2002-06

A wavelet-based prediction technique for concealment of loss-packet effects in wireless channels

Garantziotis, Anastasios

Monterey California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A WAVELET-BASED PREDICTION TECHNIQUE FOR CONCEALMENT OF LOSS-PACKET EFFECTS IN WIRELESS CHANNELS

by

Anastasios Garantziotis

June 2002

Thesis Advisor:	Murali Tummala
Second Reader:	Robert Ives

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: A Wavelet-Based Prediction Technique for Concealment of Packet-Loss Effects in Wireless Channels			**5. FUNDING NUMBERS**
6. AUTHOR(S) Anastasios Garantziotis			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

In this thesis, a wavelet-based prediction method is developed for concealing packet-loss effects in wireless channels. The proposed method utilizes a wavelet decomposition algorithm in order to process the data and then applies the well known linear prediction technique to estimate one or more approximation coefficients as necessary at the lowest resolution level. The predicted sample stream is produced by using the predicted approximation coefficients and by exploiting certain sample value patterns in the detail coefficients. In order to test the effectiveness of the proposed scheme, a wireless channel based on a three-state Markov model is developed and simulated. Simulation results for transmission of image and speech packet streams over a wireless channel are reported for both the wavelet-based prediction and direct linear prediction. In all the simulations run in this work, the wavelet-based method outperformed the direct linear prediction method.

14. SUBJECT TERMS Linear prediction, Wavelets, CELP, Denoising, Markov channel			**15. NUMBER OF PAGES** 109
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A WAVELET-BASED PREDICTION TECHNIQUE FOR CONCEALMENT OF PACKET-LOSS EFFECTS IN WIRELESS CHANNELS

Anastasios Garantziotis
Lieutenant J.G., Hellenic Navy
B.S., Hellenic Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2002

Author: Anastasios Garantziotis

Approved by: Murali Tummala
Thesis Advisor

Robert Ives
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, a wavelet-based prediction method is developed for concealing packet-loss effects in wireless channels. The proposed method utilizes a wavelet decomposition algorithm in order to process the data and then applies the well known linear prediction technique to estimate one or more approximation coefficients as necessary at the lowest resolution level. The predicted sample stream is produced by using the predicted approximation coefficients and by exploiting certain sample value patterns in the detail coefficients. In order to test the effectiveness of the proposed scheme, a wireless channel based on a three-state Markov model is developed and simulated. Simulation results for transmission of image and speech packet streams over a wireless channel are reported for both the wavelet-based prediction and direct linear prediction. In all the simulations run in this work, the wavelet-based method outperformed the direct linear prediction method.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This thesis is dedicated to my wife Elena for her unlimited support that she provided through the two difficult years of my studies. Her strength and love was the source of my accomplishments.

I also dedicate this work to my parents, whose sacrifices and love made my way of life secure and beautiful, and especially to my brother, who will always be my raw model.

Special thanks to the Hellenic Navy for making this academic experience and challenge possible for me.

I would like to thank my thesis advisor, Dr. Murali Tummala, for his efforts in helping to complete this work.

I would also like to thank my second reader Dr. Robert Ives for the help he provided in fine-tuning this work.

Lastly, I would also like to thank LT K. Kamaras H. N for providing the code for the two-state Gilbert-Elliot channel which was the basis for the development of the three-state channel.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The demand for multimedia applications, such as voice, image and video, has grown significantly over the last decade. Increasingly, multimedia information is being transmitted over packet switched networks. A more recent development is multimedia transmission over wireless packet switched networks. During transmission, some of the packets belonging to a frame of voice signal or image may be lost due to channel errors or overflow in networks buffers, thereby leading to quality degradation. Linear prediction, a method that makes use of the past data to predict future (missing) data, can be used to estimate lost packets, thereby providing error concealment to improve the quality of received signals. The work reported in this thesis comes under a broader effort of the application of advanced signal processing techniques for packet stream processing (versus the traditional sample stream processing).

When the transmitted signal samples are not highly correlated or when they contain high frequencies, the performance of the linear prediction is poor [3], making concealment of errors due to lost packets from the end user somewhat ineffective. In an effort to overcome this limitation, in this thesis, we propose the use of a wavelet-based prediction scheme for concealment of effects due to lost packets in speech and image packet streams. The proposed method utilizes a wavelet decomposition algorithm to preprocess the available parameter values from the past packets in the stream and then applies one of the well-known linear prediction techniques (covariance method) to estimate one or more approximation coefficients as necessary at the lowest resolution level. The parameter values corresponding to the lost packets are obtained by using the predicted approximation coefficients and by exploiting certain sample value patterns in the detail coefficients in the wavelet synthesis structure.

In order to test the effectiveness of the proposed scheme, a wireless channel based on a three-state Markov model is developed and simulated. The model simulates the lossy conditions of packet stream transmission over a wireless channel. The model consists of a “Good” state, where an error-free transmission is assumed; a “Bad” state, where there is a

probability of single data packets being dropped; and a “Bursty” state, where consecutive packets may be dropped.

Simulation experiments of image and speech packet stream transmission over the three-state Markov wireless channel are conducted and results reported for both the wavelet-based prediction and direct linear prediction. An objective of these simulation experiments is to compare the error concealment performance of the wavelet-based method with that of the direct linear prediction method; in all simulation runs conducted in this work, the wavelet-based method outperformed the direct linear prediction method. The reasons for improved performance can be attributed to the following. First, the wavelet method results in a smaller prediction error than the direct linear prediction since prediction is carried out on signal components at low frequencies. Second, the role that the prediction error plays in the estimation of the missing parameter values is weighted down by using known past approximation and detail coefficients and known future detail coefficients as derived from the patters in the wavelet decomposition.

During the course of this work several topics for possible future efforts were considered. Suggestions for future work include utilizing both forward and backward linear prediction, which may help improve the accuracy of prediction. Also recommended is a more in-depth investigation to develop rigorous mathematical basis in an effort to generalize the proposed approach to other classes of wavelets.

I. INTRODUCTION

A. BACKGROUND

The demand for multimedia applications, such as voice, image and video, has grown significantly over the last decade. Increasingly, multimedia information is being transmitted over packet switched networks. A more recent development is multimedia transmission over wireless packet switched networks.

During transmission, some of the packets belonging to a frame of voice signal or image file may be lost due to channel errors or overflow in network's buffers. The lost or missing packets and the possible data misalignment that results in the receiver lead to quality degradation. Furthermore, in real-time voice and video transmission, delay is an important factor. This additional constraint significantly affects the quality of the reconstructed bitstream; also, the delay constraint does not permit the use of ARQ techniques. Nevertheless, other techniques, such as error correction, error resilience and error concealment, can be applied to the received signal in order to enhance its quality [17].

Linear prediction can be used to estimate lost packets, thereby providing error concealment to improve the quality of received signals. Linear prediction makes use of the known data from past packets in order to predict the values of missing packets.

When the transmitted signal samples are not highly correlated or when they contain high frequencies, the performance of the linear prediction is poor [3]. Since high correlation among samples is not assumed for many practical signals and given that errors are inevitable in hostile environments, such as wireless channels, linear prediction is not robust enough to hide or correct the errors due to lost packets from the end user.

B. THESIS OBJECTIVES

The objective of this thesis is to investigate the use of wavelet-based linear prediction for concealment of effects due to lost packets in speech and image packet streams. In this method the received data is processed through a wavelet decomposer, and

then the low frequency part of the signal is fed to a linear predictor in order to estimate the parameters of the missing packets.

Another objective is to develop a three-state Markov model to simulate the lossy conditions of data transmission over a wireless channel. The model consists of a "Good" state, where an error-free transmission is assumed; a "Bad" state, where there is a probability of single data packets being dropped; and a "Bursty" state, where consecutive packets may be dropped.

The proposed scheme will be applied on images and real-time voice packet streams, both uncompressed and compressed. The error concealment performance of this scheme will be compared with that of linear prediction through simulation results.

C. RELATED WORK

Depending on the type of information they exploit, concealing techniques are divided into source coder-independent and source coder-dependent schemes [17]. Source coder-independent schemes are further subdivided into sender-receiver-based schemes and receiver-based schemes, linear prediction being one of them [17]. Linear prediction, a method of predicting a present sample when a sufficient number of past samples is known [6], can be used to estimate parameters in a lost packet in order to conceal the packet-loss effects.

Wavelet analysis can be used to decompose an image or a speech signal into multiple resolution versions of the same signal, i.e., decompose a given signal into multiple (typically, non-overlapping) frequency bands or components [3], [9]. Wavelets have been successfully applied to many signal processing applications, such as image [3] and speech [18] compression and transient signal analysis [19]. To the best of the author's knowledge, the wavelet analysis has not been applied to prediction for concealment applications.

A two-state Markov model has been thoroughly analyzed and successfully used to model the lossy conditions of data transmission in a wireless channel [10], [11], [12]. In this work, the two-state model has been extended to a three-state model to account for the

bursty packet losses, which for example is a suitable way to express the loss of macroblocks in image packet transmission.

D. THESIS ORGANIZATION

This thesis is organized into six chapters and two supporting appendices. Chapter II provides an overview of the wavelet theory. Chapter III provides a quick overview of the principles of linear prediction focusing on the modified covariance method. A wavelet-based prediction method is introduced in this chapter. Chapter IV presents a three-state Markov representation to model a wireless channel. In Chapter V, simulation results of linear prediction and the wavelet-based prediction applied to image and voice packet streams are presented.

Chapter VI presents a summary of the work done in the thesis and includes conclusions and recommendations for further research. Appendix A contains the Matlab code of prediction algorithms using various wavelet families and the simulated wireless channel. Appendix B presents additional simulation results for images.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WAVELETS

This chapter provides an overview of the wavelet theory. In the following chapters, wavelet decomposition is used to improve the prediction performance for application to conceal the effects of information loss in packet switched networks.

A. WAVELETS

The wavelet transform provides the time-frequency representation of a signal simultaneously. Most wavelets are based on a single function called the mother wavelet $\psi(t)$, which is a function with some special properties [2]. For example, a wavelet $\psi(t)$ is a function having zero average value:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{2.1}$$

The function $\psi(t)$ can be scaled by replacing t with t/α, where α is a parameter that regulates the amount of scaling [3]. Scaling or dilation means shrinking as well as stretching of $\psi(t)$ along the time axis. The square of the norm of $\psi(t)$ is given by:

$$\|\psi(t)\|^2 = \int_{-\infty}^{+\infty} \psi^2(t)dt \tag{2.2}$$

The squared norm of a scaled function is:

$$\left\|\psi\left(\frac{t}{\alpha}\right)\right\|^2 = \int_{-\infty}^{+\infty} \psi^2\left(\frac{t}{\alpha}\right)dt = \alpha\|\psi(t)\|^2 \tag{2.3}$$

In order to keep the same norm for the scaled function, the scaled function must be multiplied by $1/\sqrt{\alpha}$.

A translation of the mother wavelet $\psi(t)$ is obtained by replacing t with t-b, or t+b, depending on whether the mother wavelet is moved to the right or to the left on the time axis. Consequently, the normalized, scaled and translated versions of the mother wavelet $\psi(t)$ are given by [3]:

$$\psi_{\alpha,b}(t) = \frac{1}{\sqrt{\alpha}}\psi\left(\frac{t-b}{\alpha}\right) \tag{2.4}$$

For different values of α and b, a group of shifted and scaled functions forms a wavelet basis. For each wavelet, there can be a different basis, and even for the same wavelet there can be multiple bases depending on the translations chosen.

B. DISCRETE WAVELET TRANSFORM

In order to go from the continuous to the discrete representation of the mother wavelet $\psi(t)$, the following transformation is used [3]:

$$\begin{aligned} \alpha &= \alpha_0^{-j} \\ b &= kb_0\alpha_0^{-j} \end{aligned} \tag{2.5}$$

where the widely used choices for α_0 and b_0 are $\alpha_0 = 2$ and $b_0 = 1$ [3]. Substituting (2.5) into (2.4) yields:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{2.6}$$

where j is the parameter of the scale (or resolution level) and k the shift, and both take only integer values.

Another function that needs to be introduced is the scaling function $\phi(t)$. A weighted sum of the translated versions of the scaling function can be used to represent a function $x(t)$ [3]:

$$x(t) = \sum_{k=-\infty}^{+\infty} \mu_k \phi(t-k) \tag{2.7}$$

where μ_k are the weights of this representation. Similar to (2.6), scaled and translated versions of the scaling function, in different resolution levels, can be represented as follows:

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \tag{2.8}$$

A function $x(t)$ can be represented as a weighted sum of a low-pass scaling function, or a version of it, and a band-pass wavelet function, i.e., scaled and translated versions of the wavelet and the scaling function, given by [3]:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_{j_0,k}\phi_{j_0,k}(t) + \sum_{k=-\infty}^{+\infty}\sum_{j=j_0}^{+\infty} d_{j,k}\psi_{j,k}(t). \tag{2.9}$$

where $\psi_{j,k}(t)$ is given in (2.6) and $\phi_{j_0,n}(t)$ is given in (2.8); the wavelet or detail coefficients $d_{j,k}$ and the scaling or approximation coefficients $c_{j,k}$ are defined as [3]:

$$d_{j,k} = \int_{-\infty}^{+\infty} x(t)\psi_{j,k}(t)dt \tag{2.10}$$

$$c_{j,k} = \int_{-\infty}^{+\infty} x(t)\phi_{j,k}(t)dt \tag{2.11}$$

The scaling function $\phi(t)$ can be represented in terms of its dilated versions at a higher resolution level:

$$\phi(t) = \sum_{k=-\infty}^{+\infty} h_k \phi_{1,k}(t) \tag{2.12}$$

where h_k are the low-pass filter or scaling function coefficients. By using (2.8), for $j = 1$, the preceding equation becomes:

$$\phi(t) = \sum_{k=-\infty}^{+\infty} h_k \sqrt{2}\phi(2t-k) \tag{2.13}$$

which is known as the multiresolution analysis (MRA) equation [3].

The set of all functions that can be obtained as a linear combination of the set $\{\phi(t-k)\}$ is called span$\{\phi_{j,k}(t)\}$ and is denoted as:

$$V_j = \overline{Span\{\phi_{j,k}(t)\}} \tag{2.14}$$

A function $x(t) \in V_j$ at resolution j can also be described as spanning two subspaces at the next lower resolution level, represented as:

$$V_j = V_{j-1} \oplus W_{j-1} \tag{2.15}$$

where W_{j-1} is an orthogonal compliment of V_{j-1}. Since the subspaces V_{j-1} and W_{j-1} resolve back into V_j, no information is lost in the process [5].

The wavelet $\psi(t)$ can also be represented as a weighted sum of scaled versions of $\phi(t)$ at resolution j_0:

$$\psi(t) = \sum_{k=-\infty}^{+\infty} w_k \phi_{j_0,k}(t) \tag{2.16}$$

where w_k are the high-pass filter or wavelet coefficients. As done earlier, by substituting (2.8) into (2.16), for $j = 1$, we have:

$$\psi(t) = \sum_{k=-\infty}^{+\infty} w_k \sqrt{2}\phi(2t - k) \tag{2.17}$$

The wavelet and scaling coefficients satisfy the following conditions [3]:

$$w_k = \pm(-1)^k h_{N-1-k} \tag{2.18}$$

where N is a even integer number. Equation (2.18) shows that the low-pass filter or approximation coefficients and the high-pass filter or detail coefficients are related to each other. Another condition is:

$$\sum_{k=-\infty}^{+\infty} h_k w_{k-2j} = 0 \tag{2.19}$$

which shows that the low-pass filter or approximation coefficients and the high-pass filter or detail coefficients are orthogonal to each other [8]. The following equation:

$$\sum_{k=-\infty}^{+\infty} w_k = 0 \tag{2.20}$$

shows that the sum of all wavelet coefficients w_k is zero. In order for a set of scaling coefficients h_k, to support a solution, the coefficients must satisfy the linear equation [8]:

$$\sum_{k=-\infty}^{+\infty} h_k = \sqrt{2} \tag{2.21}$$

Furthermore, the sum of the squares norm of the scaling coefficients, h_k, must be unity [8]:

$$\sum_{k=-\infty}^{+\infty} {h_k}^2 = 1 \tag{2.22}$$

Another condition:

$$\sum_{k=-\infty}^{+\infty} h_k h_{k-2j} = \delta_j \tag{2.23}$$

states that the scaling coefficients h_k are orthogonal to themselves after decimation by two [8]. Finally, the individual sum of the odd terms of h_k is $1/\sqrt{2}$ [8]:

$$\sum_{k=-\infty}^{+\infty} h_{2k+1} = \frac{1}{\sqrt{2}} \tag{2.24}$$

C. COMPUTATION OF APPROXIMATION AND DETAIL COEFFICIENTS

A more efficient way to perform the discrete wavelet transform is to compute the approximation and detail coefficients. Substituting $t = 2^j t - k$ in (2.13) yields:

$$\phi(2^j t - k) = \sum_{n=-\infty}^{+\infty} h_n \sqrt{2}\phi(2^{j+1} t - 2k - n) \tag{2.25}$$

By setting $m = 2k + n$, (2.25) becomes:

$$\phi(2^j t - k) = \sum_{m=-\infty}^{+\infty} h_{m-2k} \sqrt{2}\phi(2^{j+1} t - m) \tag{2.26}$$

Substituting (2.26) into (2.8) yields [3]:

$$c_{j,k} = \sum_{m=-\infty}^{m_0} h_{m-2k} c_{j+1,k} \tag{2.27}$$

and similarly, the detail coefficients are given by:

$$d_{j,k} = \sum_{m=-\infty}^{m_0} w_{m-2k} c_{j+1,k} \tag{2.28}$$

where the filters *h* and *w* must satisfy the equations (2.18)-(2.24) and are known as quadrature mirror or perfect reconstruction filters [5].

In Figure 2.1, the wavelet decomposition structure is shown. Given a signal $x(t)$ at a resolution 2^{j-1}, we can obtain the approximation coefficients set c_j or the low frequency version of the signal. In order to do that the signal is filtered by a low-pass filter with impulse response h_k. Then the filtered signal is downsampled by 2 to obtain the approximation coefficients at a lower resolution level 2^j. By sending the original

signal through a high-pass filter with impulse response w_k and downsampling by 2, the detail coefficients d_j at the same resolution are obtained, which consist of the high frequency components of the signal $x(t)$ that are missing from the approximation coefficients [5]. Since no information is lost in the process, the reconstruction will be perfect when going from a lower to a higher resolution level. This process can be continued in order to obtain the approximation and the detail coefficients of a higher level or lower resolution, each time by using the approximation coefficients of the previous level as input until the approximation coefficients obtained consist of only one sample.

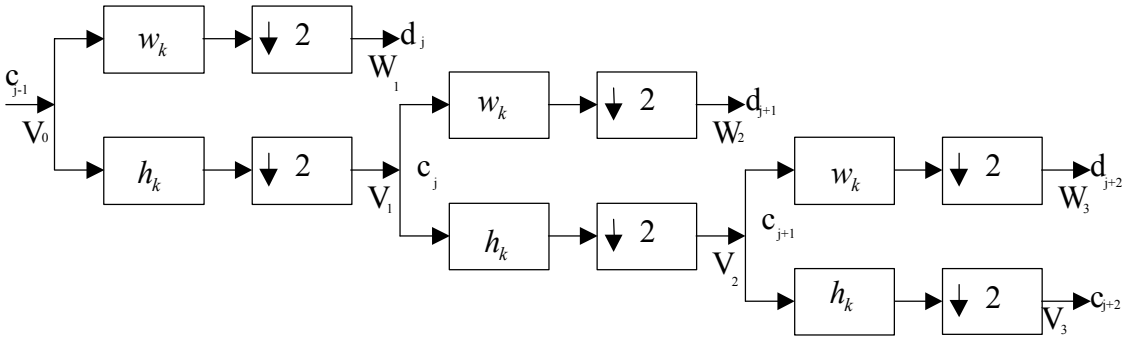


Figure 2.1. Wavelet Decomposition Structure.

The same structure shown in Figure 2.1 can also be illustrated in a signal flow form in Figure 2.2, where the original signal is decomposed into its approximation, left column, and detail coefficients, right column. Figure 2.3 illustrates the wavelet decomposition of an actual signal, which in this case is a noisy sinusoid. The signal is decomposed and its detail and approximation coefficients: d_1, d_2, d_3, d_4 and c_4 that consist of different components of the original signal $x(t)$ in different frequency bands, are obtained.

