Linear, Time Invariant and Causal Systems

Contents

1 Describing Digital Signals

Digital signals form an integral part of a Digital Signal Processing (DSP) system. For example, understanding various digital signals can help understand the operation of a DSP system. It can be very useful to start with some basic signal types and then look at how particular types of DSP systems respond if those signal types are the inputs to the system.

1.1 Unit Impulse Function

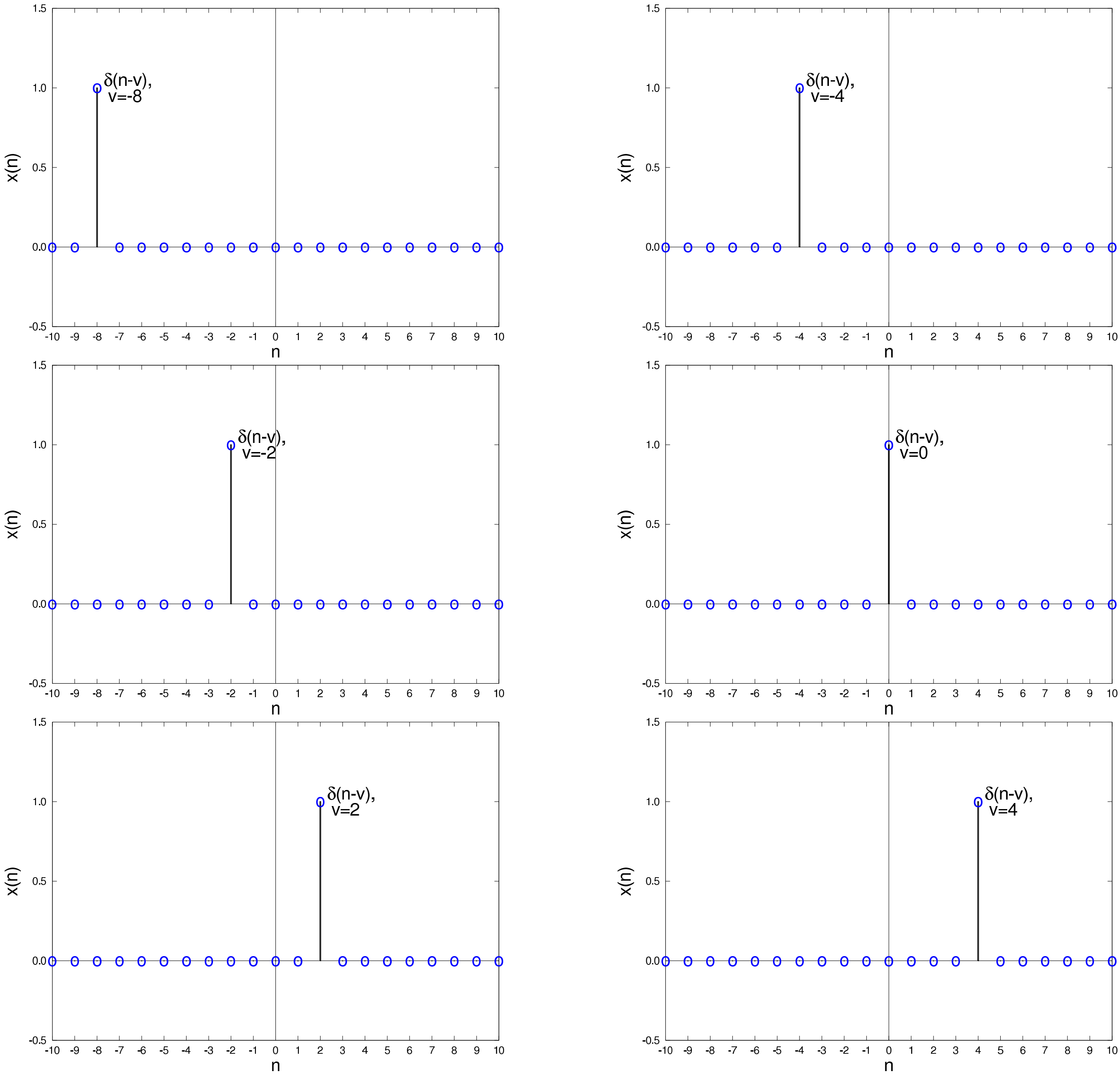
The Unit impulse function is a fundamental function in Digital Signal Processing (DSP). It can be used to characterize systems whose response or output to an unit impulse input is known as the unit impulse response. The symbol of the unit impulse function is the *Greek* delta: δ which has the following property:

- $\delta(n) = 1$ if $n = 0$

  The unit impulse can also be shifted so that,

$$\delta(n-v) = \begin{cases} 1 & \text{if} \quad (n-v) = 0, \\ 0 & \quad\;\; \text{otherwise.} \end{cases}$$

where v shifts the unit impulse to the left or the right of $n = 0$.
Some examples of impulse functions that have been shifted can be seen here:



1.2 Scaling Unit Impulse Function

It is also possible to *scale* the unit impulse function. This is useful when considering what might happen if a digital signal, consisting of many scaled and shifted unit impulses passes through a DSP system. Scaling can be performed by a single variable or by a function that varies depending on the position (shift position) of the unit impulse.

Scaling a unit impulse with any value g results in:

$$g \times \delta(n-v) = \begin{cases} g & \text{if} \quad n-v=0, \\ 0 & \text{otherwise.} \end{cases}$$

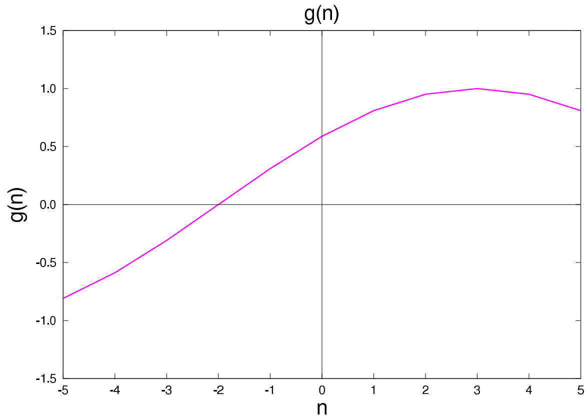
If, on the other hand g is instead a function, such as $g(n)$ then

$$g(n)\delta(n-v) = \begin{cases} g(n) & \text{if} \quad n-v=0, \\ 0 & \text{otherwise.} \end{cases}$$

