四川大學
SICHUAN UNIVERSITY

# Generation of Typical Digital Signals and Its Frequency Spectrum Analysis by Using MATLAB

Fundamentals and Experiments

Present By: Cui Ao & Li Shangwen

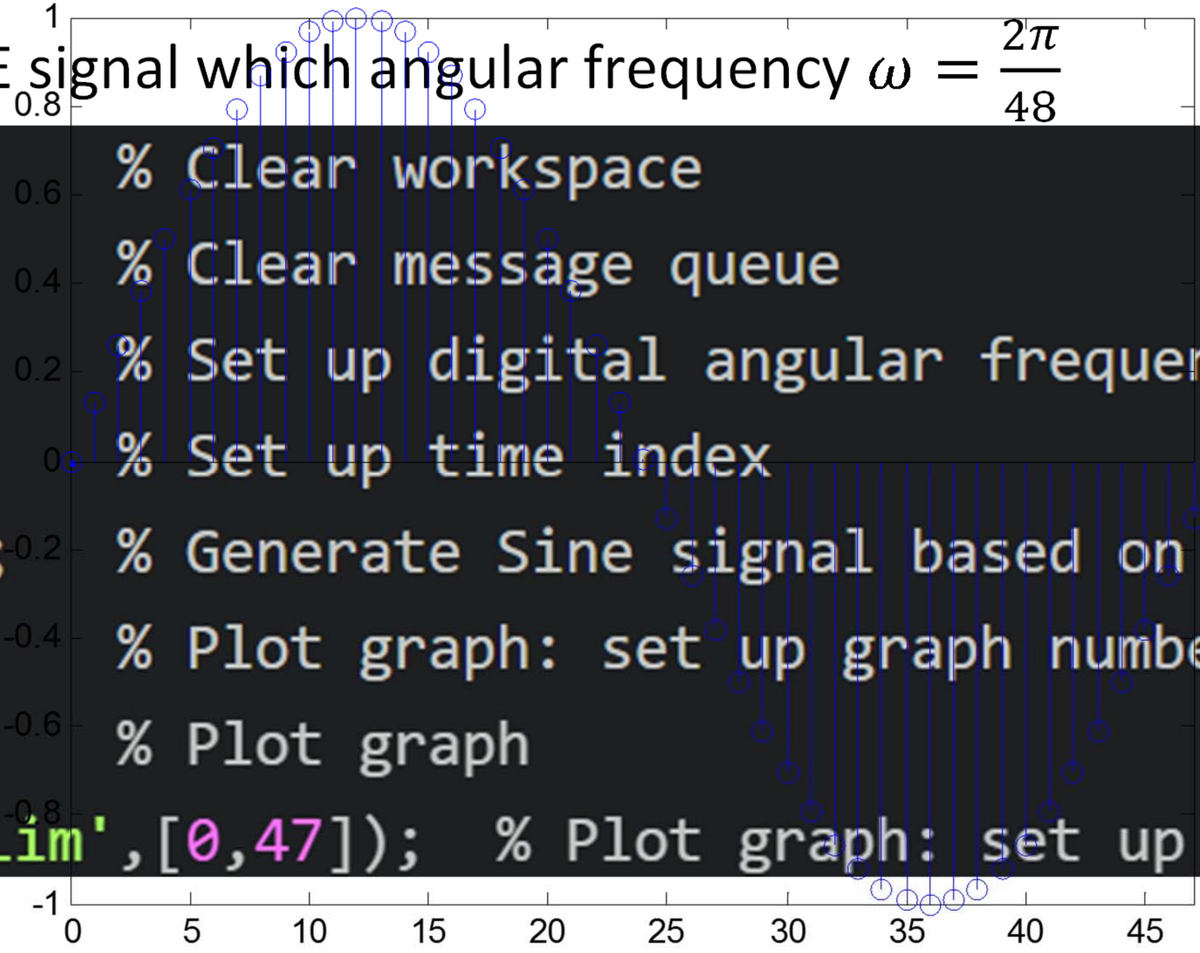
# Content

1. Generate Typical Signals in MATLAB;
2. Import an Audio File to MATLAB Workspace;
3. Analyze Signals by Using DTFT/DFT in MATLAB;
4. Experiment Tasks and Goals;

