Smoothing Techniques for the Reconstruction of Missing Samples

Jennifer A. Fulton

BE (Hons)

January 1997

A thesis submitted for the degree of Master of Engineering
of The Australian National University

Department of Engineering
Faculty of Engineering and Information Technology
The Australian National University
Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any University or other institution of tertiary education. Information derived from published or unpublished work of others has been acknowledged and a list of references is given.

Signed ..............................................

Date ..............................................
Abstract

The central theme of this thesis is the application of Kalman filtering techniques to reconstruct sampled measurements. In particular the ability of the Kalman filter to produce smoothed estimates of these measurements is investigated. The two applications considered are:

- a speech coding system which down-samples the transmitted data
- an algorithm which reconstructs the measurements at a failed sensor in an array

An algorithm for reducing the bit-rate while maintaining the quality of the output speech for a version of the Adaptive Differential Pulse Code Modulation coder is presented. The algorithm down-samples the transmitted error signal which allows less frequent transmission of data but at a higher resolution as opposed to transmitting every sample at low resolution. A Kalman filter is applied to reconstruct the down-sampled speech signal at the decoder. A comparison of the down-sampled system transmitting every second measurement with a full-sampled system operating at the same bit-rate is presented. A theoretical analysis and results from tests using real speech data are used to compare the performance of the two systems.

A method for reconstructing the measurements at a failed sensor in a linear equally spaced array is presented. The algorithm applies a Kalman filter to the data along the array and provides smoothed estimates for the measurements at the failed sensors. A comparison between this algorithm and an existing algorithm which uses linear predictive coding to produce estimates of the measurements is presented.
Acknowledgements

I would like to thank my supervisors, Prof. Bob Bitmead and Dr. Bob Williamson for their support, guidance and encouragement during the course of this thesis. I would also like to thank Dr. Jane Perkins for all her help and ideas on the array processing work.

I acknowledge the Australian Government for their funding of the Co-operative Research Centre program and the provision of Australian postgraduate research awards.

I would also like to acknowledge the support the Co-operative Research Centre for Robust and Adaptive Centre has given me. The centre has provided me with a top-up scholarship and the opportunity to spend two weeks visiting the Maritime Operations Division, DSTO in Salisbury.
Contents

1 Introduction

1.1 Thesis Motivation ........................................ 1

1.2 Thesis Outline ........................................... 3
  1.2.1 Overview ........................................... 3
  1.2.2 Summary of Original Contributions .................. 5

2 Down-Sampled ADPCM ....................................... 7

2.1 Introduction ............................................. 7

2.2 Speech Coding Systems and Speech Signals ............... 8
  2.2.1 Speech coding ....................................... 8
  2.2.2 Speech Signal ....................................... 10
  2.2.3 Speech Quality Measurement ......................... 12
  2.2.4 Adaptive Differential Pulse Code Modulation ........ 14

2.3 Speech Signal Models ................................... 15

2.4 Linear Predictive Coders and State Estimation ........... 16
List of Figures

2-1 Time and frequency domain plots for samples of voiced and unvoiced speech ........................................ 11

2-2 Simplified block diagram of ADPCM .................................................. 14

3-1 Block diagram of the Down-Sampled KF-ADPCM system .......................... 30

3-2 Comparison of the full-sampled and down-sampled KF-ADPCM systems ........ 32

3-3 Speech signal from the male speaker, its spectrogram and the resulting segmental SNR for the full and down-sampled systems ........................................... 33

3-4 Informal MOS Results for the Segments of Speech from the Male and Female Speaker .................................................. 35

4-1 Projection of a plane wave on a linear array ....................................... 43

4-2 Beam Patterns for a SNR of 20dB using a) Hamming window and b) rectangular window ........................................... 50

4-3 Beam Patterns for a SNR of 10dB using a) Hamming window and b) rectangular window ........................................... 50
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-4</td>
<td>Beam Patterns for a SNR of 3 dB using a) Hamming window and b) rectangular window</td>
</tr>
<tr>
<td>4-5</td>
<td>Comparison of the linear predictive and Kalman filtering algorithm using an estimated model for various SNRs for a) Hamming window and b) Rectangular window.</td>
</tr>
<tr>
<td>4-6</td>
<td>Comparison of the linear predictive and Kalman filtering algorithm using the actual model for various SNRs for a) Hamming window and b) Rectangular window.</td>
</tr>
<tr>
<td>4-7</td>
<td>Beam Patterns for a SNR of 20dB using a) Hamming window and b) Rectangular window</td>
</tr>
<tr>
<td>4-8</td>
<td>Beam Patterns for a SNR of 10dB using a) Hamming window and b) Rectangular window</td>
</tr>
<tr>
<td>4-9</td>
<td>Beam Patterns for a SNR of 3 dB using a) Hamming window and b) Rectangular window</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Mean Opinion Score Five Point Quality Scale .......................... 13

3.1 SNR Measures for a Segment of Speech from a Male and a Female Speaker ................................................................. 34

3.2 SNR Measures for a Segment of Speech from a Male Speaker ..... 36

3.3 SNR Measures for a Segment of Speech from a Male Speaker ..... 37
Chapter 1

Introduction

1.1 Thesis Motivation

Signals can be either classified as a continuous-time or a discrete-time signal depending on whether the independent variable (time) is continuous or discrete. It is often necessary to sample the continuous-time signals to generate a discrete-time signal for subsequent use in digital systems. If certain conditions are obeyed during the sampling process, the continuous-time and discrete-time signals are entirely equivalent in that either can be reconstructed from the other. These conditions are given by Nyquist’s (or Shannon’s) sampling theorem [20]. The theorem states that if the continuous signal is bandlimited to $\Omega_n$, the sampling frequency used must be greater than or equal to $2\Omega_n$. The resulting discrete-time signal consists of equispaced samples that lie in a continuous range and can take on any one of the infinite values in the range. If the conditions are obeyed, there exists techniques to interpolate or reconstruct the signal perfectly at other time instances. [6, 7, 23, 28, 33].

However, the discrete-time signal is still not suitable for use in a digital system. The discrete-time signal must first be quantised. The amplitude of each sample is
approximated to the nearest quantised level. The accuracy of the quantised signal can be improved to any desired degree by increasing the number of levels in the quantiser and hence the number of bits in its binary representation. It is still possible to reconstruct the signal at other time instances but the accuracy of the reconstructed samples will depend both on the accuracy of the samples and the sample rate used.

In this thesis we investigate the performance of the Kalman filter for the purpose of reconstructing measurements in a digital system for two separate applications. The Kalman filter is a statistically optimal, model-based state estimator that has the ability to produce smoothed estimates. Smoothed estimates are produced by considering measurements up to and including time $t$ and measurements past time $t$, whereas predicted estimates are produced by considering measurements only up to and including time $t - 1$.

The first part of the thesis proposes and analyses a novel algorithm for reducing the bit-rate in a speech coding system while maintaining the quality of the output speech. The ITU G.721 telephony standard based on Adaptive Differential Pulse Code Modulation (ADPCM) transmits at 32Kbits/sec ($8KHz \times 4bit$). Speech typically has a bandwidth of 3.7KHz. Therefore the sampling rate used in ADPCM satisfies Nyquist’s sampling theorem. The algorithm we propose uses a version of the ADPCM coder but only transmits some of the measurements across the channel (i.e. we down-sample the transmitted data). We specifically consider transmitting only every second sample. By down-sampling we can use less frequent transmission of measurements but at a higher resolution (eg. $4KHz \times 4bits$) as opposed to sending every measurement at a lower resolution (eg. $8KHz \times 2bits$). The algorithm uses a Kalman filter to reconstruct the down-sampled speech signal at the decoder. The Kalman filter uses a model for the speech signal to produce smoothed estimates of the speech signal.
There appears to be a contradiction of the Nyquist sampling theorem in the use of data sampled at less than twice the bandwidth of speech. However both our goals and assumptions are different to those pertaining to Nyquist’s sampling theorem. The samples we have available for reconstruction are imperfect and we only require adequate (approximate) reconstruction of the signal. Furthermore, we have a signal model which provides more information about the signal of interest than just an upper bound on their bandwidth.

The second part of the thesis proposes an algorithm for reconstructing the measurements at a failed sensor in a linear equally spaced array. In practice sensors in an array malfunction due to various factors and can dramatically affect the performance of a linear array. They can cause disruption of the beam-pattern, by adding spurious sidelobes and increasing the overall sidelobe level. The algorithm implements a Kalman filter along the array data to produce smoothed estimates of the measurements at the failed sensors.

1.2 Thesis Outline

1.2.1 Overview

Chapter 2

This chapter provides a brief introduction into the applications of speech coding systems, temporal and spectral properties of speech signals and methods used for measuring speech quality. It also provides a brief explanation of the Adaptive Differential Pulse Coded Modulation (ADPCM) algorithm for speech coding.

We then present a down-sampled version of an ADPCM coder which uses Kalman filtering techniques to reconstruct the speech signal (referred to as KF-ADPCM). We first introduce the all-pole speech model, linear prediction of the speech signal
and state estimation of the speech signal. We then discuss how the Kalman filter can account for missing measurements due to down-sampling and how it can be used to reconstruct the signal using ideas from smoothing. Finally a theoretical analysis comparing a down-sampled and full-sampled systems is presented using their respective error covariance matrices as performance means.

Chapter 3

This chapter compares the performance of the down-sampled KF-ADPCM system with the performance of a full-sampled KF-ADPCM system operating at the same bit-rate using real speech data. The relative performance of the two systems is analysed with the system first transmitting the model parameters to the decoder as side information (known as a forwards adaptive system) and then this information being estimated at both the encoder and the decoder from the transmitted data (referred to as a backwards adaptive system).

Chapter 4

In this chapter a new algorithm for reconstructing the measurements at a failed sensor for a linear equally spaced array is presented. The method proposed is similar to the algorithm used to reconstruct the down-sampled speech signal discussed in the previous chapters. The algorithm uses a Kalman filter to provide smoothed estimates of the measurements at the failed sensors. A comparison of this method to an existing method which uses a combination of linear predictors running forwards and backwards along the array is performed.

