ECE333 Signals and Systems, III Problems

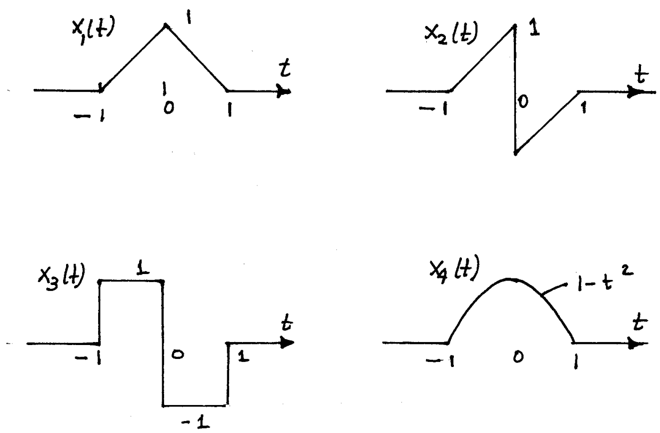
Unit 2 – Time-Domain Analysis

Problem 2.1 – Integrators

Each of the signals shown is the input to an ideal integrator, i.e.,

$$y(t) = \int_{-\infty}^{t} x(\tau) d\tau$$

Carefully, and to scale, sketch the resulting outputs.

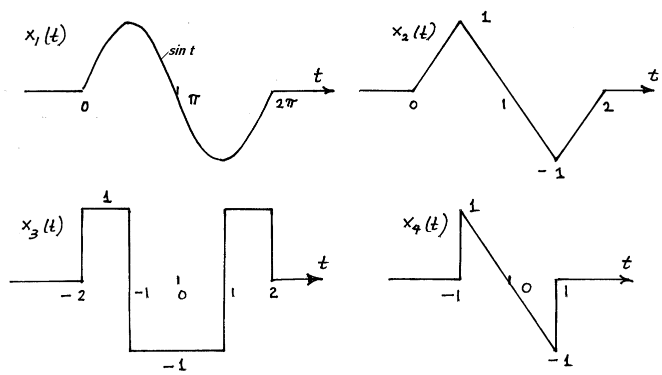


Problem 2.2 – More Integrators

Each of the signals shown is the input to an ideal integrator, i.e.,

$$y(t) = \int_{-\infty}^{t} x(\tau) d\tau$$

Carefully, and to scale, sketch the resulting outputs.



Problem 2.3 – Differentiators

Each of the signals in Problem 2.1 is the input to an ideal differentiator, i.e.,

$$y(t) = \dot{x}(t) = dx(t)/dt$$

Carefully, and to scale, sketch the resulting outputs. (Do not omit any impulse that may be present in the output.)

Problem 2.4 – More Differentiators

Each of the signals in Problem 2.2 is the input to an ideal differentiator, i.e.,

$$y(t) = \dot{x}(t) = dx(t)/dt$$

Carefully, and to scale, sketch the resulting outputs. (Do not omit any impulse that may be present in the output.)

Problem 2.5 – Convolution Integrals

The impulse response of a linear, time-invariant system is defined as follows:

$$h(t) = \begin{cases} 0, & t < 0 \\ 1 - t, & 0 < t < 1 \\ 0, & t > 1 \end{cases}$$

(a) Using the two forms of the convolution integral, find the responses to the following two inputs:

1.

$$x_1(t) = \begin{cases} 0, & t < 0 \\ e^{-t}, & t > 0 \end{cases}$$

2.

$$x_2(t) = \begin{cases} 0, & t < 0 \\ 1, & 0 < t < 2 \\ 0, & t > 2 \end{cases}$$

Hint: Don't try to do these without drawing a picture!

(b) Find the step response

$$a(t) = \int_{-\infty}^{t} h(\tau)d\tau$$

of the system.

(c) Using the result of Part(b), find the response to $x_2(t)$ **without** using convolution.

