1 Lecture Outline

Reading: Chapter 12

- Overview of a simple vocoder (LPC-based codec)
- Clipping autocorrelation pitch detector
- Encoder design
- Decoder design

2 Overview of a Simple Vocoder (LPC-based Codec)

In our next project, we will construct a vocoder based on linear predictive coefficients (LPCs). We begin with a review of the DT speech production model as in Fig. 1. The model assumes a pulse train or noise excitation signal for voiced or unvoiced speech. The excitation signal is then fed to the all-pole model, $H(z)$ resulting in a speech output. Coefficients of the all-pole model are obtained through short-time LPA. As discussed, the glottal glow, $G(z)$ and lips radiation model, $R(z)$ are lumped into $H(z)$ in the LPA.

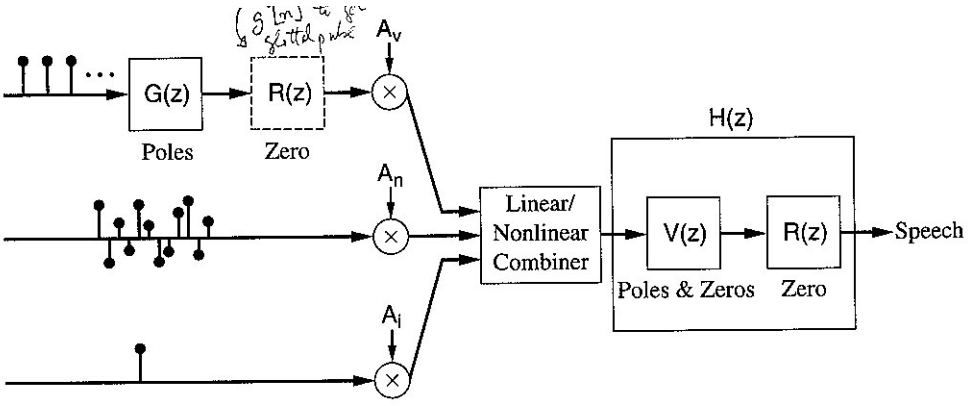

Figure 4.20 Overview of the complete discrete-time speech production model.

Figure 1: (Figure 4.20) DT speech production model

From the speech production model, it is clear the parameters we must encode: 1) (V)oiced or (U)nvoiced or (S)ilence; 2) if (V)oiced, also code the pitch period; 3) all-pole model coefficients $[\alpha_1, \cdots, \alpha_p]$ and gain A. We will use an autocorrelation-based method to estimate the fundamental (pitch) and Levinson to determine the all-pole model coefficients.

There are $p+1$ parameters (plus a few bits for V/U/S/PP) for every analysis frame (typically 20-40ms with 50% overlap). Assuming $p = 16$, a 30ms window with 15ms frame rate (50% overlap), and 8 bits/parameter, we have a data rate in the neighborhood of 8kbps.

We next investigate how to determine the excitation signal parameters. These parameters are used to construct the input to the all-model at the decoder end.

3 Clipping Autocorrelation Pitch Detector

(This section taken from Section 4.8 of *Digital Processing of Speech Signals* by L. Rabiner and R. Schafer).

We begin with a block diagram of the clipping autocorrelation pitch detector as in Fig. 2. The purpose is to determine whether the speech is voiced, unvoiced, or silence and if voiced, what the pitch period is. As we have seen, voiced speech has a certain harmonic structure. One key feature is the fundamental frequency, f_0 or the pitch period. There exist many ways to determine the pitch period including counting the number of zero crossing per unit time and dividing. We will explore a slightly more robust method using the autocorrelation function.

