Efficient System Identification based on Root Cepstral Deconvolution

Sanjeev Sarpal

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Centre for Vision, Speech and Signal Processing
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

April 2003

© Sanjeev Sarpal 2003
Summary

This thesis summarizes approximately three years of research on signal modelling for the purposes of system identification.

Improvements in signal modelling techniques have been encouraged over the years by society's demand for more efficient ways of accessing information. As a consequence, several modelling/compression techniques in both the time domain and the frequency domain have been developed as possible solutions to these problems. Cepstral deconvolution is a frequency domain modelling technique that has been successfully applied to many diverse fields, such as speech and seismic analysis. Thus far, all cepstral modelling performance has been empirical, relying on the judgement of the designer. Therefore a novel method for measuring root cepstral pole-zero modelling performance is proposed, by introducing a cost function applied directly to the root cepstral domain. It is, therefore, possible to demonstrate the optimized modelling of a pole-zero model and show that its performance is superior to that of a FIR Wiener filter and LPC.

The optimized modelling of speech data is considered by a special form of the developed cost function. It is demonstrated that the modelling performance of the root cepstral method is superior to that of the real (magnitude) cepstrum and LPC.

A novel method of model order identification for use with time domain modelling methods based around z-plane root cepstral plots is also developed and discussed. It is demonstrated that the positions of a model or plant's poles and zeros may be determined by visual inspection of the resulting z-plane plot. However, performance in noise was poor to that of LPC, leading to difficulties when trying to determine the model's order.

Finally, an investigation into the poor phase modelling performance of the algorithm when modelling signals comprised of multiple excitations is presented. It is demonstrated that all DFT/FFT based analysis techniques are fundamentally flawed due to discontinuities. As a consequence, a simple pre-filtering algorithm is presented as a possible solution.
Acknowledgements

To my supervisor, Dr. Ted Chilton, CVSSP, for his patience and in depth understanding of signal processing theory. Also not forgetting all the coffee and alcohol he has bought me over the years.

I would also like to thank various members of CVSSP and my family for their encouragement and financial support over this period. A special thank you to Professor Paul Micallef, University of Malta and all of the other members of the Department of Communications and Computer Engineering for the fruitful discussions and warm Mediterranean hospitality during my student exchange visit there.

Finally, I am truly indebted to Dr. Clive Andrews, Dr. Dave Blood and Dr. Sue Blood, for finding the time in their busy schedules to proof read this thesis.
## Contents

0.1 Acronyms and Definitions ............................................. ix  
0.2 Symbols and Definitions .................................................. xi

1 Introduction  
1.1 Adaptive Filters ............................................. 4  
1.2 The System Identification Problem ................................. 4  
1.3 Speech Coding ................................................... 6  
1.3.1 Modelling the System ........................................ 7  
1.4 Noise Cancellation .................................................. 8  
1.5 Roadmap of the Thesis .............................................. 10  
1.6 Summary and Conclusions .......................................... 12

2 Time Domain Modelling Techniques  
2.1 Introduction .......................................................... 14  
2.2 The Padé Approximation .......................................... 15  
2.3 Prony’s Method ..................................................... 17  
2.4 Other Deterministic Pole-Zero Modelling Techniques ...... 20  
2.5 The Yule-Walker Equations ....................................... 21  
2.5.1 The Modified Yule-Walker Equations (MYWE) ........ 23  
2.5.2 The Least Squares Modified Yule-Walker Equations (LSMYWE) 26  
2.6 The Wiener-Hopf Equations ...................................... 27  
2.7 Linear Prediction ................................................... 29  
2.7.1 The Autoregressive Model ................................... 29  
2.7.2 The Moving Average Model .............................. 32  
2.8 The LMS Algorithm ............................................... 32
## Contents

2.8.1 The Variable Step Size LMS Algorithm ....................................... 34
2.8.2 Variants of the LMS Algorithm ....................................................... 34
2.9 The RLS Algorithm ...................................................................................... 35
2.10 The Kalman Filter ......................................................................................... 35
2.11 Model Order Identification and Validation ................................................. 36
   2.11.1 Final Prediction Error (FPE) and The Akaike Information Crit-
   eric (AIC) ........................................................................................................ 36
   2.11.2 Minimum Description Length (MDL) ........................................... 37
   2.11.3 Criterion Autoregressive Transfer (CAT) ........................................ 38
2.12 Summary and Conclusions ............................................................................. 39

3 Cepstral Deconvolution 41
   3.1 Introduction .................................................................................................. 41
   3.2 The Real Cepstrum ............................................................................. 42
   3.3 The Complex Cepstrum .................................................................... 43
      3.3.1 Modelling Rational Systems ............................................................. 45
      3.3.2 Noise and Error Reduction Techniques .......................................... 46
   3.4 The Root Cepstrum ...................................................................................... 47
      3.4.1 Special Properties of the Root Cepstrum ....................................... 48
      3.4.2 Pole-zero Modelling and its Implications ....................................... 49
      3.4.3 Noise Immunity .................................................................................. 50
   3.5 Other Non-Linear Functions .......................................................................... 51
   3.6 The Mel Cepstrum ............................................................................. 54
   3.7 The Differential Cepstrum ........................................ 54
      3.7.1 Group Delay ..................................................................................... 55
      3.7.2 The Differential log Cepstrum and its Relationship to the Plant's
          Impulse Response ................................................................... 56
   3.8 The Bicepstrum ............................................................................. 57
   3.9 The LSMYWE-Cepstrum Recursion ............................................... 57
   3.10 Summary and Conclusions ............................................................................. 59
## 4 Theoretical Development

4.1 Introduction ...................................................................................................... 61

4.2 Root Cepstral Cost Function ........................................................................... 63

4.2.1 Properties of the Performance Function ................................................. 65

4.2.2 Real World Considerations ....................................................................... 66

4.3 ROD Algorithm Optimisation ......................................................................... 67

4.3.1 ROD Algorithm: Part A .......................................................................... 67

4.3.2 RCD Algorithm: Part B ........................................................................... 70

4.4 z-plane Root Cepstral plots ........................................................................... 71

4.4.1 The all-pole model .................................................................................... 72

4.4.2 The pole-zero model ................................................................................. 77

4.5 Summary and Conclusions ............................................................................ 80

## 5 Simulation Results

5.1 Introduction ...................................................................................................... 81

5.2 RCD Algorithm Performance ......................................................................... 81

5.2.1 The all-pole model .................................................................................... 81

5.2.2 The all-zero model ................................................................................... 91

5.2.3 The pole-zero model ................................................................................. 94

5.2.4 Compression .............................................................................................. 98

5.2.5 The RCD and its relationship to the Padé Approximation ..................... 98

5.3 RCD performance when modelling magnitude and phase independently 100

5.4 z-plane Cepstral plots .................................................................................... 104

5.4.1 Performance in Noise .............................................................................. 109

5.5 Modelling maximum phase and mixed phase systems ............................... 112

5.6 Matlab's phase unwrapping algorithm and the Differential Cepstrum ....... 114

5.7 Modelling speech .......................................................................................... 116

5.8 Summary and Conclusions .......................................................................... 121

5.8.1 Further Discussions ................................................................................. 123
## Contents

6 Phase Analysis 124
- 6.1 Introduction ................................................. 124
- 6.2 The DTFT Vs. The DFT ...................................... 125
- 6.3 Phase Analysis for signals comprised of multiple excitations ............... 126
- 6.4 Hartley Phase ..................................................... 130
- 6.5 A Pre-Filtering Algorithm .................................... 131
  - 6.5.1 Modelling Performance ................................. 132
  - 6.5.2 Modelling Speech .......................................... 133
- 6.6 Summary and Conclusions ................................... 134

7 Conclusions and Further Work 135
- 7.1 Further Work ..................................................... 138

A Algorithm Implementation 139
- A.1 RCD ................................................................. 139
  - A.1.1 Analysis ...................................................... 139
  - A.1.2 Lifter Function ............................................. 140
  - A.1.3 Synthesis ..................................................... 140
  - A.1.4 Cost function of Eqn. (4.6) ............................ 140
  - A.1.5 Cost Function of Eqn. (5.20) ......................... 141
- A.2 z-plane Root Cepstral plots ................................. 141
- A.3 Pre-filtering Algorithm ....................................... 142
0.1 Acronyms and Definitions

A/D Analogue-to-discrete converter. I.e., converting a signal from continuous time to discrete time.


AIC Akaike Information Criterion.

AR Autoregressive. A stochastic all-pole model.

ARMA Autoregressive moving average. A stochastic pole-zero model.

CAT Criterion autoregressive transfer.

CELP Code Excited Linear Prediction.

Cepstrum An anagram of the word "spectrum". A frequency domain deconvolution technique.

CF Compression factor.

D/A Discrete-to-analogue converter. I.e., converting a signal from discrete time to continuous time.

DFT Discrete Fourier transform.

DSP Digital signal processing/processor.

DTFT Discrete time Fourier transform.

FFT Fast Fourier transform.

FPE Final prediction error.

FIR Finite impulse response. Essentially an all-zero digital filter.

GSM Global system for mobile communications.

IDFT Inverse discrete Fourier transform.

IDTFT Inverse discrete time Fourier transform.
ITC Information Theoretic Criterion.

LMS Least mean square. The famous LMS algorithm.

LPC Linear predictive coding.

LSMYWE Least squares modified Yule-Walker equations.

Lifter An anagram of the word “filter” used in cepstral analysis. Generally a rectangular window function applied to the cepstral data.

MA Moving average. A stochastic all-zero model.

MDL Minimum description length.

MELP Mixed excitation linear prediction.

MMSE Minimum mean squared error.

MSE Mean square error.

MYWE Modified Yule-Walker equations.

RC Real cepstrum (magnitude cepstrum).

RCD Root cepstral deconvolution.

RLS Recursive least squares. The RLS algorithm.

SNR Signal-to-noise ratio.

SPL Sound pressure level.

VS Variable step. The variable step size variant of the LMS algorithm.
0.2 Symbols and Definitions

- $c^*(n)$: Complex conjugate of $e(n)$.
- $X^H$: Hermitian transpose of matrix $X$.
- $u(n) * x(n)$: Convolution sum. I.e., $u(n)$ convolved with $x(n)$.
- $\nabla$: Gradient vector, used in the steepest descent and LMS algorithms.
- $\varepsilon$: Statistical expectation.
- $\Sigma$: Summation.

- $c(\eta)$: Cepstral data (data transformed into the pseudo time domain).
- $l(\eta)$: Cepstral lifter function. Usually a rectangular window function used to window $c(\eta)$.

- $e(n)$: Estimation error. Defined as the difference between plant output and model output.
- $u(n)$: Kronecker delta function system/plant input.
- $v(n)$: Bandlimited Gaussian white noise input.
- $x(n)$: Plant output, usually an impulse response.
- $\hat{x}(n)$: Model output, usually an impulse response.
- $\hat{x}(n|n - \Delta)$: Estimated value of $x(n)$ (model output) at time $n - \Delta$.

- $I$: Identity matrix.
- $r_{xu}$: Cross correlation vector.
- $R$: Autocorrelation matrix.

- $p$: Number (order) of poles.
- $q$: Number (order) of zeros.
- $a(k)$: Denominator polynomial coefficients of a linear model.
- $b(k)/w(k)$: Numerator polynomial coefficients of a linear model.

- $H(z)$: Model transfer function.
- $P(z)$: Plant transfer function.
\( E(w) \)  Short time spectrum of the error.
\( \mathcal{F} \)  DTFT.
\( \mathcal{F}^{-1} \)  IDTFT.
\( G \)  Gain term.
\( L(w) \)  Short time spectra of the cepstral lifter function.
\( M(w) \)  Short time spectra of the model.
\( j \)  normalized RCD cost function.
\( U(w) \)  Short time spectra of the plant input.
\( X(w) \)  Short time spectra of the plant output.
\( \dot{X}(w) \)  Short time spectra of the model output. I.e., an estimate of the plant's output.
\( Z \)  \(z\)-transform.
Chapter 1

Introduction

Since the dawn of time, civilizations all over the world have devised efficient methods for solving problems. The Babylonians, for example used a set of rules or an algorithm for deciding points of law, whereas ancient Chinese and Japanese cultures used algorithms for categorizing martial art movements. In the 9th century, the mathematician, al-Khwarizmi\(^1\) undertook pioneering work in algebra, which was popularized in his book, "al-Mukhtasar fi Hisab al-Jabr wa l-Muqabala" [13] and altered society’s perspective of analyzing problems, be they a simple domestic chore or a complex mathematical concept.

For the squares and roots equal to a number, it is as saying: a square and ten of its roots is equal to thirty-nine dirhams. The solution is to halve roots, equal to five in this problem, then, multiplying the root by itself then this will be twenty-five. Then add it to thirty-nine and this will be sixty-four. Then take the square root, which will be eight and subtract from it half the root, which is five. The remainder is three and that is the root you are seeking and the square is nine.

Figure 1.1: An excerpt from al-Mukhtasar fi Hisab al-Jabr wa l-Muqabala for the solution to \(x^2 + 10x = 39\).

In the 21st Century, the need for fast and efficient data compression algorithms has never been greater, as society places greater demands on manufacturers for faster data processing speeds and access to more and more information. Indeed, a huge demand

\(^1\)The word “algorithm” is derived from al-Khwarizmi’s name.
for high fidelity communication systems is difficult to meet, since bandwidth is severely restricted over a satellite channel or the internet. Over the years, a multitude of papers have appeared proposing newer, more efficient data compression schemes for digital audio and the like [19, 45, 85]. However, as processing power increases with the advent of better DSP (digital signal processing) technology, modelling schemes that were once difficult to implement have now been seen to fruition and have ultimately increased the scope of tools used for signal modelling.

Fortunately several tools are available for signal modelling and ultimately data compression in either the time domain or the frequency domain. The reader may be surprised by the fact that deterministic time domain modelling techniques date as far back as the 18th Century [61, 68]. Methods such as Padé, Prony, Shanks' and Steiglitz-McBride may therefore be used to model an impulse response of any linear discrete-time system. However, the Yule-Walker equations and the Wiener-Hopf equations may be used to model signals and systems with stochastic or random properties and have proved to be invaluable in such fields as, speech coding and noise cancellation.

At this stage it is informative to introduce some additional terminology necessary for modelling stochastic signals. Stochastic processes are generated by filtering white noise with a causal linear time-invariant filter that has a rational transfer function, comprised of \( p \) poles and \( q \) zeros, pole-zero models of this form are referred to as ARMA (autoregressive moving average) processes of order \((p,q)\). There are two special cases of the ARMA process: firstly, when \( q=0 \), the process is essentially an all-pole filter filtered by white noise and is therefore referred to as an AR (autoregressive) process. Secondly, when \( p=0 \), it is an all-zero filter and therefore referred to as a MA (moving average) process.

The cepstrum is a frequency domain modelling technique that unlike the aforementioned time domain modelling techniques can model the properties of a non-linear system. Since by utilizing the Fourier transform, the cepstrum transforms data from say, the discrete-time domain to a new pseudo discrete-time domain by the use of a non-linear function in order to separate the plant’s impulse response from the excitation. The “cepstral deconvolution” operation may be summarized as follows.

1. Obtain the Spectrum of the data, by applying the DTFT (discrete time Fourier
2. Transform the data into a warped time domain by applying a non-linear operator to the spectrum of the data.

3. Take the IDTFT (inverse discrete time Fourier transform) of the resulting sequence.

The data in this new "pseudo time domain" is comprised of a mixture of pole-zero coefficients (for the mixed model case) and is represented as a single polynomial or an all-zero model. Traditionally, a logarithmic function has been employed as the "optimal" non-linear operator. However, over the years further research proposed new non-linear functions and therefore demonstrated that logarithmic was not suitable for all types of signals.

There are two classes of cepstrum: Magnitude and complex. The complex cepstrum is able to analyze both magnitude and phase and is therefore a reversible process, but is limited to the modelling of deterministic signals. Since phase is retained, the technique has signal alignment problems, since not all of the cepstral data may captured by the windowing operation, necessary for part of the compression and data extraction procedure. The windowing function is usually a rectangular function and is referred to as a lifter, as defined by Bogert et al. [6].

The magnitude cepstrum or "real cepstrum" on the other hand, may be used for modelling both stochastic and deterministic signals, and has found particular use in speech and seismic analysis. However, the method is limited since it does not retain phase and is therefore non-reversible. Notice that in both cases, the cepstrum is able to analyze both minimum and maximum phase systems without the worry of model stability. Unfortunately, a shortcoming of all cepstral methods is the fact that they have no performance function and therefore analysis has always been empirical. The cepstrum is therefore a sub-optimal method.

Another modelling technique of particular interest, couples the cepstrum together with a subset of the Yule-Walker Equations in order to model the MA coefficients of an ARMA process more accurately. The method is particularly interesting since it combines a frequency domain technique (the real cepstrum) with a time domain technique.
Subsequent improvements led to the development of the evolutionary cepstrum, since cepstral estimates were based upon the evolutionary periodogram [39] and therefore used to model non-stationary signals.

1.1 Adaptive Filters

Adaptive filters are required when it is necessary for a filter's characteristics to be variable, adapting to changing signal characteristics. Such a filter can be classed as self designing in that it relies on a recursive mathematical algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the desired signal and noise is not known. Thus, due to their flexibility, adaptive filters have been used in many real world applications such as biomedical signal enhancement, system identification and telephone echo cancelling [92, 94].

1.2 The System Identification Problem

Consider the arrangement of figure 1.2, where $P(s)$ is an “analogue plant” transfer function (i.e., a physical system modelled by a laplace transfer function), $M(z)$ is a discrete model and “A/D” are analogue to discrete converters. The error signal of this arrangement may be expressed as, $e(n) = x(n) - \hat{x}(n)$, where $\hat{x}$ denotes estimate.

![Figure 1.2: System Identification of an analogue plant using a discrete model. Where “A/D” represents analogue-to-discrete converters and $\hat{x}(n)$ denotes an estimate of $x(n)$ at time $n$.](image-url)
1.2. The System Identification Problem

The requirement is to model the characteristics of the unknown plant by a suitable mathematical model using its input/output data records such that the error, $e(n)$, is minimized. This model may then be used for analysis purposes at a later stage. In order to proceed, several questions must be answered. Firstly, what type of model all-pole, all-zero or pole-zero is to be used for the categorization of the plant? What modelling method and performance criteria are to be used? What is the order of the model? These indeed are difficult questions and to-date there is no definitive answer, since each modelling technique has its own advantages and disadvantages, depending upon the problem. A few model order identification techniques have been devised over the years, but generally perform badly in the presence of noise and for short data sequences. Therefore, these techniques can only be used for giving an approximate indication of the true order of a plant or process. Finally, due to the inevitability of noise corrupting the measurements of $u(t)$ and $z(t)$, most modelling schemes will not be able to model the plant exactly.

Typically all-zero models are favoured for most system identification schemes, in order to circumvent any model stability problems that may arise. However, several speech coding techniques as discussed in the next section, make use of an AR model due to its elegance and simplicity for real time implementation. Generally speaking it is difficult to model a system where little or no a priori knowledge is available. The useful pieces of a priori information may include details about the number of poles and zeros of the plant or details about the statistics of the input, such as the variance. Note that all time domain modelling techniques discussed herein are based upon a linear model and therefore modelling a plant with non-linear characteristics is quite difficult. The cepstrum provides a possible solution to the non-linear case, since there is no explicit declaration of the number of poles and zeros and no linear model. Therefore, the cepstrum provides an interesting non-parametric alternative to the system identification modelling problem.
1.3 Speech Coding

The Yule-Walker Equations originally proposed by Yule [96] and Walker [83] around the 1930s for sunspot analysis are a fundamental set of parametric modelling tools for system identification and form the basis of LPC (linear predictive coding), required for low bit rate communication. The basic idea behind LPC is that a speech sample can be approximated as a linear combination of past speech samples. Therefore, by minimizing the MSE between the actual speech samples and the predicted ones, an optimal set of coefficients or weights for an all-pole (synthesis) or all zero (analysis) digital filter can be determined. Although other modelling techniques, such as the cepstrum exist, LPC offers a computationally efficient and accurate method for estimating speech parameters. As a consequence, it has earned its place as an industry standard technique for speech coding.

Figure 1.3: Anatomy of the human speech production system

A simplified illustration of the human speech production system is shown in figure 1.3. The process of human speech synthesis may be summarized as follows [17]:

Air pushed out from the lungs travels into the trachea, then up into the glottis, where it is periodically interrupted by the movement of the vocal chords. The tension of the vocal chords is adjusted by the larynx so that the chords vibrate in an oscillatory fashion, resulting in the production of voiced speech. During unvoiced speech, constrictions within the vocal tract (oral cavities - mouth, throat, etc.) force air from the lungs,
1.3. Speech Coding

through the constriction to produce turbulence. An example is the /s/ sound (fricative) in the word “six”. There are many other anatomical components that contribute to the production of speech, such as the velum, teeth, lips and tongue. These are referred to as articulators and move to different positions in order to produce various speech sounds.

1.3.1 Modelling the System

A simplified block diagram of the speech production system is shown in figure 1.4. Notice that the block diagram model combines effect of the glottis and lips\(^2\) and condenses the discrete speech production model [44], containing both poles and zeros into a single all-pole model of sufficiently high order. Using the model of figure 1.4 linear predictive coding (LPC) may be used to synthesize a speech waveform, \(s(n)\) [3, 4]. The use of an all pole model of the human vocal tract excited by a pitch pulse, to represent voiced speech and white noise to represent unvoiced speech, was popularized in the 1970s, with Texas Instruments “Speak and Spell” learning aid [22].

Subsequent achievements in this field led to the development of the LPC-10 algorithm.

\(^2\)The glottis and the lips manifest themselves as two poles and one zero close to the unit circle in the discrete model, leading to the zero cancelling out one of the poles.
This algorithm was officially adopted by the U.S government for both civilian and military communications [79] and was replaced in March 1996 by MELP (mixed excitation linear prediction) [54]. MELP coders are based on the traditional LPC model, but include additional features to improve their performance [77, 82]. Finally, CELP (code excited linear prediction) [12] is another successful coding scheme that has replaced LPC-10 for low-bit rate, high fidelity communication systems. Performance comparisons of all three coding schemes and many others can be found in [42, 86].

1.4 Noise Cancellation

Separating the signals of interest from additive noise is a common problem in signal processing. A typical application is the enhancement of a pilot's speech in a fighter aircraft [67, 71], since sound pressure levels (SPL) of 90dB or greater, due to engine and wind noise manifest themselves across the entire speech spectrum, making the transmitted speech unintelligible.

A possible solution to this problem is shown in figure 1.5, where "D/A" represents a discrete to analogue converter, $M(z)$ is the discrete model (typically a FIR adaptive filter) and $N(s)$ is the analogue noise transfer function to be determined. It can be seen, that by adding a second "reference" microphone that is away from the speech, $s(t)$ and is therefore just monitoring the ambient noise field, $n(t)$, an estimate of the noise transfer function, $N(s)$ may be obtained. The speech may be enhanced by estimating the noise ($\hat{n}(n)$) that is transmitted when the pilot speaks into the primary microphone, and then subtracting this estimate from the transmitted speech. Analyzing the block diagram, it can be seen that the enhanced speech is expressed as $\hat{s}(n) = s(n) + \hat{n}(n) - \hat{n}(n)$.

Harrison et al. [29] suggested that the primary and reference microphones should be as close together as possible for the following reasons: Firstly, an adaptive filter with a shorter length reduces the effects of reverberation and requires less computation. Secondly, in a multinoise environment, the noise samples between the primary and reference microphones start to become uncorrelated with each other as the microphones are moved further apart. However, care must be taken if the microphone are too close together, since the system may cancel the speech as well as the noise! A typical fighter
1.4. Noise Cancellation

Figure 1.5: System identification for the purpose of noise cancellation, where \( N(s) \) is an analog noise transfer function that must be determined by the adaptive filter, \( M(z) \).

The pilot's helmet consists of two microphones either side of the oxygen face mask. These serve as the primary and reference microphones respectively. Therefore, the primary microphone provides the noisy speech that needs to be enhanced and the reference microphone, should ideally contain the ambient noise field and no speech. However, due to the proximity of the two microphones, the face mask can only act as a limited acoustic barrier, providing about 10dB of attenuation and an approximate separation distance of 3cm. As consequence, the reference microphone generally contains some components of speech, leading to distortion of the enhanced speech.

Further work also undertaken by Harrison et al. [30], devised a method of constraining the adaptive filtering algorithm such that the weights could only be altered during the silent intervals, i.e., when no speech was present and therefore demonstrated that a 11dB signal-to-noise ratio (SNR) improvement could be achieved.
1.5 Roadmap of the Thesis

Due to the diversity of the topics covered, this thesis is split up into several sections, as summarized below.

A comprehensive literature review of both deterministic and stochastic time domain modelling techniques is presented and discussed in chapter 2. Several interesting comparisons are made between the various types of stochastic and deterministic modelling methods, such as the similarity between the Yule-Walker equations and the Wiener-Hopf equations for one step linear prediction, and the unique relationship between the all-pole Prony equations and the all-pole Yule-Walker equations.

