MATLAB Functionality for Digital Speech Processing

- MATLAB Speech Processing Code
- MATLAB GUI Implementations

Graphical User Interface

GUI Lite 2.5

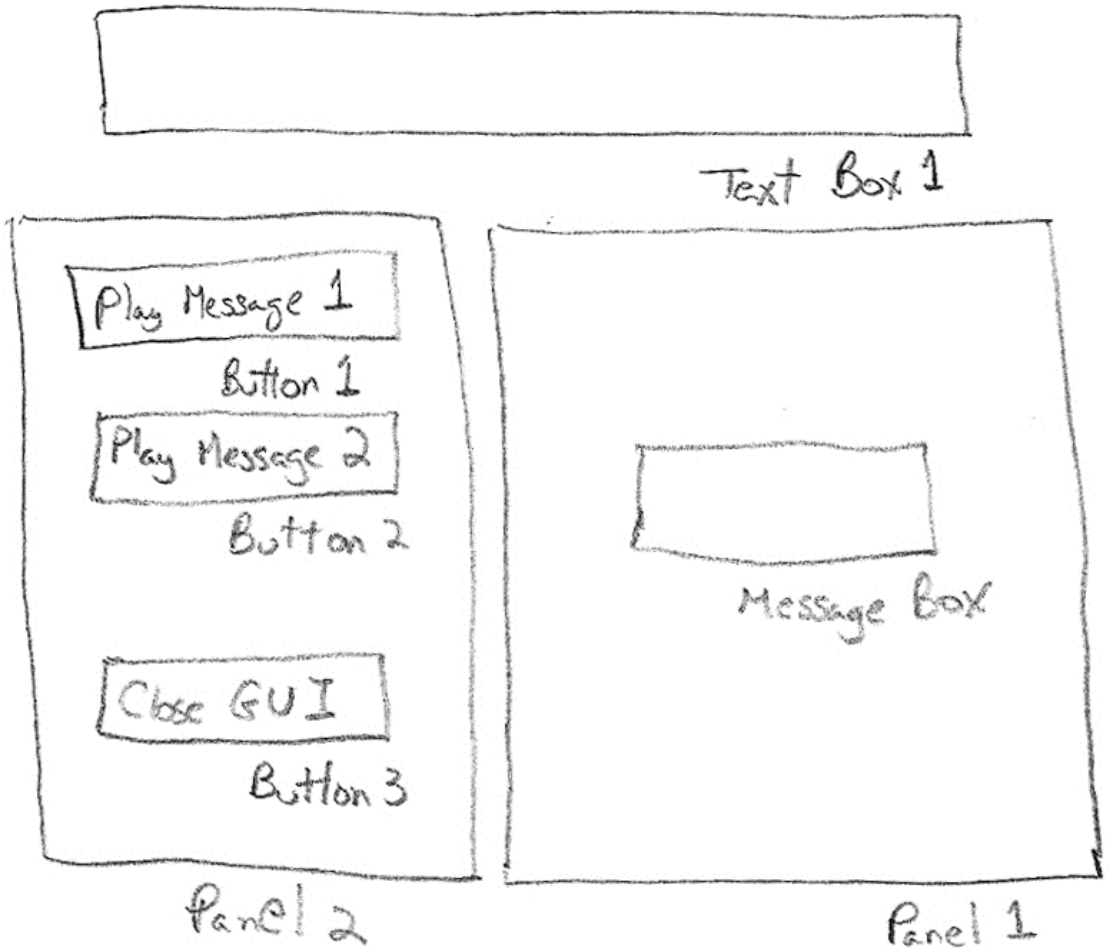
Graphical User Interface Components

- GUI Lite created by students at Rutgers University to simplify the process of creating viable GUIs for a wide range of speech and image processing exercises
- GUI Lite Elements
  - basic design tool and editor (GUI Lite 2.5)
  - panels; used to block group of buttons/graphical panels/etc., into one or more coherent blocks
  - graphics panels; used to display one or more graphical outputs (figures)
  - text block; used to display global information about the specific speech processing exercise
  - buttons; used to get and set (vary) exercise parameters; used to display a list of exercise options; used to initiate actions within the code
    - editable buttons – get and/or set parameter value
    - text buttons – display variable values
    - slider buttons – display variable range
    - popupmenu buttons – display list of variable options (e.g., list of speech files)
    - pushbuttons – initiate actions within the code

GUI LITE 2.5 Design Process

- begin with a rough sketch of the GUI 2.5 output, segmented into button panels, graphics panels, text boxes, and buttons
- run program 'runGUI.m' to create GUI elements and save as a GUI file
- edit the two programs created by GUI LITE 2.5
  - rename GUI program from 'EditrunGUI.m' to 'exercise_GUI25.m'
  - rename GUI Callbacks program from 'PanelandButtonCallbacks.m' to 'Callbacks_exercise_GUI25.m'
- run the resulting exercise and loop on GUI design and Callbacks implementation

Hello/Goodbye World Plan



Design Specs:

- 2 Panels (for linking inputs and outputs)
- 1 Text Box (for describing the Exercise GUI)
- 3 Buttons (all pushbuttons) (for embedding Callback code to play two messages and to close up the GUI)

GUI25 Initial Screen

GUI25 Creation for ‘hello_goodbye_world’

- run program ‘runGUI.m’ and click on ‘New’ button
- enter values for number of panels (2), number of graphics panels (0), number of text boxes (1), and number of buttons (3)
- enter name for GUI (‘hello_goodbye_world.mat’)
- create the GUI objects specified above, using mouse cursor to define range of each object; set GUI object properties
- save the resulting specifications for the GUI in the designated .mat file
- edit and rename the GUI exercise from ‘EditrunGUI.m to ‘hello_goodbye_world_GUI25.m’
- edit and rename the GUI Callbacks from ‘PanelandButtonCallbacks.m’ to ‘Callbacks_hello_goodbye_world_GUI25.m’

GUI25 Callback Code

```
% Callback for button 1 – present on screen message 1
function button1Callback(h,eventdata);
    uiwait(msgbox('Hello World!',  'Message1',  'modal'));

% title box
    stitle1=strcat('Hello World Using GUI2.5');
    set(titleBox1,  'String',  stitle1);
    set(titleBox1,  'FontSize',  25);
end

% Callback for button 3 – Close GUI
function button3Callback(h,eventdata);
    display Goodbye;
    close(gcf);
end
```

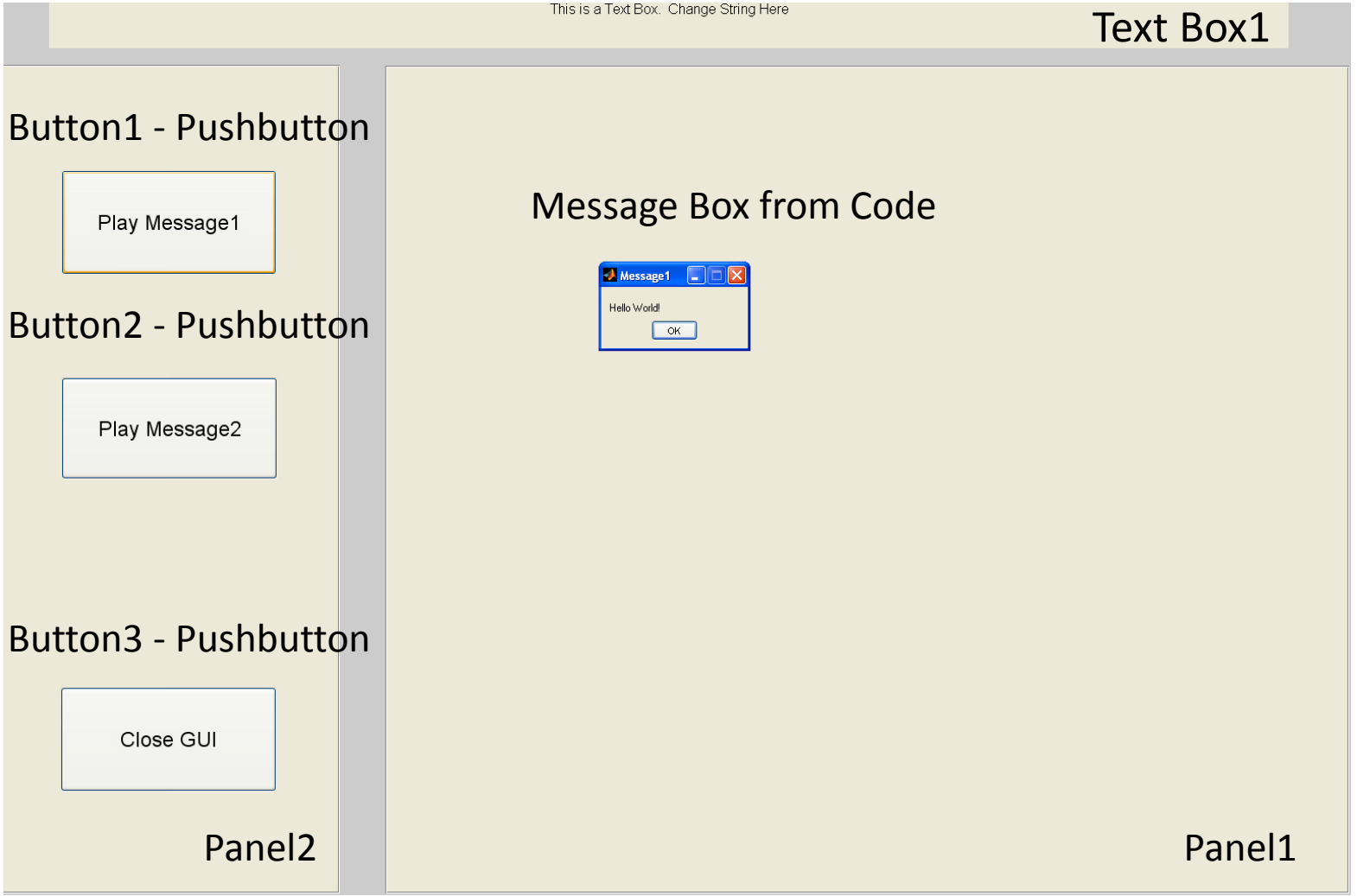
run:

