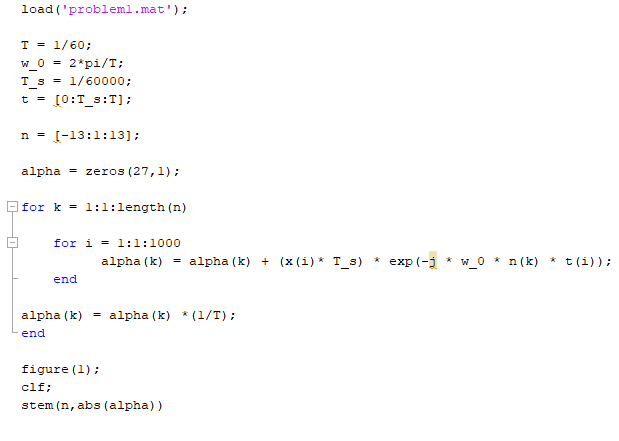
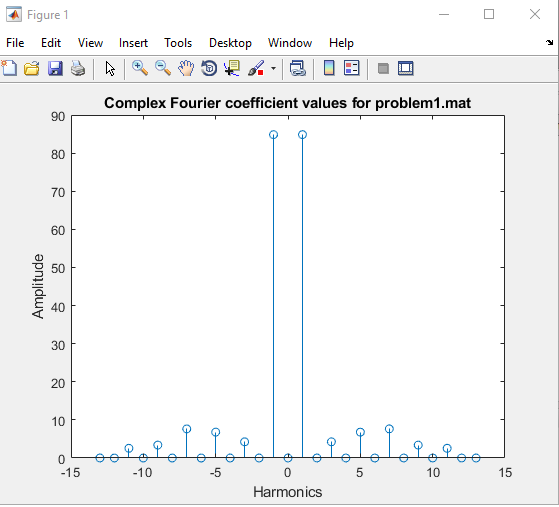
**Problem 1:** A test file was provided with a variable x containing a list of points for a sinusoidal voltage.

1. Given a voltage signal, the complex coefficients of its sine and cosine components were calculated using MATLAB. Two nested For-Loops were needed to do this without any built-in MATLAB functions. The outer loop iterates for the required number of harmonics, 27 in total, while the inner loop iterates based on numeric integration using the Fourier Series complex formula:

The results of a Fourier Series are equivalent to a periodic waveforms’ Fourier Transform.

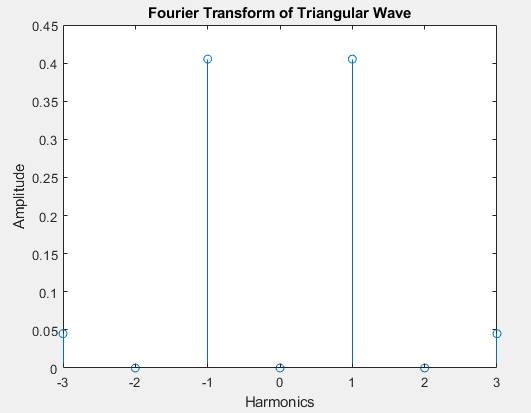
1. The transform coefficients were then plotted in MATLAB using the stem function to provide discrete point results for each harmonic point calculated in part a.

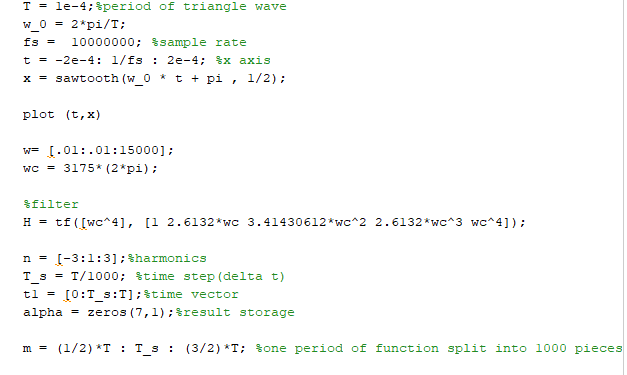


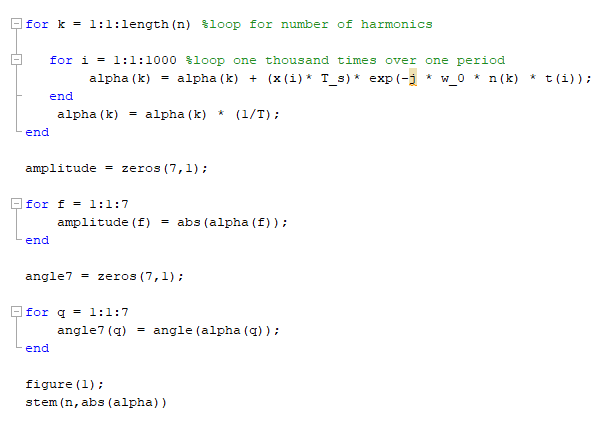


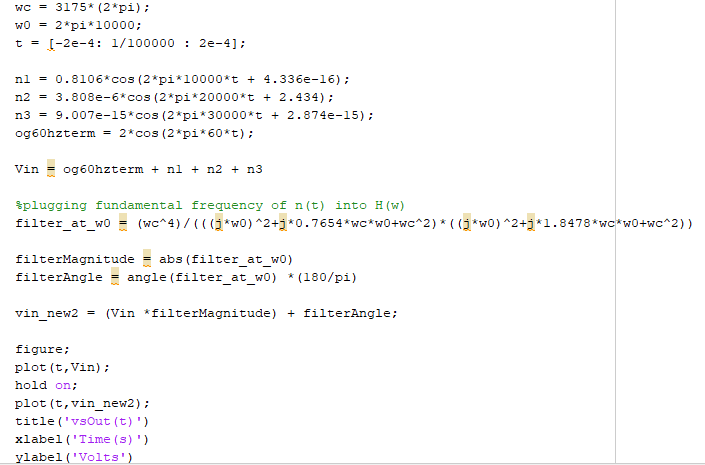
**Problem 2:** A 60Hz signal corrupted by a higher frequency sawtooth wave was provided. An appropriate filter and corner frequency were then chosen to reduce the signal as much as possible without shifting the phase of the original signal.

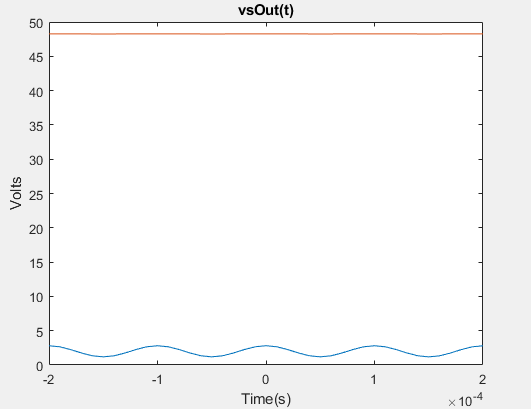
1. The sawtooth wave was reproduced using the sawtooth command in MATLAB. Using numeric integration similar to project 1, the Fourier Coefficients for positive and negative harmonics 1-3 were calculated. Those coefficients were then turned into shifted cosine form by computing their angle and magnitude in MATLAB, then multiplying the magnitude by two.



