PS403 - Digital Signal processing

I. DSP - Introduction

Key Text:

Digital Signal Processing with Computer Applications (2nd Ed.)

Paul A Lynn and Wolfgang Fuerst, (Publisher: John Wiley & Sons, UK)

We will cover in this section

Some examples of algorithmns

Basic terminology

Definition: Numerical manipulation of sampled (data) signals

Example 1. ***200 day (moving) average of the dollar price of gold !***

This genre of filter is variously referred to as:

Savitsky Golay - Moving Average - Adjacent Channel Average - Low Pass Filter

In general we will be writing y[n] = f(x[n]) where y is the table of outputs and x[n] is the list of input values.

200 point moving average: x[n] - input, y[n] - output - see Fig 1.3

$$y[n] = \frac{1}{200}\left(x[n] + x[n-1] + \ldots\ldots\ldots\ldots x[n-199]\right)$$

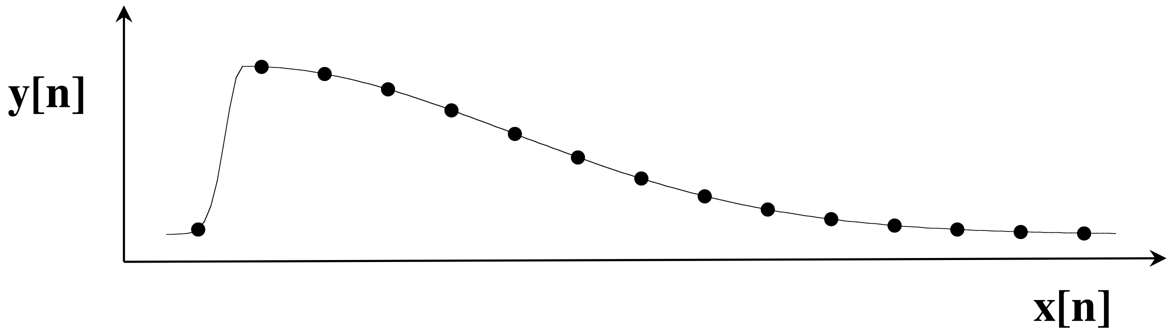
OR

$$y[n] = \frac{1}{200}\sum_{k=0}^{k=199} x[n-k] \quad - (1)$$

Recurrence Formula

NB: This is a **non-recursive formula** (filter) since each y[n] is computed from an existing (pre-recorded) set of input signal values x[n]

Recursive moving average algorithmn



We can make this filter *recursive* !!

Notice that you can write equation 1 as:

$$y[n] = y[n-1] + 0.005\{x[n] - x[n-199]\} - \textbf{(2)}$$

Much more computationally efficient than the non recursive version -

Recursive - compute current value of output y[n] from one or more previous values y[n-m], m = 1, 2, 3,...

Example 2. ***Bandstop filter***

We take the example of a typical EKG (Electro-Cardiogram) signal. Such signals are usually low power (microvolts/ nanoamps) and so power can be as low as femtowatts ! As a result, 50 Hz mains frequency often appears as 'pickup' superimposed on the EKG signal. The EKG leads act as 'aerials'. **See Fig 1.4.**

So we need a digital filter algorithmn that rejects 50 Hz (or 60 Hz in the USA)

$$y[n] = 1.8523y[n-1] - 0.94833y[n-2] + x[n] - 1.9021x[n-1] + x[n-2] \quad - (3)$$

Notice that this is also a recursive formula (filter)

Recomputing y[n] (filtered signal) from the distorted sample values x[n] you obtain the very clean EKG signal in figure 1.4c - Good eh ?????

NB1: A digital filter is designed for a specific sampling rate. Here the sampling rate is 1 KHz for 50 Hz rejection. However, if the contaminated signal is sampled at 1.2 KHz, the rejected frequency will be 60 Hz

NB2: The real skill you will learn is the ability to design these algorithms - i.e., to be able to determine what terms should be included and what the multipliers are.