The latter part of the chapter is devoted to issue of model order identification/validation, which is discussed in some depth, including the limitations of the standard identification techniques.

A comprehensive literature review of cepstral techniques is presented in chapter 3. The chapter begins with an overview of the logarithmic cepstrum, leading to the definition of the real cepstrum (RC) and complex cepstrum (CC). The discussion then continues to the root cepstrum and root cepstral deconvolution (RCD), where other non-linear operators are introduced. Some fundamental cepstral theory is presented that forms an important part of the theoretical development, as presented in chapter 4.

The latter part of chapter 3 provides an overview of the Mel cepstrum, the differential cepstrum and the Bi-cepstrum. Although these techniques do not significantly contribute to the theoretical development of the thesis, they are presented out of interest.

Chapter 4 forms the core of the author's novel theoretical development to the thesis. A root cepstral cost function is introduced together with a suitable optimization algorithm, such that root cepstral modelling performance could be measured. Finally, a rather novel method of model order identification for time domain modelling methods based around z-plane root cepstral plots is presented. Subsequent developments are discussed in chapter 5.

Using the theory developed and reviewed in chapters 2, 3 and specifically chapter 4, root cepstral modelling performance (using the developed cost function) for all-pole, all-zero
1.5. Roadmap of the Thesis

and pole-zero models is assessed in the presence of various levels of noise, and compared to some time domain methods. The discussion then continues to the derivation of several important relationships between the RCD and two time domain techniques. The concept of modelling magnitude and phase independently to one another is then considered. Finally, a comprehensive analysis and discussion of z-plane root cepstral plots and other developments are presented at the end, including a special cost function for modelling speech.

Finally, chapter 6 investigates the poor phase modelling performance of the RCD algorithm when modelling signals comprised of multiple excitations. It is demonstrated that all DFT/FFT based analysis techniques, including the Hartley spectrum are fundamentally flawed due to discontinuities in the resulting real and imaginary components of the DFT. As a consequence, a simple pre-filtering algorithm is presented as a possible solution.

The key novel contributions of the thesis are summarized as follows:

- Root cepstral cost function, modelling both magnitude and phase, Eqn. (4.6).
- Optimized modelling of speech data using a special form of developed cost function, as shown in Eqn. (5.20).
- The formulation of two special relationships between the RCD and the FIR Wiener filter and the RCD and the Padé approximation, as presented and discussed in section 5.2.5.
- Modelling Cepstral magnitude and phase independently to one another (i.e., assigning different values of $\gamma$ to the magnitude and phase spectra), yielding superior modelling performance at the expense of an increased number of cepstral coefficients, as presented and discussed in section 5.3.
- Pole-zero model order identification using z-plane cepstral plots for use with time domain modelling methods, as presented and discussed in sections 4.4 and 5.4.
- An investigation into the poor phase modelling performance of DFT/FFT based modelling methods when modelling signals comprised of multiple excitations, as presented and discussed in chapter 6.
1.6 Summary and Conclusions

Improvements in signal modelling techniques have been encouraged over the years by society's demand for more efficient ways of accessing information, whether this is for the clarity/enhancement of speech in a mobile phone or an industrial requirement for modelling a physical system. As a consequence, several modelling/compression techniques in both the time domain and the frequency domain have been developed over the years as possible solutions to these problems. Time domain techniques such as Padé, Prony, Shanks' and Steiglitz-McBride (as discussed in chapter 2) may be applied to the modelling of deterministic signals, whereas the Yule-Walker and the Wiener-Hopf equations are stochastic modelling techniques. Nevertheless, in both cases a linear model is optimized by some type of performance criterion.

A popular frequency domain modelling technique is the cepstrum, whereby data is transformed into a pseudo time domain in order to reduce the cepstral model order. The cepstrum, like the aforementioned time-domain techniques has its variants for modelling deterministic and stochastic signals, but has the virtue of being able to model non-linear systems. Due to the utilization of the Fourier transform, the technique is non-parametric and therefore does not suffer from the stability problems and constraints imposed by a linear model. Perhaps the greatest disadvantage of the cepstrum is that there is no performance or cost function and therefore unlike the time domain techniques it cannot be considered optimal.

The system identification problem is where the characteristics of an unknown plant or model are to be modelled using the input/output data records. This may be undertaken in either the frequency domain or the time domain, using the techniques as discussed above. However, several fundamental questions must be answered before the analysis or modelling can proceed. Firstly, what type of model is to be used (all-pole, all-zero or pole-zero)? What type of signals (deterministic or stochastic) are to be modelled? What domain (time or frequency) is the analysis to be undertaken? What is the order of the model? There is no definitive answer to any of these questions, since each modelling technique has its own advantages and disadvantages, depending upon the situation.
Towards the latter part of the chapter, two system identification examples are given. The speech coding example is based around linear predictive coding (LPC). Notice that this technique uses prediction for the system identification and condenses the speech production model, containing both poles and zeros into a single all-pole model of sufficiently high order. An overview of standard coding schemes is also given.

The noise cancellation example describes how speech corrupted by noise may be enhanced by the use of an adaptive filter. This may be achieved by modelling the corruptive analogue noise transfer function by its discrete equivalent. Therefore, the speech may be enhanced by simply exciting the discrete model with estimates of the ambient noise field (obtained from a reference microphone away from the speech) and subtracting the model’s output from the noisy speech, resulting in a moderate improvement in the SNR.

Finally, a roadmap of thesis is presented in section 1.5, and all novel contributions identified and summarized.
Chapter 2

Time Domain Modelling Techniques

2.1 Introduction

Expressing the properties of a "black box" or unknown plant in terms of a suitable linear mathematical model, has long been a challenging and difficult task for the control engineer. Although many time domain pole-zero modelling techniques exist, as described later on in the chapter, all modelling methods require information about the exact number of poles and zeros of the system (system order) in order to successfully model the system. Such data may not be available a priori and therefore presents many difficulties for the designer who must design the system empirically.

As illustrated in figure 2.1, at time \( n \), the model produces an output or estimate, \( \hat{x}(n) \) of the desired response, \( x(n) \). This estimate, is accompanied by an estimation error, 

\[
\hat{e}(n) = x(n) - \hat{x}(n)
\]

Figure 2.1: System identification based on pole-zero modelling.
2.2 The Padé Approximation

\[ e(n) = x(n) - \hat{x}(n) = x(n) - h(n) * u(n) \]  \hspace{1cm} (2.1)

\( u(n) \) is assumed to be a Kronecker delta function (deterministic signals) or band limited zero mean Gaussian white noise (stochastic signals) and \( e(n) \) is defined as the difference between the desired response and model's estimated output. Clearly, the smaller the estimation error, the closer the estimated output is to the desired output on a sample by sample basis. For simplicity, many algorithms minimize \( e(n) \) by computing the squared error or in some cases the MSE (mean square error) which result in a set of linear equations that may be solved quickly and efficiently [17, 18].

This chapter is broken up into two sections reviewing both deterministic and stochastic or random modelling algorithms. Deterministic modelling algorithms, such as the Padé approximation and Prony's method assume that the characteristics of the signal, say \( x(n) \) are known for all values of \( n \). However, in some cases, where \( x(n) \) is random, such as unvoiced speech, stochastic models are required, since \( x(n) \) may only be described probabilistically.

2.2 The Padé Approximation

The Padé approximation [31, 51, 61] attempts to model the characteristics of a discrete time impulse response signal, \( x(n) \), containing \( p \) poles and \( q \) zeros over the time interval \( 0 \leq n \leq p + q \) without error. In order to develop a set of Padé linear equations, consider the following transfer function:

\[ H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \]  \hspace{1cm} (2.2)

which may be expressed as:

\[ H(z)A(z) = B(z) \]  \hspace{1cm} (2.3)

Taking inverse \( z \)-transforms, yields the following difference equation

\[ h(n) + \sum_{k=1}^{p} a(k) h(n-k) = b(n) \]  \hspace{1cm} (2.4)
Chapter 2. Time Domain Modelling Techniques

where \( h(n) = 0 \) for \( n < 0 \) and \( b(n) = 0 \) for \( n < 0 \) and \( n > q \). Also, in order to be consistent with the output nomenclature, let \( x(n) = h(n) \).

Summarizing these results leads to a set of \( p+q+1 \) linear equations:

\[
x(n) + \sum_{k=1}^{p} a(k)x(n-k) = \begin{cases} b(n) & n = 0,1,...,q \\ 0 & n = q+1,...,q+p \end{cases}
\]  

(2.5)

More conveniently, this may be expressed in matrix form.

\[
\begin{bmatrix}
  x(0) & 0 & \cdots & 0 \\
x(1) & x(0) & \cdots & 0 \\
x(2) & x(1) & x(0) & 0 \\
\vdots & \vdots & \ddots & \vdots \\
x(q) & x(q-1) & \cdots & x(q-p) \\
x(q+1) & x(q) & \cdots & x(q-p+1) \\
\vdots & \vdots & \ddots & \vdots \\
x(q+p) & x(p+q-1) & \cdots & x(q) 
\end{bmatrix}
\begin{bmatrix}
a(1) \\
a(2) \\
\vdots \\
a(p) \\
\end{bmatrix}
= \begin{bmatrix}
b(0) \\
b(1) \\
\vdots \\
b(q) \\
\end{bmatrix}
\]  

(2.6)

In order to be able to extract the numerator and denominator coefficients from Eqn. (2.6) corresponding to the poles and zeros of the model, a solution for the numerator coefficients, \( a(k) \) is firstly obtained by partitioning Eqn. (2.6) and solving the lower equations.

\[
\begin{bmatrix}
x(q) & x(q-1) & \cdots & x(q-p+1) \\
x(q+1) & x(q) & \cdots & x(q-p+2) \\
\vdots & \vdots & \ddots & \vdots \\
x(q+p-1) & x(p+q-2) & \cdots & x(q) 
\end{bmatrix}
\begin{bmatrix}
a(1) \\
a(2) \\
\vdots \\
a(p) \\
\end{bmatrix}
= \begin{bmatrix}
x(q+1) \\
x(q+2) \\
\vdots \\
x(q+p) \\
\end{bmatrix}
\]  

(2.7)

The numerator coefficients, \( b(n) \), may then be determined in a similar fashion by solving the upper set of equations.

\[
\begin{bmatrix}
x(0) & 0 & \cdots & 0 \\
x(1) & x(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
x(q) & x(q-1) & \cdots & x(q-p) 
\end{bmatrix}
\begin{bmatrix}
a(1) \\
a(2) \\
\vdots \\
a(p) \\
\end{bmatrix}
= \begin{bmatrix}
b(0) \\
b(1) \\
\vdots \\
b(q) \\
\end{bmatrix}
\]  

(2.8)
As an example of modelling under the Padé approximation, consider a signal, \(x(n)\) modelled by a pole-zero model, where \(p = 2\) and \(q = 1\). As illustrated in figure 2.2 it can be seen that in the interval \(n = 0..p + q\), \(x(n)\) has been modelled without error. However, for \(n > 3\), \((p + q = 3)\) the model has not produced an accurate representation of \(x(n)\) and has in fact produced a best fit type approximation. A word of caution should be exercised when selecting the model order, since the Padé model is only exact over the first \(p+q+1\) values and is therefore not necessarily stable.

2.3 Prony’s Method

The Padé approximation is constrained in that it only uses the first \(p+q+1\) data values of the input signal, say \(x(n)\) to compute the model parameters. By examining Eqn. (2.6), in the interval \(0 \leq n \leq p + q\), it can be seen that the algorithm will model the signal without error. However, since data outside the interval \(n = 0..p + q\) is not considered in the analysis, there is no guarantee on the model’s accuracy for \(n > p + q\). Prony’s method [31] provides a solution to the problem, since \(x(n)\) is no longer modelled exactly in the interval \(0 \leq n \leq p + q\) but is better approximated over all values of \(n\).
Applying a similar concept to that of the Padé approximation, an expression for the error, \( E(z) \) may be obtained.

\[
E(z) = A(z)X(z) - B(z) \tag{2.9}
\]

Eqn. (2.9) may be expressed in the time domain, as depicted in figure 2.3,

\[
e(n) = a(n) * x(n) - b(n) \tag{2.10}
\]

Since \( b(n) = 0 \) for \( n > q \), the error may be expressed as the following difference equation

\[
e(n) = \begin{cases} 
  x(n) + \sum_{k=1}^{p} a(k)x(n - k) - b(n) & n = 0, 1, ..., q \\
  x(n) + \sum_{k=1}^{p} a(k)x(n - k) & n > q 
\end{cases} \tag{2.11}
\]

Rather than setting \( e(n) = 0 \) for \( n = 0, 1, ..., p + q \) as for the Padé approximation, Prony's method begins by finding the denominator polynomial coefficients that minimize the deterministic squared error.

\[
\sum_{n=q+1}^{\infty} |e(n)|^2 = \sum_{n=q+1}^{\infty} \left| x(n) + \sum_{k=1}^{p} a(k)x(n - k) \right|^2 \tag{2.12}
\]

The coefficients that minimize the squared error may be obtained by setting the partial derivatives of Eqn. (2.12) with respect to \( a(k) \) equal to zero [31], where \((\cdot)^*\) denotes conjugation.

\[
\sum_{n=q+1}^{\infty} \frac{\delta[e(n)e^*(n)]}{\delta a(k)} = \sum_{n=q+1}^{\infty} e(n) \frac{\delta e^*(n)}{\delta a(k)} = 0 \quad k = 1, 2, ..., p \tag{2.13}
\]

Summarizing this result yields the following relationship, referred to as the principle of orthogonality.

\[
\sum_{n=q+1}^{\infty} e(n)x^*(n - k) = 0 \quad k = 1, 2, ..., p \tag{2.14}
\]
2.3. Prony's Method

Substituting Eqn. (2.11) into Eqn. (2.14).

\[
\sum_{n=q+1}^{\infty} \left[ x(n) + \sum_{i=1}^{p} a(i)x(n-i) \right] x^*(n-k) = 0 \quad (2.15)
\]

After some simplification, a set of linear equations in terms of a deterministic autocorrelation sequence is obtained.

\[
\sum_{i=1}^{p} a(i)r(k-l) = -r(k) \quad k = 1, 2, \ldots p \quad (2.16)
\]

where, \( r(k-l) = \sum_{n=q+1}^{\infty} x(n-l)x^*(n-k) \). These equations are referred to as the Prony normal equations. Realizing that \( r(k-l) \) is conjugate symmetric, that is, \( r(k-l) = r^*(l-k) \) the normal equations may be expressed in a more suitable matrix form.

\[
\begin{bmatrix}
  r(0) & r^*(1) & \ldots & r^*(p-1) \\
  r(1) & r(0) & \ldots & r^*(p-2) \\
  \vdots & \vdots & \ddots & \vdots \\
  r(p-1) & r(p-2) & \ldots & r(0)
\end{bmatrix}
\times
\begin{bmatrix}
a(1) \\
a(2) \\
\vdots \\
a(p)
\end{bmatrix}
=
\begin{bmatrix}
r(1) \\
r(2) \\
\vdots \\
r(p)
\end{bmatrix}
\quad (2.17)
\]

Alternatively,

\[
Ra = -r \quad (2.18)
\]

with the solution

\[
a = -R^{-1}r \quad (2.19)
\]

Notice that the coefficients of the denominator polynomial, \( a(i) \) may have also been determined by solving Eqn. (2.11) directly. See below.

\[
\begin{bmatrix}
x(q) & x(q-1) & \ldots & x(q-p+1) \\
x(q+1) & x(q) & \ldots & x(q-p+2) \\
x(q+2) & x(q+1) & \ldots & x(q-p+3) \\
\vdots & \vdots & \ddots & \vdots 
\end{bmatrix}
\times
\begin{bmatrix}
a(1) \\
a(2) \\
\vdots \\
a(p)
\end{bmatrix}
=
\begin{bmatrix}
x(q+1) \\
x(q+2) \\
\vdots \\
x(q+3)
\end{bmatrix}
\quad (2.20)
\]

\[
Xa = -x \quad (2.21)
\]

\[
X^HXA = -X^Hx \equiv Ra = -r \quad (2.22)
\]
Therefore, it can be seen that the solution obtained by solving Eqn. (2.18) or Eqn. (2.21) leads to exactly the same result. The zeros are determined in exactly the same way as the Padé approximation, by solving Eqn. (2.8).

Figure 2.4 illustrates the modelling performance of Prony's method Vs. the Padé approximation. As expected, for a Prony/Padé pole-zero model of $p = 2$ and $q = 1$, Prony's method has modelled $x(n)$ more accurately for all $n$ rather than the Padé model. Notice that unlike the Padé approximation, the modelling error of Prony's method is only equal to zero in the interval $n = 0..q$. Note, if the model's order satisfies or exceeds the plant's order, the model will match $x(n)$ exactly.

Figure 2.4: The Padé Approximation Vs. Prony's Method for signal modelling.

### 2.4 Other Deterministic Pole-Zero Modelling Techniques

There are at least another two deterministic pole-zero modelling methods that are essentially extensions of Prony’s method. The first, Shanks' method [31, 74], attempts a least squares minimization of the model error in order to accurately model the zeros. Recall that Prony's method when calculating the numerator coefficients, $b(n)$ forces
the error of Eqn. (2.11) to equal zero in the interval $n = 0..q$ and therefore the model is only exact over this interval, since no attempt is made to model the data for $n > q$.

Finally, the Steiglitz-McBride method [16, 75] is an iterative algorithm that minimizes the least squared error with respect to both the numerator and denominator coefficients, by solving a set of over determined linear equations. Generally speaking, the algorithm usually converges rapidly, but is dependent upon on how close the initial guess is to the optimum solution [31].

### 2.5 The Yule-Walker Equations

In the previous sections, a set of algorithms for modelling deterministic signals was presented. The assumption in each case, was that $x(n)$ was known for all $n$ or for values of $n$ over a finite interval. However, consider the case when $x(n)$ is a stochastic or random process and is therefore not completely known. Such signals may only be described probabilistically. Recall that Prony’s Method and the Padé approximation minimize the deterministic squared error of a signal. It is therefore invalid to assume that these techniques may be used to model stochastic processes, since $x(n)$ is only known probabilistically. To circumvent this problem, a discussion of a set of stochastic modelling algorithms based around the “Yule-Walker method” is presented.

Consider the arrangement of figure 2.5, where $v(n)$ is band-limited white gaussian noise and $H(z)$ is a causal linear time-invariant filter with $q$ zeros and $p$ poles excited by $v(n)$.

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b(k)z^{-k}}{1 + \sum_{k=1}^{p} a(k)z^{-k}}$$

(2.23)

The power spectrum for Eqn. (2.23), with $P_x(e^{j\omega}) = \sigma^2_x$ is given by

$$P_x(e^{j\omega}) = \sigma^2_x \left| \frac{B(e^{j\omega})}{A(e^{j\omega})} \right|^2$$

(2.24)

Processes that have a power spectrum of the form of Eqn. (2.24) are referred to as ARMA (autoregressive moving average) processes of order $(p, q)$, or simply ARMA$(p, q)$. As mentioned in the introduction, there are two special cases of the ARMA process:
when \( q = 0 \), the model is all-pole and therefore referred to as an AR (autoregressive) process and similarly when \( p = 0 \), the model is all-zero and said to be a MA (moving average) process. This terminology will be used extensively for the remainder of this chapter.

Consider an ARMA process given by the following difference equation.

\[
x(n) + \sum_{l=1}^{p} a(l)x(n-l) = \sum_{l=0}^{q} b(l)v(n-l)
\]  

(2.25)

Multiplying both sides by \( x^*(n-k) \) and taking the expected value, yields

\[
r(k) + \sum_{l=1}^{p} a(l)r(k-l) = \sum_{l=0}^{q} b(l)E[v(n-l)x^*(n-k)]
\]

(2.26)

where, \( E[v(n-l)x^*(n-k)] = r_{ux}(k-l) \) is the cross-correlation between the input and output data. These results may be summarized as follows [38].

\[
r(k) + \sum_{l=1}^{p} a(l)r(k-l) = \begin{cases} \sum_{l=k}^{q} b(l)r_{ux}(k-l) & \text{for } k = 0, 1, \ldots, q \\ 0 & \text{for } k > q \end{cases}
\]

(2.27)

Analyzing figure 2.5, it can be seen that for the error to equal zero, \( \hat{x}(n) = x(n) \). Therefore,

\[
x(n) = h(n) * v(n) = \sum_{m=-\infty}^{\infty} v(m)h(n-m)
\]

(2.28)

The cross-correlation may now be expressed as

\[
E[v(n-l)x^*(n-k)] = E\left[\sum_{m=-\infty}^{\infty} v(n-l)v^*(m)h^*(n-k-m)\right] \\
= \sum_{m=-\infty}^{\infty} E[v(n-l)v^*(m)]h^*(n-k-m) \\
= \sigma_v^2 h^*(1-k)
\]

(2.29)
2.5. The Yule-Walker Equations

\[ E[v(n-l)v^*(m)] = \sigma_v^2 \delta(n-l-m). \]

Substituting Eqn. (2.29) into Eqn. (2.26) and assuming that \( h(n) \) is causal yields,

\[ r(k) + \sum_{l=1}^{p} a(l) r(k-l) = \sigma_v^2 \sum_{l=0}^{q-k} b(l+k)h^*(l) \]

(2.30)

The Yule-Walker Equations for an ARMA process may therefore be explicitly defined as,

\[ r(k) + \sum_{l=1}^{p} a(l) r(k-l) = \begin{cases} \sigma_v^2 c(k) & k = 0, \ldots, q \\ 0 & k > q \end{cases} \]

(2.31)

where, \( c(k) = \sum_{l=0}^{q-k} b(l+k)h^*(l) \). Expressing this in a more convenient form,

\[
\begin{bmatrix}
  r(0) & r^*(1) & \cdots & r^*(p) \\
  r(1) & r(0) & \cdots & r^*(p-1) \\
  r(2) & r(1) & r(0) & 0 \\
  \vdots & \vdots & \vdots & \vdots \\
  r(q) & r(q-1) & \cdots & r^*(p-q) \\
  \vdots & \vdots & \vdots & \vdots \\
  r(q+1) & r(q) & \cdots & r(q-p+1) \\
  \vdots & \vdots & \vdots & \vdots \\
  r(q+p) & r(p+q-1) & \cdots & r(q)
\end{bmatrix}
\times
\begin{bmatrix}
  a(1) \\
  a(2) \\
  \vdots \\
  a(p)
\end{bmatrix}
= \sigma_v^2 \begin{bmatrix}
  c(0) \\
  c(1) \\
  \vdots \\
  c(q)
\end{bmatrix}
\]

(2.32)

The Yule-Walker equations provide a relationship between the filter coefficients and the autocorrelation sequence. However, it should be noted, that due to the term \( \sum_{l=0}^{q-k} b(l+k)h^*(l) \), the Yule-Walker equations are non-linear and therefore quite difficult to solve.

2.5.1 The Modified Yule-Walker Equations (MYWE)

In order to determine the auto-regressive coefficients of an ARMA process [24, 25], of order \( p+q \), it can be seen that the Yule-Walker equations of Eqn. (2.31) become similar to the Prony equations, as seen by solving the lower set of equations of Eqn. (2.32).

\[ r(k) + \sum_{l=1}^{k} a(l) r(k-l) = 0 \quad k > q \]

(2.33)
Notice how Eqn. (2.34) and the Padé equations (2.7) are similar. In the modified or extended Yule-Walker equations, the elements consist of a sequence of autocorrelation, $r(k)$ values, whereas the Padé equations consist of actual data values of the sequence $x(n)$. Also, the matrix is toeplitz [26], but not hermitian and may be solved by Trench’s algorithm [80]. In spite of these similarities, the modelling performance is considerably different, as seen in figure 2.6.

![Figure 2.6: The Padé Approximation Vs. Modified Yule-Walker Equation (MYWE) Method for signal modelling.](image)

Finally, figure 2.7 compares the modelling performance of deterministic modelling algorithms (the Padé approximation and Prony’s Method) on a stochastic signal (impulse response + noise), to the modelling performance of the modified Yule-Walker method. As expected, for an ARMA model when $p = 2$ and $q = 1$, the modified Yule-Walker method surpasses the performance of the deterministic modelling algorithms, based on the mean squared error over all data points.
Example

Due to the complexity of the Yule-Walker equations, consider the example of a real valued ARMA process where, \( p = 1, q = 1 \) \[38\]. Using Eqn. (2.31), a set of equations may be produced

\[
\begin{align*}
  r(0) &= -a(1)r(-1) + \sigma^2 \left[ b(1)h(1) + 1 \right] & k &= 0 \\
  r(1) &= -a(1)r(0) + \sigma^2 b(1) & k &= 1 \\
  r(2) &= -a(1)r(1) & k &= 2
\end{align*}
\]

Assuming that \( b(0) = 1 \) and \( h(0) = 1 \), \( h(n) \) may be expressed in terms of the process coefficients, since

\[
\begin{align*}
  h(n) &= -a(1)h(n-1) + \delta(n) + b(1)\delta(n-1) \\
  \Rightarrow h(1) &= -a(1) + b(1)
\end{align*}
\]

Therefore,

\[
\begin{align*}
  c(0) &= b(0)h(0) + b(1)h(1) \\
  \therefore c(0) &= 1 + b(1)^2 - a(1)b(1) \\
  c(1) &= b(1)h(0) \\
  \therefore c(1) &= b(1)
\end{align*}
\]
Combining these results and re-arranging Eqn. (2.31) yields,

\[
\begin{bmatrix}
1 & a(1) \\
a(1) & 1
\end{bmatrix}
\begin{bmatrix}
r(0) \\
r(1)
\end{bmatrix}
= \begin{bmatrix}
\sigma_b^2[1 + b^2(1) - a(1)b(1)] \\
\sigma_b^2b(1)
\end{bmatrix}
\]

where,

\[a(1) = \frac{-r(2)}{r(1)}\]

NB. In practice the AR and MA coefficients are determined by firstly fitting a large order AR model to the data. These coefficients may then be used as the “data” in order to determine the MA coefficients by Durbin’s method. See Kay [38] for a more detailed explanation.

