MIT OpenCourseWare
http://ocw.mit.edu

HST.582J / 6.555J / 16.456J Biomedical Signal and Image Processing
Spring 2007

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.

Harvard-MIT Division of Health Sciences and Technology
HST.582J: Biomedical Signal and Image Processing, Spring 2007
Course Director: Dr. Julie Greenberg

HST-582J/6.555J/16.456J-Biomedical Signal and Image Processing-Spring 2007

Laboratory Project 2
Speech Coding

DUE: 4/5/07

1 Introduction

In this laboratory exercise we will explore two different methods of speech coding, a channel vocoder and a linear prediction (LP) vocoder. Both methods are based on the *source-filter model* of speech and require determination of voicing and pitch to code the *source* portion of the model. The channel vocoder uses short-time Fourier analysis to code the *filter* portion of the model, while the LP vocoder uses an all-pole model of speech to estimate the *filter*.

2 Speech

The speech signal represents the variations in acoustic pressure at a certain distance from the lips of a talker. The amplitude of these pressure variations can vary from about 30 dB above the threshold of hearing (i.e. 20 μPa) for whispered speech to 100 dB for shouted speech. Even within a single utterance, intensity variations of 40 to 60 dB are typical.

While there may be significant energy in the speech signal up to 10 kHz or more, there is essentially no loss of intelligibility if the speech signal is bandlimited to 4 or 5 kHz. Indeed, speech heard through a telephone receiver is bandlimited to 3 kHz.

2.1 The source-filter model

The speech signal can be considered as the output of a time-varying *filter* excited by a *source* signal. (See Fig. 1.) The filter and the source can be controlled almost independently by the talker. In this model of speech production, the filter represents the resonant properties of the vocal tract and the radiation characteristics at the lips.

The source is either a *voicing source* for vowels and some consonants, or a *noise source* for voiceless sounds (such as unvoiced fricative and stop consonants), or a combination of noise and voicing (for voiced fricatives). The voicing source is a quasi-periodic signal produced by vibration of the vocal folds. Its *fundamental frequency*, which listeners hear as the *pitch* of voice, is in the range 80-160 Hz for male voices, 160-320 Hz for female voices, while its

Cite as: Julie Greenberg. Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

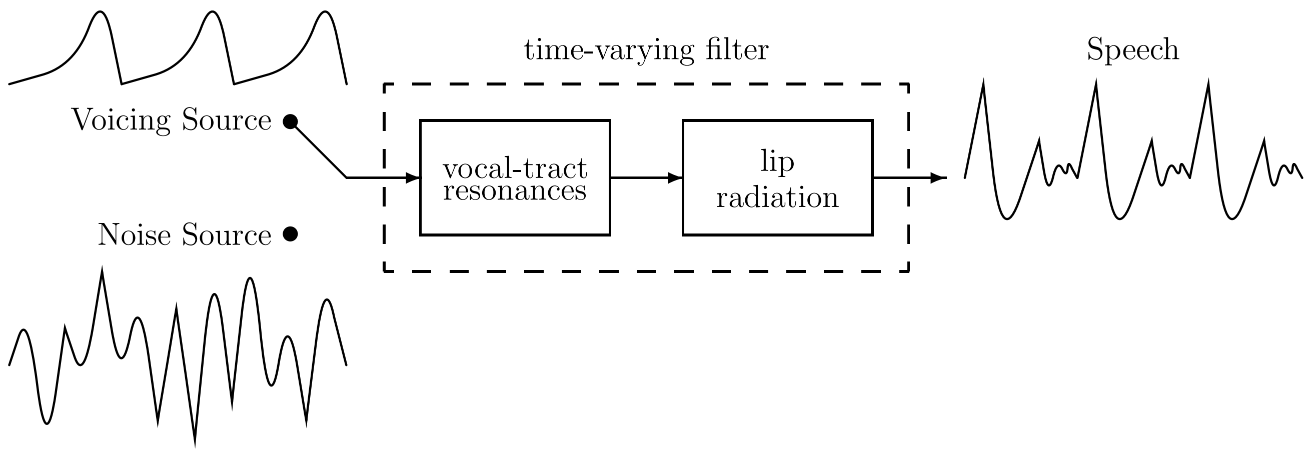

Figure 1: Source-filter model of speech production

spectral envelope falls with frequency at a rate of about 12 dB/octave (i.e. as $1/F^2$). The spectrum of the noise source can be considered flat over the frequency range of interest.

