

**Objectives:** In this lab, we will utilize the wave file recorded during the previous lab. Given such a file, can we figure out what the original digits are? In order to achieve the task, we will take advantage of Fourier transform.

**Tasking description:**

1. You are given a MATLAB script to begin with. You will use the DFT and FFT algorithms. By assuming that each length of the time domain signal corresponded to each digit is the same, we are able to transform each separated signal into the frequency domain using matrix operation.
2. Given the equation  $X = Wx$ , where  $x$  is the original time domain signal,  $W$  is the  $N$  by  $N$  square DFT matrix:

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(N-1)} \\ 1 & w^3 & w^6 & \cdots & w^{3(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix}$$

where  $w = e^{\frac{-j2\pi}{N}}$

Please use the equation above to calculate the original time domain signal. Note that  $X$  is a complex matrix.

3. Hopefully we also have `fft()` function to carry out the transformation. Please use `help fft` to see the detail. Are the matrices derived from different method the same? How about the complexity or consumed time?
4. Now, use FFT to find the frequencies in each separated signal. You should plot the spectra to visualize the answer.

5. (optional) If the lengths of the original time domain signals are not the same, can you still use FFT to generate digits automatically?

Things to turn in on LMS:

Please write a function, `phoneNumStr = getPhoneNum( WAV_PATH )`, where the name of the function is strictly limited to be `getPhoneNum`. In addition, `WAV_PATH` is the input path of the function, and `phoneNumStr` is the string format of the generated cellphone digits. Last but not least, we also provide our answers to the 8 test data for you to verify your implementation. Please refer to the code below to generate the 8 answers:

```
load TA_answer_partial.mat
keys = answer_map_partial.keys;
for ii = 1:length(keys)
    fprintf([keys{ii} ': ' answer_map_partial(keys{ii}) '\n']);
end
```

Note that `TA_answer_partial.mat` is a map container. For more information please access to <http://www.mathworks.com/help/matlab/map-containers.html>

Please turn in the `getPhoneNum` function only.

**A further question leaving for you:**

If your result of the implementation is not as good as your expectation, what are the factors that may affect the performance?