Chapter 5

The last chapter summarises the major results and suggests possible areas for future research.
1.2.2 Summary of Original Contributions

During the course of research for this Masters thesis, a number of original contributions have been made. A brief description of these contributions is listed below:

- We have proposed a novel algorithm for reducing the bit-rate while maintaining the quality of the output speech for a version of the Adaptive Differential Pulse Code Modulation coder. We down-sample the transmitted error data. This allows us to use less frequent transmission of data but at a higher resolution as opposed to transmitting every sample at low resolution. A Kalman filter which uses smoothing is applied to reconstruct the downsampled speech signal at the decoder.

- A theoretical performance analysis is presented which compares the new down-sampled system to a full-sampled system operating at the same bit-rate via their respective error covariance matrices. A blocked model for the down-sampled system was introduced to enable the comparison of the two systems to be performed. The results indicate that the down-sampled system can outperform the full-sampled systems on signals which have similar properties to voiced speech.

- The down-sampled algorithm is also tested using real speech data. The results indicate that substantial improvements in subjective quality can be obtained in using this new algorithm over the full-sampled system for a forwards adaptive system.

- A new data-dependent method for reconstructing the measurements at a failed sensor in a linear array is introduced. The method applies a Kalman filter to the data along the array and provides smoothed estimates for the measurements at the failed sensors. A comparison between this algorithm and an existing algorithm which uses a combination of linear predictors that operate
both forwards and backwards along the array is performed. The results indicate that both the linear prediction and Kalman filter algorithms recover most of the loss in performance due to the failed sensors at signal-to-noise ratios (SNR) greater than 10dB and at SNRs around 0dB approximately 50% of the loss in performance due to the failed sensors is recovered.
Chapter 2

Down-Sampled ADPCM

2.1 Introduction

Bit-rate reduction in Adaptive Differential Pulse Code Modulation (ADPCM) has traditionally been achieved by reducing the number of bits per sample. Below three bits per sample the performance of traditional ADPCM systems degrades quickly. Another possible mechanism for achieving bit-rate reduction is by down-sampling the transmitted data. By down-sampling, we can use less frequent transmission of data but at a high resolution as opposed to transmitting every sample at low resolution. For example, a down-sampled system can operate at 4KHz×4bit whereas a full-rate system with the same bit-rate would have to operate at 8KHz×2bit.

In order to use down-sampling in speech coding systems we need to use reconstruction techniques at the decoder. We investigate the use of Kalman filtering techniques to reconstruct the down-sampled speech signals. This technique was first investigated by Ramabadran and Sinha [34]. In their work, the Kalman filter was used in a system with all pole predictors for both long delay (pitch) and short delay (formant) structure and produced some encouraging results. We extend this by using a
Kalman smoother, thereby altering the reconstruction problem from extrapolation (prediction) to interpolation (smoothing). The Kalman filter of an all-pole model can provide smoothed estimates up to lag $N - 1$ at no additional computational complexity. The improvement in performance by smoothing rather than filtering is well known [1].

This chapter begins with a brief introduction into the applications of speech coding systems, temporal and spectral properties of speech signals and methods used for measuring speech quality. It also provides a brief explanation of Adaptive Differential Pulse Coded Modulation. We then present a down-sampled version of an ADPCM codec which uses Kalman filtering techniques to reconstruct the speech signal (known as KF-ADPCM). For certain classes of signals, it is capable of providing better performance than standard ADPCM and full-sampled KF-ADPCM at bit-rates of 12-16Kbits/sec. We first introduce the all-pole speech signal model, linear prediction of the speech signal and state estimation of the speech signal. We then discuss how the Kalman filter can account for missing measurements due to down-sampling and how it can be used to reconstruct the signal. A performance analysis comparing the down-sampled and the full-sampled systems using their respective error covariance matrices is then presented.

2.2 Speech Coding Systems and Speech Signals

2.2.1 Speech coding

Speech coding or speech compression is the field concerned with obtaining compact digital representation of voice signals for the purpose of efficient transmission or storage. A speech coding algorithm is evaluated based on the bit-rate, the quality of the reconstructed speech, the complexity of the algorithm and the delay introduced. The importance placed on each of these requirements varies with the intended ap-
plication of the algorithm.

Speech coding plays an important role in three broad areas, the wired telephone network, the wireless network and voice security for both privacy and encryption. In the wired network applications strong emphasis is placed on the quality, delay and complexity requirements of the algorithm. Whereas within the wireless networks the requirements on delay and quality are often relaxed but more emphasis is placed on a lower bit-rate because of the intended channel capacity. Generally with security applications lower quality, lower bit-rates and long delay algorithms are used. Lower bit-rates are used because they reduce the likelihood of jamming or interception.

Besides transmission applications there are also several other areas for which speech coding can be used, including storage and teleconferencing. One of the largest application areas for speech coding is in voice messaging and voice mail, whereby a voice message is sent to a mailbox, stored in coded form and delivered to the intended recipient when he or she is ready to receive it. Another growing application for speech coding is voice response systems. Voice response systems are being used as the front end processors for telephone enquiries to most major businesses. The voice response system speaks out a message in response to a touch tone input from a telephone system or ultimately in response to voice input commands and thereby maintains a dialogue with the user.

The field of digital coding of wide-band speech signals is an emerging area of coding especially for applications like digital audio broadcasting of compact disk audio over frequency modulation channels and for surround sound for high definition television. There is also the area of teleconferencing of coded wide-band speech which is rapidly being used within businesses and should appear in the home and in wireless systems within the next few years. Another interesting application is digital telephone answering machines where the usual tape recorder for storage of messaging has been replaced by solid-state memory. Signal storage is also used in multimedia games,
2.2.2 Speech Signal

Speech is primarily produced by the larynx, which contains a pair of muscular folds called the vocal chords and the vocal tract which is a tube leading from the larynx along the pharynx and then branching into the oral cavity leading to the lips and through the nasal cavity to the nostrils. Acoustic energy can be generated in two different ways.

The primary mechanism occurs in the larynx and is known as voiced excitation. The muscles in the larynx place the vocal chords close together and make them loose enough that when air from the lungs is driven through them they open and close periodically at an average of 110 times a second for a man and about twice that for a woman. The main instant of voiced excitation occurs when the air-flow from the lungs is suddenly stopped as the cords are pulled together by Bernoulli forces. The resulting voiced speech sounds includes all vowels and many consonant sounds. The words “roman”, “yellow” and “wiring” are composed entirely of voiced sounds.

The second mechanism for generating acoustic energy in speech is when air passes from the lungs through the larynx with the vocal chords held apart and is forced through a constriction formed by the tongue or lips causing turbulence and resulting in an aperiodic, noise-like excitation. Sounds like the “s” and “ft” in soft are generated this way and are referred to as unvoiced speech. Further information about speech production can be found in [10] and [17].

As a consequence of being produced by different mechanisms, voiced and unvoiced speech have different spectral and temporal properties. Speech signals are highly non-stationary but are generally assumed to be piecewise stationary for segments up to 20ms in length. Thus the statistical and spectral properties are defined over
short segments. An example of time and frequency domain plots for voiced and unvoiced speech is shown in Figure 2-1.

![Voice and Unvoice Speech Plots](image)

Figure 2-1: Time and frequency domain plots for samples of voiced and unvoiced speech.

Voiced speech is quasi-periodic in the time domain and this corresponds to the pitch period. The pitch period is related to the vocal cord vibration period and ranges from 30 to 120 samples (sampled at 8KHz). The energy in voiced speech is concentrated in the low frequency region with considerable redundancy both long and short term. Unvoiced speech has a noise-like character with no periodicity (the vocal chords are not vibrating) and no slowly changing temporal characteristics. The energy in unvoiced speech is spread over the whole bandwidth and its energy level is much smaller than that of voiced speech. Further information about speech signals can be found in [12] and [17].
2.2.3 Speech Quality Measurement

Speech quality is classified into four general categories, namely: broadcast, network or toll, communications and synthetic. Broadcast speech refers to very high quality speech such as found in FM radio. Toll quality refers to quality comparable to standard telephone quality speech at 3.7KHz bandwidth. Communications quality implies somewhat degraded speech quality which is nevertheless natural, highly intelligible and adequate for telecommunications. Synthetic speech is usually intelligible but can be unnatural and associated with loss of speaker recognisability.

Measuring the speech quality of a speech codec is an important but generally very difficult task. The signal-to-noise ratio (SNR) is one of the most common objective measures used for evaluating the performance of a speech codec. This is given by

$$SNR = 10 \log_{10} \frac{\sum_{n=0}^{M} s^2(n)}{\sum_{n=0}^{M} (s(n) - \hat{s}(n))^2}$$

(2.1)

where $s(n)$ is the original speech data and $\hat{s}(n)$ is the output of the speech codec. The SNR is a long term measure for the accuracy of speech reconstruction and tends to hide temporal reconstruction noise particularly for low level signals. Temporal variations of the performance can be better detected and evaluated using a short term signal-to-noise ratio (i.e. by computing the SNR for each N-point segment of speech). A performance measure that exposes weak signal performance is the segmental SNR (SEGSNR) which is given by

$$SEGSNR = \frac{10}{L} \sum_{i=0}^{L-1} \log_{10} \frac{\sum_{n=0}^{N-1} s^2(iN + n)}{\sum_{n=0}^{N-1} (s(iN + n) - \hat{s}(iN + n))^2}$$

(2.2)

Objective measures do not typically account for perceptual properties of the ear. Thus subjective tests for determining quality and intelligibility are needed to supplement or validate objective test results. Common subjective tests used include the Diagnostic Rhyme Test (DRT) [43], the Diagnostic Acceptability Measure (DAM) [42] and the Mean Opinion Score (MOS).
In the Diagnostic Rhyme Test the subject’s basic task is to recognise one of two possible words in a rhyming pair (e.g. moot and hoot). This test is used mainly for low-rate speech coding systems where intelligibility is the primary concern. The Diagnostic Acceptability Measure scores are based on results of test methods evaluating the quality of a speech coding system based on the acceptability of speech as perceived by a trained listener. The Mean Opinion Score (MOS) test involves a group of subjects rating samples from speech coding systems on a discrete five point scale and is widely used to quantify coded speech. MOS scores are obtained by averaging the responses from the group of listeners. The quality and impairment levels of the five scales are shown in Table 2.1.

