ECE 348: Digital Signal Processing Lab
Lab 3 (Spring 2020)

Waheed U. Bajwa

February 24, 2020

General Instructions

Please submit a report that carefully answers the problems posed in here, and include all graphs and Matlab code. Examples of all the graphs that you need to generate in this lab and include in your report are contained in this handout. The reports must be uploaded to Sakai under Assignments within the specified time frame. The written report will count *only if* you had attended *and* worked on the lab during the *full* double-period lab session in which you are registered.

1 DTMF decoding (20 points)

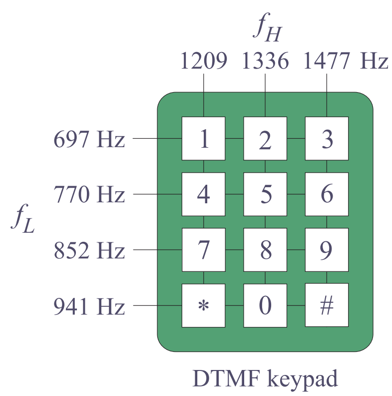

Fig. 1: Keypad of a phone and mapping of the frequencies upon pressing one of the keys on the keypad.

In the dual-tone multifrequency (DTMF) phone (aka, touch-tone phones), each pressed key generates a sum of two sinusoidal signals of two prescribed frequencies, one frequency chosen from a "low" group, and the other, from a "high" group of frequencies, as follows:

$$f_L = [697,\ 770,\ 852,\ 941]\,, \quad f_H = [1209,\ 1336,\ 1477] \quad \text{Hz}. \tag{1}$$

The generated signal goes over the phone line and is decoded at the switching center and forwarded on to its destination depending on the decoded number. One method of decoding the pressed key is to perform a DTFT on the received signal and identify the spectral peaks of the two sinusoidal frequencies. Fig. 1 shows a typical phone keypad and the pair of frequencies corresponding to each one of the keys. Mathematically, the signal generated upon pressing of each one of the keys, as well as the associated DTFT of the sampled signal, are given below:

$$x(t) = \sin(2\pi f_L t) + \sin(2\pi f_H t) = \text{analog}, \tag{2}$$

$$X(f) = \sum_n x(nT)e^{-2\pi j f n T} = \text{DTFT in Hz}, \tag{3}$$

$$f_s = \frac{1}{T} = \text{sampling rate}. \tag{4}$$

In this lab, each student is assigned a 3-digit DTMF signal based on his/her ID number and will try to decode it and plot the spectra of the 3 decoded digits. The following p-coded Matlab function, **dtmfsig**, generates the signal based on the student's ID number, to be entered as an integer:

```
y = dtmfsig(id);       % generate 3-digit DTMF signal
```

The signal vector, an example of which is shown in Fig. 2 below, consists of the concatenated signals for each digit, each having duration of 0.2 sec, with the digits being separated by a blanking period of 0.1 sec. The total duration of each digit plus blanking is 0.3 sec, and the total duration of **y** is 0.9 sec. The signal was sampled at a rate of $f_s = 8$ kHz, thus, consisting of $0.9\times 8000 = 7200$ samples, with each digit portion consisting of one-third of that, or, $N = 2400$ samples.

Problem 1.1 (2.5 points)**.** Plot your signal **y** versus time in seconds, as shown below in Fig. 2.

Problem 1.2 (7.5 points)**.** Divide your signal **y** into three length-N consecutive segments and, for each segment, compute its 7-point DTFT at the seven DTMF frequencies of (1) using the built-in function, **freqz** (2.5 points). From the two highest DTFT values, determine the two frequencies f_L, f_H contained in the segment, and hence, the dialed digit (2.5 points). Does it matter if each segment includes the blanking interval or not (2.5 points)?

Problem 1.3 (5 points)**.** Once you have identified your three digits, then for each length-N segment of **y**, compute its DTFT over a denser set of equally-spaced frequencies spanning the range $600 \le f \le 1600$ Hz, with a step increment of 1 Hz (2.5 points).

