Some Useful Summations

1. Geometric Progression

\[ \sum_{i=0}^{\infty} a^i = \frac{1}{1-a} \]

|a| < 1

2.

\[ \sum_{i=0}^{N} a^i = \begin{cases} 
0 & \text{if } a = 0 \\
N + 1 & \text{if } a = 1 \\
\frac{1-a^{N+1}}{1-a} & \text{otherwise}
\end{cases} \]

3.

\[ \sum_{i=0}^{\infty} ia^i = \frac{a}{(1-a)^2} \]

|a| < 1

(Take the derivative in Example 1 above)

4.

\[ \sum_{i=1}^{N} i = \frac{N(N+1)}{2} \]

5.

\[ \sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6} \]

6. Sinc Function

\[ \text{sinc } x = \frac{\sin x}{x} \]

7. Taylor Series Expansion of an Exponential

\[ e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} \]
ECE161A – Discrete Time Linear Systems

1. A second course in signal processing.

2. Have seen discrete time signals and systems as well as Z-transforms.

3. Will study 2 new transforms: DTFT, DFT

4. Will use Matlab for signal processing applications

5. Will discuss filters and will filter signals in Matlab

6. Will learn Fast Fourier Transform (FFT) which is used in many practical applications such as spectrum analyzer, EKG systems, compression systems, etc.

7. In summary, we will cover an introduction to the important area of DIGITAL SIGNAL PROCESSING.

Definition: A discrete time signal is one that is defined only for discrete points in time (hourly, every second, etc.)

Ex.: An image on a computer is a discrete signal. It is defined only at discrete points in space, called pixels.

Ex.: Any signal on a computer which is a list of numbers is a discrete signal.

Ex.: A picture taken with a digital camera is a discrete signal.

Ex.: A DVD format movie is a discrete signal.

Ex.: An MP3 file is a discrete signal.
Chapter 2 – Discrete-Time Signals and Systems

We assume that we derived a discrete-time signal from a continuous time signal via sampling. Given \( x_a(t) \) to be a continuous time signal, \( x_a(nT) \) is the value of \( x_a(t) \) at \( t = nT \). The discrete-time signal \( x[n] \) is defined only for \( n \) an integer. So if we derive \( x[n] \) from \( x_a(t) \) by sampling every \( T \) seconds, where \( T \) is the sample period, we get:

\[
x_a(nT) = x_a(t)|_{t=nT}
\]

\[
x[n] = x_a(nT) = x_a(t)|_{t=nT}
\]

We will not necessarily assume that \( x[n] \) is a discrete amplitude signal. A signal that is both discrete time and discrete amplitude is known as a digital signal. You will see these in later communications courses but a well-known example of a digital signal is music on a compact disk.

Note that a discrete-time signal need not be generated by explicitly sampling a continuous-time signal. Some signals are inherently discrete time, such as computer bit sequences, and some signals are implicitly sampled, such as the daily DJIA or yearly temperature averages.

Discrete-Time Signals and Systems

Remember the square brackets!

Read (skim) the example of Euler integration in the book but we will cover difference equations in Chapter 10 and when we discuss the Z-transform. Euler integration approximates the area under a curve \( x(t) \) by the sum of rectangular areas.
Discrete-Time Unit Step Function

\[ u[n] = \begin{cases} 
1, & n \geq 0 \\
0, & n < 0 
\end{cases} \]

Notice that here, the unit step is defined at \( n = 0 \), unlike for continuous time.

The time-shifted unit step function \( u[n - n_0] \) is:

\[ u[n - n_0] = \begin{cases} 
1, & n \geq n_0 \\
0, & n < n_0 
\end{cases} \]

\( n_0 = 3 \)
Discrete-Time Unit Impulse Function

\[ \delta[n] = \begin{cases} 
1, & n = 0 \\
0, & n \neq 0
\end{cases} \]

Here, there is no difficulty in defining the impulse as we had in continuous time.

Shifted unit impulse:

\[ \delta[n - n_0] = \begin{cases} 
1, & n = n_0 \\
0, & n \neq n_0
\end{cases} \]
The summation is the discrete-time analog of the running integral in continuous time, and the first difference is the analog of the derivative. With these analogies, the unit impulse has essentially the same behavior in discrete and continuous time, including the sifting property.

\[
\begin{align*}
\text{Continuous time} & & \text{Discrete time} \\
u(t) &= \int_{-\infty}^{t} \delta(\tau) d\tau & u[n] &= \sum_{k=-\infty}^{n} \delta[k] \\
\delta(t) &\equiv \frac{d}{dt}u(t) & \delta[n] &= u[n] - u[n-1] \\
x(t)\delta(t-t_0) &= x(t_0)\delta(t-t_0) & x[n]\delta[n-n_0] &= x[n_0]\delta[n-n_0] \\
\int_{-\infty}^{\infty} x(t)\delta(t-t_0) dt &= x(t_0) & \sum_{n=-\infty}^{\infty} x[n]\delta[n-n_0] &= x[n_0]
\end{align*}
\]
Recall that continuous-time signals could be represented by an equation (which might be defined in regions) or a graph. Discrete-time signals can be represented in these ways, but also using a table. For example:

\[ x[n] = u[n] - u[n - 4] \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( x[n] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \leq -1 )</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>( \geq 4 )</td>
<td>0</td>
</tr>
</tbody>
</table>

\( u[n] \)

\( u[n-4] \)
Operations on Discrete-Time Signals

Time reversal

\[ y[n] = x[m] \big|_{m=-n} = x[-n] \]

Ex: Reading an array of numbers backwards. Taking a digital image and looking at it upside down and reversed. Playing a CD backwards. This flips a signal about the vertical axis.
Time Scaling

\[ y[n] = x[m] \mid m = an = x[an] \] SPEED UP (\(|a| > 1\)) or SLOW DOWN (\(|a| < 1\)) by a factor of \(a\)!

Unlike continuous time, there are restrictions on \(a\)!

