

LINEAR PREDICTIVE CODING

Program #5

EE 8993: Speech Processing

Audrey Le

Professor: Dr. Joseph Picone

June 18, 1998



1. PROBLEM DESCRIPTION

In this project we are to implement a program that calculates the spectrum of a signal using DFT and linear prediction model. The program will read in binary 16-bit linear audio data and produce output that can be used by xmgr, a Unix plotting tool, to display the result.

The program allows the user to select various options via the command line arguments to control the operation of the program. These options include allowing the user to choose the sample frequency of the signal, the preemphasis constant, the size of the window and frame, the center time of the window, the type of window, the order of the linear prediction model, and the number of channels. The options supported are summarized and their default values are given below.

- sample frequency of the signal (default 8000Hz)
- preemphasis constant (default 0.95)
- window duration in msec (default 30 msec)
- frame duration in msec (default 20 msec)
- center time for the window in msec (default 15 msec)
- window type: rectangular or hamming window (default hamming window)
- the linear prediction order (default 16)
- number of channels (default 1)

Several experiments will be performed to evaluate the algorithm. The results obtained from the LPC and DFT methods will be plotted on a log amplitude vs. linear frequency scale will be analyzed and compared to those obtained from DFT method.

The speech files used in the experiments consists of one and two-channel data. Both type of data used in the experiments are 16-bit linear data sampled at 8000Hz. The following files are used in the experiments.

Filename	Type
710_b_8k.raw	one-channel
sw2001.raw	two-channel

Table 1: Files used in the experiments.

The one-channel data can be obtained from www.isip.msstate.edu/resources/courses/ece_8993_speech/homework/1996/data. The two-channel data is located at isip/d00/switchboard/data/20/2001.

The one-channel data has the following format:

<chan 0 byte 0>< chan 0 byte 1><chan 0 byte 2> etc...

while the two-channel data use an interleave format:

<chan 0 byte 0>< chan 1 byte 0><chan 0 byte 1><chan 1 byte 1> etc...

2. INTRODUCTION

The linear prediction model was first introduced by Gauss in 1795 [1]. Since then, it has been found to be useful in many domains. In neurophysics, it is used to describe the different spectra of the EEG signals [2]. In geophysics, it is used to model the seismic traces to determine the presence of oil [3]. In speech, it is used to model speech waveform and estimate speech parameters [4].

Linear prediction model speech waveform by estimating the current value from the previous values. The predicted value is a linear combination of previous values. The linear predictor coefficients are determined such that the coefficients minimize the error between the actual and estimated signal. The basic equation of linear prediction is given as follows:

$$\tilde{s}(n) = \sum_{k=1}^p \alpha_k s(n-k) \quad (1)$$

where $\tilde{s}(n)$ is the estimated sample of the actual sample $s(n)$ from the linear combination of p samples with α_k as the coefficients.

A prediction is useless if that prediction is inaccurate. Thus, the purpose is to minimize the prediction error. That is, to minimize Equation (2).

$$E = \sum_n e^2(n) \quad (2)$$

where E is the short-time average prediction error and $e(n) = s(n) - \tilde{s}(n)$ is the individual error.

An example of a linear predicted signal is given in Figure 1. The black waveform shown in Figure 1 is the spectrum of the actual speech waveform, and the red waveform also shown in Figure 1 is the spectrum of the predicted waveform.

The large peaks in the speech waveform contain critical information that helps the recognition system identify the signal. The small peaks often are noise and sometimes can confuse the system. One advantage of linear prediction is that it smooths over these small peaks. Another advantage of linear prediction is that because it represents the actual waveform with a small number of coefficients (for speech signal 16-20 coefficients for 10 msec window), it reduces the number of bits for transmission and storage of the actual signal.

Various formulations for efficient computation of the predictor coefficients have been derived.

The details of these derivations can be found in [5], [6], and [7].

