Experiment 3B: FIR Filter Implementation Using TMS320C5515

Introduction

In signal processing, filters are used to filter out a group of frequencies while allowing the rest of the frequencies.

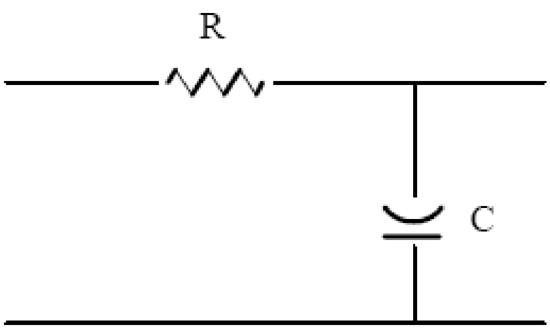


Figure 1: RC Circuit

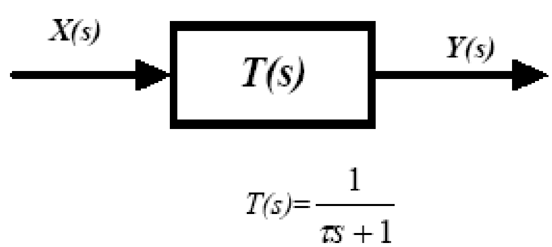


Figure 2: Block Diagram

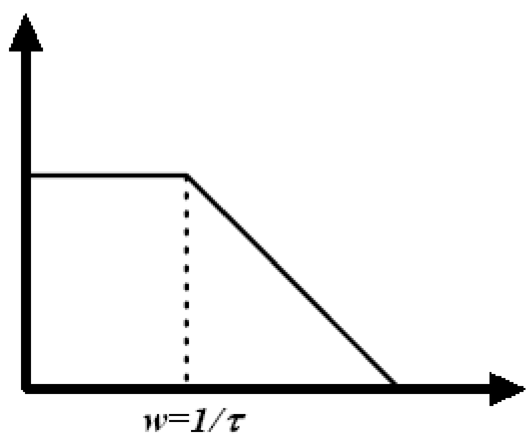


Figure 3: Bode Diagram

The analog circuit shown in Fig. 1 is effected by the physical environment by time and starts not to respond in the expected way.

Based on their behaviors, filters can be divided into four categories:

- Low-pass
- High-pass
- Band-pass
- Notch filter

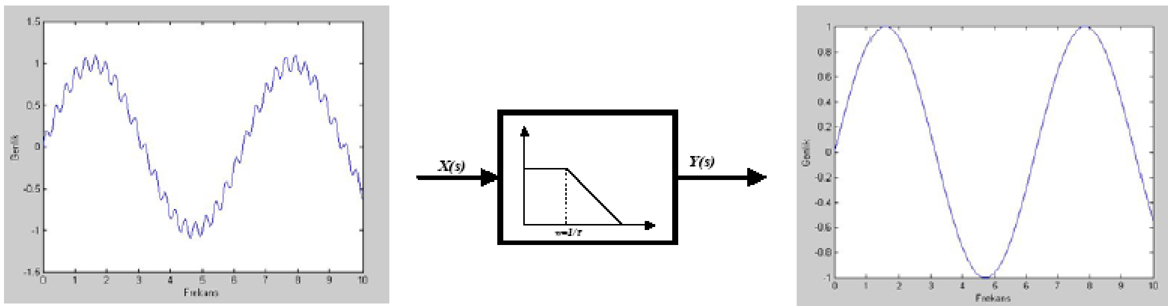


Figure 4: A sample low-pass filter

In Fig. 4, *y=sin(x)+0.1sin(20x)* is given as the input to the low-pass filter block. This is a sinusoidal signal containing noise with a high frequency. The signal holding the noise is filtered by the low-pass filter in order to obtain the desired original signal without any noise. Here determining the value of the cut-off frequency becomes a very important point.

Digital filtering is applied to signals instead of analog filters due to the fact that the filters that make use of analog circuit components are not stable. This approach provides quite good results to the designers. The main difference between the digital filter designs and analog designs is the operation domain. Analog filters operate in frequency domain while digital filters operate in time domain. Based on their structure, filters can be designed in two ways: IIR (Infinite Impulse Response) and FIR (Finite Impulse Response).

