Digital Signal Processing: Assignment #2

Due on Thursday, March 27, 2014

Prof. Lee

Kevin Zhao, E14982022

Contents

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

$$\mathcal{F}\{h(t)\} = \int_{-\infty}^{\infty} h(t)\mathrm{e}^{-j2\pi ft}\,\mathrm{d}t$$
$$= \int_{-\infty}^{\infty} A\cos(2\pi f_0 t)\mathrm{e}^{-j2\pi ft}\,\mathrm{d}t$$

Then apply integral by part, we could derive that

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

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.

$$\frac{Ae^{-j2\pi ft}}{(2\pi f_0)^2 + (j2\pi f)^2}[-j2\pi f\cos(2\pi f_0 t) + 2\pi f_0\sin(2\pi f_0 t)]$$

Integrate from $-T$ to T, we get

$$A[\frac{sin(2\pi(f - f_0)T}{2\pi(f - f_0)} + \frac{sin(2\pi(f + f_0)T}{2\pi(f + f_0)}]$$

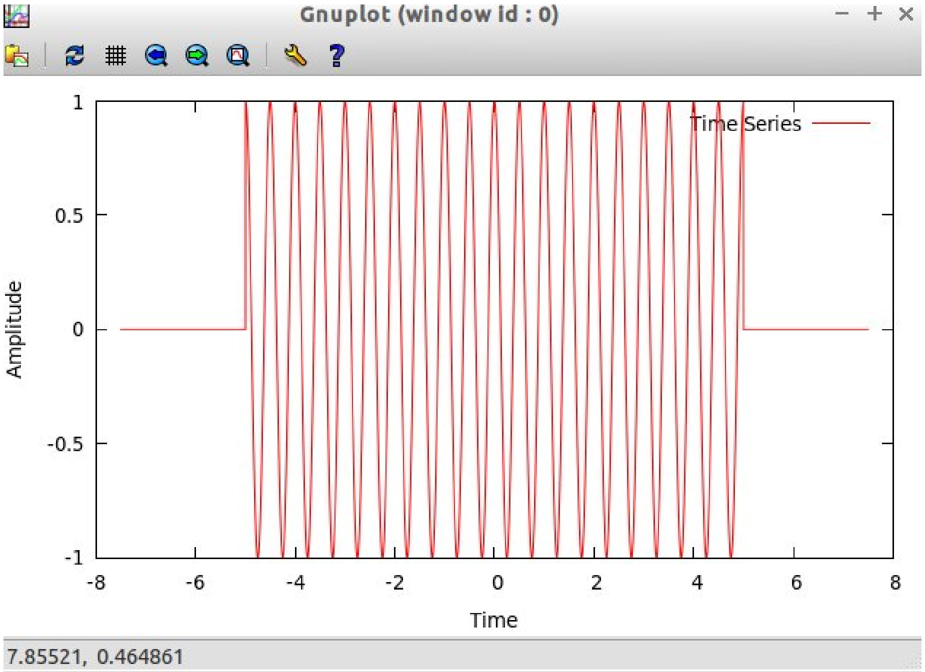
Finally, the result could be represented as

$$AT[Q(t + f_0) + Q(t - f_0)]$$

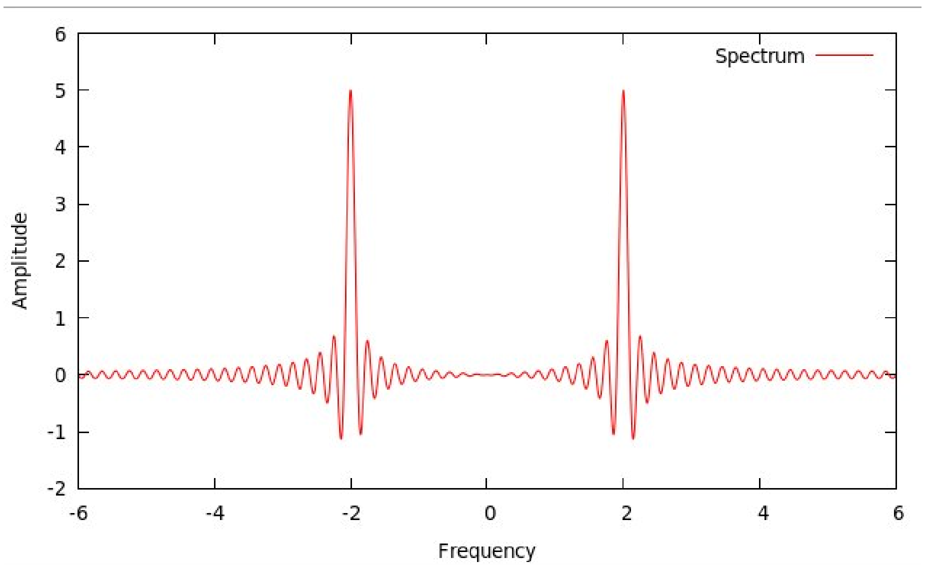
b.