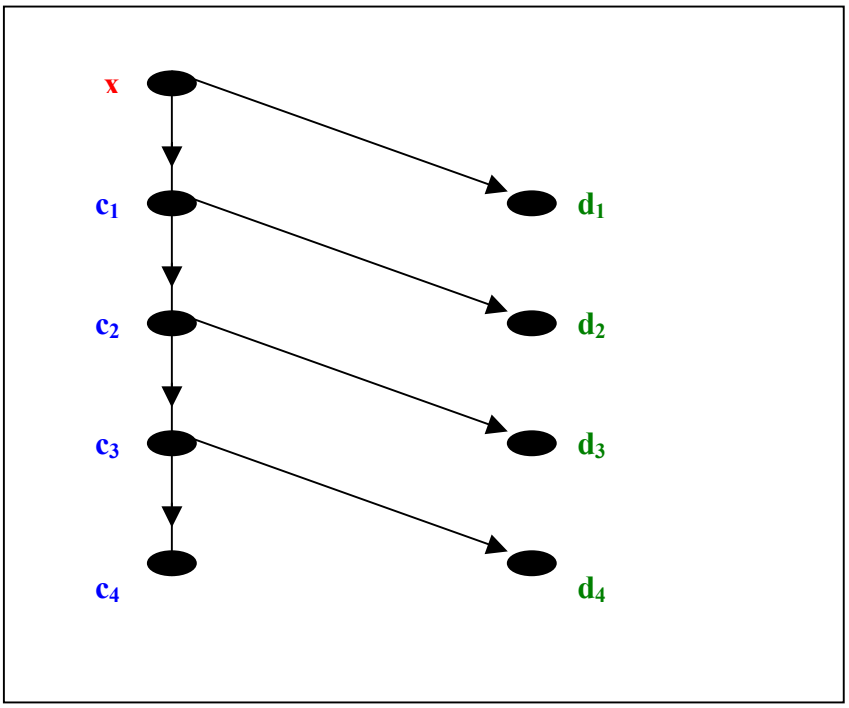


Figure 2.2. Wavelet Decomposition Tree.

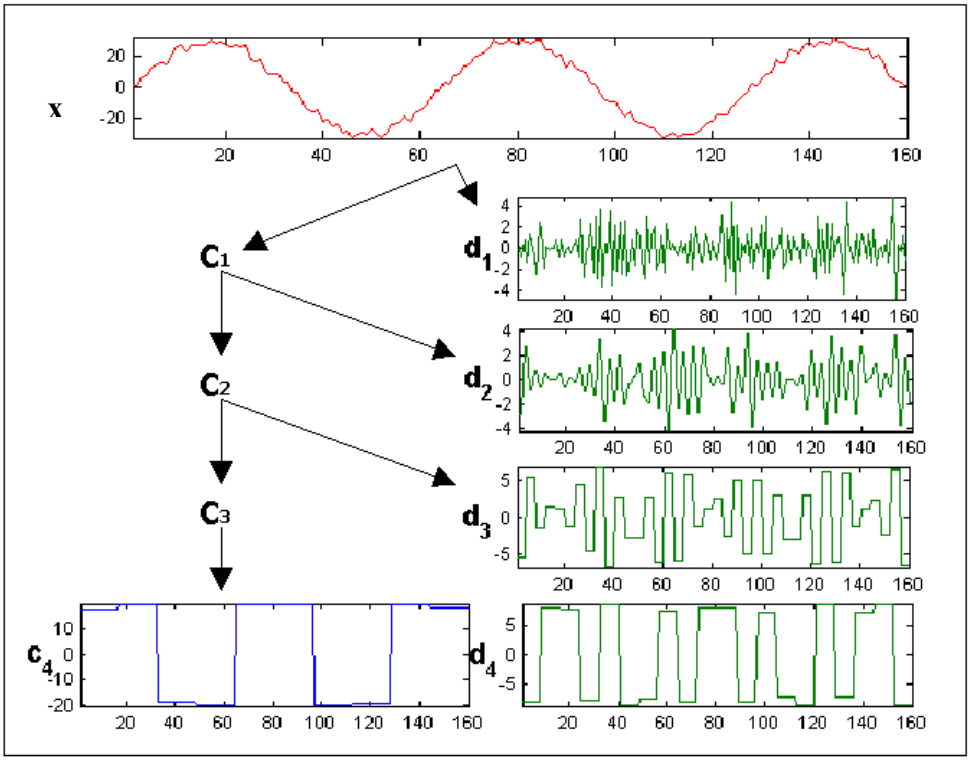


Figure 2.3. Four-Level Decomposition of a Noisy Sinusoid.

In order to reconstruct the original signal $x(t)$, given the approximation and detail coefficients, the exact opposite procedure is followed. Each time the approximation and detail coefficients of the same level of decomposition j are combined in order to reconstruct the approximation coefficients of the next higher resolution j-1. From equation (2.9) and from the analysis filter bank representation [3], if $x(t) \in V_{j+1}$, $x(t)$ can be represented in terms of $V_{j+1} = W_j \oplus V_j$, or as [3]:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_{j,k} 2^{j/2} \phi(2^j t - k) + \sum_{k=-\infty}^{+\infty} d_{j,k} 2^{j/2} \psi(2^j t - k) \tag{2.29}$$

Substituting (2.13), (2.17) and (2.26) into the preceding equation, yields:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_{j,k} 2^{j/2} \left[\sum_{m=-\infty}^{+\infty} h_{k-2m} \sqrt{2} \phi(2^{j+1} t - m) \right] + \sum_{k=-\infty}^{+\infty} d_{j,k} 2^{j/2} \left[\sum_{m=-\infty}^{+\infty} w_{k-2m} \sqrt{2} \phi(2^{j+1} t - m) \right]$$

Multiplying on both sides by $\phi(2^{j+1} t - m)$ and integrating, yields [3]:

$$c_{j+1,k} = \sum_{m=-\infty}^{m_0} h_{k-2m} c_{j,m} + \sum_{m=-\infty}^{m_0} w_{k-2m} d_{j,m} \tag{2.30}$$

Figure 2.4, shows an implementation of (2.30). At each level, the approximation and detail coefficients c_{j+2} and d_{j+2} are first upsampled by two and filtered using filters with impulse responses h_k and w_k, respectively. The output of the two filters is added, thus obtaining the approximation coefficients of the next level. The reconstruction continues by applying the same procedure to the detail and the approximation coefficients at successive levels, until the signal in its original form is obtained.

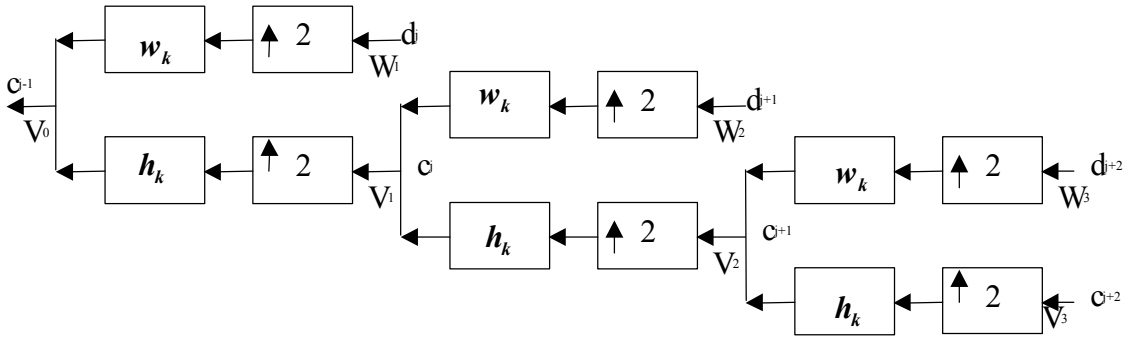


Figure 2.4. Wavelet Reconstruction Structure.

A signal flow representation of the reconstruction procedure can be seen in Figure 2.5, where the approximation and detail coefficients of each level are combined in order to obtain the approximation coefficients of the next lower level. The procedure continues until the original signal $x(t)$ is reconstructed. Figure 2.6 illustrates such a reconstruction procedure. The combined approximation and detail coefficients c_4 and d_4 yield c_3, the approximation coefficient set of the next level, which in turn, is combined with detail coefficient set d_3. If we continue combining the approximation and detail coefficients, the noisy sinusoid, decomposed in Figure 2.3, can be reconstructed.

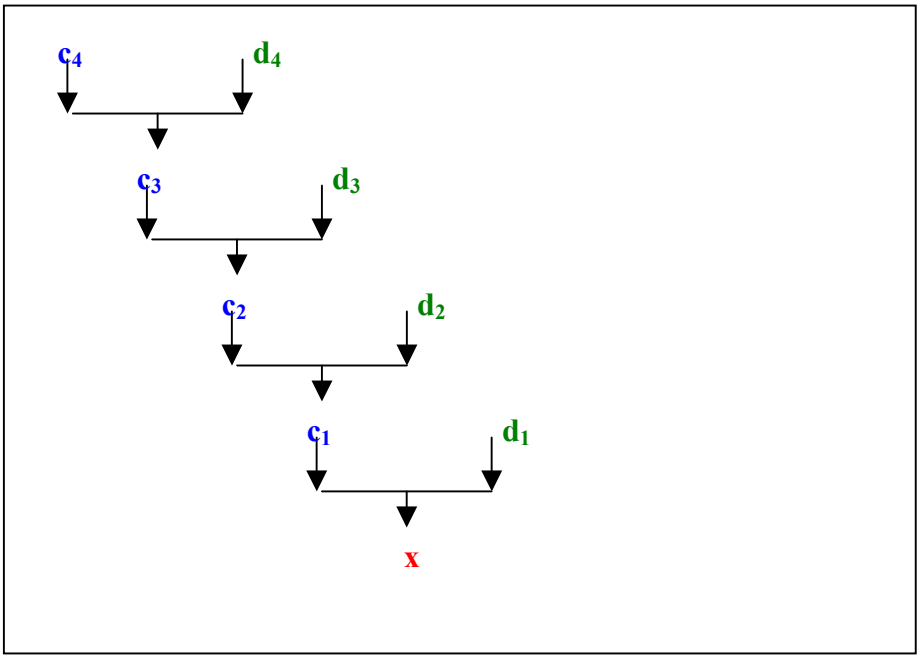


Figure 2.5. Wavelet Reconstruction Tree.

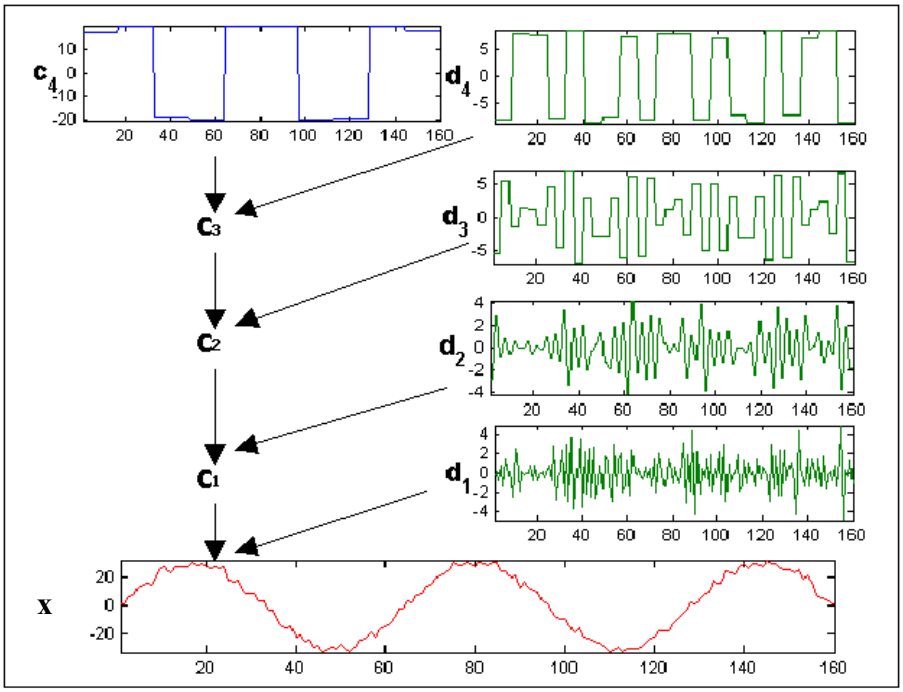


Figure 2.6. Reconstruction of a Noisy Sinusoid.

D. SUMMARY

In this chapter, we discussed the concepts of wavelets and described how the operations of wavelet decomposition and reconstruction of a signal $x(t)$ work. The wavelet function $\psi(t)$, the scaling function $\phi(t)$ and their scaled and translated versions can be used to decompose a signal $x(t)$ into orthogonal components at successively lower resolutions. The wavelet decomposition and reconstruction is a lossless process, which means that a signal $x(t)$ can be perfectly reconstructed if approximation and detail coefficients from a lower resolution are known. In the following chapter, the wavelet decomposition will be used in combination with linear prediction to enhance the prediction performance.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PREDICTION USING WAVELET DECOMPOSITION

Linear prediction can be used to predict lost data in order to conceal the effects of loss on the signal's perceptual quality. In this chapter, a wavelet based linear prediction technique will be developed in an effort to improve the prediction performance.

A. LINEAR PREDICTION

Given the past values of a signal $x[n-1], x[n-2].....x[n-p]$, the current sample $x[n]$ can be predicted as a linear combination of these past samples as [6]:

$$\hat{x}[n] = -a_1 x[n-1] - a_2 x[n-2] - a_3 x[n-3] - ..... - a_P x[n-P] \tag{3.1}$$

where a_i are the prediction filter coefficients. The objective is to find the coefficients $-a_1, -a_2, -a_3, ... - a_P$ that would minimize the error between the original and the predicted value, which is defined as [6]:

$$\varepsilon[n] = x[n] - \hat{x}[n]$$

There are several methods available in the literature of estimating the aforementioned coefficients, such as autocorrelation, covariance and the modified covariance method [6]. In this work, the modified covariance approach is used in which the criterion for estimating the prediction filter coefficients is the minimization of the sum of the squared forward and backward errors [6]:

$$S^{fb} = \left\|\varepsilon\right\|^2 + \left\|\varepsilon^b\right\|^2 \tag{3.2}$$

where $\left\|\varepsilon\right\|^2$ and $\left\|\varepsilon^b\right\|^2$ are the squared forward and backward error terms, respectively. The squared error terms are defined as:

$$\left\|\varepsilon\right\|^2 = \boldsymbol{a}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{a} \tag{3.3}$$

and

$$\left\|\varepsilon^b\right\|^2 = \boldsymbol{a}^T \tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}}\boldsymbol{a} \tag{3.4}$$

where

$$\boldsymbol{a}^T = [1 \quad a_1 \quad ..... \quad a_P]$$

is the prediction filter coefficient vector, P is the order of the filter,

$$\boldsymbol{X} = \begin{bmatrix} x[n_I] & x[n_I - 1] & ..... & x[n_I - P + 1] \\ x[n_I + 1] & x[n_I] & ..... & x[n_I - P + 2] \\ ..... & ..... & ..... & ..... \\ x[n_F] & x[n_F - 1] & ..... & x[n_F - P + 1] \end{bmatrix} \tag{3.5}$$

is the data matrix, n_I and n_F are the initial and final sample indices that define the interval over which to perform the minimization, and $\widetilde{\boldsymbol{X}}$ is the reverse data matrix given by:

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} x[n_F - P] & ..... & x[n_F - 1] & x[n_F] \\ x[n_F - P - 1] & ..... & x[n_F - 2] & x[n_F - 1] \\ ..... & ..... & ..... & ..... \\ x[n_I - P] & ..... & x[n_I - 1] & x[n_I] \end{bmatrix}$$

Minimization of Equation (3.2) yields the equation for estimating the prediction filter coefficient vector $\boldsymbol{a}$:

$$\begin{aligned} (\boldsymbol{X}^T \boldsymbol{X} + \widetilde{\boldsymbol{X}}^T \widetilde{\boldsymbol{X}})\boldsymbol{a} &= \begin{bmatrix} S^{fb} \\ \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{R}\boldsymbol{a} &= \begin{bmatrix} S^{fb} \\ \boldsymbol{0} \end{bmatrix} \end{aligned} \tag{3.6}$$

Multiplying both sides of (3.6) with the inverse of $\boldsymbol{R}$ yields the vector $\boldsymbol{a}$.

The modified covariance method has two advantages. First, twice as much data is used upon which the estimation is based. Second, no choice has to be made on whether to choose the forward or backward covariance method, which provide different but equally efficient estimations [6].

B. WAVELET BASED PREDICTION

In this thesis a wavelet-based method that improves the results of linear prediction is developed. In wavelet-based linear prediction, a signal $x(n)$ is first decomposed into its approximation and detail coefficients as shown in Figure 3.1. The number of

decomposition levels depends on the number of samples that needs to be available for linear prediction at the lowest level of resolution. After the approximation coefficients are obtained at the desired decomposition level, linear prediction is applied on them. Since the approximation coefficients, for example c_{j+2} in Figure 3.1, represent the lowest band of the signal spectrum, the prediction performance is expected to be better than that applied to the original signal.

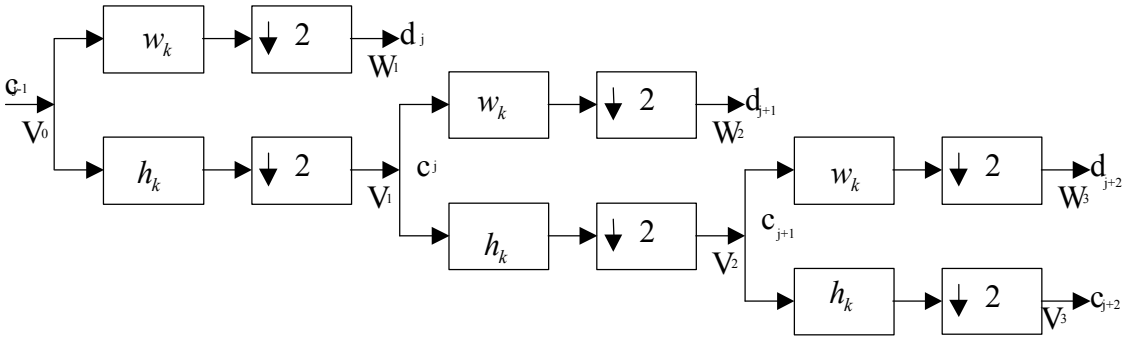


Figure 3.1. Wavelet Decomposition Structure.

1. Patterns of Approximation and Detail Coefficients

There is a certain pattern in the approximation and detail coefficients, observed when using wavelets for the decomposition of any signal $x(n)$, that can be further exploited in wavelet-based prediction. In order to demonstrate this pattern, consider the wavelet analysis scheme shown in Figure 3.1. In this work, we mainly focus on using Haar's wavelet because of its simplicity. This approach, however, can be extended to other families of wavelets. The lowpass and highpass filter coefficients for Haar's wavelet analysis are given by:

$$\boldsymbol{h_k} = \{h_0, h_1\}, \quad h_0 = \frac{1}{\sqrt{2}}, \quad h_1 = \frac{1}{\sqrt{2}} \tag{3.7}$$

$$\boldsymbol{w_k} = \{w_0, w_1\}, \quad w_0 = -\frac{1}{\sqrt{2}}, \quad w_1 = \frac{1}{\sqrt{2}} \tag{3.8}$$

respectively.

In order to estimate the approximation and detail coefficients of a given signal $x(n)$, the signal is first convolved, as shown in Figure 3.2, with $\boldsymbol{h}_k$. The result is then downsampled by two, thus obtaining the first level decomposition coefficients. In order for the successive decomposition coefficients to have an appropriate number of samples, zero-padding must take place as shown in Figure 3.3.

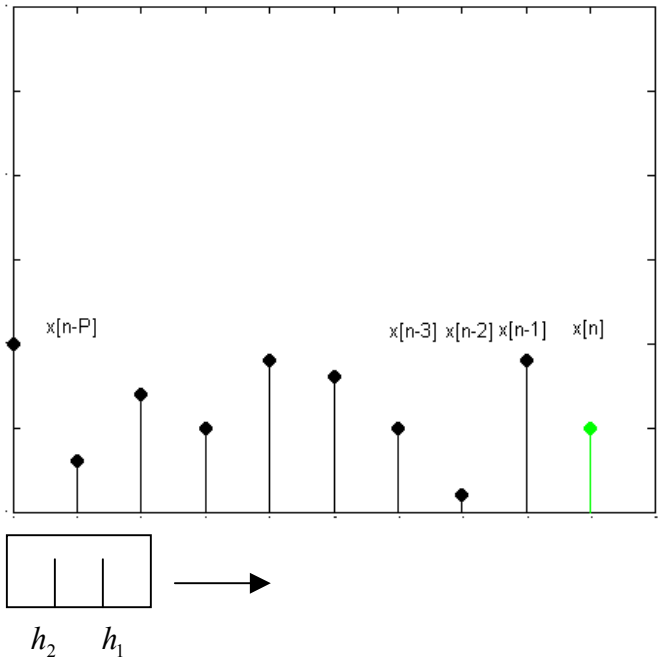


Figure 3.2. Convolution between a Signal x(n) and Haar's Wavelet.