2.5.2 The Least Squares Modified Yule-Walker Equations (LSMYWE)

Revisiting Eqn. (2.34) it can be seen that if the autocorrelation values are not exact, the modified Yule-Walker equations will no longer be satisfied, due to estimation errors. Cadzow [10, 11] proposed the addition of an error vector, to account for the errors introduced by the estimated autocorrelation values.

\[\hat{\varphi} = \hat{R}a + e\] (2.35)

For best results, Cadzow advocated the use of unbiased autocorrelation values. The least squares solution for an optimal set of AR coefficients may therefore be expressed as,

\[\hat{a} = -\left(\hat{R}^H\hat{R}\right)^{-1}\hat{R}^H\hat{\varphi}\] (2.36)

The performance of the LSMYWE is superior to that of the MYWE, since the method relies upon solving a set of over determined equations and therefore there will be more equations than unknowns [24]. Friedlander [23] demonstrated that there was an improvement (especially for narrowband processes) in estimation accuracy as the number of equations increased.

A subsequent achievement, proposed by Kay [37], developed and extended the use of the LSMYWE to spectral estimation of an ARMA process without the need to determine the MA coefficients. Since the LSMYWE sometimes led to an invalid negative spectral
2.6 The Wiener-Hopf Equations

Consider the arrangement of figure 2.8, built around a class of linear optimum discrete-time filter with input, \( u(n) \). The filter or MA model is constrained to be a \( q \)th order FIR (finite impulse response) and may therefore be characterized by its impulse response \( (w_0, w_1, \ldots, w_q) \) [32].

![Figure 2.8: System identification based on the FIR Wiener filter.](image)

The FIR filter or MA model output is defined as

\[
\hat{x}(n) = \sum_{k=0}^{q} w(k)u(n - k)
\]

(2.37)

and the estimation error, \( e(n) \), once again is defined as

\[
e(n) = x(n) - \hat{x}(n)
\]

(2.38)

The Wiener filter may be optimized by applying a gradient operator \( \nabla \), to the cost function, \( J \) of the MSE.

\[
\nabla J = \mathcal{E} \left[ \frac{\delta[e(n)e^*(n)]}{\delta w(k)} \right]
\]

(2.39)

Eqn. (2.39) therefore demonstrates the dependence of the cost function, \( J \) on the filter coefficients as a bowl shaped \( q + 2 \) dimensional space, as shown in figure 2.9. This space is known as the error-performance surface and has a unique minimum. Note that higher order performance criteria, such as the mean-quad error, may result in more than one minimum and therefore a more complicated mathematical solution.
To reach the "bottom of the bowl" or the Wiener solution, the cost function, $J$ must be reduced to its minimum value. At this point all elements of Eqn. (2.39) must be equal to zero, only then can the filter said to be optimum in the mean square error sense [48].

$$J = \frac{\delta[e(n)e^*(n)]}{\delta w(k)} = 0 \quad \text{where,} \quad \frac{\delta e^*(n)}{\delta w(k)} = -u^*(n - k) \quad (2.40)$$

After some algebraic manipulation, an expression for the principle of orthogonality is obtained. Notice the similarity between Eqn. (2.41) and the principle of orthogonality used in Prony’s method, Eqn. (2.14).

$$E[e(n)u^*(n-k)] = 0 \quad k = 0,1,...q \quad (2.41)$$

$$E\left[ (x(n) - \sum_{k=0}^{q} w_k u(n-k))u^*(n-l) \right] = 0 \quad l = 0,1,\cdots q$$

$$\sum_{k=0}^{q} w_k r(l-k) = r_{xu}(l) \quad l = 0,1,\cdots q$$

Alternatively,

$$Rw_{opt} = r_{xu} \quad (2.42)$$

where $w_{opt}$ is the optimal weight vector corresponding to the Wiener solution.
2.7 Linear Prediction

One of the most difficult problems in time series analysis is that of predicting a future value of a stationary discrete time stochastic process, given a set of past samples of the process. In linear prediction, the predicted value, say, \( \hat{x}(n) \), may be expressed as a linear combination of past samples \( x(n-1), x(n-2), \ldots x(n-M) \). Linear prediction, when constrained to an all-pole (autoregressive) model is used for speech coding, required for low bit rate communications [44] and voice recognition applications. However, it has been successfully applied to non-engineering applications such as financial forecasting.

The form of linear prediction considered in this section is said to be in the forward direction, see figure 2.10. This is referred to as forward linear prediction (FLP), since by considering the time series \( x(n), x(n-1), x(n-2), \ldots x(n-M) \) and \( \Delta = 1 \), using a set of previous input samples, \( x(n-1), x(n-2), \ldots x(u-M) \) it is possible to predict next sample, \( x(n) \) at time \( (n-1) \). Where \( \hat{x}(n|n-\Delta) \) is the estimated value of \( x(n) \) at time \( (n-\Delta) \).

![Figure 2.10: Forward Linear prediction.](image)

The estimation error, \( e(n) \) is given by

\[
e(n) = x(n) - \hat{x}(n|n-\Delta)
\]  

(2.43)

2.7.1 The Autoregressive Model

The all-pole (autoregressive) model has found a particular use in speech coding, and is therefore a fundamental component of LPC (linear predictive coding). The LPC concept condenses the discrete speech production model [44], containing both poles and zeros into a single all-pole model of sufficiently high order. The virtue of an all-pole model is thus its simplicity and ease of real time implementation.
Chapter 2. Time Domain Modelling Techniques

The model's output, may be expressed as

\[ \hat{x}(n|n - \Delta) = - \sum_{k=0}^{P} w_k x(n - k - \Delta) \]  \hspace{2cm} (2.44)

Substituting Eqn. (2.44) into Eqn. (2.43), the estimation error may be explicitly defined as

\[ e(n) = x(n) + \sum_{k=0}^{P} w_k x(n - k - \Delta) \]  \hspace{2cm} (2.45)

The principle of orthogonality of Eqn. (2.41) now becomes

\[ \mathbb{E}[e(n)x^*(n - k - \Delta)] = 0 \hspace{1cm} k = 0, 1, \ldots, q \]  \hspace{2cm} (2.46)

\[ \mathbb{E}\left[(x(n) + \sum_{k=0}^{P} w_k x(n - k - \Delta))x^*(n - l - \Delta)\right] = 0 \hspace{1cm} l = 0, 1, \ldots, p \]  \hspace{2cm} (2.47)

Summarizing the result, yields

\[ r(l + \Delta) + \sum_{k=0}^{P} w_k r(l - k) = 0 \hspace{1cm} l = 0, 1, \ldots, p \]  \hspace{2cm} (2.48)

Due to the structure of the all-pole system, \( w(0) = 1 \). Therefore, setting \( k = l = 1, \ldots, p \), yields the following

\[ \begin{bmatrix} r(0) & r^*(1) & \ldots & r^*(p - 1) \\ r(1) & r(0) & \ldots & r^*(p - 2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p - 1) & r(p - 2) & \ldots & r(0) \end{bmatrix} \begin{bmatrix} w(1) \\ w(2) \\ \vdots \\ w(p) \end{bmatrix} = - \begin{bmatrix} r(1 + \Delta) \\ r(2 + \Delta) \\ \vdots \\ r(p + \Delta) \end{bmatrix} \]  \hspace{2cm} (2.49)

Alternatively,

\[ Rw = -r_\Delta \hspace{1cm} \text{where, } l = 1, \ldots, p \]  \hspace{2cm} (2.50)

The Constrained AR Yule-Walker Equations

Consider an all-pole model (i.e., \( q = 0 \)) of the form:

\[ H(z) = \frac{b(0)}{1 + \sum_{k=1}^{P} a(k)z^{-k}} \]  \hspace{2cm} (2.51)
Revisiting Eqn. (2.26), it can be seen that when $q = 0$,

$$r(k) + \sum_{l=1}^{p} a(l)r(k-l) = b(0)\mathcal{E}[v(n)x^*(n-k)]$$

(2.52)

The right hand side of Eqn. (2.52) may be simplified by realizing that $\mathcal{E}[v(n)x^*(n-k)]$ is zero when $k > 0$, since $v(n)$ is uncorrelated with $x(n)$ at time $n-k$. Therefore, expressing Eqn. (2.31) for $k = 1, 2, \ldots, p$ yields the Yule-Walker equations for the special case of an AR model [52].

$$
\begin{bmatrix}
  r(0) & r^*(1) & \ldots & r^*(p-1) \\
  r(1) & r(0) & \ldots & r^*(p-2) \\
  \vdots & \vdots & \ddots & \vdots \\
  r(p-1) & r(p-2) & \ldots & r(0)
\end{bmatrix}
\begin{bmatrix}
  a(1) \\
  a(2) \\
  \vdots \\
  a(p)
\end{bmatrix}
= -
\begin{bmatrix}
  r(1) \\
  r(2) \\
  \vdots \\
  r(p)
\end{bmatrix}
$$

(2.53)

$$Ra = -r$$

(2.54)

From the above definitions, several analogies come to light.

Firstly, when $\Delta = 0$, Eqn. (2.53) is identical to Eqn. (2.49), since Eqn. (2.49) is constrained to a one step predictor. While setting $\Delta = 0$ may be interpreted as zero-step prediction (see figure 2.10), the two sets of equations should not be confused; since, in the Yule-Walker case (Eqn. (2.53)), an estimate of $x(n)$ is obtained from a weighted sum of past outputs $x(n-1) \ldots x(n-p)$ and so the Yule-Walker equations implement a "one step predictor" as default. Therefore, Eqn. (2.49) differs by the inclusion of a "prediction depth" term $(x^{-\Delta})$, that delays the model input data by $\Delta$ samples.

Secondly, when comparing Eqn. (2.53) to the special case of a constrained all-pole model using Prony's method, it can be seen that the two sets of equations are also identical, but differ only in the definition of the autocorrelation sequence. For the normal equations of Prony's method, $r(k)$ is a deterministic autocorrelation sequence, whereas in the case of the Yule-Walker equations, $r(k)$ is a statistical autocorrelation.
2.7.2 The Moving Average Model

The model's output may be expressed as

$$\hat{x}(n|n - \Delta) = \sum_{k=0}^{q} w_k x(n - k - \Delta) \tag{2.55}$$

Combining Eqn. (2.55) with Eqn. (2.43) and Eqn. (2.46)

$$\mathbb{E} \left[ (x(n) - \sum_{k=0}^{q} w_k x(n - k - \Delta)) x^*(n - l - \Delta) \right] = 0 \quad l = 0, 1, \ldots q \tag{2.56}$$

Summarizing the result, yields

$$r(l + \Delta) = \sum_{k=0}^{q} w_k r(l - k) \quad l = 0, 1, \ldots q \tag{2.57}$$

$$\begin{bmatrix} r(0) & r^*(1) & \cdots & r^*(q) \\ r(1) & r(0) & \cdots & r^*(q-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(q) & r(q-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(q) \end{bmatrix} = \begin{bmatrix} r(\Delta) \\ r(1 + \Delta) \\ \vdots \\ r(q + \Delta) \end{bmatrix} \tag{2.58}$$

Alternatively,

$$Rw = r_\Delta \tag{2.59}$$

A technique of “multi-step prediction” for a moving average model has been established in Eqn. (2.58). Notice that regardless of the value of \( \Delta \), the autocorrelation matrix, \( R \) remains unchanged. Another interesting analogy, for when \( l = k = 1, \Delta = 0 \) and \( w(0) = 1 \) is that the Wiener-Hopf equations of Eqn. (2.58) are similar to the all-pole Yule-Walker equations of Eqn. (2.53). This is as expected since, both sets of equations are implementing a “one-step predictor” as default. This is illustrated in figure 2.11.

2.8 The LMS Algorithm

Wiener filters are difficult to implement in real time as they require a matrix inversion of the autocorrelation matrix, \( R \), which may lead to computational errors due to finite precision. Secondly, solving the Wiener-Hopf equations of Eqn. (2.42) requires a priori
2.8. The LMS Algorithm

The LMS (least mean square) algorithm is a stochastic gradient algorithm that adjusts the weights of an FIR filter on a sample-by-sample basis as to minimize the instantaneous squared error.

According to the method of steepest descent [32], the updated value of the weight vector, \( \mathbf{w} \) at time \( n+1 \) may be computed by the simple recursive relation:

\[
\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \left[ -\nabla J(n) \right]
\]  

(2.60)

where \( \mu \) is the step size, controlling the algorithm's stability and rate of convergence. From Eqn. (2.60) it can be seen that the increment from \( \mathbf{w}(n) \) to \( \mathbf{w}(n+1) \) is in the negative gradient direction. Hence, the weights will approximately follow a steepest descent trajectory on the error performance surface to the bottom of the bowl. An exact measurement of the gradient vector Eqn. (2.61) is not possible, since this would

![Figure 2.11: One step prediction: 10th order FIR Wiener Filter Vs. 10th order All-pole Yule-Walker method filter when modelling a 4th order all-pole stochastic signal.](image)

knowledge of the cross-correlation vector, \( r_{xu} \) and \( R \), which may not be known. Also, if the signals are non-stationary, then \( \mathbf{w}_{opt} \) will have to be re-calculated, as \( R \) and \( r_{xu} \) will be changing with time.

Widrow and Hoff [93] devised a technique of obtaining \( \mathbf{w}_{opt} \) without the need for explicitly calculating \( R \) and \( r_{xu} \). The famous LMS (least mean square) algorithm is a stochastic gradient algorithm that adjusts the weights of an FIR filter on a sample-by-sample basis as to minimize the instantaneous squared error.
require knowledge of both $R$ and $r_{xu}$, which are not known. To circumvent this problem, Widrow proposed the substitution of instantaneous estimates of $R$ and $r_{xu}$ into Eqn. (2.61).

$$\nabla J(n) = -2r_{xu} + 2Rw(n)$$  \hspace{1cm} (2.61)

An instantaneous estimate of the gradient vector may therefore be expressed as:

$$\hat{\nabla}J(n) = -2u^*(n)x(n) - u^H(n)w(n) = -2u^*(n)e(n)$$  \hspace{1cm} (2.62)

Substituting Eqn. (2.62) into Eqn. (2.60) yields a simple expression, suitable for real-time implementation [91].

$$w(n+1) = w(n) + \mu u^*(n)e(n)$$  \hspace{1cm} (2.63)

### 2.8.1 The Variable Step Size LMS Algorithm

The variable step (VS) size algorithm is another stochastic gradient algorithm, that utilizes an independent step size constant, $\mu_p$, for each weight of the FIR filter implementation [20, 28]. The value of each step size constant, $\mu_p$ is allowed to vary between $(\mu_{\text{min}}$ and $\mu_{\text{max}}$) according to an estimate of the distance to the mean-square error minimum (see figure 2.9) therefore providing more rapid convergence. The algorithm was shown to reduce conventional convergence times by up to a factor of 50, with only a small increase in computation. A method for updating is the "frequency of sign change" [40]. Basically, the values are allowed to increase and decrease as the algorithm detects the crossing and non-crossing, respectively, of the minimum of the error performance surface in the $p$th dimension.

### 2.8.2 Variants of the LMS Algorithm

There are many more variants of the standard LMS algorithm, most have been designed for efficient real time implementation [46]. However, it should be noted that performance is dependent upon the actual type of algorithm used.

A final interesting variant of the standard LMS algorithm, proposed by Martinez-Ramón et al. [53] adaptively combines two LMS algorithms (i.e., two FIR filters with
different weight vectors; one with a large step size and the other with a small step size). Although the overall effect will be that of one filter, Martinez-Ramón et al. were able to demonstrate the algorithm's ability to track fast changes and yet still achieve a low modelling error once the weights converged for periods of stationary data.

2.9 The RLS Algorithm

The recursive least squares (RLS) algorithm differs from the previous set of algorithms since it minimizes a deterministic squared error. However, it is interesting to compare it to the LMS algorithm, since for the RLS algorithm, past or "old" input data is exponentially weighted in order to allow "new" data to improve the estimate of the weight vector. Data weighting is controlled by a "forgetting factor" and since the algorithm replaces \( \mu \), with an inverse correlation matrix of the input vector, the algorithm has a much faster rate of convergence. However, the improvement in performance is achieved at the expense of greater computational complexity [32].

2.10 The Kalman Filter

The Kalman filter is another linear optimum filter that is mathematically formulated in terms of state space concepts [9, 32]. Due to the state-space formulation, the Kalman filter, unlike Wiener filter, is able to analyze both linear and non-linear dynamic systems or processes. Also, due to the many similarities that exist between the Kalman and RLS algorithms, the RLS algorithm is a special case of the Kalman filter, as reported by Haykin [32]. Thus, the Kalman filter, although complicated, has found many uses in such fields as navigation and adaptive equalization of telephone channels [9].
2.11 Model Order Identification and Validation

Model validation is a fundamental part of the system identification procedure for time domain modelling techniques. Since analyzing different sets of data usually results in many models of different orders and structures. The designer is therefore left with the dilemma of which model describes the data or plant most adequately for their purpose. Although no definitive validation technique exists, for all types of data, it should be noted that "model validation" is somewhat empirical and is ultimately dependent upon the judgement of the designer. The model validation techniques discussed in this chapter may therefore only be used for giving an approximate indication to the true order of a system.

An interesting rule of thumb used for mobile phone algorithms based on the all-pole LPC model, such as the GSM algorithm [27] is the assignment of a complex conjugate pole pair per formant. Since most voiced adult human speech is limited to about 4kHz, this manifests itself as one formant/kHz and therefore may be fairly accurately represented with 8 poles. However, other speech coding algorithms [79], encode the frames of speech with 10 poles, but having too many poles may cause the formants to split in two, this is referred to as spectral line splitting [36] and inevitably leads to poor modelling performance.

2.11.1 Final Prediction Error (FPE) and The Akaike Information Criterion (AIC)

Two prominent model estimation criteria have been proposed by Akaike [1]. The first one, FPE or final prediction error criterion, attempts to estimate the true model order, \( p \) for a data sequence of length \( N \), by estimating the prediction error power of a \( k \)th order model.

\[
FPE(k) = \frac{N + k}{N - k} \hat{\rho}_k
\]

(2.64)

where \( \hat{\rho}_k \) is the prediction error power or estimate of the input white noise variance.

Therefore, for an all-pole model, the Yule-Walker equations become

\[
\hat{\rho}_k = r(0) + \sum_{k=1}^{p} a(k)r(k)
\]

(2.65)
2.11. Model Order Identification and Validation

Analyzing Eqn. (2.64) it can be seen that as $\rho_k$ decreases with $k$, but the term \( \frac{k+1}{N-k} \) actually increases with $k$. Subsequent improvements led to the development of the AIC, as shown below.

$$AIC(k) = N \ln \rho_k + 2k \quad k = 1 \cdots m$$ (2.66)

since the true model order, $p$ is not known, autoregressive models of orders ranging from 1 to $m$ are fitted to the data using the Yule-Walker equations, where $m$ is believed to be larger than $p$. $\rho_k$ is calculated using Eqn. (2.65). Notice that the AIC has a penalty term $2k$ for any extra autoregressive coefficients that do not significantly reduce the prediction error power. Kashyap [35] observed that the AIC criterion tended to underestimate the model order for non-autoregressive processes and is statistically inconsistent as $N \to \infty$. Further work undertaken by Wax and Kailath [84] suggested that the AIC actually overestimates the order as $N$ increases.

2.11.2 Minimum Description Length (MDL)

Rissanen et al. [5, 70] provided a possible solution to the model order estimation problem by replacing the AIC penalty term, $2k$ with $k \ln(N)$, since this decays faster as $N$ increases.

$$MDL(k) = N \ln \rho_k + k \ln(N) \quad k = 1 \cdots m$$ (2.67)

Rissanen demonstrated that the MDL is a consistent model order estimator, converging to the true model order as $N$ increases. Notice that the penalty for over parametrization is much more severe than for Akaike's rule.

Figure 2.12 compares the performance of the AIC over MDL for estimating the order of a 4th order AR process excited with bandlimited Gaussian white noise, where $N=500$. Notice that as expected, AIC has overestimated the model order, since the global minimum occurs at 7, whereas MDL has obtained the true order of 4. Notice that both methods have knee at 4, but this is only due to the fact that an AR model is being used, other models would produce different results.
2.11.3 Criterion Autoregressive Transfer (CAT)

Another AR (autoregressive) model order estimation technique proposed by Parzen [63] is CAT (Criterion Autoregressive Transfer) and may be defined as

\[
CAT(k) = \frac{1}{N} \sum_{j=1}^{k} \frac{N - j}{N \hat{\theta}_j} - \frac{1}{\hat{\theta}_k}
\]  

(2.68)

The AR model order, \(k\) is selected such that \(CAT(k)\) is minimized. CAT chooses the AR model order such that the error between the model and an optimal infinite length filter is minimized [38, 47].
2.12 Summary and Conclusions

In this chapter, a comprehensive overview of time domain modelling techniques has been presented. There are two classes of modelling algorithms, as summarized below.

**Deterministic**
- Padé Approximation
- Prony's Method
- Shanks' Method
- Steiglitz-McBride Method
- Recursive Least Squares (RLS)

**Stochastic**
- Yule-Walker Equations
- Modified Yule-Walker Equations (MYWE)
- Least Squares Modified Yule-Walker Equations (LSMYWE)
- Wiener-Hopf Equations
- Least Mean Square (LMS)
- Kalman Filter

The assumption for deterministic modelling, was that $x(n)$ was known for all $n$ or for values of $n$ over a finite interval. The model parameters were determined by minimizing the deterministic squared error of the signal. The Padé approximation fairs quite badly to Prony’s method, since the modelling error for $n > p + q$ is not considered in the analysis. However, this is only true if the system’s order is not met.

Shanks’ method and the Steiglitz-McBride method improve the accuracy of the pole-zero Prony model, by applying a least squares minimization to the model error in order to be able to model the numerator coefficients more accurately.

Stochastic modelling assumes that $x(n)$ is a random process and therefore may be described probabilistically. The model parameters for the Yule-Walker method and the FIR Wiener filter were determined by statistically minimizing the mean squared error or MSE.

In section 2.7 several comparisons were made between the various types of stochastic modelling methods. It was shown that in the special case of one step linear prediction, the Wiener-Hopf equations were similar to the Yule-Walker Equations when constrained to an all-pole model. The least squares modified Yule-Walker Equations (LSMYWE), have superior modelling performance than the modified Yule-Walker Equations (MYWE) since a set of overdetermined equations are used to estimate the AR coefficients.
The LMS algorithm is a stochastic gradient based algorithm, that offers an attractive solution for implementing a Wiener filter in real time. Rather than calculating an optimum set of weights all at once, the LMS algorithm produces estimates of the weights, Eqn. (2.63), which improve over time. Eventually, the weights converge and fluctuate around the Wiener solution.

The VS LMS algorithm provides a reduction in convergence time over the conventional LMS algorithm, by adaptively controlling the step size (rate of convergence) for each weight of the digital filter. There are many variants of the standard LMS algorithm mainly for efficient real time implementation.

Finally, the RLS algorithm and the Kalman filter are discussed. Due to the many similarities between the Kalman and RLS algorithms, the RLS algorithm is a special case of the Kalman filter, as reported by Haykin [32]. Also notice that the RLS algorithm is a deterministic algorithm, whereas the Kalman filter is stochastic.