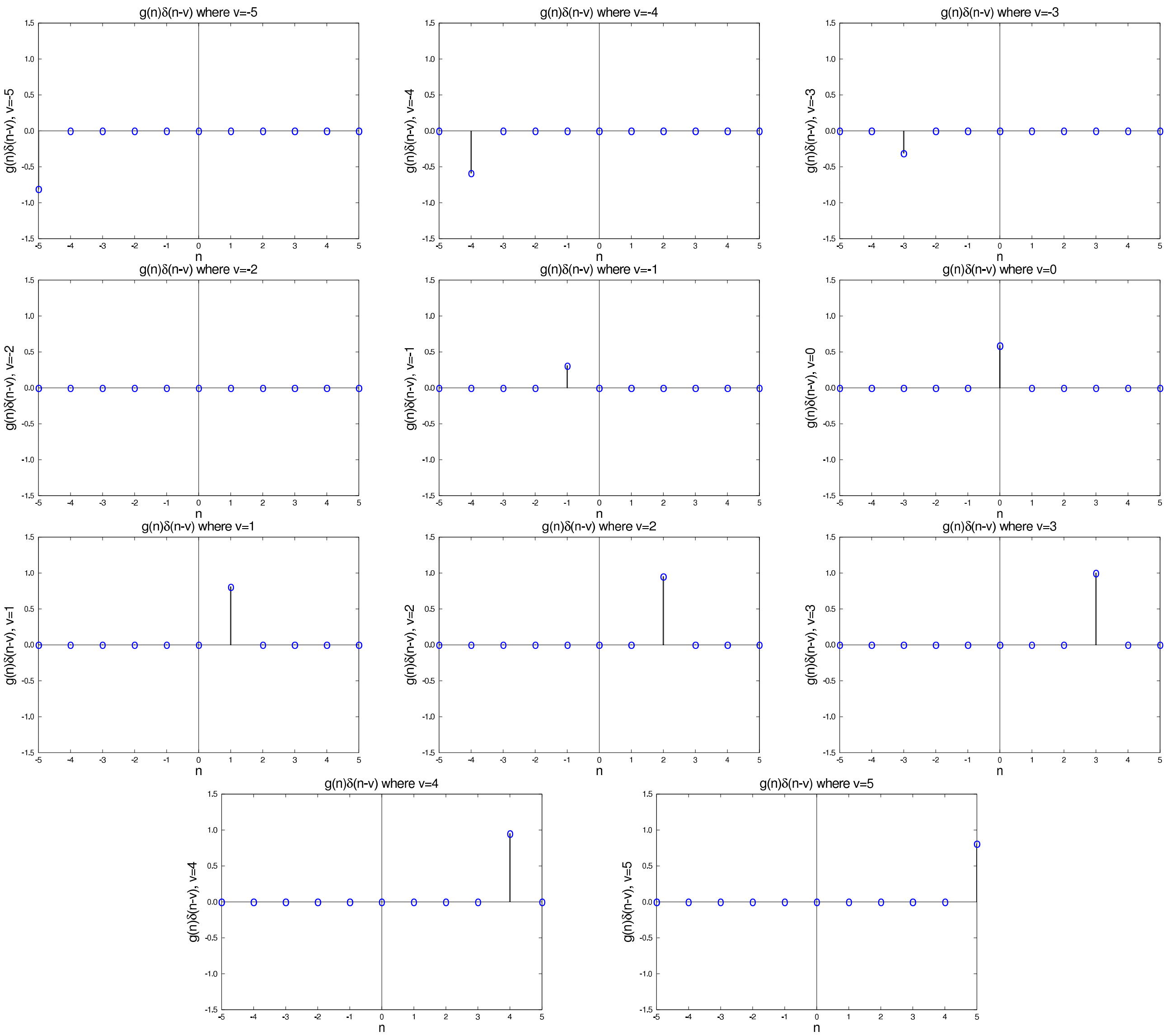
Scaling a shifted unit impulse function with another function is useful for something called *sifting*.

1.3 Sifting

For example, given a signal $g(n)$:

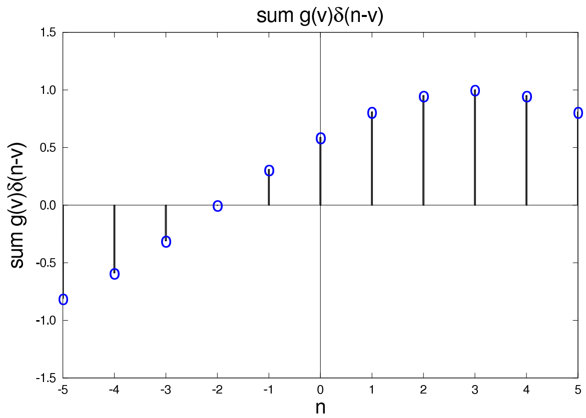


- Calculate $g(n)\delta(n-v)$ for all values of v, *i.e.*



Each of these individually scaled and shifted impulse functions can now be added (summed) altogether to get the sifted version of $g(n)$.

Adding all the delta values together we get



The result of this sifting operation is a discrete (*sifted*) representation of the original signal, $g(n)$. We can now define a discrete or digital version $x[n]$ of the original signal $g(n)$ like so

$$x[n] = ... + g(-5)\delta(n+5) + g(-4)\delta(n+4) + ... + g(4)\delta(n-4) + g(5)\delta(n-5) + ...$$

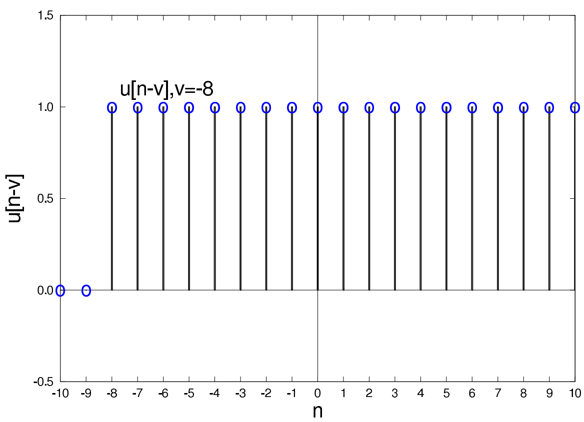
where $[\cdot]$ signifies a discrete formulation. This can be shortened to $x[n] = \sum_{k=-\infty}^{\infty} g(k)\delta(n-k)$. For our case $x[n] = \sum_{k=-5}^{5} g(k)\delta(n-k)$. This is a somewhat mathematically simplified version of the process of converting an analogue signal into digital form. Notice that the index n for the "*analogue*" signal is discrete, taking only integer values, i.e. $... -5, -4, -3, ..., 1, 2, ...$. Most analogue signals will be defined using a continuous time index, e.g. t. We will consider this aspect of sampling in more detail in a later part of the course.