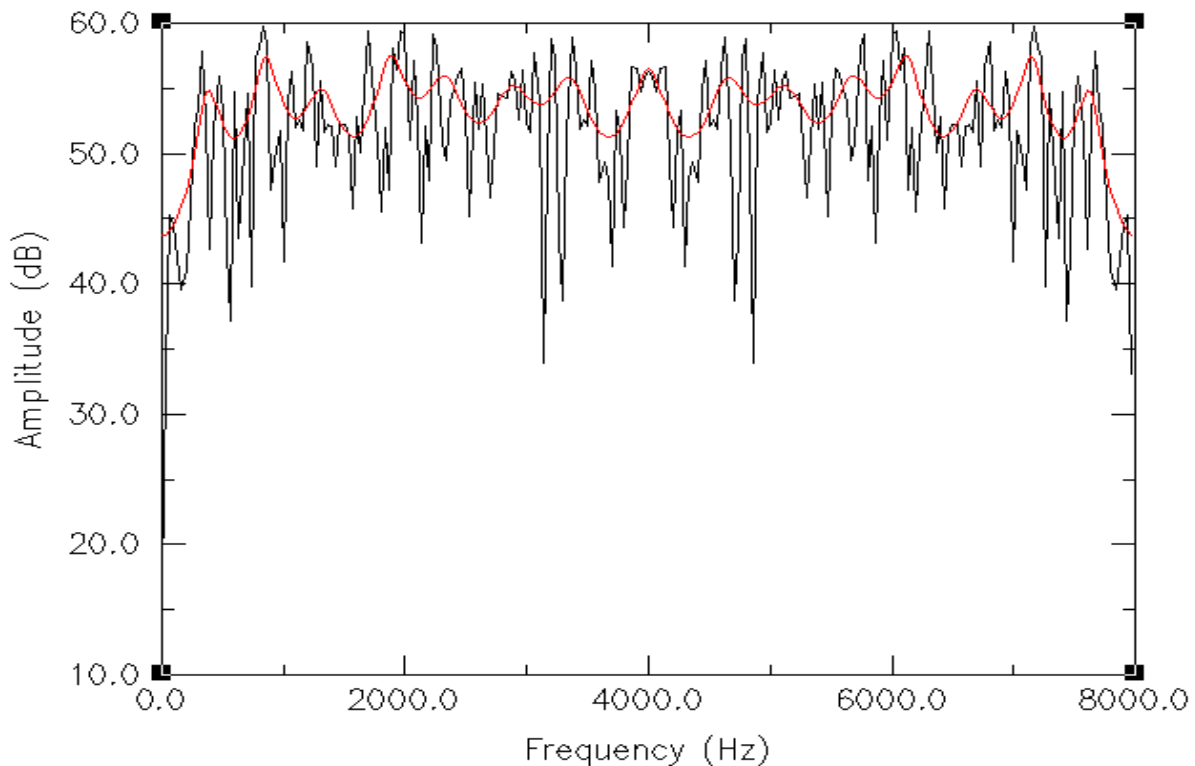


Figure 1: A sample spectrum.

3. ALGORITHMS

We implemented two methods that calculate the gain-matched spectrum of a signal. The first method is the linear prediction method and the second is the DFT method. The two methods are described below.

Levinson-Durbin's Recursion Method

There are many algorithms to find the predictor coefficients. We choose the Levinson-Durbin's recursion method due to its ease of implementation and computational efficiency. This method uses the autocorrelation coefficients to derive the reflector coefficients, and from the reflector coefficients, the predictor coefficients are obtained.

The recursion is given in Equation (3)—Equation (7).

$$E^{(0)} = R(0) \tag{3}$$

$$k_i = \left(R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right) / E^{(i-1)} \quad (4)$$

$$\alpha_i^{(i)} = k_i \quad (5)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad (6)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (7)$$

with $1 \leq i \leq p$ and $1 \leq j \leq i-1$ where i indicates the current iteration, $i-1$ indicates the previous iteration, j is the total number of iterations, and p is the order of the prediction. E is the error term, R is the autocorrelation coefficient, k is the reflection coefficient, and α is the predictor coefficient.