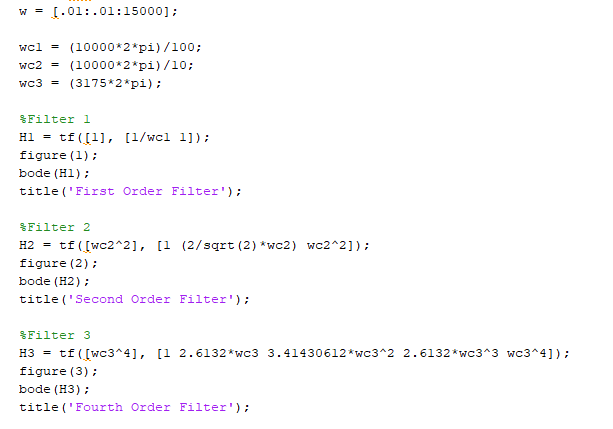


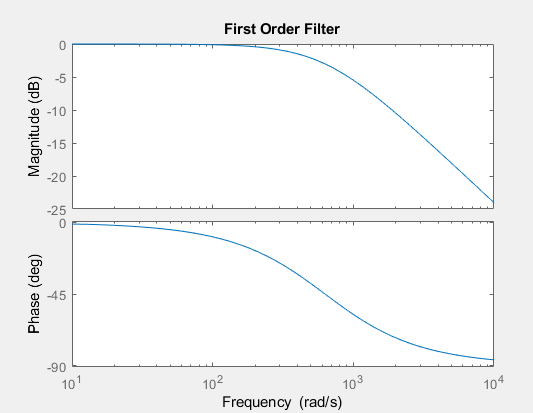


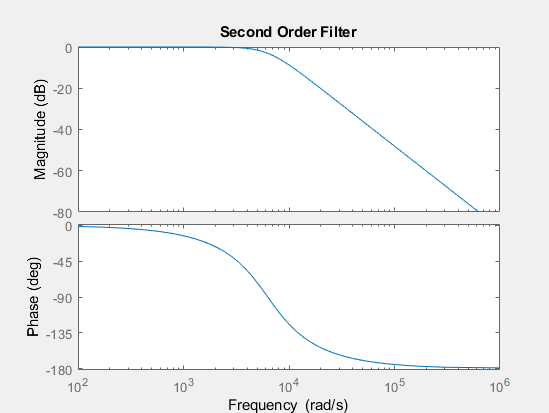


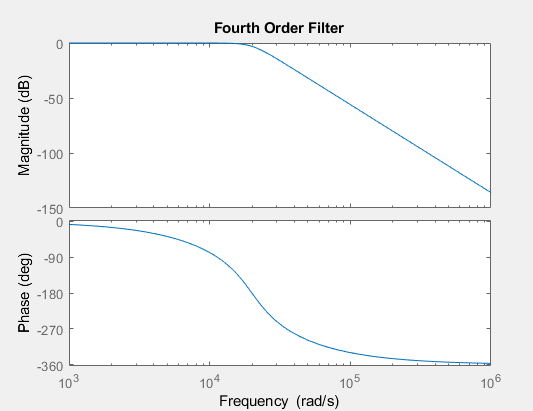


1. Given three options of filters and two constraints, the appropriate filter and corresponding corner frequency were chosen. The filters were tested within MATLAB by plugging in corner frequencies into their given transfer functions. The bode() MATLAB command was then used to analyze the performance of the filter. After plugging in corner frequency values for all filters, it was quite clear that the fourth order filter would be the most helpful in completing the task of the original problem with the smallest phase shift and a corner frequency of 2π3175 rads/sec.