hello_goodbye_world_GUI25.m

directory:

hello_goodbye_world_gui25

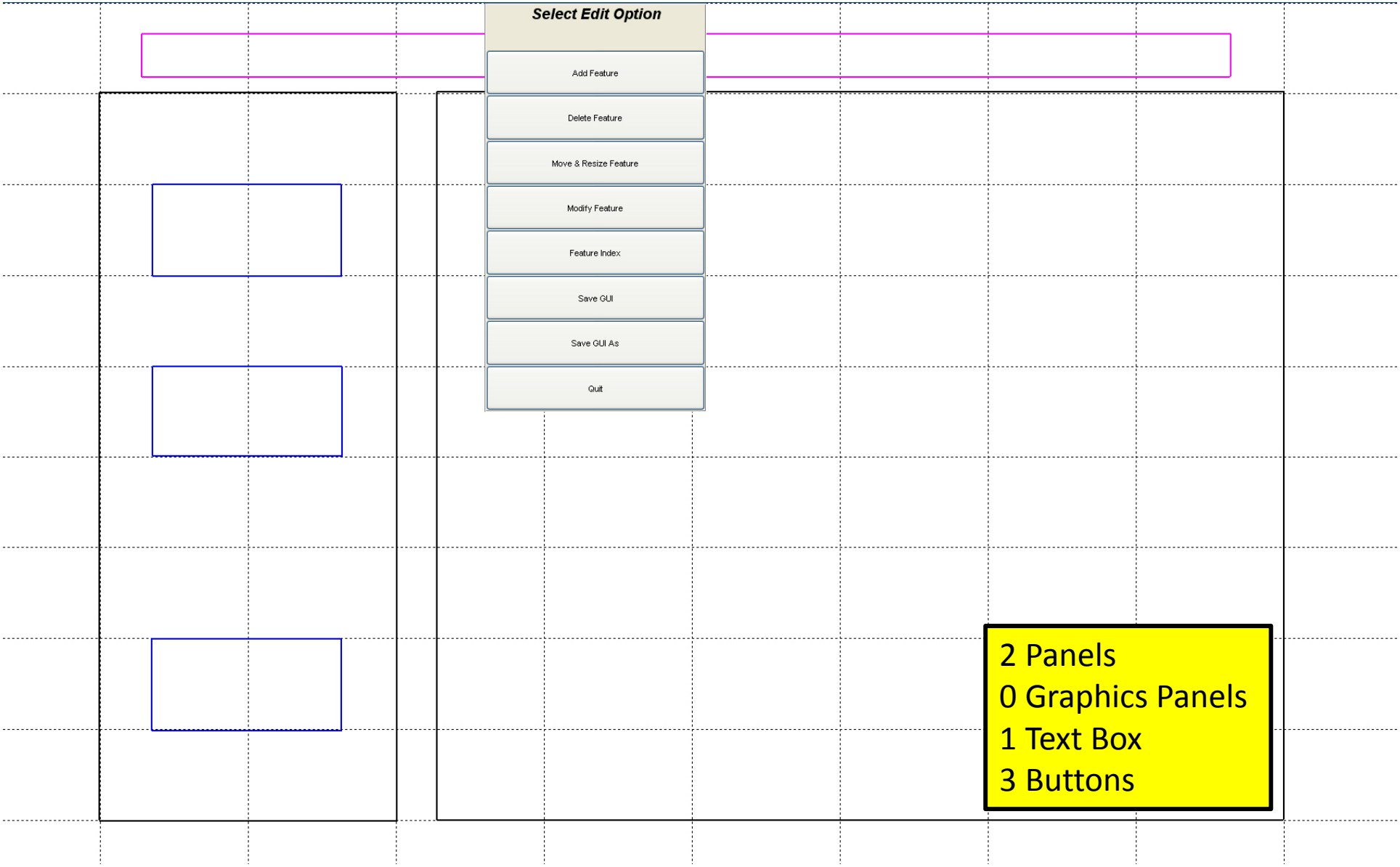
Hello/Goodbye World



Hello/Goodbye World GUI

- Run program ‘runGUI.m’ to bring up GUI Lite 2.5 editor
- Choose Mod (modify) and select GUI file ‘hello_goodbye_world.mat’ for editing
- Choose ‘Move & Resize Feature’ option
- Choose ‘Button’ option
- Left click inside button to be modified
- Choose new button coordinates by using graphics cursor to identify lower left and upper right corners of modified button
- Click ‘Save GUI’ button
- Iterate on other buttons
- Click ‘Quit’ option to terminate GUI Lite 2.5 editor

GUI Lite 25 Edit Screen



GUI LITE 2.5 Edit Screen

Add Feature

Delete Feature

Move & Resize Feature

Modify Feature

Feature Index

Save GUI

Save GUI As

Quit

GUI Lite 25 Features

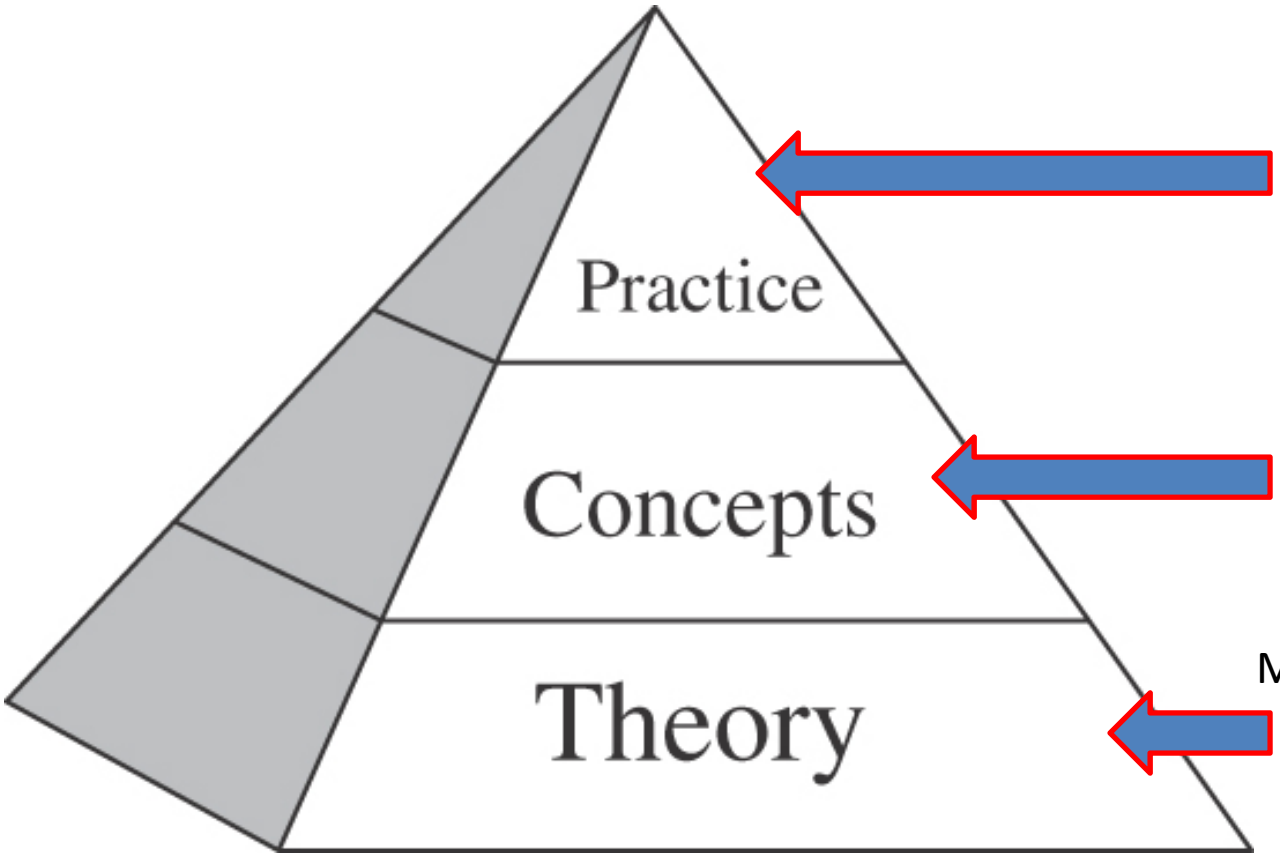
- separates GUI design from Callbacks for each GUI element
- provides a versatile editor for modifying GUI elements without impacting the Callback actions
- provides a GUI element indexing feature that enables the user to identify GUI elements with the appropriate Callback elements

Missing GUIDE Features

- radio button
- check box
- listbox
- toggle button
- table
- axes
- button group
- active X control

- are the missing features of value?
- do we need these features?
- can we create the desired set of speech processing exercises without these features?
- can we add these features to the GUI LITE editor?

