Experiment 7

M Naveen Kumar
16EE230

1 Aim of the experiment

2 Results/Graphs

2.1 Question 1

User defined function to implement Direct Form II realization of an IIR system.

Code : task1.m

The code was successfully written and used for Questions 3 to 6.

2.2 Question 2

Generate a digital band-pass elliptical filter.

Code : task2.m

The following code was successfully written and used for Question 3.

```
pkg load signal
 N=5;
[B,A]=ellip(N,-20*log10(.99),20,[.3  .4]);
```

2.3 Question 3

Code : task3.m

a) Graph

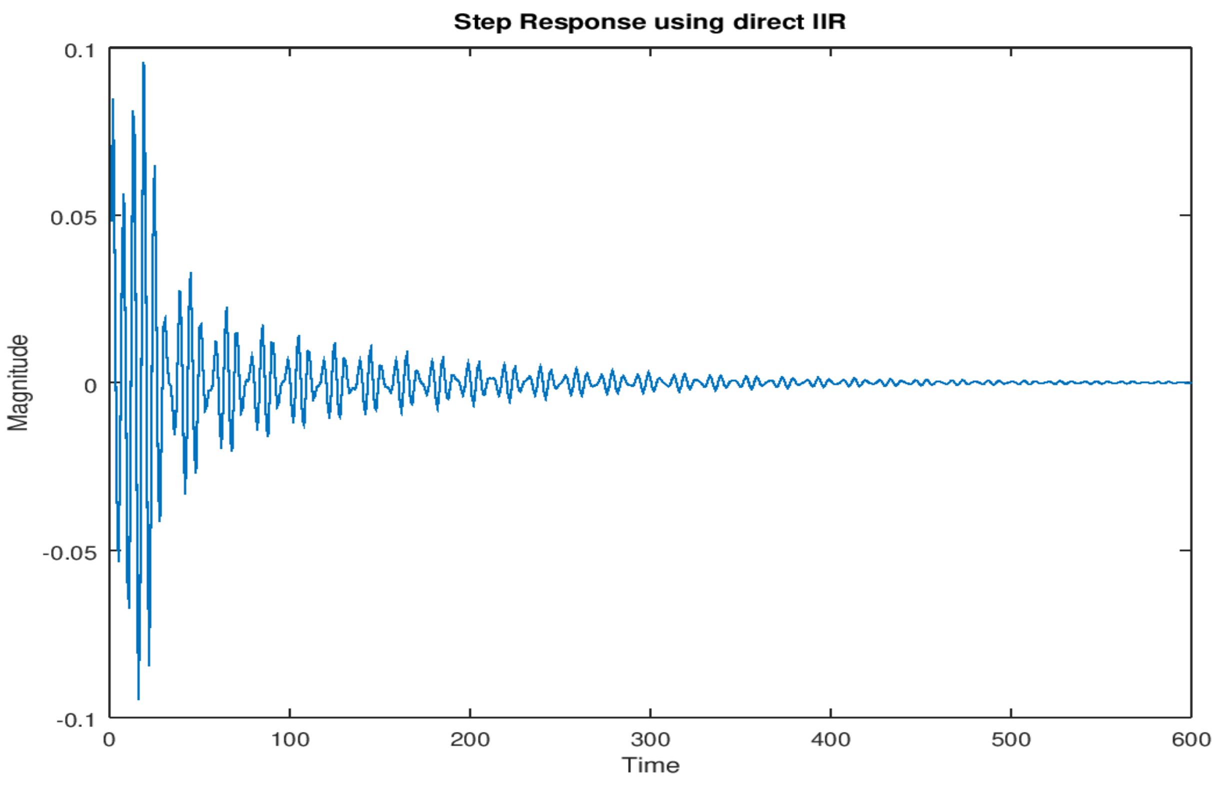

Figure 1: Step Response using direct IIR

b) Graph

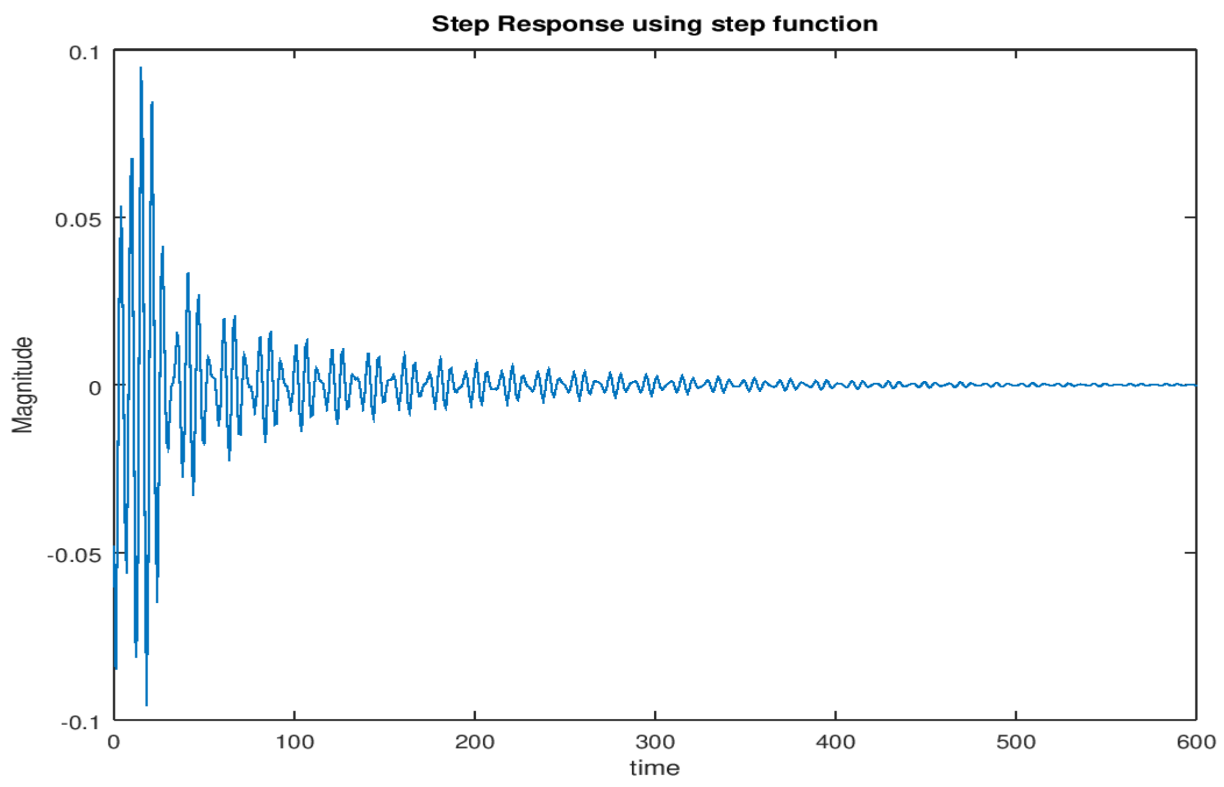

Figure 2: Step Response using step function

Observations

- A function was written to implement Direct Form II realization of an IIR system given the pseudo code.
- From the plots,we observe that the output of the DirectIIR function for step input and the step response of the inbuilt function are same. The direct form II realization of the filter produced same results.
- The magnitude of the step response seems to be decaying to 0.

Conclusion

- An IIR filter has been implemented using direct form II. This method uses less memory locations.
- The DirectIIR function took 0.11121 seconds whereas the inbuilt function took 0.035141 seconds on a Core i7 8th gen processor with the latest version of octave which implies that the inbuilt function is faster and more efficient way of calculating the step response.
- For greater number of samples time will increase proportionately.

2.4 Question 4

Use tf2sos() function from Octave which converts a system into a cascade of second order system and use this to perform a cascade implementation of the elliptical filter.

Code : task4to6.m

The code was successfully written and the tf2sos() function was explored which gives an (N+1)/2 * 6 matrix where the 6 elements corresponds to the coeffecients of the numerator's and denominator's second order cascaded tranfer functions of the system and N is the order of the filter.

2.5 Question 5

Use the Direct Form II function to calculate the step response of the filter using a cascade of equations and compare it with the responses obtained from Question 2.