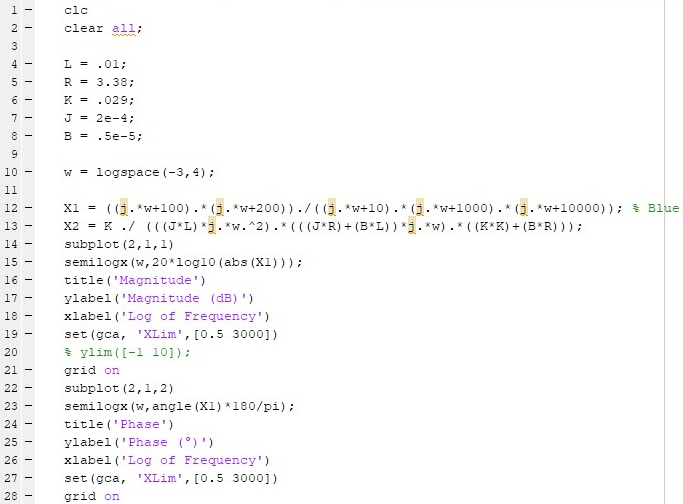


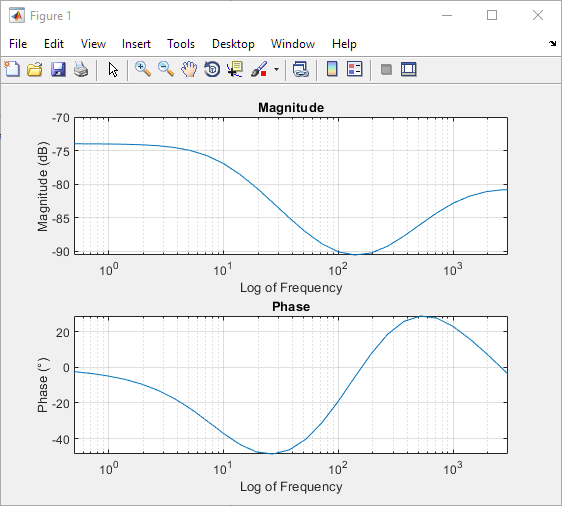




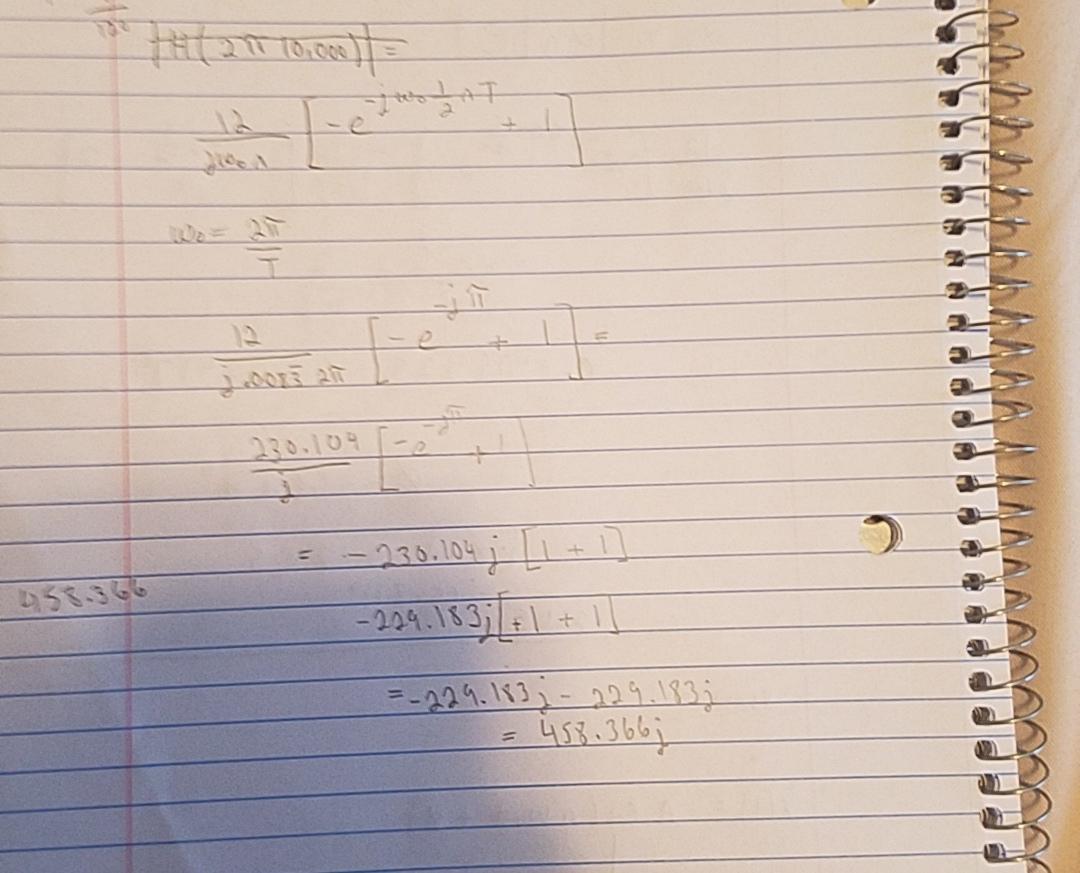
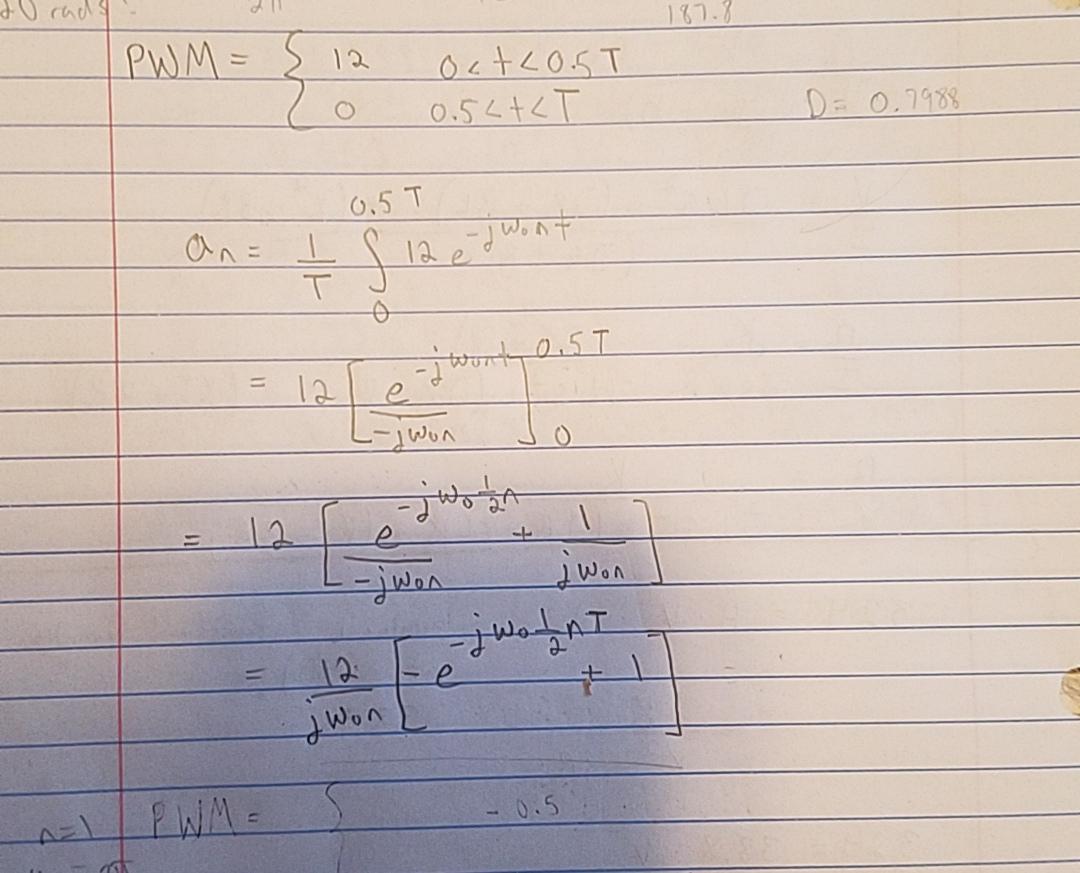
**Problem 3:** A model for a DC motor, a differential equation for its mechanical circuit components and a periodic PWM waveform were provided.

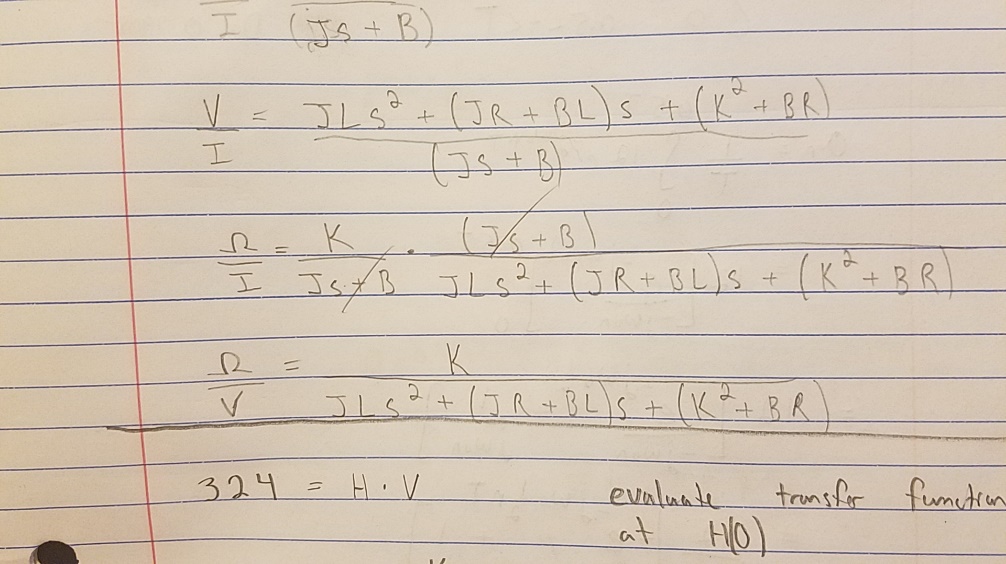
1. Referred back to Project 1, Problem 4 to reuse the differential equation of the identical DC motor’s mechanical circuit.
2. Both differential equations were computed algebraically to eliminate the term of the current in transfer function orientation. The DC motor speed represented output and the voltage represented input.
3. The frequency response diagram was computed using user-defined code in MATLAB rather than the pre-defined bode function as instructed.



