**

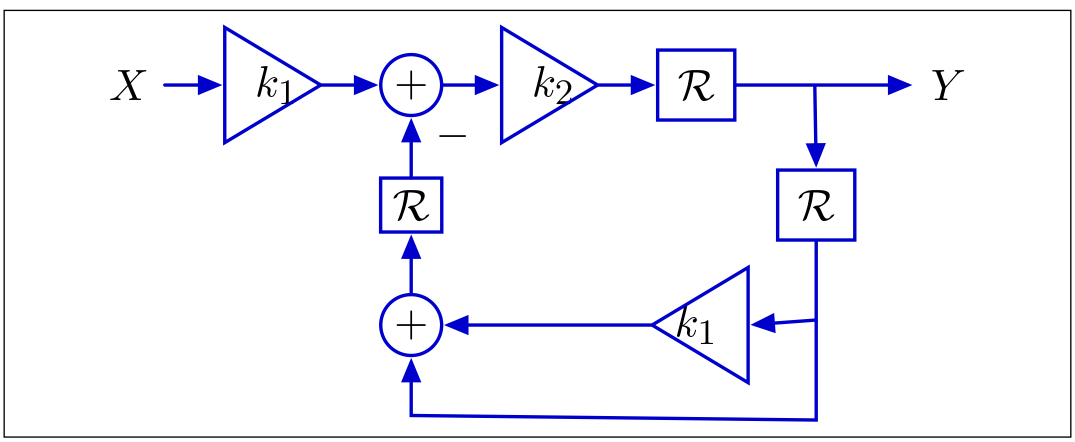
Let the output of the gain of k_2 be W. Then we can write the following set of operator equations:

$$
\begin{aligned}
W &= k_2(k_1X - (1 + k_1)\mathcal{R}Y) \\
Y &= \mathcal{R}W
\end{aligned}
$$

Eliminating W, we have:

$$
\begin{aligned}
Y &= \mathcal{R}k_2(k_1X - (1 + k_1)\mathcal{R}Y) \\
&= k_1k_2\mathcal{R}X - k_1k_2\mathcal{R}^2Y - k_2\mathcal{R}^2Y
\end{aligned}
$$

which is equal to the operator equation for the original system.



Equivalent to H (yes/no)? **No**

This is like the previous system, but with an extra delay in the feedback path, so it cannot be equivalent to the original system.

5.8.6 SM to DE

Here is the definition of a class of state machines:

```
class Thing(SM):
    startState = [0, 0, 0, 0]
    def getNextValues(self, state, inp):
        result = state[0] * 2 + state[2] * 3
        newState = [state[1], result, state[3], inp]
        return (newState, result)
```

1. What is the result of evaluating

```
Thing().transduce([1, 2, 0, 1, 3])
```

```
[0, 0, 3, 6, 6]
```

2. The state machine above describes the behavior of an LTI system starting at rest. Write a difference equation that describes the same system as the state machine.

$y[n] = 2y[n-2] + 3x[n-2]$

The important thing to see here is that the values in the state are $(y[n-2], y[n-1], x[n-2], x[n-1])$, so that the output is $2y[n-2] + 3x[n-2]$.

5.8.7 On the Verge

For each difference equation below, say whether, for a unit sample input signal:

- the output of the system it describes will diverge or not,
- the output of the system it describes (a) will always be positive, (b) will alternate between positive and negative, or (c) will have a different pattern of oscillation

1.

$$10y[n] - y[n-1] = 8x[n-3]$$

diverge? Yes or No | No

positive/alternate/oscillate | Positive

We first write the operator equation:

$$10Y - \mathcal{R}Y = 8\mathcal{R}^3 X$$

And the system function

$$\frac{Y}{X} = \frac{8\mathcal{R}^3}{10 - \mathcal{R}}$$

Find the root of the polynomial in $z = 1/\mathcal{R}$:

$$10z - 1 = 0$$
$$z = 0.1$$

The single pole is at 0.1. It is positive, so for a unit-sample input, the output will always be positive (assuming it starts at rest). It has magnitude less than 1, so it will converge.

2.

$$y[n] = -y[n-1] - 10y[n-2] + x[n]$$

diverge? Yes or No | Yes

positive/alternate/oscillate | Oscillates

We first write the operator equation:

$$Y + \mathcal{R}Y + 5\mathcal{R}^2 Y = X$$

And the system function

$$\frac{Y}{X} = \frac{1}{1 + \mathcal{R} + 10\mathcal{R}^2}$$

Find the roots of the polynomial in $z = 1/\mathcal{R}$:

$$Z^2 + Z + 10 = 0$$
$$Z = \frac{-1 \pm \sqrt{1 - 100}}{2}$$
$$Z = 0.5 \pm 4.97j$$

The magnitude of the poles is 5, which is greater than 1, so it diverges. The poles are complex, so the output will oscillate.