DFT Method

Like the previous case, there are many algorithms to calculate the DFT coefficients. However, here since we focus on the linear prediction task and not the DFT, we did not use any fast implementation of the DFT calculation but just straight implementation from the DFT equation, Equation (8).

$$\beta(i) = \sqrt{\left(\sum_{j=0}^{N-1} x(j) \cos\left(\frac{2\pi ij}{p}\right) \right)^2 - \left(\sum_{j=0}^{N-1} x(j) \sin\left(\frac{2\pi ij}{p}\right) \right)^2} \quad (8)$$

with $1 \leq i \leq p$ where i indicates the current iteration, N is the total number of iterations, and p is the order of the DFT.

4. RESULTS

We compared the spectrum obtained from the linear prediction model to that obtained from the DFT. A window of 30 msec centered at 15 msec from file 710_b_8k.raw was used as the input signal. We ran the program calculating the spectrum of this sample signal for different prediction orders while keeping the DFT order constant to model the progress of the linear prediction behavior. The results are given in Figure 2—Figure 9. As we can see, there is a direct relationship between the prediction order the prediction accuracy. As the order increases the prediction accuracy increases. The predicted signal in lower order, i.e. order 4, does not estimate the actual signal well. On the other hand, the predicted signal with higher order, i.e. order 16, does a good job of estimating the major components of the actual signal. When we get an order of 240 which is the exact same as the order of the DFT we get an exact match.

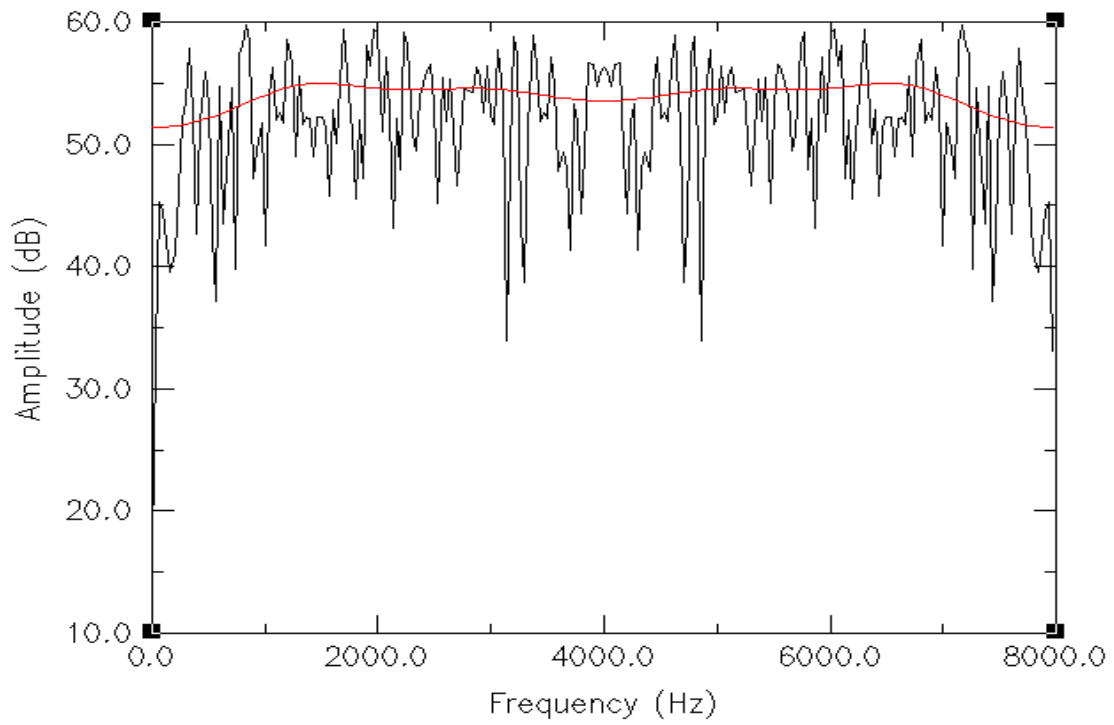


Figure 2: Spectrum of a signal using 4th order LPC (red) vs. 240 points DFT spectrum (black).

