Part I

The Science of Audio The Mathematics of Audio

1

Math1

1.1 ANALOG AUDIO SYSTEM

The recording of audio, or the transmission of audio over a distance requires that the acoustical vibrations in the air be converted to electronic form.

Figure 1 shows an audio of video system which records an audio or video message, stores it in some manner or sends it to another location, and then reproduces the message. The system includes a human sender or message source, a transducer (mic, camera) to convert the message to electronic form, a channel to transmit the message across a distance, another transducer (speaker, display) to convert the message back to a form that can be sensed by humans (sound, light), and a human receiver or message sink.

```
Figure

message    sender                         receiver      message
source    transducer        channel       transducer      sink
       ^               ^              ^             ^

     sound          electrical    electrical      sound
     light           waveform      waveform       light
```

1.1.1 Message source

The message source may be in the form of sound (speech, music) or light (still or moving image).

1.1.2 Transducer at sender

The transducer at the sender converts the message to an electronic signal.

If the message is expressed as sound (speech,music) which can be sensed by the ears, the transducer is a microphone. The microphone contains a diaphragm or membrane which vibrates in step with the sound waves (Figure). The vibrations are used to power a kind of tiny electric generator in the form of a magnet and a moving coil of wire, which creates an electrical current in step with the vibrations. The precise details of how vibrations create electrical current are explained by basic principles of physics (Maxwells theory of electromagnetic fields). The output of the microphone transducer is a time-varying voltage (AC or alternating current) (Figure).

This signal can be represented mathematically as some function s(t), i.e. for every value of t on the horizontal axis of Figure , the signal has an amplitude (voltage or current) s(t). The graph of Figure specifies the function s(t), since for each value of t, we can read the corresponding value of s(t) from the graph.

If the message is light (still or moving image) which can be sensed by the eyes, the transducer is a camera. The camera contains a lens to focus and project the image onto a light-sensitive surface made of electronic semiconductor material (e.g. a charged coupled device or CCD) which converts light to electrical signals (Figure). Bright spots in the image will generate a stronger electrical signal at that point on the surface. The precise details of how light creates electrical current are explained by advanced principles of physics (Quantum Theory). The CCD may have several million light-sensitive points, arranged in a grid of rows and columns. To create a time-varying voltage, the projected image is scanned by reading the current at each point on the surface in a predefined order, e.g. by reading each row in turn from top to bottom. (Figure) Each row may be called a scanning line. A video camera scans about 30 images (frames) per second.

1.1.3 Transducer at receiver

The transducer at the sender converts the electronic signal back to the message form which can be sensed by a human.

If the original message is sound (voice, music), then the transducer is a speaker (or headphones). A speaker is a microphone in reverse, with a magnet, moving coil and diaphragm. The electrical signal powers a kind of electric motor which moves the diaphragm and thus creates sound waves.

If the original message is light (still or moving image), then the transducer is an electronic display device such as a LCD (liquid crystal display) or a CRT (cathode ray tube). The signal is scanned across the display, and causes light to emit from the display, with stronger signals generating brighter light, and thus recreating the image. The details of how the displays create light from electronic signals depends on the type of display. The display can also be a printer, where stronger signals will deposit more ink.

1.2 DIGITAL AUDIO SYSTEM

Digital audio systems convert the message into bits (ones and zeros) before transmitting it (or storing it). One major advantage of digital is potentially higher quality, which is preserved when the message is repeated or copied. This is why virtually all music is now stored on CD instead of vinyl records. Another advantage is that digital messages can be easily stored on a computer and transmitted over computer networks such as the Internet. Computer memory and disks store the bits in groups of 8 bits called bytes. A one-megabyte (Mbyte) memory chip stores 8 million bits. An audio CD can store 650 Mbytes or 5.2 billion bits. A third advantage is that different kinds of messages (e.g. audio, video, text) can be combined and transmitted together, since they are both converted to bits.

Figure x shows the one-way point-to-point system of Figure 1 again, but with two additional blocks at both the sender and receiver end. The analog-to-digital converter (A/D) block converts the analog electronic signal (time-varying voltage) to digital form (a sequence of binary numbers). The processor (formatter) block is used to modify and rearrange the sequence of numbers as described later.

```
Figure x

analog    sampler    quantizer    processor     D/A      analog
       ^          ^            ^            ^        ^
   waveform    numbers      bits         bits    waveform
```

1.2.1 A/D, sampling, quantization

The A/D includes two parts: a sampler which takes samples (measurements) of the analog signal voltage at regular intervals, and a quantizer which assigns a numerical value in binary form to each sample measurement. Thus the output of the A/D is a sequence of binary numbers. The process of converting the analog signal to numbers is called digitizing the signal or converting the signal to digital form.

We can write the sampling mathematically as $s[n] = s(t)|_{t=nT_s} = s(t = nT_s)$ where T_s is the sampling interval, and n is an integer called the time index. Note that we use square brackets [] for the time index n and round brackets () for the actual time t. Both n and t can be either positive, negative or zero. The sampled signal can be represented in a table with two rows: the bottom row is the integers $n = -2, -1, 0, 1, 2, 3$, etc and the top row is the numbers $s[n]$ representing the audio signal.

The time interval between samples T_s determines the number of samples taken per second which defines the sampling frequency f_s in cycles per second.

For an audio CD, this sampling frequency f_s is 44,100 samples per second or 44,100 Hz.

The numerical values of the samples can be expressed approximately using binary notation with only ones and zeros (Figure). The maximum number of bits used to represent these values determines the precision, or the number of possible values for each sample. For an audio CD, this is 16 bits, so there are 2^{16} or 65,536 possible values for each sample. If the signal voltage has a maximum of 1 volt and a minimum of -1 volts, then we have 65,536 steps between -1 and +1 (Figure). The size of one of these steps in volts is called the resolution or quantization step size. The maximum quantization error is one-half the quantization step size or 1/65,536 volts (Figure).

The number of samples per second multiplied by the number of bits per samples determine the total number of bits per second needed to represent the signal in digital form. For an audio CD, we have 44,100 samples/sec times 16 bits/sample = 708,000 bits per second. For two-cbannel stereo, we need 1.4 million bits/second (1.4 Mb/s).

The sampling frequency must be at least twice the highest frequency in the message signal. This is because we need at least two samples to represent one cycle of a sine wave (Figure). For music, the highest frequency of human hearing is around 20,000 Hz, so the CD sampling rate of 44,100 Hz is considered sufficient. The quantization error results in so-called quantization noise (Figure). With 16 bits, this quantization noise is virtually inaudible, and thus 16 bits is usually considered sufficient. Audiophiles would like more safety margin, so the new DVD audio standard is 96 KHz sampling and 24 bit resolution. *Exercise:* how many bits/sec are needed for 2 channels of DVD audio ? Answer: 4.608 Mbits/sec.

For digital telephony and digital cellphones, the sampling frequency is 8 KHz with 8 bit resolution, which is sufficient for acceptable voice quality (but the music-on-hold sounds bad). *Exercise:* how many bits/sec are needed for telephony ? Answer: 64 Kbits/sec.

1.2.2 numbers

At this point, we have shown that an analog message or analog electronic signal (speech, music, image, video) is converted to digital form, i.e. a sequence of numbers $x[n]$. For audio, there is one number per sample, and for video, there is one number per pixel. These numbers can be stored in computer memory, disk, CD or tape, or can be transmitted over a distance via the Internet or other means.

1.2.3 Processor

Once the audio signal $s(t)$ has been converted to numbers $s[n]$, it can be processed or modified easily just by changing the numbers in some specified manner. The numbers $s[n]$ are changed to different numbers $y[n]$. Adding reverberation is just one example of audio processing, low pass filtering (turning down the treble control) is another.

Figure x shows the processor with input $x[n]$ and output $y[n]$. In this section, we describe and specify how the numbers are changed to achieve specified effects. These effects can be heard or seen by converting the output numbers back to an analog electronic signal and connecting it to the receiver transducer (speaker, display).

The simplest effect is no effect at all, i.e. the output numbers are the same as the input numbers. In this case, the processor is represented by the input-output equation $y[n] = x[n]$. The next simplest effect is a delay by one time interval, i.e. $y[n] = x[n-1]$. For an audio system with sampling rate 44,100 Hz, a delay by one second or 44,100 samples is written $y[n] = x[n-44100]$. A very simple (and not very good) reverberator can be made by simply adding the signal to a delayed version of itself. Mathematically, we write $y[n] = x[n] + x[n-k]$, where k is the amount of delay. For a delay of say 100 millisec, $k = 4410$. Various kind of filters, including lowpass, highpass, bandpass and notch, can be built with input-output equation $y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$. The type of filter is determined by the values of the constants a_1, a_2, b_0, b_1, b_2 called the filter coefficients.

The input-output equations represent the function or effect carried out by the processor. In what follows, we will learn how to design filters and choose these constants.

We note here that the processor may also perform other functions which we call formatting. For example, to store the sampled audio an audio CD, the bit stream output of the A/D is processed as follows (Figure x). Segmentation divides the bit stream into segments or blocks. Source coding compresses the segment, i.e. reduces the number of bits in the segment by eliminating redundancy and irrelevant bits. An example of source coding is MP3 compression. Encryption may be done to protect the data from unauthorized copying. This is done for DVD video but not for CD audio. Channel coding is used to add some redundancy which can be used to detect errors, and in some cases also correct errors. On audio CD's the channel coding allows the CD to play even if there are some scratches.

```
A/D   segmentation   source coding    encryption   channel coding
                     (compression)               (error protection)
```

The output of the formatter is then stored on CD (or may be transmitted via the internet). The formatting operations are not considered further at this point.

1.2.4 System response

The processor is also referred to as a discrete-time system (sampled system, digital system), even though it is really only part of the more complete system of Figure . The output $y[n]$ is the *system response* to the input $x[n]$. If we know the input-output equations, we can predict the system response $y[n]$ to any input $x[n]$.

If we don't know the input-output equations, i.e. we have a black box processor with unknown function, then we can test the system to discover its function by feeding different input signals $x[n]$ and observing the output $y[n]$. In general, we would have to try many signals $x[n]$, and even then we could not necessarily predict what the response (output) would be without trying it.

However, for so-called linear, time-invariant systems, the function of the system (processor) is completely characterized by its response to one particular kind of input signal called a unit impulse function or delta function. The unit impulse function is like a sharp clap or click. We call this particular response the *impulse response* of the system. The impulse response can be used to predict the output with any input. Thus we need to try only one special input signal to completely understand the function of a linear time-invariant system. Almost all processors and systems used in practice are linear and time-invariant.

For example, consider the simple reverberator mentioned above with input-output equation $y[n] = x[n] + x[n-k]$ where the echo is delayed by k time units, If the input is a delta function, then the output or the impulse response is two claps (impulses) k time units apart (Figure). If the input is three rapid claps in succession, e.g. a triple of delta functions one time unit apart, then the output can be predicted to be two triples of delta functions separated by k time units (Figure).

1.2.4.1 Mathematical notation To more clearly explain discrete-time systems, and in particular, linear time-invariant systems, and how to design processors to implement functions we want, we need to introduce some mathematical notation.

A *linear* system is one where if input $x_1[n]$ yields output $y_1[n]$ and input $x_2[n]$ yields output $y_2[n]$, then with input $ax_1[n] + bx_2[n]$ for constants a and b, the output will be $ay_1[n] + by_2[n]$ (Figure).

A it time-invariant system is one where if input $x[n]$ yields output $y[n]$, then time-shifted input $x[n-k]$ delayed by k sample time units yields output $y[n-k]$ delayed by the same amount [1].

We define a special signal called the delta function or the impulse function which is written $\delta[n] = 1$ for $n = 0$ and 0 otherwise (Figure). A delayed delta function which is 1 for time $n = 1$ and 0 otherwise is written $\delta[n-1] = 1$

[1] if k is negative, then the input is advanced instead of delayed

for $n-1=0$ or $n=1$ and 0 otherwise (Figure). Similarly, a delayed delta function which is 1 for some arbitrary time $n=k$ is written $\delta[n-k]=1$ for $n-k=0$ or $n=k$ and 0 otherwise (Figure). The delta function is always 0 except when the argument inside the square brackets is 0, in which case it is 1.

We now make an important observation which will be a useful tool to show how signals are processed or modified by a linear time-invariant system. Each sample of the signal $x[n]$ can be written as delayed impulse. The sample of the signal $x[n]$ at time $n=k$ can be written $x[k]\delta[n-k]$, which is a delta function delayed by k and weighted (multiplied) by the value $x[k]$. This one sample of $x[n]$ at time $n=k$ can be thought of as a elementary kind of signal which has value $x[k]$ at time k and is zero everywhere else. and is written $x[k]\delta[n-k]$. This elementary signal has only one delta function at time k and nothing else.

Any signal $x[n]$ which is a collection of samples (numbers) can be written as the sum of weighted delayed delta functions. The time index n can take any value $-\infty < n < \infty$. The weights are the values $x[k]$ at time delay $n=k$, and the delayed delta functions are $\delta[n-k]$, which is equal to 1 at time delay $n=k$. In other words, any signal $x[n]$ can be written as the sum of many elementary signals with only one delta function in each. In general, we write

$$
\begin{aligned}
x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] && (1.1)\\
&= \ldots + x[-2]\delta[n-(-2)] + x[-1]\delta[n-(-1)] && (1.2)\\
&+ x[0]\delta[n-0] + x[1]\delta[n-1] && (1.3)\\
&+ x[2]\delta[n-2] + \ldots && (1.4)
\end{aligned}
$$

To show how the formula works, now evaluate $x[n]$ at a particular time $n=2$. By substituting $n=2$ everywhere, we find

$$
\begin{aligned}
x[2] &= \sum_{k=-\infty}^{\infty} x[k]\delta[2-k] && (1.5)\\
&= \ldots + x[-2]\delta[2-(-2)] + x[-1]\delta[2-(-1)] && (1.6)\\
&+ x[0]\delta[2-0] + x[1]\delta[2-1] && (1.7)\\
&+ x[2]\delta[2-2] + \ldots && (1.8)\\
&= \ldots + x[-2]\cdot 0 + x[-1]\cdot 0 && (1.9)\\
&+ x[0]\cdot 0 + x[1]\cdot 1 && (1.10)\\
&+ x[2]\cdot 1 + \ldots && (1.11)\\
&= x[2] && (1.12)\\
& && (1.13)
\end{aligned}
$$

For a particular value of $n=n_0$ ($n_0=2$ in the example above), the formula [?] reduces to a single term representing the sample value of the signal $x[n]$ at time n_0. This works for any value of n_0. This formula [?] may not seem very useful at first glance, but it is fundamental for all that follows. This formula [?] has the form of a *convolution sum* which will be explained in a following section, and which will occur frequently in all subsequent work.

To summarize, any signal $x[n]$ can be represented as the sum of impulse functions $\delta[n-k]$ delayed by k and weighted (multiplied) by $x[k]$.

1.2.4.2 Illustration of the sum of impulses formula In what follows, we show by example how this formula works. The notation may seem tedious, but necessary to understand the mathematics.

Recall that $\delta[n-k] = 1$ for $n = k$ and 0 otherwise. For a signal $x[n]$ which is equal to 1 for $n = 0$ and 0 otherwise, i.e. $x[n] = 1$ for $n = 0$ and $x[n] = 0$ for $n \neq 0$ $x[n] = \delta[n]$ (Figure a).

For a signal $x[n] = 1$ for $n = 1$ and $x[n] = 0$ for $n \neq 1$ $x[n] = \delta[n-1]$ (Figure b). For a signal $x[n] = 1$ for $n = k$ for some arbitrary integer k, and $x[n] = 0$ for $n \neq k$, $x[n] = \delta[n-k]$ (Figure c).