Plot versus f the magnitude of each DTFT normalized to unity maximum (1 point). On each graph, using red markers, add the 7 DTFT values at the 7 DTMF frequencies that you computed in Problem 1.2, and label the f-axis only with the two DTMF frequencies of the two peaks (1.5 points) (see example graphs in Fig. 2 corresponding to the dialed keys "538").

Problem 1.4 (5 points)**.** Using **fprintf** commands, make a table of the normalized DTMF spectral values (i.e., the values of the red dots) for the three keys, as shown below, where the values of the keys are to be replaced with yours.

```
   f  |  key 5    key 3    key 8
------|-------------------------
  697 |  0.024    1.000    0.003
  770 |  1.000    0.018    0.020
  852 |  0.015    0.001    1.000
  941 |  0.004    0.002    0.011
 1209 |  0.011    0.003    0.009
 1336 |  0.999    0.005    0.999
 1477 |  0.007    1.000    0.006
```

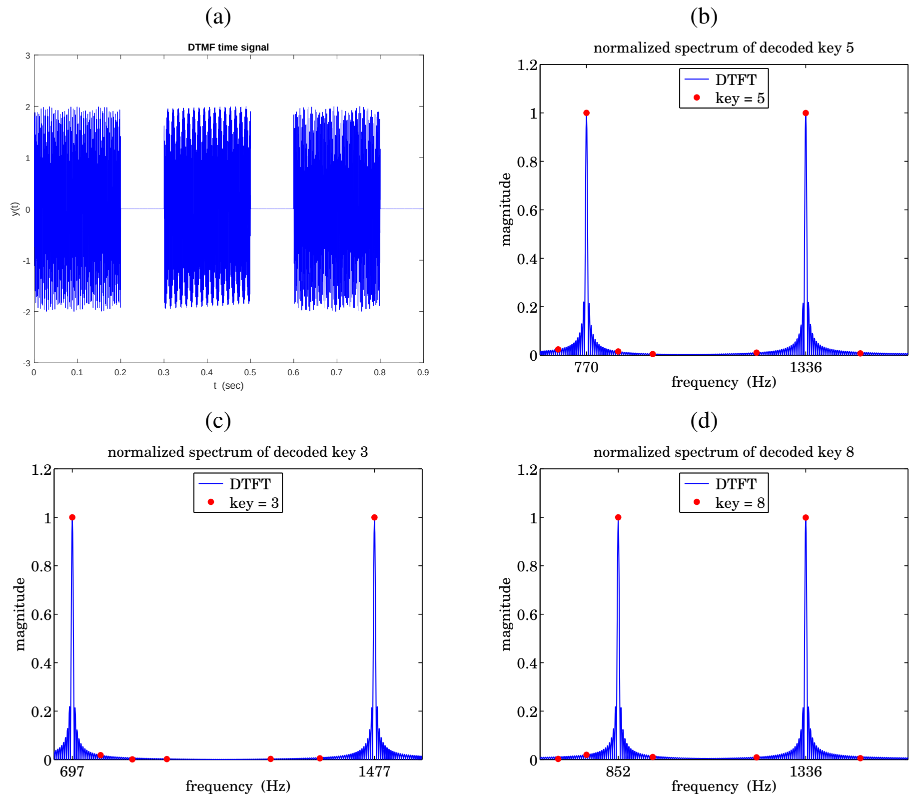

Fig. 2: Plots for Problems 1.1–1.4.

2 Staircase reconstructor and Butterworth postfilter (55 points)

In practice, analog reconstruction of a signal that was digitized at a rate of $f_s = 1/T$ samples/sec is typically done with a D/A converter that holds each sample constant for T seconds, thus filling the time gaps between samples. The resulting staircase output can be post-filtered by an analog postfilter to smooth out the sharp staircase levels. The postfilter is designed to have cutoff at the Nyquist frequency $f_s/2$. This reconstruction process is illustrated below.