The frequency response of the filter that represents modifying effects of the vocal tract on the source signal shows a series of sharp peaks or *formants* corresponding to resonance frequencies. The spacing between formant frequencies is about 1 kHz on the average for a vocal tract of 17 cm. In running speech, formant frequencies change continuously over time, reflecting the motions of the articulators. Specifically, the frequency of the first formant correlates with the degree of opening of the vocal tract, while the second formant frequency correlates with the front-back position of the tongue. The radiation characteristics of the lips result in an emphasis of high frequency components with a slope of 6 dB/octave. Thus, the short-time spectrum of a voiced sound shows an overall downward tilt of 6 dB/octave, 12 dB/octave due to the voicing source, minus 6 dB/octave due to radiation characteristics. Spectral analysis techniques usually give better results if this overall tilt is removed by an appropriate *preemphasis* filter.

2.2 Vocoders

In the transmission and storage of digitized speech signals, it is often necessary to minimize the amount of data. This is accomplished by efficient coding to reduce the number of bits per unit time, or *bit rate*, required to represent the speech signal. A vocoder consists of two parts, an analyzer, or encoder, which extracts the parameters of the speech signal for transmission, and a synthesizer, or decoder, which reconstructs the speech signal from the received parameters.

The two speech vocoders considered in this lab are based on the source-filter model of speech,