Digital Speech Processing



Ability to implement theory and concepts in working code (MATLAB, C, C++); algorithms, applications

Basic understanding of how theory is applied; autocorrelation, waveform coding, …

Mathematics, derivations, signal processing; e.g., STFT, cepstrum, LPC, …

Need to understand speech processing at all three levels

The Speech Stack

Speech Applications **— coding, synthesis, recognition, understanding, verification, language translation, speed-up/slow-down**
Speech Algorithms **— speech-silence (background), voiced-unvoiced, pitch detection, formant estimation**
Speech Representations **— temporal, spectral, homomorphic, Linear Prediction Coding**
Fundamentals **— acoustics, linguistics, pragmatics, speech production/perception**
Basics – read/write speech/audio files; display speech files; play files

MATLAB Exercise Categories

- Basic MATLAB Functions for handling speech and audio files
- Advanced MATLAB Functions for Speech Processing

MATLAB Exercise Categories

- The speech processing exercises are grouped into 5 areas, namely:
  - **Basics** of speech processing using MATLAB (5)
  - **Fundamentals** of speech processing (6)
  - **Representations** of speech in time, frequency, cepstrum and linear prediction domains (22)
  - **Algorithms** for speech processing (7)
  - **Applications** of speech processing (17)

Basic Functionality

- **read a speech file** (i.e., open a .wav speech file and read the speech sample into a MATLAB array)
- **write a speech file** (i.e., write a MATLAB array of speech samples into a .wav speech file)
- **play a MATLAB array** of speech samples as an audio file
- * **play a sequence of MATLAB arrays of speech samples** as a sequence of audio files
- **record a speech file** into a MATLAB array
- **plot a speech file** (MATLAB array) as a waveform using a strips plot format
- * **plot a speech file** (MATLAB array) as one or more 4-line plot(s)
- **convert the sampling rate** associated with a speech file (MATLAB array) to a different (lower/higher) sampling rate
- **lowpass/highpass/bandpass filter** a speech file (MATLAB array) to eliminate DC offset, hum and low/high frequency noise
- **plot a frame of speech** and its associated spectral log magnitude
- **plot a spectrogram** of a speech file (MATLAB array)
- * **plot multiple spectrograms** of one or more speech files (MATLAB arrays)

* indicates exercise not yet done

Read a Speech File into a MATLAB Array

- [xin, fs, nbits] = wavread(filename);
- [xin, fs] = loadwav(filename);
  - filename is ascii text for a .wav-encoded file which contains a speech signal encoded using a 16-bit integer format
  - xin is the MATLAB array in which the speech samples are stored (in double precision format)
  - fs is the sampling rate of the input speech signal
  - nbits is the number of bits in which each speech sample is encoded (16 in most cases)
  - program wavread scales the speech array, xin, to range $-1 \leq xin \leq 1$, whereas loadwav preserves sample values of the speech file and hence array xin is scaled to range $-32768 \leq xin \leq 32767$
- [xin1, fs, nbits] = wavread('s5.wav');
- [xin2, fs] = loadwav('s5.wav');

Read a Speech File into a MATLAB Array

```
% test_wavread.m
% test waveread function
%
% read speech samples from file 'test_16k.wav' into array x1 using wavread
% routine
   filein='test_16k.wav';
   [x1,fs1,nbits]=wavread(filein);

% print out values of fs1, nbits, wavmin1, wavmax1
   wavmin1=min(x1);
   wavmax1=max(x1);
   fprintf('file: %s, wavmin/wavmax: %6.2f %6.2f, fs1: %d, nbits: %d \n',...
      filein,wavmin1,wavmax1,fs1,nbits);

% read speech samples from same file into array x2 using loadwav routine
   [x2,fs2]=loadwav(filein);

% print out values of fs2, nbits, wavmin2, wavmax2
   wavmin2=min(x2);
   wavmax2=max(x2);
   fprintf('file: %s, wavmin/wavmax: %d %d, fs2: %d \n',...
      filein,wavmin2,wavmax2,fs2);
```

Terminal Display:

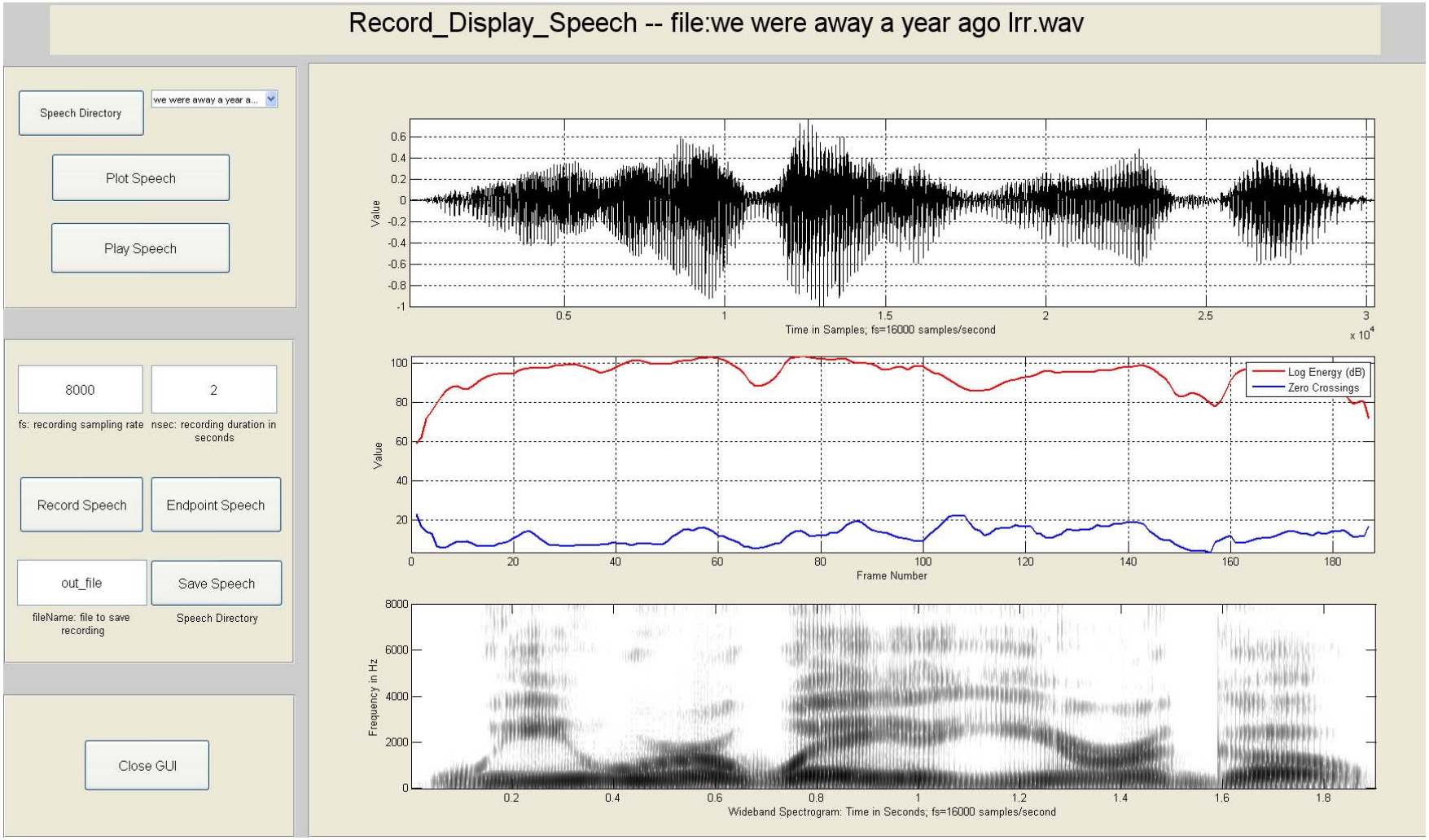
```
file: test_16k.wav, wavmin/wavmax:  -1.00   1.00, fs1: 16000, nbits: 16
file: test_16k.wav, wavmin/wavmax: -32768 32767, fs2: 16000
```

Write a Speech Array into a Speech File

- wavwrite(xout, fs, nbits, filename);
- savewav(xout, filename, fs);
  - xout is the MATLAB array in which the speech samples are stored
  - fs is the sampling rate of the output speech signal
  - nbits is the number of bits in which each speech sample is encoded
  - filename is the ascii text for the .wav-encoded file in which the MATLAB signal array is to be stored
  - for wavwrite the MATLAB array xout needs to be scaled to the range $-1 \le xin \le 1$ whereas for savewav the MATLAB array xout needs to be scaled to the range $-32768 \le xout \le 32767$
- wavwrite(xin1, fs, 's5out.1.wav');
- savewav(xin2, 's5out.2.wav', fs);

Record/Display Speech

Basics



Play a Speech File

- sound(x, fs);
- soundsc(x, fs);

  - for sound the speech array, x, must be scaled to the range $-1 \le x \le 1$
  - for soundsc any scaling of the speech array can be used
  - fs is the sampling rate f the speech signal

- [xin, fs] = loadwav('s5.wav'); % load speech from s5.wav;
- xinn = xin/abs(max(xin)); % normalize to range of − 1 to 1;

- sound(xinn, fs); % play out normalized speech file;
- soundsc(xin, fs); % play out unnormalized speech file;

* Play Multiple Speech Files

- play_multiple_files.m;
  - sequence of filenames read in via filelist, keyboard or file search

- Example of usage to play out 3 speech files in sequence:
  - kbe=filename entry via filelist(2), keyboard(1), or file search(0):1; % keyboard chosen
  - N=number of files to be played in a group:3; % play out 3 files
  - i=1; filename: s1.wav;
  - i=2; filename: s2.wav;
  - i=3; filename: s3.wav

* Play Multiple Speech Files

- test_play_files.m
  - play the following sequence of files:

s2.wav
s3.wav
s4.wav
s5.wav
s6.wav

Record Speech into MATLAB Array

- record_speech.m (calls MATLAB function audiorecorder.m, formally wavrecord.m)

- function y=record_speech(fs, nsec);
  - fs: sampling frequency
  - nsec: number of seconds of recording
  - y: speech samples array normalized to peak of 32767