<table>
<thead>
<tr>
<th>MOS Score</th>
<th>Quality Scale</th>
<th>Impairment Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Slightly Annoying</td>
</tr>
<tr>
<td>2</td>
<td>Poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Unsatisfactory</td>
<td>Very Annoying</td>
</tr>
</tbody>
</table>

Table 2.1: Mean Opinion Score Five Point Quality Scale

The MOS range relates to speech quality as follows: a MOS of 4-4.5 implies toll quality, scores between 3.5 and 4 imply communications quality and a MOS between 2.5 and 3.5 implies synthetic quality. MOS ratings vary from test to test and are not absolute measures for the comparison of speech coders.

Subjective evaluations such as the ones mentioned above, can be very time-consuming and costly. Objective methods are being developed that exploit the perceptual properties of the ear [45].
2.2.4 Adaptive Differential Pulse Code Modulation

Adaptive Differential Pulse Code Modulation (ADPCM) is the basis of the ITU G.721 32Kbits/sec telephony standard. A simplified block diagram for ADPCM is shown in Figure 2-2.

ADPCM uses linear prediction to remove redundancy from the speech signal. It achieves this by producing a prediction \((S_k^p)\) of the speech signal \((S_k)\) and subtracts this prediction from the original speech signal to produce an error signal \((e_k)\). This error signal is then quantised \((Y_k)\) and transmitted across the channel. At the decoder this error is dequantised \((Z_k)\) and a prediction signal \((S_k^p)\) is generated and added to it, to obtain a reconstructed signal \((S_k^r)\). The predictors at the encoder and decoder are nominally identical, if initial conditions and channel effects are small.

The adaptive predictor used in the ITU G.721 standard uses a combination of pole and zero components. The pole component is second order with specific constraints on the coefficients to guarantee stability of the decoder. The zero component is sixth order and is used to match the input spectra of speech adequately. The zero coefficients are estimated using a relatively simple gradient algorithm. The adaptation of the pole coefficients is more complicated and involves polarity correlations.
of the output of the quantiser and the sum of the zero components [29].

The quantiser used is a leaking 2-mode, one-word adaptive memory quantiser. The quantiser converts the prediction error into a symbol of the transmission alphabet. The quantiser step-size adapts in response to the input signal amplitude. The dequantiser restores the signal variation. Lower bit-rates are achieved by using 4 bits/symbol compared to 8 bits for the input data. The sampling rate used is 8KHz. More information about ADPCM can be found in [4].

2.3 Speech Signal Models

2.3.1 Linear Predictive Models and State Estimation

Speech is frequently modelled as all-pole filtered white noise which can be represented as

\[ S_k = \left( \frac{1}{1 - \sum_{i=1}^{N} a_i z^{-i}} \right) w_k \]  

(2.3)

where \( S_k \) is the speech signal, \( w_k \) represents the excitation sequence, \( a_i \) are the filter coefficients and \( N \) is the order of the model. The set of filter coefficients for a stationary signal can be computed efficiently from the signal autocorrelation matrix using the Levinson-Durbin algorithm [13].

Using a state space description, the all-pole model for speech can be expressed as

\[ X_k = FX_{k-1} + w_k \]  

(2.4)

\[ S_k = HX_k \]  

(2.5)

where

\[ F = \begin{pmatrix} a_1 & a_2 & \cdots & a_{N-1} & a_N \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \]
\[
\overline{w}_k = \begin{pmatrix}
w_k \\
0 \\
\vdots \\
0
\end{pmatrix}; \quad X_k = \begin{pmatrix}
S_k \\
S_{k-1} \\
S_{k-2} \\
\vdots \\
S_{k-N+1}
\end{pmatrix}; \\
H = \begin{bmatrix}
1 & 0 & \cdots & 0
\end{bmatrix}.
\]

The state vector \(X_k\) consists of \(N\) speech signals from the \(k\)th sample back to the \((k - N + 1)\)th sample.

### 2.4 Linear Predictive Coders and State Estimation

ADPCM utilises linear prediction of samples in order to remove redundancy from the signal for more efficient transmission [3, 14]. It uses a linear predictor to estimate the speech sample at discrete time \(k\) based on the previous \(N\) measurements. The least squares prediction of the sample \(S_k\) based on (2.3) is

\[
\hat{S}_{k|k-1} = a_1S_{k-1} + a_2S_{k-2} + \cdots + a_NS_{k-N}.
\] (2.6)

The standard notation \(x_{k|l}\) represents the estimate of variable \(x\) at time \(k\) using data available up to and including time \(l\). This prediction cannot be used effectively since the decoder does not have access to exact copies of the samples \(S_{k-1}\) to \(S_{k-N}\). The decoder does have reconstructed estimates of these samples available to it and thus the following prediction is used:

\[
\hat{S}_{k|k-1} = a_1\hat{S}_{k-1|k-1} + a_2\hat{S}_{k-2|k-2} + \cdots + a_N\hat{S}_{k-N|k-N} \quad (2.7)
\]

\[
\hat{S}_{k|k} = \hat{S}_{k|k-1} + Q[S_k - \hat{S}_{k|k-1}],
\] (2.8)

where \([S_k - \hat{S}_{k|k-1}]\) is the prediction error and \(Q[\cdot]\) is the quantisation function (\(Q[S_k - \hat{S}_{k|k-1}]\) is effectively the dequantised received signal). This can also be
expressed in state space form as

\[ \hat{S}_{k|k-1} = HF\hat{X}_{k-1|k-1} \]

(2.9)

\[ \hat{S}_{k|k} = \hat{S}_{k|k-1} + Q[\hat{S}_k - \hat{S}_{k|k-1}], \]

(2.10)

where

\[ \hat{X}^\text{lp}_{k-1|k-1} = \begin{pmatrix} \hat{S}_{k-1|k-1} \\ \hat{S}_{k-2|k-2} \\ \vdots \\ \hat{S}_{k-N|k-N} \end{pmatrix} \]

is a state estimate vector provided by the linear predictor.

Crisafulli et al. [9] showed how the Kalman filter can be used instead of the standard linear predictor in ADPCM and noted that the performance gain is due to the prediction filter using smoothed estimates of past speech samples. From [9, 8] the Kalman filtering equations for speech estimation based on the all-pole model (2.3) are

\[ \hat{X}_{k|k} = \hat{X}_{k|k-1} + K_kQ[\hat{S}_k - \hat{S}_{k|k-1}] \quad \text{(measurement update)} \]

(2.11)

\[ \hat{X}_{k+1|k} = F\hat{X}_{k|k} \quad \text{(time update)} \]

(2.12)

\[ \hat{S}_{k|k-1} = H\hat{X}_{k|k-1} \]

(2.13)

where

\[ K_k = P_kH^T \left( HP_kH^T + R_k \right)^{-1} \]

(2.14)

is the Kalman gain vector and \( P_{k+1} \) is the solution of the Ricatti difference equation (RDE) given by

\[ P_{k+1} = FP_kF^T - FP_kH^T \left( HP_kH^T + R_k \right)^{-1} HP_kF^T + Q_k. \]

(2.15)
The quantity $R_k$, is the measurement noise variance which is used to model the quantisation distortion of the prediction signal $[S_k - \hat{S}_{k|k-1}] \equiv n_k$. That is,

$$R_k = E[n_k^2] = \sigma_{n,k}^2,$$  \hspace{1cm} (2.16)

and

$$Q_k = E[\bar{w}_k\bar{w}_k^T] = \begin{pmatrix} \sigma_{w,k}^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$  \hspace{1cm} (2.17)

is the excitation noise variance matrix. The state estimate vector for the Kalman Filter is

$$\hat{X}_{k|k}^k = \begin{pmatrix} \hat{S}_{k|k} \\ \hat{S}_{k-1|k} \\ \vdots \\ \hat{S}_{k-N+1|k} \end{pmatrix}.$$  

Observe that the state vector $X_{k|k}$ contains the $N$ previous speech samples. The estimate $\hat{S}_{k|k}$ is known as the filtered estimate and the estimates $\hat{S}_{k-1|k}$ to $\hat{S}_{k-N+1|k}$ are known as smoothed estimates. The smoothed reconstructed speech samples can simply be extracted from the state vector. The price for this is the computation of the vector $K_k$, which replaces the vector gain $H = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ of linear prediction.

2.5 Down-sampled Speech Models and Reconstruction

By down-sampling, we can use less frequent transmission of measurements but at a higher resolution as opposed to sending every measurement at a lower resolution. The bandwidth of speech is typically 3.7KHz and the sampling rate used is 8KHz.

The ITU G.721 telephony standard based on ADPCM transmits at 32Kbits/sec.
With down-sampling we could equivalently transmit every second measurement at 8 bits (4KHz×8bit) and use a Kalman filter to reconstruct the missing measurements. The Kalman filter exploits the correlation between samples, given by the all-pole model and noise statistics information to obtain a smoothed estimate of the input sample. The Kalman filtering algorithm is well suited to down-sampling since it has the ability to take account of the missing measurements. The algorithm effectively does this by setting the measurement noise variance $R_k$, equal to infinity, which corresponds to zero quantisation levels (i.e. no information contained in this measurement). From (2.14), by setting the measurement noise variance equal to infinity, the Kalman gain $K_k$, is set to zero. In practice this means the Kalman filter performs the time update part of the algorithm (2.12) twice before looping back to the measurement update part of the algorithm (2.11).

2.6 Theoretical Performance Analysis

2.6.1 Error Covariance Matrices

In order to compare the performance of the down-sampled KF-ADPCM system with a KF-ADPCM system without down-sampling (referred to as the full-rate system) we examine the error covariance matrices for both systems. These are given by the solution of the RDE (2.15) and are a measure of the quality of the estimate $\hat{x}_{k|k-1}$ for ideal stationary signals. Information about the convergence and monotonicity properties of Ricatti equation solutions can be found in [5, Chapter 10.6]. We analyse the performance of a system which only transmits every second measurement (referred to as the half-rate system). In order to enable a fair comparison between the two systems we add the requirement that both of the systems operate at the same bit-rate. Thus the half-rate system transmits only every second sample but
with twice the number of bits.