3.

$$y[n] = -0.6y[n-1] + .16y[n-2] - 0.1x[n-1]$$

diverge? Yes or No

No

positive/alternate/oscillate

Oscillates

We first write the operator equation:

$$Y + 0.6\mathcal{R}Y - .16\mathcal{R}^2Y = -0.1\mathcal{R}X$$

And the system function

$$\frac{Y}{X} = \frac{-0.1\mathcal{R}}{1 + 0.6\mathcal{R} - 0.16\mathcal{R}^2}$$

Find the roots of the polynomial in $z = 1/\mathcal{R}$:

$$Z^2 + 0.6Z - 0.16 = 0$$

$$Z = \frac{-0.6 \pm \sqrt{.36 + .64}}{2}$$

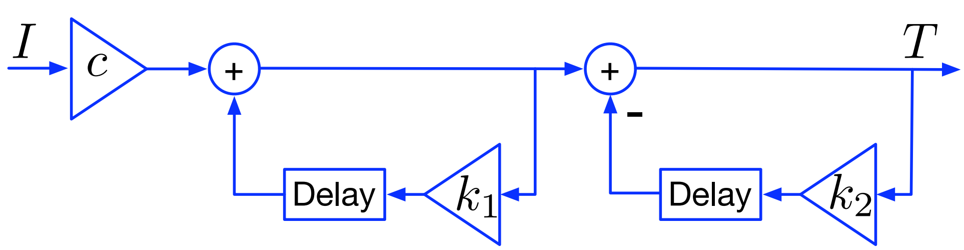
$$Z = (-0.8, 0.2)$$

The dominant pole is -0.8, because it has the largest magnitude. Its magnitude is less than 1, so the system will converge. The pole is negative, so the system will alternate positive and negative signs.

5.8.8 What's Cooking?

Sous vide cooking involves cooking food at a very precise, fixed temperature T (typically, low enough to keep it moist, but high enough to kill any pathogens). In this problem, we model the behavior of the heater and water bath used for such cooking. Let I be the current going into the heater, and c be the proportionality constant such that Ic is the rate of heat input.

The system is thus described by the following diagram:



1. **a.** Give the system function:

$$\frac{c}{(1 - k_1R)(1 + k_2R)}$$

If we name the signal coming out of the first adder W, then we have operator equations

$$W = cI + k_1\mathcal{R}W$$
$$T = W - k_2\mathcal{R}T$$

Solving, we get

$$W - k_1\mathcal{R}W = cI$$
$$W(1 - k_1\mathcal{R}) = cI$$
$$W = \frac{cI}{1 - k_1\mathcal{R}}$$
$$T = \frac{cI}{1 - k_1\mathcal{R}} - k_2\mathcal{R}T$$
$$T(1 - k_1\mathcal{R}) = cI - k_2\mathcal{R}T(1 - k_1\mathcal{R})$$
$$T - k_1\mathcal{R}T + k_2\mathcal{R}T - k_1k_2\mathcal{R}^2T = cI$$
$$T(1 - k_1\mathcal{R})(1 + k_2\mathcal{R}) = cI$$
$$\frac{T}{I} = \frac{c}{(1 - k_1\mathcal{R})(1 + k_2\mathcal{R})}$$