Consider an example signal $x[n] = 1$ for $n = 0, 1, 2$ and 0 otherwise (Figure),which looks like a square pulse. The signal can be written as a infinite vector $\mathbf{x}' = [...\ 0\ 1\ 1\ 1\ 0\ ...\ ]$ where the index vector $\mathbf{n}'$= [... -1 0 1 2 3 ...]. $\mathbf{x}$ and $\mathbf{n}$ are the equivalent column vectors. In practice, we consider only a finite range, and define $\mathbf{x}'$=[0 1 1 1 0] and $\mathbf{n}'$=[-1 0 1 2 3] or equivalently $\mathbf{n}'$=(-1:3). The figure is plotted in Matlab using the command plot(n,X,'bo') with a blue circle at each point. We can tabulate the signal as

n	$x[n]$
-1	0
0	1
1	1
2	1
3	0

For this signal $x[n]$, we can write a formula as follows. For $n = -1$ and $k = ..., -1, 0, 1, 2, 3, ...,$

$$
\begin{aligned}
x[-1] &= \sum_{k=-\infty}^{\infty} x[k]\delta[-1-k] && (1.14)\\
&= ... + x[-1]\delta[-1-(-1)] + x[0]\delta[-1-0] + x[1]\delta[-1-1] && (1.15)\\
&\quad +x[2]\delta[-1-2] + x[3]\delta[-1-3] + ... && (1.16)\\
&= ... + 0\cdot 1 + 1\cdot 0 + 1\cdot 0 + 1\cdot 0 + 0\cdot 0 + ... && (1.17)\\
&= ... + 0 + 0 + 0 + 0 + 0 + ... && (1.18)\\
&= 0 && (1.19)
\end{aligned}
$$

For $n = 0$ and $k = ..., -1, 0, 1, 2, 3, ...,$

$$\begin{aligned}
x[0] &= \sum_{k=-\infty}^{\infty} x[k]\delta[0-k] && (1.20)\\
&= ... + x[-1]\delta[0-(-1)] + x[0]\delta[0-0] + x[1]\delta[0-1] && (1.21)\\
&\quad + x[2]\delta[0-2] + x[3]\delta[0-3] + ... && (1.22)\\
&= ... + 0\cdot 0 + 1\cdot 1 + 1\cdot 0 + 1\cdot 0 + 0\cdot 0 + ... && (1.23)\\
&= ... + 0 + 1 + 0 + 0 + 0 + ... && (1.24)\\
&= 1 && (1.25)
\end{aligned}$$

For $n = 1$ and $k = ..., -1, 0, 1, 2, 3, ...,$

$$\begin{aligned}
x[1] &= \sum_{k=-\infty}^{\infty} x[k]\delta[1-k] && (1.26)\\
&= ... + x[-1]\delta[1-(-1)] + x[0]\delta[1-0] + x[1]\delta[1-1] && (1.27)\\
&\quad + x[2]\delta[1-2] + x[3]\delta[1-3] + ... && (1.28)\\
&= ... + 0\cdot 0 + 1\cdot 0 + 1\cdot 1 + 1\cdot 0 + 0\cdot 0 + ... && (1.29)\\
&= ... + 0 + 0 + 1 + 0 + 0 + ... && (1.30)\\
&= 1 && (1.31)
\end{aligned}$$

For $n = 2$ and $k = ..., -1, 0, 1, 2, 3, ...,$

$$\begin{aligned}
x[2] &= \sum_{k=-\infty}^{\infty} x[k]\delta[2-k] && (1.32)\\
&= ... + x[-1]\delta[2-(-1)] + x[0]\delta[2-0] + x[1]\delta[2-1] && (1.33)\\
&\quad + x[2]\delta[2-2] + x[3]\delta[2-3] + ... && (1.34)\\
&= ... + 0\cdot 0 + 1\cdot 0 + 1\cdot 0 + 1\cdot 1 + 0\cdot 0 + ... && (1.35)\\
&= ... + 0 + 0 + 0 + 1 + 0 + ... && (1.36)\\
&= 1 && (1.37)
\end{aligned}$$

For $n = 3$ and $k = ..., -1, 0, 1, 2, 3, ...,$

$$\begin{aligned}
x[3] &= \sum_{k=-\infty}^{\infty} x[k]\delta[3-k] && (1.38)\\
&= ... + x[-1]\delta[3-(-1)] + x[0]\delta[3-0] + x[1]\delta[3-1] && (1.39)\\
&\quad + x[2]\delta[3-2] + x[3]\delta[3-3] + ... && (1.40)\\
&= ... + 0\cdot 0 + 1\cdot 0 + 1\cdot 0 + 1\cdot 0 + 0\cdot 1 + ... && (1.41)\\
&= ... + 0 + 0 + 0 + 1 + 0 + ... && (1.42)\\
&= 0 && (1.43)
\end{aligned}$$

Thus $x[-1] = 0, x[0] = 1, x[1] = 1, x[2] = 1.x[3] = 0$ as expected.

Consider another example signal $x[n] = 1$ for all n (Figure), which is a constant level DC signal. For this signal, we can write

$$\begin{aligned} x[1] &= \sum_{k=-\infty}^{\infty} 1 \cdot \delta[1-k] && (1.44)\\ &= \sum_{k=-\infty}^{\infty} \delta[1-k] && (1.45)\\ &= \ldots + \delta[1+1] + \delta[1+0] + \delta[1-1] + \delta[1-2] + \ldots && (1.46)\\ &= +0+0+1+0 && (1.47)\\ &= 1 && (1.48) \end{aligned}$$

In general for any n

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} \delta[n-k] && (1.49)\\ &= \ldots + \delta[n+1] + \delta[n] + \delta[n-1] + \delta[n-2] + \ldots + \delta[n-k] + && (1.50)\\ &= +0+0+0+0+\ldots+1+\ldots && (1.51)\\ &= 1 && (1.52) \end{aligned}$$

for any value of n. In the last equation, for any value of n, all the delta functions are zero except the one for which $n = k$.

Consider a third example signal $x[n] = n$ for $n = 1, 2, 3$ and 0 otherwise (Figure) i.e. $x[1] = 1, x[2] = 2, x[3] = 3, x[n] = 0$ for all other n We can tabulate this signal as

n	$x[n]$
-1	0
0	0
1	1
2	2
3	3
4	0

This signal $x[n]$ can also be written as a vector $\mathbf{x} = [\mathbf{01230}]$, where the first element of the vector is $x[0]$.

This signal can also be written in terms of weighted delayed impulses as

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} x[k] && (1.53)\\ &= \sum_{k=1}^{3} k\delta[n-k] && (1.54)\\ &= \ldots + 0\delta[n-0] + 1\delta[n-1] + 2\delta[n-2] + 3\delta[n-3] + 0\delta[n-4]\ldots && (1.55)\\ &= 1 \text{ for } n = 1, 2 \text{ for } n = 2, 3 \text{ for } n = 3, 0 \text{ otherwise} && (1.56) \end{aligned}$$

For a general signal $x[n]$ for any n where the values of $x[n]$ may be arbitrary,

$$\begin{aligned}
x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] && (1.57)\\
&= \ldots + x[-2]\delta[n-(-2)] + x[-1]\delta[n-(-1)] && (1.58)\\
&+ \; x[0]\delta[n-0] + x[1]\delta[n-1] && (1.59)\\
&+ \; x[2]\delta[n-2] + \ldots && (1.60)
\end{aligned}$$

Thus any signal $x[n]$ can be represented as the sum of impulse functions $\delta[n-k]$ delayed by k and weighted (multiplied) by $x[k]$. A signal $x[n]$ can also be written as a vector

1.2.4.3 Impulse response We can now use this notation to prove a most important result that lies at the foundation of signal processing: that for linear, time-invariant (LTI) systems, the function of the system (processor) is completely characterized by its response to a unit impulse function or delta function.

If the input $x[n]$ is a single impulse function $\delta[n]$ which yields output (impulse response) $y[n] = h[n]$, then from the rules for a *time-invariant* system, the delayed input $x[n-k = \delta[n-k]$ yields the identical but delayed output $y[n-k] = h[n-k]$. In other words, if we have a system such that for input

$$x[n] = \delta[n] \tag{1.61}$$

the output

$$y[n] = h[n] \tag{1.62}$$

then from the rules for a *time-invariant* system, for delayed input

$$x[n-k] = \delta[n-k] \tag{1.63}$$

the delayed output

$$y[n-k] = h[n-k] \tag{1.64}$$

Thus for a time-invariant system, a delayed input results in a delayed, but otherwise unchanged, output.

If the input $x[n]$ is written as one weighted delayed impulse function, i.e.

$$x[n] = x[k]\delta[n-k] \tag{1.65}$$

then from the rules for a *linear* system, the output

$$y[n] = x[k]h[n-k] \tag{1.66}$$

Thus the output $y[n]$ is a delayed impulse response weighted by $x[k]$.

For systems which are both time-invariant and linear, we can combine these results. If the input $x[n]$ is written as the sum of many weighted delayed impulse functions [2], i.e.

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{1.67}$$

then from the combined rules for *linear* and *time-invariant* system, the output

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{1.68}$$

Thus the output $y[n]$ is a sum of delayed impulse responses weighted by the input samples $x[k]$.

Because the system is both linear and time-invariant, the impulse response $h[k]$ is unchanged if the input signal is scaled [3] or delayed, except for scaling and delay. Thus we can predict the response $y[k]$ of an LTI system to any signal $x[k]$, just by knowing the response $h[k]$ of the system to a single impulse $\delta[k]$. *The impulse response uniquely represents the behaviour of an LTI system. This is the foundation of all signal processing with LTI systems.*

Since the system can be represented by its impulse response $h[k]$, we prefer to make a change of variable and write the output [4]

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{1.69}$$

Thus the output $y[n]$ of any LTI system is the sum of delayed inputs $x[n-k]$ weighted by the impulse response $h[k]$. In other words, the output $y[n]$ is a sum of products obtained by taking each delayed (past) input $x[n-k]$, multiplying it by by $h[k]$, and adding them all up.

This sum of products calculations is a fundamental operation which occurs everywhere in digital signal processing, and special hardware (DSP chips) are designed to efficiently compute sums of products.

The expression for $y[n]$ is the so-called *convolution* of $x[n]$ with $h[n]$. In practice, to compute $y[n]$, the summation limits have to be finite, i.e. $h[k] = 0$ for $k < k_0$ and $k > k_1$ for finite constants k_0, k_1, i.e.

$$y[n] = \sum_{k=k_0}^{k_1} h[k]x[n-k] \tag{1.70}$$

[2] This is where we need the notation derived in the previous section

[3] We use the word *weighting* for multiplying a single sample by some weighting constant (called a weight) and *scaling* for multiplying an entire signal by some scaling constant (also called scaling factor).

[4] This change of variable is made by letting $m = n - k$, so that $k = n - m$ and we write [?] as $y[n] = \sum_{m=-\infty}^{\infty} x[n-m]h[m]$ and then relabelling $m = k$.

An example with $h[k] = 0$ for $k_0 = 0$ and $k_1 = 2$, the output

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] && (1.71) \\ &= \sum_{k=0}^{2} h[k]x[n-k] && (1.72) \\ &= h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] && (1.73) \end{aligned}$$

The sum goes from $k = 0$ to 2 because $h[k] = 0$ for all other values of k, and so those terms drop out.

1.2.5 Terminology

We now introduce some terminology, followed by examples.

For a *causal* system, the output for any time $n = n_0$ depends only on the inputs for earlier times $n \leq n_0$, i.e. depends only on the past, not on the future. For this case $h[k] = 0$ for any $k < 0$.

For a *(causal) finite impulse response (FIR)* system, $h[k] = 0$ for $k > N$ where N is some finite number. For an *infinite impulse response (IIR)* system, $h[k] \neq 0$ for arbitrarily large k. A *bounded* signal has a finite value for all times k. [5] For a *stable* system, a bounded input signal $x[n]$ yields a bounded output signal $y[n]$,

$$\sum_{k=-\infty}^{\infty} |h[k]| < \infty \qquad (1.74)$$

Thus the example system given above

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] \qquad (1.75)$$

is a causal stable FIR system. Examples of non-causal, unstable and IIR systems will be given later.

1.2.5.1 Example - impulse reponse of FIR filter Consider the FIR system

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] \qquad (1.76)$$

with $h[0] = 1, h[1] = l, h[2] = -1$.

k	$h[k]$
-1	0
0	1
1	3
2	-1
3	0

[5] whereas an *unbounded* signal has an infinite value for at least one or more samples.

We show how to compute the output with specified input. In this example, the problem statement is: given input $x[n] = \delta[n]$, find the output $y[n]$. We will find that $y[n] = h[n]$, the impulse response, as expected. The solution may appear long and tedious, but is worth going through once, and then easily programmed on a computer (and particularly easy in Matlab).

Solution

In general

$$\begin{aligned} y[n] &= \sum_{k=0}^{2} h[k]x[n-k] && (1.77) \\ &= h[0]x[n-0] + h[1]x[n-1] + h[2]x[n-2] && (1.78) \end{aligned}$$

To find $y[n]$, the method of solution is to write a separate expression for $y[n]$ for each value of n.

First consider $n = 0$ and $k = 0, 1, 2$

$$\begin{aligned} y[0] &= \sum_{k=0}^{2} h[k]x[0-k] && && (1.79) \\ &= h[0]x[0-0] && (n = 0, k = 0) && (1.80) \\ &+ h[1]x[0-1] && (n = 0, k = 1) && (1.81) \\ &+ h[2]x[0-2] && (n = 0, k = 2) && (1.82) \\ &= h[0]x[0] + h[1]x[-1] + h[2]x[-2] && && (1.83) \end{aligned}$$

Since $x[k] = \delta[k]$, we know that $x[k] = 0$ for all $k \neq 0$ and $x[k] = 1$ for $k = 0$. Thus for $n = 0$,

$$\begin{aligned} y[0] &= \sum_{k=0}^{2} h[k]x[0-k] && (1.84) \\ &= h[0]x[0] = 1 \cdot x[0] = 1 \cdot 1 = 1 && (1.85) \end{aligned}$$

Only one term in the sum is non-zero, because $x[k] = \delta[k]$.

Similarly, consider $n = 1$

$$\begin{aligned} y[1] &= \sum_{k=0}^{2} h[k]x[1-k] && (1.86) \\ &= h[0]x[1-0] + h[1]x[1-1] + h[2]x[1-2] && (1.87) \\ &= h[0]x[1] + h[1]x[0] + h[2]x[-1] && (1.88) \\ &= h[1]x[0] = h[1]\delta[0] = 3 \cdot 1 = 3 && (1.89) \end{aligned}$$

Similarly, consider $n = 2$

$$y[2] = \sum_{k=0}^{2} h[k]x[2-k] \qquad (1.90)$$

$$\begin{aligned} &= h[0]x[2-0] + h[1]x[2-1] + h[2]x[2-2] && (1.91)\\ &= h[0]x[2] + h[1]x[1] + h[2]x[0] && (1.92)\\ &= h[2]x[0] = 1 \cdot (-1) = -1 && (1.93) \end{aligned}$$

For all other values of $n \neq 0, 1, 2$, the sum will be 0.

The final result is

$$y[0] = 1, y[1] = 3, y[2] = -1, y[k \neq 0, 1, 2] = 0 \qquad (1.94)$$

Recall that $h[0] = 1, h[1] = 3, h[2] = -1$, and $x[k] = \delta[k]$, so that $y[k] = h[k]$ as expected.

Figure x shows a diagram of this system with input $x[n]$, unit delay elements, gain elements $h[0], h[1], h[[2]$, a summer and output $y[n]$.

1.2.6 FIR filter with arbitrary input

Now we can compute the output $y[n]$ for any arbitrary input signal $x[n]$. For example, consider $x[0] = 1$, $x[1] = 3$ and $x[n] = 0$ for all other $n \neq 1, 3$ (Figure), and $h[0] = 1, h[1] = 2, h[2] = -1$ as before.

k	$x[k]$	$h[k]$
0	1	1
1	3	2
2	0	-1
3	0	0

We compute $y[n]$ two ways, using algebra as in the previous example, and again by looking at the system diagram.

In general, given the input $x[n]$ and the impulse response $h[n]$, we calculate the output for any n using the *convolution* sum

$$y[n] = \sum_{k=0}^{2} h[k]x[n-k] \qquad (1.95)$$

For causal systems, we start by considering $n = 0$

$$\begin{aligned} y[0] &= \sum_{k=0}^{2} h[k]x[0-k] && (1.96)\\ &= h[0]x[0] + h[1]x[-1] + h[2]x[-2] && (1.97)\\ &= h[0]x[0] = 1 \cdot x[0] = 1 \cdot 1 = 1 && (1.98) \end{aligned}$$

Similarly, consider $n = 1$

$$\begin{aligned} y[1] &= \sum_{k=0}^{2} h[k]x[1-k] && (1.99)\\ &= h[0]x[1-0] + h[1]x[1-1] + h[2]x[1-2] && (1.100) \end{aligned}$$

$$\begin{aligned}
&= h[0]x[1]+h[1]x[0]+h[2]x[-1] && (1.101)\\
&= 1\cdot 3+2\cdot 1+-1\cdot 0 && (1.102)\\
&= 3+2+0=5 && (1.103)
\end{aligned}$$

Similarly, consider $n = 2$

$$\begin{aligned}
y[2] &= \sum_{k=0}^{2} h[k]x[2-k] && (1.104)\\
&= h[0]x[2-0]+h[1]x[2-1]+h[2]x[2-2] && (1.105)\\
&= h[0]x[2]+h[1]x[1]+h[2]x[0] && (1.106)\\
&= 1\cdot 0+2\cdot 3+-1\cdot 1 && (1.107)\\
&= 0+6-1=5 && (1.108)
\end{aligned}$$

Similarly, consider $n = 3$

$$\begin{aligned}
y[3] &= \sum_{k=0}^{2} h[k]x[3-k] && (1.109)\\
&= h[0]x[3-0]+h[1]x[3-1]+h[2]x[3-2] && (1.110)\\
&= h[0]x[3]+h[1]x[2]+h[2]x[1] && (1.111)\\
&= 1\cdot 0+2\cdot 0+-1\cdot 3 && (1.112)\\
&= 0+0-3=-3 && (1.113)
\end{aligned}$$

Similarly, consider $n = 4$

$$\begin{aligned}
y[4] &= \sum_{k=0}^{2} h[k]x[4-k] && (1.114)\\
&= h[0]x[4-0]+h[1]x[4-1]+h[2]x[4-2] && (1.115)\\
&= h[0]x[4]+h[1]x[3]+h[2]x[2] && (1.116)\\
&= 1\cdot 0+2\cdot 0+-1\cdot 0 && (1.117)\\
&= 0+0-0=-0 && (1.118)
\end{aligned}$$

The output $y[n]$ may be written as a vector $\mathbf{y}' = [\mathbf{1\ 5\ 5\ -3}]$ For $n \geq 3$ and $n \leq -1$, $y[n] = 0$.