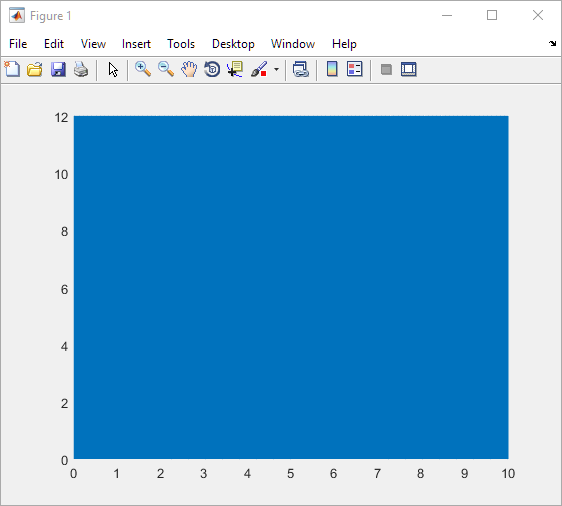


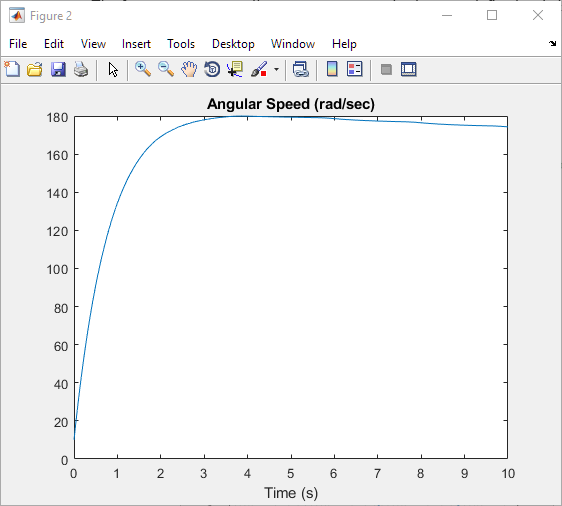
1. The frequency response plot from C was used to determine that the switching frequency should be set to 120 rad/sec, which would allow the amplitude of the signal to be about 1% of the DC value of the PWM. To determine the DC voltage required to turn the motor at 324 rad/sec, the transfer function was set to 0 frequency and solved to be 9.586V.
2. To compute the DC term of the Fourier Series as a function of D, the 9.586V was set equal to 12D, producing D = 0.7988.
3. See Below

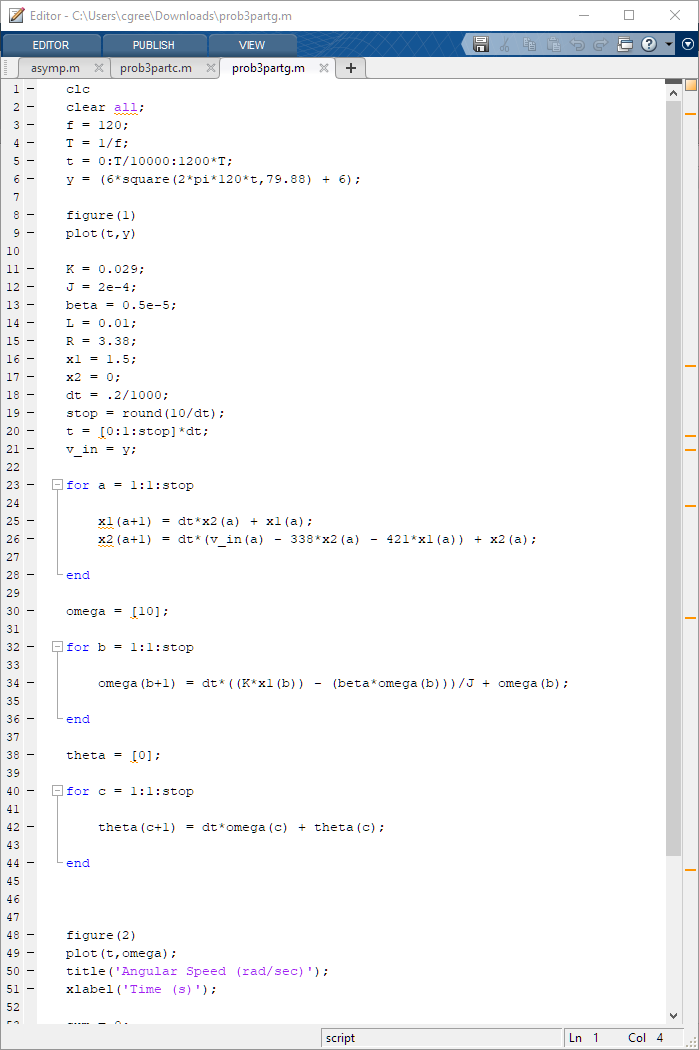




1. The code from Project 1 and the square function (with a duty ratio of 0.7988 and an amplitude of 12) to simulate the PWM was used to simulate the motor’s behavior. The average value of the speed in steady-state was 177.366, which was quite far from the approximations. The peak-to-peak value at the switching frequency was 0.9. One possible reason for the difference in the peak-to-peak variation of the switching frequency is that the prediction used a duty cycle of 0.5 but the actual PWM waveform had a duty cycle of 0.7988. Another possible explanation is a possible programming error for the MATLAB model of the DC motor.





