Signals in Nonlinear Bandpass Systems

Ian W. Dall, B.E. (Hons.)

Department of Electrical and Electronic Engineering
University of Adelaide
Adelaide, South Australia.

A thesis submitted for the degree of Doctor of Philosophy in:

May 1991

Abstract

Nonlinear distortion is a factor limiting performance in many bandpass systems such as are found in communications and radar. This thesis addresses a number of theoretical and practical issues relevant to the modelling and correction of distortion in High Frequency Radar Receivers. The theoretical results and the experiment methods are applicable to other bandpass systems.

Techniques for modelling nonlinear communication systems, using power series, amplitude and phase describing functions and Volterra series are reviewed. Some effort is expended in determining the class of nonlinear systems described by each model. In many systems, it often the case that only the class of systems with bandpass input and output is of interest. The usefulness of these models in predicting the performance of communications systems is evaluated.

It is shown that, for all three models considered, distortion in a bandpass context can be considered equivalent to distortion of the complex envelope of signal. It is also shown that the power series model and the amplitude and phase describing function model are special cases of the Volterra series model.

Many bandpass systems perform a frequency translation function. Radio receivers invariably use mixers to achieve frequency translation. Mixers are two input devices and in general the output is a nonlinear functional of both the input signal and the local oscillator signal. In classical communications receivers the local oscillator is a sinusoidal function of time, but in applications such as frequency hopping communications and in radar, the local oscillator may be a more complex signal.

The number of terms required to model a general, nonlinear functional of n signals goes up as $O(k^n)$ so it is undesirable to model a mixer as a general two input nonlinearity if it is avoidable. Further, modelling a real receiver

containing mixers as a general, multi-input nonlinearity would require extra instrumentation to monitor the local oscillators.

The Wiener model of nonlinear systems is extended to the case of two input nonlinear systems. This representation is then used to examine the conditions under which a real mixer can be modelled by a combination of one input nonlinearities and ideal mixers. The extended Wiener model is also be used to show that, in a bandpass context, a phase shift in one local oscillator results in the same phase shift in all the output components.

The discrete time version of the Volterra series is chosen as a suitable means of modelling and correcting nonlinear distortion, since it is capable of approximating arbitrarily closely a very large class of nonlinear systems. An adaptive nonlinear filter based on the discrete time Volterra series is discussed and extended to the two input case. A number of different adaption algorithms are discussed and two nonlinear recursive least squares (RLS) algorithms (lattice and fast Kalman) are derived.

An experiment to estimate nonlinear distortion in an HFR receiver is described. There are a number of characteristics which make this experiment different from others described in the literature. Specifically:

- The receiver must be modelled as a two input device. The relative phase of the inputs can not be measured or controlled[1].
- The receiver has slight, but important, nonlinear distortion in the presence of large linear distortion. These two factors place severe constraints on the adaption algorithms, and prompted the development of the nonlinear RLS algorithms mentioned above.
- The receiver's inputs and outputs are at different frequencies. This was

[1] All the signals are phase locked, but there is no way to set the initial phase.

dealt with by estimating the distortion in the complex envelope instead of estimating the distortion of the original signals.

- The band width of the signals applied to the receiver's inputs should be very much greater than the band width of the receiver output to adequately probe the front end circuitry. Consequently, it is difficult to provide enough input samples so that enough information is available in the output for the estimation to be done. The problem can be alleviated by sampling the input data at slightly different rates. Different sampling rates complicates the processing of the data considerably, but the problem is tractable.

Sufficient experimental results are presented to verify that the methodology is capable of estimating the receiver's Volterra kernels. The results also verify some of the theoretical results in the thesis.

It is concluded that more efficient algorithms and/or more powerful computing hardware are necessary before the nonlinear filter approach is a practical way of estimating or correcting nonlinear distortion in systems similar to the HFR radar receiver.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any University, and to the best of the author's knowledge and belief contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

The author hereby consents to this thesis being made available for photocopying and for loans as the University deems fit, should the thesis be accepted for the award of the degree of Doctor of Philosophy.

Ian Dall

Acknowledgements

I would like to thank Professor Robert E. Bogner for his effective and conscientious supervision of this work. Professor Bogner displayed great patience and understanding, especially at times when I tended to become side tracked. His personal library, catalogued and cross referenced in his head, was of immense value. For almost any idea I came up with, he was able to come up with background reading, often from studies in apparently unrelated areas.

I would also like to acknowledge the financial support of the Australian Department of Defence through their "Research Scientist Cadetship" scheme. The High Frequency Radar Division (HFRD), Surveillance Research Laboratory (SRL), Defence Science and Technology Organisation (DSTO), provided technical assistance in the form of equipment and facilities for the experimental program. Thanks particularly to Mr. Lyndon Durbridge, Mr. Peter Kerr and Mr. Geoff Warne for their assistance in setting up and operating this equipment. Thanks also to Dr. Fred Earl, Dr. Malcolm Golley and Dr. Douglas Kewley for their advice and moral support. Dr. Kewley also proofread the manuscript which was much appreciated.

Thanks are due to Dr. Richard Stallman, Jim Kingdon and others from the Free Software Foundation (FSF). The software for the experimental work was written using *gcc* which is a C complier from the FSF. Extensive use was made of *gcc* extensions to the C language, which made the numerical code much

simpler[2]. Mention should also be made of *GNU emacs* and *gdb*, also from the FSF, which contributed substantially to productive software development. *GNU emacs* was used for preparing the manuscript as well as software development. David Gillespie's *calc* (a programable calculator package for *emacs*) proved valuable for verifying the correctness of C implementations of algorithms.

This thesis was prepared almost entirely using free software. Credit is due to Professor Donald Knuth (TEX), Dr. Leslie Lamport (LATEX), the MIT X Consortium (*X*), Brian Smith (major extensions to *xfig*), Thomas Williams and Colin Kelley (*gnuplot*), Micah Beck (*transfig*), Kevin Coombes (*dvi2ps*) and Olin Shivers (*cmutex*). Contributions to many of these programs were also made by others too numerous to mention. In many cases, the software needed modification to meet my requirements which would have been impossible had the source code not been made freely available by these authors.

Thanks to Mr. Michael Liebelt for cooperation in providing highly accessible computing facilities, without which it would have been much more difficult to install all the fore-mentioned software. Thanks also to Mr. Michael Pope for introducing me to *GNU emacs* and helping maintain and install many of the free software packages.

Thanks to Miss Brianna Ferguson for her understanding and emotional support, even when it seemed I had no time for anything except work and the list of things postponed until after this thesis was completed appeared to be growing without limit.

Credit is due to the FESOSA for preserving my sense of perspective and to the business dinner patrons who helped maintain my sanity. The University of Adelaide Underwater Hockey Team, Team Trevor and Crabs Canoe Polo teams maintained my physical health.

---

[2]Without these extensions, C is rather cumbersome when handling two (or higher) dimensional arrays.

Contents

Abstract i

Declaration iv

Acknowledgements v

1 Introduction 1

1.1 Definitions 3

1.1.1 Linearity 3

1.1.2 Bandpass Systems 3

1.1.3 Systems with Memory 4

1.2 Motivation 4

1.3 Techniques for Reducing or Correcting Nonlinear Distortion . . 6

1.3.1 Demanding Applications 6

1.3.2 Feedback 7

1.3.3 Feedforward 9

1.3.4 Pre or Post Distortion 10

1.3.5 Correction by Use of Nonlinear Digital Filters (Equalizers) 13

1.4 Outline 13

1.4.1 Signal Dependent Characterization 13

1.4.2 Parametric Characterization 14

1.4.3 Multi-input Systems 15

1.4.4 Nonlinear Digital Filters 15

1.4.5 HFR Experiments 15

1.5 Original Contributions 16

2 Signal Dependent Characterization **20**

2.1 Introduction 20

2.2 Harmonic Distortion 21

2.3 Intermodulation Distortion 26

2.4 Cross Modulation Distortion 30

2.5 Polyspectra 31

2.6 Summary 34

3 Parametric Characterization **36**

3.1 Introduction 36

3.2 Instantaneous Distortion 38

3.3 Amplitude and Phase Describing Functions 40

3.4 Volterra Series 42

3.5 The Wiener Model 45

3.6 Comparison of Models 49

3.7 Frequency Domain Analysis 50

3.8 Summary 55

4 Multi-input Systems **57**

4.1 Introduction 57

4.2 A Simplified Model Of Two Input Nonlinearities 59

4.3 The Wiener Model Of Multi-input Nonlinearities 61

4.3.1 The Two Input Wiener Model Applied to Mixers 62

4.3.2 Sinusoidal Local Oscillators 68

4.4 Phase Transfer 69

4.5 Summary 71

5 Nonlinear Digital Filters **74**

5.1 Introduction 74

5.2 Discrete Time Volterra Series 75

5.3 Nonlinear Adaptive filters 77
5.3.1 Nonlinear Prediction Estimation and Equalization 78
5.3.2 Gradient Algorithms 80
5.3.3 Direct Solution Of the Wiener-Hopf Equations 85
5.3.4 Recursive Least Squares Algorithms 87
5.4 Orthogonalization 93
5.5 Multi-input Nonlinear Filters 94
5.6 Implementation Considerations 96
5.6.1 Initialization 96
5.6.2 Software 96
5.7 Summary 100

6 HFR Receiver Experiments **102**
6.1 Introduction 102
6.2 The HFR Receiver and Its Operating Conditions 104
6.3 Design 105
6.3.1 Equipment 108
6.3.2 Test Signals 111
6.3.3 Chirp Generation by Phase Accumulation 112
6.3.4 Sampling Rates 115
6.4 Analysis Algorithms 120
6.4.1 Complex Envelope Processing 120
6.4.2 The Ideal Receiver and Sample Interpolation 122
6.4.3 Adaptive filter Algorithms 125
6.5 Results 128
6.5.1 Equipment and Procedure Verification 128
6.5.2 Nonlinear Adaptive Filter Verification 134
6.5.3 Estimation 142
6.6 Summary 145

7 Conclusion **149**

A Complex Envelope Equivalent Distortion. **155**

B Multidimensional Orthogonal Functions **158**

C Mixers with Sinusoidal Local Oscillators **161**

D Recursive Least Squares Algorithms **163**

D.1 The General Least Squares Problem 165

D.1.1 Properties of the Transition Matrices 170

D.2 Time Update Recursions 172

D.3 The Nonlinear Lattice Filter 176

D.4 Order Update Recursions 189

D.4.1 Forward Prediction Coefficient Update 193

D.4.2 Backward Prediction Coefficient Update 201

D.5 Error Update 206

D.6 Auxiliary Variables Update 207

D.7 Kalman Gain Update 208

D.8 Gamma Update 212

D.9 The Generalized Fast Kalman Algorithm 213

D.10 The Generalized LS Lattice Filter 215

Bibliography **222**

List of Figures

1.1 The bandpass context. 3
1.2 Feedback. 7
1.3 Feedforward. 9
1.4 A piece-wise linear distortion and its inverse. 12

2.1 Harmonic distortion. 25
2.2 IMD intercept point. 28

3.1 The bandpass context. 38
3.2 Amplitude and phase describing functions typical of a TWT. . . 41
3.3 The in-phase and quadrature model. 42
3.4 Samples of $x(t)$. 43
3.5 The single input Wiener model. 46
3.6 Frequency range of interest for a linear system. 51
3.7 A cube of interest in 3 dimensional frequency space. 52
3.8 A hypothetical system. 54

4.1 A simple model of a mixer. 60
4.2 The two input Wiener model. 64
4.3 A simple two input non-linearity. 71
4.4 Phase dependent distortion 72

5.1 The discrete time volterra series. 76
5.2 A nonlinear digital filter. 76
5.3 Estimation. 78
5.4 Equalization. 79

5.5 Prediction. 79
5.6 An error surface with a small eigenvalue spread. 83
5.7 An error surface with a large eigenvalue spread. 84
5.8 Organization of nonlinear terms into column vectors. 88
5.9 Organization of nonlinear terms into column vectors. 89
5.10 Reduced order filter "taps". 91
5.11 A two input nonlinear filter based on the Volterra series. 95

6.1 Experimental radar receiver block diagram. 105
6.2 Estimation of a two input receiver. 106
6.3 Approximate estimation of a two input receiver using a single input estimator. 107
6.4 Equalization of a two input receiver. 107
6.5 Equipment configuration for HFR receiver experiments. 109
6.6 VHF frequency translator. 110
6.7 Illustration of mixing of aliased components (common sampling rate). 116
6.8 Illustration of mixing of aliased components (different sampling rates). 118
6.9 Estimation of a two Input receiver 121
6.10 Illustration of ideal receiver simulation. 126
6.11 Measured output (doublet LO, triplet RF in). 129
6.12 Measured output (doublet LO, triplet RF in) omitting start up transient. 130
6.13 Simulated output (doublet LO, triplet RF in). 131
6.14 Measured output ("chirp" LO, noise RF in), omitting start up transient. 132
6.15 Measured output ("chirp" LO, noise RF in), omitting start up transient. 133
6.16 Measured output ("chirp" LO, noise RF in), omitting start up transient. 134

6.17 Simulated output ("chirp" LO, noise RF in). 135
6.18 Simulated output ("chirp" LO, noise RF in). 136
6.19 Simulated output ("chirp" LO, noise RF in). 137
6.20 Comparison of simulated and measured output ("chirp" LO, noise RF in). 138
6.21 Receiver gain ("chirp" LO, noise RF in). 139
6.22 Receiver gain (repeat)("chirp" LO, noise RF in). 140
6.23 Adaptive filter testing. 141
6.24 Estimation error due to rounding. 141
6.25 Estimation error vs number of samples. 142
6.26 Estimation error vs noise. 143
6.27 Estimation of a pure delay, $\lambda_{min}/\lambda_1 = 10^{-6}$ ("chirp" LO, noise RF in). 144
6.28 Estimation of a pure delay, $\lambda_{min}/\lambda_1 = 10^{-5}$ ("chirp" LO, noise RF in). 145
6.29 Estimation of the receiver (SVD). 146

D.1 Organization of nonlinear terms into unequal column vectors. . . 164
D.2 Organization of linear terms into unequal column vectors. 164
D.3 The lattice filter. 178
D.4 The generalized lattice filter. 219
D.5 Reduced generalized lattice filter. 221

List of Tables

2.1 Third order cross modulation terms. 31

6.1 Test signal parameters. 113

6.2 Sampling parameters. 120

D.1 Dimensions of variables. 167

D.2 $\hat{\boldsymbol{T}}$ example. 220

Chapter 1

Introduction

As signals are processed by any real equipment, they suffer degradation. The degradation can be classified as noise or distortion, although the situation is complicated by noise which also undergoes distortion. Distortion is a deterministic process which acts on the input signal(s), whereas noise is an extraneous signal[1].

Distortion is such a broad term that little can be said about it unless the term is qualified. A general distorting system could literally have *any* output. A major class of distortion is *linear* distortion. Linear distortion includes gain and phase errors. Nonlinear distortion covers everything else. Nonlinear distortion is still an extremely general term. Indeed, since a nonlinear system might differ only infinitesimally from a linear system, it can still be said that nonlinear systems could have *any* output. In a sense, linear systems can be considered a special case of nonlinear systems. A linear system is defined in section 1.1.1. Nonlinear are systems which lack this property. It is not very helpful to define a class of systems in terms of the properties they lack! To be able to say

[1] "Noise" is sometimes used to imply a degree of randomness, but "Random Noise" is a more correct term in this case.

anything useful about the behaviour of nonlinear systems, interesting classes of nonlinearity must be identified.

One approach is to consider only a class of distortion based on a particular mathematical form, most commonly the power series. Much of the traditional work in nonlinear distortion considers the effect of that distortion on a narrow class of signals (typically sinusoids). However, the power series is not capable of describing nonlinear distortion with memory (see section 1.1.3 for a definition).

The approach taken in this thesis is to concentrate on nonlinearities in bandpass systems (see section 1.1.2 for a definition). Many practical systems can be considered to be bandpass. Virtually all communications, radar, sonar and electronic warfare systems contain bandpass subsystems. Apart from the bandpass restriction, the class of nonlinearities considered has been kept as broad as possible. In particular, nonlinearities with memory *are* considered.

The experimental work in this thesis has actually been done on an experimental HF Radar Receiver. However, there is little in this work which is not applicable more widely. Perhaps the most common examples of bandpass systems are in communications.

This chapter will describe the layout of the rest of the thesis, give an overview of previous related work and set the context for the authors work. More specific discussion of previous work is in separate chapters (particularly chapter 2 and parts of chapters 3 and 5).

1.1 Definitions

1.1.1 Linearity

A linear system is defined as follows: Define y_1 and y_2 to be the outputs for inputs of x_1 and x_2 respectively. The inputs and outputs are typically functions of time but might also be functions of other independent variables. If the output is $ay_1 + by_2$ for an input of $ax_1 + bx_2$, for any x_1, x_2, and any constant a and b then the system is linear.

1.1.2 Bandpass Systems

A nonlinear system in a bandpass context can be considered to be preceded and followed by ideal bandpass filters. In figure 1.1, the inputs of system S is guaranteed to have no components outside a certain band. The output, y, may have components outside the passband but they can be neglected since they will not effect z.

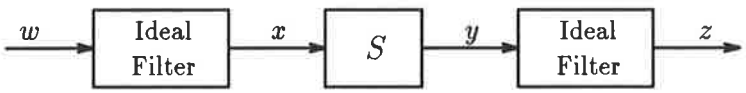


Figure 1.1: The bandpass context.

The filters may be there explicitly or implicitly. For example, the signal w might always be band limited eliminating the need for the input filters. Similarly, in an audio system, if high frequencies are inaudible then there may be no need to explicitly filter them out to consider a system bandpass.

In chapter 4 we consider two input bandpass systems. The definition is

essentially the same as for single input bandpass systems except there is an implicit filter on *each* of the inputs.

1.1.3 Systems with Memory

Nonlinear distortion can be classified as being with or without memory. A memoryless system is one whose output $y(t)$ at any time t is a function of its input $x(t)$ at that same instant. Generally a pure delay can be ignored so that a system $y(t) = f(x(t-T))$ can usually be treated as though it were memoryless. A system with memory is one whose output at any instant is dependent on the input at some instants other than the present. Some writers refer to systems whose memory is short compared to the period of the modulation as being memoryless, which can create confusion[2]. In this thesis we will refer to systems with short memory as such, reserving the terms "memoryless" and "zero-memory" to describe systems whose output can be validly considered to be a function only of the input at that instant.

1.2 Motivation

The study of nonlinear systems has a long history, but they are none the less much less well under stood than linear systems. This is partly because they are inherently more difficult to understand and partly because they have not received such intense attention. In much of engineering practice the assumption of linearity is a good one. Further, the linearity property is a powerful one in terms of the mathematical manipulations it allows, so there is a strong temptation to coerce systems into a linear form to make the analysis tractable.

[2]Note that a truly memoryless system has no phase shift so it is not correct to describe a TWT, which has amplitude dependent phase shift as "memoryless".

None the less, there are circumstances where nonlinear characteristics cannot be ignored. In most systems, there is a requirement that the energy output due to nonlinear distortion be below some level[3]. With complex inputs, predicting the output distortion level is not a trivial exercise. Except for specific inputs it is not simple to distinguish the nonlinear components of the output from the linear components, which is necessary to *measure* nonlinear distortion. One approach to determining the distortion output, is to first produce a *model* and then use it to simulate the output. This is most useful when it is difficult or impractical to apply experimental inputs to the original system. Some models might also be of scientific interest in their own right, especially if they provide insights into natural phenomena. Finally, if the level of nonlinear distortion components in the output of a system is too high, then steps can be taken to *correct* the distortion. We have identified three areas, measurement, modelling and correction, of nonlinear engineering systems which require understanding. This thesis contributes to the understanding of modelling and correction. The measurement problem is discussed in greater detail in chapter 2.

The primary motivation for the work in this thesis was modelling and correcting distortion in High Frequency Radar (HFR) receivers. The method used has so far proven to be impractical for the particular case of the HFR Receiver because of the amount of computation required. Investigating more efficient algorithms is a suggested topic for further research. However, a number of theoretical results were developed to validate the experimental approach taken. These results are significant in their own right. See section 1.5 for a more detailed statement of the contributions made in this thesis.

[3]There may also be a requirement on the form that the distortion might take (such as spectral distribution, energy in a certain band etc).

1.3 Techniques for Reducing or Correcting Nonlinear Distortion

There are a number of known methods for reducing nonlinear distortion. The simplest is to operate on a more linear portion of the devices characteristic. Usually that implies operating at a lower power level[4]. This technique is commonly used for microwave Travelling Wave Tube (TWT) amplifiers where it is known as "back off".

1.3.1 Demanding Applications

There are many applications where the linearity of systems is a matter for concern. Two examples are TWT amplifiers as used in satellite repeaters and HFR receivers.

"Backing off," a TWT implies that the device is over specified in terms of power handling capability compared to the case where nonlinear distortion need not be considered. In the case of satellite repeaters, the weight of a larger amplifier and its accompanying power supply (such as batteries and solar panels) is a significant incentive to find other methods of reducing nonlinear distortion. As more sophisticated modulation schemes are used, the permissible level of nonlinear distortion goes down.

HFR receivers are another demanding application. Because HF radar operates at a frequency where ionospheric propagation is good, it is subject to interference from distant HF broadcast stations. Also, like any radar, it may be subject to deliberate interference (jamming). With this high level of in-

[4] "Cross-over" distortion is an exception where operating at a lower level *increases* the distortion.

terference, it is desirable that nonlinear distortion products be smaller than the external noise floor so that the minimum detectable cross section is not degraded.

1.3.2 Feedback

Probably the most widely used method of reducing distortion is to apply feedback. H.S. Black [14] is generally credited with the invention of feedback. Although feedback control systems in the form of governors for steam engines were already in existence, Black was apparently the first to identify feedback as a general technique and analyze it. The potential for feedback to reduce distortion (in addition to its other features) was recognized in Black's original paper.

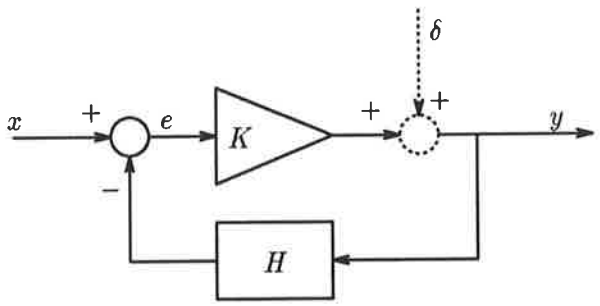


Figure 1.2: Feedback.

Consider the feedback system in figure 1.2. Assuming that the feedback is effective, the output level y, is held substantially constant for variations in the parameter K. If the distortion is caused primarily by the output stage of the amplifier, then the distortion can be considered to be injected at the output and, to a good approximation, constant. The distortion is represented by δ in figure 1.2.

To see the distortion reducing effect of feedback, we need only work out the

transfer function between the input δ and the output y which is

$$y = \frac{\delta}{1 + KH}. \tag{1.1}$$

The feedback concept contains a causal contradiction. It is an attempt to correct errors after they have happened. This is only effective if the signal e is essentially unchanged during the time taken for the corresponding feedback signal to be generated. The applicability of feedback is therefore constrained by the band width of the loop gain KH, the band width of the input signal and delay in KH.

There is, of course, a large body of theoretical and practical knowledge pertaining to the stability of feedback networks and it is not appropriate to discuss it here in any detail (some of the significant early advances are documented in [14] and [16]). Note, however, that the wider the band width and the greater the delay, the more difficult it is to ensure that the feedback network is stable. The TWT is a particularly extreme example of a device for which a feedback network is not usable. Seidel [70] gives approximately 19,000 electrical degrees as the delay of a particular TWT.

In the Laplace domain, consider the case of $KH = 10^7/(50 + s)$ which is representative of a low cost operational amplifier. From equation 1.1, the output distortion is approximated by

$$Y(s) \approx \frac{\delta(s)(50 + s)}{10^7 + s}. \tag{1.2}$$

An interesting feature of this example is that even if $\delta(t)$ is due to a "zero-memory" distortion, the feedback network as a whole can no longer be considered to be "zero-memory". Also, there are phase variations due to the zero in equation 1.2 which would normally be well within the closed loop passband.

1.3.3 Feedforward

Feedforward is a technique for reducing distortion by detecting errors amplifying them and injecting them into an appropriately delayed signal in such a way as to cancel the errors. See figure 1.3. Seidel [70] cites Black [13] as inventing feedforward in 1924 prior to his invention of feedback. However, the practical implementation of feedforward networks appears to be pioneered by Seidel [71] [70].

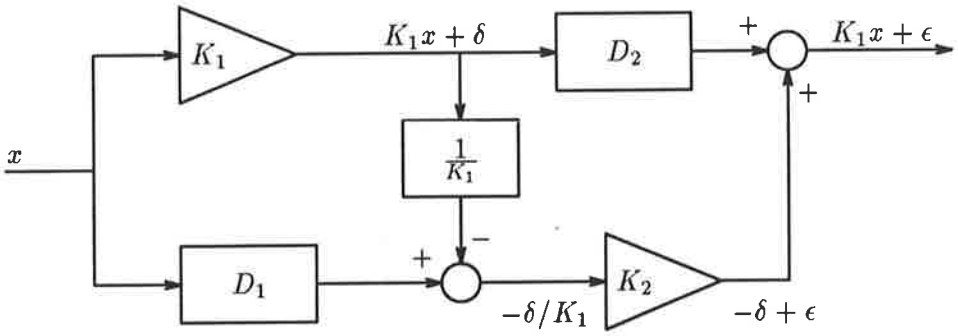


Figure 1.3: Feedforward.