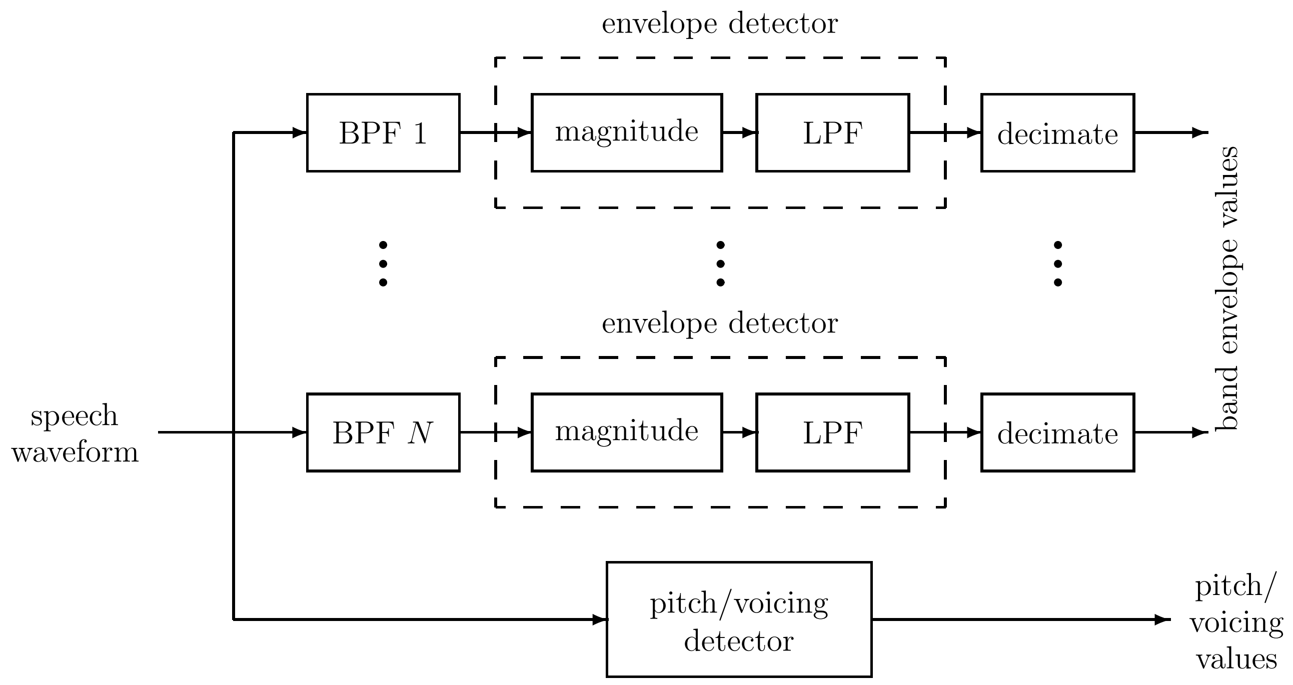

Figure 2: Channel vocoder analyzer

that is, the idea that the speech signal can be accurately represented by, and reconstructed from, two sets of parameters, representing the *source* and the *filter*. The first set of parameters contains source information indicating whether the current data frame is voiced or unvoiced. If the frame is voiced, it includes an estimate of the frequency of the voicing source (pitch). Many different algorithms have been proposed to detect voicing and pitch of speech signals; a simple method based on the autocorrelation function will be described and implemented in Sec. 3.2.

2.2.1 Channel vocoder

One of the earliest methods of speech coding is the *channel vocoder*. (See Figs. 2 and 3.)

The channel vocoder analyzer extracts the *filter* information by starting with a bandpass filter bank that splits the incoming speech signal into a number of frequency bands, or channels. The envelope of each bandpass filtered signal is determined by full-wave rectification (taking the magnitude) and lowpass filtering. Each band signal is then *decimated*, or *downsampled*; this function is primarily responsible for achieving the reduction in bit rate obtained by the vocoder. The resulting *band envelope values* comprise the second set of outputs from the channel vocoder analyzer. For each frame of data, these signals indicate the magnitude of the envelope of the speech waveform in a different frequency region.