Problem 2.6 – More Convolution Integrals

The impulse response of a linear, time-invariant system is defined as follows:

$$h(t) = \begin{cases} 0, & t < 0 \\ t, & 0 < t < 1 \\ 0, & t > 1 \end{cases}$$

(a) Using the two forms of the convolution integral, find the responses to the following two inputs:

1.

$$x_1(t) = \begin{cases} 0, & t < 0 \\ e^{-t}, & t > 0 \end{cases}$$

2.

$$x_2(t) = \begin{cases} 0, & t < 0 \\ 1, & 0 < t < 1 \\ -1, & 1 < t < 2 \\ 0, & t > 2 \end{cases}$$

Hint: Don't try to do these without drawing a picture!

(b) Find the step response

$$a(t) = \int_{-\infty}^{t} h(\tau) d\tau$$

of the system.

(c) Using the result of Part(b), find the response to $x_2(t)$ **without** using convolution.

Problem 2.7 – Convolution Using Step Response

Find the response of the system of Problem 2.5 for each of the inputs, by convolving the step response you found in Problem 2.5 with the time-derivatives of the inputs. Use both forms of the convolution integral.

Problem 2.8 – More Convolution Using Step Reponse

Repeat Problem 2.7 for the signals of Problem 2.6.

Problem 2.9 – Numerical Convolution (Matlab Problem)

Using Matlab, generate arrays, each of 100 points, filled with the numerical values of $h(t)$, $x_1(t)$, and $x_2(t)$ of Problem 2.5. Plot these to verify that they are correct. Then, using the Matlab function `conv` find the response of system of Problem 2.5 to the two inputs and plot the results. Compare with the analytical results obtained above.

Do your work using an m-file and submit the m-file and your plot results.

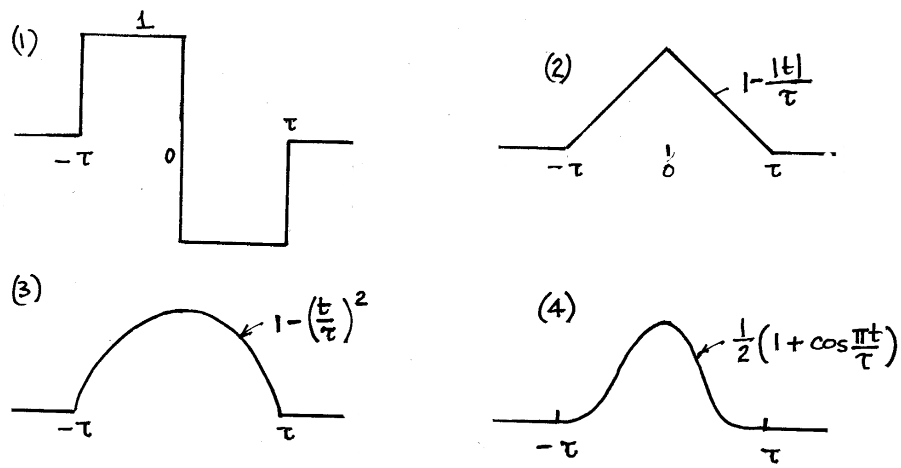
Problem 2.10 – Numerical Convolution (Matlab Problem)

Repeat Problem 2.9 for the signals of Problem 2.6.

Unit 3 – Fourier Series for Periodic Signals

Problem 3.1 — Computation of Fourier Series

One cycle of four periodic waveforms is shown below:



(a) Evaluate the Fourier coefficients for each of the waveforms using 2 methods:

1. Using the defining integral, either computed by PAP (pencil and paper), or with the aid of the Matlab symbolic tool box.
2. Using appropriate properties of Fourier series.

b)Plot the amplitude and phase spectra for each of the waveforms, including X_0 to X_{10} for (1) $\tau = T/4$ and (2) $\tau = T/2$. (Use of Matlab is encouraged.)

Problem 3.2 — Numerical Computation of Fourier Series (Matlab Problem)

(a) For each of the waveforms of Problem 3.1 and for $\tau = T/4$ create and run an m-file to do the following:

1. Fill an array of 256 points with corresponding values of the waveforms.
2. Plot the results to verify correctness.
3. Calculate the Fourier series coefficients using the Matlab function fft
4. Plot the amplitude and phase spectra.
5. Compare the results with those obtained in Problem 3.1.