However if we just consider one frame of speech for the half-rate system, it can be seen that the steady state solution of the RDE will alternate between two error covariance matrices, $P_{2k-1|2k-2}$ and $P_{2k|2k-2}$ (corresponding to the two time updates following a measurement update). Thus both of these will need to be considered when comparing the half-rate system to the full-rate system. To enable this to be done a stationary blocked model for the speech signal $S_k$ can be used. The blocked model produces an error covariance matrix that contains both error covariance matrices, $P_{2k-1|2k-2}$ and $P_{2k|2k-2}$.

### 2.6.2 Stationary Blocked Model

The stationary blocked model for the speech signal $S_k$ is shown below

\[
\begin{bmatrix}
x_{2k+2} \\
x_{2k+1}
\end{bmatrix} =
\begin{bmatrix}
F^2 & 0 \\
F & 0
\end{bmatrix}
\begin{bmatrix}
x_{2k} \\
x_{2k-1}
\end{bmatrix} +
\begin{bmatrix}
G & FG \\
0 & G
\end{bmatrix}
\begin{bmatrix}
w_{2k+1} \\
w_2k
\end{bmatrix}
\]

(2.18)

\[
\begin{bmatrix}
S_{2k} \\
S_{2k-1}
\end{bmatrix} =
\begin{bmatrix}
H & 0 \\
0 & H
\end{bmatrix}
\begin{bmatrix}
x_{2k} \\
x_{2k-1}
\end{bmatrix}
\]

(2.19)

where $G = [1 \ 0 \ \cdots \ 0]^T$.

If we represent $\tilde{S}_{k|k}$ by $Z_k$, the signal measurement model for the full-rate and represent $\tilde{S}_{k|k}$ by $\tilde{Z}_k$, the signal measurement model for the half-rate system then:

\[
\begin{bmatrix}
x_{2k+2} \\
x_{2k+1}
\end{bmatrix} =
\begin{bmatrix}
F^2 & 0 \\
F & 0
\end{bmatrix}
\begin{bmatrix}
x_{2k} \\
x_{2k-1}
\end{bmatrix} +
\begin{bmatrix}
G & FG \\
0 & G
\end{bmatrix}
\begin{bmatrix}
w_{2k+1} \\
w_2k
\end{bmatrix}
\]

(2.20)

\[
\begin{bmatrix}
Z_{2k} \\
Z_{2k-1}
\end{bmatrix} =
\begin{bmatrix}
H & 0 \\
0 & H
\end{bmatrix}
\begin{bmatrix}
x_{2k} \\
x_{2k-1}
\end{bmatrix} +
\begin{bmatrix}
v_{2k} \\
v_{2k-1}
\end{bmatrix}
\]

(2.21)

Full-rate
\[
\begin{bmatrix}
\tilde{Z}_{2k}
\end{bmatrix} =
\begin{bmatrix}
H & 0
\end{bmatrix}
\begin{bmatrix}
x_{2k}
\cr
x_{2k-1}
\end{bmatrix} + \tilde{v}_{2k} \quad \text{Half-rate} \tag{2.22}
\]

The excitation noise variance matrix for both systems is \(\bar{Q}\) and the measurement noise variance for the full-rate system is \(\bar{R}_f\) and half-rate system is \(\bar{R}_h\). Then

\[
\bar{Q} = \begin{bmatrix}
Q & 0 \\
0 & Q
\end{bmatrix}; \quad \bar{R}_f = \begin{bmatrix}
R & 0 \\
0 & R
\end{bmatrix}; \quad \bar{R}_h = \begin{bmatrix}
R/2^n & 0 \\
0 & \infty
\end{bmatrix},
\]

where \(n\) is the number of bits used to transmit the error signal across the channel for the half-rate system. Since both systems are operating at the same bit-rate, the full-rate system will be using \(n/2\) bits to transmit the error signal across the channel. If the full-rate system was using 2 bits/sample and the measurement noise variance was \(R\) then for the half-rate system every second measurement noise variance would be \(R/16\).

The blocked model produces an error covariance matrix of the form

\[
P_m = \begin{bmatrix}
P_1 & * \\
* & P_2
\end{bmatrix} \tag{2.23}
\]

where

\[
P_1 = E \left( \tilde{x}_{2k/2k-2} - x_{2k} \right) \left( \tilde{x}_{2k/2k-2} - x_{2k} \right)^T \tag{2.24}
\]

\[
P_2 = E \left( \tilde{x}_{2k-1/2k-2} - x_{2k-1} \right) \left( \tilde{x}_{2k-1/2k-2} - x_{2k-1} \right)^T. \tag{2.25}
\]

Thus for the full-rate system the only part of the error covariance matrix that contains meaningful information is \(P_2\) which is the one-step ahead error covariance matrix. However to analyse the performance of the half-rate system, \(P_1\) the two-step ahead prediction covariance matrix must be considered in addition to \(P_2\). The diagonal elements of the two sub-matrices \(P_1\) and \(P_2\) contain the information in which we are interested. The first diagonal element, \(P_{11}\), is the variance of the error for the predicted estimate. The second diagonal element \(P_{22}\), is the variance of the
error for the filtered estimate and the remaining diagonal elements are the error variance for the smoothed estimates to various smoothing lags. There also exists a relationship between the diagonal elements of $P_1$ and $P_2$ such that

$$P_{2}^{t,i} = P_{1}^{t+1,i+1} \quad (2.26)$$

This is due to the form of the $F$ matrix.

2.6.3 Theoretical Results

In order to test the feasibility of the idea of down-sampling, data was generated by choosing pole locations and modelling $n_k$ and $w_k$ by white noise with variances $R$ and $\sigma_{w,k}^2$ respectively. Thus the matrices $Q$ and $F$ and the variance $R$ are all known exactly for the analysis. This data was used to compute the matrix $P_m$. The diagonal elements from the full-rate and half-rate error covariance matrices were compared to obtain information about the performance of the two systems. A fourth order system with the poles all located equal distance $D$ from the origin was used to generate the data. The poles were located at $\pm D$ and $2^{-\frac{1}{2}}(D \pm jD)$. The variance of the measurement noise, $R$, is the variance of the quantisation distortion. The power of the quantisation noise for a uniform mid-rise quantiser, assuming that the quantisation error is equally likely to lie anywhere in the range $(-\Delta/2, \Delta/2)$, is given by

$$R = \frac{\Delta^2}{12} \quad (2.27)$$

The power of the signal generated depends on both the pole locations and the variance of the process noise, $\sigma_{w,k}^2$. By Parseval’s Theorem

$$\text{Power} = \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \sigma_{w,k}^2 d\omega = \sum_{i=1}^{\infty} h_i^2 \sigma_{w,k}^2 \quad (2.28)$$

where $\{h_i\}_{i=0}^{\infty}$ is the impulse response of the system. Thus to keep the power of the signal constant for different pole locations we have to change $Q$ according to (2.28).
It was found that with poles close to the unit circle and the process noise, \( Q \) less than the measurement noise, \( R \) the performance of the half-rate system was substantially better than that of the full-rate system when smoothing was used. With poles close to the unit circle the simulated data has similar temporal and spectral properties to voiced speech. In voiced speech the energy is concentrated in the low frequency region with considerable redundancy both short term and long term. Unvoiced speech is harder to predict with energy spread over the whole bandwidth and corresponds to poles close to the origin. With poles close to the origin the performance of both systems dropped dramatically and the half-rate system rarely outperformed the full-rate system as measured by error covariance.

For example, when the power of the signal was kept constant at 0.2, the full-rate system transmitting each measurement using 2 bits and the half-rate system transmitting at 4 bits and the poles located at a distance of 0.9 from the origin, both the corresponding diagonal elements of \( P_1 \) and \( P_2 \) of the half-rate system were less than that of the full-rate system after the first smoothing lag. However when the poles are located at a distance of 0.4 from the origin, one of the respective error covariances for the half-rate system was always worse than that of the full-rate system even when smoothing is used.

It is also possible to have signals for which the half-rate system performs better than the full-rate system even on the predicted and filtered estimate. One such example is:

\[
F = \begin{bmatrix} 1 - \epsilon & 1 \\ 0 & \epsilon \end{bmatrix}; \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}
\]

with \( \sigma_w^2 = 0.01, R_f = 0.01, R_d = 0.0025 \) and \( \epsilon = 1 \times 10^{-6} \).

The smoothed estimates for this system were found experimentally to be substantially better than the predicted and filtered estimates especially for the half-rate system.
The relative performance of the two systems depends greatly on the characteristics of the input signal. The half-rate system performs better when the poles are located close to the unit circle and the process noise is small. Whereas the full-rate system performs better when the poles are located close to the origin.

2.7 Conclusion

In this chapter we have presented a new algorithm for reducing the bit-rate in a version of the Adaptive Differential Pulse Code Modulation coder while maintaining the quality of the output speech. The algorithm down-samples the transmitted error data and uses a Kalman filter at the decoder for a forwards adaptive system (or at the encoder and decoder for a backwards adaptive system) to produce smoothed estimates of the speech signal.

We have also presented a theoretical method for analysing the performance of this new algorithm. This was achieved by examining the error covariance matrix which is given by the solution of the Ricatti difference equation. If the down-sampled system transmits only every second error measurement, it can be shown that the steady state solution of the Ricatti difference equation will alternate between two error covariance matrices for one frame of speech. Both of these error covariance matrices will need to be considered when examining the performance of this half-rate system. To enable this to be done we introduced a blocked model for the speech signal and this blocked model produces an error covariance matrix that contains both error covariance matrices.

Using this theoretical method for analysing the performance of the down-sampled system we compared a half-rate system and full-rate system operating at the same bit-rate. This comparison indicates that the half-rate system can outperform the full-rate system when smoothing is applied for certain classes of signals. In particular
the half-rate system was found to have better performance on signals which have similar properties to voiced speech.