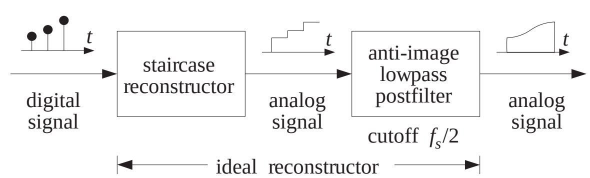

Fig. 3: Block diagram of practical reconstruction in DSP systems.

We begin with some background. Please consult also Appendix A.10 of your textbook. We will work with frequency f in units of Hz (instead of radians/second), and in this notation, the Fourier transform of an analog signal $x_a(t)$ and its inverse are given by:

$$X_a(f) = \int_{-\infty}^{\infty} x_a(t) e^{-2\pi j f t}\, dt \quad \Leftrightarrow \quad x_a(t) = \int_{-\infty}^{\infty} X_a(f) e^{2\pi j f t}\, df. \tag{5}$$

If the signal $x_a(t)$ is sampled at a rate of $f_s = 1/T$, the spectrum (i.e., DTFT) of the discrete-time signal

$x_a(nT)$ will consist of the periodic replication of the original analog spectrum $X_a(f)$ with period f_s,

$$x_d(t) = \sum_{n=-\infty}^{\infty} x_a(nT)\delta(t-nT) = \text{ideally sampled time signal,} \tag{6}$$

$$X_d(f) = \sum_{n=-\infty}^{\infty} x_a(nT)e^{-2\pi jfTn} = \frac{1}{T}\sum_{m=-\infty}^{\infty} X_a(f-mf_s) = \text{Poisson summation formula.} \tag{7}$$

As an analog filter, the staircase reconstructor has impulse response and frequency response:

$$h(t) = u(t) - u(t-T) \quad \Leftrightarrow \quad H(f) = \frac{1-e^{-2\pi jfT}}{2\pi jf} = T\,G(f)\,e^{-\pi jfT}\,, \quad G(f) = \frac{\sin(\pi fT)}{\pi fT}. \tag{8}$$

In this lab, the signal to be sampled is taken to be a sinusoid of frequency f_0, with f_0 assumed to lie in the Nyquist interval $[-f_s/2, f_s/2]$,

$$x_a(t) = e^{2\pi jf_0t} \quad \Leftrightarrow \quad X_a(f) = \delta(f-f_0).$$

The spectrum of the sampled sinusoid consists of the replications of f_0 at multiples of f_s, that is, the replicated frequencies, $f_m = f_0 + mf_s$, $-\infty < m < \infty$,

$$X_d(f) = \frac{1}{T}\sum_{m=-\infty}^{\infty} X_a(f-mf_s) = \frac{1}{T}\sum_{m=-\infty}^{\infty}\delta(f-f_0-mf_s) = \frac{1}{T}\sum_{m=-\infty}^{\infty}\delta(f-f_m).$$

After passing through a reconstruction filter $H(f)$, the reconstructed signal in the frequency and time domains will be:

$$X_r(f) = H(f)X_d(f) = H(f)\frac{1}{T}\sum_{m=-\infty}^{\infty}\delta(f-f_m) = \frac{1}{T}\sum_{m=-\infty}^{\infty}H(f_m)\delta(f-f_m), \tag{9}$$

$$x_r(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty}H(f_m)e^{2\pi jf_mt}. \tag{10}$$

For an ideal reconstructor, since f_0 was assumed to be in the Nyquist interval, the filter will reject completely all f_m for $m \neq 0$, and result in the original sinusoid, $x_a(t) = e^{2\pi jf_0t}$. But for the staircase reconstructor, the out-of-band components will only be partially removed. A subsequent postfilter can remove them more completely. We will be considering these issues in this lab.

