Laboratory Assignment 5
Speech Modeling, Prediction, and Synthesis

PURPOSE

In this laboratory assignment, you will learn how to generate digitally synthesized speech by using a difference equation model for digital speech. Using MATLAB, you can easily implement digital filters defined by difference equations (for real-time speech synthesis, the digital filters are implemented on a dedicated DSP board). Using your model, you will explore the quality of speech you can digitally synthesize and the associated storage and transmission requirements. Using the Fourier transform, you will determine the bandwidth required for speech transmission, explore how speech frequency content changes with time, and compare the spectra of true and synthesized speech.

6.1 OBJECTIVES

By the end of this assignment, you should be able to

Time Domain

1. Use difference equations to model and synthesize speech.
2. Use MATLAB to simulate the time-domain response of difference equation models for DT LTI systems.
3. Compute transmission and storage requirements for speech technologies.

Transform Domain

4. Identify formant frequencies from sampled speech data and compute formant frequencies from linear prediction filter coefficients.
5. Determine the bandwidth and minimum sampling rate required for speech data.
6. Analyze effects of digital filtering using frequency-domain techniques and create inverse filters.
7. Compare spectra and signal to noise ratios (SNR5) for digital speech transmitted by quantized error signals and by quantized speech signals.

6.2 REFERENCES

Review topics

1. Difference equation models for discrete-time (DT) systems
2. Characteristic roots and responses for DT systems

Exploratory Topics

1. Data windowing and windowing functions

2. Frequency response of digital filters and resonant frequencies
3. MATLAB's **filter** command
4. Speech modeling and prediction for transmission and synthesis

Application Reference

1. Rabiner and Schafer, *Digital Processing of Speech* (Englewood Cliffs: Prentice-Hall, 1978)

6.3 LABORATORY PREPARATION

Problems

Time Domain

Question 1. Assume that you have a digitized speech signal that was sampled at 8 kHz. If the speech is broken down into 20 ms blocks, how many samples *NS* are there per block? If one second of a recorded speech signal is in a MATLAB vector, how many 20 ms blocks, *NBLKS*, are there (assuming no overlap of blocks)? What is a one-line MATLAB command that will extract the *n*th 20ms segment of speech, where $n = 1, ..., NBLKS$?

Question 2. Suppose you wish to use as input to your speech model a train of equally spaced DT unit pulses, and would like the pitch to be 200 Hz. If the speech is assumed to be sampled at 8 kHz, how many DT samples are there per period, i.e., what is *N* in

$$x[n] = \sum_{i=0}^{NS-1} \delta[n - iN]$$

Question 3. Look up the description of the MATLAB command **filter**. Determine how you need to define the vectors **a** and **b**, used as input to **filter**, in terms of α_i and *G* to create difference equations that will allow you to perform the following operations:

1. Give you $\hat{s}[n]$ as output when *s[n]* is the input (linear prediction);

2. Give you *e[n]* as output when *s[n]* is input (prediction error);

3. Give you $\tilde{s}[n]$ as output when *e[n]* is input (synthesis).

In each case, how can you ensure that you use the correct initial conditions for each successive speech block, defined as the final values from the previous block when using **filter**. Assume that you are in a loop that generates one 20 ms block at a time.

Question 4. For the signal $x(t) = \sin(\omega t)$, where $\omega = 2\pi(100 + 50t)$, $0 < t < 1$ sec, plot a two-dimensional time vs. frequency representation of the signal. Put time on the horizontal axis and frequency on the vertical axis.

Question 5. Given $e[n] = x[n] - a_1x[n — 1] - a_2x[n — 2]$,

(a) What is the transfer function for the linear prediction filter, where x is the input and e is the output?

(b) What is the transfer function of the inverse filter, where e is the input and x is the output?

(c) Given that $a_1 = 1.3789$ and $a_2 = -0.9506$, what are the formant frequencies in radians per second for the transfer function from part (b)?

