Bilkent University
Electrical and Electronics Engineering Department
Fall 2026-27

# EEE-321: Signals and Systems

Lab Assignment 3

You will use the Fourier transform in two everyday applications: decoding the tones a telephone keypad sends (DTMF, Parts 1–3) and removing echoes from a recording (Part 4). Part 5 has three optional experiments that build on Parts 3 and 4.

## Part 1: DTMF Transmitter

Dual-tone multi-frequency (DTMF) is the standard way an analog telephone line tells the telephone exchange (*telefon santrali*) which number is dialled. Each key sends the sum of two cosines: one "row" frequency and one "column" frequency from Table 1.

Table 1: DTMF frequencies

| | **1209 Hz** | **1336 Hz** | **1477 Hz** | **1633 Hz** |
|---|---|---|---|---|
| **697 Hz** | 1 | 2 | 3 | A |
| **770 Hz** | 4 | 5 | 6 | B |
| **852 Hz** | 7 | 8 | 9 | C |
| **941 Hz** | * | 0 | # | D |

For example, while key 5 is held down, the phone sends

$$x(t) = \cos(2\pi\, 770\, t) + \cos(2\pi\, 1336\, t). \tag{1}$$

In this lab every key press lasts 0.25 s: the tone is on for the first 0.20 s and is followed by 0.05 s of silence. For the number 2026 the transmitted signal is therefore

$$x(t) = \begin{cases} \cos(2\pi\, 697t) + \cos(2\pi\, 1336t) & \text{for } 0 \le t < 0.20, \\ \cos(2\pi\, 941t) + \cos(2\pi\, 1336t) & \text{for } 0.25 \le t < 0.45, \\ \cos(2\pi\, 697t) + \cos(2\pi\, 1336t) & \text{for } 0.50 \le t < 0.70, \\ \cos(2\pi\, 770t) + \cos(2\pi\, 1477t) & \text{for } 0.75 \le t < 0.95, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In MATLAB we can only compute samples of $x(t)$. We use the sampling rate $f_s = 8000$ Hz (the rate used in telephony), so the samples are $x(n/f_s)$ for $n = 0, 1, \ldots, 2000K - 1$ when $K$ digits are dialled. Digit $k$ occupies the 2000 samples $n = 2000(k-1), \ldots, 2000k - 1$, and its tone is on during the first 1600 of them.

**(a)** Write a function `x = DTMFTRA(Number)` that returns the samples of the transmitted signal for the digits in `Number`, a $1\times K$ vector with entries $0, 1, \ldots, 9$, such as `[2 0 2 6]`. Use the global time $t = n/f_s$ inside the cosines, as in Eq. (2), so that all bursts at one frequency are pieces of the same cosine. Self-check: `length(DTMFTRA([2 0 2 6]))` must be 8000, and `plot((0:7999)/8000, DTMFTRA([2 0 2 6]))` must show four 0.20 s bursts separated by short silences.

**(b)** Listen to the signal for your own phone number with `soundsc(DTMFTRA(Number), 8000)`. Does it sound familiar? Can you hear the silences between the digits?

# Part 2: Fourier Transform of a Tone Burst

At the exchange, the receiver must find which frequencies are present in the incoming signal. The Fourier transform of $x(t)$,

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt, \tag{3}$$

tells us how much of each complex exponential $e^{j\omega t}$ is needed to build $x(t)$. Before using MATLAB, find what the Fourier transform of a single tone burst looks like. You may take the transform pairs in (a) from the textbook. In (b), use the properties of the Fourier transform rather than evaluating the integral.

**(a)** Write $X(\omega)$ for $x(t) = \cos(\omega_0 t)$, where $\omega_0 = 2\pi f_0$, and for $x(t) = \mathrm{rect}(t/T)$, where $\mathrm{rect}(t) = 1$ for $|t| < \frac{1}{2}$ and $\mathrm{rect}(t) = 0$ for $|t| > \frac{1}{2}$. At which nonzero frequencies is the second transform zero? Evaluate its value at $\omega = 0$ by taking a limit.