(b) (Optional–Extra credit) Repeat Part (a) for $\tau = T/2$
Submit a print-out of your m-file and the plot results.

Problem 3.3 — Application of Parseval's Theorem

For each of the waveforms of Problem 3.1 and for $\tau = T/4$ do the following:

(a) Compute the mean square value using the integral definition:

$$\overline{x^2} = \frac{1}{T}\int_T x^2(t)dt \qquad (1)$$

(b) Compute the N-harmonic approximate mean square value using the frequency domain formula

$$\overline{x^2}_N = \sum_{n=-N}^{N} |X_n|^2 \qquad (2)$$

where N is chosen such that the error (i.e., the difference between the result of calculation (1) and (2) is 1 % or less. Tabulate N for each of the waveforms.

(c) Repeat Part (b), except for an error of 0.01 % or less.

Hint: Write a Matlab m-file to perform this calculation.

Problem 3.4 — Signal Compression

The Matlab m-file compress.m illustrates the rudiments of frequency-domain "lossy" signal compression. Check it to verify that does the following:

- Reads a .wav file into Matlab's work space as the variable signal
- Plots the signal
- Calculates and plots the *fft* of the signal
- Determines the number of FLOPS that calculation of the FFT requires.
- Zeros all the harmonics that fall below a specified threshold.
- Determines the compression ratio, i.e., the ratio of the number of zeroed harmonics to the total number originally present.
- Calculates the inverse transform ifft of the "compressed" signal.
- Superimposes the plots of the starting signal and the compressed signal.
- Writes a .wav file of the resulting time-domain output.

(a) Locate several (at least 4) "interesting" wave files that are present on your computer, copy them into Matlab's work space and then do the following:

- Listen to the original signal using your favorite multimedia player.
- Compress the signal using several different thresholds and determine the corresponding level of compression.
- Compare the spectra of the original and the compressed signal.
- Compare the plots of the corresponding time-domain signals.

- Listen to the resulting signal and determine if you can detect any difference.

Report your results (submit hard copies of graphics) and the original and compressed signals on a floppy disk.

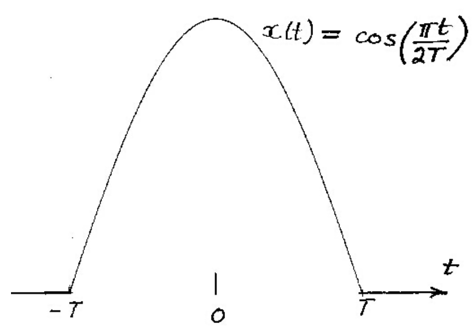
(b) For signal(s) the length of which are not 2^n with $n =$ integer, truncate the signal (by removing the % sign in the m-file and setting NT to the largest number less than the number of terms in the original signal that's 2^n. (Alternatively, pad the original signal with enough zeros to make it 2^n.) Repeat Part(a), and take note of the number of flops required to perform the calculations.

(c) **Extra credit.** The threshold used in the m-file is the ratio of the magnitude of tested harmonic to the fundamental. This may not be the best way to choose the threshold. Investigate other ways (e.g., ratio of magnitude of harmonic to RMS value of entire signal) to choose the threshold.

(d) **Extra credit.** The m-file illustrates the *effect* of compression, but does not really achieve compression because it simply replaces the harmonics that are below the threshold with zeros, so no compression is actually achieved. To achieve the benefits of compression, you must actually *remove* the harmonics that are zero from the transmitted file, but you must encode their locations in a "header", prior to transmission, so that the zero harmonics can effectively be restored at the receiving end. Devise an efficient algorithm for doing this and demonstrate the actual compression and restoration.