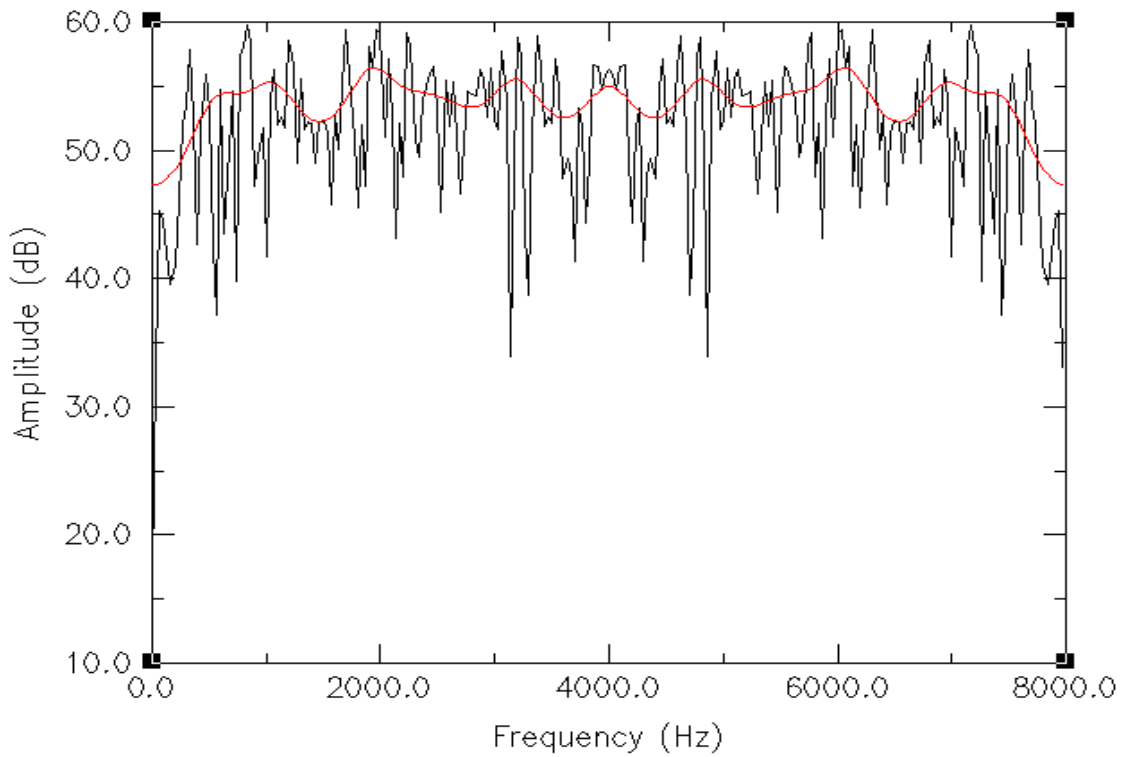


Figure 3: Spectrum of a signal using 12th order LPC (red) vs. 240 points DFT spectrum (black).