(d) For the coefficients given in (c) above, sketch the frequency response. You might wish to read the text corresponding to Figure 10.4.3 in the Background section of Laboratory Assignment 10.

6.4 BACKGROUND

You work for a company that is developing a digital telephone answering machine for home computers. The system will sample data from a telephone line, detect rings, pick up the phone, speak a greeting, and record a message. The greeting will be a text message you type in, and the system will synthesize your speech from samples of your speech which will be recorded and analyzed when the system is installed. In addition, it compresses the recorded phone messages before saving to the hard drive so as to save space.

You are in charge of developing the compression and synthesizer portions of the system. As a first step, you attempt to model your own speech and determine how much compression is possible.

Speech Fundamentals

Physically, CT speech is produced when air from your lungs excites your vocal tract system. Sampling and quantizing CT speech results in digital speech. In telecommunications, speech is digitized by sampling at 8 kHz, using 8 bits per sample. The vocal tract behaves as a resonant cavity so that the signal emanating from your mouth is a weighted sum of delayed versions of the original vocal signal plus the excitations. We can model speech as a linear difference equation; the weights on the delayed signal versions are the coefficients of the model. Different sounds can be produced by using different inputs to and coefficients of this model.

Different types of speech sounds can be roughly categorized as either voiced or unvoiced, where the category is determined by the type of input used to produce the sound. Voiced

sounds are produced by using a periodic sequence of pulses as input; the fundamental period of this sequence determines the resulting pitch. Vowels are voiced sounds; if you say “aah,” you can feel the vibrations at the top of your vocal tract. Unvoiced sounds are produced by using random white noise as input (alone it sounds like static). These sounds generally are produced more by turbulent air flow in the mouth, such as “sh.”

Discrete-Time Speech Models

A mathematical difference equation model for the vocal tract can be developed as follows. Since each successive DT speech sample is very closely related to previous samples, the value of the current speech sample can be estimated as a linear combination of previous samples.

$$\hat{s}[n] = \sum_{i=1}^{p} \alpha_i s[n-i]$$

$\hat{s}[n]$ is the estimate of the speech signal s En] for the nth sample. The error between the estimate and the original signal is

$$e[n] = s[n] - \hat{s}[n]$$

Prediction Model

Combining the two equations above yields a difference equation model of the prediction process for speech:

$$s[n] - \sum_{i=1}^{p} \alpha_i s[n-i] = e[n]$$

This prediction model is used in telecommunications to increase the number of voice signals that can be transmitted over a channel. If the coefficients α_i are known at both the transmitting and receiving ends, then only the error needs to be transmitted and the speech signal can be *reconstructed* at the receiving end using the difference equation above. At the transmitting end *s[n]* is the prediction filter input and *e[n]* is the filter output. It turns out that sending a sampled error signal can result in substantial channel bandwidth savings; this idea is explored further in this laboratory assignment.

Synthesis Model

We can modify this same basic speech prediction model for use in speech synthesis. If our goal is to create a signal $\tilde{s}[n]$ that mimics the original sampled speech segment *s[n]*, then we can replace the error *e[n]* by an input signal *x[n]* multiplied by a gain *G*. Using the same form as the difference equation model for prediction results in the following difference equation model for speech synthesis:

$$\tilde{s}[n] - \sum_{i=1}^{p} \alpha_i \tilde{s}[n-i] = Gx[n]$$

If *Gx[n] = e[n]*, then the synthesized speech $\tilde{s}[n]$ should exactly match the original sampled speech segment *s[n]*; in this case the process is called reconstruction rather than synthesis.

Typically the coefficients ci., change every 10-20 msec as the vocal tract changes to produce different sounds. In synthesis, you apply a sequence of excitations to the model that has coefficients appropriate for that time interval to generate a sequence of sounds corresponding to a speech utterance.