For a real-valued sinusoid, and the staircase reconstructor of (8), we have,

$$x_a(t) = \cos(2\pi f_0t), \tag{11}$$

$$x_r(t) = \sum_{m=-\infty}^{\infty} G(f_m)\cos(2\pi f_mt - \pi f_mT) = \text{staircase-reconstructed signal,}$$

and if we truncate the sum to $-M \leq m \leq M$, we obtain the computable approximation,

$$x_r(t) = \sum_{m=-M}^{M} G(f_m)\cos(2\pi f_mt - \pi f_mT) = \text{staircase-reconstructed signal.} \tag{12}$$

The truncation causes some unwanted ripples in the staircase levels, which can be seen in the graphs in Fig. 4. These ripples can be reduced by using some weights w_m, like Hamming weights (to be studied later in the class),

$$x_h(t) = \sum_{m=-M}^{M} w_m\, G(f_m)\cos(2\pi f_m t - \pi f_m T) = \text{staircase-reconstructed signal}, \tag{13}$$

$$w_m = 0.54 + 0.46\cos\left(\frac{\pi m}{M}\right), \quad -M \le m \le M\,, \qquad \text{(Hamming weights)}. \tag{14}$$

If the reconstructed signal $x_h(t)$ of (13) is postfiltered by an ideal postfilter that adequately removes the components, f_m, $m \neq 0$, then, only the f_0 component will survive at the postfilter output:

$$x_p(t) = G(f_0)\cos(2\pi f_0 t - \pi f_0 T). \tag{15}$$

This is essentially the original analog signal of (11), but subject to an attenuation and phase shift due to the staircase reconstructor. It is possible to digitally compensate for this effect, but we will not consider it in this lab (see further discussion in Appendix A.10). In practice, we cannot use an ideal postfilter, but we may design it to approximate the ideal one. The signal $x_p(t)$ will serve as an ideal reference to be compared to the output from the actual designed postfilter.

For this lab, the postfilter is chosen to be a 6th order Butterworth analog lowpass filter with a 3-dB cutoff frequency at Nyquist, $f_{3\text{dB}} = f_s/2$. Such a filter can be designed with the MATLAB function, **butter**, using the following command which generates the numerator and denominator coefficients, **b**, **a**, of the filter,

```
[b,a] = butter(6, 2*pi*f3dB, 's');
```

We recall that the frequency response of an analog filter with order-K numerator and order-N denominator is given as follows in terms of the filter coefficients, $\mathbf{b} = [b_0, b_1, \ldots, b_K]$, $\mathbf{a} = [a_0, a_1, \ldots, a_N]$,

$$H(f) = \left.\frac{b_0 s^K + b_1 s^{K-1} + \cdots + b_K}{a_0 s^N + a_1 s^{N-1} + \cdots + a_N}\right|_{s=2\pi j f}.$$

This can be evaluated with the built-in function, **freqs**. But for some unknown reason, **freqs**, as well as **freqz**, cannot evaluate it at a single frequency value, requiring instead a vector of frequencies. Therefore, you may use the following little function to define the frequency response of the designed postfilter,

```
Hpost = @(f) polyval(b, 2*pi*j*f) ./ polyval(a, 2*pi*j*f);
```

We are now ready to proceed with the lab.

Problem 2.1 (12.5 points)**.** Consider the sinusoid of (11) of frequency, $f_0 = 0.125$ kHz, where t is in units of milliseconds. The sampling rate is taken to be $f_s = 1$ kHz, and choose $M = 20$ in the finite summations of (12) and (13).

Compute the four signals $x_a(t)$, $x_r(t)$, $x_h(t)$, $x_p(t)$, defined in (11)–(15), over the time interval $0 \le t \le$ 20 msec at equally-spaced times with a step increment of 0.01 msec. The computation of each signal must be carried out with a single MATLAB command (no loops) (4 points).

