Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Number	ELE 532
Course Title	Signals and Systems I
Semester/Year	Fall 2025
Instructor	Beheshti, Soosan
TA Name	Punya Cheema

Lab/Tutorial Report No.	04
Report Title	The Fourier Transform: Properties and Applications

Submission Date	Nov 23, 2025
Due Date	Nov 23, 2025

Student Name	Student ID	Signature
Vinci Fajardo	501239903	VF
Keunhyeok Choi	500958125	KH

1. Objective

The objective of this assignment is to utilize the Fourier Transform to analyze and study the frequency-domain characteristics of time-domain waveforms. This report will explore key properties of the Fourier Transform and apply them to solve a practical problem: designing a communications system to transmit a voice signal over a bandpass transmission channel.

2. Methodology

The laboratory exercises were conducted using MATLAB to investigate the properties of the Fourier Transform and to simulate a communication system. The experiments were divided into two main parts:

A. Properties of the Fourier Transform

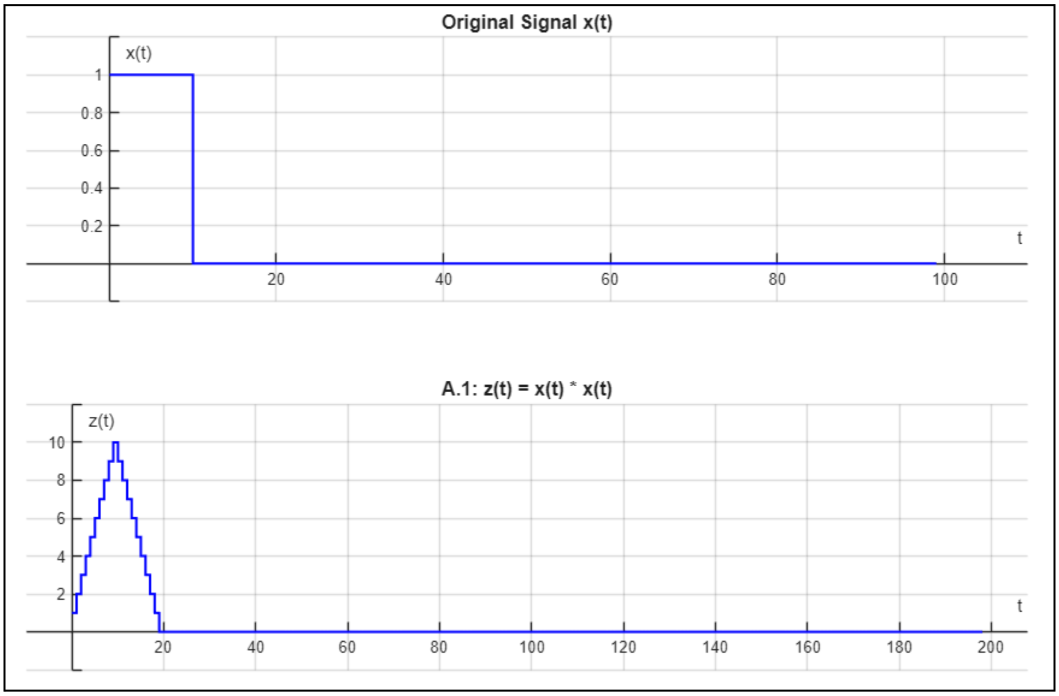
1. Signal Generation: A discrete-time rectangular pulse x(t) of width 10 samples was generated. The MATLAB functions stairs and plot were used to visualize the waveform.
2. Convolution Property: The self-convolution of the pulse, z(t) = x(t) * x(t), was computed using two methods:
    a. Time-Domain: Using the linear convolution command conv.
    b. Frequency-Domain: Computing the Fast Fourier Transform (FFT) of the signal, squaring the spectrum $Z(\omega) = X(\omega) \cdot X(\omega)$ and applying the Inverse FFT (ifft). The results were compared to verify the convolution theorem.
3. Time-Scaling Property: The spectral characteristics of the pulse were analyzed by varying the pulse width (P=5, 10, 25). The magnitude and phase spectra were plotted to observe the inverse relationship between time-domain duration and frequency-domain bandwidth.
4. Frequency-Shifting Property: The pulse x(t) was multiplied by complex exponentials ($e^{j\omega_0 t}, e^{-j\omega_0 t}$) and a cosine wave ($cos(\omega_0 t)$) with $\omega_0 = \frac{\pi}{3}$. The resulting spectra were plotted to demonstrate the translation of frequency components.

B. Application: Speech Transmission System A complete Amplitude Modulation (AM) communication system was designed to transmit a baseband speech signal (xspeech) over a bandpass channel (hChannel).

1. Spectral Analysis: The frequency content of the original speech signal and the frequency response of the transmission channel were analyzed using the FFT to identify bandwidth mismatches.
2. Transmitter Design (Coder):
   a. Filtering: The speech signal was passed through a 2000 Hz Low-Pass Filter (hLPF2000) to limit its bandwidth to fit the channel requirements.
   b. Modulation: The filtered signal was modulated using a carrier frequency (F_c) chosen to center the signal within the channel's passband (approx. 2–6 kHz). This shifted the spectrum to the appropriate transmission frequency.
3. Channel Simulation: The transmission was simulated by convolving the modulated signal with the channel impulse response (hChannel).
4. Receiver Design (Decoder):
   a. Demodulation**:** The received signal was multiplied by a local carrier synchronous with the transmitter (Coherent Detection) to shift the spectrum back to the baseband.
   b. Reconstruction: A final Low-Pass Filter (hLPF2500) was applied to remove high-frequency replicas generated during demodulation and recover the original audio signal.
5. Verification: The recovered signal was compared to the original signal in both the time and frequency domains, and auditory verification was performed using the sound command.

