Julia Frothingham

**Lab 2 - Touch Tone Phone and Fourier Analysis**

*a) Create row vectors d0 through d9 to represent all 10 digits for the interval 0<=n<=999. Listen to each signal using sound. For example, sound(d2, 8192) should sound like the tone you hear when you push a '2' on the telephone.*

The signal associated with each keypad digit is a sinusoid composed of two sine waves of different frequencies depending on the digit's location on the keypad. In Matlab, I stored the frequencies associated with each row and column of the keypad in arrays for reference and created variables to store the signal associated with each digit. I also used the Matlab function `sound` to play each signal, sampled at 8192 Hz, to check that each signal sounded qualitatively like a dial tone.

*Figure 1. Matlab code to represent touch tone signals for all ten possible digits.*

```
% list row and column frequencies
omega_r = [0.5346 0.5906 0.6535 0.7217];
omega_c = [0.9273 1.0247 1.1328];

% create signals for each digit
n = 0:999;
d0 = sin(omega_r(4)*n) + sin(omega_c(2)*n);
d1 = sin(omega_r(1)*n) + sin(omega_c(1)*n);
d2 = sin(omega_r(1)*n) + sin(omega_c(2)*n);
d3 = sin(omega_r(1)*n) + sin(omega_c(3)*n);
d4 = sin(omega_r(2)*n) + sin(omega_c(1)*n);
d5 = sin(omega_r(2)*n) + sin(omega_c(2)*n);
d6 = sin(omega_r(2)*n) + sin(omega_c(3)*n);
d7 = sin(omega_r(3)*n) + sin(omega_c(1)*n);
d8 = sin(omega_r(3)*n) + sin(omega_c(2)*n);
d9 = sin(omega_r(3)*n) + sin(omega_c(3)*n);

% uncomment to listen to each signal
%sound(d0,8192)
%sound(d1,8192)
%sound(d2,8192)
%sound(d3,8192)
%sound(d4,8192)
%sound(d5,8192)
%sound(d6,8192)
%sound(d7,8192)
%sound(d8,8192)
%sound(d9,8192)
```

In the code above, the number 8192 used with the function `sound` is the sampling rate in Hz. d0, d1, d2, etc. are all continuous time signals, so they have to be sampled and reconstructed in order to be played by the computer. This means that the signal played is sampled 8192 times per second. Plotting the signal over time would have a resolution of 8192 points per second.

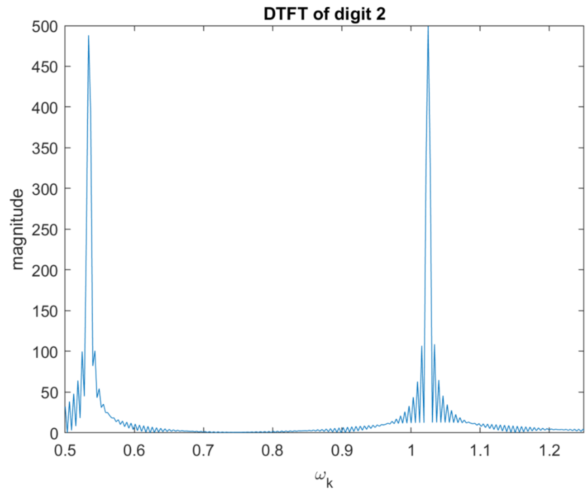
*b) Use the function fft to compute samples of $D_2(e^{j\omega})$ and $D_9(e^{j\omega})$ at $\omega_k$=2πk/2048. Plot the magnitudes of the DTFTs for these signals, and confirm that the peaks fall at the specified frequencies.*