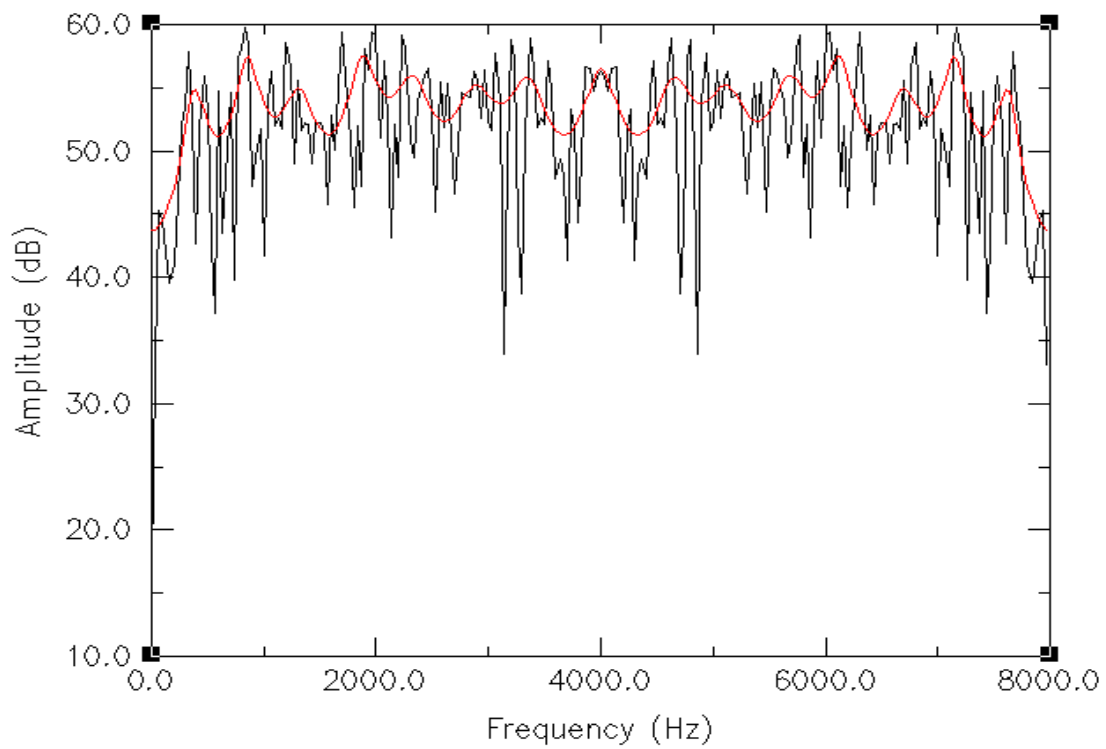


Figure 4: Spectrum of a signal using 16th order LPC (red) vs. 240 points DFT spectrum (black).

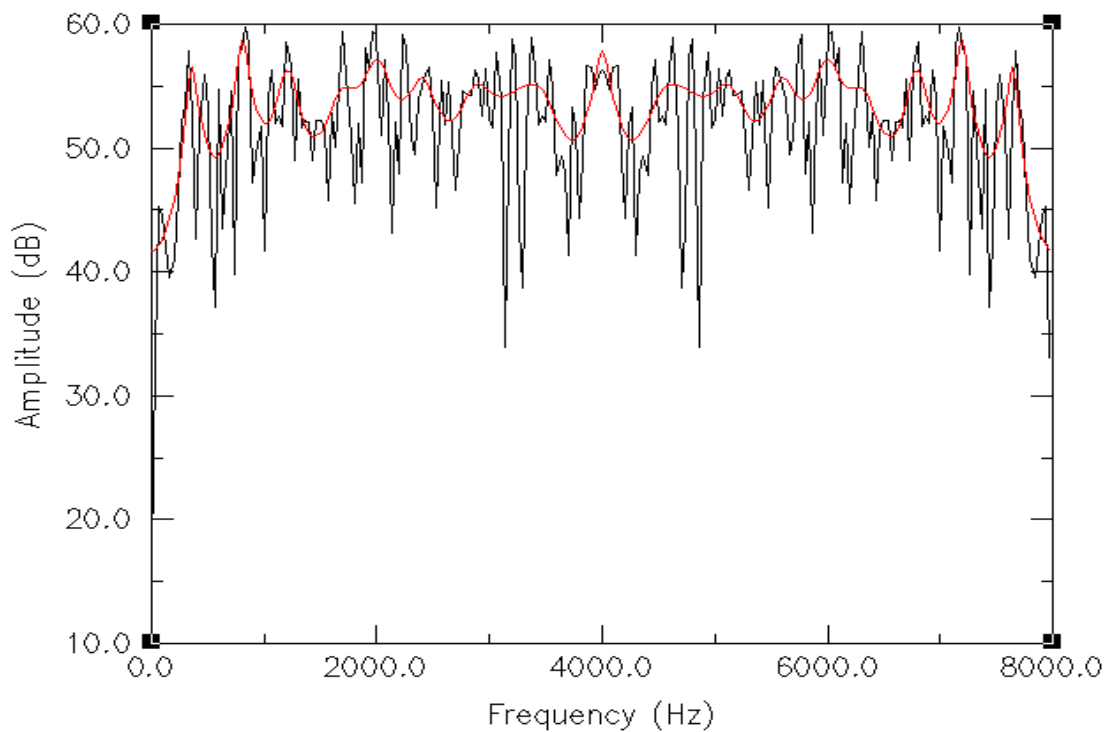


Figure 5: Spectrum of a signal using 20th order LPC (red) vs. 240 points DFT spectrum (black).

