Phone dial-up analysis
By Fourier Transformation

Numerical Computation (1001) (Mr. Tian TANG)

[Semester 1 of 2021-2022]

Group 6

Group Members

许嘉芸 **Jenny 1930026140**

黄一纯 **Athena 1930026050**

韩昕尧 **Daisy 1930026040**

李迪欣 **Andi 1930026063**

Content

1.Introduction

1.1 Objective

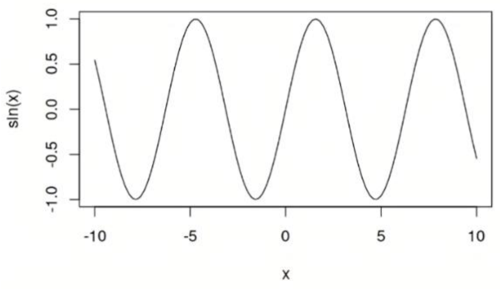
Seeing many novels have plots about using sound frequency to recognize the telephone number, our group is interested in making a phone number detector. Due to the fact that different keys of the telephone have different pitches and corresponding frequencies when pressed, we can use the Fourier transform to recognize specific numbers by frequency.

In this report, we will first give the background knowledge of sound detection and our design ideas. Then, we will further explain our methods and code implementation. Finally, we will give some suggestions for practical applications.

1.2 Background knowledge

In order to analyze sound, we first need to understand some background knowledge of sound. Sound consists of pitch, volume, and pitch, and is correspondingly determined by the shape, amplitude, and frequency of its waves. Among them, the pitch and volume of a telephone key are fixed, only the pitch is different, which means the frequency is different.

What's more, the telephone used in our research is an old-fashioned one, and when dialing it produces a combination of high and low frequency SIN signals. Then the telephone resolves two frequencies from the combined signal to know which key was pressed.

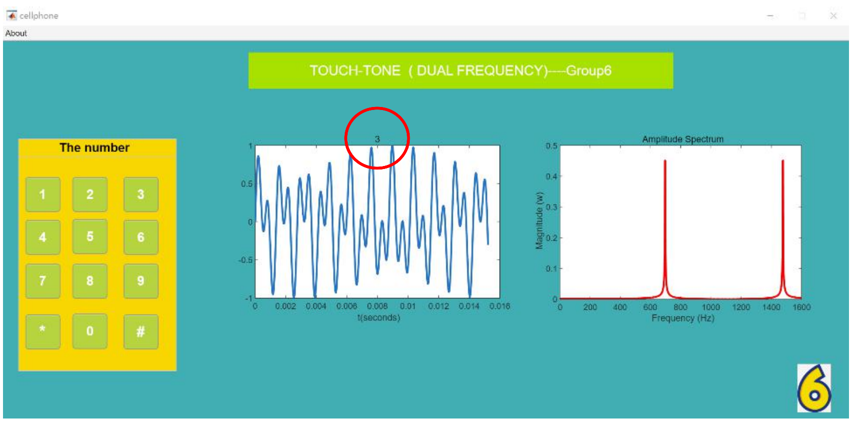


So, as long as we take the sound wave of the phone, we can resolve these two frequencies by Fourier transform. Before the transformation, the wave is mixed, and changes over time. After transformation, the wave is extracted according to its frequency. Then by looking up the corresponding frequency table, we can find the exact key number.

1.3 Design idea

We use Matlab to generate an interactive telephone interface.

When pressing a key, Matlab simulates the phone to generate a pair of SIN waves and release the sound. It also plots the sound waves before and after the Fourier transform, and displays the corresponding number. For example, when we press 3, it shows 3 at the top accordingly.



2.Data Preprocessing

By comparing the sound waves of the phone buttons in Matlab with the real sound waves of the phone buttons, we determine whether the model is correct.

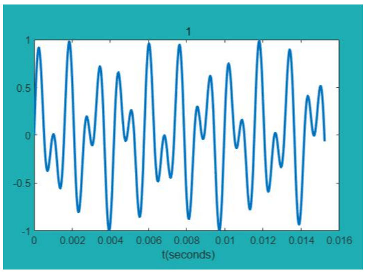
The sound of each phone button was recorded individually, with a total of 12 recordings, and then their sound waves were displayed separately using Adobe Audition software, and finally compared with the sound waves generated by each key in Matlab.

Compare waves of each button

The frequencies of sound vibrations in these two pictures are the same. Because the speed of vibration is the same at the same time, and the number of sound waves in the pictures is 19. Therefore, they have the same pitch.