Example 3. ***Bandpass filter*** - **See Fig 1.5.**

$$y[n] = 1.5y[n-1] - 0.85y[n-2] + x[n] \quad - (4)$$

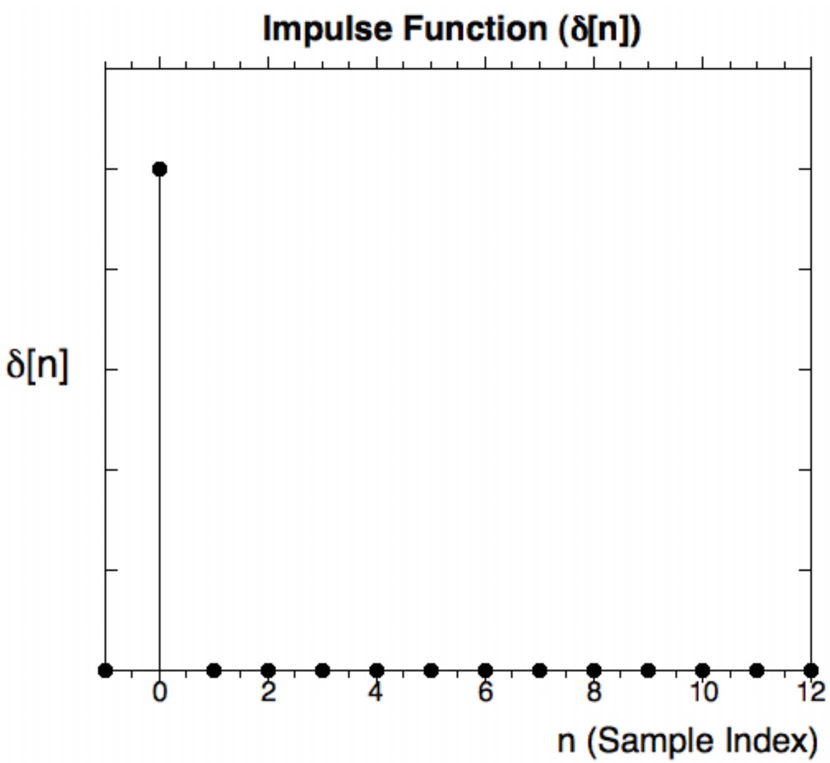
Notice that this is also a recursive formula (filter)

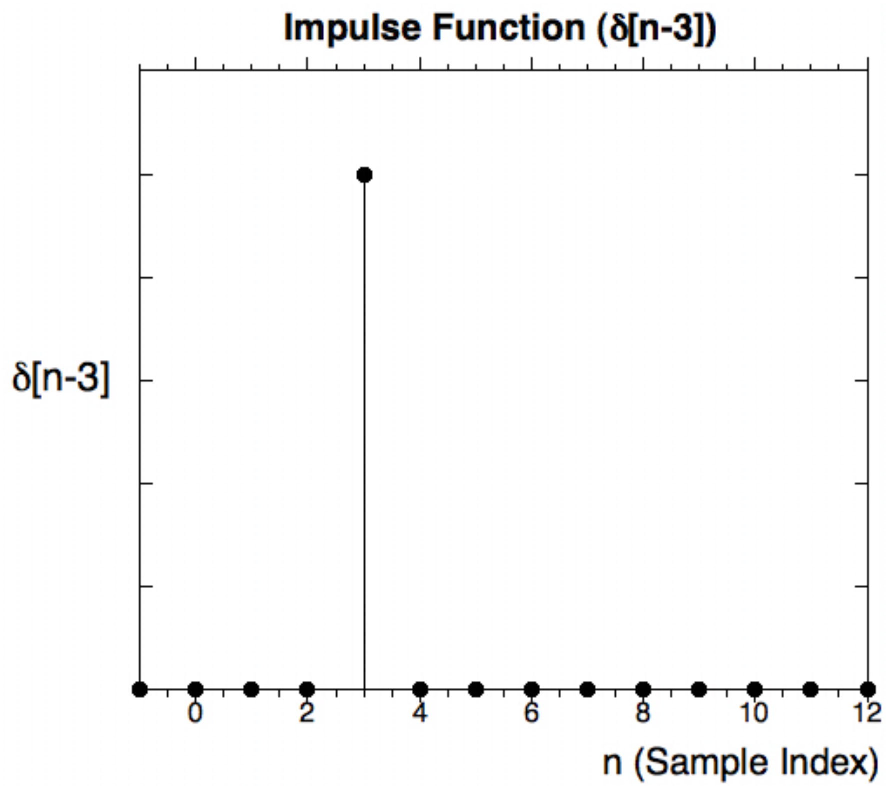
Recursive filter designed to transmit a sinusoidal signal which has been recorded at 10 samples/cycle. For lower or higher sample rates (samples/cycle) the gain is lower. Hence the filter exhibits a 'bandpass' action.