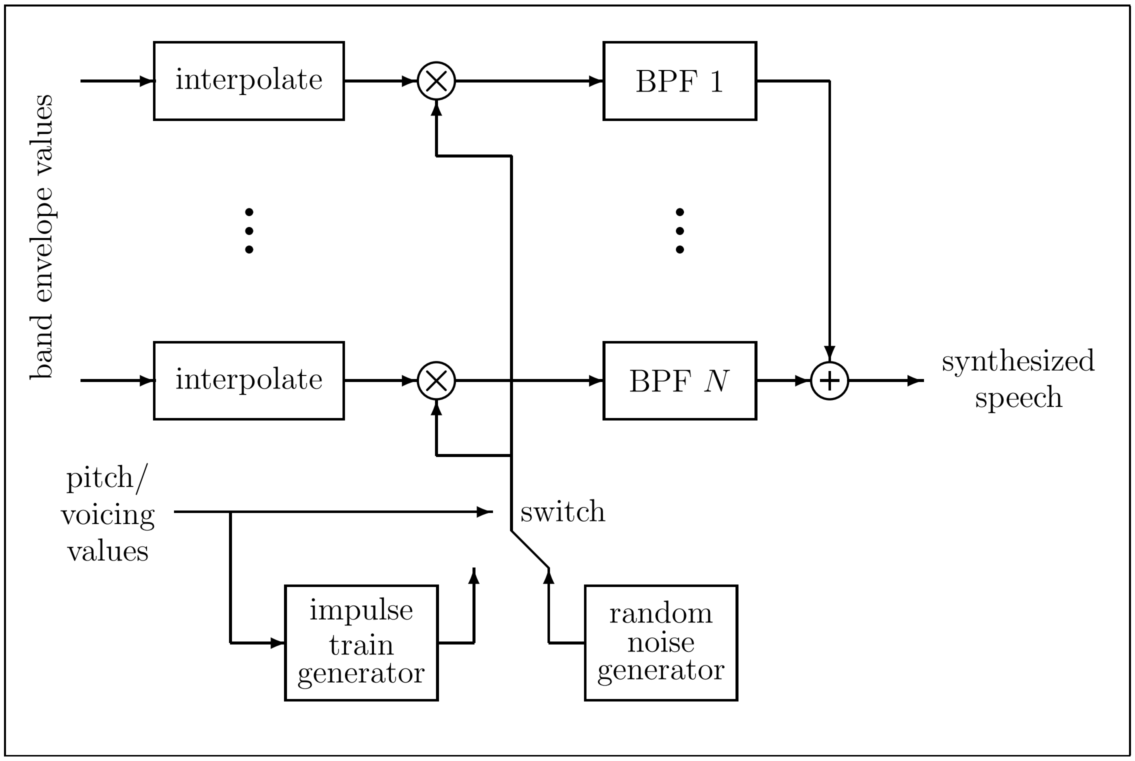

Figure 3: Channel vocoder synthesizer

The channel vocoder synthesizer uses the two outputs of the analyzer (pitch/voicing values and band envelope values) to reconstruct the speech waveform. The pitch/voicing values control the detailed structure of the output for each frame of data. The synthesizer contains two sources, a random noise generator and an impulse train generator. The pitch/voicing values determine which of these two sources is activated for a given frame of data. For unvoiced frames, the random noise generator is used, while for voiced frames, the impulse train generator is used, and the frequency of the impulses is determined from the value of the pitch for that frame. The band envelope values control the amplitude of the contribution of a particular channel. For each channel, the band envelope values are *interpolated*, or *upsampled*, to the desired sampling rate and then used to scale the source signal. These intermediate signals are processed by the same bandpass filter bank used in the analyzer. Finally, the bandpass filter outputs are summed to produce the synthesized speech output.

2.2.2 Linear prediction vocoder

One of the most effective methods of speech coding is the linear-prediction (LP) vocoder. Linear prediction is a technique that deconvolves the contributions of the source and the filter by fitting an *all-pole* or *autoregressive* model to the signal. Because all-pole models are well motivated by the acoustics of speech production and deviations from the all-pole model have only weak perceptual effects, properly-designed LP vocoders give very high speech quality. (See Fig. 4.)