Chapter 3

Performance of Down-sampled ADPCM

3.1 Introduction

In this chapter we present a comparison of the performance of the down-sampled KE-ADPCM system and the full-sampled KE-ADPCM system using real speech data. We first consider the parameters that are required to be estimated for the implementation of the down-sampled KE-ADPCM system. We then compare the performance of the full-sampled and down-sampled systems operating both in a forwards and backwards adaptive manner. These comparisons are done using both objective and subjective methods.

3.2 Implementation of the Algorithm

The parameters that we are required to estimate for the implementation of the down-sampled KE-ADPCM system are the filter coefficients (i.e., the L vectors),
Chapter 3

Performance of Down-sampled ADPCM

3.1 Introduction

In this chapter we present a comparison of the performance of the down-sampled KF-ADPCM system and the full-sampled KF-ADPCM system using real speech data. We first consider the parameters that are required to be estimated for the implementation of the down-sampled KF-ADPCM system. We then compare the performance of the full-sampled and down-sampled systems operating both in a forwards and backwards adaptive method. These comparisons are done using both objective and subjective methods.

3.2 Implementation of the Algorithm

The parameters that we are required to estimate for the implementation of the down-sampled KF-ADPCM system are the filter coefficients (i.e. the $F$ matrix),
the process noise covariance matrix, $Q$ and the measurement noise variance, $R$.

For a stationary signal the set of $a_i$ coefficients can be computed efficiently from the signal autocorrelation matrix through the use of the Levinson-Durbin algorithm. Although speech is non-stationary it is reasonable to assume piecewise stationarity for segments up to 20ms. The Levinson-Durbin algorithm is a direct method for computing the filter coefficients by solving the Yule-Walker prediction equations. The Yule-Walker prediction equations are given by

$$
\begin{bmatrix}
R_{xx}(1) \\
R_{xx}(2) \\
\vdots \\
R_{xx}(N)
\end{bmatrix}
= 
\begin{bmatrix}
R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(N-1) \\
R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(N-2) \\
\vdots & \vdots & \ddots & \vdots \\
R_{xx}(N-1) & R_{xx}(N-2) & \cdots & R_{xx}(0)
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_N
\end{bmatrix}
$$

(3.1)

where

$$
R_{xx}(k) = E[X(n)X(n + k)]
$$

and $a_i$ are the filter coefficients.

The Levinson-Durbin method is recursive by nature and makes use of the Toeplitz structure of the autocorrelation matrix. The advantage in using the Levinson-Durbin algorithm over other methods is its computational efficiency. This method only requires $2N$ storage locations and $N^2 + O(N)$ operations (multiplications and divisions). From [27] Levinson-Durbin’s algorithm can be written as

$$
E_0 = R_{xx}(0)
$$

(3.2)

$$
k_i = -\frac{R_{xx}(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R_{xx}(i-j)}{E_{i-1}}
$$

(3.3)

$$
a_1^{(i)} = k_i
$$

(3.4)

$$
a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad 1 \leq j \leq i - 1
$$

(3.5)
The above equations are solved recursively for \( i = 1, 2, \ldots, N \). The final solution is given by

\[
a_j = a_j^N, \quad 1 \leq j \leq N.
\]  

The quantity \( R_k \), is the variance of the measurement noise which in our case is the variance of the quantisation error. The quantisation error can be modelled by uniformly distributed zero mean white noise under certain reasonable conditions [19]. In this case

\[
R_k = \frac{\Delta_k^2}{12},
\]  

where \( \Delta_k \) is the quantiser step-size.

There are two methods one could use to estimate the process noise covariance matrix, \( Q_k \). From (2.17) the only element to estimate is \( \sigma_{w,k}^2 \), the excitation noise sequence variance. If \( w_k \) is considered as zero mean white noise with a non-stationary variance then it is possible to obtain an estimate by estimating the variance of \([S_k - \hat{S}_{k|k-1}]\) since the statistics of this are the same as those of the process noise. Thus a recursive variance estimator can be used

\[
\hat{\sigma}_{w,k}^2 = \beta \hat{\sigma}_{w,k-1}^2 + (1 - \beta) [S_k - \hat{S}_{k|k-1}]^2.
\]  

where \( \beta \) is some constant less than but close to 1.

The second method uses the first \( N \) elements of the signal autocorrelation matrix which is calculated in any case for the Levinson-Durbin algorithm. We can rewrite equation (2.3) as

\[
S_k = \sum_{i=1}^{N} a_i S_{k-i} + w_k.
\]
To consider the autocorrelation function of the output process $S_k$, we multiply both sides of equation (3.10) by $S_k$ and take expected values. We also note that $w_k$ is independent of the previous values $S_i$, ($i < k$). Thus (3.10) becomes

$$\sigma_S^2 = R_{SS}(0) = \sum_{i=1}^{N} a_i R_{SS}(i) + \sigma_w^2$$

(3.11)

where $R_{SS}(i)$ is the autocorrelation of $S$ at lag $i$. This allows the excitation noise sequence variance for each frame of speech to be calculated from the signal autocorrelation matrix.

A comparison of the accuracy of both estimators in estimating the process noise covariance matrix, $Q_k$ was performed. The results indicate that the method of estimating the process noise covariance matrix, $Q_k$ using the first $N$ elements of the autocorrelation matrix produced a slightly more accurate representation of the process noise covariance matrix. Thus this method is used in all tests on the downsampled KF-ADPCM system.

### 3.3 Forwards-Adaptive System

#### 3.3.1 Algorithm

The tests done in this section used a forwards adaptive coding scheme. A simplified block diagram is shown in Figure 3-1.

The quantity $\sigma^2_{w,k}$, the process noise covariance matrix for each frame of speech, was estimated using the first $N$ elements of the autocorrelation matrix using (3.11).

This estimate was multiplied by five. This is a standard method in Kalman filtering of improving filter stability at the expense of optimality [1], Chapter 6.2. The parameters of the coders are not quantised. Both full-rate and half-rate methods use double precision parameters.
The quantiser was a mid-rise backwards adaptive quantiser based on Jayant's One-Word Memory algorithm [19]. The multipliers were those recommended for DPCM in Table 4.14 of [19]. A filter of order ten and a smoothing lag of three was used.

### 3.3.2 Results

In order to illustrate how both the half-rate and the full-rate systems operate, a small segment (15 samples) of speech (represented by the solid line) and the predicted (represented by the crosses), filtered (represented by the circles) and smoothed estimates (represented by asterisks) of this segment of speech are plotted in Figure 3-2. In the full-sampled system (8KHz×2bit), it can be seen that there is considerable gain in accuracy in reconstructing the original speech signal when the filtered estimates are used over the predicted estimates but only marginal improvement in accuracy in using the smoothed estimates over the filtered estimates.

In the down-sampled system (4KHz×4bit), it can be seen that every second predicted and filtered estimates are the same. This occurs because at these time instances, no measurements were sent and thus the filtered estimates have no extra information contained in them over the predicted estimates. It is at these time in-
stances that the advantage in using smoothing in the down-sampled system can be seen clearly. There is a large improvement in the accuracy of the smoothed estimates over the filtered and predicted estimates at these time instances (corresponding to even numbered samples). At the odd numbered samples there is only a marginal gain in accuracy in reconstructing the original speech signal using the smoothed estimates over the filtered estimates.

Overall the smoothed estimates from the down-sampled system reconstruct the original speech signal better than the smoothed estimates from the full-sampled system. The improvement in performance in using smoothing over prediction on frequent, lower quality measurements as in the full-rate system is much smaller than using smoothing on less frequent, higher quality measurements as in the down-sampled system.

The forwards adaptive KF-ADPCM system was tested using 2 bits per sample for the full-rate system and 4 bits per sample for the half-rate system on a 6 second segment of speech from a male speaker and a 7 second segment of speech from a female speaker. The sentence used for both speakers was “Mechanical, acoustical or electrical vibrations are the sources of sounds in musical instruments”. The speech signal from the male speaker along with its spectrogram and the segmental SNR for both the full-sampled and down-sampled system is plotted in Figure 3-3. From these figures it appears that the down-sampled system performs better for voiced speech in agreement with the expected performance from the theoretical analysis.

The resulting SNRs and segmental SNRs are shown in Table 3.1.

From these results it can be seen that there is a substantial advantage in using the smoothed estimate over the predicted and the filtered estimates especially for the half-rate system. Informal listening tests have also corroborated these results. A group of four untrained listeners were used to evaluate the subjective quality of the above-mentioned segments of speech using the Mean Opinion Score (MOS) [19]
Figure 3-2: Comparison of the full-sampled and down-sampled KF-ADPCM systems.
Figure 3-3: Speech signal from the male speaker, its spectrogram and the resulting segmental SNR for the full and down-sampled systems.

and randomised trials. The results are shown in Figure 3-4. From these graphs, the improvement in quality of the half-rate KF-ADPCM system over the full-rate KF-ADPCM system with smoothing can be seen.

3.3.3 Cost of Forwards-Adaptation

The linear predictive coding (LPC) coefficients are typically obtained at the rate of 50 frames/sec. There exist three commonly used methods for quantising the LPC coefficients for transmission in a forwards adaptive system. These are scalar quantisation, vector quantisation and vector-scalar quantisation.
<table>
<thead>
<tr>
<th>Male Speaker</th>
<th>Full-sampled</th>
<th>Down-sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR(dB)</td>
<td>SNRSEG(dB)</td>
</tr>
<tr>
<td>Predicted Output</td>
<td>7.14</td>
<td>8.24</td>
</tr>
<tr>
<td>Filtered Output</td>
<td>13.10</td>
<td>14.40</td>
</tr>
<tr>
<td>Smoothed Output</td>
<td>13.76</td>
<td>14.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Female Speaker</th>
<th>Full-sampled</th>
<th>Down-sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR(dB)</td>
<td>SNRSEG(dB)</td>
</tr>
<tr>
<td>Predicted Output</td>
<td>9.82</td>
<td>9.85</td>
</tr>
<tr>
<td>Filtered Output</td>
<td>15.77</td>
<td>16.01</td>
</tr>
<tr>
<td>Smoothed Output</td>
<td>16.64</td>
<td>16.38</td>
</tr>
</tbody>
</table>

Table 3.1: SNR Measures for a Segment of Speech from a Male and a Female Speaker

Scalar quantisation, quantises the LPC coefficients individually. Different sets of parameters that represent the LPC coefficients have been used. These include log area ratios [41], arc sine reflection coefficients [15] and the line spectral frequency representation [18]. Studies [2] have suggested that 32-40 bits are necessary to quantise each frame of LPC information with reasonable accuracy.