Problem 3.5 — Numerical Convolution

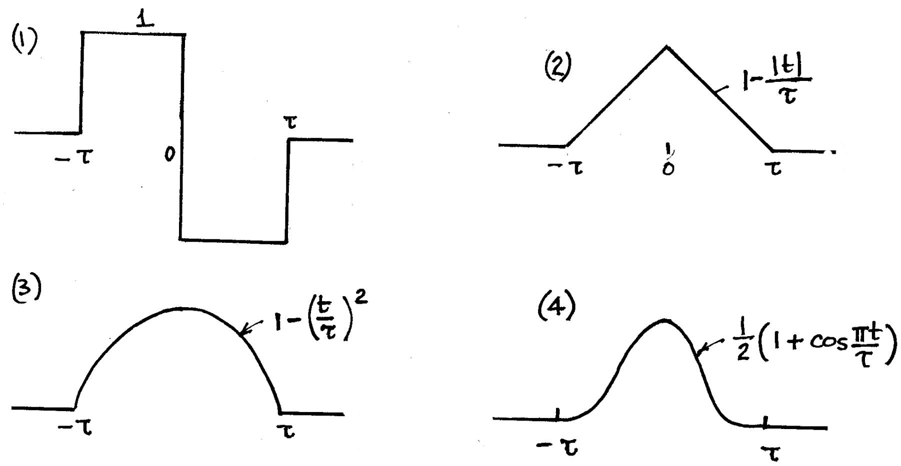
The signal shown is sampled at $t = -T/2, -T/4, 0, T/4$
(a) Calculate the 4 point discrete Fourier transform (DFT) of this signal.
(b) Calculate the inverse DFT of the result obtained in part (a) and verify that it returns the original signal.
(c) Check your results with Matlab



Unit 4 – Fourier Transforms

Problem 4.1 — Fourier Transforms of Pulses

Calculate the Fourier transform of each of the "pulses" shown in the figure.



Hint: Use results of Problem 3.1.

Problem 4.2 — More Fourier Transforms

(a) Calculate the Fourier transforms of each of the following signals:

$$x_1(t) = \begin{cases} e^{at}, & t < 0 \\ -e^{-at}, & t > 0 \end{cases}$$

$$x_2(t) = e^{-(t/2\tau)^2}$$

(b) Plot the amplitude and phase spectra of these signals.

Problem 4.3 — Signals with Band-Limited Spectra

Using the Fourier transform inversion formula, find and plot the signals having the following "band-limited" Fourier transforms:

(a) $X_1(j\omega) = \begin{cases} 1, & |\omega| < W \\ 0, & |\omega| > W \end{cases}$

(b) $X_2(j\omega) = \begin{cases} 1 - |\omega|/W, & |\omega| < W \\ 0, & |\omega| > W \end{cases}$

Problem 4.4 — Numerical Evaluation of Responses and Spectra Using Matlab

The following m-file demonstrates the use of Matlab for numerical calculation of time-domain and frequency-domain responses of linear systems.

```
%Fourier and Laplace Transform Demonstration

clf                 %initialize

t=[0:.02:2.54];      %Create time base of 256 points.
% ----- Filter Characteristics ----------
syms s
H=1/(s+4)           % Transfer function of filter
h=ilaplace(H)       % Find expression for impulse response
hh=eval(h);         % Evaluates the expression for h at the points in t.
HH=fft(hh);         % Take the Fourier transform of numerical values of impulse res
HHS=fftshift(HH);   % Shift to make graphs look nice.
figure(1)           % Plot magnitude and phase spectra.
subplot(2,1,1) plot(abs(HHS)),title('Spectrum of
Filter'),ylabel('Magnitude') subplot(2,1,2)
plot(angle(HHS)),ylabel('Phase')
figure(2)           % Compare original impulse response with ifft(fft(h))
hold on
plot(t,hh),title('Impulse response') % Solid line represents original signal
hi=ifft(HH);
plot(t,real(hi),'.')                 % Dotted line represents ifft(fft(signal))
hold off
% ------ Input Signal -----------------
head=ones(1,64); tail=zeros(1,64); x=[head,tail]; figure(3)
plot(t,x),title('Input Signal')
%------- Generate Output in Time Domain ----
y=conv(hh,x);
y(256)=0;   %Add one more element to make size(y)=256.
X=fft(x); Y=fft(y);
% ------ Compute Y(jw)= H(jw)*X(jw) ------
YY=HH.*X;    % The .* operator means multiply each element of HH by the correspondi
             % element of X

figure(4)    % Compare result of time-domain calculation with frequency domain calc
hold on tt=[0:.02:5.10]; plot(tt,y),title('Output Signal')
yy=ifft(YY);
plot(t,real(yy),'.') % Dots for inverse Fourier transform of H(jw)*X(jw)
hold off
```