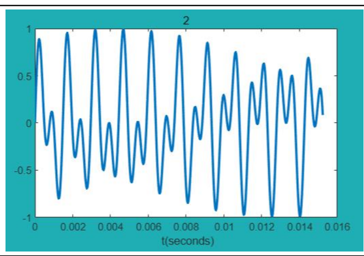
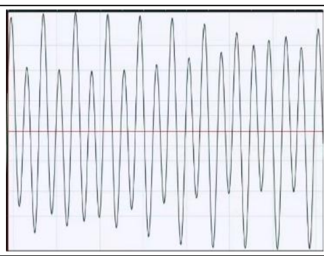
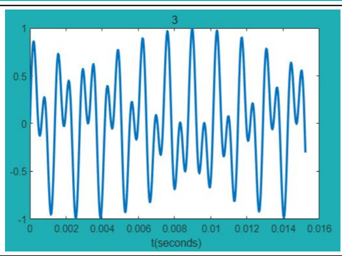
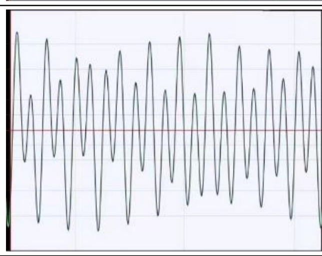
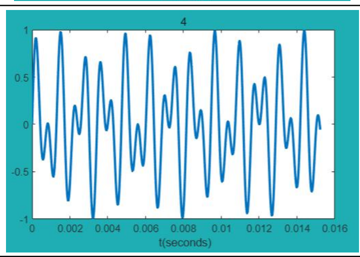
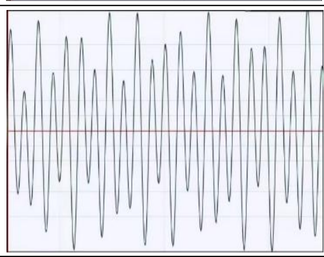
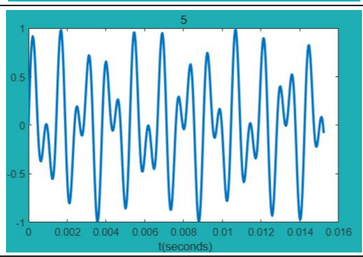
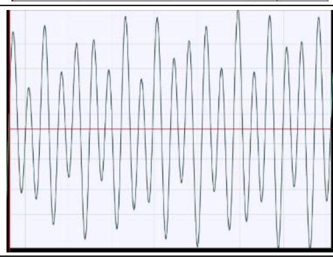
We can see that they have a similar waveform, so their timbre is identical, and then they have the common sound producing body. So, these two waves are equal.