Using the system diagram, we write the input $x[n]$ as a vector $\mathbf{x}'$=[1 3], the impulse response $h[n]$ as a vector $\mathbf{h}'$=[1 2 -1]. The output $y[n]$ may be written as a vector $\mathbf{y}'$ = [1 5 5 -3]. The convolution is computed by sliding the input across the top with the most recent sample first (i.e. the vector is written in reverse order with the lowest index $n = 0$ on the right most side, and higher indices to the left of that). The numbers which are vertically adjacent are multiplied, and the results of these multiplications are added.

Figure shows the convolution calculation for each value of $n = 0, 1, 2, 3, 4$ corresponding to the equations above. $\mathbf{x}' = [\mathbf{1\ 3}]$ is reversed with the lowest index first and slides in from the left. Thus older samples (lower time index) of $\mathbf{x}'$ and $\mathbf{y}'$ are to the right, i.e. time *decreases* (time index gets smaller) from left to right. [6]

$\mathbf{x}'$	3	1	0	0	0	0	0		
$\mathbf{h}'$			1	2	-1				
			0	0	0				
$\mathbf{y}'$				0					

$\mathbf{x}'$	0	3	1	0	0	0	0		
$\mathbf{h}'$			1	2	-1				
			1	0	0				
$\mathbf{y}'$				1	0				

$\mathbf{x}'$	0	0	3	1	0	0	0		
$\mathbf{h}'$			1	2	-1				
			3	2	0				
$\mathbf{y}'$				5	1	0			

$\mathbf{x}'$	0	0	0	3	1	0	0		
$\mathbf{h}'$			1	2	-1				
			0	6	-1				
$\mathbf{y}'$				5	5	1	0		

$\mathbf{x}'$	0	0	0	0	3	1	0		
$\mathbf{h}'$			1	2	-1				
			0	0	-3				
$\mathbf{y}'$				-3	5	5	1	0	

$\mathbf{x}'$	0	0	0	0	0	3	1		
$\mathbf{h}'$			1	2	0				
			0	0	0				
$\mathbf{y}'$				0	-3	5	5	1	0

1.2.7 Difference equations

We now have two alternative ways to represent a system: the input-output equations relating $y[n]$ with $x[n]$, and the impulse response. The input-output equations are also referred to as the *difference equations*.

[6] These convolution diagrams can also be drawn the other way with time *increasing* from left to right. In this case, the signal slides from right to left, and the impulse response $h[k]$ is time-reversed, i.e. $h[2]h[1]h[0]$.

Given the difference equations, it is straightforward to find the impulse response, by selecting the input $x[n]$ to be $\delta[n]$ and computing the output $y[n]$ which will be the impulse response $h[n]$.

1.2.7.1 FIR, no feedback Example: consider the same FIR system as before

$$y[n] = x[n] + 2x[n-1] - x[n-2] \tag{1.119}$$

The impulse response is found by setting $x[n] = \delta[n]$

$$h[n] = \delta[n] + 2\delta[n-1] - \delta[n-2] \tag{1.120}$$

so that $h[0] = 1, h[1] = 2, h[2] = -1$. We can write $h[n]$ as a vector $\mathbf{h}' = [\mathbf{1}\ \mathbf{2}\ -\mathbf{1}]$. This is a *finite impulse response* (FIR) system.

We can also set $x[n] = \delta[n]$ and compute the output $y[n] = h[n]$ for each n using the difference equation as follows. For $n = 0$

$$\begin{aligned}
y[n] &= x[n-0] + 2x[n-1] - x[n-2] && (1.121)\\
y[-1] &= x[-1-0] + 2x[-1-1] - x[-1-2] && (1.122)\\
&= x[-1] + 2x[-2] - x[-3] && (1.123)\\
&= \delta[-1] + 2\delta[-2] - \delta[-3] && (1.124)\\
&= 0 + 0 + 0 = 0 && (1.125)\\
y[0] &= x[0-0] + 2x[0-1] - x[0-2] && (1.126)\\
&= x[0] + 2x[-1] - x[-2] && (1.127)\\
&= \delta[0] + 2\delta[-1] - \delta[-2] && (1.128)\\
&= 1 + 0 + 0 = 1 && (1.129)\\
y[1] &= x[1-0] + 2x[1-1] - x[1-2] && (1.130)\\
&= x[1] + 2x[0] - x[-1] && (1.131)\\
&= \delta[1] + 2\delta[0] - \delta[-1] && (1.132)\\
&= 0 + 2 + 0 = 2 && (1.133)\\
y[2] &= x[2-0] + 2x[2-1] - x[2-2] && (1.134)\\
&= x[2] + 2x[1] - x[0] && (1.135)\\
&= \delta[2] + 2\delta[1] - \delta[0] && (1.136)\\
&= 0 + 0 - 1 = -1 && (1.137)\\
y[3] &= x[3-0] + 2x[3-1] - x[3-2] && (1.138)\\
&= x[3] + 2x[2] - x[1] && (1.139)\\
&= \delta[3] + 2\delta[2] - \delta[1] && (1.140)\\
&= 0 + 0 - 0 = 0 && (1.141)\\
& && (1.142)
\end{aligned}$$

We could also find $h[n]$ by inspection, since the difference equation

$$y[n] = x[n] + 2x[n-1] - x[n-2] \tag{1.143}$$

is a special case of the convolution sum

$$y[n] = \sum_{k=0}^{2} h[k]x[n-k] \tag{1.144}$$

with $h[0] = 1, h[1] = 2, h[2] = -1$ and $h[k] = 0$ for $k \neq 0, 1, 2$.

Thus we observe that given the impulse response $h[n]$ of finite length (i.e a finite impulse response filter), the difference equations are the same thing as the convolution sum.

1.2.7.2 IIR, feedback The convolution sum is a kind of difference equation which computes the output $y[n]$ as the sum of weighted and delayed *inputs* $x[n]$, i.e. the sum of the current and past inputs.

Another kind of difference equation computes the output as the sum of past *outputs* $\mathbf{y}$ as well as current and past inputs $\mathbf{x}$. For example

$$y[n] = ay[n-1] + bx[n] \tag{1.145}$$

or

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \tag{1.146}$$

In general,

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] \tag{1.147}$$

$$+ \sum_{k=0}^{M} b_k x[n-k] \tag{1.148}$$

which is a convolution sum (using b_k instead of h_k) plus extra feedback terms using a_k.

Note that we could equally well use the subscript notation a_k or the index notation $a[k]$, since they are equivalent. If $a_k = 0$ for all $k >= 0$, then we have the convolution sum and no feedback.

Given the difference equation

$$y[n] = ay[n-1] + bx[n] \tag{1.149}$$

we can draw a system diagram as in Figure . Note that the past output $y[n-1]$ is weighted by factor a and fed back and added to the weighted current input $x[n]$. The presence of feedback makes this a *recursive* system [7], for which the current output $y[n]$ depends on the previous output $y[n-1]$ as well as the current input $x[n]$.

[7] so named because the output recurs (again) in the input

We can compute the impulse response by choosing $x[n] = \delta[n]$, and computing the output $y[n]$ for each n. We will find that this (and any) feedback system has an infinite impulse response which is non zero for any arbitrarily large value of n. We assume a causal system, so that outputs $y[n]$ before the start time $n = 0$ are set to 0, i.e. $y[n < 0] = 0$.

First consider $n = 0$

$$\begin{aligned} y[0] &= ay[0-1] + bx[0] && (1.150) \\ &= ay[-1] + b\delta[0] && (1.151) \\ &= 0 + b \cdot 1 && (1.152) \\ &= b && (1.153) \end{aligned}$$

For $n = 1$,

$$\begin{aligned} y[1] &= ay[1-1] + bx[1] && (1.154) \\ &= ay[0] + b\delta[1] && (1.155) \\ &= a \cdot b + b \cdot 0 && (1.156) \\ &= ab && (1.157) \end{aligned}$$

For $n = 2$,

$$\begin{aligned} y[2] &= ay[2-1] + bx[2] && (1.158) \\ &= ay[1] + b\delta[2] && (1.159) \\ &= a \cdot ab + b \cdot 0 && (1.160) \\ &= a^2 b && (1.161) \end{aligned}$$

For $n = 3$,

$$\begin{aligned} y[3] &= ay[3-1] + bx[3] && (1.162) \\ &= ay[2] + b\delta[3] && (1.163) \\ &= a \cdot a^2 b + b \cdot 0 && (1.164) \\ &= a^3 b && (1.165) \end{aligned}$$

Since the input $x[n] = \delta[n]$, the output $y[n]$ is the impulse response $h[n]$. From inspection of the computations above,

$$h[n] = a^n b \qquad (1.166)$$

or in vector notation

$$\mathbf{h}' = [\mathbf{b}\ \mathbf{ab}\ \mathbf{a^2b}\ \mathbf{a^3b}...] \qquad (1.167)$$

The system has an infinite impulse response. The 'shape' of this impulse response (i.e. the appearance of a plot of $h[n]$ versus n depends on the numerical value of a. There are 7 possibilities: $a < -1, a = -1, -1 < a < 0, a = 0, 0 < a < 1, a = 1, a > 1$ (Figure).

1.2.7.3 FIR and IIR systems with Matlab The general IIR system

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] \tag{1.168}$$

$$+ \sum_{k=0}^{M} b_k x[n-k] \tag{1.169}$$

If all the a_k's are zero ($a_k = 0$ for all $k > 0$), then the system is FIR.

The IIR system may also be written

$$y[n] - \sum_{k=1}^{N} a_k y[n-k] \tag{1.170}$$

$$= \sum_{k=0}^{M} b_k x[n-k] \tag{1.171}$$

or

$$\sum_{k=0}^{N} \alpha_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{1.172}$$

where $\alpha_0 = 1$ and $\alpha_k = -a_k$.

In Matlab, the filter function can be used to compute the output $y[n]$ given the input $x[n]$ as follows

$$\mathbf{y} = \mathbf{filter}[\mathbf{b}, \alpha, \mathbf{x}] \tag{1.173}$$

where $\mathbf{b} = [\mathbf{b_0} \quad \mathbf{b_1} \quad \mathbf{b_2} \quad ...], \alpha = [\mathbf{1} \quad -\mathbf{a_1} \quad -\mathbf{a_2} \quad ...]$. Note that we use $\alpha_k = -a_k$ and not the a_k's from the difference equation 2.70. Also note that in Matlab, the vector indices start at 1 (not 0). For an FIR system

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{1.174}$$

the Matlab filter function is

$$\mathbf{y} = \mathbf{filter}[\mathbf{b}, \mathbf{1}, \mathbf{x}] \tag{1.175}$$

Since for an FIR filter, the filter coefficients $b_k = h_k$ are the same as the impulse response, we can also write

$$\mathbf{y} = \mathbf{filter}[\mathbf{h}, \mathbf{1}, \mathbf{x}] \tag{1.176}$$

Examples difference equations and corresponding matlab filters:

$$y[n] = x[n], \mathbf{b} = [\mathbf{1}]\alpha = [\mathbf{1}] \tag{1.177}$$

so that in this case $\mathbf{b}$ and α are vectors with only one element.

$$y[n] = x[n] + b_1 x[n-1] \tag{1.178}$$
$$\mathbf{b} = [\mathbf{1} \quad \mathbf{b_1}] \alpha = [\mathbf{1}] \tag{1.179}$$

$$y[n] = a_1 y[n-1] + b_0 x[n] \tag{1.180}$$
$$\mathbf{b} = [\mathbf{b_0}] \alpha = [\mathbf{1} \quad -\mathbf{a_1}] \tag{1.181}$$

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1] \tag{1.182}$$
$$\mathbf{b} = [\mathbf{b_0} \quad \mathbf{b_1}] \alpha = [\mathbf{1} \quad -\mathbf{a_1}] \tag{1.183}$$

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \tag{1.184}$$
$$\mathbf{b} = [\mathbf{b_0} \quad \mathbf{b_1} \quad \mathbf{b_2}] \alpha = [\mathbf{1} \quad -\mathbf{a_1} \quad -\mathbf{a_2}] \tag{1.185}$$

1.2.7.4 Difference equations, differential equations and transforms The difference equations can always be solved for any input $x[n]$ iteratively by considering $n = 0$, $n = 1$, etc. in turn and computing $y[n]$ for each n.

The difference equations could also be solved in two steps, first by setting the input $x[n] = \delta[n]$ and thus finding the impulse response $h[n]$, and then convolving the input $x[n]$ with $h[n]$ to get $y[n]$.

Recall the following items.

- Differential equations can be solved using the Laplace transform.
- Difference equations are a discrete-time version of the continuous-time differential equations.

In the next chapter, we will show how difference equations can be solved using the discrete time version of the Laplace transform, called the z-transform.

2

Z transform

2.1 INTRODUCTION

We introduce the z-transform as a tool to solve LTI system difference equations of the types (both FIR and IIR) in the previous chapter. The z-transform also yields insights into the frequency domain behaviour of LTI systems.

The z-transform transforms a sequence of numbers $x[n]$ (or equivalently, a vector $\mathbf{x}$) into a function $X(z)$ of a complex variable z. The z=transform converts difference equations in the time domain to algebraic equations in the z domain.

Similarly, the Laplace transform converts a continuous function of time $x(t)$ into another function of a complex variable s. The Laplace transform converts differential equations in the time domain to algebraic equations in the s domain.

The z-transform is essentially a 'sampled' Laplace transform, where $z = e^{sT}$ and T is the time between samples (the sampling frequency $f_s = 1/T$). To show this, given a continuous time (analog) signal $x(t)$ and it's Laplace transform

$$X(s) = \int_{t=-\infty}^{\infty} x(t)e^{-st}dt \tag{2.1}$$

[1] and substituting $z = e^{sT}$ and sampling $x(t)$ at times $t = kT$ to obtain samples $x[k] = x(t = kT)$ [2] the z-transform is written

$$X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \tag{2.2}$$

where the Laplace integral becomes a sum. [3] The details are are follows. From the previous chapter, we can write the sampled $x(t)$ by defining values only at $t = nT$ integer multiples of the sampling time T. Thus

$$x(t = nT) = x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{2.3}$$

is a sum of weighted delayed impulses [4]. The continuous time version of this is

$$x(t) = \sum_{k=-\infty}^{\infty} x(kT)\delta(t-kT) \tag{2.4}$$

which for the sampling times $t = nT$ is equivalent to (2.3) [5]. [6] We convert from the continuous time Laplace transforms to the discrete time z-transform as follows.

$$\begin{aligned} X(s) & \quad \int_{t=-\infty}^{\infty} x(t)e^{-st}dt & (2.5) \\ &= \int_{t=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(kT)\delta(t-kT)e^{-st}dt & (2.6) \\ &= \sum_{k=-\infty}^{\infty} x(kT) \int_{t=-\infty}^{\infty} e^{-st}\delta(t-kT)dt & (2.7) \\ &= \sum_{k=-\infty}^{\infty} x(kT)e^{-skT} & (2.8) \\ &= \sum_{k=-\infty}^{\infty} x[k]z^{-k} & (2.9) \end{aligned}$$

[1]the lower limit is zero instead of $-\infty$ for the unilateral Laplace transform, which is used to solve problems with initial conditions

[2]Note that in the previous chapter, we used the variable n as a general time index, and k for specified values. In this chapter, we may use k as a general time index also.

[3]The negative sign in the exponent is used by convention. In principle, the positive sign could be used, and all the subsequent definitions and theorems rewritten.

[4]and may also be interpreted as the product of $x[n]$ with the comb function $\sum_{k=-\infty}^{\infty} \delta[n-k]$.

[5]and may also be interpreted as the product of $x(t)$ with the comb function $\sum_{k=-\infty}^{\infty} \delta[t-kT]$.