3. Results and Analysis

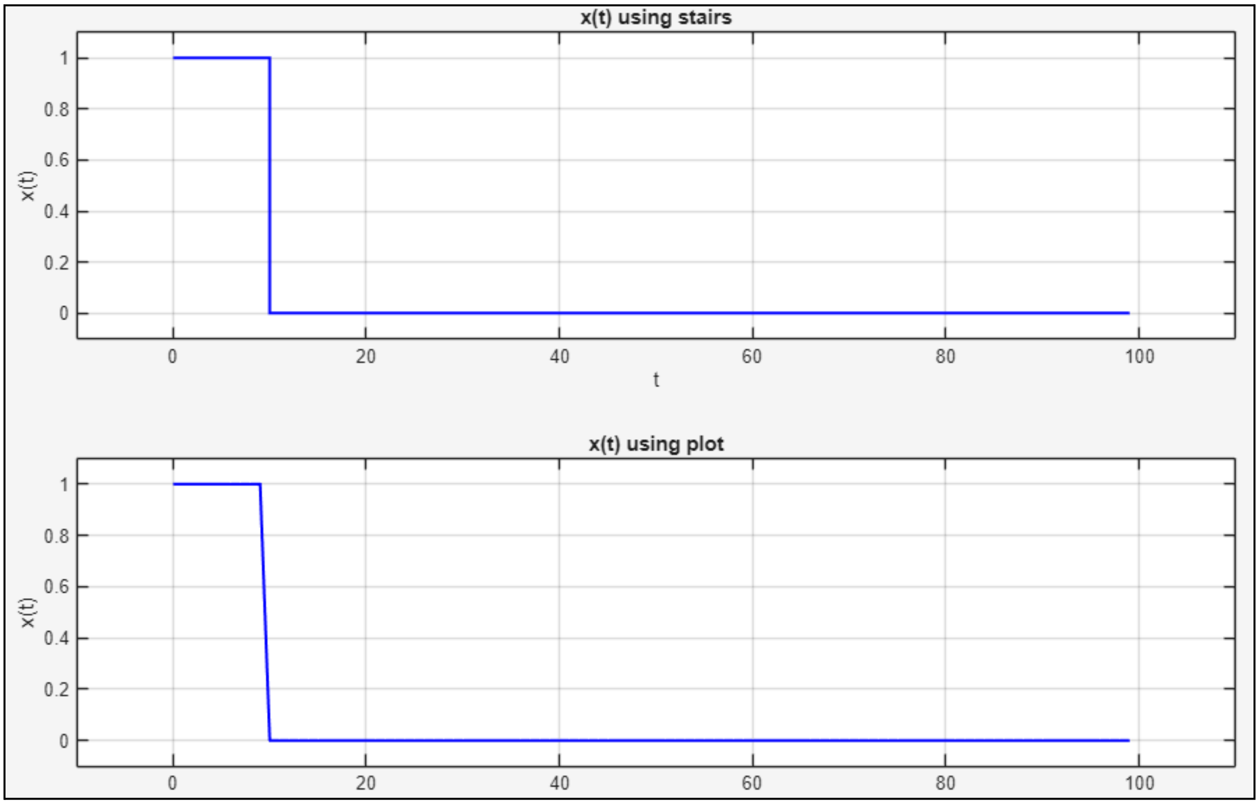
A.1



[Figure 1: Plots of x(t) and z(t) = x(t) * x(t).]

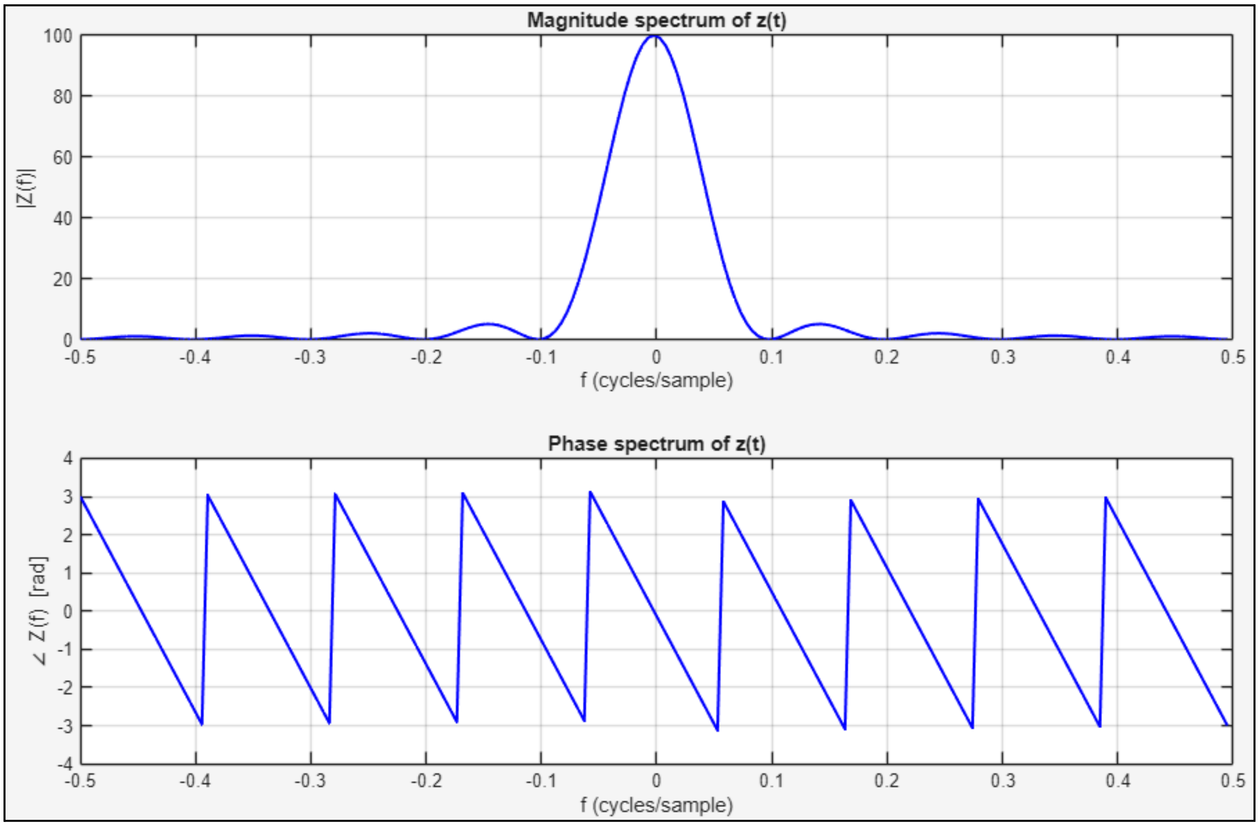
The convolution z(t)=x(t)*x(t) produces a triangular waveform whose support is twice the length of the original rectangular pulse. The linear rise and fall of z(t) reflect the increasing and then decreasing overlap between the two shifted copies of x(t) during convolution. The peak of z(t)occurs at the midpoint of the overlap interval, where the overlap between the two pulses is maximal. The resulting shape and duration therefore match the expected analytic form of the autocorrelation-like convolution of a finite-width rectangular pulse.

A.2



[Figure 2: Plots of x(t) using *stairs* and *plot* command in MATLAB.]

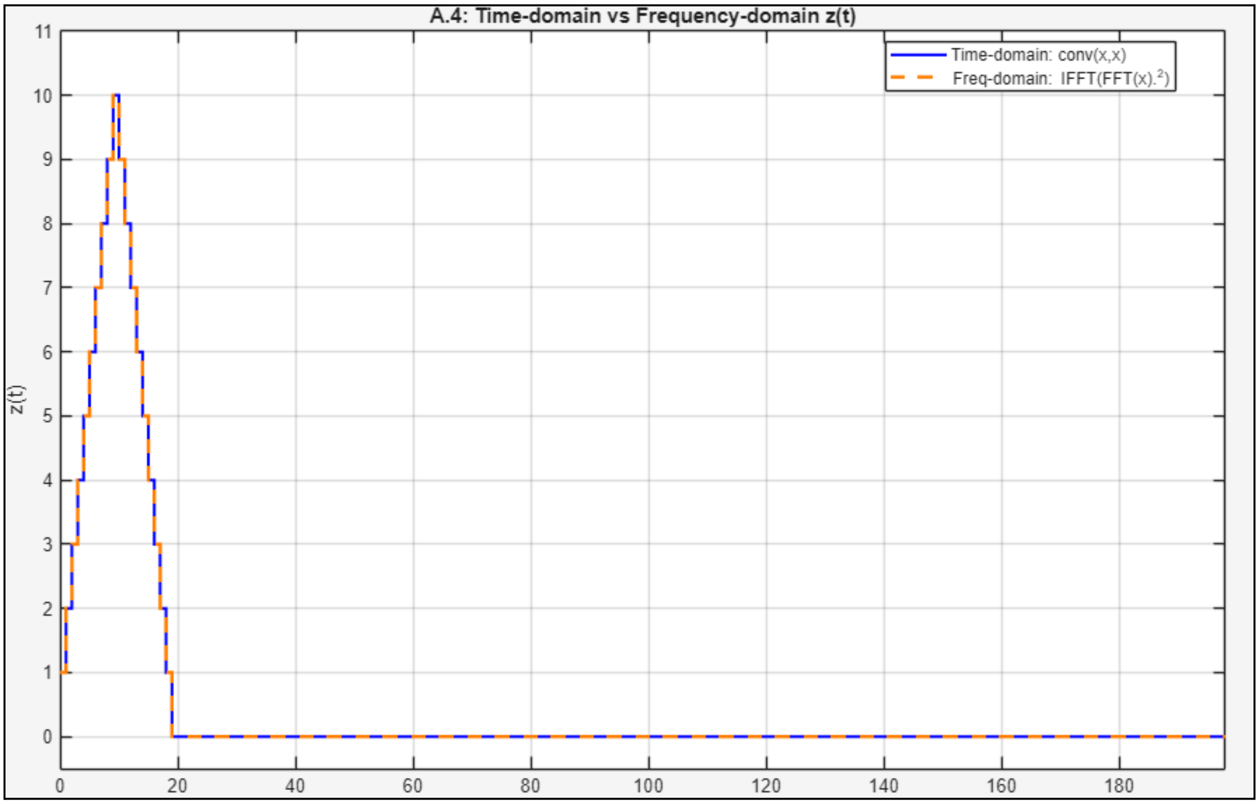
A.3.