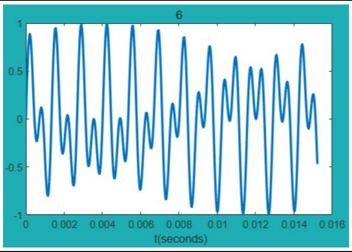
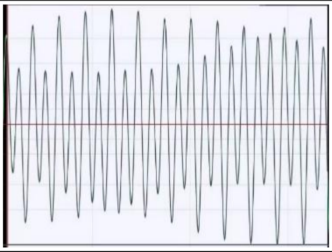
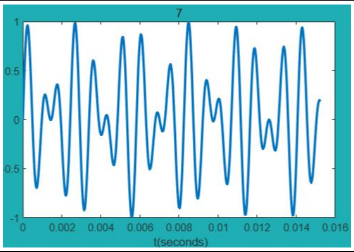
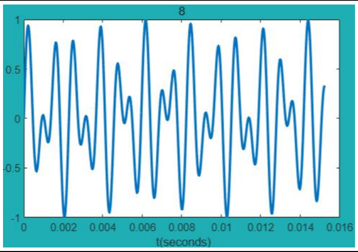
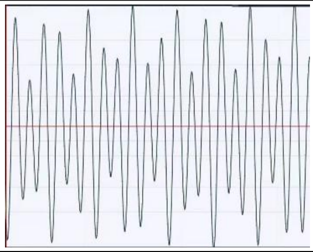
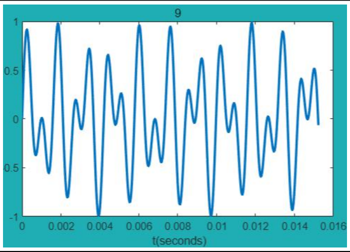
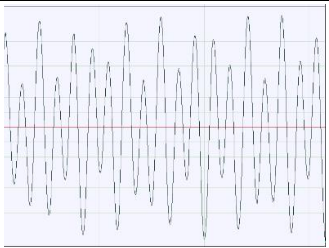
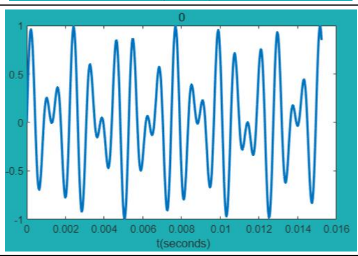
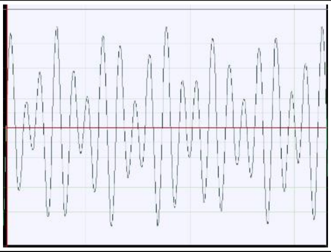
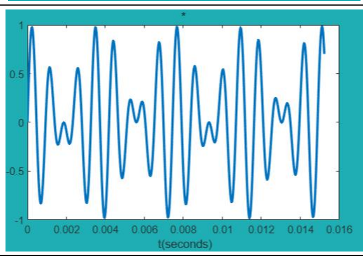
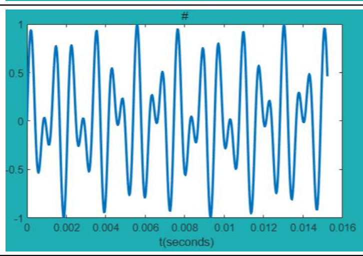
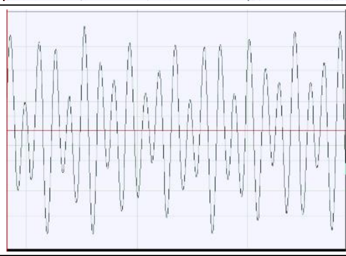
For button 1:



Therefore, under the same principle, we can know that the sound waves of these keys are identical.

For the remaining bottoms:

2		
3		
4		
5		

6	6 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
7	7 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
8	8 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
9	9 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
0	0 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
*	* 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	
#	# 1 0.5 0 -0.5 -1 0 0.002 0.004 0.006 0.008 0.01 0.012 0.014 0.016 t(seconds)	

Then we can get that the sound of the button generated in Matlab is the same as the sound of the real button.