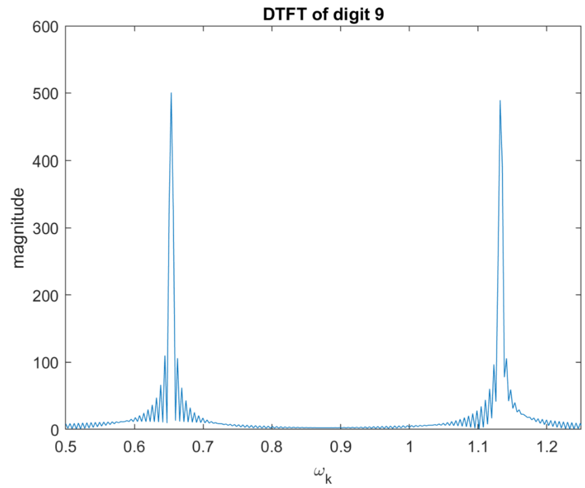
The Matlab function `fft` can be used to calculate the Discrete Fourier Transform (DFT) of a signal. In the code below, the argument `2048` is passed to the function to indicate that 2048 evenly spaced samples should be calculated. The DFT is evaluated at evenly spaced intervals where ω is integer multiples of 2π divided by the number of points in the DFT. In the code below, there are 2048 samples, so there are 2048 integer multiples ranging from 0 to 2047.

*Figure 2. Matlab code to calculate DTFT for each digit and plot for digits 2 and 9.*

```
% make sure omega is same length as fft
k = 0:2047;
omega = k*((2*pi)/2048);

% find the fft for each digit
X1 = fft(d1,2048);
X2 = fft(d2,2048);
X3 = fft(d3, 2048);
X4 = fft(d4, 2048);
X5 = fft(d5, 2048);
X6 = fft(d6, 2048);
X7 = fft(d7, 2048);
X8 = fft(d8, 2048);
X9 = fft(d9, 2048);

% plot the magnitude of the fft for digit 2
figure
plot(omega, abs(X2));
title('DTFT of digit 2');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

% plot the magnitude of the fft for digit 2
figure
plot(omega, abs(X9));
title('DTFT of digit 9');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');
```

*Figure 3. DTFT for phone digits 2 (left) and 9 (right)*




The given table of digits and frequencies states that for a digit of 2 sampled at 8192 Hz, the frequencies should be 0.5346 and 1.0247. It states that for a digit of 9 sampled at 8192 Hz, the frequencies should be 0.6535 and 1.1328. By visual inspection, the peaks in the figure above seem to occur at the correct frequencies.

*c) Define space to be a row vector with 100 samples of silence using zeros. Define phone to be your phone number by appending the appropriate signals and space. Play your phone number using sound and plot the spectra for each digit.*

In order to play my phone number and have each digit's dial tone perceptibly distinct from the others, there needs to be some silence between the signals for each digit. The variable `phone` contains the signal for each digit in my phone number separated by 100 samples of a zero-amplitude signal for silence. The Matlab function `sound` is used to sample the signals at 8192 Hz and play them out loud. The DFTs for each digit were calculated in the Matlab code in Figure 2. They are plotted in the code shown below in Figure 4a.

