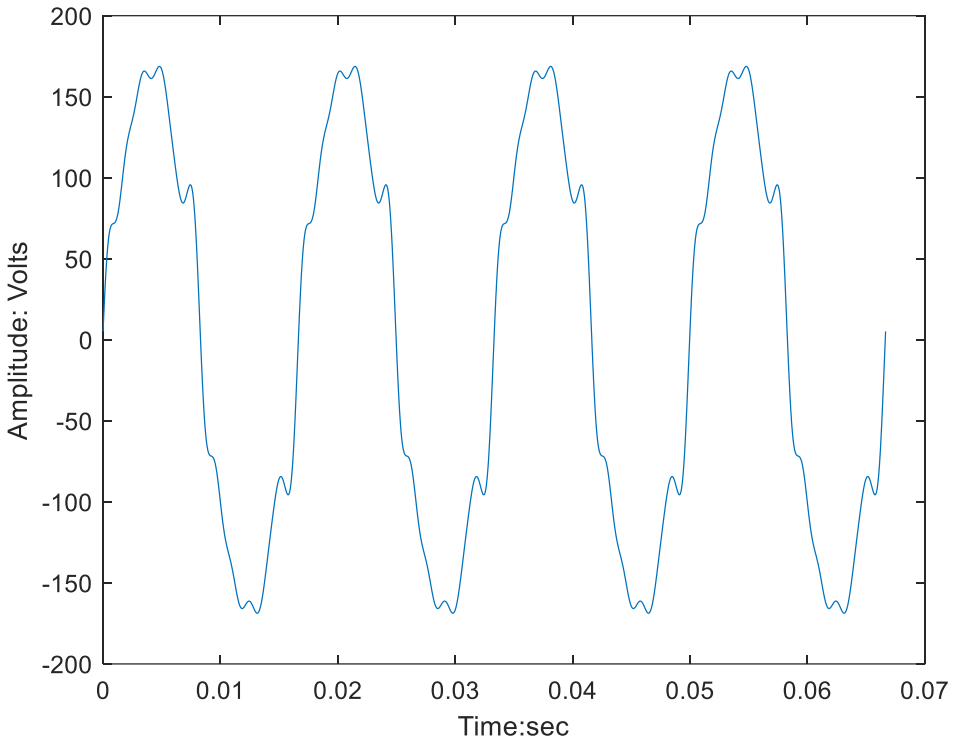
ECGR 2254

Project 2

Due: 11/22 at class time

Problem 1 – A simple power quality analyzer

Download the file test1.mat from the Canvas page. Load this file into MATLAB using the command `load('problem1.mat)`. This should load a variable `x` into your workspace. x contains 4 periods of the grid voltage in a local building. This voltage appears as shown below:



The measured voltage contains several harmonics of the fundamental 60Hz voltage. Please note that the voltage was sampled at a frequency $f_s = 60{,}000$.

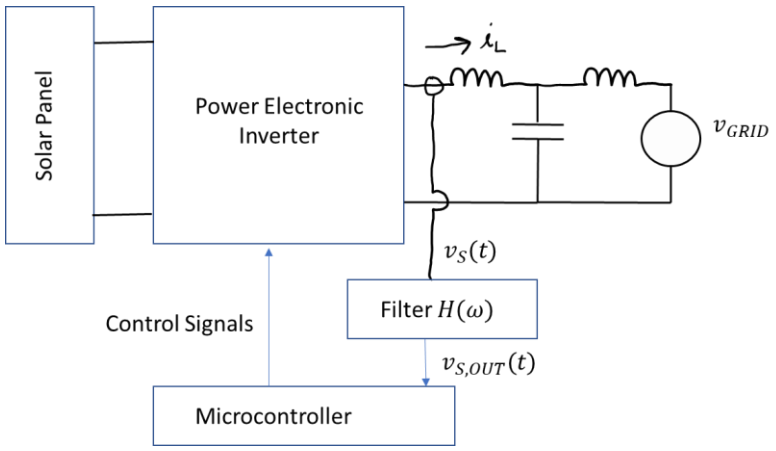
In this problem, you are asked to do the following:

