Lab 2 – DT system applications

Objectives: In this lab we will learn to build simple discrete time (DT) systems to perform some tasks and recognize the patterns of pole-zero, ROC properties, and impulse response of a second order system.

2.1. A **Moving Average** (MA) system is used to detect trends from a given signal. It is related to the accumulator. It finds the average of the signal over the past few samples.

Accumulator: $y[n] = \sum_{k=-\infty}^{n} x[k]$ Moving average system: $y[n] = \frac{1}{N} \sum_{k=n-N+1}^{n} x[k]$

a) Write a Matlab script to implement the above MA system.
b) Test the system with a unit step function ($u[n]$) as input.
c) Find the trend of the given test sequence $s[n]$, the signal is provided in q1.mat file.
d) Experiment with different values for N and find the value appropriate for $s[n]$. Why do you think it is appropriate?
e) Find the impulse response of the MA system and implement it using convolution. Find the trend of $s[n]$ using this implementation. Is there any difference in the result? What are the pros and cons of the two implementations?

2.2. An **Upsampler** is a system which increases the length of a given sequence and interpolates to find the values of the new samples. A popular application of upsampling is magnifying/zooming an image.

Upsampling step 1:

$y[n] = \frac{n}{M}$ if n is an integer multiple of $M > 1$

= 0 otherwise

Upsampling step 2: Estimate the value of the newly inserted samples, i.e. do interpolation.

a) Write a script to implement the upsampler with $M = 2$ and 3. Experiment with zero order hold and linear interpolation.
b) Upsample the given test sequences present in q2_1.mat and q2_2.mat files. What do you observe?

2.3. A **digital differentiator** is given as y[n] = x[n]-x[n-1]. Write a script to implement this system and find the output of this system for the following three inputs. Plot the input and output sequences.

a. $x[n] = 5(u[n] - u[n-20])$
b. $x[n] = n(u[n] - u[n-10]) + (20 - n)(u[n-10] - u[n-20])$
c. x[n] = $sin[\frac{\pi}{25}](u[n] - u[n - 100])$

2.4. A finite difference equation can be written as,

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

The solution for this equation can be found numerically using the *filter* function in Matlab.

Matlab code:

```
y = filter(b,a,x)

b = [b0, b1, b2.. bM]; a = [a0, a1, a2.. aN].
```

Note that you have to choose $a_0 \neq 0$; Output y[n] is same length as x[n].

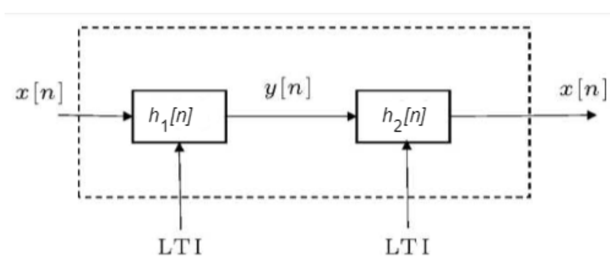
To find the impulse response h[n]

```
h = impz(b,a,n)
```

2.4.1. A second order feedback system is given as y[n] + αy[n-1] + βy[n-2] = x[n]. Plot the impulse response h[n] for this system (using the above code) for different coefficient values, i.e. α and β.

a. α = -1 and β = 0.9
b. Find the coefficients for the system such that h[n] decays monotonically.
c. Find the coefficients such that h[n] diverges monotonically.
d. Find coefficients such that h[n] grows initially and then decays as n increases to ∞.
e. Find the coefficients such that h[n] oscillates for all n.

2.5. Consider the **digital differentiator** system **S** from **2.3**. Find the difference equation for a system **S*** such that it acts as the inverse of the digital differentiator system. Thus, the cascade connection of **S** and **S*** acts like an identity system as shown below.



2.5.1 Derive and write a Matlab code to implement the **S*** system. Make subplots for $x[n]$, $y[n]$ at the input and output of each system.

2.5.2 Find the impulse response of the systems h_1 and h_2 and compare their plots.

2.5.3 Comment on the result of convolution of the systems h_1 and h_2