Cite as: Julie Greenberg. Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

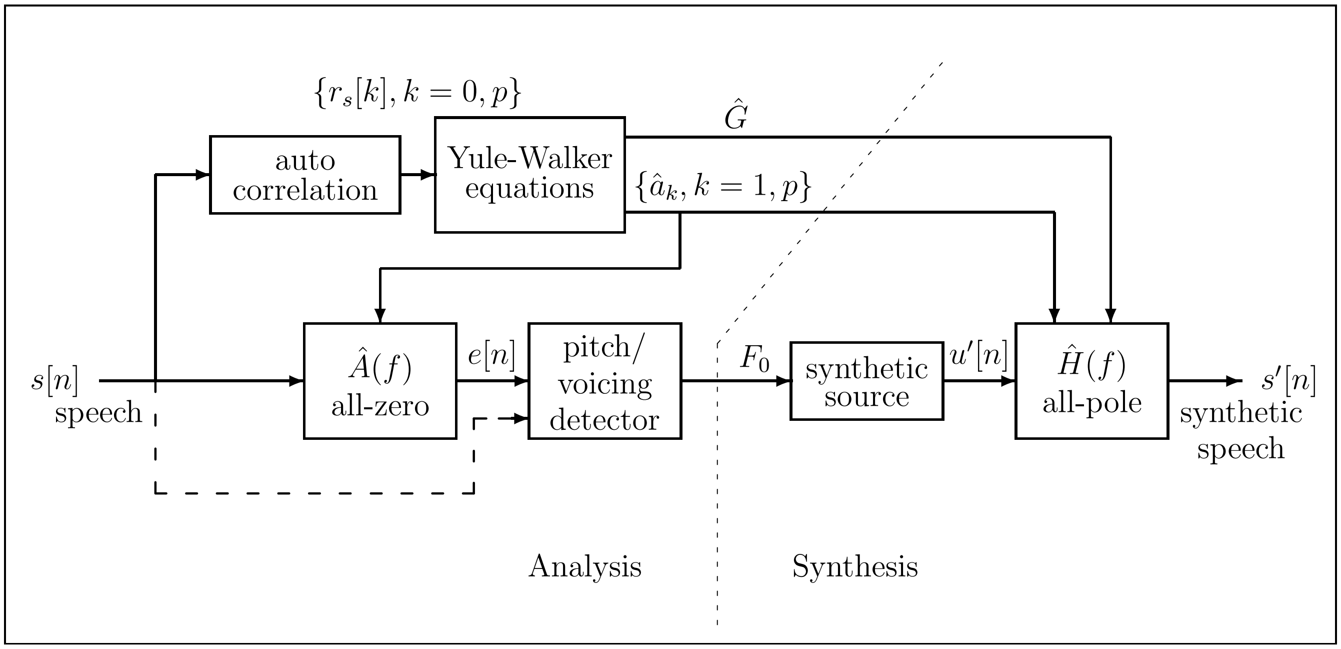

Figure 4: LP vocoder

The LP analyzer determines the *filter* information by fitting an all-pole filter to the speech signal. The original speech signal is filtered by an inverse filter based on the estimated filter coefficients to determine the error signal. The error signal approximates the *source*, and pitch detection is performed on this error signal.

The LP vocoder synthesizer uses the two outputs of the analyzer (pitch/voicing values and all-pole filter coefficients) to reconstruct the speech waveform. Like the channel vocoder synthesizer, the LP synthesizer contains two sources, a random noise generator and an impulse train generator. The pitch/voicing values determine which of these two sources is activated for a given frame of data. For unvoiced frames, the random noise generator is used, while for voiced frames, the impulse train generator is used, and the frequency of the impulses is determined from the value of the pitch for that frame. The synthesized speech signal is obtained by simply passing the appropriate source signal through the all-pole filter for each frame.

References

Rabiner, L.R. and Schafer, R.W. “Short-Time Fourier Analysis,” chapter 6 of *Digital Processing of Speech Signals.* Prentice-Hall, Inc., Englewood Cliffs, 1978.

Cite as: Julie Greenberg. Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

3 Specific Instructions

3.1 Data acquisition

The sound boards on the Sun workstations record and play sound at a fixed sampling rate of 8 kHz. Although Matlab appears to allows you to specify F_s when using the **sound** or **soundsc** command on a Sun workstation, the specified value of F_s is ignored, and the sound is played at 8 kHz.

1. **Optional: Record your own speech segment.** Select an utterance that is roughly 2-3 seconds in duration, and rich in different types of speech sounds, including vowels and unvoiced consonants such as "sh", "ch", "t", "s", "k", or "p". To record the sentence, at the Athena* prompt type "sdtaudio". Sdtaudio works like a tape recorder. If you are using headphones, verify that the built-in speaker is turned off by running "sdtaudiocontrol" and deselecting the build-in speaker option. Do not speak directly into the microphone to avoid large pressure transients. Save the recorded utterance into an audio file (.au) or wav file (.wav) and then read it into matlab using **auread** or **wavread**, respectively.
If you do not wish to record your own speech segment, you can simply **load** *one of the prerecorded sentences in* **/mit/6.555/data/vocod/cw16*_8k.mat**.

2. Listen to the utterance (**soundsc**).

3. Plot the time waveform. Make an attempt to identify the different phonemes within the utterance.

Question 1 *What is the speech segment that you have recorded/chosen? In your report, include the time plot with your attempt at labelling the phonemes.*

3.2 Pitch Detection

1. Extract a 100 ms segment from the utterance corresponding to a portion of a stressed vowel. (Save this segment for use in later portions of the lab exercise.) Lowpass filter the vowel segment to preserve frequencies below 500 Hz.