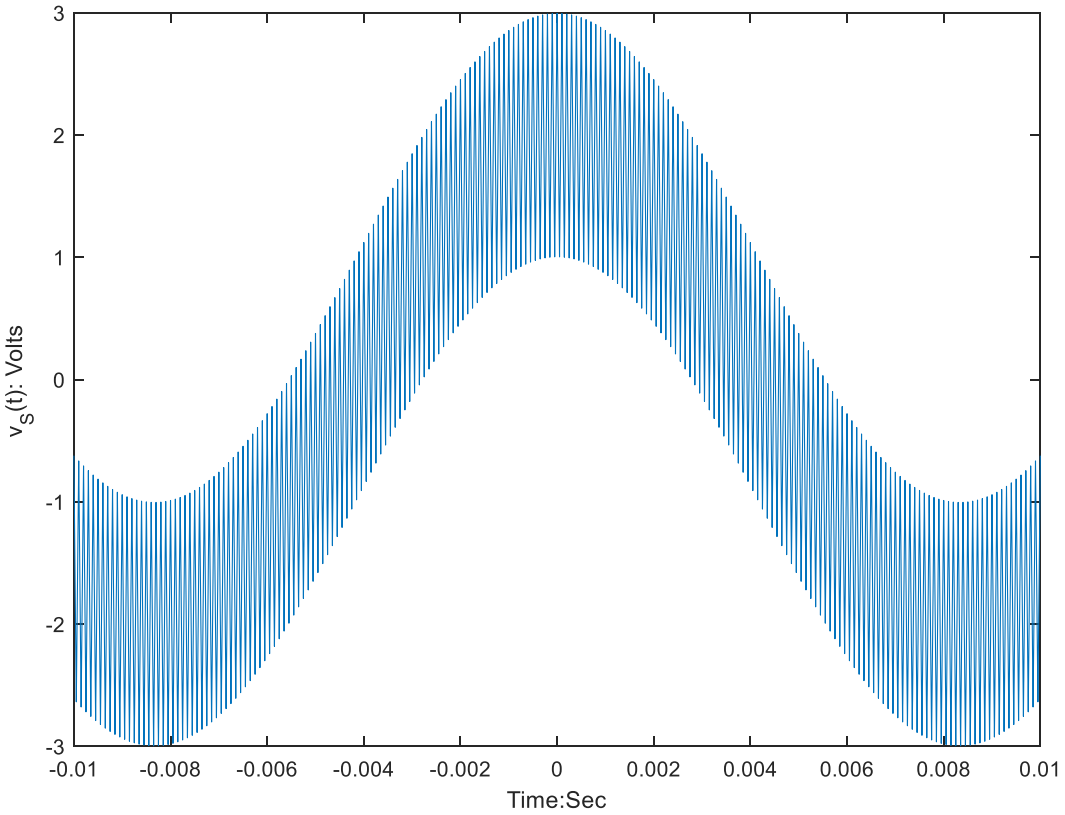
a) Determine the amplitude of the sine and cosine components at each harmonic of the fundamental from $n = 1$ to $n = 13$. To solve this problem, you will need to integrate numerically.
b) Create a graph that shows the Fourier transform of x from $n = -13$ to $n = 13$.

In your answers, carefully explain your work.