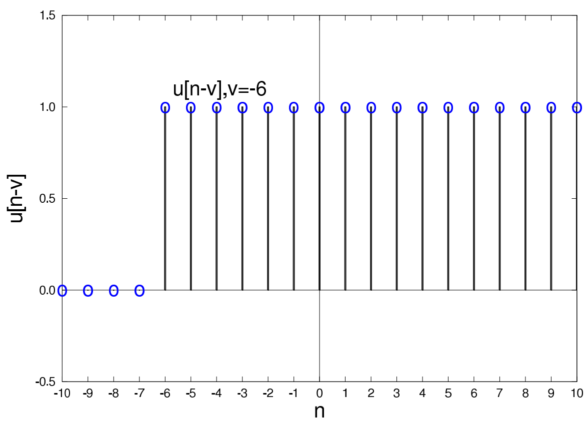
1.4 Unit Step Function

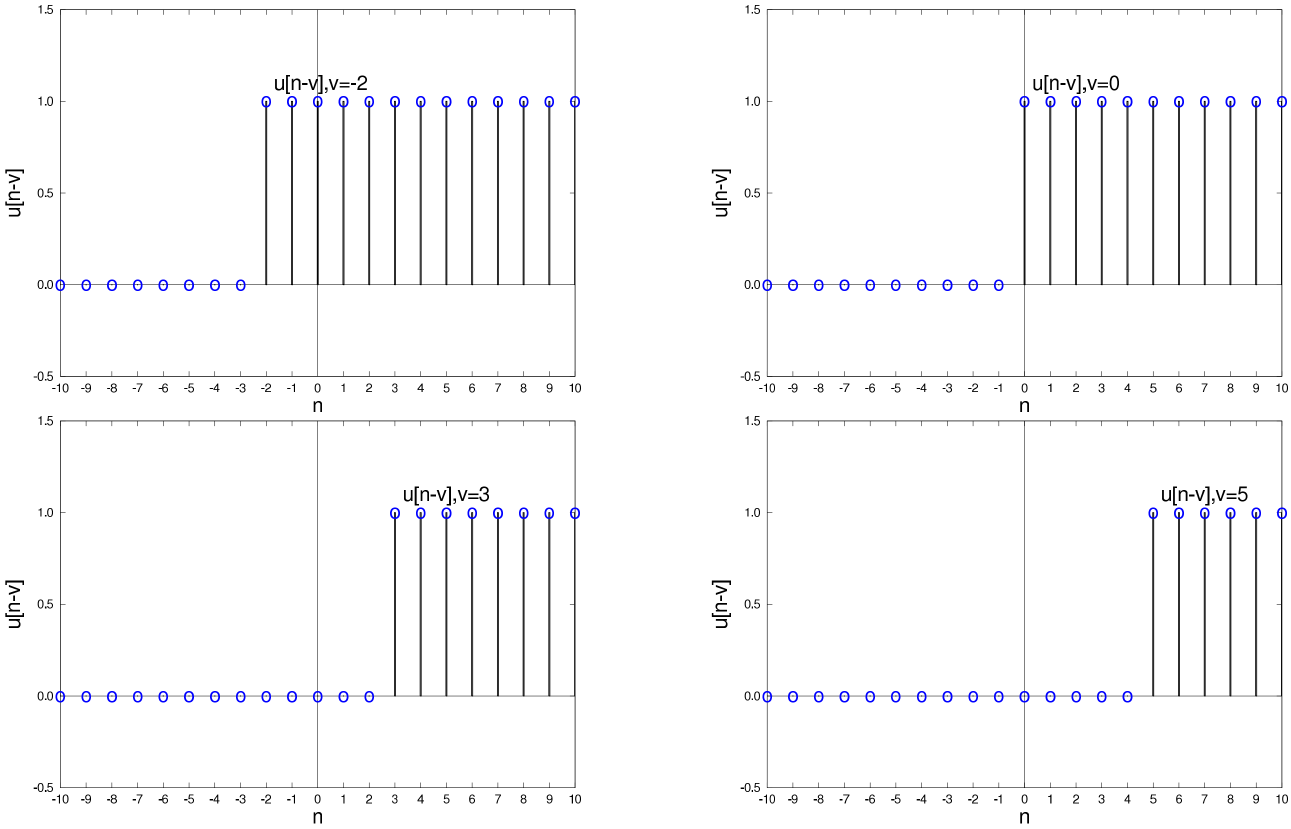
The unit step function is another common type of signal. It is used in many DSP and control engineering applications. The unit step function can be defined like so:

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

The unit step function is bit like an on/ off switch, switching from zero to an on value (1). Examples of shifted unit step functions are illustrated below.







The digital unit step function can also be defined using a series of unit impulse functions ($\delta[n-v]$) like so:

$$u[n-v] = \sum_{m=v}^{\infty} \delta[n-m]$$

Using this, we may also turn it around and define the unit impulse function in terms of unit step functions!

$$\delta[n-v] = u[n-v] - u[n-1-v].$$

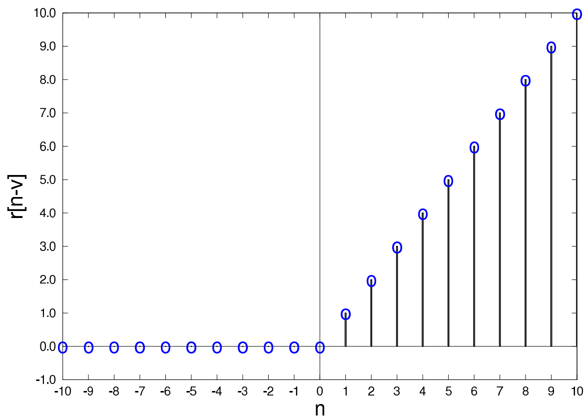
This is known as a *recurrence* formula. Recurrence formula are where the current signal value is dependent on past signal values. This is a technique that is extensively used in DSP because DSP systems often calculate an output based on the current input combined with past input values. This is useful for all manner of things, such as calculating approximations to derivatives, integrals, performing filtering, estimating future input signal values.

1.5 Ramp Function

Another interesting function type is the ramp function. The equation for a ramp function is given by

$$r[n-v] = (n-v)u[n-v].$$

A ramp function is illustrated below. *Example*



1.6 Digital Sinusoidal Functions

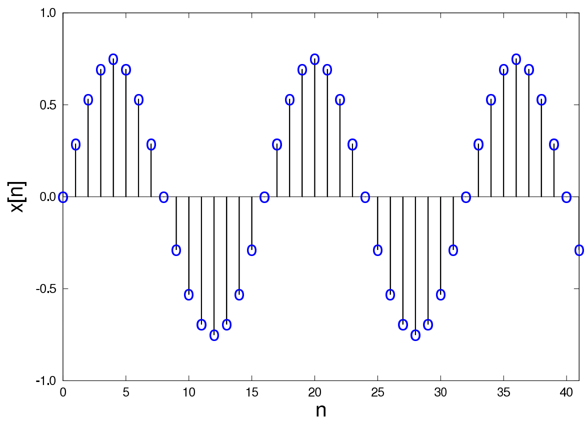
The sine wave is one of the most well known waveforms that is often associated with signal processing. The digital version of the sine wave is of the form:

$$x[n] = a\sin(n\Omega + \theta).$$

The corresponding digital cosine wave is given by:

$$x[n] = a\cos(n\Omega + \theta).$$

The sine and cosine waves, both make use of Ω which is *digital "frequency"* and it is measured in *radians per sample*. Also $\Omega = 2\pi/N$ so that $N = 2\pi/\Omega$. This digital waveform has 1 cycle across N samples.
Example
For the digital sine wave above, $a = 0.75$, $\theta = 0$ and $\Omega = \pi/8$, therefore $N = 2 \times 8 = 16$
i.e. $x[n] = 0.75\sin(n\pi/8)$:



1.7 Comparison with Analog Sine Function

The analog or continuous form of a sine wave or function is commonly encountered in conventional physics and mathematics. A continuous analog sine wave is often formulated like so:

$$x(t) = a\sin(t\omega + \theta)$$

where t could be time in seconds and $\omega = 2\pi f$ is the angular frequency, therefore in *radians per second*. The interval between each sample n is T_s seconds, so there is a sample at every $t = nT_s$ seconds. The continuous sine wave can then be written as

$$x(t) = x(n)|_{t=nT_s} = a\sin(nT_s 2\pi f + \theta)$$

If we equate the continuous and digital versions, then

$$x[n] = x(n)$$

so that

$$a\sin(n\Omega + \theta) = a\sin(nT_s 2\pi f + \theta).$$

Therefore $\Omega = T_s 2\pi f$ *or* if sampling frequency is $f_s = 1/T_s$ then $\Omega = 2\pi f/f_s$. We can also make the observation that $\Omega = 2\pi/N$. Equating the two representations for Ω we can see that:

$$2\pi\frac{f}{f_s} = 2\pi\frac{1}{N}$$

so that

$$N = f_s/f.$$

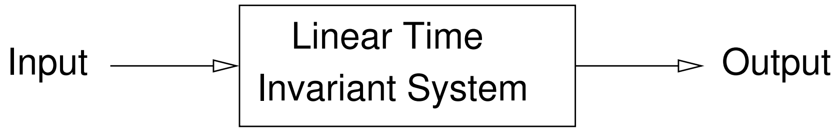
This is interesting because if we remember the Nyquist sampling theorem in which $f < f_s/2$ so that if we set $f = f_s/2$ at the upper limit then we find:

$$N = \frac{f_s}{f_s/2} = 2.$$

So if we sample at the lowest frequency possible (twice the maximum frequency) then we will get just 2 samples per cycle ($N = 2$) which is the minimum number of samples required in order to be able to reconstruct the original analogue signal.

2 Linear Time Invariant Systems

A Linear Time Invariant (LTI) System is a generalised type of system that possesses some properties that are common for many types of system. Furthermore these properties make certain calculations, including signal processing operations simpler to compute. A simplified illustration of an LTI system is shown below, emphasising that we are interested in a system with an input signal and an output that corresponds to some operation or set of calculations on the input signal.



The special properties that make LTI systems simpler to work with are:

- **Time Invariance**: The same response to the same input at any time.
  - ◇ For example, if we have two signals q_1 and q_2 that are equal but at different times $n = v_1$ and $n = v_2$ which can be represented with

    $$q_1[n - v_1] = q_2[n - v_2]$$

    with constants v_1 and v_2 representing two different time shifts (*i.e.* $v_1 \neq v_2$) then if F represents an LTI system the following must be true

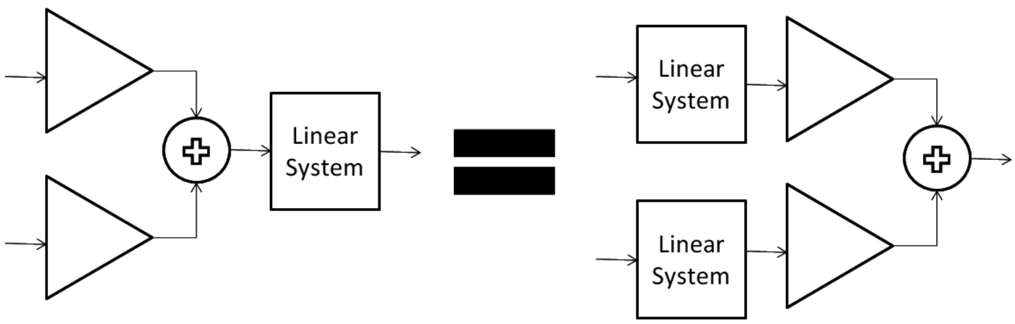
    $$F(q_1[n - v_1]) = F(q_2[n - v_2])$$

    because q_1 is the same as q_2 but at different times. An LTI system will output the same output no matter the given time, whether it is at $n = v_1$ or $n = v_2$, if the input signal is the same.
- **Linear System**:
  - ◇ *Principle of Superposition*:
    - ⋆ If the input consists of a sum of signals then the output is the sum of the responses to those signals.

If the output of a system is $y_1[n]$ and $y_2[n]$ in response to two different inputs $x_1[n]$ and $x_2[n]$ respectively then the output of the same system for the two inputs weighted and combined *i.e.* $ax_1[n]+bx_2[n]$ will be $ay_1[n]+by_2[n]$ where a and b are constants.

For a linear system $y[n] = F(x[n])$

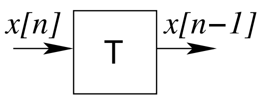
$$F(ax_1[n] + bx_2[n]) = aF(x_1[n]) + bF(x_2[n])$$



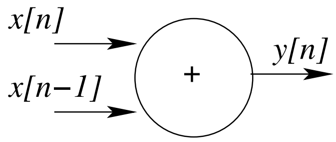
2.1 LTI System Constituent Parts

A Linear Time Invariant (LTI) system may consist of a number of different components consisting of:

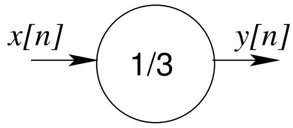
- Storage / Delay:



- Addition / Subtraction: *e.g.* $y[n] = x[n] + x[n-1]$



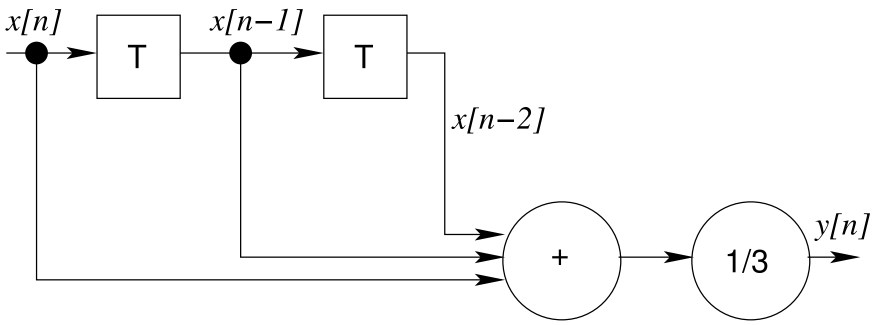
- Multiplication by Constants: *e.g.* $y[n] = \frac{1}{3}x[n]$



Simple LTI System Example

A simple example of an LTI system is a moving average filter with equation:

$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}$$



Here we can see that the input signal $x[n]$ is combined with the input that has been delayed $x[n-1]$ and $x[n-2]$. The delay is represented by a box with a T indicating a single time instance. Sometimes instead of T you may see z^{-1} which we will see later in the course. The three parts of the signal are then combined together by adding them altogether: $x[n] + x[n-1] + x[n-2]$. The result of this addition operation is then passed through another stage that multiplies the result by a third: $(x[n] + x[n-1] + x[n-2]) \times \frac{1}{3}$, thus yielding the output of the moving average filter: $y[n]$.