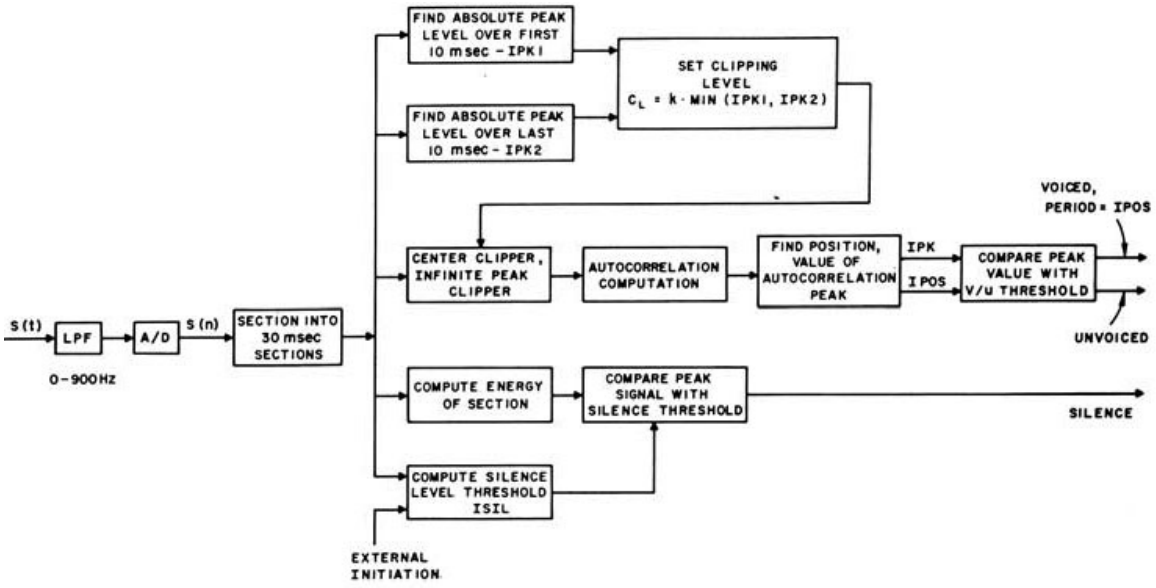

Figure 2: [Figure 4.36 (Rabiner and Schafer)] clipping autocorrelation pitch detector

The autocorrelation of a signal $s[n]$ with lag m is defined as

$$r_s[m] \equiv E\{s[n]\, s[n+m]\}. \tag{1}$$

It is easy to show that if $s[n]$ is periodic with a period P, then the autocorrelation is also periodic with the same period

$$r[m] = r[m+P] \tag{2}$$

and attains a maximum at $m = 0, \pm P, \pm 2P, \ldots$. That is regardless of the time origin of the signal, the period can be estimated by finding the location of the first maximum in the autocorrelation function. Thus the autocorrelation function provides a convenient method of estimating periodicities in speech.

As it turns out, the major limitation of the autocorrelation representation is that in a sense, it retains too much of the information in the speech signal. Specifically, for speech the autocorrelation function may have too many peaks, many of which can be attributed to the damped oscillations of the vocal tract response as opposed to the underlying periodic glottal flow.

To avoid this problem, it is useful to preprocess the speech signal so as to make the periodicity more prominent while suppressing other distracting features of the signal. Generically these methods are called "spectral flatteners" since their objective is to remove the effects of the vocal tract transfer function, thereby bringing each harmonic to the same amplitude level as in the case of a periodic impulse train. We will focus on one particular method namely, the *center clipper*.

The center clipped speech signal proposed by Sondhi, is obtained by a nonlinear transformation, $y[n] = C\{x[n]\}$ shown in Fig. 3. The operation of the center clipper is shown in Fig. 4. For samples above/below $+C_L/-C_L$, the center clipper outputs these same samples minus the clipping level. Examples of center clipped speech are given in Fig. 5.

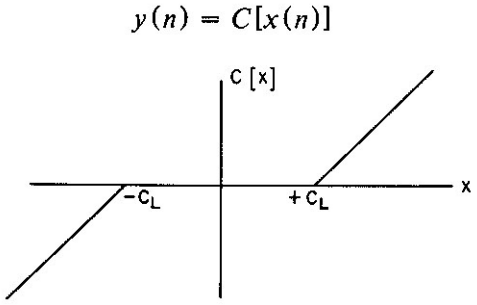

Fig. 4.31 Center clipping function.

Figure 3: [Figure 4.31 (Rabiner and Schafer)] center clipping function

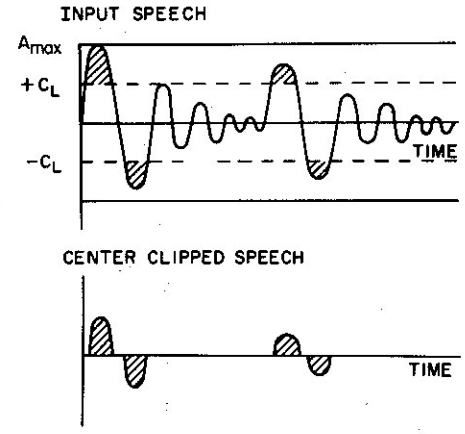

Fig. 4.32 An example showing how center clipping affects a speech waveform. (After Sondhi [17].)