System Characteristic Response and Roots

The characteristic response for a difference equation can be found from the characteristic roots or poles of the system, much in the same way that the characteristic roots for CT systems determine the system behavior. Here our DT development closely parallels that followed in Laboratory 4 for the CT case.

A response of the form z^n, where z is a complex number, can be shown to satisfy a linear, constant-coefficient difference equation with zero input. We can assume that the zero-input response is of this form, and $z^{-N}y[n]$, for $y[n] = z^n$, corresponds to a delayed version of *y[n]*, i.e., *y[n — N]*. Setting the input to zero and replacing each delay in the prediction or synthesis difference equations above by a power of z^{-1} results in the characteristic polynomial. The roots of the characteristic equation, given below, are the characteristic roots, which define the system characteristic response.

$$Q(z) = 1 - \sum_{i=1}^{p} \alpha_i z^{-i} = \prod_{i=0}^{p-1} (z - z_i)$$

Since the equation is order p, there are p characteristic roots z_i. For typical male speech, $p = 10$, and the roots form complex conjugate pairs so that all α_i are real valued.

Resonance and Formant Frequencies

Each complex conjugate pair of characteristic roots defines a damped sinusoidal characteristic response. Consider the root pair $z_i = |z_i| \exp(j \angle z_i)$; this root contributes a time-domain characteristic mode of the form $A|z_i|^n \cos(\angle z_i n + \theta)$. Note that the characteristic root magnitude determines the damping factor, and the phase of the characteristic root determines the frequency of oscillation, i.e. $\Omega = \angle z_i$.

For speech synthesis, these DT sinusoids represent sampled CT sinusoids. In this case, the CT frequency of oscillation is related to the DT frequency by $\Omega = \omega T$, where T is the

sampling rate and ω is the CT frequency in radians per second. The corresponding frequency $f = \omega/2\pi$ Hz is called a formant frequency; it defines a resonant frequency of the vocal tract. Typical male speech (p = 10)is characterized by five formant frequencies. When the coefficients of the model change, different formant frequencies are produced, resulting in different intonations.

Analyzing and Synthesizing Speech

Synthesizing speech using the difference equation model requires that we first analyze a real speech segment to determine the best coefficients α_i for each 20 ms speech segment. Given these coefficients, we can synthesize speech by applying an appropriate input for each model. For voiced speech, a good model of the input source is a train of ideal impulses at a certain frequency (where the frequency determines the pitch). For unvoiced speech, a good model for the input signal is random or "white" noise. These source models rely on the impulse response of the linear prediction filter to generate the output sound.

The analysis procedure is as follows. First, the sampled speech signal is broken into 20-ms long sections using an analysis window. Windowing of data is discussed in a later section. Next, statistical analysis of the data, which determines how correlated adjacent sample values are, is used to determine the filter coefficients that provide the best speech prediction (i.e., that minimize the prediction error power)—see Rabiner and Schafer, *Digital Processing of Speech Signals*. For synthesis, the filter coefficients are used in the model and a periodic impulse train or white noise are used as inputs for each segment, generating synthesized speech.

