

Digital Signal Processing: Assignment #2

Due on Thursday, March 27, 2014

Prof. Lee

Kevin Zhao, E14982022

Contents

Problem 1	3
a. $h(t)$	3
b. $g(t)$	3
Problem 2	3
a.	3
b.	3
Problem 3	4
a.	4
b.	5
$T \times f_0 = 10, T = 5, f_0 = 2$	5
$T \times f_0 = 100, T = 5, f_0 = 20$	6
$T \times f_0 = 1000, T = 20, f_0 = 50$	6
c.	7
Problem 4	7
a.	8
b.	8
Appendix Code	8

Problem 1

Impulse function and its Fourier Transform

a. $h(t)$

We could describe the function $h(t)$ as

$$h(t) = \sum_{k=-\infty}^{\infty} \delta(t - nT) \times (-1)^n$$

The Fourier Transform could be determined intuitively by shifting

$$H(f) = f_0 \sum_{k=-\infty}^{\infty} (-1)^n \delta(f - nf_0)$$

b. $g(t)$

We could describe the function $g(t)$ as

$$g(t) = \sum_{k=-\infty}^{\infty} [(-1)^n \delta(t - 2nT) + \delta(t - (2n + 1)T)]$$

The Fourier Transform could be determined intuitively by shifting

$$G(f) = \frac{1}{2T} \sum_{k=-\infty}^{\infty} (-1)^n \delta(f - (2n + 1)\frac{1}{T})$$

Problem 2

Convolution and its theory

a.

The convolution could be determined as

$$\begin{aligned} x(t) * [\delta(t + \frac{T_0}{2}) + \delta(t - \frac{T_0}{2})] \\ = x(t - \frac{T_0}{2}) + x(t + \frac{T_0}{2}) \end{aligned}$$

b.

The convolution could be determined as

$$\begin{aligned} x(t) * [\delta(t + \frac{T_0}{4}) + \delta(t - \frac{T_0}{2})] \\ = x(t + \frac{T_0}{4}) + x(t - \frac{T_0}{2}) \end{aligned}$$

Problem 3

a.

The Fourier Transform is

$$\begin{aligned}\mathcal{F}\{h(t)\} &= \int_{-\infty}^{\infty} h(t)e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} A \cos(2\pi f_0 t)e^{-j2\pi ft} dt\end{aligned}$$

Then apply integral by part, we could derive that

$$\int e^{ax} \cos(bx) dx = \frac{e^{ax}(a \cos(bx) + b \sin(bx))}{a^2 + b^2}$$

Compare the coefficients

$$a = -j2\pi f$$

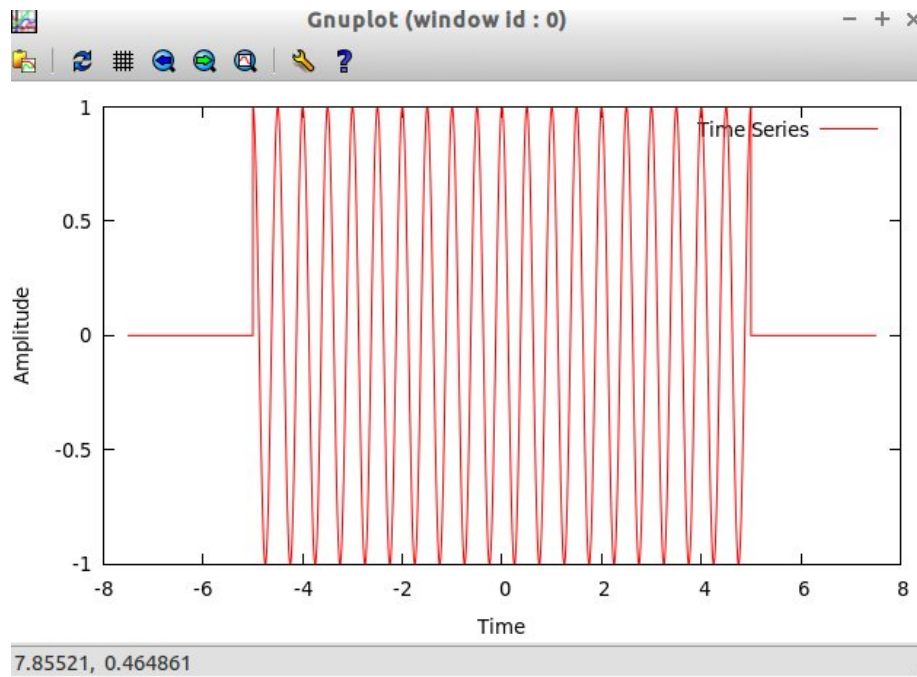
$$b = 2\pi f_0$$

Then frequency-shifting theory tells that exponential function in time domain implies f_0 shifting in frequency domain.