Figure 4: [Figure 4.32 (Rabiner and Schafer)] speech before and after center clipping

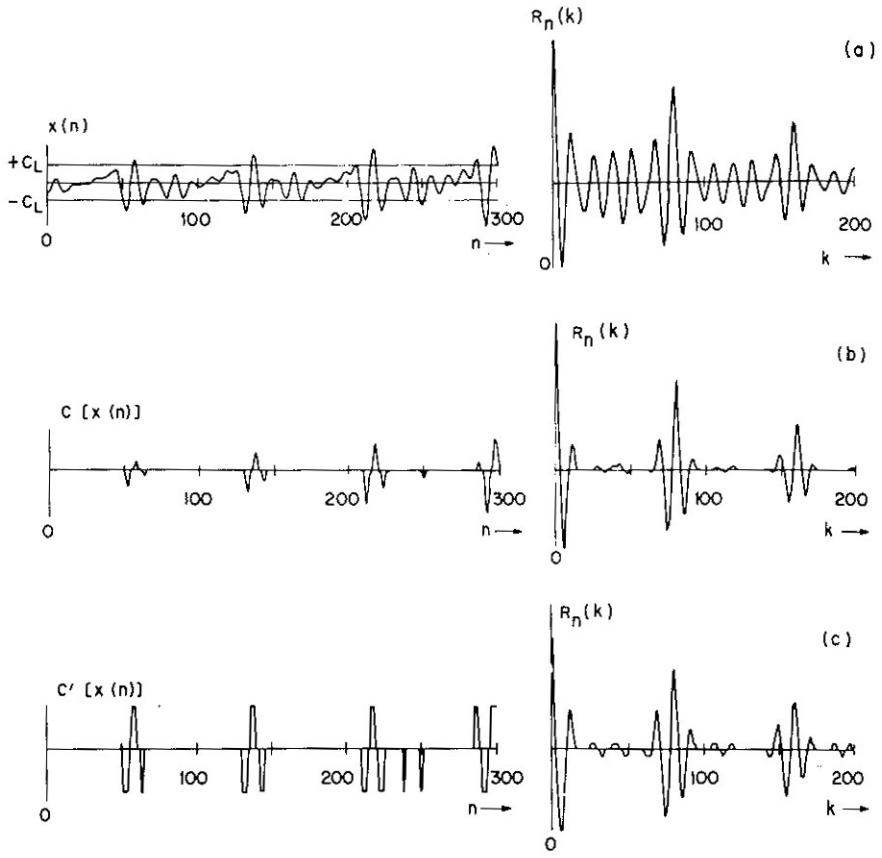

Fig. 4.33 Example of waveforms and correlation function; (a) no clipping; (b) center clipped; (c) 3-level center clipped. (All correlation functions normalized to 1.0.) (After Rabiner [18].)

Figure 5: [Figure 4.33 (Rabiner and Schafer)] center clipped speech

Now, if we compute the autocorrelation function on center clipped speech, it has considerably fewer extraneous peaks thus aiding in the voiced/unvoiced decision as well as pitch period estimation. Examples of the autocorrelation function of clipped speech are found in Fig. 6. The clearest indication of periodicity is when the clipping level is set high. However, if there is significant variance in the amplitude of the speech signal, much of the speech signal may fall below the clipping level and be lost. Therefore, we normally set C_L to 30% of the maximum signal amplitude.