Display Speech Waveform

Strips Plot

*** 4-Line Plots**

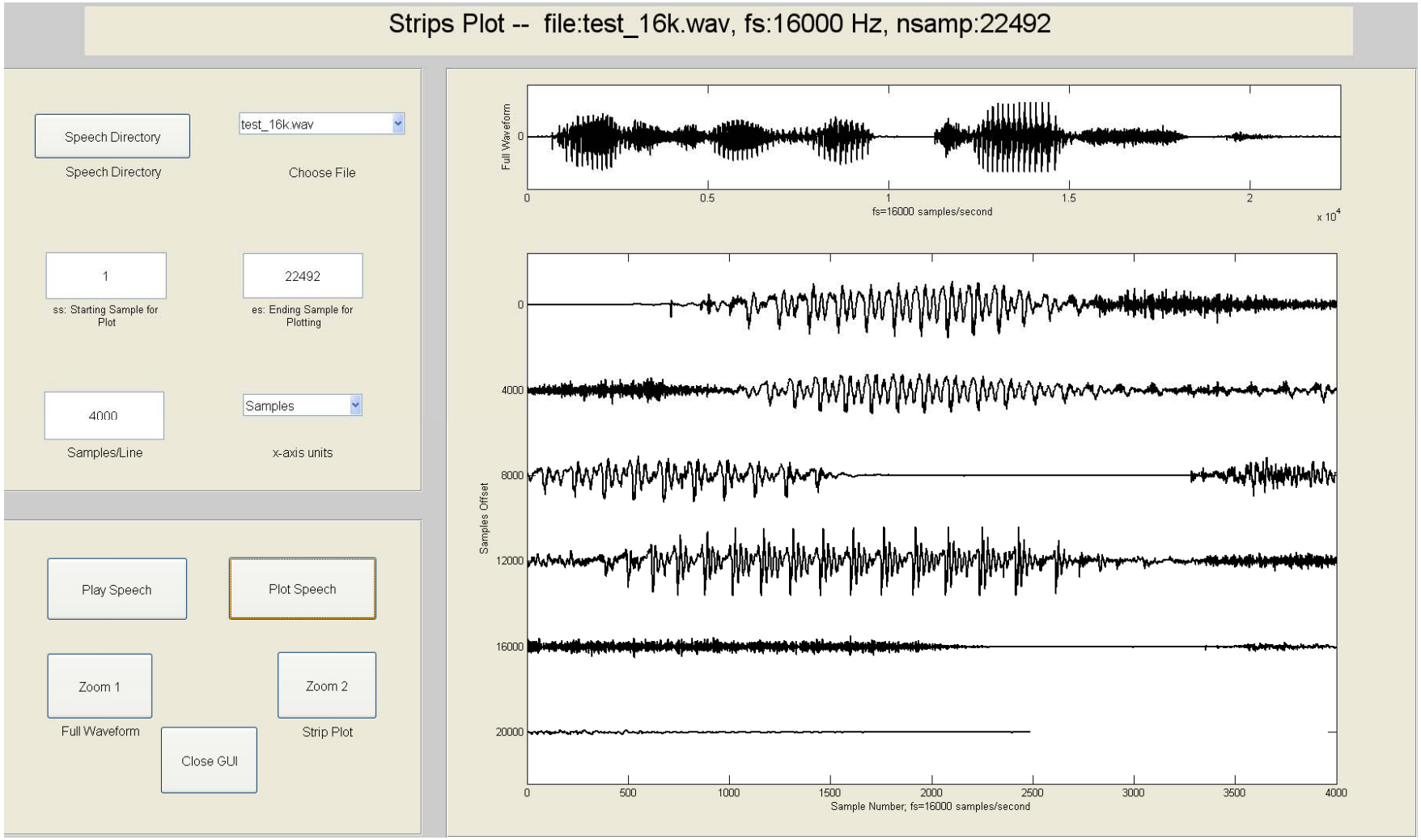
Basics

Waveform Zoom Strips Plot

- Plotting and examining speech/audio waveforms is one of the most useful ways of understanding the properties of speech and audio signals.
- This MATLAB Exercise displays a speech/audio waveform as a single running plot of samples (called a Strips Plot).
- Exercise plots from designated starting sample to designated ending sample, with a user-specified number of samples/line.
- Zoom feature to select region of signal for display.
- Plots use either samples or seconds, as specified by the user.

