# EEE6440
# Advanced Digital Signal Processing (ADSP)

**Dr Charith Abhayaratne**

Tel: 25893
Email: c.abhayaratne@sheffield.ac.uk
Office: F176

- Course Delivery:
  - Lectures: 2 hours/week Monday @ 12 pm (WL) & **Thursday @ 2 pm (CA)**
  - For 12 weeks

- This unit aims to

  - provide an understanding of filter design concepts
  - extend the filter design into scenarios where sampling rate conversions, filter bank and adaptive filtering are required.
  - introduce the concept of transforms.
  - introduce the concept of random signals and their analysis.
  - provide hands-on experience in advanced signal processing.

- By the end of the unit, students will be able to demonstrate the ability to

  - carry out filter design and implementation for sampling rate conversions including decimation (d), interpolation (i) and a rational factor (i/d). (CA)
  - understand the poly-phase representations of filter banks, formulate different filter bank design and provide the corresponding solutions; (CA)
  - perform simple analysis and compute statistics of random signals; (WL)
  - understand the Wiener filter solution and the Least Mean Square type adaptive algorithms and apply them to solve adaptive filtering problems; (WL)
  - design, implement and use simple signal transforms in various applications. (CA)
  - use MATLAB in designing and implementing the above concepts and using them in suitable applications (CA &WL)

- Syllabus for CA's Lectures
  - Filters
    - How information is represented in signals
    - Time-domain /frequency domain parameters
    - Low-pass/high-pass, band-pass filters
    - Moving average/ recursive filetrs
    - Windowed-Sinc /Cherbyshev filters
  - Filter Banks
    - Sampling rate conversions
    - Poly-phase representation
    - Multirate filtering
    - Filter-bank design
  - Transforms
    - Introduction to signal transforms
    - Discrete Cosine transform / Wavelet transform
    - Transform-domain processing

- Assessment

  - Final exam 100% in January
  - 2 hours
  - Two parts (50% each)
    - Part A: CA lectures
    - Part B: WL lectures
  - For each part
    - Answer 2 questions from 3

- Prerequisites
  - Content in EEE309/EEE6033 (in S2)
    - http://hercules.shef.ac.uk/eee/teach/resources/eee309/eee309.html
    - Discrete time signals & systems
    - z-transform Sampling of continuous-time signals
    - Transform analysis of linear time-invariant (LTI) systems
    - Structures for discrete-time Systems
    - Discrete Fourier Transform
    - IIR & FIR Filter Design
  - If not familiar with above,
    - Revise ASAP

- Reference Books (for background reading only)

  - Digital Signal Processing
    J. Proakis & D. Manalokis (Prentice Hall).

  - Wavelets & Subband coding
    M. Vetterli & J. Kovacavic.

(available online at http://www.waveletsandsubbandcoding.org/)

## Topic 01: Revision – Background knowledge

- Signal Processing Preliminaries
  - Discrete time signals & systems
  - Convolution
  - Impulse & Frequency response
  - Filters (low pass and high pass)
  - Transforms

Background reading: Digital Signal Processing (Proakis / Manolakis)
Chapters 1 and 2.
(Or Introduction and Discrete time systems and signals chapters on any DSP text book)

# Discrete time signals

- A discrete time signal x(n) is a function of an independent variable that is an integer.

- We can assume that x(n) is defined for all integer values of n for $-\infty < n < \infty$