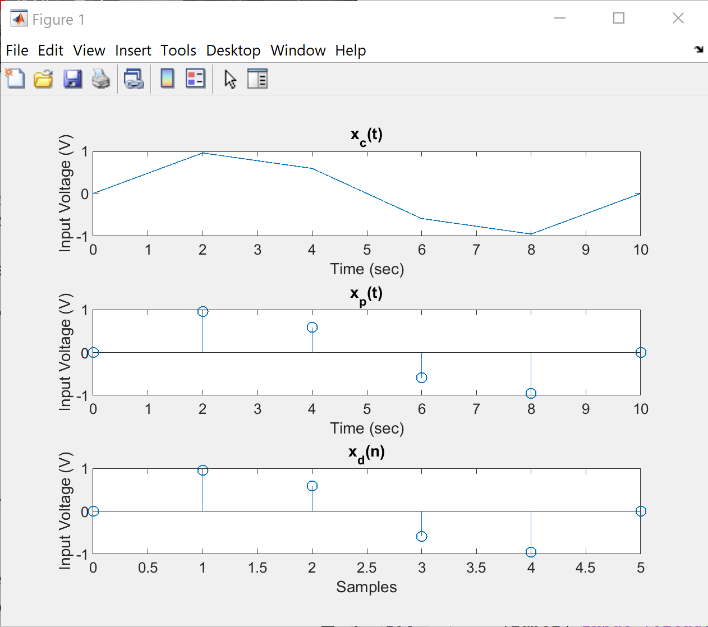


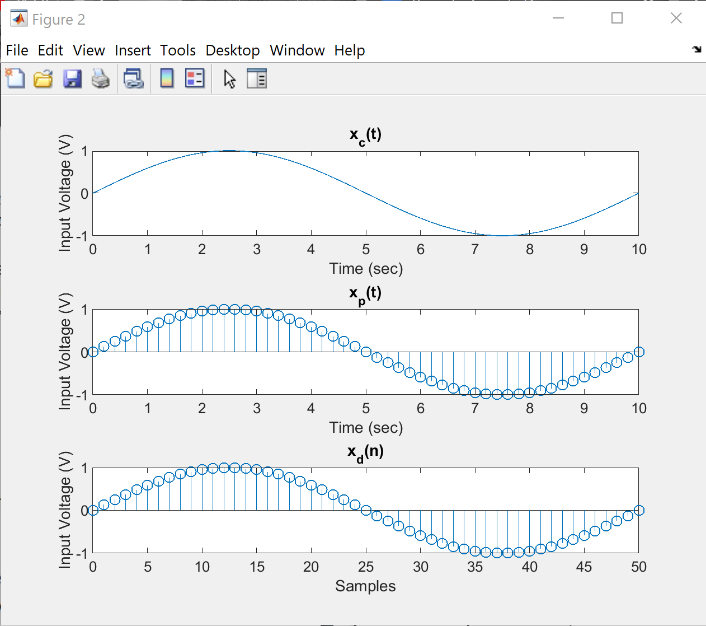


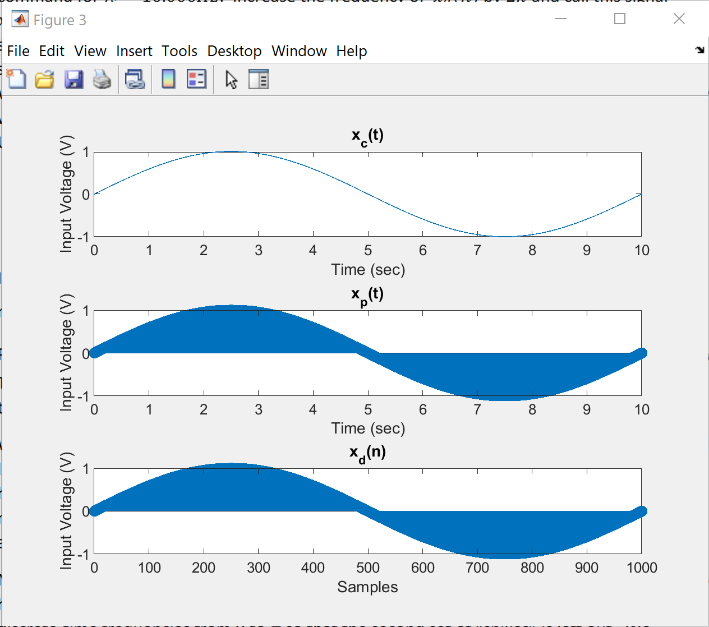
1. To attenuate the amplitude of the speed to 0.01% of the DC value, a switching frequency of 12000 rad/sec was chosen because it is a factor of 100 different than the previous switching frequency for the 1% value at 120 rad/sec.

**Problem 4:** A sample signal with three different sample periods was provided. DFT was introduced and used to find Fourier Transforms for given signals.

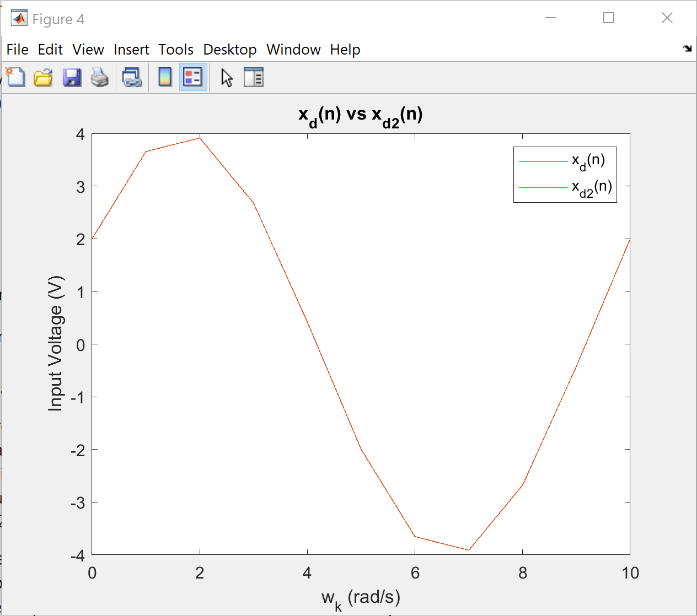
1. Using subplot, 3 graphs were shown for each sample period given, plotted over one period each.