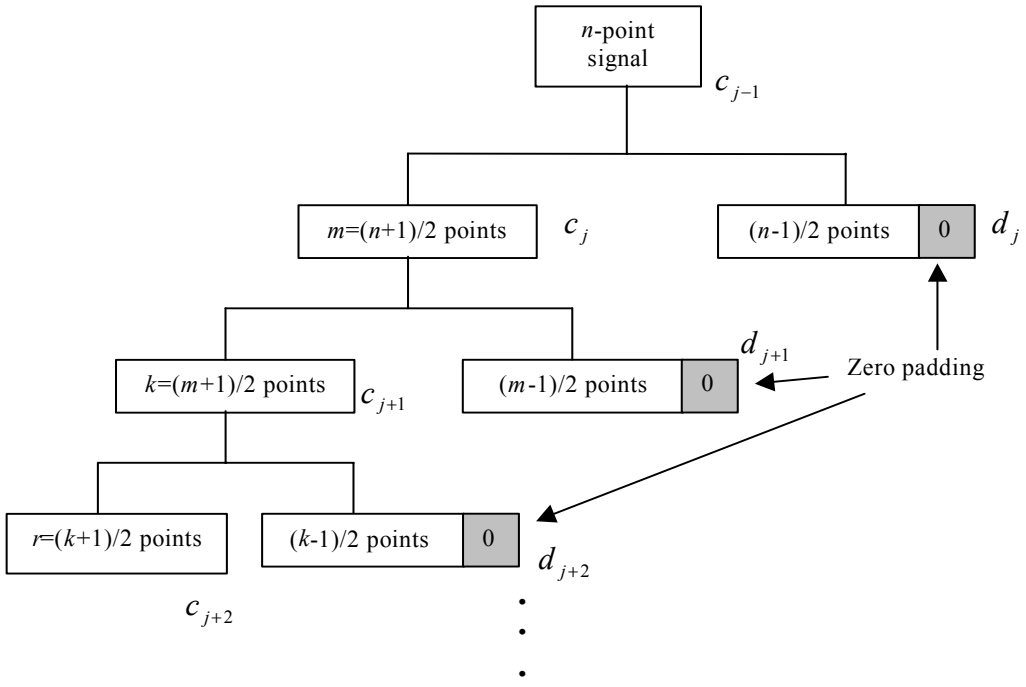


Figure 3.3. Decomposition Diagram Indicating the Number of Decomposition Sample Values at Successive Resolution Levels.

For a better understanding of the preceding procedure, Figure 3.4 illustrates the decomposition sample sizes and the necessary zero-padding for example input signal sizes of $n = 17$, 18, 19 and 20, where it is shown that each time an odd-numbered sample set is decomposed, zero-padding takes place.

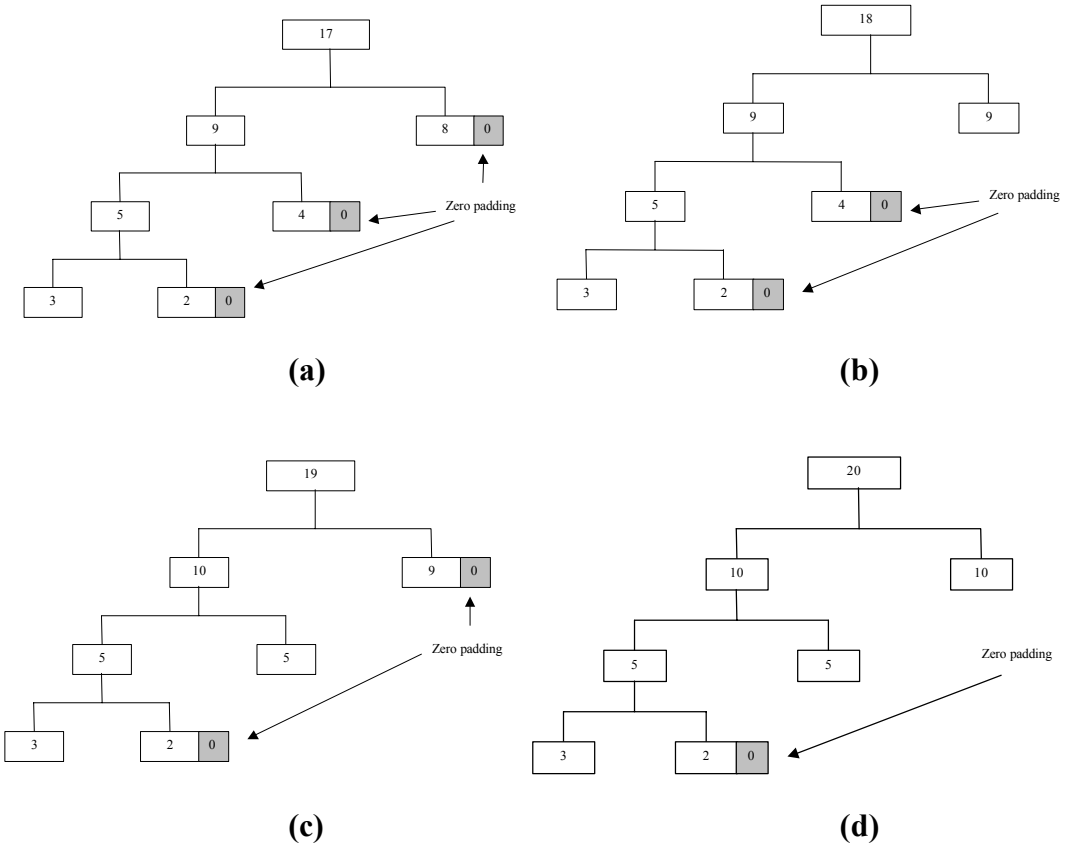


Figure 3.4. Decomposition Diagram Indicating the Number of Decomposition Sample Values at Successive Resolution Levels for Different Input Signal Sizes of (a) $n = 17$, (b) $n = 18$, (c) $n = 19$, (d) $n = 20$.

In the following, we will show that the approximation coefficient sets of the same level for two signal sizes of n and n+1 samples, respectively, are exactly the same with the exception of the very last sample.

Consider two signal vectors $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n+1}$ that differ from one another only in the last sample:

$$\begin{aligned} \boldsymbol{x}_n &= [x_1, x_2, x_3, ..., x_{n-2}, x_{n-1}, x_n] \\ \boldsymbol{x}_{n+1} &= [\boldsymbol{x}_n \; x_{n+1}] \\ &= [x_1, x_2, x_3, ..., x_{n-2}, x_{n-1}, x_{n+1}] \end{aligned}$$

Let the filter response be:

$$\boldsymbol{h_n} = [h_1, h_2]$$

By convolving $\boldsymbol{x_n}$ and $\boldsymbol{x_{n+1}}$ with $\boldsymbol{h_n}$, the resulting signals will be:

$$\begin{aligned}
\boldsymbol{y_n} &= \boldsymbol{x_n} * \boldsymbol{h_n} &= [h_1x_1, h_1x_2 + h_2x_1, ..., h_1x_{n-1} + h_2x_{n-2}, h_1x_n + h_2x_{n-1}, h_2x_n] \\
\boldsymbol{y_{n+1}} &= \boldsymbol{x_{n+1}} * \boldsymbol{h_n} &= [h_1x_1, h_1x_2 + h_2x_1, ..., h_1x_n + h_2x_{n-1}, h_1x_{n+1} + h_2x_n, h_2x_{n+1}]
\end{aligned}$$

For *n* even, downsampling $\boldsymbol{y_n}$ and $\boldsymbol{y_{n+1}}$ by 2 yields:

$$\boldsymbol{c_1^{(n_e)}} = [h_1x_1, h_1x_3 + h_2x_2, ..., h_1x_{n-1} + h_2x_{n-2}, h_2x_n] = [\boldsymbol{z_{(n_e/2)}}, h_2x_n], \quad \frac{n}{2} + 1 \text{ values} \qquad (3.9)$$

$$\boldsymbol{c_1^{(n_e+1)}} = [h_1x_1, h_1x_3 + h_2x_2, ..., h_1x_{n+1} + h_2x_n] = [\boldsymbol{z_{(n_e/2)}}, h_1x_{n+1} + h_2x_n], \quad \frac{n}{2} + 1 \text{values} \qquad (3.10)$$

For *n* odd, downsampling $\boldsymbol{y_n}$ and $\boldsymbol{y_{n+1}}$ by 2 yields:

$$\boldsymbol{c_1^{(n_o)}} = [h_1x_1, h_1x_3 + h_2x_2, ..., h_1x_{n-2} + h_2x_{n-3}, h_1x_n + h_2x_{n-1}] = [\boldsymbol{z_{(n_o+1/2)}}], \quad \frac{n+1}{2} \text{ values} \qquad (3.11)$$

$$\boldsymbol{c_1^{(n_o+1)}} = [h_1x_1, h_1x_3 + h_2x_2, ..., h_1x_n + h_2x_{n-1}, h_2x_{n+1}] = [\boldsymbol{z_{(n_o+1/2)}}, h_2x_{n+1}], \quad \frac{n+1}{2} + 1 \text{values} \qquad (3.12)$$

From (3.9)-(3.12), we observe the pattern that the decomposed values of $\boldsymbol{x_n}$ and $\boldsymbol{x_{n+1}}$ differ only in the last sample of the approximation coefficients. We will exploit this pattern in wavelet-based linear prediction.

Example 3.1

Consider a signal given by:

$$x(t) = 10\sin(2\pi t) + w(t)$$

where $w(t)$ is zero-mean white Gaussian noise with variance $\sigma^2 = 0.285$. Let us construct four signal vectors of size 17-20 as follows:

$$\boldsymbol{x_{17}} = [x_1, x_2, ..., x_{17}]$$

$$\boldsymbol{x_{18}} = [x_1, x_2, ..., x_{17}, x_{18}] = [\boldsymbol{x_{17}}, x_{18}]$$

$$\boldsymbol{x_{19}} = [x_1, x_2, ..., x_{18}, x_{19}] = [\boldsymbol{x_{18}}, x_{19}]$$

$$\boldsymbol{x_{20}} = [x_1, x_2, ..., x_{19}, x_{20}] = [\boldsymbol{x_{19}}, x_{20}]$$

The first level approximation coefficients are obtained from (3.9)-(3.12):

$c_1^{(17)} = [1.5491, 3.1829, 5.3555, 5.8636, 9.1067, 10.3957, 11.4955, 12.4582, 14.7135]$

$c_1^{(18)} = [1.5491, 3.1829, 5.3555, 5.8636, 9.1067, 10.3957, 11.4955, 12.4582, 14.5117]$

$c_1^{(19)} = [1.5491, 3.1829, 5.3555, 5.8636, 9.1067, 10.3957, 11.4955, 12.4582, 14.5117, 15.2920]$

$c_1^{(20)} = [1.5491, 3.1829, 5.3555, 5.8636, 9.1067, 10.3957, 11.4955, 12.4582, 14.5117, 14.2501]$

We may observe that successive approximation coefficient vectors differ from each other only in the last sample value. Continuing with the second and third level decomposition for $x_{17}, x_{18}, x_{19}, x_{20}$, the approximation and detail coefficients of Tables 3.1, 3.2, 3.3 and 3.4 are obtained.

	1st Level Detail Coefficients
$d_1^{(17)}$	-0.3052 -0.5128 -0.7090 -0.2939 -0.9653 0.5954 -1.3221 -0.8489 0
$d_1^{(18)}$	-0.3052 -0.5128 -0.7090 -0.2939 -0.9653 0.5954 -1.3221 -0.8489 0.2018
$d_1^{(19)}$	-0.3052 -0.5128 -0.7090 -0.2939 -0.9653 0.5954 -1.3221 -0.8489 0.2018 0
$d_1^{(20)}$	-0.3052 -0.5128 -0.7090 -0.2939 -0.9653 0.5954 -1.3221 -0.8489 0.2018 1.0420

Table 3.1. First Level Detail Coefficients.

	2nd Level Detail Coefficients
$d_2^{(17)}$	-1.1553 -0.3593 -0.9114 -0.6807 0
$d_2^{(18)}$	-1.1553 -0.3593 -0.9114 -0.6807 0
$d_2^{(19)}$	-1.1553 -0.3593 -0.9114 -0.6807 -0.5517
$d_2^{(20)}$	-1.1553 -0.3593 -0.9114 -0.6807 0.1850

Table 3.2. Second Level Detail Coefficients.

	3rd Level Detail Coefficients
$d_3^{(17)}$	-3.2436 -2.2257 0
$d_3^{(18)}$	-3.2436 -2.2257 0
$d_3^{(19)}$	-3.2436 -2.2257 0
$d_3^{(20)}$	-3.2436 -2.2257 0

Table 3.3. Third Level Detail Coefficients.

	3rd Level Approximation Coefficients
$c_3^{(17)}$	7.9756 21.7280 29.4270
$c_3^{(18)}$	7.9756 21.7280 29.0234
$c_3^{(19)}$	7.9756 21.7280 29.8037
$c_3^{(20)}$	7.9756 21.7280 28.7618

Table 3.4. Third Level Approximation Coefficients.

The results in the tables exhibit two patterns. One, the last value of the approximation coefficients changes as the size of the signal vector increases. Two, the last values of the detail coefficients consist of alternating zero and non-zero values. Figure 3.5 shows plots of the last values of approximation and detail coefficients for signal vectors of sizes 17-40.

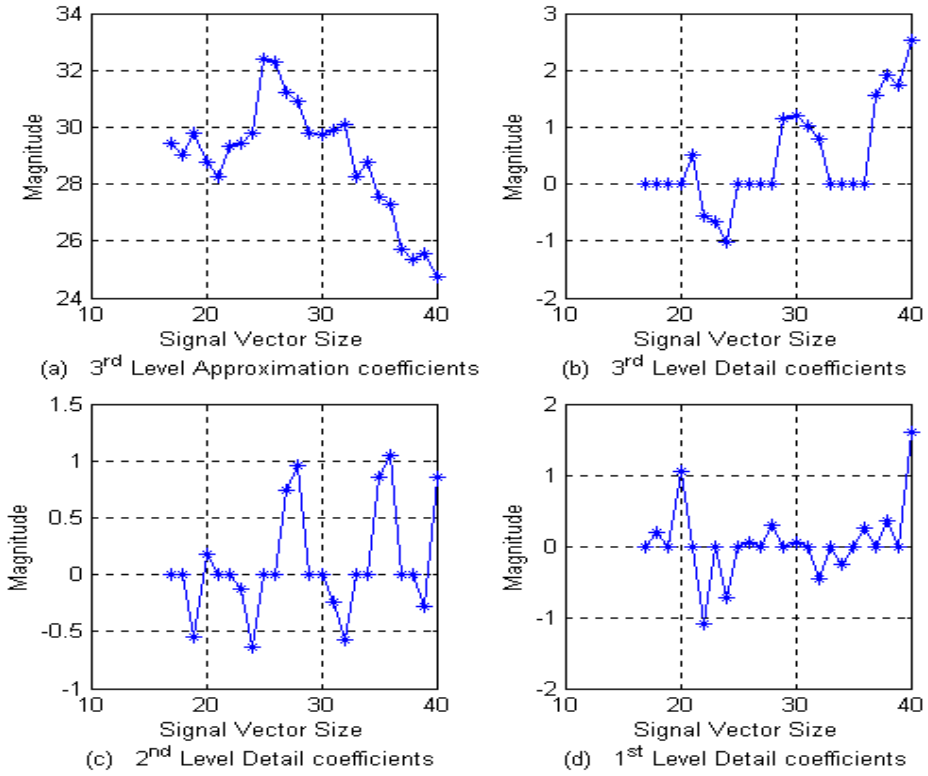


Figure 3.5. Pattern of Approximation and Detail Coefficients for Third Level Wavelet Decomposition.

C. WAVELET BASED PREDICTION ALGORITHM

The wavelet-based prediction consists of three steps. First, the given signal samples are wavelet decomposed to the desired resolution level. Second, the last sample of the highest-level approximation coefficients is predicted using the modified covariance method. Third, the detail coefficients at different resolution levels are zero-padded as necessary, and the signal is reconstructed. The reconstructed signal has all the original samples and a new predicted sample.

Let us assume that a signal $\boldsymbol{x}_n$ is given and that we wish to predict the next sample $x[n+1]$. First, the signal is wavelet decomposed to the j^{th} resolution level. The appropriate level of resolution is based on the length of the original signal $\boldsymbol{x}_n$.

Using k approximation coefficients of the highest level of decomposition j, the coefficient $\hat{c}_j[k+1]$ is predicted and appended at the end of the set. In order to do that, a matrix $\boldsymbol{C}$ on the lines of (3.5) is constructed. The filter order P depends on the number k of the available approximation coefficients. Using (3.6) of the modified covariance method, the filter coefficients a_i are obtained and then used in (3.1) to obtain $\hat{c}_j[k+1]$.

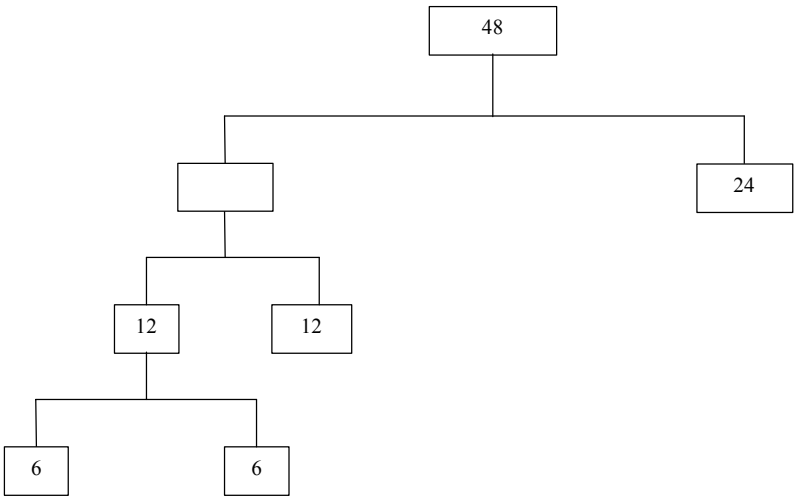
After having predicted $\hat{c}_j[k+1]$, one zero is padded at the end of the detail coefficient set in each level of decomposition, since the approximation and detail coefficient sets in each level must consist of the same number of elements. The sample $\hat{x}[n+1]$ is obtained by reconstructing the signal using the newly constructed approximation and detail coefficient sets. The procedure is further explained in Example 3.2.

Example 3.2

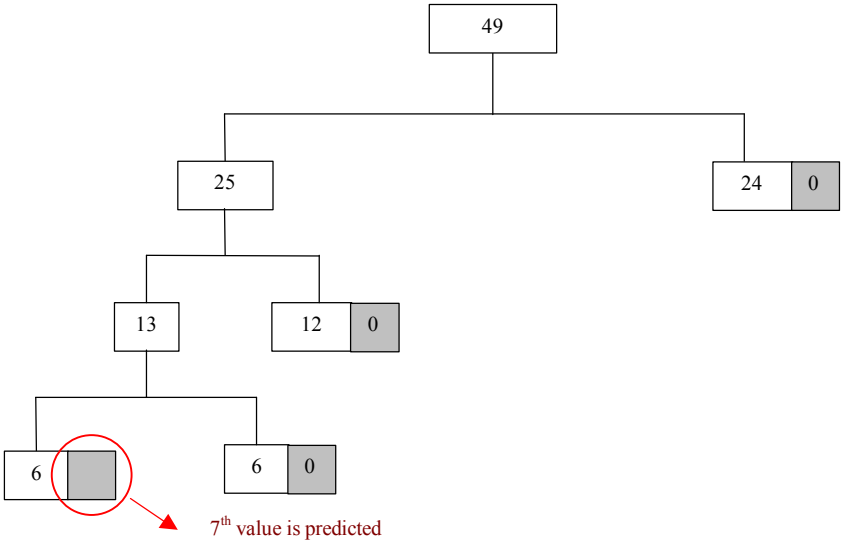
Assume that the first 48 values of the signal in Example 3.1 are known, and the goal is to predict the 49th value. For the purposes of the example, a three level wavelet decomposition is used.

Recall that, for successive signal vector sizes, only the last value of each coefficient set changes. The difference between the signal vector $\boldsymbol{x_{48}}$ and the signal vector $\boldsymbol{x_{49}}$ is in the 7th value of the 3rd level approximation coefficient set, as shown in Figure 3.6. Decomposing $\boldsymbol{x_{48}}$ yields 6 samples in each coefficient set at level 3, as shown in Figure 3.6(a). To predict the 49th sample $\hat{x}_{49}$, we need to predict the 7th sample at level 3 as shown in Figure 3.6(b) and zero-pad the detail coefficients at all levels. Applying the modified covariance method on the third level approximation coefficient set, the sample $\hat{c}_3[7]$ is predicted. The detail coefficient sets remain the same with the only difference being the padded zero at the end of each set in order to have the same number of samples in the approximation and detail sets of each level. The 49th sample of signal vector $\boldsymbol{x_{49}}$ is obtained by performing the wavelet reconstruction procedure using the

structure in Figure 3.6(b). Upon reconstruction, we have 49 values and the 49th sample is the predicted sample $\hat{x}[49]$.



(a) Decomposition of signal vector $\boldsymbol{x}_{48}$



(b) Reconstruction of signal vector $\boldsymbol{x}_{49}$

Figure 3.6. Decomposition Sample Sizes for Signal Vectors $\boldsymbol{x}_{48}$ and $\boldsymbol{x}_{49}$.