Figure 1.5 shows the action of this filter on a 'swept' frequency waveform running from 20 samples/cycle (on the left hand side) to 5 samples per cycle (on the right hand side). As you can see gain is maximum at 10 samples/cycle.

Standard Digital Signals

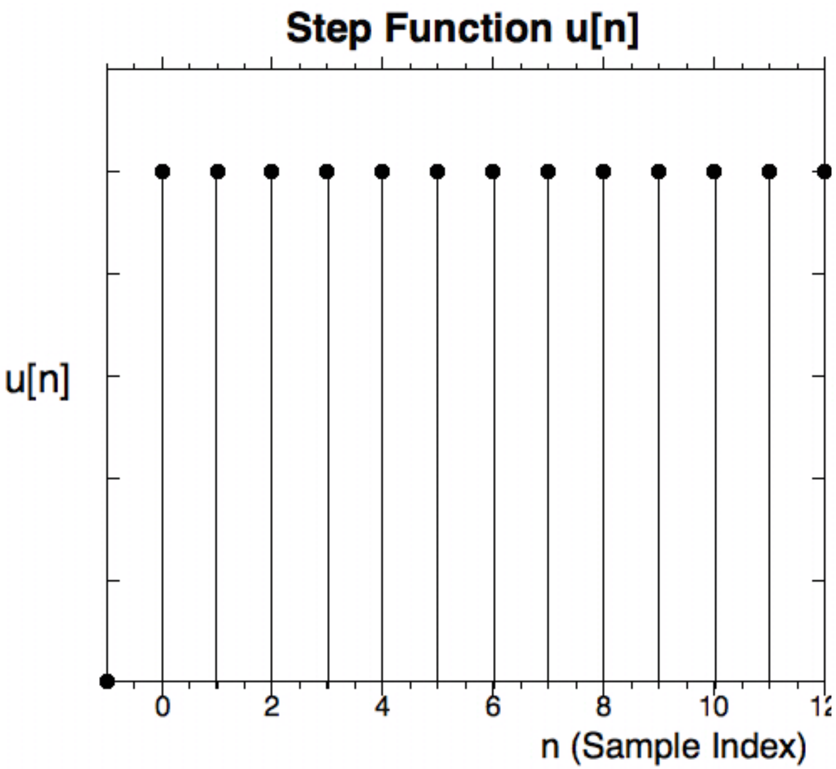
1. Impulse or 'δ' function

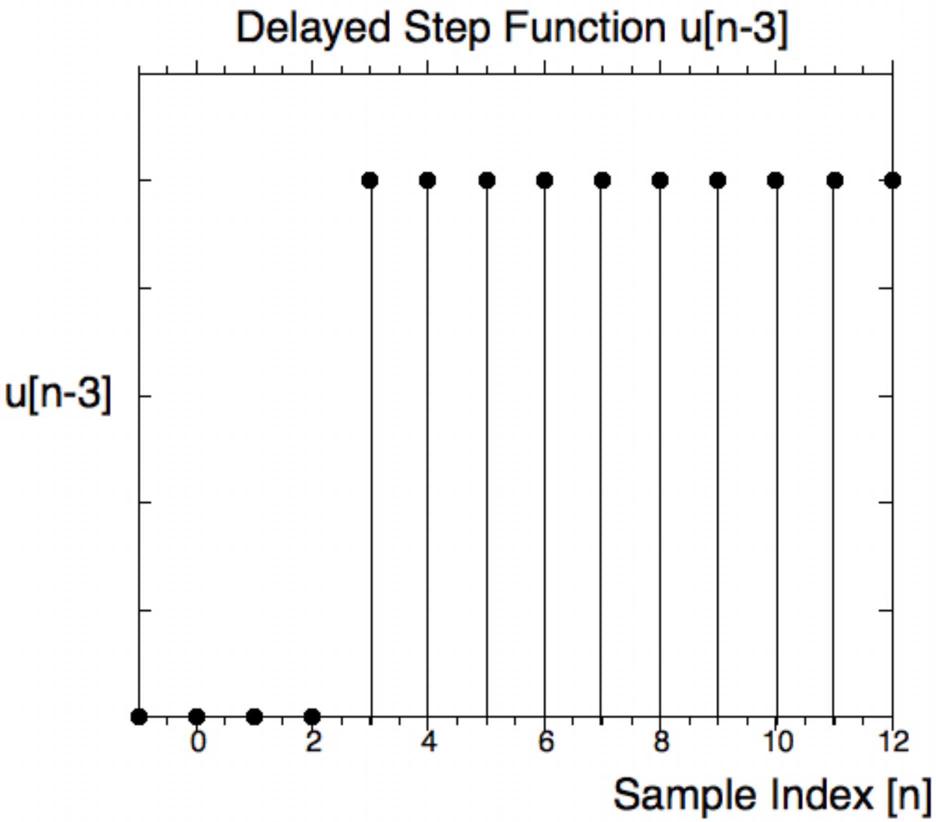




Standard Digital Signals

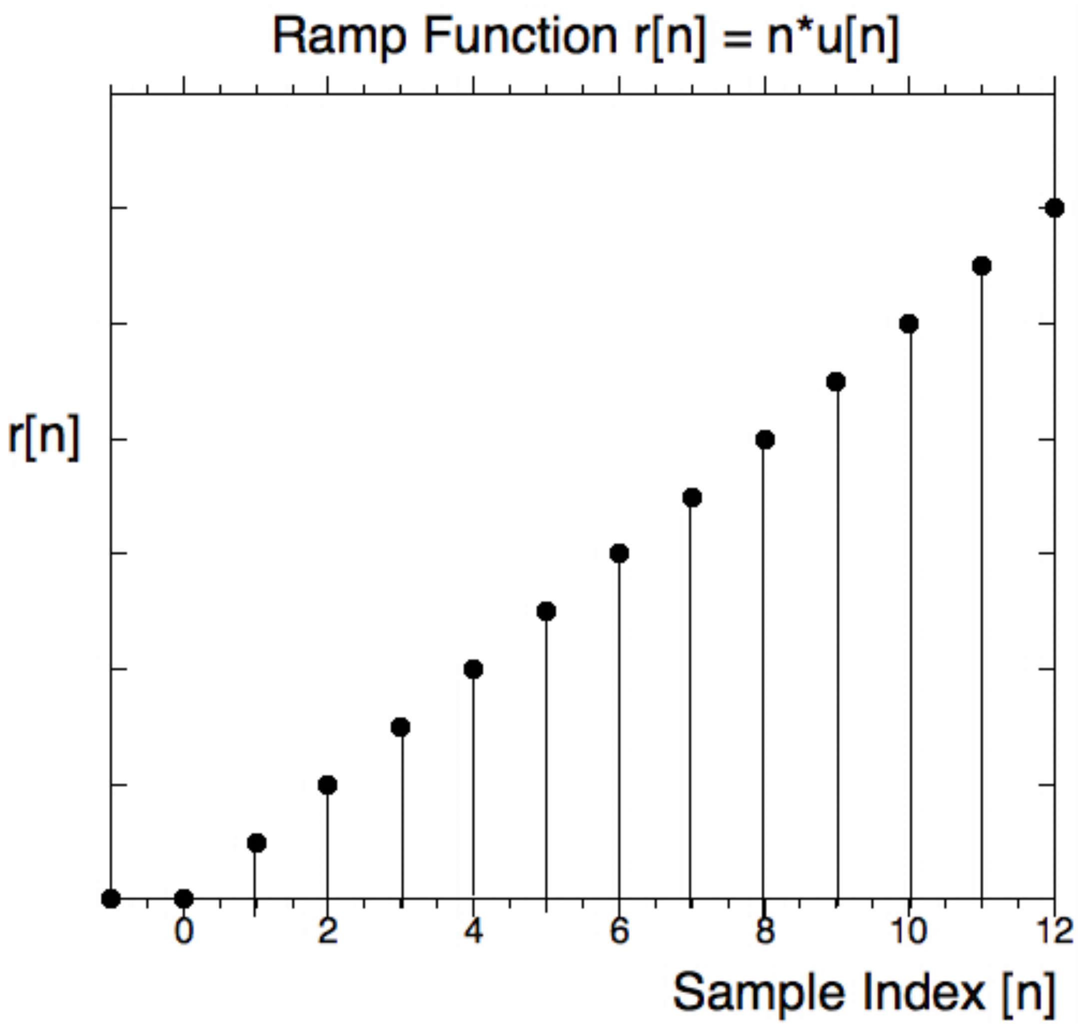
2. Step function





