MATLAB Exercises – LPC Frame Solution

Program Directory: `matlab_gui\lpc_frame`
Program Name: `lpc_frame_GUI25.m`
GUI data file: `lpc_frame.mat`
Callbacks file: `Callbacks_lpc_frame_GUI25.m`
TADSP: Sections 9.2.1-9.2.4, pp. 477-485, Section 9.5.3, pp. 516-525, Problem 9.25

This MATLAB exercise shows how the method of linear predictive coding (LPC) models a speech frame in terms of its fit to the log magnitude spectrum of the speech frame STFT.

LPC Frame Solution – Theory of Operation

In this exercise, all three methods of linear predictive analysis, as discussed in TADSP, are used and compared as to their ability to match the short-time spectral properties of a frame of speech. The three methods of linear prediction that are used are:

1. the autocorrelation method,
2. the covariance method,
3. the lattice method

Each method of linear prediction is used to analyze a designated frame of speech and to illustrate the way in which the log magnitude spectrum of the LPC analysis model provides a spectral fit to the log magnitude of the STFT of the given speech frame.

This MATLAB exercise computes the log magnitude of the STFT of a specified frame of speech. Then, using the same frame of speech, the exercise computes LPC log spectral matches to the speech frame, using the autocorrelation method, the covariance method, and the lattice method. The four resulting log magnitude responses (from the STFT of the speech frame and from the three LPC analysis methods) are then plotted on a common plot, and compared in terms of how each LPC analysis method approximates the STFT log magnitude response.

The frame duration, L_m, is first converted from its value in msec to its value, `L`, in samples at the speech file sampling rate. Then the speech file is read in and the STFT log magnitude spectrum (of the frame of speech starting at sample `ss`)[1] is calculated using an FFT size of `nfft=1024`. A log spectral level correction factor is computed due to the use of different window sizes and types for the LPC analysis routines. For a window type of 1 (Hamming window), the log spectral correction factor is applied to both the covariance method solution and to the lattice method solution, and is of the form:

$$U = \sum_{l=0}^{L-1} \frac{[w(l)]^2}{L+p} \tag{1}$$

and for a window type of 0 (rectangular window), the log spectral correction factor is:

$$U = \frac{L}{L+p} \tag{2}$$

with the final log spectral correction of $U_{\text{log}} = 10 * \log_{10}(U)$.

The next step in the solution to this MATLAB Exercise is to compute the LPC model for the designated speech frame for the three LPC analysis methods. The autocorrelation method is computed using the MATLAB function `durbin.m`; the covariance method is computed using the MATLAB function `cholesky.m`; and the lattice solution is computed using the MATLAB function `lattice.m`. The LPC polynomials for the three methods of solution are converted to log magnitude spectral fits using the MATLAB function `freqz.m`, and the four log magnitude spectral responses are plotted on a common plot.

[1] The frame starting sample is interactively chosen using the method described in the next section.

