Lab09: Speech Coding

EECS20: Introduction to Real-Time Digital Systems

By K. H. Chiang, William T. Huang, Brian L. Evans.
URL: `http://www-inst.eecs.berkeley.edu/~ee20`
News: `ucb.class.ee20`

Assigned: 09 Apr 97, Checkoff: 16 Apr 97, Writeup Due: 18 Apr 97

1 Introduction

Five seconds of speech, quantized into eight bits and sampled at 8 kHz, requires 40 KB of memory, if stored. Suppose we wish to reduce the amount of storage space, at the expense of the quality of the speech.

Alternately, consider a situation in which we would like to transmit this speech over a communication channel, but the maximum bit rate permitted through the channel does not allow us to send the original speech signal.

One way to get around these problems is to use speech coding, as diagrammed in Figure 1. First the speech is divided into segments, or groups of samples. Assuming that each segment contains S samples, a technique called linear predictive coding (LPC) can be used to reduce the S samples in each segment to N coefficients, in effect compressing the speech by a factor of $\frac{S}{N}$.

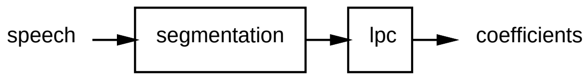

Figure 1: Largely content-free top level diagram.

2 Segmentation

Segmentation is easily accomplished by extracting groups of S samples from the vector containing the original speech sample. 50% overlapping of successive segments is suggested, as in Figure 2.

3 Linear Predictive Coding (LPC)

The heart of the coding process is in the determination of the N coefficients for each segment. If we assume that the nth sample in a given speech segment can be approximated by an appropriately scaled sum of the previous kth samples, the Matlab routine `lpc` will give us the N coefficients that we're looking for.

Unfortunately, the description of the LPC algorithm is beyond the scope of this course. Suffice to say that if we constructed a DT system with these coefficients and gave the right input to

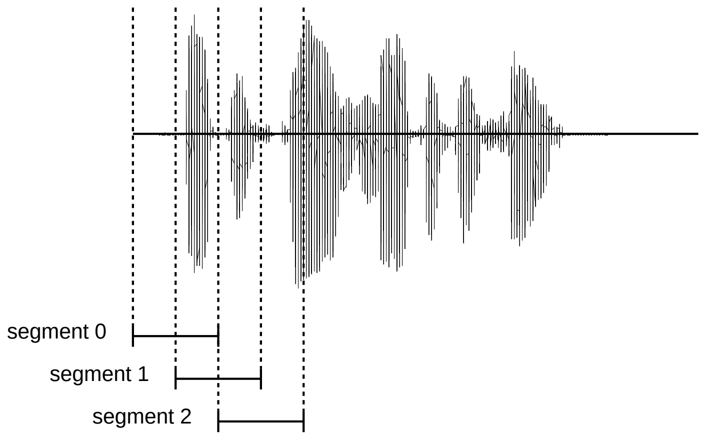

Figure 2: Segmentation with 50% overlapping.

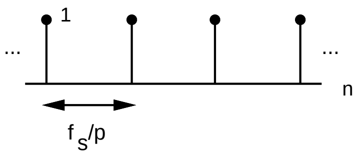

Figure 3: An impulse train with period $\frac{f_s}{p}$.

the system, we would obtain a sequence of samples that would closely approximate the original sequence of samples.

Using the notation of the `filter` command, the `b` vector would be `[1]` and the `a` vector would be the coefficients returned by the `lpc` command.

4 Excitation

What's the right input though? It turns out that for voiced speech, when the vocal cords are forced to vibrate, a train of impulses as in Figure 3 is the correct input. If the impulse train is supposed to be at pitch p,[1] the spacing between impulses is the ratio of the sampling rate and the pitch: $\frac{f_s}{p}$. All other samples in the train are zeros.

5 Questions

1. Record a speech signal and remove the zeros at the beginning and end of the sample [the LPC routine fails on zero inputs].

2. Segment the speech sample into 10 ms lengths with 50% overlapping. For the S samples in each segment, use the `LPC` command with an order of 9 [$N = 10$] to get the coefficients of

[1]Pitch is the perceived frequency of a sound. Pitch and frequency do not necessarily correspond; in the case of a sound consisting of the sum of 2 kHz, 3 kHz, 4 kHz, and 5 kHz sinusoids, the listener "hears" a 1 kHz tone.

the corresponding DT system for that segment.

3. For each set of LPC coefficients, construct the corresponding DT system, using the `filter` command and an impulse train of pitch 100 Hz as input. Since the output of the `filter` command is a DT signal with as many samples as the input, make sure that your impulse train is of the correct length. Sum the resulting filter outputs appropriately to get the reconstructed speech and listen to the results.

4. Vary the pitch of the impulse train and listen to the results.

5. Quantize the LPC coefficients and listen to the reconstructed speech. At what level of quantization is the speech still intelligible?