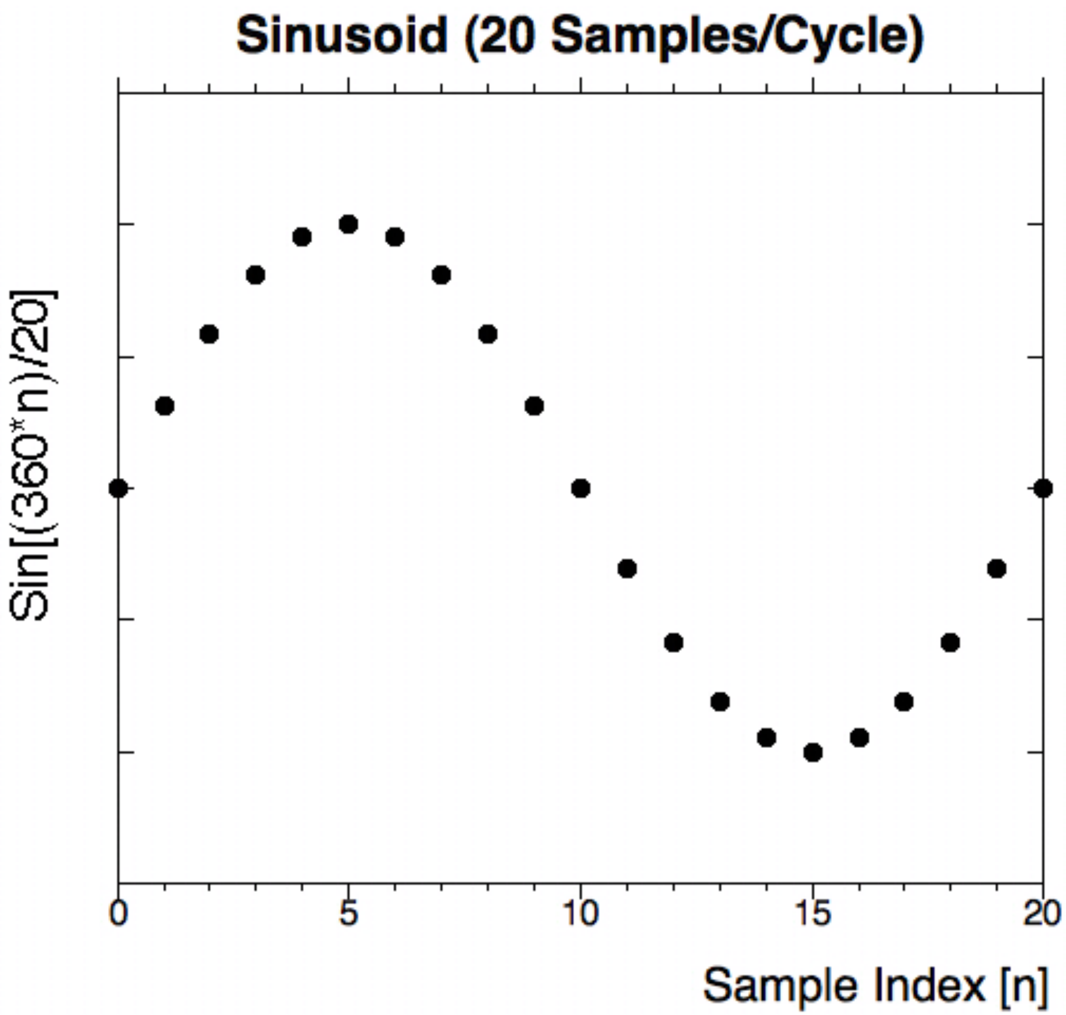
Standard Digital Signals

3. Ramp function



Standard Digital Signals

4. Sinusoidal Function



Standard Digital Signals

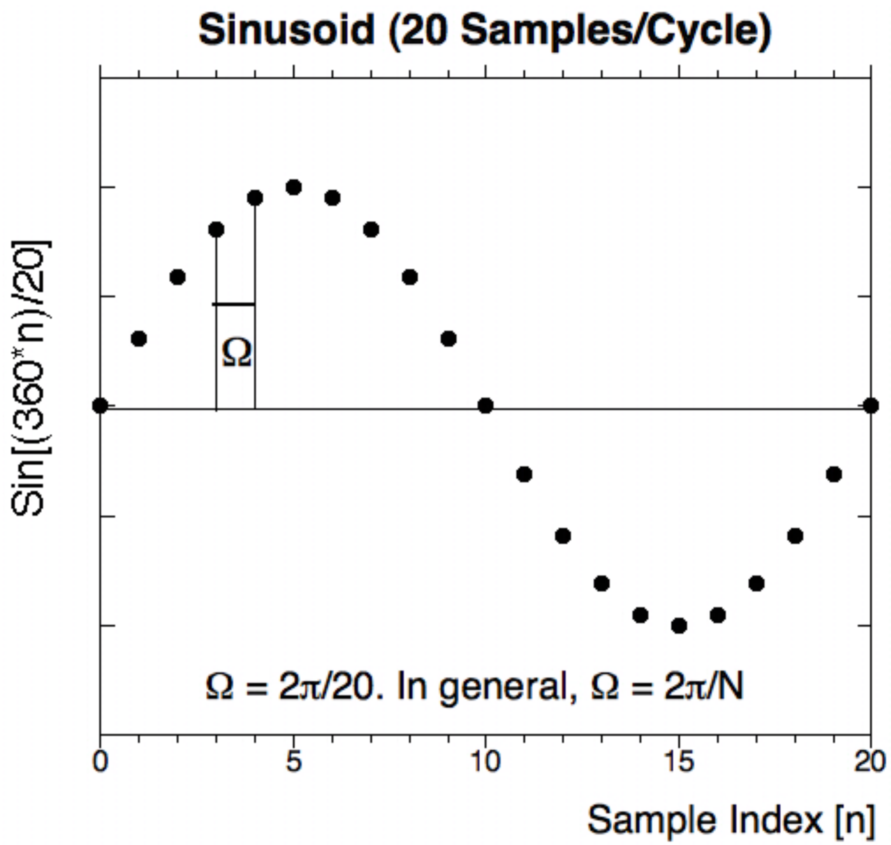
Q. How would you synthesise a pulse of width (or duration) 8 samples, starting at $n = -3$ and ending at $n = 4$?

$$\text{Ans. } x[n] = u[n+3] - u[n-5]$$

i.e., Add a unit step function starting at $n = -3$ to a step function, delayed by 5 samples to start at $n = 5$,which has an amplitude of -1 per point !!

Draw it and see for yourself !!!

Sampling frequency and samples/cycle



A DSP system does not understand the concept of absolute frequency !!

However, it can understand/deal with 'samples per cycle' and this is how frequency is represented in DSP.

So a 1 Hz sinusoid sampled at 20 Hz gives the same data set as a 1MHz sinusoid sampled at 20 MHz - i.e., 20 samples/cycle !

NB: When we design DSP filters (algorithmns), we do so for a specific 'sampling frequency' given in 'samples per cycle' but not a specific absolute frequency !!

Sampling frequency and samples/cycle

From the previous figure, 1 full cycle = 2π radians.
The sampling interval = Ω radians, so N = $2\pi/\Omega$

In absolute time, the sampling interval = T_s (seconds)
The sampling frequency, $f_s = 1/T_s$, or $\omega_s = 2\pi/T_s$.

Connection: $\omega_s T_s = 2\pi = N\Omega$.