[Figure 3: Magnitude and Phase Spectrum of z(t)]

The magnitude spectrum of z(t) exhibits the expected low-pass structure produced by convolving a finite rectangular pulse with itself. Because z(t) is a wider, smoother signal than x(t), its Fourier Transform is correspondingly more concentrated around zero frequency, with a dominant central lobe and secondary side-lobes that decay symmetrically. The phase spectrum shows a linear, sawtooth-like progression across frequency, reflecting the real and even symmetry of z(t); phase discontinuities arise from numerical wrapping at $\pm\pi$. Together, the spectra are consistent with the Fourier characteristics of a triangular time-domain waveform.

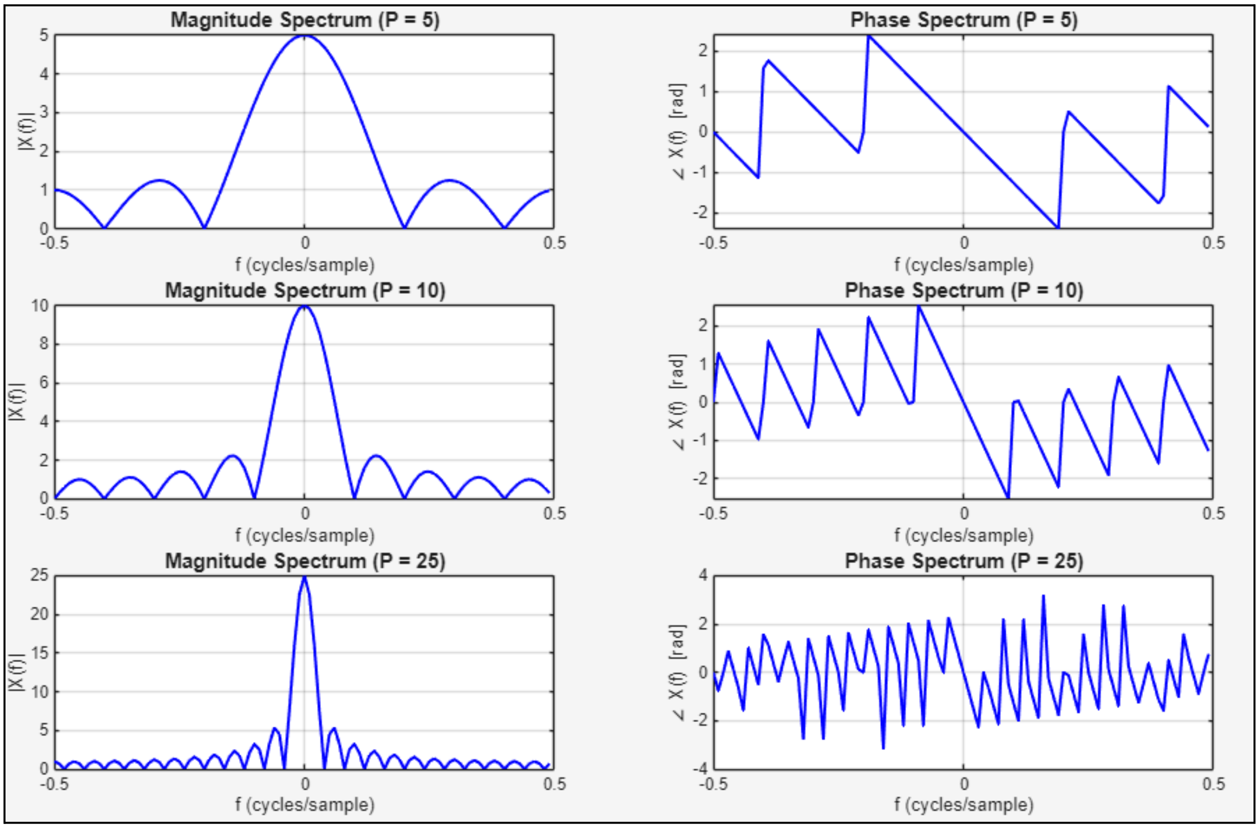
A.4.



[Figure 4: Time-Domain vs. Frequency-Domain Operations of z(t)]

The numerical convolution of x(t)x and the inverse–FFT reconstruction of Z(ω)=X(ω)X(ω) produce identical waveforms because the Fourier Transform maps convolution in the time domain to pointwise multiplication in the frequency domain while preserving the full amplitude-phase structure of the signal under inversion. The complete agreement between the analytic expression for z(t) from A.1, the time-domain convolution, and the frequency-domain method therefore demonstrates the Convolution Theorem.

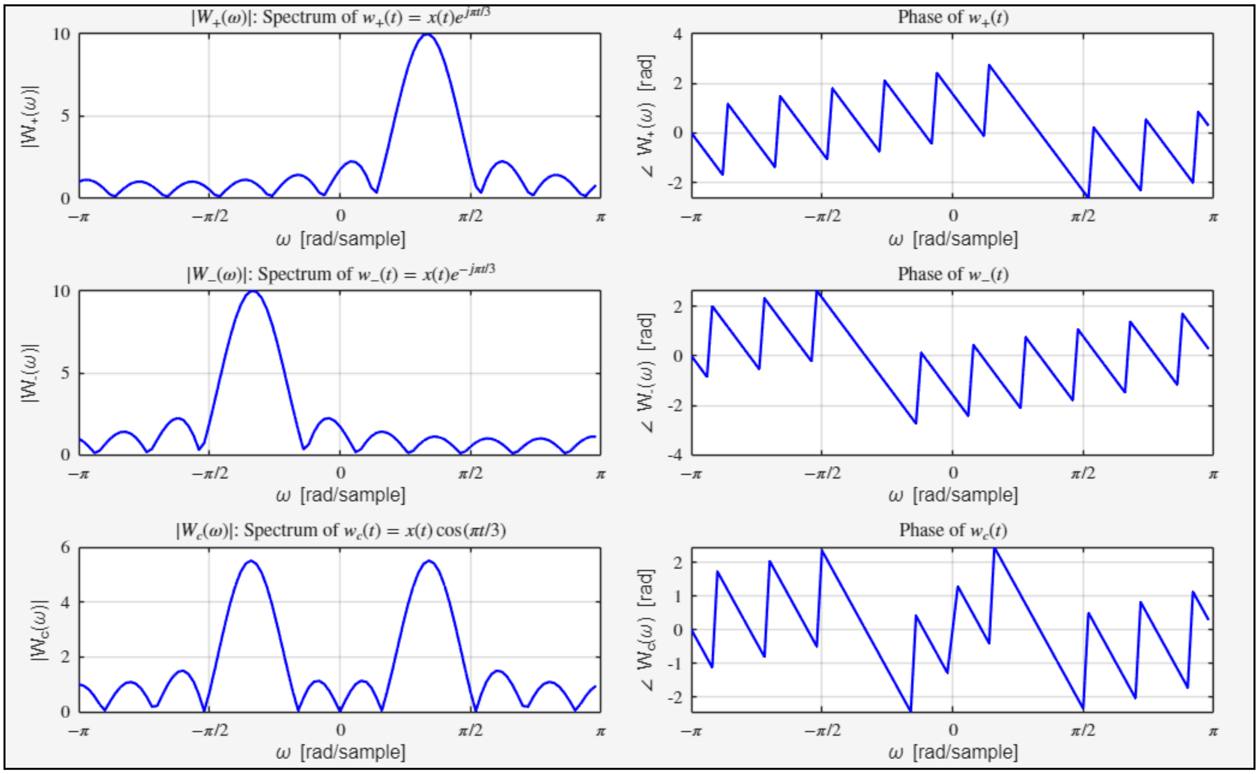
A.5.