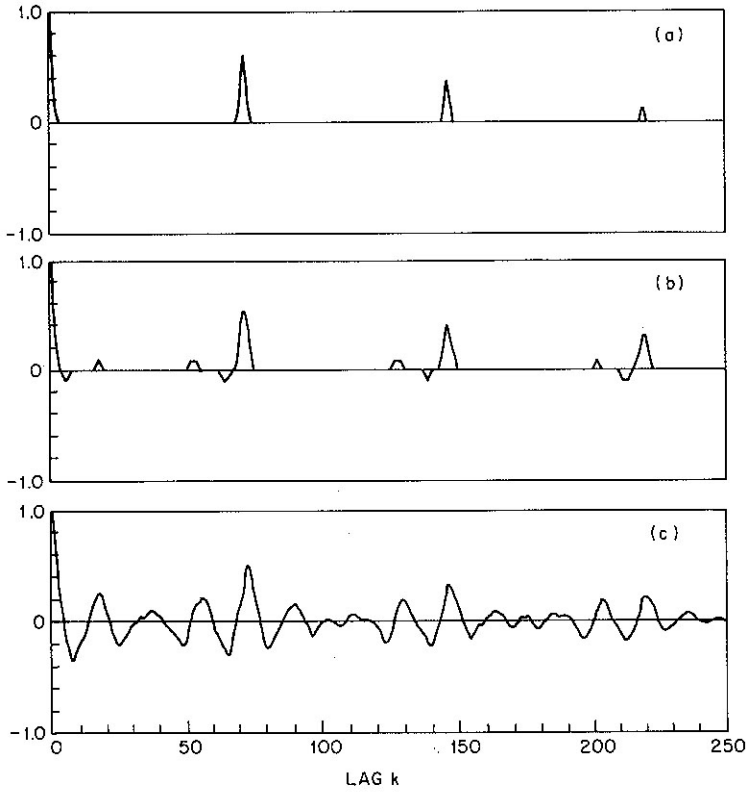

Fig. 4.34 Autocorrelation functions of center clipped speech using $N = 401$; (a) C_L set at 80% of maximum; (b) 64%; (c) 48%. (Speech segment same as for Fig. 4.26a.)

Figure 6: [Figure 4.34 (Rabiner and Schafer)] autocorrelation function of center clipped speech

We can significantly reduce the computational complexity associated with the center clipper with no appreciable degradation in utility for pitch detection. The modification is simple:

$$y[n] = \begin{cases} +1, & x[n] > +C_L \\ -1, & x[n] < -C_L \end{cases} \tag{3}$$

This modified center clipper is referred to as the *3-level center clipper* shown in Fig. 7.

There are a few remaining components of the clipping autocorrelation pitch detector to discuss.

Clipping Levels

One method for using higher clipping levels (clearer indication of periodicity) is to determine the maximum amplitudes over the first and last thirds of speech frame. Then the clipping level for each frame is set to

$$C_L \quad = \quad k \min\left(\text{IPK1}, \text{IPK2}\right) \tag{4}$$

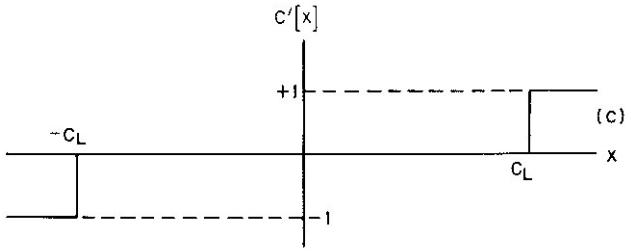

Fig. 4.35 3-level center clipping function.

Figure 7: [Figure 4.35 (Rabiner and Schafer)] 3-level center clipper

where k is the percentage of the peak level to use and IPK1, IPK2 are the maximum amplitude levels for the first and last one-third segments of the speech frame. We typically set $k = 60\% - 80\%$.

Voiced/Unvoiced Thresholds

If there are strong peaks in the autocorrelation function of the center clipped speech frame, then we consider the frame to contain (V)oiced speech and the location of the first "big" peak to determine the pitch period. On the other hand, if there are no strong peaks in the autocorrelation function of the center clipped speech frame, then we consider the frame to contain either (U)nvoiced speech or silence. Unfortunately, making this decision with the autocorrelation function may not be so clear cut. Thus we introduce a V/U threshold which we will compare to the peak in the autocorrelation function. If the peak exceeds the threshold, we will assumed voiced speech otherwise we will assume either unvoiced speech or silence. We typically set the V/U threshold to 30% of the signal variance, $r[0] \approx \frac{1}{N}\sum_{n=0}^{N-1} s^2[n]$ or energy in the frame.

Silence Thresholds

Since silence is a natural part of speech, we must make provisions for its detection. First, we estimate the background noise level by examining the first 50ms of speech and noting the peak amplitude value or silence threshold, ISIL. It is assumed the initial 50ms of signal does not contain speech. (Some variations first low-pass filter the 50ms of signal). Then for the first 100 samples of each frame, we compare the peak amplitude value with ISIL. If the peak amplitude value is less than ISIL we assume the signal is composed of silence, otherwise we assume either voiced or unvoiced speech.

4 A Linear Predictive enCoder (LPC)

Regardless of whether the speech is classified as voiced or unvoiced, we must estimate the parameters of the all-pole filter in order to filter an excitation signal (impulse train for voiced or random noise for unvoiced).

We have already investigated in Chapter 5, the procedure for this estimation and developed the Levinson algorithm. Thus in addition to using the clipping autocorrelation pitch detector to determine if the speech frame is voiced, unvoiced, or silence and if voiced what the pitch period is, we must have a parallel block which determines the coefficients of the all-pole model. Obviously, if we determine the frame is silence, we do not transmit the coefficients. A block diagram of the encoder is given in Fig. 8.

