Istanbul Technical University
Faculty of Computer and Informatics

BLG438E Digital Signal Processing Lab
Experiment 4

Cem Yusuf Aydoğdu
150120251

First Part: Matlab

Initially, different types of signals was produced by a IIR filter in Matlab. Following simulink models were constructed in order to produce these signals.

Equation of the filter is given as:

$$Y(Z) \ = \ AY(Z)Z^{-1} \ + \ BY(Z)Z^{-2} \ + \ CX(Z)Z^{-1}$$

Solving the equation results:

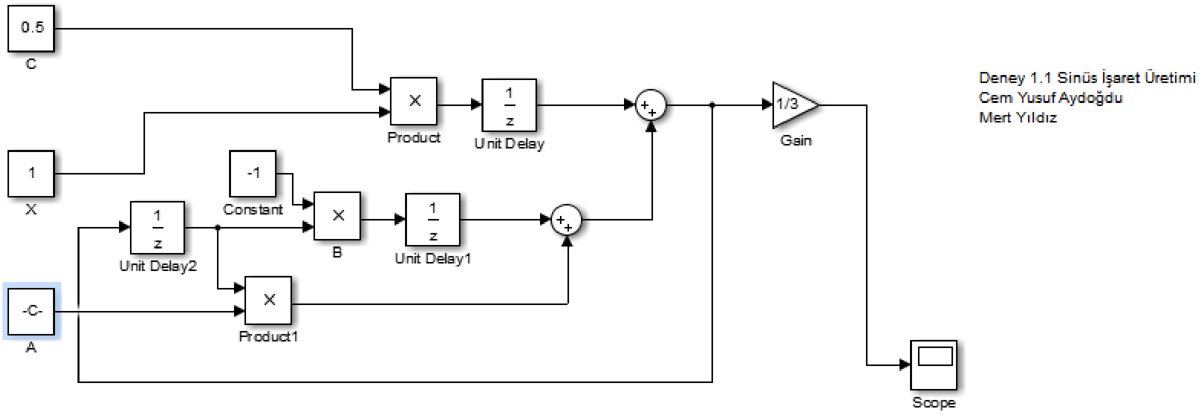
$$A = 2cos(\theta)$$

$$B = -1$$

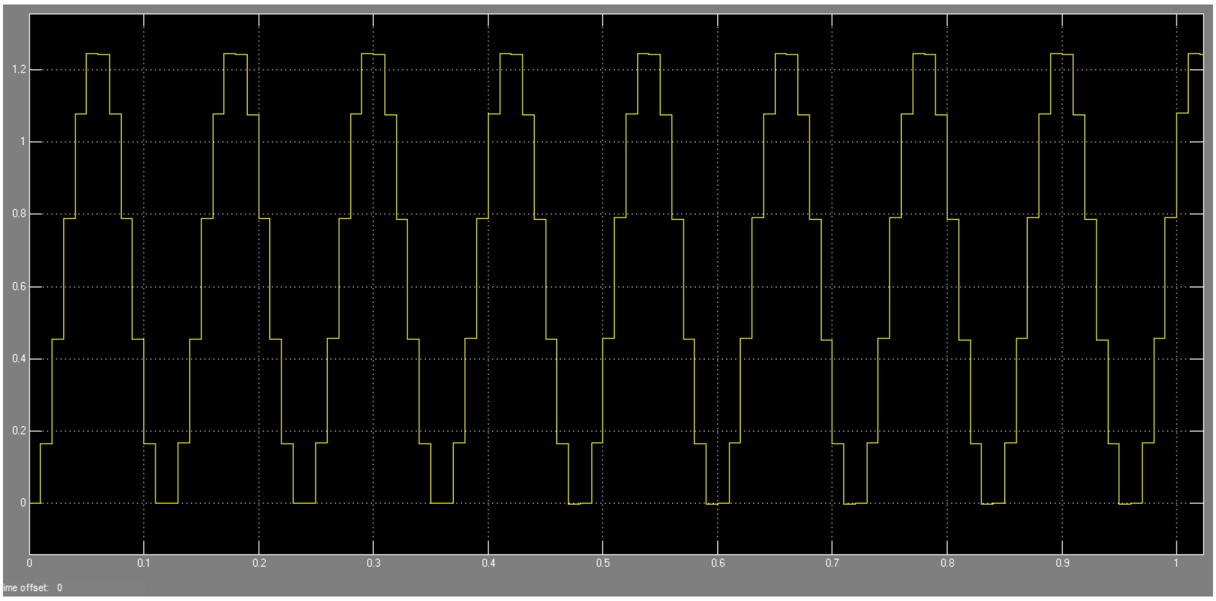
$$C = sin(\theta)$$

$$\theta = \ (f_{wanted}/f_{sampled}) * 360$$

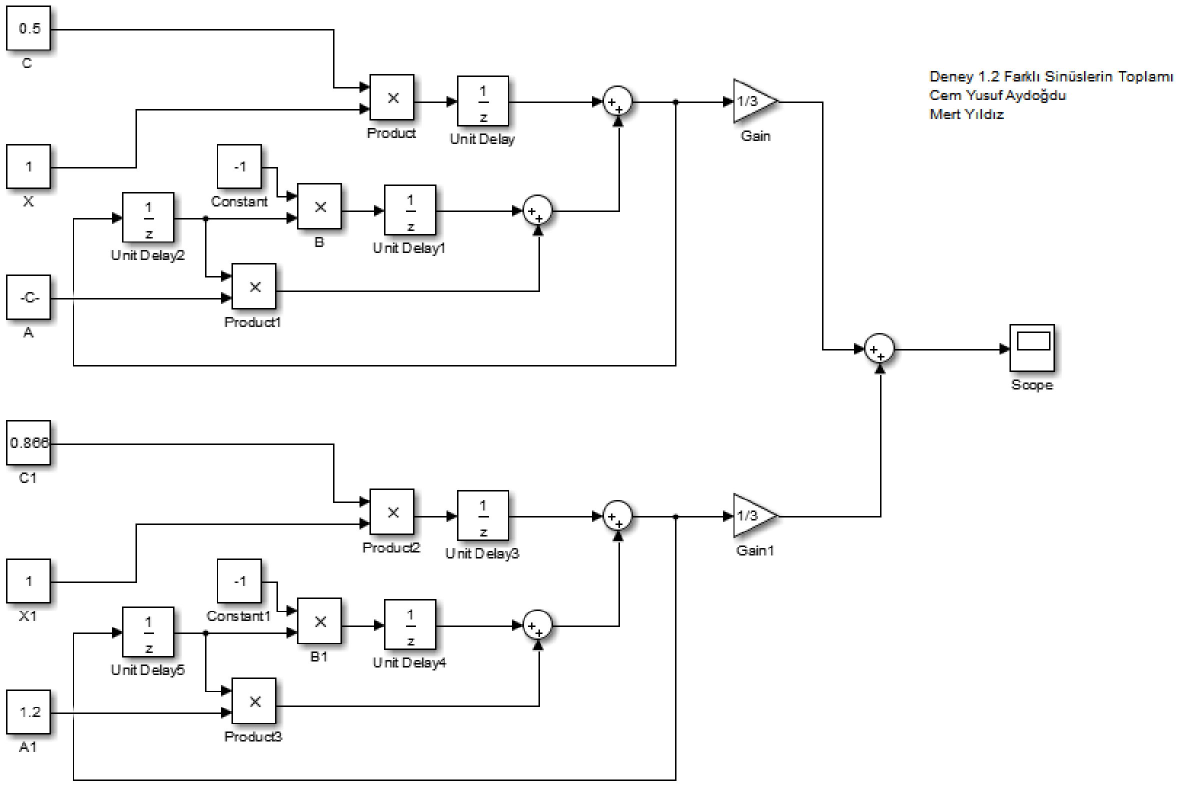
For sinusoidal wave, θ was selected as 30, which results A=0.5 and C=1.7320