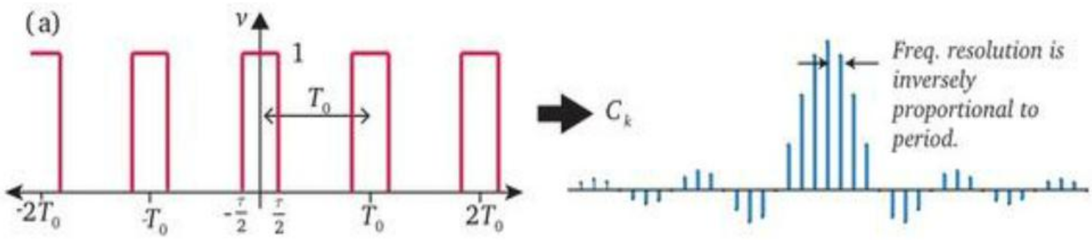
3.DFT&FTT Theories

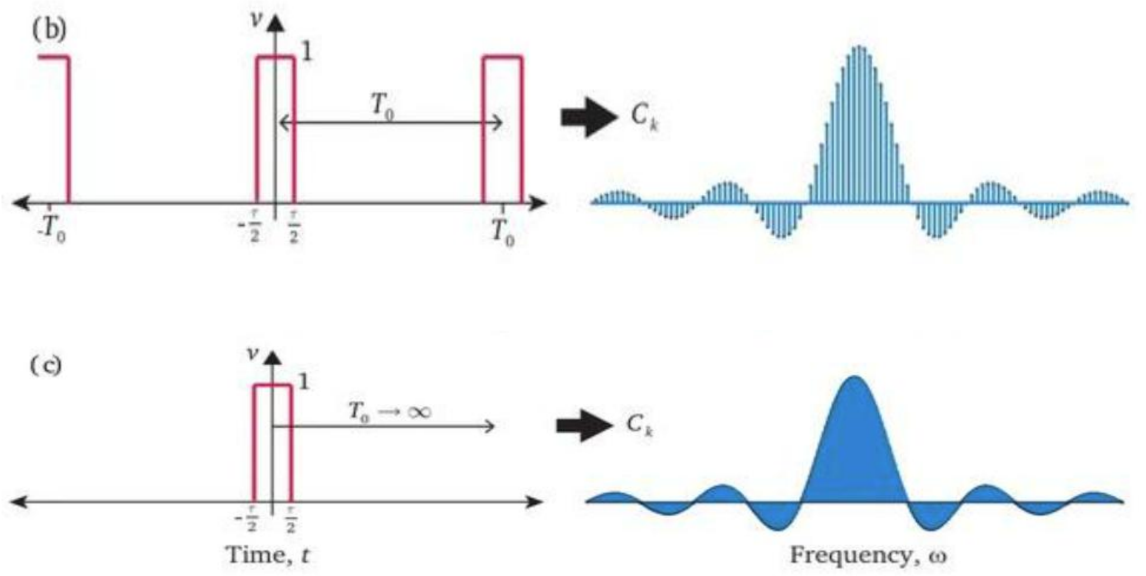
Non-periodic signals can be regarded as periodic signals with infinite periods. We call the Fourier analysis method of non-periodic signals the Fourier transform. There are so many functions that can explain the Theories of DFT and FFT. So this part will contain the Fourier Transformation to the Discrete Fourier Transformation and Fast Fourier Transformation.

3.1DFT

When the period T of the periodic function gradually tends to infinity, since $\omega=2\pi/T$, the frequency spectrum of the periodic signal is discrete, and the discrete interval is ω. So when T tends to infinity, ω tends to 0, the discrete interval gradually becomes 0, and the frequency spectrum becomes a continuous spectrum.

The graph is like that:





From the above figure, we can see that when the period T of the pulse signal continues to increase, the spectrum width gradually narrows. At this time we draw the spectral density function of F(nw)/w. The red rectangle in the picture has a width of w, a length of F(nw)/w, and an area of F(nw).

Continue DFT:X(F)=$\int_{-\infty}^{\infty} x(t)e^{-j2\pi Ft}dt$

Discrete DFT:$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$,(the K-th frequency, it is evaluating n of N sample).

Also, using the Euler's Formula to expand this function:

If take e to the power of some numbers of t tours i, then we can reach x amount of unit counterclockwise of a circle of radius one.

$X_k = F(w) = \int_{-\infty}^{\infty} f(t)e^{-2\pi wt}dt$, and the F(w) is the function in frequency domain and f(t) is function in time domain.

Also $e^{2\pi}$ $is\ a\ full\ rotation\ of\ the\ circle$, and the $f(t)e^{-2\pi wt}$ is the original complex wave f(t) and cropping it around the circle.

Discrete:$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$

$X_k = x_0 e^{-b_0 j} + x_1 e^{-b_1 j} + \ldots\ldots + x_n e^{-b_{N-1} j}$

$e^{ix} = cosx + jsinx(\ using\ Euler's\ Formula)$

$X_k = x_0[cos(-b_0) + jsin(-b_0)] + \ldots\ldots$

X_k=$A_k + jB_k$

Finally, computing the time complexity of the DFT:

$$F_K = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$$

The time complexity is the O(n^2).

3.2FFT

Fast Fourier Transform (FFT), FFT uses dividing and conquering method, recursively break down DFT into small's DFT. Therefore, it is a method for efficiently computing the discrete DFT.

$F_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$,

then dividing into:

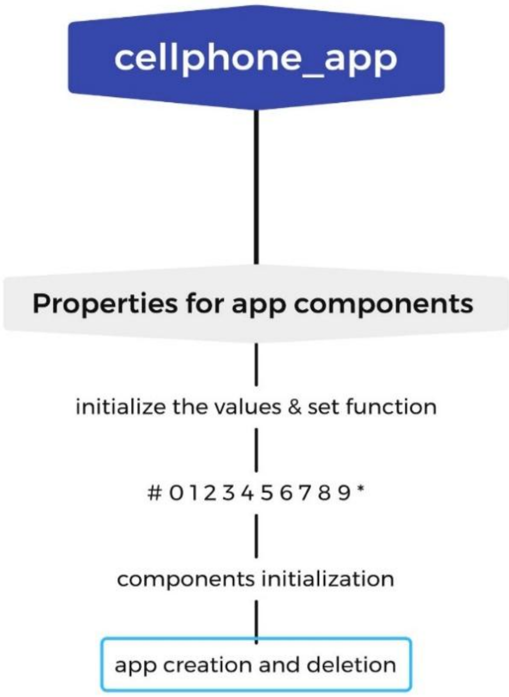
$$F_k = \sum_{d=0}^{\frac{N}{2}-1} x_{2m} e^{\frac{-j2\pi k(2d)}{N}} + \sum_{d=0}^{\frac{N}{2}-1} x_{2m+1} e^{\frac{-j2\pi k(2m+1)}{N}}$$