[Figure 5: Magnitude and Phase Spectrums of x(t) at different pulse widths]

Reducing the pulse width spreads its Fourier spectrum, and increasing the pulse width concentrates it. The narrower pulse (width 5) produces a much broader magnitude spectrum with slower phase variation, while the wider pulse (width 25) exhibits a correspondingly narrower main lobe and more rapid phase transitions around its spectral zeros. These differences follow directly from the inverse relationship between temporal duration and spectral extent: compressing a signal in time expands its frequency-domain representation, and stretching it in time contracts the spectrum. The comparison across the three pulse widths therefore demonstrates the time–scaling property of the Fourier Transform.

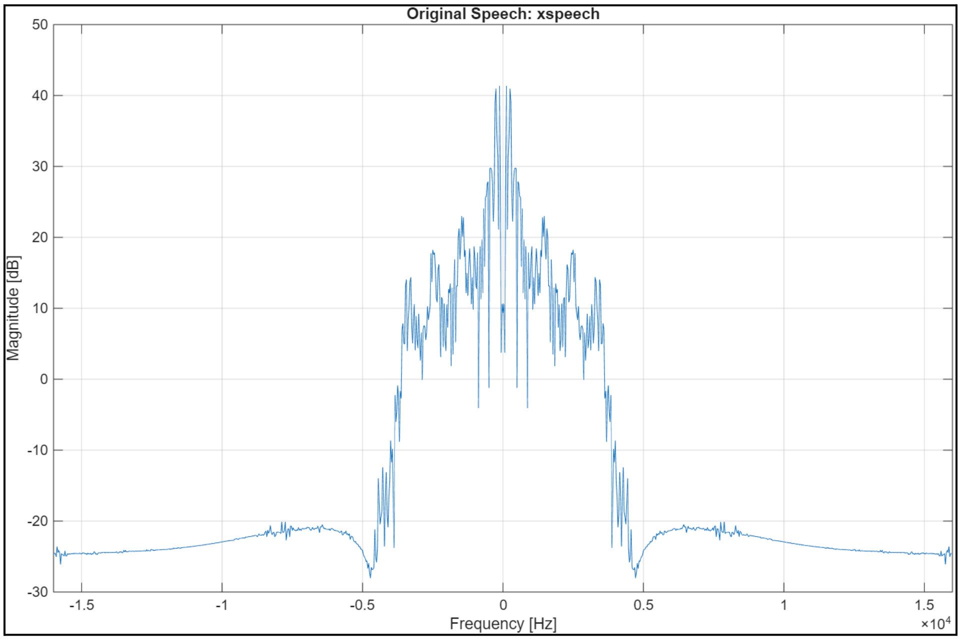
A.6.



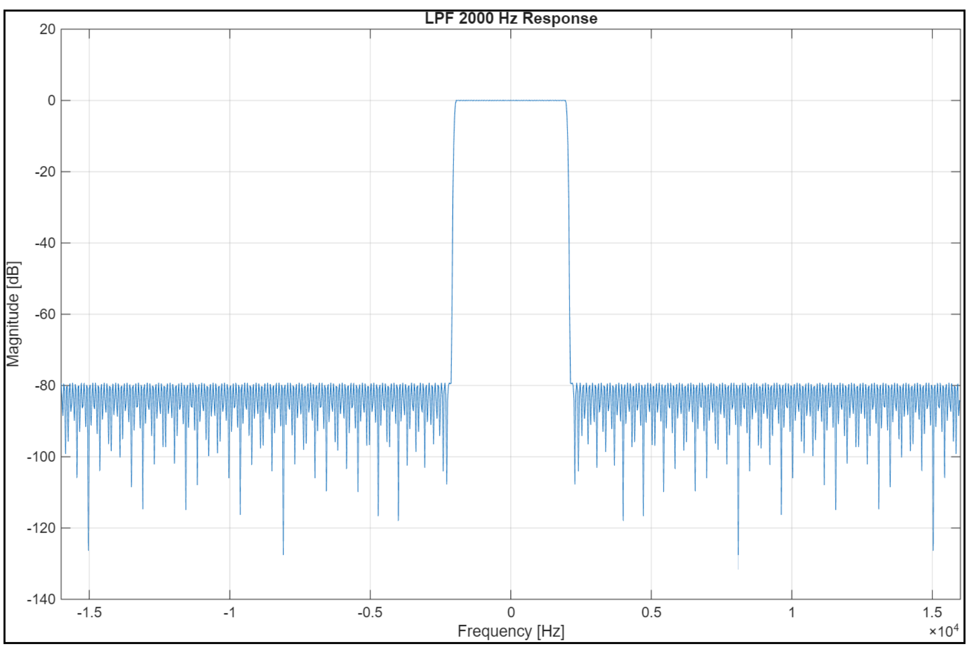
[Figure 6: Magnitude and Phase Spectrums of $\omega_+(t)$, $\omega_-(t)$ and $\omega_c(t)$]

Multiplying the pulse x(t) by the complex exponential e^j(π/3)t shifts its spectrum upward by π/3, while multiplying by e^-j(π/3)t shifts the spectrum downward by the same amount. The magnitude spectra of $\omega_+(t)$ and $\omega_-(t)$ therefore match the original spectrum from A.3 but are displaced along the frequency axis, whereas the phase spectra exhibit correspondingly translated phase structure. For $\omega_c(t)$=x(t)cos(π/3 t), the cosine modulation produces two symmetric shifts because cos(Ωt) = ½(e^jΩt + e^-jΩt), yielding two copies of the original spectrum centred at +π/3 and −π/3. These outcomes demonstrate the frequency–shifting property of the Fourier Transform, which states that multiplication by a complex exponential in time translates the spectrum without altering its shape.