The latter part of the chapter addressed the issue of model order identification, where several important model order identification techniques were discussed. Further work undertaken by Xu and Kaveh [95] showed that model order estimation, based on AIC and MDL was very sensitive to deviations of the statistical structure of the idealized Gaussian white noise model. Since, when a data series was corrupted with coloured Gaussian noise, both MDL and AIC tended to overestimate the true model order. Livaas and Regalia [49] provided some rules and cases in which application of the ITC (Information theoretic criterion) was likely to lead to overestimation or underestimation. Landers and Lacos [47] provide an interesting comparison of all four model order identification discussed in section 2.11.
Chapter 3

Cepstral Deconvolution

3.1 Introduction

Cepstral deconvolution approaches the pole-zero modelling problem in the frequency domain, by applying a non-linear function to the signal's spectrum in order to reduce the model's order. Traditionally, the "log cepstrum" has been successfully applied to speech coding and speech recognition. Since, according to the basic speech synthesis model (see figure 1.4), speech is composed of an excitation sequence linearly convolved with the impulse response of the vocal tract transfer function. The process of cepstral deconvolution attempts to deconvolve the excitation from the vocal tract transfer function without making any of the assumptions that were necessary for LPC. Thus, the cepstrum is able to model the effects of both poles and zeros, since the deconvolution process makes no assumption about the statistics of the excitation and has no linear model. Therefore, the cepstrum can be viewed as an alternative method of system modelling.

The cepstral deconvolution process may be described by the block diagram of figure 3.1, where, "$F$" represents the Discrete Time Fourier Transform (DTFT) and $\eta$ is discrete quefrency (as explained later).

\[ s(n) \xrightarrow{F} \text{Non-linear Function} \xrightarrow{F^{-1}} c(\eta) \]

Figure 3.1: Cepstral Deconvolution.
3.2 The Real Cepstrum

Consider a frame of speech data, \( s(n) \). The real cepstrum (RC) makes use of a logarithmic function as the non-linear operator and may be calculated by determining the logarithm of the magnitude of the Fourier transform of \( s(n) \), then obtaining the inverse Fourier Transform of the resulting sequence.

\[
e(n) = \mathcal{F}^{-1} \left\{ \log |\mathcal{F}(s(n))| \right\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |S(w)| e^{i\omega n} dw
\]  

(3.1)

Viewing the speech magnitude spectrum, \( |S(w)| \), as consisting of a “quickly varying” part, \( |U(w)| \) and a “slowly varying” part, \( |P(w)| \) Noll [59] obtained the following relationship.

\[
\log |S(w)| = |\log(U(w))| + |\log(P(w))| 
\]  

(3.2)

The excitation spectra and vocal tract spectra are now additive rather than convolved. Noll believed that by analyzing the two signals as time signals, one high frequency (excitation) and one low frequency (envelope), the excitation would manifest itself as “high frequency ripple” in the logarithm spectrum, whereas, the envelope would appear as “low frequency ripple”. Hence, the effects of the vocal tract and excitation may be separated. The word “quefrency” was assigned to describe the frequency of the spectral ripples in this new pseudo time domain [6].

Figure 3.2 shows the real cepstrum for a frame of voiced speech. Since the real cepstrum is “the magnitude of spectrum” and is therefore symmetrical about the centre point, the analysis may be restricted to the left hand side only. For this particular piece of speech, the RC has provided two pieces of information - the impulse response of the vocal tract and the pitch or excitation. Notice how the pitch, \( \tau_p \) manifests itself as a large spike at about 60 quefrency time samples. The log operator has rapidly decayed all of the vocal tract information and has pushed it into the corners, thus providing good separation. The pitch information or the vocal tract information may be extracted by applying window function or lifter to the relevant section of the cepstrum. Figure 3.3 shows the resulting spectral envelope of the vocal tract and compares it to the spectral envelope obtained by 12th order LPC analysis. At first glance, figure 3.3 appears to show that both the LPC envelope (blue) and the RC envelope (red) model the original
3.3. The Complex Cepstrum

The real cepstrum provides a method for analyzing the magnitude frequency response of a system, such as the human vocal tract. However, as an analytic process it is limited since the phase is discarded and therefore only part of the information is retained. The complex cepstrum [17, 69] provides an analysis method which retains all of the signal’s information and is therefore a fully reversible process. However, for the complex cepstrum to exist, it is required that \( \log S(w) \) must be a continuous function of \( w \) [60].
therefore requiring a phase unwrapping algorithm. This indeed is a difficult task, as phase unwrapping algorithms are seldom totally successful, as their performance is severely affected in the presence of noise [65].

The complex cepstrum of the signal $s(n)$ is defined similarly to the real cepstrum.

$$c(\eta) = \mathcal{F}^{-1}\{\log \mathcal{F} \{s(n)\}\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |S(w)| e^{j\eta} dw$$  \hspace{1cm} (3.3)

Utilizing the properties of the logarithm and realizing that the logarithm of the spectrum, $S(w)$ may be may be expressed as,

$$\log S(w) = \log |S(w)| + j\theta$$  \hspace{1cm} (3.4)

where, $\theta = \arg[S(w)]$. Analyzing Eqn. (3.3) more carefully, it can be seen that the complex cepstrum is simply the real cepstrum + phase cepstrum.

$$c(\eta) = \mathcal{F}^{-1}\{\log |S(w)|\} + \mathcal{F}^{-1}\{j\eta\}$$  \hspace{1cm} (3.5)

Figure 3.4 shows the associated phase cepstrum for the frame of voiced speech shown in figure 3.3, using Matlab's phase unwrapping algorithm [87].
3.3. The Complex Cepstrum

Figure 3.4: Phase cepstrum for the frame of voiced speech of figure 3.3.

3.3.1 Modelling Rational Systems

Consider a class of sequences with the rational z-transform given by

\[
X(z) = G \cdot z^{-r} \cdot \prod_{k=1}^{q_1} (1 - a_k z^{-1}) \prod_{k=1}^{q_0} (1 - b_k z) \prod_{k=1}^{p_1} (1 - c_k z^{-1}) \prod_{k=1}^{p_0} (1 - d_k z)
\]  

(3.6)

where \((1 - a_k z^{-1})\) and \((1 - c_k z^{-1})\) are the zeros and poles “inside” the unit circle and \((1 - b_k z)\) and \((1 - d_k z)\) are the zeros and poles “outside” the unit circle, with \(|a_k|, |b_k|, |c_k|, |d_k| < 1\). The term \(z^{-r}\) represents a delay of the sequence to the time origin, which may be eliminated by a linear shift of the sequence. Assuming that the phase term can be estimated and therefore removed and that the system gain, \(G\) is positive, the logarithm function may be applied to Eqn. (3.6).

\[
\tilde{X}(z) = \log(G) + \sum_{k=1}^{q_1} \log(1 - a_k z^{-1}) + \sum_{k=1}^{q_0} \log(1 - b_k z) - \sum_{k=1}^{p_1} \log(1 - c_k z^{-1}) - \sum_{k=1}^{p_0} \log(1 - d_k z)
\]

(3.7)

where \(\log(G)\) is a constant and so under the inverse Fourier transform it becomes a delta function at the origin scaled by \(\log(G)\). All other terms sum to zero at \(n=0\). The
contribution of the poles and zeros may be expressed as a power series expansion, since
\[
\log(1 - \alpha) = -\alpha - \frac{\alpha^2}{2} - \frac{\alpha^3}{3} \cdots \text{ for } |\alpha| < 1
\]

\[
\log(1 - a_k z^{-1}) = -\sum_{n=1}^{\infty} \frac{a_k^n}{n} z^{-n}, \quad |a_k z^{-1}| < 1 \quad (3.8)
\]

\[
\log(1 - b_k z) = -\sum_{n=1}^{\infty} \frac{b_k^n}{n} z^n, \quad |b| < 1 \quad (3.9)
\]

Analyzing Eqn. (3.8) it can be concluded that the z-transform is for a causal sequence, corresponding to the right hand side of the cepstral domain. Similarly, Eqn. (3.9) is a non-causal sequence, corresponding to the left hand side of the cepstral domain.

\[
\tilde{\alpha}(\eta) = \log(G)\delta(n) - \left[ \sum_{k=1}^{\eta} \sum_{n=1}^{\infty} \frac{a_k^n}{n} - \sum_{k=1}^{\eta} \sum_{n=1}^{\infty} \frac{b_k^n}{n} \right] u(n-1)
\]

\[
+ \left[ \sum_{k=1}^{\eta} \sum_{n=1}^{\infty} \frac{b_k^n}{n} - \sum_{k=1}^{\eta} \sum_{n=1}^{\infty} \frac{d_k^n}{n} \right] u(-n+1) \quad (3.10)
\]

Therefore, the poles and zeros inside the unit circle manifest themselves on the right hand side of the cepstral domain, whereas the poles and zeros outside the unit circle manifest themselves on the left hand side of the cepstral domain, as shown in figure 3.5. It should be clear that this is only true for the complex cepstrum and not the RC, since phase is discarded.

### 3.3.2 Noise and Error Reduction Techniques

Since noise is usually present in most data sequences, it seems reasonable to suggest that higher order quefrency components may contain more noise than signal. Childers et al. [15] suggested that zeroing the higher order quefrency components, by a rectangular or Hanning lifter, resulted in a significant improvement in signal recovery.

Childers also suggested that interpolating (padding with zeros) benefits the complex cepstrum in two ways: firstly, increasing the sampling rate in the frequency domain reduces the aliasing of the complex cepstrum. Secondly, since the sampling instants or steps of the phase curve are much more closer together, phase unwrapping errors can be reduced, since there are now fewer “jumps” greater than \(\pi\).
3.4 The Root Cepstrum

Further work undertaken by Lim [50] suggested that log was not suitable for all types of signals. Lim’s root cepstral deconvolution (RCD) system makes use of an alternative non-linear function, whereby the log and antilog (inverse process) functions are replaced by $(\cdot)^\gamma$ and $(\cdot)^{1/\gamma}$ and where $\gamma$ is constrained to lie between $-1 \leq \gamma \leq 1$. The root cepstrum is defined as follows:

$$c(\eta) = \mathcal{F}^{-1}\left\{ \left(S(w)\right)^\gamma \right\}$$

(3.11)

Applying Eqn. (3.6) to the root cepstral concept.

$$X(z) = G^\gamma \cdot \frac{\prod_{k=1}^{q_a} (1 - a_k z^{-1})^\gamma \prod_{k=1}^{q_b} (1 - b_k z)^\gamma}{\prod_{k=1}^{p_a} (1 - c_k z^{-1})^\gamma \prod_{k=1}^{p_b} (1 - d_k z)^\gamma}$$

(3.12)

This may be simplified further by discarding all poles and zeros outside the unit circle.
and therefore assuming that all systems are minimum phase.

\[ X(z) = G^n \cdot \prod_{k=1}^{\infty} \frac{(1 - a_k z^{-1})^\gamma}{(1 - c_k z^{-1})^\gamma} \]  

The contribution of the poles and zeros may be expressed as a binomial series expansion, where \( 0 \leq k \leq \infty \)

\[(1 - a z^{-1})^\gamma = 1 - a z^{-1} + \frac{a^2 z^{-2}}{2!} - \gamma(\gamma - 1) + \cdots + \frac{-a^k z^{-k}}{k!} - \gamma(\gamma - 1)(\gamma - 2) \cdots (\gamma - k + 1) \]  

(3.14)

As with the logarithmic cepstrum, \( G^n \) is a constant and so under the inverse Fourier transform it becomes a delta function scaled by \( G^n \).

It is interesting to note, that a relationship between the traditional log cepstrum and the root cepstrum may be expressed as follows [50]. Let \( \hat{X}(z) = \log X(z) \) and \( \hat{X}(z) = (X(z))^\gamma \)

\[ \hat{X}(z) = e^{\gamma X(z)} = 1 + \gamma \hat{X}(z) + \frac{\gamma^2}{2!} \hat{X}^2(z) + \cdots \]  

(3.15)

Consider the signal \( s(n) = u(n) * p(n) \), where \( u(n) \) is an impulse excitation and \( p(n) \) is the plant transfer function.

\[ S(w) = U(w) \cdot P(w) \quad \text{where,} \quad P(w) = \frac{B(z)}{A(z)} \]

\[ \therefore S(w) = G \cdot \left. \frac{B(z)}{A(z)} \right|_{z = e^{j\omega}} \]  

(3.16)

where \( A(z) \) and \( B(z) \) are polynomials in "z" with roots which are the system poles and zeros respectively. Therefore, an estimate of the plant may be expressed as,

\[ \hat{p}(n) = \mathcal{F}^{-1} \left\{ \left( \mathcal{F} \left( c(\eta) l(\eta) \right) \right)^{1/\gamma} \right\} \]  

(3.17)

where \( l(\eta) \) is a rectangular window function or lifter.

3.4.1 Special Properties of the Root Cepstrum

There are two special cases of the root cepstrum, when \( \gamma = \pm 1 \). Consider the special case of an all pole system.

\[ c(\eta) = \mathcal{F}^{-1} \left\{ \left( \frac{G}{A(z)} \right)^\gamma \right\} \]  

(3.18)
3.4. The Root Cepstrum

Since $\gamma$ alters the rate of convergence of the cepstral coefficients in the “$\gamma$-cepstral domain”, when $\gamma = -1$ maximum convergence or compression is achieved, since the all-pole system is transformed into an all-zero system of finite length. Thus, the length of lifter, required to capture all of the system’s coefficients, has been reduced to the order of the system. The same can be said about an all-zero system, when $\gamma = 1$ [50].

Note that for a noiseless signal, the cepstral coefficients obtained from these two special cases are identical to the coefficients derived from the all-pole equations of linear prediction, ($\gamma = -1$) when and those obtained from the Wiener-Hopf equation when ($\gamma = 1$).

To illustrate the concept of convergence in the cepstral domain, figure 3.6 compares the log cepstrum to the root cepstrum for $\gamma = \pm 1$. The impulse response, $p(n)$ for a second order all-pole system is shown in (a). The log cepstrum (b) performs quite badly, as hardly any of the original system, $p(n)$ has been reconstructed. As expected, for the root cepstrum, when $\gamma = -1$ (c), $p(n)$ has been perfectly reconstructed. Finally, when $\gamma = 1$, a truncated version of $p(n)$ is obtained. In all cases, the lifter was held constant at 3 (i.e., 3 cepstral coefficients). Lim demonstrated that as $\gamma \rightarrow 0$ the non linear function became similar to that of a log function, and therefore behaved identically to the log cepstrum.

3.4.2 Pole-zero Modelling and its Implications

In section 3.4.1 it was shown that the RCD method was able to model an all-pole or all-zero system, when $\gamma = \pm 1$. However, a problem arises for pole-zero or mixed models, since regardless of the value of $\gamma$, the impulse response will be of infinite duration, as evident from Eqn. (3.14). Practically speaking, it is not possible to capture all of the information, since the data records are finite. Lim empirically demonstrated that it was possible to model an a priori pole-zero process by varying $\gamma$ between $\pm 1$. The problem, like all cepstral methods, is that no cost function is available and it is therefore impossible to know whether the cepstral coefficients are the “best fit” for a given system.
Chapter 3. Cepstral Deconvolution

Figure 3.6: Reconstruction using the root cepstrum and the log cepstrum (lifter=3) for a second order all-pole system. The impulse response, $p(n)$ to be modelled (a). log cepstrum (b). Root cepstrum, $\gamma = -1$ (c). Root cepstrum, $\gamma = 1$ (d). Notice that when $\gamma = -1$, as shown in (c) the reconstruction is perfect.

3.4.3 Noise Immunity

Revisiting Eqn. (3.11), it can be seen that by raising $S(w)$ to the power $\gamma$, for certain values of $\gamma$, such as 1, 0.5, etc. it reasonable to assume that the root cepstrum is more immune to noise than the traditional log cepstrum [2, 72]. To demonstrate this, figure 3.7 compares the noise immunity performance of the log cepstrum (a) and the root cepstrum $\gamma = 0.5$ (b) for a mixed model. The model's impulse response, $y(n)$ was corrupted with bandlimited Gaussian white noise, $v(n)$ and applied to the two modelling methods. As expected, the effect of $v(n)$ has severely affected the performance of the log cepstrum, but not the root cepstrum.
3.5. Other Non-Linear Functions

Another non-linear function proposed by Koboyashi and Imai [41], replaced Lim’s \((\cdot)^\gamma\) and \((\cdot)^{1/\gamma}\) operators with \(1/\gamma((\cdot)^\gamma - 1)\) and \([1+(\cdot)^1/\gamma]^{1/\gamma}\). The “generalized log function” as it has come to be known, converges better to the log operator as \(\gamma \to 0\). Consider a complex function, \(s_\gamma(x)\) of a complex variable \(x\)

\[
s_\gamma(x) = \frac{1}{\gamma} (x^\gamma - 1), \quad \text{where, } \gamma \neq 0
\]

(3.19)

Since \(s_\gamma(x)\) cannot be defined when \(\gamma = 0\), Kobayashi demonstrated that limit of \(s_\gamma(x)\) as \(\gamma \to 0\) may be expressed as a Taylor series.

\[
s_\gamma(x) = x - 1 + \sum_{k=2}^{\infty} \frac{(\gamma - 1)(\gamma - 2)\ldots(\gamma - k + 1)}{k!} (x - 1)^k
\]

(3.20)

\[
\lim_{\gamma \to 0} s_\gamma(x) = x - 1 - \frac{(x - 1)^2}{2!} + \frac{(x - 1)^3}{3!} \ldots = \log x
\]

(3.21)
\[ s_\gamma(x) = \begin{cases} \frac{1}{\gamma}(x^\gamma - 1), & 0 < \gamma \leq 1 \\ \log x, & \gamma = 0 \end{cases} \quad (3.22) \]

From the definition of \( s_\gamma(x) \), the inverse function of \( s_\gamma(x) \) is given by,

\[ s_\gamma^{-1}(x) = \begin{cases} (1 + \gamma x)^{1/\gamma}, & 0 < \gamma \leq 1 \\ \exp x, & \gamma = 0 \end{cases} \quad (3.23) \]

Figure 3.8: Kobayashi and Imai's Non-linear function.

Figure 3.9: Lim's Non-linear function.
These functions were taken further by Tokuda et al. [78], who demonstrated that the generalized cepstrum, \( c_{\gamma}(k) \) is the defined as the inverse Fourier transform of the generalized logarithm of a spectrum, \( X(z) \)

\[
c_{\gamma}(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} s_{\gamma}(X(z)) e^{i\omega k} d\omega
\] (3.24)

Tokuda suggested that the speech spectrum, \( P(e^{j\omega}) \) may be modelled by generalized cepstral coefficients, as follows:

\[
s_{\gamma}\left(P(z)\right) = \sum_{k=0}^{M} c_{\gamma}(k)z^{-k}
\] (3.25)

\[
P(z) = s_{\gamma}^{-1}\left(\sum_{k=0}^{M} c_{\gamma}(k)z^{-k}\right)
\] (3.26)

\[
P(z) = \begin{cases} 
K \cdot \left(1 + \gamma \sum_{k=1}^{M} \bar{c}_{\gamma}(k)z^{-k}\right)^{1/\gamma}, & 0 < |\gamma| \leq 1 \\
K \cdot \exp\sum_{k=1}^{M} \bar{c}_{\gamma}(k)z^{-k}, & \gamma = 0
\end{cases}
\] (3.27)

where, \( K = s_{\gamma}^{-1}(c_{\gamma}(0)) \) and \( \bar{c}_{\gamma}(k) \) are the normalized cepstral coefficients.

\[
\bar{c}_{\gamma}(k) = \frac{c_{\gamma}(k)}{(1 + \gamma c_{\gamma}(0))}, \quad k \geq 1
\] (3.28)

Spectral estimates obtained by the generalized cepstrum lead to spectral bias caused by linear smoothing (lifiting the cepstral coefficients) of the generalized cepstra. To circumvent this problem, Tokuda applied the “Spectral criterion function” used in the unbiased estimate of the log cepstrum to the spectral model in order to minimize the error. As a consequence, when \( \gamma = -1 \) the resulting magnitude spectra was identical to that obtained from LPC for the following reasons: Firstly, Eqn. (3.27) is similar to the all-pole equations of linear prediction. Secondly, the minimization for \( \gamma = -1 \) is identical to that used for LPC. Likewise, when \( \gamma = 0 \), the results are similar to the unbiased real cepstrum.

Further developments replaced the LPC analysis section of a CELP coder [44] with the generalized cepstrum for speech coding applications [43].
3.6 The Mel Cepstrum

The concept of the Mel was developed from early speech analysis work undertaken in the 1940s [76]. A Mel is a unit of perceived pitch or frequency of a tone. Since the human auditory tract does not perceive pitch in a linear manner, the researchers were able to plot the characteristics of the "way we hear pitch". Their results demonstrated that the mapping between "physical frequency" and "perceived frequency" is approximately linear below 1kHz and logarithmic above, such an assumption is heavily used in speech recognition systems today. Further work undertaken by Fant [21] proposed an approximate formula for calculating Mels from physical frequency.

\[ F_{Mel} = 1127 \ln \left( 1 + \frac{f_{Hz}}{700} \right) \]  

(3.29)

The real cepstrum is particularly well suited to the computation of Eqn. (3.29) since it only operates on the magnitude of the spectrum.

3.7 The Differential Cepstrum

The differential cepstrum circumvents the need for phase unwrapping of the complex log cepstrum, by computing the derivative of the logarithm [66].

\[ \frac{d \log S(w)}{d w} = \frac{1}{S(w)} \cdot \frac{d S(w)}{d w} \]  

(3.30)

For a convolution, \( s(n) = u(n) \ast p(n) \), the logarithmic derivative may be expressed as,

\[ \frac{d \log S(w)}{d w} = \frac{1}{U(w)} \cdot \frac{d U(w)}{d w} + \frac{1}{P(w)} \cdot \frac{d P(w)}{d w} \]  

(3.31)

This concept may be extended to the root cepstrum, where the derivative of the spectrum, \( S(w)^{\gamma} \) may be expressed as,

\[ \frac{d S(w)^{\gamma}}{d w} = \gamma \cdot S(w)^{\gamma - 1} \cdot \frac{d S(w)}{d w} \]  

(3.32)

Therefore the Differential root cepstrum may be defined as,

\[ c_d(n) = F^{-1} \left\{ \gamma \cdot S(w)^{\gamma - 1} \cdot \frac{d S(w)}{d w} \right\} \]  

(3.33)

Once again, the differential root cepstrum does not require any phase unwrapping techniques and therefore provides an interesting alternative to analyzing phase cepstra, but at the expense of much more severe cepstral aliasing [60].
3.7.1 Group Delay

The first derivative of the phase spectrum, $\theta(w)$ is also referred to as the group delay, $\tau_g$.

$$\tau_g = -\frac{d \theta(w)}{dw}$$  \hspace{1cm} (3.34)

The differentiation is of particular use for analyzing phase, since an impulse response that does not start at time zero produces a phase spectrum shown in figure 3.10(a). Two important observations can be made: firstly, notice that the gradient of the unwrapped phase (a) is equal to the first value of the group delay (b). Secondly, it is not possible to see the "plant's phase" from (a), due to the initial offset, but this can be clearly seen in (b), although the reader should remember that this is not the actual system's phase, but the differential phase. Modifying the standard phase unwrappng algorithm, so that the time offset is removed can also lead to good results (see section 5.6).

Figure 3.10: (a) unwrapped phase, (b) group delay.
3.7.2 The Differential log Cepstrum and its Relationship to the Plant’s Impulse Response

Consider the plant impulse response, $p(n)$ of Eqn. (3.35), comprised of minimum, $i(n)$ and maximum, $o(n)$ phase components [57, 66].

$$p(n) = i(n) * o(n) \quad (3.35)$$

using Eqn. (3.30) and the taking inverse Fourier transform,

$$p_d(\eta) = i_d(\eta) + o_d(\eta) \quad (3.36)$$

where, $i_d(\eta)$ and $o_d(\eta)$ are the differential cepstra of $i(n)$ and $o(n)$. A relationship between the differential log cepstrum and plant’s impulse response [62] may be expressed as,

$$i_d(\eta) = \begin{cases} 
A^{(n-1)} & n \geq 2 \\
0 & n \leq 1 
\end{cases} \quad (3.37)$$

$$o_d(\eta) = \begin{cases} 
0 & n \geq 1 \\
-B^{(n-1)} & n \leq 0 
\end{cases} \quad (3.38)$$

where, $A^n$ and $B^n$ are the cepstral coefficients containing the minimum and maximum phase information.

$$A^n = \sum_{i=1}^{q_i} a^n_i - \sum_{i=1}^{p_i} c^n_i$$

$$B^n = \sum_{i=1}^{g^n} b^n_i \quad (3.39)$$

and $a^n_i, b^n_i$ and $c^n_i$ correspond to poles and zeros inside and outside the unit circle, as defined in Eqns. (3.6) and (3.7).