[6]Here $\delta(t-kT)$ is the continuous time δ function with value ∞ at $t = kT$ and zero elsewhere, with area $\int_{-\infty}^{\infty} \delta(t-kT) = 1$, and sifting property $\int_{-\infty}^{\infty} x(t)\delta(t-kT) = x(kT)$.

$$= X(z) \quad (2.10)$$

$$(2.11)$$

where $z = e^{sT}$ and we write $x(kT) = x[k]$. There are other transforms, including the Fourier transform, Fourier series, discrete Fourier transform and discrete-time Fourier transform. The relationship between all these transforms is discussed in section ...

2.2 BASIC Z-TRANSFORMS

We want to use the z-transform to solve the general LTI IIR system difference equations

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] \quad (2.12)$$

$$+ \sum_{k=0}^{M} b_k x[n-k] \quad (2.13)$$

for arbitrary values of the coefficients a_k, b_k. Using the z-transform will avoid the computations needed for the iterative solution, solving for $y[0], y[1]$,... etc in turn. The z-transform will also yield insights into the behaviour of the system and the shape of the impulse response, which is not always obvious from the values of a_k, b_k [7] The z-transform also provides insight into the frequency domain behaviour of LTI systems, as will be seen later.

The solution requires us to take the z-transform of both sides of the difference equation. In this section, we derive the basic z-transform pairs needed to do that, and then do it in the next section.

2.2.0.5 z-transform for finite sequence Recall that the z-transform

$$X(z) = \sum_{k=-\infty}^{\infty} x[k] z^{-k} \quad (2.14)$$

converts a sequence of numbers or samples $x[k]$ (which may be written as a vector of samples $\mathbf{x}$) to a function of the complex variable z. For example, if we define a finite vector of samples $x[0] = 2, x[1] = 4, x[2] = -3$, or $\mathbf{x}' = [\mathbf{103}]$, [8]

[7] In the simple first order case where $M = N = 1$, we have the IIR feedback example in the previous chapter, where we did predict the impulse response behaviour.

[8] Recall that in Matlab, the vector index starts at 1, so that in the Matlab script we would have to write $x[1] = 2, x[2] = 4, x[3] = -3$, or $\mathbf{x}' = [\mathbf{103}]$.

k	x[k]
0	2
1	4
2	-3

and it is implied that all values of $x[k] = 0$ for $k \neq 0, 1, 2$, then

$$\begin{aligned} X(z) &= \sum_{k=0}^{2} x[k] z^{-k} && (2.15) \\ &= x[0]z^0 + x[1]z^{-1} + x[2]z^{-2} && (2.16) \\ &= 2 + 4z^{-1} + -3z^{-2} && (2.17) \end{aligned}$$

We can also write the vector using the basic definition that any sequence is the sum of weighted delayed samples

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \quad (2.18)$$

In this case, evaluating for $k = 0, 1, 2$, we obtain

$$\begin{aligned} x[n] &= \sum_{k=0}^{2} x[k]\delta[n-k] && (2.19) \\ &= x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] && (2.20) \\ &= 2\delta[n] + 4\delta[n-1] + -3\delta[n-2] && (2.21) \end{aligned}$$

Comparing the expressions for $x[n]$ and $X(z)$ term by term, we can see that

time domain	z-domain
$x[n]$	X(z)
$2\delta[n]$	2
$4\delta[n-1]$	$4z^{-1}$
$-3\delta[n-2]$	$-3z^{-2}$
$a\delta[n-k]$	az^{-k}

In general, we write the z-transform pair

$$f[n] = a_k \delta[n-k] \leftrightarrow a_k z^{-k} \quad (2.22)$$

for a signal $f[n]$ which is a_k at time $n = k$, and zero at all other times n. [9] In this case, the sum in the definition of the z-transform reduces to a single term. If $a_k = a_0 = 1$, then we write

$$f[n] = \delta[n-k] \leftrightarrow z^{-k} \quad (2.23)$$

so we have the z-transform pair

$$\delta[n-k] \leftrightarrow z^{-k} \quad (2.24)$$

[9] we use a subscript a_k so that we can have a different weight for each different value of k.

i.e. the z-transform of a δ-function delayed by k is z^{-k}. If $k = 0$, then

$$f[n] = \delta[n] \leftrightarrow z^0 = 1 \tag{2.25}$$

so we have the z-transform pair

$$\delta[n] \leftrightarrow 1 \tag{2.26}$$

so that the z-transform of the unit impulse at $k = 0$ is the constant 1 with no z-dependence.

Since any signal $x[n]$ is a sum of weighted delayed δ-functions as per (2.18), and we know the z-transform of a δ-function delayed by k is z^{-k} we can find the z-transform of any signal $x[n]$ as the sum of terms which are negative powers of z (i.e. powers of z^{-1}. To be precise,

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \leftrightarrow X(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \tag{2.27}$$

which is exactly the definition of the z-transform.

2.2.0.6 z-transform of a semi-infinite sequence The second basic z-transform considers the case where the signal $x[n]$ has an infinite number of terms. The same definition applies.

Consider

$$f[k] = \left\{ \begin{array}{cc} a & k \geq 0 \\ 0 & k < 0 \end{array} \right\} = au[k] \tag{2.28}$$

where we define the unit step function

$$u[k] = \left\{ \begin{array}{cc} 1 & k \geq 0 \\ 0 & k < 0 \end{array} \right\} \tag{2.29}$$

Equivalently, we can write a semi-infinite vector

$$\mathbf{f}' = [a\ a\ a\ a\ a...] \tag{2.30}$$

or

k	x[k]
0	a
1	a
2	a
3	a

and it is implied that all values of $x[k] = 0$ for $k < 0$. By definition

$$F(z) = \sum_{k=-\infty}^{\infty} f[k]z^{-k} \tag{2.31}$$

Thus we can write

$$
\begin{aligned}
F(z) &= \sum_{k=0}^{\infty} az^{-k} && (2.32)\\
&= a\sum_{k=0}^{\infty}(z^{-1})^k && (2.33)\\
&= a(z^0 + z^{-1} + z^{-2} + ...) && (2.34)\\
&= a((z^{-1})^0 + (z^{-1})^1 + (z^{-1})^2 + ...) && (2.35)\\
&= a(w^0 + w^1 + w^2 + ...) && (2.36)\\
& && (2.37)
\end{aligned}
$$

where $w = z^{-1}$. This expression can be simplified by recalling the geometric series

$$\sum_{k=0}^{\infty} w^k = \frac{1}{1-w} \tag{2.38}$$

for $|w| < 1$ and substituting $w = z^{-1}$ to obtain

$$\sum_{k=0}^{\infty}(z^{-1})^k = \frac{1}{1-z^{-1}} \tag{2.39}$$

for $|z^{-1}| < 1$ (or equivalently $z > 1$) so that

$$F(z) = a\frac{1}{1-z^{-1}} = \frac{az}{z-1} \tag{2.40}$$

for $|z| > 1$. Thus we can write the z-transform pair

$$au[k] \leftrightarrow \frac{az}{z-1} \tag{2.41}$$

Another example is

$$f[k] = a\alpha^k u[k] \tag{2.42}$$

where α is a complex constant.

$$
\begin{aligned}
F(z) &= \sum_{k=-\infty}^{\infty} f[k]z^{-k} && (2.43)\\
&= \sum_{k=0}^{\infty} a\alpha^k z^{-k} && (2.44)\\
&= a\sum_{k=0}^{\infty}(\alpha z^{-1})^k && (2.45)\\
&= a\frac{1}{1-(\alpha z^{-1})} && (2.46)\\
&= \frac{az}{z-\alpha} && (2.47)\\
& && (2.48)
\end{aligned}
$$

valid for

$$
\begin{aligned}
|\alpha z^{-1}| &< 1 && (2.49)\\
|z^{-1}| &< \frac{1}{\alpha} && (2.50)\\
|\frac{1}{z}| &< \frac{1}{\alpha} && (2.51)\\
|z| &> |\alpha| && (2.52)\\
& && (2.53)
\end{aligned}
$$

2.2.0.7 Properties of z-transform Since z-transforms are used with general LTI (linear, time-invariant) systems, we have the following properties.

Linearity

If

$$
\begin{aligned}
f[k] &\leftrightarrow F(z) && (2.54)\\
g[k] &\leftrightarrow G(z) && (2.55)\\
& && (2.56)
\end{aligned}
$$

then

$$af[k] + bg[k] \leftrightarrow aF(z) + bG(z) \tag{2.57}$$

Time-shifting

If

$$f[n] \leftrightarrow F(z) \tag{2.58}$$

then

$$f[n-m] \leftrightarrow z^{-m}F(z) \tag{2.59}$$

The time-shifting property of the z-transform can be shown using the basic definition of signals and the z-transform

$$f[n] = \sum_{k=-\infty}^{\infty} f[k]\delta[n-k] \leftrightarrow F(z) = \sum_{k=-\infty}^{\infty} f[k]z^{-k} \tag{2.60}$$

and replacing n with $n-m$, thus writing

$$
\begin{aligned}
f[n-m] = \sum_{r=-\infty}^{\infty} f[r]\delta[n-m-r] \quad &\leftrightarrow \quad \sum_{r=-\infty}^{\infty} f[r]z^{-m-r} && (2.61)\\
&= \quad z^{-m}\sum_{r=-\infty}^{\infty} f[r]z^{-r} && (2.62)\\
&= \quad z^{-m}F(z) && (2.63)
\end{aligned}
$$

In effect, k is replaced by $m-r$.

2.2.0.8 z-transform and convolution Using these properties, we are now almost ready to apply the the z-transform to help solve the general LTI IIR system equation

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] \tag{2.64}$$

$$+ \sum_{k=0}^{M} b_k x[n-k] \tag{2.65}$$

First we consider the simpler but less general case of an FIR system or convolution sum where $a_k = 0$ for all k, and we write $b_k = h[k]$, so that

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k] \tag{2.66}$$

Take the z-transform of both sides we find

$$Y(z) = H(z)X(z) \tag{2.67}$$

The proof is a straigthforward but tedious exercise in algebra and using the z-transform properties. The proof also works if $M = \infty$.

Thus *convolution in the time domain corresponds to multiplication in the frequency domain.* This is another fundamental result in signal processing which will recur often.

Thus if we have a LTI system $h[n]$ and input $x[n]$, then we can compute the output $y[n]$ in one of two ways

- compute the convolution sum $y[n] = \sum_{k=0}^{M} h[k]x[n-k]$ iteratively for each value of n to get $y[n]$
- 1. $x[n] \to X(z)$,
  2. $h[n] \to H(z)$,
  3. $Y(z) = H(z)X(z)$
  4. take the inverse z-transform of $Y(z)$ to get $y[n]$.

The first method is well suited to computer calculation, but yields no insight, i.e. you just get numbers $y[n]$ for a finite number of values of n when you stop computing. The second method is an analytical method which can yield a closed form solution for $y[n]$ for all n, and can yield insight as to the behaviour of $y[n]$. In practice, $y[n]$ will be samples of a continuous time function $y(t)$ (typically combinations of exponentials and sine waves), and the z-transform method can help to find that function and its parameters (frequencies and damping factors).

2.2.0.9 Transfer function For the general LTI system, $H(z)$ is called the transfer function, since it provides the recipe for transferring $X(z)$ to $Y(z)$ via the equation $Y(z) = H(z)X(z)$. If we know the input $x[n]$ and the output $y[n]$, then the transfer function

$$H(z) = \frac{Y(z)}{X(z)} \tag{2.68}$$

We now show a very fundamental result: **the transfer function $H(z)$ is the z-transform of the impulse response $h[n]$.**

The proof is straightforward, we assume the input is an impulse $x[n] = \delta[n]$ so that $X(z) = 1$ [10]. Thus $Y(z) = H(z)X(z) = H(z) \cdot 1 = H(z)$ and taking the inverse z-transform of both sides of this result $Y(z) = H(z)$ yields the output $y[n] = h[n]$ equal to the impulse response.

As mentioned in the previous chapter, if we know the impulse response $h[k]$ of a LTI system, then we know the response $y[k]$ to *any* input $x[k]$, since any $x[k]$ is just the sum of weighted delayed impulses., so the output $y[k]$ is just the sum (superposition) of weighted delayed impulse responses. The input $x[k]$ may be a sampled sine wave, or any waveform.

Thus if we know the transfer function $H(z)$ of a LTI system, then again we know the response $y[k]$ to any input $x[k], X(z)$.

The algebraic form of $H(z)$ tells us about the response characteristics of the system.

2.2.0.10 Transfer function of general LTI system We can now find transfer function for a general LTI system including feedback

$$\begin{aligned} y[n] &= \sum_{k=1}^{N} a_k y[n-k] && (2.69) \\ &+ \sum_{k=0}^{M} b_k x[n-k] && (2.70) \end{aligned}$$

We can, of course, obtain the transfer function by computing the impulse response iteratively for each value of $n = 0, 1, 2..., M$ by assuming $x[k] = \delta[k]$ to find $h[k]$ up to some finite value of time index M, and then taking the z-transform

$$H(z) = \sum_{k=0}^{M} h[k] z^{-k} \tag{2.71}$$

However, the result will be a lot of terms in powers of z, and will be an approximation if the system is IIR (i.e. if there is feedback and some of the $a_k \neq 0$). Also, the numerical values of the terms $h[n]$ may not yield much

[10] Recall $X(z) = \sum_k x[k]z^{-k} = \sum_k \delta[k]z^{-k} = 1 \cdot z^0 + 0 \cdot z^{-k} = 1$

insight into the general behaviour or response characteristics. In general, $h[n]$ will look like the sum of sine waves at different amplitudes, frequencies, phases and exponential damping factors. It is not easy to extract the sine wave parameters from the number sequence $h[n]$. Hence we are motivated to find the transfer function $H(z)$ in a simpler form, and we will show how it it used to find these parameters.

To keep the algebra simple, we consider $M = N = 2$ and write the system equation

$$\begin{aligned} y[n] &= a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] && (2.72) \\ &+ b_1 x[n-1] + b_2 x[n-2] && (2.73) \\ & && (2.74) \end{aligned}$$

or equivalently

$$\begin{aligned} & y[n] - a_1 y[n-1] - a_2 y[n-2] && (2.75) \\ =\ & b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] && (2.76) \\ & && (2.77) \end{aligned}$$

This system is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{b}, \alpha, \mathbf{x}] \tag{2.78}$$

where $\mathbf{b} = [\mathbf{b_0}\ \mathbf{b_1}\ \mathbf{b_2}]$ and $\alpha = [1\ \ -a_1\ \ -a_2]$. Thus $\alpha_1 = -a_1$ and $\alpha_2 = -a_2$.

To find the transfer function, we take the z-transform of both sides term by term, and then gather terms to compute $H(z) = Y(z)/X(z)$. Using the linearity and time-shifting properties of the z-transform, (if $f[n] \leftrightarrow F(z)$ then $f[n-m] \leftrightarrow z^{-m}F(z)$) we write

$$\begin{aligned} & Y(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z) && (2.79) \\ =\ & b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) && (2.80) \\ & && (2.81) \end{aligned}$$

Simplifying

$$\begin{aligned} & Y(z)(1 - a_1 z^{-1} - a_2 z^{-2}) && (2.82) \\ =\ & X(z)(b_0 + b_1 z^{-1} + b_2 z^{-2}) && (2.83) \\ & && (2.84) \end{aligned}$$

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} && (2.85) \\ &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} && (2.86) \\ & && (2.87) \end{aligned}$$

Thus the transfer function $H(z)$ is a ratio of polynomials in the variable z^{-1}.

For convenience, we can also multiply top and bottom by z^2 and write $H(z)$ as a ratio of polynomials in z

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} && (2.88) \\ &= \frac{b_0 z^2 + b_1 z + b_2}{z^2 - a_1 z - a_2} && (2.89) \\ &= b_0 \frac{z^2 + (b_1/b_0)z + (b_2/b_0)}{z^2 - a_1 z - a_2} && (2.90) \\ & && (2.91) \end{aligned}$$

Note that the z^2 term has coefficient 1 in both numerator and denominator. To gain further insight into $H(z)$, we factor the polynomials and write

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} && (2.92) \\ &= b_0 \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} && (2.93) \end{aligned}$$

where z_1, z_2 are the roots of $z^2 + b_1 z + b_2$ and p_1, p_2 are the roots of $z^2 - a_1 z - a_2$ [11]. The roots are found from the quadratic formula[12]. Thus

$$z_1 = \frac{(b_1/b_0) + \sqrt{(b_1/b_0)^2 - 4(b_2/b_0)}}{2} \tag{2.94}$$

$$z_2 = \frac{(-b_1/b_0) - \sqrt{(b_1/b_0)^2 - 4(b_2/b_0)}}{2} \tag{2.95}$$

$$p_1 = \frac{-a_1 + \sqrt{a_1^2 - 4a_2}}{2} \tag{2.96}$$

$$p_2 = \frac{-a_1 - \sqrt{a_1^2 - 4a_2}}{2} \tag{2.97}$$

The numbers z_1, z_2 are called the zeros of $H(z)$, because if we choose $z = z_1$ or $z = z_2$, $H(z) = 0$. The numbers p_1, p_2 are called the poles of $H(z)$, because if we choose $z = p_1$ or $z = p_2$, $H(z) = \infty$.

