MATLAB Exercise – Cepstrum Computation

Program Directory: `matlab_gui\cepstrum_computation`
Program Name: `compute_cepstrum_GUI25.m`
GUI data file: `compute_cepstrum.mat`
Callbacks file: `Callbacks_compute_cepstrum_GUI25.m`
TADSP: Section 8.2.3, pp. 409-412, Section 8.4.1, pp. 429-434

This MATLAB exercise compares two different methods of cepstrum analysis of a finite duration frame of speech, namely the conventional method based on FFTs (with frame unwrapping for the complex cepstrum), and an analytical method based on solving for the roots of a finite order numerator polynomial and directly computing the cepstrum from the locations of the polynomial roots.

Cepstrum Computation – Theory of Operation

This exercise analyzes a finite duration frame of windowed speech and computes both the real and the complex cepstrum of that frame of speech using both the conventional FFT method (using the MATLAB Signal Processing Toolbox routines `cceps` and `rceps`) and using an analytical method based on solving for the numerator roots of the speech polynomial.

Analytical Computation of the Complex Cepstrum

The MATLAB function, `compute_cep.m` computes the complex cepstrum of a finite duration sequence (e.g., a frame of speech) by factoring the finite order polynomial (the windowed speech frame of length L samples) using the MATLAB root finding routine, `roots.m`, and then sequentially combining the cepstral contributions of each (complex/real) root of the speech polynomial to give the composite complex cepstrum. The detailed steps that must be carried out in the MATLAB function, `compute_cep.m` are explained in the discussion below.

Consider the case of computing the complex cepstrum of an FIR sequence (weighted speech frame) of the form:

$$x[n] = \sum_{m=0}^{L-1} X_m \delta[n-m], \qquad 0 \leq n \leq L-1 \tag{1}$$

where L is the duration (in samples) of the FIR sequence (typically L is on the order of 40 msec, or 400 samples at a sampling rate of $f_s = 10000$ Hz), and X_m represents the polynomial coefficient for the m^{th} degree term.

We can form the z-transform of Eq. (1) and express it in the form:

$$\begin{aligned} X(z) &= \sum_{n=0}^{L-1} x[n] z^{-n} && (2) \\ &= \prod_{n=1}^{L-1} (1 - b_n z^{-1}) && (3) \\ &= \prod_{n_1=1}^{N_1} (1 - b_{n_1} z^{-1}) \cdot \prod_{n_2=1}^{N_2} (1 - b_{n_2} z^{-1}) && (4) \end{aligned}$$

(where $L = N_1 + N_2$) by factoring the $z-$transform polynomial into two sets of (complex) roots (using the MATLAB function `roots.m`), where we have assumed (without any loss of generality) that the leading term has amplitude of 1, i.e., $x[0] = X_0 = 1$. We see that when $|b_{n_1}| < 1$, the first term in Eq. (4) represents the set of N_1 zeros of $X(z)$ that are inside the unit circle, and when $|b_{n_2}| > 1$ the second term represents the set of N_2 zeros of $X(z)$ that are outside the unit circle. All zeros are either complex (in which case they occur in complex conjugate pairs), or real.[1]

[1] In practice there are never any complex roots on the unit circle, i.e., the root magnitude is never exactly 1.0; hence all complex roots are either inside or outside the unit circle.

We factor the term with the zeros outside the unit circle into the form:

$$\prod_{n_2=1}^{N_2}(1-b_{n_2}z^{-1}) = z^{-N_2}\cdot\prod_{n_2=1}^{N_2}[-b_{n_2}]^{-1}\cdot\prod_{n_2=1}^{N_2}\left(1-\frac{z}{b_{n_2}}\right) \tag{5}$$

We can now express the cepstrum of the individual terms in Eqs. (4) - (5) using the logarithm power series of the form:

$$\log(1-X) = -\sum_{n=1}^{\infty}\frac{X^n}{n} \tag{6}$$

$$\log(1+X) = -\sum_{n=1}^{\infty}(-1)^n\frac{X^n}{n} \tag{7}$$

leading to the z-transform of the complex cepstrum of $x[n]$ in the form:

$$\hat{X}(z) = \log[z^{-N_2}] + \sum_{n_2=1}^{N_2}\log|b_{n_2}|^{-1} + \sum_{n_1=1}^{N_1}\log(1-b_{n_1}z^{-1}) + \sum_{n_2=1}^{N_2}\log\left(1-\frac{z}{b_{n_2}}\right) \tag{8}$$