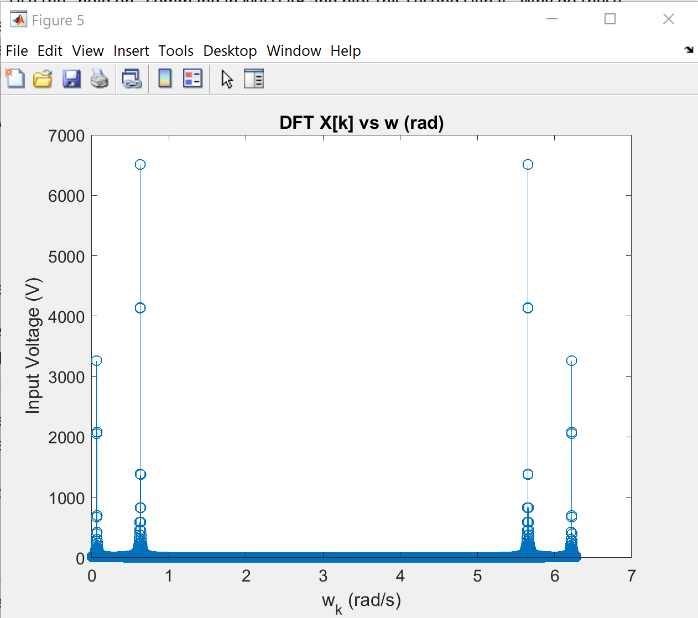




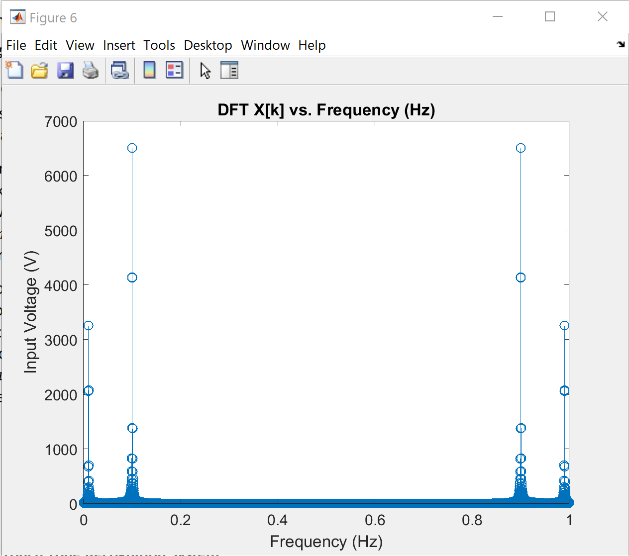
1. For both signals, the frequency is the same, but the graphs end up resting on top of each other because increasing the frequency by 2π only causes it to be another period ahead of the signal. This causes the graphs to look the same as both are sinusoidal graphs that have a fundamental period of 2π.

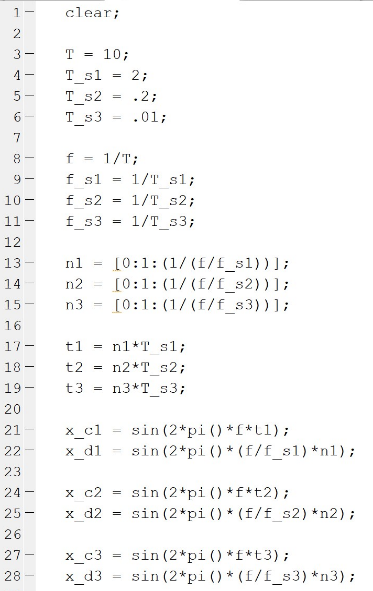


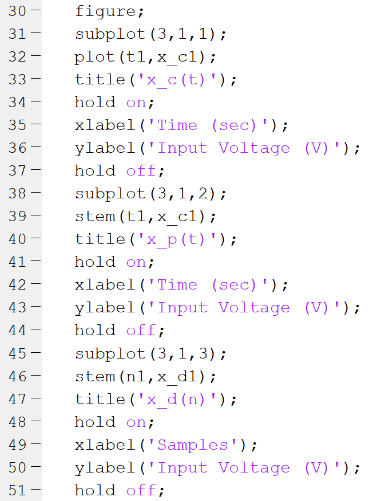
1. SEE IN CODE.
2. The relative values of the spikes for each wave make sense because at some point the input voltage will spike due to the input signal being real, thus being conjugate-symmetric within the frequency domain. If the input signal was imaginary, only one spike would be seen per wave.

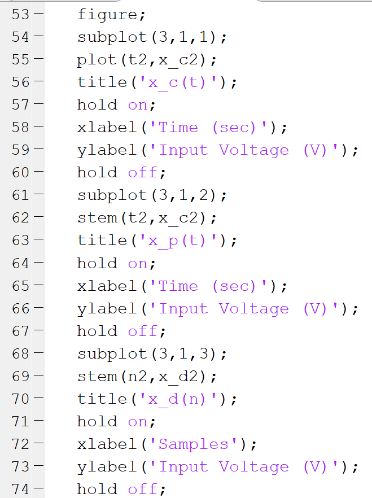


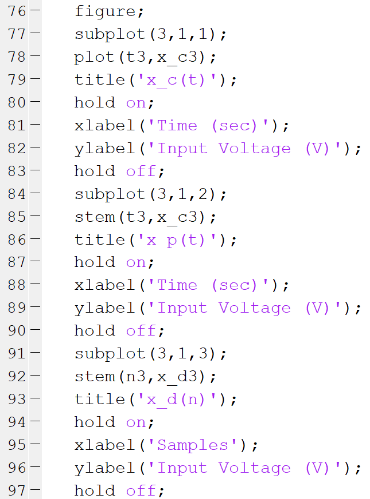
1. Plotted for frequency vs DFT X[k].

