Lab 7

Simulation of Discrete Time Systems

7.1 Introduction

Discrete time systems work on digital signals. They are the counterparts of analog systems. The input and output to these systems are digital. Thus, an analog signal that is to be processed with digital signal processors must be converted to digital representation using sampling. Like analog systems, digital systems are linear and non-linear, causal and anticausal, time variant and in-variant, stable and unstable etc.

Digital systems also have an impulse response like analog systems, which is basically the output of the digital system for an input impulse to the system. The output to any other type of input can be calculated with the help of this response.

7.1.1 Time Invariant System

A system is time invariant if for every input $x[n]$ it satisfies the following relationship (see Fig. 7.1)

$$y[n] = w[n] \tag{7.1}$$

7.1.2 Linear System

To test whether a system is linear or not, we perform the following test (shown in Fig. 7.2)

$$u[n] = v[n] \tag{7.2}$$

If the system fails the test, it is not linear.

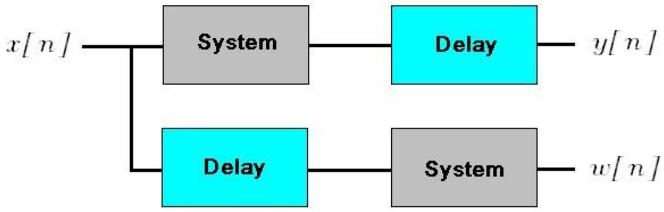

Figure 7.1: Test to check whether the system is time invariant or not. If *y*[*n*] equals *w*[*n*], system is time invariant

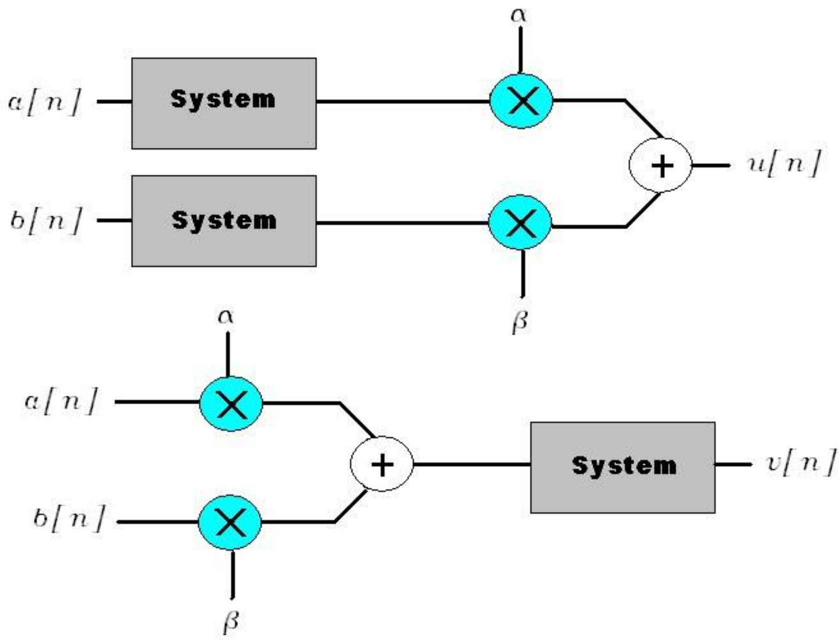

Figure 7.2: Test to check whether the system is linear or not. If *u*[*n*] and *v*[*n*] are equal, then the system is a linear system

7.1.3 Stability of a System

An LTI discrete-time system is" Bounded Input: Bounded Output" (BIBO) stable if its impulse response is absolutely summable. It therefore follows that a necessary condition for an IIR LTI system to be stable is that its impulse response decays to zero as the sample index gets larger. An alternate procedure to find the stability of a digital system is to check whether there is any pole of $H(z)$ whose magnitude is greater than 1. If that's so, the system is unstable.