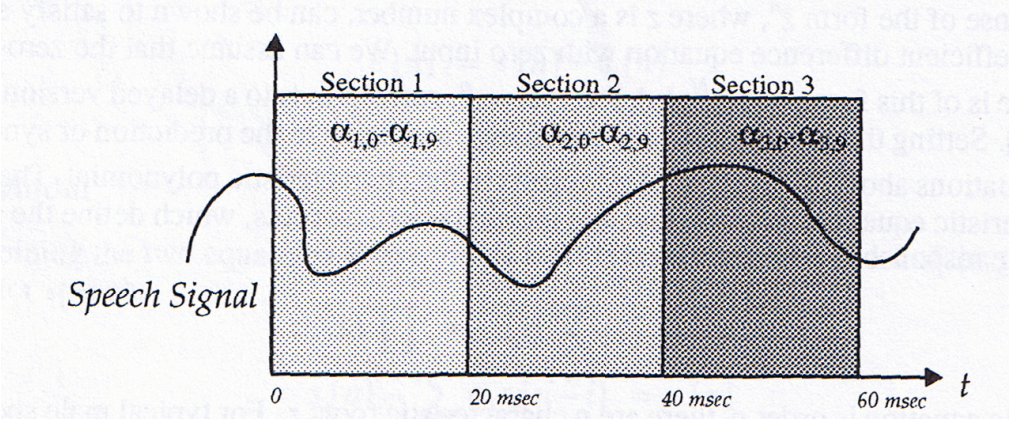

Figure 6.4.1 Analysis Process on a Data Stream

Transmission of Speech

A normal telephone line operates by sampling a person's speech, quantizing the samples to 8 bits, and transmitting these bits to the other end, where they are converted back into speech. An alternative method is to perform analysis and prediction as outlined previously, quantize the error signal, and transmit the resulting digital error signal and linear prediction coefficients.

Why do this? Normal speech requires 8 bits x 8 kHz = 64,000 bits per second to be transmitted. Suppose that the error signal can be quantized with 4 bits instead of 8 and that each coefficient can be represented with 16 bits. Then to transmit an equivalent amount of information, 4 bits x 8 kHz + 16 x 10 coefficients x 100 ten ms chunks per second = 48,000 bits per second - 75% of the previous rate. If only 1 quantization bit is necessary for the error signal, 24,000 bits per second are necessary—37.5% of the previous rate. Using this technique, two people can have conversations in the same space as one person, a tremendous savings.

Both quantizing the sampled speech, as in the first paragraph, and quantizing the prediction error, as above, add distortion to the reconstructed speech. In some of the lab experiments, you will observe the differences between original speech segments and their reconstructed versions resulting from using different numbers of quantization levels. Representing the coefficients with 16 date bits also introduces some quantization error, which can lead to poor quality reproduction on the receiving end. This property is explored in the analysis questions.

In the computerized answering machine, the number of bits to be stored directly relates to the amount of space required. Since the library of sounds necessary to reproduce the answering machine greetings can become very large very quickly, having a good compression technique will allow more message flexibility and require less memory.

Windowing a Data Stream

As discussed above, the α_i coefficients change every 10-20 ms. For every 10 ms block of speech a new set of cc coefficients must be calculated from the sampled speech data. The process of extracting a 10 ms block of speech from the entire segment is called windowing.

The simplest type of windowing involves taking the speech samples in the current 10 ms segment as data. This operation is mathematically equivalent to multiplying the entire signal by a rectangular function having a value of 1 in the region of interest and 0 everywhere else, just as when you multiply a signal by a difference of time-shifted unit step functions. This window function is called a rectangular window. At the edges of the data region, there is a sharp transition from signal to nothing, which can cause problems in analysis.

A better way to window the data sequence is to multiply by a function that has a smooth transition from one end to the other. The most common function that does this is called the Hamming window, which can be calculated using MATLAB for any length by the function **hamming**.

To understand why the Hamming window is preferred to the simpler rectangular window, it is instructive to look at the impact of windowing in the frequency domain. Since windowing a signal is a multiplication operation in the time domain, it corresponds to convolving the Fourier transform of the window function with the frequency spectrum of

the speech segment. If the window transform approximates an impulse in frequency, then this convolution operation yields a frequency spectrum identical to the original speech spectrum. However, the less the window transform is like an impulse, the more windowing distorts the original speech signal spectrum. In Figure 6.4.3, the Fourier transforms of a rectangular window and a Hamming window are compared. Note that the Hamming window has more drop-off before flattening out, called stopband attenuation, but its main lobe is twice as wide as that of a rectangular window.

Figure 6.4.2 shows both Hamming and rectangular window functions.