The location of these poles and zeros determines the shape of the impulse response. Of course, since the factored and unfactored versions of $H(z)$ are equivalent, the values of the filter coefficients a_1, a_2, b_1, b_2 also determines the

[11] or using the Matlab notation, p_1, p_2 are the roots of $z^2 + \alpha_1 z + \alpha_2$

[12] For any polynomial $az^2 + bz + c$, there are two roots given by $r_1, r_2 = -b \pm \frac{\sqrt{b^2-4ac}}{2a}$.

shape of the impulse response. However, we will find that the poles and zeros give more insight, and thus we will want to factor $H(z)$. We can write

$$
\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} && (2.98)\\
&= b_0\frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)} && (2.99)\\
&= b_0\frac{(z^2-(z_1+z_2)z+z_1z_2}{(z^2-(p_1+p_2)z+p_1p_2} && (2.100)\\
&= b_0\frac{z^2+(b_1/b_0)z+(b_2/b_0)}{z^2-a_1z-a_2} && (2.101)\\
& && (2.102)
\end{aligned}
$$

so that $b_1/b_0 = -(z_1+z_2), b_2/b_0 = z_1z_2, a_1 = p_1+p_2, z_2 = -p_1p_2$.

2.2.0.11 Transfer function of first order LTI system with one pole In chapter xxx, we found the shape of the impulse response for the difference equation

$$y[n] = ay[n-1] + bx[n] \tag{2.103}$$

By relabelling $a = a_1$ and $b = b_0$, this is a special case of the general LTI system equation with $N = 1$ and $M = 0$

$$
\begin{aligned}
y[n] &= a_1y[n-1] + b_0x[n] && (2.104)\\
& && (2.105)
\end{aligned}
$$

or equivalently

$$
\begin{aligned}
& y[n] - a_1y[n-1] && (2.106)\\
=\ & b_0x[n] && (2.107)\\
& && (2.108)
\end{aligned}
$$

This system is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{b}, \alpha, \mathbf{x}] \tag{2.109}$$

where $\mathbf{b} = [\mathbf{b_0}]$ (one element vector) and $\alpha = [1\ -a_1]$, i.e. $\alpha_0 = 1, \alpha_1 = -a_1$.

We derived the impulse response $h[n]$ for this system in Chapter xx using direct calculation assuming $x[n] = \delta[n]$, and found that the shape of $h[n]$ depended on the numerical value of a, with 7 possible shapes. Here we will find the transfer function, and observe the same 7 possibilities.

To find the transfer function, we take the z-transform of both sides term by term, and then gather terms to compute $H(z) = Y(z)/X(z)$.Using the linearity and time-shifting properties of the z-transform, (if $f[n] \leftrightarrow F(z)$ then $f[n-m] \leftrightarrow z^{-m}F(z)$) we write

$$
\begin{aligned}
& Y(z) - a_1z^{-1}Y(z) && (2.110)\\
=\ & b_0X(z) && (2.111)\\
& && (2.112)
\end{aligned}
$$

Simplifying

$$
\begin{aligned}
& Y(z)(1 - a_1 z^{-1}) && (2.113)\\
= \; & X(z)(b_0) && (2.114)\\
& && (2.115)
\end{aligned}
$$

$$
\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} && (2.116)\\
&= \frac{b_0}{1 - a_1 z^{-1}} && (2.117)\\
& && (2.118)
\end{aligned}
$$

For convenience, we can also multiply top and bottom by z and write $H(z)$ as a ratio of polynomials in z

$$
\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} && (2.119)\\
&= \frac{b_0 z}{z - a_1} && (2.120)\\
& && (2.121)
\end{aligned}
$$

In this case, there is no need to factor, we can simply write

$$
\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} && (2.122)\\
&= b_0 \frac{(z - z_1)}{(z - p_1)} && (2.123)\\
& && (2.124)
\end{aligned}
$$

where $z_1 = 0$ and $p_1 = a_1$.

The shape of the impulse response depends on the numerical value of the pole a_1. The gain factor b_0 adds scaling and the zero $z_1 = 0$ adds delay.

2.2.0.12 Transfer function, poles and zeros for some example LTI systems *Example 1.*

To find the transfer function for the simple system

$$y[n] = b_0 x[n] \tag{2.125}$$

we take the z-transform of both sides and get

$$Y(z) = X(z) \tag{2.126}$$

or

$$H(z) = \frac{Y(z)}{X(z)} = b_0 \tag{2.127}$$

so there are no poles and no zeros.

Example 2.

To find the transfer function for the system

$$y[n] = b_1 x[n-1] \tag{2.128}$$

we take the z-transform of both sides and get

$$Y(z) = b_1 z^{-1} X(z) \tag{2.129}$$

or

$$H(z) = \frac{Y(z)}{X(z)} = b_1 z^{-1} = \frac{1}{z} = \frac{1}{z-0} = \frac{1}{z-p_1} \tag{2.130}$$

so there is one pole $p_1 = 0$ and no zeros.

Example 3.

To find the transfer function for the system

$$y[n] = b_0 x[n] + b_1 x[n-1] \tag{2.131}$$

we take the z-transform of both sides and get

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) \tag{2.132}$$

or

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} = b_0 \frac{z + b_1/b_0}{z} = b_0 \frac{z - z_1}{z - p_1} \tag{2.133}$$

so there is one pole $p_1 = 0$ and one zero $z_1 = -(b_1/b_0)$.

Example 4.

For completeness we consider the system

$$y[n] = x[n+1] \tag{2.134}$$

which is *anti-causal* since the output $y[n]$ is a future (advanced) sample $x[n+1]$. We take the z-transform of both sides and get

$$Y(z) = z^{+1} X(z) \tag{2.135}$$

or

$$H(z) = \frac{Y(z)}{X(z)} = z = \frac{z+0}{1} \tag{2.136}$$

so there is one zero $z_1 = 0$ and no poles.

2.2.0.13 Inverse z-transforms Inverse z-transforms are useful for finding the impulse response from the transfer function.

Inverse z-transforms are also useful for for finding the output of a general LTI system with given input. Consider an LTI system defined by $h[n]$, given the with input $x[n]$, we can find the output $y[n]$ either by convolution $y[n] =$

$x[n] * h[n]$ or by finding the transfer function $H(z)$, and the z-transform of the input $X(z)$, multiplying $H(z)X(z) = Y(z)$, and taking the inverse z-transform of $Y(z)$ to get the output $y[n]$.

$Y(z)$, just like $H(z)$ is a ratio of polynomials which can be factored, and then broken up into the sum of partial fractions of the form

$$F(z) = \frac{cz}{z-p} = \frac{c}{1-pz^{-1}} \tag{2.137}$$

[13] We then find the inverse z-transform of each partial fraction, and sum the result. Since we want a z term in the numerator of the partial fraction terms, we

The inverse z-transform of the partial fraction can be found directly from the geometric series, and comparing with the definition of the z-transform.

$$\begin{aligned}
F(z) &= \frac{c}{1-pz^{-1}} && (2.139)\\
&= c\sum_{k=0}^{\infty}(pz^{-1})^k && (2.140)\\
&= c + cpz^{-1} + cp^2z^{-2} + ... && (2.141)\\
&= f[0] + f[1]z^{-1} + f[2]z^{-2} + ... && (2.142)\\
&= \sum_{n=0}^{\infty} f[n]z^{-n} && (2.143)\\
& && (2.144)
\end{aligned}$$

Thus by comparing terms, we find

$$\begin{aligned}
f[0] &= c && (2.145)\\
f[1] &= cp && (2.146)\\
f[2] &= cp^2 && (2.147)\\
& && (2.148)
\end{aligned}$$

or in vector notation

$$\mathbf{f}\ [\mathbf{c}\ \ \mathbf{cp}\ \ \mathbf{cp^2}\ \ ...] \tag{2.149}$$

We could also apply the z-transform pair

$$az^{-k} \leftrightarrow a\delta[n-k] \tag{2.150}$$

[13]Note that there is a z term in the numerator of the partial fraction term. Thus, we usually divide $F(z)$ by z, and take the partial fraction of

$$\frac{F(z)}{z} = \frac{c}{z-p} \tag{2.138}$$

to eliminate the z term in the numerator, and thus allow the standard method of partial fractions to be used.

term by term to each line of (2.144) to obtain

$$\begin{aligned} f[n] &= \sum_{k=0}^{\infty} cp^k \delta[n-k] && (2.151) \\ &= c\delta[n] + cp\delta[n-1] + cp^2\delta[n-2] + ... && (2.152) \\ &= f[0]\delta[n] + f[1]\delta[n-1] + f[2]\delta[n-2] + ... && (2.153) \\ &= \sum_{k=0}^{\infty} f[k]\delta[n-k] && (2.154) \\ & && (2.155) \end{aligned}$$

Note that the last line is the familiar sum of weighted delayed delta functions used to represent any signal, except that the lower limit is 0 instead of $-\infty$. To check, evaluate for different values of n

$$\begin{aligned} n = 0: &\quad f[0] = c && (2.156) \\ n = 1: &\quad f[1] = cp && (2.157) \\ n = 2: &\quad f[2] = cp^2 && (2.158) \\ & && (2.159) \end{aligned}$$

In general

$$f[n] = cp^n u[n] \tag{2.160}$$

which is equivalent to the first line of (2.155) above. The factor $u[n]$ appears [14] because $f[n] = 0$ for all $n < 0$. Thus we can write the z-transform pair

$$\frac{c}{1 - pz^{-1}} = \frac{cz}{z - p} \leftrightarrow \sum_{k=0}^{\infty} cp^k \delta[n-k] = cp^n u[n] \tag{2.161}$$

Thus if

$$F(z) = \frac{c}{1 - pz^{-1}} = \frac{cz}{z - p} \tag{2.162}$$

then

$$f[n] = cp^n u[n] \tag{2.163}$$

One pole one zero transfer function

The first application of this z-transform pair is to the single pole transfer function above, which we obtained from the difference equation (LTI system)

$$y[n] = a_1 y[n-1] + b_0 x[n] \tag{2.164}$$

[14]Recall the definition $u[n] = 1$ for all $n/ge0, u[n] = 0$ for all $n < 0$.

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.165)$$
$$= \frac{b_0 z}{z - a_1} \quad (2.166)$$
$$(2.167)$$

In this case, $H(z)$ is already in partial fractions form

$$F(z) = \frac{cz}{z-p} = \frac{c}{1 - pz^{-1}} \quad (2.168)$$

with $c = b_0$ and $p = a_1$. So applying the z-transform pair

$$\frac{cz}{z-p} \leftrightarrow cp^n u[n] \quad (2.169)$$

we take the inverse z-transform to obtain the impulse response

$$h[n] = b_0 a_1^n u[n] \quad (2.170)$$

or

$$n = 0: \quad h[0] = b_0 \quad (2.171)$$
$$n = 1: \quad h[1] = b_0 a_1 \quad (2.172)$$
$$n = 2: \quad h[2] = b_0 a_1^2 \quad (2.173)$$
$$(2.174)$$

or in vector notation

$$\mathbf{h}' = [\mathbf{b_0 b_0 a_1 b_0 a_1^2} ...] \quad (2.175)$$

which is identical to the result obtained in Chapter xxx. The shape of the impulse response depends on the numerical value of a_1, as before (Figure ...).

Two pole two zero transfer function

We now apply this result to the two-pole two-zero transfer function

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.176)$$
$$= b_0 \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \quad (2.177)$$
$$(2.178)$$

We first break $H(z)$ into partial fractions

$$H(z) = \frac{c_1 z}{z - p_1} + \frac{c_2 z}{z - p_2} \quad (2.179)$$

where the values of c_1, c_2 will depend on b_0, z_1, z_2, p_1, p_2. Note that p_1, p_2 may both be real and distinct, or may be a complex conjugate pair $p_1 =$

$u+jv, p_2 = u-jv$ where $j=\sqrt{-1}$ [15]. [16]. Using the z-transform pair (2.189)

$$h[n] = c_1 p_1^n u[n] + c_2 p_2^n u[n] \tag{2.181}$$

or

$$\begin{aligned} n=0: &\quad h[0] = c_1 + c_2 &(2.182)\\ n=1: &\quad h[1] = c_1 p_1 + c_2 p_2 &(2.183)\\ n=2: &\quad h[2] = c_1 p_1^2 + c_2 p_2^2 &(2.184)\\ & &(2.185)\end{aligned}$$

or in vector notation

$$\mathbf{h}' = [\mathbf{c_1 + c_2} \;\; \mathbf{c_1 p_1 + c_2 p_2} \;\; \mathbf{c_1 p_1^2 + c_2 p_2^2} \;\; ...] \tag{2.186}$$

2.2.0.14 Characteristic of impulse response for two-pole two-zero transfer function The characteristic of the impulse response of the two-pole two-zero transfer function is determined by the values of the poles p_1, p_2. Since the poles may be complex, we consider the pole locations in the complex z-plane (Figure x). Both poles may be either real or a complex conjugate pair. [17]

Real and distinct poles

If the poles are real and distinct (different), then we consider each term in $h[n]$ separately, and use

$$H(z) = \frac{cz}{z-p} \leftrightarrow h[n] = cp^n u[n] \tag{2.189}$$

The results for $h[n]$ for different values of p along the real axis are tabulated below. There are 7 regions total, for p inside, on and outside the unit circle, both positive and negative. For each region, the shape of the impulse response is some kind of exponential.

[15] Methods of finding partial fractions are reviewed in Appendix X. Complex numbers are reviewed in Appendix Y (see 'DSP First' for example)

[16] When the poles are real and identical, i.e. the case of repeated poles $p_1 = p_2$, then the partial fraction expression for $H(z)$ is

$$\frac{c_1 z}{(z-p)} + \frac{c_2 z}{(z-p)^2} \tag{2.180}$$

In this case $h[n]$ is found as shown in the next section.

[17] The case of one-pole one-zero was dealt with in the previous chapter. There we saw the characteristics of the impulse response of a first order difference equation

$$y[n] = ay[n-1] + bx[n] \tag{2.187}$$

is determined by the value of a. This result applies directly to the corresponding one-pole, one-zero transfer function

$$H(z) = \frac{bz}{z-a} \tag{2.188}$$

with the pole at a.

$H(z)$	region	$h[n]$	shape
$\frac{cz}{z-p}$	$p < -1$	$cp^n u[n]$	alternating increasing exponential
$\frac{cz}{z-p}$	$p = -1$	$cp^n u[n]$	alternating constant
$\frac{cz}{z-p}$	$-1 < p < 0$	$cp^n u[n]$	alternating decaying exponential
$\frac{cz}{z-p}$	$p = 0$	$cp^n u[n]$	zero
$\frac{cz}{z-p}$	$0 < p < 1$	$cp^n u[n]$	decaying exponential
$\frac{cz}{z-p}$	$p = 1$	$cp^n u[n]$	constant
$\frac{cz}{z-p}$	$p > 1$	$cp^n u[n]$	increasing exponential

Complex conjugate poles

If the poles are a complex conjugate pair $p_1 = u + jv = re^{j\theta} = p$, and $p_2 = u - jv = re^{-j\theta} = p^*$, then there are 3 regions total, p inside, on, and outside the unit circle, and in each region, the shape of the impulse response is the sum of two complex exponentials, which is some kind of sinusoid. In this case, the partial fraction expansion

$$H(z) = \frac{c_1 z}{z - p_1} + \frac{c_2 z}{z - p_2} = \frac{cz}{z - p} + \frac{c^* z}{z - p^*} \tag{2.190}$$

where we find $c_1 = c = \rho e^{j\phi}$, and $c_2 = c* = \rho e^{-j\phi}$. It may turn out that $\phi = 0$, in which case $c = c^*$ is real.