2.2 Other System Properties

An LTI system is

- Associative, where a system can be broken down into simpler subsystems for analysis or synthesis
- Commutative, where if a system is composed of a series of subsystems then the subsystems can be arranged in any order

LTI systems may also have

- Causality: output does not depend on future input values
- Stability: output is bounded for a bounded input (see Lecture 04)
- Invertibility: input can be uniquely found from the input (*e.g.* the square of a number is not invertible)
- Memory: output depends on past input values

An LTI Causal System is also a commonly referred to type of system with the abbreviation LTIC system.

2.3 Examples of Linear Mathematical Operations

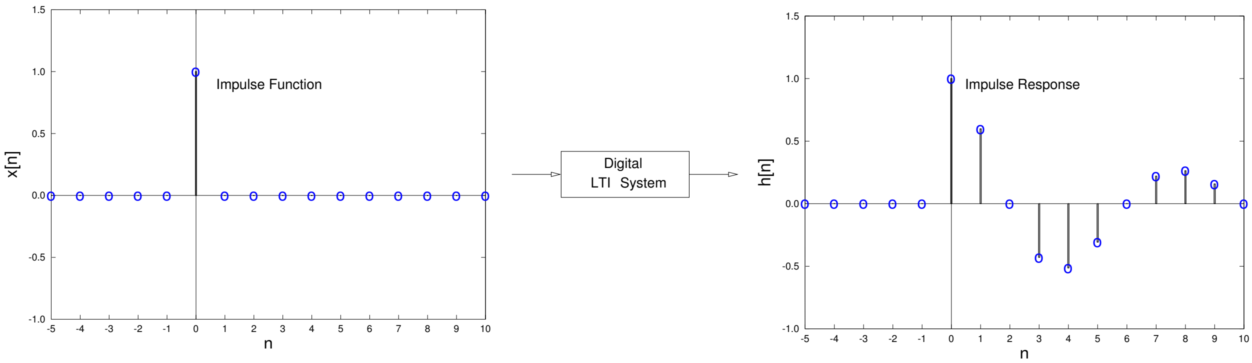
Many types of mathematical operations can be described as being LTI. These include operations involving the following

- Scaling (i.e. idealised gain or attenuation)
- Differentiation
- Integration
- The Laplace transform
- The Fourier transform
- The z-transform

Sometimes even non-LTI systems are assumed to be LTI because this assumption of LTI makes some calculations much simpler to perform.

2.4 Impulse Response

An LTI system possesses an Impulse Response which characterizes the system's output if an impulse function is applied to the input.
Example Impulse Response



The impulse response is a really useful characterisation of an LTI system because it can be used to determine the response of the LTI system to any type of signal.

Remember the moving average filter:

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

If the input is the impulse function: $x[n=0] = \delta(0)$, then $y[n]$ is the output in response to an impulse function, which can be calculated like so:

$$\begin{array}{l|rl|rl}
\ldots & & \ldots & & \ldots \\
n=-2 & x[n=-2]= & \delta[-2]=0 & y[n=-2]= & 0 \\
n=-1 & x[n=-1]= & \delta[-1]=0 & y[n=-1]= & 0 \\
n=0 & x[n=0]= & \delta[0]=1 & y[n=0]= & (\delta[0]+\delta[-1]+\delta[-2])/3 = 1/3 \\
n=+1 & x[n=+1]= & \delta[+1]=0 & y[n=+1]= & (\delta[+1]+\delta[0]+\delta[-1])/3 = 1/3 \\
n=+2 & x[n=+2]= & \delta[+2]=0 & y[n=+2]= & (\delta[+2]+\delta[+1]+\delta[0])/3 = 1/3 \\
n=+3 & x[n=+3]= & \delta[+3]=0 & y[n=+3]= & 0 \\
n=+4 & x[n=+4]= & \delta[+4]=0 & y[n=+4]= & 0 \\
\ldots & & \ldots & & \ldots
\end{array}$$

For any LTI system, such as this moving average filter, the response of the system to a unit impulse function is described as the **Impulse Response**.

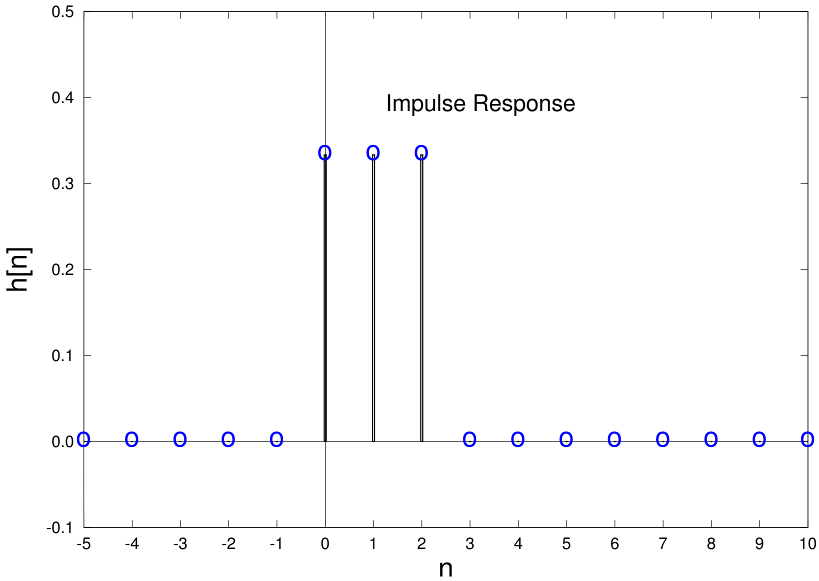
For the example shown here, the output has been calculated given that the Input Signal = Impulse Function:

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

Therefore the impulse response $h[n]$ is the same as the values of $y[n]$ that have already been calculated. This $h[n]$ is called the impulse response:

- $h[n < 0] = y[n < 0] = 0$
- $h[0] = y[0] = \frac{1}{3}(\delta[0] + \delta[-1] + \delta[-2]) = \frac{1}{3}$
- $h[1] = y[1] = \frac{1}{3}(\delta[+1] + \delta[0] + \delta[-1]) = \frac{1}{3}$
- $h[2] = y[2] = \frac{1}{3}(\delta[+2] + \delta[+1] + \delta[0]) = \frac{1}{3}$
- $h[n > 2] = y[n > 2] = 0$