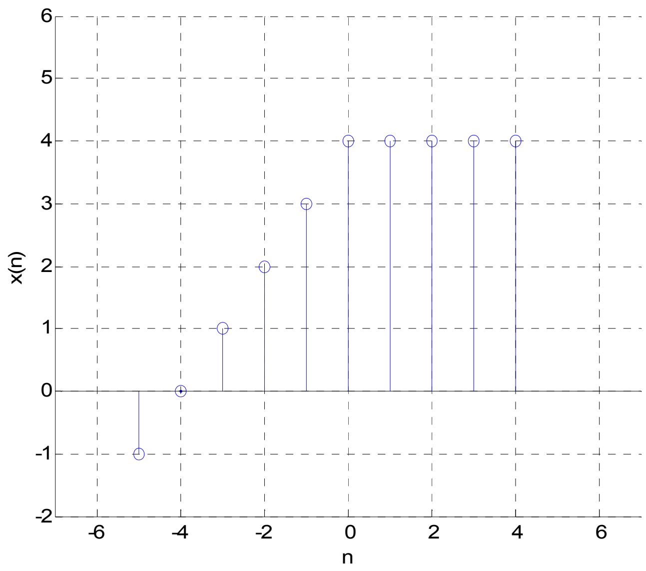
- We refer x(n) as the $n^{th}$ sample of the signal.
- $x(n) \equiv x_a(nT)$ , where $x_a$ is the analogue signal, T is the sampling interval and n is the sampling index.

- Commonly used signals:
  - Unit impulse function ---------- ?
  - Unit step signal ------------ ?

# Discrete time signals

- Simple manipulations of discrete time signals
- What is x(n)?

  x(n) =




- Time shifting ---- x(n-k)

  x(n-3)?

  x(n+2)?

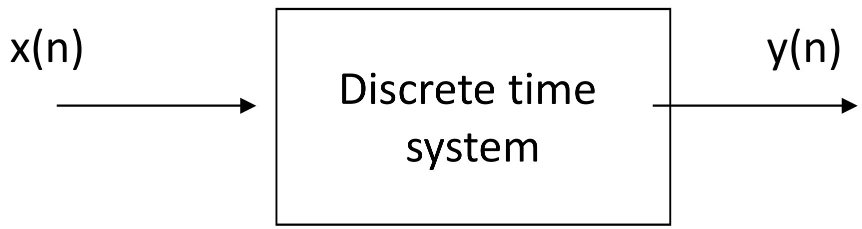
- Folding ----- x(-n)
- Time scaling ---- x(mn)

  x(2n)?

# Discrete time systems

- A discrete time system is an operation or a set of operations performed on a discrete time input signal x(n) to produce the discrete time output signal y(n).

- We can also say x(n) is transformed to y(n) by the system.




$$y(n) \equiv T[x(n)]$$

- The output when the input is the impulse function is called the impulse response of a system . $h(n,k) = T[\delta(n-k)]$

# Discrete time systems

**<u>Time (shift or translation) invariant systems</u>**

- A system is called time invariant if its input-output characteristics do not change with time.
- That means for a system

$$x(n) \rightarrow y(n)$$

$$x(n-k) \rightarrow y(n-k)$$, for every input signal x(n) and every time shift k.

- How to check? Check whether the shifted output (y(n-k)) is the same as the output computed using the shifted input (T[x(n-k)]).

# Discrete time systems

**<u>Time (shift or translation) invariant systems</u>**

- Determine the following are time invariant or not
  - $y(n)=x(n)-x(n-1)$
  - $y(n)=nx(n)$
  - $y(n)=x(-n)$
  - $y(n)=x(2n)$
  - $y(n)=x(n)\cos(wn)$

# Discrete time systems

**Linear systems**

- A system is called linear if it satisfies the superposition principle.
- The response of the system to a weighted sum of signals is the same as the corresponding weighted sum of the responses of the system to each of the individual input signals.

- $T[a_1x_1(n) + a_2x_2(n)] = a_1T[x_1(n)] + a_2T[x_2(n)]$
- This is due to scaling and additive properties of a linear system.

# Discrete time systems

**Linear systems**

- Determine the following are linear or non-linear
  - $y(n)=nx(n)$
  - $y(n)=x(n^2)$
  - $y(n)=x^2(n)$
  - $y(n)=x(2n)$

# Discrete time systems

**Causal systems**

- A system is called causal if the output of the system at any time [y(n)] depends only on the present [x(n)] and past inputs [x(n-1), x(n-2),….], but not the future inputs [x(n+1), x(n+2),….].

- Otherwise the system is called non-causal.

- What are the practical implications?