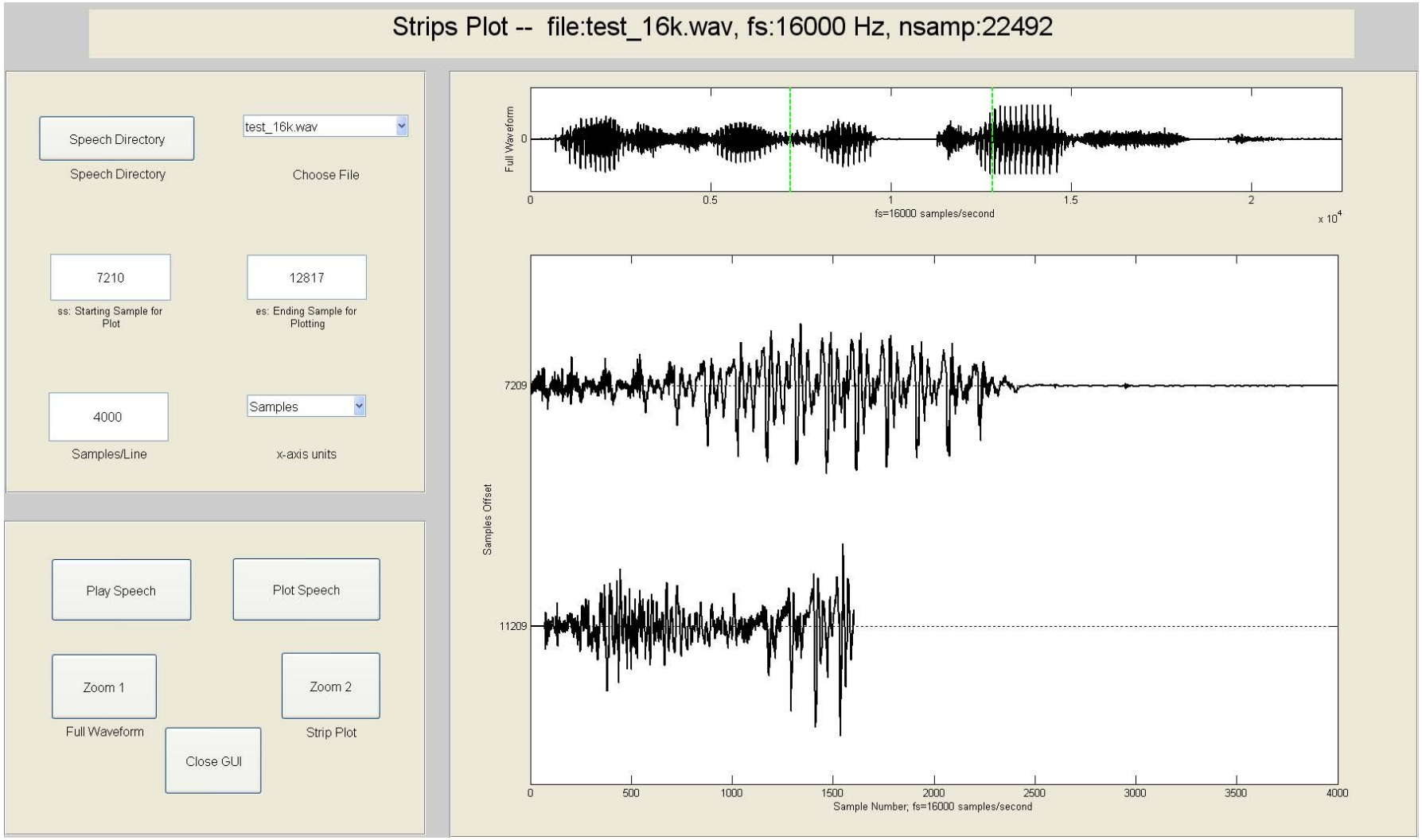
Basics

Waveform Strips Plot



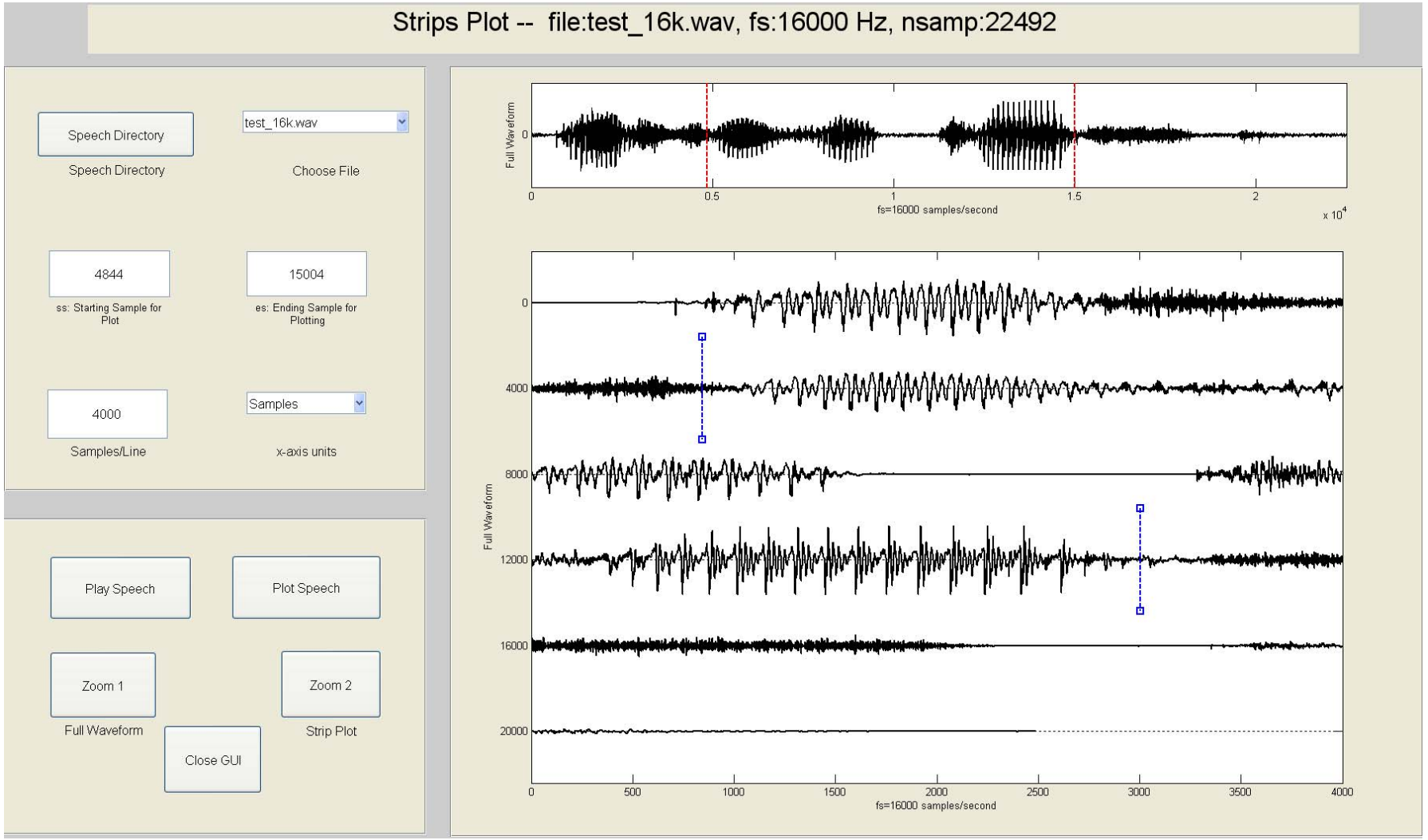
Basics

Waveform Strips Plot – Zoom 1

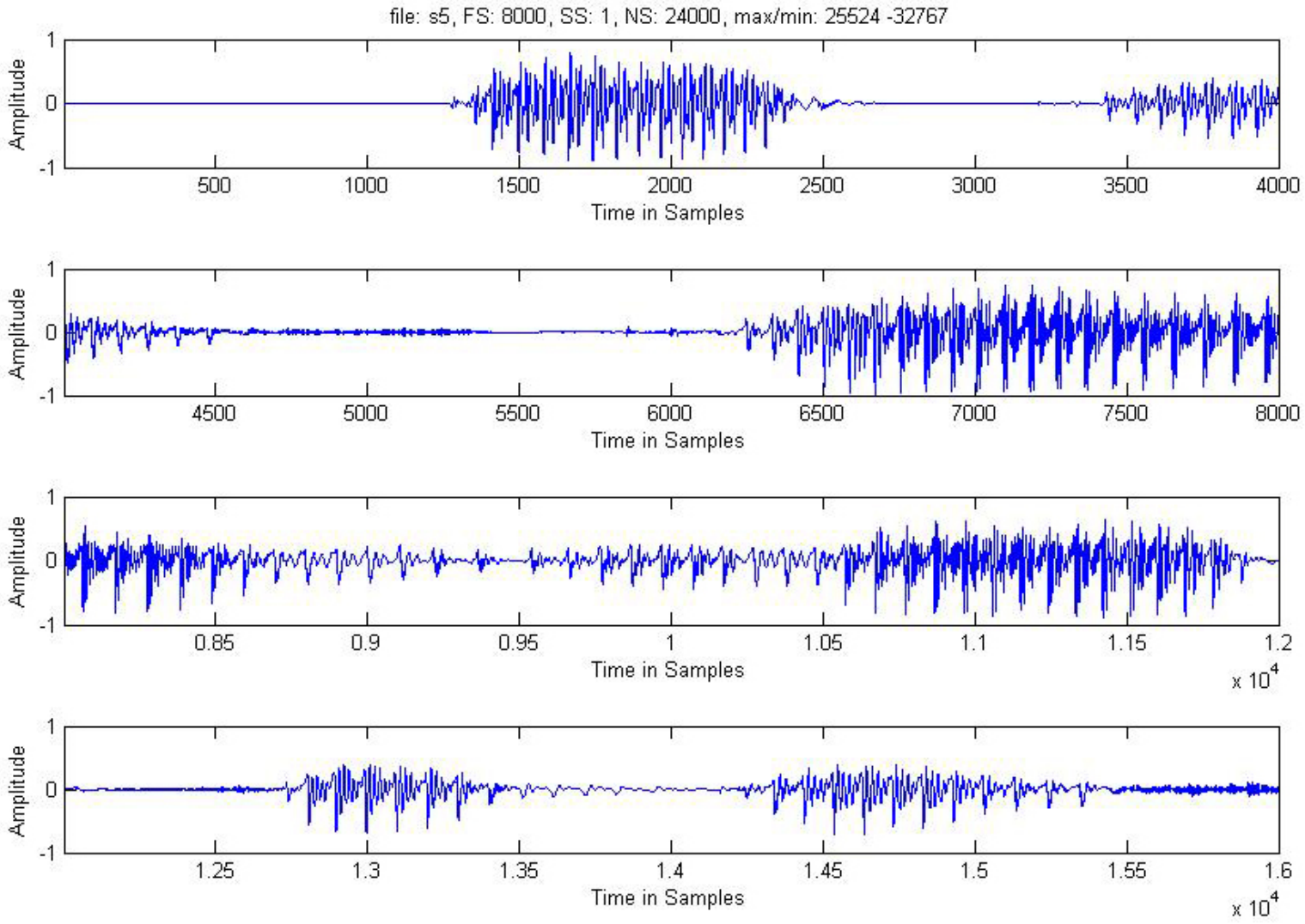


Basics

Waveform Strips Plot – Zoom 2



* Plot Speech Using 4-Line Plot



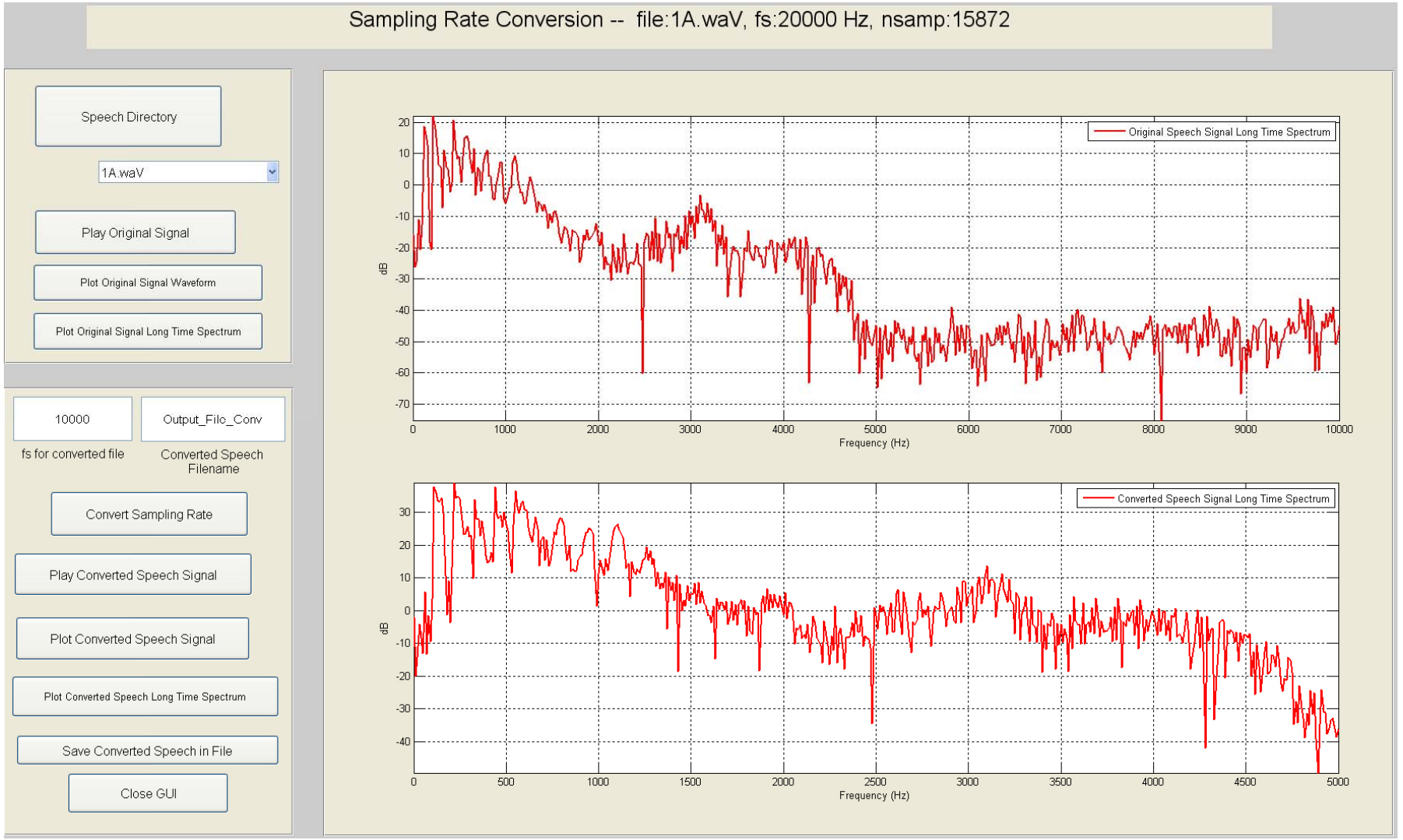
Sampling Rate Conversion

- y = srconv(x, fsin, fsout);
  - x: input speech array;
  - fsin: input speech sampling rate;
  - fsout: desired speech sampling rate;