For speeding up (also known as “subsampling”), \(a\) must be an integer. Example: For \(a = 2\), you only take every other sample of \(x[n]\).

\[
y[0] = x[2] \\
\]

Find \(w_1[n] = x[2n]\) and \(w_2[n] = x[2n + 1]\).

\[
w_2[0] = x[1] \\
w_2[1] = x[3]
\]
For **slowing down** (expanding) a signal, you need $a = 1/K$ where $K$ is an integer.

Example: Let $K = 2$ ($a = 1/2$) and find $z[n] = b[n/2]$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$z[n]$</th>
<th>$b[n/2]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$z[0]$</td>
<td>$b[0]$</td>
</tr>
<tr>
<td>1</td>
<td>$z[1]$</td>
<td>??</td>
</tr>
<tr>
<td>3</td>
<td>$z[3]$</td>
<td>??</td>
</tr>
</tbody>
</table>

Values like $b[1/2]$ and $b[3/2]$ are not defined so how do we find $z[1]$ and $z[3]$?

One solution is to **INTERPOLATE**

A simple interpolation is

$$z[n] = \begin{cases} 
    b[n/2], & n \text{ even} \\
    1/2 \{b[(n-1)/2] + b[(n+1)/2]\}, & n \text{ odd}
\end{cases}$$

Interpolation can be used in a simple compression scheme – just send every other sample and fill in missing values.

Sometimes it works well and sometimes it doesn’t.
Back to our example, Form $z_1[n] = w_1[\frac{n}{2}]$ and $z_2[n] = w_2[\frac{n}{2}]$. Which do you prefer?

$$z_1[n] = w_1\left[\frac{n}{2}\right] = \mathcal{H}[2 \cdot m] = \mathcal{H}[n]$$

$$z_2[n] = w_2\left[\frac{n}{2}\right] = \mathcal{H}[2m+1] = \mathcal{H}[n+1]$$
Time Shifting

\[ y[n] = x[m]\big|_{m=n-n_0} = x[n-n_0] \]

Here, \( y[n] \) is a time-shifted version of the original signal \( x[n] \).

Ex.: Given \( x[n] = a^n u[n], |a| < 1 \), find and plot \( y[n] = x[n-3] \).
Combination of Time Shift and Time Scale

Ex. Find $u[3 - n]$.

There are two direct ways to find this (in addition to the book method):

1. Reverse then delay ($x[a(n + \frac{b}{a})]$):

   $$z[n] = u[-n]$$
   $$y[n] = z[n - 3] = u[-(n - 3)] = u[-n + 3]$$

2. Advance then reverse ($x[an + b]$):

   $$w[n] = u[n + 3]$$
   $$y[n] = w[-n] = u[-n + 3]$$
When we have a combination of shifting and scaling or reversing, you need to be careful. For example, if we try to form:
\[ z[n] = x[3 - 2n] = x[-2(n - \frac{3}{2})], \]

What does it mean to shift a signal by \( \frac{3}{2} \)?!! Method 1 does not work here; in other cases method 2 does not work. To be safe, plug in values in a table instead or as a check.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( z[n] )</th>
<th>( x[3 - 2n] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( z[0] )</td>
<td>( x[3] )</td>
</tr>
<tr>
<td>1</td>
<td>( z[1] )</td>
<td>( x[1] )</td>
</tr>
<tr>
<td>2</td>
<td>( z[2] )</td>
<td>( x[-1] )</td>
</tr>
<tr>
<td>-1</td>
<td>( z[-1] )</td>
<td>( x[5] )</td>
</tr>
<tr>
<td>-2</td>
<td>( z[-2] )</td>
<td>( x[7] )</td>
</tr>
</tbody>
</table>

Ex. Let \( x[n] = 2u[n + 2] \). Find \( z[n] = x[3 - 2n] \).
Ex.
Find $y[n] = x[2 - 2n]$:

$$x[2 - 2n] = x[-2(n - 1)]$$

$v[n] = x[-2n]$, then delay by 1. Or, just plug in values of $n$ in a table.
They cover *elementary operations* or *amplitude transformations* in the book, which you should review but it is similar to continuous time. Basically, given $y[n] = Ax[n] + B$, if $A < 0$, you get amplitude reversal; $|A|$ controls the amplitude scaling; and $B$ controls amplitude shifting. In addition, you should be able to do several other amplitude operations on signals, including: finding magnitude and phase (or real and imaginary parts) on complex signals, and adding or multiplying two signals. Remember: elementary amplitude operations require point-by-point repetition of an operation.

Example: Find $x[n] = (u[n + 1] - u[n - 5])(nu[2 - n])$
Characteristics of Discrete-Time Signals

Length of a Discrete-Time Signal

If a signal, $x$, is finite length, it is only defined on a finite time interval $-\infty < N_1 < n < N_2 < \infty$. Usually this interval is a superset of the time interval where $x$ has non-zero values. The process of lengthening a signal by adding more zeros is called zero padding.

On the other hand, we classify the infinite length signals to right-sided, left-sided, and two-sided signals. Can you define them?
Size (Norm) of a Signal

The $\mathcal{L}_p$ norm of a signal is defined

$$
||x||_p = \left( \sum_{n=-\infty}^{\infty} |x[n]|^p \right)^{1/p}
$$

When $p = 2$, this is related to the energy of the signal. How?