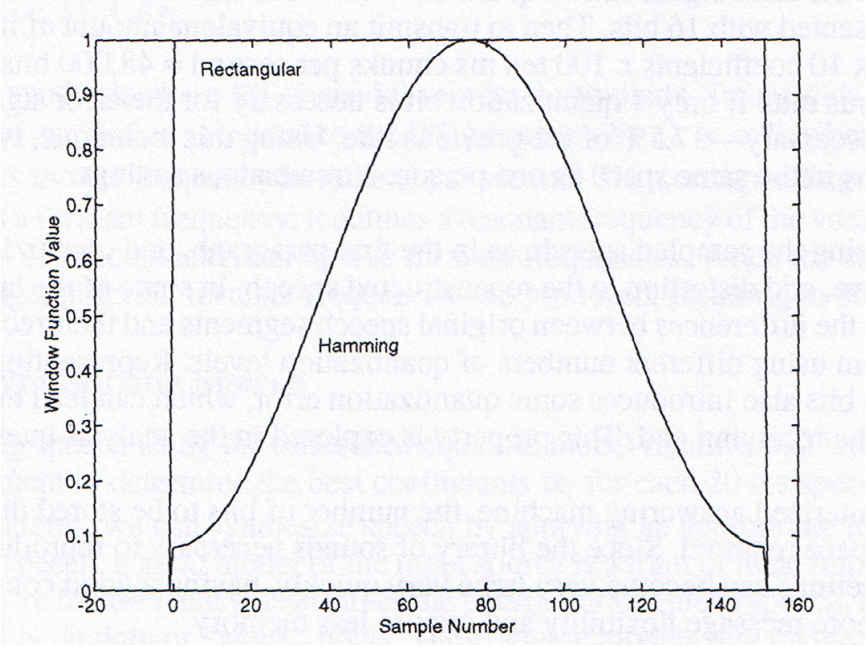

Figure 6.4.2 Hamming and Rectangular Windows

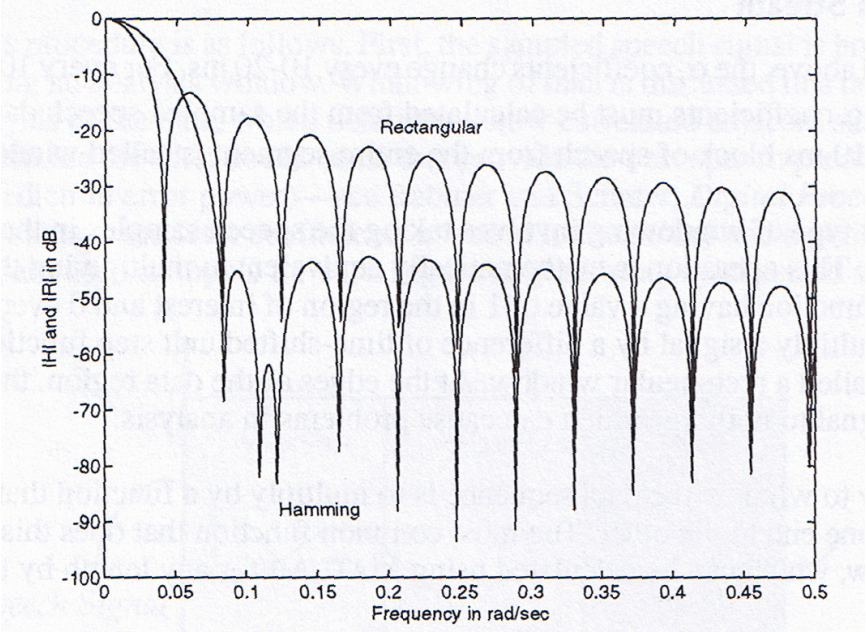

Figure 6.4.3 Window Functions in Frequency Domain

Implementing Difference Equation Models in MATLAB

The **filter** command in MATLAB can be used to compute the response of a difference equation model to specified input signals and initial conditions. Using difference equations to operaate on an input signal is called filtering. The initial conditions of the various delay elements play an important part in the output of a difference equation. For example, computing *y[n]* recursively using the difference equation *y[n] = x[n] + x[n – 1] – y[n — 1]* requires knowledge of *y[n — 1]* at each time. The initial value of *y[n - 1]* is the value of the output prior to application of a new input; different initial values result in different initial responses to the input. Often, when using difference equations for continual filtering of a constantly applied input, the initial conditions are assumed to be zero, as the response to these initial conditions will only affect the filter response at system start-up.