These equations may be solved by a recursive technique, as discussed in [66] and [62]. Note that this method requires no a priori knowledge of the plant (all-pole, all-zero or mixed) and circumvents the need for model order estimation, as the minimum and maximum phase components of the plant’s impulse response, $p(n)$ are reconstructed separately.
3.8 The Bicepstrum

The Bicepstrum combines the Bispectrum [55, 57, 58] together with the complex log cepstrum in order to model non-minimum phase systems. The virtue of the Bicepstrum over the traditional complex log cepstrum is its ability to model both deterministic and stochastic signals, since the plant's impulse response is reconstructed using third-order moments or cumulants without the need of a phase unwrapping algorithm. Nikias [57] demonstrated that the cepstral coefficients of Eqn. (3.39) may be calculated by using a two-dimensional Fourier transform, as shown in Eqn. (3.40) from which the cepstral coefficients may be calculated.

\[ n \cdot b_h(n, l) = \mathcal{F}_2^{-1} \left\{ \frac{\mathcal{F}_2[n \cdot m^3_{32}(n, l)]}{m^2_{32}(n, l)} \right\} \]  

(3.40)

where, \( m^3_{32}(n, l) \) are the cumulants, \( b_h(n, l) \) is the Bicepstrum of \( P(z) \) and \( \mathcal{F}_2 \) is a 2D Fourier transform. Alternatively, a least squares approach may be used, but requires some a priori knowledge of the pole-zero distribution [62] in order to be successful. However, both methods are dependent upon the cepstral coefficients \( \{A^n, B^n\} \) decaying exponentially, and are therefore limited to the logarithmic operator (i.e., the complex log cepstrum).

Perhaps the most useful feature of the Bicepstrum is its ability to suppress the effects of additive Gaussian noise of unknown spectrum characteristics [64, 65], and therefore it performs better than the complex cepstrum, due to the use of cumulants.

3.9 The LSMYWE-Cepstrum Recursion

Traditionally, the MYWE have been used to estimate the AR and MA coefficients of an ARMA process, but this requires a large AR model in order to be successful. Kaderli and Kayhan [33] proposed a novel scheme by combining the LSMYWE and the RC in order to calculate the MA coefficients. The LSMYWE method is used to estimate the AR coefficients and the MA coefficients are determined by combining the LSMYWE with the RC of the periodogram. Since, by differentiating the logarithm of the transfer function, \( P(z) \), the logarithm can be eliminated from the analysis and
therefore a relationship between the ARMA and cepstral coefficients may be obtained, as shown in Eqn. (3.41). Where, \( a(k), b(k), \) and \( c(k) \) are the autoregressive, moving average and cepstral coefficients respectively.

\[
b(k) = \frac{1}{k} \left[ kc(k)b(0) - \sum_{m=1}^{k-1} mb(m)a(k-m) + \sum_{m=1}^{k} ma(m)b(k-m) \right.
\]
\[
+ \sum_{i=1}^{k-1} i c(i) \sum_{m=0}^{k-i} a(m)b(k-i-m) \left. \right] \quad 1 \leq k \leq q \quad (3.41)
\]

Assuming that \( b(0) = 1 \) and the AR and cepstral coefficients are known, the MA coefficients (starting with \( k = 1 \)) may be determined in a recursive fashion. Kaderli reported that performance was poor when poles of an ARMA process were close to the unit circle. Therefore, the method was modified, such that cepstral estimates of the MA residual and not cepstral estimates from the actual ARMA process, \( x(n) \) were used. The method may be summarized as follows:

1. Apply an inverse filter to the original data sequence, \( x(n) \), using the AR coefficients obtained by the LSMYBE, in order to obtain the MA residual.
2. Obtain the RC of the MA residual.
3. Estimate MA coefficients using Eqn. (3.41) with \( a(0) = 1, b(0) = 1 \) and \( a(k) = 0, k > 0 \)

The results obtained demonstrated an improvement over Durbin's method [18], typically used for MA coefficient estimation with the MYBE method. Further work undertaken in this area, modified the original formulation of the log cepstrum of the periodogram, by including the Evolutionary periodogram [39] for non-stationary signals. Thus this new modified cepstrum was called the Evolutionary cepstrum [34] and is basically an extension to Kaderli and Kayhan's original method.
3.10 Summary and Conclusions

In this chapter several cepstral techniques for modelling pole-zero or mixed models in the frequency domain have been discussed. Cepstral deconvolution attempts to compress as much information into the lower order frequency indices, by applying a non-linear operator to the spectrum. Traditionally, a logarithmic function has been employed to perform the compression, since it has a fast rate of convergence. However, as demonstrated herein, a logarithmic function is not suitable for all types of signals and led researchers such as Lim, Kobayashi and Imai to devise new non-linear functions.

As shown empirically, the root cepstrum is more immune to noise than the log cepstrum, since, for certain powers of $\gamma$ such as, 1, 0.5... etc. the noise samples are not significantly amplified.

Analyzing phase cepstra becomes more difficult when an impulse response does not start at time zero leading to potentially more phase unwrapping errors. The differential cepstrum provides a possible solution to this problem, by differentiating the spectrum and therefore calculating the system's group delay, but at the expense of increased cepstral aliasing.

Comparing cepstral methods to LPC, it should be noted that all cepstral techniques may be used to model the effects of both poles and zeros of the plant, but are sub-optimal, since no performance function is used in the analysis. However, the complex cepstrum has two unique virtues: Firstly, it is able to analyze both minimum and maximum phase systems without the worry of stability. Secondly, since phase is retained, it possible to reconstruct the original signal, a quality nonexistent with the real cepstrum and LPC.

The Bicepstrum is a higher order spectral analysis technique that uses the cumulants or higher order moments and the complex log cepstrum in order to re-construct a non-minimum phase impulse response. The cepstra of the higher-order spectra or simply the polycepstra are of particular importance since, unlike the complex cepstrum, they may be used for stochastic (non-Gaussian) and deterministic modelling and do not require the use of a phase unwrapping algorithm.

Unfortunately, the Bicepstrum is dependent upon a logarithmic operator (the complex
log cepstrum) and is therefore unsuitable for the generalized modelling approach of the root cepstrum method.

Finally, the LSMYWE-Cepstrum recursion is discussed. This technique is a time-frequency domain technique that combines the least squares modified Yule-Walker equations (LSMYWE), as discussed in section 2.5.2 together with the real cepstrum. The recursion does not require any initial estimates or large AR model fitting and provides a simple alternative to existing methods for estimating the MA coefficients of an ARMA model.
Chapter 4

Theoretical Development

4.1 Introduction

As mentioned in the chapters 2 and 3, system identification can be undertaken in either the time domain or the frequency domain and there are a multitude of modelling techniques that are available to the system designer. However, time domain pole-zero techniques require some a priori knowledge of the number of poles and zeros in order to model a plant successfully. This information may not be available a priori but may be obtained by one of the model identification techniques as discussed in section 2.11. However, a level of scepticism should be exercised when using these techniques, since they may overestimate or underestimate the true model order in the presence of noise and their performance is usually poor for data sequences of a short duration.

As discussed in chapter 2, ARMA processes can be modelled using the Yule-Walker equations when constrained to an AR model of sufficiently high order. Although this circumvents the need for knowledge of the exact number of poles and zeros, the model is comparatively superfluous and by its very structure prone to instability in the presence of noise.

The Wiener-Hopf equations when implemented as an MA model are another useful set of equations for the system identification problem. The virtue of the MA model, is its ability to circumvent the model stability problems that are inherent to many all-pole/AR modelling schemes. However, in order to model the effect of the plant's poles
and zeros, the model is usually large and therefore not suitable for many applications where data conciseness is important.

Frequency domain modelling techniques, such as the cepstrum, approach the system identification modelling problem in an entirely different way to their time domain counterparts. Since, the cepstrum is a “non-parametric” modelling method (i.e., no linear model parameters are optimized) there is no need for model order identification and no issue of model stability, since by definition there is no linear model and therefore the technique uses windowed data in a warped time domain in order to deconvolve the excitation from the transfer function, such that only the transfer function is modelled. The method is particularly attractive since it does not require any a priori information about the structure of the plant (i.e., all-pole, all-zero or pole-zero). Although this is also true for the Yule-Walker and Wiener-Hopf equations, when constrained to an AR/MA model as mentioned above, the issue of an excessive number of coefficients remains unanswered, since it would be imprudent to use a large all-zero model for modelling an all-pole plant!

Finally, the complex cepstrum models the effect of both magnitude and phase and therefore provides a simple way for signal reconstruction. In comparison, almost all time domain techniques are phase blind due to the use of the autocorrelation matrix. However, this failure is compensated by the ease of efficient real time implementation.

Revisiting chapter 3 it should be clear that the cepstrum applies a non-linear operator to the spectrum of the signal in order to reduce the cepstral model’s order. The “non-linear operator” may be any type of non-linear function, but has traditionally been a logarithmic function, since the data in the cepstral domain is exponentially weighted and therefore compacted into the lower order cepstral coefficients. However, as mentioned in section 3.4, the function is not suitable for modelling all types of models and has led researchers to propose other more suitable functions for general cases – see figure 3.6.

All cepstral techniques rely upon the empirical good judgment of the designer when deciding the order of the cepstral model. This is greatly complicated for the root cepstrum, since the designer must now not only make a decision on the order, but
also the value of $\gamma$. In contrast, all time domain modelling techniques discussed herein have some sort of model error minimization or cost function. The cepstrum has no performance or cost function, since no error is minimized and therefore the resulting cepstral coefficients, using any subset of the cepstrum cannot be considered optimal. Although, simple spectral subtraction of the model and plant's magnitude spectra may prove to be adequate in some cases, it is a rather crude performance measure for signal modelling and generally only gives a rough fit for a given system.

Theoretically speaking, the RCD algorithm should be able to model the characteristics of a plant to a specified a priori modelling error, by optimizing the lifter length and value of $\gamma$. However, adopting a more pragmatic approach, it can be seen that a cost function must be defined, such that the modelling error is minimized with respect to both the lifter length and value of $\gamma$. This concept will therefore be considered next.

### 4.2 Root Cepstral Cost Function

Consider the frequency domain interpretation of the system identification problem, as shown in figure 4.1.

\[
P(w) \xrightarrow{X(w)} M(w) \xrightarrow{\hat{X}(w)} E(w)
\]

Figure 4.1: System identification in the frequency domain.

Since data records are finite, an approximation of the plant, $P(w)$ is given in Eqn. (4.1), where $X(w)$ and $U(w)$ are short time spectra.

\[
P(w) \approx \frac{X(w)}{U(w)}
\]

(4.1)

Also, applying Eqn. (3.17) to figure 5.1, it can be seen that the model, $M(w)$ is given by

\[
M(w) = \left[ \left( X(w) \right)^\gamma \ast L(w) \right]^{1/\gamma}
\]

(4.2)
where, $L(w)$ is the lifter in the frequency domain and "*" represents a convolution. The error signal, $E(w)$ is therefore simply defined as,

$$E(w) = X(w) - \hat{X}(w)$$

$$= U(w) \left[ P(w) - \left[ \left( X(w) \right) \star L(w) \right]^{1/\gamma} \right]$$

(4.3)

Therefore, the mean square error which will be a minimum for an optimal model, is given by Eqn. (4.4), where $E$ is an expectation operator over the continuous frequency domain.

$$J = \frac{1}{2\pi} \int_{-\pi}^{\pi} E(w) \cdot E^*(w) \, dw = E \left[ E(w) \cdot E^*(w) \right]$$

(4.4)

As with Lim's original method [50], the input is restricted to an impulse function. $U(w)$ is assumed to be spectrally flat for all frequencies of interest and may therefore be considered as a gain term, independent of frequency. Thus, significantly simplifying Eqn. (4.4), as shown below [73].

$$J = U^2 E \left[ \left( P(w) - M(w) \right) \left( P^*(w) - M^*(w) \right) \right]$$

$$J = U^2 E \left[ P(w) P^*(w) - P(w) M^*(w) - M(w) P^*(w) + M(w) M^*(w) \right]$$

$$J = U^2 E \left[ |P(w)|^2 + |M(w)|^2 - |P(w)| e^{j\theta_p} \cdot |M(w)| e^{-j\theta_m} \right.$$

$$- |M(w)| e^{j\theta_m} \cdot |P(w)| e^{-j\theta_p} \left. \right]$$

$$J = U^2 E \left[ |P(w)|^2 + |M(w)|^2 \right.$$

$$- |P(w)| \cdot |M(w)| \left( e^{j(\theta_p - \theta_m)} + e^{-j(\theta_p - \theta_m)} \right) \left. \right]$$

$$J = U^2 E \left[ |P(w)|^2 + |M(w)|^2 - 2|P(w)| \cdot |M(w)| \cos(\theta_p - \theta_m) \right]$$

(4.5)

An expression for the normalized performance function or cost function, $\hat{J}$ may be expressed as,

$$\hat{J} = E \left[ |P(w)|^2 + |M(w)|^2 - 2|P(w)| \cdot |M(w)| \cos(\theta_p - \theta_m) \right]$$

(4.6)

Notice that this cost function depends on both the magnitudes and phases ($\theta_m$ and $\theta_p$) of the model and plant, and gives the ability to discriminate between the number of cepstral coefficients and the associated modelling error, thus, potentially reducing the number of coefficients needed to represent a system. Notice also that there are two variables that need to be optimized, the lifter length and $\gamma$. Differentiating Eqn.
4.2. Root Cepstral Cost Function

(4.6) in the same manner as Eqn. (2.39) is not possible since there are no linear model coefficients to optimize. Therefore, the design of an optimization scheme is not obvious, since as demonstrated in chapter 5, the shape of the cost function is entirely dependent upon the type of data used.

4.2.1 Properties of the Performance Function

1. It should be noted that Eqn. (4.6) is particularly useful for system identification, since a modelling error or acceptable error (AE) can now be specified and therefore allows the designer to model a system to any accuracy required. For example, a modelling error of 1% may be acceptable for some applications, whereas, 0.01% error may be suitable for others. This error is similar in concept to the modelling errors of Yule-Walker and Wiener-Hopf, as discussed in chapter 2, but is applied in the frequency domain. Therefore, an iterative technique may be designed to automatically find a suitable lifter length and value of γ for a specified acceptable modelling error.

2. Notice that in the definition of Eqn. (4.6) it is assumed that the input to the system is a Kronecker delta function. Other types of input, such as white noise would complicate matters significantly as data would fill the whole cepstral domain regardless of the value of γ. Secondly, each white noise innovation or input sample would produce a mini impulse response, resulting in many impulse responses colliding into each other, therefore the extraction of a single impulse response would be infeasible using this method. Therefore, Eqn. (4.6) is limited to the analysis of a single impulse response or multiple impulse responses that are well separated.

Although cumulants may overcome this problem, the Bicepstrum in comparison, uses log cepstral coefficients in order to re-construct the plant's impulse response, and is therefore an unsuitable choice for a generalized model.

3. There is no linear model and therefore no need to use model order identification techniques to determine the number of poles and zeros.

4. No a priori knowledge of the structure (i.e., mixed, all-pole or all-zero) of the
5. No stability problems are present in the optimized model, since the method is non-parametric.

4.2.2 Real World Considerations

Realistically speaking, there are no such things as "poles" and "zeros" in the real world. Theoretical analysis requires this mathematical abstraction in order to implement a linear model, and hence assign maxima and minima in order to mimic the characteristics of a plant's spectrum. Since most plants or systems have some sort of non-linearity to their characteristics, whether this is due to noise or an inherent feature, it raises the question of why use a linear model comprised of poles and zeros to model a plant? Recall the performance of the LPC AR model, for the speech coding example of figure 3.3. It can be seen that a LPC model is only capable of modelling the spectral resonances or formants of the speech waveform, but not the spectral valleys. Also recall that the performance of other time domain modelling algorithms, such as Padé and Prony are heavily dependent upon selection of the model order (i.e., knowing how many poles and zeros are present).

As mentioned in section 2.11, a rule of thumb used in speech coding is the assignment of a conjugate pole pair per formant. Although this assumption produces satisfactory results for speech, it is not necessarily valid or acceptable for other types of signals. Therefore if pole/zero models are to be used, it would be prudent to have a dynamic method of assigning the number of poles and zeros for a given modelling problem or even adaptively switching between an all-pole/all-zero/pole-zero model for a given modelling problem.

The root cepstrum provides a possible solution to this problem, for the following reasons: Firstly, the signal is reconstructed rather than analyzed, circumventing the need for model order selection and overcoming any model stability issues associated with time domain methods. The use of $\gamma$ allows dynamic assignment of either an all-pole, all-zero or pole-zero (mixed) type cepstral model to the modelling of the data sequence. Therefore, knowledge of the poles and zeros is no longer relevant, as there is no linear model! Finally, the order of the cepstral model can be set by adjusting the lifter length...
such that the modelling error requirement is satisfied. One obvious problem is how to optimize $\gamma$ and the lifter length. This indeed is a difficult question for which there is no simple solution, since both variables are dependent upon one another.

### 4.3 RCD Algorithm Optimization

As demonstrated in chapter 5, due to the non-parametric complexity of the proposed method, an exhaustive search of the performance surface is required in order to find the optimal solution. As discussed in the previous section, there are two variables that need to be optimized - the lifter length and $\gamma$. At first glance, it would seem almost impossible to design an simple and efficient optimization routine, since both variables are dependent upon one another. However, one possible solution is to hold the lifter length constant and vary $\gamma$, such that the AE requirement is satisfied. In order to implement such a method, a performance table, shown below in table 4.1 is proposed. Note that if the AE has not been satisfied, for any value of $\gamma$, then

<table>
<thead>
<tr>
<th>MMSE</th>
<th>\begin{align*} &amp;\text{minimum mean squared error for the present lifter length} \ \end{align*}</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPT $\gamma$</td>
<td>\begin{align*} &amp;\text{optimal value of } \gamma \text{ for the present lifter length} \ \end{align*}</td>
</tr>
<tr>
<td>LIFTER</td>
<td>\begin{align*} &amp;\text{present lifter length} \ \end{align*}</td>
</tr>
<tr>
<td>AE flag</td>
<td>\begin{align*} &amp;\text{Acceptable error flag: } \begin{align*} &amp;AE = 1 \text{ when } MMSE \leq AE \ &amp;AE = 0 \text{ when } MMSE &gt; AE \end{align*} \ \end{align*}</td>
</tr>
</tbody>
</table>

**Table 4.1: RCD algorithm performance table.**

the lifter length must be increased as more information is required. This requirement complicates the algorithm significantly, since reducing the number of coefficients is paramount. Therefore, a method of reducing the lifter length to the AE threshold point, such that the error is just satisfied must also be incorporated into the design.

#### 4.3.1 RCD Algorithm: Part A

The RCD optimization algorithm is split up into two parts, part A is the main part of the algorithm, concerned with adjusting $\gamma$, calculation of the cepstral coefficients and $\tilde{J}$, as shown in figure 4.2. Part B is dedicated to setting the lifter length and ensuring
Chapter 4. Theoretical Development

Figure 4.2: RCD Algorithm optimization (part A).
4.3. RCD Algorithm Optimization

Figure 4.3: RCD Algorithm optimization (part B): Lifter adjustment and error satisfaction.
that the AE requirement is satisfied (see figure 4.3). An explanation of part A is as follows:

1. The algorithm begins by initializing several variables.
   
   (a) $\gamma$ is initialized to 1 and the lifter length is set at 2, in order to minimize the number of coefficients used for the optimal model.
   
   (b) $\alpha$ is used to adjust the lifter length (as described in part B) and "k" corresponds to the iteration number as $\gamma$ is varied in steps of 0.01. NB. That $\gamma = 0$ cannot be computed numerically and is therefore omitted.
   
   (c) An AE (acceptable error) is specified by the designer.

2. Calculate the spectrum of the impulse response, $x(n)$.

3. Calculate the root cepstrum and the model estimate.

4. Calculate $J$ for each iteration or value of $\gamma$.

5. The iteration is repeated until $\gamma = -1$. The algorithm then scans through all values of MSE and finds the minimum (MMSE), at which point part B of the algorithm is invoked.

4.3.2 RCD Algorithm: Part B

The purpose of part B of the algorithm is to efficiently adjust the lifter length such that the AE is satisfied. Part B makes extensive use of table 4.1. Continuing the discussion from Part A, by checking for a difference of 1 between adjacent values of the lifter length and AE flags (to ensure that the lifter length has been reduced to its minimum value), a quick method of finding the optimal lifter length and $\gamma$ can be achieved. Since, a difference of one for both values signifies the threshold for the AE requirement.

In order to expedite the time taken to reach the final solution, the lifter is length is adjusted by the following formula,

$$ \text{LIFTER} = \text{LIFTER} \pm \alpha $$

(4.7)
where $\alpha$ is “step size” and is initially equal to 4, but is dynamically adjusted by the algorithm, as seen in figure 4.3. Adjustments to the value of $\alpha$ are slightly similar to VS LMS algorithm, as discussed in section 2.8.1. Basically, the algorithm begins with a large step size that is reduced in steps of 2 (i.e., 4, 2, 1) according to the proximity of the optimal solution. Note that after assigning a new lifter length, any re-calculation of duplicate values may be avoided by scanning through the entries of the performance table and adjusting the lifter length accordingly. The algorithm will only stop when the AE requirement has been satisfied and the lifter length has been optimized or if a very large lifter length is required.

Notice that the “AE” flag could have been replaced with the operation $\text{MSE} \leq \text{AE}$ and therefore excluded from the performance table.

### 4.4 z-plane Root Cepstral plots

For any particular value of $\gamma$, the root cepstral coefficients represent an estimation of the plant impulse response in the “warped time domain”. The corresponding frequency response can be evaluated by taking the $z$-transform of the sequence of cepstral coefficients, as shown below (where, $m$ defines the lifter length)

$$C(z) = Z\{C_i\} \quad \text{where, } 1 \leq i \leq m$$

it should be clear that the cepstral coefficients can be thought of as a single all-zero polynomial, $C(z)$ in the warped frequency domain. Therefore, it is reasonable to suggest that as $\gamma$ is varied between $\pm 1$, the roots of the polynomial (i.e., the location of the cepstral zeros) may yield some important information about the positions of the plant’s dominant poles and zeros. Therefore, by tracking the movement of these cepstral zeros, loci can be plotted and may be used for model order identification purposes for use with time domain modelling methods. The cepstral zeros should not be confused with the plant zeros, since the cepstral coefficients, for an mixed model, are comprised of a non-linear combination of the plant’s numerator and denominator coefficients. For simple cases, such as an all-zero/all-pole models, when $\gamma = \pm 1$, the cepstral zeros will coincide with the plant’s zeros/poles, as shown in section 4.4.1. The $z$-plane loci
4.4.1 The all-pole model

In order to demonstrate the usefulness of the $z$-plane cepstral plots, consider the following 2nd order all-pole model, consisting of a conjugate pole pair, where $a = 0.81$.

$$P(z) = \frac{G}{1 + az^{-2}}$$  \hspace{1cm} (4.8)

Alternatively, this may be expressed as

$$P(z) = \frac{G}{(1 - a_1z^{-1})(1 - a_2z^{-1})}$$  \hspace{1cm} (4.9)

where $a_1$ and $a_2$ are the two roots of the polynomial. Therefore, by using Eqn. (3.14) it can be seen that when $\gamma = -1$, Eqn. (4.9) may be expressed as a finite series, since dominant “poles” become dominant “zeros”.

$$\left( P(z) \right)^{\gamma=-1} = \left( \frac{(1 - a_1z^{-1})(1 - a_2z^{-1})}{G} \right)$$  \hspace{1cm} (4.10)

Therefore, the resulting cepstral coefficients relating to the 2nd order all-pole model may be summarized as follows, where $G = 1$.

$$\left( P(z) \right)^{\gamma=-1} = 1 - (a_1 + a_2)z^{-1} + a_1a_2z^{-2}$$  \hspace{1cm} (4.11)

Solving the denominator of Eqn. (4.8) yields the roots of the polynomial or the poles, which are $0 \pm 0.9j$ and therefore Eqn. (4.11) simplifies to the following expression

$$\left( P(z) \right)^{\gamma=-1} = 1 + 0.81z^{-2}$$  \hspace{1cm} (4.12)

which is the desired result. Similarly, when $\gamma = 1$, the cepstral coefficients are identical to the impulse response of the system. See below.

$$\frac{X(z)}{U(z)} = \left( \frac{G}{(1 - a_1z^{-1})(1 - a_2z^{-1})} \right)^{\gamma=1} = G(1 - a_1z^{-1})^{-1}(1 - a_2z^{-1})^{-1}$$

$$\frac{X(z)}{U(z)} = G \left( 1 + (a_1 + a_2)z^{-1} + (a_1^2 + a_1a_2 + a_2^2)z^{-2} + \cdots \right)$$  \hspace{1cm} (4.13)
4.4. z-plane Root Cepstral plots

START

INITIALIZE
\( \gamma = 1, \text{LIFTER} = l(\eta) \)

\( X(w) = \mathcal{F}\{x(n)\} \)

\( c(\eta) = \left[ \mathcal{F}^{-1}\{X(w)^\gamma\} \right] \cdot l(\eta) \)

\( C(z) = \mathcal{Z}\{c(\eta)\} \)

Calculate roots of \( C(z) \) and plot on z-plane.