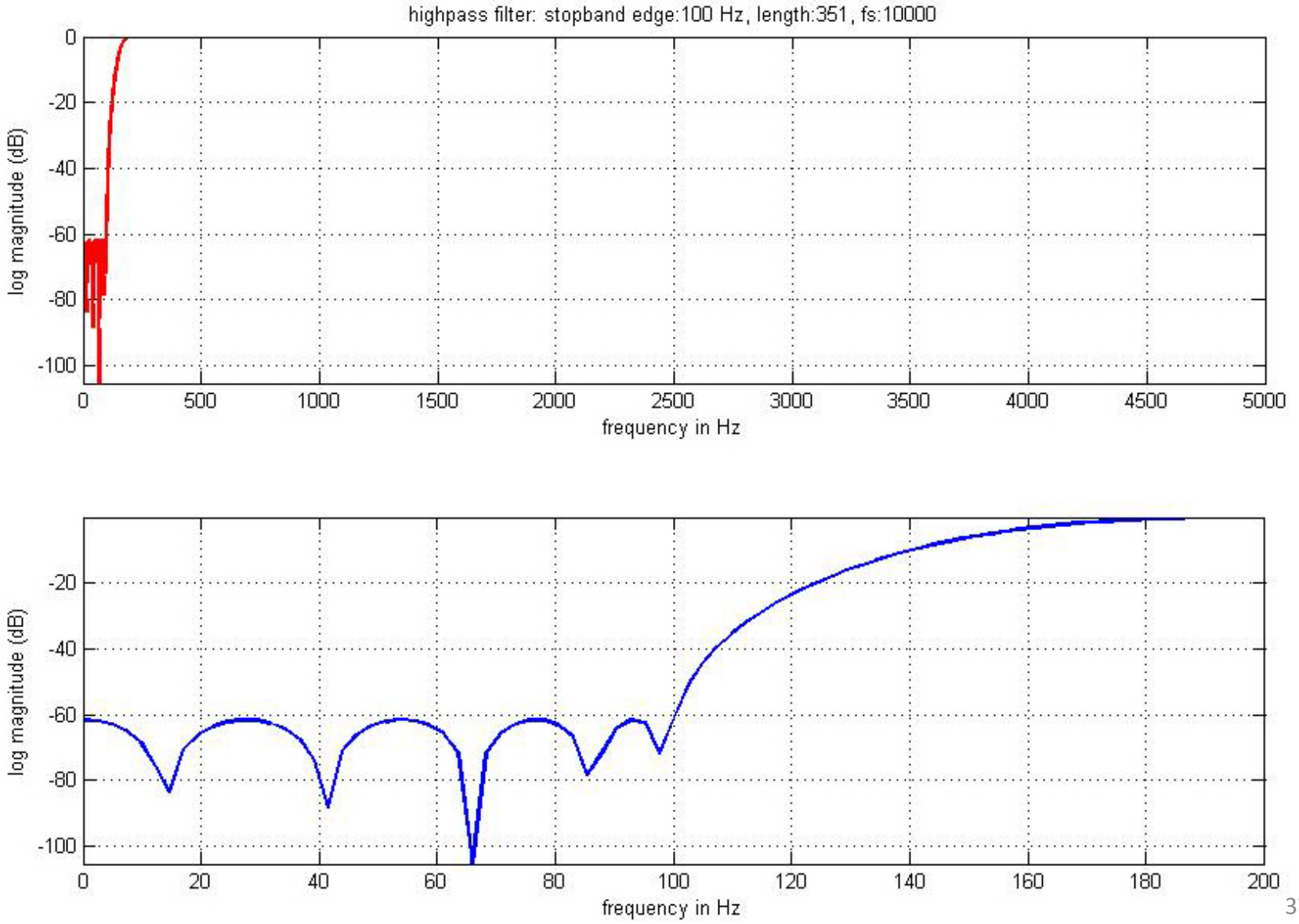
- Example:
  - [xin, fsin] = loadwav('s5.wav'); % fsin=8000;
  - fsout = 10000; % desired sampling rate;
  - y = srconv(xin, fsin, fsout);