Then we can get T(N)=2T($\frac{N}{2}$)+O(n)

Then get the O(n)=nlogn.

4. Code Analysis

The whole code is divided into four parts:



First, properties for app components, then initializing variables and setting basic functions, then designing phone numbers respectively, and then initializing components and app creation and deletion.

The code about properties for app components is automatically generated by Matlab.

```
% Properties that correspond to app components
properties (Access = public)
    figure1    matlab.ui.Figure
    About      matlab.ui.container.Menu
    uipanel2   matlab.ui.container.Panel
    star       matlab.ui.control.Button
    diez       matlab.ui.control.Button
    zero       matlab.ui.control.Button
    seven      matlab.ui.control.Button
    eight      matlab.ui.control.Button
    nine       matlab.ui.control.Button
    one        matlab.ui.control.Button
    two        matlab.ui.control.Button
    five       matlab.ui.control.Button
    four       matlab.ui.control.Button
    three      matlab.ui.control.Button
    six        matlab.ui.control.Button
    HSHS       matlab.ui.control.Label
    Image      matlab.ui.control.Image
    axes1      matlab.ui.control.UIAxes
    axes2      matlab.ui.control.UIAxes
end
```

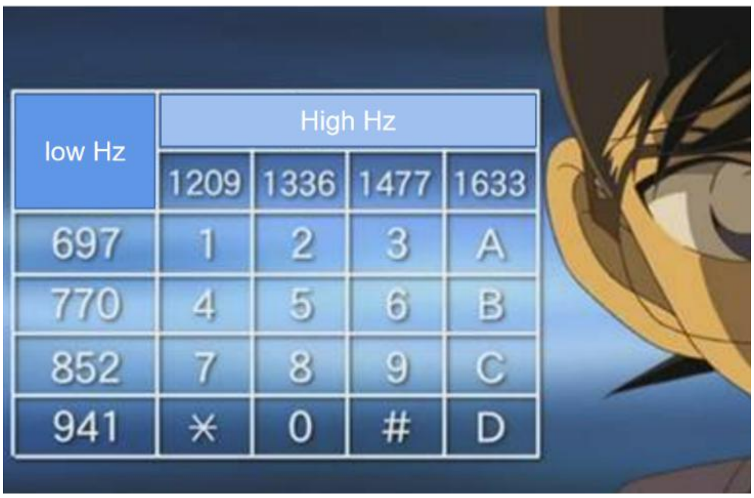
and set function and initialization variables are the core code. In the set function, we defined the length of frequency, processed the data, made FFT, and set the correlation function of independent variable and dependent variable for the making of chart.

```
methods (Access = private)
    function [Power, N, f] = FFT(app, y, Fs)
        L=length(y);%the length of frquency
        N=ceil(log2(length(y)));%process the data to avoid unfriend number
        fy=fft(y,2^N)/(L/2);%fft transformation
        Power=fy.*conj(fy);%get the y of amplitude spectrum
        f=(Fs/2^N)*(0:2^(N-1)-1);%get the x of amplitude spectrum
        return
    end

end
```

When setting the initial value, we set the sample rate and time. The most important thing is the setting of high group frequency (1209,1336,1477) and low frequency (697,770,852,951). Each dial pitch is composed of two frequencies. As shown below:

```
global t Fs fr fc one two three four five six seven eight nine zero star diez
Fs=32768;           % Sampling rate
t=0:1/Fs:0.25;      % time points
fr=[697 770 852 941];%Basic frequencies
fc=[1209 1336 1477];%high frenquencies
```