Unlike the case of feedback, there is no causal contradiction in the concept of feedforward. In figure 1.3, the time taken for the signal to transit amplifiers K_1 and K_2 are explicitly matched by the delays D_1 and D_2. Assume that the amplifiers are consist of a pure gain, a pure delay, and an additive error term. In the case of amplifier K_1 the error is δ. In the case of amplifier K_2 the error in the output ϵ includes mismatch between the nominally equal gains K_1 and K_2.

Provided the individual amplifiers are stable, the feedforward network is always stable. For the feedforward configuration of figure 1.3 to succeed in reducing distortion, the second amplifier need only have sufficiently low distortion that ϵ be very much less than δ. It is also necessary that the gain K_2 precisely match the attenuation $1/K_1$ and that the delay D_2 precisely match the delay

in the amplifier K_2. The delay D_1 need not match the delay in amplifier K_1 as closely, but if the match is poor, the amplifier K_2 will have to handle a larger signal than would otherwise be necessary making it more difficult to meet the requirement that ϵ be very much less than δ.

The chief difficulty with feedforward is the difficulty of matching the delays D_1 and D_2 to the delays in the amplifiers K_1 and K_2. In Seidel's work [70], he used servo controlled electro-mechanical gain and phase controllers. The servo control loops added significant complexity to the system. If the amplifier delay is a function of frequency, then the problem of matching the delays in the paths of the feedforward network becomes even more difficult.

1.3.4 Pre or Post Distortion

Both feedforward and feedback have problems when applied to broad band high frequency applications. In the case of feedback, the problem is one of maintaining stability and in the case of feedforward the problem is one of matching an artificial delay to the amplifier characteristics.

There are also situations where neither method is applicable. One such example is for the correction of distortion in mixers where the output is frequency shifted from the input. Another situation where neither feedback nor feedforward is applicable is where the input and outputs to the distortion are not both accessible. For example, it is not possible to use feedback to correct the distortion in a communication link whether the distortion be due to an existing repeater, or the transmission medium. Typically, one might be presented the problem of minimizing distortion over an existing satellite link. Adding feedforward to existing satellite TWTs is unlikely to be feasible.

In these cases, it may still be possible to correct for distortion by predistorting or post-distorting the signal in such a way as to reduce the over-all

distortion. A simple example of such a pre-distortion is shown in figure 1.4.

Schetzen [66], shows that it is always possible to invert distortion represented as a Volterra series up to the pth order. If the inverse is stable as $p \rightarrow \infty$, then the inverse is exact and the pre-inverse and the post-inverse are the same.

A number reports appear in the literature of attempts to linearize TWT amplifiers by pre-distortion or post-distortion [20] [37] [64] [36] [65]. Most of these correction schemes assume that an adequate corrector can be represented by amplitude and phase describing functions (discussed in section 3.3).

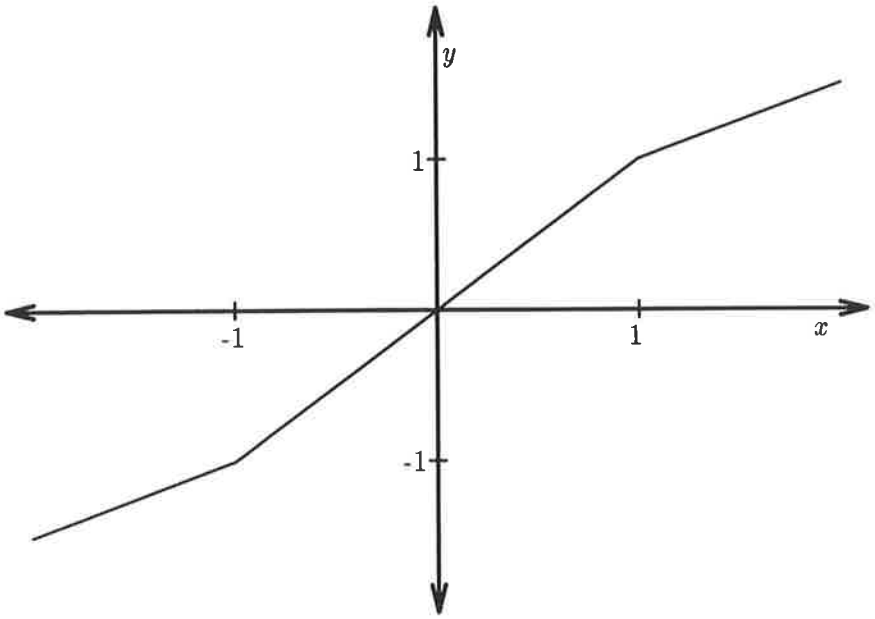
The distortion in figure 1.4 is of the form

$$y = \begin{cases} x, & \text{for } -1 < x < 1; \\ \frac{1}{2}(x+1), & \text{for } x \geq 1; \\ \frac{1}{2}(x-1), & \text{for } x \leq -1; \end{cases} \tag{1.3}$$

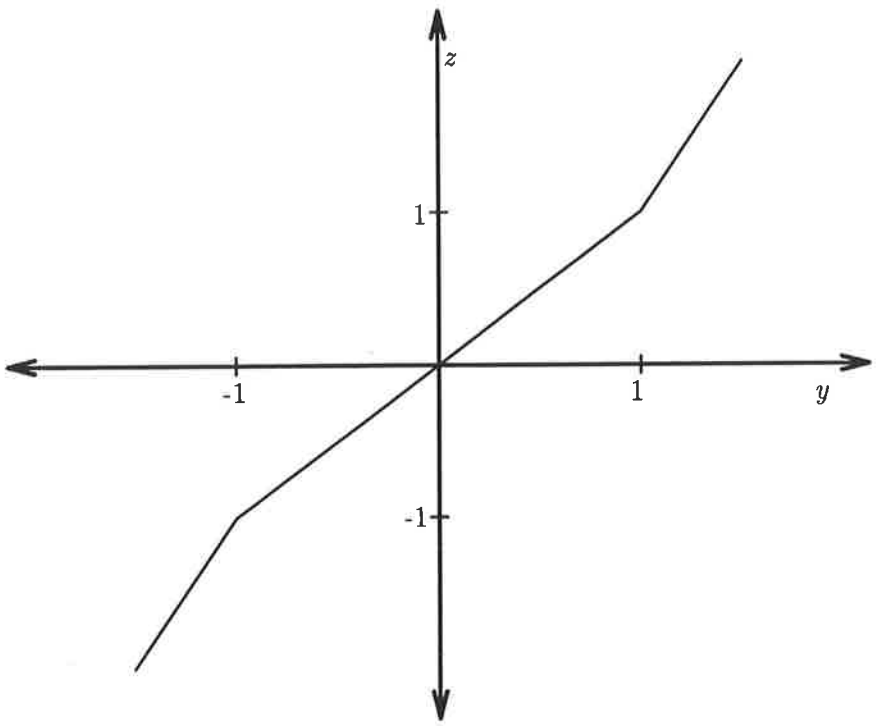
and its inverse is clearly given by

$$z = \begin{cases} y, & \text{for } -1 < y < 1; \\ 2y-1, & \text{for } y \geq 1; \\ 2y-1, & \text{for } y \leq -1. \end{cases} \tag{1.4}$$

In this case the correction is exact and the pre-inverse is exactly the same as the post inverse. This is not always the case. Some nonlinearities do not have an inverse. For example the inverse of $y = x^2$ does not exist because $z = x^{1/2}$ is multi-valued. However, even this case can be inverted if x is constrained to be only positive or only negative. Even where a distortion does not have an exact inverse, there is still a best approximation to the inverse which can be found. The optimum pre-inverse and the optimum post inverse are not necessarily the same [66].



Distortion



Inverse Distortion

Figure 1.4: A piece-wise linear distortion and its inverse.

1.3.5 Correction by Use of Nonlinear Digital Filters (Equalizers)

The method of pre-distortion or post-distortion mentioned in section 1.3.4 raises the question of just how can the corrector be implemented. One method which is capable of optimally correcting nonlinear distortion up to a given order is a nonlinear digital filter based on the discrete time Volterra Series. This method is discussed extensively in chapter 5. The problem of setting the coefficients of such a filter can be handled by adaptive techniques. The implementation and use of nonlinear adaptive filters is a major part of this thesis. As well as chapter 5 see also sections 6.4.3 and 6.5.2.

1.4 Outline

1.4.1 Signal Dependent Characterization

Chapter 2 discusses the way distortion can be characterized in terms of the distortion products which occur when specific inputs are used. Linear systems have the property that there are no components in the output at frequencies which were not present in the input. If the input consists of the sum of a few sine waves, then any outputs at other frequencies must be due to nonlinear distortion. Several traditional measures of distortion which take advantage of this property are discussed. These are "Harmonic Distortion", "Intermodulation Distortion" and "Cross Modulation Distortion".

It is observed that these techniques are certainly measures of distortion, but do not, in general, characterize distortion sufficiently well to enable the output to be estimated when a more complex input is applied. Nor do these measures contain enough information to allow a corrector to be constructed.

Section 2.5 discusses the application of polyspectra to the observation of nonlinear distortion products when the input is a Gaussian distributed random variable. This technique gives a better estimate of how a nonlinear system will behave with complex inputs.

1.4.2 Parametric Characterization

Chapter 3 discusses models of nonlinear distortion. The signal dependent characterization discussed in chapter 2 are primarily *measures* of distortion, and are not appropriate for modelling the output of a nonlinear system with a given input. The measures of distortion can, at least in principle, be used to estimate the parameters of a model. In chapter 3, bandpass systems are treated by using a complex envelope represention of the signals.

Section 3.2 discusses instantaneous (memoryless) distortion which can be represented by a series of functions, most commonly the power series. Section 3.3 discusses the amplitude and phase describing function distortion model which is commonly used to describe distortion in TWTs. Section 3.4 introduces the Volterra series, a general way of modelling distortion with memory. Section 3.5 introduces the Wiener model which separates a general nonlinearity with memory into a linear, multi-output, linear section followed by a multi-input, zero-memory, nonlinear section. The power series model, the amplitude and phase describing function model and the Volterra series model are then compared by expressing the former two models as special cases of the Volterra series.

Section 3.7 provides insights into the nature of bandpass systems by considering the Fourier transforms of the Volterra kernels. The frequency domain representation of the power series, amplitude and phase describing functions and the general Volterra series are considered.

1.4.3 Multi-input Systems

The signal in high performance receivers typically goes through two or three mixers which perform frequency conversion operations. Further, in many modern communication systems the mixer does not perform a simple frequency conversion. In radar systems the local oscillator might be a "chirp" (linear frequency sweep) signal. Systems such as this must, in general, be modelled as multi-input systems.

Chapter 4 addresses the important question of when a two input nonlinearity can be represented by a more simple model containing only single input nonlinearities. Section 4.4 shows that changing the phase of a local oscillator in a bandpass system changes the phase of all output components equally. This is important and desirable behaviour, but it by no means an obvious result.

1.4.4 Nonlinear Digital Filters

Chapter 5 defines the discrete time Volterra series and shows how it can be put in the form of a nonlinear digital filter. Section 5.3 shows how an adaptive digital filter can solve the difficult problem of estimating the coefficients of a nonlinear filter. Applications of adaptive nonlinear digital filters as predictors, estimators and correctors are also discussed. Sections 5.3.2–5.3.4 discuss algorithms for estimating the coefficients of an adaptive nonlinear filter. The advantages and disadvantages of orthogonalization are discussed in section 5.4.

1.4.5 HFR Experiments

Chapter 6 describes experiments performed on HFR receivers which presented some difficulties in experiment design. Section 6.3 describes the design of the

experiments in terms of equipment, test signal selection and generation and choice of sampling rates. Section 6.4 describes the techniques used to analyze the results including calculation of complex envelopes, interpolation to common sampling rate and the adaptive filter algorithms used for the estimation. Section 6.5 describes the results obtained including steps taken to confirm that the hardware and software were all working correctly.

1.5 Original Contributions

To the best of the author's knowledge, this thesis makes original contribution in the following areas.

In chapter 3.2 it is demonstrated that a power series distortion of a bandpass signal is equivalent to distortion of its complex envelope. This result doesn't appear explicitly in the literature, but it is a special case of a result by Benedetto et al [8]. In section 3.6, the Volterra kernels corresponding to the amplitude and phase describing function model are derived and in section 3.7, by consideration of the Fourier transform of those kernels, it is shown that amplitude and phase describing functions are a valid representation of any system if the band width is narrow enough. It is also shown that frequency independence of the amplitude and phase describing functions, as measured by a single sinusoid, is a necessary, but not sufficient, condition for the amplitude and phase describing function model to be applicable.

In chapter 4, a two input version of the Wiener model is proposed and it is used to treat a two input nonlinearity as a multi-variable polynomial of the output of a number of narrowband filters. This model is used to eliminate components outside the passband and hence it is shown;

- That a mixer with a sinusoidal[5] input can be modelled as an ideal mixer cascaded with a single input single output nonlinearity. The nonlinearity, is however a function of the local oscillator amplitude.
- That a mixer with a broad band local oscillator input can not, *in general,* be modelled as an ideal mixer cascaded with a single input single output nonlinearity.
- That a phase shift in a local oscillator results in a constant phase shift of all components within the passband. This is not true when the system is not bandpass.

In section 5.3.4 (and appendix D which is referenced from section 5.3.4), a nonlinear filter operating on a scalar is represented as a linear filter operating on vector quantities $\boldsymbol{\xi}_i$ which vary in length. This allows the development of the nonlinear RLS adaptive lattice filter algorithm and the nonlinear Fast Kalman algorithm.

As part of the development of the nonlinear RLS algorithms a number of relationships which are well known for linear systems had to be re-derived for the "variable length vector" case. These are as follows:

- Derivation of the Wiener-Hopf equations.
- Derivation of the definition of the Kalman gain.
- Demonstration that there is a (non-adaptive) lattice filter which is the equivalent of any FIR filter.
- Derivation of an algorithm for converting between the lattice filter and the transversal filter.

[5]At least the component within the passband must be sinusoidal. Harmonic distortion of the local oscillator input has no effect.

- For a stationary process, derivation of the relationship between the optimum forward and backward predictor coefficients.
- Demonstration that the optimum forward and backward prediction errors form an orthogonal basis for the space spanned by the variable length vectors $\boldsymbol{\xi}_i$.
- For a stationary process, derivation of the relationship between the forward and backward partial correlation coefficients $\boldsymbol{K}_m^f = \boldsymbol{K}_m^{b\,*}$.

In chapter 6, an experimental procedure is developed for nonlinear estimation of an HFR Receiver. There are several characteristics of the HFR Receiver which required the development of novel techniques. Specifically:

- The receiver must be modelled as a two input device. The phase of the local oscillator input, relative to the RF input, can not be measured or controlled[6].
- The receiver has slight, but important, nonlinear distortion in the presence of large linear distortion. The linear distortion is inherent in the high frequency selectivity of the receiver and implies a long impulse response. These factors place severe constraints on the adaption algorithms, which prompted the development of the nonlinear RLS algorithms mentioned above.
- The receiver's inputs and outputs are at different frequencies. The solution of this problem by estimating the distortion in the complex envelope instead of estimating the distortion of the original signals is not new [8].
- The band width of the signals applied to the receiver's inputs should be very much greater than the band width of the receiver output to adequately probe the front end circuitry. Consequently, it is difficult to provide enough input samples so that enough information is available in the

[6] All the signals are phase locked, but there is no way to set the initial phase.

output for the estimation to be done. The problem can be alleviated by sampling the data for the two inputs at slightly different sampling rates.

Experimental evidence is presented to support the theoretical result derived in section 4.4 where where it was shown that a constant phase shift of a local oscillator, results in a constant phase shift of all the components (linear and nonlinear) in the passband.

Chapter 2

Signal Dependent Characterization

2.1 Introduction

Traditionally, distortion has often been characterized by its effect on an input consisting of one or more sinusoids. This leads to terms such as "Harmonic Distortion" and "Intermodulation Distortion". These are indeed measures of distortion, but they do not in themselves constitute identification of a nonlinear system. For example, it is not possible to know what the output of a system with $x\%$ harmonic distortion will be for any input other than a sine wave of the same amplitude and frequency as the original test. However, with enough such measurements, it is possible under certain constraints, to derive an estimation of a nonlinear system. The more general the model, the more measurements are required. If one has an adequate parametric characterization (a model), it is relatively easy to predict the signal dependent characterization. Unfortunately the reverse is not generally true.

The distortion characterizations based on sinusoids (sections 2.2–2.4) are

quite old and are covered here chiefly as a contrast to the Parametric Models introduced in chapter 3. Signal dependent characterization (notably Intermodulation Distortion and Harmonic Distortion) is used by most manufacturers in specifying the linearity of their devices. The application of polyspectra (section 2.5) to the characterization of nonlinear systems is much newer and, to the authors knowledge, no manufacturers specify linearity in terms of polyspectra, although it might make sense to do so.

To understand the signal dependent characterizations of sections 2.2–2.4, we will examine the output of a power series with various inputs. The power series is the simplest commonly used parametric model of distortion. To an extent we are pre-empting chapter 3, but the power series is simple enough to need little introduction.

2.2 Harmonic Distortion

It is not clear when harmonic distortion was first theoretically predicted. Since second harmonic distortion arises fairly directly from the "double angle" trigonometric identities the concept is undoubtedly quite old. Richards [58] traces the theory of *sub*-harmonics (which only arise in systems with memory) to Mathieu in 1868[1], so it would be surprising if the generation of harmonics where not understood well before this. Some treatment of harmonic distortion can be found in text books by Panter [55] and Calson [19].

There are several approaches which can be taken to demonstrate the generation of harmonics by a nonlinear system. First, consider a power series

$$y(t) = \sum_{i=1}^{n} a_i x(t)^i. \tag{2.1}$$

[1]Sub-harmonics where *observed* as early as 1831 by Faraday.

Now we can represent a sinusoidal input by

$$x(t) = \cos(\omega_0 t), \tag{2.2}$$

and by the use of trigonometric identities we can show the production of harmonics. For example, in the case of the term $n = 2$ in equation 2.1,

$$x(t)^2 = \frac{1}{2} + \frac{1}{2}\cos(2\omega_0 t). \tag{2.3}$$

Alternatively, we can represent the sinusoidal input as a sum of complex exponentials,

$$\begin{aligned} x(t) &= A\cos(\omega_0 t + \phi_0) \\ &= \frac{A}{2}\left(e^{j\omega_0 t+\phi_0} + e^{-j\omega_0 t-\phi_0}\right). \end{aligned} \tag{2.4}$$

This tends to be easier to handle, especially for higher powers of n or when $\phi_0 \neq 0$.

$$\begin{aligned} x(t)^n &= \left(\frac{A}{2}\right)^n \left(e^{j\omega_0 t+\phi_0} + e^{-j\omega_0 t-\phi_0}\right)^n \\ &= \left(\frac{A}{2}\right)^n \sum_{i=0}^{n} \binom{n}{i} e^{j(n-i)(\omega_0 t+\phi_0)} e^{-ji(\omega_0 t+\phi_0)} \\ &= \left(\frac{A}{2}\right)^n \sum_{i=0}^{n} \binom{n}{i} e^{j(n-2i)(\omega_0 t+\phi_0)}. \end{aligned} \tag{2.5}$$

If we now note that the ith term has the same magnitude centre frequency as the lth term if $n - 2i = -(n - 2l)$ i.e. $l = n - i$, then we can group the terms in equation 2.5 in pairs. The expansion now takes a slightly different form if n is even or odd. For n odd

$$x(t)^n = \left(\frac{A}{2}\right)^n \sum_{i=0}^{(n-1)/2} \left[\binom{n}{i} e^{j(n-2i)(\omega_0 t+\phi_0)} + \binom{n}{n-i} e^{-j(n-2i)(\omega_0 t+\phi_0)}\right] \tag{2.6}$$

and for n even

$$\begin{aligned} x(t)^n = \left(\frac{A}{2}\right)^n \binom{n}{n/2} + \left(\frac{A}{2}\right)^n \sum_{i=0}^{(n/2)-1} \Bigg[\binom{n}{i} e^{j(n-2i)(\omega_0 t+\phi_0)} \\ + \binom{n}{n-i} e^{-j(n-2i)(\omega_0 t+\phi_0)}\Bigg]. \end{aligned} \tag{2.7}$$

Now we use the fact that $\binom{n}{i} = \binom{n}{n-i}$ to simplify equations 2.6 and 2.7. For n odd,

$$x(t)^n = \left(\frac{A}{2}\right)^n \sum_{i=0}^{(n-1)/2} \binom{n}{i} \cos\left((n-2i)(\omega_0 t + \phi_0)\right), \tag{2.8}$$

and for n even,

$$x(t)^n = \left(\frac{A}{2}\right)^n \binom{n}{n/2} + \left(\frac{A}{2}\right)^n \sum_{i=0}^{(n/2)-1} \binom{n}{i} \cos\left((n-2i)(\omega_0 t + \phi_0)\right). \tag{2.9}$$

It is worth observing that a common short cut when dealing with linear systems is to represent a sinusoid by $e^{j\omega_0 t+\phi_0}$, effectively forgetting about the negative frequency component of equation 2.4. If the signal is to go through a nonlinear process, this simpler representation does not give the correct result. Instead of equation 2.5 we would simply have $x(t)^n = \Re e^{jn(\omega_0 t+\phi_0)}$ which would correctly predict only one of the components in equations 2.8 and 2.9.

Yet a third method of determining the components of a power series of a sinusoid is to perform the manipulations in the frequency domain. We note that multiplication in the time domain is equivalent to convolution in the frequency domain. Also, a sinusoid has a very simple representation in the frequency domain as two delta functions. The final factor which makes this approach attractive is that the convolution of a function with a delta function is particularly easy to do,

$$\mathbf{X}(\omega) * \delta(\omega - \omega_0) = \mathbf{X}(\omega - \omega_0). \tag{2.10}$$

The method is less attractive when an arbitrary phase shift must be incorporated. If

$$x(t) = \cos(\omega_0 t)$$

then

$$\begin{aligned} \mathbf{X}(\omega) &= \frac{\pi}{2}(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)) \\ \mathcal{F}(x(t)^2) &= X(\omega) * X(\omega) \\ &= \left(\frac{\pi}{2}\right)^2 (\delta(\omega - 2\omega_0) + 2\delta(\omega) + \delta(\omega + 2\omega_0)) \end{aligned}$$

and higher orders can be handled by recursion

$$\mathcal{F}(x(t)^n) = \mathcal{F}(x(t)^{n-1}) * \mathbf{X}(\omega) \tag{2.11}$$

This method is intuitively satisfying, especially to someone with a familiarity with frequency translation processes. First, start with the spectrum of the original sine wave, then translate up and down by ω_0 to get the second order components. Then translate up and down again by ω_0 to get the third order components etc. This process is illustrated in figure 2.1.

Some observations regarding harmonic distortion are in order. From equations 2.8 and 2.9 note that a DC (zero frequency) component is produced only by even order distortion. Note that for n even, only even order harmonics are produced and for n odd, only odd order harmonics are produced. The highest harmonic produced by nth order distortion is n. Note that a high order odd(even) distortion produces harmonics at the same frequency as lower order odd(even) distortion. Finally, note that the magnitude of the nth order distortion components are proportional to A^n where A is the magnitude of the input signal. In section 3.4 we will generalize this result to systems with memory, which can not be represented by a power series.

As a consequence of the above observations, it is a non-trivial problem to determine the power series parameters from the harmonic distortion measurements. In principle, one could start from the highest order harmonic present, work out the corresponding power series coefficient, then determine the magnitude of the $(n-1)$th harmonic due to the nth order distortion, subtract that from the total $(n-1)$th harmonics to leave only the part due to the $(n-1)$th order distortion, and so on until the 1st order coefficient has been determined. The problem is that there in not, in general, a clearly defined maximum order distortion in most real systems and the process is subject to the accumulation of errors.