Problem 2 – Filtering a Signal for Control

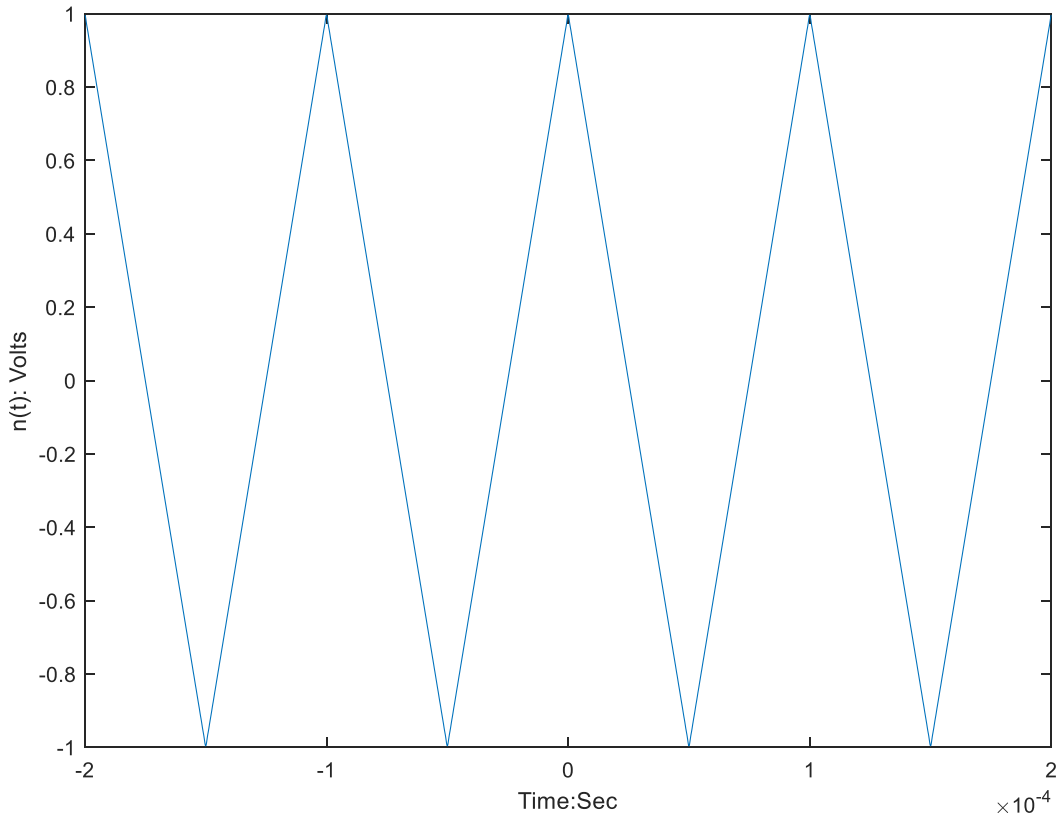
Solar panels are connected to the grid through a power-electronic inverter circuit that creates a sinusoidal current. An example system is shown in the figure below. Note that a current sensor measures the current $i_L(t)$ and converts it to an output voltage $v_S(t)$. An example version of this voltage $v_S(t)$ is also shown. Note that there is a 60Hz term added to a triangular signal. This triangular signal results from the action of the switches inside the inverter, and it is largely removed from the current as the signal passes through the LCL filter into the grid.





In a real control system, the signal $v_S(t)$ coming from the current sensor might be needed inside the microcontroller to be able to decide how to drive the switches. Typically, however, we only want the microcontroller to see the 60Hz signal and not the high-frequency triangle wave. To make sure this happens, we have added the filter $H(\omega)$, which would need to be constructed using op-amps, resistors, and capacitors. This is a common problem in embedded controls. Our goal here is to design the filter $H(\omega)$ using typical design specifications.

We are told that $v_S(t) = 2\cos(2\pi 60t) + n(t)$. The noise $n(t)$ is shown below.



a) Determine the Fourier Transform of $n(t)$ for $n = \pm 1, \pm 2, \pm 3$ by integrating numerically. To generate the signal $n(t)$:
   a. Determine the period from the figure shown above
   b. Use the sawtooth command in MATLAB. Check the Mathworks page on the sawtooth command to see an example on how to use the sawtooth command to create a triangle wave.
   c. Make sure the triangle wave you create has the same phase as the signal shown in the figure above.

   When integrating, I would recommend dividing the period into 1000 evenly space samples. You will need to determine the time step Δt.