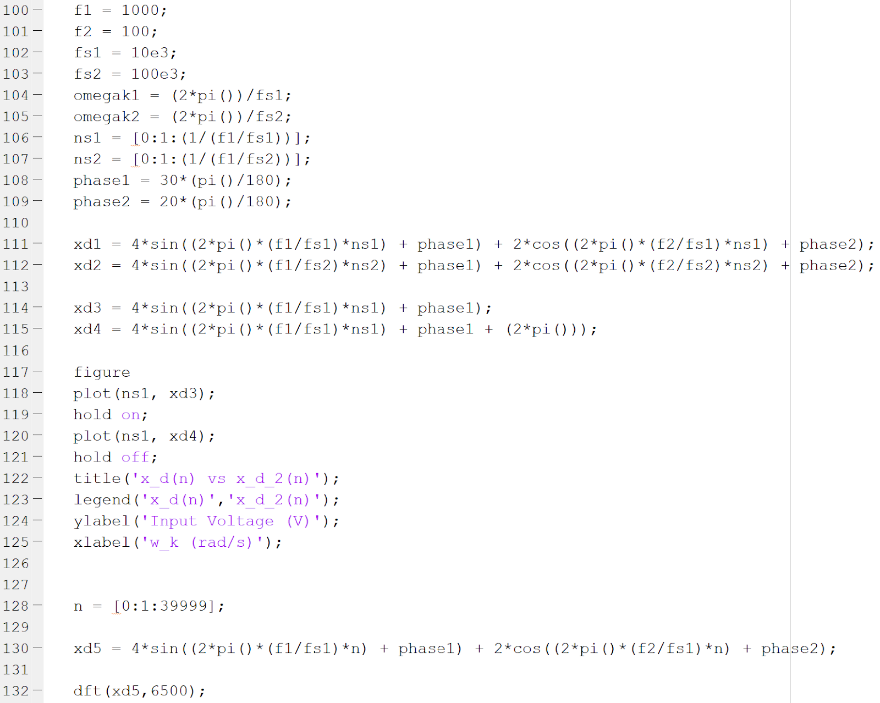


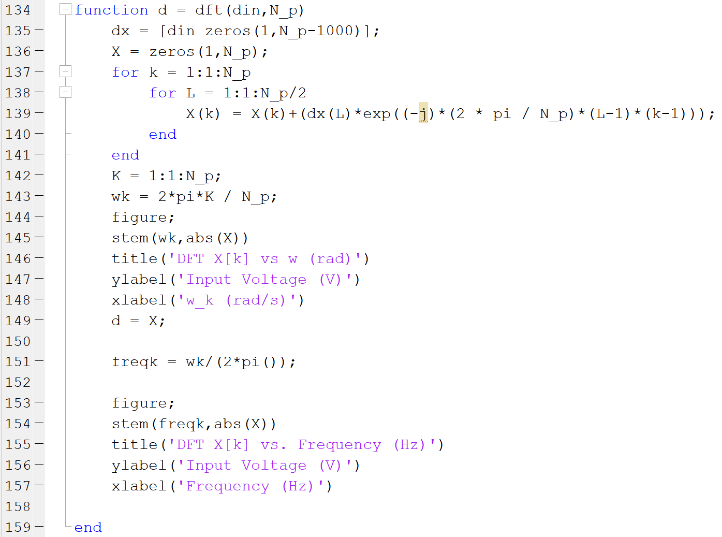






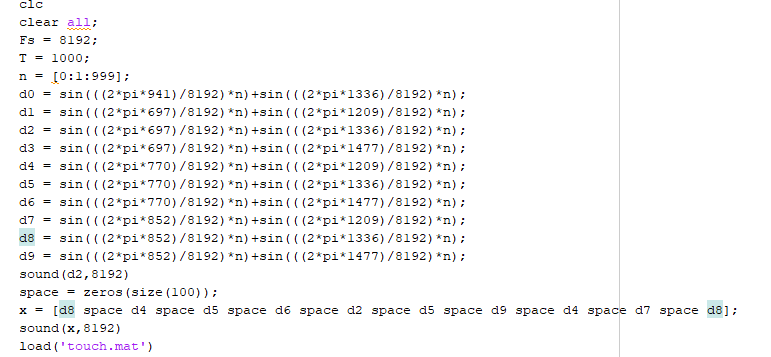


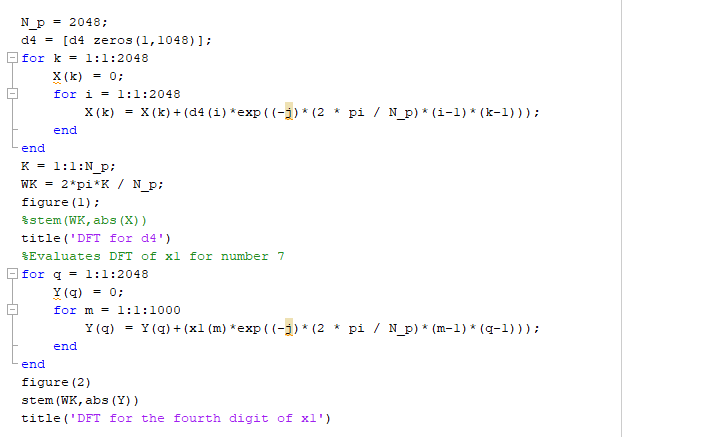


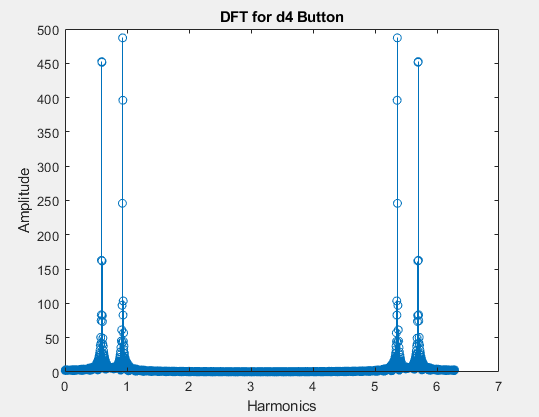


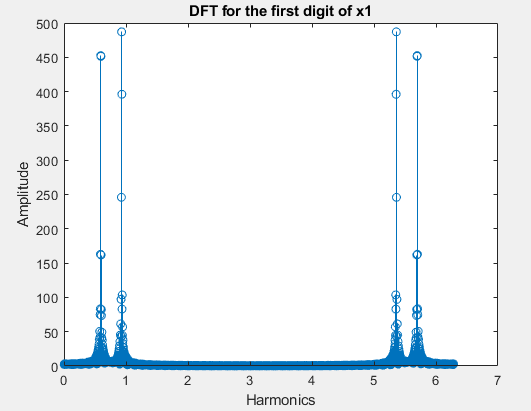
**Problem 5:** A vector consisting of seven digits of a mock phone number with each digit being 1000 samples separated by vectors of space was given.

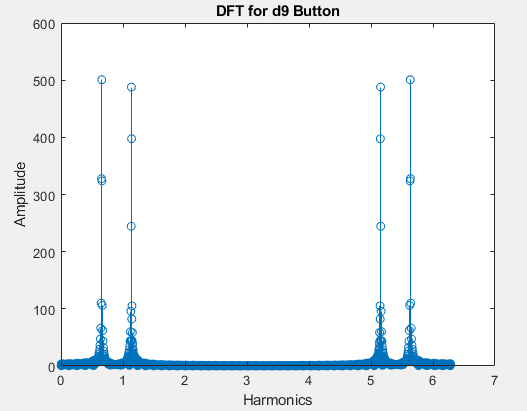
1. To represent the digits 0-9, ten vectors were created, each consisting of a sum of two sine waves with differing frequencies between each digit. The first sinusoidal frequency is the row of the touchpad, the second frequency is the column of the touchpad.
2. Sound(d3,fs) was used to play the frequency for #3, using d3 as the signal and fs as the sample frequency of 8192 Hz.
3. A zero-vector called space was used to delay the signals in between playing each tone, adding the effect of someone dialing a telephone. The space vector can easily be lengthened or shortened to increase or decrease the dial tone spacing.
4. Each digit was analyzed using a double nested loop in order to represent their digital Fourier Transforms. This resulted in plotting using stem to identify each digit by plot.
5. The touch.mat file supplied was loaded into the script in MATLAB and each digit was isolated and had its DFT graphed to compare it to the known digital signals from the given dial tone frequencies. By analyzing each number individually, we were able to cut down on inaccuracy by confusing plots and were able to come up with the correct phone number 491-5877, which correctly adds up to 41 as expected.