**(b)** Let $x(t) = \cos(\omega_0 t)\ \mathrm{rect}\Big(\frac{t-t_0}{T}\Big)$, a tone burst of length $T$ centred at $t_0$. Only the pulse is shifted; the cosine is still $\cos(\omega_0 t)$. Find $X(\omega)$. Name each property you use.

**(c)** Take $f_0 = 697$ Hz and $T = 0.20$ s. Near $\omega = \omega_0$ the term of $X(\omega)$ centred at $-\omega_0$ is negligible. Sketch $|X(\omega)|$ against $f = \omega/2\pi$ for $670 \le f \le 725$ Hz. Mark the peak value and the first zero on each side of the peak. What is the null-to-null width of the peak in Hz? What would it be for $T = 0.10$ s?

**(d)** Does $|X(\omega)|$ near $\omega_0$ depend on $t_0$? **Optional extension.** Now add a second burst at the same frequency, $\cos(\omega_0 t)\ \mathrm{rect}\Big(\frac{t-t_1}{T}\Big)$ with $t_1 \neq t_0$. Show that near $\omega_0$ the magnitude of the sum is $2\left|\cos\Big((\omega-\omega_0)(t_1-t_0)/2\Big)\right|$ times the magnitude of one burst. Does it depend on $t_0$ and $t_1$ separately, or only on $t_1 - t_0$? What does this suggest about what the magnitude spectrum of a whole dialled number can tell you about the order of its digits?

# Part 3: DTMF Receiver

You will now play the receiver for a 4-digit number that you do not know. Run the code below. It draws the digits at random using your student ID (the same ID always gives the same number). The signal has no added noise. Do not display `secret` until Part 3(d) asks you to.

```
ID = 12345678;                % <-- your own 8-digit student ID
rng(ID);                      % the random numbers below now depend on your ID
secret = randi([0 9], 1, 4);  % a 4-digit number (keep the semicolon!)
x = DTMFTRA(secret);
fs = 8000;
N = length(x);
t = (0:N-1)/fs;               % time axis (s)
[X, omega] = FT(x, fs);       % X(omega) on the frequency grid omega (rad/s)
f = omega/(2*pi);             % the same grid in Hz
```

FT returns very close approximations to samples of $X(\omega)$ of Eq. (3) (see Appendix A), so you can compare its output directly with your results from Part 2. Its grid points are $f_s/N = 1$ Hz apart and include every whole number of hertz, so each DTMF frequency is exactly a grid point.

**(a)** Listen to the received signal with `soundsc(x, fs)`. How many key presses do you hear? Can you tell which presses are the same digit?

**(b)** Plot $|X|$ against $f$ for $0 \le f \le 2000$ Hz. At which frequencies are the peaks, and which DTMF frequencies are they? Why can four digits never give eight peaks here, and why are some peaks taller than others? Can you tell the dialled number from this plot alone?

**(c)** **Optional: Where is the order of the digits?** Compute the spectra of your number and of the same digits dialled in reverse order. Append 3 s of zeros to each signal: this does not change $X(\omega)$, but FT then samples it every 0.25 Hz instead of every 1 Hz, which is fine enough to show the shape inside a peak (see also (d)).

```
s = secret;
if isequal(s, fliplr(s)), s = [2 5 3 3]; end  % same backwards? use 2533
pad = zeros(1, 3*N);                          % 3 s of zeros
[X0, omega4] = FT([DTMFTRA(s), pad], fs);
Xr = FT([DTMFTRA(fliplr(s)), pad], fs);
f4 = omega4/(2*pi);                           % grid in Hz, 0.25 Hz apart
```

Listen to both signals and plot $|X_0|$ and $|X_r|$ against `f4` on the same axes (use `hold on`). Do the signals sound the same? Are the magnitude spectra equal, or only close? (Print `max(abs(abs(X0) - abs(Xr)))`.) Now zoom into the peak of a frequency that two or more of the digits use, for example with `xlim([fr-10, fr+10])` where `fr` is that frequency. Why does this peak look so different from the 10 Hz wide peak of a single burst in Part 2(c)? Why does reversing the digits leave it (almost) unchanged? Use Part 2(d). Would every reordering of the digits leave $|X|$ unchanged? Use Part 2(d) to predict whether 2266 and 2626 have the same $|X|$ near 1336 Hz, then check (pad them in the same way).