# 1. Generate Typical Signals

# 1. Generate Typical Signals in MATLAB

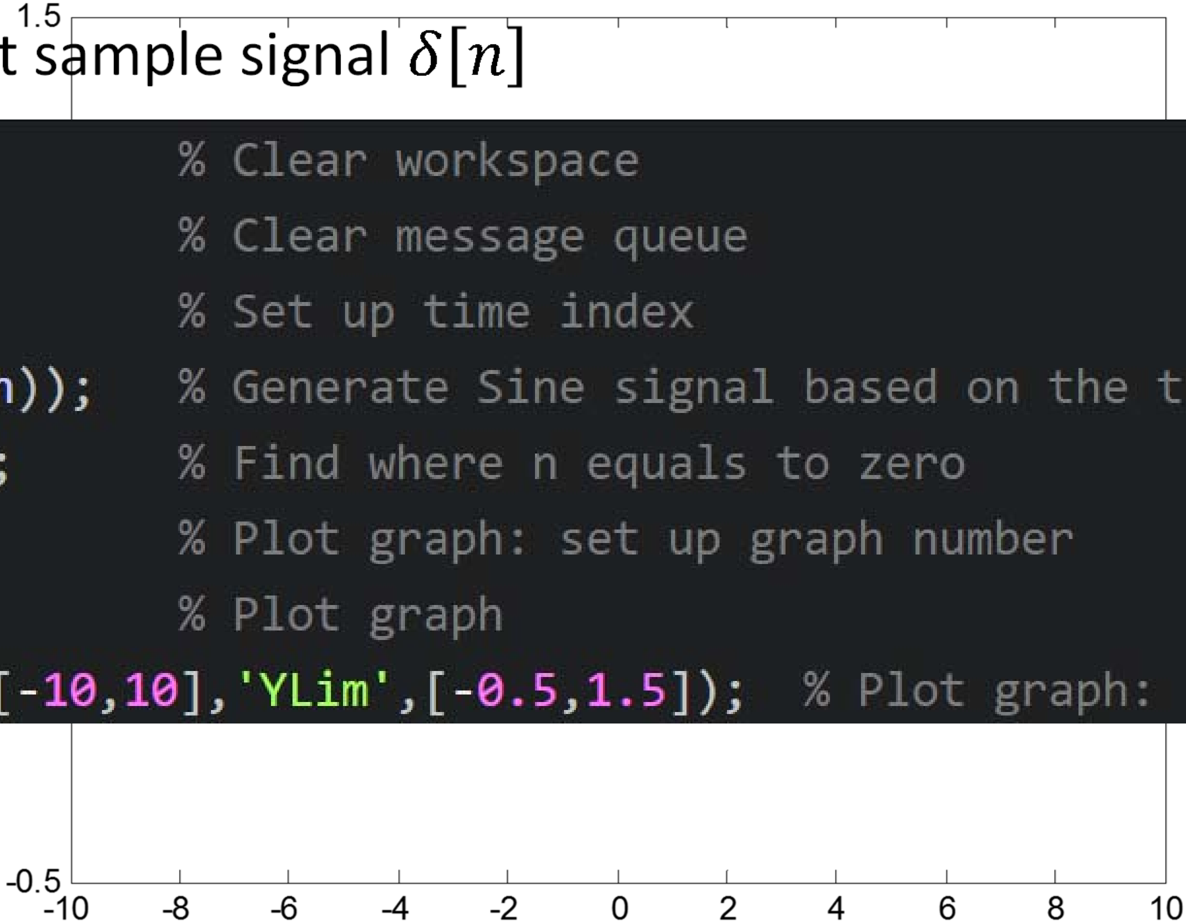
Generate a SINE signal which angular frequency $\omega = \frac{2\pi}{48}$




```
clear;              % Clear workspace
clc;                % Clear message queue
w=2*pi/48;          % Set up digital angular frequency
n=0:1:47;           % Set up time index
y=sin(w*n);         % Generate Sine signal based on the time index
figure(1)           % Plot graph: set up graph number
stem(n,y);          % Plot graph
set(gca,'XLim',[0,47]);  % Plot graph: set up proper axis
```

