ECE 5630: Programming #2

Due on Tuesday, November 24, 2014

Scott Budge 3:00pm

Tyler Travis A01519795

Contents

Problem 1 **3**
(a) . . . 3
(b) . . . 3

Problem 2 **4**
(a) . . . 5
(b) . . . 5

Problem 1

(a)

Figure 1 shows the Impulse response of the filter h[n].

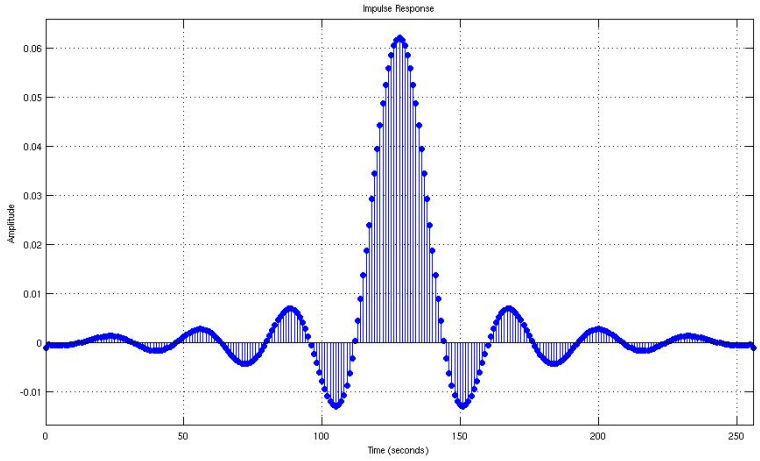


Figure 1: Impulse Response

(b)

Figure 2 shows the Magnitude and Phase response of the filter h[n].

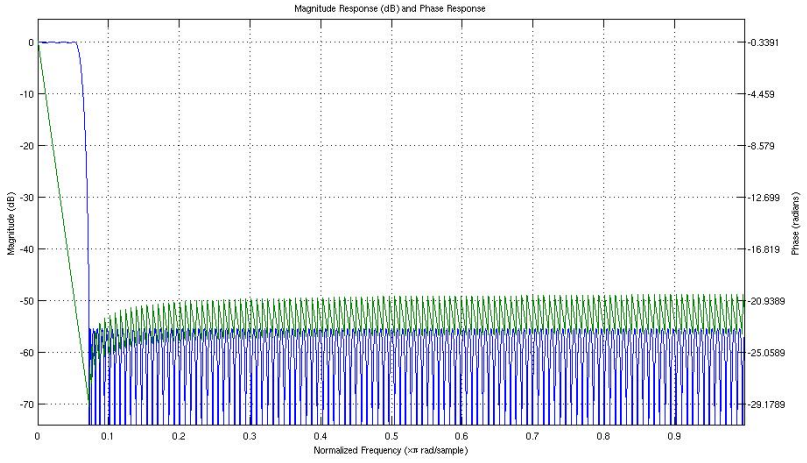


Figure 2: Impulse Response

Problem 2

Listing 1 shows the first program.

Listing 1: Program 1 - part1.cpp

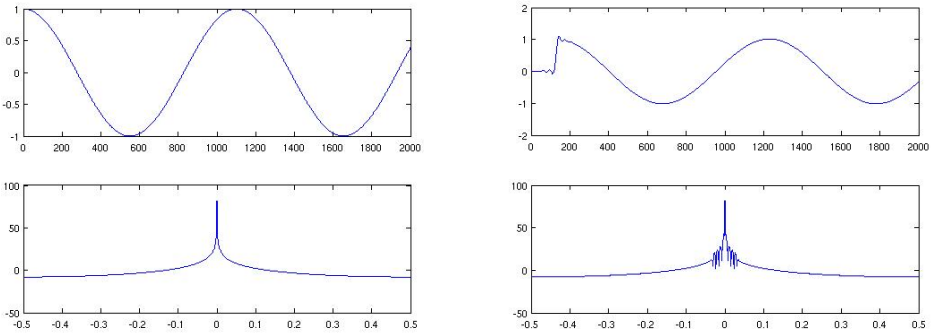
```
#include <iostream>
#include <fstream>
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include "../includes/fft842.c"

// Filter Length
#define Nf 256
// Length of Signal
#define N 25600
// Sampling frequency
const double Fs = 11025;

int main(int argc, char** argv)
{
  // Input stream for filter
  std::ifstream filterIn("../data/LowPassFilter.dat");

  // filter of length Nf = 256
  double h[Nf];

  // input vairable
  double in;

  // Read in the filter data
  for(int n = 0; n < Nf; ++n)
  {
    filterIn >> in;
    h[n] = in;
  }

  // Output streams for the input x signal
  // and the output y signal
  std::ofstream x_dat("../data/x.dat");
  std::ofstream y_dat("../data/y.dat");

  // input x signal of length N = 25600
  double x[N];

  // output y signal of Length N = 25600
  double y[N];

  // f0 = f/Fs
  // Normalized frequency
  double f = atof(argv[1]);
  double f0 = f/Fs;
```

```
50    // Generate input signal x[n]
      for(int n = 0; n < N; ++n)
      {
        x[n] = cos(2*M_PI*f0*n);
        x_dat << x[n] << std::endl;
55    }

      double temp;
      for(int n = 0; n < N; ++n)
      {
60      temp = 0;
        for(int k = 0; k < Nf; ++k)
        {
          temp += x[n-k]*h[k];
        }
65      y[n] = temp;
        y_dat << y[n] << std::endl;
      }

      return 0;
70  }
```