\( \gamma = \gamma - 0.01 \)

\( \gamma = -1 ? \)

No

Yes

STOP

Figure 4.4: z-plane root cepstral loci plot algorithm. NB. That \( \gamma = 0 \) is omitted since it cannot be computed numerically.
Chapter 4. Theoretical Development

Limiting the analysis to 2nd order for analytical convenience,

\[
\frac{X(z)}{U(z)} = G \left( 1 + a^2 z^{-2} \right) \tag{4.14}
\]

Therefore, taking inverse z-transforms of Eqn. (4.14), it can be seen that the resulting difference equation is

\[
x(n) = u(n) - au(n-2) \tag{4.15}
\]

where \( a = |a^2| \) and \( G = 1 \), therefore

\[
\begin{align*}
x(0) &= 1 \\
x(1) &= 0 \\
x(2) &= -a \\
&\vdots
\end{align*}
\tag{4.16}
\]

which is the impulse response of Eqn. (4.8).

Therefore, by finding the roots of the cepstral model and over specifying the order, as depicted in figure 4.4, quite beautiful and yet very useful z-plane cepstral zero plots can be used to analyze the properties of the plant. Figure 4.5 shows the loci for the plant zeros, for a lifter length of 10 and where \( 0.01 < \gamma \leq 1 \). Notice that 2 zeros always sit at the origin in order to model the plant’s zeros, as seen in Eqn. (4.17) and therefore there are a total of 8 loci.

\[
P(z) = \frac{x^2}{z^2 + 0.81} \tag{4.17}
\]

The arrows relate to the trajectory of the loci and the plant’s poles are denoted by “x” at \( \pm 0.9j \). As you can see, the radius of all the cepstral zeros are equal when \( \gamma = 1 \), and as \( \gamma \) is varied in steps of 0.01, the zeros move towards the centre, stopping when \( \gamma = 0.01 \). Note that \( \gamma = 0 \) is omitted for computational reasons.

Figure 4.6 depicts the loci for negative values of \( \gamma \). Upon \( \gamma \) reaching \(-0.01\), notice that the sign change in Eqn. (3.14) causes the cepstral zeros to be linearly rotated, so as to model the effects of the plant’s poles. As \( \gamma \) is decreased, notice that all the diagonal and horizontal loci seem to foldback on themselves and then converge towards the centre, since they are not required to model the plant. The two vertical loci continue to the position of the plant’s poles at \( \gamma = -1 \).
4.4. z-plane Root Cepstral plots

Figure 4.5: z-plane cepstral zero plot for a 2nd order all-pole model for $0.01 \leq \gamma \leq 1$.

Figure 4.6: z-plane cepstral zero plot for a 2nd order all-pole model for $-1 \leq \gamma \leq -0.01$. 
Figure 4.7: z-plane cepstral zero plot for a 2nd order all-pole model for $-1 \leq \gamma \leq 1$.

Finally, figure 4.7 shows the combined effect for both positive and negative values of $\gamma$. Note that these plots may be used not only for model order identification, but also for identification of the position of the plant's poles. Note also, that for all-pole models, $\gamma = -1$ would be sufficient in order to determine the position of the plant’s poles, as seen in figure 4.6.

Cepstral loci radius

As one might expect, the radius of the loci is governed by the value of $\gamma$. The two coefficients of the all-pole model, $a_1$ and $a_2$ may therefore be analyzed as shown in Eqn. (4.18).

$$ P(z) = \frac{1}{(1 - a_1z^{-1})^\gamma \cdot (1 - a_2z^{-1})^\gamma} $$

when $\gamma = 1$ the cepstral model is a truncated version of the plant’s impulse response, and therefore the radius is equal to the roots of Eqn. (4.16). Likewise, when $\gamma = -1$, the cepstral model is finite, comprised only of the plant’s coefficients, yielding two roots at $\pm 0.9j$. For fractional values of $\gamma$ the cepstral model is infinite, but is truncated by
the lifter operation. Revisiting Eqn. (3.14), it can be seen that both terms of Eqn. (4.18) will produce an infinite time series expansion, and therefore by combining these two series expansions a cepstral model is obtained.

### 4.4.2 The pole-zero model

Consider the simple plant transfer function of Eqn. (4.19).

\[
\left( P(z) \right)^\gamma = \left( \frac{1-bz^{-1}}{1-az^{-1}} \right)^\gamma
\]  
(4.19)

It can be seen that regardless of the value of \( \gamma \), the impulse response and therefore the unfettered cepstral model will be of infinite duration. Therefore assigning a value of \( \gamma \) and lifter length becomes harder. Taking inverse \( z \)-transforms of Eqn. (4.19), when \( \gamma = 1 \), it can be seen that the resulting difference equation is

\[
x(n) = u(n) - bu(n-1) + ax(n-1)
\]  
(4.20)

resulting in the following impulse response

\[
x(0) = 1 \\
x(1) = a - b \\
x(2) = a(a - b) \\
\vdots \\
x(n) = a^{n-1}(a - b)
\]

(4.21)

Similarly, when \( \gamma = -1 \),

\[
x(n) = u(n) - au(n-1) + bx(n-1)
\]  
(4.22)

resulting in the following impulse response

\[
x(0) = 1 \\
x(1) = b - a \\
x(2) = b(b - a) \\
\vdots \\
x(n) = b^{n-1}(b - a)
\]

(4.23)
As seen from both impulse responses, it is difficult to extract the coefficient values if the model order is not known. However, figures 4.8 and 4.9 show the identification method for a 6th order mixed plant that consists of 4 poles and 2 zeros in conjugate pairs of radius 0.8. The basic idea is to look for large gaps in the circle of cepstral zeros, in order to find the plant's poles and zeros. Notice that this example has worked particularly well, since the poles and zeros are of equal radius, and in effect they can all be classed as dominant poles and zeros.
4.4. z-plane Root Cepstral plots

Figure 4.8: z-plane cepstral zero plot for a 4 pole, 2 zero mixed model for $\gamma = -1$.

Figure 4.9: z-plane cepstral zero plot for a 4 pole, 2 zero mixed model for $\gamma = 1$. 
4.5 Summary and Conclusions

In this chapter several important ideas have been developed and discussed. Cepstral techniques are non-parametric modelling techniques that approach the system identification problem in the frequency domain. The virtue of these methods are that they require no model order identification techniques and have no model stability problems as no linear model is used in the analysis. All cepstral methods do not discriminate between the modelling of poles and zeros and the complex cepstrum has the virtue of being a reversible process, since it retains phase.

As demonstrated in chapter 3, the root cepstrum may be used to model all-pole and all-zero systems fairly efficiently when $\gamma = \pm 1$. However, for pole-zero systems, the technique requires a performance or cost function to ensure that an optimal solution has been reached. Therefore, a root cepstral MSE performance function for system identification was developed and discussed in section 4.2. The normalized performance cost function, $\tilde{J}$ (Eqn. (4.6)), assumes that the input to the plant is a Kronecker delta function and that both input and output short time spectra records are available. Eqn. (4.6) has several advantages, such as there is no need to specify the number of poles and zeros and no need to specify the model order, as the modelling error can be compared to the AE directly, thus allowing the designer to model a plant to any accuracy required. Also note, that for a small AE, Eqn. (4.6) will automatically set $\gamma = \pm 1$ for an all-pole or all-zero model.

Since the root cepstrum retains phase information, the cost function, quite uniquely has a phase term. Note that the modelling error concept is similar to the Yule-walker and Wiener-Hopf errors as discussed in chapter 2, but is applied in the frequency domain. In section 4.3 details of a suitable optimization algorithm were presented and discussed.

Section 4.4 developed the concept of root cepstral z-plane plots for model order identification for use with time domain modelling methods. Three examples were discussed: an all-pole model, an all-zero model and a pole-zero model and theory developed. It was shown that by setting $\gamma = \pm 1$, positions of the plant’s poles and zeros could be determined by visual inspection of the resulting z-plane plot.
Chapter 5

Simulation Results

5.1 Introduction

This chapter is broken up into several sections assessing the performance of the RCD algorithm. The first section presents detailed results of the RCD algorithm's performance in noise for all-pole, all-zero and pole-zero models respectively. The discussion then continues to the derivation of several important relationships with two time domain techniques. Finally, a comprehensive analysis and discussion of z-plane root cepstral plots and other developments are presented at the end, including a special cost function for modelling speech.

5.2 RCD Algorithm Performance

5.2.1 The all-pole model

Consider the fourth order all-pole model, consisting of two conjugate poles pairs of radius 0.8, as shown in Eqn. (5.1).

\[ P(z) = \frac{1}{1 - 0.0444z^{-1} + 0.8398z^{-2} - 0.0284z^{-3} + 0.4096z^{-4}} \]  \hspace{1cm} (5.1)

Initializing the RCD algorithm with an acceptable error (AE) of 0.1% and providing the algorithm with no other a priori information, the optimization algorithm, as described
in section 4.3 produced an optimal solution after only 3 adjustments to the lifter length - see table 5.1. The corresponding optimal magnitude and phase are shown in the upper two plots of figure 5.1, where red and blue denote the model and plant respectively. Notice that for a lifter=5 (i.e., lifter length=5), $\gamma = -1$ the error is equal to $1.6 \times 10^{-17}$ and the composite magnitude and phase response coincides with the plant's magnitude and phase response.

<table>
<thead>
<tr>
<th>MMSE</th>
<th>0.8605</th>
<th>$2.3852 \times 10^{-18}$</th>
<th>0.0786</th>
<th>$1.6263 \times 10^{-17}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal $\gamma$</td>
<td>1</td>
<td>-1</td>
<td>0.34</td>
<td>-1</td>
</tr>
<tr>
<td>lifter length</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.1: RCD optimization performance for Eqn. (5.1).

The bottom left plot of figure 5.1 shows the reconstructed signal ($p$-hat) Vs. the plant impulse response, $p(n)$ and therefore demonstrates that the signal has been reconstructed perfectly. The associated modelling error for a lifter length of 5 is shown on the bottom right of figure 5.1, notice how the minima is at $\gamma = -1$.

![Figure 5.1: The optimal magnitude and phase for Eqn. (5.1).](image-url)
Figure 5.2 depicts how the optimal value of $\gamma (\gamma_{opt})$ changes as the lifter is increased. Notice that for a lifter $\geq 5$, $\gamma_{opt}$ remains fixed at -1. This is attributed to the dramatic change in MMSE, as shown in figure 5.3, as the lifter changes from 4 to 5, since the number of cepstral coefficients is equal to the plant's order - see section 4.4.1 for a simplified mathematical analysis.

A 3D plot depicts the performance surface for this example, as shown in figure 5.4. The plot can be thought of as a more elaborate combination of figures 5.2 and 5.3 and
Figure 5.4: 3D plot for the 4th order all-pole plant of Eqn. (5.1).

is best analyzed by considering the plot evolving from left to right (i.e., as the lifter length is increased). Notice the sudden change in MSE when $\gamma = -1$, for a lifter $\geq 5$, since the plant’s (model) order has been satisfied and that for other values of $\gamma$ the performance surface is fairly flat. Therefore, figure 5.3 may be used as an alternative to the model order identification techniques, such as MDL and AIC as discussed in chapter 2.11.

Performance in Noise

For an AE=0.1% in the presence of band limited Gaussian white noise, it can be seen that even for a modest variance, such as $\sigma^2 = 0.01$ or a high signal-to-noise ratio (SNR) the modelling performance has been affected, as seen in figure 5.5. Examining the entries of table 5.2, it can be seen that for a lifter of 5 the MMSE is considerably larger, and the optimal value of $\gamma$ is now equal to -0.99.
5.2. RCD Algorithm Performance

<table>
<thead>
<tr>
<th>MMSE</th>
<th>0.8605</th>
<th>$3.034 	imes 10^{-4}$</th>
<th>0.0783</th>
<th>$3.789 	imes 10^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal $\gamma$</td>
<td>1</td>
<td>-0.99</td>
<td>0.35</td>
<td>-0.99</td>
</tr>
<tr>
<td>optimal lifter</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.2: RCD optimization performance for Eqn. (5.1), $\sigma_v^2 = 0.01$.

A comparison between LPC and RCD coefficients is given in Table 5.3. Note that the cepstral coefficients are not directly equivalent to the LPC coefficients, when $\gamma = -0.99$ since the cepstral model is theoretically infinite. Nevertheless, notice the similarity in the coefficient values and therefore as seen in Figure 5.6 for a SNR=23.3dB, the modelling performance of LPC (4th order) and the RCD is fairly comparable.

<table>
<thead>
<tr>
<th>liftered cepstral coefficients</th>
<th>1.0000</th>
<th>-0.0432</th>
<th>0.8310</th>
<th>-0.0233</th>
<th>0.4054</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC coefficients</td>
<td>1.0000</td>
<td>-0.0417</td>
<td>0.8355</td>
<td>-0.0241</td>
<td>0.4061</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison between LPC and RCD coefficients for Eqn. (5.1).
Figure 5.6: Spectral envelopes comparing RCD (lifter=5) to 4th order LPC, $\sigma_r^2 = 0.01$.

For a SNR=11.9dB (where, $\sigma_r^2 = 0.18$) and AE=1% the optimal value of $\gamma$ is 0.11, as shown in figure 5.7. Note that this value of $\gamma$ does not invert $P(z)$ and will dampen the higher order cepstral coefficients, comprised of signal + noise, by compressing the cepstral data into the lower order coefficients. As a consequence, the RCD is more robust to the effects of additive band-limited Gaussian white noise than 4th order LPC. This indeed is an interesting result, since LPC autocorrelates the data, it is therefore assumed that the effects of the noise are significantly minimized in the higher order lags. Examining the optimization, shown in table 5.4, it can be seen that the algorithm attempts to reduce the number of coefficients by adjusting the lifter length from 10 to 8 and then to 9. However, the first adjustment is to no avail, as the value does not satisfies the AE requirement. Finally, the algorithm stops for a lifter length of 9.

<table>
<thead>
<tr>
<th>MMSE</th>
<th>0.8624</th>
<th>0.0528</th>
<th>0.0093</th>
<th>0.0174</th>
<th>0.0098</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal $\gamma$</td>
<td>1</td>
<td>-0.72</td>
<td>0.2</td>
<td>-0.42</td>
<td>0.11</td>
</tr>
<tr>
<td>optimal lifter</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 5.4: RCD optimization performance for Eqn. (5.1), $\sigma_r^2 = 0.18$. 
The RCD magnitude with a SNR=11.9dB is once again compared to LPC, as seen in figure 5.8. Notice that by increasing the number of LPC coefficients (8th order) - see figure 5.9 that there is only a slight improvement in performance over the 4th order response. However, notice how the 4th order LPC spectral envelope formants are centred at the correct resonant frequencies of \( P(z) \). Comparing this to the 8th order LPC and RCD spectral envelopes, it can be seen that both methods are slightly off centre, when estimating the second formant of \( P(z) \), but the RCD spectral envelope is spectrally closest to \( P(z) \).

The poor performance of LPC in noise is attributed to the noise variance modifying the trace of the autocorrelation matrix, \( R \). As an example, consider the signal \( x(n) = d(n) + v(n) \), where \( d(n) \) is the desired signal, \( v(n) \) is zero-mean Gaussian white noise and \( d(n) \) and \( v(n) \) are uncorrelated with each other. Applying \( x(n) \) to Eqn. (2.46), it can be seen that the Yule-Walker equations of Eqn. (2.49) may be summarized as,

\[
[R_d + \sigma^2_v I]w = -r_{dx}
\]  

Note that as \( \sigma^2_v \to 0 \), the coefficients approach the noise free case [31, 88].

Figure 5.10 compares the spectral envelopes of LPC, RCD and the log cepstrum. The complex log cepstrum, as expected has the worst modelling performance, since the log operator has “amplified” the samples of noise. The corresponding 3D plot, shown in figure 5.11, shows the effect of the noise with a SNR = 11.9dB, where the black line denotes the optimal trajectory. Comparing figure 5.11 to figure 5.4 notice how the noise has altered the shape of the performance surface and how it is now quite difficult to determine the plant’s order.

Finally, in order to conclude this section for all-pole models, the RCD was applied to a 4th order all-pole model, consisting of two conjugate pole pairs of different radii, as seen in figure 5.12. As expected, the RCD has modelled the magnitude and phase perfectly with 5 coefficients, as shown in figure 5.13.
Chapter 5. Simulation Results

\[ \gamma = 0.11 \]

\[ \text{Present Lifter} = 9 \]

\[ \text{Frequency} \]

\[ \text{P-hat Vs p} \]

\[ \text{Error (Lifter = 9)} \]

\[ \text{MSE} \]

\[ \text{Amplitude} \]

\[ \text{Phase (red)} \]

\[ \text{Log magnitude} \]

\[ \text{Frequency} \]

\[ n \]

Figure 5.7: The optimal solution. Model (red), plant (blue) and \( \sigma_n^2 = 0.18 \).

\[ \text{Figure 5.8: Spectral envelopes comparing RCD (lifter=9) and 4th order LPC, } \sigma_n^2 = 0.18. \]
5.2. RCD Algorithm Performance

Figure 5.9: Spectral envelopes comparing RCD (lifter=9), LPC (4th and 8th order), $\sigma_v^2 = 0.18$.

Figure 5.10: Spectral envelopes comparing RCD (lifter=9), LPC (4th and 8th order) and the complex log cepstrum (lifter=9), $\sigma_v^2 = 0.18$. 

Figure 5.11: 3D performance surface, $\sigma_e^2 = 0.18$.

Figure 5.12: $z$-plane plot for a 4th order all-pole.
5.2. RCD Algorithm Performance

Figure 5.13: RCD optimization performance for all-pole model consisting of two conjugate pole pairs of different radii.

5.2.2 The all-zero model

Consider the fourth order all-zero model, consisting of two conjugate poles pairs of radius 0.8, as shown in Eqn. (5.3).

\[ P(z) = 1 + 1.4165z^{-1} + 1.2389z^{-2} + 0.9066z^{-3} + 0.4096z^{-4} \] (5.3)

The corresponding optimal magnitude and phase spectra are shown in the upper two plots of figure 5.14. Notice that for a lifter=5, \( \gamma = 1 \) the error is approximately equal to \( 9 \times 10^{-17} \) and the composite magnitude and phase response coincides with the plant’s magnitude and phase response.

<table>
<thead>
<tr>
<th>MMSE</th>
<th>0.0751</th>
<th>(-9.8012 \times 10^{-17})</th>
<th>0.0141</th>
<th>(-9.0206 \times 10^{-17})</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{optimal } \gamma</td>
<td>-0.17</td>
<td>1</td>
<td>0.61</td>
<td>1</td>
</tr>
<tr>
<td>\text{lifter length}</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.5: RCD optimization performance for Eqn. (5.3).
Chapter 5. Simulation Results

The 3D-plot is shown figure 5.15. Just as with the all-pole model there is a sudden change in MSE when $\gamma = 1$, for a lifter $\geq 5$, since the plant's (model) order has been satisfied. Notice that unlike the all-pole performance surface of figure 5.4, the all-zero performance surface is much more rugged, containing both peaks and valleys especially for the lower order lifter values.

RCD and its relationship to the FIR Wiener filter

A special relationship can be formulated between the RCD and the FIR Wiener filter when modelling an impulse response. Recall Eqn. (2.42) and the block diagram of figure 2.8 as discussed in chapter 2. Notice that if the reference input to the Wiener filter is a kronecker delta function, then the autocorrelation matrix, $R$, simplifies to the identity matrix and therefore the Wiener-Hopf equations for an impulse response may be expressed as,

$$Iw = r_{xu}$$
5.2. RCD Algorithm Performance

Therefore, Eqn. (5.4) states that all $q$ coefficients of the weight vector, $w$, corresponding to the Wiener solution are equal to the first $q$ values of the cross correlation vector, $r_{xu}$. As a consequence, when $\gamma = 1$, the RCD algorithm is identical to solving Eqn. (5.4). It can therefore be concluded that the RCD, when $\gamma = 1$ is identical to the performance of an identical order FIR Wiener filter. Finally, it is assumed that the impulse response starts from time index zero. However, if this is not the case, then $r_{xu}$ must be extended so that it captures all of the relevant data.

Figure 5.16 shows the modelling performance for an AE=1% in the presence of additive band limited Gaussian white noise, where $\sigma^2 = 0.27$ and SNR=15dB. Note that as $\gamma = 1$, the modelling performance is identical to an equivalent order FIR Wiener filter.
5.2.3 The pole-zero model

The pole-zero model is perhaps the most difficult modelling scenario, since it is impossible to make the cepstral model truly finite, as discussed in the previous chapters. However, as demonstrated in this section, by varying $\gamma$ and adjusting the lifter length, satisfactory modelling performance for mixed models can be achieved. Once again, consider the mixed model consisting of four poles and four zeros in conjugate pairs of radius 0.8, as shown in Eqn. (5.5).

$$P(z) = \frac{1 + 1.41652z^{-1} + 1.23892z^{-2} + 0.90662z^{-3} + 0.40962z^{-4}}{1 - 0.0294z^{-1} + 0.0887z^{-2} - 0.0187z^{-3} + 0.4097z^{-4}}$$  (5.5)

<table>
<thead>
<tr>
<th>MMSE</th>
<th>1.1736</th>
<th>0.0925</th>
<th>0.0057</th>
<th>$2.429 \times 10^{-4}$</th>
<th>0.0046</th>
<th>$8.5932 \times 10^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal $\gamma$</td>
<td>0.01</td>
<td>0.38</td>
<td>-0.01</td>
<td>-0.36</td>
<td>-0.48</td>
<td>0.08</td>
</tr>
<tr>
<td>lifter length</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>14</td>
<td>12</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 5.6: RCD optimization performance for Eqn. (5.3).
5.2. RCD Algorithm Performance

The corresponding optimal magnitude and phase spectra are shown in the upper two plots of figure 5.17. For an AE=0.1%, a lifter=13 (i.e., 13 coefficients) is required. Clearly this is much more than the order of model, which has a total of 10 coefficients. However, compare this result to an equivalent order FIR Wiener filter, as shown in figure 5.18. Upon examining the MMSE, it can be seen that for 13 coefficients, the equivalent of a 12th order Wiener filter ($\gamma = 1$), the MMSE is $2.76 \times 10^{-2}$, comparing this result to the RCD, it can be seen that the MMSE is $8.59 \times 10^{-4}$, which is approximately a 32 fold improvement. The superior modelling performance of the RCD is attributed to the mixed (pole-zero) cepstral model, unlike the Wiener filter, which is constrained to an all-zero model. Analyzing the shape of the MSE curve, shown on the bottom right of figure 5.17, it can be seen that there is a peak and a trough, therefore re-enforcing the need for an exhaustive search of the performance surface.

Notice that it takes at least a 19th order FIR Wiener filter to match to the model error performance of the RCD, as shown in figure 5.19, therefore requiring an additional 7 coefficients. The Wiener filter also requires the normalization of impulse response prior to the filtering operation in order to remove the excitation and therefore estimate $P(z)$. Recall that the RCD automatically scales the excitation by $\gamma$ and then uses this value to normalize the rest of the cepstral data (see Appendix A for further details).

Finally, the 3D performance surface is shown in figure 5.20. Notice the mountain type terrain, especially at the lower order lifter values and how the surface, unlike the previous two examples does not reveal any obvious clues to the order of the model. Also notice that the optimal trajectory is a zigzag, thus relying upon an exhaustive search of the surface.
Chapter 5. Simulation Results

\( \gamma = 0.08 \)

Present Lifter = 13

Figure 5.17: The optimal magnitude and phase for Eqn. (5.5).

\[ \beta - \frac{1}{5} \]

\[ \beta - \frac{1}{10} \]

\[ \beta - \frac{1}{15} \]

\[ \beta - \frac{1}{20} \]

Figure 5.18: 12th order FIR Wiener filter magnitude and phase response (shown in red) for modelling Eqn. (5.5), shown in blue.
5.2. RCD Algorithm Performance

Figure 5.19: 19th order FIR Wiener filter magnitude and phase response for modelling Eqn. (5.5), modelling error equivalent to the RCD (lifter length=13, $\gamma = 0.08$).

Figure 5.20: 3D performance surface for modelling Eqn. (5.5).
5.2.4 Compression

The issue of cepstral model compression for a pole-zero model has received little attention in this thesis and will therefore be discussed in this section. Analyzing the results of figure 5.21 it can be seen that for a 24th order pole-zero model, comprised of 12 poles and 12 zeros (see figure 5.22 for the distribution) the number of cepstral coefficients is far less than the order of the model! Therefore, a compression factor (CF) of 2.36, corresponding to (26 plant coefficients)/(9 cepstral coefficients + \(\gamma\) + lifter length) can be obtained for an AE=0.1%. However, a compression factor of this magnitude is not generally common, as seen from the other pole-zero models discussed herein and therefore the compression factor will depend upon the type of data used (i.e., the pole-zero distribution).