2. Compute the autocorrelation of the lowpass filtered vowel segment (**xcorr**). In order to extract the relevant portion of the result, note that the zeroth lag of the correlation is in the middle of the output sequence. Examine the autocorrelation for lags up to 30 ms. Manually estimate the pitch (fundamental frequency) from the autocorrelation of the vowel segment.

3. Typically, the autocorrelation function of vowel segments contains peaks due to both the fundamental period of speech and the vocal tract transfer function. In order to use the autocorrelation function for automatic pitch detection, it is helpful to suppress the peaks due to the vocal tract transfer function. This can be accomplished by center clipping the

* "Note for OCW users. Athena is MIT's UNIX-based academic computing facility. The necessary Athena-based files and directories referenced herein have been provided in the supporting ZIP archives."

lowpass filtered speech signal (**cclip**). The clipping limits are typically selected to be 75% of the minimum and maximum values of the speech segment. Apply center clipping to the vowel segment and examine your result. Manually estimate the pitch from the autocorrelation of the center clipped vowel segment.

4. Write a Matlab function to automatically determine pitch based on the autocorrelation function. The input to your pitch detector will be one segment, or *frame*, of speech data. If the largest peak after $R_x[0]$ in the autocorrelation function is small (less than 15–25% of $R_x[0]$), then that frame is determined to be unvoiced, and your function should return a pitch value of zero. Otherwise, that frame is classified as voiced, and your function should return the pitch value computed from the lag of the largest peak after $R_x[0]$ using the function **peak**. *Do not apply the lowpass filter within your pitch detector, since the entire utterance must be lowpass filtered* **before** *it is chopped up into frames.* An outline of the steps required is in **/mit/6.555/matlab/vocod/pitch_detect.m**. Feel free to copy this file to your subdirectory and use it as a framework to design your own pitch detector.

5. Test your pitch detector on the vowel segment from step 1 and, time permitting, on other voiced and unvoiced segments you can identify. Defer testing of the pitch detector on the complete utterance until the next section. Keep your pitch detector for use in the remainder of this lab exercise.

Question 2 *Describe how you determined the pitch manually. Include a plot of one autocorrelation sequence from either step 2 or step 3. How does the pitch value determined by the automated pitch detector compare to the values estimated manually in steps 2 and 3? Based on the manual and automatic estimates, what do you think is the pitch of the vowel segment? Is this value reasonable, given what you know about speech?*

3.3 Channel vocoder

In this section you will design a channel vocoder analyzer to encode a speech signal into pitch values and band envelope values corresponding to a number of frequency channels. Then you will test it by decoding the encoded speech with a channel vocoder synthesizer kindly provided by the 6.555 staff.

1. Write a Matlab function to generate a bandpass filter bank. We will rely on the technique demonstrated in Problem Set 2, #1. Your filter bank should have 18 bands, each with a 200 Hz bandwidth, centered at evenly spaced frequencies from 100 Hz to 3500 Hz. (In order to obtain a reasonable tradeoff between frequency resolution and processing speed, we suggest that you use 65-point FIR filters designed from a Kaiser window with $\beta = 3$.) See **/mit/6.555/matlab/vocod/filt_bank.m** for a template of the function. *You must name this function* **myownfilt_bank.m** *in order for the synthesizer function to find it.*

Question 3 *Select a few representative frequency bands in the filter bank and plot their fre-*

quency responses. Also plot the composite frequency response of the filter bank. **Hint: Think carefully about how to determine the composite frequency response. Should you add the frequency responses of the the individual bands before or after computing their magnitude?** *Is the composite frequency response what you expected? Why or why not?*