Plot the results where

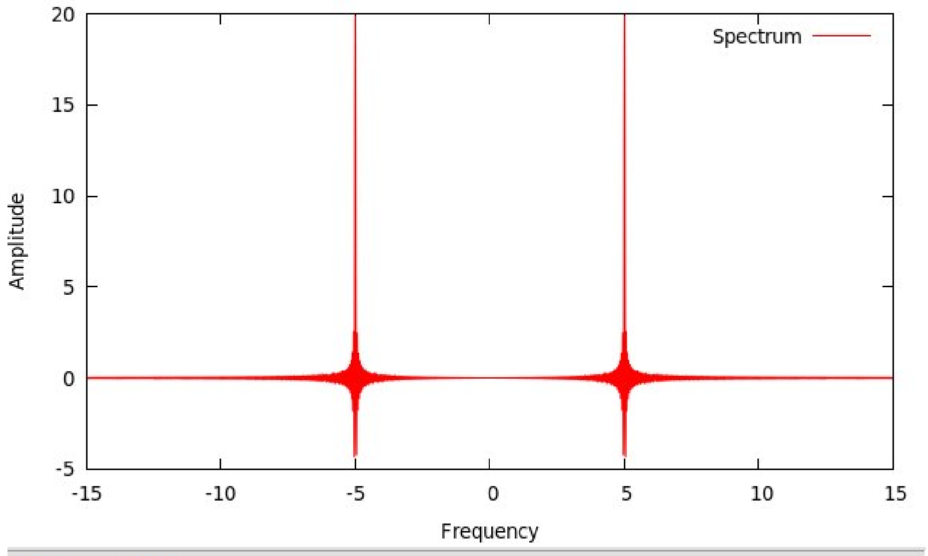
$T \times f_0 = 10, T = 5, f_0 = 2$



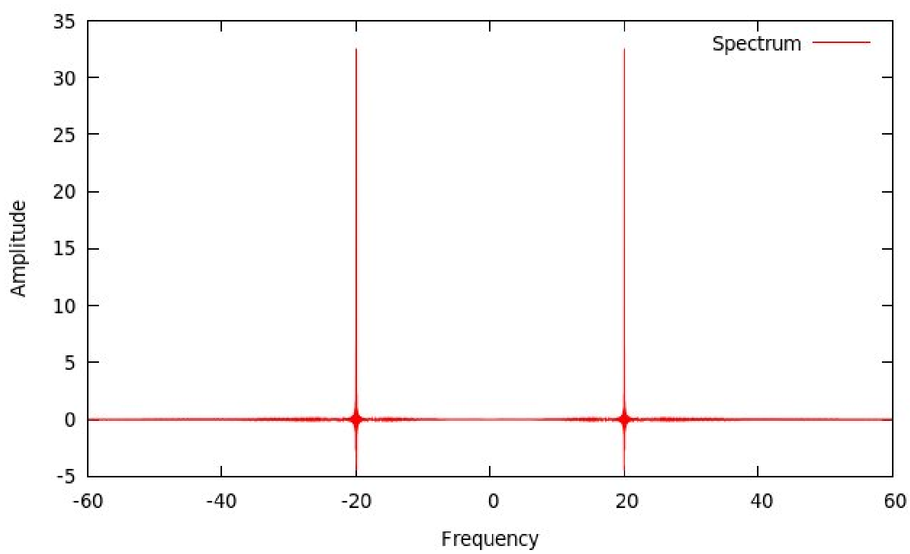
Cause the following time-series figure too oscillating to show, then we only demonstrate the spectrum.



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



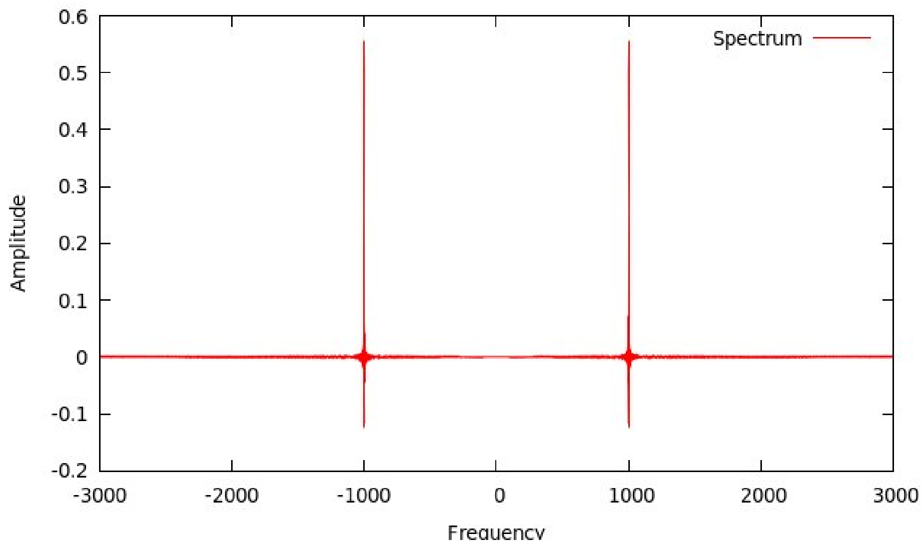
$T \times f_0 = 1000, T = 20, f_0 = 50$



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.

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



Problem 4

Prove the frequency convolution theory

a.

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

b.

Show the Associativity

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

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

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

Problem 5

Prove the frequency convolution theory

Appendix Code

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

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

#define SLEEP_LGTH  3
// Tolerance
```

```c
#define TOL     0.0001
#define A           1

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

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

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) {
            h[i] = 0;
        } else {
            h[i] = A*cos(2*M_PI*f0*t[i]);
        }
    }
}

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

void freq_init(double freq_range[2], double out[NUM_POINTS]) {
    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;
    }
}

void time_init(double time_range[2], double out[NUM_POINTS]) {
    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;
    }
}
/* Resume reading here */

int main(int argc, char *argv[]) {
```

```
    if (argc >1) {
        f0 = atoi(argv[1]);
        T  = atoi(argv[2]);
    } else {
        f0 = 2;
        T  = 5;
    }
    gnuplot_ctrl *h;

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

    /* Spectrum */
    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);

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

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

    gnuplot_resetplot(h) ;

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

```
    return 0;
}
```