On the same graph, plot the three signals, $x_a(t)$, $x_r(t)$, $x_p(t)$ versus t, for the case of rectangular weights ($w_m = 1$) (3 points). Then, do a similar graph for the signals, $x_a(t)$, $x_h(t)$, $x_p(t)$, for the case of Hamming weights (3 points). See example graphs below in Fig. 4. Note the slight attenuation and phase shift of $x_p(t)$ versus $x_a(t)$. Calculate that attenuation and phase delay and verify that it is consistent with what you see in the graphs (2.5 points).

Problem 2.2 (7.5 points)**.** Filter the Hamming-weighted signal $x_h(t)$ of (13) through the Butterworth postfilter to obtain the output $x_f(t)$ (2.5 points). The analog filtering operation can be carried out with the built-in function, **lsim**,

```
xf = lsim(b,a,xh,t);
```

Next, we compare the reference output $x_p(t)$ with the actual output $x_f(t)$. By focusing your attention on the time subinterval $8 \leq t \leq 10$ msec, determine using the MATLAB function, **max**, the time-delay of the peak of $x_f(t)$ relative to the peak of $x_p(t)$ (2.5 points). Also, calculate the exact time delay from the phase-delay of the Butterworth filter at frequency f_0, which is defined by (2.5 points)

$$\tau(f_0) = -\frac{\text{Arg}\, H_{\text{post}}(f_0)}{2\pi f_0}.$$

Problem 2.3 (15 points)**.** Calculate the magnitude responses, $|G(f)|$ and $|H_{\text{post}}(f)|$, of the staircase reconstructor and Butterworth postfilter over the frequency interval $0 \leq f \leq 4$ kHz and plot them on the same graph (5 points). Add to the graph the magnitude response of the combined filter, $|G(f)H_{\text{post}}(f)|$, as well the discrete values $|G(f_m)|$ at the sinusoid's replicated frequencies $f_m = f_0 + mf_s$ for $m = 0, 1, 2, 3$ (5 points). See example graphs below in Fig. 4.

Next, redo the same graph, but in this case add the post-filtered values, $|G(f_m)H_{\text{post}}(f_m)|$, at the replicated frequencies (5 points). This graph clearly shows that the only surviving frequency is effectively the main one at f_0.

Problem 2.4 (5 points)**.** Repeat Problem 2.1 and Problem 2.2 for the case $f_0 = 0.25$ kHz.

Problem 2.5 (5 points)**.** For both cases of f_0, use **fprintf** commands to make a table of the filter magnitudes $|G(f_m)$ and $|G(f_m)H_{\text{post}}(f_m)|$, as shown below. The table demonstrates how the out-of-band frequencies are attenuated by the staircase reconstructor, and even more completely by the postfilter, with both filters having only a minor impact on the f_0 component.

```
fm = f0 + m*fs    |         |G(fm)|        |    |G(fm)Hpost(fm)|
------------------------------------------------------------
0.1250  0.2500    |  0.974495  0.900316  |  0.974495  0.900206
1.1250  1.2500    |  0.108277  0.180063  |  0.000835  0.000738
2.1250  2.2500    |  0.057323  0.100035  |  0.000010  0.000012
3.1250  3.2500    |  0.038980  0.069255  |  0.000001  0.000001
```

Problem 2.6 (10 points)**.** Make a plot of the phase-delay of the Butterworth postfilter over $0 \leq f \leq 4$ kHz (5 points), and add the exact and estimated phase delays from Problem 2.2, for both cases of f_0 (2.5 points).