2. Use your two new Matlab functions (pitch detector and filter bank generator) to create the analyzer (encoder) portion of a channel vocoder. Your analyzer will have two outputs, a $N \times B$ matrix of band envelope values, where N is the number of frames and B is the number of frequency bands, and a N-dimensional pitch vector. The pitch detector portion of the analyzer splits the speech signal into overlapping frames of data, with the amount to advance between successive frames corresponding to the decimation rate used in the filter bank portion of the analyzer. For each frame, the pitch detector determines if it is voiced or unvoiced; if it is voiced, it will determine the pitch. Calling the pitch detector repeatedly will produce an output signal with one pitch value corresponding to each data frame. *A median filter* **(medfilt1)** *should be applied to this pitch signal to remove spurious values.* The filter bank portion of the analyzer splits the speech signal into frequency bands, computes the envelope values for each band by taking the magnitude (**abs**) and then lowpass filtering. The reduction in bitrate is then achieved by downsampling. (The Matlab command **decimate** includes the lowpass filter.) See **/mit/6.555/matlab/vocod/chvocod_ana.m** for a template of the function.

3. Process your utterance and examine the outputs. You may wish to start with a decimation rate of 100.

Question 4 *Plot the pitch values produced for the entire utterance. How well does your automated pitch detector perform? (Note that it is not necessary for your pitch detector to work perfectly, so long as it produces reasonable pitch values for most frames in the utterance.)*

4. Test your encoder by passing the outputs to the decoder provided (**chvocod_syn**) and listening to the synthesized speech output. *Be sure to save your synthesized utterance for submission and for comparison with the results of Sec. 3.5.*

5. **Optional**. Consider the effect of bit rate on speech quality by testing the channel vocoder with different decimation rates.

6. **Optional**. Produce monotone speech from the synthesizer by replacing the pitch signal with a constant value (forcing a pulse train of constant frequency for the source).

7. **Optional**. Produce whispered speech by replacing the pitch signal with a vector of zeros (forcing white noise for the source).

8. **Optional**. Change a male voice to a female voice by scaling the pitch vector by a factor of 2. (Or change a female voice to a male voice by scaling the pitch vector by 0.5.)

Cite as: Julie Greenberg. Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

Question 5 *If you have attempted any of these optional parts, briefly describe your results and observations. (If you have not, there is no penalty for skipping this question.)*

3.4 LP analysis of a vowel

1. Using the vowel segment that you selected in Sec. 3.2, compute the linear-prediction coefficients and the gain (**lpcoef**) for a model of order 12. Note that the **lpcoef** function returns the linear predicition coefficients with the initial coefficient of unity and after the sign change, that is, in the form $[1 \quad -\hat{a}_k]$.

2. Compute the linear-prediction spectrum (that is, the frequency response of the all-pole LP model filter) from the gain and coefficients (**freqz**). Plot the magnitude of the LP spectrum in dB vs Hz. Plot the DFT spectrum of the original segment on the same coordinates.

3. Repeat steps 1 and 2 for LP model orders of 8 and 20.

Question 6 *Compare the magnitudes of the LP and DFT spectra, including plots of both. Which spectrum gives a better representation of the formants? Which spectrum gives a better representation of the fundamental frequency? Which model order gives the best representation of the formant frequencies? Please justify your choice.*

4. **Optional**. Using the best model order from step 3, compute the LP error signal by applying the inverse of the LP model filter to the vowel. Note that the error signal is your estimate of the source in the source-filter model. Verify that the LP gain is equal to the square root of the energy in the error signal. Plot the DFT spectrum of the error signal and verify that its spectral envelope is approximately flat.

Question 7 *If you did this optional part, include relevant plots and briefly describe your results and observations. (If you have not, there is no penalty for skipping this question.)*

3.5 LP vocoder

In this section you will design a linear prediction analyzer to encode a speech signal into pitch, LP coefficients, and gain values. Then you will design a linear prediction synthesizer to decode those values and reconstruct an approximation to the original speech signal.