The wavelet-based prediction method has two advantages. The first is the reduction of the prediction error, since prediction is carried out on signal components of low frequencies, which is the highest-level approximation coefficient set. Secondly, the role that the prediction error plays in the estimation of the $\hat{x}[n+1]$ sample is weighted

down by using the known past values of the approximation and detail coefficient sets and the known future values, i.e., the zeros padded at the end of the detail coefficient sets.

D. EXPERIMENTAL RESULTS

In order to have a better understanding of how the quality of the wavelet based prediction scheme is affected by either the available number of samples of the approximation coefficient set or the level of the wavelet decomposition, simulations were run for several signals. The parameters of these experiments, such as the number of past values used and the number of the predicted values or the level of the wavelet decomposition, were changed. For the results shown in the following figures, the signal of Example 3.1 was used. The plots shown are based on averaging results from six simulation runs.

At the beginning we attempted to predict groups of 4, 8, 12 and 16 consecutive future samples using 48, 96, 144, 192 and 240 signal samples. In order to do that, both linear prediction and a third level wavelet-based prediction were used. From Figure 3.7, we can see that the performance of both the linear prediction and the wavelet-based prediction is proportional to the number of the samples used. Furthermore, the performance of both is inversely proportional to the number of consecutive future values predicted. Therefore, the more future values there are to predict, the worse the performance becomes. This is due to the fact that the prediction of future values is based upon erroneous previously predicted values, and the error accumulates. The metric used in the figure is the mean squared prediction error (MSE):

$$D = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2 \tag{3.13}$$

where N is the number of signal samples, x_i are the original samples and $\hat{x}_i$ are recovered samples.

Figure 3.8 indicates that the wavelet-based prediction performs better than linear prediction as the number of the consecutive predicted future samples increases. This is expected since the error that accumulates each time another consecutive sample is predicted is larger in the case of linear prediction. The fact that the performance of the

wavelet-based prediction is worse than that of the linear prediction when the number of the previous samples used is small is because after the wavelet decomposition is performed, the number of samples in the approximation coefficient set available for linear prediction is very small. For example, for a 48-point signal, after a 3-level decomposition, there are only 6 values upon which the linear prediction would be based. As the number of previous samples increases, the wavelet method performs better than the direct method.

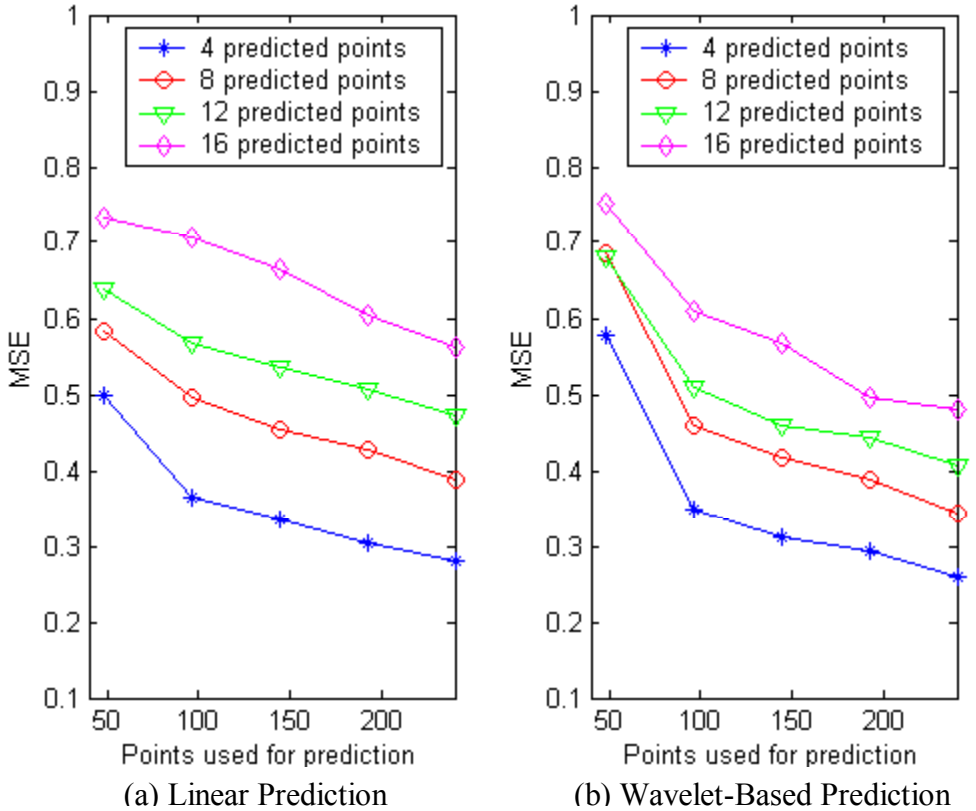


Figure 3.7. Mean Squared Prediction Error versus the Number of Signal Values Used for Prediction.

The performance of the wavelet-based prediction improves as the level of decomposition increases as shown in Figure 3.9. Again, there is a constraint based on the number of samples in the approximation coefficient set, which are available at each

decomposition level. For this reason, the error for the 3-level decomposition, shown in Figure 3.9, is larger than that of the 2-level decomposition when only 48 points are used.

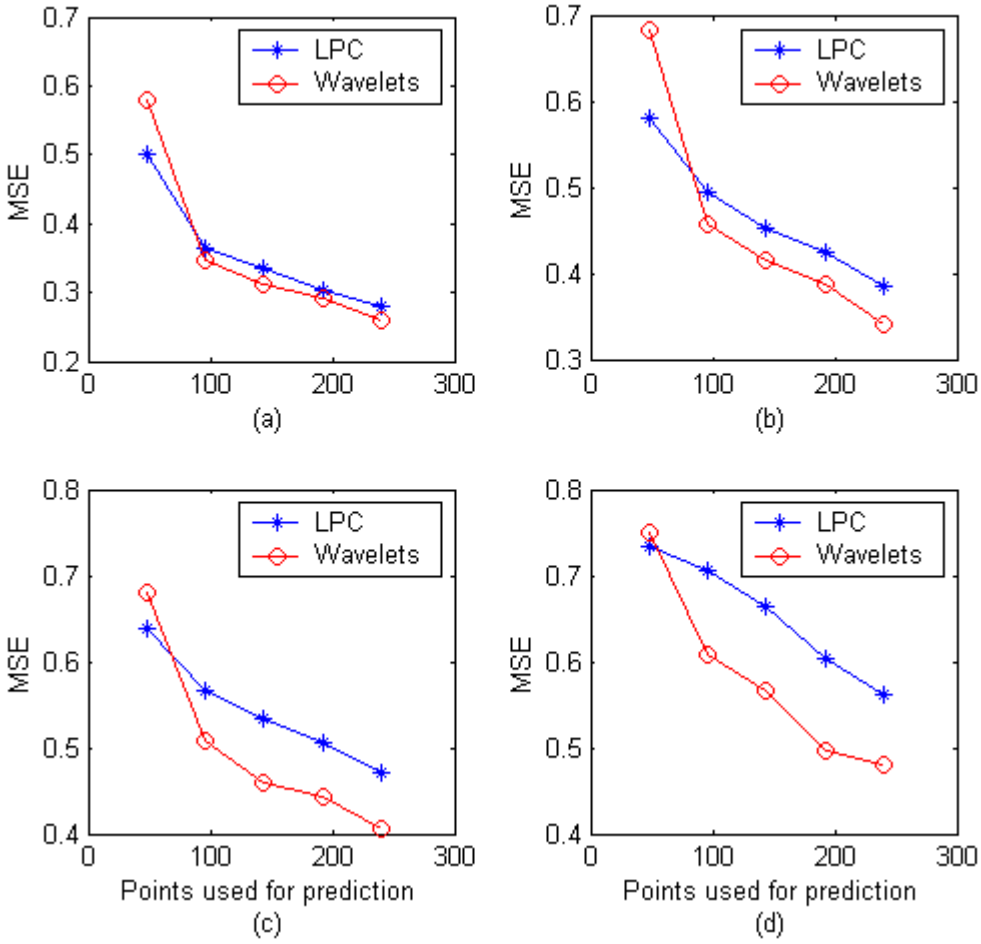


Figure 3.8. Performance Comparison between Direct Linear Prediction and Wavelet-Based Prediction for (a) 4 Future Values, (b) 8 Future Values, (c) 12 Future Values, and (d) 16 Future Values

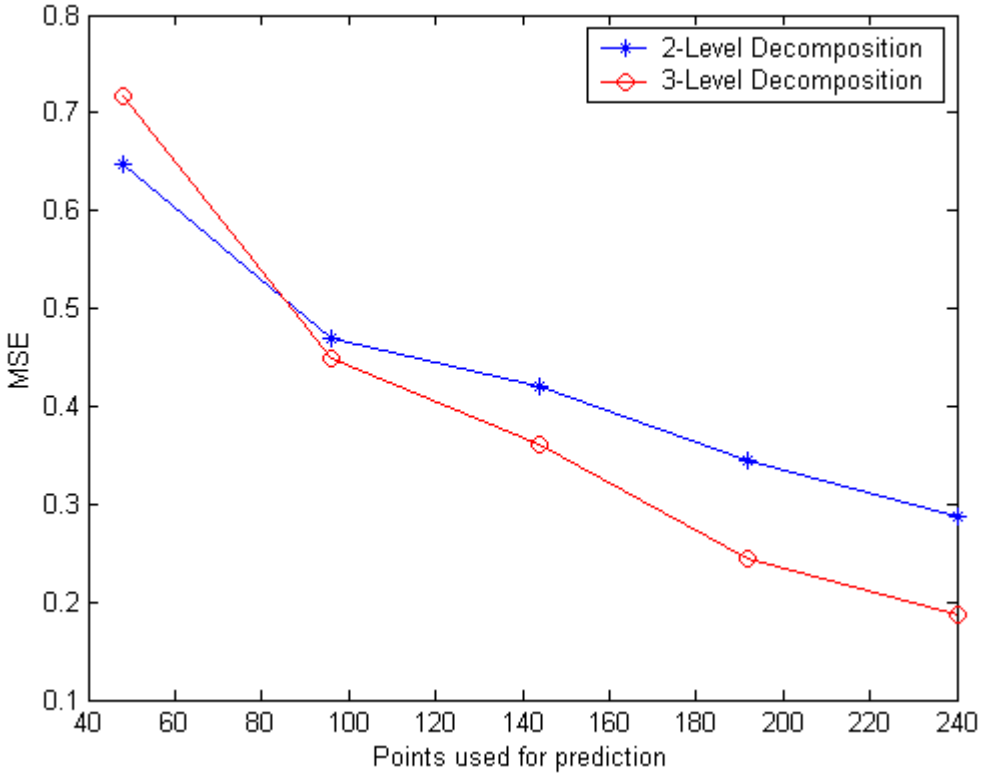


Figure 3.9. Mean Squared Prediction Error for Different Decomposition Levels.

E. SUMMARY

Linear prediction works well with highly correlated signal samples. However, when a signal consists of high frequencies, the performance of the linear prediction degrades. In this chapter, a method that excludes the high frequencies in a given signal by using wavelet decomposition was examined. First, the signal is wavelet decomposed to the desired resolution level. Next, the last sample of the highest-level approximation coefficients is predicted using linear prediction and then the detail coefficients at different resolution levels are zero-padded as necessary. Finally, the signal is reconstructed having all the original samples and a new predicted sample.

In the next chapter, a three-state Markov model will be developed. This model will be used to simulate a wireless channel and the losses that it introduces in a transmitted signal.

IV. WIRELESS CHANNEL MODEL

In this chapter a two-state Markov model of a wireless channel will be examined, and then a three-state Markov model will be developed. The purpose of this model is to simulate both single and burst-packet errors that a wireless channel introduces in a transmitted packet stream.

A. TWO-STATE MODEL

When data is transmitted through a mobile radio channel, absorption, reflection, diffraction and scattering are the cause of degradation of the received signal quality. There may be obstacles, such as buildings between the transmitter and the receiver, which cause further degradation in signal quality. Also, weather conditions may lead to burst errors, which in turn make the received signal quality unacceptable. A two-state Markov representation of the wireless channel, shown in Figure 4.1, can be used to model the preceding conditions [11], [12]. In the "Good" state, no losses occur while in the "Bad" state, the probability of error can be significant. The transition probabilities P_{ij} represent the probability of state transition from state i to state j.

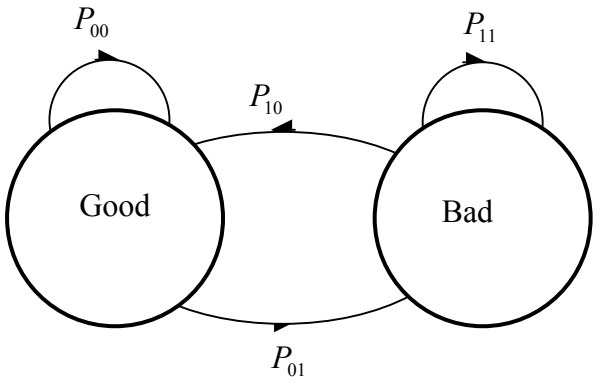


Figure 4.1. Transition Diagram for the Two-State Markov Model.

The transition matrix that describes the channel is given by:

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} \tag{4.1}$$

where P_{00} and P_{11} are the probabilities of the channel remaining in the same state as before, and the rest are the transition probabilities that the channel changes from a given state to another state. In (4.1), "0" indicates the "Good" state and "1" the "Bad" state. The steady state probabilities π_j of this model are given by [13]:

$$\pi_j = \sum_{i=0}^{1} \pi_i P_{ij}, \text{ for } j=0,1 \tag{4.2}$$

$$\sum_{i=0}^{1} \pi_i = 1 \tag{4.3}$$

Solving these equations yields:

$$\pi_0 = \frac{P_{10}}{P_{01} + P_{10}} \tag{4.4}$$

$$\pi_1 = \frac{P_{01}}{P_{01} + P_{10}} \tag{4.5}$$

From the principle of total probability, the probability of channel error can be expressed as:

$$\begin{aligned} P_e &= \Pr[\text{error} \mid \text{Good state}]\Pr[\text{Good state}] + \Pr[\text{error} \mid \text{Bad state}]\Pr[\text{Bad state}] \\ &= P_G\pi_0 + P_B\pi_1 \end{aligned} \tag{4.6}$$

where P_G and P_B are the probabilities of an error occurring in the "Good" and the "Bad" state, respectively. A Venn-diagram representation of P_e is shown in Figure 4.2.

Figure 4.3 shows plots of the probability of channel error, P_e, as a function of the probability of error in the "Bad" state, P_B, for theoretical and simulated two-state channels. The probability of error for the theoretical channel is given by (4.6) with $P_{00} = 0.8, P_{11} = 10^{-8}$ and various values of P_B. Using these same values, the channel is simulated in Matlab. The simulation results shown are based on averaging the results from six simulation runs.

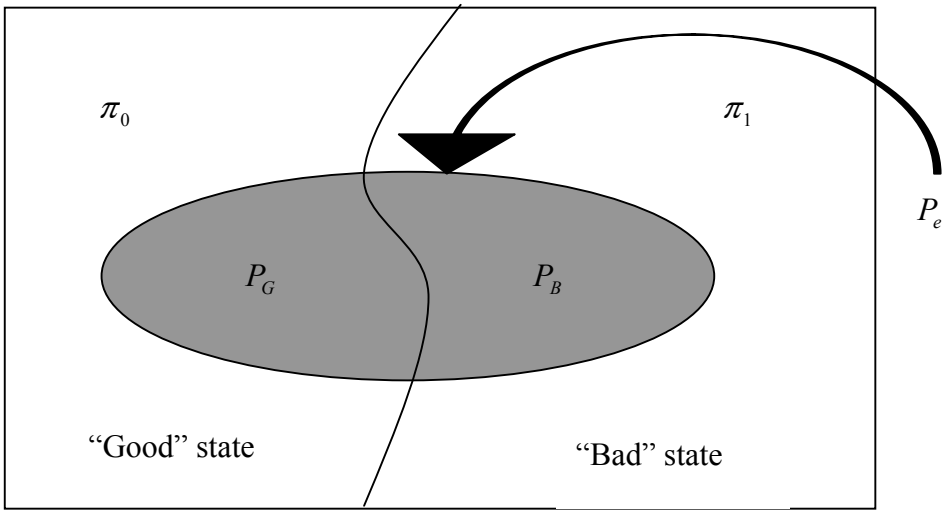


Figure 4.2. Average Error Probability of a Two-State Channel.

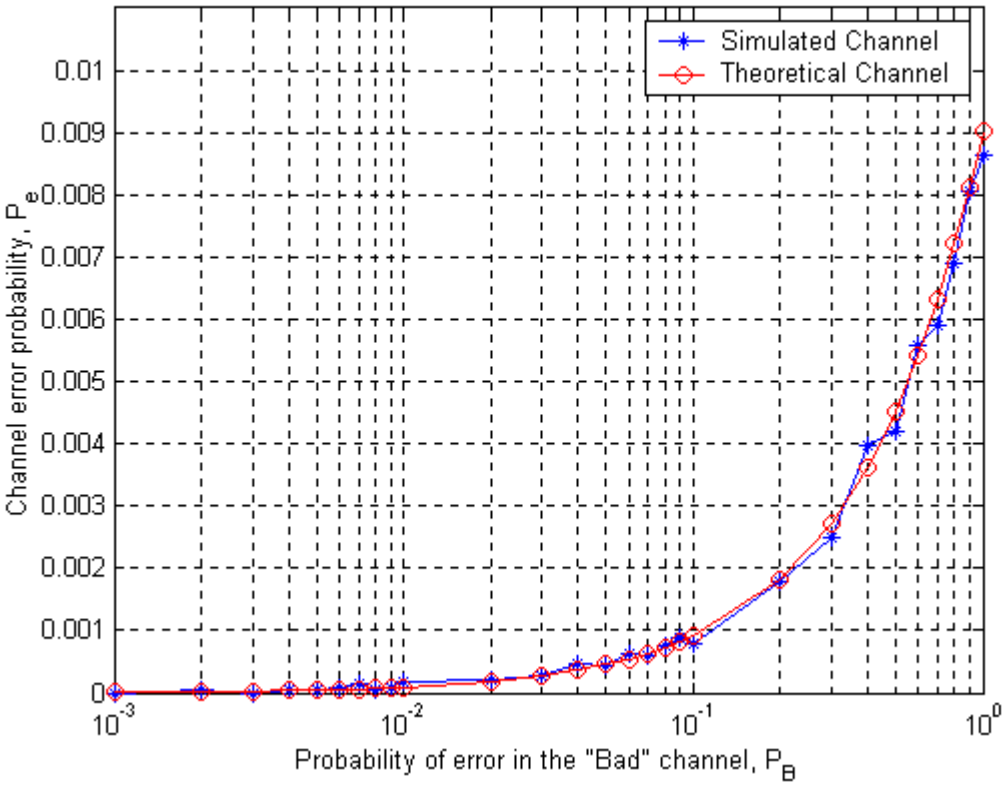


Figure 4.3. Probability of Channel Error P_e for Theoretical and Simulated Channels as a Function of the "Bad" state Probability, P_B.

B. THREE-STATE MODEL

In this thesis, the preceding two-state model is extended to a three-state Markov channel as shown in Figure 4.4. The three possible states of the channel are the "Good" state, where packets are dropped with probability P_G, the "Bad" state, where packets are dropped with probability P_B and the "Bursty" state, where packets are dropped with probability P_M. The channel transition matrix that characterizes the channel is given by:

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} \tag{4.7}$$

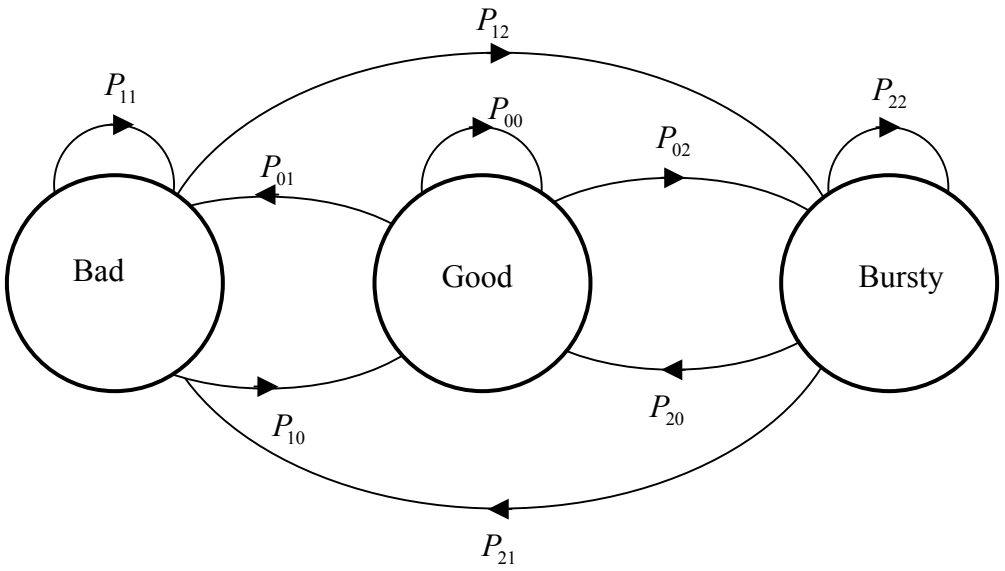


Figure 4.4. Transition Diagram for the Three-State Model.