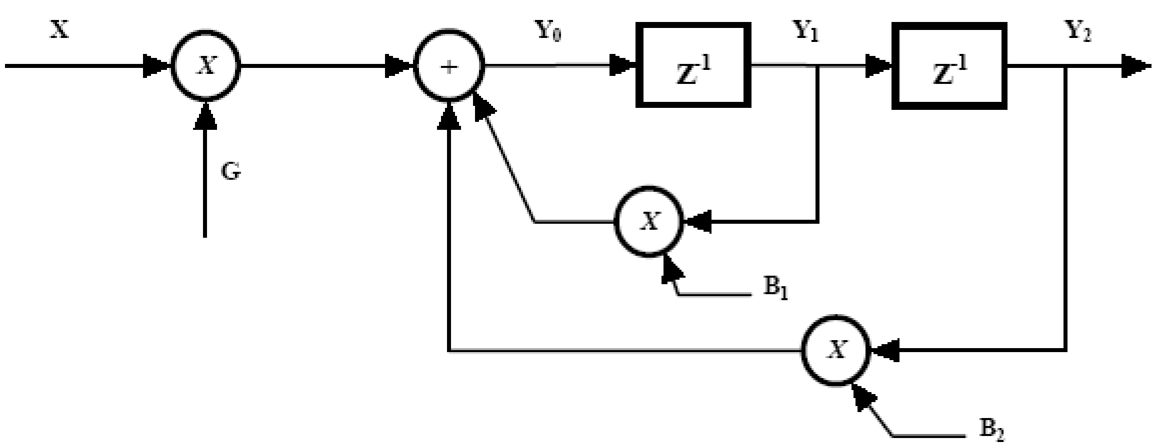


Figure 5: A discretized block diagram of an IIR filter

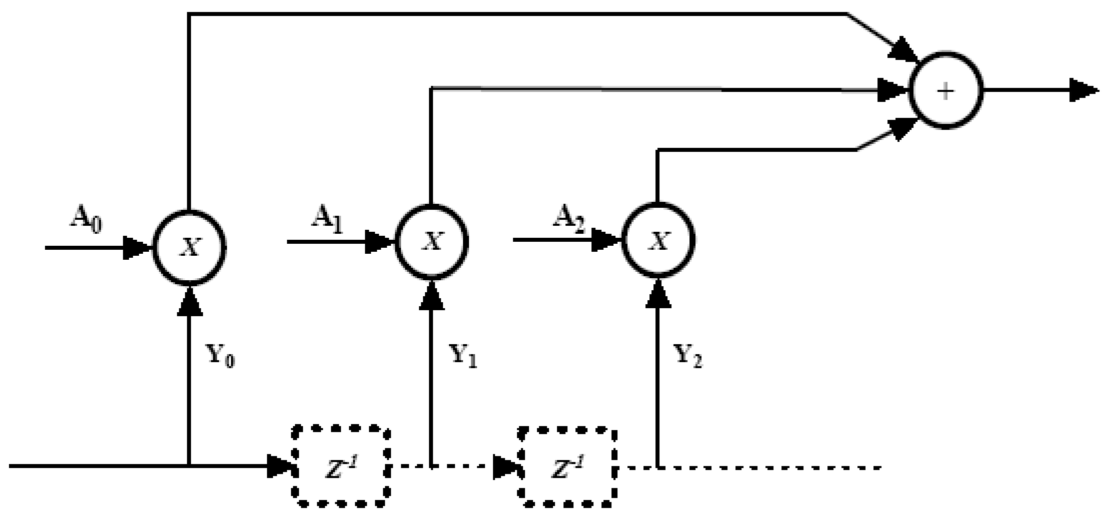


Figure 6: A discretized block diagram of an FIR filter

Unit impulse responses of a third order FIR and IIR filter are shown below..

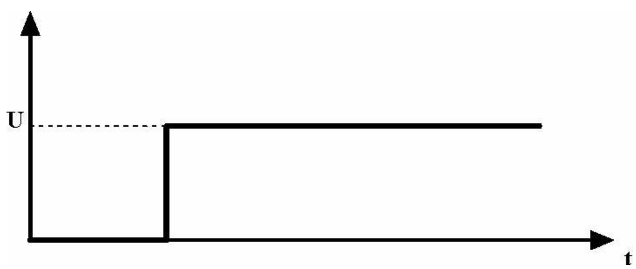
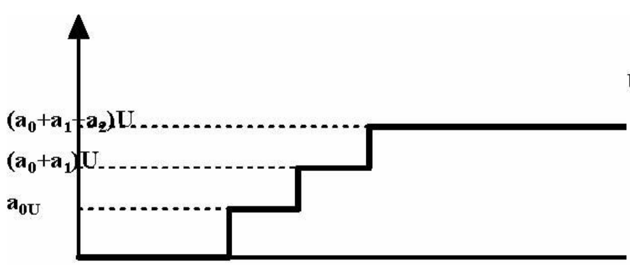