# 1. Generate Typical Signals in MATLAB

Generate an unit sample signal $\delta[n]$

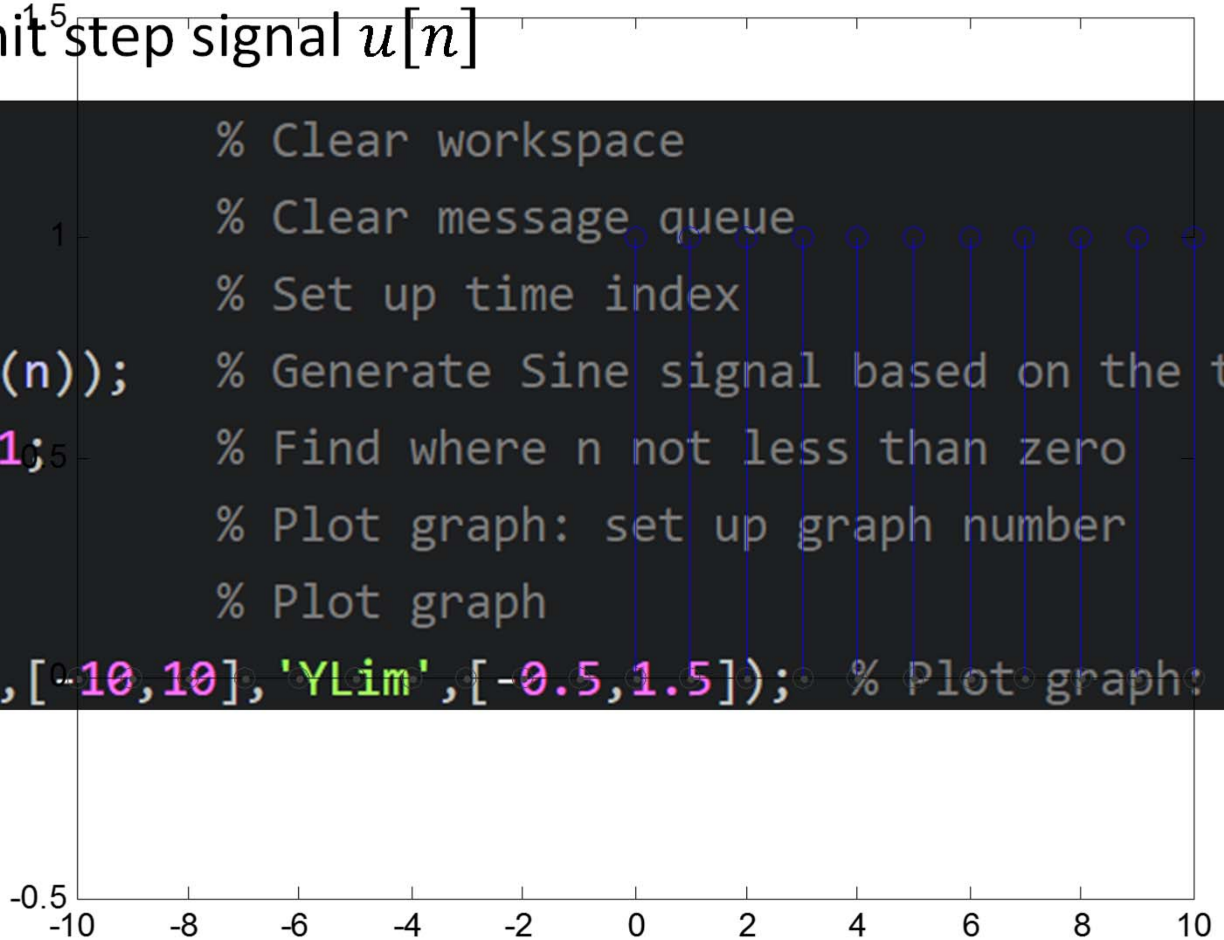



```
clear;                      % Clear workspace
clc;                        % Clear message queue
n=-10:1:10;                 % Set up time index
y=zeros(length(n));         % Generate Sine signal based on the time index
y(find(n==0))=1;            % Find where n equals to zero
figure(1)                   % Plot graph: set up graph number
stem(n,y);                  % Plot graph
set(gca,'XLim',[-10,10],'YLim',[-0.5,1.5]);  % Plot graph: set up proper axis
```