The steady state probabilities are determined by extending (4.2) and (4.3) for $j = 0$, 1, 2 where 0, 1, 2 correspond to "Good", "Bad" and "Bursty" states, respectively. By solving these extended equations, we have the steady state equations:

$$\pi_0 = 1 - \pi_1 - \pi_2 \tag{4.8}$$

$$\pi_1 = \frac{-P_{01} - \pi_2 (P_{21} - P_{01})}{P_{11} - P_{01} - 1} \tag{4.9}$$

$$\pi_2 = \frac{P_{02}P_{12} - P_{02}P_{11} + P_{02}}{P_{21}(P_{02} - P_{12}) + P_{01}(P_{12} - P_{22} + 1) + (P_{11} - 1)(P_{22} - P_{02} - 1)} \tag{4.10}$$

The channel error probability is then given by:

$$\begin{aligned} P_e &= \Pr[\text{error} \mid \text{Good state}]\Pr[\text{Good state}] + \Pr[\text{error} \mid \text{Bad state}]\Pr[\text{Bad state}] \\ &\quad + \Pr[\text{error} \mid \text{Bursty state}]\Pr[\text{Bursty state}] \\ &= P_G\pi_0 + P_B\pi_1 +_2 P_M\pi \end{aligned} \tag{4.11}$$

where P_G, P_B and P_M are the probabilities of an error occurring in the "Good", the "Bad" and the "Bursty" state, respectively. A Venn-diagram representation of P_e is shown in Figure 4.5.

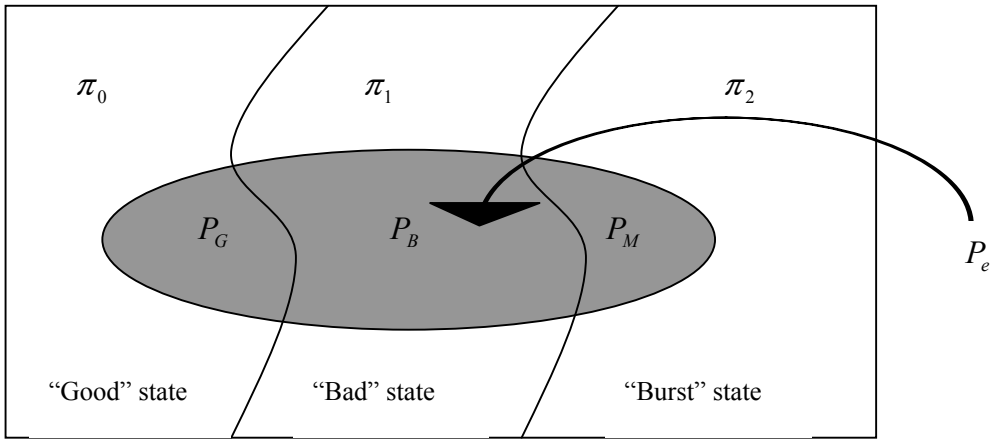


Figure 4.5. Average Error Probability of a Three-State Channel.

The actual channel that was used for the simulations, shown in Figure 4.6, was a simplified version of the channel shown in Figure 4.4. The difference between the two channels is that in Figure 4.4, a transition from any state to any state could be made regardless of the current state. On the other hand, in the simplified model of Figure 4.6, the state transitions are somewhat restrictive. Transitions could only occur between

adjacent states as illustrated in Figure 4.6. This is achieved by setting $P_{12} = P_{21} = 0$ in Figure 4.4. Consequently, the transition matrix becomes:

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & 0 \\ P_{20} & 0 & P_{22} \end{bmatrix} \tag{4.12}$$

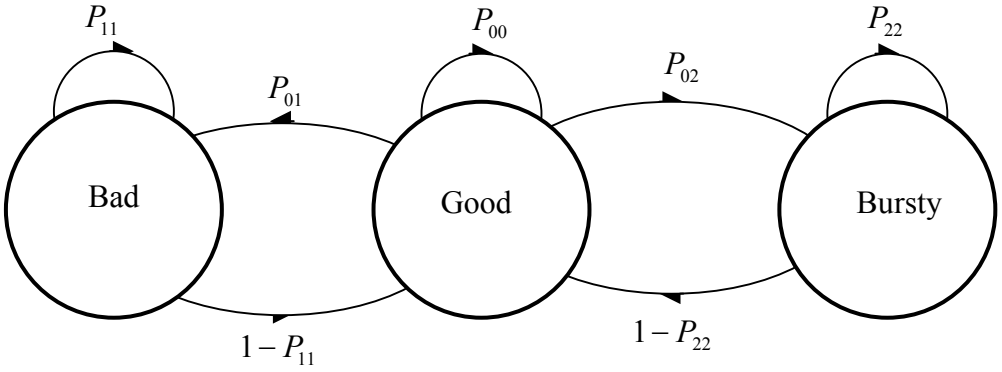


Figure 4.6. Transition Diagram for the Simplified Three-State Channel.

Equations (4.9) and (4.10) are correspondingly simplified to:

$$\pi_1 = \frac{-P_{01} + \pi_2 P_{01}}{P_{11} - P_{01} - 1} \tag{4.13}$$

$$\pi_2 = \frac{P_{02} - P_{02} P_{11}}{P_{01}(1 - P_{22}) + (P_{11} - 1)(P_{22} - P_{02} - 1)} \tag{4.14}$$

Additionally, in the simplified model, we assume that no transmission errors occur in the "Good" state; therefore, from (4.11) the channel error probability simplifies to:

$$P_e = P_B \pi_1 + P_M \pi_2 \tag{4.15}$$

Figure 4.7 shows plots of the probability of channel error, P_e, as a function of the probability of error in the "Bad" state, P_B, for theoretical and simulated three-state channels. The probability of error for the theoretical channel is given by (4.15) for three different sets of values for $P_{00}, P_{11}, P_{22}, P_{01}, P_{02}, P_M$. Using these same values, the channel

is simulated in Matlab. The simulation results shown are based on averaging the results from six simulation runs.

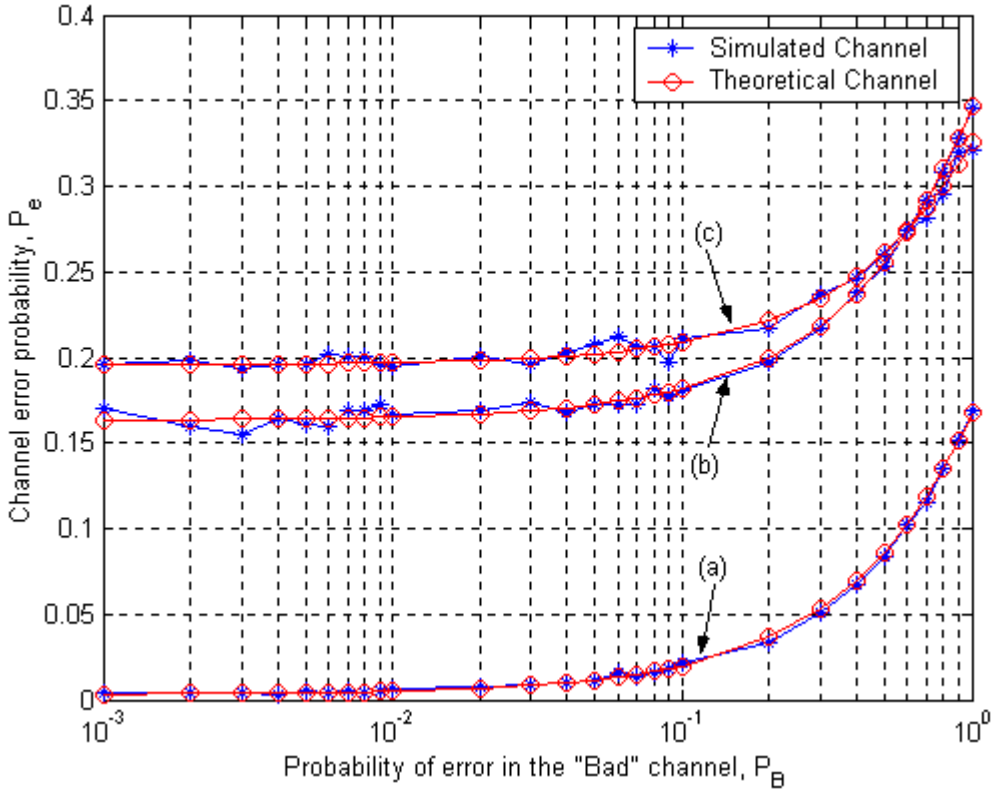


Figure 4.7. Probability of Channel Error P_e for Theoretical and Simulated Channel as a Function of the "Bad" state Probability, P_B. Curve (a) is obtained by using $P_{00} = 0.8,\ P_{11} = 10^{-8},\ P_{22} = 0.5,\ P_{01} = 0.198,\ P_{02} = 0.002,\ P_M = 1$. For curve (b), $P_{00} = 0.7,\ P_{11} = 10^{-3},\ P_{22} = 0.7,\ P_{01} = 0.2,\ P_{02} = 0.1,\ P_M = 0.9$. Curve (c) used $P_{00} = 0.6,\ P_{11} = 10^{-5},\ P_{22} = 0.7,\ P_{01} = 0.3,\ P_{02} = 0.1,\ P_M = 0.8$.

C. SUMMARY

In this chapter, a two-state Markov model was examined and a three-state Markov model was developed. The three possible states of data transmission are the "Good" state, where an error-free transmission occurs, the "Bad" state, where single packets are dropped and the "Bursty" state, where consecutive packets are dropped. This model will be used to simulate transmission of various kinds of data, including images and

uncompressed and compressed speech signals. After receiving the packet stream with missing packets, both linear prediction and the wavelet-based prediction will be applied in an effort to recover missing information, thereby concealing the effects of packet loss from the end user. This technique applies only to signals that are error tolerant, e.g., video, image and speech signals. Applications, such as file transfer, typically require error-free transmission. In such applications, other techniques, such as automatic repeat request (ARQ), are appropriate.

The next chapter presents simulation results of image and speech transmission over wireless channels using the technique developed in Chapter III and the three-state model developed in this chapter.

V. SIMULATION RESULTS

Simulation results of the image and voice packet stream transmission over a wireless channel implemented using a three-state Markov model are presented in this chapter. The received packet stream at the receiver (i.e., at the output of the channel) has missing packets due to channel errors (and possibly buffer overflows). Direct and wavelet-based linear prediction algorithms are applied on the received packet stream in order to recover the information in the lost packets. Since both image and speech signals are loss tolerant, the objective is to recover enough information to conceal the effects of loss in a way that the perceptual signal quality is acceptable to the end user. Results of both direct and wavelet-based prediction are compared for image and voice packet streams.

The simulation scheme can be illustrated as shown in Figure 5.1. The transmitter packetizes the image or speech signals and transmits a packet stream over the channel. The prediction algorithms reside in the receiver. All blocks are implemented in Matlab. Appendix A contains some of the Matlab code developed.

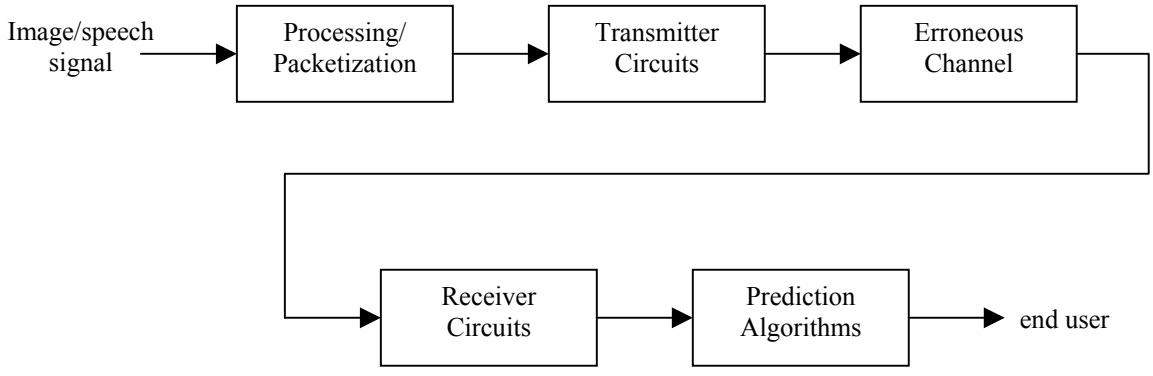


Figure 5.1. Block Diagram of the Simulation Scheme.

A. UNCOMPRESSED SPEECH SIGNAL

The speech bit stream was packetized into packets of size 48 bytes each. These packets are then transmitted through the channel as illustrated in Figure 5.1. The speech

signal is sampled at 8000 samples/sec with 8 bits/sample; thus each packet corresponds to a 6 msec time duration.

In the channel, whenever the transmission took place in the "Good" state, an error-free transmission was assumed; in the "Bad" state, a single packet was lost; and in the "Bursty" state consecutive packets were lost. The transition probabilities chosen for the simulations regulated the total number of consecutive packets that were dropped.

In order to recover the lost packets, both the direct and the wavelet-based linear prediction were applied. The modified-covariance prediction algorithm with a filter order of 18 was used for direct and wavelet-based methods. Through experimentation, a filter order of 18 was found to provide, on average, the best results. For the wavelet approach, a 2-level decomposition using Haar wavelets is implemented.

Figure 5.2 shows the plot of signal distortion versus packet loss probability P_B. The plots are based on averaging results from six simulation runs. The distortion is measured using:

$$D = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2 \tag{5.1}$$

where N is the number of speech samples, x_i are the original samples and $\hat{x}_i$ are the recovered samples. The three curves in the figure indicate distortion in the received signal (dotted line), distortion in the signal with linear prediction based error recovery (dashed line) and distortion with the wavelet approach (solid line). As packet loss increases, the linear prediction method performs worse than the received signal, i.e., it seems to introduce additional distortion for the range of packet loss probability used in this simulation. Better signal quality is observed by the author for the wavelet method through listening tests of sample speech signals.

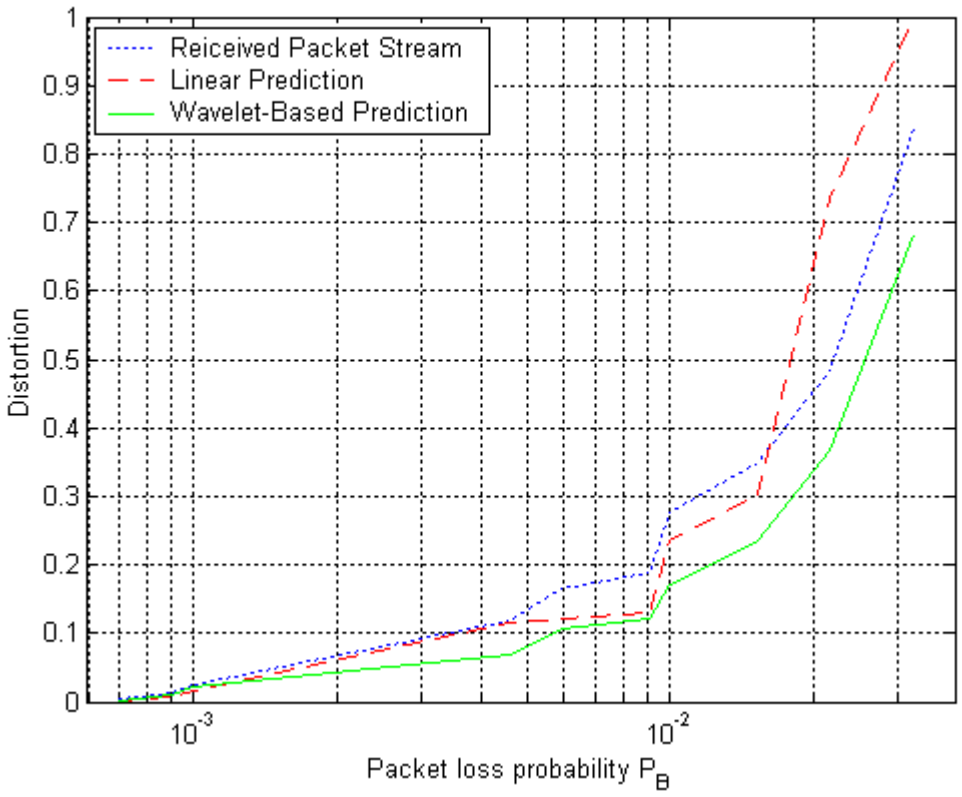


Figure 5.2. Distortion versus Packet Loss Probability for Speech Packet Stream.

B. CELP CODED SPEECH SIGNAL TRANSMISSION

In many applications, compression of speech signals is needed. For example, in digital cellular technology, where the bandwidth is limited, compression allows more users to share the system. The digital speech signals are typically sampled at a rate of 8000 samples/sec and 8 bits/sample. Code Excited Linear Prediction (CELP) is one of the widely used methods for speech compression. This synthesis-by-analysis coder essentially consists of a stochastic codebook, an adaptive codebook, and a linear prediction filter, as shown in Figure 5.3. The adaptive and stochastic codebooks provide the excitation that is driven through the linear prediction filter for reconstructing the original speech [3]. By comparing the synthesized speech with the original speech, the excitation that produces the minimum error is chosen.

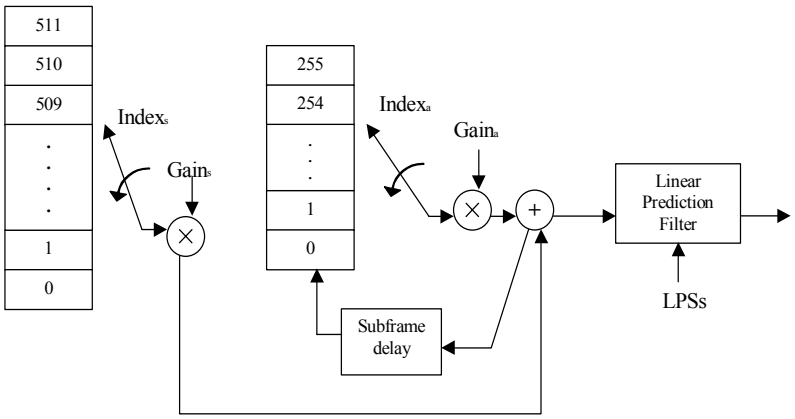


Figure 5.3. Basic CELP Coder.

Compression is achieved by transmitting the parameters of the prediction filter and the codebook indices and gains. The filter parameters are represented using the line spectrum pairs (LSP), which are obtained by transformation of the filter coefficients [3]. In CELP, the speech is fragmented into 30-ms frames, and the frames are further divided into four subframes, each 7.5 ms long. Each frame is coded into 144 bits; thus CELP provides an overall bit rate of 4.8 kbps. The manner in which these bits are allocated is shown in Table 5.1. Frames of 144 bits in CELP coded speech contain 10 line spectrum pair values (30-ms long), and four sets of codebook indices and gains (each 7.5-ms long).

Parameters	**Number of Bits**
Line Spectrum Pair	34
Pitch Prediction Filter	48
Codebook Indices	36
Gains	20
Synchronization	1
Forward Error Correction	4
Future Expansion	1
Total	144

Table 5.1. Bit Allocation in CELP.

Each CELP frame is packetized into one packet and transmitted over the simulated wireless channel (see Figure 5.1). When the transmission takes place in the "Good" sate, no packets are dropped; when the erroneous channel states are used, single or consecutive packets are dropped with a probability P_B or P_M, respectively.

In a lost CELP frame, the following parameters are estimated through direct and wavelet-based prediction:

- 10 line spectrum pair coefficients (30 ms)
- 4 stochastic codebook indices, one in each subframe (7.5 ms)
- 4 stochastic gains, one in each subframe
- 4 adaptive codebook indices, one in each subframe
- 4 adaptive gains, one in each subframe

A prediction filter of order five was used for both direct and wavelet-based methods. Parameter values from nine previous packets were used, leading to a delay of 270 msec. Using additional past packets would lead to higher delay, which may not be acceptable for real-time applications.

Figure 5.4 shows plots of distortion versus packet loss probability P_B of the "Bad" state for the received signal without error recovery (dotted line), direct prediction method (dashed line) and wavelet-based method (solid line). The wavelet-based method consistently produced better performance than the direct method.