What can you say when $p = 1$?

$$
||x||_1 = \sum_{n=-\infty}^{\infty} |x[n]|
$$

It is also possible to define the norm for $p = \infty$, in which case,

$$
||x||_\infty = \max_n |x[n]|
$$

how large of a number at any point
Even and Odd Signals

Any discrete-time signal can be expressed as the sum of an even signal and an odd signal:

\[ x[n] = x_e[n] + x_o[n] \]

Even: \( x_e[n] = x_e[-n] \)
Odd: \( x_o[n] = -x_o[-n] \)

\[ x_e[n] = \frac{1}{2}(x[n] + x[-n]) \]
\[ x_o[n] = \frac{1}{2}(x[n] - x[-n]) \]
\[ x[n] = x_e[n] + x_o[n] \]
Ex. Given $x[n]$, find $x_e[n]$ and $x_o[n]$. 
Signals Periodic in Discrete Time

How do we tell if a discrete-time signal \(x[n]\) is periodic? That is, given \(n\) and \(N\) are integers, is there some period \(N > 0\) such that

\[x[n] = x[n + N].\]

Let’s examine a signal that did not necessarily come from sampling a continuous time signal:

\[x[n] = Ca^n\]

Let \(a = e^{j\Omega_0} \Rightarrow \text{then } x[n] = Ce^{j\Omega_0 n} \text{ is a complex exponential.} \]

If \(x[n]\) is periodic, then \(x[n] = x[n + N]\) and

\[Ce^{j\Omega_0 n} = Ce^{j\Omega_0 (n+N)} = Ce^{j\Omega_0 n} e^{j\Omega_0 N}\]

which implies \(e^{j\Omega_0 N} = 1.\)

When does this happen? Only if \(\Omega_0 N\) is an integer multiple of \(2\pi\), because \(e^{j2\pi} = 1\) and so, \(e^{j2\pi k} = 1\) for \(k\) an integer. Therefore,

\[\Omega_0 N = 2\pi k \quad \text{or} \quad \frac{\Omega_0}{2\pi} = \frac{k}{N}\]

\(\frac{\Omega_0}{2\pi}\) is the normalized frequency – it must be a RATIONAL number for the complex sinusoid to be periodic and there are \(k\) cycles of the sinusoid in \(N\) samples.

If \(\frac{\Omega_0}{2\pi}\) is irrational, then \(e^{j\Omega_0 n}\) is not periodic and we never get the samples repeated no matter how many samples we see. The same is true for sinusoids\(^1\) (since they are made of complex exponentials).

---

\(^1\)Note that the frequency \(\Omega_0\) is not always the same as the fundamental frequency. Since the period \(N\) must be an integer for a discrete-time signal, the fundamental frequency is \(2\pi/N = \frac{\Omega_0}{N}\), which is the same as \(\Omega_0\) only for cases where \(k = 1\). The fundamental period can be found as

\[N = \frac{2\pi k}{\Omega_0}\]

where \(k\) is the smallest integer such that \(N\) is an integer

or by normalizing frequency and reducing to the simplest ratio of integers

\[\frac{\Omega_0}{2\pi} = \frac{k}{N}\]
Determine which of the signals below are periodic. For the ones that are, find the fundamental period and fundamental frequency.

1. \( x_1[n] = e^{\frac{j\pi}{2}n} \)

2. \( x_2[n] = \sin\left(\frac{3\pi}{5}n + 1\right) \)

3. \( x_3[n] = \cos(2n - \pi) \)

4. \( x_4[n] = \cos(1.2\pi n) \)

5. \( x_5[n] = e^{-j\frac{n}{3}} \)
There is a major difference between discrete and continuous time. For continuous time, distinct values of frequency produce distinct sinusoids. For discrete-time, complex exponentials and sinusoids with frequency $\Omega_0$ and $\Omega_0 + 2\pi$ are indistinguishable.

Example: for integer $n$

$$e^{j\frac{\pi}{4}n} = e^{j\frac{2\pi}{4}n} = e^{-j\frac{2\pi}{4}n} = e^{j(200\pi + \frac{\pi}{4})n}$$

but for real $t$

$$e^{j\frac{\pi}{4}t} \neq e^{j\frac{2\pi}{4}t} \text{ etc.}$$

$\Rightarrow$ So only consider frequency interval of length $2\pi$ such as $[0, 2\pi)$, $[-\pi, \pi)$. We’ll visit this again with the Discrete Time Fourier Transform, but in the mean time we’ll highlight the difference by using the notation $\Omega$ for frequency (vs. $\omega$ for continuous time).
Given $x_1[n]$ periodic with period $N_1$ and $x_2[n]$ periodic with period $N_2$, let
$$x[n] = x_1[n] + x_2[n].$$

1. Is $x[n]$ periodic?

2. What is its period?

**ANSWERS**

1. Since $x_1[n]$ is periodic with period $N_1$, that means:
$$x_1[n] = x_1[n + N_1] = x_1[n + k_1 N_1].$$

Likewise,
$$x_2[n] = x_2[n + N_2] = x_2[n + k_2 N_2].$$

Therefore,
$$x[n] = x_1[n] + x_2[n] = x_1[n + k_1 N_1] + x_2[n + k_2 N_2].$$

For $x[n]$ to be periodic with period $N$, we require that:
$$k_1 N_1 = k_2 N_2 = N.$$ 

You can always find integers $k_1$ and $k_2$ to satisfy this equation, so therefore $x[n]$ is periodic.

2. Its period is simply the least common multiple of $N_1$ and $N_2$, i.e. \( LCM(N_1, N_2) \).
Energy and Power Signals

A signal $x[n]$ is said to be an energy signal if and only if

$$||x||_2^2 < \infty$$

Examples:
1. If a signal has finite-length, it is an energy signal. why?

2. Consider $x_a[n] = u[n].a^{-n}$. Is this an energy signal?
A signal of infinite energy is said to be a power signal if

$$\mathcal{P}_x = \lim_{K \to \infty} \frac{1}{2K + 1} \sum_{n=-K}^{K} |x[n]|^2 < \infty$$

Example:
Consider the second example from last page. Put $a = 1$. Is this an energy signal? How about a power signal? why?

Now consider $x[n] = \cos(n\pi/2)$. Discuss if $x$ is an energy or power signal?

Can you generalize your test to identify power signals to any arbitrary periodic signal?
A Very Common Discrete-Time Signals

\[ x[n] = Ca^n \quad C \text{ and } a \text{ can be complex} \]

1. If \( C \) and \( a \) are real, then \( x[n] \) is a real exponential.
   
   (a) \( a > 1 \)

   This is a growing exponential.
   