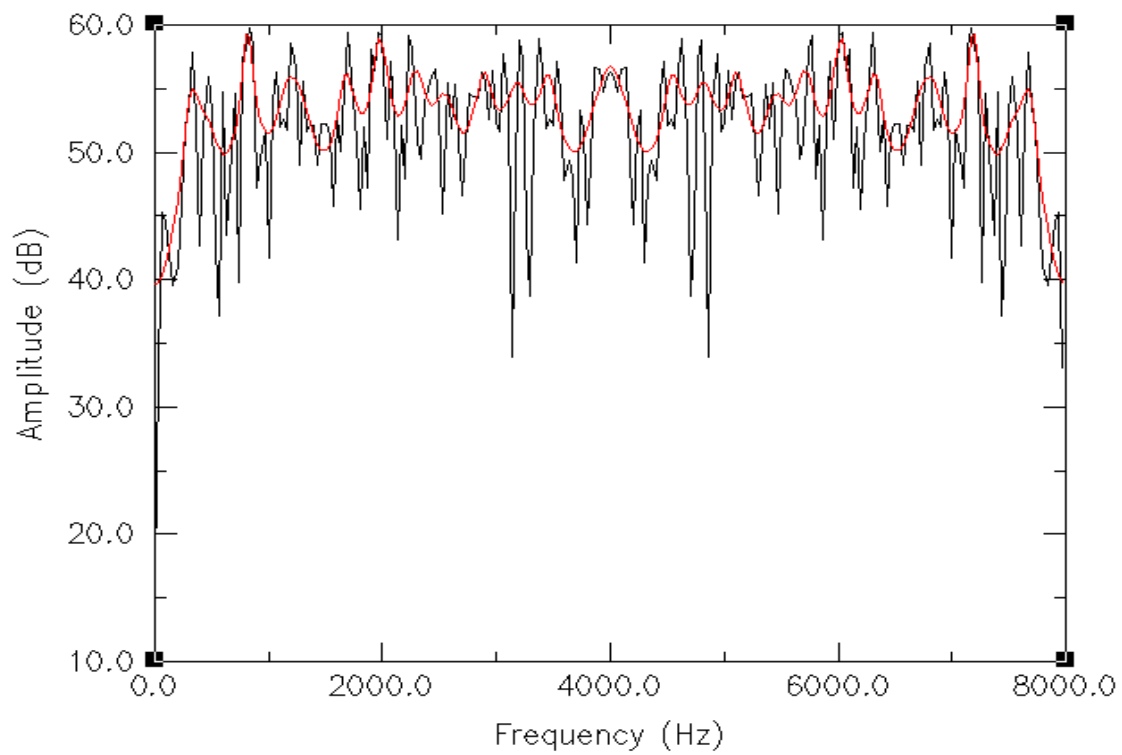


Figure 6: Spectrum of a signal using 28th order LPC (red) vs. 240 points DFT spectrum (black).