# 1. Generate Typical Signals in MATLAB

Generate an unit step signal $u[n]$

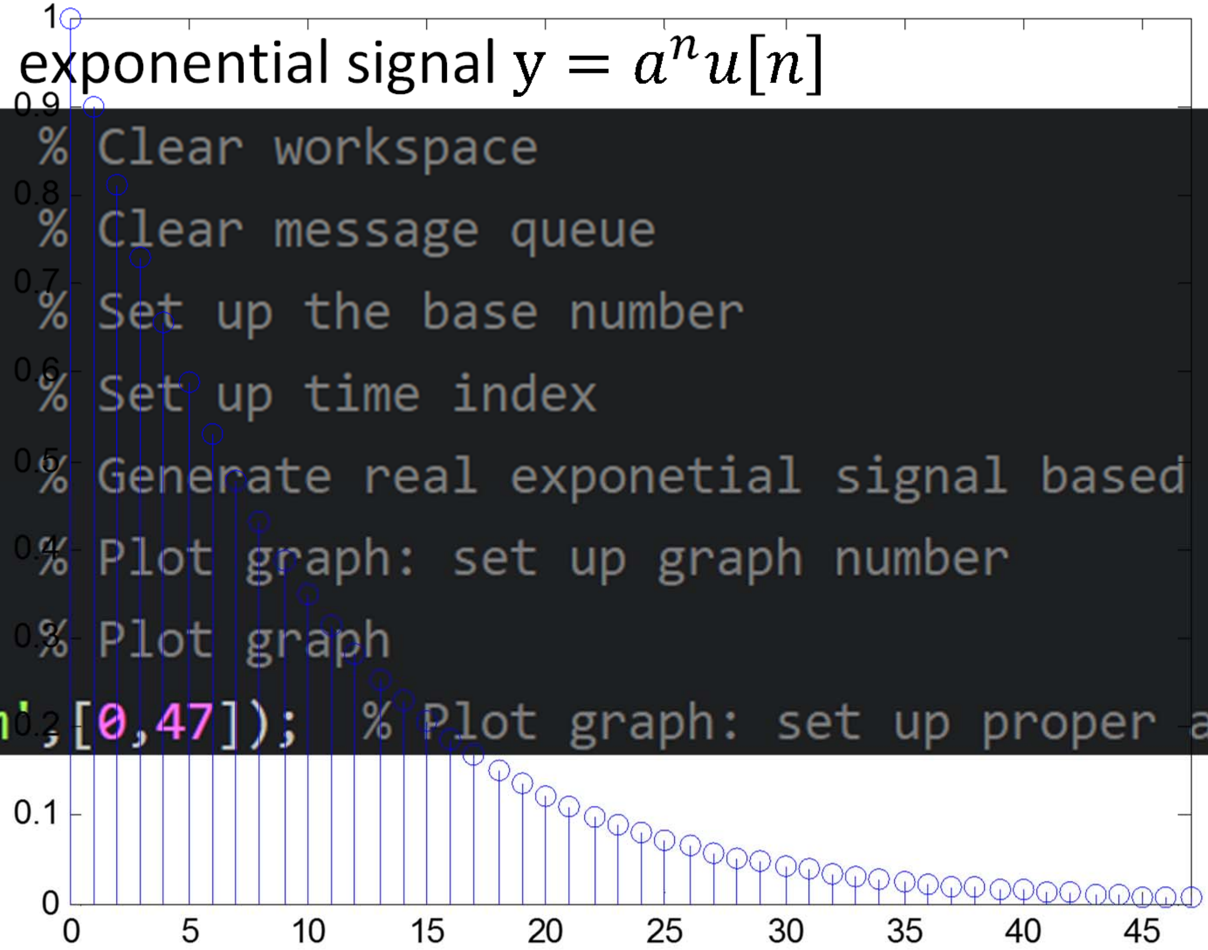
```
clear;                       % Clear workspace
clc;                         % Clear message queue
n=-10:1:10;                  % Set up time index
y=zeros(length(n));          % Generate Sine signal based on the time index
y(find(n>=0))=1;             % Find where n not less than zero
figure(1)                    % Plot graph: set up graph number
stem(n,y);                   % Plot graph
set(gca,'XLim',[-10,10],'YLim',[-0.5,1.5]);  % Plot graph: set up proper axis
```



# 1. Generate Typical Signals in MATLAB

Generate a real exponential signal $y = a^n u[n]$

```
clear;                  % Clear workspace
clc;                    % Clear message queue
a=0.9;                  % Set up the base number
n=0:1:47;               % Set up time index
y=a.^n;                 % Generate real exponetial signal based on the time index
figure(1)               % Plot graph: set up graph number
stem(n,y);              % Plot graph
set(gca,'XLim',[0,47]);  % Plot graph: set up proper axis
```