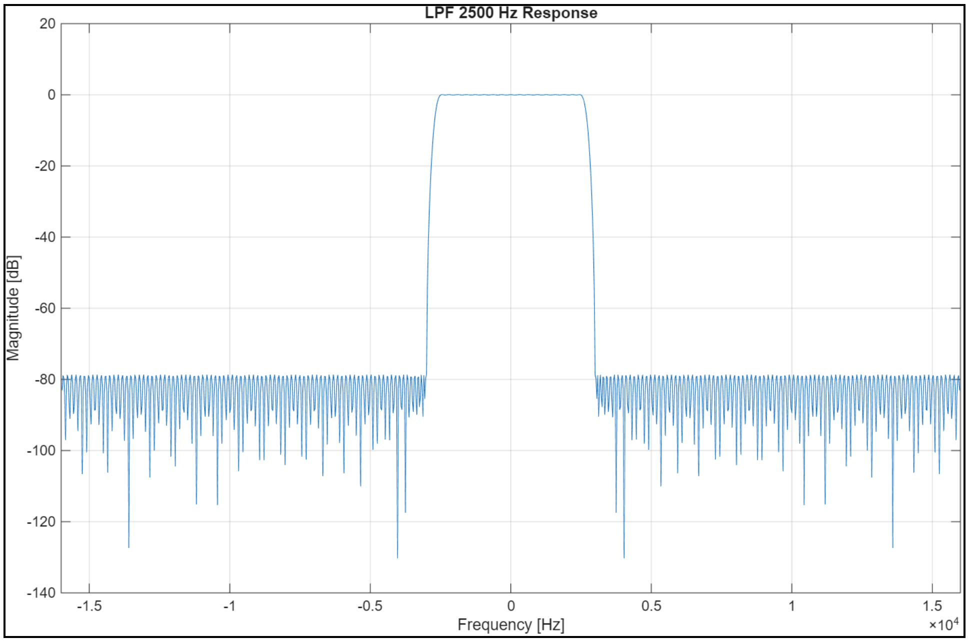
B.1



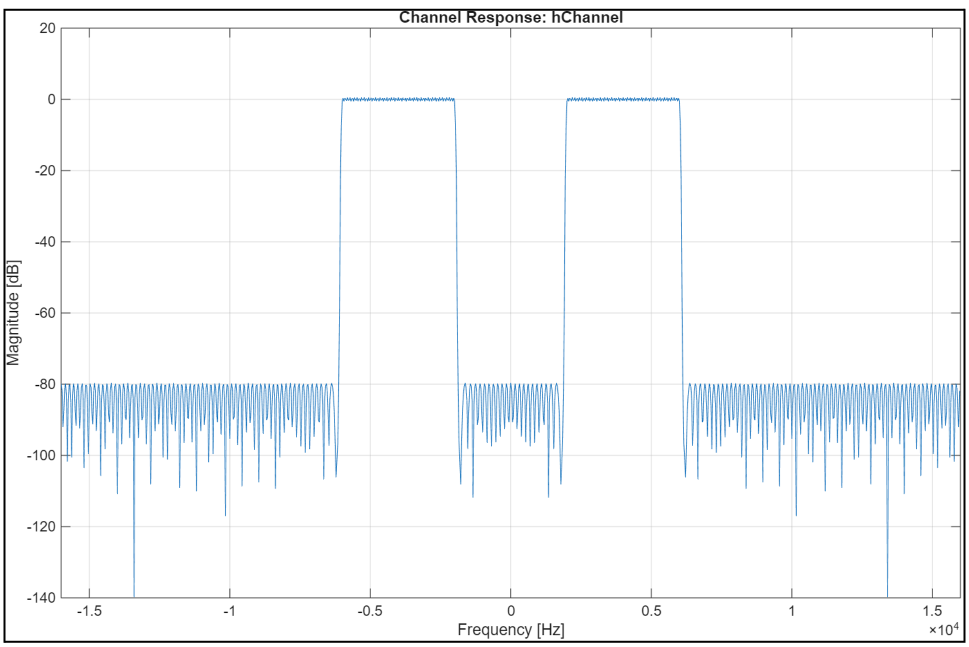
[Figure 7: Plot of xspeech using MagSpect]



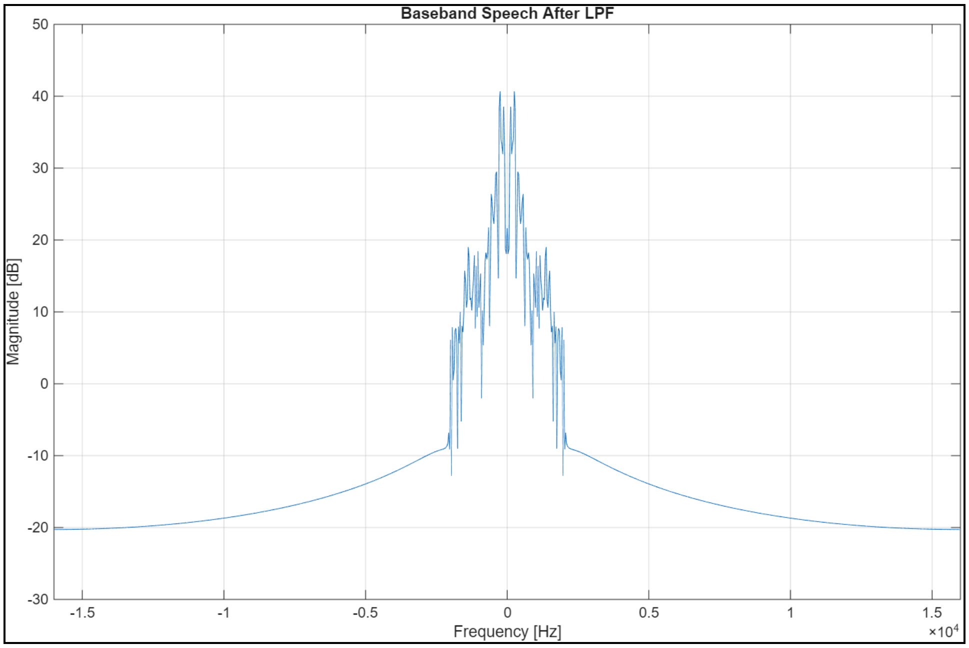
[Figure 8: Plot of hLPF2000 using MagSpect]



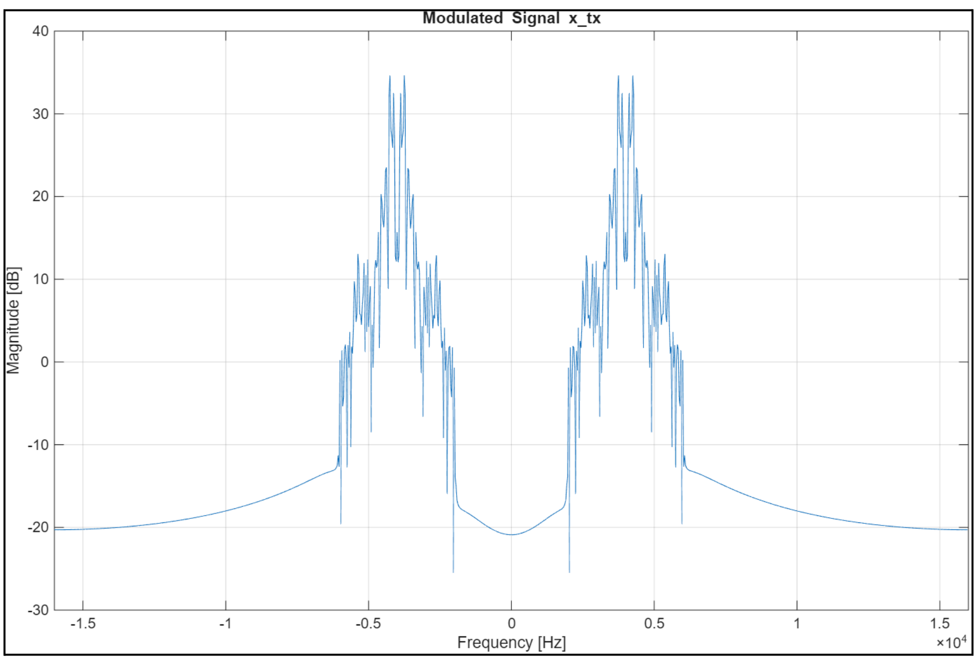
[Figure 9: Plot of hLPF2500 using MagSpect]