The term "Harmonic Distortion" is usually reserved for the production of nth order harmonics for $n > 1$. Note that the fundamental has also been

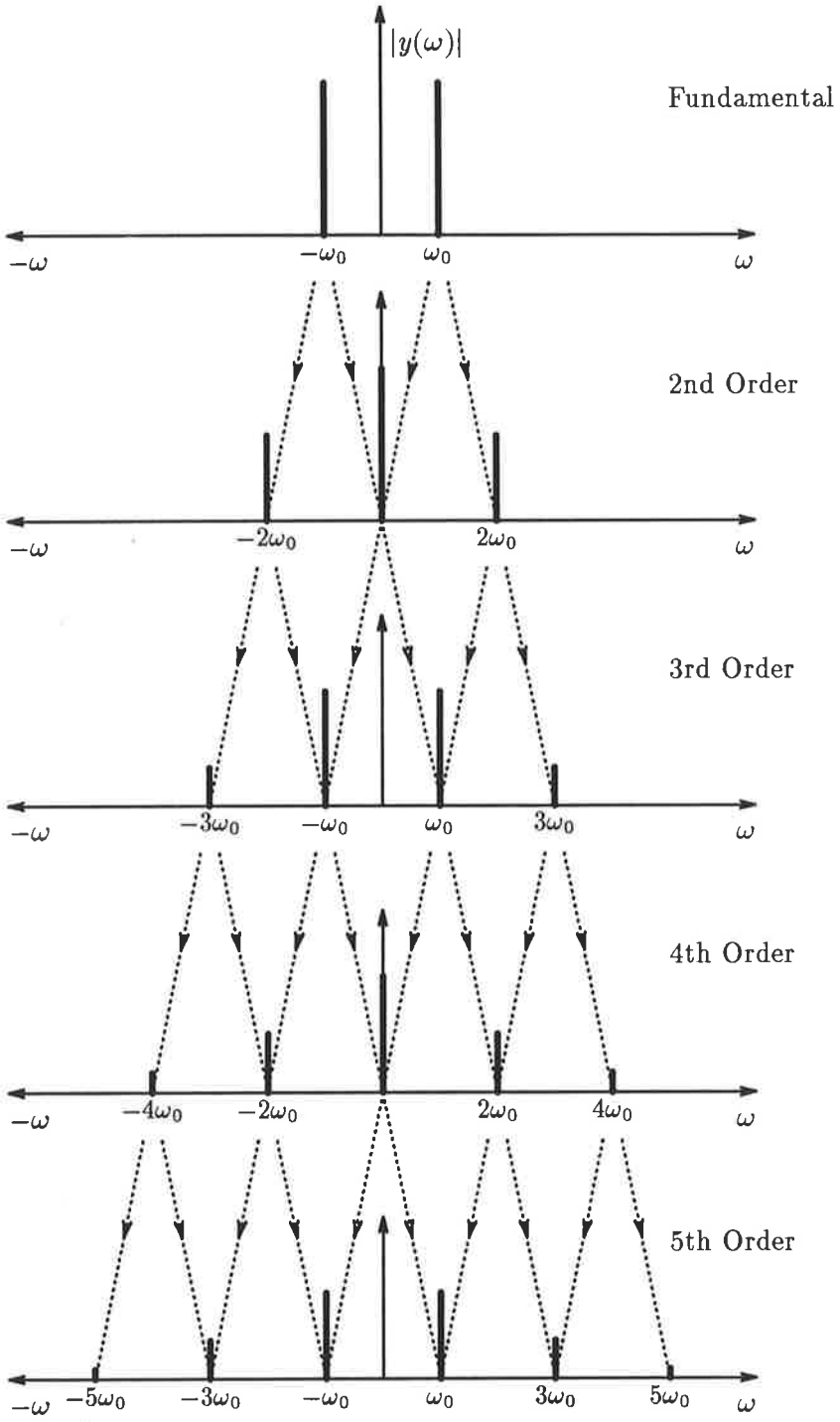


Figure 2.1: Harmonic distortion. Observe how the higher order harmonics can be derived from translations of lower order harmonics.

distorted because higher odd order distortion produces components at the fundamental frequency. The amplitude of the fundamental is no longer linearly related to to the amplitude of the input. This is one form of "in-band" distortion, a term which arises when considering bandpass systems where higher order harmonics are outside the systems passband.

This nonlinear relation between the input amplitude and the amplitude of the fundamental, or the amplitude of a higher order harmonic, can be exploited to identify the coefficients of the power series [53][12].

2.3 Intermodulation Distortion

We saw, in the case of harmonic distortion, that we could view harmonic distortion as a sine wave translated n times by it's own frequency. The method is summarized equations 2.10–2.11 and figure 2.1. When the input consists of components at more than one frequency, we have the case where components at one frequency are translated by a different frequency resulting in components at sums and differences of the original frequencies. These components are known as Intermodulation Distortion (IMD) products.

It is difficult to trace the first theoretical prediction of the phenomenon of intermodulation distortion. Like harmonic distortion, the existence of intermodulation distortion can be predicted fairly directly from basic trigonometric identities. Certainly intermodulation distortion was of concern by 1934 when Black wrote his paper on feedback [14], since he cites reduction of "modulation products" in multi-channel telephone systems as a major feature of feedback. Intermodulation distortion is treated in some depth in a text book by Panter [56].

To see how intermodulation terms arise, consider the case of a two tone

input into a system described by the same power series (equation 2.1). If the input is

$$x(t) = A\left(\cos(\omega_1 t) + \cos(\omega_2 t)\right) \tag{2.12}$$

then the nth term in the power series is given by

$$x(t)^n = A^n \sum_{i=0}^{n} \binom{n}{i} \cos^{n-i}(\omega_1 t) \cos^i(\omega_2 t). \tag{2.13}$$

In the treatment of harmonic distortion (section 2.2), we derived expansions for $\cos^n(\omega_0 t)$ which can be substituted into equation 2.13. However, the expression becomes complicated and does not provide great insight. Instead consider the particular case of $n = 3$,

$$\begin{aligned} x(t)^3 = A^3\Big(&\cos^3(\omega_1 t) + 3\cos^2(\omega_1 t)\cos^1(\omega_2 t) \\ &+ 3\cos^1(\omega_1 t)\cos^2(\omega_2 t) + \cos^3(\omega_2 t)\Big). \end{aligned} \tag{2.14}$$

The $\cos^3$ terms give rise to harmonic distortion but the remaining terms give rise to Intermodulation Distortion components at $\omega = 2\omega_1 - \omega_2$ and $\omega = 2\omega_2 - \omega_1$. Since equation 2.13 contains terms which correspond to both harmonic distortion and intermodulation distortion, it is apparent that the occurance of these components is more a feature of the input than of the nonlinear system. All the components of equation 2.14 increase in proportion to A^3. Calculation of all the IMD terms for high orders or for many input tones can be readily performed numerically [69].

As with the case of harmonic distortion, different order IMD components may have the same frequency. For example, there are fifth order components at $2\omega_1 - \omega_2 + \omega_2 - \omega_2$ which are the same frequency as the $2\omega_1 - \omega_2$ third order component. The fifth order component increases in proportion to A^5 and the third order component increases in proportion to A^3. For small A, the frequency component will increase as if it were a third order system and for a larger A it will increase as if it were a fifth order system. If plotted on a logarithmic scale, the amplitude of the distortion component as a function of the input amplitude

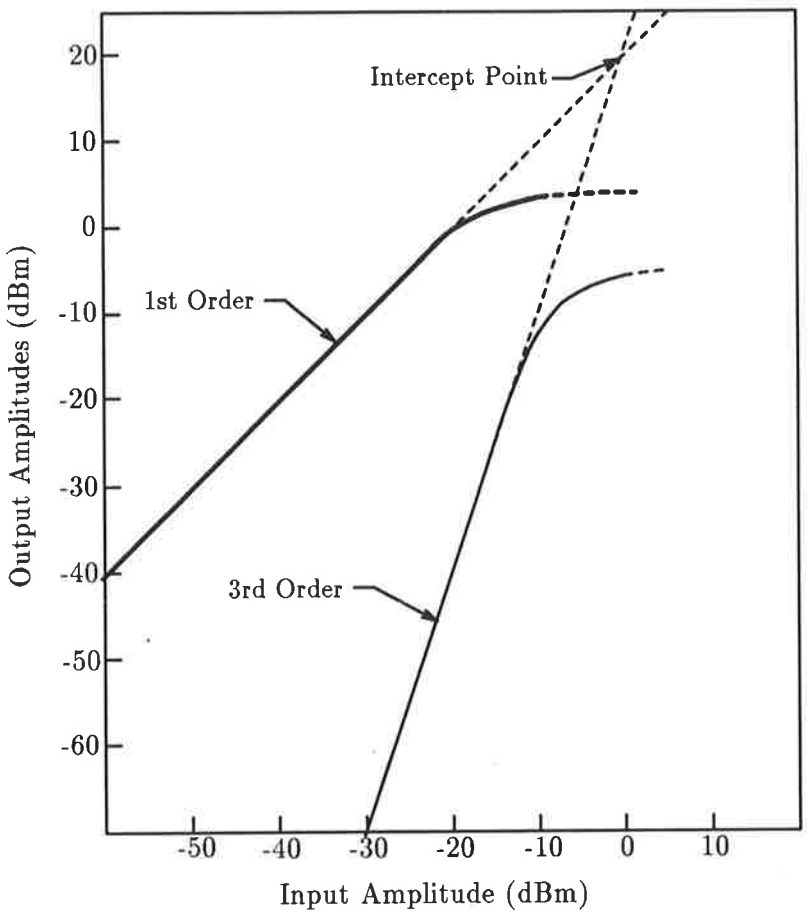


Figure 2.2: The IMD intercept point is the extrapolated point where the amplitude of the distortion equals the amplitude of the fundamental.

will be linear over the range where the third order approximation holds, with third order distortion components increasing 3dB in amplitude for every 1dB increase in input amplitude. This is illustrated in figure 2.2.

It is common in the specification of RF components, for manufacturers to quote a "third order intercept point" [21]. This is the point where the extrapolated distortion amplitude equals the input amplitude. It is important to note that this input amplitude is, for most cases, quite hypothetical and many devices would suffer catastrophic failure if subject to input amplitudes equal

to the IMD intercept value. The advantage of specifying the performance this way is that it is quite easy to work backwards using the rule of 3dB change in distortion level per 1dB change in input level to the operating point actually used.

It is possible to analyze Intermodulation Distortion with distortion models other than the power series [10] [72] [63] [43] [73] [50] [49] [56]. The nonlinearities in microwave communication systems are often described using amplitude and phase describing functions. These systems are often Frequency Division Multiple Access (FDMA) and it is of interest to know how intermodulation distortion between channels interfere with other channels. Analysis of such systems can become quite complex due to the number of channels, although simplifications can be made if it is assumed that the distortion is only of low order (eg third order only). The simplest such analysis considers only the (sinusoidal) carriers of each channel but some analysis which assumes simple modulation has been performed [10] [72] [63] [43] [73] [56]. Analysis of system performance with more accurate representations of the input has been performed numerically [30].

The term Intermodulation, rather implies an effect between two or more separate signals. Of course, any input can be approximated arbitrarily well as a sum of sinusoids (effectively performing a Fourier Transform), and the distortion can, at least in principle, be determined by considering the intermodulation between all the sinusoids. Whilst this is a perfectly valid method of analysis, the author doubts whether "Intermodulation Distortion" is the most appropriate term to apply to such a system. When considering the cumulative effect of all the different order intermodulation products arising from all the input components, the phase of the distortion products must be taken into account. The commonly measured "IMD Intercept Point" contains no information on the phase of the resulting components.

2.4 Cross Modulation Distortion

When a modulated carrier is passed through a nonlinear system with another carrier (modulated or not), the modulation of the first signal is transferred (albeit distorted) to the second carrier. This is termed cross modulation distortion.

It is difficult to trace the first theoretical prediction of the phenomenon of cross modulation distortion. One of the most interesting manifestations of cross-modulation distortion is due to nonlinear ionospheric propagation. This was first observed by Tellegen in 1933 [74] when it was observed that the Luxembourg radio station (252 kHz) could be heard on a radio tuned to a signal from Beromünster (652 kHz). The phenomenon has come to be known as the Luxembourg effect. Cross-modulation in electronic circuits must have already been known since Tellegen rules this out (on the basis of field strengths) as the cause of his observations. For a discussion of the mechanism of distortion in ionospheric propagation see the text by Budden [18].

To see how cross-modulation can happen, consider an input with the frequency domain representation

$$\begin{aligned}\mathbf{X}(\omega) = a_1 &[\delta(\omega-\omega_1)+\delta(\omega+\omega_1)] \\ &+ a_2[\delta(\omega-(\omega_2-\Delta\omega))+\delta(\omega+(\omega_2-\Delta\omega))] \\ &+ a_3[\delta(\omega-\omega_2)+\delta(\omega+\omega_2)] \\ &+ a_4[\delta(\omega-(\omega_2+\Delta\omega))+\delta(\omega+(\omega_2+\Delta\omega))]\,. \qquad (2.15)\end{aligned}$$

This is a representation of an unmodulated signal at a frequency of ω_1 and a carrier at a frequency of ω_2 with two modulation sidebands at frequencies of $\omega_2+\Delta\omega$ and $\omega_2-\Delta\omega$. This could represent AM modulation with a frequency of $\Delta\omega$. By choice of a_2, a_3 and a_4, AM, SSB and DSB could be represented. Other modulation schemes also contribute to cross modulation distortion, but they aren't easily incorporated in this simple analysis. Note that there is no difference mathematically between a number of independent sine waves and

a single modulated carrier. The only difference is that a decoder (or human listener) might interpret information in the transferred modulation. In that sense, cross modulation distortion is in the ear of the beholder!

Now, consider only the the third order term of equation 2.1. The third order components can be obtained by considering that $\mathcal{F}(x(t)^3) = \mathbf{X}(\omega) * \mathbf{X}(\omega) * \mathbf{X}(\omega)$ consists of delta functions whose frequency is the sum of the frequencies of the terms in $\mathbf{X}(\omega)$ taken three at a time. It becomes a little cumbersome to write out the full expansion in the form of an equation, so we will tabulate the terms Referring to table 2.1, terms 1–4 are all of the carrier frequency ω_1. Terms 5–8

	Frequency			Coeff.
1	$\omega_1 + (-\omega_1) + \omega_1$	=	ω_1	a_1^3
2	$(\omega_2 + \Delta\omega) - (\omega_2 + \Delta\omega) + \omega_1$	=	ω_1	$a_1 a_2 a_4$
3	$(\omega_2 - \Delta\omega) - (\omega_2 - \Delta\omega) + \omega_1$	=	ω_1	$a_1 a_2 a_4$
4	$\omega_2 - \omega_2 + \omega_1$	=	ω_1	$a_1 a_3^2$
5	$(\omega_2 + \Delta\omega) - \omega_2 + \omega_1$	=	$\omega_1 + \Delta\omega$	$a_1 a_3 a_4$
6	$(\omega_2 - \Delta\omega) - \omega_2 + \omega_1$	=	$\omega_1 - \Delta\omega$	$a_1 a_3 a_2$
7	$\omega_2 - (\omega_2 + \Delta\omega) + \omega_1$	=	$\omega_1 - \Delta\omega$	$a_1 a_3 a_4$
8	$\omega_2 - (\omega_2 - \Delta\omega) + \omega_1$	=	$\omega_1 + \Delta\omega$	$a_1 a_3 a_2$
9	$(\omega_2 + \Delta\omega) - (\omega_2 - \Delta\omega) + \omega_1$	=	$\omega_1 + 2\Delta\omega$	$a_1 a_3 a_4$
10	$(\omega_2 - \Delta\omega) - (\omega_2 + \Delta\omega) + \omega_1$	=	$\omega_1 - 2\Delta\omega$	$a_1 a_3 a_4$

Table 2.1: Third order cross modulation terms. Only terms with a frequency near ω_1 are tabulated.

represent the modulation transferred to the carrier at frequency ω_1 and terms 9 and 10 represent harmonic distortion of the transferred modulation.

2.5 Polyspectra

A complete discussion of polyspectra (also known as "higher order spectra") is outside the scope of this section. However, the specific application of polyspec-

tra to the observation of nonlinear phenomena will be briefly discussed.

In the previous sections of this chapter, we have seen that nonlinear distortion results in components in the output at frequencies which may not be present in the input. In the case of a noisy output, there may be a spontaneously generated component at the same frequency as the distortion product. The component due to nonlinear distortion bears a definite phase relationship to the input of the nonlinearity whereas the spontaneously generated component will be uncorrelated to the input of the nonlinearity. Polyspectra provide a way of differentiating between these components. A most convenient way of observing the nonlinear components is to select the input in such a way that the polyspectra of order greater than 2 are zero if the system is linear. Then any non-zero polyspectra of order greater than 2 must be due to nonlinearities. It is sufficient that the input be a Gaussian random process.

Polyspectra are defined in the literature [45] [54], as follows. First, define the joint moment generating function of the set of n real random variables, $\{x_1, x_2, \ldots, x_n\}$, by

$$\Phi(\omega_1, \omega_2, \ldots, \omega_n) = \mathcal{E}(e^{j(\omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n)}). \tag{2.16}$$

Equation 2.16 is also known as the joint characteristic function. The term "moment generating function" arises since the moments are given by

$$m_{k_1,\ldots,k_n} \equiv \mathcal{E}(x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}) \tag{2.17}$$

$$= (-j)^r \left. \frac{\partial^r \Phi(\omega_1, \omega_2, \ldots, \omega_n)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \cdots \partial \omega_n^{k_n}} \right|_{\omega_1 = \omega_2 = \cdots = \omega_n = 0} \tag{2.18}$$

where

$$r = k_1 + k_2 + \cdots + k_n.$$

Equation 2.18 can be readily verified by substituting equation 2.16 into equation 2.18 and exchanging the order of the expectation and differentiation operations. The joint cumulants are then defined

$$c_{k_1 \cdots k_n} \equiv (-j)^r \left. \frac{\partial^r \log \Phi(\omega_1, \omega_2, \ldots, \omega_n)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \cdots \partial \omega_n^{k_n}} \right|_{\omega_1 = \omega_2 = \cdots = \omega_n = 0}. \tag{2.19}$$

The definitions 2.16 to 2.19 are straight forward extensions of equivalent functions of a single random variable [76] [57].

Now consider a real stationary random process

$$\{X(k)\}, \quad k = \ldots, -2, -1, 0, 1, 2, \ldots$$

and define

$$c_N(\tau_1, \tau_2, \ldots, \tau_{N-1}) \equiv c_{X(k),X(k+\tau_1),X(k+\tau_2),\ldots,X(k+\tau_{N-1})}. \tag{2.20}$$

Equation 2.20 is for the discrete case with $\tau_i = \ldots, -2, -1, 0, 1, 2, \ldots$. This definition follows the notation of Nikias and Raghuveer [54] although they appear to have omitted the actual definition from their paper. Finally the Nth order spectrum is defined as the $N-1$ dimensional Fourier Transform of equation 2.20. Bendat [6], defines higher order spectra as the Fourier transform of higher order auto-correlation functions which is the same as the moment of the random variables $X(k), X(k+\tau_1), X(k+\tau_2), \ldots, X(k+\tau_{N-1})$. In short, Bendat [6], defines higher order spectra as the Fourier transform of the moment, rather than the Fourier transform of the cumulant. It happens that these definitions are the same for $N < 3$ but differ for $N \geq 3$ [54].

Some properties of polyspectra are:

1. The second order spectrum is the familiar power spectrum [54].
2. For a Gaussian process, the nth order spectrum is zero for $n \geq 3$ [54] [45].
3. The bispectrum $B(\omega_1, \omega_2)$ represents the correlation of the component at a frequency $\omega_1 + \omega_2$ with the product of the components at frequencies of ω_1 and ω_2 [54].

As a consequence of the above properties, polyspectra provide no extra information if a Gaussian input is applied to a linear system. However, with a

non-Gaussian random input, it is possible to identify a linear system [54] [34] [75] [35]. With only the power spectrum it is possible to identify the magnitude of a systems transfer function, but not it's phase. Further, if the input *is* Gaussian and the higher order spectra for $n > 2$ are *not* zero, then the system must be nonlinear and property 3 (above) gives a reasonably intuitive interpretation.

Linear system identification using polyspectra has the advantage that the actual input does not need to be known, however, where the input *is* known (or measurable), then methods using polyspectra are unlikely to ever compete with more traditional methods. Without attempting a rigorous analysis (which could be made obsolete by more efficient methods) we simply observe that discarding the input information is unlikely to make the identification task easier. At the time of writing, it does not appear possible to identify a *general* nonlinearity using polyspectra[2].

2.6 Summary

In this chapter we have examined three traditional measures of distortion, "Harmonic Distortion", "Intermodulation Distortion" and "Cross Modulation Distortion". These names suggest that the distortions are somehow different, but these effects can occur from the same nonlinearity, the only thing to change being the complexity of the input signal. If the input consists of only a single sinusoid, then we only see harmonic distortion. If the input has two sinusoids then we see intermodulation distortion as well as harmonic distortion. If the system consists of more than two sinusoids and we consider one of them to be a modulation sideband then we can observe cross modulation distortion in addi-

[2]There does not appear to be any cases in the literature of nonlinear identification using solely (auto)-polyspectra. However, cross-polyspectra (of input and output) have been used for system identification [44] [7]. The method can be applied to systems distributed in space as well as time [59].

tion to harmonic distortion and intermodulation distortion. Note that when the distortion products from different input components have the same frequency, then the phase of the products must be taken into account when working out the cumulative effect.

Polyspectra (or higher order spectra) provide a way of detecting nonlinear distortion when the input is an unknown Gaussian process.

These terms "Harmonic Distortion", "Intermodulation Distortion", "Cross Modulation Distortion" and "Polyspectra" (when the input is Gaussian), are similar to the extent that they are all *measures* of distortion, but they are not *models* of distortion. None of these methods are suitable for predicting the output of a system with a different input (which is the prime purpose of a model).

As has been mentioned, it is possible to use these measures of distortion to evaluate the parameters of a model although, as we will see in chapter 3, a general model can not be evaluated with simple inputs consisting of only a few sinusoids.

Chapter 3

Parametric Characterization

3.1 Introduction

Most of the work in this chapter was covered in a paper by myself and Bogner [24]. Signal degradation may be due to noise or to distortion. Linear distortion may be caused by filters, dispersion or multipath effects. These effects are well understood and can often be minimized by linear equalizers. To make further improvements it is necessary to consider the correction of nonlinear distortion. Distortion correction can be performed by using pre or post distortion to compensate ([5] , [8] [28], [63], [64] and [68]). It is desirable to know whether the structure chosen for a distortion compensator is capable of correcting a particular class of distortion. To this end it is necessary to have a valid model of the distortion to be corrected. The model must be able to simulate the output quite accurately. This application is more demanding than some where a reasonable estimate of the magnitude of the distortion may be all that is required. For example, harmonic distortion measurements and intermodulation distortion measurements discussed in chapter 2 give an estimate of how severe the nonlinearity is (and even that may be inaccurate) but are quite unsuitable for predicting the form of the distortion.

Nonlinear distortion can be classified as being with or without memory. A memoryless system is one whose output $y(t)$ at any time t is a function of its input $x(t)$ at that same instant. Generally a pure delay can be ignored so that a system $y(t) = f(x(t-T))$ can usually be treated as though it were memoryless. A system with memory is one whose output at any instant is dependent on the input at some instants other than the present. Some writers refer to systems whose memory is short compared to the period of the modulation as being memoryless, which can create confusion. In this chapter we will refer to systems with short memory as such, reserving the terms "memoryless" and "zero-memory" to describe systems whose output can be validly considered to be a function only of the input at that instant.

Many of the physical phenomena causing nonlinearities are essentially memoryless but they are often embedded in a network of linear components which cause the component as a whole to have memory. Further, virtually all communication systems have filters intentionally included to reduce susceptibility to thermal noise in receivers and to reduce out of band spectral emissions. Detailed discussion of the design of these components is not appropriate in this chapter, but in general terms the band width of the filters will be a compromise between reducing thermal noise and reducing intersymbol interference. Consequently it is unlikely that practical communication system nonlinearities can be adequately modelled as memoryless.

In this chapter we will first review some established methods of modelling distortion. These are, in order of increasing generality, infinite series of functions (such as power series), amplitude and phase describing functions, the Volterra series and the Wiener model. These have all been used extensively in the literature but the question of under what circumstances the less general models may be adequate has been largely ignored. We will attempt to redress this balance by comparing the less general models with the Volterra series model of the same system.

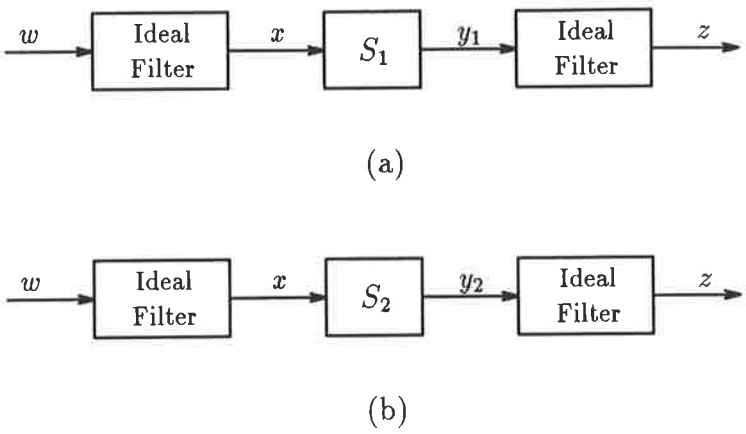


Figure 3.1: S_2 is equivalent, in a bandpass context, to S_1 if the outputs, z, are identical for identical inputs w.

We will be concerned chiefly with distortion in a bandpass context where the input is bandpass and the output is bandpass, or at least, distortion products far from the band of interest can be ignored. A nonlinear system in a bandpass context can be considered to be preceded and followed by ideal bandpass filters (figure 3.1a). It will be shown that S_1 can be replaced with S_2 (figure 3.1b) without affecting the output $z(t)$ due to the input $w(t)$.

3.2 Instantaneous Distortion

Memoryless distortion is conveniently represented by a (possibly infinite) series of functions. The output $y(t)$ is related to the input $x(t)$ by

$$y(t) = \sum_{n=1}^{\infty} G_n(x(t)). \tag{3.1}$$

Typically $G_n(x) = a_n x^n$ implying a power series. By appropriate choice of $G_n(x)$, Fourier series or series of Bessel functions are possible, and may have more desirable convergence and numerical conditioning characteristics.

It is appropriate to note that a power series can be made to approximate any function, and hence any memoryless distortion, arbitrarily well over a given range (the Weierstrass theorem). Few communication systems are memoryless or can be accurately modelled by a power series, yet the power series model has been extensively used, and continues to be used because applying more accurate models can present intractable problems. Sometimes it is possible to use the zero memory model to work out worst case results. Because it is still used, and also because it is the basis for later results we will now consider in some detail the behaviour of a system modelled by a power series in a band limited context.