# 1. Generate Typical Signals in MATLAB

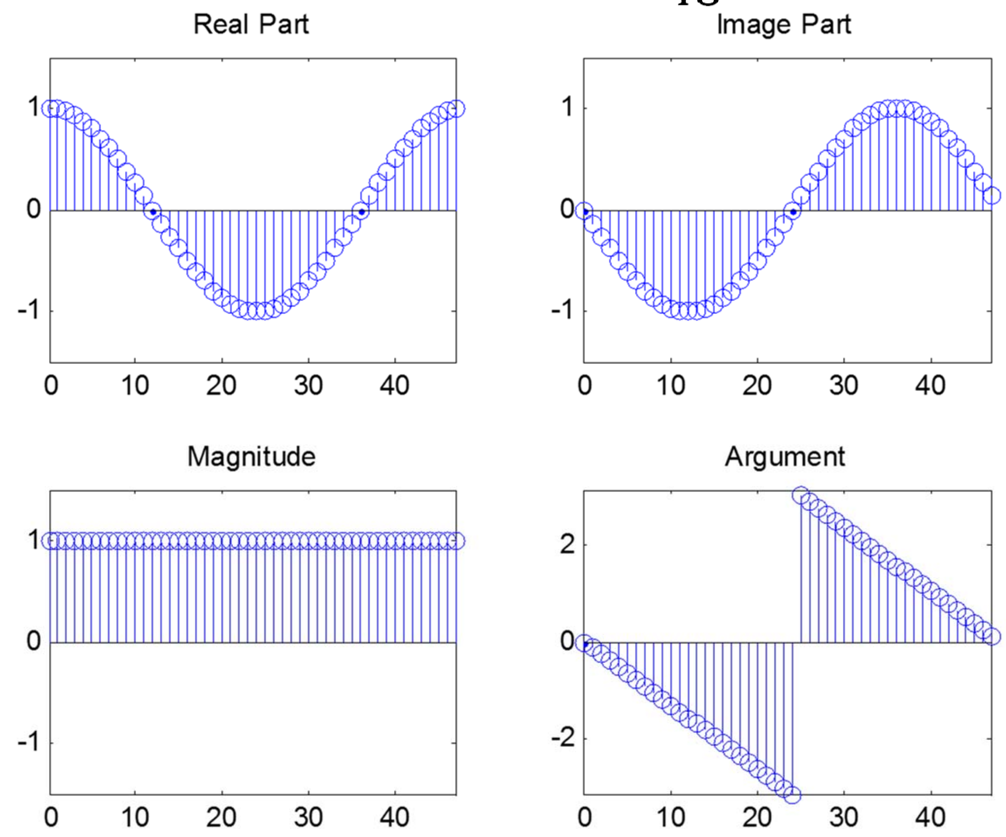
Generate a complex exponential signal y $= e^{j\omega n}u[n]$, where $\omega = \frac{2\pi}{48}$

```
clear;          % Clear workspace
clc;            % Clear message queue
w=2*pi/48;      % Set up digital angular frequency
n=0:1:47;       % Set up time index
y=exp(-1i*w*n); % Generate complex exponetial signal based on the time index
y_re=real(y);   % Calculate the real part of y
y_im=imag(y);   % Calculate the image part of y
y_mag=abs(y);   % Calculate the magnitude of y
y_arg=angle(y); % Calculate the argument of y
```

# 1. Generate Typical Signals in MATLAB

Generate a complex exponential signal $y = e^{j\omega n}u[n]$, where $\omega = \frac{2\pi}{48}$

```
figure(1)         % Plot graph: set up graph number
subplot(2,2,1)
stem(n,y_re);
title('Real Part')
set(gca,'XLim',[0,47],'YLim',[-1.5,1.5]);
subplot(2,2,2)
stem(n,y_im);
title('Image Part')
set(gca,'XLim',[0,47],'YLim',[-1.5,1.5]);
subplot(2,2,3)
stem(n,y_mag);
title('Magnitude')
set(gca,'XLim',[0,47],'YLim',[-1.5,1.5]);
subplot(2,2,4)
stem(n,y_arg);
title('Argument')
set(gca,'XLim',[0,47],'YLim',[-pi,pi]);
```