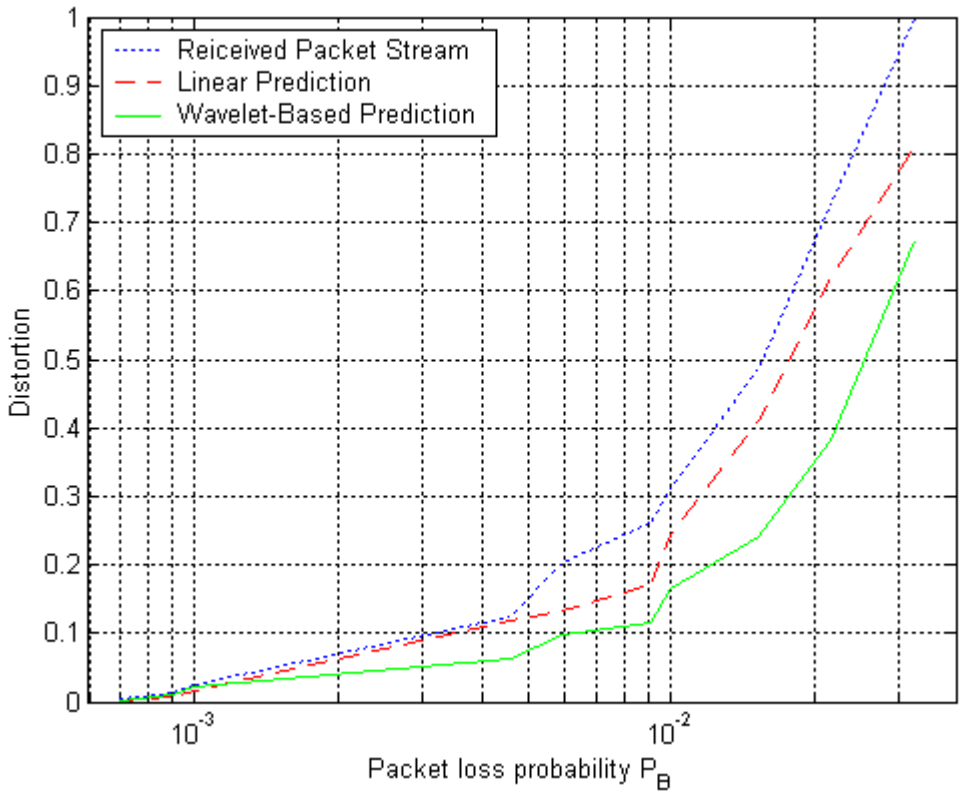


Figure 5.4. Distortion versus Packet Loss Probability for CELP-Coded Speech Packet Stream.

C. TRANSMISSION OF IMAGE DATA

The given image was packetized into 8×8 blocks and then transmitted through the three-state channel as illustrated in Figure 5.1. In the channel, whenever the transmission took place in the "Good" state, an error-free transmission was assumed; in the "Bad" state, an 8×8 block was lost; and in the "Bursty" state consecutive packets, forming 16×16 macroblocks, were lost. A block and a macroblock are illustrated in Figure 5.5. The transition probabilities chosen for the simulations regulated the total number of the consecutive packets that were lost.

In order to recover the lost packets, both the direct and the wavelet-based linear prediction were applied. The modified-covariance prediction algorithm with a filter order of 18 was used for direct and the wavelet-based methods. Since image transmission is not a case of real-time transmission, there is flexibility concerning the number of previous samples that can be used. In the simulations, 12 previous blocks or 96 previous data

samples were used. For the wavelet approach, a 3-level decomposition using Haar wavelets was implemented.

Figure 5.6 shows plots of peak signal-to-noise ratio (pSNR) versus packet loss probability P_B. The plots are based on averaging results from six simulation runs. The peak signal-to-noise ratio, in dB, is measured using:

$$pSNR_{dB} = 10\log_{10}\left(\frac{255^2}{D}\right) \tag{5.2}$$

where D is the mean squared error between the original pixel values x_{ij} and the recovered pixel values $\hat{x}_{ij}$ given by:

$$D = \frac{1}{MN}\sum_{j=0}^{N-1}\sum_{i=0}^{M-1}(x_{ij} - \hat{x}_{ij})^2$$

and M and N are the number of rows and columns, respectively, of the transmitted image.

The three curves in Figure 5.6 indicate the peak signal-to-noise ratio of the received image (dotted line), the image with linear prediction based error recovery (dashed line) and the wavelet approach (solid line). The wavelet-based method performs better than direct linear prediction throughout the loss probability range used in the simulations. Appendix B contains additional results on images.

In Figure 5.7, the original image, the received image without error recovery, the recovered image using direct linear prediction and the recovered image using the wavelet-based scheme are shown.

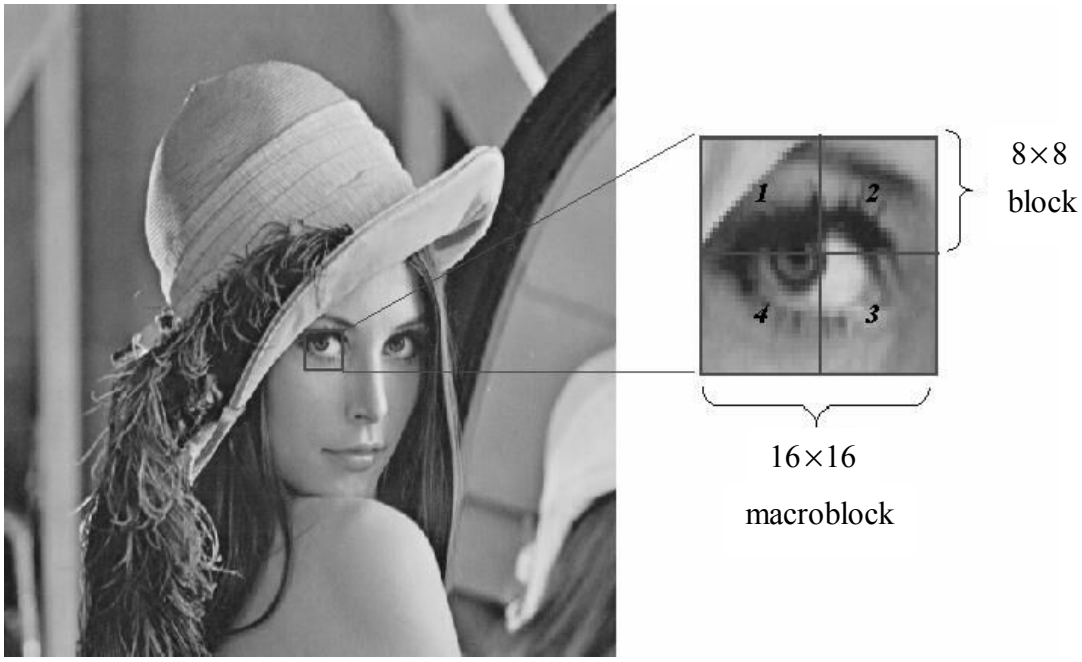


Figure 5.5. Blocks and Macroblocks in an Image.

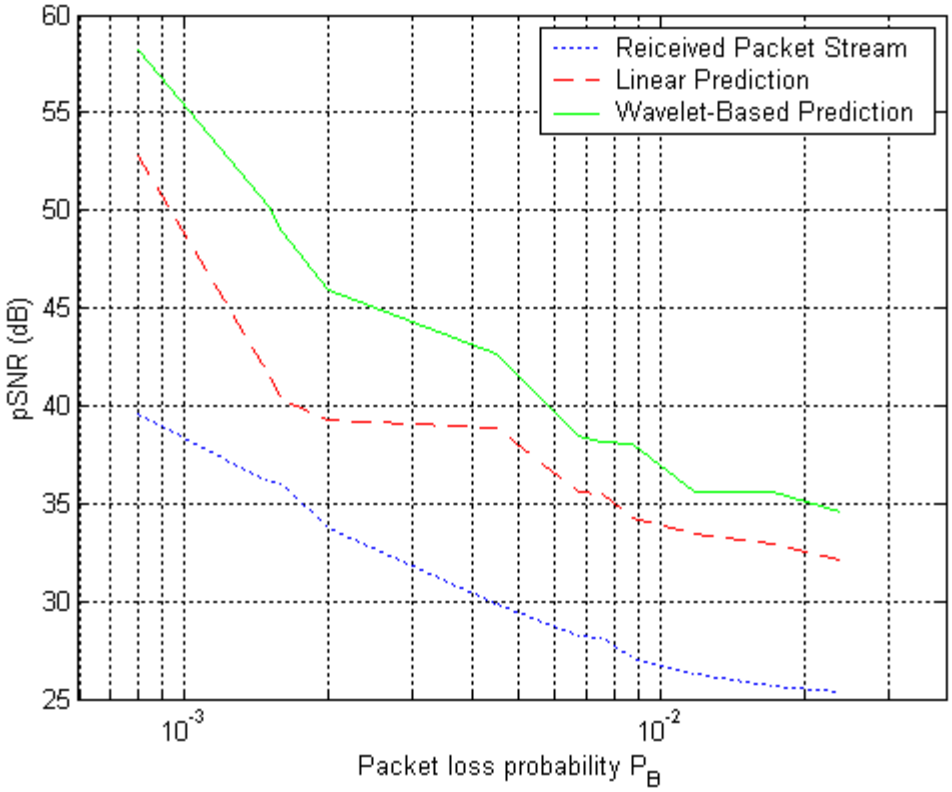


Figure 5.6. Signal-to-Noise Ratio versus Packet Loss Probability P_B for Image Packet Stream.

Figure 5.7. (a) Original Image, (b) Image without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet Based Scheme.

D. WAVELET-BASED PREDICTION WITH DAUBECHIES' WAVELETS

Using the simulation scheme shown in Figure 5.1, the wavelet-based method was successfully extended to Daubechies wavelets. Exploiting properties such as compact support, averaging, orthogonality, and regularity [15], several wavelets of the Daubechies family were successfully applied to implement wavelet-based prediction. For the following results, Daubechies wavelet "db10," the scale and wavelet function of which are shown in Figure 5.8, was used.

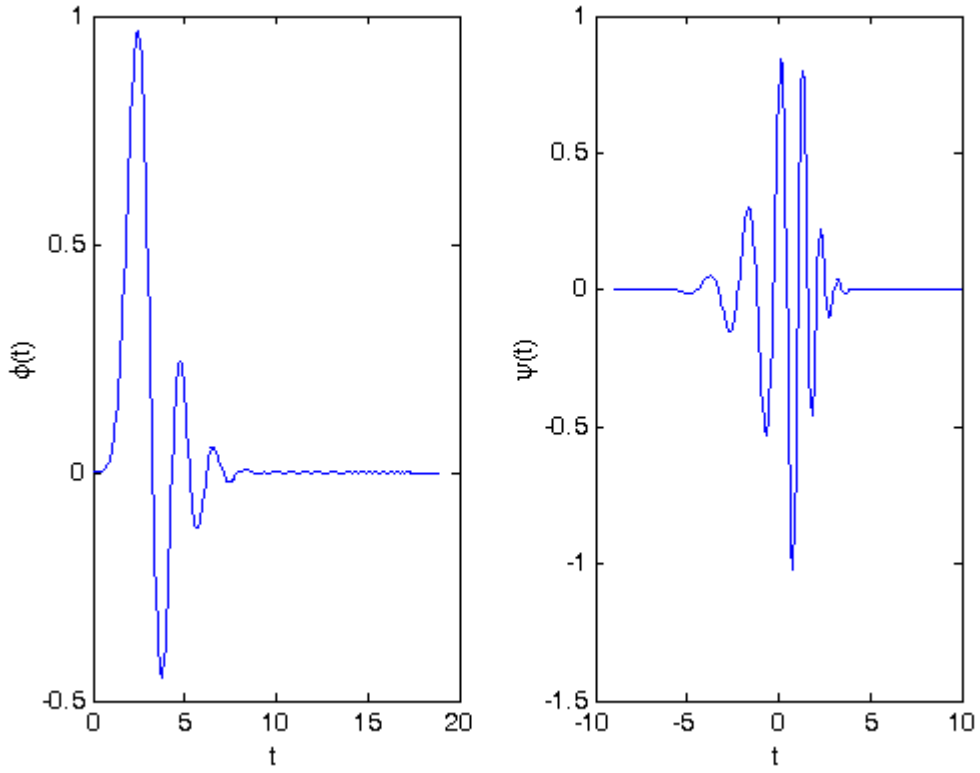


Figure 5.8. Scaling and Wavelet Function of Daubechies Wavelet "dB10".

The three curves in Figure 5.9 indicate peak signal-to-noise ratio of the received image (dotted line), the image with linear prediction based error recovery (dashed line) and the wavelet approach (solid line). The wavelet-based method performed better than direct linear prediction over the loss probability range used in the simulations.

In Figure 5.10, the original image, the received image without error recovery, the recovered image using direct linear prediction and the recovered image using the wavelet-based prediction are shown. The performance of the wavelet-based prediction using Haar wavelet is better than the one using Daubechies wavelets.

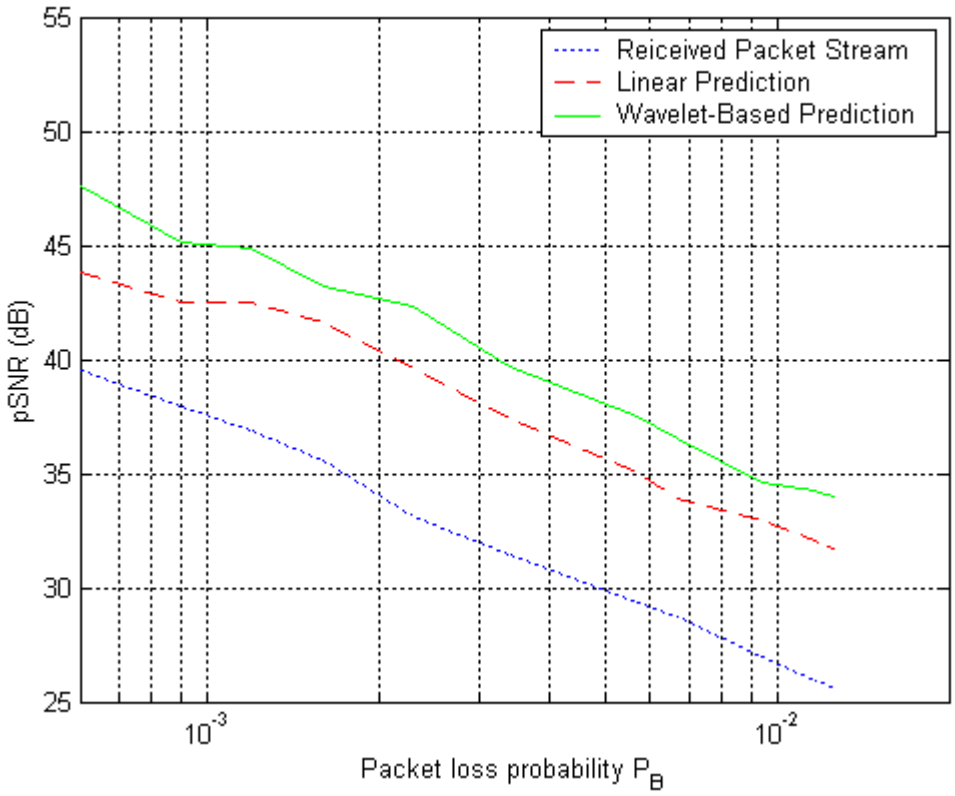


Figure 5.9. Signal-to-Noise Ratio versus Packet Loss Probability P_B for Image Packet Stream.

(a) (b)

(c) (d)

Figure 5.10. (a) Original Image, (b) Image without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet Based Scheme.

E. DENOISING

When noise is present in a signal, denoising is used to reduce the effects of noise in wavelet-based analysis. After the approximation and detail coefficients have been computed, they are subjected to a threshold, thus removing all coefficients below a certain value. The energy of a signal is typically concentrated in a small number of coefficients. These coefficients have larger values than the ones due to a noise or other distortion that spreads its energy over a large number of coefficients [9]. Removing the coefficients that do not exceed the set threshold and reconstructing the signal from the remaining coefficients eliminate much of the noise. In this process, some information will be lost, but the overall signal-to-noise ratio will be improved.

There are two thresholding options called hard and soft thresholding. In this work hard thresholding was used, as given by [9]:

$$d_m(x) = \begin{cases} x, & |x| > T \\ 0, & |x| \le T \end{cases}$$

where d_m is the thresholding function, x is the value of the coefficient that is examined and T is the threshold value, given by [9]:

$$T = \sqrt{2\log_2(n\log_2(n))}$$

where n is the length of the signal.

1. Implementation using Prediction and Denoising

The given image was packetized into 8×8 blocks and then transmitted through the channel (see Figure 5.1) as described in the previous section. In order to conceal the packet-loss effects, two different approaches were attempted. In the first approach, after the image was received, shown in Figure 5.11 (a), denoising was applied to it (predenoising), shown in Figure 5.11 (b), and then the missing blocks were predicted using both the direct and the wavelet-based scheme (Figures 5.11 (c) and (d)). In the second approach, after the image was received, shown in Figures 5.12 (a) and (d), the missing blocks were predicted using both the direct and the wavelet-based scheme, shown in Figures 5.12 (b) and (e), and then denoising was performed (postdenoising), see

Figures 5.12 (c) and (g). In both cases, the plots are based on averaging results from six simulation runs.

As shown in Table 5.2, predenoising performs marginally better than postdenoising, and improves the peak signal-to-noise ratio by approximately 1.5-2 dB. The wavelet-based prediction provides just a fraction of an improvement over linear prediction; the main reason of quality degradation is the additive noise and not the lost packets.

			Image with Missing Blocks	LP Method	Wavelet-based Method
PREDENOISING	**pSNR (dB)**	**Before Denoising**	20.1251	----------	----------
		After Denoising	20.9728	21.6971	21.8332
POSTDENOISING		**Before Denoising**	20.1251	20.7707	20.8693
		After Denoising	----------	21.5357	21.6500

Table 5.2. Comparison between Predenoising and Postdenoising.

(a) **(b)**

(c) **(d)**

Figure 5.11. Predenoising: (a) Noisy Image Without Error Recovery, (b) Denoised Image Without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet-Based Scheme.

Figure 5.12. Postdenoising: (a) Noisy Image Without Error Recovery, (b) Recovered Image Using Linear Prediction, (c) Denoised Image, (d) Noisy Image Without Error Recovery, (e) Recovered Image using Wavelet-Based Prediction, (f) Denoised Image.

F. SUMMARY

In this chapter, simulation of speech and image packet stream transmission is implemented in order to compare the performance of the direct and wavelet-based linear prediction. After receiving the packet stream with the missing packets, both methods were applied in an effort to conceal the effects of packet loss from the end user. The wavelet-based prediction was found to perform consistently better than direct linear

prediction. Application of denoising to error recovery only provided marginal improvement in peak signal-to-noise ratio for image packet streams.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

The objective of this thesis was to investigate a new prediction scheme that utilizes the wavelet decomposition in linear prediction and compares its performance with that of direct linear prediction. The two methods are used for concealing packet-loss effects when transmitting packet streams over wireless channels and their performance compared through simulation results based on image and speech packet streams.

A. SIGNIFICANT RESULTS

From the simulation results shown in Chapter V, the superior performance of the wavelet-based approach over direct linear prediction was demonstrated. The wavelet method results in a smaller prediction error than the direct linear prediction. This may be attributed to the fact that prediction in the wavelet method is carried out on signal components of low frequencies. The role that the prediction error plays in the estimation of the $\hat{x}[n+1]$ sample is weighted down by using the known past values of the approximation and detail coefficient sets and the known future values of the detail coefficients, i.e., the zeros padded at the end of the detail coefficient sets. The wavelet prediction method has been successfully extended to Daubechies wavelets. Denoising of wavelet coefficients only provided marginal improvement in signal quality.

A three-state Markov model that simulates a wireless channel was developed. From the plots shown in Chapter IV, we remark that the model provides an accurate representation of a lossy wireless channel.

B. FUTURE WORK

Two families of wavelets were used in this thesis: Haar wavelet and Daubechies. A more in-depth investigation is required in order to generalize the proposed method to other wavelets.

Although the wavelet-based prediction was experimentally shown to perform better than linear prediction, rigorous mathematical support was not provided in this thesis. A future effort may focus on developing a rigorous mathematical basis for this work.

The prediction method did not work well when packet losses caused a loss of information in the high frequency region of the bitstream; for example, in regions where an abrupt change in pixel values occurs in an image. A possible solution may consider first determining where this change takes place and then using forward prediction for the part of the block where the values are highly correlated and backward prediction for the rest of the block, as shown in Figure 6.1. This way the known data to the left and to the right of the missing block would be better exploited.