Vector quantisation [22] considers the entire set of LPC parameters and allows for direct minimisation of quantisation distortion. Hence vector quantisers result in smaller distortion than the scalar quantisers at a given bit-rate but have a much higher computational complexity. For high quality speech coders the memory required to store the vector codebook and the number of computations used in comparing the input vector to each codebook vector is prohibitive.

The last method, vector-scalar quantisation attempts to exploit the advantages offered by vector quantisation while reducing the practical problems of it. A split vector quantisation approach [32] has been reported to be able to quantise LPC in-
formation in 24 bits/frame with 1dB average spectral distortion. This method uses a LPC vector consisting of 10 line spectral frequencies and this vector is divided into two parts and each part is quantised separately. The results for the forwards adaptive system in the previous section used frames of length 20ms and an overlap of 50%. Using the split vector quantisation method, an extra 2.4Kbits/sec is required to transmit the LPC information to the decoder.

3.4 Backwards Adaptive System

The results in the previous section are based on a forwards adaptive system. In a forwards adaptive system the filter coefficients are computed based on the input speech and these filter coefficients are transmitted to the decoder as side information, thus requiring extra bit-rate. Backwards adaptive systems make use of the reconstructed signal that is available at the decoder and use this to produce a set of filter coefficients for the prediction process. In this section we examine the performance of two different methods for calculating the filter coefficients based on a backwards adaptive scheme. The first uses the Levinson-Durbin algorithm discussed in the previous section and the second method uses a recursive least squares estimator to
calculate the filter coefficients.

All test results in the following sections use a filter of order ten and a smoothing lag of three. The quantiser used is a mid-rise backwards adaptive quantiser based on Jayant’s one-word memory algorithm. The down-sampled system transmits every second sample using 4 bits while the full-sampled system transmits every sample using 2 bits. Thus both system are operating at 16Kbits/sec.

3.4.1 Levinson-Durbin Algorithm

The filter coefficients for the speech model were estimated from the signal autocorrelation matrix through the use of the Levinson-Durbin algorithm (discussed previously in Section 3.2). Initially the autocorrelation matrix was estimated using the previous frame of predicted outputs from the Kalman filter. The resulting SNRs and SEGSNRs for a segment of speech from a male speaker are shown in Table 3.2

<table>
<thead>
<tr>
<th>Male Speaker</th>
<th>Full-sampled</th>
<th></th>
<th>Down-sampled</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR(dB)</td>
<td>SNRSEG(dB)</td>
<td>SNR(dB)</td>
<td>SNRSEG(dB)</td>
</tr>
<tr>
<td>Predicted Output</td>
<td>5.25</td>
<td>6.24</td>
<td>4.69</td>
<td>5.67</td>
</tr>
<tr>
<td>Filtered Output</td>
<td>9.20</td>
<td>11.10</td>
<td>7.83</td>
<td>9.33</td>
</tr>
<tr>
<td>Smoothed Output</td>
<td>9.69</td>
<td>11.62</td>
<td>9.11</td>
<td>11.54</td>
</tr>
</tbody>
</table>

Table 3.2: SNR Measures for a Segment of Speech from a Male Speaker

The frame length used in these tests is 10ms. Informal listening tests indicate that the quality of both the full-sampled system and the down-sampled system are noticeably inferior to those from the forwards adaptive system. The previous frame of smoothed outputs from the Kalman filter should be able to the provide better estimates of the autocorrelation matrix than the predictions. Thus the previous frame of smoothed outputs was used to provide the estimated autocorrelation matrix. This change in estimating the filter coefficients brought about significant improvement in
results. These results are shown in Table 3.3. The output of the full-sampled system

<table>
<thead>
<tr>
<th>Male Speaker</th>
<th>Full-sampled</th>
<th>Down-sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR(dB)</td>
<td>SNRSEG(dB)</td>
</tr>
<tr>
<td>Predicted Output</td>
<td>5.98</td>
<td>7.15</td>
</tr>
<tr>
<td>Filtered Output</td>
<td>10.15</td>
<td>12.55</td>
</tr>
<tr>
<td>Smoothed Output</td>
<td>10.70</td>
<td>13.21</td>
</tr>
</tbody>
</table>

Table 3.3: SNR Measures for a Segment of Speech from a Male Speaker

still has a noticeable amount of quantisation noise in the output. While the output of the down-sampled system is cleaner in most parts, there is some parts of the output speech which have a noticeable aliasing effect. Since the Levinson-Durbin algorithm is a block orientated scheme which relies on the autocorrelation matrix, it is possibly not the ideal algorithm to use in estimating the filter coefficients for the speech model. A recursive least squares estimator may produce improvements in the performance of the two systems.

3.4.2 Recursive Least Squares

The recursive least squares estimator [39, 35] was used to estimate the filter coefficients for the all-pole speech model. The estimator often became unstable when the input signals characteristics changed rapidly (i.e. the transition from unvoiced to voiced speech). Two methods were investigated to prevent the estimator becoming unstable. The first method involved a reset back to initial conditions when the estimator was on the verge of becoming unstable. With this reset in place both the full-sampled and down-sampled systems had very poor subjective quality. The second method involved introducing a leakage factor into the recursive estimator. However to prevent the estimator from becoming unstable, the leakage factor had to be made too large for the overall system to maintain a reasonable level of performance.
The stability and performance of the recursive least-squares lattice estimator was then examined. From [25] the square-root normalised least squares algorithm is:

for each new data sample $x_T$

$$R_T = \lambda R_{T-1} + x_T^2$$ (3.12)

$$\nu_{0,T} = \eta_{0,T} = \frac{x_T}{\sqrt{R_T}}$$ (3.13)

where $R_T$ is the estimated variance of $x$, $\rho_{i,T}$ are the reflection coefficients, $\nu_{i,t}$ are the normalised forward prediction errors, $\lambda$ is the exponential weighting factor and $\eta_{i,T}$ are the normalised prediction errors.

For each stage of the lattice, $i = 0$ to $\min(T,P) - 1$

$$\rho_{i+1,T} = \sqrt{1 - \nu_{i,T}^2 \sqrt{1 - \eta_{i,T}^2}} \rho_{i+1,T-1} + \nu_{i,T} \eta_{i,T-1}$$ (3.14)

$$\nu_{i+1,T} = \frac{\nu_{i,T} - \rho_{i+1,T} \eta_{i,T-1}}{\sqrt{1 - \rho_{i+1,T}^2} \sqrt{1 - \eta_{i,T}^2}}$$ (3.15)

$$\eta_{i+1,T} = \frac{\eta_{i,T-1} - \rho_{i+1,T} \nu_{i,T}}{\sqrt{1 - \rho_{i+1,T}^2} \sqrt{1 - \eta_{i,T}^2}}$$ (3.16)

This algorithm has similar stability problems to those encountered by the recursive least squares estimator and thus the output of both the full and down-sampled systems using this algorithm to estimate the filter coefficients had poor subjective quality. It appears the recursive schemes examined could not cope well with the time-varying characteristics of the speech signal.

### 3.5 Conclusion

This chapter demonstrated the potential of the down-sampled KF-ADPCM system by comparing the performance of the full-sampled KF-ADPCM with the down-sampled KF-ADPCM system using real speech data. Both systems were operating
at the same bit-rate and the tests were performed using both subjective and objective measures. The improvement in performance gained by using smoothing in the down-sampled system is clearly illustrated in Figure 3-2. When a measurement is not sent due to down-sampling, the predicted and filtered estimates are the same and the performance gain in using the Kalman filter is derived from its ability to provide smoothed estimates.

With both systems obtaining the filter coefficients for the speech model in forwards adaptive manner, we have shown that the down-sampled system can provide substantial improvements in subjective quality over the full-sampled system. It was also shown that the down-sampled system does perform better for voiced speech than the full-sampled system in agreement with the expected performance from the theoretical analysis in the previous chapter.

Different algorithms for estimating the filter coefficients in a backwards adaptive systems were investigated. None of the algorithms examined produced suitable results but all the recursive schemes examined were simple least-squares estimators. Maximum likelihood estimators or spectrum estimating techniques [39] may be more suitable. Further work needs to be performed to find a suitable estimator which copes adequately with the time varying characteristics of the speech signal. One possible solution is using a bank of Kalman filters and adaptively selecting between them based on prediction performance.
Chapter 4

Failed Sensors in Array Processing

4.1 Introduction

An array comprises a set of sensors located at distinct spatial locations and is used to measure a propagating wavefield. The array samples the wavefield at each sensor location and at time instances \( \{t_n\} \). It is thus sampled in time and space. The outputs of the sensors are combined such that the number of sources of propagating energy and the locations of the sources can be determined. This is commonly referred to as beamforming. There exist many applications for arrays, these include radar, sonar, communications, imaging, geophysical exploration and biomedical engineering [40].

The results presented in this chapter are from a delay-and-sum beamformer applied to the data from a linear equal-spaced array. A brief introduction to these topics is presented in the next section. Further information can be found in [16, 21, 30, 37].

In practice sensors malfunction due to various factors. Besides complete electrical or mechanical failure, sensors which have a faulty gain or excessive self noise are
also referred to as failed sensors. At the failed sensors there will be no reliable
data available for beamforming. Failed sensors can dramatically affect the perfor-
mance of a linear array. They can cause disruption of the beam-pattern, by adding
spurious sidelobes and increasing the overall sidelobe levels. Several algorithms for
compensating for failed sensors have been proposed.

These algorithms can be classified into two main groups, data dependent and data
independent methods. Data dependent algorithms use some form of interpolation
to reconstruct estimates of the measurements at the failed sensors. These include
the linear prediction algorithm [38] and the Fourier interpolation of excitations al-
gorithm [11]. Whereas data independent methods attempt to compensate for the
failed sensors by reweighting the remaining elements of the array in a way which
minimises the effect of the failed elements. These include the Lagrange multiplier
method [46] and the linear programming method [36].