Assume the system is described by 3.1 with

$$G_n(x(t)) = a_n x^n(t) \tag{3.2}$$

and the input is

$$\begin{aligned} x(t) &= \Re\left[\tilde{x}(t)e^{j\omega_0 t}\right] \\ &= \frac{1}{2}\left(\tilde{x}(t)e^{j\omega_0 t} + \tilde{x}^*(t)e^{-j\omega_0 t}\right). \end{aligned} \tag{3.3}$$

The variable $\tilde{x}$ is the complex envelope of x defined by $\tilde{x} = (x + j\hat{x})e^{j\omega_0 t}$, where $\hat{x}$ is the Hilbert transform of x. The asterisk superscript is used to indicate the complex conjugate. Substituting 3.3 into equation 3.2 gives for one term of the power series

$$G_n(x(t)) = \frac{a_n}{2^n}\sum_{i=0}^{n}\binom{n}{i}\tilde{x}^{n-i}(t)\tilde{x}^{*i}(t)e^{(n-2i)jw_0 t}. \tag{3.4}$$

If we now note that the ith term has the same magnitude centre frequency as the lth term if $n - 2i = -(n - 2l)$ i.e. $l = n - i$. For n odd,

$$G_n(x) = \frac{a_n}{2^n}\sum_{i=0}^{(n-1)/2}\left[\binom{n}{i}\tilde{x}^{n-i}\tilde{x}^{*i}e^{(n-2i)jw_0 t} + \binom{n}{n-i}\tilde{x}^{i}\tilde{x}^{*n-i}e^{-(n-2i)jw_0 t}\right] \tag{3.5}$$

and for n even

$$G_n(x) = \frac{a_n}{2^n}\binom{n}{n/2}\tilde{x}^{n/2}\tilde{x}^{*n/2} + \sum_{i=0}^{(n/2)-1}\left[\binom{n}{i}\tilde{x}^{n-i}\tilde{x}^{*i}e^{(n-2i)jw_0t} + \binom{n}{n-i}\tilde{x}^{i}\tilde{x}^{*n-i}e^{-(n-2i)jw_0t}\right]. \tag{3.6}$$

If we now make use of the narrowband constraint and discard all terms involving $e^{\pm njw_0t}$ where $n \neq 1$ it can be seen by inspection of 3.5 and 3.6 that the only term remaining has n odd and $i = (n-1)/2$. Also making use of $\binom{n}{i} = \binom{n}{n-i}$ and $\tilde{x}\tilde{x}^* = |\tilde{x}|^2$ results in

$$\begin{aligned} G_n(x) &= \frac{a_n}{2^n}\binom{n}{(n-1)/2}|\tilde{x}|^{n-1}\left(\tilde{x}e^{j\omega_0t} + \tilde{x}^*e^{-j\omega_0t}\right) \\ &= \frac{a_n}{2^{(n-1)}}\binom{n}{(n-1)/2}|\tilde{x}|^{n-1}x \\ &\text{For n=1,3,5....} \end{aligned} \tag{3.7}$$

Equation 3.7 shows that a zero-memory distortion can, when the inputs and outputs are narrowband, be considered as a distortion of the complex envelope. The output complex envelope $\tilde{y}$ is given by

$$\tilde{y} = \left[\sum_{k=0}^{\infty}\frac{a_{2k+1}}{2^{2k}}\binom{2k+1}{k}|\tilde{x}|^{2k}\right]\tilde{x}. \tag{3.8}$$

Since the multiplicative factor in 3.8 is a real function, then it has been shown that there can be no phase shift in a system with zero memory distortion.

3.3 Amplitude and Phase Describing Functions

The memoryless distortion represented by 3.8 can be generalized to include a limited class of distortion with memory by allowing phase shift to be repre-

sented. Thus

$$\tilde{y} = A(|\tilde{x}|)\, e^{j\Phi(|\tilde{x}|)}\tilde{x}, \tag{3.9}$$

or equivalently,

$$\tilde{y} = [I(|\tilde{x}|) + jQ(|\tilde{x}|)]\,\tilde{x}. \tag{3.10}$$

$A(|\tilde{x}|)$ is a real function known as the amplitude describing function and is said to give rise to AM/AM distortion. $\Phi(|\tilde{x}|)$ is a real function said to give rise to AM/PM distortion. This representation (equation 3.9) is very convenient since no further effort is needed to deal with the case of bandpass signals. See figure 3.2.

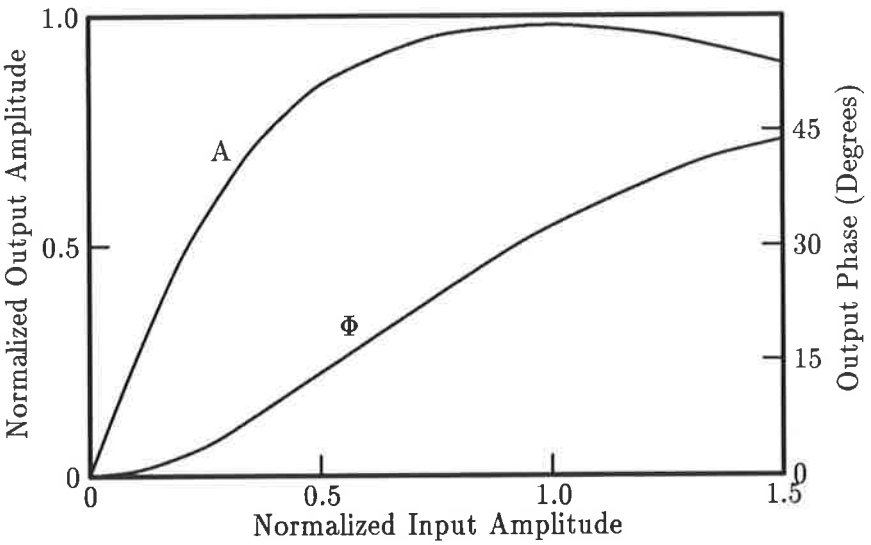


Figure 3.2: Amplitude and phase describing functions typical of a TWT.

The alternate representation 3.10 is useful from the implementation point of view (figure 3.3). The functions $I(|\tilde{x}|)$ and $Q(|\tilde{x}|)$ are necessarily even functions because they are functions of $|\tilde{x}|$. They can therefore, be represented by a series of even powers of $|\tilde{x}|$ so that $I(|\tilde{x}|)\,\tilde{x}$ and $Q(|\tilde{x}|)\,\tilde{x}$ can be represented in the form of equation 3.8, and applying the result of section 3.2 there must exist two corresponding functions of x, $\bar{I}(x)$ and $\bar{Q}(x)$, which, in the bandpass context, result in the same distortion. $\bar{I}(x)$ and $\bar{Q}(x)$ are more amenable to a

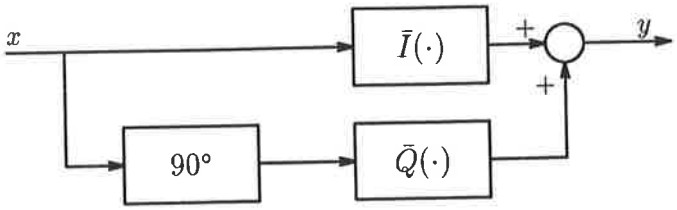


Figure 3.3: Illustration of the in-phase and quadrature model (equation 3.10).

direct implementation since there is no need to attempt to generate a signal $|\tilde{x}|$. Amplitude and phase describing functions are very widely used to model communication systems ([10], [62], [63], [64], [72]) and it is desirable to understand the range of validity of the model (section 3.6).

3.4 Volterra Series

Amplitude and phase describing functions are not sufficient to model general nonlinearities with memory. A more general representation is the Volterra series. Schetzen [68][67] credits Volterra with generalizing the Taylor series in the 1880s. The derivation which follows is essentially that given by Schetzen [67]. A causal nonlinear system with memory can be approximated by multi-variable polynomial of samples of the past input, $x(t - iT)T$ where T is the sampling interval. Non-causal systems can be approximated by a multi-variable polynomial of samples of the past and future inputs. Figure 3.4 illustrates $x(t - iT)T$ as the area in a rectangle approximating the area under the $x(t)$ curve.

Consider for the moment, just the nth order terms of such a polynomial

$$H_n(x(t)) \approx \sum_{i_1=-N}^{N} \cdots \sum_{i_n=-N}^{N} h(i_1T \ldots i_nT) \prod_{p=1}^{n} x(t - i_pT)T \qquad (3.11)$$
$$\approx \sum_{i_1T=-NT}^{NT} \cdots \sum_{i_nT=-NT}^{NT} h(i_1T \ldots i_nT) \prod_{p=1}^{n} x(t - i_pT)T$$

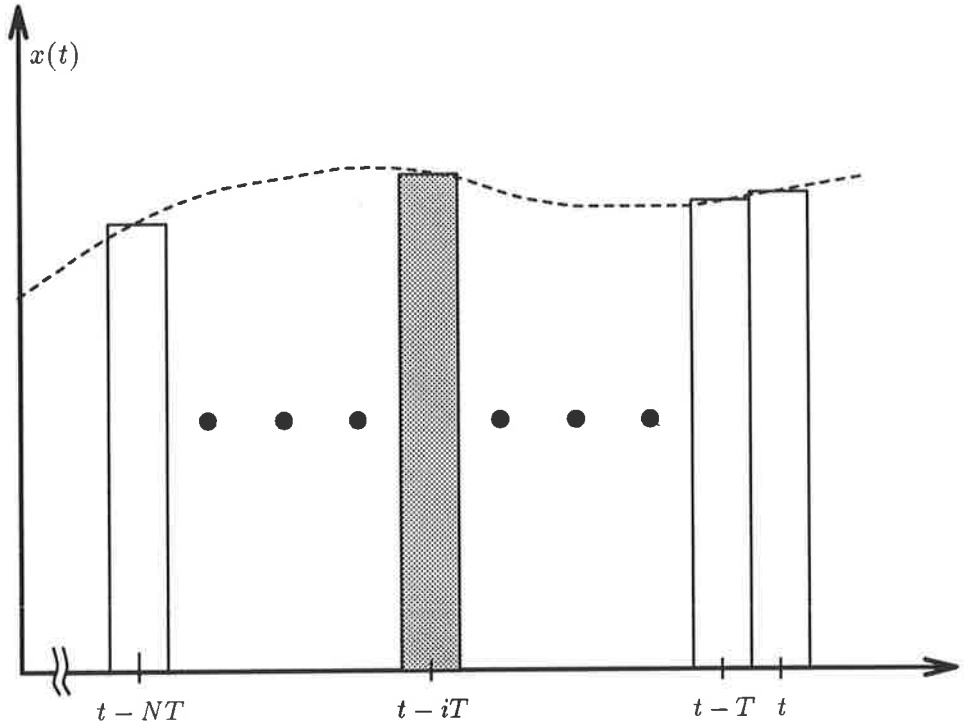


Figure 3.4: Samples of $x(t)$. The shaded area is $x(t - iT)T$.

and put $i_p T = \tau_{i_p}$ and $NT = t_1$

$$H_n(x(t)) \approx \sum_{\tau_1=-t_1}^{t_1} \cdots \sum_{\tau_n=-t_1}^{t_1} h(\tau_1 \ldots \tau_n) \prod_{p=1}^{n} x(t - \tau_p)T$$

and in the limit as $T \to 0$

$$H_n(x(t)) = \int\int \ldots \int_{-t_1}^{t_1} h_n(\tau_1 \ldots \tau_n) \prod_{p=1}^{n} x(t - \tau_p)\, d\tau_1\, d\tau_2 \cdots d\tau_n. \qquad (3.12)$$

The Volterra series is the limit as $t_1 \to \infty$ of the sum of $H_n(x(t))$,

$$y(t) = \sum_{n=0}^{\infty} H_n(x(t)), \qquad (3.13)$$

where

$$H_n(x(t)) = \int\int \ldots \int_{-\infty}^{\infty} h_n(\tau_1 \ldots \tau_n) \prod_{p=1}^{n} x(t - \tau_p)\, d\tau_1\, d\tau_2 \cdots d\tau_n. \qquad (3.14)$$

The $h_n(\tau_1 \ldots \tau_n)$ are known as the Volterra kernels and may be thought of as a generalization of the impulse response of a linear system. As an aid to understanding the Volterra series consider two special cases.

1. If $h_n(\tau_1, \tau_2 \ldots \tau_n) = a_n \prod_{p=1}^{n} \delta(\tau_p)$, then $y(t) = \sum_{n=0}^{\infty} a_n x^n(t)$, which is a power series (memoryless distortion).

2. If $h_n(\tau_1, \tau_2 \ldots \tau_n) = 0$ for all $n > 1$, then the system is a linear one with an impulse response of $h_1(\tau_1)$.

There is a theorem due to Fréchet which states that any functional can be arbitrarily well approximated over a given range by a Volterra series. This makes the Volterra series very general although systems with more than one stable state (such as flip-flops) cannot be characterized by a Volterra series. Determining the Volterra kernels $h_n(\tau_1, \tau_2 \ldots \tau_n)$ of a system experimentally is cumbersome and evaluating 3.13 in a simulation is expensive in computing resources. The Volterra series does, however, provide a useful theoretical tool and extensive use will be made of it in the rest of this chapter.

Note that the nth order term scales in proportion to A^n where A is the amplitude of the input signal. This can be verified by substituting $Ax(t - \tau_p)$ for $x(t - \tau_p)$ in equation 3.14. This property was previously (section 2.2) shown for the case of zero memory distortion.

It can be shown ([8], Appendix A) that in the bandpass context, distortion represented by 3.13 may equivalently be represented by

$$\tilde{y}(t) = \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \binom{2k+1}{k} \int\int \cdots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) \prod_{i=1}^{k} \tilde{x}^*(t - \tau_i) e^{-j\omega_0 \tau_i} \times \prod_{i=k+1}^{2k+1} \tilde{x}(t - \tau_i) e^{j\omega_0 \tau_i} \, d\tau_1 \, d\tau_2 \cdots d\tau_n. \tag{3.15}$$

This is significant because it shows that, in a bandpass context, quite general distortion of a signal can be modelled by distortion of the signal's complex envelope. It is possible to go further and define a low pass equivalent of the Volterra kernel [8] but this is not required for the present purpose.

3.5 The Wiener Model

Wiener developed a model which is capable of representing any single input single output distortion with finite memory [78]. Schetzen presents a detailed derivation of the Wiener model by representing the Volterra series as a series of Wiener's G functionals [67][68]. Here a more direct approach will be taken since none of the intermediate results involving G functionals are required, and that approach would have to be reworked to extend the model to multiple inputs (section 4.3).

Consider a system with input $x(t)$ and an output $y(t)$. We can represent a function such as $x(t)$ as a sum of orthogonal functions $x(t) = \sum_{n=1}^{\infty} a_n \alpha_n(t)$, where $\alpha_n(t)$ are orthogonal functions in t. If $\int_{-\infty}^{\infty} x(t)^2\, dt < \infty$, then this is always possible. The function $x(t)$ is then described entirely by the a_n.

If only causal systems with finite memory are considered then the output $y(t)$, is only dependent on the input $x(t)$ over the interval $[t-T, t]$ where T is regarded as the maximum "memory" of the system. Most real systems will not have a well defined time T, but never the less it will be possible define some T large enough to ensure that the error in modelling $y(t)$ is arbitrarily small.

The signal $x(t)$ for t in the interval $[t-T, t]$ can be approximated by a sum of orthonormal functions. Equivalently, $x(t-\tau)$ for τ in the interval $[0, T]$ can be approximated by a sum of orthonormal functions. The latter form will be used as it leads to a model with a reasonable physical interpretation:

$$x(t-\tau) = \lim_{M \to \infty} \sum_{m=0}^{M} a_m(t) \alpha_m(\tau), \tag{3.16}$$

where

$$a_m(t) = \int_0^T \alpha_m(\tau) x(t-\tau)\, d\tau. \tag{3.17}$$

All the information about the history of the signal over the period $[t-T, t]$, is contained in the current values of $a_m(t)$. Consequently the output can be

expressed

$$y(t) = \lim_{M \to \infty} f\left(a_1(t), a_2(t), \ldots, a_M(t)\right). \tag{3.18}$$

Where $f(a_1(t), a_2(t), \ldots, a_M(t))$ is a function of M variables (and consequently has no memory of its inputs). Note that 3.17 is in the form of a convolution integral, so that the calculation represented by 3.17 can be performed by a linear system whose impulse response is $\alpha_m(t)$ (figure 3.5).

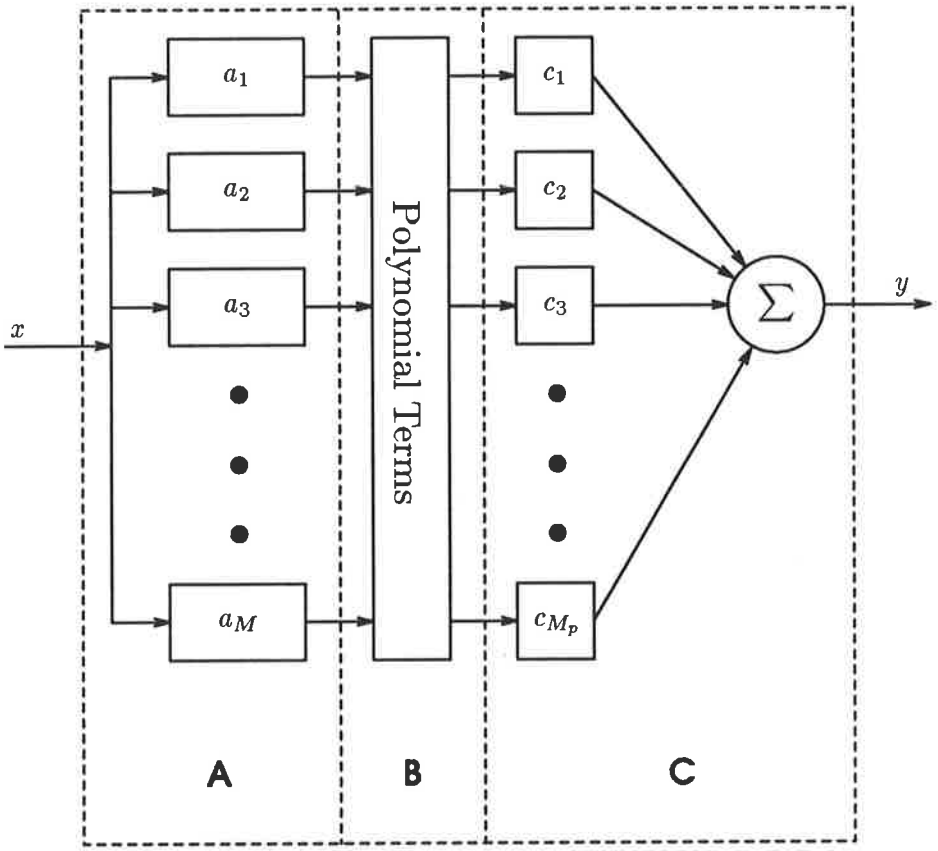


Figure 3.5: The single input Wiener model.

The function $f(a_1(t), a_2(t), \ldots, a_M(t))$ can be represented by a power series in M variables. The weights c in figure 3.5 represent the coefficients of the terms in such a series. In the Wiener model, section C of figure 3.5 is implemented as a series of orthogonal polynomials rather than as a power series (a series of non-orthogonal polynomials) which has certain advantages when it comes to

devising a rapidly convergent strategy for adjusting the weights c.

It can be shown (Appendix B equation B.4) that a function of several variables may be represented as

$$f(a_1(t), a_2(t), \ldots, a_M(t)) = \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \cdots \sum_{m_M=0}^{\infty} b_{m_1,m_2,\ldots,m_M} \prod_{q=1}^{M} \beta_{m_q}(a_q(t)). \tag{3.19}$$

The $\beta_{m_q}(a_q(t))$ in equation 3.19 are single dimensional orthonormal functions such that

$$\langle \beta_i(a_m(t))\beta_j(a_m(t)) \rangle = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j. \end{cases} \tag{3.20}$$

Since the $a_m(t)$ are functionals of $x(t)$ defined by equation 3.17 we can express equation 3.19 as

$$f(a_1(t), a_2(t), \ldots, a_M(t)) = \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \cdots \sum_{m_M=0}^{\infty} b_{m_1,m_2,\ldots,m_M} \phi_{m_1,m_2,\ldots,m_M}(x(t)). \tag{3.21}$$

From equation 3.19 and equation 3.21

$$\phi_{m_1,m_2,\ldots,m_M}(x(t)) = \beta_{m_1}(a_1(t))\beta_{m_2}(a_2(t)) \cdots \beta_{m_M}(a_M(t)),$$

and

$$\phi_{k_1,k_2,\ldots,k_M}(x(t)) = \beta_{k_1}(a_1(t))\beta_{k_2}(a_2(t)) \cdots \beta_{k_M}(a_M(t)).$$

Consequently the $\phi_{m_1,m_2,\ldots,m_M}(x(t))$ can be considered orthonormal functionals with the property that

$$\langle \phi_{m_1,m_2,\ldots,m_M}(x(t))\phi_{k_1,k_2,\ldots,k_M}(x(t)) \rangle = \begin{cases} 1, & \text{if } m_i = k_i \text{ for } i = 1, 2, \ldots, M; \\ 0, & \text{otherwise.} \end{cases} \tag{3.22}$$

The notation in equation 3.22 can become a little cumbersome, so we define the Kronecker delta function

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j. \end{cases}$$

We can then rewrite equation 3.22 as

$$\langle \phi_{m_1,m_2,\ldots,m_M}(x(t))\phi_{k_1,k_2,\ldots,k_M}(x(t))\rangle = \prod_{q=1}^{M} \delta_{m_q k_q}. \tag{3.23}$$

Multiplying both sides of equation 3.21 by $\phi_{k_1,k_2,\ldots,k_M}(x(t))$, taking the average, exchanging the order of the summation and the averaging, and eliminating terms which are zero by property equation 3.23, we obtain

$$b_{k_1,k_2,\ldots,k_M} = \langle f(a_1(t), a_2(t), \ldots, a_M(t))\phi_{k_1,k_2,\ldots,k_M}(x(t)),\rangle$$

and by simply changing the variables,

$$b_{m_1,m_2,\ldots,m_M} = \langle f(a_1(t), a_2(t), \ldots, a_M(t))\phi_{m_1,m_2,\ldots,m_M}(x(t))\rangle \tag{3.24}$$

Equation 3.24 implies that the $b_{m_1,m_2,\ldots,m_M}$ can be determined by a simple correlation.

Up to this point the development has been independent of the choice of the sets of orthogonal functions. The choice of the $\beta_{m_q}(a_q(t))$ is complicated by the fact that $a_q(t)$ is typically a random variable whose distribution must be taken into account. The $\beta_{m_q}(a_q(t))$ must satisfy equation 3.20 which can be rewritten as

$$\int_{-\infty}^{\infty} \beta_i(a)\beta_j(a)\rho(a)\,da = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j; \end{cases} \tag{3.25}$$

where $\rho(a)$ is the probability density function of a and can be thought of as the fraction of the time which a spends at a particular amplitude. More precisely, $\rho(a_0)da$ is the fraction of the time for which $a_0 < a < a_0 + da$.

The choice of the $\beta_{m_q}(a_q(t))$ is governed by the probability density function of its inputs. Wiener and others have assumed that the a_q are Gaussian distributed with a probability density function of $\rho(a) = e^{-a^2/2\sigma^2}/(\sigma\sqrt{2\pi})$. If the $\beta_m(a)$ are considered to be polynomials of the form $\beta_m(a) = \sum_{k=0}^{m} c_k a^m$ then the coefficients c_k can be determined uniquely by applying equation 3.25. The mth degree polynomial must be orthogonal to all lower order polynomials. If

it is normalized (also implied by 3.25), there are $m+1$ equations in $m+1$ variables $c_k, k = 0, 1, 2, \ldots, m$. In the case of a_q being Gaussian distributed, the polynomials are the Hermite polynomials.

The $\alpha_m(t)$ in section A of figure 3.5 represent equation 3.17. That is to say that the blocks in this part of the figure have impulse responses $\alpha_m(t)$ which form a set of orthogonal functions. Wiener used Laguerre functions for the $\alpha_m(t)$ [78][67][68], because they are readily realizable by a cascade of single pole all-pass networks. Any set of orthogonal functions can be used, however, and in chapter 4 sinusoids will be used which makes it rather easy to eliminate components of the input signal outside a certain passband. The treatment of bandpass signals in the Wiener model is covered in chapter 4 and so will not be covered further here.

3.6 Comparison of Models

For comparison, we would like to express the power series model and the amplitude and phase describing function model as special cases of the Volterra series model.

Expressing a power series as a special case of the Volterra series is trivial. It can be seen by inspection that a Volterra kernel

$$h_n(\tau_1, \tau_2 \ldots \tau_n) = a_n \prod_{p=1}^{n} \delta(\tau_p) \tag{3.26}$$

substituted in 3.14 results in the power series form for y:

$$y(t) = \sum_{n=0}^{\infty} a_n x^n(t)$$

and 3.26 substituted in 3.15 yields 3.8.