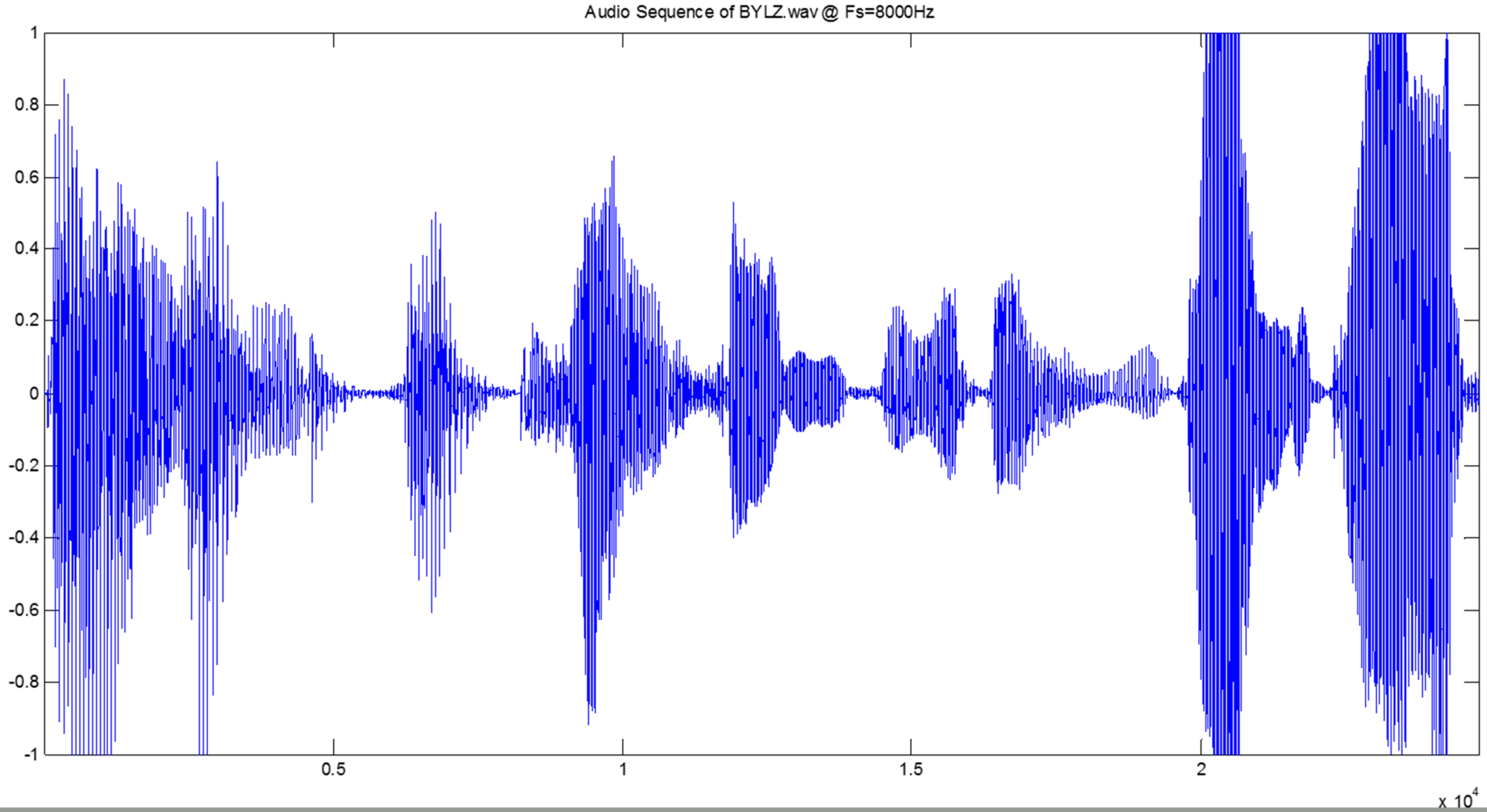


# 2. Input an Audio File to Workspace

# 2. Import an Audio File to MATLAB Workspace

```
FileName='BYLZ.wav';     % Set up audio file name
Speed=1;                 % Set up play speed
[AudioSequence,SampleFrequency]=audioread(FileName);  % Read audio file into workspace
sound(AudioSequence,SampleFrequency*Speed); % Play the audio sequence in different Fs
figure(1) % Plot
plot(1:1:length(AudioSequence),AudioSequence);
title('Audio Sequence of BYLZ.wav');
set(gca,'XLim',[1,length(AudioSequence)],'YLim',[-1,1])
```

# 2. Import an Audio File to MATLAB Workspace



# 3. Analyze Signals by Using DTFT/DFT

# 3. Analyze Signals by Using DTFT/DFT in MATLAB

$$DTFT : X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

In general, the computers are conditioned by processing time and resource, which means they can not calculate the proper DTFT of an infinite signal or can not express this signal in frequency domain continuously.

If we planning to examine a signal in frequency domain by using the DTFT, we should describe this signal as a polynomial function of $\omega$.