Applying the z-transform pair (2.161) to each term we obtain

$$\begin{aligned} h[n] &= c_1 p_1^n u[n] + c_2 p_2^n u[n] && (2.191)\\ &= cr^n e^{jn\theta} u[n] + c^* r^n e^{-jn\theta} u[n] && (2.192)\\ & && (2.193) \end{aligned}$$

The results for different values of r are tabulated below. The positive and negative complex exponentials can be combined into a sinusoid using

$$e^{jn\theta} + e^{-jn\theta} = 2cos(n\theta) \tag{2.194}$$

If $c = c^*$ is real, then we can write

$$\begin{aligned} h[n] &= cr^n [e^{jn\theta} + e^{-jn\theta}] u[n] && (2.195)\\ &= 2cr^n cos(n\theta) u[n] && (2.196)\\ & && (2.197) \end{aligned}$$

If c, c^* are complex, then

$$\begin{aligned} h[n] &= \rho e^{j\phi} e^{jn\theta} u[n] + \rho e^{-j\phi} r^n e^{-jn\theta} u[n] && (2.198)\\ &= 2\rho r^n cos(n\theta + \phi) u[n] && (2.199)\\ & && (2.200) \end{aligned}$$

Thus we can write the z-transform pair

$$\frac{\rho e^{j\phi} z}{z - p} + \frac{\rho e^{-j\phi} z}{z - p^*} \leftrightarrow 2\rho r^n cos(n\theta + \phi) u[n] \tag{2.201}$$

$H(z)$	region	$h[n]$	shape
$\frac{cz}{z-p}+\frac{c^*z}{z-p^*}$	$0<r<1$	$2cr^n cos(n\theta+\phi)u[n]$	decaying sinusoid
$\frac{cz}{z-p}+\frac{c^*z}{z-p^*}$	$r=1$	$2ccos(n\theta+\phi)u[n]$	sinusoid
$\frac{cz}{z-p}+\frac{c^*z}{z-p^*}$	$r>1$	$2cr^n cos(n\theta+\phi)u[n]$	increasing sinusoid

Repeated poles

If the poles of $H(z)$ are real and identical, i.e. $\theta = 0, p_1 = p_2 = u + j0 = re^{j0}$, i.e.

$$H(z) = \frac{Y(z)}{X(z)} \tag{2.202}$$

$$= b_0\frac{(z-z_1)(z-z_2)}{(z-p)^2} \tag{2.203}$$

$$\tag{2.204}$$

then the partial fraction expression for $H(z)$ is

$$\frac{c_1 z}{(z-p)} + \frac{c_2 z}{(z-p)^2} \tag{2.205}$$

The z-transform of the second term

$$\frac{cz}{(z-p)^2} \leftrightarrow cnp^{n-1}u[n-1] \tag{2.206}$$

We can write either $u[n]$ or $u[n-1]$, since the term for $n=0$ is zero in any case, and $u[n] = u[n-1]$ for all $n > 0$. [18] For this second term, there are 3 regions total, p inside, on, and outside the unit circle, as tabulated below.

[18] This result can be verified by taking the derivative of the geometric series

$$\frac{cz}{z-p} = \frac{c}{1-pz^{-1}} \tag{2.207}$$

$$= c\sum_{k=0}^{\infty}(pz^{-1})^k \tag{2.208}$$

$$= c\sum_{k=0}^{\infty} w^k \tag{2.209}$$

$$= c\frac{1}{1-w} = \frac{c}{1-w} \tag{2.210}$$

with respect to $w = pz^{-1}$ to obtain

$$\frac{d}{dw}\frac{c}{1-w} = \frac{c}{(1-w)^2} = \frac{d}{dw}c\sum_{k=0}^{\infty} w^k = c\sum_{k=1}^{\infty} kw^{k-1} \tag{2.211}$$

$H(z)$	region	$h[n]$	shape
$\frac{cz}{(z-p)^2}$	$r < -1$	$cnp^n u[n]$	alternating increasing faster than exponential
$\frac{cz}{(z-p)^2}$	$r = -1$	$cnp^n u[n]$	alternating increasing linear ramp
$\frac{cz}{(z-p)^2}$	$-1 < r < 0$	$cnp^n u[n]$	alternating rise and then fall towards zero
$\frac{cz}{(z-p)^2}$	$r = 0$	$cnp^n u[n]$	zero
$\frac{cz}{(z-p)^2}$	$0 < r < 1$	$cnp^n u[n]$	rise and then fall towards zero
$\frac{cz}{(z-p)^2}$	$r = 1$	$cnp^n u[n]$	linear ramp increasing
$\frac{cz}{(z-p)^2}$	$r > 1$	$cnp^n u[n]$	increasing faster than exponential

This completes the discussion of characteristics of the impulse response of a two-pole two-zero transfer function.

Exercise: what happens as a pair of complex poles converges toward the real axis (both positive and negative), and become a double pole ?

To summarize:

For a one-pole transfer function, the impulse response will be some kind of exponential, which may be increasing or decreasing and alternating. For a two-pole transfer function, the impulse response may be some kind of sinusoid, which is increasing or decreasing exponentially.

2.2.0.15 Other z-transform pairs Other useful z=transform pairs can be derived using the definition and properties, and are listed in the following table.

Table of z-transforms

Substituting $w = pz^{-1}$

$$\frac{c}{(1-pz^{-1})^2} = \frac{cz^2}{(z-p)^2} = c\sum_{k=1}^{\infty} k(pz^{-1})^{k-1} = c\sum_{k=1}^{\infty} kp^{k-1}z^{-k+1} \tag{2.212}$$

Multiplying by z^{-1}

$$\frac{cz^{-1}}{(1-pz^{-1})^2} = \frac{cz}{(z-p)^2} = c\sum_{k=1}^{\infty} kp^{k-1}z^{-k} \tag{2.213}$$

and comparing with the definition of the z-transform yields the result.

$f[k]$	F(z)
$\delta[n]$	1
$\delta[n-k]$	z^{-k}
$u[k]$	$\frac{z}{z-1} = \frac{1}{1-z^{-1}}$
$p^k u[k]$	$\frac{z}{z-p} = \frac{1}{1-pz^{-1}}$
$p^{k-1}u[k-1]$	$\frac{1}{z-p} = \frac{z^{-1}}{1-pz^{-1}}$
$ku[k]$	$\frac{z}{(z-1)^2} = \frac{1}{1-z^{-1}}$
$kp^{k-1}u[k-1]$	$\frac{z}{(z-p)^2} = \frac{z^{-1}}{(1-pz^{-1})^2}$
$\frac{k(k-1)}{2}p^{k-2}u[k-2]$	$\frac{z}{(z-p)^3}$
$k^2u[k]$	$\frac{z(z+1)}{(z-1)^3}$
$af[k]+bg[k]$	$aF(z)+bG(z)$
$f[n-k]$	$z^{-k}F(z)$
$\rho r^n cos(n\theta+\phi)u[n]$	$\frac{\rho e^{j\phi}z}{z-re^{j\theta}} + \frac{\rho e^{-j\phi}z}{z-re^{-j\theta}}$

2.2.0.16 Applications of the inverse z-transform The inverse z-transform is used to find the impulse response $h[n]$ from the transfer function $H(z)$ of an LTI system.

The inverse z-transform is also used to find the response $y[n]$ of the system $H(z)$ to any input $x[n]$ (since $x[n]$ is the sum of weighted delayed impulses. Consider an LTI system defined by $H(z)$, given the with input $x[n]$, we can find the output $y[n]$ by finding the z-transform of the input $X(z)$, multiplying $H(z)X(z) = Y(z)$, and taking the inverse z-transform of $Y(z)$ to get the output $y[n]$.

A second way to find the output is to first find the impulse response $h[n]$ and then use convolution $y[n] = x[n] * h[n]$. A third way is to find the difference equations corresponding to $H(z)$, and solving for $y[n]$ iteratively for $n = 0, 1, 2, ...$ as was done in Chapter xxx.

The convolution method emphasizes that the output is the sum of delayed impulse responses

$$y[n] = \sum_{k=0}^{\infty} x[k]h[n-k] \tag{2.214}$$

weighted by the input samples $x[k]$. The shape of the impulse response is determined by the locations of the poles in $H(z)$, as shown above. Since any input signal can be defined as the sum of weighted delayed impulses, the output signal will be the sum of weighted delayed 'shapes'.

2.2.0.17 Example inverse z-transforms to be added

2.2.0.18 Difference equations from transfer function To find the difference equations from the transfer function, we write

$$H(z) = \frac{Y(z)}{X(z)} \tag{2.215}$$
$$= b_0\frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)} \tag{2.216}$$
$$= \frac{Y(z)}{X(z)} \tag{2.217}$$
$$\tag{2.218}$$

and then rearrange

$$Y(z)[(z-p_1)(z-p_2)] = X(z)b_0[(z-z_1)(z-z_2)] \tag{2.219}$$
$$Y(z)[z^2-(p_1+p_2)z+p_1p_2] = X(z)b_0[z^2-(z_1+z_2)z+z_1z_2] \tag{2.220}$$
$$Y(z)[1-(p_1+p_2)z^{-1}+p_1p_2z^{-2}] = X(z)b_0[1-(z_1+z_2)z^{-1}+z_1z_2z^{-2}] \tag{2.221}$$
$$Y(z)-(p_1+p_2)z^{-1}Y(z)+p_1p_2z^{-2}Y(z0 = b_0X(z)-b_0(z_1+z_2)z^{-1}X(z)+b_0z_1z_2z^{-2}X(z) \tag{2.222}$$

Now take the inverse z-transform of both sides by inspection, recalling the time-shifting property

$$Y(z) \leftrightarrow y[k] \tag{2.223}$$
$$z^{-1}Y(z) \leftrightarrow y[k-1] \tag{2.224}$$
$$z^{-2}Y(z) \leftrightarrow y[k-2] \tag{2.225}$$
$$\tag{2.226}$$

to obtain

$$y[n]-(p_1+p_2)y[n-1]+p_1p_2y[n-2] = b_0x[n]-b_0(z_1+z_2)x[n-1]+b_0z_1z_2x[n-2] \tag{2.227}$$

$$y[n] = (p_1+p_2)y[n-1]-p_1p_2y[n-2] \tag{2.228}$$
$$+ \quad b_0x[n]-b_0(z_1+z_2)x[n-1]+b_0z_1z_2x[n-2] \tag{2.229}$$
$$= \sum_{k=1}^{2} a_k y[n-k] + \sum_{k=0}^{2} b_k x[n-k] \tag{2.230}$$

where

$$a_1 = p_1+p_2 \tag{2.231}$$
$$a_2 = -p_1p_2 \tag{2.232}$$
$$b_1 = -b_0(z_1+z_2) \tag{2.233}$$
$$b_2 = b_0z_1z_2 \tag{2.234}$$

2.2.0.19 Summary We have defined LTI systems in 3 ways, by the difference equations, the impulse response $h[n]$ and the transfer function $H(z)$. An LTI system is completely characterized by it's impulse response. We have seen how the locations of the poles in the z-plane affect the characteristics of that impulse response. We have seen three methods of finding the response $y[n]$ of an LTI system $h[n]$ to any input $x[n]$.

3

filters

3.1 FREQUENCY RESPONSE OF LTI SYSTEM

We have considered the response of LTI systems to an impulse function, and thereby obtained the impulse response.

In this section, we consider the response of LTI systems to sinusoidal inputs. We will find the if the input is a sinusoid, then the output will be another sinusoid of the same frequency, but with different amplitude and phase.

We define the complex frequency response of an LTI system as the change in amplitude and phase of the sinusoid at each frequency f. Thus if the input sinusoid is $cos(2\pi fnT)$ and the output is $rcos(2\pi fnT+\phi)$, then the frequency response at the frequency f is $re^{j\phi}$. The complex frequency response has two parts: the magnitude response r and the phase response ϕ. The complex frequency response is often called just the 'frequency response', where 'complex' is implied.

The frequency response of an LTI system at all frequencies f is given by the transfer function evaluated on the unit circle [1]

$$H(f) = H(z)|_{z=e^{j\omega}} = H(z)|_{z=e^{j2\pi fT_s}} = H(z)|_{z=e^{j2\pi f/f_s}} \tag{3.1}$$

where $\omega = wT_s = 2\pi f/f_s$ is the normalized radian frequency, and $f_s = 1/T_s$ is the sampling frequency. For $\omega = \pi, f = f_s/2$. Going around the unit circle from $\omega = -\pi$ to π, we can write

[1]The unit circle in the complex z-plane is a circle or radius 1 where the complex number $z = e^{j\omega}$ has magnitude 1 and phase $-\pi < \omega < \pi$.

ω	f
$-\pi$	$-f_s/2$
$-\pi/2$	$-f_s/4$
$-\pi/4$	$-f_s/8$
0	0
$\pi/4$	$f_s/8$
$\pi/2$	$f_s/4$
π	$f_s/2$

Thus we span all frequencies $|f| < f_s/2$, i.e. all frequencies less than the Nyquist frequency, half the sampling rate.

$H(f)$ at a particular value of f is a complex number $re^{j\phi}$. In general $H(f)$ is a complex function of the real variable f, i.e.

$$H(f) = |H(f)|e^{j\,\arg(H(f))} = r(f)e^{j\phi(f)} \tag{3.2}$$

The magnitude response

$$|H(f)| = |H(z)|_{z=e^{j\omega}} = r(f) \tag{3.3}$$

and the phase response

$$arg(H(f)) = arg(H(z)|_{z=e^{j\omega}}) = \phi(f) \tag{3.4}$$

where both r and ϕ are real functions of f.

For any LTI system samples at $f_s = 1/T_s$, the frequency response (magnitude and phase) can be verified by choosing the input as a cosine wave at frequency f sampled at times $t = nT_s$

$$\begin{aligned} x[n] = x(t)|_{t=nT_s} &= cos(2\pi ft)|_{t=nT_s} && (3.5)\\ &= cos(2\pi fnT_s) = cos(2\pi nf/f_s) = cos(\omega n) && (3.6)\end{aligned}$$

for any value of $|f| < f_s/2$ or $|\omega| < \pi$, and computing the output either from the difference equations or by convolution with the impulse response. Either way, the output will be another cosine wave shifted in magnitude and phase

$$y[n] = rcos(\omega n + \phi) = rcos(2\pi nf/f_s + \phi) \tag{3.7}$$

so from the definition of frequency response above, the frequency response at frequency f is $re^{j\phi}$. [2]

By repeating this calculation over a range of values of $|f| < f_s/2$, curves can be obtained for the magnitude response as a function of f

$$|H(f)| = |H(z)|_{z=e^{j\omega}} = r = r(f) \tag{3.8}$$

[2] Recall, if the input sinusoid is $cos(2\pi fnT)$ and the output is $acos(2\pi fnT + \phi)$, then the frequency response is $ae^{j\phi}$.

and the phase response as a function of f

$$arg(H(f)) = arg(H(z)|_{z=e^{j\omega}} = \phi = \phi(f) \tag{3.9}$$

The complex frequency response has two parts: the magnitude response and the phase response.

3.2 EXAMPLE FREQUENCY RESPONSE OF SIMPLE FILTER

Consider the filter

$$H(z) = 1 + \beta z^{-1} \tag{3.10}$$

(Figure x) [3]. The frequency response

$$\begin{aligned} H(f) &= H(z)|_{z=e^{j2\pi f/f_s}} && (3.15)\\ &= 1 + \beta e^{-j2\pi f/f_s} && (3.16)\\ &= 1 + \beta e^{-j\omega} && (3.17)\\ &= re^{j\phi} && (3.18) \end{aligned}$$

Recall that the delay element z^{-1} in Figure x is a delay of $T_s = 1/f_s$ time units, so that the filter output is the sum of the delayed input scaled by β plus the original input. For digital filters, the delay is always an integer number of samples spaced by T_s (in this case, one sample).

The frequency response can also be obtained by considering a continuous time input $x(t) = e^{j2\pi ft} = cos(2\pi ft) + jsin(2\pi ft)$, which is complex sinusoid [4] or phasor at frequency f. The resulting output is, as stated above, the sum of the original input plus the delayed input scaled by β

$$y(t) = e^{j2\pi ft} + \beta e^{j2\pi f(t-T_s)} \tag{3.19}$$

is another sinusoid. Factoring out the phasor and using $T_s = 1/f_s$

$$y(t) = e^{j2\pi ft}(1 + \beta e^{-j2\pi f/f_s}) \tag{3.20}$$

[3] The difference equation for this filter is obtained from

$$H(z) = 1 + \beta z^{-1} = \frac{Y(z)}{X(x)} \tag{3.11}$$

$$Y(z) = X(z) + \beta z^{-1} X(z) \tag{3.12}$$

$$y[n] = x[n] + \beta x[n-1] \tag{3.13}$$

so that $b_0 = 1, b_1 = \beta$. The impulse response is obtained by taking the inverse z-transform of $H(z)$ term by term

$$h[n] = \delta[n] + \beta\delta[n-1] \tag{3.14}$$

[4] A real sinusoidal wave input $x(t) = cos(2\pi ft)$ or $x(t) = sin(2\pi ft)$ could also be used, and the frequency response would be the same. The algebra would be messy, and the easiest approach would be to write the input as a sum of phasors $cos(2\pi ft) = 0.5(e^{j2\pi ft} + e^{-j2\pi ft})$

$$
\begin{aligned}
&= e^{j2\pi ft} H(f) &(3.21)\\
&= e^{j2\pi ft} r e^{j\phi}) &(3.22)
\end{aligned}
$$

$H(f)$ represents the shift in magnitude and phase of the complex sinusoid, and is equivalent to the frequency response $re^{j\phi}$ calculated above.