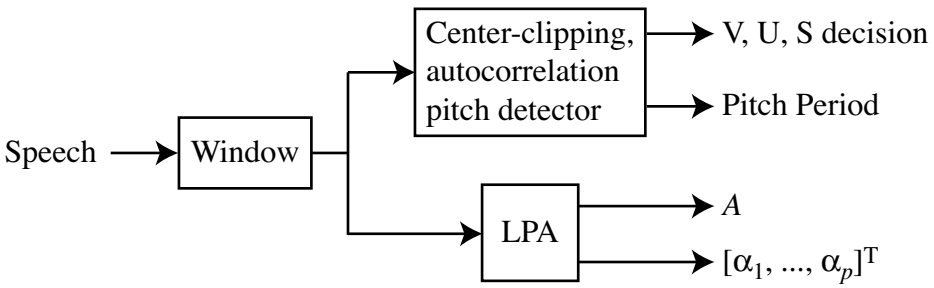

Figure 8: Block diagram of LPC-based speech encoder

Although the feedback coefficients, α_k define the all-pole model,

$$H(z) = \frac{A}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{5}$$

the poles, p_k also define the all-pole model

$$H(z) = \frac{A}{\prod_{k=1}^{p} (1 - p_k z^{-k})} \tag{6}$$

as do the PARCOR or lattice filter coefficients, k_i. Since $H(z)$ is causal and stable, we know $|p_k| < 1$ and $|k_i| < 1$ but we cannot say anything regarding the numerical range of α_k. Thus for purposes of quantizing the parameters of the all-pole model, we normally quantize (and transmit) either the poles or the PARCORs (usually PARCORs since these are a by-product of Levinson). Note that in MATLAB, "latcfilt" filters a signal through the lattice defined by the reflection parameters.

Figure 9: [Figure 5.17] All-pole lattice using PARCOR coefs

We might organize the compressed speech file as a series of data frames (each data frame represents the coded speech frame). A (variable length) data frame could be organized as follows:

1. Voiced/unvoiced/silence decision (1 bit to represent V or U, silence represented with $A = 0$)
2. Pitch (if voiced) (6 bits, uniform)
3. Gain, A (5 bits, non-uniform)
4. PARCORS k_i (12 PARCORS @ 8 bits/PARCOR)

Given a fixed number of bits per frame, we must allocate these bits (quantize) in a way that carefully represents the parameters so as to recreate the speech signal with high quality. With the above parameters and assuming 67 data frames/s with $1 + 6 + 5 + 96$ bits/data frame results in 7.2 kbps. Compare this to unencoded speech–8,000 samples/s with 8 bits/sample which results in 64 kbps. The coder results in a factor of 9× compression.

5 Decoder Design

The objective of the decoder is to recreate the speech frame given a data frame composed of the parameters in the above list. We use the speech production model as a guide to this three-step procedure.

Step 1: Generate an excitation signal. If the signal is unvoiced, the excitation signal is white noise. If the signal is voiced, the excitation signal is an impulse train with the period equal to the estimated pitch period. Excitation signal length must be the same as the speech frame length.

Step 2: Filter the excitation signal through the IIR filter defined by parameters of the all-pole model.

Step 3: Overlap (as defined in the encoder) and add the filter outputs to rebuild the speech signal.

A block diagram of the decoder is given in Fig. 10

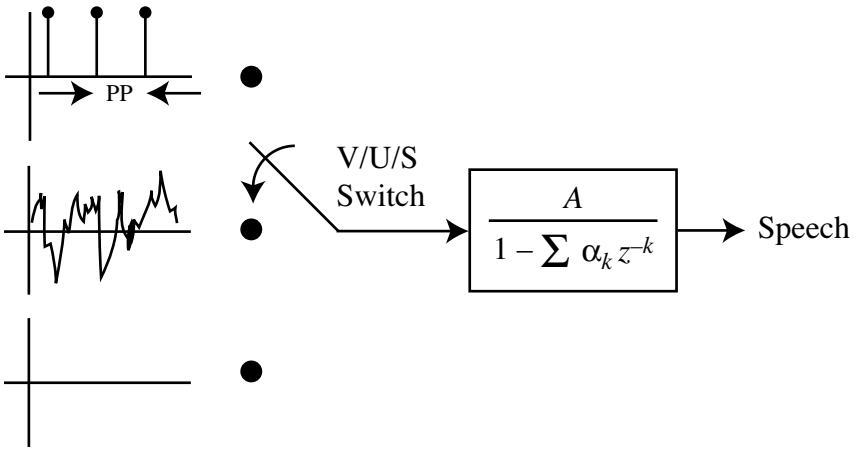

Figure 10: Block diagram of LPC-based speech decoder