   An example is a GOOD investment.
   
   \( C = \)initial investment, 10\% interest rate, \( a = 1 + 0.1 = 1.1 \).

   (b) \( 0 < a < 1 \)

   BAD investment.
(c) $-1 < a < 0$

Alternates and decays (Pendulum).

(d) $a < -1$

Alternate and grows (Teenage mood swings).
2. If both $C$ and $a$ are complex, i.e. $C = Ae^{j\phi}$ and $a = e^{\Sigma_0 + j\Omega_0}$, then we get

\[
x[n] = Ca^n = Ae^{j\phi}e^{(\Sigma_0 + j\Omega_0)n} = Ae^{\Sigma_0 n}e^{j(\Omega_0 n + \phi)}
\]

\[
= Ae^{\Sigma_0 n} \cos(\Omega_0 n + \phi) + jAe^{\Sigma_0 n} \sin(\Omega_0 n + \phi)
\]

This will be a damped complex exponential, i.e. it will have sinusoidal real and imag components that either grow or decay depending on whether $\Sigma_0 < 0$ (decays) or $\Sigma_0 > 0$ (grows)

The notation above follows what we used in continuous time, e.g.

\[
x(t) = Ae^{j\phi}e^{(\sigma + j\omega)t} = Ae^{\sigma t}e^{j(\omega t + \phi)}
\]

but in discrete time it is often more convenient to put $a$ directly in polar form rather than write it as a complex exponential. In this case:

\[
x[n] = Ca^n = Ae^{j\phi}(re^{j\Omega_0})^n = Ar^n e^{j(\Omega_0 n + \phi)}
\]

\[
= Ar^n \cos(\Omega_0 n + \phi) + jAr^n \sin(\Omega_0 n + \phi)
\]

where $r$ determines whether it is decaying ($|r| < 1$) or growing ($|r| > 1$) and $\Omega_0$ determines the oscillation.
Discrete-Time Systems

Recall: A system is an operator on signals, so a discrete-time system is an operator on discrete-time signals.

We will see examples of different discrete-time systems throughout the quarter and on your MATLAB exercises. A particularly important class of LTI systems that we will work with are filtering systems, which include:

- Low-pass filters – systems that remove high frequencies in an input signal
- High-pass filters – systems that remove low frequencies in an input signal
- Band-pass filters – systems that only pass frequencies in a certain frequency band

Another example of a discrete-time system is the Euler integrator. It has the equation:

\[ y[n] = y[n-1] + Hx[n-1]. \]

Here, \( x[n] \) is the input to the system and \( y[n] \) is the output of the system. Note that when \( H = 1 \), this system reduces to an accumulator discussed in the textbook (page 68).

We can also write:

\[ y[n] = T(x[n]) \]

which represents a transformation. Given the input \( x[n] \), we solve equations to obtain the output \( y[n] \).

Another simple example of a system, described by a difference equation, is a digital filter:

\[ y[n] = (1 - \alpha)y[n-1] + \alpha x[n] \]

where \( 0 < \alpha < 1 \). (We will later see that this is a simple low-pass filter.)

See the textbook for more examples.
Properties of Discrete-Time Systems

We’ll see that these properties are very similar to those in continuous time.

Memory

Condition is same as in continuous time. \( y[n_0] = f(x[n_0]) \) alone \( \rightarrow \) system is memoryless. Otherwise, the system has memory, meaning that its output depends on inputs other than just at the time of the output.

- \( y[n] = x[n] + 5 \) is memoryless
- \( y[n] = (n + 5)x[n] \) is memoryless
- \( y[n] = x[n + 5] \) has memory

Invertibility

Formal definition: A system \( T \) has an inverse \( T_i \) if when cascaded with \( T \) gives the identity system (the output of the two systems is the original input):

\[
T_i[T(x[n])] = x[n]
\]

Unit advance and Unit delay are Inverses.

- \( T : y[n] = x[n + 1] \)
- \( T_i : x[n] = y[n - 1] \)

Accumulator and First Difference are Inverses.

- \( T : y[n] = \sum_{k=-\infty}^{n} x[k] \)
- \( T_i : x[n] = y[n] - y[n - 1] \)

A rectifier \( y[n] = |x[n]| \) is not invertible.

For simple systems, we can easily find the inverse and thereby show invertibility, or we can find two inputs that give the same output and thereby show that the system is not invertible. For more complex systems, the transforms that we will learn later will be useful for determining if a system is invertible.
Causality

Formal definition: A system is causal if output $y[n]$ at $n = n_0$ depends only on $x[n]$ for $n \leq n_0$. The output DOES NOT depend on future inputs but only on past and present inputs. Test this by looking at the time inside the $x[\cdot]$ relative to the time inside $y[\cdot]$.

Intuition: A causal system does not laugh before it is tickled. The output does not start before the input. (Note: having the output be non-zero does not always mean that the output has “started” – consider $y(t) = 1 + x(t)$.)

All real-time physical systems are causal. BUT, you can have a noncausal system – processing images (and other signals) on a computer for later viewing (or playing).

Memoryless implies causal but not vice versa.

Examples:
Passive and Lossless Systems

A system is called *passive*, if the energy of its output is no more than the energy of its input, i.e.

\[
\sum_{n=-\infty}^{\infty} |y[n]|^2 \leq \sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty
\]

If the equality is attained, the system is called *lossless*.
Bounded-Input Bounded-Output (BIBO) Stability

Formal definition: A system is BIBO stable if an input \(|x[n]| \leq B_x, \forall n\) produces an output \(|y[n]| \leq B_y, \forall n\).

Intuition: Reasonable (well-behaved) inputs do not cause the system to blow up.