The Lagrange multiplier algorithm produces optimal weighting coefficients which
minimise the total power received by the array from all directions exclusive of the
mainbeam. The linear programming algorithm attempts to reweight the array to
reduce the sidelobe levels to the original design level. This is achieved normally by
slightly widening the mainbeam and generally produces sidelobes which are equiv-
alued.

The Fourier interpolation of excitations (FIX) algorithm uses excitation information
from all of the working sensors to estimate the complex Fourier coefficients at the
location of the failed sensors. The algorithm first beamforms the data and converts
the signals to power and these powers are averaged to obtain a threshold. The
beams which have power greater than this threshold are kept and the remaining
ones are set to zero. The resultant is inverse beamformed to produce estimates of
the excitations of the failed array elements. These are substituted into the original
data and this process is repeated to refine further the excitation estimates.
The linear prediction method [38] claims to have performance similar to the FIX algorithm but has less computational complexity than FIX. It uses linear prediction at each time instance to reconstruct estimates of the measurements for the failed sensors from the available data from the working sensors. Where possible it averages both forwards and backwards (along the array) linear predictions to obtain the final estimate.

In this chapter we will introduce another data dependent method that uses a Kalman filter to reconstruct estimates of the measurements at the failed sensors. This algorithm is similar to the one used to reconstruct the missing samples in the downsampled speech signal discussed in the previous chapters. We will compare the performance of this algorithm to the existing linear prediction method.

4.2 Delay-and-Sum Beamforming

The positioning of the sensors within an array determines both the resolution and the processing algorithm which can be applied to the array data. The sensors are sometimes positioned an equal distance apart along a linear array. This positioning of the sensors permits fast algorithms, such as the fast Fourier transform to be used on the array data.

One powerful and commonly used array signal processing technique is, delay-and-sum beamforming. The beamformer delays and weights the outputs of the sensors by appropriate amounts and these are added together to reinforce the signal with respect to noise or waves propagating in other directions. The delays that reinforce the signal are directly related to the length of time required for the signal to propagate between sensors. The output of the delay-and-sum beamformer is given by

\[
y(t) = \sum_{m=0}^{M-1} a_m x_m(t - \Delta_m) \tag{4.1}
\]
where $M$ is the number of sensors, $x_m$ is the waveform measured at the $m$th sensor, $a_m$ is the weighting function applied to the output of each sensor and $\Delta_m$ is the delay applied to the $m$th sensor. Weighting functions such as the Hanning, Hamming, Blackmann and Bartlett windows are all commonly used. These weighting functions have the effect of lowering the sidelobes at the expense of widening the mainbeam in the beam pattern for the array.

For example, if we have a complex sinusoid of angular frequency, $\omega$ propagating through space as a plane wave at angle, $\phi$ relative to a linear equal spaced array as shown in Figure 4-1.

![Figure 4-1: Projection of a plane wave on a linear array](image)

The plane wave has to travel an additional distance to reach the third sensor relative to the first sensor (distance $u$ in Figure 4-1) and hence there is an equivalent time delay involved in the plane wave reaching this sensor. For the $m$th sensor this time delay, $\tau_m$ is given by

$$\tau_m = \frac{\sin \phi (m - 1)d}{c}$$

(4.2)

where $c$ is the speed of light and $d$ is the distance between adjacent sensors. These time delays, $\tau_m$ would be the appropriate time delays to be used in the delay-and-sum beamformer (4.1) for this array geometry.

This conventional time domain approach for beamforming can be replaced by an
equivalent frequency domain approach which uses the fast Fourier transform \[21\]. It is given by

$$X_{nl} = \sum_{m=0}^{M-1} \sum_{i=0}^{N-1} a_m x_{mi} e^{-j\frac{\pi}{M} m} e^{-j\frac{\pi}{M} i}$$  \hspace{1cm} (4.3)

where \(x_{mi}\) is the output of the \(m\)th sensor at the \(i\)th time instant and \(N\) is the number of time samples used and \(X_{nl}\) is the complex beam pattern. This is equivalent to first performing a discrete Fourier transform on a block of data from each sensor and then processing the resulting data across the array at each frequency bin by performing another discrete Fourier transform (this time in the spatial domain) to produce the complex beam pattern at each frequency bin.

With failed sensors these coherent signals are replaced by zero or by noise which distorts the beamforming gains. There exists several approaches to reconstruct the missing data at the failed sensors. We shall apply techniques similar to those of the previous chapters used to reconstruct down-sampled speech signals.

### 4.3 LPC Algorithm

The linear prediction method proposed in \[38\] can be applied to estimate the data at the failed sensors. The failed sensors can usually be identified by comparing the broadband power at each sensor to the average power of all sensors. The algorithm linearly combines both the forwards and backwards linear predictions when they are available. The forwards prediction for the \(k\)th faulty sensor is given by

$$\hat{X}_k^f = \sum_{j=1}^{p} X_{k-j} a_j \hspace{1cm} p + 1 \leq k \leq M$$  \hspace{1cm} (4.4)

where \(M\) is the number of sensors and \(a_j\) are the coefficients for a \(p\)th order model.

The backward prediction is given by

$$\hat{X}_k^b = \sum_{j=1}^{p} X_{k+j} a_j^* \hspace{1cm} 1 \leq k \leq M - p$$  \hspace{1cm} (4.5)
The final estimate used is a linear combination of the backwards and forwards predictions given by

$$\hat{X}_k = \frac{1}{2}(\hat{X}_k^f + \hat{X}_k^b)$$  \hspace{1cm} (4.6)

If either of the forwards or backwards estimate is not available for a particular sensor, the estimate which is available is used. This method cannot be used unless there is at least one segment of working sensors of length $p$.

### 4.4 Kalman Filtering Algorithm

The algorithm proposed in this section for estimating the measurements at the failed sensors is similar to the algorithm discussed in the previous chapters for reconstructing the missing samples in the down-sampled speech signal. The algorithm uses a Kalman filter to reconstruct the measurements at the failed sensors. The algorithm requires that the failed sensors have been identified.

The first step in the algorithm is to obtain a model for the measurements received at all of the sensors. We choose to use an all-pole model. It has the advantage that smoothed estimates can be obtained at no additional computational complexity. The filter coefficients, $a_i$ for the model are estimated from the signal autocorrelation matrix through the use of the Levinson-Durbin algorithm (discussed in Section 3.2).

Using a state-space description, the all-pole model of order, $p$ for the measurements at the sensors can be written as

$$X_k = F X_{k-1} + \bar{w}_k$$  \hspace{1cm} (4.7)

$$S_k = H X_k$$  \hspace{1cm} (4.8)
where
\[
F = \begin{pmatrix}
a_1 & a_2 & \cdots & a_{p-1} & a_p \\
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0 \\
\end{pmatrix}
\]
\[
\bar{w}_k = \begin{pmatrix}
w_k \\
0 \\
\vdots \\
0 \\
\end{pmatrix}; \quad X_k = \begin{pmatrix}
S_k \\
S_{k-1} \\
S_{k-2} \\
\vdots \\
S_{k-p+1}
\end{pmatrix};
\]
\[
H = \begin{bmatrix}
1 & 0 & \cdots & 0
\end{bmatrix}.
\]
The state vector \(X_k\) consists of \(p\) sensor measurements from the \(k\)th sensor back to the \((k - p + 1)\)th sensor. The “time” index \(k\) evolves along the array.

Kalman filtering is applied to the sensor measurements along the array using
\[
\dot{X}_{k|k} = \dot{X}_{k|k-1} + K_k Q [S_k - \hat{S}_{k|k-1}] \quad \text{(measurement update)} \tag{4.9}
\]
\[
\dot{X}_{k+1|k} = F \dot{X}_{k|k} \quad \text{ (“time” update)} \tag{4.10}
\]
\[
\hat{S}_{k|k-1} = H \dot{X}_{k|k-1} \tag{4.11}
\]
where
\[
K_k = P_k H^T (HP_k H^T + R_k)^{-1} \tag{4.12}
\]
is the Kalman gain vector and \(P_{k+1}\) is the solution of the Ricatti difference equation (RDE) given by
\[
P_{k+1} = FP_k F^T - FP_k H^T (HP_k H^T + R_k)^{-1} HP_k F^T + Q_k. \tag{4.13}
\]
The quantity $R_k$, is the measurement noise variance and $Q_k$, is the excitation noise covariance matrix given by

$$Q_k = E\left[\overline{w}_k \overline{w}_k^T\right] = \begin{pmatrix}
\sigma_{w,k}^2 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 
\end{pmatrix}$$

(4.14)

The only non-zero element in the matrix is $\sigma_{w,k}^2$ and this can be estimated by using

$$\sigma_S^2 = R_{SS}(0) = \sum_{i=1}^{N} a_i R_{SS}(i) + \sigma_{w,k}^2$$

(4.15)

where $R_{SS}(i)$ is the autocorrelation of $S$ at lag $i$ and is calculated in any case for the Levinson-Durbin algorithm.

The measurement noise variance, $R_k$ is set to infinity for the measurement at the failed sensor (indicating no information contained in this measurement). Thus the "time" update part of the algorithm (4.10) is performed twice before looping back to the measurement update (4.9). The filter is continued past the failed sensor to obtain the fixed lag smoothed estimate for the failed sensor. The smoothed estimate for the failed sensor is substituted into the data for the complete array and standard beamforming is carried out. If the failed sensor is positioned such that the fixed lag smoothed estimate for this sensor cannot be obtained by running the Kalman filter along the array data (ie. the sensor is positioned near the end of the array), the filter can be applied first forwards across the array data and once the end of the array is reached the filter can then be applied in the opposite direction across the array data to obtain the smoothed estimate for the failed sensor.

### 4.5 Comparison of the LPC and KF Algorithm

In this section we compare the performance of the Kalman filtering algorithm and the linear predictive coding algorithm for reconstructing the measurements at the
failed sensors. We also examine the performance of the array with the measurements from the failed elements set to zero and the performance of the array if the sensors had not failed. These results are used as reference points for the performance of the two algorithms. We examine the performance of both algorithms with a Hamming window being applied to the data and with a rectangular window applied to the data.