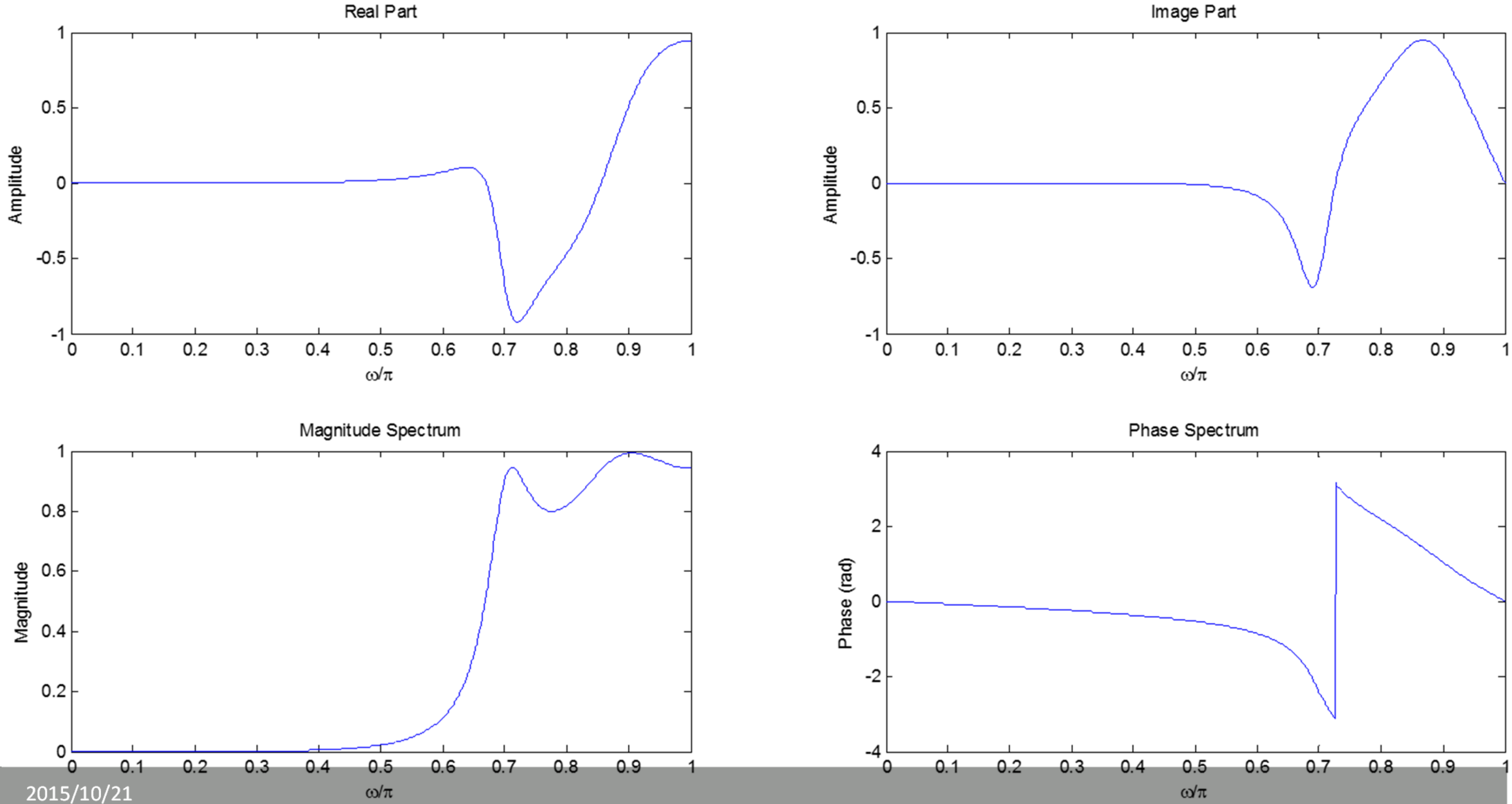
$$X(e^{j\omega}) = \frac{P(e^{j\omega})}{D(e^{j\omega})} = \frac{p_0 + p_1 e^{-j\omega} + \cdots + p_M e^{-j\omega M}}{d_0 + d_1 e^{-j\omega} + \cdots + d_N e^{-j\omega N}}$$

# 3. Analyze Signals by Using DTFT/DFT in MATLAB

$$[X(e^{j\omega_n}), \omega_n] = freqz([p_0, p_1, \ldots, p_M], [d_0, d_1, \ldots, d_N], \boldsymbol{\omega}_n)$$

freqz is a useful MATLAB function, which can compute the complex values of DTFT of a signal sequence, which described as a rational function in $e^{j\omega}$ in the form of a polynomial at a prescribed set of discrete frequency points $\omega = \omega_n$. For reasonably accurate plot, a fairly large number of frequency points should be selected.

# 3. Analyze Signals by Using DTFT/DFT in MATLAB



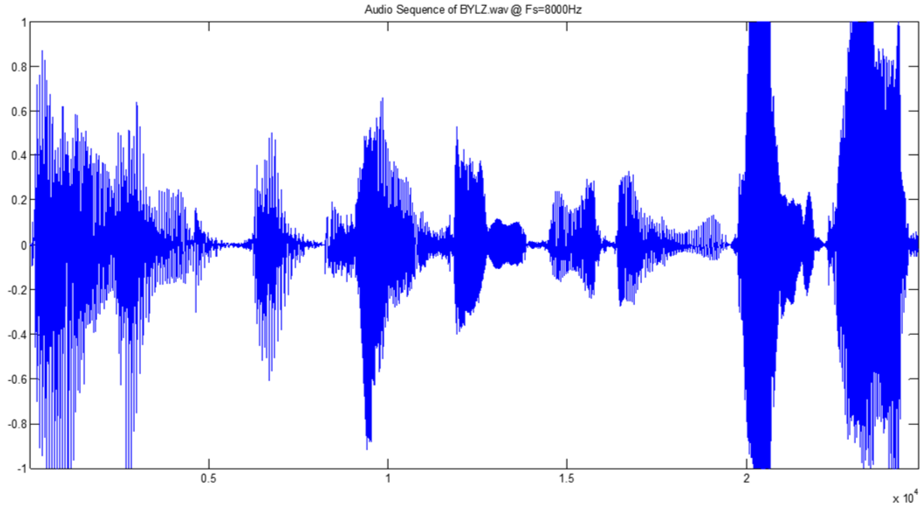
# 3. Analyze Signals by Using DTFT/DFT in MATLAB