7.1.4 Moving-Average Filter

Here we present an example of a discrete system that will be utilized in the coming labs. A Moving Average Filter is a filter that is based on the principle of averaging the waveform's adjacent points. The equation of the running (moving) average filter is given by:

$$out[n] = \frac{\sum_{k=0}^{M-1} in[n-k]}{M} \tag{7.3}$$

This equation can be reduced to the form:

$$out[n] = out[n \quad 1] + \frac{in[n] \quad in[n \quad M]}{M} \tag{7.4}$$

We can see that all the coefficients of a moving-average filter are 1/*M*.

7.2 Practical

1. Simulate a Moving average System using *in*[*n*] equal to a sum of two sinusoid of frequency 10 Hz and 200 Hz, sampled at 8000 Hz, and vary *M* from 5 to 10. Plot *out*[*n*]. The screen shot of the expected results is shown in Fig. 7.3.

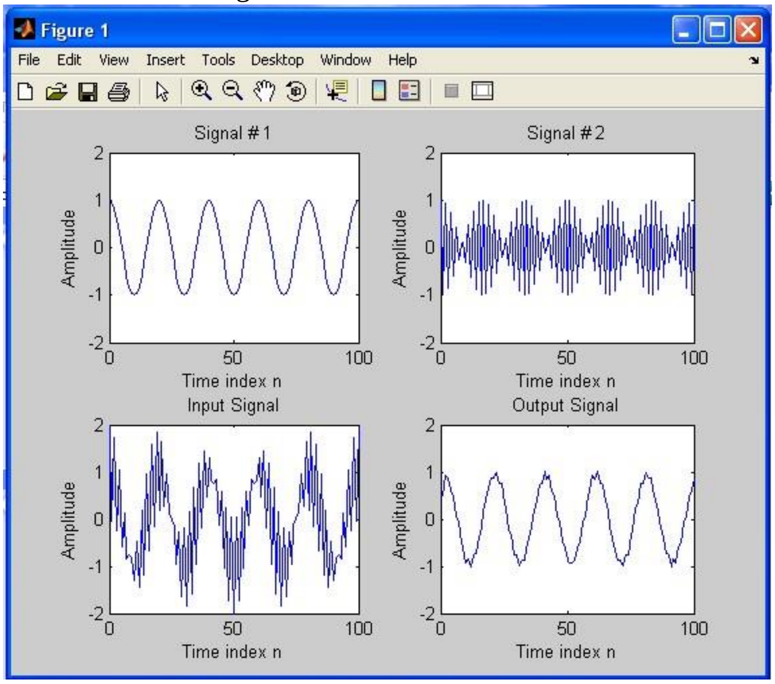

Figure 7.3: The screen shot of a moving averaging filter

2. Find the impulse response of the system

$$y[n] = 2.2403x[n] + 2.4908x[n\text{-}1] + 2.2403x[n\text{-}2] + 0.4y[n\text{-}1] - 0.75y[n\text{-}2] \tag{7.5}$$

Find whether the system given by the above equation is stable, time variant or linear using the first 50 samples of the output. Take *x*[*n*] the sum of two sinusoids of frequency 10 and 20 Hz. Plot *y*[*n*], both in MATLAB and VisualDSP++ 4.5. The screen shots are expected to be as given in Fig. 7.4 and 7.5.