Examples:

Unit delay \(y[n] = x[n − 1]\) is stable

\(y[n] = \cos(x[n])\) is stable

Accumulator is not stable

\[ y[n] = \sum_{k=-\infty}^{n} x[k] \]

We will see later that this system has an impulse response of \(h[n] = u[n]\) and we will see this is not BIBO stable.
Time-Invariance (textbook refers to this as Shift-Invariance)

Formal definition: A system is time-invariant if a time shift in the input only results in the same time shift in the output. Mathematically, we can write this as:

\[
\begin{align*}
T[x[n]] &= y[n] \\
T[x[n - n_0]] &= y[n - n_0]
\end{align*}
\]

Intuition: A system is time-invariant if its behavior doesn’t change with time.

Formal test for time-invariance:
Examples:

1. $y[n] = x[2n]$

2. $y[n] = \sum_{k=-\infty}^{n} x[k]$

3. $y[n] = \sum_{k=0}^{n} x[k]$

4. $y[n] = nx[n]$

5. $y[n] = x[n]u[n]$
Linearity

Formal definition: A system is linear if both additivity and scaling hold:

\[ T[x_1[n]] = y_1[n] \text{ and } T[x_2[n]] = y_2[n] \Rightarrow \]
\[ T[ax_1[n] + bx_2[n]] = ay_1[n] + by_2[n] \]

Special case test for non-linear systems: Zero input must produce zero output due to scaling property:

\[ T[ax_1[n]] = ay_1[n] \]

Let \( a = 0 \), then \( T[0] = 0 \).

General (formal) test for linearity:
Examples:
Time Domain Characterization of Discrete-Time LTI Systems

We will study discrete-time systems that are both linear and time-invariant and see that their input/output relationship is described by a discrete-time convolution.

Impulse Representation of Discrete-Time Signals

We can write a signal $x[n]$ as:

$$x[n] = \ldots + x[-1] \delta[n + 1] + x[0] \delta[n] + x[1] \delta[n - 1] + x[2] \delta[n - 2] + \ldots$$

or

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

which is writing $x[n]$ as a series of impulse functions shifted in time, all scaled with weights $x[k]$. We will see this again when we show that the I/O relationship of a DT LTI system is a DT convolution.
Convolution for Discrete-Time Systems

Using the result from 10.1:

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

and the fact that the system is linear plus knowledge that the response to \( \delta[n - k] \) is \( h_k[n] \)

\[ x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k] \]
\[ y[n] = \sum_{k=-\infty}^{\infty} x[k] h_k[n] \]

Due to Time-Invariance, we get \( h_k[n] = h[n - k] \)
\[ y[n] = \sum_{k=-\infty}^{\infty} x[k] h_k[n] \]
\[ = \sum_{k=-\infty}^{\infty} x[k] h[n - k] \]

CONVOLUTION!!
Note that \( h[n] \) is the impulse response.
We get the Convolution Equation:
\[ y[n] = x[n] \ast h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k] \]

The output of an LTI system is the input convolved with the impulse response where \( h[n] \) is the impulse response.

Convolution is useful in image processing— a KERNEL is passed over each pixel of the image to effect a desired image processing operation such as filtering, edge detection, etc.

As for CT, DT convolution is commutative:
Let \( m = n - k \) in above equation,
\[ \sum_{n-m=-\infty}^{\infty} x[n - m] h[m] \Rightarrow \sum_{-m=-\infty}^{\infty} h[m] x[n - m] \Rightarrow \]
\[ \sum_{m=-\infty}^{\infty} h[m] x[n - m] = h[n] \ast x[n] \]

It doesn’t matter which signal gets flipped.
Steps to perform convolution:

1. Time reverse $h[k]$ and shift by $n$ to form $h[n - k]$ (flip and shift)
2. Rewrite $x[n]$ as $x[k]
3. Multiply $x[k]$ and $h[n - k]$ for all values of $k$
4. Sum up $x[k]h[n - k]$ over all $k$ to get $y[n]$
5. Do for all values of $n$

I HAVE TWO IMPORTANT RULES FOR PERFORMING A CONVOLUTION:

1. FLIP THE EASY FUNCTION!
2. DRAW A PICTURE!

Ex.

Find $x[n] * h[n] = y[n]$. 
Note: $N_y = N_x + N_h - 1$, where $N_i$ is the nonzero length of $i[n]$. 

43
Ex. Find $x[n] * \delta[n - n_0]$ ⇒ This is CONVOLUTION WITH DISCRETE-TIME IMPULSE ⇒ Result is: Convolving a function with an impulse shifts the function to where the impulse is

What is $x[n] * \delta[n]$?
Ex. Find \( y[n] = x[n] * h[n] \) where \( x[n] = a^n u[n] \) and \( h[n] = u[n] \). Try it both ways (first flip \( x[n] \) and do the convolution and then flip \( h[n] \) and do the convolution). Which method do you prefer?
Ex. Find $y[n] = x[n] * h[n]$ where:
More on DT convolution

\( x[n] = h[n] = u[n] \). Find \( y[n] = x[n] * h[n] \).

1. Do it Graphically:

2. Use convolution equation

\[
y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} u[k]u[n-k]
\]

\( \Rightarrow \sum_{k=0}^{\infty} u[n-k] \) since \( u[k] = 0, \ k < 0 \)

Now,

\( u[n-k] = 0, \ n-k < 0 \) or \( k > n \) \( \Rightarrow \) \( y[n] = \sum_{k=0}^{n} (1) = n + 1 \)

BUT what values of \( n \) is this good for?

\( u[k] = 0, \ k < 0 \) and \( u[n-k] = 0, \ k > n \)

\( \Rightarrow \) only good for \( 0 < k \leq n \) \( \Rightarrow n \geq 0 \).
Ex.

$$x[n] = b^n u[n]$$
$$h[n] = a^n u[n + 2]$$

where $a \neq b$

Find $y[n] = x[n] * h[n]$. 
Compute output of $x[n] = u[-n]$ to system with impulse response -

$$h[n] = a^n u[n - 2], \ |a| < 1$$

Answer is:

$$\frac{a^2}{1 - a} u[2 - n] + \frac{a^n}{1 - a} u[n - 3]$$
Ex. Given $x[n] = u[n]$ and $h[n] = a^n u[n + 2]$, find

$$y[n] = x[n] \ast h[n]$$
Ex. Given \( x[n] = u[-n + 2] \) and \( h[n] = a^n u[-n] \), find

\[ y[n] = x[n] * h[n] \]

The output should be left-sided.
Here are some examples of discrete-time impulse responses:

Unit delay: \( h[n] = \delta[n - 1] \)
Unit advance: \( h[n] = \delta[n + 1] \)
Accumulator: \( h[n] = u[n] \)
Edge detector: \( h[n] = \delta[n] - \delta[n - 1] \)
FIR and IIR LTI systems

If $h[n]$ is finite-length, the system is called *Finite Impulse Response* (FIR) system. For an FIR, the convolution equation becomes:

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

Is this a FIR system?