In general -

$$x[n] = \mathrm{Sin}(n\Omega) = \mathrm{Sin}\,(n.2\pi/N)$$

Linear Time-Invariant (LTI) Systems

This module will deal only with linear signals and systems, i.e., we will not cover more advanced topics such as non-linear DSP !

Time invariant means we will ignore systems whose properties vary with time and hence require the application of adaptive DSP strategies to control them.

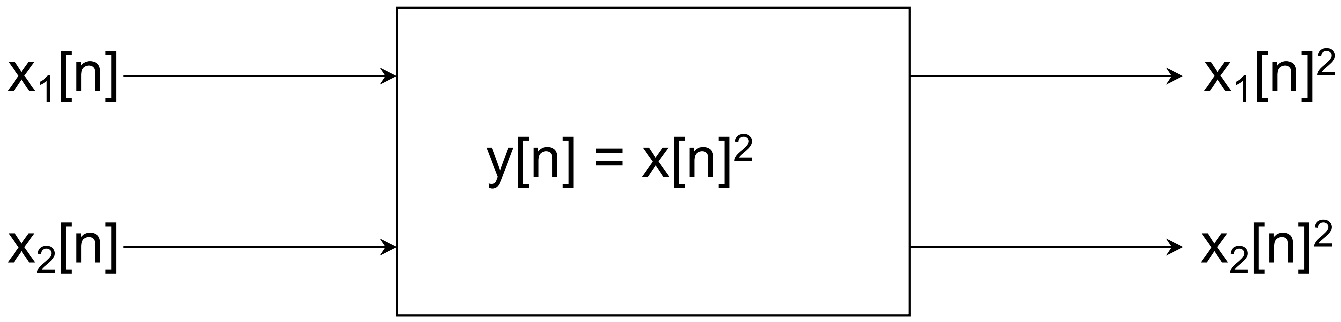
Linear:
If the input to a system is composed of a superposition of signals (stimuli), then the output is simply a superposition of the system's response to each individual stimulus (or input signal component).

i.e., if $x[n] = a.x_1[n] + b.x_2[n] + c.x_3[n] + \ldots\ldots\ldots$

then $y[n] = a.y_1[n] + b.y_2[n] + c.y_3[n] + \ldots\ldots\ldots$

Linear Time-Invariant (LTI) Systems

Non-linear example: Squaring system



For $x[n] = x_1[n] + x_2[n]$, $y[n] = (x_1[n] + x_2[n])^2 \neq x_1[n]^2 + x_2[n]^2$

Also, frequency is not preserved in non linear systems !!

For $x[n] = \mathrm{Cos}(n\Omega)$, $y[n] = \mathrm{Cos}^2(n\Omega) = 1/2\{1 + \mathrm{Cos}\,(2n\Omega)\}$,
i.e., the squaring operation doubles the frequency of the input sinusoid.

Other Properties of LTI Systems

Association - A complex LTI system may be analysed by breaking it down into a number of simpler systems. Conversely, a complex system can be synthesised from a number of simpler component systems

Commutation - Any array of LTI systems arranged in series (cascaded) may be re-arranged in any order without affecting the system response

Causality - In a causal system, the output signal depends only on the present (or previous) input values

Stability - A stable system is one which produces a finite (or bounded) output for a bounded input

Invertibility - If a digital processor produces an output y[n] for an input x[n], it will produce and output x[n] for an input y[n] - (reversibile system)

Memory - A DSP system must possess memory if the present output y[n] depends on previous input x[n-m] or output y[n-m] values

LTI Systems

Digital LTI Processor Operations:

1. Storage and delay
2. Addition and subtraction
3. Multiplication by a constant

See **Figure 1.19** for block diagram representations of typical DSP processors employing 1, 2 & 3 above.

(1.19a) $y[n] = x[n] + x[n-1] + x[n-2] + x[n-3] + x[n-4] + ...$

(1.19b) $y[n] = y[n-1] + x[n]$ (recursive filter - feedback !)

(1.19c) $y[n] = 1.8523y[n-1] - 0.94833y[n-2] + x[n] - 1.9021x[n-1] + x[n-2]$
- Bandstop filter for 20 samples/cycle !

NEXT SECTION -
TIME DOMAIN ANALYSIS

IMPULSE RESPONSE AND CONVOLUTION