**(d)** The $k$-th digit lies in $0.25(k-1) \le t < 0.25k$. Write the window $w_k(t)$, using rect, such that $x_k(t) = x(t)\, w_k(t)$ contains only the $k$-th digit. In MATLAB, build $w_k$ as a vector of zeros and ones with the same length as `x`, compute `Xk = FT(xk, fs)`, and plot $|X_k|$ against $f$ for $k = 1, \ldots, 4$ in one figure using `subplot(2,2,k)`. Decode the four digits, and only then check your answer with `disp(secret)`. We keep the zeroed samples so that every $X_k$ is on the same 1 Hz grid as $X$. **Optional extension.** What would the grid spacing be if you kept only the 2000 samples of the $k$-th slot? Would that change the width of the peaks, or only how finely they are sampled?

**(e)** Zoom into one peak of $|X_1|$, for example with `xlim([fr-30, fr+30])` where `fr` is the frequency of that peak. Estimate the peak value and the distance between the first minima on either side of the peak, and compare them with your sketch in Part 2(c). The other tone of the digit and the numerical approximation can keep the minima from being exactly zero, so small differences are expected even without added noise. Which sets the width: the 0.25 s window or the 0.20 s tone?

# Part 4: Echo Cancellation

A sound recorded in a hallway between two parallel walls is followed by weaker copies of itself, one after each round trip of the sound between the walls. We model the recording as

$$y(t) = x(t) + \sum_{i=1}^{M} A_i \, x(t - t_i), \tag{4}$$

where $x(t)$ is the original sound, and $A_i > 0$ and $t_i > 0$ are the amplitude and the delay of the $i$-th echo. Our goal is to recover $x(t)$ from $y(t)$. In this lab $M = 3$, $A_i = 2^{-i}$ (each round trip halves the amplitude) and $t_i = i\tau$. The round-trip time depends on your student ID: $\tau = 0.5 + d/16$ s, where $d$ is the last digit of your ID (for example, $d = 8$ gives $\tau = 1$ s).

First answer the following on paper.

**(a)** Find $h(t)$ such that $y(t) = x(t) * h(t)$, i.e., the impulse response of the LTI system in Eq. (4). Find its frequency response $H(\omega)$. What is the relation between $X(\omega)$, $Y(\omega)$ and $H(\omega)$?

**(b)** Given $Y(\omega)$ and $H(\omega)$, how can you find $X(\omega)$? At which frequencies would this fail? Show that $|H(\omega)| \geq 1 - \sum_i |A_i|$ for every $\omega$, and conclude that $\sum_i |A_i| < 1$ is a sufficient condition for recovery by division at every frequency. This condition is not necessary in general. Do our $A_i$ satisfy this sufficient condition?

**(c)** For our parameters, show that $H(\omega)$ is periodic in $\omega$ and find its period. Compute $|H(0)|$ and $|H(\pi/\tau)|$, and use the period to find $|H(2\pi/\tau)|$. Now explain the values at $2\pi/\tau$ and $\pi/\tau$ without any formula: feed $\cos(\omega t)$ into the system and think about how the original and its three delayed copies line up (in phase or out of phase) when $\omega = 2\pi/\tau$ and when $\omega = \pi/\tau$.