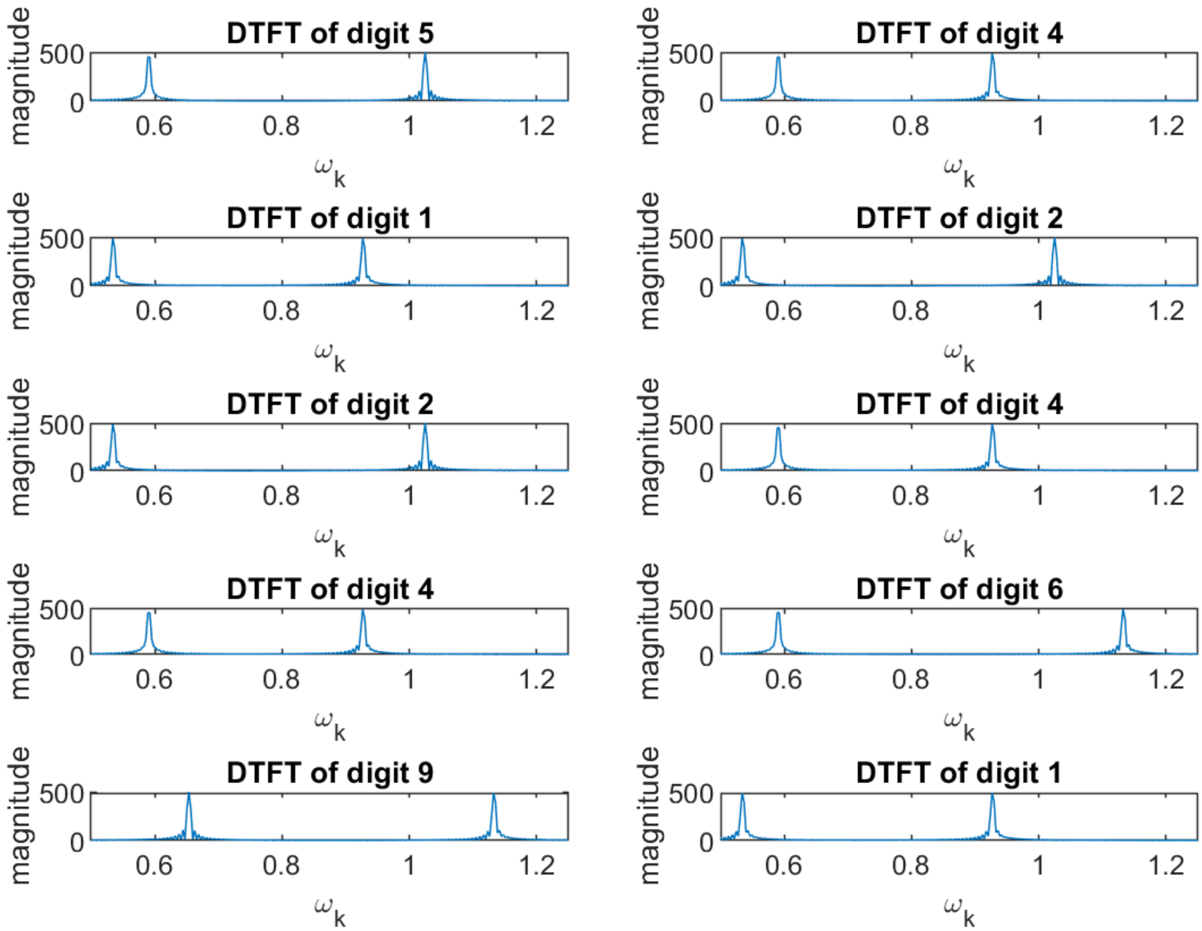
*Figure 4a. Matlab code to play the digits of my phone number and plot each digit's DFT.*

```
% define a variable to hold signals for each digit of my phone number,
% separated by spaces for ease of listening
empty = zeros(100);
space = empty(1,:);
phone = [d5 space d4 space d1 space d2 space d2 space d4 space d4 space d6 space d9 space d1];

% play the dialtones for my phone number
sound(phone, 8192);

figure  % plot the magnitudes of each digit's signal's DTFT

subplot(5,2,1);
plot(omega, abs(X5));
title('DTFT of digit 5');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,2);
plot(omega, abs(X4));
title('DTFT of digit 4');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,3);
plot(omega, abs(X1));
title('DTFT of digit 1');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,4);
plot(omega, abs(X2));
title('DTFT of digit 2');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');
```

*Figure 4b. (Matlab code from Figure 4a continued)*

```
subplot(5,2,5);
plot(omega, abs(X2));
title('DTFT of digit 2');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,6);
plot(omega, abs(X4));
title('DTFT of digit 4');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,7);
plot(omega, abs(X4));
title('DTFT of digit 4');
xlim([0.5 1.25])
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,8);
plot(omega, abs(X6));
title('DTFT of digit 6');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,9);
plot(omega, abs(X9));
title('DTFT of digit 9');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(5,2,10);
plot(omega, abs(X1));
title('DTFT of digit 1');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');
```

*Figure 5. The DTFT of each digit of my phone number.*




By visual inspection, the peaks of each digit seem to match up with the expected frequencies based on keypad location. When played using `sound`, the phone number sounded like a typical set of dialtones. Both indicate that the DTFT was calculated and plotted correctly.

*d) Using fft, compute 2048 evenly spaced samples of the DTFT for each digit of x1.*

It is known that the signal x1 contains seven digits of 1000 samples each, with 100 samples of silence between each digit. The first step in determining the DTFT of each digit is to separate the digit. The first digit consists of the first 1000 samples contained in x1. Then, after skipping 100 samples, the second digit consists of the next 1000 samples, and so on. The Matlab function `fft` was used to calculate the DTFT of each digit. To check the results and determine frequencies, these DTFTs were plotted.