1. Write a Matlab function to create the analyzer (encoder) portion of a LP vocoder. There will be three outputs: a $p \times N$ matrix of LP coefficients, where p is the model order and N is the number of frames; an N dimensional vector of LP gain values; and an N dimensional pitch vector. The function should split the speech signal into 30-ms frames of data that overlap by 15 ms. Then, for each frame, compute the LP coefficients

and the gain of the windowed frame (**lpcoef**), compute the LP error signal by passing the frame of speech through the inverse filter, lowpass filter the error signal, and perform pitch detection on the lowpass-filtered error signal. After all the frames are processed, a median filter (**medfilt1**) should be applied to remove spurious values from the pitch signal. See **/mit/6.555/matlab/vocod/lpvocod_ana.m** for a template of the function.

2. Write a Matlab function to create the synthesizer (decoder) portion a LP vocoder. The synthesis stage takes as input the parameters produced by the analysis stage. As in the analysis stage, these operations are performed frame by frame, at 15-ms frame intervals. The synthesis follows the source-filter model of speech production, first generating a source signal using pitch/voicing information. The Matlab functions **randn** and **pulse_train** can be used to generate the appropriate source signals. The source signal generated for each frame should be normalized so the energy over the frame is unity. (This makes the energy in the source signal the same as that of the LP error signal in the analysis stage.) For each frame, the source signal is then processed by a time-varying filter defined by the LP coefficients and the gain (**filter**). The only tricky part is to keep track of the filter state from one frame to another in order to avoid perceptually-undesirable discontinuities in the synthesized waveform. (Read carefully the help for **filter** and be sure to (i) set the initial filter state to the final state of the previous frame, and (ii) recover the filter state at the end of the new frame.) See **/mit/6.555/matlab/vocod/lpvocod_syn.m** for a template of the function.

3. Use your LP analyzer and synthesizer to encode and then decode the entire speech utterance. Listen to the synthesized utterance and compare it to the original. Also compare the synthetic utterances from the LP vocoder to the utterance synthesized by the channel vocoder.

Question 8 *Based on your listening, describe how the synthesized speech from the channel and LP vocoders compare to each other and to the original speech. Submit your original speech, channel vocoder synthesized speech, and LP synthesized utterances (using the function* **submit_lab2***) so the instructors can listen to them when grading your lab writeup. You may reuse the submit function as many times as you wish, but only the last set of sentences submitted with each vocoder type will be saved.*

Question 9 *Assuming 16-bit quantization, what is the bit rate (in bits/sec) of the original speech? Now assume that each pitch value can be encoded in 8 bits and that each band envelope value requires 12 bits. What is the bit rate of your channel vocoder? Assuming 8 bits for the pitch and 12 bits for each LP coefficient and the gain value, what is the bit rate of your LP vocoder? How do these three bit rates compare? Be sure to state all relevant vocoder parameters (for example, decimation rate, model order) that affect your calculation.*

3.6 Spectrograms

1. Preemphasize the original speech and the two synthesizer outputs (**diff**).

2. Compute the spectrogram (**specgram6555**) of each of the three preemphasized signals. Choose the window length to get a broadband spectrogram (300 Hz resolution).

Question 10 *What window shape/length did you use to compute the spectrogram? How do the three spectrograms compare? Include your spectrogram plots. How do the two different vocoders affect the formants?*

Question 11 *Compare and contrast the basic approaches to speech coding used in the channel vocoder and in the linear prediction vocoder. In what ways are the designs of these two systems similar? In what ways do they differ? (For each system, consider the major components and their functions.)*

Question 12 *What did you find most challenging about this lab exercise?*

Question 13 *What is the most important thing that you learned from this lab exercise? (Suggested length: one sentence)*

Question 14 *What did you like/dislike the most about this lab exercise? (Suggested length: one sentence)*

Be sure to submit your results for both the channel vocoder and the LP vocoder using the function submit_lab2.

Cite as: Julie Greenberg. Course materials for HST.582J / 6.555J / 16.456J, Biomedical Signal and Image Processing, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].