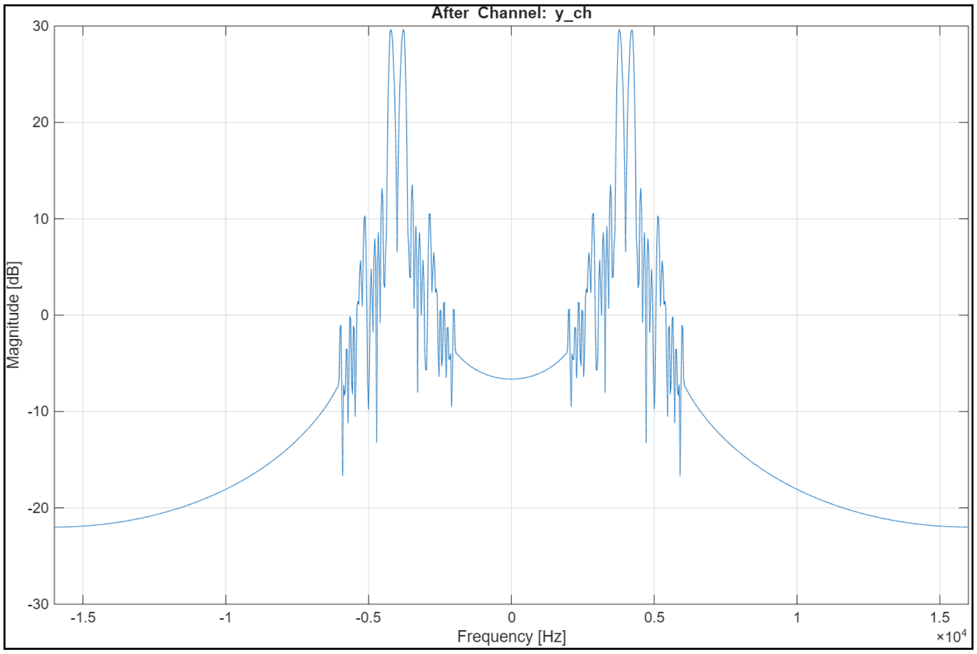
[Figure 10: Plot of hChannel using MagSpect]



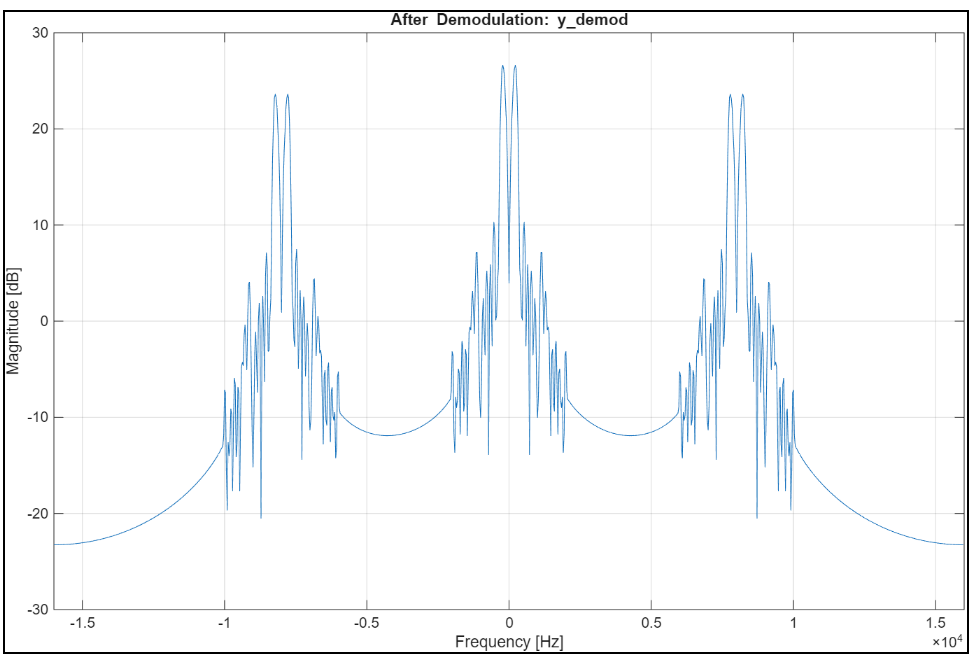
[Figure 11: Plot of xspeech after LPF]



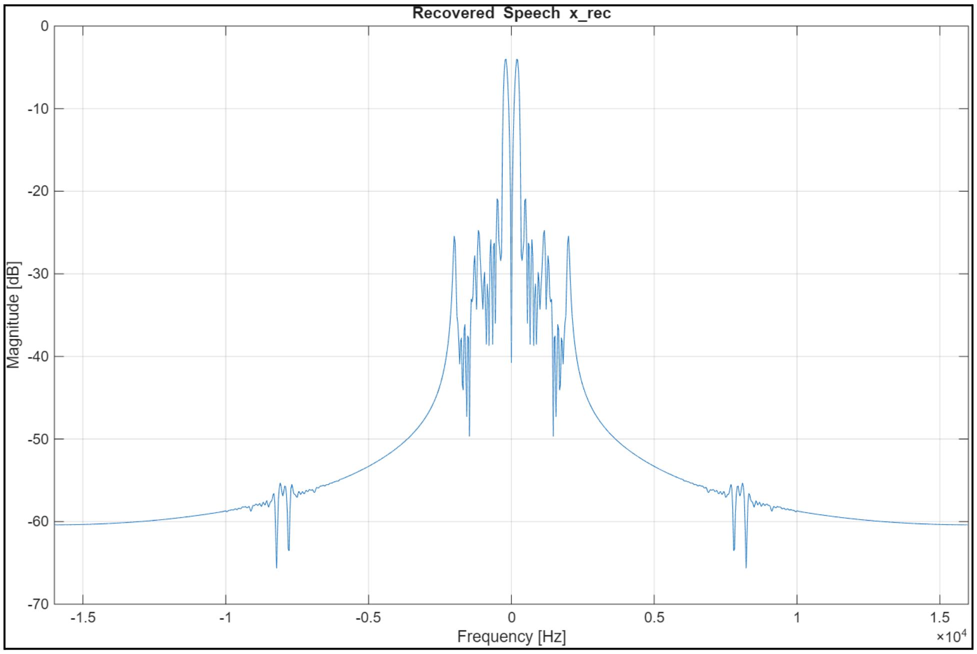
[Figure 12: Plot of xspeech after modulation]



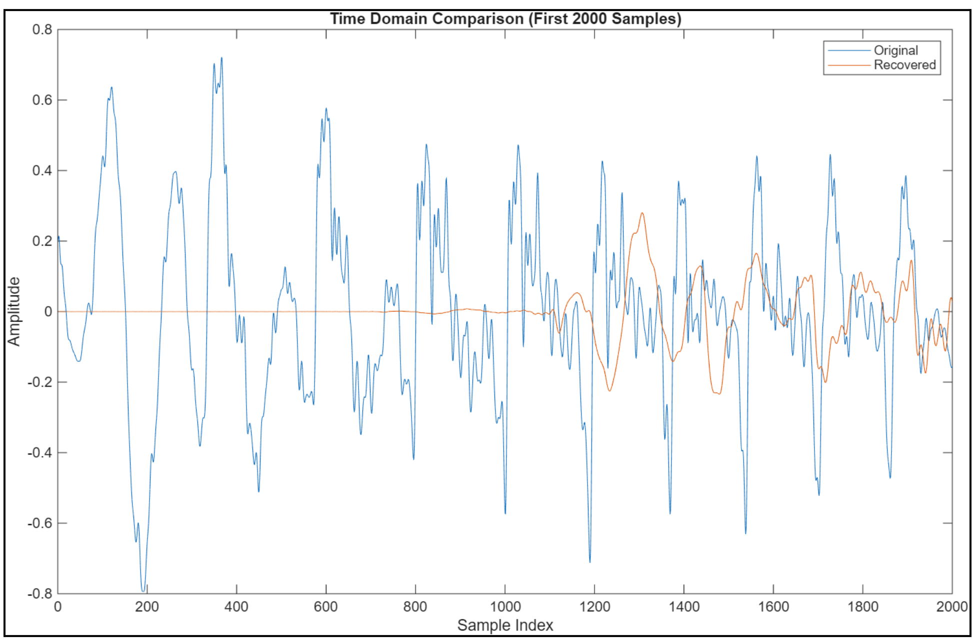
[Figure 13: plot of xspeech after passing through hChannel]



[Figure 14: plot of xpseech after demodulation]



[Figure 15: plot of recovered xspeech]



[Figure 16: Comparison between original vs. recovered.]