b. Give a difference equation for the system: $t[n] =$

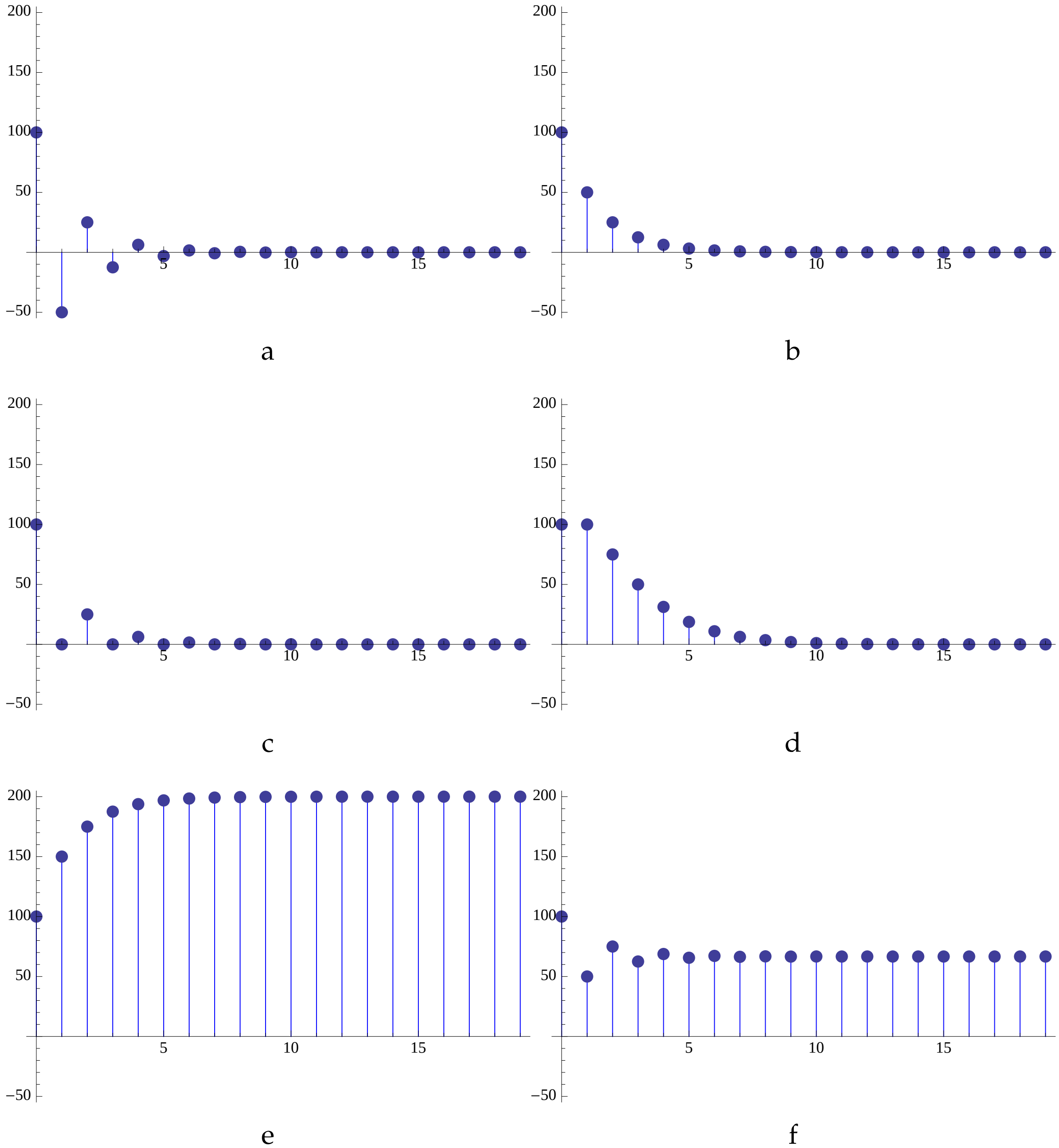
$$t[n] = (k_1 - k_2)t[n-1] + k_1k_2t[n-2] + ci[n]$$

Starting with this form of the operator equation, taken from the derivation above, and then rearranging terms

$$T(1 - k_1\mathcal{R}) = cI - k_2\mathcal{R}T(1 - k_1\mathcal{R})$$
$$T = cI + k_1\mathcal{R}T - k_2\mathcal{R}T(1 - k_1\mathcal{R})$$

We get an equation that's easy to convert to the difference equation above.

2. Let the system start at rest (all signals are zero). Suppose $I[0] = 100$ and $I[n] = 0$ for $n > 0$. Here are plots of $T[n]$ as a function of n for this system for $c = 1$ and different values of k_1 and k_2.



a. Which of the plots above corresponds to $k_1 = 0.5$ and $k_2 = 0$?

Circle all correct answers: a **(b)** c d e f **none**

The denominator as written above is already factored, and so we know that the poles are k_1 and $-k_2$. So, with these values for k_1 and k_2, there is a single pole at 0.5. So, we know that the system will converge monotonically, and that the each magnitude will be 0.5 of the magnitude on the previous step. The only plot that has this property is **b**.

b. Which of the plots above corresponds to $k_1 = 1$ and $k_2 = 0.5$?