The linear equispaced array used in the simulations consisted of 20 sensors and it was assumed that one of the sensors had failed and that the position of the failed sensor was known. It was noted that the closer the missing sensor is to the middle of the array the more it degrades the performance of the array (resulting in higher sidelobes). The results shown in this section are for the failed sensor being located at position 12 along the array of 20. The order of the Kalman filter and linear predictor used in all simulations is order five. The beam pattern is computed by using the frequency domain approach (4.3).

Figures 4-2, 4-3 and 4-4 show the normalised beam power for a narrowband plane wave arriving from an angle of approximately -0.51 radians for different signal-to-noise ratios. The main-lobe of the beam pattern indicates the phase difference between adjacent sensors. The phase difference between sensors is a function of the frequency of the plane wave and the time delay between sensors and hence the direction of arrival of the wave (4.2). In these experiments the phase difference between sensors is -1.2 radians. Figure 4-2 shows two beam patterns for a SNR of 20dB. The first beam response is for a Hamming window applied to the data and the second beam plot is for a rectangular window applied to the data. Similarly for Figure 4-3 where the SNR is 10dB and Figure 4-4 where the SNR is 3dB. The dot-dash line represents the beam pattern for the array if there had been no failed sensors and the dotted line is the beam pattern for the array when the measurements at the failed sensor have been set to zero. The solid line is the beam pattern for the
array using the Kalman filtering algorithm to reconstruct the measurements at the failed sensor and the dashed line represents the beam response for the array using the linear predictive coding algorithm to reconstruct the measurements at the failed sensor.

From these plots we can see that there is a substantial advantage in using either the Kalman filtering algorithm or the linear predictive coding algorithm in reconstructing the measurements at the failed sensor over setting the measurements at the failed sensor to zero. Both methods regain most of the performance loss from the failed sensor for a SNR of 10dB and 20dB. For the SNR of 3dB the improvement drops to approximately 50% of the difference between the performance of the array with no failed sensors and the performance of the array with the measurements at the failed sensor set to zero.

There exist different methods for evaluating the performance of the array, these include the total power of the beam response, the maximum level of the sidelobes and the average level of the sidelobes. To investigate the performance of the two methods we chose to examine the total power of the normalised beam response (area under the beam pattern) for various signal-to-noise ratios. This is shown in Figure 4-5. The first plot shows the performance of the array using a Hamming window and the second plot indicates the performance of the array using a rectangular window. The same types of lines as in the previous Figures are used to indicate the performance of the different algorithms.

These plots indicate that at SNRs greater than 10dB, the Kalman filtering algorithm recovers almost all of the loss in performance from the failed sensors. At SNRs around 0dB both the LPC and the KF algorithm recover approximately 50% of the loss in performance due to the failed sensor. At these lower SNRs the linear predictive algorithm performs marginally better than the Kalman filtering algorithm. We thought that the Kalman filter could be more sensitive to the modelling
Figure 4-2: Beam Patterns for a SNR of 20dB using a) Hamming window and b) rectangular window.

Figure 4-3: Beam Patterns for a SNR of 10dB using a) Hamming window and b) rectangular window.

Figure 4-4: Beam Patterns for a SNR of 3 dB using a) Hamming window and b) rectangular window.
Figure 4-5: Comparison of the linear predictive and Kalman filtering algorithm using an estimated model for various SNRs for a) Hamming window and b) Rectangular window.
errors than the linear predictive algorithm and these modelling errors would be more apparent at lower SNRs. To investigate this, we examined the performance of the two algorithms using the actual model for the incoming wave. This is shown in Figure 4-6.

From these plots we can see that the performance of both algorithms is almost identical to the performance of the array with no failed sensors.

Lastly we examined the performance of the algorithms for an incoming wave having two sources. The first source is located at an angle of -0.51 radians and the second source has a quarter of the power and is located at an angle of 0.3 radians. Thus the mainlobes should be located at -1.2 radians and 0.72 radians. Again the beam patterns are plotted for different SNRs and for both a Hamming window and a rectangular window. These are shown in Figures 4-7, 4-8 and 4-9.

Again both the linear predictive and the Kalman filtering algorithm regain most of the performance loss due to the failed sensor at SNRs greater than 10dB and at lower SNRs both algorithms regain approximately 50% of the loss due to the failed sensor.

4.6 Conclusion

In this chapter we proposed a new data dependent method for improving the performance of an array with failed sensors. This method uses a Kalman filter to produce fixed lag smoothed estimates of the measurements at the failed sensors. The performance of this algorithm was compared to another commonly used data dependent method which utilises linear predictive coding to produce estimates of the measurements at the failed sensors. Both methods regained most of the performance loss due to the failed sensors at SNRs greater than 10dB. At SNRs around 0dB both algorithms recover approximately 50% of the performance loss due to the failed sensor.
Figure 4.6: Comparison of the linear predictive and Kalman filtering algorithm using the actual model for various SNRs for a) Hamming window and b) Rectangular window.
Figure 4-7: Beam Patterns for a SNR of 20dB using a) Hamming window and b) Rectangular window

Figure 4-8: Beam Patterns for a SNR of 10dB using a) Hamming window and b) Rectangular window

Figure 4-9: Beam Patterns for a SNR of 3 dB using a) Hamming window and b) Rectangular window
sensors. There appears to be very little difference between the performance of the linear predictive coding and Kalman filtering algorithms. The higher computational complexity of the Kalman filter suggests that the linear predictive coding algorithm should be the preferred data dependent algorithm.

Chapter 5

Conclusions

5.1 Conclusions

In the above paragraphs, various algorithms were described and compared. The linear predictive coding algorithm was shown to perform well compared to the Kalman filter algorithm. However, the higher computational complexity of the Kalman filter suggests that the linear predictive coding algorithm should be the preferred data dependent algorithm.
Chapter 5

Conclusions

5.1 Conclusions

In this thesis, we have applied Kalman filtering techniques (in particular ideas from smoothing) to two separate applications that require the reconstruction of sampled measurements:

1. a speech coding system which down-samples the transmitted data
2. an algorithm which reconstructs the measurements at a failed sensor in an array

The first part of the thesis proposes a novel algorithm for reducing the bit-rate in a version of the Adaptive Differential Pulse Code Modulation coder while maintaining the quality of the output speech. This is achieved by down-sampling the transmitted error data and using a Kalman filter to reconstruct the speech signal. The advantage in using this technique over a full-sampled system operating at the same bit-rate is gained by using the Kalman filter to reconstruct smoothed estimates of the speech signal.
These methods are of interest in that they appear to violate the Nyquist’s theorem for sampled signal reconstruction. A 3.7KHz bandwidth signal is reconstructed from a speech coding system which uses a sampling rate of 4KHz. However our goals and assumptions are different to those pertaining to the Nyquist sampling theorem. We only require adequate (approximate) reconstruction of the signal from approximate samples and we have a signal model, which provides more information about the signal of interest than is assumed by Nyquist.

The performance of the down-sampled system was analysed by comparing the performance of a half-rate system \((4KH \times 4\text{bit})\) and a full-rate system \((8KH \times 2\text{bit})\). A method for comparing the theoretical performance of the systems was presented. This was achieved by examining the respective error covariance matrices for the two systems which is given by the solution of the the Ricatti difference equations. For the half-rate system it can be seen that the steady-state solution of the Ricatti difference equation will alternate between two error covariance matrices for one frame of speech. Both of these error covariance matrices will need to be considered when analysing the performance of the half-rate system. To enable this to be done we have presented a block model for the speech signal that produced an error covariance matrix that contains both error covariance matrices. It was shown that the half-rate system can outperform the full-rate for certain classes of signals. In particular the half-rate system performed better on signals which had the similar temporal and spectral properties to voiced speech.

The relative performance of the two systems was also examined using objective and subjective measures on real speech data. With the two systems operating in a forwards adaptive manner the half-rate system was found to offer substantial improvements in subjective quality over the full-rate system.

The second part of this thesis proposes a new algorithm for improving the performance of a linear equally spaced array with failed sensors. The algorithm uses
a Kalman filter to produce smoothed estimates of the measurements at the failed sensors. The performance of this algorithm was compared to an existing algorithm which make use of linear prediction methods. Both algorithms produced similar improvements in performance and at SNRs above 10dB, most of the loss in performance due to the failed sensors is recovered. At SNRs close to 0dB, 50% of the performance loss due to the failed sensors is recovered.

5.2 Future Research Directions

In this section we will present a number of topics which are logical extensions to the work presented in this thesis. These topics could provide future research areas and are listed below.

- The performance of other decimation schemes for the KF-ADPCM system could be investigated. For example only sending every third or fourth sample or possibly transmitting a sequence of data, i.e. sending the first, second, fourth and seventh sample and then repeating this sequence. Some of these schemes may have little practical implementation but could possibly provide an insight into how down-sampling affects systems.

- Many standard speech coding techniques which enhance the subjective performance exist, such as perceptual weighting. These techniques could be added to the down-sampled KF-ADPCM coder and the improvement in the performance of the system could be investigated.

- There exists a method [44] for significantly reducing the number of computations required to implement the Kalman filter and the effect (if any) of this algorithm on the performance of the down-sampled KF-ADPCM system could be examined.
• Dithering techniques are also becoming common-place in applications where quantisation is required to reduce the word-length of audio data. The use of dither can render the total error signal audibly equivalent to steady white noise [24]. The effect of dithering on the down-sampled KF-ADPCM system could be investigated.

• Further applications for Kalman filtering in other speech coding frameworks could be investigated to see whether a similar advantage in using smoothing exists.

• Other methods for estimating the filter coefficients in a backwards adaptive system could be investigated. Some have been suggested in Chapter 3.

• We only analysed the performance of the linear predictive coding and Kalman filtering algorithms for the case of one failed sensor in the array. The performance of the two algorithms could be investigated for multiple failed sensors in an array.

• Fixed-interval smoothing could be used to estimate the measurements at the sensors and the performance of this system could be analysed.

• The algorithms for improving the performance of an array with failed sensors could be tested on real data.
Bibliography