We can now solve for the complex cepstrum in the form:

$$\hat{x}[n] = \sum_{n_2=1}^{N_2}\log|b_{n_2}|^{-1} \quad n=0 \tag{9}$$

$$= -\sum_{n_1=1}^{N_1}\frac{(b_{n_1})^n}{n} \quad n>0 \tag{10}$$

$$= \sum_{n_2=1}^{N_2}\frac{(b_{n_2})^{-n}}{n} \quad n<0 \tag{11}$$

We can combine complex terms (since they occur in complex conjugate pairs) simplifying Eqs. (9) - (11) and giving the result:

$$b_{n_1} = \mathcal{R}[b_{n_1}] + j\mathcal{I}[b_{n_1}] \tag{12}$$

$$= |[b_{n_1}]|\cdot e^{j\angle b_{n_1}} \tag{13}$$

$$|[b_{n_1}]| = \sqrt{(\mathcal{R}[b_{n_1}])^2 + (\mathcal{I}[b_{n_1}])^2} \tag{14}$$

$$\angle b_{n_1} = \arctan\left(\frac{\mathcal{I}[b_{n_1}]}{\mathcal{R}[b_{n_1}]}\right) \tag{15}$$

By combining complex conjugate terms we get:

$$(b_{n_1})^n + (b_{n_1}^*)^n = |[b_{n_1}]|^n\cdot e^{jn\angle b_{n_1}} + |[b_{n_1}]|^n\cdot e^{-jn\angle b_{n_1}} \tag{16}$$

$$= |(b_{n_1})|^n\cdot 2\cdot\cos(n\angle b_{n_1}) \tag{17}$$

Using these results we can readily evaluate the complex (or real) cepstrum from the roots of the FIR polynomial, sorted into those inside and those outside the unit circle, and sorted into real and complex roots (which occur in complex conjugate pairs).

Cepstrum Computation – GUI Design

The GUI for this exercise consists of two panels, 3 graphics panels, 1 title box and 11 buttons. The functionality of the two panels is:

1. one panel for the graphics display,

2. one panel for parameters related to the cepstral computation, and for running the program.

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

1. a user-designated section of the speech waveform on a normalized amplitude scale, initially showing the location of the current speech frame for analysis (the outline of a Hamming window is shown over the frame of speech for analysis), and finally showing the current windowed frame of speech,
2. the real cepstrum of the current frame of speech, computed using both rceps (FFT-based approach from the MATLAB Signal Processing Toolbox) and the analytical solution based on root-finding for the numerator polynomial,
3. the complex cepstrum of the current frame of speech, computed using both cceps (FFT-based approach from the MATLAB Signal Processing Toolbox) and the analytical solution based on root-finding for the numerator polynomial.

The title box displays the information about the selected file for cepstral analysis. The functionality of the 11 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 that enables the user 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 length of the cepstral sequence, `ceplm`, (in msec) that is to be plotted; (the default value is `ceplm=30` msec),
7. an editable button that specifies the size of FFT, `nfft`, used for the computation of `rceps` and `cceps`; (the default value is `nfft=2048`),
8. a pushbutton to determine the single frame starting sample, `ss`, using the iterative method described below; this starting sample defines the current frame of the speech signal,
9. a pushbutton to run the cepstral 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,
10. a pushbutton to run the cepstral 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,
11. a pushbutton to close the GUI.