---

**Ex.** The output-input relationship for a *moving average* (MA) filter is

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

Is this a FIR system?

---

If $h[n]$ is not finite, the system is called an *Infinite Impulse Response* (IIR) filter. The simplest IIR system is an extension of an accumulator and is called an *autoregressive* (AR) model:

$$y[n] = x[n] - \sum_{k=1}^{N} d_k y[n - k].$$

Can you guess the output-input difference equation for an ARMA model?
Step Response of a Discrete-Time System

The step response of an LTI system is just the response of the system to an input equal to unit step. We can denote this as \( s[n] \).

Ex. Compute DT step response of an LTI system with \( h[n] = a^n u[n-2] \).

55
Properties of Discrete-Time LTI Systems

The I/O characteristics of an LTI system are completely characterized by its impulse response \( h[n] \) (and the output is just the convolution of the input with \( h[n] \)). We can derive properties of LTI systems based on this by putting constraints on \( h[n] \). Notice that the basic definitions remain the same, the result here are specific quick tests for LTIs.

**Memoryless Systems**

The impulse response of a memoryless LTI system can only have the form

\[
h[n] = K\delta[n].
\]

Anything else would cause inputs other than at the present time to appear in the output.

**Invertible Systems**

An LTI system with impulse response \( h[n] \) is invertible if there exists another function \( h_i[n] \) such that

\[
h[n] \ast h_i[n] = \delta[n]
\]

**Ex.** What is the inverse of \( h[n] = 3\delta[n + 5] \)?

56
Causality

\[ h[n] = 0, \ n < 0 \]

so that

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

depends only on past and present values of input.

Ex. We can also see the requirements for an LTI system to be causal through convolution. Let \( h_1[n] = u[n] \) and form \( h_1[n-k] \). What inputs appear in the output?

Now let \( h_2[n] = u[n+2] \) and form \( h_2[n-k] \). How does this relate to causality?
Stability

Requirement for Bounded-input/Bounded-output.

Stability is

\[ \sum_{k=-\infty}^{\infty} |h[k]| < \infty, \]

That is, the impulse response must be absolutely summable for a system to be BIBO stable.

This is because:

Given \(|x[n]| \leq M\) for all \(n\),

examine the magnitude of the output. It must be finite for BIBO stability.

\[ |y[n]| = \left| \sum_{k=-\infty}^{\infty} x[n-k]h[k] \right| \leq \sum_{k=-\infty}^{\infty} |x[n-k]| |h[k]| = \sum_{k=-\infty}^{\infty} |x[n-k]| |h[k]| \leq \sum_{k=-\infty}^{\infty} M|h[k]| = M \sum_{k=-\infty}^{\infty} |h[k]| \]

Since \(M < \infty\), we only need \(\sum_{k=-\infty}^{\infty} |h[k]| < \infty\) for \(y[n]\) to be finite.

**Ex.** Is

\[ h[n] = (\frac{1}{3})^u[n] \]

BIBO stable?
More examples:

1. $h_1[n] = u[n]$ (the accumulator)

2. $h_2[n] = 3^n u[n]$

3. $h_3[n] = (3)^n u[-n]$

4. $h_4[n] = \cos(\frac{\pi}{3} n) u[n]$

5. $h_5[n] = u[n + 2] - u[n]$
Unit-Step Response

\[ x[n] = u[n] \]

\[ s[n] = \sum_{k=-\infty}^{\infty} h[k]u[n-k] = \sum_{k=-\infty}^{n} h[k] \]

Can get \( h[n] \) from \( s[n] \) as:

\[ h[n] = s[n] - s[n-1] \]

**Ex.** Given we already determined that for an impulse response \( h[n] = a^n u[n] \), \( s[n] = u[n] * a^n u[n] = \frac{1-a^{n+1}}{1-a} u[n] \), show that you can obtain the impulse response back from the step response. You might find it helpful to remember that \( u[n-1] = u[n] - \delta[n] \).
Summary of DT LTI Systems.

1. System Attributes,
   Memory, linearity, TI, Causality, Stability, Invertibility.
   Saw how to determine attribute from impulse response (except inverse).

2. Saw how an LTI system has its I/O relationship described by convolution.

3. Superposition—Break an input down into basis functions for which it is easy to calculate system response.
   Ex. Impulses, step functions, exponentials.


\[
\begin{align*}
\text{Continuous Time} & & \text{Discrete Time} \\
\delta(t) = \frac{d}{dt} u(t) & & \delta[n] = u[n] - u[n - 1] \\
h(t) = \frac{d}{dt} s(t) & & h[n] = s[n] - s[n - 1] \\
\int_{-\infty}^{t} \delta(\tau) d\tau & & \sum_{k=\infty}^{n} \delta[k] \\
\int_{-\infty}^{t} h(\tau) d\tau = h(t) * u(t) & & \sum_{k=\infty}^{n} h[k] = h[n] * u[n] \\
u(t) = \int_{-\infty}^{t} \delta(\tau) d\tau & & u[n] = \sum_{k=\infty}^{n} \delta[k] \\
s(t) = \int_{-\infty}^{t} h(\tau) d\tau = h(t) * u(t) & & s[n] = \sum_{k=\infty}^{n} h[k] = h[n] * u[n]
\end{align*}
\]
Difference-Equation Models (sections 4.6.1-4.6.3 of the textbook)

LTI discrete-time systems are usually modeled by linear difference equations with constant coefficients. For example, a digital filter is modeled by a difference equation.