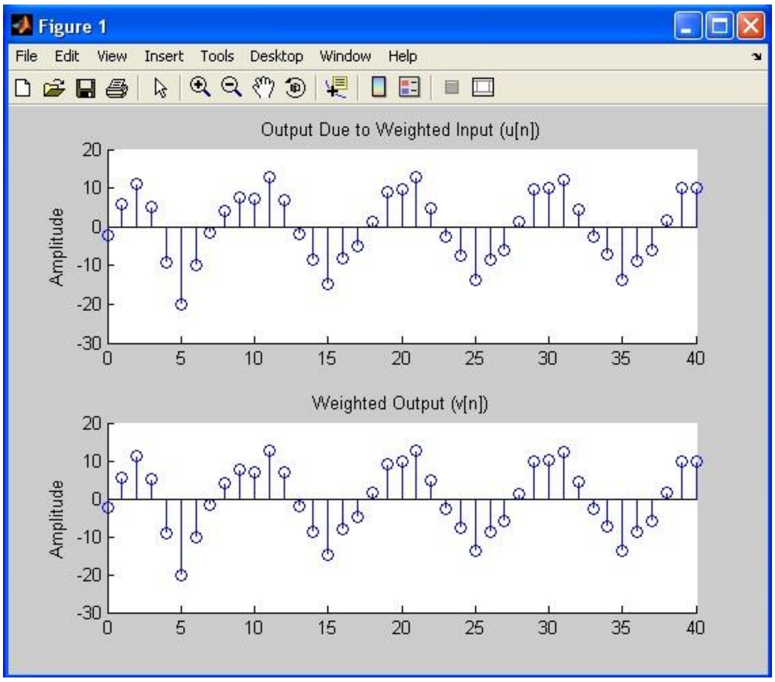

Figure 7.4: The screen shot of results obtained for testing linearity

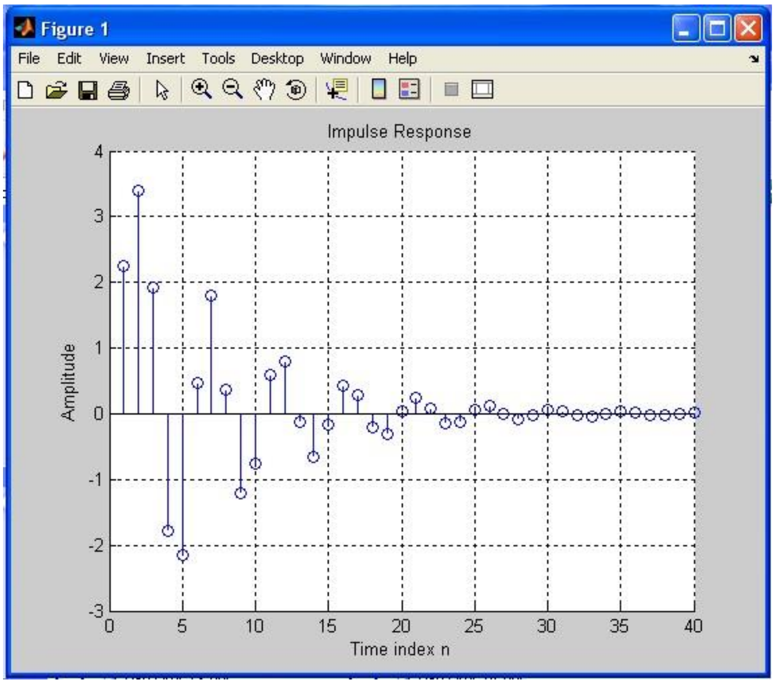

Figure 7.5: The screen shot of the impulse response of the system under test

7.3 Hints

To calculate the response of the moving average filter, use the MATLAB's *filter*() function. See MATLAB's Help for reference. Take the Z-transform of the equation for calculating the impulse response. Use the function *impz*() to calculate the impulse response. Set all initial conditions equal to zero. To determine the stability, take the sum of the impulse response in time domain and find whether the sum is getting stable or increasing. Or use the MATLAB's *root*() function.

7.4 Questions

- Replace *in*[*n*] in the practical by a sinusoid of frequency 15 Hz and contaminated it with noise. Perform the moving average. What value of M removes most of the noise? What is the drawback of the moving-averaging filter?
- What is the value of M which must not be taken for a moving average in the above question?
- Change *x*[*n*] to sum of three sinusoids with the third of frequency 30 Hz and repeat the procedure. Does the properties of the system change?
- Generate a unit ramp function for 0.2 sec sampled at 4000 Hz in MATLAB. Using convolution, obtain the output of a system with time domain system response:

  $$h[n] = 2\delta[n] + 3\delta[n-3]$$

  Perform the procedure in VisualDSP++ 4.5.
- Using VisualDSP in-built functions, convolve x and y. Both are integer arrays of 256 values. Verify the results by plotting, in both MATLAB and VisualDSP.