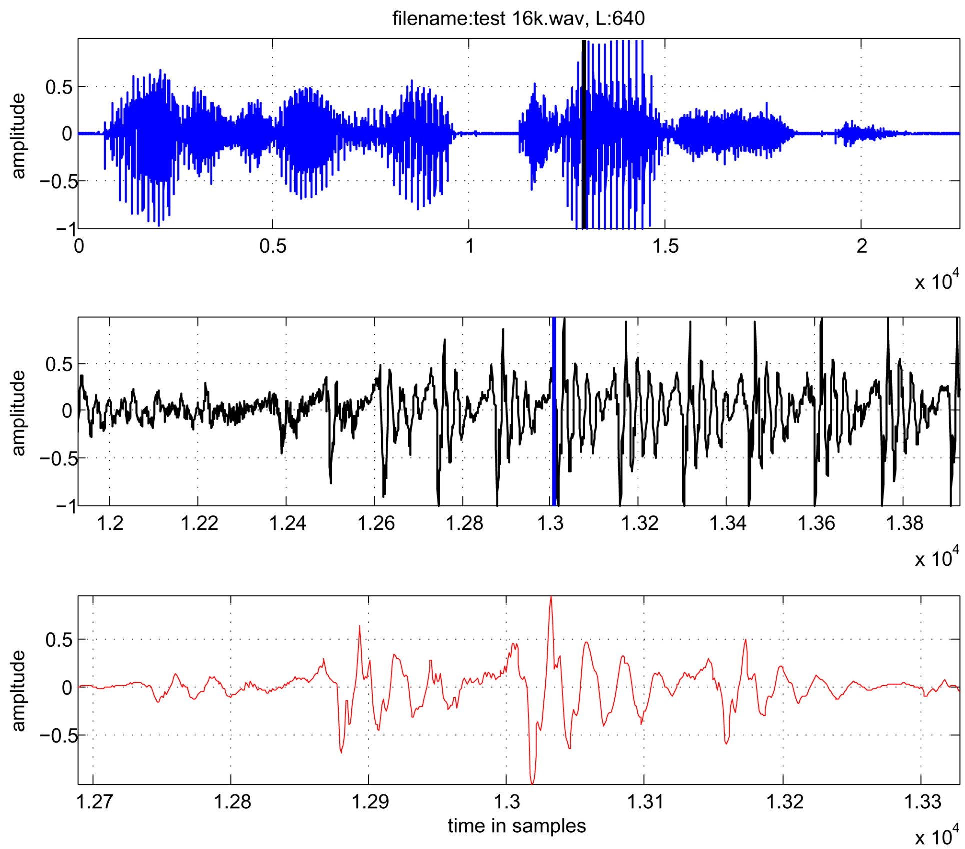

Figure 1: Sequence of waveform plots defining how the user can interactively choose a starting sample for the current analysis frame.

Interactive Method of Defining the Speech Analysis Frame Starting Sample

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.

Cepstrum Computation – Scripted Run

A scripted run of the program 'compute_cepstrum_GUI25.m' is as follows:

1. run the program 'compute_cepstrum_GUI25.m' from the directory 'matlab_gui\cepstrum_computation',
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 out the designated speech file,
5. using the editable buttons, choose 40 msec as the initial value for the frame length, L_m, 10 msec for the value of frame shift, R_m, 30 msec for the cepstral sequence length, `ceplm`, and 2048 for the FFT size, `nfft`
6. 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 1673 so as to match the plotted results for this example exercise,
7. hit the 'Run Current Frame' button to initiate single frame cepstral analysis of the speech beginning at the current frame starting sample, `ss`; the results of cepstral 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,
8. hit the 'Run Next Frame' button to initiate single frame cepstral 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,
9. experiment with different choices of speech file, and with different values for L_m, R_m, `ceplm`, and `nfft`,
10. 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 window-weighted speech waveform (top graphics panel), the short-time real cepstrum computed using the FFT method and also computed using the analytical method (middle graphics panel), the short-time complex cepstrum computed using the FFT method and also computed using the analytical method (bottom graphics panel). For this example the alignment between the analytically computed complex (and real) cepstrum, and the MATLAB code for the complex (`cceps`) and real (`rceps`) cepstrums is extremely close, with no major areas of difference. For other speech frames, the alignment is not nearly so good (especially for the complex cepstrum computations) and the user can find frames of speech where the pair of cepstrums do not align well, clearly due to phase unwrapping and aliasing issues that invariably arise when computing complex cepstrums with FFTs and phase unwrapping routines. In general the alignment between the analytical real cepstrum and the real cepstrum computed using FFT methods (i.e., the MATLAB routine `rceps`) is extremely good as there are no phase unwrapping problems to deal with, and aliasing can be made small by using a sufficiently large size FFT.