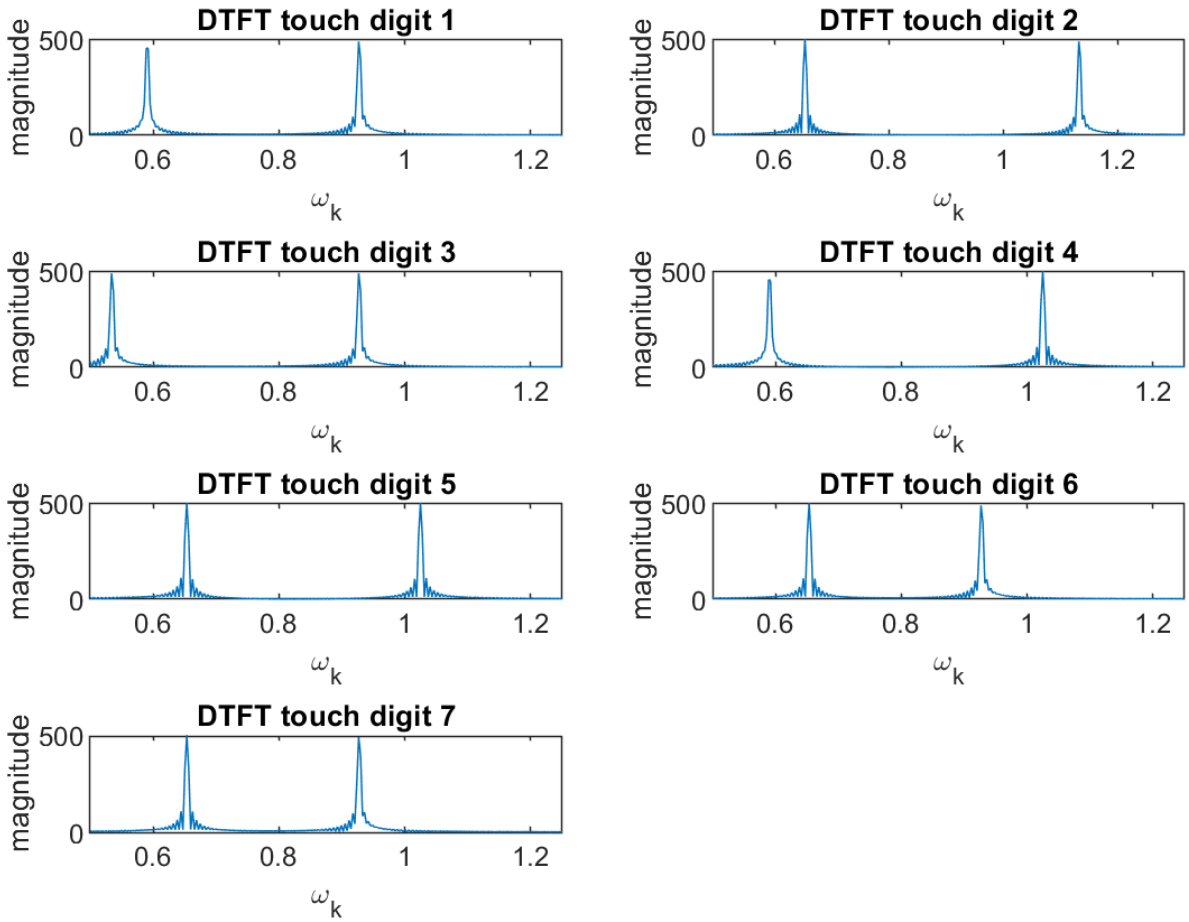
*Figure 6a. Matlab code to calculate and plot the DTFT of each digit in the given file 'touch.mat.'*

```
% load file containing variable x1
load touch.mat;

% parse out each digit of x1
d1_t = x1(1:1000);
d2_t = x1(1101:2100);
d3_t = x1(2201:3200);
d4_t = x1(3301:4300);
d5_t = x1(4401:5400);
d6_t = x1(5501:6500);
d7_t = x1(6601:7600);

% find the fft of each digit of x1
X1_t = fft(d1_t,2048);
X2_t = fft(d2_t,2048);
X3_t = fft(d3_t, 2048);
X4_t = fft(d4_t, 2048);
X5_t = fft(d5_t, 2048);
X6_t = fft(d6_t, 2048);
X7_t = fft(d7_t, 2048);

% plot the magnitude of the DTFT of each digit of x1
figure
subplot(4,2,1);
plot(omega, abs(X1_t));
title('DTFT touch digit 1');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(4,2,2);
plot(omega, abs(X2_t));
title('DTFT touch digit 2');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');
```