5.2.5 The RCD and its relationship to the Padé Approximation

A special noiseless relationship exists between the RCD and the Padé Approximation for when \(\gamma = \pm 1\). In order to demonstrate this, consider a mixed model of the form,

\[
P(z) = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}
\]  

(5.6)

with the difference equation

\[
x(n) = u(n) + b_1 u(n - 1) + b_2 u(n - 2) - a_1 x(n - 1) - a_2 x(n - 2)
\]  

(5.7)

when \(\gamma = -1\), the coefficients obtained by the ROD are identical to inverting \(P(z)\). Therefore, Eqn. (5.7) may be re-written as,

\[
x(n) = u(n) + a_1 u(n - 1) + a_2 u(n - 2) - b_1 x(n - 1) - b_2 x(n - 2)
\]  

(5.8)

with the following impulse response,

\[
x(0) = 1
\]  

(5.9)

\[
x(1) = a_1 - b_1
\]  

(5.10)

\[
x(2) = a_2 - b_2 - b_1 a_1 + b_1^2
\]  

(5.11)
5.2. RCD Algorithm Performance

\[ \gamma = -0.42 \]

Present Lifter = 9

\[ \theta \]

\[ p \hat{=} \theta \]

\[ \text{Amplitude} \]

\[ \text{Error (Lifter = 9)} \]

\[ \text{MSE} \]

Figure 5.21: Compression of a high order pole-zero (12 poles, 12 zeros) model.

\[ \text{z-plane plot for a 12 pole, 12 zero model.} \]
It can be seen that when the Padé equations of Eqn. (2.6) are constrained to an all-pole model (i.e., q = 0, therefore making no assumptions about the zeros) and applied to the modelling of \( x(n) \), the results are identical to those obtained by the RCD, when \( \gamma = -1 \), as demonstrated below.

\[
\begin{bmatrix}
  x(0) & 0 \\
  x(1) & x(0)
\end{bmatrix}
\times
\begin{bmatrix}
  w(1) \\
  w(2)
\end{bmatrix}
= -
\begin{bmatrix}
  x(1) \\
  x(2)
\end{bmatrix}
\tag{5.12}
\]

Solving Eqn. (5.12) yields the solution for \( w_1 \) and \( w_2 \).

\[
w_1 = \frac{x(1)}{x(0)} = a_1 - b_1
\]

\[
w_2 = \frac{x(1)^2 - x(0)x(2)}{x(0)^2} = a_2 - b_2 - b_1a_1 + b_1^2
\]

Notice that \( w_1 \) and \( w_2 \) are identical to Eqn. (5.10) and Eqn. (5.11). Therefore, the first three cepstral coefficients are identical to the first three Padé coefficients. This relationship should not be surprising, since the Padé approximation models \( x(n) \) without error for the first \( p + q + 1 \) values and as \( q \) is equal to zero, \( p + q + 1 = 3 \). Therefore, it can be concluded that when \( q = 0 \), the Padé equations are identical to the RCD when \( \gamma = -1 \). Likewise, when \( \gamma = 1 \) and \( p = 0 \) the solutions are also identical.

### 5.3 RCD performance when modelling magnitude and phase independently

Since Fourier magnitude and phase are orthogonal, they may be separated and analyzed independently to one another. In order to develop this concept, consider the following definition.

\[
M(w) = |M(w)| \cdot e^{j\theta}
\tag{5.15}
\]

Therefore, separating the magnitude and phase components of Eqn. (5.15) and applying this concept to Eqn. (4.2), it can be realized that,

\[
M(w) = \left[ |X(w)|^{\gamma_{mag}} \ast L(w) \right]^{1/\gamma_{mag}} \times \left[ e^{j\theta_{phase}} \ast L(w) \right]^{1/\gamma_{phase}}
\tag{5.16}
\]

where, \( L(w) \) is a sinc function.
5.3. RCD performance when modelling magnitude and phase independently

Figure 5.23: optimal $\gamma$ for separated magnitude and phase for modelling Eqn. (5.5).

Figure 5.24: MMSE Vs. lifter length for modelling Eqn. (5.5).
where, $\gamma_{mag}$ and $\gamma_{phase}$ are independent to each other. Modifying the standard RCD algorithm, such that Eqn. (5.16) may be implemented, the new algorithm was used to model the pole-zero model of Eqn. (5.5). Analyzing figure 5.23 it can be seen that the optimal values of $\gamma$ are in fact different for the magnitude and phase, but coincide for certain values. This is especially noticeable for a lifter (lifter length) of 13, where $\gamma_{mag} = \gamma_{phase}$, which was the optimal result for an AE=0.1$. Figure 5.24 shows the corresponding MMSE Vs. the lifter length, where the dashed black line denotes the AE threshold value. Notice how the phase has satisfied the AE requirement when the lifter=9, but the magnitude requires another 4 values (lifter=13).

Similarly, for 4th order all-pole and all-zero models, figures 5.25, 5.26, 5.27 and 5.28 show once again that the optimal values of $\gamma$ are different for the magnitude and phase, but converge for lifter lengths $\geq 5$.

![Figure 5.25: MMSE Vs. lifter length for modelling Eqn. (5.1).](image)

Although these results seem promising and suggest that magnitude and phase should be modelled independently, this technique requires many more coefficients than the integrated method of section 4.3 and therefore may only be suitable for cases when
5.3. RCD performance when modelling magnitude and phase independently

Figure 5.26: optimal $\gamma$ for modelling Eqn. (5.1).

Figure 5.27: MMSE Vs. lifter length for modelling Eqn. (5.3).
compression is not important. Recall that magnitude or phase cepstra requires liftering either side of the origin, therefore requiring twice the amount of coefficients. Although, as seen from figure 5.26 and figure 5.28, the two solutions have converged for lifter lengths $\geq 5$, thus demonstrating that modelling magnitude and phase independently is not necessary for all-pole or all-zero models.

5.4 z-plane Cepstral plots

Continuing the brief investigation of section 4.4, the corresponding z-plane cepstral plot of the optimal solution ($\gamma = 0.08$) for the pole-zero model example is shown in figure 5.29. This reveals no obvious clues to the order of the model or position of the poles and zeros. However, upon examining figure 5.30 it can be seen that the order of the model may be determined by looking at the colour of all the uncrossed loci. Recall that for modelling zeros, the loci manifest themselves as dark colours, whereas for modelling poles, the loci manifest themselves as light colours.
5.4. \textit{z-plane Cepstral plots}

Figure 5.29: \textit{z-plane plot for modelling Eqn. (5.5) $\gamma = 0.08$, lifter length=13.}

Figure 5.30: \textit{Complete \textit{z-plane} plot for modelling Eqn. (5.5), $\gamma = 0.08$, lifter length=13.}
Therefore, counting anti-clockwise it can be seen that there are twelve uncrossed loci, corresponding to two poles in the first and fourth quadrant, two zeros bordering the first and second quadrant and two poles and two zeros in the second and third quadrant. Notice how the proximity of the two zeros in the second and third quadrant has distorted the loci. The "crossed loci" phenomena can be seen more explicitly in figure 5.31, where the "zeros loci" arc outwards, therefore making way for one of the "pole loci" to pass through the centre and hence cross over the actual position of the plant's pole. Some general observations may be summarized as follows:

1. Dominant poles/zeros manifest themselves as shown in figure 5.31.

2. Pole/zero models with different radii will generally perform badly, since the more dominant poles/zeros will, understandably, dominate the response.

3. As discussed in section 4.4, the initial cepstral zeros radius will be equal to the radius of the dominant poles or zeros. Since all previous examples have been comprised of equal radii poles/zero models, this issue has not received any attention in this thesis, and therefore will be considered next.

![Figure 5.31: Model order identification: crossed loci phenomena.](image)
Figure 5.32 shows the performance of a model consisting of two conjugate pole pairs of different radii and a conjugate zero pair also of different radii. As expected, the RCD has modelled the two dominant poles, but has neglected the other two poles and has also modelled the two zeros, since they are the “dominant zeros”, for positive $\gamma$.

![Diagram of z-plane cepstral zeros plot for a pole-zero sequence of unequal radius.](image)

It is informative to note that when $\gamma = -1$, the roots of an equivalent order optimized LPC model (see Eqn. (2.54)) will produce similar results to the RCD, as shown in figures 5.33 and 5.34. Likewise, when $\gamma = 1$ the roots of the weight vector, $w$ corresponding to the Wiener solution (see Eqn. (2.42)) will also produce an identical result. However, rather than use two algorithms for model order identification, the RCD can be used to get the best of all worlds. Finally, it should be noted that a marginal improvement in performance may be achieved by using a Hamming window as the lifter.
Figure 5.33: Model order identification using RCD, lifter length=13.

Figure 5.34: Model order identification using 12th order LPC.
5.4.1 Performance in Noise

As one might expect, in the presence of noise the loci trajectory may appear wobbly and any estimate of pole/zero positions would be smeared. A comparison between LPC and RCD for identifying the all-pole model of Eqn. (5.1) in the presence of additive bandlimited Gaussian white noise, where \( \sigma^2 = 1 \) is shown in figure 5.35 and figure 5.36. As you can see, the performance of the RCD has been severely affected by the noise, since the spectrum has been inverted, thus amplifying the noise samples. As a consequence, it is difficult to determine the positions of the plant’s poles from figure 5.35. LPC fares much better for two reasons, firstly it computes the autocorrelation of the signal+noise and will reduce the effect of the noise. Secondly, the signal is not inverted, it is analyzed in the time domain.

Analyzing figures 5.37 and 5.38, it can be seen that the model order identification technique has completely failed for the RCD, whereas the positions of the plant’s zeros are still discernable for LPC. However, analyzing figure 5.37 further, notice how there are several cepstral zeros outside the unit circle. Applying the rules of spectral factorization [31], stating that each pole or zero will have a matching pole or zero at the conjugate reciprocal location, it would reasonably be suggested that if these zeros were moved to their conjugate reciprocal locations, the plot would be more useable. NB. Moving zeros around in this fashion would alter the phase spectrum, but not the magnitude spectrum.
Figure 5.35: Pole location using RCD, lifter length=13, with additive noise, $\sigma_v^2 = 1$.

Figure 5.36: Pole location using 12th order LPC with additive noise, $\sigma_v^2 = 1$. 
5.4. *z*-plane Cepstral plots

Figure 5.37: Model order identification using RCD, lifter length=13, with additive noise, $\sigma_v^2 = 1$.

Figure 5.38: Model order identification using 12th order LPC with additive noise, $\sigma_v^2 = 1$. 
5.5 Modelling maximum phase and mixed phase systems

Until now all models and results have assumed that the plant is minimum phase (i.e., all the poles and zeros are inside the unit circle). However, this assumption is not realistic for most systems and therefore a discussion assessing performance of maximum and mixed phase models is presented.

In order to model maximum phase systems the definition of the lifter must be altered such that all of the relevant data is captured (see section 3.3.1). Analyzing figure 5.39, it can be seen that varying the value of $\gamma$ splits the data up into the left and right regions of the cepstral domain respectively. However, when $\gamma = \pm 1$ the cepstral data is bounded to either the left or right regions. As one might expect, the modelling performance is fairly limited unless the lifter function has been appropriately defined. However, when $\gamma = 1$, the impulse response is aligned left and may be successfully modelled, although $\gamma = 1$ is no compression.

Continuing the discussion to mixed phase (i.e., poles and zeros inside and outside the unit circle) models, it can be seen in figure 5.40 that for a 4th order pole-zero model, consisting of a conjugate pole pair inside the unit circle and a conjugate zero pair outside the unit circle, when $\gamma = -1$ a symmetrical lifter function is required. As expected, for fractional values of $\gamma$, the cepstral data is spread out across the whole cepstral domain.

Therefore an important conclusion can be made from these findings. The modelling of maximum/mixed phase systems requires re-defining the lifter function and liftering either side of the origin. This leads to many more coefficients and generally does not yield satisfactory modelling performance.
5.5. Modelling maximum phase and mixed phase systems

Figure 5.39: 2nd order all-zero maximum phase model.

Figure 5.40: Mixed phase model pole-zero model.
5.6 Matlab's phase unwrapping algorithm and the Differential Cepstrum

As briefly discussed in chapter 3, the alignment of the impulse response is paramount for satisfactory performance. Until now, all examples have been limited to the analysis of a single impulse response that is aligned left (starting from time index zero). However, this assumption is rather idyllic and therefore not representative of real world situations, since most data records will generally have a gap before the start of the data. In order to ensure that the RCD performs correctly, any phase shifts due to signal misalignment must be removed by an appropriate pre-filtering algorithm.

![Signal with multiple excitations](image)

Figure 5.41: Signal with multiple excitations.

For a single excitation response, this problem may be overcome in several ways. For example, the standard phase unwrapping algorithm may be modified, such that the phase corresponding to integer lag may be calculated and removed. The integer lag may then be re-introduced at the synthesis stage. The following piece of Matlab code [89] achieves this, by subtracting a diagonal line from the unwrapped phase (phase.X) as shown overleaf.
5.6. Matlab's phase unwrapping algorithm and the Differential Cepstrum

```matlab
phase_X = unwrap(angle(X));
n = length(phase_X);
nh = fix((n+1)/2);
nd = round(phase_X(nh+1)/pi);
phase_X(:) = phase_X(:)' - (pi*nd*(0:(n-l))/nh);
```

However, the differential cepstrum may be utilized by differentiating the phase cepstra, as discussed in section 3.7. Analyzing figure 5.42 it can be seen both methods have failed to extract the true phase. Notice that the differential phase cepstra has many discontinuities that appear oscillatory due to an insufficient DFT frequency step size. This subject will be treated in some detail in chapter 6.

Therefore, both techniques are limited to the analysis of a single impulse response, and as a consequence neither method is suitable for analyzing signals with multiple excitations, such as voiced speech. Windowing the signal of figure 5.41 with a Hamming or Blackman window may produce some satisfactory results. However, for voiced speech, where the excitations are much closer together and therefore the impulse response actually collide into one other, the technique performs quite badly.

![Figure 5.42: (a) unwrapped phase, (b) Phase of the differential spectrum ($\gamma = 1$).](image-url)
5.7 Modelling speech

In order to avoid the problems discussed in the previous section, the RCD cost function of Eqn. (4.6) must be modified in order to model speech signals. Therefore the cost function, of Eqn. (4.4) must be re-defined to exclude the phase terms, such that the error function of Eqn. (4.3) is constrained to modelling magnitude only. Since, the excitation, \( U(w) \) can be any function, the cost function must include \( U(w) \).

\[
J = \mathcal{E} \left[ |E(w)|^2 \right] \tag{5.17}
\]

where, \(|E(w)| = |X(w)| - |\hat{X}(w)|\), which may be defined as,

\[
X(w) = U(w) \cdot P(w) \tag{5.18}
\]
\[
\hat{X}(w) = U(w) \cdot M(w) \tag{5.19}
\]

Therefore, it can be seen that by combining the previous equations the magnitude cost function may be explicitly defined as,

\[
J = \mathcal{E} \left[ |X(w)|^2 + |\hat{X}(w)|^2 - 2|X(w)| \cdot |\hat{X}(w)| \right] \tag{5.20}
\]

Properties of the Magnitude Cost function

As with Eqn. (4.6) the magnitude cost function may be used to model any type of system with no a priori knowledge. However, unlike its predecessor, Eqn. (5.20) may be used to model processes (i.e., models excited with white noise, such as unvoiced speech), and therefore may be used instead of the standard real cepstrum.

Eqn. (5.20) has several disadvantages, such as twice the number of coefficients are now required, since the magnitude spectrum is used. Secondly, as with the real cepstrum the technique is not reversible, since phase is discarded and therefore it will not be able to model a signal exactly - even a single impulse response.

Modelling Performance

Eqn. (5.20) was applied to a frame of voiced speech and unvoiced speech. As seen in figures 5.43 and 5.45, the RCD system outperforms the modelling performance of
5.7. Modelling speech

the real cepstrum. Analyzing figure 5.45 further, notice that the RCD is much more spectrally closer to \( X(w) \) than LPC or the real cepstrum, and has produced some extra information in the 1-1.5kHz region. This is much more apparent in figure 5.43, where the RCD has modelled the 1.8kHz valley much more accurately. Also notice that LPC has modelled the voiced speech spectra with two distinct formants and has pretty much neglected the rest of the signal. This is as expected, since LPC is constrained to a relatively short all-pole model (10th order) that will perform badly when modelling zeros. Thus, the RCD will generally have superior modelling performance over LPC.

Finally, rather than show lots of examples to cover all types of speech classes, the results of a single speech waveform is shown in figure 5.47. The sentence, "card games are hard to play", spoken by an adult male speaker was processed by the RCD and Eqn. (5.20), in frames of 256 with no overlapping and no windowing. Notice how \( \gamma \) is limited to positive values only. This is attributed to negative values of \( \gamma \) giving rise large resonant peaks (like LPC) that are spectrally distant from \( X(w) \). Note that when \( \gamma = \pm 0.01 \), the RCD is similar to the real cepstrum [50]. It may therefore be concluded, that a logarithmic function is not necessarily the best operator for modelling speech, and a dynamic method of assigning \( \gamma \) is required, depending upon the class of sound (fricative, plosive etc.).

NB. That increasing the lifter length results in the usual problem of the spectral envelope containing anomalies, and therefore as demonstrated empirically, a lifter length of 11 is sufficient to model speech.

An RCD speech synthesis system, devised by Lim [50] suggested that when \( \gamma = -\frac{1}{3} \), the fidelity of the synthesized speech was superior to that obtained by the real cepstrum. This could be attributed to sharper formants, although Lim's results were inconclusive. Further work undertaken by Murthy [56], proposed a method of pitch extraction by extending the range of \( \gamma \), where, \(-2 \leq \gamma \leq 2\). Murthy demonstrated that a low value of \( \gamma \) was required for the extraction of high pitched voice, such as female speech, and \( \gamma \approx 1.5 \) was required for extraction in noise. However, extending \( \gamma > 1 \) generally leads to unsatisfactory modelling performance, as seen in figure 5.48.
Chapter 5. Simulation Results

Figure 5.43: RCD performance for a frame of voiced speech, lifter length=11.

Figure 5.44: MSE Vs. \( \gamma \) for figure 5.43, lifter length=11.
5.7. Modelling speech

\[ Y = 0.45 \]

Figure 5.45: RCD performance for a frame of unvoiced speech, lifter length=11.

\[ \gamma = 0.45 \]

Figure 5.46: MSE Vs. \( \gamma \) for figure 5.45, lifter length=11.
Chapter 5. Simulation Results

Figure 5.47: Optimal $\gamma$ per frame for the utterance "card games are hard to play".

Figure 5.48: Optimal $\gamma$ for a frame of voiced speech, lifter length=11, when $\gamma > 1$. 
5.8 Summary and Conclusions

After studying this chapter it is hoped that the reader has appreciated the effectiveness of the RCD algorithm for compactly modelling the properties of an unknown plant or model with relatively few adjustments to the lifter length. It was demonstrated that the RCD's modelling performance for an all-pole model in noise was superior to that of an equivalent order LPC model. Although the LPC formants were centred at the correct resonant frequencies, the RCD was spectrally closer to the plant's magnitude spectra.

Next the all-zero model was considered. It was shown that when modelling an impulse response, a special relationship exists between the RCD, when $\gamma = 1$ and the FIR Wiener filter, such that the modelling performance was identical. This concept was developed further for the pole-zero model. Therefore, when modelling a particular pole-zero model it was demonstrated that the RCD outperformed the modelling performance of an equivalent order FIR Wiener filter 32 fold. The superior modelling performance of the RCD was attributed to the mixed pole-zero cepstral model, unlike the Wiener filter that was constrained to an all-zero model.

The compression performance of the RCD was tested for a 24th order pole-zero model, consisting of 12 poles and 12 zeros, equally spaced and of equal radius. The compression factor (CF) was calculated by the following formula,

$$CF = \frac{\text{total number of plant or model coefficients}}{\text{Cepstral coefficients}(1..\text{lifter length}) + \gamma_{\text{opt}} + \text{lifter length}}$$

It was shown that a CF of 2.36 could be achieved, meaning that a system could be modelled with fewer coefficients than the model itself. However, a compression factor of this magnitude is not generally common, as seen from the other pole-zero models discussed herein and therefore the compression factor will depend upon the type of data used (i.e., the pole-zero distribution).

In section 5.2.5, a special relationship between the RCD and the Padé approximation was developed and discussed. When modelling a signal containing no noise it was shown that when $\gamma = -1$ and constraining the Padé model to all-pole ($q = 0$), the solutions obtained for modelling a mixed model were identical. Likewise, when the Padé model was constrained to an all-zero model ($p = 0$), the results were identical to when $\gamma = 1$. 
Chapter 5. Simulation Results

In section 5.3 the separation of the Fourier magnitude and phase and analyzing the two independently was considered. It was shown that for mixed models, separating magnitude and phase and assigning different values of $\gamma$ was indeed a good idea, yielding superior modelling performance at the expense of an increased number of cepstral coefficients. However, in the absence of noise, when modelling an all-zero or all-pole model, separation is unnecessary, since the optimal solution would always be $\gamma = \pm 1$. Therefore, this technique may only be used in circumstances where modelling accuracy is more important than model order compression.

In order to model systems with maximum and mixed phase properties, it is necessary to re-define the standard lifter function. However, this re-definition results in many more coefficients and generally does not yield good compression results. As discussed in section 5.5, when modelling an unknown plant with minimum, mixed or maximum phase properties, the system could be modelled exactly when $\gamma = 1$, which is no compression at all.

Model order identification using z-plane cepstral plots was developed further in section 5.4 and some general observations made concerning model order identification. As demonstrated in section 5.4, the model order may be determined by the sum of all the uncrossed loci, and the poles and zeros determined by the colour of the loci. Results of pole/zero models of different radii were also presented and it was seen that the technique only modelled the more dominant poles or zeros of the system. Unfortunately, performance in noise was worse than that of LPC, leading to difficulties when trying to determine the model's order.

Finally, the modelling of speech signals was considered. The standard cost function of Eqn. (4.6) was modified to exclude any phase terms, such that any modelling problems due to signal alignment and multiple excitation were avoided. It was shown that using the RCD and Eqn. (5.20), the modelling performance for almost all classes of speech was superior to that of the real cepstrum and in many cases LPC. However, the use of Eqn. (5.20) now results in twice the amount of coefficients than the original definition of Eqn. (4.6). Nevertheless the superior modelling performance over the real cepstrum makes this technique a strong competitor to LPC.
5.8. Summary and Conclusions

5.8.1 Further Discussions

Table 5.7 summarizes the number of coefficients required to model a system using the proposed minimum phase RCD implementation. Notice that as the first cepstral value is always equal to one, this may be omitted from storage, and introduced at the synthesis stage. Also, if the lifter length is kept constant (modelling speech data), the number of coefficients may be reduced even further by omitting the lifter length parameter. However, recall that when modelling speech, the lifter must be applied to both sides of the cepstral domain, resulting in twice the amount of cepstral data than the minimum phase method.

<table>
<thead>
<tr>
<th>lifter length</th>
<th>$\gamma_{opt}$</th>
<th>$[1 \ldots \text{lifter length}]$</th>
</tr>
</thead>
</table>

Table 5.7: Frame of compressed data.

A possible set of cepstral data frames for modelling speech is shown in table 5.8. Comparing the number of cepstral parameters to LPC and assuming that the model order is known, it can be seen that there are approximately 23 cepstral coefficients compared to 11 LPC coefficients (10th order LPC). Designing a low bandwidth communication system and assuming that the residual or phase information is identical for both techniques, the cepstral method seems less attractive, but has the virtue of modelling the effects of both poles and zeros.

<table>
<thead>
<tr>
<th>lifter length</th>
<th>$\gamma_{opt}$</th>
<th>$[1 \ldots 2 \times (\text{lifter length})]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma_{opt}$</td>
<td>$[1 \ldots 2 \times (\text{lifter length})]$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

Table 5.8: Frame of RCD compressed data when modelling speech.

Of course, many more fundamental issues must be addressed for efficient real time implementation and require much more careful consideration than the superficial treatment given in this thesis.
Chapter 6

Phase Analysis

6.1 Introduction

As demonstrated in chapter 5, the RCD performance is limited to the analysis of a single impulse response that is suitably aligned for the lifter function. However, when modelling signals comprised of multiple excitations, such as voiced speech, the traditional approach has been to employ the real cepstrum, since any signal alignment and phase unwrapping issues can be avoided. Therefore an important question is why retain phase at all? Revisiting chapter 5, it was demonstrated that phase retention plays a crucial role in signal reproduction and is an integral part of the cost function of Eqn. (4.6). For example, when analyzing a speech signal the phase information contains the accent of the speaker, and its omission results in a robot like sound quality. However, vocoders, such as the GSM algorithm [27] use LPC to compute the magnitude spectrum and an inverse LPC filter to obtain a residual. The residual contains the phase information and is compressed by the algorithm and transmitted to the decoder along with the LPC coefficients for re-construction [45].

Revisiting section 5.6 it was originally thought that the phase unwrapping algorithm was the culprit for poor modelling performance when analyzing a signal with a multiple excitation. However, further analysis undertaken by the author will demonstrate that the problem is much more fundamental, and is actually due to discontinuities in the computed DFT (discrete Fourier transform). Some alternative techniques, such as the
Hartley transform are presented and their modelling performance assessed.

6.2 The DTFT Vs. The DFT

There are two types of Fourier transform for analyzing finite length discrete time signals. The DTFT (discrete time Fourier transform) is a theoretical analysis tool that is not computationally implementable on a DSP, since it is a function of continuous frequency. The DFT (discrete Fourier transform) on the other hand, circumvents the implementation problem by computing the exact samples of the DTFT at finite spaced frequency intervals, and as a consequence has limitations over the DTFT.

