MSc Laboratory Experiment

Title: **Speech Signal Processing**

Aims: To study the method of linear prediction and to investigate its application to speech signals.

Equipment: This experiment requires access to a PC/workstation running MATLAB.

Introduction: This experiment is based on a section of the book by Burrus et al [1] beginning at page 352. You may find it helpful to read from this book and from the relevant chapters of [3] before beginning the practical work. It is important that you are able to understand the theoretical aspects of this work as well as perform the practical tasks.

Files: The data file you will work with is s5.mat. Please check with the lab staff how to access this file or download it by clicking here.

References:

[1] Burrus, C.S., et. al., "Computer-based exercises for signal processing using MATLAB", Prentice Hall, 1994

[2] Makhoul, J., "Linear prediction: a tutorial review", Proc. IEEE, Vol 63, No. 4, pp 561-580, April 1975.

[3] Rabiner, L.R. and Schafer, R.W., "Digital processing of speech signals", Prentice Hall, 1978.

Linear Prediction

This document is based on [1].

Overview

There is an excellent overview of linear prediction in the paper by Makhoul [2]. Another very good source of information is Rabiner and Schafer's textbook [3].

Given a signal $x(n)$ we wish to predict future values of $x(n)$ from the previous samples. We can therefore define a linear (FIR) predictor as

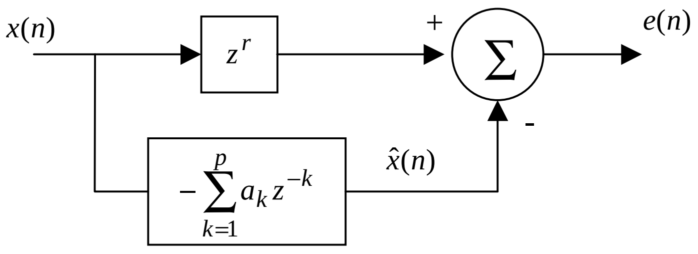
$$\hat{x}(n) = -\sum_{k=1}^{p} a_k x(n-k) \tag{1}$$

The coefficients a_k of the predictor can be designed by minimising, over a range of samples n, the sum of the squared difference between the true samples of $x(n)$ and our estimate $\hat{x}(n)$. The function to minimise can be written

$$E = \sum_n \left(x(n+r) - \hat{x}(n)\right)^2 . \tag{2}$$

The choice of the range n leads to two different methods: the autocorrelation method and the covariance method.

Figure 1 - Linear Predictor
If r=0 then the predictor predicts the current sample, if r>0, then predictor predicts a future sample.



The solutions for a_k can be found by setting partial derivatives of (2) to zero which leads to the normal equations [3]

$$\begin{array}{rcll}
-x(1+r) & = & a_1 x(0) + a_2 x(-1) + \cdots + a_p x(1-p) & (n=1) \\
\vdots & & & \\
-x(p+r) & = & a_1 x(p-1) + a_1 x(p-2) + \cdots + a_p x(0) & (n=p) \\
\vdots & & & \\
-x(L-1) & = & a_1 x(L-2-r) + \cdots + a_p x(L-1-r-p) & (n=L-1-r) \\
\vdots & & & \\
-x(L-1+p+r) & = & 0 + \cdots + a_p x(L-1) & (n=L-1+p)
\end{array}$$

which can be written in matrix form

$$-\mathbf{x} = \mathbf{Xa} . \tag{3}$$

Such systems of simultaneous equations can be solved by matrix inversion using the MATLAB \ operator.

When r=0 there are two methods of linear prediction of interest.

a) In the autocorrelation method all possible equations for n=1 to L-1+p are included. Thus if the extent of the input data $x(n)$ is $0 \le n < L$, the prediction distance is r, and the length of the predictor is p, there will be L-1+p equations. Note that $x(L) = x(L+1) = \cdots = x(L-1+p+r) = 0$.

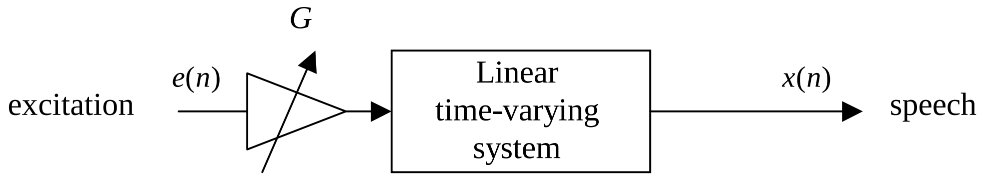
b) The covariance method includes only those equations for which all values of $x(n)$ needed on both sides are present in the data. This method uses fewer equations but does not predict past the end of the data.