To work out values for r and ϕ is an exercise in complex algebra [5].

$$H(f) = 1 + \beta cos(2\pi f/f_s) - j\beta sin(2\pi f/f_s) \tag{3.23}$$

$$
\begin{aligned}
r = |H(f)| &= [1 + \beta cos(2\pi f/f_s)]^2 + [\beta sin(2\pi f/f_s)]^2 &(3.24)\\
&= 1 + 2\beta cos(2\pi f/f_s) + \beta^2 cos^2(2\pi f/f_s) + \beta^2 sin^2(2\pi f/f_s) &(3.25)\\
&= 1 + \beta^2 + 2\beta cos(2\pi f/f_s) &(3.26)
\end{aligned}
$$

$$\phi = arg(H(f)) = arctan\left(\frac{-\beta sin(2\pi f/f_s)}{1 + \beta cos(2\pi f/f_s)}\right) \tag{3.27}$$

r and ϕ are plotted as a function of f in Figure ...

3.3 FREQUENCY RESPONSE DEPENDS ON LOCATION OF POLES AND ZEROS

In this section, we find the frequency response of an LTI system $H(z)$ as a function of the pole and zero locations. For the case of one pole and one zero,

$$
\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} &(3.28)\\
&= b_0 \frac{z - z_1}{z - p_1} &(3.29)\\
&= \frac{Y(z)}{X(z)} &(3.30)\\
& &(3.31)
\end{aligned}
$$

The frequency response

$$
\begin{aligned}
H(f) &= H(z)|_{z=e^{j2\pi f/f_s}} &(3.32)\\
&= b_0 \frac{e^{j2\pi f/f_s} - z_1}{e^{j2\pi f/f_s} - p_1} &(3.33)\\
& &(3.34)
\end{aligned}
$$

[5] Recall for $w = u + jv$, $|w| = \sqrt{u^2 + v^2}$, $arg(w) = arctan(v/u)$

For any particular value of f, both the numerator and denominator are complex numbers whose value depends on f, z_1, p_1. Thus we can write [6]

$$\begin{aligned} e^{j2\pi f/f_s} - z_1 &= r_1 e^{j\phi_1} && (3.39)\\ e^{j2\pi f/f_s} - p_1 &= d_1 e^{j\theta_1} && (3.40)\\ &= e^{j2\pi f/f_s} - p_1 && (3.41)\\ & && (3.42) \end{aligned}$$

and

$$\begin{aligned} H(f) &= b_0 \frac{r_1 e^{j\phi_1}}{d_1 e^{j\theta_1}} && (3.43)\\ &= b_0 \frac{r_1}{d_1} e^{j\phi_1 - \theta_1} && (3.44)\\ & && (3.45) \end{aligned}$$

which is in the form of a frequency response, with magnitude response

$$\begin{aligned} |H(f)| &= b_0 \frac{r_1}{d_1} && (3.46)\\ & && (3.47) \end{aligned}$$

and phase response

$$\begin{aligned} arg(H(f)) &= \phi_1 - \theta_1 && (3.48)\\ & && (3.49) \end{aligned}$$

To find the frequency response $H(f)$, we can simply evaluate (??) by computer for all frequencies $-f_s/2 < f < f_s/2$, and plot a curve of the magnitude response and phase response.

However, we can get an intuitive insight by considering a geometrical interpretation of (??).

In Figure xxx, we show the unit circle with a pole z_1 and zero z_1. At any point $z = e^{j2\pi f/f_s}$ along the unit circle, the value of r_1 is the length of the vector between $e^{j2\pi f/f_s}$ and z_1, i.e. the vector subtraction of the vectors $e^{j2\pi f/f_s}$ and z_1 in the complex plane. Similarly, the value of d_1 is the length

[6] We find r_1, ϕ_1 as follows using $z_1 = x_1 + jy_1$

$$\begin{aligned} e^{j2\pi f/f_s} - z_1 &= cos(\pi f/f_s) + jsin(2\pi f/f_s) - (x_1 + jy_1) && (3.35)\\ &= [cos(\pi f/f_s) - x_1] + j[sin(2\pi f/f_s) - y_1] && (3.36) \end{aligned}$$

and

$$r = \sqrt{cos(\pi f/f_s) - x_1]^2 + [sin(2\pi f/f_s) - y_1]^2} \qquad (3.37)$$

$$\phi = arctan\left(\frac{sin(\pi f/f_s) - y_1}{cos(2\pi f/f_s) - x_1}\right) \qquad (3.38)$$

of the vector between $e^{j2\pi f/f_s}$ and p_1. The magnitude response $b_0 r_1/d_1$ is the ratio of the lengths of the two vectors r_1, d_1 (multiplied by b_0). As we move f along the unit circle from 0 to $f_s/2$, we can observe the lengths of the vectors r_1, d_1, estimate their ratio, and thus obtain an estimate of the magnitude response. The phase response is obtained by observing that at any point $z = e^{j2\pi f/f_s}$ along the unit circle, the value of θ_1 is the angle of the vector between $e^{j2\pi f/f_s}$ and z_1 with respect to the positive x-(real) axis, and similarly for ϕ_1.

We can consider a two-pole two-zero filter, and write

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} && (3.50) \\ &= b_0 \frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)} && (3.51) \\ &= \frac{Y(z)}{X(z)} && (3.52) \\ & && (3.53) \end{aligned}$$

with frequency response

$$\begin{aligned} H(f) &= H(z)|_{z=e^{j2\pi f/f_s}} && (3.54) \\ &= b_0 \frac{(e^{j2\pi f/f_s}-z_1)(e^{j2\pi f/f_s}-z_2)}{(e^{j2\pi f/f_s}-p_1)(e^{j2\pi f/f_s}-p_2)} && (3.55) \\ &= b_0 \frac{r_1 e^{j\phi_1} r_2 e^{j\phi_2}}{r_1 e^{j\theta_1} d_1 e^{j\theta_2}} && (3.56) \\ &= b_0 \frac{r_1 r_2}{d_1 d_2} e^{j\phi_1+\phi_2-\theta_1-\theta_2} && (3.57) \\ & && (3.58) \end{aligned}$$

with magnitude response

$$\begin{aligned} |H(f)| &= b_0 \frac{r_1 r_2}{d_1 d_2} && (3.59) \\ & && (3.60) \end{aligned}$$

and phase response

$$\begin{aligned} arg(H(f)) &= \phi_1 + \phi_2 - \theta_1 - \theta_2 && (3.61) \\ & && (3.62) \end{aligned}$$

From Figure xxy, we can estimate the magnitude response by moving f from 0 to $f_s/2$ and estimating the ratio $r_1 r_2/d_1 d_2$.

In what follows, we estimate the magnitude response for filters with various pole and zero locations. The phase response can be estimated also. Note that we always have at least as many poles as zeros, so that the filters are causal.

3.3.0.20 one pole, no zeros For one pole and no zero,

$$H(f) = \frac{b_0}{e^{j2\pi f/f_s} - p_1} = \frac{b_0}{d_1} e^{-j\theta_1} \tag{3.63}$$

with magnitude response

$$|H(f)| = \frac{b_0}{d_1} \tag{3.64}$$

For the pole $p_1 = 0.9$ on the positive real axis (Figure), we see that d_1 is smallest for $f = 0$, and becomes larger for increasing f. Since d_1 is in the denominator of the expression for $H(f)$, we find $H(f)$ is largest for $f = 0$ and decreases with increasing f. This is a *low pass* filter which passes low frequencies near $f = 0$ and rejects high frequencies. For the pole $0 < p_1 < 0.9$, the slope of the frequency response curve is less steep, so the cutoff of high frequencies is less abrupt. For the pole $1 > p_1 > 0.9$, the slope of the frequency response curve is greater, so the cutoff of high frequencies is more abrupt. for the pole $p_1 > 1$, the filter becomes unstable.

For the pole $p_1 = 0$, we see that d_1 is constant for all frequencies $f_s/2 < f < f_s/2$ and thus $H(f)$ is constant for $f = 0$. This is an *all pass* filter which passes all frequencies with equal magnitude [7]

For the pole $p_1 = -0.9$ on the negative real axis (Figure), we see that d_1 is smallest for $f = \pm f_s/2(\omega = \pm\pi)$, and becomes larger for increasing f. Since d_1 is in the denominator of the expression for $H(f)$, we find $H(f)$ is largest for $f = 0$ and decreases with increasing f. This is a *high pass* filter which passes high frequencies near $f_s/2$ and reject low frequencies. For the pole $-0.9 < p_1 < 0$ ($-1 < p_1 < -0.9$), the slope of the frequency response curve is less (greater), so the cutoff of high frequencies is less(more) abrupt.

p_1	p_2	z_1	z_2	response
0.9				lowpass
0				allpass
-0.9				highpass

3.3.0.21 one pole, one zero For a filter with one pole and one zero,

$$\begin{aligned} H(f) &= b_0 \frac{e^{j2\pi f/f_s} - z_1}{e^{j2\pi f/f_s} - p_1} && (3.65) \\ &= b_0 \frac{r_1}{d_1} e^{j\phi_1 - \theta_1} && (3.66) \\ & && (3.67) \end{aligned}$$

For the pole $p_1 = 0.9$ on the positive real axis (Figure), and the zero $z_1 = -1$ on the unit circle, we get the same low pass response as before, except that now the frequency response is precisely zero at $f = f_s/2$.

[7] we will see later that such a filter still does something. For this example the output is delayed by one sample compared to the input, but otherwise unchanged.

If we move the pole p_1 to 0 and leave the zero $z_1 = -1$ (Figure), $d_1 = 1$ is constant for all frequencies, and r_1 is largest for $f = 0$ and smallest (actually 0) for $f = f_s/2$. Thus $H(f)$ is largest for $f = 0$ and 0 for $f = f_s/2$. For this case, we again get a low pass response, but with a higher cutoff frequency and a different shape than before.

For the pole $p_1 = -0.9$ on the negative real axis (Figure), and the zero $z_1 = +1$ on the unit circle, we get the same high pass response as before, except that now the frequency response is precisely zero at $f = 0$.

If we move the pole p_1 to 0 and leave the zero $z_1 = +1$ (Figure), $d_1 = 1$ is constant for all frequencies, and r_1 is largest for $f = f_s/2$ and smallest (actually 0) for $f = 0$. Thus $H(f)$ is largest for $f = f_s/2$ and 0 for $f = 0$. For this case, we again get a high pass response, but with a lower cutoff frequency and a different shape than before.

p_1	p_2	z_1	z_2	response
0.9		-1		lowpass
0		-1		lowpass
-0.9		-1		lowpass
0.9		1		highpass
0		1		highpass
-0.9		1		highpass

3.3.0.22 two poles, no zeros For a complex conjugate pair of poles and no zeros,

$$H(f) = \frac{b_0}{(e^{j2\pi f/f_s} - p_1)(e^{j2\pi f/f_s} - p_1^*)} = \frac{b_0}{d_1 d_2} e^{-j(\theta_1+\theta_2)} \tag{3.68}$$

with magnitude response

$$|H(f)| = \frac{b_0}{d_1 d_2} \tag{3.69}$$

For the pole $p_1 = 0.9j$ on the positive imaginary axis and $p_1^* = -0.9$ (Figure), we see that d_1 is smallest for $f = f_s/4, \omega = \pi/2$, and becomes larger for f greater than or less than $f_s/4$. d_2 doesn't change very much over the whole range $0 < f < f_s/2$, and thus does not affect the magnitude response significantly. Since d_1 is in the denominator of the expression for $H(f)$, we find $H(f)$ is largest for $f = f_s/4$ and decreases as f moves away from $f_s/4$. This is a *bandpass* filter which passes frequencies near the *center frequency* $f_s/4$ and rejects all other frequencies removed from the center frequency.

For the poles $p_1 = 0.7 + j0.7, p_2 = 0.7 - j0.7$ (Figure), d_1 is smallest for $f = f_s/8, \omega = \pi/4$, and becomes larger for f greater than or less than $f_s/8$. This is a bandpass filter at $f_s/8$. When the poles are on the positive real axis and the maximum $H(f)$ is at $f = 0$, then the bandpass filter is centered at $f = 0$ and so we call it a low pass filter. As the complex pair of poles is moved along (but inside) the unit circle the frequency of maximum $|H(f)|$

moves from $f = 0$ to $f_s/2$. For this case, $|p_1| < 1$ and $|p_2| < 1$ for the filter to be stable.

When the poles are on the real axis, then only one pole need have magnitude less than 1, the other pole can have magnitude greater than 1. If $p_2 = 1/p_1$, then an interesting geometrical result is that the product $d_1 d_2$ is constant for all frequencies $-f_s/2 < f < f_s/2$ (Figure). Thus $H(f)$ is constant, and we have an it allpass filter.

Table xxx Pole-zero locations and corresponding frequency response

p_1	p_2	z_1	z_2	response
$0.9j$	$-0.9j$			bandpass
$0.7 + 0.7j$	$0.7 - 0.7j$			bandpass
0.9	0.9			lowpass
0.5	2.0			allpass
-0.5	-2.0			allpass

3.3.0.23 two poles, two zeros For a filter with a complex conjugate pair of poles and two zeros,

$$H(f) = b_0 \frac{(e^{j2\pi f/f_s} - z_1)(e^{j2\pi f/f_s} - z_2)}{(e^{j2\pi f/f_s} - p_1)(e^{j2\pi f/f_s} - p_1^*)} \quad (3.70)$$

$$= b_0 \frac{r_1 r_2}{d_1 d_2} e^{j\phi_1 + \phi_2 - \theta_1 - \theta_2} \quad (3.71)$$

$$(3.72)$$

For the poles $p_1 = 0.9j, p_1* = -0.9j$, and the zeros $z_1 = j, z_2 = -j$ (Figure), as before, d_1 is smallest for $f = f_s/4, \omega = \pi/2$, but $r_1 = 0$ and thus $|H(f)| = 0$ at $f = f_s/4$. For frequencies f greater than or less than $f_s/4$, $|H(f)|$ increases. This is a *bandstop* or *notch* filter which rejects frequencies near the center frequency $f_s/4$ and passes all other frequencies removed from the center frequency. The notch is made narrower by moving the poles closer to the unit circle, and broader by moving them away.

Another way to get a bandstop filter is to place the poles $p_1 = p_2 = 0$ and leave the zeros at $z_1 = j, z_2 = -j$. In this case, $d_1 = d_2 = 1$ is a constant for all frequencies, and $r_1 = r_2 = 0$ at $f = f_s/4$ and increase to non-zero values for f on either side of $f_s/4$.

A third way to get a bandstop filter is to place the poles $p_1 = +1, p_2 = -1$ and leave the zeros at $z_1 = j, z_2 = -j$.

For a two-pole, two-zero filter, we can also obtain all of the responses above, e.g. lowpass, highpass, bandpass, allpass in several ways.

Table xxx Pole-zero locations and corresponding frequency response

p_1	p_2	z_1	z_2	response
$0.9j$	$-0.9j$	j	$-j$	bandstop
1	-1	j	$-j$	bandstop
0	$0j$	j	$-j$	bandstop
$0.9j$	$-0.9j$	1	-1	bandpass
$0.9j$	$-0.9j$	0	0	bandpass
0	0	1	-1	bandpass
0.9	0.9	-1	-1	lowpass
0.9	0.9	0	0	lowpass
0	0	-1	-1	lowpass
-0.9	-0.9	1	1	highpass
-0.9	-0.9	0	0	highpass
0	0	1	1	highpass
0.5	2.0	0	0	allpass
0.8	1.25	0	0	allpass
-0.5	-2.0	0	0	allpass

3.3.0.24 3 poles, 3 zeros Consider a filter with 3 poles and 3 zeros, choosing the poles to be one complex conjugate pair, and one real pole.

For the poles near 1 on the unit circle, e.g. $p_1 = 0.9, p_2 = 0.9 + j0.1, p_3 = 0.9 - 0.1j$ and $z_1 = z_2 = z_3 = -1$ (Figure), we get a low pass filter, but with a steeper rolloff than with only one or two poles. The precise location of the poles can be optimized according to the desired shape of the magnitude and phase response. One such shape is the so-called Butterworth response, discussed later in Section xxx.

Similarly, for the poles near -1 on the unit circle, e.g. $p_1 = -0.9, p_2 = -0.9 + j0.1, p_3 = -0.9 - 0.1j$ and $z_1 = z_2 = z_3 = +1$ (Figure), we get a high pass filter, but with a steeper rolloff than with only one or two poles.