# Discrete time systems

**Interconnection of systems**

- Determine the combined system (T) of two systems ($T_1$ and $T_2$) interconnected:

(a) in cascade or

(b) in parallel

- For cascade interconnections, is the order of performance ($T_1$ followed by $T_2$ or $T_2$ followed by $T_1$) important?

# Convolution

**<u>Response of a linear time invariant (LTI) system to an arbitrary input x(n).</u>**

We know: $y(n) = T[x(n)]$

$$h(n) = T[\delta(n)]$$

An arbitrary signal x(n) can be expressed as a sum of weighted impulses:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

Now we can write the output y(n):

$$\begin{aligned}
y(n) = T[x(n)] &= T\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] \\
&= \sum_{k=-\infty}^{\infty} x(k)T\left[\delta(n-k)\right] \\
&= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
&= x * h
\end{aligned}$$

# Convolution

**Response of a linear time invariant (LTI) system to an arbitrary input x(n).**

Steps:

1) folding: fold h(k) about k=0 to get h(-k)
2) Shifting: shift h(-k) by $n_0$ to the right if n is positive to get $h(n_0-k)$
3) Multiplication: multiply x(k) by $h(n_0-k)$ to get the product sequence
4) Summation: sum all the values of the product sequence.

Repeat the above steps 2 to 4 for all n.

# Convolution

**Computation by hand**

A good way to compute h*x is to arrange it as an ordinary multiplication. But don't carry digits from one column to the other.

e.g., consider {x(0), x(1), x(2)} & {h(0), h(1), h(2)}

| | | | | |
|---|---|---|---|---|
| | | x(2) | x(1) | x(0) |
| | | h(2) | h(1) | h(0) |
| | | (2) | (1) | (0) |
| | (3) | (2) | (1) | |
| (4) | (3) | (2) | | |
| y(4) | y(3) | y(2) | y(1) | y(0) |

Compute the convolution for x(n)={4,2,3} and h(n)={2,5,1}

# Convolution

Convolution of x(n) by h(n) in time domain becomes multiplication of X by H in frequency domain, where X & H are the Fourier transform of h.

$$X\left(e^{j\omega}\right)=\sum_{-\infty}^{\infty}x(n)e^{-jn\omega}=X\left(\omega\right)$$

$$H\left(e^{j\omega}\right)=\sum_{-\infty}^{\infty}h(n)e^{-jn\omega}=H\left(\omega\right)$$

$$Y(\omega)=H(\omega)X\left(\omega\right)$$

Similarly in the z-transfom domain

$$X(z)=\sum_{-\infty}^{\infty}x(n)z^{-n}$$

$$H(z)=\sum_{-\infty}^{\infty}h(n)z^{-n}$$

$$Y(z)=H(z)X(z)$$

# Filters

A filter is a linear time-invariant operator.

It acts on input signal x and the output signal y is the convolution sum of x with the fixed vector h, which is the impulse response of the system.

The values of the vector h are known as the filter coefficients. E.g., h(0), h(1),……

Low pass filters & High pass filters  (later in detail)

# Transforms

A transform of a signal is a new representation of that signal.

Consider the following system of equations:

$y(0)= x(0) + x(1)$

$y(1)= x(0) - x(1)$

$y(2)= x(2) + x(3)$

$y(3)= x(2) - x(3)$

This transforms x into y using a 4-point transform.

In other words the signal is represented in y-domain using the linear combinations of signal components in the x domain.

What is the inverse transform for the above transform?

# Homework: MATLAB

- Exercise 1:
  - Create the time axis values for 10 cycles with 512 data points using t=linspace(0,10, 512);
  - Consider the signal x=3sin(5t)-6cos(9t)
  - Plot x
  - Add random noise n to obtain a noisy signal y=x+n
  - Consider you are using a 3 point moving average filter. What is "h" for this filter?
  - Use convolution to find the cleaned signal "z"
  - Check the size of the output z
  - Plot all x, y and z in the SAME figure

  - Think of an alternative approach for de-noising using the Fourier Transform and implement it using MATLAB