Circle all correct answers: a b c d e **(f)** **none**

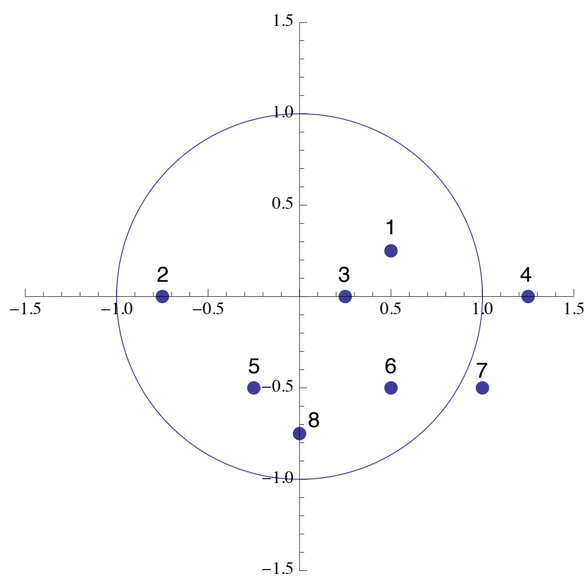
In this case, the poles are 1 and -0.5. With the dominant pole at 1, we expect the system to neither converge to 0, nor to diverge. The other pole at -0.5 will generate a component with alternating signs, but whose magnitude dies away over time. The graph in **f** shows a signal that is the sum of a long-term constant signal and a signal that is converging to zero with alternating signs.

3. Let $k_1 = 0.5$, $k_2 = 3$, and $c = 1$. Determine the poles of H, or **none** if there are no poles.

Looking at the factored form of the denominator, we can easily see that the poles are at k_1 and $-k_2$. If you didn't see that factored form, then you could explicitly see that the poles are roots of the equation $z^2 + (k_2 - k_1)z - k_1 k_2 = 0$, which in this case is $z^2 + 2.5z - 1.5 = 0$. We can use the quadratic formula to find that the roots are at 0.5 and −3.

5.8.9 Pole Position

Consider eight poles located at the following locations in the z plane. The plots below show the unit-sample responses of eight linear, time-invariant systems. Match them with the dominant pole for each system (remember that the system may have more than one pole).



A. This signal is alternating in sign and converging. Each magnitude is about 0.75 of the magnitude of the previous sample. So, we'd expect the dominant pole to be about −0.75, which corresponds to pole 2 on the plot.

B. This signal is converging monotonically; each sample is about 0.25 of the previous sample value. So, we expect a dominant pole of about 0.25. This corresponds to pole 3 on the plot.

C. This signal is diverging monotonically; each sample is about 1.25 of the previous sample value. So, we expect a dominant pole of about 1.25. This corresponds to pole 4 on the plot.

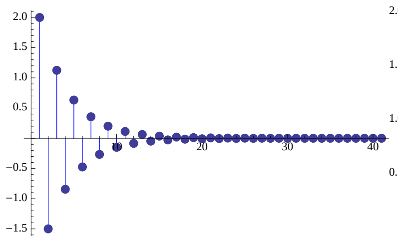
D. This signal is converging; it is neither monotonic, nor alternating in sign. It is oscillating with a period of 4, so we expect the dominant poles to be complex, with angle $\pm 2\pi/4 = \pm\pi/2$. We can see that the magnitude is about 0.3 of the previous magnitude after 4 steps, which means that the magnitude of the pole is about 0.75 (because $0.75^4 = 0.316$). Pole 8 has angle $-\pi/2$ and magnitude 0.75.

E. This signal is converging and oscillating. The period seems to be 8. So, we'd expect a pole at angle ±pi/4. The magnitude is a bit tricky to estimate. It seems to get from 2 to about 0.1 in 8 steps, so it's something like 0.7. That corresponds well to pole 6.

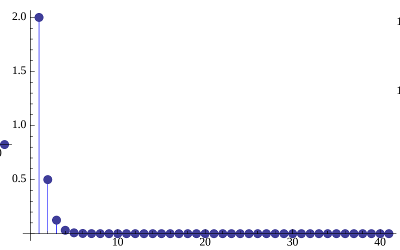
F. This signal is converging and oscillating. The period seems to be something like 3, and the magnitude even smaller than the previous two. Pole 5 has these characteristics.

G. This signal is diverging and oscillating. It seems to have a period of about 12, which would mean an angle of $\pm\pi/6$. The only pole on our picture with magnitude greater than 1 is at the correct angle, and so it must be 7.