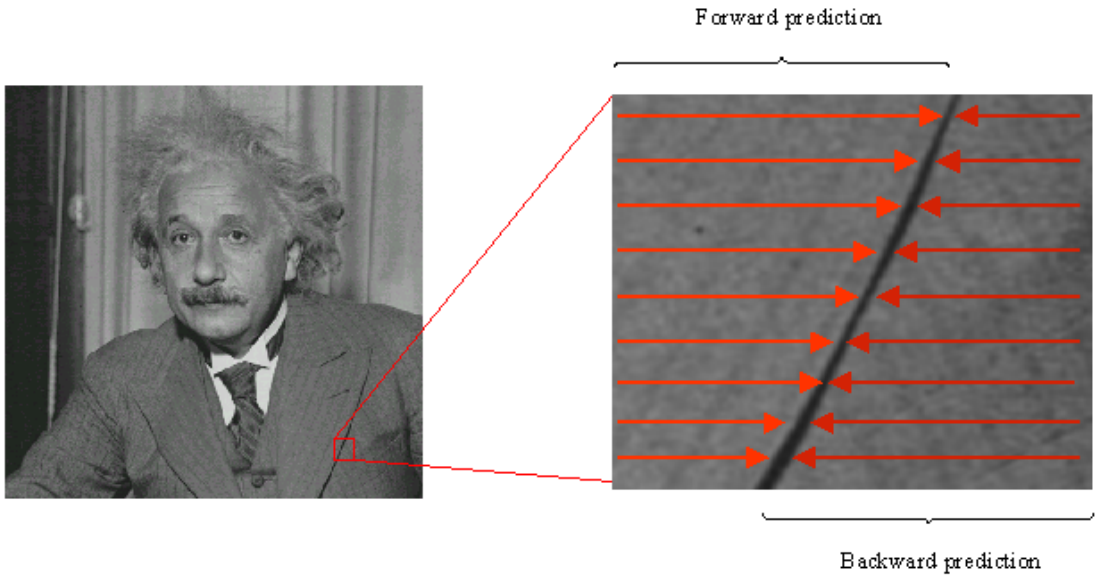
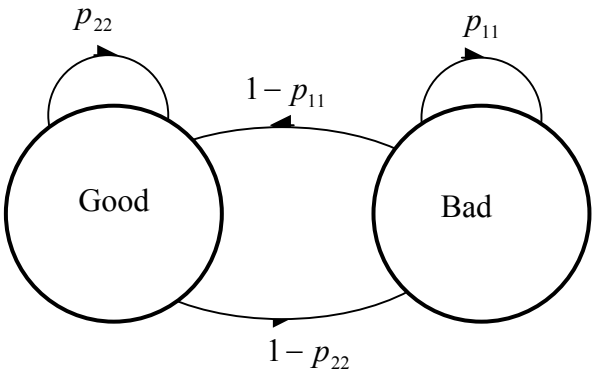


Figure 6.1 Application of Forward and Backward Linear Prediction in the High Frequency Region of the Image.

APPENDIX A. MATLAB CODES

A. BURST NOISE CHANNEL

1. Two-State Channel

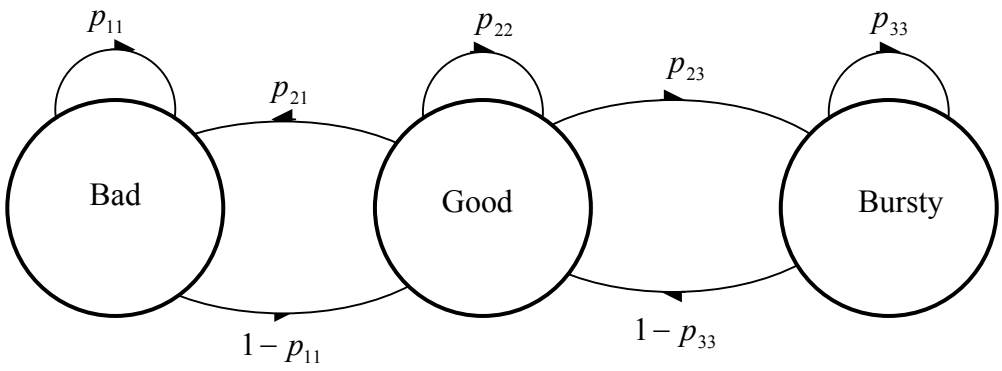



```
%The following code simulates a channel that alternates between two states
%according to a binary Markov process.
% p11 = prob. of staying in the bad channel
% p22 = prob. of staying in the good channel
% Pe1 = prob. of error in the bad channel
% Pe2 = prob. of error in the good channel.

clear
clc
S=[];S1=[];S2=[];

s=[];s2=[];s4=[];s5=[];s6=[];
for Pe1=0:0.001:.01
   Pe1
   s3=[];
   s=[s,Pe1];
   TOTAL_lost_packet_rate=[];
```

```
for times=1:100
   times
   total_packets=10000;
   p11=10^-2;
   p22=0.991;

   mu=[0.1,0.9];                        % Initial distribution
   P=[[p11 1-p11];[1-p22 p22]];         % Transition matrix
   n=total_packets;                     % Number of time steps to take
   state=zeros(1,n);
   t=1:n;                               % Time indices
   state(1)=rando(mu);          % generate first x value (time 0, not time 1),if state(1)=2
                                        % I am in the Bad channel else in the Good
   lost_blocks=[];

   if state(1)==1
      lost_blocks=[lost_blocks,1];
   end
   for i=1:n-1
      state(i+1) = rando(P(state(i),:));      % Pr(same state or transition)
      if state(i+1)= =1
         [error]=error_probab(Pe1);         % if it goes to the bad state....
         if error= =1
            lost_blocks=[lost_blocks,i+1];
         end
      end
   end
   plot(t, state, '*');
   axis([1 n 0 (length(mu)+1)]);
   lost_blocks  ;
```

```
    total=size(lost_blocks);
    how_many=total(2);  %how many blocks are lost
    s3=[s3,how_many];
    lost_packet_rate=how_many/total_packets;
    TOTAL_lost_packet_rate=[TOTAL_lost_packet_rate,lost_packet_rate];
  end                 % of times=1:10

  TOTAL_lost_packet_rate_2=mean(TOTAL_lost_packet_rate);
  s5=[s5,TOTAL_lost_packet_rate_2];
  avg=mean(s3);
  s2=[s2,avg];

  % THEORETICAL PROBABILITY

  Pb_theor=((1-p22)/((1-p22)+(1-p11)))*Pe1;
  s6=[s6,Pb_theor];
  how_many_theor=Pb_theor*total_packets;
  s4=[s4,how_many_theor];

end % of Pe1=0.1:0.1:1

S=[S,s];S1=[S1,s5];S2=[S2,s6];
figure(27)
semilogx(S,S1,'b*-',S,S2,'ro-'),grid   % x axis: function of Pe1,y axis: Average error
probability
legend('Simulated Channel','Theoretical Channel')
xlabel('Probability of error in the "bad" channel - P_B')
ylabel('Average error probability - (P_e)_{avg}')
```

2. Three-State Channel




```
%The following code simulates a channel that alternates between two states
%according to a binary Markov process.
% p11 = prob. of staying in the bad channel
% p22 = prob. of staying in the good channel
% p21 = prob. of transition from the good to the bad channel
% p23 = prob. of transition from the good to the bursty channel
% p33 = prob. of staying in the bursty channel
% Pe1 = prob. of error in the bad channel
% Pe2 = 0 ==> there are NO errors in the good channel.
% Pe3 = prob. of error in the bursty channel
% p_g = steady state probability of being in the good channel
% p_b = steady state probability of being in the bad channel
% p_m = steady state probability of being in the burst channel

% LTJG Anastasios Garantziotis (Hellenic Navy)
% April 2002

clear
clc

S=[];S1=[];S2=[];
```

```
s=[];s2=[];s4=[];s5=[];s6=[];

for Pe1=0:0.001:.01
   Pe1
   s3=[];
s=[s,Pe1];
TOTAL_lost_packet_rate=[];

   for times=1:10
      times
total_packets=10000;
block_counter=0;       %********************
help=[];               %********************
COUNTER=[];            %********************
burst=0;               %********************
p11=10^-8;
p22=0.8;
p33=0.5;
p21=0.198;
p23=0.002;
Pe3=1;

mu=[0.85,0.1,0.05];%initial distribution
P=[[p22 p21 p23];[1-p11 p11 0];[1-p33 0 p33]];% transition matrix

n=total_packets;% number of time steps to take
state=zeros(1,n); % clear out any old values
t=1:n;         % time indices
```

```
state(1)=rando(mu); % generate first x value (time 0, not time 1),if state(1)=2, I am in the
                    %Bad channel else in the Good
lost_blocks=[];

if state(1)==2 | state(1)==3
   lost_blocks=[lost_blocks,1];
end
 for i=1:n-1,
   state(i+1) = rando(P(state(i),:));      % check whether it will stay in the same state or it
                                           %will go to the other state
   if state(i+1)==2
      block_counter=block_counter+1;              % SINGLE ERRORS
      [error]=error_probab(Pe1);             % if it goes to the bad state....
      if error==1                    % then the probability that the packet will be lost is.....
         lost_blocks=[lost_blocks,i+1];
      end
   elseif state(i+1)= =3
      help=[help,i+1];
      [error]=error_probab(Pe3);             % if it goes to the bursty state....
      if error= =1                    % then the probability that the packet will be lost is.....
         lost_blocks=[lost_blocks,i+1];
      end
   end
end


% ******* This part gives me the number of single and  ***********
% *********              burst errors                          ***********
for i=2:max(size(help))
```

```
   counter=help(i)-help(i-1);
   COUNTER=[COUNTER,counter];
end
for i=2:max(size(COUNTER))
   if (COUNTER(i)==1) & (COUNTER(i-1)~=1)
      burst=burst+1;
   end
end
Single_blocks__bursts=[block_counter,burst]

total=size(lost_blocks);
how_many=total(2);  %how many blocks are lost
s3=[s3,how_many];

lost_packet_rate=how_many/total_packets;
TOTAL_lost_packet_rate=[TOTAL_lost_packet_rate,lost_packet_rate];
%TOTAL_lost_packet_rate;

end % of times=1:10

TOTAL_lost_packet_rate_2=mean(TOTAL_lost_packet_rate);
s5=[s5,TOTAL_lost_packet_rate_2];
avg=mean(s3);
s2=[s2,avg];

% THEORETICAL PROBABILITY

q1=p23-(p23*p11);
q2=p21*(1-p33);
q3=(p11-1)*(p33-p23-1);
```

```
p_m=q1/(q2+q3);
p_b=(-p21+(p21*p_m))/(p11-p21-1);
p_g=1-p_b-p_m;                % this is not needed since there are no errors in the good
state
Pb_theor=p_b*Pe1+p_m*Pe3;
s6=[s6,Pb_theor];
how_many_theor=Pb_theor*total_packets;
s4=[s4,how_many_theor];

end % of Pe1=0.1:0.1:1

S=[S,s];S1=[S1,s5];S2=[S2,s6];
figure(27)
semilogx(S,S1,'b*-',S,S2,'ro-'),grid   % x axis: function of Pe1,y axis: Average error
probability
legend('Simulated Channel','Theoretical Channel')
xlabel('Probability of error in the "bad" channel - P_B')
ylabel('Average error probability - (P_e)_{avg}')
```

B. PROPOSED SCHEME

1. Wavelet-Based Prediction Algorithm

```
function [photo4,photo5,photo99]=photo_pred(photo,x_start,x_end,y_start,y_end,Points)

% INPUT
% photo=Original Photo
% x_start=Start of missing block for rows
% x_end  =End of missing block for rows
% y_start=Start of missing block for columns
% y_end  =End of missing block for columns
% OUTPUT
% photo_4=prediction of the missing block using LPC
```

```
% photo_5=prediction of the missing block using Wavelets

x_zero=max(size(x_start:x_end));
y_zero=max(size(y_start:y_end));
photo3=photo;
photo3(x_start:x_end,y_start:y_end)=zeros(x_zero,y_zero);
% ********************************************
%      Determining if the Missing Block                *
%        is Low or High Frequency                      *
% ********************************************
Mean_of_Block=(mean(photo(x_start-y_zero:x_end-y_zero,y_start-y_zero:y_end-
y_zero)'))'; % For each row
Total_mean=mean(Mean_of_Block);
Difference=Mean_of_Block-Total_mean;

% ********************************************
%      Prediction using LPC and Wavelets               *
% ********************************************
%photo4=photo3;
%photo5=photo3;
photo4=[];
photo5=[];
photo99=[];

% ********************************************
%    Prediction Using Modified Covariance              *
% ********************************************
filter=18;
%count_x=x_start;
count_x=1;
```

```
for i=1:x_zero
   Row=[];
   Prediction1=photo(count_x,y_start-Points:y_start);
   Prediction1=double(Prediction1);
   for j=1:y_zero    % So many predictions per row
      [a1,S1,R1]=mcovar(Prediction1',filter);
      estim1=-a1(2:filter+1)'*Prediction1(end:-1:end+1-filter)';
      Prediction1=[Prediction1(2:end),estim1];
      Row=[Row,estim1];
   end
%    photo4(count_x,y_start:y_end)=Row;
   photo4(count_x,1:y_zero)=Row;
   count_x=count_x+1;
end
photo0=double(photo);
photo44=double(photo4);
Error=[(photo0(x_start:x_end,y_start:y_end)-photo44).^2];
Error=(sum(Error'))';
[Min_Error,Min_Row_Error]=min(Error);
Row_Error_LPC=Error;

% ************************************
%      Prediction Using Wavelets                    *
% ************************************

count_x=1;
for k=1:x_zero
   Prediction2=photo(count_x,y_start-Points:y_start);
   Prediction2=double(Prediction2);
   wav_row=[];
```

```
   wav_row99=[];
   total=y_zero/4;       % How many times I will perform the wavelet prediction
   for i=1:total         % So many times I will predict in groups of 4
      for j=1:4
         wav_final=[];
         [pred_value]=wav_pred3(Prediction2,Points,3,1);
         wav_final=[wav_final,pred_value];
         Prediction2=[Prediction2(2:end),pred_value];
      end
      wav_row=[wav_row,wav_final];




      [wav_final,CXD,LXD] = wden(wav_final,'sqtwolog','s','one',5,'db4');
      wav_row99=[wav_row99,wav_final];
   end
   photo99(count_x,1:y_zero)=wav_row99;
   photo5(count_x,1:y_zero)=wav_row;
   count_x=count_x+1;
end
Mean_of_Block2=(mean(photo5'))'; % For each row
Total_mean2=mean(Mean_of_Block2);
if Difference<10
   scale=Total_mean/Total_mean2;
else
   scale=1;
end
photo5=scale*double(photo5);
photo99=scale*double(photo99);
```

```
photo0=double(photo);
photo55=double(photo5);
photo999=double(photo99);
Error2=[(photo0(x_start:x_end,y_start:y_end)-photo55).^2];
Error2=(sum(Error2'))';
[Min_Error2,Min_Row_Error2]=min(Error2);
Row_Error_Wavelet=Error2;

Error29=[(photo0(x_start:x_end,y_start:y_end)-photo999).^2];
Error29=(sum(Error29'))';
[Min_Error29,Min_Row_Error29]=min(Error29);
Denoising_Error=Error29;
LPC_Error__Wavelet_Error=[Row_Error_LPC,Row_Error_Wavelet,Denoising_Error]

% ********************************************************************
function [pred_value]=wav_pred3(signal,k,m,n)

% Inputs
% signal= signal that will be wavelet decomposed
%   k   = the number of signal points counting from the end
%         that will be used
%   m   = level of decomposition
%   n   = number of the consequtive lost packet from the 'Errors.m'
% Output
% pred_value= the next 4 predicted values of the signal


help=1;%rem(n,2);
pred_value=[];
mm=1:m; % used in the detcoef
```

```
s1=signal(end-k+1:end);
[c,l] = wavedec(s1,m,'db1');           % wavelet decomposition
ca3 = appcoef(c,l,'db1',m);            % approximation coefficients at level 2
[cd1,cd2,cd3] = detcoef(c,l,mm);   % Extract detail coefficients at levels 1, 2

if size(ca3)>20
   order1=18;
else
   order1=5;
end

if size(cd2)>20
   order2=18;
else
   order2=5;
end

if size(cd1)>20
   order3=18;
else
   order3=5;
end

% PREDICTION

% APPROXIMATION 3 ******************************************
for j=ceil((k+1)/8)     % these are the coefficients I want to estimate (of the lost frame)
  [signal_estim1_a3]=Predict2(ca3,j,order1);
end  % end j=25:28
ca3=[ca3,signal_estim1_a3];
```

```
% ************************************************************
% DETAIL 3 ***************************************************
if help==1
   cd3=[cd3,0];
else
   for j=ceil((k+1)/8)
      [signal_estim1_d3]=Predict2(cd3,j,order1);
   end
   cd3=[cd3,signal_estim1_d3];
end
% ************************************************************
% DETAIL 2 ***************************************************
   for j=ceil((k+1)/4)
   [signal_estim1_d2]=Predict2(cd2,j,order1);
end
D2=[cd2,0];                % corresponds to the 49th & 50th point
cd2=[cd2,signal_estim1_d2];   % corresponds to the 51st & 52nd point
% ************************************************
% DETAIL 1 *************************************
for j=ceil((k+1)/2)
   [signal_estim1_d1]=Predict2(cd1,j,order2);
end
D1=[cd1,0];
cd1=[cd1,signal_estim1_d1];
for j=ceil((k+2)/2)
   [signal_estim1_d1]=Predict2(cd1,j,order2);
end
D11=[cd1,0];
cd1=[cd1,signal_estim1_d1];
% ************************************************
```

```
% RECONSTRUCTION**********************************
Rec1=[ca3,cd3,D2,D1];L1=[ceil((k+1)/8),ceil((k+1)/8),ceil((k+1)/4),ceil((k+1)/2),k+1];
Rec2=[ca3,cd3,D2,cd1];L2=[ceil((k+1)/8),ceil((k+1)/8),ceil((k+1)/4),ceil((k+1)/2),k+2];
Rec3=[ca3,cd3,cd2,D11];L3=[ceil((k+1)/8),ceil((k+1)/8),ceil((k+1)/4),ceil((k+2)/2),k+3];
Rec4=[ca3,cd3,cd2,cd1];L4=[ceil((k+1)/8),ceil((k+1)/8),ceil((k+1)/4),ceil((k+2)/2),k+4];
X1= waverec(Rec1,L1,'db1');
X2= waverec(Rec2,L2,'db1');
X3= waverec(Rec3,L3,'db1');
X4= waverec(Rec4,L4,'db1');
pred_value=[pred_value,X1(end),X2(end),X3(end),X4(end)];
```

2. Wavelet-Based Prediction with Pre-Denoising

```
%function
[TOTAL_lost_packet_rate,PSNR_BLOCK,PSNR_LPC,PSNR_WAVELETS,PSNR_DE
NOISING]=lena_transmit_noise(p22,p11,point)

% Input
% p22=probability to stay in the good channel
% p11=probability to stay in the bad channel
% point=just a number in order to get the figures numbered
% Output
% pretty obvious

%     PREDENOISING
%     FIRST DENOISING THEN PREDICTION

clc
clear
repetition=5;point=240;
load('photo','photo')
load('photo3','photo3')
```

```
load('x_start','x_start')
load('x_end','x_end')
load('y_start','y_start')
load('y_end','y_end')
load('how_many','how_many')
figure(1)
imshow(photo)

% ************************************
%      DENOISING THE IMAGE                    *
% ************************************
% First I denoise the image that I inserted noise in and then I'll
% use the output XC in order to make the predictions

xxx=double(photo3);
[THR,SORH,KEEPAPP] = ddencmp('den','wp',xxx)
[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('gbl',xxx,'db3',10,33,SORH,KEEPAPP);
X=uint8(XC);
LPC_photo=X;
Wavelet_photo=X;
figure(repetition+14)
imshow(X),title('Denoised Picture')

% ************************************
%          PREDICTED BLOCKS                    *
% ************************************
% First Predicted Block
for i=1:how_many
   photo_1=[];photo_11=[];
 [photo_1,photo_11,photo_99]=photo_pred(X,x_start(i),x_end(i),y_start(i),y_end(i),96);
```

```
   LPC_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_1;
   Wavelet_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_11;
   Denoising_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_99;
end
% ************************************************************
figure(repetition+7+point)
imshow(LPC_photo)%,title('Predicted Image using Linear Prediction')
figure(repetition+13+point)
imshow(Wavelet_photo)%,title('Predicted Image using Wavelets')

% PSNR
psnr_miss_block=psnr(photo,photo3)
psnr_miss_block_denoised=psnr(photo,X)
psnr_LPC=psnr(photo,LPC_photo)
psnr_Wavelet=psnr(photo,Wavelet_photo)
```