The case of the amplitude and phase describing function model is a little

more involved. Consider

$$h_n(\tau_1, \tau_2 \dots \tau_n) = h'_n(\tau_1, \tau_2 \dots \tau_n) + h''_n(\tau_1, \tau_2 \dots \tau_n), \tag{3.27}$$

where

$$h'_n(\tau_1, \tau_2 \dots \tau_n) = a_n \prod_{p=1}^{n} \delta(\tau_p)$$

and

$$h''_n(\tau_1, \tau_2 \dots \tau_n) = b_n \prod_{p=1}^{n} \frac{1}{\pi \tau_p}.$$

Since $\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{1}{\tau} x(t-\tau)\, d\tau = \hat{x}$, the Hilbert transform of x, substitution of 3.27 in 3.14 results in

$$H_n(x(t)) = a_n x^n(t) + b_n \hat{x}^n(t)$$

and

$$y(t) = \sum_{n=0}^{\infty} a_n x^n(t) + \sum_{n=0}^{\infty} b_n \hat{x}^n(t). \tag{3.28}$$

Using the results from section 3.2 we put both summations of 3.28 in the form of 3.8

$$\tilde{y} = \left[\sum_{k=0}^{\infty} \frac{a_{2k+1}}{2^{2k}} \binom{2k+1}{k} |\tilde{x}|^{2k} \right] \tilde{x} + \left[\sum_{k=0}^{\infty} \frac{b_{2k+1}}{2^{2k}} \binom{2k+1}{k} |\tilde{\hat{x}}|^{2k} \right] \tilde{\hat{x}}. \tag{3.29}$$

Now using $\tilde{x} = (x + j\hat{x}) e^{j\omega_0 t}$ it is easy to show that $\tilde{\hat{x}} = -j\tilde{x}$ and therefore

$$\tilde{y} = \left[\sum_{k=0}^{\infty} \frac{a_{2k+1}}{2^{2k}} \binom{2k+1}{k} |\tilde{x}|^{2k} \right] \tilde{x} + j \left[\sum_{k=0}^{\infty} \frac{-b_{2k+1}}{2^{2k}} \binom{2k+1}{k} |\tilde{x}|^{2k} \right] \tilde{x}, \tag{3.30}$$

which is of the same form as equation 3.10 as required.

To summarize, if $h_n(\tau_1, \tau_2 \dots \tau_n) = a_n \prod_{p=1}^{n} \delta(\tau_p)$, then the distortion is zero memory, if $h_n(\tau_1, \tau_2 \dots \tau_n) = a_n \prod_{p=1}^{n} \delta(\tau_p) + b_n \prod_{p=1}^{n} \frac{1}{\pi \tau_p}$, then gain and phase describing functions are applicable and otherwise the Volterra series must be used.

3.7 Frequency Domain Analysis

If the input to a linear system is always limited to a certain band then the real system and the model of the system need only approximate each other within

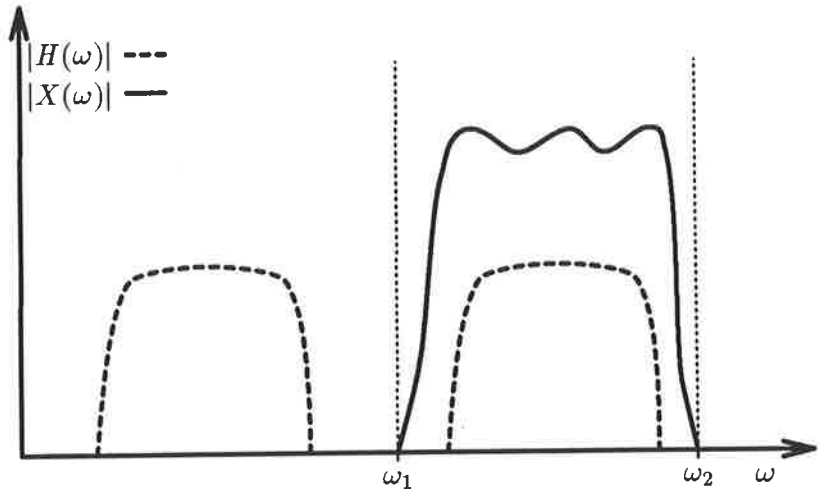


Figure 3.6: Frequency range of interest for a linear system. If the input, $|\mathbf{X}(\omega)|$, is band limited to ω_1–ω_2, then the system response outside that range need not be considered.

that band (see figure 3.6). It would be useful to be able to relax in a similar way the approximation of a real nonlinear system by a model. Just as for the linear case, it is most convenient to handle this problem in the frequency domain.

Define

$$y_n(t) = H_n(x(t)),$$

also define

$$\breve{y}_n(t_1, t_2, \ldots, t_n) = \int\int \ldots \int_{-\infty}^{\infty} d\tau_1\, d\tau_2 \cdots d\tau_n\, h_n(\tau_1 \ldots \tau_n) \prod_{p=1}^{n} x(t_p - \tau_p). \tag{3.31}$$

By reference to 3.14 it is apparent that

$$y_n(t) = \breve{y}_n(t, t, \ldots, t). \tag{3.32}$$

Taking the n-dimensional Fourier transform of 3.32 results in

$$\breve{\mathbf{Y}}_n(\omega_1, \omega_2, \ldots \omega_n) = \mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n) \prod_{p=1}^{n} \mathbf{X}(\omega_p). \tag{3.33}$$

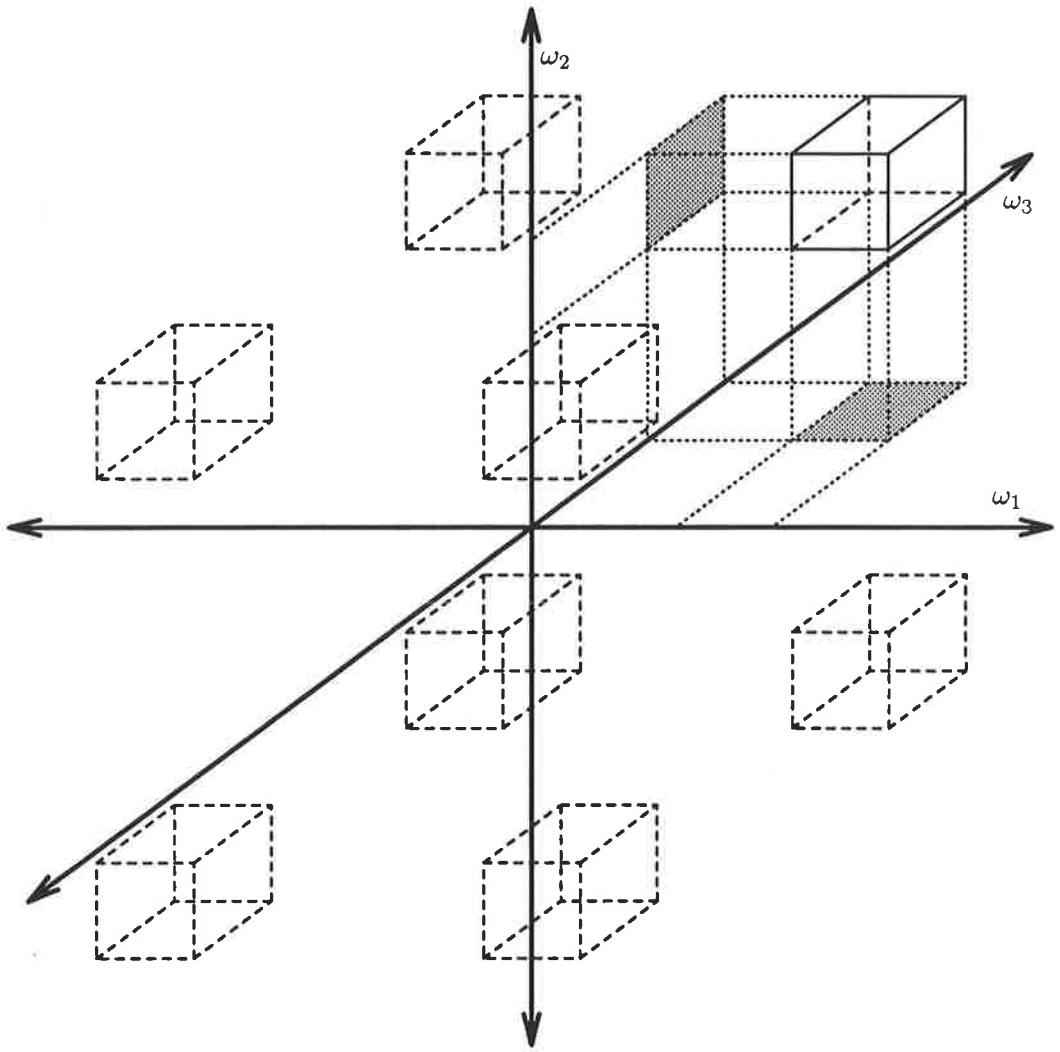


Figure 3.7: A cube of interest in 3 dimensional frequency space.

It is apparent from 3.32 that $\mathbf{Y}_n(\omega)$ is dependent on $\breve{\mathbf{Y}}_n(\omega_1, \omega_2, \ldots, \omega_n)$ although we don't need to find the exact relationship for our present purpose. If we now assume that x is bandpass, i.e.

$$\mathbf{X}_p(w) = 0, \quad \text{for } |\omega_p| < \omega_0 - \epsilon \text{ , or } |\omega_p| > \omega_0 + \epsilon, \tag{3.34}$$

then the output is independent of $\mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n)$ except for $\boldsymbol{\omega}$ within the hyper-cube defined by $\omega_0 - \epsilon < |\omega_p| < \omega_0 + \epsilon$ (see figure 3.7).

Let us now apply the limited frequency space of interest to the case of the

power series in amplitude and phase describing function models. We need to find $\mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n)$ for these two cases. Taking the n-dimensional Fourier transform of 3.26 and 3.27 results in

$$\mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n) = a_n, \tag{3.35}$$

for the case of a power series and

$$\mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n) = a_n + b_n \prod_{p=1}^{n} -j \operatorname{sgn}(\omega_p), \tag{3.36}$$

for the case of amplitude and phase describing functions.

The n-dimensional Fourier transform $\mathbf{H}_n(\omega_1, \ldots \omega_n)$ of any real $h_n(\tau_1, \ldots, \tau_n)$ will have its real part symmetric with respect to ω_p and its imaginary part anti-symmetric with respect to ω_p. This property is well known for single dimensional Fourier transforms, and its extension to n-dimensional Fourier transforms follows by considering an n-dimensional transform as n single dimensional Fourier transforms. It follows therefore that *any* system representable by a Volterra series can be approximated arbitrarily well by amplitude and phase describing functions over a narrow enough band width as long as the limit as $\epsilon \to 0$ of $\mathbf{H}_n(\omega_0 + \epsilon, \ldots, \omega_0 + \epsilon)$ exists.

As a direct consequence, testing a system with a sinusoid (a signal of zero band width) is insufficient to determine whether a system can adequately be modelled by amplitude and phase describing functions (unless, of course, the model will only be used with a sinusoidal input). Both 3.35 and 3.36 are independent of ω_i and it might be thought that testing with a sinusoid at a number of different ω_0 would be sufficient to check whether amplitude and phase describing functions are an adequate description. This is *not* the case however. The response of a system to a sinusoid of every frequency within the band of interest depends on $\mathbf{H}_n(\omega_1, \ldots, \omega_n)$ along a line, $\omega_1 = \omega_2 = \ldots = \omega_n = \omega_0$, through the hyper-cubes of interest in the n-dimensional frequency space. As an example consider the third order system defined by

$$\mathbf{H}_1(\omega_1) = a_1,$$

$$\mathbf{H}_3(\omega_1, \omega_2, \omega_3) = a_3((\omega_1^2 - \omega_2^2) + (\omega_2^2 - \omega_3^2) + (\omega_3^2 - \omega_1^2)). \qquad (3.37)$$

This system 3.37 shows no distortion at all when tested with a sine wave since the third order term vanishes when $\omega_1 = \omega_2 = \omega_3 = \omega_0$ and yet the system is certainly not distortionless. It is also worth noting that a system might have zero memory for its linear term and have a long memory for its higher order terms. Such a system could have a very large small signal band width but for larger signals would require a distortion model with memory.

As a further example (possibly easier to visualize) refer to figure 3.8. If a single sinusoid is input to the system in figure 3.8, then it will only pass through one of the bandpass filters and hence at least one of the inputs of each of the multipliers will be zero and hence the nonlinear distortion will be nil. Similarly if the input is a sum of two sinusoids. It is only when the input has energy in each of the three bands that there is any nonlinear component in the output.

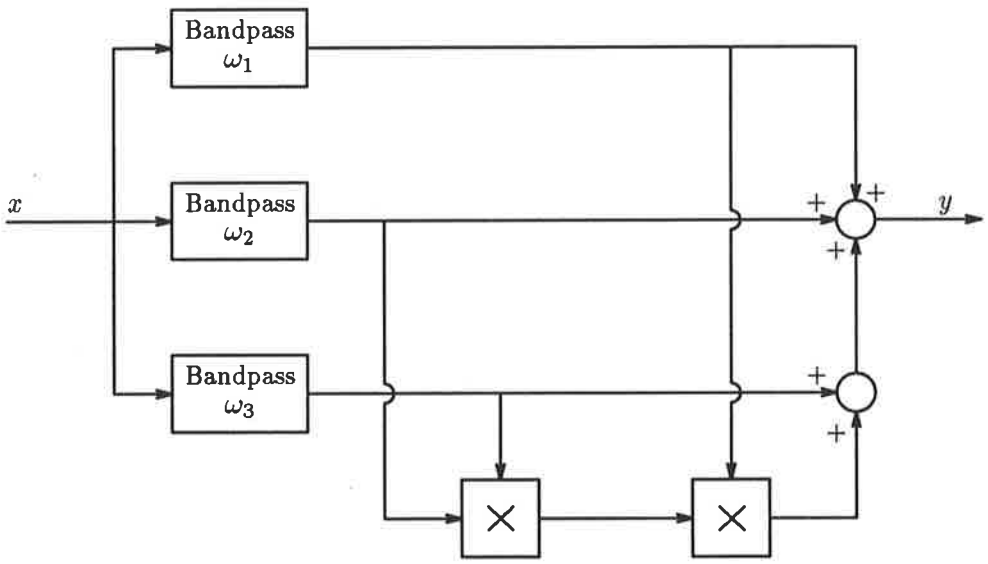


Figure 3.8: A hypothetical system.

Determining which model is satisfactory can be done by determining the Volterra kernels for the general model and observing whether or not they approximate 3.26 or 3.27. Identifying the Volterra kernels is a difficult exercise

which has received some attention in the literature [4] [68]. Here we shall do no more than consider the minimum complexity signal which could be used to identify the Volterra kernels. Since there are n independent variables in $\mathbf{H}_3(\omega_1, \ldots, \omega_n)$ it is reasonable to consider an input of n sinusoids whose transform has the form

$$\mathbf{X}(\omega) = \sum_{p=1}^{n} [a_n \delta(\omega - \gamma_p) + a_n^* \delta(\omega + \gamma_p)]. \tag{3.38}$$

When substituted in 3.34, $\breve{\mathbf{Y}}_n(\omega_1, \omega_2, \ldots \omega_n)$ consists of a sum of $(2n)^n$ terms, some of which are like

$$a_1 a_2 \cdots a_n \mathbf{H}_n(\gamma_1, \gamma_2, \ldots \gamma_n). \tag{3.39}$$

Actually isolating terms like 3.39 from all the rest of the terms requires a testing strategy which is cumbersome to devise and impractical to implement (see [4] for some special cases). However, the existence of terms like 3.39 show that n sinusoids of frequency γ_p can be used to identify an nth order system at most. It is important to note that the vector $(\gamma_1, \gamma_2, \ldots, \gamma_n)$ must be set to every point in the hypercube of interest within the n-dimensional frequency space defined by $\omega_0 - \epsilon < \gamma_p < \omega_0 + \epsilon$ in order to competely characterize $\mathbf{H}_n(\omega_1, \omega_2, \ldots \omega_n)$.

3.8 Summary

The distortion in systems modelled by power series, amplitude and phase describing functions and Volterra series has been analyzed for the case of bandpass inputs and outputs. In all cases it has been shown that distortion in a bandpass context can be considered as distortion of the complex envelope of the signal. This is of significance from the point of view of distortion correction in communication systems, since it means that distortion can be corrected at baseband or at some intermediate frequency as effectively as it can be at RF.

The power series model was shown to be a special case of the amplitude and phase describing function model. Both the power series model and the amplitude and phase describing function model were shown to be special cases of the Volterra series model and the form of the Volterra kernels was deduced. The Wiener model of nonlinearities has been derived.

From consideration of the frequency domain representation of the Volterra kernels it was shown that, for band limited inputs, the Volterra kernels of the model and of the actual system need only match for ω within a hyper-cube of interest. This is analogous to requiring the model of a linear system to match the actual transfer function over a frequency range of interest.

It was also possible to show that for a sufficiently narrowband input signal, any distorting system describable by a Volterra series could be approximated by the amplitude and phase describing function model. It was also shown that testing with a single sinusoid, even at a number of different frequencies, is inadequate to check whether amplitude and phase describing functions are an adequate model or whether a more general Volterra series model is necessary. Frequency independence of the amplitude and phase describing functions, as measured by a single sinusoid is a necessary, but not sufficient condition for the amplitude and phase describing function model to be applicable.

Chapter 4

Multi-input Systems

4.1 Introduction

The work in this chapter was mostly covered in an earlier paper by myself [23]. Radio communication systems make extensive use of mixers to perform frequency translation. A discussion of the architecture of receivers is beyond the scope of this chapter but high performance receivers typically have two or three separate frequency conversion operations.

The distortion of a signal going through a time invariant single input single output system can be modelled as a distortion of the complex envelope of the signal [8][24]. If the frequency translation operations are ideal (but the rest of the system is not) then the complex envelope of the signal is unchanged by the frequency translation process and distortion of a signal going through the whole system can be modelled as a distortion of the complex envelope of the signal.

Unfortunately real mixers are not perfect and contribute to the distortion of the signal. Further, in many modern communication systems the mixer does not perform a simple frequency conversion. In radar systems the local

oscillator might be a "chirp" (linear frequency sweep) signal, and in frequency hopping spread spectrum systems the local oscillator might follow a series of step functions in frequency. These two factors prevent the treatment of the distortion caused by a receiver as a single input, single output, time-invariant distortion of the complex envelope of the signal.

A typical communication system is, in reality, a multiple input device where the inputs consist of the signal being processed and all the local oscillator signals. It is of interest therefore to determine a model of minimum complexity, which is still sufficiently general to represent such systems adequately.

In section 4.2 of this chapter a simple model of distortion in mixers will be presented and it will be shown to be capable of representing the distortion in a mixer under restricted conditions.

The Wiener model of a nonlinear system presented in chapter 3.5 will then be extended to systems with more than one input. The extended Wiener model (which is quite general) will then be compared with the simple model presented in section 4.2 and the conditions under which the simple model is adequate will thereby be established.

In many systems the phases of the local oscillators relative to the input signal and to each other are not controlled. In these cases the absolute phase of the output is not important. If the phase of the local oscillator changes by a constant amount, the phase of the desired output signal changes by a constant amount. It will be shown that under appropriate bandpass conditions the distortion components also change phase by a constant amount. This is important because otherwise the spectrum (and maybe even the total power) of the distortion would depend on the phase relationship between the input signal and the local oscillator and it would impossible to devise a deterministic model.

Mixers are almost always used in a bandpass context. That is to say that a

mixer's inputs are bandpass and that its output is bandpass filtered (or at least that frequency components outside a certain band of interest can be ignored). In this chapter only the bandpass case will be considered.

The two input bandpass context is defined the same way as single input bandpass context (see section 1.1.2) except there is an ideal filter in each input. In the case of two-input bandpass systems, the output will depend on the frequencies of the inputs and of the filters centre frequencies. The output filter is normally arranged to be centred on either the sum or the difference of the two input frequencies. It is possible that harmonics of an input signal will also be within the output passband. We assume in this chapter, that the system is designed so that this does not occur.

4.2 A Simplified Model Of Two Input Nonlinearities

The distortion specifications quoted by most mixer manufacturers are the two tone intermodulation intercept points and the compression point. Since a distortion with memory can not be adequately tested by one or two tone measurements [24], manufacturers and designers are implicitly assuming that mixers are zero memory devices.

Further, since the local oscillator signal does not appear as a parameter in the distortion specification (except to specify how large a signal it should be), the mixer is implicitly being treated as a single input device. Certainly the specifications based on these assumptions are convenient from the point of view of the designer, but the question of how valid the assumptions are, remains.

Figure 4.1 illustrates such a system. The multiplier represents an ideal mixer and N' and N'' represent single input, single output distortions. First assume

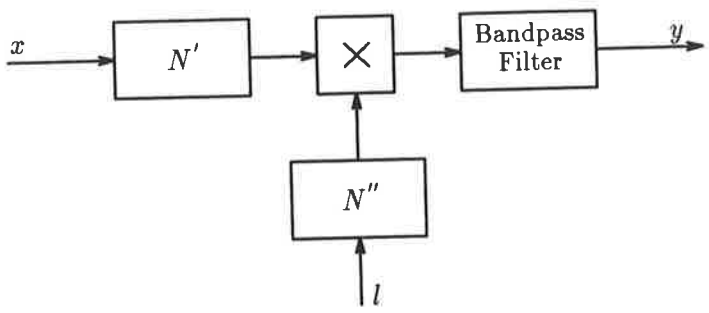


Figure 4.1: A simple model of a mixer.

that the distortion $N'(x(t))$ is a zero memory distortion in accordance with the widely adopted assumptions noted above. Also assume that $l(t)$ is a pure sinusoid. In this case the nature of $N''(l(t))$ is not important since a distorted sine wave always has a sinusoidal in band component. Finally assume $l(t)$ has a frequency ω_l and that $x(t)$ is bandpass and of a centre frequency ω_x such that $n\omega_l + m\omega_x$ is not within the filter's passband for any $|n| < p$ and any $|m| < p$ except for $n = m = \pm 1$ where p is the maximum order distortion present. This last assumption essentially states that only the upper sideband of the mixer output is passed by the bandpass filter. It would be a trivial matter to reformulate the problem for the case where only the lower sideband was being selected.

A general two input zero memory distortion can be modelled as a power series in two variables

$$\begin{aligned} y(t) &= \sum_{\substack{i\geq 0 \\ j\geq 0 \\ i+j\leq p}} a_{i,j} x^i(t) l^j(t), \\ &= \sum_{\substack{i\geq 0 \\ j\geq 0 \\ i+j\leq p}} A^j a_{i,j} x^i(t) \cos^j(\omega_l t + \phi). \end{aligned} \tag{4.1}$$

The local oscillator component represented by $A^j \cos^j(\omega_l t + \phi)$ can be expressed as a sum of harmonics of $\cos(\omega_l t + \phi)$. Similarly, since $x(t)$ is bandpass with centre frequency ω_x, then $x^i(t)$ will have components centred around the har-

monics of ω_x. The only terms which have energy within the passband are of the form

$$\begin{aligned} y(t) &= \sum_{\substack{i\geq 0 \\ j\lesseqgtr 0 \\ i+j\leq p}} A^j a_{i,j} x^i(t) \cos(\omega_l t + \phi) \\ &= \cos(\omega_l t + \phi) \sum_{\substack{i\geq 0 \\ i\lesseqgtr p}} a_i x^i(t), \end{aligned} \tag{4.2}$$

where

$$a_i = \sum_{\substack{j\geq 0 \\ i+j\leq p}} A^j a_{i,j}. \tag{4.3}$$

Equation 4.2 is in the form of a product of the local oscillator signal and the distorted signal. The same result is derived rather more rigorously in appendix C using a complex envelope representation of $x(t)$. Also observe that in 4.2 the local oscillator signal has not been distorted and so $N''(l(t)) = l(t)$. It has been shown, therefore, that a two input device with zero memory and a pure sinusoid applied to one input can, in a bandpass context, be modelled as a combination of a single input distortion and an ideal mixer (multiplier). Note, however, that the distortion is dependent on the *amplitude* of the local oscillator.

A nonlinearity with memory implies that the output at any instant is dependent on the inputs at all preceding instants within the memory of the system. Such a system is rather more difficult to describe in a general way.

4.3 The Wiener Model Of Multi-input Nonlinearities

Before the Wiener model can be applied to two input devices such as mixers, it must be extended. A similar argument will be used for the two input case as was used for the one input case (chapter 3.5). For the purposes of this chapter, there is no advantage in representing f of equation 3.18 as a series of orthogonal functions so the simpler power series representation will used.

The two inputs $x(t)$ and $l(t)$ can both be represented by a series of orthonormal functions

$$x(t-\tau_x) = \lim_{M\to\infty} \sum_{m=1}^{M} a_m(t)\alpha_m(\tau_x), \tag{4.4}$$

where

$$a_m(t) = \int_0^T \alpha_m(\tau)x(t-\tau_x)\,d\tau_x, \tag{4.5}$$

and

$$l(t-\tau_l) = \lim_{N\to\infty} \sum_{n=1}^{N} b_n(t)\beta_n(\tau_l), \tag{4.6}$$

where

$$b_m(t) = \int_0^T \beta_m(\tau_l)x(t-\tau_l)\,d\tau_l. \tag{4.7}$$

All the information about the two inputs over the interval $[t-T, t]$ is contained in the current values of $a_m(t)$ and $b_n(t)$ so that the output of the system can be expressed

$$y(t) = f(a_1(t), a_2(t), \ldots, a_M(t), b_1(t), b_2(t), \ldots, b_N(t)).$$

The function, f, is of the same form as for the single input case (but with more variables) and the same comments regarding implementation apply. The operation performed by the function f is represented by the blocks in sections B and C of figure 4.2.