Now load a sound and set up the echo parameters. The built-in recording of a gong is sampled at $f_s = 8192$ Hz. With $\tau = 0.5 + d/16$ s, the delay of the $i$-th echo is exactly $D_i = i\,(4096 + 512d)$ samples. The 5.1 s gong is padded with zeros to 10 s so that even the last echo, which starts at $3\tau \leq 3.2$ s, ends inside the record. This matters: `addEcho` below keeps the length of `x`, so an echo running past the end would be cut off, and $y$ would no longer be exactly $x * h$.

```
S = load('gong.mat');          % built-in sound: S.y (samples) and S.Fs (8192 Hz)
fs = S.Fs;
x = S.y.';                     % row vector
N = 10*fs;                     % work with exactly 10 s
x = [x, zeros(1, N - length(x))];   % zero-pad the 5.1 s gong to 10 s
t = (0:N-1)/fs;
[X, omega] = FT(x, fs);        % X(omega) on the frequency grid omega (rad/s)
d = 8;                         % <-- last digit of your student ID
D = (4096 + 512*d)*(1:3);      % echo delays in samples
td = D/fs;                     % echo delays in seconds
tau = td(1);
A = [1/2 1/4 1/8];             % echo amplitudes
```

**(d)** Write a function `y = addEcho(x, A, D)` that returns $y[n] = x[n] + \sum_i A_i\, x[n - D_i]$, taking $x[n] = 0$ for $n < 0$ and keeping the length of `x`. Generate `y = addEcho(x, A, D)`. Listen to $x$ and $y$ and describe the difference. Plot $x$ and $y$ against $t$ in one figure (`subplot(2,1,1)` and `subplot(2,1,2)`). Then measure the impulse response the way its name suggests: feed a unit impulse, `delta = [1, zeros(1, N-1)]`, into `addEcho` and plot the output with `stem` for $0 \le t \le 4$ s. Add `'Marker', 'none'` to the `stem` call so that it does not draw a circle at every zero sample, and use `xlim([-0.1 4])` so that the axis does not hide the impulse at $t = 0$. Compare its values with the weights (areas) of the impulses in your $h(t)$ from (a).

**(e)** Compute $H(\omega)$ on the grid `omega` (one line of code; `A*exp(-1j*td(:)*omega)` is a $1 \times N$ vector). Plot $|H|$ against $\omega$ for $0 \le \omega \le 6\pi/\tau$. The grid `omega` is coarse (spacing $0.2\pi$ rad/s), so for a smooth plot also evaluate your formula on `w = linspace(0, 6*pi/tau, 2000)`. Read the maximum, the minimum and the period from your plot and compare them with (c). **Optional extension.** Go back to the time domain: plot `h2 = IFT(H, fs)` against $t$ for $0 \le t \le 4$ s and compare it with the impulse response you measured in (d). Where are the spikes, and why are their values $f_s = 8192$ times larger than the values you measured in (d)? (Hint: $\delta(t)$ has area 1. How tall must a pulse of width $1/f_s$ be to have area 1?)

**(f)** Recover the sound: compute `Y = FT(y, fs)`, `Xe = Y./H` and `xe = IFT(Xe, fs)`. Listen to $x_e$ and plot it against $t$. Compute the relative error `norm(xe - x)/norm(x)`. Is the echo gone? Explain your result using (a) and (b).

# Part 5: Explore (Optional)

**(a) Windows that do not match the digits.** Part 4 replaced `x`, `fs`, `N` and `t`, so first run the Part 3 code again. Using `x`, you will plot the magnitude spectrum of $x(t)\,w(t)$ for (i) $w(t) = \mathrm{rect}\left(\frac{t-0.05}{0.1}\right)$ and (ii) $w(t) = \mathrm{rect}\left(\frac{t-0.25}{0.25}\right)$. (If the first two digits of your number are the same, use `DTMFTRA([2 5 3 3])` in place of `x` for (ii).) Before plotting (i), use Part 2(c) to predict the height and the null-to-null width of its peaks; then plot it and compare. What do you see in (ii), and why? A real receiver does not know when a key is pressed. Using (i) and (ii), what limits how long and how short its windows can be?

**(b) Echo cancellation without Fourier transforms.** Run the Part 4 code again and regenerate `y = addEcho(x, A, D)`. Then compute `z = addEcho(y, -1/2, D(1))`, i.e., $z(t) = y(t) - \frac{1}{2}\,y(t-\tau)$. Listen to $z$, and plot $z - x$ against $t$. Is any echo left, and where? Show that $z(t) = x(t) - \frac{1}{16}\,x(t-4\tau)$. What would $z(t)$ be if $y(t)$ contained infinitely many echoes with $A_i = 2^{-i}$ and $t_i = i\tau$?