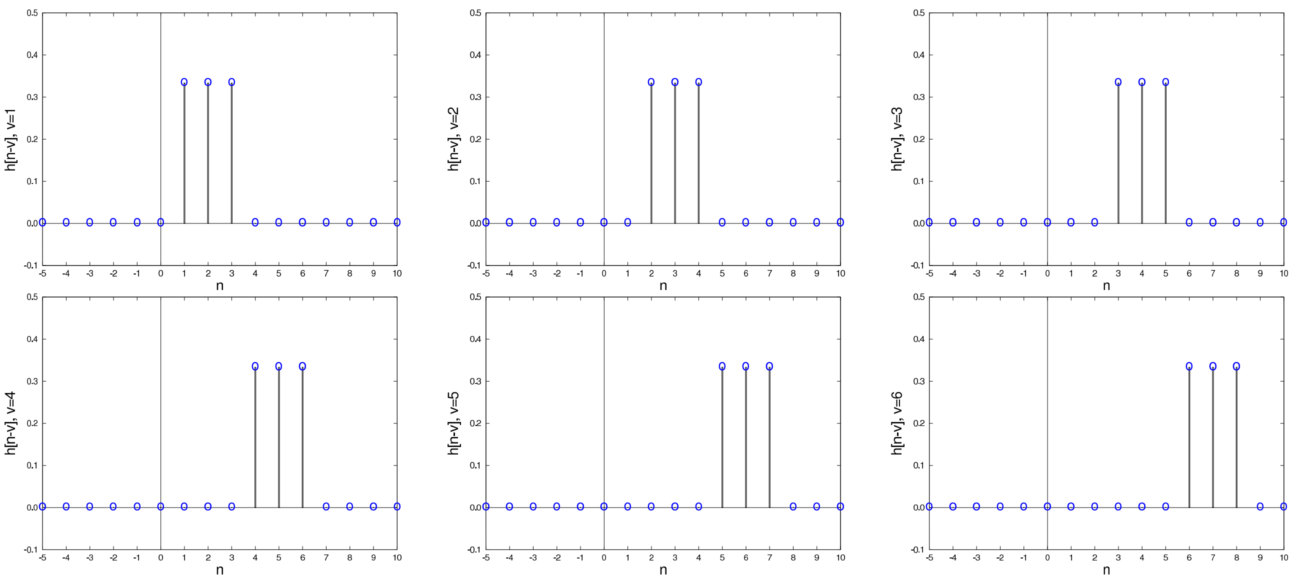
Thus, the response $h[n]$ is known as the **Impulse Response**.



This impulse response can now be used to determine the response of the system to any other signal input. To see how this is done some more concepts need to be introduced.

2.5 Impulse Response *Examples - shifting*

The impulse response can also be determined for a shifted impulse function, *i.e.* $\delta[n - v]$



What will the system output ($y[n]$) be if the input consists of more than one impulse function shifted by different amounts?

2.6 System Response to Multiple Shifted Impulse Responses

Remember that all LTI systems obey the *"Principle of Superposition"*...
So, for the inputs

$$x_1[n] = a\delta[n] \text{ and } x_2[n] = b\delta[n-1],$$

where a and b are constants, the corresponding outputs will be

$$y_1[n] = ah[n] \text{ and } y_2[n] = bh[n-1],$$

i.e. impulse responses. Therefore if $x[n] = x_1[n] + x_2[n] = a\delta[n] + b\delta[n-1]$ then

$$y[n] = ah[n] + bh[n-1].$$

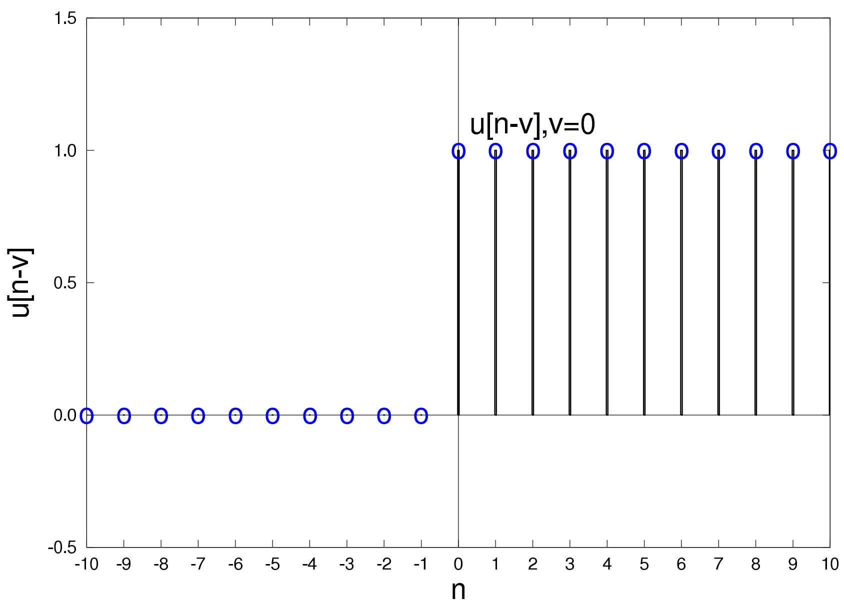
2.7 Other Signals: Step Function

- The discrete step function can be thought of as a series of impulse functions (remember sifting).
- Each impulse function creates an impulse response.
- The output is then the joint response of all the impulse responses scaled by the inputs.
- A discretely sampled step input (starting at $n = 0$) is given by:

$$x[n] = \sum_{k=0}^{\infty} \delta(n-k).$$

- Therefore, using the *Principle of Superposition* we get

$$y[n] = \sum_{k=0}^{\infty} h(n-k).$$



2.8 Step Function Moving Average

Moving average (with $k = 3$) has an impulse response:

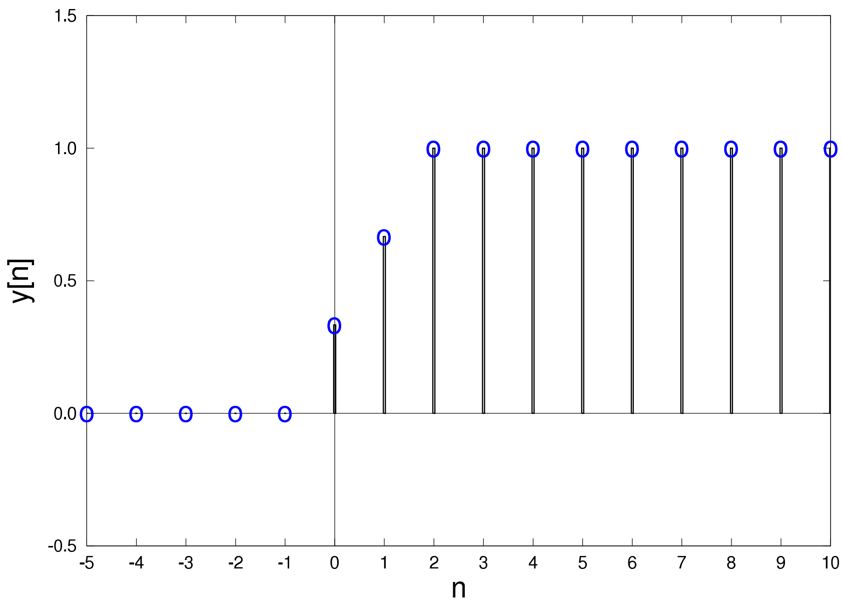
- $h[n < 0] = y[n < 0] = 0$
- $h[0] = y[0] = \frac{1}{3}(\delta[0] + \delta[-1] + \delta[-2]) = \frac{1}{3}$
- $h[1] = y[1] = \frac{1}{3}(\delta[+1] + \delta[0] + \delta[-1]) = \frac{1}{3}$

- $h[2] = y[2] = \frac{1}{3}(\delta[+2] + \delta[+1] + \delta[0]) = \frac{1}{3}$
- $h[n > 2] = y[n > 2] = 0$