For long input sequences there should be little or no difference in the solutions from the two methods.

Linear Prediction of Speech

The subject of linear prediction of speech is discussed in detail in [3] chapters 1, 2, 3, and 8. In summary, speech is modelled as the output of an all-pole linear system $H(z)$ which is stationary of short time intervals.

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{4}$$



Linear prediction is a technique to finding the set of predicition coefficients a_k that minimize the mean-squared prediction error between the signal $x(n)$ and a predicted signal based on a linear combination of past samples. It can be seen in [3] that the autocorrelation method of linear prediction yields predictor coefficients which satisfy

$$\mathbf{Ra} = -\mathbf{r} \tag{5}$$

where **R** is a p x p Toeplitz matrix of the autocorrelation sequence of $x(n)$, **a** is a p x 1 vector of prediction coefficients and **r** is a p x 1 vector of autocorrelation values.

When applying linear prediction to speech, it is usual to make the assumption that the predictor coefficients a_k are equal to the parameters α_k of the speech model. Then it can be seen that the output of the prediction error filter with system function

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{6}$$

is

$$f(n) = x(n) + \sum_{k=1}^{p} a_k\, x(n-k) \quad \equiv \quad G\, e(n) \qquad (7)$$

i.e. the excitation of the model is defined to be the input that produces the given output $x(n)$ for the prediction coefficients estimated from $x(n)$. The gain constant G is the constant that is required so that $e(n)$ has unit mean-squared value and it can be found from the autocorrelation values used in computation of the prediction coefficients [3].

An efficient method called the Levinson recursion [3] can be applied to solve simultaneous equations of the form of (5) when **R** is Toeplitz.

Exercise 1

Write a MATLAB function which performs autocorrelation LPC having the following interface:

```
function [A, G, r, a] = autolpc(x, p)
%AUTOLPC   Autocorrelation Method for LPC
%    Usage: [A, G, r, a] = autolpc(x, p)
%    x : input samples
%    p : order of LPC
%    A : prediction error filter, (A = [1; -a])
%    G : rms prediction error
%    r : autocorrelation coefficients
%    a : predictor coefficients
```

Exercise 2

This exercise uses the speech data file `S5.MAT` which contains male speech of the sentence "Oak is strong and also gives shade". The data can be replayed on a PC using the MATLAB command `soundsc`.

Apply your LPC analysis on this data file using 12th-order prediction using a Hamming window of length 320 samples for two different phonemes: SH as in *shade* and AA as in *shade*. For both phonemes, make a plot of the frequency responses of the prediction error filter and the log magnitude responses of the vocal tract model filter both on the same graph. Also use `zplane()` to plot the zeros of the prediction error filter for both cases. Keep these plots and data as you will need them for later exercises.

What do you observe about the relationship between the zeros of the prediction filter and the poles of the vocal tract model? What do you observe about the relationship between the zeros of the prediction filter and the peaks in the frequency response of the vocal tract model? What do you observe about the relationship between the zeros of the prediction filter and the dips in the frequency response of the prediciton error filter?

Exercise 3

For both of the two phonemes, compute the DFT of the windowed segment of speech and plot is magnitude (in dB) on the same graph as the vocal tract model. Try to adjust the gain parameter *G* to get the plots to line up for convenience.

State your observations about the similarities and differences between the phonemes AA and SH. What do we mean by voiced and unvoiced sounds in this context?

Exercise 4

Repeat the above tests on other phonemes from the same file and make comparisons. Interesting phomenes to look at are in *strong* and *gives* but feel free to experiment as you wish.

Exercise 5

Investigate the effect of varying the model order on the magnitude spectral plots. It is only necessary to do this for one or two phonemes. Consider $p = 8, 10, 12, 20$.

Exercise 6

It is common to apply a preemphasis filter to speech before using linear prediction analysis. Such filters can be implemented in MATLAB using

```
y = filter([1, -0.98], 1, s5);
```

What type of filter is this and why is preemphasis used (see [3])?

Repeat some of your previous test using preemphasis and compare the results with and without the filtering.