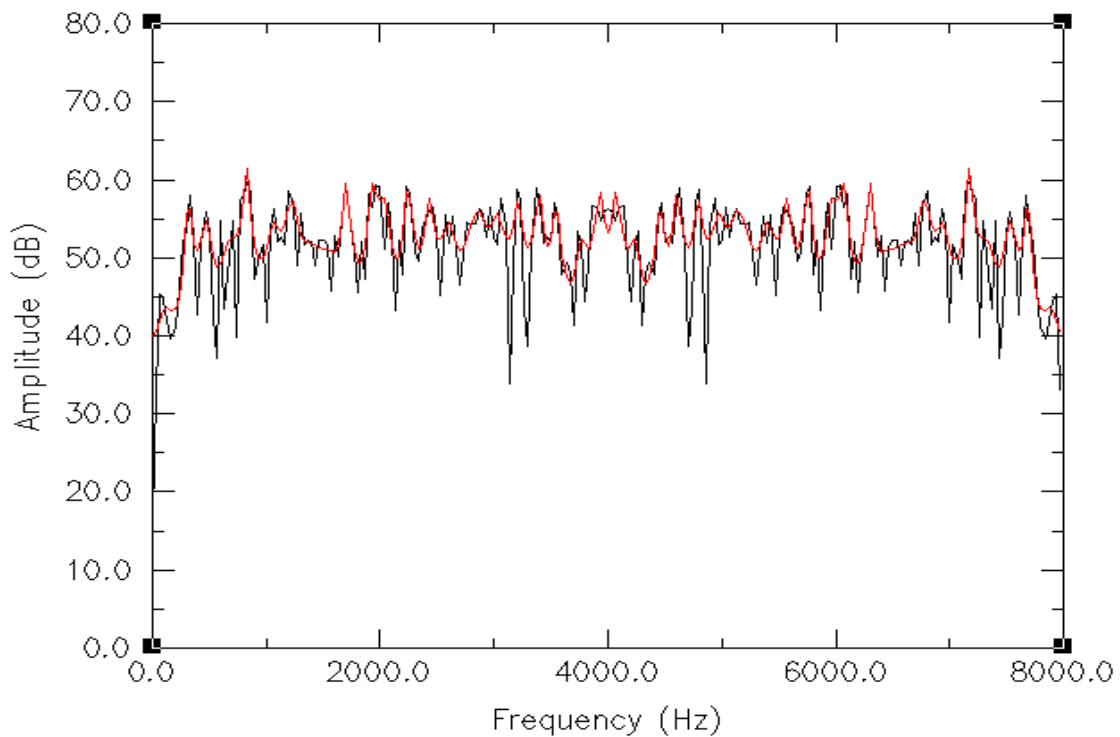


Figure 7: Spectrum of a signal using 48th order LPC (red) vs. 240 points DFT spectrum (black).

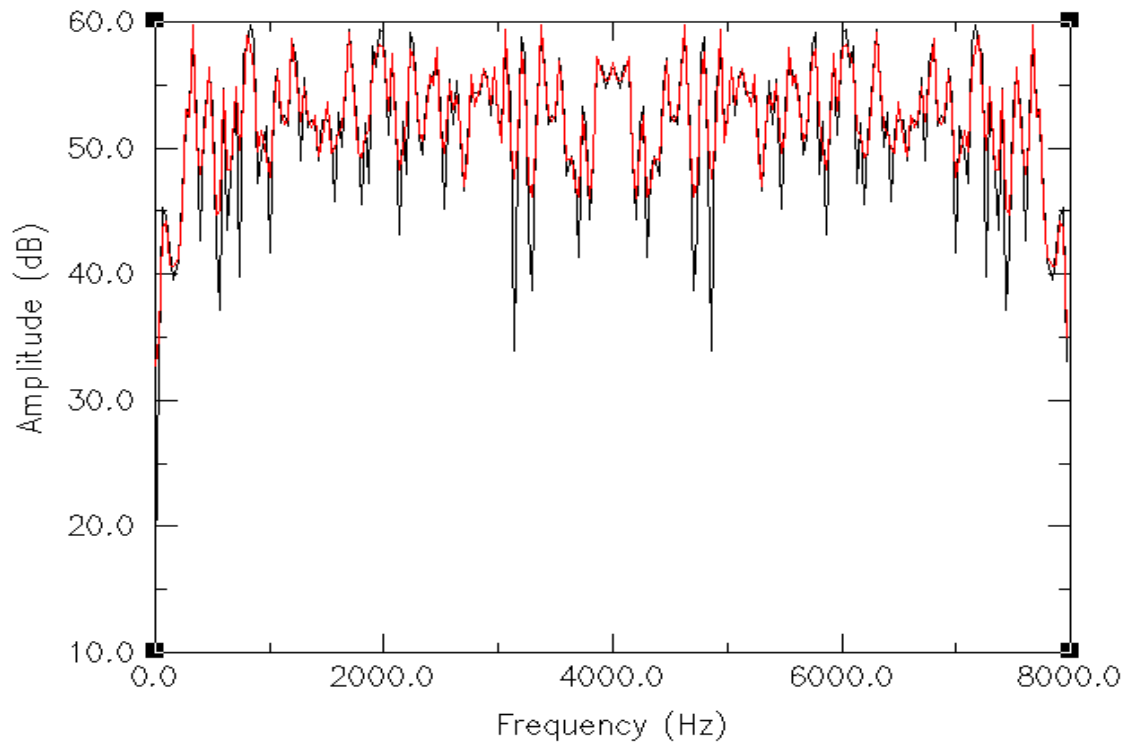


Figure 8: Spectrum of a signal using 100th order LPC (red) vs. 240 points DFT spectrum (black).