Finally, using three **fprintf** commands, display the exact and estimated phased delays in a table as follows (2.5 points), where the * entries are to be replaced by the computed numerical values,

```
 phase delay      exact       estimated
----------------------------------------
 f0 = 0.125   |  *.****    |   *.****
 f0 = 0.250   |  *.****    |   *.****
```

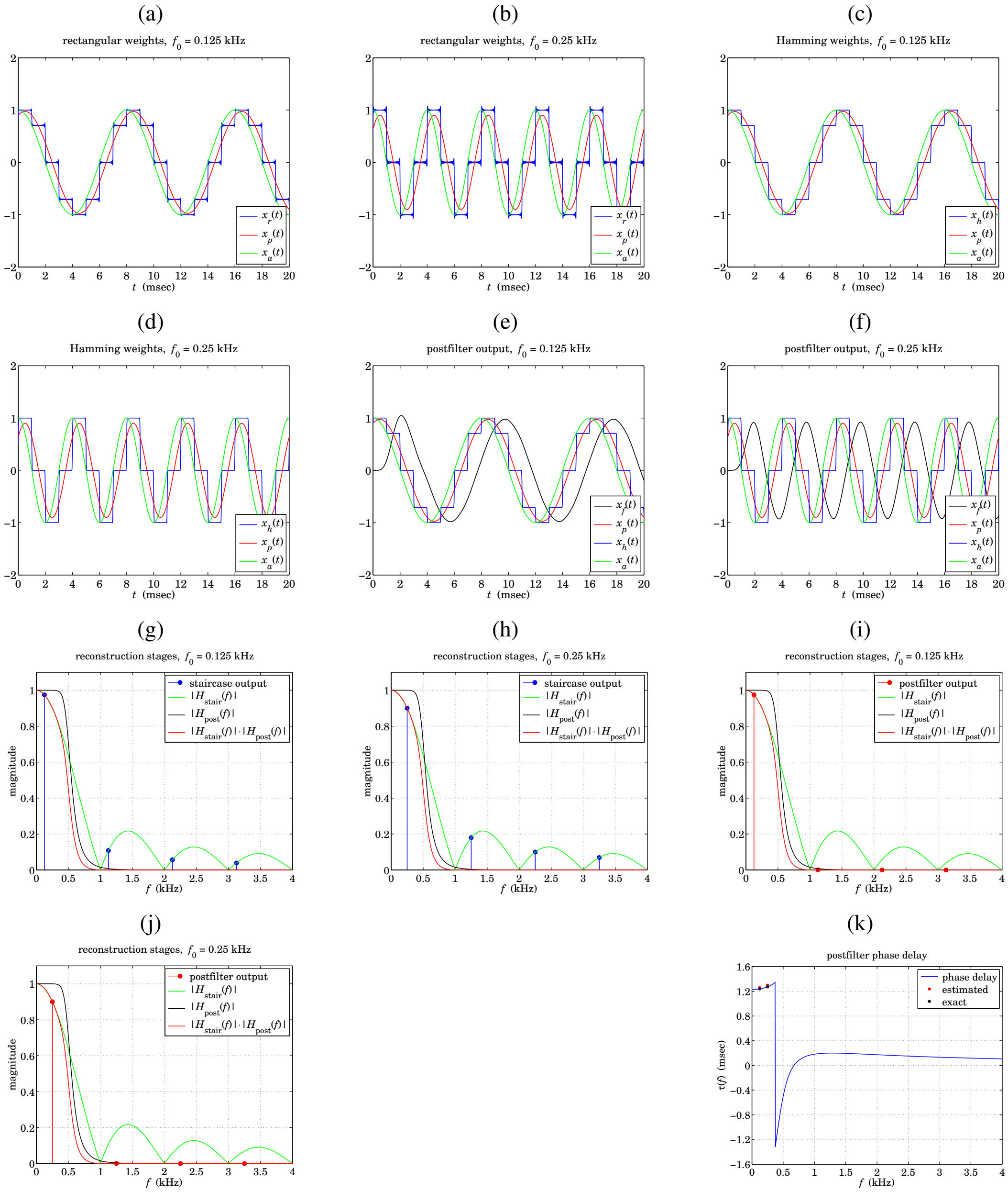

Fig. 4: Plots for Problems 2.1–2.6.