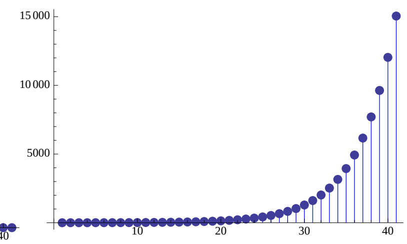
H. Finally, we have a signal that converges and oscillates. The period seems to be 10 or 12 and the rate fairly fast; pole 1 has a smaller magnitude than pole 6, and this converges faster than E, so this must be pole 1.



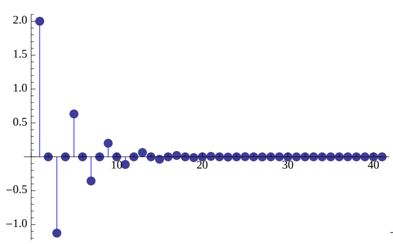
A = 2



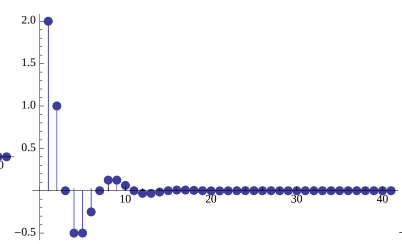
B = 3



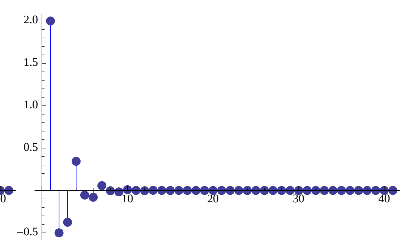
C = 4



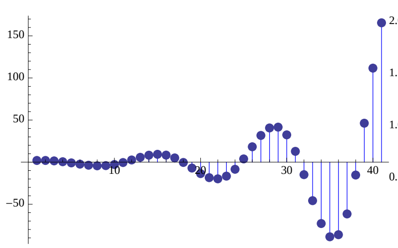
D = 8



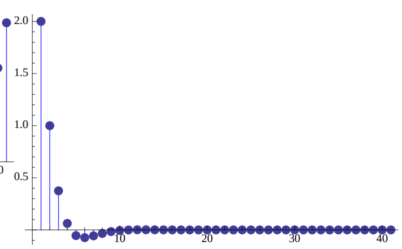
E = 6



F = 5



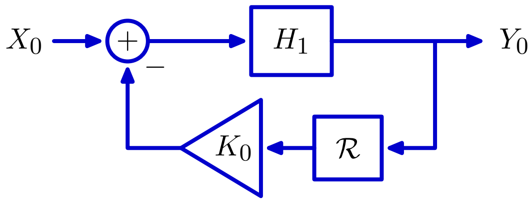
G = 7



H = 1

5.8.10 System functions

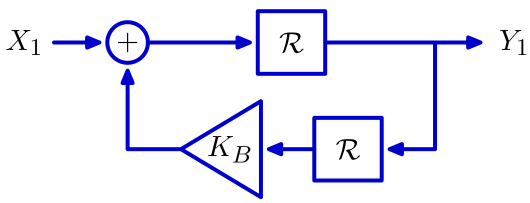
Let H_1 represent a subsystem that is part of the larger system shown below.



The system function for the larger system can be written as

$$H_0 = \frac{Y_0}{X_0} = \frac{H_1}{1 + K_0 R H_1} .$$

Assume that $H_1 = H_{1B} = \frac{Y_1}{X_1}$ as shown below.



What is the system function for H_{1B}?

$$H_{1B} = \frac{\mathcal{R}}{1 - K_B \mathcal{R}^2}$$

Determine the system function H_0 for the larger system when $H_1 = H_{1B}$.

$$H_0 = \frac{\mathcal{R}}{1 + (K_0 - K_B) R^2}$$

Under what conditions on K_0 and K_B is this system stable? Explain.

There are poles at $z = \pm\sqrt{K_B - K_0}$. To be stable, the poles should all have magnitued less than 1. Thus the system is stable if $|K_B - K_0| < 1$.

Under what conditions on K_0 and K_B does the unit-sample response decay monotonically? Explain.

None. For monotonic convergence, **both** poles must have magnitudes between 0 and 1 (since there are two poles of equal magnitude). If $K_B < K_0$ then the poles have non-zero imaginary parts, and the response oscillates. If $K_B > K_0$ then one pole is on the positive real axis and one is on the negative real axis. The pole on the negative real axis causes the unit sample response to alternate. Thus there are no values of K_O and K_B for which there is monotonic decay.