4.3.1 The Two Input Wiener Model Applied to Mixers

It is convenient to choose the impulse responses $\alpha_m(t)$ and $\beta_n(t)$ to be sinusoids over the interval $[0, T]$. These impulse responses will be harmonics of $\sin(2\pi t/T)$ and $\cos(2\pi t/T)$. The blocks in section A of figure 4.2 become, therefore, narrowband filters whose band width tends towards zero as T tends towards infinity. Since the signals $x(t)$ and $l(t)$ are assumed to be bandpass, then any of the filters α and β which have a passband outside the band of the signals need not

be included in the model. The impulse responses of the filters which are within the passband can be described

$$\alpha_m(t) = \begin{cases} \cos\left[\left(\omega_x + \Delta\omega\dfrac{(m-1)}{2}\right)t\right], \\ \qquad\qquad \text{for } m \text{ odd and } 0 \le t \le T; \\ \cos\left[\left(\omega_x + \Delta\omega\dfrac{(m-2)}{2}\right)t + \pi/2\right], \\ \qquad\qquad \text{for } m \text{ even and } 0 \le t \le T; \\ 0, \qquad\qquad \text{otherwise;} \end{cases} \tag{4.8}$$

and

$$\beta_n(t) = \begin{cases} \cos\left[\left(\omega_l + \Delta\omega\dfrac{(n-1)}{2}\right)t\right], \\ \qquad\qquad \text{for } n \text{ odd and } 0 \le t \le T; \\ \cos\left[\left(\omega_l + \Delta\omega\dfrac{(n-2)}{2}\right)t + \pi/2\right], \\ \qquad\qquad \text{for } n \text{ even and } 0 \le t \le T; \\ 0, \qquad\qquad \text{otherwise,} \end{cases} \tag{4.9}$$

where

$$\Delta\omega = 2\pi/T.$$

The outputs of the filters $a_m(t)$ and $b_n(t)$ are narrowband and tend towards being sinusoidal as T tends towards infinity (the $a_n(t)$ and the $b_m(t)$ then become a representation of one sided Fourier transform of the signal). In a practical implementation it would be undesirable to let T be larger than necessary because of the associated increase in the complexity of the model. To develop the theoretical results in the rest of the chapter, however, the limiting case $T \to \infty$ will be used and the outputs of the filters in section A of figure 4.2 can be written

$$a_m(t) = \cos(\zeta_m),$$

where

$$\zeta_m = \begin{cases} \left(\omega_x + \Delta\omega\frac{(m-1)}{2}\right)t + \theta_m, & \text{for } m \text{ odd;} \\ \left(\omega_x + \Delta\omega\frac{(m-2)}{2}\right)t + \pi/2 + \theta_m, & \text{for } m \text{ even;} \end{cases}$$

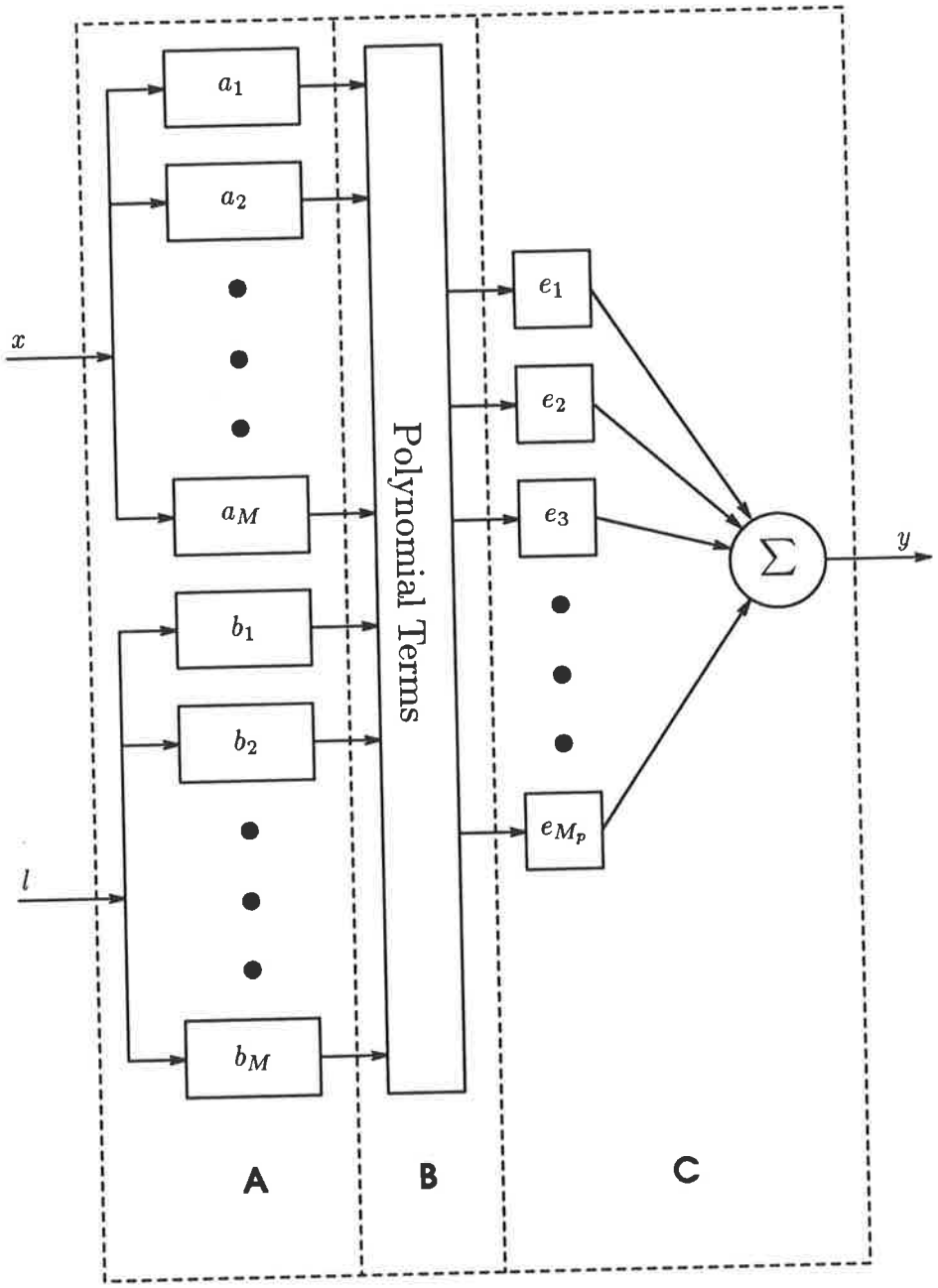


Figure 4.2: The two input Wiener model.

(4.10)

and

$$b_n(t) = \cos(\eta_m),$$

where

$$\eta_n = \begin{cases} \left(\omega_l + \Delta\omega\frac{(n-1)}{2}\right)t + \phi_n, & \text{for } n \text{ odd;} \\ \left(\omega_l + \Delta\omega\frac{(n-2)}{2}\right)t + \pi/2 + \phi_n, & \text{for } n \text{ even.} \end{cases} \tag{4.11}$$

It is now only necessary to write down a representation for the function $f\left(a_1(t), a_2(t), \ldots, a_M(t), b_1(t), b_2(t), \ldots, b_N(t)\right)$ to complete the model. We choose a multi dimensional power series,

$$\begin{aligned} f(a_1, a_2, \ldots, a_M, \quad &= \lim_{p\to\infty} \sum_{\substack{j_1=0 \\ j_2=0 \\ \vdots \\ j_M=0}}^{p} \sum_{\substack{k_1=0 \\ k_2=0 \\ \vdots \\ k_N=0}}^{p} (e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N} \\ b_1, b_2, \ldots, b_N) \quad &\qquad \times a_1^{j_1} a_2^{j_2} \cdots a_M^{j_M} \\ &\qquad \times b_1^{k_1} b_2^{k_2} \cdots b_N^{k_N}). \end{aligned} \tag{4.12}$$

The $e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N}$, in equation 4.12 are constant coefficients. Since the j_m and k_n range from 0 to p then every possible term of order less than or equal to p is present. However, equation 4.12 also contains terms of order greater than p. For example when every variable is raised to the maximum power p then the term is of order $(N+M)^p$. There is no loss of generality in including these terms since the associated coefficients can always be set to zero if a finite order system is to be represented.

This model will now be used to show that a two input nonlinearity with memory can not, in general, be put in the form of figure 4.1.

Substituting 4.10 in 4.12 and setting all the $b_n = 0$, the single input system

$N'(x(t))$ can be written

$$N'\left(x(t)\right) = \sum_{\substack{j_1=0\\ j_2=0\\ \vdots\\ j_M=0}}^{p} \left(c_{j_1,j_2,\ldots,j_M} \cos^{j_1}(\zeta_1)\cos^{j_2}(\zeta_2)\cdots\cos^{j_M}(\zeta_M)\right), \tag{4.13}$$

and substituting 4.11 in 4.12 and setting all the $a_m = 0$, the single input system $N''(l(t))$ can be written

$$N''\left(l(t)\right) = \sum_{\substack{k_1=0\\ k_2=0\\ \vdots\\ k_N=0}}^{p} \left(d_{k_1,k_2,\ldots,k_N} \cos^{k_1}(\eta_1)\cos^{k_2}(\eta_2)\cdots\cos^{k_N}(\eta_N)\right). \tag{4.14}$$

The output of a system of the form in figure 4.1 can be written as the product of equations 4.13 and 4.14

$$\begin{aligned} N'\left(x(t)\right)N''\left(l(t)\right) = \sum_{\substack{j_1=0\\ j_2=0\\ \vdots\\ j_M=0}}^{p} \sum_{\substack{k_1=0\\ k_2=0\\ \vdots\\ k_N=0}}^{p} \Big(&c_{j_1,j_2,\ldots,j_M} d_{k_1,k_2,\ldots,k_N} \\ &\times \cos^{j_1}(\zeta_1)\cos^{j_2}(\zeta_2)\cdots\cos^{j_M}(\zeta_M) \\ &\times \cos^{k_1}(\eta_1)\cos^{k_2}(\eta_2)\cdots\cos^{k_N}(\eta_N)\Big). \end{aligned} \tag{4.15}$$

and a general two input nonlinearity can be written

$$\begin{aligned} N\left(x(t),l(t)\right) = \sum_{\substack{j_1=0\\ j_2=0\\ \vdots\\ j_M=0}}^{p} \sum_{\substack{k_1=0\\ k_2=0\\ \vdots\\ k_N=0}}^{p} \Big(&e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N} \\ &\times \cos^{j_1}(\zeta_1)\cos^{j_2}(\zeta_2)\cdots\cos^{j_M}(\zeta_M) \\ &\times \cos^{k_1}(\eta_1)\cos^{k_2}(\eta_2)\cdots\cos^{k_N}(\eta_N)\Big). \end{aligned} \tag{4.16}$$

Clearly 4.15 is of the form of 4.16 if, and only if

$$e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N} = c_{j_1,j_2,\ldots,j_M} d_{k_1,k_2,\ldots,k_N},$$

$$\text{for } 0 \le j_m \le p \text{ and } 0 \le k_n \le p. \tag{4.17}$$

We will now show that in most cases, there are more simultaneous equations 4.17 defining $e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N}$ than there are independent variables ($c_{j_1,j_2,\ldots,j_M}$ and $d_{k_1,k_2,\ldots,k_N}$), so a solution is not generally possible. There are $(p+1)^M(p+1)^N$ equations in 4.17 describing $(p+1)^M(p+1)^N$ values of $e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N}$ and there are $(p+1)^M + (p+1)^N$ independent variables. The condition for being able to represent a two input nonlinearity in the form of figure 4.1 is, therefore,

$$(p+1)^M(p+1)^N \leq (p+1)^M + (p+1)^N. \tag{4.18}$$

Many of the $e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N}$ are coefficients of terms which do not have any energy in the passband. If the passband constraint is taken into account there will be fewer equations of the form of those in 4.17 but there will also fewer independent variables. Any $e_{j_1,j_2,\ldots,j_M,k_1,k_2,\ldots,k_N}$ which correspond to terms of 4.16 which do not have a component at a frequency near $\pm(\omega_x + \omega_l)$ can be ignored. The product of cosines in 4.16 can be converted to a sum of cosines by repeated application of the identity $\cos(A)\cos(B) = (\cos(A+B)+\cos(A-B))/2$, this, together with 4.10 and 4.11 implies that the only way that frequencies within the passband can be produced is if $\sum_{i=1}^{M} j_i$ is odd and $\sum_{i=1}^{N} k_i$ is also odd.

For the case of p odd, then the j_i will take on as many odd values as even values and the $\sum_{i=1}^{M} j_i$ will be odd in as many cases as it is even. Similarly $\sum_{i=1}^{N} k_i$ will be odd as often as it is even. The number of equations 4.17 then becomes $\frac{1}{2}(p+1)^M \cdot \frac{1}{2}(p+1)^N$ and the number of independent variables is $\frac{1}{2}(p+1)^M + \frac{1}{2}(p+1)^N$. A very similar argument can be applied to the case of p even. The conditions for representing a bandpass two input system in the form of figure 4.1 then is

$$\frac{(p+1)^M}{2}\frac{(p+1)^N}{2} \leq \frac{(p+1)^M}{2} + \frac{(p+1)^N}{2}, \quad \text{for } p \text{ odd;} \tag{4.19}$$

and

$$\frac{\left[(p+1)^M-1\right]}{2}\frac{\left[(p+1)^N-1\right]}{2} \leq \frac{(p+1)^M-1}{2}+\frac{(p+1)^N-1}{2},$$

for p even. (4.20)

This is only true for trivial cases. For example, if $p = 1$, $M = 2$ and $N = 2$ then 4.19 holds but not for anything more complex.

Neither in the bandpass case nor in the wideband case, can a general two input nonlinearity be represented in the form of figure 4.1.

4.3.2 Sinusoidal Local Oscillators

The case of a sinusoidal local oscillator can be treated by putting $N = 1$ in equations 4.19 and 4.20. However, this results in a more restrictive interpretation than is really required. In applications where there is a sinusoidal local oscillator, the amplitude is usually fixed and it is acceptable if the nonlinearity $N'(x(t)$ is a function of the local oscillator amplitude.

In the bandpass case, we are only interested in terms of equations 4.15 and 4.16 which have $\sum_{i=1}^{M} j_i$ is odd and k_1 is also odd. Suppose the local oscillator is $l(t) = A\cos\eta_1$. Note, from equation 2.8, that

$$l(t)^{k_1} = \left(\frac{A}{2}\right)^{k_1} \sum_{i=0}^{(k_1-1)/2} \binom{k_1}{i} \cos\left((k_1-2i)(\eta_1)\right) \tag{4.21}$$

and since, in the bandpass case, we are only interested in the fundamental where $k_1 - 2i = 1$, equation 4.21 becomes

$$x(t)^{k_1} = \left(\frac{A}{2}\right)^{k_1} \binom{k_1}{i} \cos\left(\eta_1\right). \tag{4.22}$$

We can re-use the work in section 4.3.1 if we write

$$d_{k_1} = \left(\frac{A}{2}\right)^{k_1} \binom{k_1}{i}, \tag{4.23}$$

substitute equations 4.21 and 4.23 into equation 4.16 and factorize

$$N\left(x(t), l(t)\right) = \cos(\eta_1) \sum_{\substack{j_1=0 \\ j_2=0 \\ \vdots \\ j_M=0}}^{p} \sum_{k_1=1}^{p} e_{j_1,j_2,\ldots,j_M,k_1} \times \cos^{j_1}(\zeta_1)\cos^{j_2}(\zeta_2)\cdots\cos^{j_M}(\zeta_M), \tag{4.24}$$

and writing

$$\sum_{k_1=1}^{p} e_{j_1,j_2,\ldots,j_M,k_1} = e_{j_1,j_2,\ldots,j_M} \tag{4.25}$$

gives

$$N\left(x(t), l(t)\right) = \cos(\eta_1) \sum_{\substack{j_1=0 \\ j_2=0 \\ \vdots \\ j_M=0}}^{p} e_{j_1,j_2,\ldots,j_M} \cos^{j_1}(\zeta_1)\cos^{j_2}(\zeta_2)\cdots\cos^{j_M}(\zeta_M). \tag{4.26}$$

equation 4.26 is clearly of the form of figure 4.1. Note the the $e_{j_1,j_2,\ldots,j_M}$ depend on the amplitude of the local oscillator.

4.4 Phase Transfer

In many practical mixer applications, the phase relationship between the local oscillator and the signal is not controlled. Equation 4.16 can still be used to describe the output if the η_k are redefined

$$\eta_m = \begin{cases} \left(\omega_x + \Delta\omega\frac{(n-1)}{2}\right)t + \phi_n + \gamma, & \text{for } n \text{ odd;} \\ \left(\omega_x + \Delta\omega\frac{(n-2)}{2}\right)t + \pi/2 + \phi_n + \gamma, & \text{for } n \text{ even;} \end{cases} \tag{4.27}$$

where γ is the phase difference between the local oscillator $l(t)$ and the signal $x(t)$.

The product of cosines in 4.16 can be converted to a sum of cosines using the identity $\cos(A)\cos(B) = (\cos(A+B)+\cos(A-B))/2$. The terms then have the form $\cos(\Phi)$ where

$$\begin{aligned}
\Phi = &\pm\zeta_1 \pm \zeta_1 \cdots && (j_1 \text{ terms}) \\
&\pm\zeta_2 \pm \zeta_1 \cdots && (j_2 \text{ terms}) \\
&\vdots && \vdots \\
&\pm\zeta_M \pm \zeta_M \cdots && (j_M \text{ terms}) \\
&\pm\eta_1 \pm \eta_1 \cdots && (k_1 \text{ terms}) \\
&\pm\eta_2 \pm \eta_1 \cdots && (k_2 \text{ terms}) \\
&\vdots && \vdots \\
&\pm\eta_N \pm \eta_N \cdots && (k_N \text{ terms}).
\end{aligned} \tag{4.28}$$

Substituting 4.10 and 4.27 in 4.28

$$\Phi = r\omega_x t + s\omega_l + \epsilon t + v\pi/2 + s\gamma. \tag{4.29}$$

where r can be interpreted as the net number of ζ_m's added together and s can be interpreted as the net number of η_n's added together. The ϵ in 4.29 is a small frequency deviation which results from the addition and subtraction of the $\Delta\omega$ terms in 4.10 and 4.27 and v is the net number of $\pi/2$ terms which have been added together. Only the terms where $r = s = \pm 1$ have a frequency near $\pm(\omega_x + \omega_l)$ and

$$\Phi = \pm(\omega_x t + \omega_l + \gamma) + \epsilon t + v\pi/2. \tag{4.30}$$

Thus a change in phase of the local oscillator of γ results in a uniform change in phase of all the output components within the passband.

It is *not* true in a *broad*-band system that a change in phase of the local oscillator of γ results in a uniform change in phase of all the output components. For example, the phase of the fifth harmonic of a signal changes phase by five times as much as the change in phase of the signal. This is clearly illustrated by considering the simple two input system in figure 4.3. Where N is given by

$$y = \begin{cases} z, & \text{for } -1 < z < 1; \\ \frac{1}{2}(z+1), & \text{for } z \geq 1; \\ \frac{1}{2}(z-1), & \text{for } z \leq -1; \end{cases} \tag{4.31}$$

$$z = x + l. \tag{4.32}$$

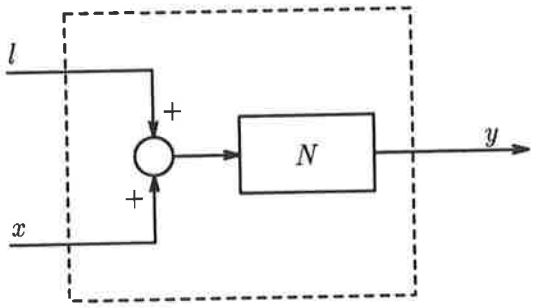


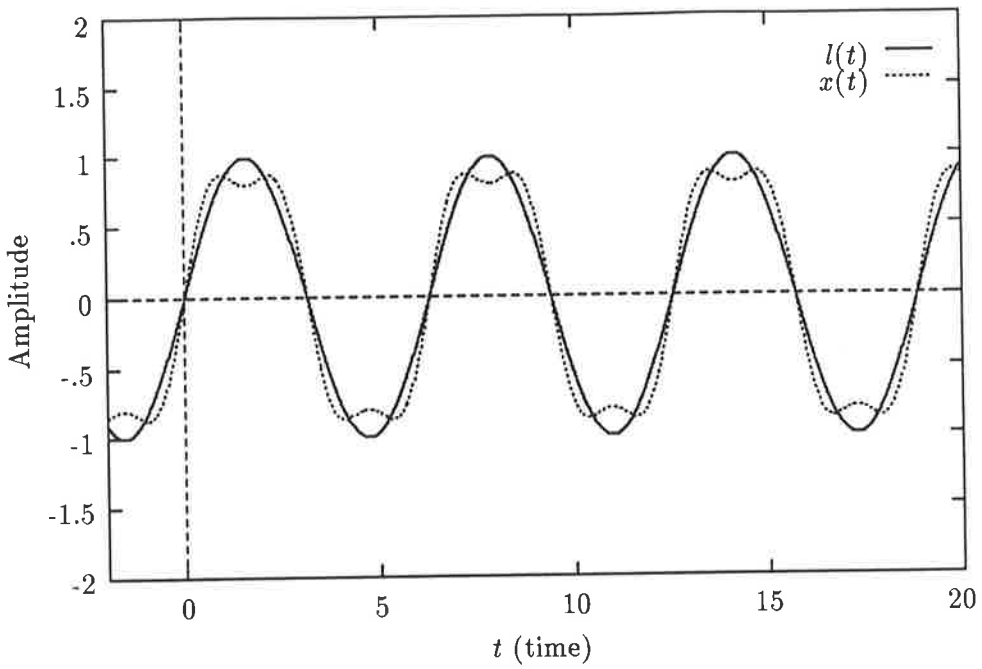
Figure 4.3: A simple two input non-linearity. The nonlinearity, N, is defined by equation 4.32

Consider two different cases. In the case of figure 4.4a, the sum of the signals l and x clearly exceeds the linear range of N (equation 4.32). In the case of figure 4.4b, the phase of $l(t)$ has changed 180° and the sum of the signals l and x clearly is within the linear range of N (equation 4.32). Thus, in this case the distortion is very much dependent on the phase of the local oscillator system. Note that this combination of signals and system does not fit the extended definition of bandpass systems in 4.1.

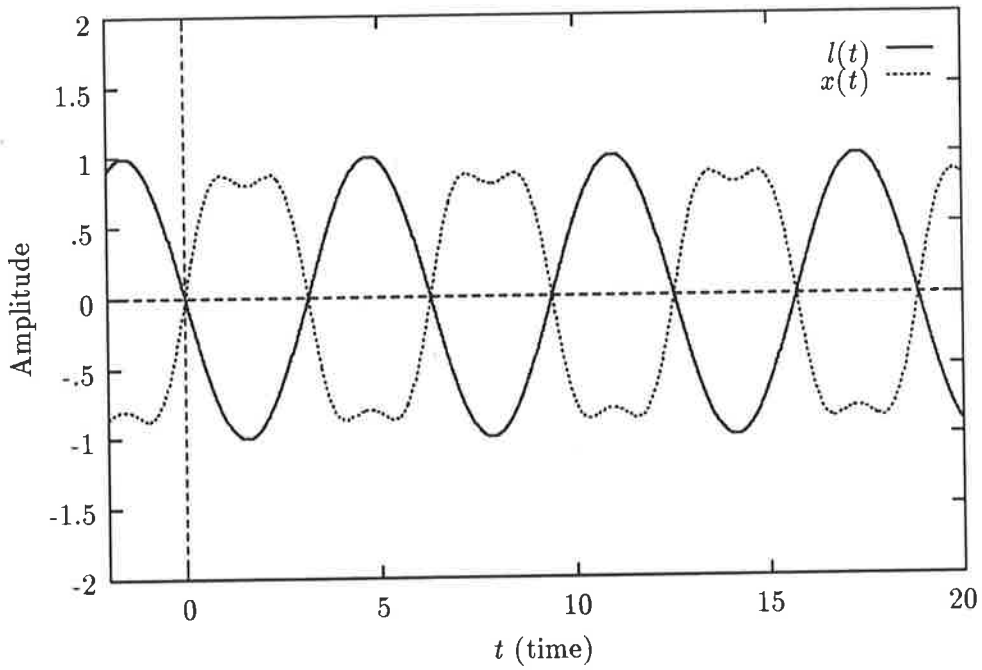
4.5 Summary

It was observed that mixer manufacturers' linearity specifications implicitly assume a rather simple model of distortion in a two input device. This assumption has been critically examined and found to be invalid if the mixer does not have a "pure"[1] sinusoidal local oscillator.

[1] A distorted sinusoid is satisfactory since the distortion can be considered part of the block N'' in figure figure 4.1. A "broadened" sinusoid (perhaps due to amplitude or phase modulation) is not satisfactory.



(a)



(b)

Figure 4.4: Phase dependent distortion. In case (a), the sum of the two signals exceeds the linear range of the non-linearity whereas the sum of the two signals in case (b) does not exceed the linear range.

Most mixer applications employ a sinusoidal local oscillator but some modern mixer applications use non-sinusoidal local oscillator signals such as a linear frequency sweep (a "chirp") and caution must be exercised in applying performance measurements made with a sinusoidal local oscillator to these applications. It may be that a "chirp" changed frequency slowly enough to be considered a sinusoid, but this would depend on the mixer and the frequency sweep rate. The author speculates that useful relationships might arise from studying the case of constant envelope signals as local oscillators.