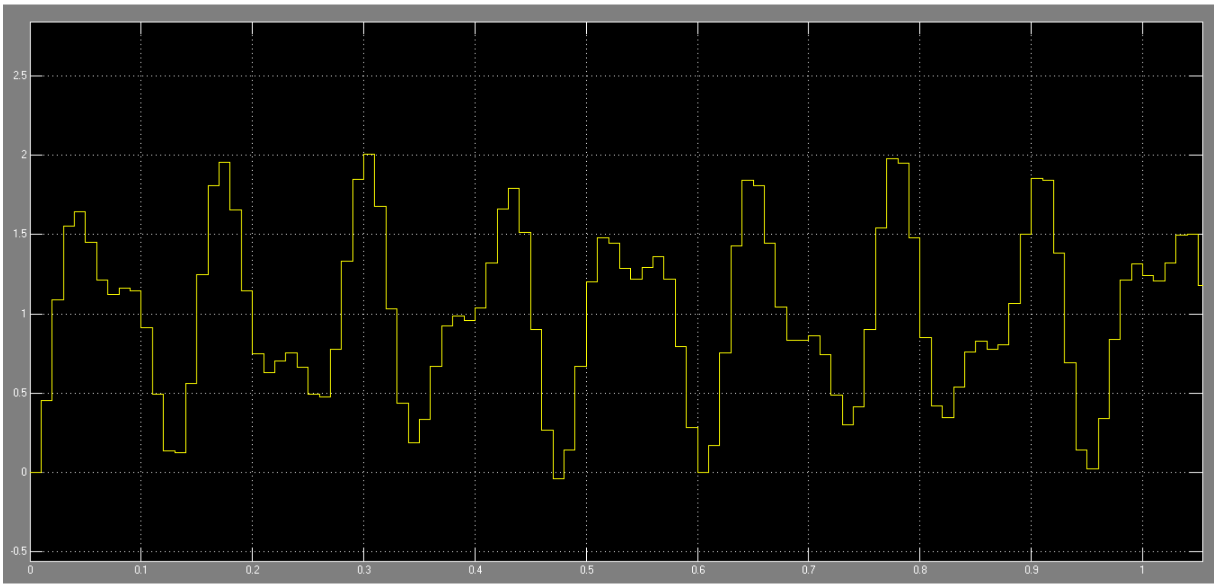
Running the simulation for with start time=0.0, stop time=10.0, step size=0.01



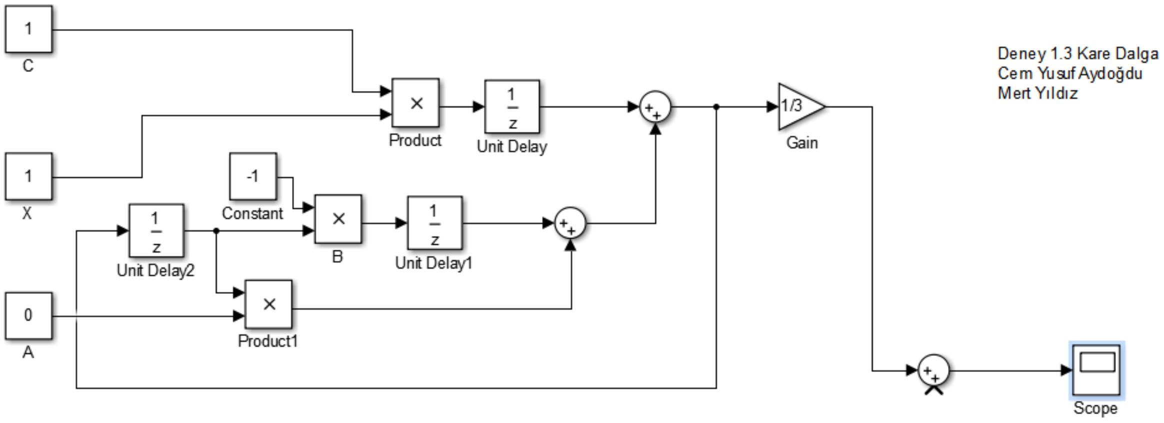
Summing two different sinusoidal functions. In the first function, coefficients are A=0.5 and C=1.7320, in the second function A=1.2 and C=0.866:



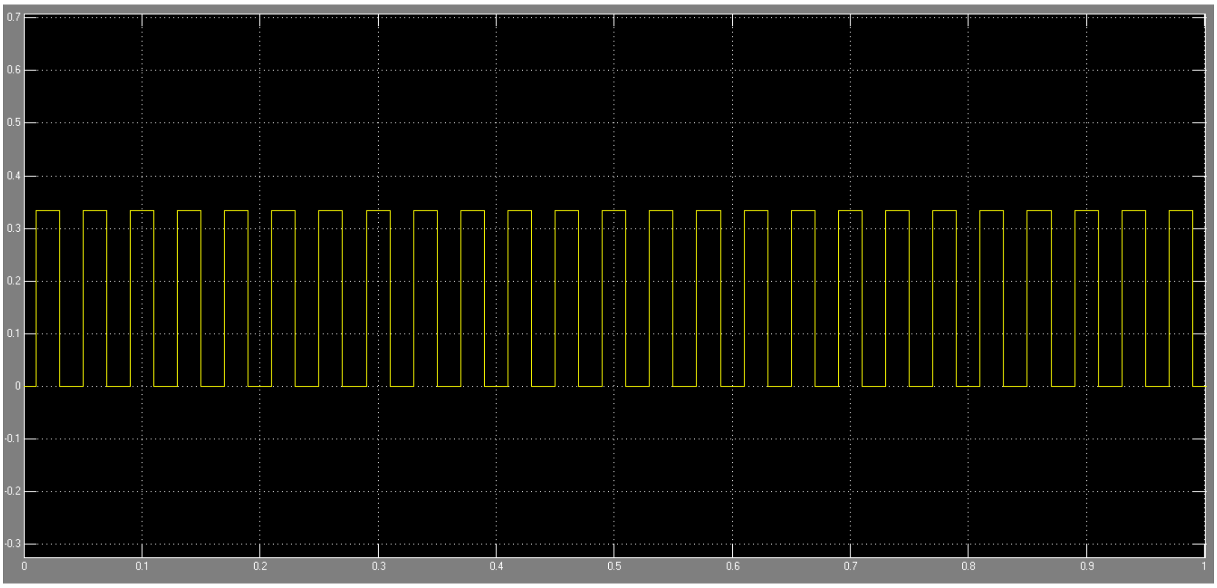
Running the simulation for with start time=0.0, stop time=10.0, step size=0.01:



In order to produce square waves, A=0 and C=1 selected.



Running the simulation for with start time=0.0, stop time=10.0, step size=0.01:



Second Part: C5515 Kit

In the second part, code for the transfer function was written. This code runs in a sample interrupt function with 1kHz.

For the sinusoidal wave. First, coefficients for transfer function were initialized as float. x[0] holds an initial value as input, x[1] holds $X(Z)Z^{-1}$. Similarly, y[0] holds $Y(Z)$, y[1] holds $Y(Z)Z^{-1}$ and y[2] holds $Y(Z)Z^{-2}$.

Unit delays were calculated and multiplied with coefficients and the transfer function was calculated. After that, output of the transfer function was multiplied with 10000 in order to preserve the data and it was sent to the output as Int16 data type.

```
float A=1.7320, B=-1, C=0.5;

float x[2] = {1.0 ,0.0};               // x, xz^-1
float y[3] = {0.0, 0.0, 0.0};          // y, yz^-1, yz^-2

while(1)
{
      while((XmitR & I2S0_IR) == 0);//Wait for transmit interrupt to be pending
      y[0] = A*y[1] + B*y[2] + C*x[1];

      y[2]=y[1];          //yz^-2
      y[1]=y[0];          //yz^-1

      x[1]=x[0];          //xz^-1

      //write to output
      I2S0_W0_MSW_W = ((Int16)((10000*y[0]))) ; // 16 bit left channel
      I2S0_W1_MSW_W = ((Int16)((10000*y[0]))) ; // 16 bit right channel
}
```

For the square wave. Only coefficients are different:

```
float A=0, B=-1, C=1;

float x[2] = {1.0 ,0.0};               // x, xz^-1
float y[3] = {0.0, 0.0, 0.0};          // y, yz^-1, yz^-2

while(1)
{
      while((XmitR & I2S0_IR) == 0);//Wait for transmit interrupt to be pending
      y[0] = A*y[1] + B*y[2] + C*x[1];

      y[2]=y[1];          //yz^-2
      y[1]=y[0];          //yz^-1

      x[1]=x[0];          //xz^-1

      //write to output
      I2S0_W0_MSW_W = ((Int16)((10000*y[0]))) ; // 16 bit left channel
      I2S0_W1_MSW_W = ((Int16)((10000*y[0]))) ; // 16 bit right channel
}
```