The DTFT of a finite length discrete signal, \( x(n) \), where \( n \in [0, N - 1] \) may be expressed as,

\[
X(w) = \sum_{n=0}^{N-1} x(n)e^{-jwn}
\]  

and its inverse (IDTFT) is given by,

\[
x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(w)e^{jwn} dw
\]

The DFT of \( x(n) \) over the same interval may be expressed in terms of \( N \) equally spaced frequency points,

\[
X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)kn}, \quad k = 0..N - 1
\]

and its inverse (IDFT)

\[
x(n) = \sum_{k=0}^{N-1} X(k)e^{j(2\pi/N)kn}, \quad n = 0..N - 1
\]

Thus, the DFT calculates \( N \) equally spaced frequency samples of the DTFT of the finite signal, \( x(n) \) in the interval \( k = 0..N - 1 \). Note that the two indices, \( n \) and \( k \) correspond to a time index and frequency index respectively. Therefore, as long as \( N \) is greater than or equal to the length of the sequence, \( x(n) \), there will be no time aliasing \([14]\) and the IDFT=IDTFT. As one might expect, the resolution or frequency step size is defined by, \( \frac{2\pi}{N} \). Therefore computation time may be reduced by pre-calculating the twiddle factors of Eqn. (6.5).

\[
W = e^{-j2\pi k/N}, \quad k = 0..N - 1
\]

Of course this is an implementation issue and forms the basis of the FFT (fast Fourier transform) algorithm, which is essentially a fast method of computing the DFT coefficients. Finally, realizing that \( X(w) \) is periodic with periodicity \( 2\pi \), each frequency step may be defined as, \( \frac{2\pi}{N} k \). Therefore, substituting this into the DTFT of Eqn. (6.1), results in the definition of the DFT, as shown below [14].

\[
X\left(\frac{2\pi}{N} k\right) = \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N} k\right)n}
\]

(6.6)

Therefore, it is reasonable to suggest that the results obtained by the DFT will differ from that of the DTFT in between the sampling instants.

### 6.3 Phase Analysis for signals comprised of multiple excitations

As mentioned in the introduction, phase analysis is poor for signals comprised of multiple excitations. As an example, consider a signal of length 256 comprised of 25 equally spaced unit samples, with an effective periodicity of 10 samples (intermediate sample values set to zero). Analyzing the magnitude spectrum, shown in figure 6.1, it can be seen that there are 10 harmonics, approximately 25 frequency samples apart. Analyzing figure 6.2, notice that the real components of the DFT (a) have 10 discontinuities, whereas the imaginary components of the DFT (b) only have 8 and unlike (a), there are no discontinuities at zero and the nyquist frequency. These results suggest that modelling signals of this type is fundamentally flawed by use of the DFT. Indeed, it was originally believed that these discontinuities were caused by the phase unwrapping algorithm [81, 87]. However, it is possible that the phase unwrapping algorithm may cause additional discontinuities. Figure 6.3 shows the corresponding unwrapped phase, notice how the transitions are inside the \( \pm\pi \) limits, presenting problems for the unwrapper. However, analyzing figure 6.3 more carefully, notice that there is an apparent discontinuity in between the second and third harmonic, due to the real and imaginary components simultaneously approaching zero (see figure 6.2). In essence, this is a computational error due a specific condition of the synthetic data.
6.3. Phase Analysis for signals comprised of multiple excitations

Figure 6.1: Magnitude spectra (DFT of 25 unit samples).

Figure 6.2: DFT of 25 unit samples (a) Real components (b) Imaginary components.
Figure 6.3: unwrapped phase (DFT of 25 unit samples).

Figure 6.4: Phase of the differential spectrum, $\gamma = 1$ (DFT of 25 unit samples).
6.3. **Phase Analysis for signals comprised of multiple excitations**

Analyzing figures 6.2, 6.3 and 6.4 more closely, notice that the positions of the discontinuities all occur at the same place (the harmonic frequencies), suggesting that all methods are unsuitable for modelling signals of this kind. Also note, that the phase of the differential spectrum, as shown in figure 6.4, circumvents the "apparent discontinuity" problem that manifests itself in the unwrapped phase.

Analyzing the DFT in more detail, another explanation for the poor modelling performance comes to light - the sampling frequency. Since the frame length is 256, this corresponds to an approximate frequency step size of 0.025. Therefore, it is possible that the discontinuities may be due to an insufficient sampling rate. In order to overcome this, the signal was interpolated by extending the frame length to $2^{16}$ (i.e., 65536 unit samples), as shown in figure 6.5. Notice that the real and imaginary components of figure 6.5 still have discontinuities in the same places as figure 6.2, but the finer sampling rate has reduced the tails of the discontinuities. NB. That simply interpolating (padding with zeros) the original frame of 256 would cause damped oscillations in between the peaks. Therefore it can be concluded that even a large frame length does not overcome the discontinuity problem, when analyzing signals comprised of multiple excitations using the DFT.

However, since the DFT is evaluated around the unit circle, any zeros or poles that lie on the unit circle will cause a discontinuity. Evaluating the roots of the synthetic signal, it was observed that complex conjugate zero pairs manifest themselves on the unit circle at the harmonic frequencies. Therefore, even a large frame length will not overcome this problem. One possible solution may be to warp the z-plane, such that any poles/zeros on the unit circle will be moved either inside or outside the unit circle, thus ensuring that the phase spectra will be continuous. However, much more work is needed in this area.
6.4 Hartley Phase

The Hartley transform [7, 8] originates from the Fourier transform. Therefore the Hartley spectrum, \( H(w) \) is the real-imaginary components of the Fourier spectrum.

\[
H(w) = \text{real} \{X(w)\} + \text{imag} \{X(w)\} \quad (6.7)
\]

The Hartley phase, \( V(w) \) is defined as,

\[
V(w) = \frac{H(w)}{M(w)} \quad (6.8)
\]

where, \( M(w) \) is the magnitude spectrum of Eqn. (6.7) respectively. Analyzing figure 6.6, notice the discontinuities in the spectra. Therefore suggesting that even the Hartley spectrum technique has problems when analyzing signals with multiple excitations. This is reasonable, since the Hartley spectrum is "built" from Fourier components. Notice that due to the real and imaginary components of the DFT (see figure 6.2) simultaneously approaching zero, Eqn. (6.8) has computed an indeterminant result.
As seen from the previous sections, both the DFT and Hartley transform suffer from discontinuities when analyzing signals comprised of multiple excitations. Therefore, an appropriate pre-filtering algorithm, that removes the repetition from the signal to be analyzed is proposed. In essence, the periodicity and position of the “first impulse response” is calculated by a cross and autocorrelation respectively. A window function (rectangular or Blackman) is then applied to the data in order remove the repetitive responses (as shown in figure 6.7). The method is summarized as follows:

1. autocorrelate $x(n)$ to obtain the pitch period.
2. cross-correlate $x(n)$ with a Kronecker delta function that is aligned left, in order to establish the position of first impulse response.
3. window $x(n)$ with a rectangular or Blackman window function, with position and width obtained from steps 1 and 2.
4. use the Matlab integer lag offset procedure, as described in section 5.6 in order to remove any offsets.

Figure 6.6: Hartley Phase (DFT of 25 unit samples).
5. apply data to the RCD algorithm.

*NB. See the implementation section for more details.*

![Figure 6.7: "Sliding the window function" over the first impulse response.](image)

**6.5.1 Modelling Performance**

Revisiting the all-pole model of section 5.2.1, a contrast can be made between the standard RCD algorithm and the proposed scheme, as discussed in this chapter (all-pole model excited by two equally spaced excitations). Examining the entries of table 6.1 and comparing them to the values obtained in table 5.1 it can be concluded that the newer method does not adversely affect the modelling performance. Although, a slight reduction in performance is to be expected, due to the effects of windowing and phase unwrapping errors. Naturally, other window functions can be used, such as Hamming, Blackman, Bartlett etc., but simulations have showed that good modelling performance can be achieved with the either a rectangular or Blackman window function.

Figure 6.8 shows the corresponding optimal magnitude and phase responses for a lifter=5 (rectangular window). Notice once again that the spectral match is perfect. In general this will not be true, since it is assumed that each impulse response has
6.5. A Pre-Filtering Algorithm

<table>
<thead>
<tr>
<th>MMSE</th>
<th>0.86048</th>
<th>$7.6304 \times 10^{-15}$</th>
<th>0.0786</th>
<th>$7.6304 \times 10^{-12}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimal $\gamma$</td>
<td>1</td>
<td>-1</td>
<td>0.34</td>
<td>-1</td>
</tr>
<tr>
<td>lifter length</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 6.1: RCD optimization performance for Eqn. (5.1)

sufficiently decayed between each pitch period. Violation of this rule usually results in the window function truncating part of the signal, leading to difficulties for the re-construction section.

![Graphs showing phase and magnitude](image)

Figure 6.8: The optimal magnitude and phase for Eqn. (5.1).

Applying the new scheme to the pole-zero model of Eqn. (5.3), it was seen that the modelling performance was identical to the entries of table 5.6, therefore re-enforcing the usefulness of the method for analyzing signals comprised of multiple excitations.

6.5.2 Modelling Speech

Modelling voiced speech still has its problems. Although locating the position of the first excitation and extracting the pitch is a well established technique, the modelling
performance, once the data is windowed is poor. One possible explanation is that the excitation separation rule, as discussed in section 6.5.1, has been violated and the impulse responses collide into one another. Note also, that the proposed method does not circumvent the problems associated with unvoiced speech and is therefore still unable to model AR, MA or ARMA processes.

6.6 Summary and Conclusions

The purpose of this chapter was to investigate the poor phase modelling performance of the ROD algorithm when modelling signals comprised of multiple excitations. As demonstrated and discussed in section 6.3 the components of DFT and its relatives, all suffer from discontinuities, even for large frame lengths. It may therefore be concluded that phase analysis under the DFT/FFT is fundamentally flawed, as complex conjugate zero pairs manifest themselves directly on the unit circle at the harmonic frequencies, although further work is needed in this area.

A possible solution for analyzing multiple excitation signals was proposed in section 6.5. Utilizing the cross and auto correlation functions, the position of the first impulse response may be determined respectively. An appropriate window function (rectangular or Blackman) may then be applied to the data in order to remove the repetitive responses. The proposed method exhibited similar modelling performance to the original method as discussed in chapter 5, but with the limitation that the impulse responses were well separated in the time domain. As a consequence, the new method was found to be unsuitable for modelling voiced speech and stochastic processes.
Chapter 7

Conclusions and Further Work

The fundamental versatility of the RCD algorithm lies in its ability to model the characteristics of an unknown plant or model to a specified acceptable modelling error (AE) with no a priori knowledge of the pole-zero distribution. The algorithm's superior modelling performance over the real cepstrum and LPC can be attributed to the utilization of a root cepstral cost function, as defined by Eqn. (4.6) being applied directly to the root cepstral domain. As discussed and demonstrated in chapters 4 and 5, the RCD algorithm assumes that the excitation is a Kronecker delta function, and therefore the signal to be analyzed is an impulse response. Although the RCD is based upon the complex cepstrum, which is generally used for deterministic modelling, the proposed algorithm is flexible enough to model deterministic signals corrupted with bandlimited Gaussian white noise, as demonstrated throughout chapter 5. The log cepstrum on the other hand, tends to "amplify" the samples of noise and is generally unsuccessful when modelling signals of this kind, as seen in section 3.4.3.

For a majority of the thesis it is assumed that the impulse response is aligned left (i.e., starting from time index zero). However, this assumption is rather idyllic and therefore not representative of real world situations, since most data records will generally have a gap before the start of the data. However, Matlab's phase unwrapping algorithm [81, 87] may be modified, such that the phase corresponding to integer lag may be calculated and removed. The integer lag may then be re-introduced at the synthesis stage. Alternatively, utilizing the differential cepstrum and hence differentiating the
phase cepstra, as discussed in section 3.7, any offsets may also be removed from the analysis. Therefore either method may be used for an appropriate pre-filtering algorithm to ensure that any offsets are removed and the impulse response is aligned left. Unfortunately, both techniques are limited to the analysis of a single impulse response, and as a consequence neither method is suitable for analyzing signals with multiple excitations, such as voiced speech.

The analysis of signals comprised of multiple excitations was discussed in some depth in chapter 6. It was shown that the components of DFT and its relatives, all suffer from discontinuities, even for large frame lengths. It may therefore be concluded that phase analysis under the DFT/FFT is fundamentally flawed, as complex conjugate zero pairs manifest themselves directly on the unit circle at the harmonic frequencies, although further work is needed in this area. However, a simple pre-filtering technique was proposed with some success as a possible solution to the problem.

Special relationships between the RCD algorithm and two time domain modelling methods were developed and discussed in chapter 5. From the discussions of section 5.2.5 it may be concluded that when modelling a pole-zero plant in the absence of noise, the RCD ($\gamma = -1$) is mathematically identical to an equivalent order all-pole Padé model and also equivalent to an all-zero Padé model when $\gamma = 1$.

A second relationship between the RCD ($\gamma = 1$) and the FIR Wiener filter was then developed. Revisiting section 5.2.3 it may be concluded that when modelling a particular pole-zero plant, the modelling performance of the RCD surpassed that of an equivalent order Wiener filter 32 fold. The superior modelling performance of the RCD was attributed to the mixed pole-zero cepstral model, unlike the Wiener filter that was constrained to an all-zero model.

Due to the fact that Fourier magnitude and phase are orthogonal, the discussion of section 5.3 proposed the separation and independent analysis of magnitude and phase to improve the modelling accuracy. It was shown that for mixed models, separating magnitude and phase and assigning different values of $\gamma$ was indeed a good idea, yielding superior modelling performance at the expense of an increased number of cepstral coefficients. However, in the absence of noise, when modelling an all-pole or all-zero model, separation was unnecessary, since the optimal solution would always be $\gamma = \pm 1$. 
Therefore, this technique may only be used in circumstances where modelling accuracy is more important than model order compression.

In order to model plants with maximum or mixed phase properties, it is necessary to redefine the standard lifter function (see section A.1.2). However, this re-definition results in many more coefficients, since the cepstrum must be liftered left and right, yielding unsatisfactory compression results. As discussed in section 5.5, when modelling an unknown plant with minimum, mixed or maximum phase properties, the system could be modelled exactly when $\gamma = 1$, which is no compression at all. However, for fractional values of $\gamma$, the cepstral data tends to be spread out across the whole frequency domain, leading to difficulties for the statically placed lifter function.

A novel method of model order identification for use with time domain modelling methods based around z-plane root cepstral plots was developed and discussed in sections 4.4 and 5.4. As demonstrated herein, the positions of a model or plant's poles and zeros may be determined by visual inspection of the resulting z-plane plot and the model's (plant) order may be determined by the sum of all the uncrossed loci, and the poles and zeros determined by the colour of the loci respectively. The technique is limited to the modelling of dominant poles and zeros only and performance in noise was poor compared to that of LPC, leading to difficulties when trying to determine the true model order.

The proposed identification technique stands alongside those of chapter 2.11, but has the versatility of analyzing poles or zeros in unbiased fashion, since the analysis is undertaken in the frequency domain.

Modelling speech signals required some re-definition of the standard cost function of Eqn. (4.6) in order to exclude any phase terms, such that any modelling problems due to signal alignment and multiple excitation were avoided. It was shown that using the RCD and Eqn. (5.20), the modelling performance for almost all classes of speech was superior to that of the real cepstrum and in many cases LPC. However, the use of Eqn. (5.20) now results in twice the amount of coefficients than the original definition of Eqn. (4.6) and is no longer a reversible process. Note that extending the range of $\gamma$ (i.e., $-2 \leq \gamma \leq 2$) generally leads to unsatisfactory modelling performance.
In terms of computational requirements the RCD requires at least two Fourier transforms, two phase unwrapping operations and many multiplications, which is much more computationally intensive than the comparatively simple LPC algorithm. Nevertheless, in spite of these shortcomings the superior modelling performance over the real cepstrum makes this technique a strong competitor to LPC.

7.1 Further Work

The results and findings of this thesis have demonstrated that the RCD may be used to model a minimum phase model or plant efficiently and compactly. This was developed further for modelling speech signals, where a special form of cost function was devised and shown to out perform the real cepstrum and LPC. However, the issue of RCD implementation has received very little attention. Areas for further work may therefore include the issues of coefficient quantization, the effects of noise on RCD coefficients and general efficient real time implementation of the cost functions and the RCD algorithm.

The Bicepstrum also has received very little attention, since it is not suitable for the generalized modelling approach of the root cepstrum, due to the dependence of the log operator. However, recalling the brief discussion of section 3.8, it can be seen that the Bicepstrum overcomes the problems of the standard phase unwrapping algorithm and suppresses the effects of noise by the use of cumulants. Therefore, another area for further work would be to define a Bi-root cepstrum for the purposes of modelling a mixed phase system in the presence of noise. This may then be extended to the modelling of speech signals.

Finally, the poor phase modelling performance of the RCD algorithm, when modelling signals comprised of multiple excitations, requires much more careful analysis. As discussed in chapter 6, the components of DFT and its relatives, all suffer from discontinuities. It may therefore be concluded that phase analysis under the DFT/FFT is fundamentally flawed, as complex conjugate zero pairs manifest themselves directly on the unit circle at the harmonic frequencies. However, much more work is required in this area in order to substantiate these claims.
Appendix A

Algorithm Implementation

A.1 RCD

The purpose of this appendix is to summarize and clarify any of the implementation issues concerning the RCD algorithm and any of its components. All software shown in this appendix is Matlab code. For a more detailed explanation of any specific Matlab command, please refer to the Mathworks website [90].

A.1.1 Analysis

This code builds the root cepstrum by joining both the real and imaginary components of the IDFT separately.

$$X = \text{fft}(x); \quad \% \text{spectra of plant output}$$
$$\text{mag}_X = \text{abs}(X); \quad \% \text{magnitude and}$$
$$\text{phase}_X = \text{unwrap}(	ext{angle}(X)); \quad \% \text{unwrapped phase}$$
$$f = (\text{mag}_X \cdot \gamma) \cdot \cos(\gamma \cdot \text{phase}_X); \quad \% \text{build root cepstrum}$$
$$p\_\text{ceps\_real} = \text{real}(\text{ifft}(f)); \quad \% \text{real components}$$
$$g = (\text{mag}_X \cdot \gamma) \cdot \sin(\gamma \cdot \text{phase}_X);$$
$$p\_\text{ceps\_imag} = \text{imag}(\text{ifft}(g)); \quad \% \text{imaginary components}$$
$$\text{root\_cepstrum} = p\_\text{ceps\_real} - p\_\text{ceps\_imag}; \quad \% \text{join the two together}$$
A.1.2 Lifter Function

The minimum phase or default rectangular lifter (window) function is implemented as follows, where lifter is the lifter length and 256 is the frame length.

\[
\text{root}_\text{cepstrum}(\text{lifter}+1:256) = 0;
\]

*Important.* Notice that this type of implementation results in lifter coefficients.

The lifter function for maximum phase or the real cepstrum may be modified to include both sides of the cepstrum, as shown below.

\[
\text{root}_\text{cepstrum}(\text{lifter}+1:256-\text{lifter}-1) = 0;
\]

The Cepstral data is normalized by the following function

\[
\text{liftered}_\text{ceps} = \frac{\text{root}_\text{cepstrum}(\text{lifter})}{\text{root}_\text{cepstrum}(1)};
\]

A.1.3 Synthesis

An estimate of the plant (p.hat) is determined by a reverse procedure to that of section A.1.1. I.e., the (·)\(^{-T}\) operator is replaced by (·)\(^{1/7}\). NB. That the model output is referred to as “p.hat”.

\[
\text{M} = \text{fft}(\text{liftered}_\text{ceps});
\]

\[
\text{mag}_\text{M} = \text{abs}(\text{M});
\]

\[
\text{phase}_\text{M} = \text{unwrap}(\text{angle}(\text{M}));
\]

\[
\text{f} = (\text{mag}_\text{M}^{-(1/\gamma)}) \cdot \text{cos}(\text{phase}_\text{M} / \gamma);
\]

\[
\text{m}_\text{ceps}_\text{real} = \text{real}(\text{ifft}(\text{f}));
\]

\[
\text{g} = (\text{mag}_\text{M}^{-(1/\gamma)}) \cdot \text{sin}(\text{phase}_\text{M} / \gamma);
\]

\[
\text{m}_\text{ceps}_\text{imag} = \text{imag}(\text{ifft}(\text{g}));
\]

\[
\text{p.hat} = \text{m}_\text{ceps}_\text{real} - \text{m}_\text{ceps}_\text{imag};
\]

\[
\% \text{p.hat} \text{ is the model's estimate of the plant}
\]

A.1.4 Cost function of Eqn. (4.6)

The following piece of code calculates \( \hat{J} \) for Eqn. (4.6), where the frame of data is assumed to be of length 256 for computational convenience.
A.2. z-plane Root Cepstral plots

\[
P_{\hat{\text{h}}} = \text{mag}_M \cdot \frac{1}{\gamma}; \quad \% \text{ plot spectra magnitude and phase)
\]
\[
P_{\hat{\text{h}}}, \text{phase} = \frac{1}{\gamma} \cdot (\text{phase}_M);
\]
\[
P_{\text{mag}} = |\text{fft}(p)|;
\]
\[
P_{\text{phase}} = \text{unwrap}(\text{angle}(\text{fft}(p)));
\]
\[
\text{phase_error} = \text{P}_{\text{phase}} - \text{P}_{\hat{\text{h}}},\text{phase};
\]
\[
J_{\hat{\text{h}}} = \sum((P_{\text{mag}})^2 + (P_{\hat{\text{h}}},\text{phase})^2 - 2 \cdot (P_{\text{mag}} \cdot P_{\hat{\text{h}}},\text{phase}) \cdot \cos(\text{phase_error})) / 256;
\]

A.1.5 Cost Function of Eqn. (5.20)

The following piece of code calculates \( J \) for Eqn. (5.20), where the frame of data is assumed to be of length 256 for computational convenience and \( \text{voice} \) is the present frame of speech data. NB. That log has only been used for display purposes.

\[
\text{spec}_\text{env} = \log(\text{real}(\text{fft}(\text{root_cepstrum})));\n\]
\[
\text{spec} = \log(\text{abs}(\text{fft}(\text{voice})));\n\]
\[
\text{model_spec} = \frac{1}{\gamma} \cdot \text{spec}_\text{env};\n\]
\[
J = \sum(\text{spec}^2 - \text{model_spec}^2 - 2 \cdot \text{spec} \cdot \text{model_spec}) / 256;
\]

A.2 z-plane Root Cepstral plots

This section of code calculates the roots of cepstral model (polynomial) and plots them on the z-plane. The colour of the “zeros” are dependent upon the value of \( \gamma \).

\[
\text{hold on};\n\]
\[
\text{zero_loc} = \text{roots}(\text{polynomial});\n\]
\[
\text{if} \ (\gamma \leq 1) \ & \ (\gamma > 0)\n\]
\[
\quad \text{colour} = [\gamma \ 0 \ 0];\n\]
\[
\text{else}\n\]
\[
\quad \text{colour} = [0.2 \ 0.8 \ \text{abs}(\gamma)];\n\]
\[
\text{end};\n\]
\[
\text{plot}(\text{real}(\text{zero_loc}), \text{imag}(\text{zero_loc}), 'o', 'MarkerEdgeColor', \text{colour});\n\]
\[
\text{hold off};\n\]
Appendix A. Algorithm Implementation

A.3 Pre-filtering Algorithm

The following piece of code calculates the pitch period and the position of the first impulse response. See section 6.5 for more details. NB. imp is a Kronecker delta function and y is the time domain signal to be analyzed.

```matlab
max_lag=150;
cross_corr=xcorr(y,imp,max_lag);
acf=xcorr(y,max_lag);
acf=acf(max_lag+1:max_lag*2);

peak=0; % determine pitch period
for d=7:max_lag,
    if acf(d)>peak
        peak=acf(d);
        width=d;
    end;
end;

peak=0; % determine position of first
for d=max_lag:max_lag*2, % impulse response
    if cross_corr(d)>peak
        peak=cross_corr(d);
        pos=d-max_lag;
    end;
end;

window=boxcar(2*width); % window data sequence
y(pos-width:pos+width-1)=y(pos-width:pos+width-1).*window;
y(1:pos-width)=0;
y(width+pos-2:256)=0;
```


Bibliography


[27] GSM. *Digital cellular telecommunications system (phase 2+) full rate speech transcoding*, 1999. ETSI EN 300 961, GSM 06.10 version 8.1.1.


Overview of all modern types of compression algorithms.

A website devoted to speech coding algorithms. A practical comparison between all modern speech coding algorithms is given.

Matlab implementation details and theory about Tribolet's algorithm.

A good report on the effects of noise on LPC coefficients.

Matlab implementation details and theory about the complex cepstrum.

The main Mathworks (the creators of Matlab) website.