Interactive Method of Defining the Speech Analysis Frame Starting Sample

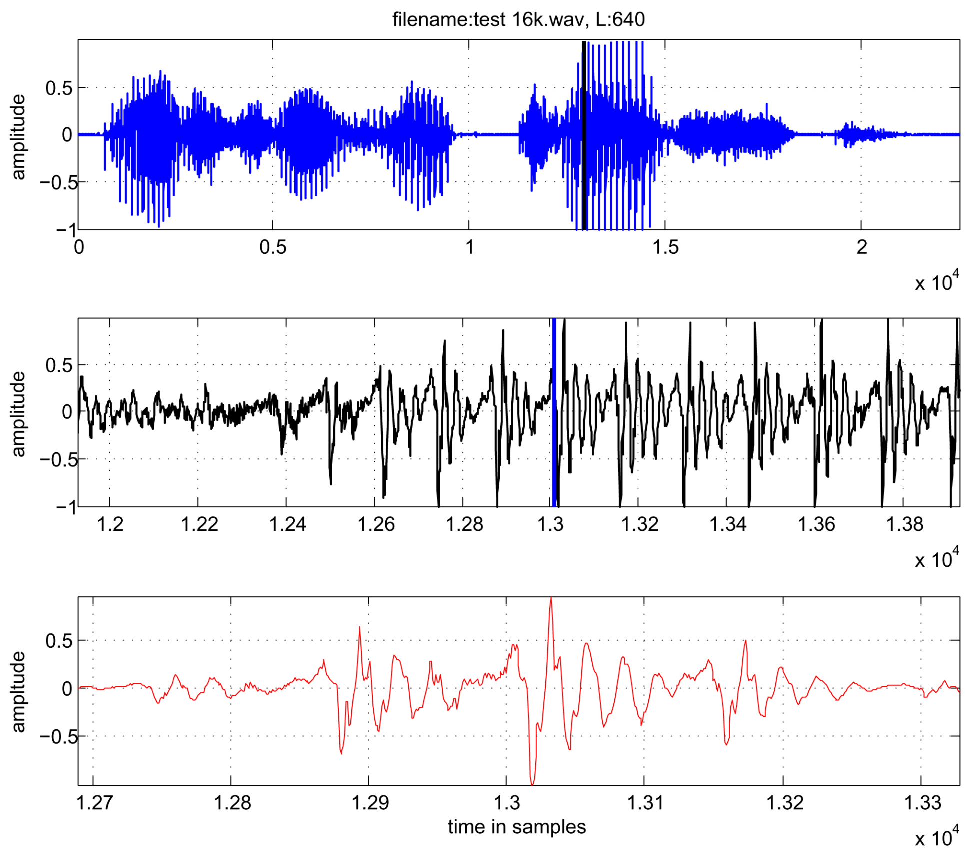

Figure 1: Sequence of waveform plots defining how the user can interactively choose a starting sample for the current analysis frame.

Several MATLAB Exercises rely on frame-based analysis methods where the user needs to specify both the speech file for analysis, and the starting sample of the speech analysis frame of interest. The method that we have chosen to define the frame starting sample is an interactive analysis which homes in on an appropriate analysis frame in a series of steps. The operations of this interactive method for determining the starting sample of the speech analysis frame for autocorrelation analysis proceed as follows:

1. In a specified graphics frame (or figure sub-frame) a single line plot of the entire speech waveform is obtained, as illustrated at the top panel of Figure 1. A graphics curser then appears allowing the user to move the cursor to the region of speech that is of interest for specifying the current analysis frame. A solid vertical cursor is shown at the place selected by the user. For the example of Figure 1 the cursor location is approximately sample 13000, as indicated by the solid red bar.

2. In another specified graphics frame (or figure sub-frame) a plot of the speech signal over a region that is about ± 1000 samples around the location of the cursor in the previous step; i.e., from sample 12000 to sample 14000. A second graphics cursor appears allowing the user to move the cursor to the exact starting sample of interest (to within the resolution of the display) for specifying the current analysis frame, as illustrated in the middle graphics panel of Figure 1. Here the cursor is again shown in the area of sample 13000.

3. The current analysis frame is then defined as the frame of speech from the starting sample of step 2 minus half

the window length, to the starting sample of step 2 plus half the window length. The designated analysis frame is then weighted by the analysis window (Hamming in the case here) and plotted in the bottom graphics panel.

It should be clear that the three steps of the above process for choosing an analysis frame can be implemented in either a single graphics panel or frame (by simply overwriting the graphics panel with the new speech signal) or in a series of graphics panels or frames. The current exercise uses one of the 8 graphics panels and overwrites the speech waveform plot at each step of the analysis. This process is a very useful and efficient one for choosing a region of interest within the speech signal, and then homing into a particular analysis frame using the steps outlined above.

LPC Frame Solution – GUI Design

The GUI for this exercise consists of two panels, 3 graphics panels, 1 title box and 12 buttons. The functionality of the two panels is:

1. one panel for the graphics display,
2. one panel for parameters related to the linear prediction analysis, and for running the program.

The set of three graphics panels is used to display the following:

1. the section of the speech waveform in the vicinity of the selected speech frame,
2. the speech frame, suitably weighted by the LPC analysis window,
3. the log magnitude response of the speech frame STFT along with the LPC log magnitude spectrum from each of the three LPC analysis methods.