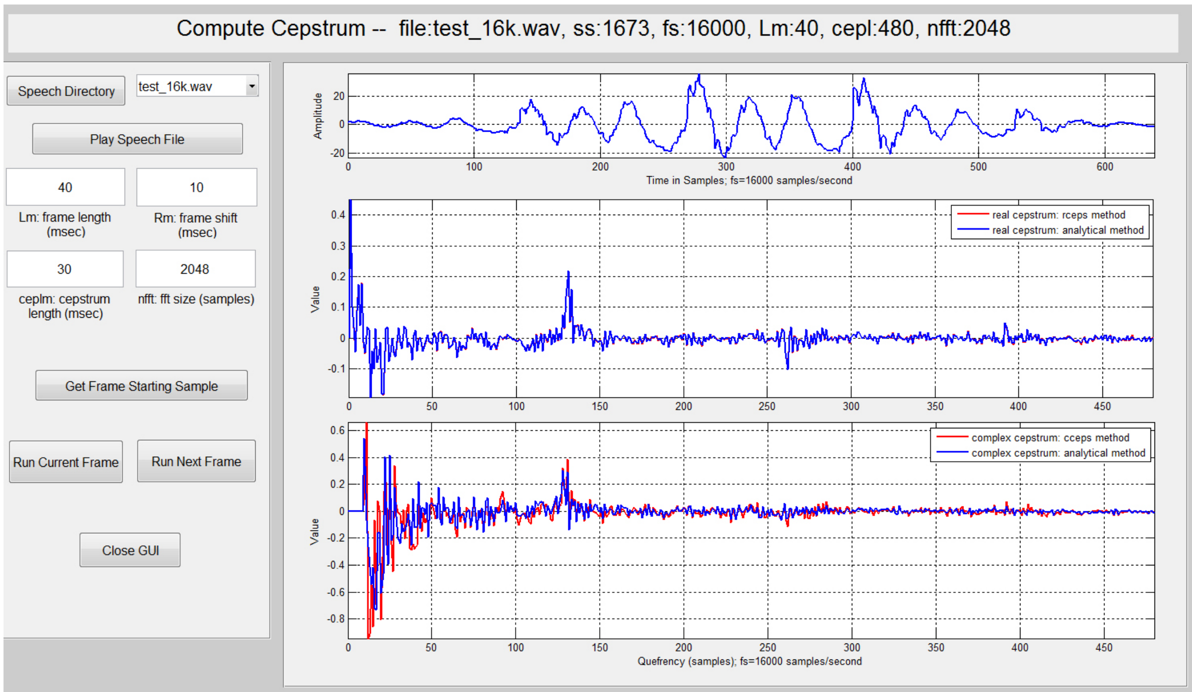

Figure 2: Short-time cepstral analysis of the speech file 'test_16k.wav'. The graphics panels show the region of the speech waveform (top graphics panel), the short-time real cepstrum computed using both `rceps` and an analytical method (middle graphics panel), and the short-time complex cepstrum computed using both `cceps` and an analytical method (bottom graphics panel

Cepstrum Computation – Issues for Experimentation

1. run the scripted exercise above, and answer the following:

   - estimate the pitch period (in samples) of the section of the window-weighted waveform from the speech file 'test_16k.wav' in the vicinity of sample 1673
   - estimate the pitch period from the real cepstrum displays of the middle graphics panel; does the period agree (approximately) with the period estimated from the waveform plot?
   - compare the real cepstrum, as obtained from direct computation using an FFT and an inverse FFT, with the real cepstrum, as obtained from the analytical method based on factoring the numerator polynomial of the speech frame and using the power series expansions to sum together the cepstrum associated with each root (both real and complex roots)
   - compare the complex cepstrum, as obtained from direct computation using an FFT and an inverse FFT, with phase unwrapping, to the complex cepstrum obtained from the analytical method based on root finding; why are there many more regions of difference for the complex cepstrum computation than for the real cepstrum computation?

2. change the size of the FFT (double it) and repeat the above experiments; do the magnitudes of the errors become a lot smaller? if so, what is the reason for the decline; if not, why is the error not decreasing?

3. select a frame of voiced speech from any of the speech files in the speech file library. Run the cepstral analysis program on this frame (by iterating the 'Run Current Frame' button), and change the FFT size (`nfft`) from 512 to 8192 in powers of 2. Observe the effects of cepstrum aliasing in the real cepstrum.

4. in the complex cepstrum comparison, the errors are quite large for small values of `nfft`. Explain what could be the cause of such errors.