Code : task4to6.m

a) Graph

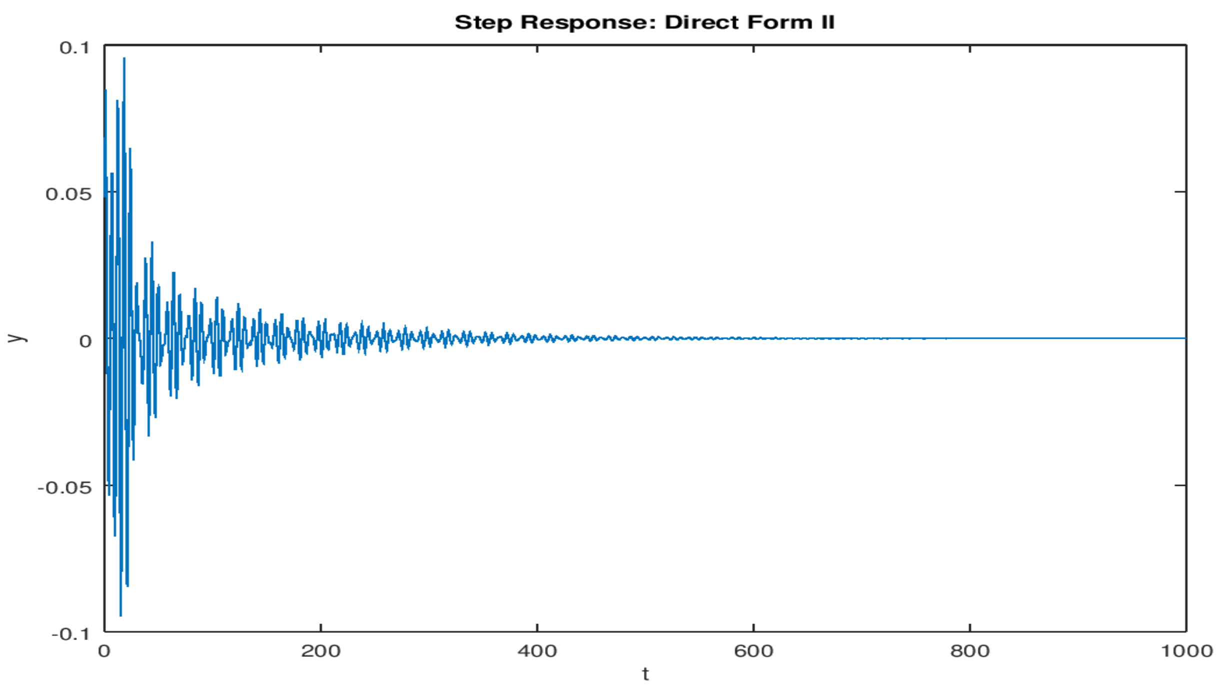

Figure 3: Step Response using direct IIR

b) Graph

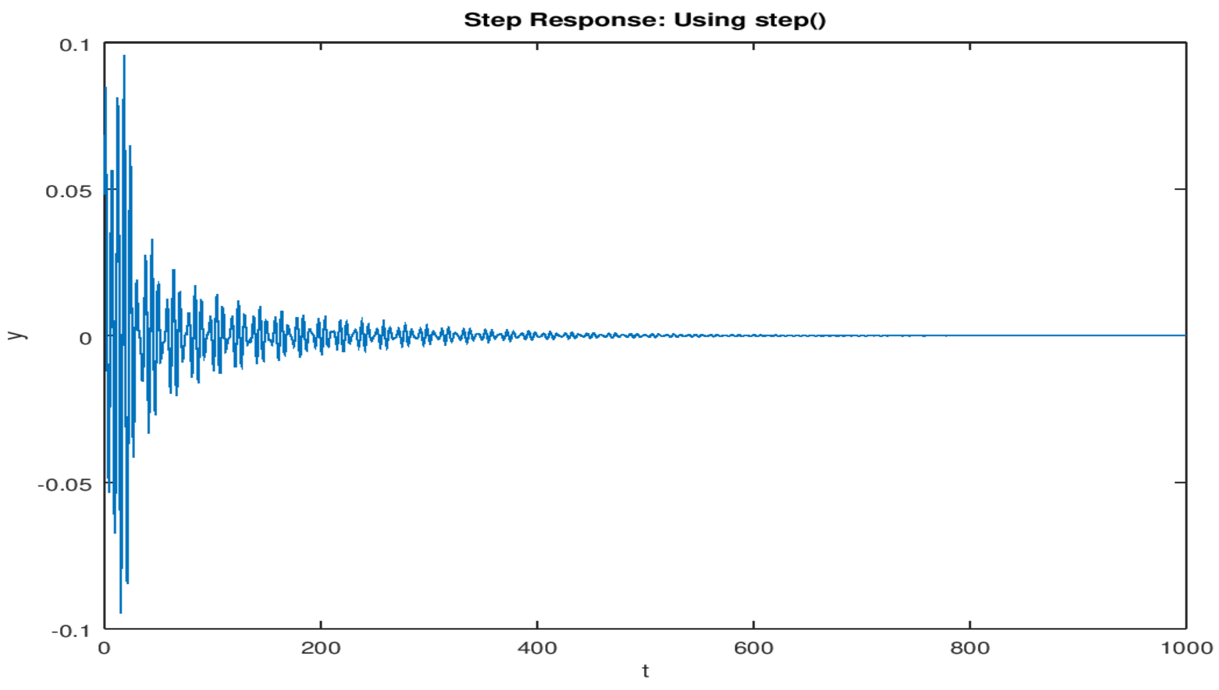

Figure 4: Step Response using step function

c) Graph

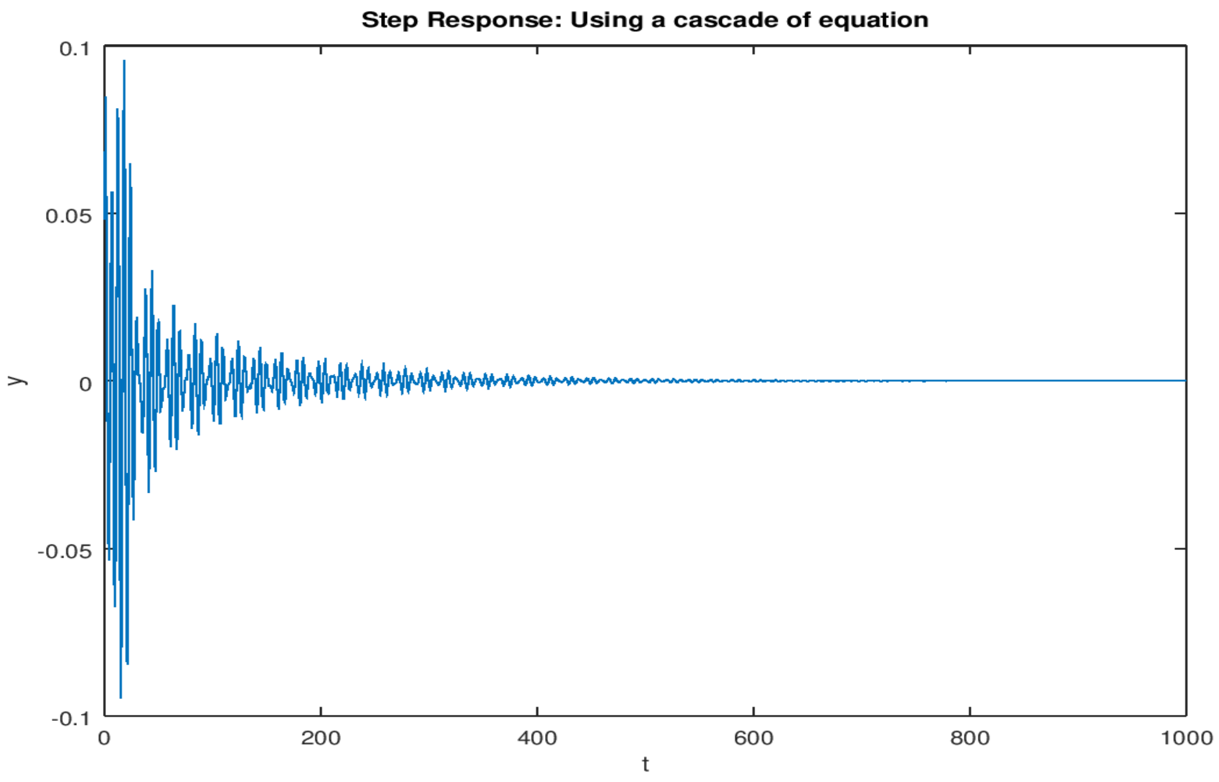

Figure 5: Step Response using cascaded functions obtained from tf2sos()

Observations

- The figure 3 is the one obtained from the cascade of equations and as seen in question 2 and again over here the first 2 plots are same and so is the one obtained with the cascaded equations.
- The respective time taken for each functions are :

  Direct Form II implementation: 0.189958s
  Inbuilt step function: 0.0346129s
  Cascaded form: 0.240696s
- The above time taken varies with ones obtained in question 2 due to different implementations and time vector's size (was 600 in Qn 2 and 1001 over here).
- The inbuilt step function was the fastest. Cascaded form took the most time among the three forms.

Conclusion

- The cascaded realization is more general and more flexible than the parallel realization. Therefore it is more preferred in most applications of IIR filters.
- All the three implementations gave the same results but the inbuilt function was the fastest. Thus, for applications where time is a constraint the inbuilt function without the cascaded form is preferred.

2.6 Question 6

Exploring tic() and toc() functions to find time taken.

Code : task4to6.m and task3.m

The tic() and toc() functions were successfully used to find the computation time by the computer and compare the speed of various methods.