   Determine the corresponding signals $C_n \cos(n\omega_o t + \phi_n)$ for $n = 1,2,3$.

You can verify your answers by integrating by hand, but it will be messy.

b) We now must design the filter. Let's assume we have the following specifications:
   - The fundamental component of $n(t)$ must be reduced to about 1% of its input value. Use this spec to determine an approximate desired filter magnitude at the fundamental frequency of $n(t)$.
   - The 60Hz component of $v_S(t)$ must not be significantly affected. This means that the amplitude of the output 60Hz component must be almost the same as the amplitude of the input component. The resulting phase shift should be less than 4 degrees.

You are given the following choices for the filters:

$$H_1(\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

$$H_2(\omega) = \frac{\omega_c^2}{(j\omega)^2 + j\frac{2}{\sqrt{2}}\omega_c\omega + \omega_c^2}$$

$$H_4(\omega) = \frac{\omega_c^4}{((j\omega)^2 + j0.7654\omega_c\omega + \omega_c^2)((j\omega)^2 + j1.8478\omega_c\omega + \omega_c^2)}$$

In each case, the frequency ω_c is a constant known as the corner frequency. This is the frequency at which the magnitude of the transfer function has fallen -3dB below its value at DC.

Select an appropriate filter to meet the requirements. To do so, you must choose ω_c for the various filters. All three of the filters can meet one of the requirements and only one can meet both.

Explain your filter choice in detail using appropriate bode plots.

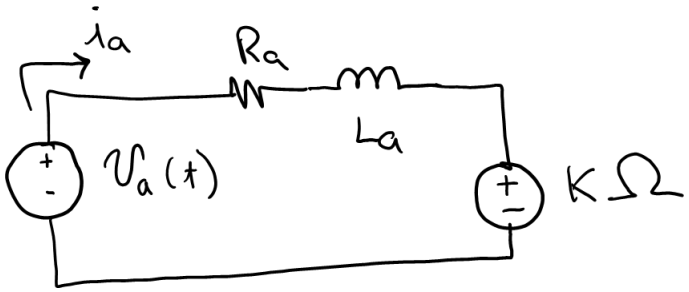
Determine the output signal for $v_{s,OUT}(t)$ for your filter choice. You should include the 60Hz term, and first, second, and third harmonics of $n(t)$.

A few hints:

- You should consider using the bode command in MATLAB. To use bode, you should go back to your learnings from 2112, where you learned that $s = j\omega$ and wrote $H(s)$.
- Watch the video on how to use the bode command in MATLAB.

Problem 3 – A PWM motor drive

In project 1, you were given the model for a DC motor. The circuit model for the DC motor is shown below:



The variable Ω is the speed of the motor in rad/sec. Note that the model includes a speed-dependent voltage source.

The mechanical component of the motor is represented by Newton's Second Law for rotation, which states:

$$(Moment\ of\ Inertia) \times (Angular\ Acceleration) = Net\ Torque$$

Mathematically, this is:

$$J\frac{d\Omega}{dt} = Ki_a - \beta\Omega$$

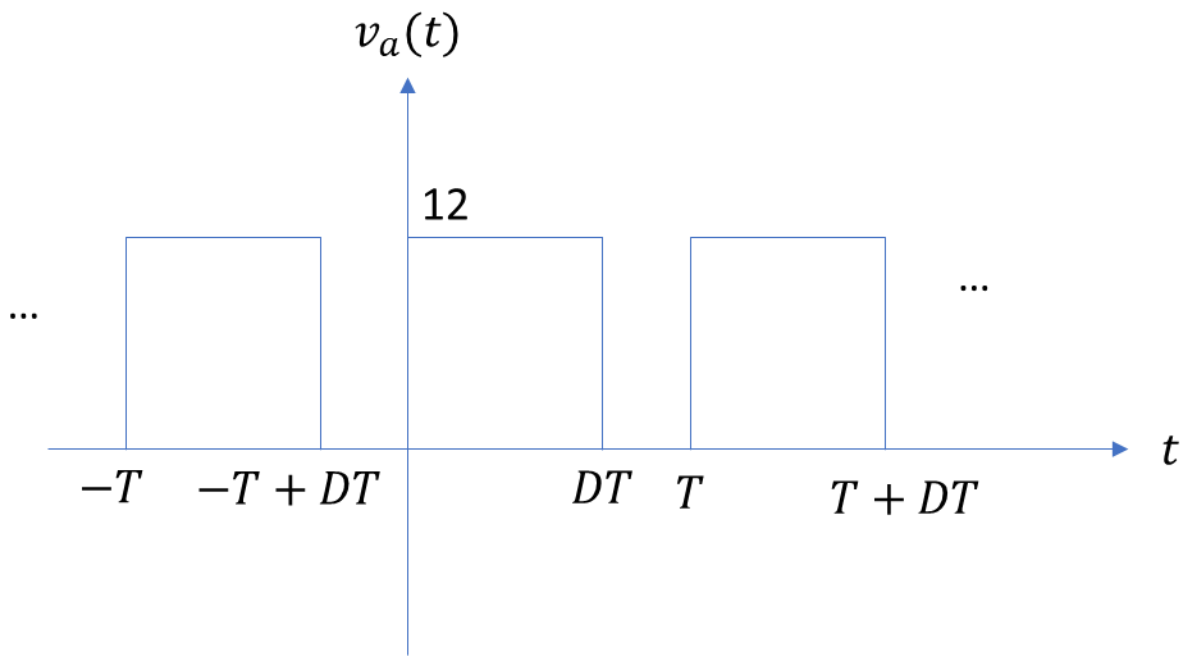
Where:

- β: Damping ratio
- J: Moment of inertia
- K: Motor constant

The motor you are controlling is has the following parameter values:

- $L = 0.01\ H$
- $R = 3.38\ Ohms$
- $K = 0.029\ Vs/rad$
- $J = 2 \times 10^{-4}\ kg\ m^2$
- $beta = 0.5 \times 10^{-5}\ Nms/rad$

In this case, the motor will be driven by a pulse width modulated (PWM) signal such as the one shown below:



This waveform is periodic with a period T , a peak value of 12, a minimum value of 0, and a duty ratio D. Please note that the duty ratio can vary but is a fraction between 0 and 1.

a) Determine a differential equation for $i_a(t)$ from analyzing the electrical circuit. You should be able to find this from Project 1.
b) Using the two differential equations, create a transfer function that has $V_a(\omega)$ as its input and $\Omega(\omega)$ as its output.
c) Plot the magnitude of the transfer function you created in part b. In this particular case, generate the plot without using the Bode command. To do so:
    a. Use the logspace command to create 10,000 frequencies between 0.01 rad/sec and 10,000 rad/sec.
    b. Generate a vector containing the elements of the complex function $H(\omega) = \Omega(\omega)/V_a(\omega)$.
    c. Plot the magnitude of the transfer function versus the log of frequency using the semilogx command. The vector that you plot should be $20\log_{10}|H(\omega)|$. Use the abs command in MATLAB to create the magnitude and use the log10 command.
d) Use your plot from part c to do the following:
    - Determine a switching frequency $f = \frac{1}{T}$. Select a switching frequency so that the amplitude of the signal at the switching frequency will be about 1% of the DC value. Explain your choice for the switching frequency.
    - Determine the DC voltage that will be required to turn the motor at a speed of 324 rad/sec.
e) Compute the DC term of the Fourier series for the PWM waveform as a function of D. Determine the value of D you need to apply the appropriate DC voltage to the motor that you determined in part d.
f) To keep things simple, determine the complex Fourier series coefficients for a PWM waveform with $D = 0.5$. What is the amplitude C_1 of the cosine $C_1\cos(\omega_0 t + \phi)$ at the

fundamental frequency $\omega_0 = 2\pi/T$? Use this value to predict the amount of speed variation that will occur at switching frequency. Note that the output of the filter is a speed variation of the form $\Omega(t) = C_1 \cos(\omega t + \phi)$. This means the speed is varying a little bit around a constant value.