3 Poisson Summation Formula (25 points)

Consider the following analog signal and its Fourier transform:

$$x(t) = te^{-at}e^{j\Omega_0 t}u(t)\,, \qquad X(\Omega) = \frac{1}{\big[a + j(\Omega - \Omega_0)\big]^2}\,, \tag{16}$$

where $\Omega_0 = 2\pi f_0$. The signal is sampled at a rate of $f_s = 1/T$ (see Fig. 5). The Fourier transform of the sampled signal can be expressed in two equivalent ways, which are a consequence of the Poisson summation formula: (i) in terms of the DTFT of the discrete-time samples $x(nT)$, and (ii) in terms of the summation of the spectral replicas of $X(\Omega)$ at multiples of the sampling frequency,

$$X_d(\Omega) = \sum_{n=0}^{\infty} x(nT)e^{-j\Omega nT} = \frac{Te^{-aT}e^{-j(\Omega-\Omega_0)T}}{\left[1 - e^{-aT}e^{-j(\Omega-\Omega_0)T}\right]^2}, \tag{17}$$

$$X_d(\Omega) = \frac{1}{T}\sum_{m=-\infty}^{\infty} \frac{1}{\left[a + j(\Omega - \Omega_0 - m\Omega_s)\right]^2}, \qquad \Omega_s = 2\pi f_s = \frac{2\pi}{T}, \tag{18}$$

and if we truncate the sum:

$$X_M(\Omega) = \frac{1}{T}\sum_{m=-M}^{M} \frac{1}{\left[a + j(\Omega - \Omega_0 - m\Omega_s)\right]^2}. \tag{19}$$

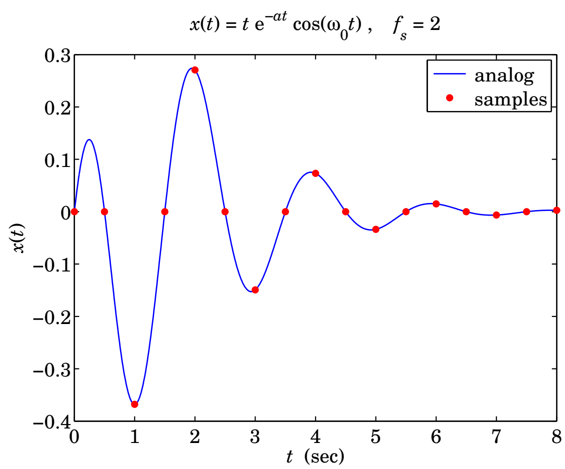

Fig. 5: Plots for Problem 1.2, Problem 1.3, and Problem 1.4.

The purpose of this lab is to compare $X(\Omega)$, $TX_d(\Omega)$, and $TX_M(\Omega)$ for various values of T and M.

Problem 3.1 (5 points)**.** Consider the following values of the parameters, $a = 1$ sec^{-1}, $f_0 = 0.5$ Hz, with t in units of seconds. For the following two values of the sampling frequency, $f_s = 1$ Hz and $f_s = 2$ Hz, make a plot of the real-part of the above signal $x(t)$ over the time interval $0 \le t \le 8$ sec, and add the time samples $x(nT)$ on the graphs (see Fig. 5).

Problem 3.2 (20 points)**.** For each of the three sampling frequencies $f_s = 0.5, 1, 2$ Hz, and for each of the two values, $M = 1, 2$, of the parameter M that defines the finite sum in (19), compute and plot the magnitude spectra, $|X(\Omega)|$, $T|X_d(\Omega)|$ of (17), and $T|X_M(\Omega)|$ over the frequency interval $0 \le f \le 4$ Hz (see Fig. 6).

Remarks: Observe how the agreement of all three spectra improves with increasing f_s and with increasing M. Clearly, no finite sum (19) can approximate the full DTFT of (17) over all frequencies, because any finite sum ultimately tends to zero for large Ω. However, as you will observe, the agreement between (17) and (19) extends over an ever increasing band of frequencies as M increases.