Block Diagram:

Original Speech → Low-Pass Filter (Baseband Shaping) → Modulator → Channel → Demodulator → Low-Pass Filter (Recovery) → Recovered Speech.

The original speech is first passed through a low-pass filter to restrict its bandwidth to the range that can safely be transmitted through the channel after modulation. This band-limited signal is then multiplied by the carrier during modulation, shifting its entire spectrum upward so it occupies the 2–6 kHz band passed by the channel. The channel eliminates all frequency components outside this band, so only the shifted speech survives transmission. After the channel, the signal is demodulated by multiplying with the same carrier, which returns the surviving band back down to baseband. A final low-pass filter is applied to isolate the baseband speech and remove the high-frequency terms generated during demodulation, producing the recovered version of the original signal.

4. Conclusion

This lab successfully demonstrated the key properties of the Fourier Transform. The Convolution, Time-Scaling, and Frequency-Shifting properties were observed experimentally. Furthermore, these principles were applied to a practical engineering problem. By using filtering and modulation (based on the Frequency-Shifting property), a baseband speech signal was successfully transmitted over a bandpass channel and subsequently recovered, validating the design and the theoretical principles of the Fourier Transform.

5. References

1. B. P. Lathi, *Linear Systems and Signals*, 2nd Edition, Oxford University Press. (Chapter 7)
2. ELE 532 Lab Manual 4: The Fourier Transform, Department of Electrical, Computer, & Biomedical Engineering, Toronto Metropolitan University