Moving average of a step function is then:

$$y[n] = \sum_{k=0}^{\infty} h(n-k)$$

$$= \begin{cases} 0 & \text{if} \quad n \leq 0 \\ 1/3 & \text{if} \quad n = 0 \\ 2/3 & \text{if} \quad n = 1 \\ 1 & \text{if} \quad n \geq 2 \end{cases}$$



2.9 Scaled Impulse Function Inputs

What happens when the step function is given by:

$$u[n-v] = \begin{cases} a & \text{if} \quad n - v \geq 0, \\ 0 & \quad \text{otherwise} \end{cases} ?$$

The discrete impulse function version is

$$x[n] = \sum_{k=0}^{\infty} a\delta[n-k].$$

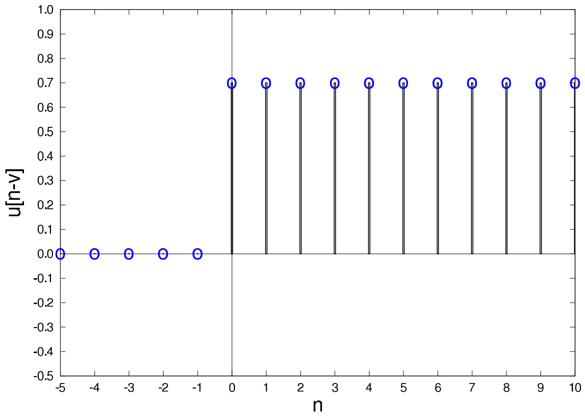
Using the *Principle of Superposition*:

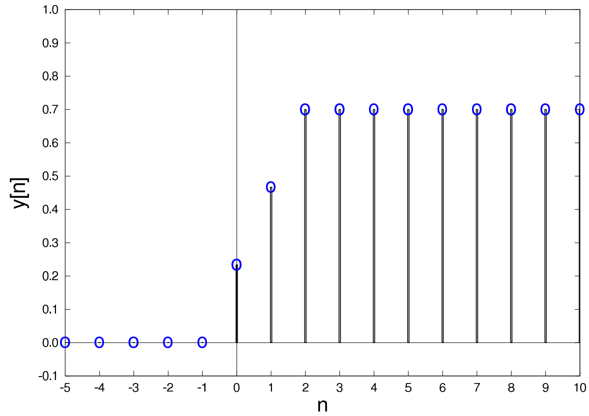
$$y[n] = \sum_{k=0}^{\infty} ah[n-k].$$

Example Moving average filter, $k = 3$

$$y[n] = \begin{cases} 0 & \text{if} \quad n \leq 0 \\ a/3 & \text{if} \quad n = 0 \\ 2a/3 & \text{if} \quad n = 1 \\ a & \text{if} \quad n \geq 2 \end{cases}$$

Example Moving Average and $a = 0.7$





3 Digital Convolution

What happens if the scale of the input impulse functions (a) varies with n? *i.e.*

$$x[n] = a[n]\delta[n-k].$$

Using the *Principle of Superposition* we get

$$y[n] = \sum_{k=-\infty}^{\infty} a[k]h[n-k].$$

This is known as the **Convolution Sum**.
Example

$$x[n] = \begin{cases} 0 & \text{if} \quad n < 0 \\ a[0] & \text{if} \quad n = 0 \\ a[1] & \text{if} \quad n = 1 \\ 0 & \text{if} \quad n \geq 2 \end{cases},$$

which is the same as $x[n] = a[0]\delta[n] + a[1]\delta[n-1]$. Then

$$y[n] = a[0]h[n] + a[1]h[n-1].$$

3.1 Example

Q. Find $y[n]$ if $a[0] = 1$ and $a[1] = 2$ using the impulse response of the moving average filter, $k = 3$.
A.

$$y[n] = a[0]h[n] + a[1]h[n-1] = h[n] + 2h[n-1]$$
$$y[-1] = h[-1] + 2h[-2] = 0 + 0 = 0$$
$$y[0] = h[0] + 2h[-1] = 1/3 + 0 = 1/3$$
$$y[1] = h[1] + 2h[0] = 1/3 + 2/3 = 1$$
$$y[2] = h[2] + 2h[1] = 1/3 + 2/3 = 1$$
$$y[3] = h[3] + 2h[2] = 0 + 2/3 = 2/3$$
$$y[4] = h[4] + 2h[3] = 0 + 0 = 0$$

3.2 Digital Convolution Trivia

Convolution is often represented by an asterik:

$$y[n] = \sum_{k=-\infty}^{\infty} a[k]h[n-k] = a[n] * h[n]$$

Convolution is commutative:

$$y[n] = a[n] * h[n] = h[n] * a[n]$$

$$= \sum_{k=-\infty}^{\infty} h[k]a[n-k].$$

Convolution is associative: *cascaded systems*

$$\{x[n] * h_1[n]\} * h_2[n] = x[n] * \{h_1[n] * h_2[n]\}$$

Convolution is distributive: *systems in parallel*

$$x[n] * \{h_1[n] + h_2[n]\} = x[n] * h_1[n] + x[n] * h_2[n]$$

3.3 Example

Original signal: (1 cycle every 104 samples)

$$x_1[n] = \sin(n\pi/52)$$

Input signal:

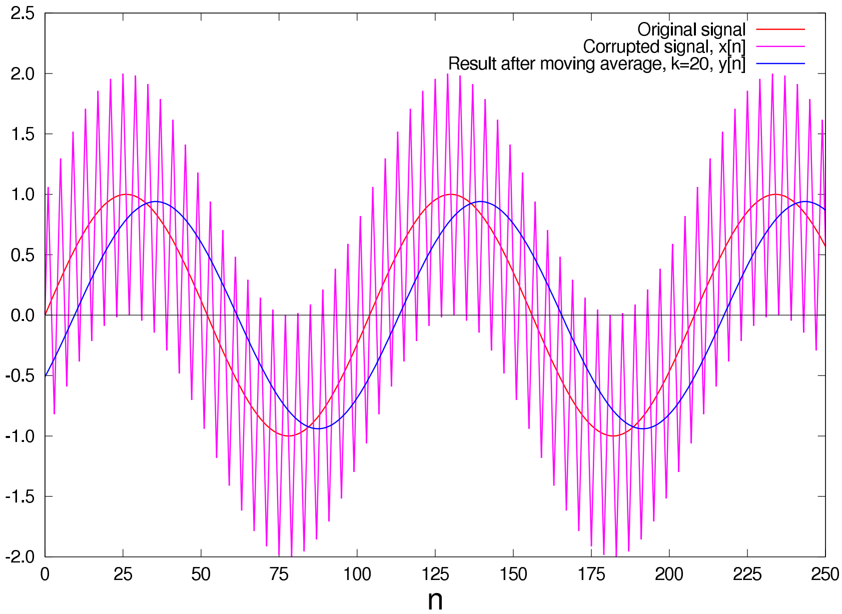
$$x[n] = x_1[n] + x_2[n].$$

Noise signal: (1 cycle every 4 samples)

$$x_2[n] = \sin(n\pi/2).$$

Moving average filter, $k = 20$:

$$y[n] = \frac{1}{20} \sum_{k=0}^{k=19} x[n-k].$$



4 Digital Cross-Correlation

Cross-correlation can be used to compare 2 signals.