The title box displays the information about the selected file along with the set of LPC analysis parameters. The functionality of the 12 buttons is:

1. a pushbutton to select the directory with the speech file that is to be analyzed using short-time analysis methods; the default directory is 'speech_files',
2. a popupmenu button that allows the user to select the speech file for analysis,
3. a pushbutton to play the current speech file,
4. an editable button that specifies the frame duration, L_m, (in msec) for short-time analysis; (the default value is $L_m = 40$ msec),
5. an editable button that specifies the frame shift, R_m, (in msec) for short-time analysis; (the default value is $R_m = 10$ msec),
6. an editable button that specifies the LPC system order, p; (the default value is $p = 12$),
7. a popupmenu button that lets the user choose either a Hamming or Rectangular window as the short-time analysis window used to compute the LPC solution (strictly for the autocorrelation method of LPC analysis); (default is Hamming window),
8. a popupmenu button that lets the user choose either a Hamming or Rectangular window as the short-time analysis window used to compute the speech frame STFT; (default is Hamming window),
9. a pushbutton to determine the single frame starting sample, `ss`, using the iterative method described above; this starting sample defines the current frame of the speech signal,
10. a pushbutton to run the analysis code and display the signal processing results using the current frame of the speech signal; this button can be pressed and used as often as desired, changing one or more analysis parameters while keeping the frame starting sample the same,

11. a pushbutton to run the analysis code and display the signal processing results using the next frame of signal; i.e., the frame with starting sample set to `ss+R` where `R` is the frame shift in samples; this button can be pushed repeatedly to provide a frame-by-frame analysis,

12. a pushbutton to close the GUI.

LPC Frame Solution – Scripted Run

A scripted run of the program 'lpc_frame_GUI25.m' is as follows:

1. run the program 'lpc_frame_GUI25.m' from the directory 'matlab_gui\lpc_frame_solution',

2. hit the pushbutton 'Directory'; this will initiate a system call to locate and display the filesystem for the directory 'speech_files',

3. using the popupmenu button, select the speech file for short-time feature analysis; choose the file 'test_16k.wav' for this example,

4. hit the pushbutton 'Play Speech File' to play the selected speech file,

5. using the editable buttons, choose the following initial values for the speech analysis parameters; 40 msec for the frame length L_m, 10 msec for the frame shift, R_m, and 12 for the LPC system order, p,

6. using the popupmenu button choose Hamming for the STFT analysis window,

7. using the popupmenu button choose Hamming for the LPC analysis window,

8. hit the 'Get Frame Starting Sample' button to interactively choose the initial analysis frame starting sample, `ss`, using the iterative method described above; try to choose the starting sample as close to the value of 12879 so as to match the plotted results for this example exercise,

9. hit the 'Run Current Frame' button to initiate single frame analysis of the speech beginning at the current frame starting sample, `ss`; the results of LPC frame analysis are shown in the various graphical plots; the 'Run Current Frame' button can be hit repeatedly after making changes in the analysis frame parameters,

10. hit the 'Run Next Frame' button to initiate single frame analysis on the next frame of speech, i.e., where the starting sample of the next frame is set to `ss+R`, where `R` is the frame shift in samples,

11. experiment with different choices of speech file, and with different values for Lm, Rm, p, and both window types,

12. hit the 'Close GUI' button to terminate the run.

An example of the graphical output obtained from this exercise using the speech file 'test_16k.wav' is shown in Figure 2. The graphics panels show the speech waveform in the vicinity of the chosen starting frame (top graphics panel), the speech frame used for STFT and LPC spectrum analysis (second graphics panel), and the log magnitude spectral response of the STFT analysis along with the LPC spectrum for each of the three methods of LPC analysis (bottom graphics panel). It should be clear from the Figure 2 that the LPC fits to the speech STFT log magnitude spectrums for this frame of speech are excellent, and that there are only small differences between the LPC models as computed for the three methods of LPC analysis.