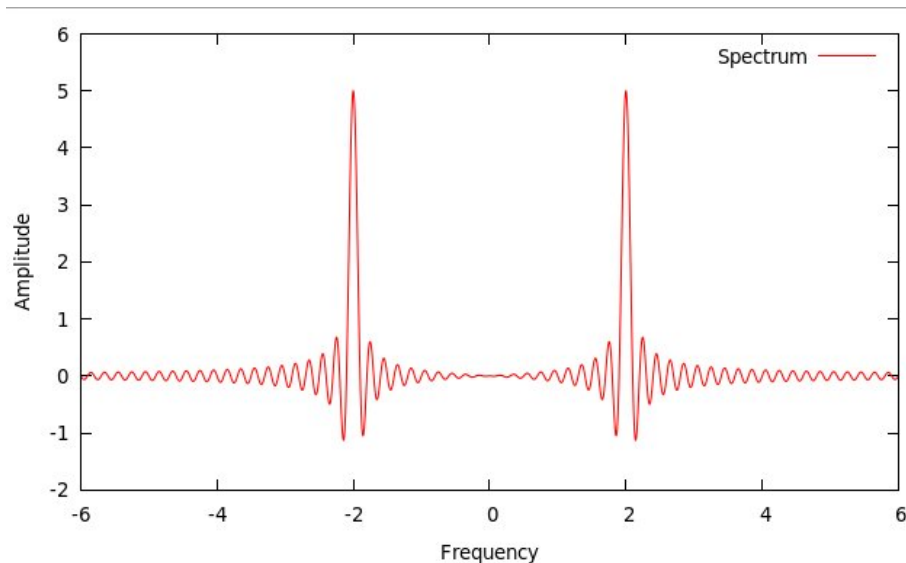
b.