Note also the frequency shifting that takes place because of the presence of the sinusoidal factor $e^{j\Omega_0 t}$ in $x(t)$, causing the centering of $X(\Omega)$ at Ω_0.

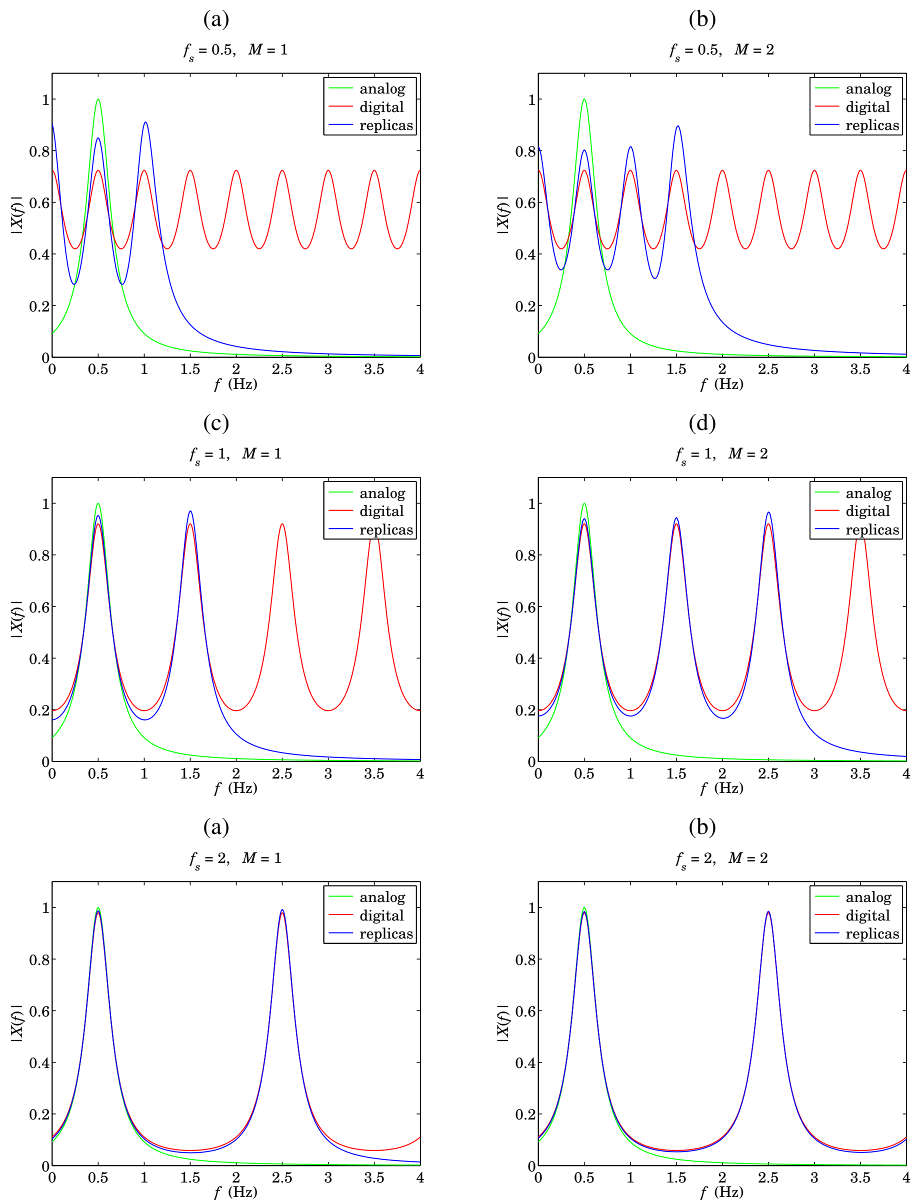

Fig. 6: Plots for Problem 3.2.