Basics

Sampling Rate Conversion

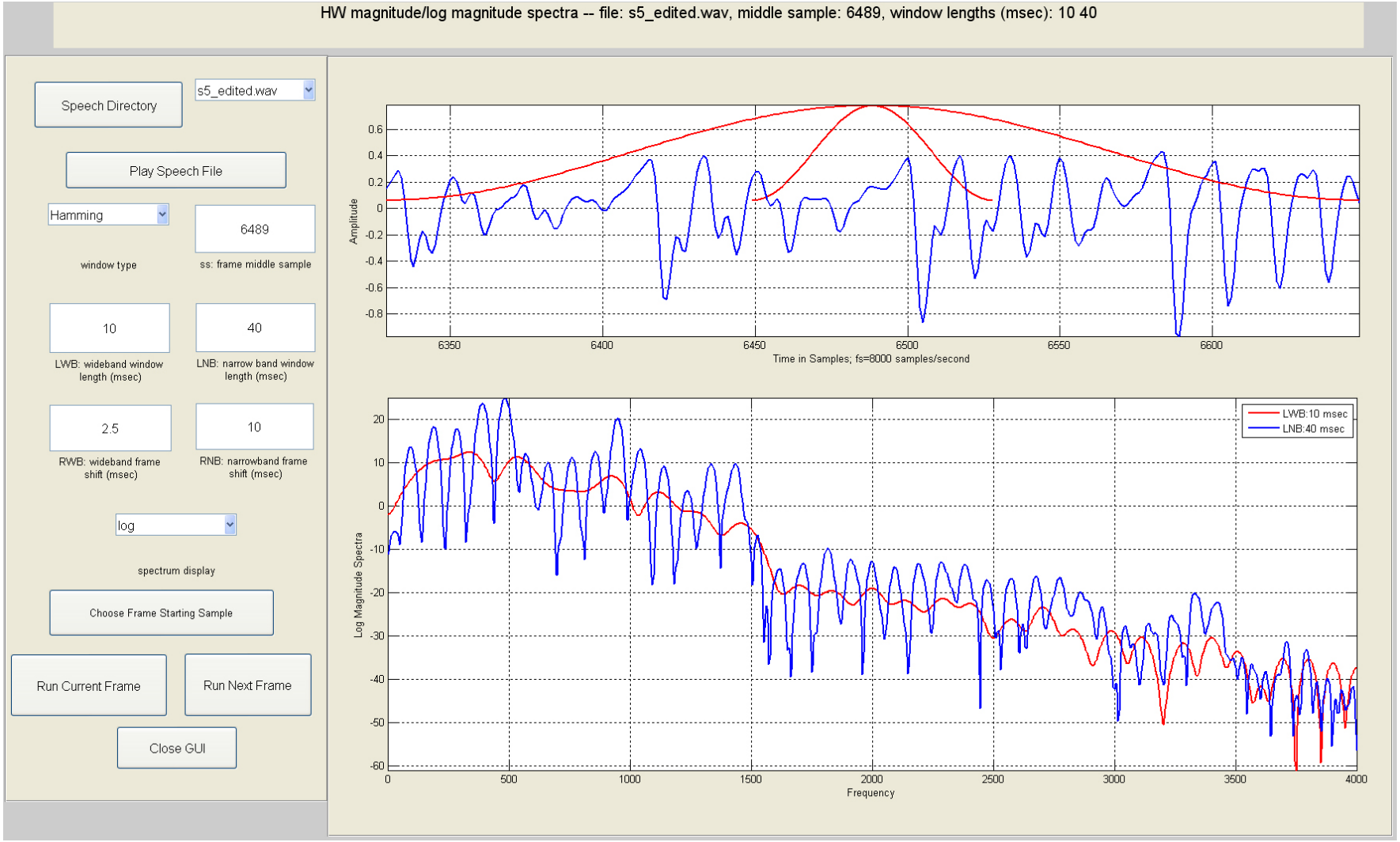


Filter Speech Waveform



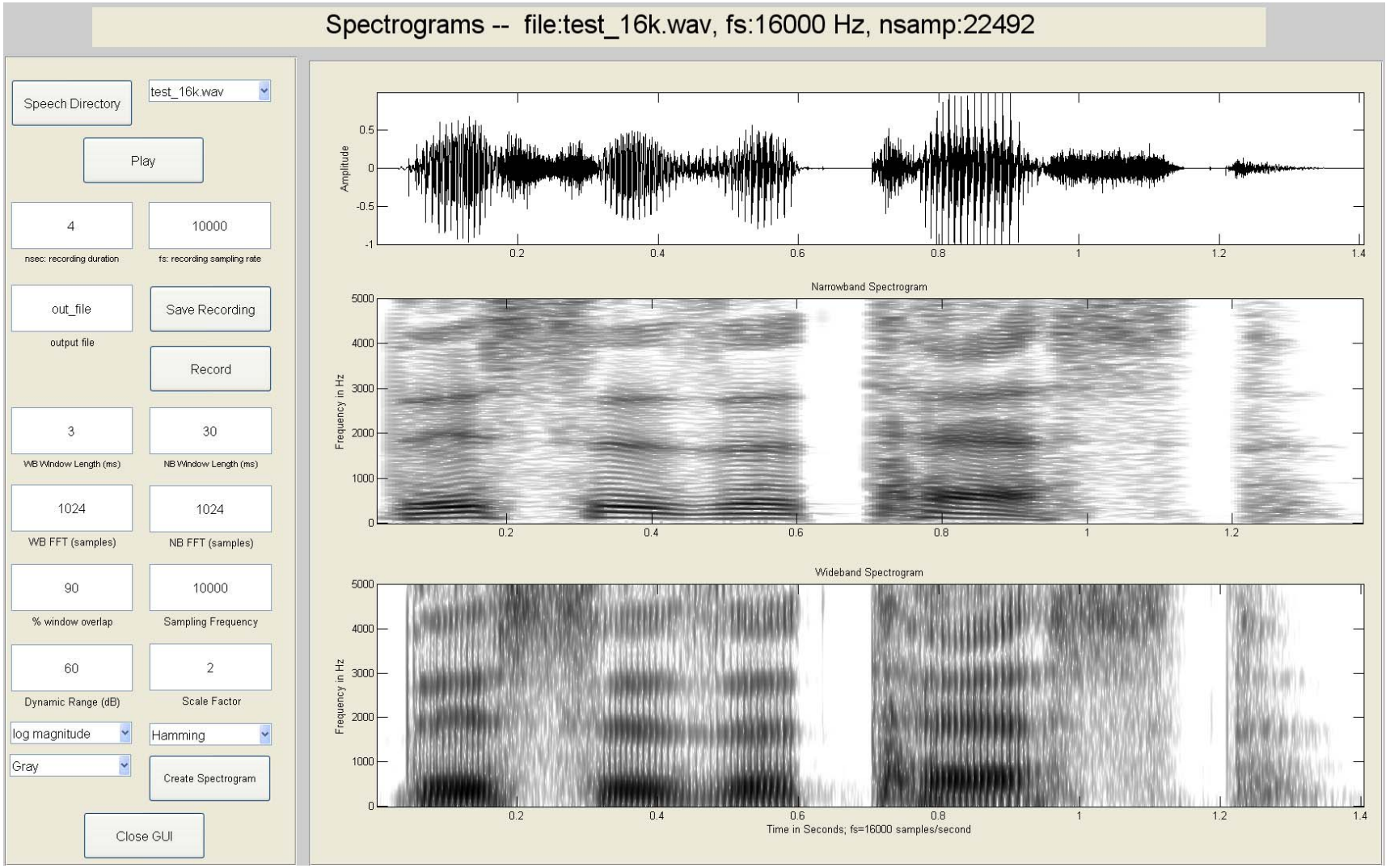
Basics

Frame-Based Spectrums

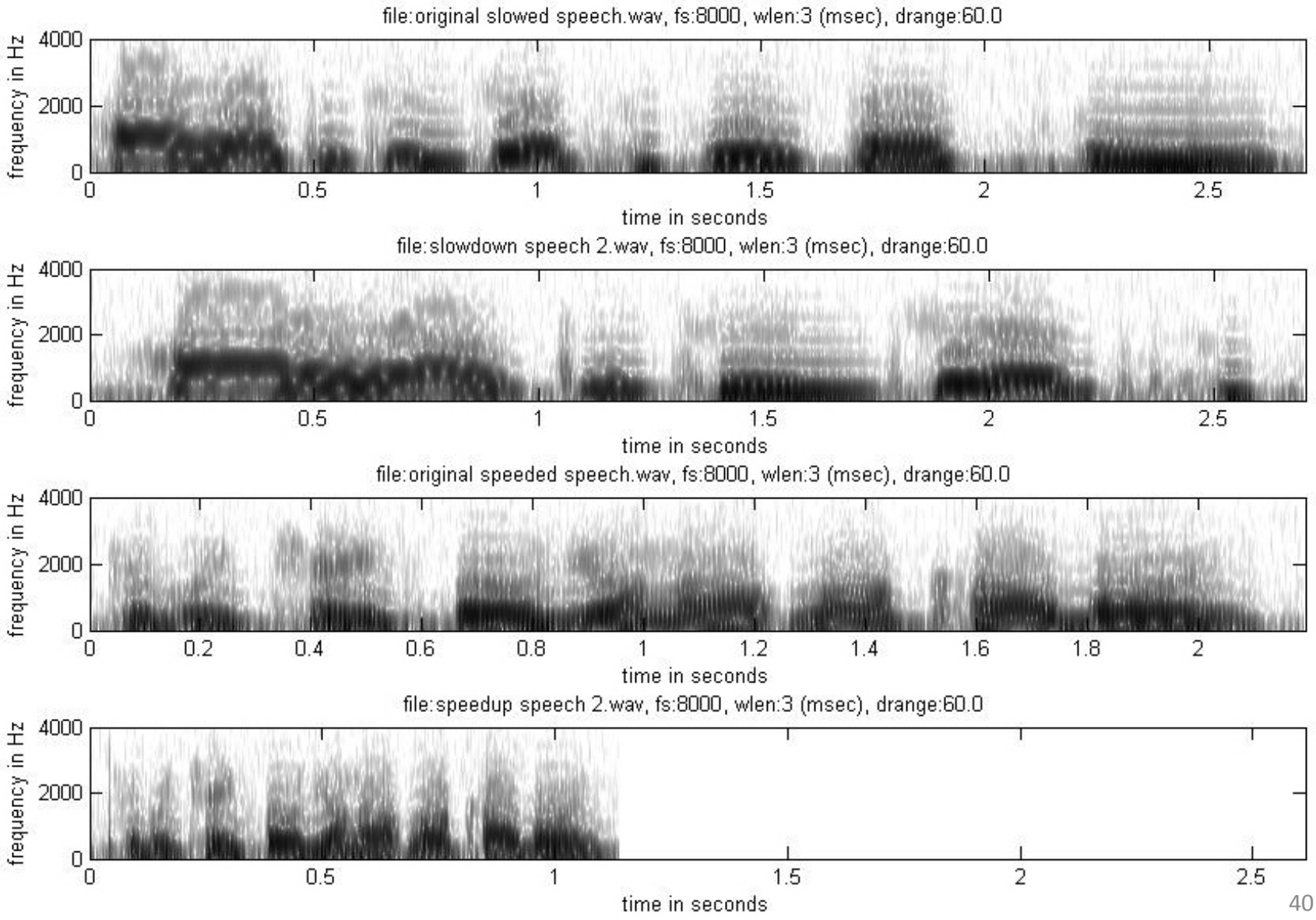


Wideband/Narrowband Spectrogram

Basics



* Plot Multiple Spectrograms



Fundamentals

- 2-tube vocal tract model
- 3-tube vocal tract model
- p-tube vocal tract model
- glottal pulse model and spectrum
- composite vocal tract model and spectrum
- ideal vocal tract model and spectrum

Representations

- **time domain exercises**
  - windows; features; autocorrelation estimates; amdf
- **frequency domain exercises**
  - phase/magnitude; overlap-add windows; WSOLA
- **cepstral domain exercises**
  - analytical cepstrum; single pole cepstrum; FIR sequence cepstrums; cepstrum aliasing; cepstrum liftering; cepstral waterfall
- **linear prediction exercises**
  - LPC frames; LPC error; LPC varying p; LPC varying L; LSP roots; plot roots

Algorithms

- endpoint detector
- Voiced-Unvoiced-Background estimation method
- autocorrelation pitch detector
- log harmonic spectral waterfall plots
- cepstral pitch detector
- SIFT pitch detector
- formant estimation method

Applications – Part 1

- Speech waveform coding;
  - statistical properties of speech; quantization characteristics of a B-bit uniform or mu-law compressed and quantized speech file; uniform quantization; mu-law compression; mu-law quantization; Signal-to-Noise Ratio (SNR) of uniform and mu-law quantizers
- Automatic Gain Control (AGC)
- Adaptive Differential Pulse Code Modulation (ADPCM) waveform speech coder
- Vector Quantizer (VQ); VQ Cells
- Synthetic vowel synthesizer

Applications – Part 2

- LPC error synthesis
- LPC vocoder
- Play pitch period contour
- Two-Band subband coder
- Phase Vocoder
- Isolated, speaker-trained, digit recognizer