It was shown that in the case of a bandpass system, a change in phase of one of the input signals results in the same change in phase of all components of the output signal within the passband. This result is of practical importance in the modelling of distortion in a receiver where it is impossible or impracticable to measure the local oscillator signals.

Chapter 5

Nonlinear Digital Filters

5.1 Introduction

Nonlinear digital filters can be considered a discrete time version of the Volterra series discussed in section 3.4. Many of the concepts of linear digital filters can be applied to nonlinear digital filters. In particular, the thorny problem of estimating the Volterra kernels can be performed "automatically" by an adaptive nonlinear digital filter.

Nonlinear digital filters can be used as predictors, estimators and equalizers in the same way as linear digital filters can be.

In this chapter we discuss the basis for nonlinear digital filters in the discrete time Volterra series. We discuss a number of adaption algorithms and their relative merits. Finally we discuss some of the features of a software implementation.

5.2 Discrete Time Volterra Series

To obtain the discrete time Volterra series, take equation 3.11 and put $x_n \equiv x(t-nT)$ and $h_{i_1 \ldots i_n} \equiv T^n h(i_1 T \ldots i_n T)$

$$\begin{aligned} H_n(x(t)) &\approx \sum_{i_1=-N}^{N} \cdots \sum_{i_n=-N}^{N} h_{i_1 \ldots i_n} \prod_{p=1}^{n} x_{i_p} \\ &\approx H_n(\ldots x_{-1}, x_0, x_1 \ldots), \end{aligned} \tag{5.1}$$

where the discrete time Volterra series is defined as

$$y_k = \sum_{n=0}^{\infty} H_n(\ldots x_{-1}, x_0, x_1 \ldots), \tag{5.2}$$

where

$$H_n(\ldots x_{-1}, x_0, x_1 \ldots) = \sum_{i_1=-N}^{N} \cdots \sum_{i_n=-N}^{N} h_{i_1 \ldots i_n} \prod_{p=1}^{n} x_{i_p},$$

and in the case of causal systems

$$H_n(\ldots x_{-1}, x_0, x_1 \ldots) = \sum_{i_1=0}^{N} \cdots \sum_{i_n=0}^{N} h_{i_1 \ldots i_n} \prod_{p=1}^{n} x_{i_p}. \tag{5.3}$$

Alternatively equation 5.2 can be expanded by substituting equation 5.3,

$$\begin{aligned} y_k = &\sum_{i_1=0}^{N} h_{i_1} x_{i_1} + \sum_{i_1=0}^{N} \sum_{i_2=0}^{N} h_{i_1, i_2} x_{i_1} x_{i_2} \\ &+ \cdots \sum_{i_1=1}^{N} \cdots + \sum_{i_n=1}^{N} h_{i_1 \ldots i_n} \prod_{p=1}^{n} x_{i_p} + \cdots . \end{aligned} \tag{5.4}$$

Equation 5.4 can be put in block diagram form (figure 5.1).

The values of the discrete time Volterra kernel $h_{i_1 \ldots i_n}$ can be interpreted as the coefficients of a multi-dimensional polynomial of the past input samples. We can redraw figure 5.1 in a form which emphasizes the similarity with a conventional linear transversal filter (figure 5.2).

The application of Volterra series to system modelling has been attributed to Wiener [79], but the form of equation 5.4 or of figure 5.2 seems to have been proposed by others. The first accessible publication appears to be by

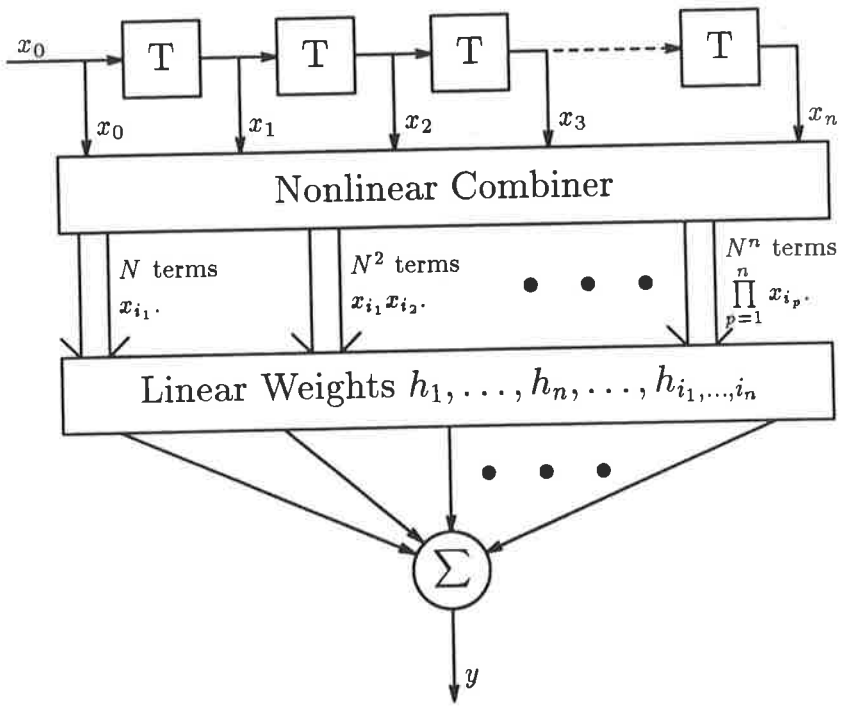


Figure 5.1: The discrete time volterra series.

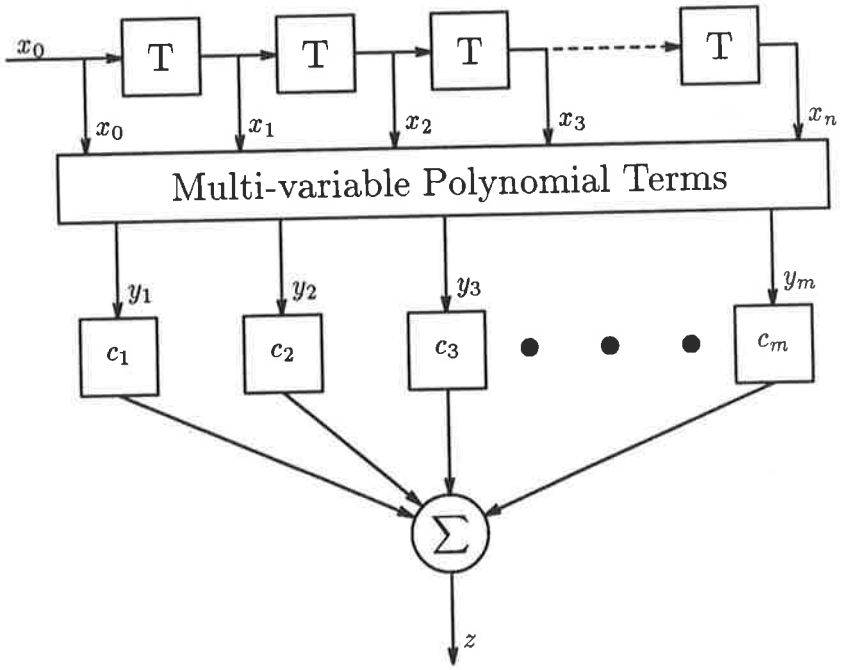


Figure 5.2: A nonlinear digital filter based on the Volterra series.

Gabor[1] [31].

The discrete time Volterra series can be made an arbitrarily close approximation to the continuous time Volterra series (equation 5.1) by increasing N and decreasing T. As has been stated in Chapter 3.4, any functional can be arbitrarily well approximated over a given range by a continuous time Volterra series. It follows that any functional can be approximated arbitrarily well by a discrete time Volterra series.

5.3 Nonlinear Adaptive filters

The form of a digital filter (linear or nonlinear) is relatively simple. The more difficult problem is to determine the parameters of the digital filter so that it has the desired characteristic.

Determining the Volterra kernels from an analytic description of the distortion is difficult (and may not always be possible in "closed form"). Further, in many applications, there might not even be an analytic description to start from. These difficulties can be neatly circumvented by the use of adaptive techniques [61].

There is some similarity between the nonlinear transversal filter (figure 5.2) and a linear transversal filter. The linear case can be obtained as a special case of the nonlinear filter by restricting the considering the "Multi-variable Polynomial Terms" box of figure 5.2 to first order, in which case $y_i = x_i$. For given y_i, the variation of z as a function of c_i is the same whether the

[1] To quote Gabor: "It is difficult to say how much of this is new, as so much written on the subject appears only in reports. From the abstracts circulated by the C.C.I.R it appears that nonlinear filters operating on samples were first considered by H.E. Singleton in 1951, and after him by W.D. White, also 1951, in reports of the General Electric Co., Schenectady and of the Airborne Instruments Lab., Mineola, respectively."

system is linear or nonlinear, since the nonlinear part is before the application of the weights c_i. Consequently much of the theory of linear adaptive filters is applicable to nonlinear adaptive filters. However, the more efficient recursive linear adaptive algorithms make use of the fact that $y_i(n) = y_{i-1}(n-1)$ which is not true for the nonlinear case. Consequently, the recursive adaptive filter algorithms need to be modified for the nonlinear case. See section 5.3.4.

5.3.1 Nonlinear Prediction Estimation and Equalization

In order to adaptively adjust filter parameters we need to know what the ideal output of the filter should be for a given input. Depending on how we obtain that desired filter output, the nonlinear filter can be used to perform different functions. Most commonly, those functions can be classified as estimation, equalization and prediction exactly the same as for the linear case.

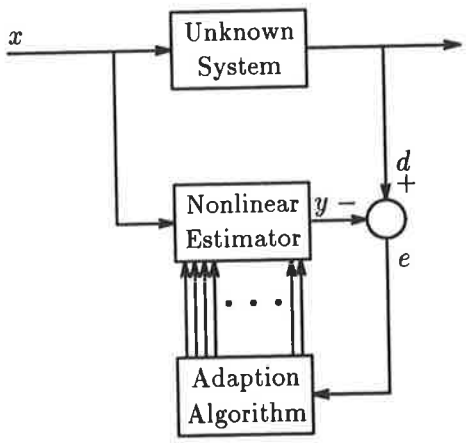


Figure 5.3: Estimation.

Estimation (figure 5.3) is where the nonlinear filter and an unknown system have the same input and the parameters of the adaptive filter are adjusted so as to mimic the output of the unknown system. The result of the exercise is the

values of the filter weights. Conceptually, the adaption process can be turned off once the weights have converged, and the adaptive filter is then a model of the system, which can be used for simulations of the system by applying synthetic inputs and observing the output.

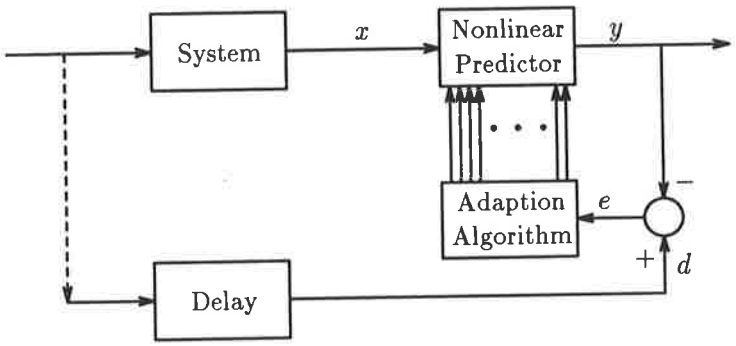


Figure 5.4: Equalization.

Equalization (figure 5.4) is where the output of the system to be equalized is the input to the adaptive filter and the input to the system to be equalized is the desired output of the adaptive filter. There is an apparent logical flaw in this arrangement: If the desired output is available, why not just use it directly instead of attempting to equalize a distorted signal? In practice, a known training sequence is used to adapt the filter and then the weights are frozen while the system has unknown inputs applied.

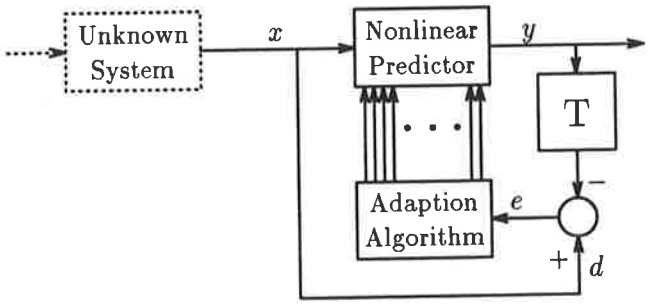


Figure 5.5: Prediction.

Prediction (figure 5.5) is where we use an adaptive filter to estimate future values of a sequence. The sequence is the input to the adaptive filter. We delay the last prediction of the filter and compare that with the current input to see how good the prediction was.

The adaption algorithms in figures 5.3–5.5 all act to minimize in some sense the signal e. In practice, e is not the only input to the adaption algorithm.

It is possible to have fixed equalizers and predictors, but it makes no sense to speak of a fixed estimator. In the case of predictors and equalizers, adaptive filters become necessary when the system to be equalized is time varying (but quasi-stationary), but even when the system is truly stationary, an adaptive filter is often the simplest way to determine the optimum filter weights.

Adaptive nonlinear estimators were probably first proposed by Gabor [31] [32][2]. Some representative uses of adaptive digital filters as estimators or correctors are reported in [28] [47] [33] [11] [8] [9].

5.3.2 Gradient Algorithms

The LMS gradient algorithm for the nonlinear case is identical to the linear case. We need only consider the linear combiner part of figure 5.2. A linear combiner can be written

$$y(n) = \boldsymbol{c}(n-1)^{*}\boldsymbol{x}(n). \tag{5.5}$$

The $*$ superscript is used to indicate the conjugate-transpose. Note the change of variable names from previous usage. This has been forced by simply running out of symbols (especially in section 5.3.4 and appendix D). Following

[2]The work by Gabor [32] was ambitious considering the technology available. The tapped delay line of figure 5.2 was implemented with a 20 track tape recorder with 20 independently positionable heads. Multiplication (for the generation of the nonlinear) terms was by piezomagnetic devices and the adaption used analogue computer techniques.

derivations do not make reference to previous sections of this chapter, so the confusion should be minimal.

The error is defined

$$e(n) = d(n) - \boldsymbol{y}(n), \tag{5.6}$$

and substituting equation 5.5, we get

$$e(n) = d(n) - \boldsymbol{c}(n-1)^* \boldsymbol{x}(n). \tag{5.7}$$

Equation 5.7 defines the error before the coefficients have been updated. *After* the coefficients have been updated, we can define a new error

$$\tilde{e}(n) = d(n) - \boldsymbol{c}(n)^* \boldsymbol{x}(n), \tag{5.8}$$

and a variant form,

$$\tilde{e}(n,k) = d(k) - \boldsymbol{c}(n)^* \boldsymbol{x}(k). \tag{5.9}$$

We want to choose the value of $\boldsymbol{c}(n)$ so as to minimize the mean squared value of $\tilde{e}(n,k), \quad 0 \leq k \leq n$. The cost function to be minimized is

$$\epsilon(n) = \sum_{k=0}^{n} \tilde{e}(n,k)\tilde{e}(n,k)^*. \tag{5.10}$$

We define the gradient of $\epsilon(n)$ as

$$\nabla\left[\epsilon(n)\right] = \begin{pmatrix} \dfrac{\partial \epsilon(n)}{\partial c_1} \\ \dfrac{\partial \epsilon(n)}{\partial c_2} \\ \vdots \\ \dfrac{\partial \epsilon(n)}{\partial c_m} \end{pmatrix}. \tag{5.11}$$

Differentiating equation 5.10 with respect to c_i,

$$\begin{aligned} \frac{\partial \epsilon(n)}{\partial c_i(n)} &= \sum_{k=0}^{n} \frac{\partial}{\partial c_i(n)} \left[\tilde{e}(n,k)\tilde{e}(n,k)^*\right] \\ &= \sum_{k=0}^{n} \frac{\partial \tilde{e}(n,k)}{\partial c_i(n)} \tilde{e}(n,k)^* + \left[\frac{\partial \tilde{e}(n,k)}{\partial c_i(n)} \tilde{e}(n,k)^*\right]^*. \end{aligned} \tag{5.12}$$

Differentiate equation 5.9,

$$\frac{\partial \tilde{e}(n,k)}{\partial c_i(n)} = \frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(k),$$

and substitute in equation5.12,

$$\frac{\partial \epsilon(n)}{\partial c_i(n)} = \sum_{k=0}^{n} \left(\frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(k) \tilde{e}(n,k)^* + \left[\frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(k) \tilde{e}(n,k)^* \right]^* \right). \quad (5.13)$$

The gradient algorithm involves adjusting the weight vector $\boldsymbol{c}$ in the direction of steepest descent. Equation 5.13 gives the direction of steepest descent, however, it is difficult to evaluate in practice since the $e(n,k)$ need to be re-evaluated for all k each time $\boldsymbol{c}$ is updated.

In practice the "LMS Stochastic Gradient Algorithm" (or more simply the "LMS Algorithm"), is used [38]. This involves simply approximating the gradient by the most recent term in the summation of equation 5.13,

$$\begin{aligned} \frac{\partial \hat{\epsilon}(n)}{\partial c_i(n)} &\approx \frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(n) \tilde{e}(n)^* + \left[\frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(n) \tilde{e}(n)^* \right]^* \\ &\approx 2\Re \left(\frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(n) \tilde{e}(n)^* \right). \end{aligned} \quad (5.14)$$

Now, adopting the notation that $\mathbf{0}_i$ is a null vector of length i,

$$\begin{aligned} \frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \boldsymbol{x}(n) \tilde{e}(n)^* &= \left(\mathbf{0}_{i-1}^* \,\middle|\, 1 \,\middle|\, \mathbf{0}_{m-i}^* \right) \begin{pmatrix} x_1(n) \\ \hline x_2(n) \\ \hline \vdots \\ \hline x_i(n) \\ \hline \vdots \end{pmatrix} \tilde{e}(n)^* \\ &= x_i(n) \tilde{e}(n)^*. \end{aligned} \quad (5.15)$$

Substitute equation 5.15 into equation 5.14 and (comparing with equation 5.11) we can write

$$\nabla \left[\hat{\epsilon}(n) \right] \approx 2\Re \left(\boldsymbol{x}(n) \tilde{e}(n)^* \right). \quad (5.16)$$

The algorithm, then consists of setting

$$\boldsymbol{c}(n+1) = \boldsymbol{c}(n) - 2\beta\Re \left(\boldsymbol{x}(n) \tilde{e}(n)^* \right). \quad (5.17)$$

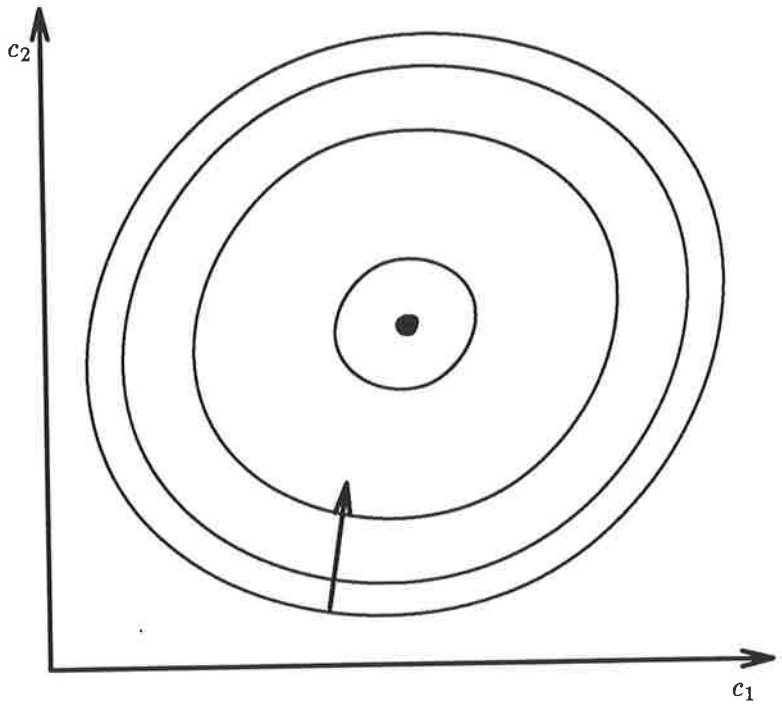


Figure 5.6: An error surface with a small eigenvalue spread. The arrow indicates the direction of steepest descent.

The parameter β effects the speed of adaption as well as the final error. A value of β which is too large can result in an unstable algorithm. The error surface is a paraboloid, and contours of constant error are elliptical (figures 5.6 and 5.7).

The length of the axes of the ellipse are inversely proportional in length to the eigenvalues of the correlation matrix (equation 5.22) and the eigenvectors are in the direction of the axis. Compare figure 5.6 with 5.7. The arrow on these figures indicates the direction of steepest descent. Observe that this is not the quickest route to the optimum point on the error surface, particularly when the eigenvalue spread is large. Heuristically, instability can be envisaged when β is large enough so that $\boldsymbol{c}(n+1)$ is up the other side of the "valley" in the error surface further than $\boldsymbol{c}(n)$. The largest stable value of β is governed by

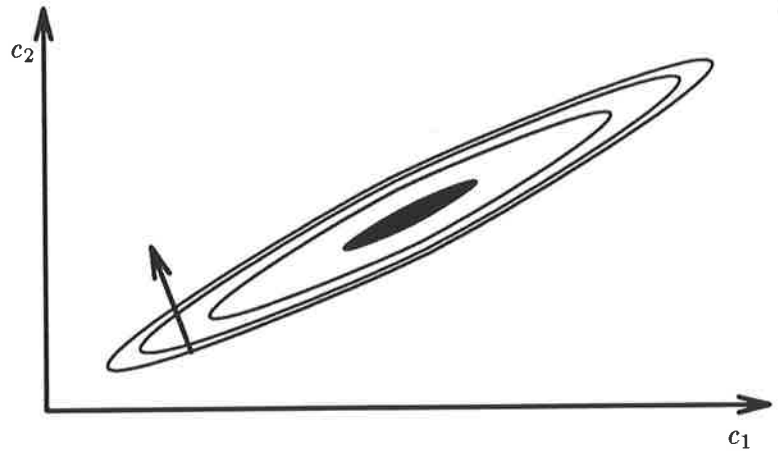


Figure 5.7: An error surface with a large eigenvalue spread. The arrow indicates the direction of steepest descent.

the largest eigenvalue. The convergence rate is also governed by β and it can be shown [39], that for the optimum value of β the error in $\boldsymbol{c}(n)$ is proportional to

$$\left(\frac{\lambda_{\max}/\lambda_{\min}-1}{\lambda_{\max}/\lambda_{\min}+1}\right)^n. \tag{5.18}$$

When the ratio $\lambda_{\max}/\lambda_{\min}$ is large, the quantity inside the brackets of equation 5.18 approaches unity and the rate of convergence is slow. As a consequence, the LMS algorithm is impractical when there is a very large eigenvalue spread.

Note that the error in the coefficients $\boldsymbol{c}(n) - \boldsymbol{c}_{\text{opt}}$ is not proportional to the error defined in equation 5.6. The LMS algorithm minimizes the largest source of error (corresponding to the largest eigenvalue) quite efficiently. In the case of a few large dominant eigenvalues and a number of extremely small eigenvalues, the prediction, estimation or equalization error might become acceptable small quite quickly even though the coefficients are still changing. In the terms of figure 5.7, the descent into the valley is rapid, but travelling along the almost horizontal floor of the valley makes little improvement. In the case where we wish to use a nonlinear filter to correct a small amount of nonlinear distortion

in the presence of a large undistorted (or linearly distorted) signal, then it is necessary to allow the adaption to continue until the contribution due to the small eigenvalues has been corrected. It is the ratio of the largest eigenvalue to the smallest *significant*[3] eigenvalue which is important.

The LMS Algorithm is very efficient in terms of arithmetic operations per step. Referring to equations 5.5, 5.6 and 5.17, there are m multiplications and $m-1$ additions to calculate $y(n)$, 1 subtraction to calculate $e(n)$ and $m+1$ multiplications and m additions to calculate $\boldsymbol{c}(n+1)$. That is a total of $2m+1$ multiplications, $2m-1$ additions and 1 subtraction per step.

5.3.3 Direct Solution Of the Wiener-Hopf Equations

A closed form expression for the optimum weights of a filter can be found. Rewrite equation 5.13 as

$$\frac{\partial \epsilon(n)}{\partial c_i(n)} = \frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \sum_{k=0}^{n} \boldsymbol{x}(k)\tilde{e}(n,k)^* + \left[\frac{\partial \boldsymbol{c}(n)^*}{\partial c_i(n)} \sum_{k=0}^{n} \boldsymbol{x}(k)\tilde{e}(n,k)^*\right]^*. \quad (5.19)$$

The cost $\epsilon(n)$ is minimized by setting $\frac{\partial \epsilon(n)}{\partial c_i(n)} = 0$ which can be achieved if

$$\sum_{k=0}^{n} \boldsymbol{x}(k)\tilde{e}(n,k)^* = 0, \quad (5.20)$$

and substituting equation 5.9 gives

$$\sum_{k=0}^{n} \boldsymbol{x}(k)\boldsymbol{x}(k)^*\boldsymbol{c}(n) = \sum_{k=0}^{n} \boldsymbol{x}(k)d(k)^*,$$

which is of the form

$$\boldsymbol{A}(n)\boldsymbol{c}(n) = \boldsymbol{v}(n), \quad (5.21)$$

where

$$\boldsymbol{A}(n) = \sum_{k=0}^{n} \boldsymbol{x}(k)\boldsymbol{x}(k)^* = \boldsymbol{A}(n-1) + \boldsymbol{x}(n)\boldsymbol{x}(n)^* \quad (5.22)$$

and

[3]The definition of "significant" varies with the application.

$$\boldsymbol{v}(n) = \sum_{k=0}^{n} \boldsymbol{x}(k)d(k)^* = \boldsymbol{v}(n-1) + \boldsymbol{x}(n)d(n)^*. \tag{5.23}$$

Equation 5.21 (together with the definitions 5.22 and 5.23) is known as the Wiener-Hopf solution to the problem of minimizing equation D.17 [77].