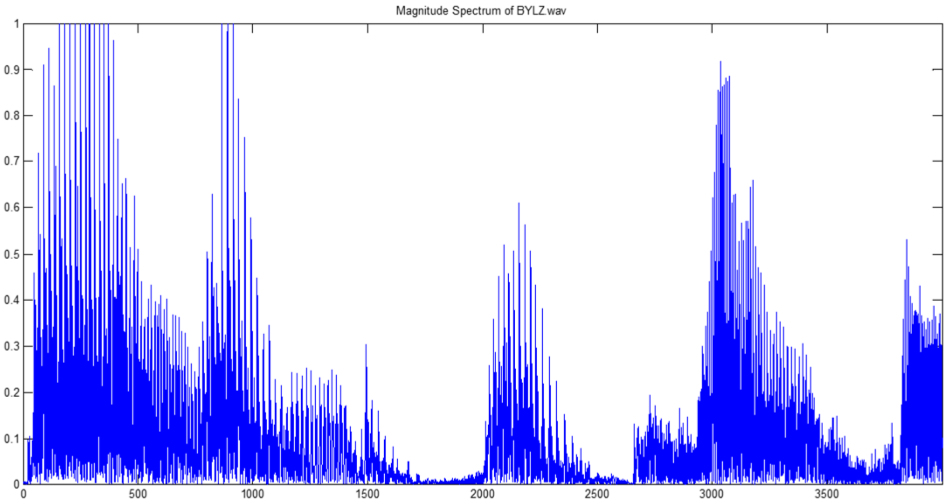
$$DFT: X(k) = X(e^{j\omega})\big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, 0 \le k \le N-1$$

The signals we dealing with in practice are finite discrete time sequences. In addition, they are continuous in frequency domain. Thus we can not obtain a polynomial function of $\omega$ of this signal directly.

However, DFT provide an approach to compute the discrete frequency spectrum based on a N-length sequence in time domain.

# 3. Analyze Signals by Using DTFT/DFT in MATLAB




```
clear;
FileName='BYLZ.wav';     % Set up audio file name
Speed=1;                 % Set up play speed
[AudioSequence,SampleFrequency]=audioread(FileName);  % Read audio file into workspace
AudioSequence_DFT=fft(AudioSequence); % DFT

figure(1) % Plot
plot(0:SampleFrequency/length(AudioSequence):SampleFrequency/2-SampleFrequency/length(AudioSequence),abs(AudioSequence(1:end/2)));
title('Magnitude Spectrum of BYLZ.wav');
set(gca,'XLim',[0,SampleFrequency/2-SampleFrequency/length(AudioSequence)],'YLim',[0,1])
```

# 4. Experiment Tasks and Goals

# Experiment Tasks and Goals

| | |
|---|---|
| **Generate Typical Signals** | 1. Unit sampling sequence;<br>2. Unit step sequence;<br>3. Real exponential sequence;<br>4. Complex exponential sequence;<br>5. Sine sequence; |
| **Frequency Analyze the Signal from above** | 1.DTFT: $X(e^{j\omega}) = \frac{1+0.3544e^{-j\omega}+0.3508e^{-j2\omega}+0.1736e^{-j3\omega}+0.2401e^{-j4\omega}}{1-1.3817e^{-j\omega}+1.5632e^{-j2\omega}-0.8843e^{-j3\omega}+0.4096e^{-j4\omega}}$;<br>2.DFT: Signal 1~5 in above; |
| **Import an Audio File and Analyze It by using Techniques Learn From This Lesson.** | 1. Record both male and female saying “Hello World” speech signals and import them to MATLAB;<br>2. Using DFT method to analyze these two signals and find the difference. |
| **Further Goals for Advanced Learners** | 1. Try to plot the complex exponential sequence in 3-D(Real Axis, Imaginary Axis and Time Index Axis) with different $\omega$;<br>2. Try to realize the DFT all by yourself instead of using fft().<br>3. Compare the performance of fft() and the DFT realized by yourself by using tic\toc; |

# Thank You!

Present By: Cui Ao & Li Shangwen