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

In the next code for generating dialing keys, the GUI part about the display is generated by Matlab, the wave is composed of two frequencies.

```
% Button pushed function: diez
function diez_Callback(app, event)
    % Create GUIDE-style callback args - Added by Migration Tool
    [~, ~, handles] = convertToGUIDECallbackArguments(app, event); %#ok<ASGLU>

    % hObject    handle to diez (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    global t fr fc Fs diez
    y1=sin(2*pi*t*fr(4));
    y2=sin(2*pi*t*fc(3));
    diez=(y1+y2)/2;%get the frequence of number in phone
    axes(handles.axes1);
    plot(t(1:500),diez(1:500),'LineWidth',2) ,xlabel(' t(seconds)'),title('#');%plot the frquency of number
    sound(diez,Fs)%send  signal to voice at sample rating of Fs.
    [Power,~,f]=FFT(app, diez,Fs);
    axes(handles.axes2);
    plot(f(1:1600),sqrt(Power(1:1600)),'r','LineWidth',2),  xlabel('  Frequency (Hz)'), ylabel(' Magnitude (w)');%plot the amplitude spectrum of number
    title(' Amplitude Spectrum'),axis([0 1600 0 0.5])

    F=sort([fr fc]);
    set(gca,'XTickLabel',F);
end
```

The two plots correspond to the two generated pictures respectively.

```
% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create figure1 and hide until all components are created
        app.figure1 = uifigure('Visible', 'off');
        app.figure1.Color = [0.1176 0.6824 0.702];
        app.figure1.Colormap = [0 0 0.5608;0 0 0.6235;0 0 0.6863;0 0 0.749;0 0 0.8118;0 0 0.8745;0 0 0.9373;0 0 1;0 0.0627 1;1 1 1;0 0.1882 1;0 0.251 1;0 0.3137 1;0 0.3765 1;0 0.439
        app.figure1.Position = [535 -181 1234 551];
        app.figure1.Name = 'cellphone';
        app.figure1.Resize = 'off';
        app.figure1.CloseRequestFcn = createCallbackFcn(app, @figure1CloseRequest, true);
        app.figure1.HandleVisibility = 'callback';
        app.figure1.Tag = 'figure1';

        % Create About
        app.About = uimenu(app.figure1);
        app.About.Accelerator = 'A';
        app.About.Text = 'About';
        app.About.Tag = 'About';

        % Create uipanel2
        app.uipanel2 = uipanel(app.figure1);
        app.uipanel2.Title = '            The number';
        app.uipanel2.BackgroundColor = [0.9882 0.8471 0.0314];
        app.uipanel2.Tag = 'uipanel2';
        app.uipanel2.FontWeight = 'bold';
        app.uipanel2.FontSize = 18;
        app.uipanel2.Position = [25 72 230 339];

        % Create star
        app.star = uibutton(app.uipanel2, 'push');
        app.star.ButtonPushedFcn = createCallbackFcn(app, @star_Callback, true);
        app.star.Tag = 'star';
        app.star.BackgroundColor = [0.702 0.8314 0.302];
        app.star.FontSize = 24;
        app.star.FontWeight = 'bold';
        app.star.FontColor = [1 1 1];
        app.star.Position = [10 32 50.9612480314961 50.9612480314961];
        app.star.Text = '*';

        % Create diez
        app.diez = uibutton(app.uipanel2, 'push');
        app.diez.ButtonPushedFcn = createCallbackFcn(app, @diez_Callback, true);
        app.diez.Tag = 'diez';
        app.diez.BackgroundColor = [0.702 0.8314 0.302];
        app.diez.FontSize = 21.3333333333333;
        app.diez.FontWeight = 'bold';
        app.diez.FontColor = [1 1 1];
        app.diez.Position = [153 32 50.961248031496 50.9612480314961];
        app.diez.Text = '#';
```

```
            % Show the figure after all components are created
            app.figure1.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = cellphone_App(varargin)

            runningApp = getRunningApp(app);

            % Check for running singleton app
            if isempty(runningApp)

                % Create UIFigure and components
                createComponents(app)

                % Register the app with App Designer
                registerApp(app, app.figure1)

                % Execute the startup function
                runStartupFcn(app, @(app)cellphone_OpeningFcn(app, varargin{:}))
            else

                % Focus the running singleton app
                figure(runningApp.figure1)

                app = runningApp;
            end

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.figure1)
        end
    end
end
```

5. Conclusion

Our topic is about deciphering the dialed number by dialing the voice. First, we performed the data and processing, then we introduced the theory related to DFT and FFT, and then we analyzed the code. Finally, after we showed the sound waves of different buttons through the code, we got the correctness of our model by comparing it with the real sound waves.

Our model can also be applied in life to help police officers solve crimes, and other meaningful situations.