(a) Study the m-file to understand how it works. Run the m-file to see that operations in the time-domain and frequency-domain produce the same results (to the limits of resolution of your graphics.)

(b) Modify the m-file and re-run it for input pulses of different lengths.

(c) Modify the m-file to determine the response to the filter of having the transfer function

$$H(s) = \frac{1}{s^2 + 8s + 32}$$

In submitting your homework, include your name in the title of each figure as evidence that you did the work.

The actual m-file is included in the folder `mfiles` to save you the need for retyping it yourself.

Problem 4.5 — Bandwidth and Time Duration

Calculate the bandwidth W and time duration T_s of the signals in Problems 4.1 and 4.2 using the definitions:

$$W^2 = \frac{\int_{-\infty}^{\infty} \dot{x}^2(t)dt}{E}$$

$$T_s^2 = \frac{\int_{-\infty}^{\infty} t^2 x^2(t)dt}{E}$$

$$E = \int_{-\infty}^{\infty} x^2(t)dt$$

Unit 5 – Discrete-Time Systems

Problem 5.1 — Calculation of Z-transforms

(a) A sinusoidal signal $x(t) = \sin(2\pi t)$ is sampled with a sampling period T to give

$$x[n] = x(nT)$$

Calculate the Z-transform of $x[n]$ for the following values of T:

$$T = 1, \quad 1/2, \quad 1/3, \quad 1/4$$

(b) Repeat part (a) for

$$x(t) = e^{-t}\sin(2\pi t)$$

Problem 5.2 — Calculation of Inverse Z-transforms

Find the inverse Z-transform of the following signals by

- Partial fractions, using pencil-and-paper.
- Long division
- Matlab

(a)

$$X(z) = \frac{z^2}{z^2 - 1}$$

(b)

$$X(z) = \frac{z}{z^2 - 1}$$

(c)

$$X(z) = \frac{z^2 - z}{6z^3 - 11z + 6z - 1}$$

(d)

$$X(z) = \frac{2z^2}{2z^2 - 2z + 1}$$

Problem 5.3 — Solution of difference equation

A discrete-time system is characterized by the following difference equation:

$$2y[n+2] - y[n+1] - y[n] = 2x[n-1]$$

(a) Find the discrete-time transfer function $H(z) = Y(z)/X(z)$ of the system.
(b) Find the "unit response", i.e., the response to $x[n] = \delta[n]$

- Analytically, using the transfer function calculated in (a) and the Z-transform of the input.

- Numerically, using the difference equation directly.

(c) Find the (discrete-time) step response, i.e. the response to $x[n] = 1$

- Analytically, using the transfer function calculated in (a) and the Z-transform of the input.
- Numerically, using the difference equation directly.

(d) Find the response to the input

$$x[n] = \begin{cases} 1, & n = 0,\ 1,\ 2,\ 3 \\ 0, & \text{otherwise} \end{cases}$$

- Analytically, using the transfer function calculated in (a) and the Z-transform of the input.
- Numerically, using the difference equation directly.
- Approximately, by convolution of the unit response with the input.

Problem 5.4 — Digitizing analog filters.

Calculate the (sampled-and-held) discrete-time equivalents of the filters having the following analog transfer functions:

- By pencil and paper calculation
- Using the M-file af2df.m

(a)

$$H(s) = \frac{1}{(s+1)^2}$$

(b)

$$H(s) = \frac{s}{s^2 + 2s + 2}$$

The sampling interval is $T = .05$ sec.