(a)

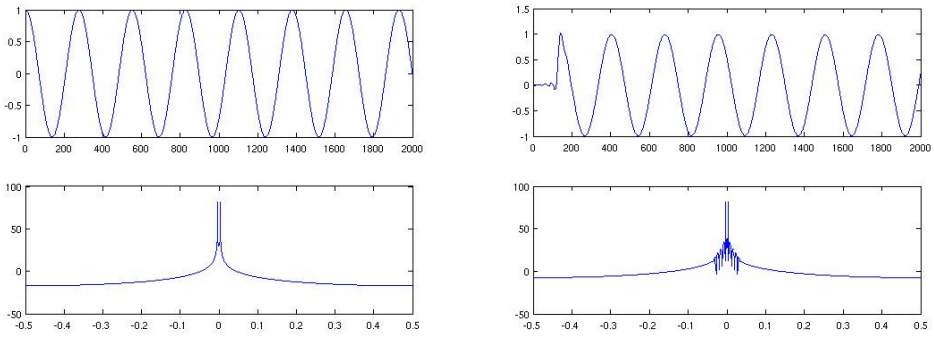
The number of multiples is

(b)



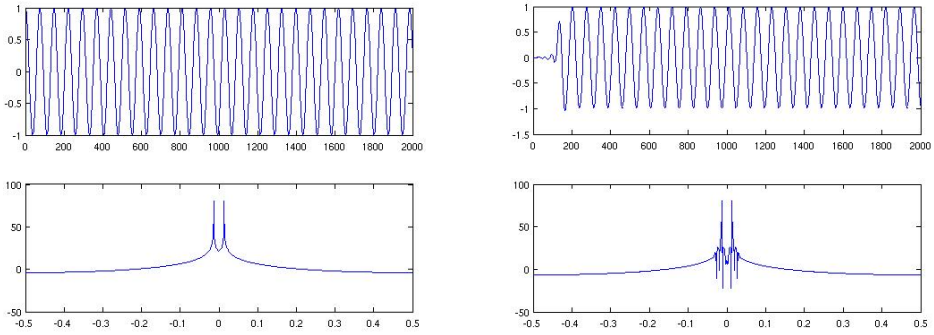
(a) x[n] (b) y[n]

Figure 3: Input(a) and Output(b) with f = 10Hz



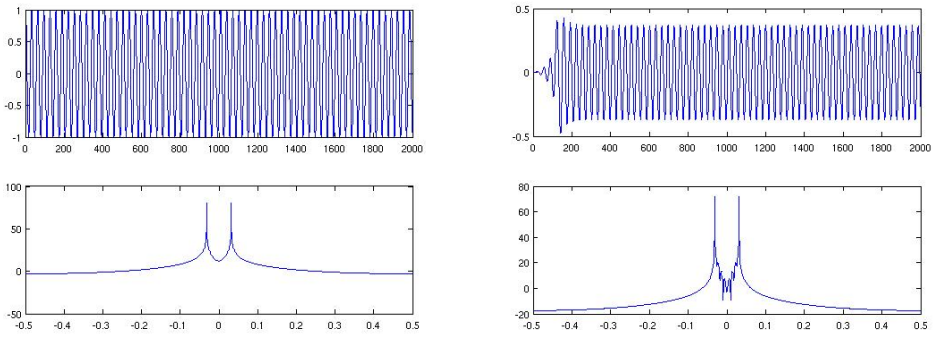
(a) x[n] (b) y[n]

Figure 4: Input(a) and Output(b) with f = 40Hz



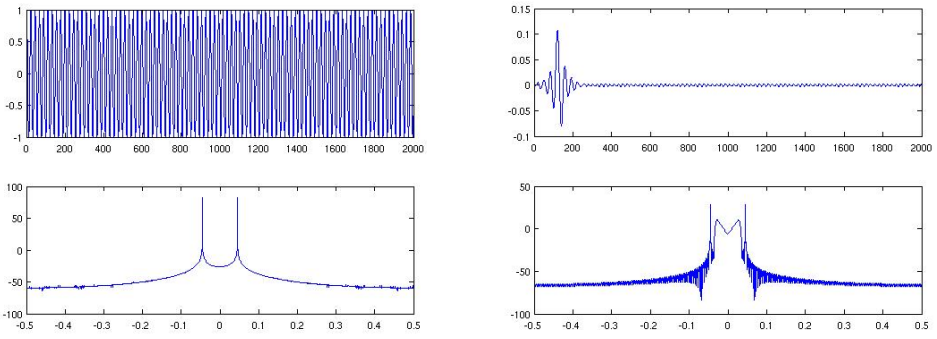
(a) x[n] (b) y[n]

Figure 5: Input(a) and Output(b) with f = 150Hz



(a) x[n] (b) y[n]

Figure 6: Input(a) and Output(b) with f = 350Hz



(a) x[n] (b) y[n]

Figure 7: Input(a) and Output(b) with f = 500Hz