An example of a difference equation is:

\[ y[n] = x[n] + x[n - 1] + x[n - 2]. \]

A general \( N \)th order \((N \geq M)\) linear difference equation with constant coefficients (LCCDE) is:

\[
d_0y[n] + d_1y[n - 1] + \cdots + d_{N-1}y[n - N + 1] + d_Ny[n - N] = \\
p_0x[n] + p_1x[n - 1] + \cdots + p_{M-1}x[n - M + 1] + p_Mx[n - M]
\]

which we can write as:

\[
\sum_{k=0}^{N} d_ky[n - k] = \sum_{k=0}^{M} p_kx[n - k]
\]

where \( a_k \) and \( b_k \) are real constants.

An important case to be familiar with is the first-order system

\[ y[n] = ay[n - 1] + bx[n] \]

in which the output is a function of a delay of only one time unit.

The Classical method for the solution is to express the output \( y[n] \) as the sum of complementary or natural \((y_c[n])\) and particular or forced \((y_p[n])\) solutions:

\[ y[n] = y_c[n] + y_p[n] \]
**Natural response**  The natural response is the solution to the homogeneous equation:

\[ \sum_{k=0}^{N} d_k y[n - k] = 0 \]

where \( d_0 \neq 0 \).

We assume solutions of the form \( y_c[n] = \alpha \lambda^n \).

We can see that:

\[
\begin{align*}
y_c[n] &= \alpha \lambda^n, \
y_c[n - 1] &= \alpha \lambda^{n-1} = \alpha \lambda^{-1} \lambda^n, \ldots \
y_c[n - N] &= \alpha \lambda^{n-N} = \alpha \lambda^{-N} \lambda^n
\end{align*}
\]

and substituting in the homogeneous equation yields:

\[
(d_0 \lambda^N + \cdots + d_{N-1} \lambda + d_N) \alpha \lambda^{-N} \lambda^n = 0
\]

and we get the characteristic equation:

\[
d_0 \lambda^N + d_1 \lambda^{N-1} + \cdots + d_{N-1} \lambda + a_N = a_0 (\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_N) = 0.
\]

Clearly, \( N \) values of \( \lambda \) satisfy this equation.

The solution is of the form:

\[
y_c[n] = \alpha_1 \lambda_1^n + \alpha_2 \lambda_2^n + \cdots + \alpha_N \lambda_N^n
\]

assuming there are no repeated roots (which is all we will cover).
Ex. Given a first-order difference equation

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

find its homogeneous solution. Your answer should be in terms of a constant $\alpha$.

\[
\begin{align*}
\alpha \lambda^n + .2\alpha \lambda^{n-1} &= 0 \\
\alpha \lambda^{n-1}(\lambda + .2) &= 0 \\
\lambda + .2 &= 0
\end{align*}
\]

$\lambda = -2$

\[
\begin{align*}
y_c[n] &= \alpha (-2)^n \\
y_c[0] &= 1 \\
y_c[-1] &= 0
\end{align*}
\]

$\implies y[n] = (-2)^n$ for $n \geq 0$
**Forced response** The forced response $y_p[n]$ solves the equation

$$\sum_{k=0}^{N} d_k y_p[n - k] = \sum_{k=0}^{M} p_k x[n - k].$$

The form of the solution is determined by the input $x[n]$. For an exponential input $x[n] = Aa^n$, the solution would be $y_p[n] = Pa^n$ where $A, a,$ and $P$ are constants.

**Ex.** For the previous example, given an input $x[n] = 9(\cdot 7)^n$, find the particular solution $y_p[n]$.

\[
\begin{align*}
y[n] + 0.2 y[n-1] &= 9 (\cdot 7)^n \\
\text{Guess} \quad y[n] &= P (\cdot 7)^n \\
\text{Solve} \quad P (\cdot 7)^n + 2 P (\cdot 7)^{n-1} &= 9 (\cdot 7)^n \\
\alpha (\cdot 7) + 2 P &= 9 \cdot 7 \\
\Rightarrow P &= 7
\end{align*}
\]

**Ex.** Now, assuming that the system is initially at rest, i.e., initial conditions of 0 ($y[0] = 0$), solve for the constant $\alpha$ in your overall solution $y[n] = y_c[n] + y_p[n]$.

\[
\begin{align*}
y[0] + 0.2 y[0-1] &= \alpha (\cdot 7)^0 = 9 (\cdot 7)^0 \\
y[0] &= 0 \\
\Rightarrow \alpha (\cdot 7) + 7 (\cdot 7)^0 &= 0 \\
\alpha (\cdot 2) + 7 &= 0 \\
\Rightarrow \alpha &= -\frac{7}{2}
\end{align*}
\]
Ex. Given

\[ y[n] - .3y[n - 1] = x[n] \]

with \( y[-1] = 0 \) and \( x[n] = (.6)^n \), find \( y[n] \).

\[
y(n) = \alpha (-.2)^n + 7 (.7)^n
\]

\[
\alpha (-.2)^0 + 7 (.7)^0 = 0 \Rightarrow \alpha = -7
\]
Terms in the Natural Response

Recall that the natural solution was

\[ y_c[n] = \alpha_1 \lambda_1^n + \alpha_2 \lambda_2^n + \cdots + \alpha_N \lambda_N^n \]

where \( \lambda_i \) is the root of the characteristic equation.

\[ a_0 \lambda^N + a_1 \lambda^{N-1} + \cdots + a_{N-1} \lambda + a_N = a_0(\lambda - \lambda_1)(\lambda - \lambda_2)\ldots(\lambda - \lambda_N) = 0. \]

Each general term is \( \alpha_i \lambda_i \) where \( \lambda_i^n \) is a system mode. The root \( \lambda_i \) can be either real or complex. Its value will determine if the overall system is BIBO stable or not.