For this experiment, however, the initial conditions are very important. Since speech is a continuous phenomenon but we are breaking it into 20 ms chunks, we would like the filter output at a 20 ms boundary to be consistent with the values from the previous block. Otherwise, there will be "pops" - caused by the errors in the initial conditions - in the output. The last output samples in the previous 20 ms segment should be used as initial conditions for the current 20 ms segment.

6.5 LABORATORY EXPERIMENT

Voiced Speech Models

Your objective is to generate a model for your speech, and then to try synthesizing your speech. As a first pass, you want to try to synthesize a purely voiced segment of speech.

Problem 1. Record yourself saying "We were away a year ago." Sample at 8000 samples/s and store the resulting signal in a MATLAB vector. To save time, you may want to process only a portion of this sentence when you are troubleshooting.

Problem 2. We have provided you with a MATLAB function file **P_6.m**. Given your original sampled speech vector, an excitation vector, and the number of full 20 ms blocks in your speech segment, **P_6.m** generates the prediction error, predicted speech, speech reconstructed using the prediction error as input, and speech synthesized using the excitation vector as input. Information regarding how to use this function is included as comments in the file. Variable definitions can be found by typing help **P_6.m**.

You need to edit **P_6.m** and add in the commands to create the appropriate *a* and *b* vectors, segmented original speech, excitation vectors, and filter command to generate (a) *e[n]*, (b) synthesized speech $\tilde{s}[n]$, and (c) speech synthesized using *x[n]* as input. The use of the filter command is illustrated in the code by generating $\hat{s}[n]$ from *s[n]*. You may wish to review your answers to Question 1 and 3 prior to coding.

Problem 3. Generate a DT signal that is a periodic sequence of DT unit impulses. Pick your period to generate a pitch somewhere in the range of 50-300 Hz. Use your answer to Question 2 for guidance.

Problem 4. Use **P_6.m** to generate synthesized speech using the impulse train. Look at the resulting plots of and listen to the signals generated by **P_6.m**.

How do the synthesized speech predicted speech, and error signal compare to your original sampled speech? Consider both the perceived quality of the sound and the visual similarities and differences for the time- and frequency-domain signal representations. Be sure to identify what parameters you selected, e g pitch. What is the impact of these parameters? What happens if you assume zero initial conditions for each segment?

Improvements on the Voiced Model

Some ideas to consider and possible ways to improve your synthesized speech are described below. Use these suggestions to further explore speech synthesis using difference equations.

Problem 5. Try different pitches for your voiced excitation in Problem 3. You may want to try to estimate a reasonable pitch period from your original speech vector by looking for the significant periodicity present in the time- or frequency-domain plots of each speech segment.

Problem 6. Try synthesizing a different segment of speech, such as "Sally sells sea shells by the sea shore," using the same approach as above. Is it intelligible? Does it retain the same perceptual characteristics as the original speech segment? Pay particular attention to the "sh" sound.

Problem 7. Try using an unvoiced excitation vector as input for the speech segment in Problem 6 by using the MATLAB command **randn** to generate white Gaussian noise having zero mean and unit variance (the default for **randn**). How does using this input instead of periodic impulses impact the perceived quality of the synthesized speech segments for the sentences from Problems 1 and 6?