**(c) When is an echo not an echo?** With the gong `x` from Part 4, compute `v = addEcho(x, 0.9, 41)`: a single strong echo that arrives 41 samples (about 5 ms) after the sound. Listen to $x$ and $v$. Do you hear an echo? Plot $|H|$ of this system against $f$ for $0 \le f \le 2000$ Hz. On the same axes, plot the spectrum of the gong scaled to a peak of 1, `abs(X)/max(abs(X))` against `omega/(2*pi)`. How far apart are the dips of $|H|$? Which of the gong's strong frequencies does the echo boost, and which does it cut? Use this to describe what the short echo does to the sound.

# Appendix A: `FT.m` and `IFT.m`

Save the two functions below as `FT.m` and `IFT.m` in your working folder. `[X, omega] = FT(x, fs)` treats `x` as the samples $x(n/f_s)$, $n = 0, 1, \ldots, N-1$, of a signal that is zero outside the record, and replaces the integral in Eq. (3) by a sum:

$$X(\omega) \approx \frac{1}{f_s} \sum_{n=0}^{N-1} x(n/f_s)\, e^{-j\omega n/f_s}.$$

It returns this sum on $N$ frequencies `omega`, spaced $2\pi f_s/N$ rad/s apart from $-\pi f_s$ up to just below $\pi f_s$ (for even $N$, as in all of this lab), including $\omega = 0$. `IFT(X, fs)` undoes `FT`; it warns you if the result is not a real signal, which usually means that `X` was not computed on the grid returned by `FT`. You will learn why the grid stops at $\pm\pi f_s$ when we cover sampling.

A quick check that ties `FT` to Part 2(a): for a pulse of length $T = 0.2$ s, the line below prints 0.2, the value of $X(0) = T$.

```
disp(max(abs(FT([ones(1,1600), zeros(1,6400)], 8000))))
```

### FT.m

```
function [X, omega] = FT(x, fs)
% FT  Approximate continuous-time Fourier transform of a sampled signal.
%   [X, omega] = FT(x, fs) takes the samples x(n/fs), n = 0, 1, ..., N-1,
%   of a signal that is zero outside 0 <= t < N/fs, and returns
%
%       X(k) = (1/fs) * sum over n of x(n) * exp(-1j*omega(k)*n/fs),
%
%   which approximates X(omega) = integral of x(t)*exp(-1j*omega*t) dt.
%   The frequencies (in rad/s) are omega = 2*pi*fs*((0:N-1) - floor(N/2))/N:
%   they are 2*pi*fs/N apart, run from -pi*fs to just below pi*fs (for
%   even N) and include omega = 0. X and omega are row vectors.

if nargin < 2
    error('FT:fs', 'Call FT with the sampling rate: [X, omega] = FT(x, fs).');
end
x = x(:).';
N = length(x);
X = fftshift(fft(x))/fs;
omega = 2*pi*fs*((0:N-1) - floor(N/2))/N;
end
```

### IFT.m

```
function x = IFT(X, fs)
% IFT  Inverse of FT.
%   x = IFT(X, fs) returns the samples x(n/fs), n = 0, 1, ..., N-1, of the
%   signal whose FT is X. X must be given on the frequency grid returned by
%   FT. If X is the transform of a real signal, x is real; otherwise IFT
%   warns and returns a complex x.

if nargin < 2
    error('IFT:fs', 'Call IFT with the sampling rate: x = IFT(X, fs).');
end
X = X(:).';
x = ifft(ifftshift(X))*fs;
if max(abs(imag(x))) <= 1e-9*max(abs(x))
    x = real(x);
else
    warning('IFT:complex', ['The result is not real. Is X given on the ' ...
        'frequency grid returned by FT?']);
end
end
```