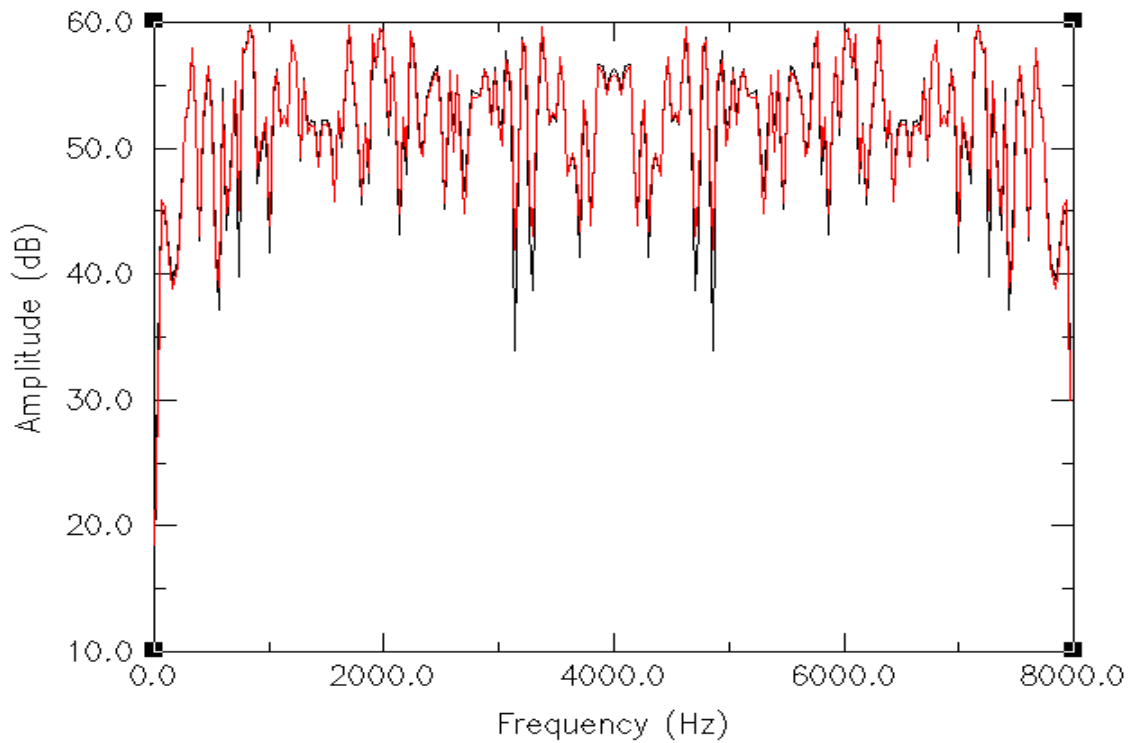


Figure 9: Spectrum of a signal using 240th order LPC (red) vs. 240 points DFT spectrum (black).

5. REFERENCES

- [1] J. Markel and A. Gray Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.
- [2] T. Bohlin, "Comparison of Two Methods of Modeling Stationary EEG Signals," *IBM J. Res. Dev.*, pp. 194-205, May, 1973.
- [3] L. Wood and S. Treitel, "Seismic Signal Processing," *Proc. IEEE*, vol. 63, pp. 649-661, 1975.
- [4] B. Atal and L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.*, vol. 50 (2), pp. 637-655, 1971.
- [5] J. Makoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE*, vol. 63, pp. 561-580, 1975.
- [6] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.
- [7] J. Picone, "Speech Processing Lecture Notes," Mississippi State University, 1996.