Assume we have a causal LTI system. The solution is of the form

\[ y[n] = y_c[n] + y_p[n]. \]

Since \( y_p[n] \) is of the form \( x[n] \), if the input \( x[n] \) is bounded, then \( y_p[n] \) will also be bounded.

Let’s examine

\[ y_c[n] = \alpha_1 \lambda_1^n + \alpha_2 \lambda_2^n + \cdots + \alpha_N \lambda_N^n. \]

Clearly, as long as all roots (also called poles) of this equation, \(|\lambda_i^n| < 1\), then each term in \( y_c[n] \) will be bounded.

Our condition for stability of a causal LTI system is that all roots of the system characteristic equation lie within the unit circle in the \( \lambda \)-plane, that is, \(|\lambda_i| < 1, \forall i\).
Given a causal system described by the difference equation

\[ y[n] - 2.5y[n - 1] + y[n - 2] = x[n] \]

determine if the system is BIBO stable.

Characteristic eqn: \( \lambda^2 - 2.5\lambda + 1 = 0 \)

\[ \lambda = \frac{2.5 \pm \sqrt{(2.5)^2 - 4}}{2} \]

\[ \lambda = \frac{2.5 \pm \sqrt{6.25 - 4}}{2} \]

\[ \lambda = \frac{2.5 \pm \sqrt{2.25}}{2} \]

\[ \lambda = \frac{2.5 \pm 1.5}{2} \]

\[ \lambda_1 = 2, \quad \lambda_2 = 0.5 \]

\[ y_c[n] = a_1(2)^n + a_2(0.5)^n \]

Causal system

\[ y_c[0], \quad y_c[-1] \text{ would be given} \]

\[ y_c[n] = a_1(2)^n + a_2(0.5)^n \quad n \geq 0 \]

Not BIBO Stable
Ex. Given a causal system described by the difference equation

\[ y[n] - 1.25y[n-1] + 0.375y[n-2] = x[n] \]

determine if the system is BIBO stable.

Do it at home!
System Response for Complex Exponential Inputs

Given an input $x[n] = Xz^n$ to a BIBO stable LTI system modeled by an $N$th order linear difference equation with constant coefficients, we will examine the steady-state system response. Here, $X$ and $z$ are complex.

If the system is stable, then the natural system response will die out. The forced (steady-state) response of the system to this input is of the same form as the input, i.e.

$$y_p[n] = y_{ss}[n] = Yz^n.$$

From the difference equation describing the system,

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k],$$

plugging in $x[n]$ and $y_{ss}[n]$, we get:

$$\sum_{k=0}^{N} a_k Yz^{n-k} = \sum_{k=0}^{M} b_k Xz^{n-k}$$

or

$$Y \sum_{k=0}^{N} a_k z^{n-k} = X \sum_{k=0}^{M} b_k z^{n-k}$$

or

$$Y = X \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$

which we can write as

$$Y = X H(z)$$

where

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$

is a transfer function.

So given an input $x[n] = Xz_1^n$ to such an LTI system, the steady-state response is $y_{ss}[n] = X H(z_1) z_1^n$. 

70
Similar to the difference equation, the transfer function $H(z)$ completely characterizes the LTI system (we can derive the difference equation from $H(z)$ and vice versa).

In general, by superposition, given an input $x[n] = \sum_{k=1}^{M} X_k z_k^n$, the output of the system is $y_{ss}[n] = \sum_{k=1}^{M} X_k H(z_k) z_k^n$.

Finally, if $x[n] = \sum_{k} a_k \phi_k[n]$ and
\[
y[n] = \sum_{k} a_k \psi_k[n]
\]
where $\psi_k[n] = \phi_k[n] * h[n]$ AND $\psi_k[n] = b_k \phi_k[n]$ (input and output basis functions have the same form), then $\phi_k[n]$ is an eigenfunction of the LTI system with eigenvalue $b_k$.

Analogous to CT, eigenfunctions of DT LTI systems are complex exponentials:
\[
\phi[n] = z^n
\]

Check: What is $z^n * h[n]$?

\[
\psi[n] = \phi[n] * h[n] = z^n * h[n] = \sum_{k=-\infty}^{\infty} z^{n-k} h[k] = z^n \sum_{k=-\infty}^{\infty} h[k] z^{-k} = z^n H(z)
\]

where $H(z) = \sum_{k=-\infty}^{\infty} h[k] z^{-k}$ is the eigenvalue. This motivates the Z-transform and the Discrete Time Fourier Transform (DTFT):

- $H(z)$ is known as the $z$-transform of $h[n]$.
- If $z = e^{j\Omega}$, then we get $H(\Omega)$, the DTFT of $h[n]$.  

71
Ex.
Given an input \( x[n] = (\frac{2}{3})^n \), and

\[
\begin{align*}
h[n] &= \delta[n] \\
y[n] &= 3 \cdot 2^n
\end{align*}
\]

find its steady-state output is the forced response

YOU FINISH:

\[
\begin{align*}
y[c][n] &= \alpha_1(\lambda_1)^n + \alpha_2(\lambda_2)^n \\
y_p[n] &\quad ? \\
y[n] &= y[c][n] + y_p[n] \\
y[n] &= y[c][n] & n > 0
\end{align*}
\]

\[
\begin{bmatrix}
y[0] \\
y[-1]
\end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y[0] \\
y[-1]
\end{bmatrix}
\]

\[
\begin{align*}
y[c][n] &= \alpha_1(0.2)^n \\
y_p[n] &= \left(\frac{10}{3}\right)(2)^n \\
y_p[n] &= P(2)^{n-1} \\
y[n] &= \delta[n] + (2^{n-1} + 2^n)u[n-1] \\
y[n] &= (\delta[n] + 3(2^{n-1})u[n-1]) \\
y[0] &= 0 \\
y[1] &= 1 \\
y[2] &= 2 \\
y[3] &= 4 \\
y[4] &= 8
\end{align*}
\]