- If $x_1[n]$ and $x_2[n]$ are two signals then digital cross-correlation is defined:

$$y[n] = \sum_{m=-\infty}^{\infty} x_1^*[m]x_2[n+m]$$

  where $x_1^*[n]$ is the complex conjugate of $x_1[n]$.

- For a real signal $x_1^*[n] = x_1[n]$.
- l is the *lag*.
- If $x_1[n]$ and $x_2[n]$ are the same signal but with a delay between them, then $y[l]$ is at a maximum when l is equal to this delay.

4.1 Example

Q. Given $x_1 = (0\ 0\ 0.5\ 0.7\ 0)^{\mathrm{T}}$ and $x_2 = (0\ 0.5\ 0.7\ 0\ 0)^{\mathrm{T}}$.
Calculate the cross-correlation for these two real signals.
A. Cross correlation for a real signal is:

$$y[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n+m].$$

There are 5 elements in these vectors so (changing the limits):

$$y[n] = \sum_{m=0}^{4} x_1[m]x_2[n+m].$$

We can then calculate the results. Some example calculations:

$$\begin{aligned}
y[l=0] =& \overbrace{x_1[0]\times x_2[0]}^{l=0,m=0} + \overbrace{x_1[1]\times x_2[1]}^{l=0,m=1} + x_1[2]\times x_2[2] + x_1[3]\times x_2[3] + \overbrace{x_1[4]\times x_2[4]}^{l=0,m=4} \\
=& 0\times 0 + 0\times 0.5 + \underline{0.5\times 0.7} + 0.7\times 0 + 0\times 0 = \underline{0.5\times 0.7} = 0.35 \\
y[l=1] =& \overbrace{x_1[0]\times x_2[0+1]}^{l=1,m=0} + \overbrace{x_1[1]\times x_2[1+1]}^{l=1,m=1} + x_1[2]\times x_2[2+1] + \\
& x_1[3]\times x_2[3+1] + \overbrace{x_1[4]\times x_2[4+1]}^{l=1,m=4} \\
=& 0\times 0.5 + 0\times 0.7 + 0\times 0 + 0\times 0 + 0\times 0 = 0
\end{aligned}$$

Here are the results for each combination of l and m values:

l	m **0**	**1**	**2**	**3**	**4**	$y[l]$
-5	0	0	0	0	0	0
-4	0	0	0	0	0	0
-3	0	0	0	0	0	0
-2	0	0	0	0.35	0	0.35
-1	0	0	0.25	0.49	0	$\underbrace{0.74}_{PEAK}$
0	0	0	0.35	0	0	0.35
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

Some observations can be made:

- A peak is located at $l = -1$.
- l is the *lag*, so there is a lag of -1.
- This means that x_1 has some similar signal as x_2 but lagged by 1 step.
- We can also see from the signal definitions

  $$x_1 = (0\ 0\ 0.5\ 0.7\ 0)^{\mathrm{T}}$$

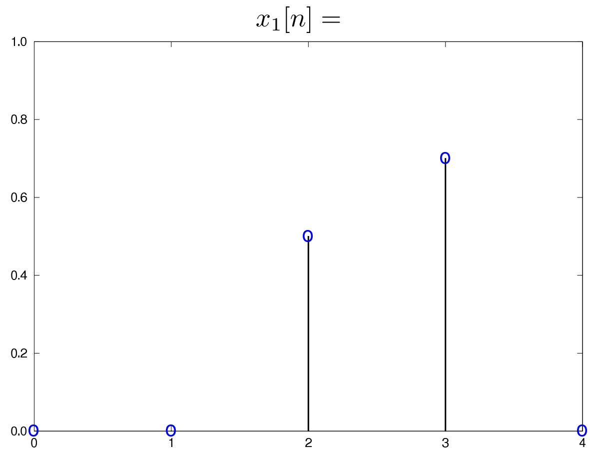
  and

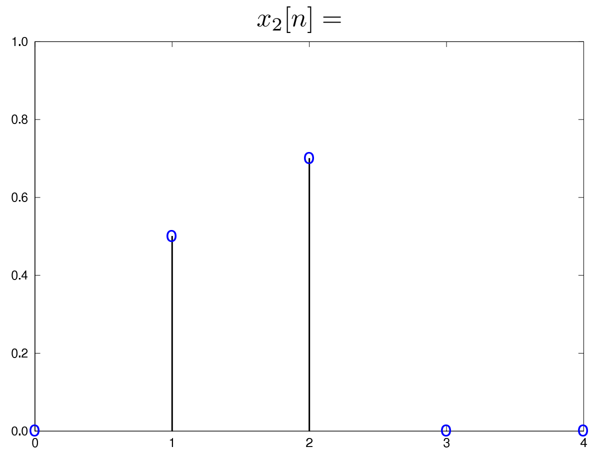
  $$x_2 = (0\ 0.5\ 0.7\ 0\ 0)^{\mathrm{T}}$$

  so that

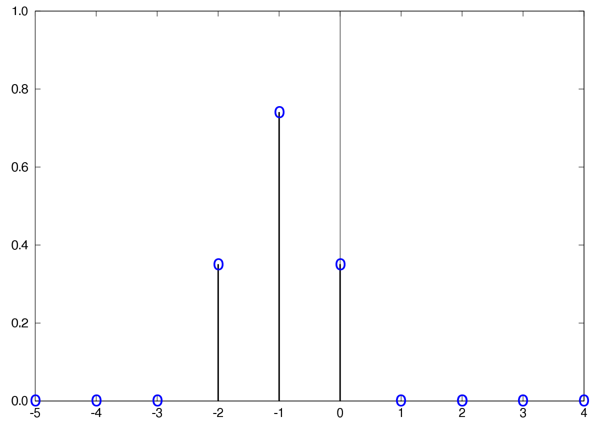
  $$x_1[n-1] = x_2[n]$$

  .





The result of the digital cross-correlation, $y[l] =$



5 Difference Equations

Difference equations are the name given to the equations that describe the digital signals and systems. For example the equation for the moving average filter with $k = 3$:

$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}$$

is known as a difference equation.
Difference equations for LTI systems can always be put in the form:

$$\sum_{m=0}^{N} a[m]y[n-m] = \sum_{m=0}^{M} b[m]x[n-m].$$

So for our moving average filter:

- $M = 2$ and $N = 0$.
- $a[m]$ and $b[m]$ are known as coefficients.
- For the moving average output y there is only one coefficient, $a[0] = 1$.
- For the moving average input x, there are three coefficients $b[0] = b[1] = b[2] = \frac{1}{3}$.

6 Summary

There are many types of signal however there are some types of signal that are commonly found in the DSP literature such as the unit impulse function, the scaled or shifted unit impulse function, the unit step function, the ramp function and sinusoidal functions. Various operations can be applied to these various signal types and if we carefully select particular operations with particular signal types, it can give us insight into the functioning of that operation. For example, convolution of a unit impulse function with some other function just results in the reproduction of that other function. This is useful for characterizing Linear Time Invariant systems and it is called the unit impulse response.