Plot the results where

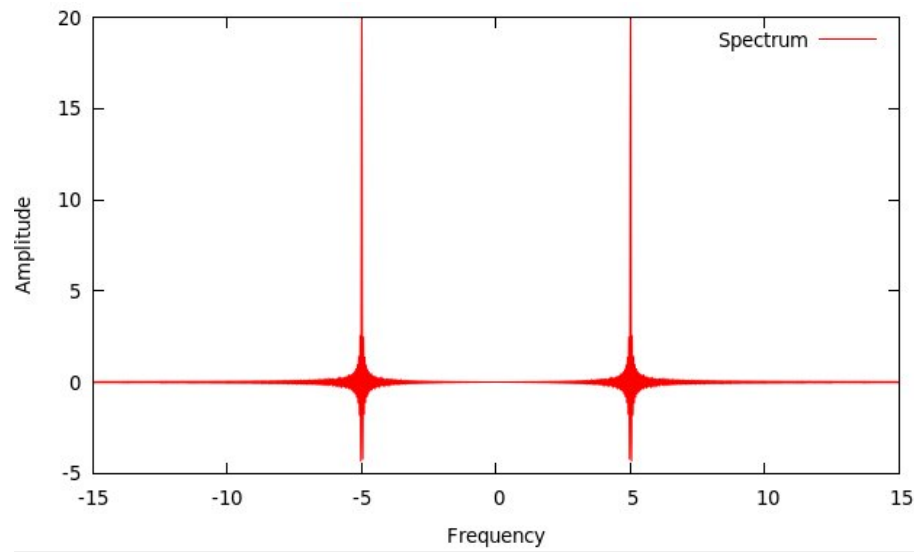
$$T \times f_0 = 10, T = 5, f_0 = 2$$



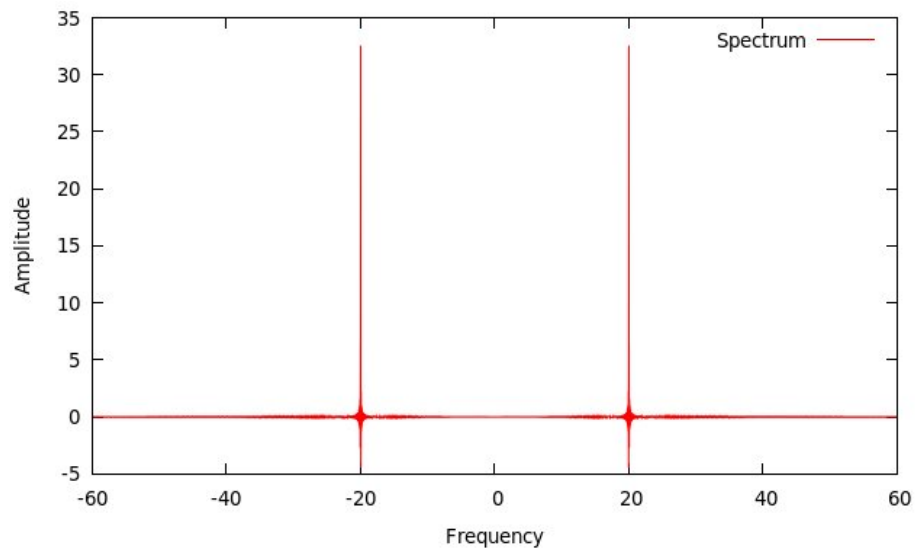
Cause the following time-series figure too oscillating to show, then we only demonstrate the spectrum.



$$T \times f_0 = 100, T = 5, f_0 = 20$$



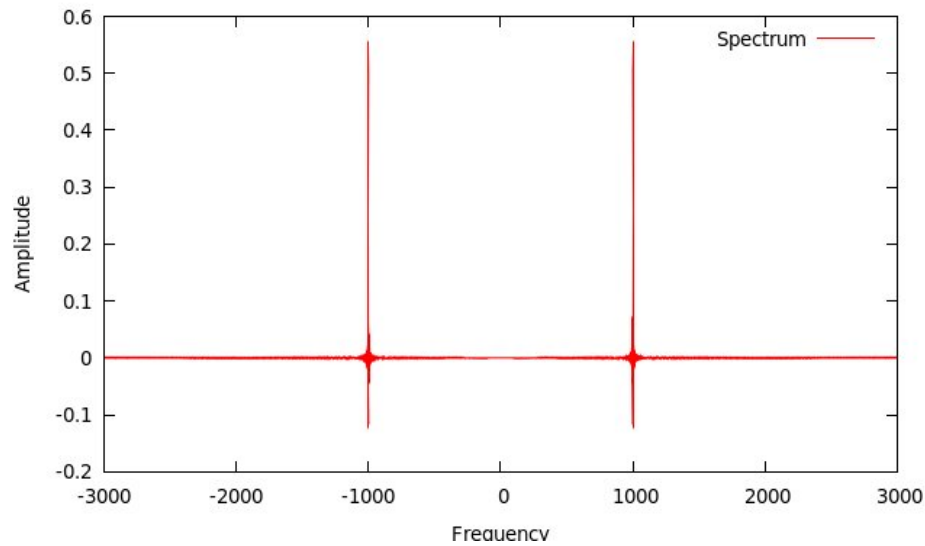
$$T \times f_0 = 1000, T = 20, f_0 = 50$$



c.

As we can find that when $T \times f_0$ get larger, the spectrum gets thinner. The original sinc function gradually become impulse function.

$$\lim_{\tau \rightarrow \infty} \tau \text{sinc}(f\tau) = \delta(f)$$



Problem 4

Prove the frequency convolution theory

a.

$$\begin{aligned}
\mathcal{F}\{x(t)h(t)\} &= \int_{-\infty}^{\infty} x(t)h(t)e^{-j\omega t} dt \\
&= \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega_1)e^{j\omega_1 t} dt \right] h(t)e^{-j\omega t} dt \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega_1) \left[\int_{-\infty}^{\infty} h(t)e^{j\omega_1 t} e^{-j\omega t} dt \right] d\omega_1 \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega_1) \left[\int_{-\infty}^{\infty} h(t)e^{-j(\omega - \omega_1)t} dt \right] d\omega_1 \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega_1)H(j(\omega - \omega_1)) d\omega_1 \\
&= X(f) * H(f)
\end{aligned}$$

b.