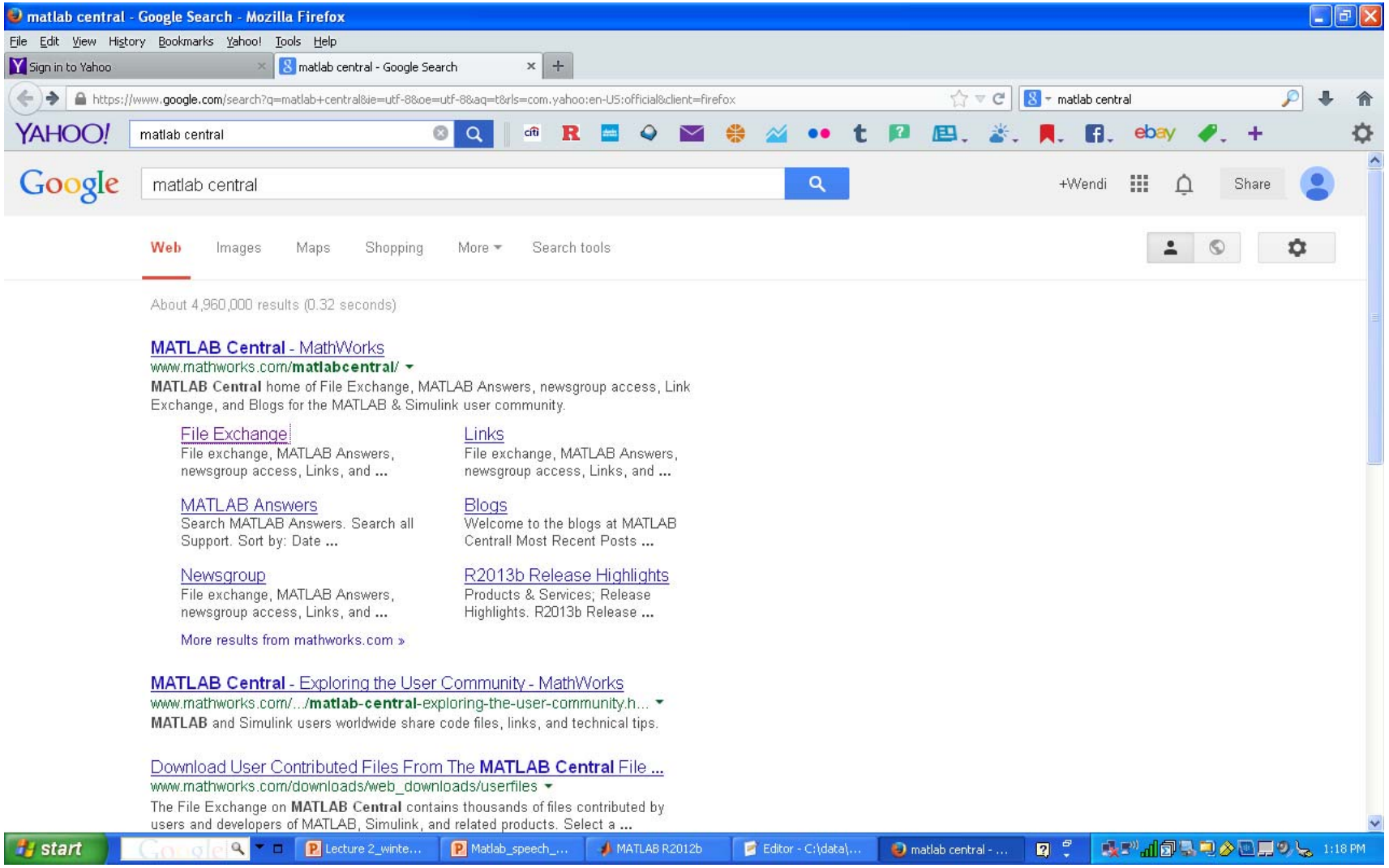
Summary

- Set of about 60 MATLAB speech processing exercises
- Exercises aligned with distinct sections in the textbook TADSP by Rabiner/Schafer
- Each exercise has an associated Graphical User Interface created using a GUI LITE program and created expressly for these speech processing exercises
- GUI LITE design and implementation Callbacks are in totally separate code packages

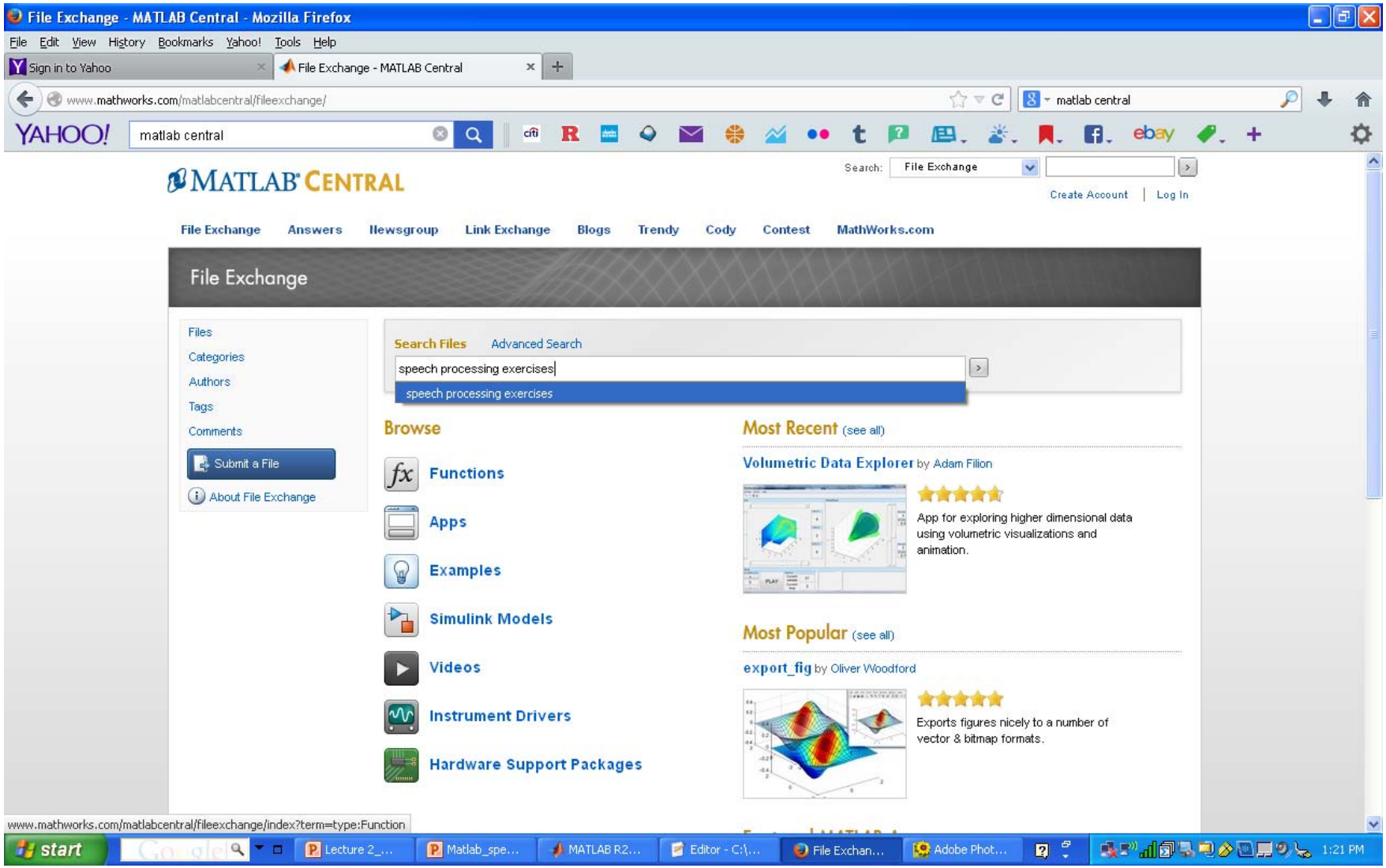
MATLAB Central APPs

- **Search**: Matlab Central
- **Click on:** 'File Exchange'
- **Local Search:** 'speech processing exercises'
- **Click on desired exercise** (be sure to download speech/audio files before downloading any exercise: e.g., Zoom Strips Plot)
- **Click on** downloaded exercise to get to user guide information

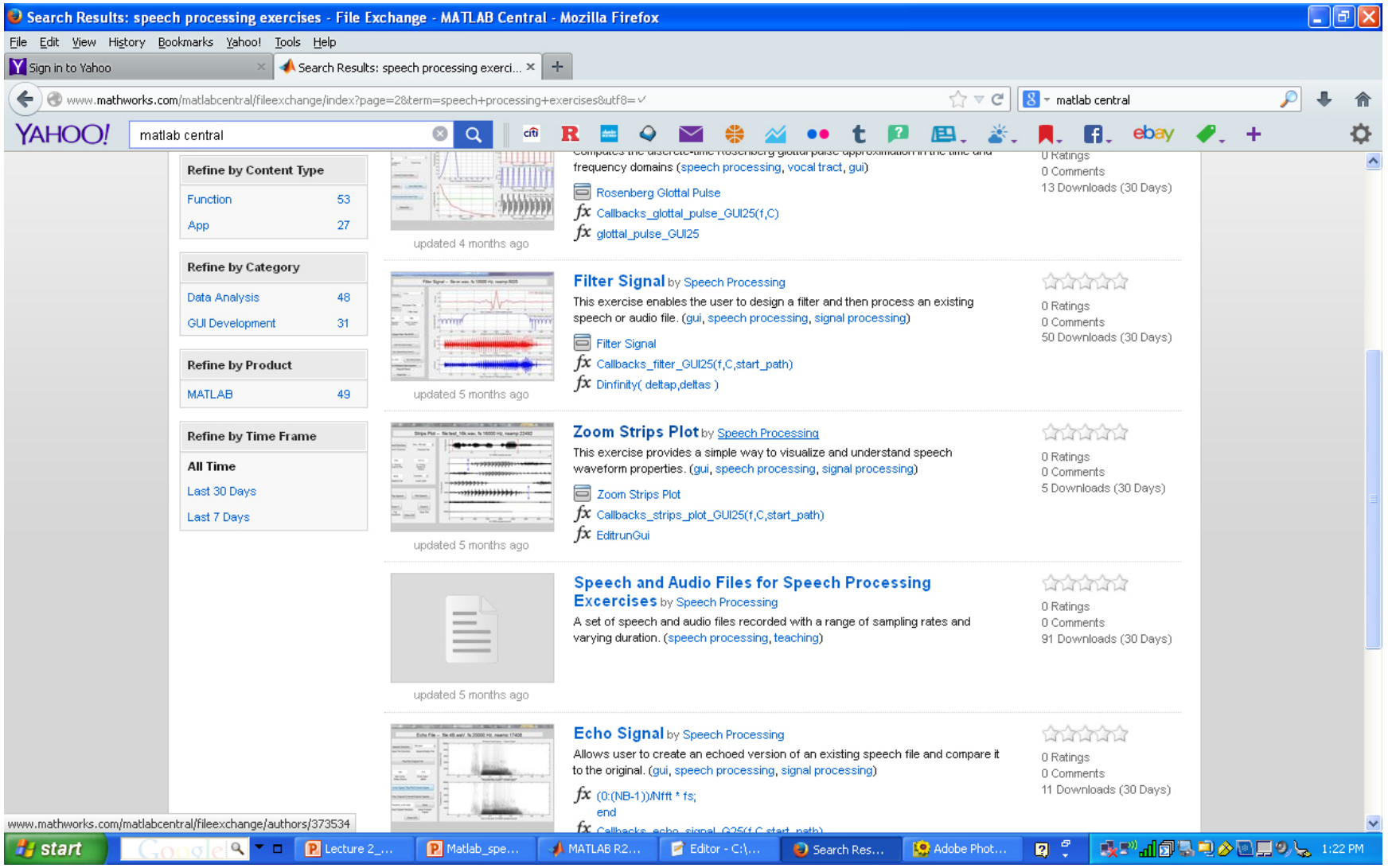
MATLAB Central APPs



MATLAB Central APPs



MATLAB Central APPs



MATLAB Central APPs