University of Maryland at College Park
Department of Electrical and Computer Engineering

ENEE 624 Advanced Digital Signal Processing

Project 2
Spring 2003

Issued: Wednesday, April 16, 2003 **Due:** Monday, May 12, 2003

Problem 2.1 Implementation of the Levinson-Durbin Algorithm

Write a Matlab program that implements the Levinson-Durbin recursion. Your program should take as inputs a vector of autocorrelation values

$$\mathbf{r_x} \triangleq [r_x[0] \;\; r_x[1] \;\; \cdots \;\; r_x[N]]^T \;,$$

and an integer p (such that $N \geq p \geq 0$). Your program should return as outputs

(i) the optimal all-pole parameter vector

$$\mathbf{a}_p \triangleq [a_p[0] \;\; a_p[1] \;\; \cdots \;\; a_p[p]]^T \;,$$

(where $a_p[0] = 1$),

(ii) $b[0]$,

(iii) a vector of squared errors

$$\mathbf{e} \triangleq [\varepsilon_0 \;\; \varepsilon_1 \;\; \cdots \;\; \varepsilon_p]^T \;,$$

(iv) and a vector of reflection coefficients

$$\mathbf{g} \triangleq [\Gamma_1 \;\; \Gamma_2 \;\; \cdots \;\; \Gamma_p]^T \;.$$

Problem 2.2 Debugging Your Algorithm

In this part your task is to verify that program you wrote in Problem 2.1 works correctly. Let $x[n]$ denote a wide-sense stationary (WSS) process whose spectrum is given by

$$P_x[k] = 4\,Q(z)\,Q^*(1/z^*)\ ,$$

where

$$Q(z) = \frac{1}{(1-0.5\,z^{-1})\,(1+0.25\,z^{-1})}\ .$$

Verify that your program indeed correctly identifies the all-pole model characterizing this system.

Problem 2.3 Signal Modeling

Consider the random process $x_{\mathrm{A}}[n]$ generated by the Matlab program `BlackBoxA`. We would like to model this process as the response of a zero-mean white WSS unit-variance process through an all-pole filter. Note that any time you run the program `BlackBoxA` you get a different sample path of the process $x_{\mathrm{A}}[n]$. Based on observation of sample paths from `BlackBoxA` develop an all-pole model for the process. Clearly indicate (and justify) any assumptions you make in the process.

Problem 2.4 Stochastic Modeling and Lossy Compression

The objective of this part of the project is to develop effective models for lossy compression of the real-valued signals that are outputs of the Matlab program `BlackBoxB`. Every time you run the Matlab program `BlackBoxB` you get a different sample path of a random process $x_{\mathrm{B}}[n]$. You are allowed to generate as many sample paths as you wish, in order to obtain an understanding of the types of models that may be appropriate for this process. Note that you should justify all the assumptions you may make in the sequel, such as whether the process is zero-mean, wide-sense stationary, etc.

Once you obtain what you deem as an accurate model (or class of models) for the process $x_{\mathrm{B}}[n]$, you may use that to develop algorithms for compression of the individual sample paths. In each of the following cases you should provide: (i) an encoder program, which, given as input a sample path of length $N = 1000$, generates as output an encoded description; (ii) a decoder program, which, given as input the description generated by the encoder, generates an approximation signal $\hat{x}_B[n]$.

(a) Design an encoder/decoder pair that generates signal approximations to the sample paths of $x_{\mathrm{B}}[n]$ that satisfy the quality criterion:

$$\frac{\sum_{n=1}^{N} |\hat{x}_B[n] - x_B[n]|^2}{\sum_{n=1}^{N} |x_B[n]|^2} \leq 0.05;$$

while minimizing the average number of bits used by your encoded description.

(b) Design an encoder/decoder pair that generates signal approximations to the sample paths of $x_{\mathrm{B}}[n]$ that satisfy the quality criterion:

$$E\left[|\hat{x}_B[n] - x_B[n]|^2\right] \leq 0.05\,E\left[|x_B[n]|^2\right]\ ;$$

while minimizing the average number of bits used by your encoded description.

Problem 2.5 Channel Deconvolution

The objective of this part of the project is to develop a channel deconvolution algorithm to be used for recovering a binary-valued information bearing sequence. In particular, the real-valued signals that are outputs of the Matlab program `BlackBoxC` correspond to sample path segments of the output of a causal and stable LTI system with unknown (but fixed) impulse response $h[n]$, to a (different in each case) sample path of an IID sequence $s[n]$ of equally likely ± 1's. Specifically, calling the Matlab program `BlackBoxC(N)` produces as output an N-point vector $[x_{\mathrm{C}}[1] \;\; x_{\mathrm{C}}[2] \;\; \cdots \;\; x_{\mathrm{C}}[N]]^T$, which is a segment of the sequence

$$x_{\mathrm{C}}[n] = \sum_{k=0}^{\infty} h[k]\, s[n-k]$$

where $\{s[n]\}_{n=-\infty}^{\infty}$, is an arbitrary sample path of the binary-valued IID information-bearing process. Given as many observations as you wish of the output of the channel, your task is to design a method for estimating the channel $h[n]$, or more precisely, mitigating its effects so that the information bearing signal can be recovered. Together with your code you should list also the assumptions you have made in the process and the extent to which the information bearing sequence is recoverable.

Apendix 2.1 Matlab Programs for Project 2

You can get a copy of the functions `BlackBoxA`, `BlackBoxB`, and `BlackBoxC` from the course web page. The usage of these the three programs is given by

```
x = BlackBoxA(N);
```

```
x = BlackBoxB(N);
```

and

```
x = BlackBoxC(N);
```

where `x` is the output vector of length N (to be specified by the user).

For your convenience, you can also pass a second (integer) argument to `BlackBoxA`, `BlackBoxB`, and `BlackBoxC`; this allows you to fix the random seed in the programs; for instance,

```
x = BlackBoxA(N,0);
```

sets the random seed of the function `BlackBoxA` to 0, so the program returns the same sample path every time it is run (namely, the one corresponding to the seed equal to 0).

Apendix 2.2 Fodder for Individual Projects

You may wish to substitute Problems 2.4 and 2.5 with an alternative set of project exercises (though, you would still be responsible for Problems 2.1–2.3). Here is a list of ideas that can be used as a starting point for selecting such a set of project exercises.

- Single-input multiple-output channel estimation based on second-order statistics;
- Higher-order methods for channel identification;
- Pilot-assisted channel identification of wireless channels;
- Robust signal modeling over noisy channels;
- Comparative studies of subband coding and linear predictive coding for image compression;
- System identification methods for chaotic systems.