Show the Associativity

$$\begin{aligned}
(f_1 * f_2) * f_3 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(\tau_1)f_2(\tau_2)f_3((t - \tau_1) - \tau_2)d\tau_1d\tau_2 \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(\tau_1)f_2((\tau_1 + \tau_2) - \tau_1)f_3(t - (\tau_1 + \tau_2))d\tau_1d\tau_2
\end{aligned}$$

That we can find $\tau_3 = \tau_1 + \tau_2$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} f_1(\tau_1) \left[\int_{-\infty}^{\infty} f_2(\tau_2)f_3(t - \tau_2)d\tau_2 \right] d\tau_1 \\
&= f_1 * (f_2 * f_3)
\end{aligned}$$

Appendix Code

```

/* Start reading here */
#include <fftw3.h>
#include "gnuplot_i.h"
#define NUM_POINTS 8192

5  /* Never mind this bit */
#include <stdio.h>
#include <math.h>

10 #define SLEEP_LGTH 3
    // Tolerance
#define TOL 0.0001
#define A 1

15 double f0, T;
double HF[NUM_POINTS];
double ht[NUM_POINTS];
double freq_linspan[NUM_POINTS];

```



```
double time_linspace[NUM_POINTS];

20 double Q(double f) {
    double x = 2*M_PI*T*f;
    return sin(x)/(x);
}

25 void gen_ht(double T, double f0, double t[NUM_POINTS], double h[NUM_POINTS]) {
    int i;
    for (i=0; i < NUM_POINTS; ++i) {
        if (t[i] > T || t[i] < -T) {
30             h[i] = 0;
        } else {
            h[i] = A*cos(2*M_PI*f0*t[i]);
        }
    }
35 }

void gen_Hf(double f0, double f[NUM_POINTS], double H[NUM_POINTS]) {
    int i;
    for (i =0; i < NUM_POINTS; i++) {
40         H[i] = A*A*T*(Q(f[i]+f0) + Q(f[i]-f0));
    }
}

void freq_init(double freq_range[2], double out[NUM_POINTS]) {
45     int i;
    double f = freq_range[0];
    double f_increment = (freq_range[1]-freq_range[0]) / NUM_POINTS;
    for (i =0; i < NUM_POINTS; i++, f += f_increment) {
        out[i] = f;
50     }
}

void time_init(double time_range[2], double out[NUM_POINTS]) {
55     int i;
    double t = time_range[0];
    double t_increment = (time_range[1]-time_range[0]) / NUM_POINTS;
    for (i =0; i < NUM_POINTS; i++, t += t_increment) {
        out[i] = t;
60     }
}

/* Resume reading here */

int main(int argc, char *argv[]) {
65     if (argc >1) {
        f0 = atoi(argv[1]);
        T = atoi(argv[2]);
    } else {
        f0 = 2;
70         T = 5;
    }
}
```

```
gnuplot_ctrl *h;

/* Initialize the gnuplot handle */
75 printf("*** Digital Signal Processing through C ***\n") ;
h = gnuplot_init() ;

/* Spectrum */
80 double freq_range[2] = {-3*f0, 3*f0};
double time_range[2] = {-1.5*T, 1.5*T};

freq_init(freq_range, freq_linspan);
time_init(time_range, time_linspan);

85 gen_ht(T, f0, time_linspan, ht);
gen_Hf(f0, freq_linspan, HF);

/* Plot the signal */
gnuplot_resetplot(h) ;
90 gnuplot_setstyle(h, "lines") ;
gnuplot_set_xlabel(h, "Time");
gnuplot_set_ylabel(h, "Amplitude");
gnuplot_plot_xy(h,
                time_linspan,
95                ht,
                NUM_POINTS,
                "Time Series") ;
sleep(SLEEP_LGTH) ;

100 gnuplot_resetplot(h) ;

/* Plot the signal */
gnuplot_resetplot(h) ;
gnuplot_setstyle(h, "lines") ;
105 gnuplot_set_xlabel(h, "Frequency");
gnuplot_set_ylabel(h, "Amplitude");
gnuplot_plot_xy(h,
                freq_linspan,
                HF,
110                NUM_POINTS,
                "Spectrum") ;
sleep(SLEEP_LGTH) ;
printf("\n\n") ;
printf("*** end of DSP example ***\n") ;
115 /* Kill plotting Handler */
gnuplot_resetplot(h);
gnuplot_close(h);
return 0;
}
```