The other filter types (bandstop, bandpass) can also be obtained by following the same ideas as for the 2-pole 2-zero case.

3.3.0.25 4 poles, 4 zeros For a filter with 4 poles and 4 zeros, all of the shapes above can be obtained and modified. In addition, by using p_1, p_2, z_1, z_2 to obtain one shape, and p_3, p_4, z_3, z_4 to obtain another shape, interesting and complicated shapes can be obtained.

For example, by choosing $p_1 = 0.7 + j0.7, p_2 = 0.7 - j0.7$, $p_3 = -0.7 + j0.7, p_4 = -0.7 - j0.7$, we obtain a double bandpass with two peaks in $H(f)$ at $f = f_s/8, 3f_s/8(\omega = \pi/4, 3\pi/4)$.

The design of filters with specified characteristics such as Butterworth, or Bessel, is beyond the scope of this chapter.

3.4 FILTER DESIGN AND ANALYSIS

In the previous section, we have learned how to design an LTI system (filter) to obtain a desired frequency response by choosing the locations of the poles and zeros.

In this section, we analyze each filter to obtain the LTI difference equations, impulse response and transfer function. and to set the filter gain equal to unity at a preferred frequency.

The general difference equations

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] \tag{3.73}$$

$$+ \sum_{k=0}^{M} b_k x[n-k] \tag{3.74}$$

have a corresponding transfer function

$$H(z) = \frac{Y(z)}{X(z)} \tag{3.75}$$

$$= b_0 \frac{(z-z_1)(z-z_2)...(z-z_M}{(z-p_1)(z-p_2)...(z-p_N)} \tag{3.76}$$

In what follows we consider only up to 2 poles and 2 zeros with difference equations

$$y[n] = a_1 y[n-1] a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \tag{3.77}$$

and transfer function

$$H(z) = \frac{Y(z)}{X(z)} \tag{3.78}$$

$$= b_0 \frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)} \tag{3.79}$$

3.4.1 one pole, no zeros

$$H(z) = b_0 \frac{1}{z-p_1} \tag{3.80}$$

3.4.1.1 low pass and high pass We first consider the case $p_1 = \alpha$, which is lowpass for $0 < \alpha < 1$, highpass for $-1 < \alpha < 0$. Thus

$$H(z) = b_0 \frac{1}{z-\alpha} = b_0 \frac{z^{-1}}{1-\alpha z^{-1}} = \frac{Y(z)}{X(z)} \tag{3.81}$$

From the z-transform pair

$$F(z) = \frac{c}{1-pz^{-1}} = \frac{cz}{z-p} \leftrightarrow f[n] = cp^n u[n] \tag{3.82}$$

and the shifting property

$$z^{-m}F(z) \leftrightarrow f[n-m] \tag{3.83}$$

with $m = 1$, we obtain

$$h[k] = b_0\alpha^{k-1}u[k-1] \tag{3.84}$$

The characteristics of this impulse response depend on the value of α as shown in Section xxx. Note that the high pass impulse responses (for $-1 < \alpha < 0$) are alternating. Rearranging (??)

$$Y(z)[1-\alpha z^{-1}] = b_0 z^{-1}X(z) \tag{3.85}$$

Taking the inverse z-transform term by term

$$y[n] - \alpha y[n-1] = b_0 x[n-1] \tag{3.86}$$

and rearranging

$$y[n] = \alpha y[n-1] + b_0 x[n-1] \tag{3.87}$$

Comparing with the difference equation (??), $\alpha = a_1$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.88}$$

where

$$B = [0b_0], A = [1 + \alpha] \tag{3.89}$$

Recalling that

$$H(z) = b_0\frac{1}{z-\alpha} \tag{3.90}$$

To set the gain b_0 equal to one at $f = 0$ or $z = e^{j2\pi f/f_s} = 1$ for the low pass filter, we must have $|H(z)|_{z=1} = 1$ or $b_0 = 1 - \alpha$. To set the gain b_0 equal to one at $f = f_s/2$ or $z = e^{j2\pi f/f_s} = -1$ for the high pass filter, we must have $|H(z)|_{z=-1} = 1$ or $b_0 = -1 - \alpha$.

3.4.1.2 allpass We next consider the case $p_1 = 0$, allpass. Thus

$$H(z) = b_0\frac{1}{z} = b_0 z^{-1} = \frac{Y(z)}{X(z)} \tag{3.91}$$

Using the z-transform pair

$$z^{-k} \leftrightarrow \delta[n-k] \tag{3.92}$$

with $k = 1$, we obtain

$$h[n] = b_0\delta[n-1] \tag{3.93}$$

which is just a delay by one time sample. Rearranging ()

$$Y(z) = b_0 z^{-1}X(z) \tag{3.94}$$

Taking the inverse z-transform term by term

$$y[n] = b_0 x[n-1] \tag{3.95}$$

which shows that the output is delayed by one time sample. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.96}$$

where

$$B = [0 b_0], A = [1] \tag{3.97}$$

The frequency response $|H(z)| = 1$ for any f or any z on the unit circle.

3.4.2 one pole, one zero

$$H(z) = b_0 \frac{z - z_1}{z - p_1} \tag{3.98}$$

3.4.2.1 lowpass and highpass We first consider the case $p_1 = \alpha, z_1 = 0$, which is lowpass for $0 < \alpha < 1$, highpass for $-1 < \alpha < 0$. Thus

$$H(z) = b_0 \frac{z}{z - \alpha} = b_0 \frac{1}{1 - \alpha z^{-1}} = \frac{Y(z)}{X(z)} \tag{3.99}$$

Using (refzt1),

$$h[n] = b_0 \alpha^n u[n] \tag{3.100}$$

The characteristics of this impulse response depend on the value of α as shown in Section xxx. Rearranging (),

$$Y(z)[1 - \alpha z^{-1}] = b_0 X(z) \tag{3.101}$$

Taking the inverse z-transform term by term

$$y[n] - \alpha y[n-1] = b_0 x[n] \tag{3.102}$$

and rearranging to obtain the difference equation

$$y[n] = \alpha y[n-1] + b_0 x[n] \tag{3.103}$$

Comparing with the difference equation (**??**), $\alpha = a_1$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.104}$$

where

$$B = [b_0], A = [1 - \alpha] \tag{3.105}$$

Recalling that

$$H(z) = b_0 \frac{z}{z - \alpha} \tag{3.106}$$

To set the gain b_0 equal to one at $f = 0$ or $z = e^{j2\pi f/f_s} = 1$ for the low pass filter, we must have $|H(z)|_{z=1} = 1$ or $b_0 = 1 - \alpha$. To set the gain b_0 equal to one at $f = f_s/2$ or $z = e^{j2\pi f/f_s} = -1$ for the high pass filter, we must have $|H(z)|_{z=-1} = 1$ or $b_0 = 1 + \alpha$.

3.4.2.2 lowpass We next consider the case $p_1 = \alpha, 0 < \alpha < 1, z_1 = -1$, which is lowpass.

$$H(z) = b_0 \frac{z+1}{z-\alpha} = b_0 \frac{z}{z-\alpha} + b_0 \frac{1}{z-\alpha} \tag{3.107}$$

$$= b_0 \frac{1+z^{-1}}{1-\alpha z^{-1}} = \frac{Y(z)}{X(z)} \tag{3.108}$$

Using (??),(??),

$$h[k] = b_0 \alpha^k u[k] + b_0 \alpha^{k-1} u[k-1] \tag{3.109}$$

Rearranging,

$$Y(z)[1 - \alpha z^{-1}] = b_0[1 + z^{-1}]X(z) \tag{3.110}$$

and taking the inverse z-transform term by term

$$y[n] - \alpha y[n-1] = b_0 x[n] + b_0 x[n-1] \tag{3.111}$$

we get the difference equations

$$y[n] = \alpha y[n-1] + b_0 x[n] + b_0 x[n-1] \tag{3.112}$$

Comparing with the difference equation (??), $\alpha = a_1$ and $b_1 = b_0$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.113}$$

where

$$B = b_0[11], A = [1 - \alpha] \tag{3.114}$$

Recalling that

$$H(z) = b_0 \frac{z+1}{z-\alpha} \tag{3.115}$$

To set the gain b_0 equal to one at $f = 0$ or $z = e^{j2\pi f/f_s} = 1$ for the low pass filter, we must have $|H(z)|_{z=1} = 1$ or $b_0 = (1-\alpha)/2$.

3.4.2.3 highpass Similarly we can consider the case $p_1 = \alpha, -1 < \alpha < 0, z_1 = 1$, which is highpass, to obtain the difference equations

$$y[n] = \alpha y[n-1] + b_0 x[n] - b_0 x[n-1] \tag{3.116}$$

Comparing with the difference equation (??), $\alpha = a_1$ and $b_1 = -b_0$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.117}$$

where

$$B = b_0[1 - 1], A = [1 - \alpha] \tag{3.118}$$

Recalling that

$$H(z) = b_0 \frac{z-1}{z-\alpha} \tag{3.119}$$

To set the gain b_0 equal to one at $f = f_s/2$ or $z = e^{j2\pi f/f_s} = -1$ for the low pass filter, we must have $|H(z)|_{z=-1} = 1$ or $b_0 = (1+\alpha)/2$.

3.4.2.4 lowpass and highpass We next consider the case $p_1 = 0, z_1 = \beta$, which is highpass for $0 < \beta$, lowpass for $\beta < 0$. Thus

$$H(z) = b_0 \frac{z-\beta}{z} = b_0 \frac{1-\beta z^{-1}}{1} = \frac{Y(z)}{X(z)} \tag{3.120}$$

Using (??) with $k = 0$ and again with $k = 1$,

$$h[n] = b_0(\delta[n] - \delta[n-1] \tag{3.121}$$

Rearranging the expression for $H(z)$,

$$Y(z) = b_0[1 - \beta z^{-1}]X(z) \tag{3.122}$$

and taking inverse z-transform term by term, we get the difference equation

$$y[n] = b_0 x[n] - b_0 \beta x[n-1] \tag{3.123}$$

Comparing with the difference equation (??), $b_1 = b_0\beta$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.124}$$

where

$$B = b_0[1\beta], A = [1 \quad] \tag{3.125}$$

If $\beta = 1$,

$$y[n] = b_0(x[n] - x[n-1]) \tag{3.126}$$

which is a differentiator. If $\beta = -1$,

$$y[n] = b_0(x[n] + x[n-1]) \tag{3.127}$$

which is an averager.

3.4.3 two poles, no zeros

$$H(z) = C\frac{1}{(z-p_1)(z-p_2)} \tag{3.128}$$

3.4.3.1 bandpass Consider the case of a complex conjugate pair $p_1 = p, p_2 = p^*$, which is bandpass for $|p| < 1$ and unstable for $|p| \geq 1$. Thus

$$H(z) = C\frac{1}{(z-p)(z-p^*)} = C\frac{z^{-2}}{(1-pz^{-1})(1-p^*z^{-1})} = \frac{Y(z)}{X(z)} \tag{3.129}$$

To take the inverse z-transform Breaking $H(z)/z$ into partial fractions.

$$\frac{H(z)}{z} = Cz^{-1}\left(\frac{c}{z-p} + \frac{c^*}{z-p*}\right) \tag{3.130}$$

$$H(z) = Cz^{-2}\left(\frac{cz}{z-p} + \frac{c^* z}{z-p*}\right) \tag{3.131}$$

From the previous chapter and (??) with $k = 2$, and given $p = re^{j\theta}, c = \rho e^{j\phi}$

$$h[n] = 2Ccr^(n-2)cos((n-2)\theta + \phi)u[n-2] \tag{3.132}$$

The characteristics of this impulse response depend on the value of r as shown in Section xxx. Rearranging (??

$$Y(z)(1 - pz^{-1})(1 - p * z^{-1}) = Cz^{-2}X(z) \tag{3.133}$$

$$Y(z)[1 - 2Re(p)z^{-1} + |p|^2 z^{-2}] = Cz^{-2}X(z) \tag{3.134}$$

Taking the inverse z-transform term by term

$$y[n] - 2Re(p)y[n-1] + |p|^2 y[n-2] = Cx[n-2] \tag{3.135}$$

and rearranging to get the difference equation

$$y[n] = 2Re(p)y[n-1] - |p|^2 y[n-2] + Cx[n-2] \tag{3.136}$$

Comparing with the difference equation (??), $a_1 = 2Re(p), a_2 = |p|^2, b_0 = 0, b_1 = 0, b_2 = C$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.137}$$

where

$$B = [00C], A = [1 \quad -2Re(p) \quad |p|^2] \tag{3.138}$$

3.4.3.2 lowpass and highpass Consider the case of a double pole on the real axis, i.e. a complex conjugate pair $p_1 = p, p_2 = p^*$ with $p = re^{j\theta}$ and $Im(p) = 0$ (i.e. $\theta = 0$ or π). This case is lowpass for $0 < p < 1$ and highpass for $-1 < p < 0$. and unstable for $|p| \geq 1$. Thus

$$H(z) = C\frac{1}{(z-p)^2} = Cz^{-1}\frac{z}{(z-p)^2} = C\frac{z^{-2}}{(1-pz^{-1})^2} = \frac{Y(z)}{X(z)} \tag{3.139}$$

To take the inverse z-transform we use the z-transform pair

$$\frac{cz}{(z-p)^2} \leftrightarrow cnp^{n-1}u[n-1] \tag{3.140}$$

from the previous chapter and the shifting property (??) with $k = 1$ to obtain

$$h[n] = C(n-1)p^{n-2}u[n-2] \tag{3.141}$$

Note that this is equivalent to the $h[n]$ obtained for a complex conjugate pair, but with $\theta = 0$ or π. The characteristics of this impulse response depend on the value of p as shown in Section xxx. Rearranging (??

$$Y(z)(1 - pz^{-1})^2 = Cz^{-2}X(z) \tag{3.142}$$

$$Y(z)[1 - 2pz^{-1} + p^2 z^{-2}] = Cz^{-2}X(z) \tag{3.143}$$

Taking the inverse z-transform term by term

$$y[n] - 2py[n-1] + p^2 y[n-2] = Cx[n-2] \tag{3.144}$$

and rearranging to get the difference equation

$$y[n] = 2py[n-1] - p^2 y[n-2] + Cx[n-2] \tag{3.145}$$

Comparing with the difference equation (??), $a_1 = 2p, a_2 = p^2, b_0 = 0, b_1 = 0, b_2 = C$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.146}$$

where

$$B = [00C], A = [1 \quad -2p \quad p^2] \tag{3.147}$$

The filter should be scaled to unity output at a specified frequency, e.g. $f = 0$ for lowpass, or $f = f_s/2$ for highpass.

3.4.4 two poles, two zeros

$$H(z) = C\frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \tag{3.148}$$

Consider the case of real $z_1 = w, z_2 = -w$, and a complex conjugate pair $p_1 = p, p_2 = p^*$. Thus

$$H(z) = C\frac{(z - w)(z + w)}{(z - p)(z - p^*)} = C\frac{(1 - wz^{-1})(1 + wz^{-1})}{(1 - pz^{-1})(1 - p^* z^{-1})} = C\frac{Y(z)}{X(z)} \tag{3.149}$$

To take the inverse z-transform, break $H(z)/z$ into partial fractions.

$$\frac{H(z)}{z} = Cz^{-1}\left(\frac{c}{z - p} + \frac{c^*}{z - p*}\right) \tag{3.150}$$

$$H(z) = Cz^{-2}\left(\frac{cz}{z - p} + \frac{c^* z}{z - p*}\right) \tag{3.151}$$

From the previous chapter and (??) with $k = 2$

$$h[n] = 2Ccr^{(n-2)}cos((n-2)\theta + \phi)u[n-2] \tag{3.152}$$

The characteristics of this impulse response depend on the value of r as shown in Section xxx. Rearranging (??

$$Y(z)(1 - pz^{-1})(1 - p * z^{-1}) = Cz^{-2}X(z) \tag{3.153}$$

$$Y(z)[1 - 2Re(p)z^{-1} + |p|^2 z^{-2}] = Cz^{-2}X(z) \tag{3.154}$$

Taking the inverse z-transform term by term

$$y[n] - 2Re(p)y[n-1] + |p|^2 y[n-2] = Cx[n-2] \tag{3.155}$$

and rearranging to get the difference equation

$$y[n] = 2Re(p)y[n-1] - |p|^2 y[n-2] + Cx[n-2] \tag{3.156}$$

Comparing with the difference equation (??), $a_1 = 2Re(p), a_2 = |p|^2, b_0 = 0, b_1 = 0, b_2 = C$. The filter is written in Matlab as

$$\mathbf{y} = \mathbf{filter}[\mathbf{B}, \mathbf{A}, \mathbf{x}] \tag{3.157}$$

where

$$B = [00C], A = [1 \quad -2Re(p) \quad |p|^2] \tag{3.158}$$