Problem 8. Using what you've learned in Problems 5 through 7, try creating an excitation vector which uses both unvoiced and variable-pitch voiced inputs to create more realistic-sounding speech. Try to do this for an arbitrary speech segment, as well as for the sentences above. You may want to analyze *e[n]* to determine whether voiced or unvoiced excitation is appropriate for synthesizing a given segment of speech. For segments where *e[n]* looks more random, use an unvoiced excitation, and where it looks more periodic, use a voiced excitation.

Compare and contrast the linear prediction error signal with the excitation use for synthesis each of the cases above. Consider the impact of any differences on the intelligibility of the synthesized speech your ability to recognize the speaker and the overall quality of the reproduction for different types of sounds in the speech segments.

Transmission of Speech

Problem 9. Obtain an error signal for a speech sequence using **P_6.m**. Normalize the error signal so that the maximum value is 1.0 and the minimum value is -1.0. Quantize the resulting error signal using 5 bits by using **P_6_9.m**. Plot the normalized error signal and the quantized error signal at the same time, and note any differences. Also plot the quantized error signal subtracted from the normalized error signal. What is the maximum value of the difference? Repeat for 4 bits, 3 bits, 2 bits, and 1 bit. How does the number of quantization levels affect the accuracy of your quantized representation of *e[n]*?

Problem 10. Resynthesize the speech waveform using quantized error signals for the El input. Try using 4 bits, 2 bits, and 1 bit for quantization. Listen to the sampled speech and resynthesized speech, and comment on the differences and the relative rate of improvement observed as you increase the number of bits. Remember to undo your normalization of the error signal so that the volume of the synthesized speech will be the same.

Problem 11. Calculate how many bits per second are needed to transmit the speech signal using 5-bit, 4-bit, 3-bit, 2-bit, and 1-bit quantization of the error signal and 16-bit quantization for each filter coefficient. If you want to satisfy the conflicting goals of minimizing the bit rate and maximizing the quality of reproduction, which quantization level (using your results from Problem 10) provides the best trade-off?

Compare the bit rate for sampled speech with that of a system which transmits only the quantized prediction error. Try to quantify the bit rate vs. quality trade-offs and suggest potential applications.

Frequency Domain Analysis of Speech

Problem 12. Record yourself saying “Ick Ack” at 44.1 kHz. Perform a Fourier transform of the first 8192 sampled data points using **fft**. and plot the magnitude of the frequency spectrum. Find the highest frequency represented, and determine the maximum sampling rate needed to accurately represent the data.

How much information is lost if the sampling rate is 8 kHz, the telephone standard?

Problem 13. Speech is the output of a time-varying system with time-varying inputs. You are to perform a time-frequency analysis of your sampled speech using the MATLAB function **specgram**. From your time-frequency plot, identify the three major formants in your speech segment and track how they change over time.

Problem 14. Obtain a set of coefficients for one speech segment. Find the frequency response of the synthesis filter for these coefficients, and compare it to the frequency spectrum of the speech data for that segment. What similarities and differences do you notice?

How well does the linear prediction filter model the frequency content of the speech?

Analysis Questions

Problem 15. Coefficient Variation Between Blocks: Determine the model coefficients, characteristic roots, and formant frequencies for some neighboring speech segments and for some different sounds. Supply a table of these values below and attach your calculations (which can be done via MATLAB). How much variation do you observe in these from segment to segment and for different sounds?

Problem 16. Finite-Precision Effects: For the segments you used in Problem 15, what can you say about the stability of the difference equation for each segment? Now try reducing the number of decimal places you use to represent each coefficient. How do the characteristic roots, formant frequencies, and system stability of each segment change, if at all? What are the implications for digital storage of the model coefficients?

Problem 17. Storage and Compression Implications: For the original voiced speech segment that you recorded, determine the storage savings achievable by storing only the coefficients, rather than the sampled speech segment. Be sure to use as many bits per coefficient as needed to ensure that the model is not compromised. How does your answer change if the speech was digitized as high quality audio (44.1 kHz sampling rate and 16 bits per sample)?