3. Wavelet-Based Prediction With Post-Denoising

```
%function
[TOTAL_lost_packet_rate,PSNR_BLOCK,PSNR_LPC,PSNR_WAVELETS,PSNR_DE
NOISING]=lena_transmit_noise(p22,p11,point)
%     POSTDENOISING
%     FIRST PREDICTION THEN DENOISING
% Input
% p22=probability to stay in the good channel
% p11=probability to stay in the bad channel
% point=just a number in order to get the figures numbered
% Output
% pretty obvious

clc
```

```
clear
repetition=5;point=240;
load('photo','photo')
load('photo3','photo3')
load('x_start','x_start')
load('x_end','x_end')
load('y_start','y_start')
load('y_end','y_end')
load('how_many','how_many')

figure(1)
imshow(photo)
LPC_photo=photo3;
Wavelet_photo=photo3;
figure(repetition+14)
imshow(photo3
% ************************************
%          PREDICTED BLOCKS                    *
% ************************************
% First Predicted Block
for i=1:how_many
   photo_1=[];photo_11=[];
[photo_1,photo_11,photo_99]=photo_pred(photo3,x_start(i),x_end(i),y_start(i),y_end(i),
96);
   LPC_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_1;
   Wavelet_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_11;
   Denoising_photo(x_start(i):x_end(i),y_start(i):y_end(i))=photo_99;
end
% ************************************************************
figure(repetition+7+point)
```

```
imshow(LPC_photo),title('Predicted Image using Linear Prediction')
[psnr_noise_LPC,psnr_Denoising_LPC]=denoise(photo,LPC_photo,repetition)
figure(repetition+13+point)
imshow(Wavelet_photo),title('Predicted Image using Wavelets')
[psnr_noise_Wavelet,psnr_Denoising_Wavelet]=denoise(photo,Wavelet_photo,repetition
+4)

% PSNR
psnr_miss_block=psnr(photo,photo3)
%psnr_LPC=psnr(photo,LPC_photo)
%psnr_Wavelet=psnr(photo,Wavelet_photo)
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. SIMULATION RESULTS FOR VARIOUS IMAGES

A. WAVELET-BASED PREDICTION USING HAAR WAVELET

In order to show that the wavelet-based prediction performs better than direct linear prediction, independent of the type of image used, the simulations described in Chapter V are carried out on a variety of images. The results of these simulations indicate that the wavelet-based prediction performs better than the direct prediction in all cases.

In the following figures, the original image, the received image without error recovery, the recovered image using direct linear prediction and the recovered image using the wavelet-based prediction are shown. The three curves shown in the plots indicate peak signal-to-noise ratio of the received image (dotted line), the image with linear prediction based error recovery (dashed line) and the wavelet prediction (solid line). All results are based on averaging six simulation runs.

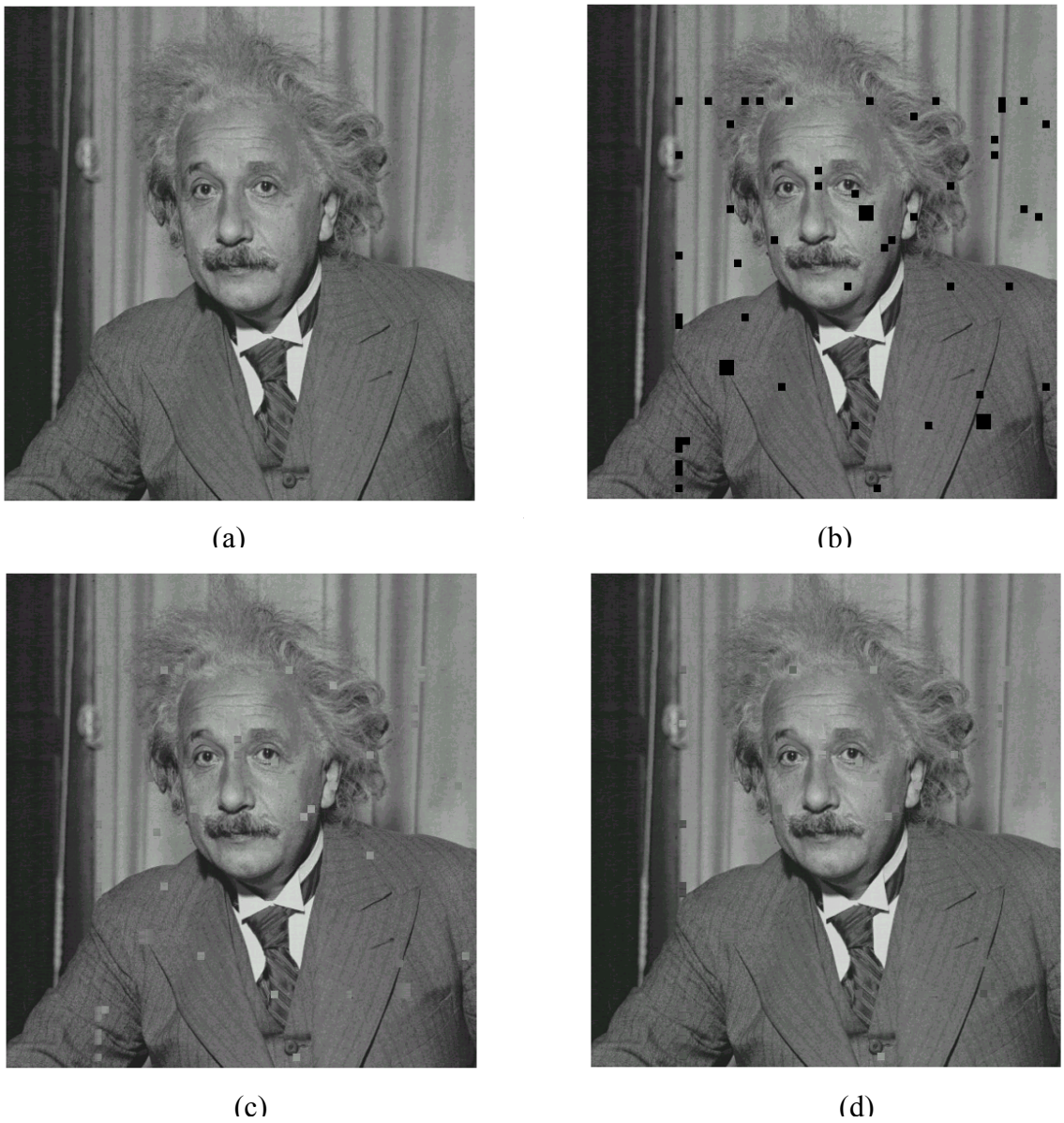


Figure B.1. (a) Original Image, (b) Image without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet Based Prediction.

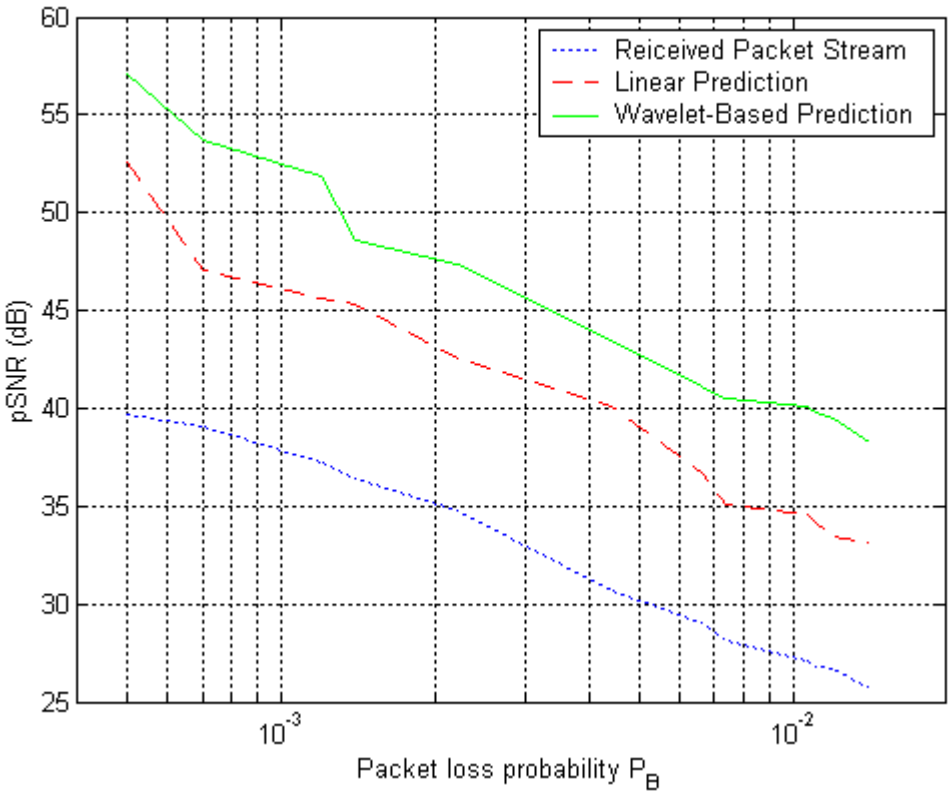


Figure B.2. Peak Signal-to-Noise Ratio versus Packet Loss Probability P_B for Image Packet Stream.

(a)

(b)

(c)

(d)

Figure B.3. (a) Original Image, (b) Image without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet Based Prediction.

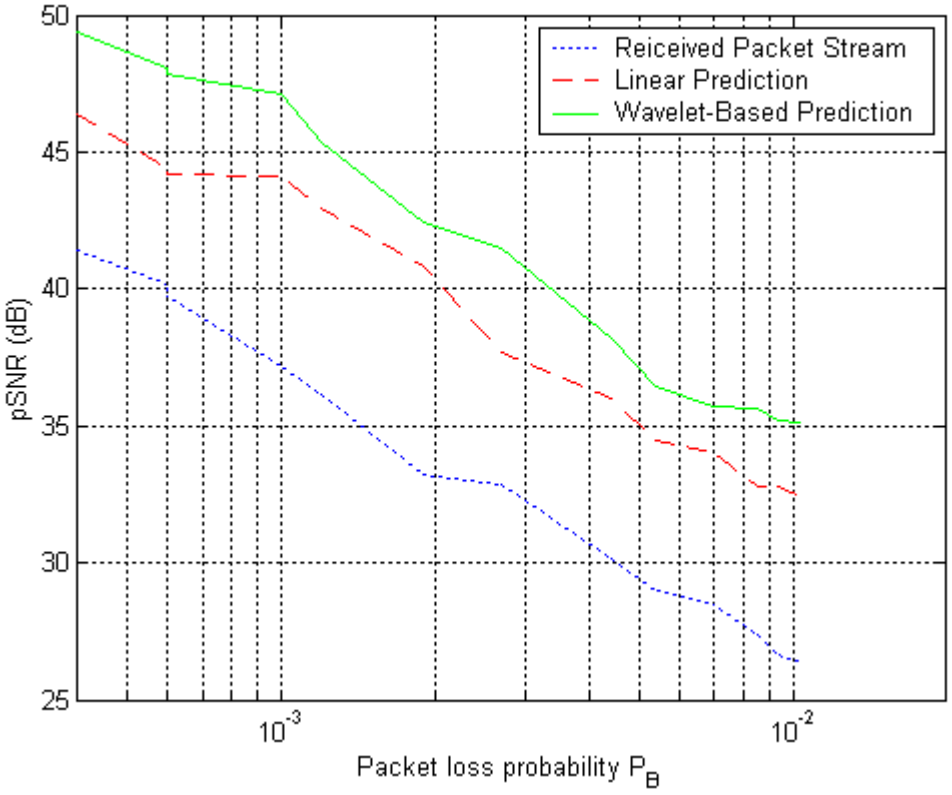


Figure B.4. Peak Signal-to-Noise Ratio versus Packet Loss Probability P_B for Image Packet Stream.

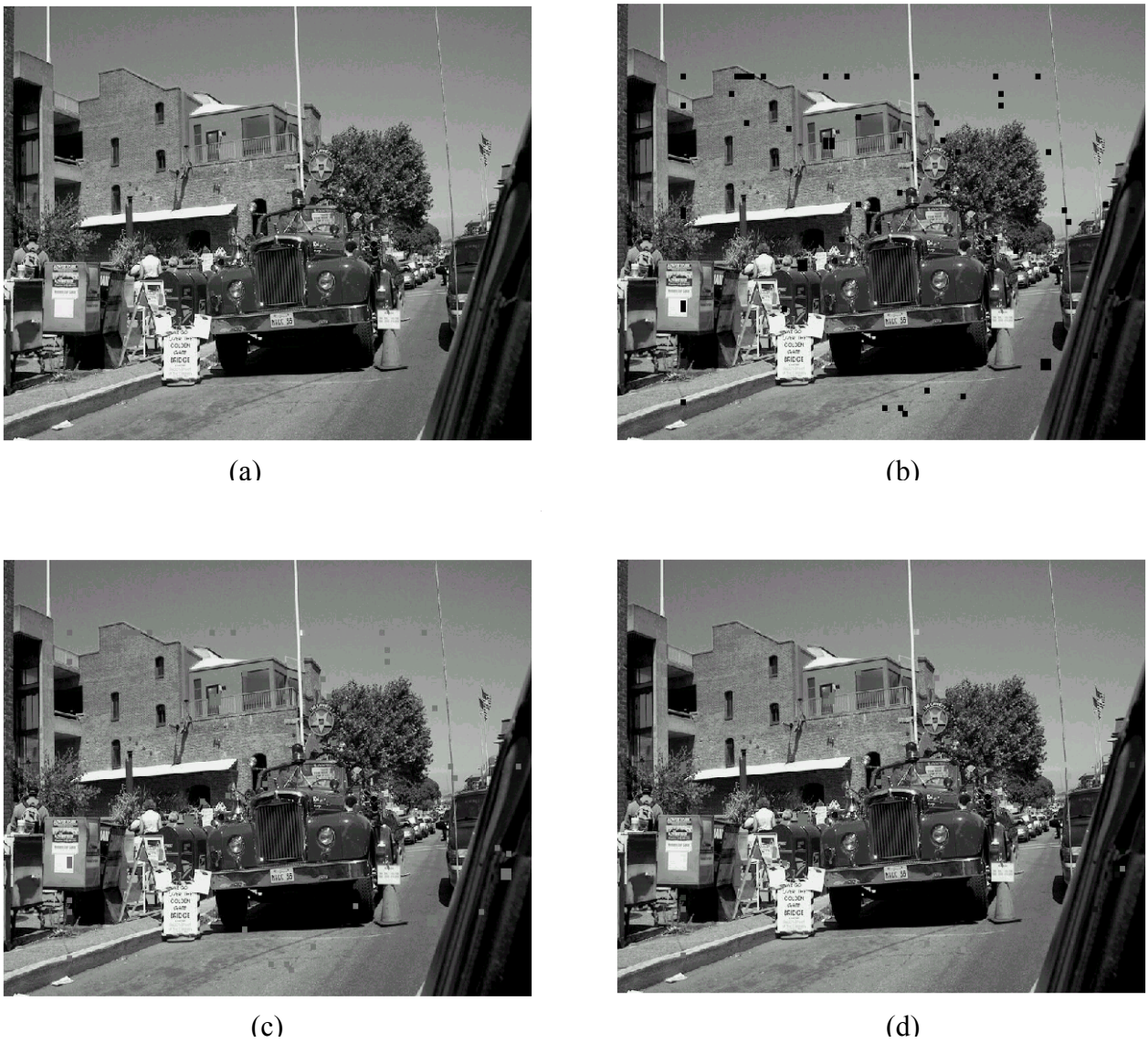


Figure B.5. (a) Original Image, (b) Image without Error Recovery, (c) Recovered Image Using Linear Prediction, (d) Recovered Image Using Wavelet Based Prediction.

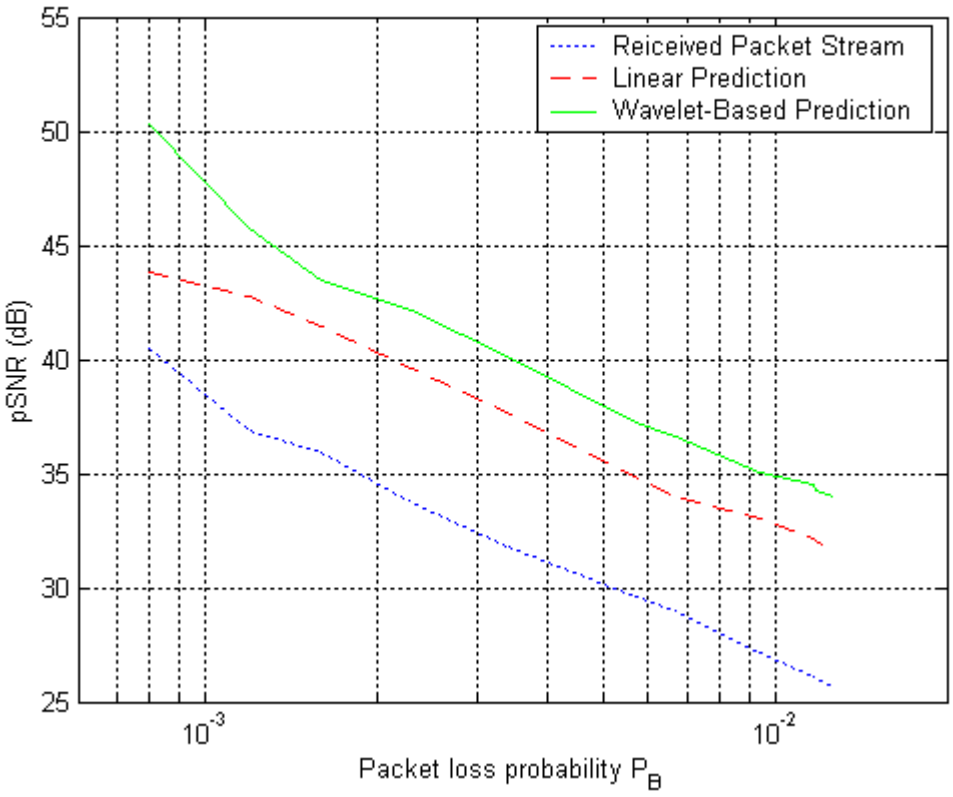


Figure B.6. Signal-to-Noise Ratio versus Packet Loss Probability P_B for Image Packet Stream.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Robi Polikar, *The Wavelet Tutorial*," Rowan University, 1999. [http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html]

2. Guy P. Nason and B.W. Silverman, "*The discrete wavelet transform*," S. J. Comp. Graph. Statist., 3, 163-191, 1994.

3. Khalid Sayood, "*Introduction to Data Compression*," Morgan Kaufmann, 2000.

4. Matthew S. Crouse, Robert D. Nowak and Richard G. Baraniuk, "*Wavelet-Based Statistical Signal Processing Using Hidden Markov Models*," IEEE Transaction on Signal Processing, Vol. 46, No. 4, April 1998.

5. V. Lakshmanan, "*A Short Write-up On Wavelets*," January 2000. [http://www.nssl.noaa.gov/~lakshman/Papers/wavelet.pdf]

6. Charles W. Therrien, "*Discrete Random Signals And Statistical Signal Processing*," Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1992.

7. John Proakis, Dimitris Manolakis, "*Digital Signal Processing - Principles Algorithms and Applications*," Third Edition Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1995.

8. C. Sidney Burrus, Ramesh A. Gopinath, Haitao Guo, "*Introduction to Wavelets and Wavelet Transforms*," Edition Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1997.

9. Stéphane Mallat , "*A Wavelet Tour of Signal Processing*," Second Edition, Academic Press, 1999.

10. F. Babich and G. Lombard, "*A Markov Model for Mobile Propagation Channel*," IEEE trans. of Vehicular Technology, Vol. 49, No. 1, January 2000.

11. E. N. Gilbert, "*Capacity of a Burst-Noise Channel*," Bell System Tech. Journal, Vol. 39, pp. 1253-1266, Sept. 1960.

12. E. O. Elliot, "*Estimates of Error Rates for Codes on Burst-Error Channels*," Bell System Tech. Journal, Vol. 42, p. 1977, Sep. 1963.

13. A. Leon-Garcia, "*Probabilities and Random Process for Electrical Engineering*," Addison-Wesley Publishing Co., Massachusetts, 1994.

14. A. Spanias and T. Painter, "*Matlab Simulation of NSA FS-1016 CELP v3.2*," Arizona State University, 1995-1999. [www.eas.asu.edu/~spanias]

15. E. Aboufadel and Steven Schlicker "*Discovering Wavelets*," Wiley-Interscience, 1999.

16. A. V. Oppenheim, A. S. Willskyand, S. H. Nawab, "*Signals & Systems*," Second Edition, Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1997.

17. Benjamin W. Wah, Xiao Su and Dong Lin, "*A Survey of Error-Concealment Schemes for Real-Time Audio and Vide0 Transmissions over the Internet*," IEEE International Symposium on Multimedia Software Engineering, December 2000.

18. Ankit Patel, Mark Tonkelowitz,Xiao and Mike Vernal, "*Lossless Sound Compression using the Discrete Wavelet Transform*," Harvard University, January 14, 2002. [http://www.people.fas.harvard.edu/~mtonkel/sndcomp.pdf]

19. W. Leung and F. Chang, "*Transient Analysis via Fast Wavelet-Based Convolution*," 1995 ISCAS Symp. Digest, Vol. 3, 1995, pp. 1884-1887.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman
   Department of Electrical and Computer Engineering
   Monterey, California

4. Professor Murali Tummala, Code EC/Tu
   Department of Electrical and Computer Engineering
   Monterey, California

5. Professor Robert Ives, Code EC/Ir
   Electrical Engineering Department, USNA
   Annapolis, Maryland

6. Dr. Rich North
   SPAWAR Systems Center
   San Diego, California

7. Hellenic Navy General Staff
   Department B/2, Stratopedo Papagou, Mesogeion 151
   Holargos, 155-00, Athens
   Greece

8. Lt JG Anastasios Garantziotis H.N
   21 Agisilaou,
   Tzitzifies, Athens, 176 74
   Greece