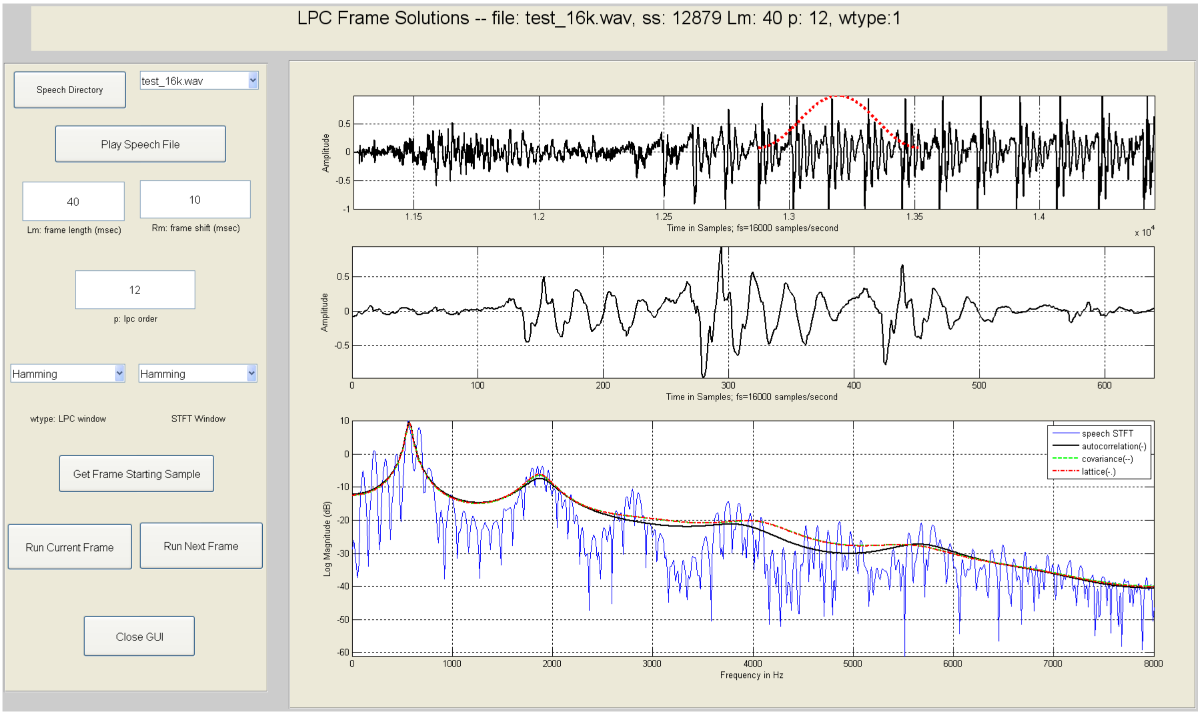

Figure 2: Graphical output from LPC analysis of a frame of voiced speech. The upper graphics display shows the speech waveform in the vicinity of the selected speech frame; the middle graphics panel shows the window-weighted speech signal; the bottom graphics panel shows the log magnitude response of the STFT of the speech frame along with the LPC spectrum from each of the three methods of LPC analysis.

LPC Frame Solution – Issues for Experimentation

1. run the scripted exercise above, and answer the following:
   - how closely does the resulting LPC model match the STFT peaks and valleys?
   - how can you explain the matching properties of the LPC model?
   - what accounts for the differences between the 3 types of LPC model, and their resulting matches to the STFT log magnitude spectrum?
   - what would change in terms of the matching properties of the LPC system if the LPC order was increased from $p = 12$ to $p = 30$?
   - what would change in terms of the matching properties of the LPC system if the LPC order was decreased from $p = 12$ to $p = 6$?
   - is there an ideal LPC model order for this example? If so, what is it? If not, what happens as p increases to very large size LPC systems?
2. repeat the LPC system analysis on an unvoiced section of the current speech signal, or on an unvoiced section of another speech signal
   - how well does the resulting LPC model match the unvoiced signal STFT log magnitude spectrum?
   - is there any reason why the matches during unvoiced sections of speech are not as good as during voiced sections?
   - what happens to the LPC model match as p gets as big as the frame duration, L_m?