*Figure 6b. (Matlab code continued from Figure 6a)*

```
subplot(4,2,3);
plot(omega, abs(X3_t));
title('DTFT touch digit 3');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(4,2,4);
plot(omega, abs(X4_t));
title('DTFT touch digit 4');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(4,2,5);
plot(omega, abs(X5_t));
title('DTFT touch digit 5');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(4,2,6);
plot(omega, abs(X6_t));
title('DTFT touch digit 6');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');

subplot(4,2,7);
plot(omega, abs(X7_t));
title('DTFT touch digit 7');
xlim([0.5 1.25]);
xlabel('\omega_k');
ylabel('magnitude');
```

*Figure 7. The DTFT of each digit of the phone number dialled in the given file.*




By inspection, each digit only has two frequency spikes, and they seem to be frequencies that match the expected frequencies available. The value of each digit (its position on the keypad) can be determined by manually matching up the frequency spikes in the plotted DTFT to the corresponding pair of frequencies associated with the correct numerical digit.

By visual inspection:

Digit 1 → spikes ~0.59 and ~0.92 → **4**
Digit 2 → spikes ~0.65 and ~1.13 → **9**
Digit 3 → spikes ~0.53 and ~0.92 → **1**
Digit 4 → spikes ~0.59 and ~1.02 → **5**
Digit 5 → spikes ~0.65 and ~1.02 → **8**
Digit 6 → spikes ~0.65 and ~0.92 → **7**
Digit 7 → spikes ~0.65 and ~0.92 → **7**

So the number dialled in the given file is 491-5877. The sum of these digits is 41, which satisfies the condition given as a hint to check the solution.

*e) Use your knowledge of the sampling rate to determine the actual continuous time frequencies for each digit on the phone keypad. List all frequencies in a matrix corresponding to the keypad, and describe how you calculated them.*

The given frequencies $\omega_{row}$ and $\omega_{column}$ are sampled at 8192 Hz. Normally, to convert between an angular frequency and a frequency in Hz, the frequency in Hz is the angular frequency divided by 2π. However, in this case $\omega$ is sampled. So to convert to a continuous time frequency in Hz, the sampled angular frequency needs to be multiplied by the sampling rate as well as divided by 2π.

*Figure 8. Matlab code to calculate the actual CT frequencies for each keypad digit.*

```
% define conversion factor from sampled frequency to frequency in Hz
conversion = 8192 / (2*pi);

% calculate and display keypad frequencies in Hz
freq_r = conversion * omega_r;
freq_c = conversion * omega_c;
freq_r
freq_c
```

```
freq_r = 1×4
    697.0100   770.0227   852.0315   940.9505

freq_c = 1×3
10^3 ×
      1.2090     1.3360     1.4769
```

*Figure 9. Comparison of sampled DTFT frequencies and frequencies in Hz.*

| **Digit** | $\omega_{row}$ | $\omega_{column}$ | $f_{row}$ **(Hz)** | $f_{column}$ **(Hz)** |
|---|---|---|---|---|
| 0 | 0.7217 | 1.0247 | 941 | 1336 |
| 1 | 0.5346 | 0.9273 | 697 | 1209 |
| 2 | 0.5346 | 1.0247 | 697 | 1336 |
| 3 | 0.5346 | 1.1328 | 697 | 1477 |
| 4 | 0.5906 | 0.9273 | 770 | 1209 |
| 5 | 0.5906 | 1.0247 | 770 | 1336 |
| 6 | 0.5906 | 1.1328 | 770 | 1477 |
| 7 | 0.6536 | 0.9273 | 852 | 1209 |
| 8 | 0.6535 | 1.0247 | 852 | 1336 |
| 9 | 0.6535 | 1.1328 | 852 | 1477 |