g) Using your work from Project 1, simulate the response of the motor when the voltage is a PWM waveform with the duty ratio D and the switching frequency you determined in part e. In this case, you should do the following:
   - Use the square command to create $v_a(t)$
   - Assume a time step small enough that you can see 10,000 points in one switching period T.
   - Run your simulation for 10 seconds

   What is the average value of the speed in steady-state? Does it match your approximation well? It should match very closely (way less than 1% error).

   What is the peak-to-peak value of the variation at the switching frequency? This value should be pretty close to your prediction, but the error will be a bit bigger. There are at least two reasons. Name them and explain.

h) It is very easy to increase switching frequency with a microcontroller. Select a switching frequency such that the amplitude of the speed at the switching frequency is attenuated to 0.01% of the DC value. What is this switching frequency? Run a simulation and show that your result is approximately correct.

Problem 4 – Discrete Time Signals

Let's suppose you're given the signal below:

$$x_c(t) = A \sin(\omega t + \phi)$$

a) This signal is sampled using an ideal sampler. Use the subplot command to show the interpolated version of $x_c(t)$, $x_p(t)$, and $x_d(n)$ for
   - $T = 10$ sec, $A = 1, \phi = 0, T_s = 2$ sec
   - $T = 10$ sec, $A = 1, \phi = 0, T_s = 0.2$ sec
   - $T = 10$ sec, $A = 1, \phi = 0, T_s = 0.01$ sec

   Please create three separate figures with three graphs on each. When creating these plots, use plot for $x_c(t)$ and stem for $x_p(t)$ and $x_d(n)$. You only need to plot over one period. In each case, what is the period N of $x_d(n)$ and the corresponding frequency ω in radians?

b) Let $x_c(t)$ be $x_c(t) = 4\sin(2\pi 1000t + 30^\circ) + 2\cos(2\pi 100t + 20^\circ)$. Create the signal $x_d(n)$ with
   - $f_s = 10{,}000$Hz
   - $f_s = 100{,}000Hz$

   In each case, what is the period N and frequency ω (in radians) of the two terms in the signal?

c) Let $x_c(t)$ be $x_c(t) = 4\sin(2\pi 1000t + 30°)$. Over one period, plot $x_d(n)$ using the stem command for $f_s = 10{,}000Hz$. Increase the frequency of $x_d(n)$ by 2π and call this signal $x_{d2}(n)$. Use the "hold on" command in MATLAB and plot this second signal. Why do these signals sit on top of one another? The frequency is different in each case, but somehow the signals sit on top of each other. Explain. (Note: this would not happen in continuous time).

d) We again have $x_c(t) = 4\sin(2\pi 1000t + 30°) + 2\cos(2\pi 100t + 20°)$. Create the signal $x_d(n)$ with $f_s = 10{,}000$Hz over 40,000 points.
Use MATLAB to compute the Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N_p-1} x(n)e^{-j\left(\frac{2\pi}{N_p}\right)kn}$$

In each case, chose a value N_p so that the distance $\Delta\omega = \frac{2\pi}{N_p}$ between frequencies in the DFT is no larger than $0.001\ rad$.

e) Plot the magnitude of the DFT $X[k]$ versus ω (in radians). Create a frequency vector $\omega_k = \frac{2\pi}{N}k$. There should be two "spikes" for each sine wave. Although the peak values are quite large, do the relative values of the "spikes" make sense? Explain.