Figure 7: Applied unit impulse function



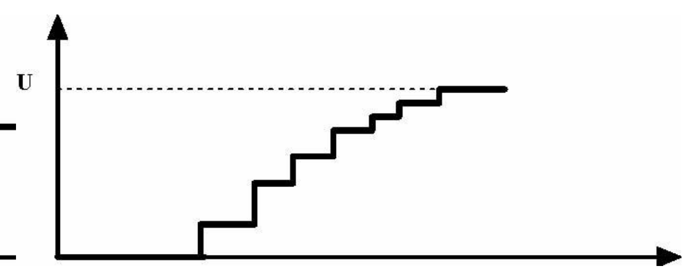


Figure 8: Impulse response of a third order FIR filter

Figure 9 : Impulse response of a third order IIR filter

Before the Experiment

1. Review IIR and FIR filters, Bode diagrams.
2. Analze the Bode digrams of the first and second order FIR filters and their equivalents in s domain.

Experiment 3B.1

Draw the Bode diagram of a first order low-pass FIR filter. Find the equivalence of the filter in s domain. Transform your filter to z domain considering the cut-off frequency of 10 KHz. Draw the block diagram of your filter.

Experiment 3B.2

In this part of the experiment, you are asked to implement the filter you designed in Experiment 3B.1 on TMS320C5515 DSP kit. In order to do that, you can modify the sample code snippets given below. Note that the sample code snippets are for a FIR filter with three coefficients. You need to make the necessary changes on the code snippet to implement your filter on the sample project.

```
/* ------------------------------------------------------------------------ *
 *                                                                          *
 *  aic3204_test( )                                                           *
 *                                                                          *
 * ------------------------------------------------------------------------ */
Int16 aic3204_test( )
{
     SYS_EXBUSSEL = 0x6100;         // Enable I2C bus
     USBSTK5515_I2C_init( );        // Initialize I2C

    /* Codec tests */
    printf( " -> 1 KHz Tone on Headphone (J4).\n" );
    if ( aic3204_tone_headphone( ) )
        return 1;

    USBSTK5515_wait( 100 );  // Wait

    printf( "<-> Audio Loopback from Stereo IN 1 (J3) --> to HP (J4)\n" );
    if ( aic3204_loop_stereo_in1( ) )
        return 1;

    return 0;
}
```

aic3204_loop_stereo_in1 function: Here, input can be obtained either from left or right channels. It is obtained from the left channel within the sample code. On the other hand, output is being sent to both of the channels.

```
Int16 input_16, output_16;
float input, output;
float x[] = {0,0,0};
float coef[] = {…,…,…}; // FIR filter katsayıları


Previous Lines are skipped...

    /* Play Tone */

     for (;;){

             /* Read Digital audio */
             while((Rcv & I2S0_IR) == 0);
             // Wait for interrupt pending flag

              input_16= I2S0_W0_MSW_R;  // 16 bit left channel received audio data

              data1 = I2S0_W0_LSW_R;

              data4 = I2S0_W1_MSW_R;  // 16 bit right channel received audio data
              data2 = I2S0_W1_LSW_R;

              /* Write Digital audio */
              while((Xmit & I2S0_IR) == 0);
              // Wait for interrupt pending flag


              input=float(input_16/pow(2,16));
              x[2]=x[1];
              x[1]=x[0];
              x[0]=input;

              output= x[0]*coef[0]+ x[1]*coef[1]+ x[2]*coef[2];
              output_16 = output*pow(2,16);

              I2S0_W0_MSW_W = output_16;//16 bit left channel transmit audio data
              I2S0_W0_LSW_W = 0;

              I2S0_W1_MSW_W = output_16;//16 bit right channel transmit audio data
              I2S0_W1_LSW_W = 0;

    }

    /* Disble I2S */
    I2S0_CR = 0x00;
```

Experiment 3B.3

Modify your filter implementation for a first order high-pass and a first order band-pass FIR filter.

Report

1. Include all necessary information including your graphics, observations etc. to the report.