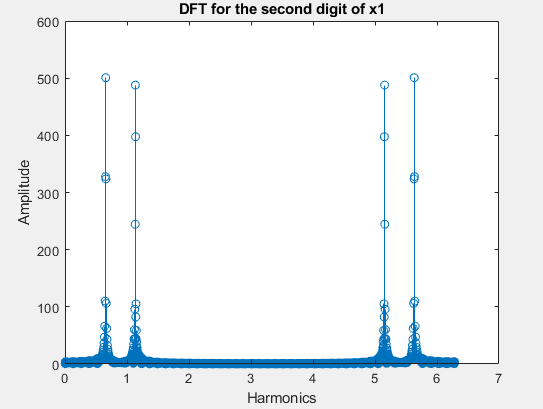


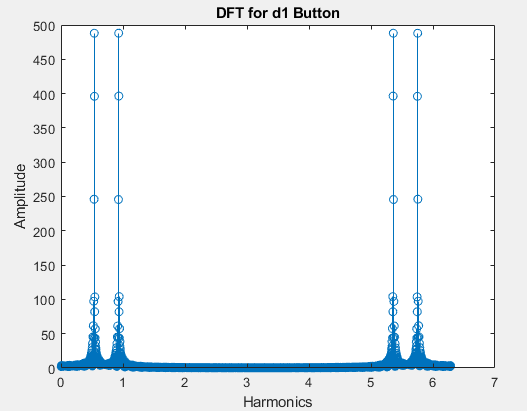


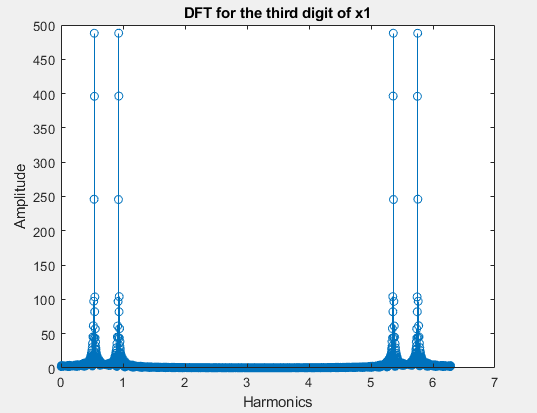


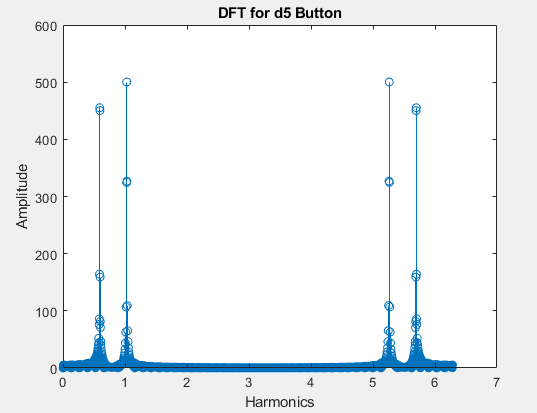


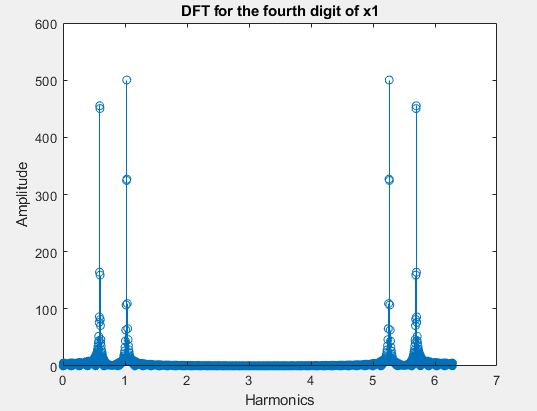


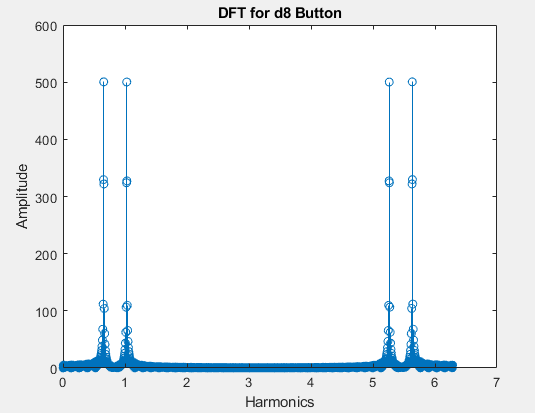


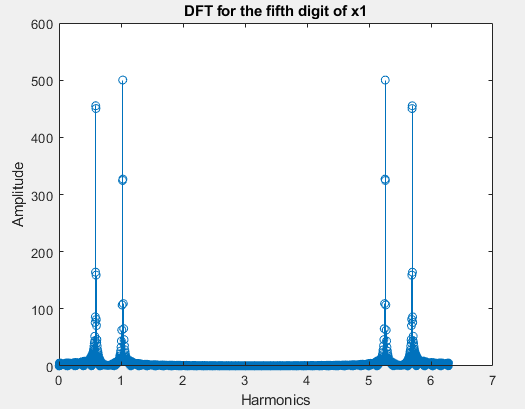


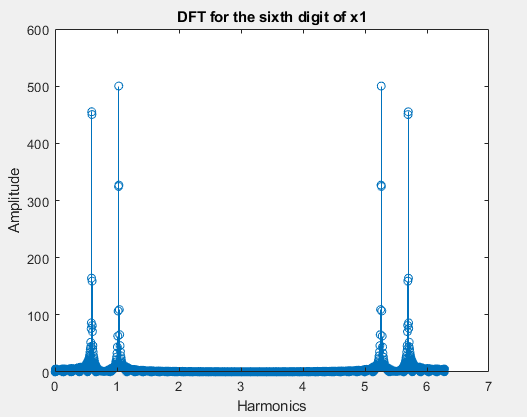
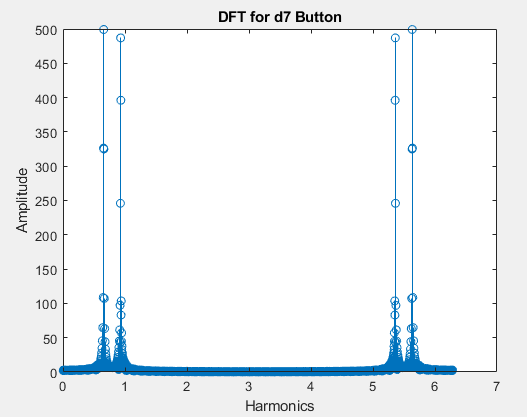


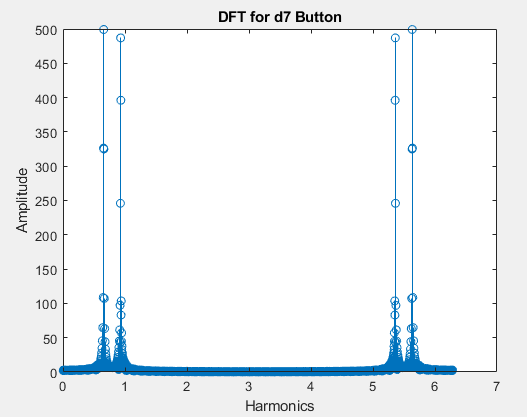


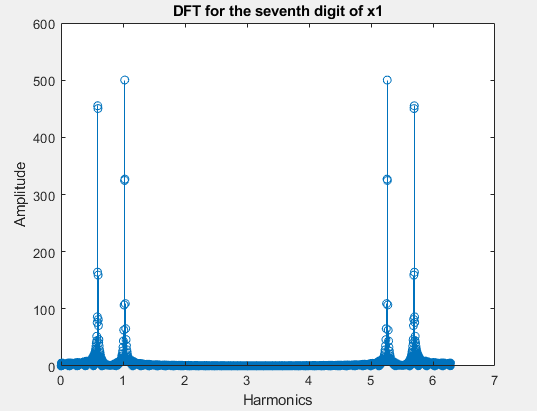












For each pair of graphs above, the first DFT graph shows the signal of the number and corresponds it to the matching frequency from the phone number within touch.mat.