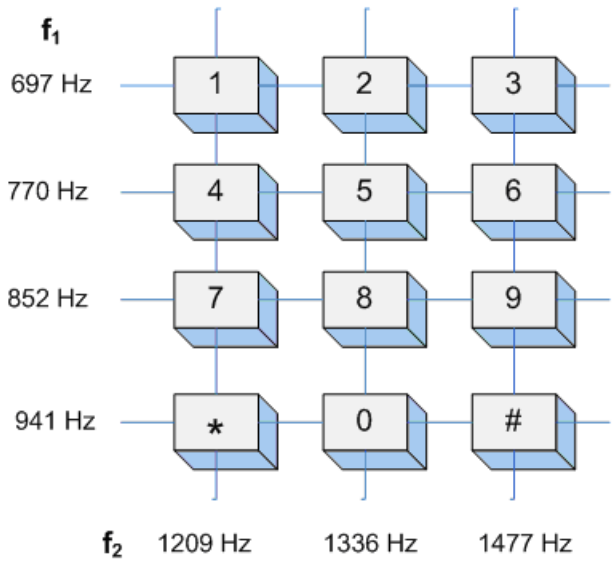
Why are there two sets of spikes for each term? Think about what the Fourier Transform would look like for a continuous time signal consisting of 2 sine waves. How may impulses would it have with 2 sine waves? Knowing this, explain why there is a second set of "spikes" appearing near 2π. You should be able to justify the exact value of the frequencies of the "spikes" that appear near 2π. Your result from part c may help to explain what is happening.

f) Now, plot the magnitude of the DFT $X[k]$ versus the corresponding frequency in Hz. Determine how to appropriately scale the frequency axis. This is a more common plot. Only plot for discrete-time frequencies from 0 to π so that the second set of "spikes" is left out. We commonly only plot the first half of the DFT since the information for frequencies between π and 2π is redundant.

g) Now recalculate the DFT with N_p chosen so that $\Delta\omega = 0.01$ rad. Now, recreate the plot from part e. Why has the lower frequency "spike" disappeared? Explain what is happening.

Problem 5 – A Touch Tone Recognition System

Touch-tone telephones use signals of different frequencies to indicate which key has been pressed. The sound you hear when you press a key is the sum of two sinusoids. The higher frequency sine wave indicates the column of the key, and the lower frequency sine wave indicates the row. The figure below shows how the frequencies are assigned to each row and column.



For example, the CT signal produced when pressing the "5" is thus

$$d_5(t) = \sin(2\pi 770t) + \sin(2\pi 1336t)$$

In a cellular phone, these signals are stored as digitized waveforms, sampled at 8192Hz. In this assignment, you are going to create each digit. Later you will use the DFT to decode several numbers in Matlab.

a) Since we can only save discretized sine waves on the cell phone, compute the discrete-time frequencies that correspond to each of the column and row frequencies.

b) In Matlab, create row vectors d0 through d9 to represent all 10 digits for the interval $0 \leq n \leq 999$. Listen to each signal using the command `sound`. For example, `sound(d2,8192)` should sound like the tone your hear when you push a "2" on the phone.

c) Create a vector called `space` consisting of 100 samples of zeros. Now, create your phone number using a piece of code like the following:

```
>> x = [d7 space d0 space d4 space d6 space d8 space d7 space d8
space d4 space d0 space d2];
```

If you type,

```
>> sound(x,8192)
```

You will hear your phone number.

d) In Matlab, compute the DFT of `d2` and `d5` using $N_p = 2048$. Plot the magnitude versus ω_k (in radians). Since $d2$ and $d5$ only include 1000 samples, you will need to add zeros to the end of $d2$ and $d5$ to be able to allow your code to properly function. You do not need to submit anything for this part. It is intended to prepare you for part e.

e) From the Canvas page, download the file `touch.mat`. Store this file in your current Matlab directory and type `load touch` at the Matlab prompt. This loads four "phone number" vectors into Matlab. One of these is called `x1`. This vector consists of 7 digits of 1000 samples each, separated by 100 samples of silence. Determine the digits of the phone number stored in `x1`. To do so, compute 2048 samples of the DFT of each digit. Again, you will need to add zeros to the end of each digit.

Determine the phone number that was entered. You should include appropriate plots that explain your answers. As a hint, the digits in the phone number sum to 41.