6. Appendix: MATLAB Code

```
%% ELE532 - Lab: Fourier Transform and its Properties

% A.1 — Compute z(t) = x(t) * x(t)

clear; close all; clc

% ==========================

% A.0 — Signal per handout

% ==========================

N          = 100;

PulseWidth = 10;

t          = 0:(N-1);

x          = [ones(1, PulseWidth), zeros(1, N - PulseWidth)];

% ==========================

% A.1 — Convolution

% ==========================

z  = conv(x, x);

tz = 0:(numel(z)-1);

% ==========================

% One figure, centered axes

% ==========================

figure('Name','ELE532 Lab A.1','NumberTitle','off','Color','w');

% ---- Top: Original Signal x(t)

subplot(2,1,1);

stairs(t, x, 'b', 'LineWidth', 1.5);

grid on;

ax = gca;

ax.XAxisLocation = 'origin';   % x-axis centered
```

```
ax.YAxisLocation = 'origin';    % y-axis centered

ax.Box = 'off';                 % remove box outline

xlim([-10 110]);

ylim([-0.2 1.2]);

xlabel('t','FontSize',11);

ylabel('x(t)','FontSize',11);

title('Original Signal x(t)','FontSize',12);

% ---- Bottom: Convolution Result z(t)

subplot(2,1,2);

stairs(tz, z, 'b', 'LineWidth', 1.5);

grid on;

ax = gca;

ax.XAxisLocation = 'origin';    % x-axis centered

ax.YAxisLocation = 'origin';    % y-axis centered

ax.Box = 'off';

xlim([-10 tz(end)+10]);

ylim([-2 max(z)+2]);

xlabel('t','FontSize',11);

ylabel('z(t)','FontSize',11);

title('A.1: z(t) = x(t) * x(t)','FontSize',12);

% ==========================================================

% A.2 — Using MATLAB, calculate Z(ω) = X(ω)X(ω)

% ==========================================================

% Recreate x(t) per handout (safe even if defined earlier)

N          = 100;

PulseWidth = 10;
```

```
t           = 0:(N-1);
x           = [ones(1,PulseWidth), zeros(1,N-PulseWidth)];
% Compute Z(ω) in frequency domain
Xf = fft(x);
Zf = Xf .* Xf;
% Compare stairs vs plot (single figure)
figA2 = figure('Name','ELE532 Lab — A.2','NumberTitle','off');
subplot(2,1,1);
stairs(t, x, 'b', 'LineWidth', 1.5);
grid on; axis([-10 110 -0.1 1.1]);
xlabel('t'); ylabel('x(t)'); title('x(t) using stairs');
subplot(2,1,2);
plot(t, x, 'b', 'LineWidth', 1.5);
grid on; axis([-10 110 -0.1 1.1]);
xlabel('t'); ylabel('x(t)'); title('x(t) using plot');
drawnow;
% ============================================================
% A.3 — Plot the magnitude- and phase-spectra of z(t)
% ============================================================
% Ensure x(t) and z(t) exist (rebuild if needed)
if ~exist('x','var') || ~exist('t','var')
   N = 100; PulseWidth = 10; t = 0:(N-1);
   x = [ones(1,PulseWidth), zeros(1,N-PulseWidth)];
end
if ~exist('z','var')
   z = conv(x,x);
```

```
end

% FFT of z(t)

M  = numel(z);

Zf = fft(z, M);

f  = (-M/2 : M/2-1) / M;    % cycles/sample

% One figure, two subplots

figure('Name','ELE532 Lab — A.3','NumberTitle','off');

subplot(2,1,1);

plot(f, fftshift(abs(Zf)), 'b', 'LineWidth', 1.5);

grid on; xlabel('f (cycles/sample)'); ylabel('|Z(f)|');

title('Magnitude spectrum of z(t)');

subplot(2,1,2);

plot(f, fftshift(angle(Zf)), 'b', 'LineWidth', 1.5);

grid on; xlabel('f (cycles/sample)'); ylabel('\angle Z(f) [rad]');

title('Phase spectrum of z(t)');

drawnow;

% ==========================================================

% A.4 — Compute z(t) using time- and frequency-domain ops

%         and compare the two time-domain results

% ==========================================================

% Recreate x(t) exactly as in the handout (safe in one script)

N          = 100;

PulseWidth = 10;

t          = 0:(N-1);

x          = [ones(1,PulseWidth), zeros(1,N-PulseWidth)];

% ----- Time-domain: linear convolution
```

```
z_td = conv(x, x);                 % length = 2*N - 1

tz   = 0:(2*N-2);                  % correct time indices for z[n]

% ----- Frequency-domain: zero-pad, multiply spectra, IFFT

L    = 2*N - 1;                    % minimum padding for linear conv

X    = fft(x, L);

Z    = X .* X;                     % Z(ω) = X(ω) X(ω)

z_fd = real(ifft(Z));              % numerical round-off -> take real

z_fd = z_fd(1:2*N-1);              % match linear-convolution length

% ----- Plot: both results overlaid for comparison

figure('Name','ELE532 Lab — A.4','NumberTitle','off');

% Plot time-domain result first (blue)

stairs(tz, z_td, 'b', 'LineWidth', 1.5); hold on;

% Overlay frequency-domain result (orange, dashed)

stairs(tz, z_fd, '--', 'Color', [1 0.5 0], 'LineWidth', 1.8);

grid on;

xlabel('t'); ylabel('z(t)');

title('A.4: Time-domain vs Frequency-domain z(t)');

legend('Time-domain: conv(x,x)', 'Freq-domain: IFFT(FFT(x).^2)',
'Location','best');

xlim([0, tz(end)]);

ylim([-0.5, PulseWidth+1]);

% ----- Numerical agreement report (console)

maxErr = max(abs(z_td - z_fd));

fprintf('A.4: max |z_{TD} - z_{FD}| = %.3e\n', maxErr);

% ===========================================================

% A.5 — Pulse width sweep: magnitude- and phase-spectra
```

```
% ============================================================
% Setup
N   = 100;
f   = (-N/2 : N/2-1) / N;        % frequency vector (cycles/sample)
Pw  = [5 10 25];                 % pulse widths to test
% Compute FFTs for each pulse width
Xs = cell(numel(Pw),1);
for k = 1:numel(Pw)
   Pk = Pw(k);
   xk = [ones(1,Pk), zeros(1,N-Pk)];
   Xs{k} = fft(xk);
end
% ------------------------------------------------------------
% Single figure, 3 rows × 2 cols layout
% ------------------------------------------------------------
figure('Name','ELE532 Lab — A.5','NumberTitle','off');
for k = 1:numel(Pw)
   % Magnitude spectrum (left column)
   subplot(3,2,(k-1)*2 + 1);
   plot(f, fftshift(abs(Xs{k})), 'b', 'LineWidth', 1.5);
   grid on;
   xlabel('f (cycles/sample)', 'FontSize', 10);
   ylabel('|X(f)|', 'FontSize', 10);
   title(sprintf('Magnitude Spectrum (P = %d)', Pw(k)), 'FontSize', 11);
   % Phase spectrum (right column)
   subplot(3,2,(k-1)*2 + 2);
```

```
    plot(f, fftshift(angle(Xs{k})), 'b', 'LineWidth', 1.5);
    grid on;
    xlabel('f (cycles/sample)', 'FontSize', 10);
    ylabel('\angle X(f) [rad]', 'FontSize', 10);
    title(sprintf('Phase Spectrum (P = %d)', Pw(k)), 'FontSize', 11);
end
drawnow;
% ==========================================================
% A.6 — Spectra of w_+(t), w_-(t), and w_c(t)  (aligned axes)
% ==========================================================
% Base pulse (width 10, N=100)
if ~exist('x','var') || numel(x)~=100
    N = 100; PulseWidth = 10;
    t = 0:(N-1);
    x = [ones(1,PulseWidth), zeros(1,N-PulseWidth)];
else
    N = numel(x); t = 0:(N-1);
end
Omega0 = pi/3;                       % rad/sample
% Modulated signals
w_plus  = x .* exp( 1j*Omega0*t);
w_minus = x .* exp(-1j*Omega0*t);
w_cos   = x .* cos(Omega0*t);
% FFTs and frequency axis
Wplus  = fft(w_plus);
Wminus = fft(w_minus);
```

```
Wcos   = fft(w_cos);
w      = 2*pi*((-N/2):(N/2-1))/N;    % ω in rad/sample
% Figure and grid
fig = figure('Name','ELE532 Lab — A.6','NumberTitle','off');
tl  = tiledlayout(fig,3,2,'Padding','compact','TileSpacing','compact');
ax = gobjects(6,1);
% Row 1: w_+(t)
ax(1) = nexttile(tl);
plot(w, fftshift(abs(Wplus)), 'b', 'LineWidth', 1.6); grid on
ylabel('|W_+(\omega)|'); xlabel('\omega [rad/sample]');
title('$|W_+(\omega)|$: Spectrum of $w_+(t)=x(t)e^{j\pi t/3}$', ...
     'Interpreter','latex','FontSize',12);
ax(2) = nexttile(tl);
plot(w, fftshift(angle(Wplus)), 'b', 'LineWidth', 1.6); grid on
ylabel('\angle W_+(\omega) [rad]'); xlabel('\omega [rad/sample]');
title('Phase of $w_+(t)$', 'Interpreter','latex','FontSize',12);
% Row 2: w_-(t)
ax(3) = nexttile(tl);
plot(w, fftshift(abs(Wminus)), 'b', 'LineWidth', 1.6); grid on
ylabel('|W_-(\omega)|'); xlabel('\omega [rad/sample]');
title('$|W_-(\omega)|$: Spectrum of $w_-(t)=x(t)e^{-j\pi t/3}$', ...
     'Interpreter','latex','FontSize',12);
ax(4) = nexttile(tl);
plot(w, fftshift(angle(Wminus)), 'b', 'LineWidth', 1.6); grid on
ylabel('\angle W_-(\omega) [rad]'); xlabel('\omega [rad/sample]');
title('Phase of $w_-(t)$', 'Interpreter','latex','FontSize',12);
```

```
% Row 3: w_c(t)
ax(5) = nexttile(tl);
plot(w, fftshift(abs(Wcos)), 'b', 'LineWidth', 1.6); grid on
ylabel('|W_c(\omega)|'); xlabel('\omega [rad/sample]');
title('$|W_c(\omega)|$: Spectrum of $w_c(t)=x(t)\cos(\pi t/3)$', ...
     'Interpreter','latex','FontSize',12);
ax(6) = nexttile(tl);
plot(w, fftshift(angle(Wcos)), 'b', 'LineWidth', 1.6); grid on
ylabel('\angle W_c(\omega) [rad]'); xlabel('\omega [rad/sample]');
title('Phase of $w_c(t)$', 'Interpreter','latex','FontSize',12);
% Enforce identical x-axis range and π-ticks on all subplots
for k = 1:6
   set(ax(k), 'XLim', [-pi pi], ...
              'XTick', [-pi -pi/2 0 pi/2 pi], ...
              'TickLabelInterpreter','latex', ...
              'FontSize', 11);
   ax(k).XTickLabel = {'$-\pi$','$-\pi/2$','0','$\pi/2$','$\pi$'};
end
linkaxes(ax,'x');
Drawnow;
```

```matlab
% ==========================
% B.1
% ==========================
load Lab4_Data.mat
Fs = 32000;
figure; MagSpect(xspeech);       title('Original Speech: xspeech');
figure; MagSpect(hLPF2000);      title('LPF 2000 Hz Response');
figure; MagSpect(hLPF2500);      title('LPF 2500 Hz Response');
figure; MagSpect(hChannel);      title('Channel Response: hChannel');
hLPF = hLPF2000;
x_base = filter(hLPF,1,xspeech);
figure; MagSpect(x_base);        title('Baseband Speech After LPF');
Fc = 4000;
c  = osc(Fc, length(x_base));
x_tx   = x_base .* c;
figure; MagSpect(x_tx);          title('Modulated Signal x\_tx');
y_ch   = filter(hChannel,1,x_tx);
figure; MagSpect(y_ch);          title('After Channel: y\_ch');
y_demod = y_ch .* c;
figure; MagSpect(y_demod);       title('After Demodulation: y\_demod');
x_rec  = filter(hLPF,1,y_demod);
figure; MagSpect(x_rec);         title('Recovered Speech x\_rec');
N = 2000;
figure;
plot(xspeech(1:N)); hold on;
```

```
plot(x_rec(1:N));

xlabel('Sample Index');

ylabel('Amplitude');

legend('Original','Recovered');

title('Time Domain Comparison (First 2000 Samples)');
```