Direct solution of equation 5.21 at every step is impractical, however there is typically no need to update $c(n)$ at every n. The matrix $\boldsymbol{A}$ and the matrix $\boldsymbol{v}$ must still be updated every step. Some saving can be made by noting that $\boldsymbol{A}(n)$ is Hermitian. In the linear case, $\boldsymbol{x}(n)\boldsymbol{x}(n)^*$ can be calculated relatively efficiently by defining

$$\Delta\boldsymbol{A}(n) \equiv \boldsymbol{x}(n)\boldsymbol{x}(n)^* \tag{5.24}$$

and noting that the (i,j)th element is

$$\begin{aligned} \Delta a_{i,j}(n) &\equiv x_i(n)x_j(n)^* \\ &= a_{i-1,j-1}(n-1); \quad i > 0, j > 0. \end{aligned} \tag{5.25}$$

Thus $\Delta\boldsymbol{A}(n)$ can be updated with $(m-1)^2$ copy operations and the new leading row and column can be calculated with $2m-1$ multiplications. The matrix $\boldsymbol{A}(n)$ can thus be updated with a total of $(m-1)^2$ copy operations, $2n-1$ multiplications and m^2 additions. For large m the m^2 additions and $(m-1)^2$ copies dominate. If advantage is taken of the Hermitian nature of $\boldsymbol{A}(n)$, and only upper (or lower) triangular sub-matrix of $\boldsymbol{A}(n)$ is stored, then the number of operations can be almost halved. However, there is still $m(m-1)/2$ additions, m multiplications and $(m-1)(m-2)/2$ copies per step. Note that in many computer implementations, copying takes of the order of the same time as addition.

In the nonlinear case, the same idea for efficiently calculating $\boldsymbol{A}(n)$ can be used since many of the nonlinear terms are delayed versions of other nonlinear terms, but the actual operation count is dependent on the actual nonlinearity. However, the number of operations is always more than the linear case, and remains $O(m^2)$ per step or $O(nm^2)$ over n inputs. This excludes the operations

for the solution of equation 5.21 but assuming $\boldsymbol{c}(n)$ is updated infrequently the time to calculate $\boldsymbol{A}(n)$ is the dominant factor.

Direct solution of the Wiener-Hopf provides an exact least squares solution. Solution of a system of linear equations of the form 5.21 is a well understood problem and robust algorithms such as the QR method [25], or singular value decomposition [26] are available as library routines. These routines are well behaved even in the face of relatively poorly conditioned matrices (which corresponds to a large eigenvalue spread).

5.3.4 Recursive Least Squares Algorithms

Gradient algorithms (section 5.3.2) work well when the eigenvalue spread is small. Direct solution of the Wiener-Hopf equations gives a true optimum over the samples input with no convergence time required and when a suitable method is used to solve the equation, accurate results can be achieved even when there is a large eigenvalue spread. However, the execution time (and storage requirements) scales as $O(nm^2)$ and when a large number of taps is required the computation becomes excessive.

A variety of algorithms have emerged which have a range of compromises between convergence rate, final error rate, robustness and computational complexity. Lattice algorithms perform well on many of these counts [40], but cannot be used unmodified to model general nonlinear systems. A class of algorithms known as Recursive Least Squares (RLS) have been invented which recursively calculate the true least squares weights. In m steps an mth order weight vector can be calculated [52]. RLS algorithms exist in both transversal and lattice forms [41] [51] [22] [48]. None of these algorithms can be applied directly in the nonlinear case.

The linear RLS algorithms achieve computational efficiency by making use

of *a priori* information about the nature of the inputs to the linear combiner. In particular, if the filter is defined by $y(n) = \boldsymbol{c}^*\boldsymbol{x}(n)$, then the linear RLS algorithms make use of the fact that the elements of $\boldsymbol{x}(n+1)$ are simply those of $\boldsymbol{x}(n)$ shifted along, with the last element removed and a new first element.

To extend these algorithms to the nonlinear case, we need to similarly take advantage of any *a priori* information on the nature of the elements of $\boldsymbol{x}$. To do that we cast the nonlinear filter in a form of a filter operating on vector (instead of scalar) inputs. For example the terms of a nonlinear filter might be defined as where x_i is the input $x(t - iT)$. At the next sampling instant, all

delay $\longrightarrow$

$\boldsymbol{\xi}_1$	$\boldsymbol{\xi}_2$	$\boldsymbol{\xi}_3$	$\boldsymbol{\xi}_4$	$\boldsymbol{\xi}_5$	$\boldsymbol{\xi}_6$
x_1	x_2	x_3	x_4	x_5	x_6
$x_1x_1x_1$	$x_2x_2x_2$	$x_3x_3x_3$	$x_4x_4x_4$	$x_5x_5x_5$	$x_6x_6x_6$
$x_1x_1x_2$	$x_2x_2x_3$	$x_3x_3x_4$	$x_4x_4x_5$	$x_5x_5x_6$	$x_6x_6x_7$
$x_1x_2x_2$	$x_2x_3x_3$	$x_3x_4x_4$	$x_4x_5x_5$	$x_5x_6x_6$	$x_6x_7x_7$
$x_1x_2x_3$	$x_2x_3x_4$	$x_3x_4x_5$	$x_4x_5x_6$	$x_5x_6x_7$	$x_6x_7x_8$

Figure 5.8: Organization of nonlinear terms into column vectors.

the columns vectors shift right so that $\boldsymbol{\xi}_i(n) = \boldsymbol{\xi}_{i-1}(n-1)$. If the nonlinear filter can be put in the form of figure 5.8, then the algorithms developed in [51] are directly applicable. It is would also be a relatively straight forward exercise to convert other algorithms such as those in [22] [48] to operate on vector quantities. Unfortunately, a nonlinear filter can't *in general* be put in the form of figure 5.8. Note that, in figure 5.8, there are third order terms with delays of up to 8 samples whilst the linear terms have a maximum delay of 6 samples. Ideally, we would like to be able to have different ranges of delay for each order. Typically, a system model has fewer significant high order terms than low order terms and so a more appropriate model would be of the form of figure 5.9. The problem with this form is that the length of the vectors $\boldsymbol{\xi}_i$ varies with i. The author has derived an algorithm for an RLS Transversal Filter and an RLS Lattice Filter operating on the representation in figure 5.9.

delay $\longrightarrow$

ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6
x_1	x_2	x_3	x_4	x_5	x_6
$x_1x_1x_1$	$x_2x_2x_2$	$x_3x_3x_3$			
$x_1x_1x_2$	$x_2x_2x_3$				
$x_1x_2x_2$	$x_2x_3x_3$				
$x_1x_2x_3$					

Figure 5.9: Organization of nonlinear terms into column vectors.

The particular RLS Transversal Filter derived is known as the "Fast Kalman" filter. The method is based on that of Mueller [51], but allowing the length of $\boldsymbol{\xi}_i$ to vary with i complicates the algorithm (and its derivation) considerably. The derivation is lengthy and so has been consigned to appendix D. Here we will discuss only the the formulation of the problem and the over all strategy.

The author implemented a linear version of the Fast Kalman algorithm in software and it was found to be numerically unstable. Errors due to finite precision effects increase exponentially with time. This experience has been confirmed in the literature. Cioffi and Kailath [22] develop a number of improvements to alleviate the problem. The Lattice filter, on the other hand appears to be much more robust. Since the extension to nonlinear filters is likely to be at least as susceptible to these problems, we will concentrate on the lattice filter. The nonlinear Fast Kalman algorithm has been developed in appendix D principally because it is little extra effort since much of the derivation is common to both the RLS Lattice Algorithm and the Fast Kalman algorithm.

Strategy

The vector of inputs to the linear combiner is defined in equation D.1 by

$$\boldsymbol{x}_m(n)^* = \left(\boldsymbol{\xi}_1(n)^* \,\middle|\, \boldsymbol{\xi}_2(n)^* \,\middle|\, \cdots \,\middle|\, \boldsymbol{\xi}_m(n)^* \right). \tag{5.26}$$

For a conventional multiple input linear transversal filter $\boldsymbol{\xi}_1(j)$ is the vector of inputs at the jth time interval and $\boldsymbol{\xi}_i(j) = \boldsymbol{\xi}_{i-1}(j-1)$. In the example 5.9, $\boldsymbol{\xi}_i(1)$ are the column vectors. For this general case where the $\boldsymbol{\xi}_i$ are not all the same length,

$$\boldsymbol{\xi}_i(j) = \boldsymbol{T}_{i-1}\boldsymbol{\xi}_{i-1}(j-1) \quad \text{for } i > 1;$$

and

$$\boldsymbol{T}_i = \boldsymbol{I} \quad \text{for } i \leq 1. \tag{5.27}$$

The matrix $\boldsymbol{T}_{i-1}$ defined in equation 5.27 is such that $\boldsymbol{\xi}_i(j)$ is the same as $\boldsymbol{\xi}_{i-1}(j-1)$ but with some elements removed and possibly some elements rearranged. If equation 5.27 is applied recursively

$$\boldsymbol{\xi}_i(j) = \boldsymbol{T}_{i-1}\boldsymbol{T}_{i-2}\cdots\boldsymbol{T}_1\boldsymbol{\xi}_1(j-i+1). \tag{5.28}$$

We need to also define a row reduced version of $\boldsymbol{x}_m(j)$

$$\begin{aligned}\boldsymbol{x}_m^q(n) &= \hat{\boldsymbol{T}}_{q,m+q-1}\hat{\boldsymbol{T}}_{q-1,m+q-2}\cdots\hat{\boldsymbol{T}}_{1,m}\boldsymbol{x}_m(n) \\ &= \hat{\boldsymbol{T}}_{q,m+q-1}\boldsymbol{x}_m^{q-1}(n)\end{aligned} \tag{5.29}$$

and

$$\boldsymbol{x}_m^0(n) = \boldsymbol{x}_m(n), \tag{5.30}$$

where

$$\hat{\boldsymbol{T}}_{i,j} = \left(\begin{array}{c|c|c|c} \boldsymbol{T}_i & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{T}_{i+1} & \cdots & \mathbf{0} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{T}_j \end{array}\right). \tag{5.31}$$

It follows from equations 5.31, 5.29 and 5.28 that

$$\boldsymbol{x}_{m-q}^q(n) = \boldsymbol{x}_{q+1,m}(n+q) \tag{5.32}$$

$$= \left(\ \boldsymbol{\xi}_{q+1}(n+q)^* \ \middle|\ \boldsymbol{\xi}_{q+2}(n+q)^* \ \middle|\ \cdots \ \middle|\ \boldsymbol{\xi}_m(n+q)^*\ \right). \tag{5.33}$$

That is to say $\boldsymbol{x}_{m-q}^q(n)$ denotes a vector made up of the last $m-q$ subvectors of $\boldsymbol{x}_m(n+q)$. This is illustrated in figure 5.10.

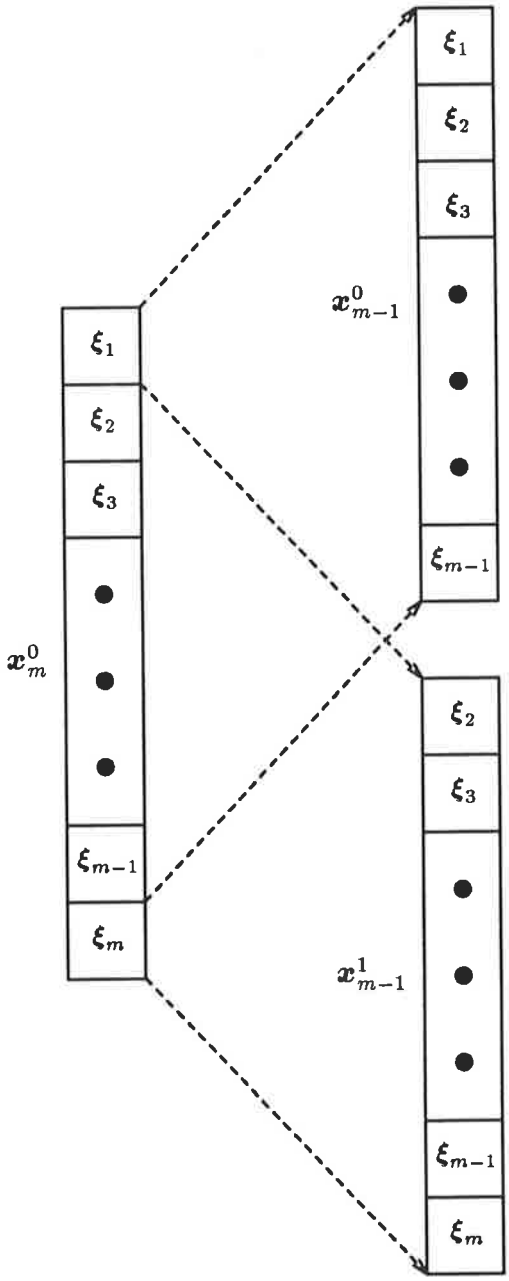


Figure 5.10: Reduced order filter "taps".

The method basically involves finding time and order recursive relationships for the various filter parameters. The time recursive relationships arise from the recursive forms for equations 5.22 and 5.23. Two order recursive forms are needed. One is derived by reducing the order from the right and the other is derived by reducing the order from the left (figure 5.10). Thus we describe the parameters of the filter based on $\boldsymbol{x}_m^0(n)$ in terms of the parameters of the filters based on $\boldsymbol{x}_{m-1}^0(n)$ and $\boldsymbol{x}_{m-1}^1(n-1)$. The main reason for the increased complexity of the derivation of the nonlinear version of these algorithms, is that in the linear case, $\boldsymbol{x}_{m-1}^1(n-1) = \boldsymbol{x}_{m-1}^0(n-1)$. In the linear case we can describe a mth order filter in terms of parameters of the $(m-1)$th order filter and the delayed parameters of $(m-1)$th order filter. In the nonlinear case we need to maintain filters with values of m and q of $(1, m_1 - 1), (2, m_1 - 1), (3, m_1 - 2) \ldots (m_1, 0)$ in order for the final filter to have $q = 0$ and $m = m_1$. The process is illustrated in figure D.4. In section D.10 we demonstrate that much of this computational burden can be removed if some of the $\hat{\boldsymbol{T}}_{q,m+q-1} = \boldsymbol{I}$. The more efficient method is illustrated in figure D.5. For the application of identifying a system with only small nonlinear distortion in the presence of significant linear distortion, it will often be the case that $\hat{\boldsymbol{T}}_{q,m+q-1} = \boldsymbol{I}$ for large m. For m in this region, the computational burden increases as $O(nm)$.

Verification

With a derivation as lengthy as that in appendix D, it is difficult to be confident that their are no errors. To gain confidence in the derivation a verification process was performed. This was done using a "programmable calculator" program[4]. This program has a significant number of built in matrix operations.

Programs (*calc* macros) where written to evaluate the filter parameters for a range of m and n using straight forward methods. For both the forward

[4]The program is called *calc* by David Gillespie and is written in GNU emacs lisp.

and backward predictor case, matrix $\boldsymbol{A}$ is evaluated using equation D.24, $\boldsymbol{v}^s$ is evaluated using equation D.25, $\boldsymbol{d}^s$ is evaluated using equation D.12, $\boldsymbol{c}^s$ is calculated by solving equation D.23 using the built in matrix inversion function, the error $\boldsymbol{e}^s$ is calculated using equation D.13, $\boldsymbol{\epsilon}^s$ was calculated using equations D.34 and D.35, the Kalman gain $\boldsymbol{g}$ was calculated from equation D.38, and $\tilde{\boldsymbol{e}}^s$ was calculated from equation D.15, and $\boldsymbol{k}^f$ was calculated using equation D.148. All these parameters were evaluated for $1 < n < 10$ and $m = 2, 3$ and $q = 0, 1$. These parameters were then substituted both sides of equations D.167, D.168, D.191, D.192, D.207 and D.216 and checked for equality.

Essentially, we have used the method of direct solution of the Wiener-Hopf equations (section 5.3.3) plus some auxiliary equations to verify the order recursive relationships of the RLS algorithms.

5.4 Orthogonalization

The recursive least squares algorithms perform a partial orthogonalization of the x vector (see section D.3) and the high performance is attributable to this. Also, the better algorithms for direct solution of the Wiener-Hopf equation perform orthogonalization.

It is possible to perform this orthogonalization explicitly by replacing the nonlinear box in figure 5.2 with one which has orthogonal outputs. Then the coefficients could be evaluated by correlation or by the gradient algorithm (the maximum convergence rate would be much increased). The question arises of how to determine the contents of the nonlinear box which would have the orthogonal outputs required. One approach is based on having *a priori* knowledge of the probability density function of the input. This is the approach used in the Wiener model[5] discussed in section 3.5 and has the disadvantage that the

[5]The orthogonalization in the Wiener model is actually performed in two stages.

probability density function of the input signal must be known.

Another method of orthogonalization is to calculate an orthogonalizing matrix from the eigenvectors of the correlation matrix. However, if we have the correlation matrix we might as well use it to solve the Wiener-Hopf equation directly. In either case, explicit orthogonalization is an $O(m^2)$ operation and so provides no advantage in number of computations over direct solution of the Wiener-Hopf equation assuming that the weights do not have to updated for every input sample.

There are potential gains to be made by performing approximate orthogonalization at a lower computational cost. For example, if a large signal can be cancelled, leaving only a smaller signal which must be estimated, then the eigenvalue spread is reduced, but not eliminated. This can be considered an approximate orthogonalization. The applicability of this technique depends on the specific problem.

5.5 Multi-input Nonlinear Filters

The extension of the nonlinear filter to two or more inputs is straight forward and is illustrated in figure 5.11.

The adaption algorithms similarly require very little modification. The gradient algorithms (section 5.3.2) and the direct solution method (section 5.3.3) need no modification at all since they require no knowledge of how the outputs of the nonlinear section were generated. The RLS methods (section 5.3.4) can also be used provided the outputs are suitable arranged into the vectors $\boldsymbol{\xi}_m$.

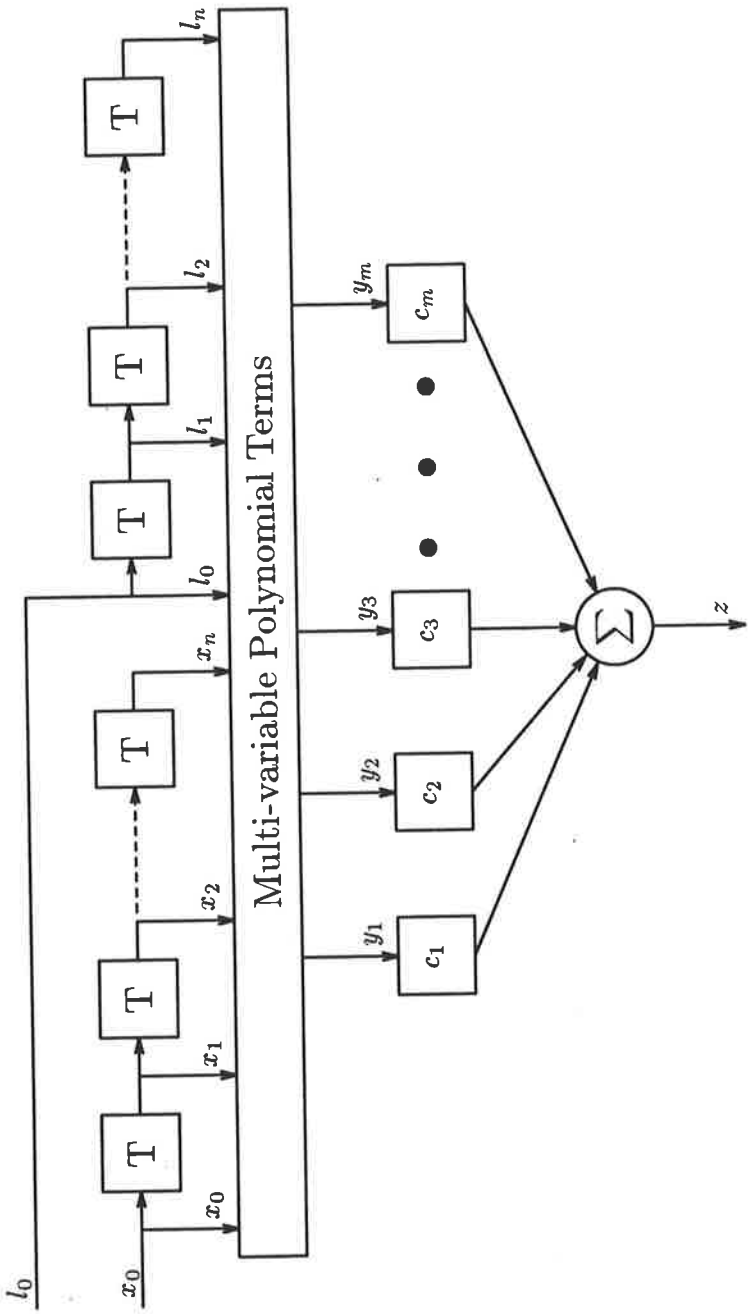


Figure 5.11: A two input nonlinear filter based on the Volterra series.

5.6 Implementation Considerations

5.6.1 Initialization

Regardless of the algorithm used, the question arises what to initialize the delay line of figure 5.2 to at time $t = 0$. If we have no information about how the input behaves prior to this time, then initializing the delay line to zero seems as good as any, however it is certainly not the only strategy possible. In the experiments which are described in chapter 6, the signals are all periodic. In these circumstances it is possible to do much better than initialization with zero.

In the software implementation of the direct solution method (section 5.3.3) written for the experiments in chapter 6 an option was included to allow the delay line to be pre-loaded. If pre-loading was specified, the input data was cycled through once to initialize the delay line and a second time to calculate $\boldsymbol{A}(n)$ and $\boldsymbol{v}(n)$ according to equations 5.22 and 5.23.

5.6.2 Software

A suite of programs were developed to implement and test nonlinear adaptive filters. Data files were organized so that the first four bytes contained a magic number indicating whether the rest of the file contained integers, floating point numbers, complex floating point numbers etc. The absence of a valid magic number indicated an ascii file. This feature proved invaluable (once the input and output library was written) since short tests could be conducted with data created with a text editor but a more efficient binary representation was used when large amounts of data had to be processed. Each program had an option for specifying binary or ascii output. In the case of binary output, some programs output integers and others output floating point numbers and others

output complex floating point numbers. In any case, the appropriate magic number would be output first so that other programs in the suite would be able to properly interpret the data. In many cases, the output of one program would be fed into another program via a UNIX pipe avoiding the need for a temporary disk file.

The suite consisted of programs to generate random test data, perform linear combinations of two streams, perform arbitrary nonlinear filtering operations, calculate the mean squared difference between two streams and calculate optimum filter weights. The nonlinear filter and the weight calculation programs were able to take an arbitrary number of inputs (up to some maximum dependent on a system imposed limit on the number of open files).

The Nonlinear Filter Program

The weights of the nonlinear filter are specified by a parameter file which contains entries of the form

$$\{n_1 tuple,\ n_2 tuple,\ \ldots, n_i tuple\} \quad value$$

where $n_i tuple$ is of the form

$$\{d_{i,1}, d_{i,2}, \ldots, d_{i,n_i}\}$$

where $d_{i,j}$ is the ith input delayed by j and value is either a real number or a complex number written as {real,imag}. for example, the entry

$$\{\{4,4\},\{4\}\} \quad \{0, 0.7\}$$

indicates that $\text{input}_{1,4} \times \text{input}_{1,4} \times \text{input}_{2,4}$ has the complex weight $0 + 0.7j$.

Internally, the history of each input is kept in an array. At each sample instant, the elements of the input arrays are all shifted one position and a new element is read into each one. Each weight has associated with it a list of

pointers to elements in the input arrays. Thus the nonlinear product terms are simply created by de-referencing each of the pointers in the list and forming a running product. This does involve some redundant calculations since many product terms will have common sub-expressions. These could probably be eliminated by constructing a tree of all the multiplications to be done, but care needs to be taken to ensure that the more complex code doesn't take longer than the operations saved. This is machine dependent.

The Weight Calculation Program (Direct Solution)

The weight calculation program calculates the nonlinear products in a similar manner to the Nonlinear Filter program. However, the specification of the product terms is somewhat different. The product terms are specified in a parameter file which has elements of the form

$$tuple_1,\ tuple_2,\ \ldots,\ tuple_n \quad canon$$

where $tuple_i$ is of the form

$$\{order, max.\ delay\}.$$

The $tuple_i$ represents the products of *order* using delays from zero up to *max. delay* for the ith input. The program automatically works out all the possible products across all the inputs specified. The *canon* field is used to optionally indicate that additional terms be included which are the product of the inputs indicated by the rest of the entry with all terms delayed by the same amount up to *canon*.

For example, an entry in the parameters file of {2,3},{1,1} results in the following terms (which are in the form discussed in section 5.6.2 above):

{{0,0},{0}}	{{3,1},{1}}	{{3,2},{1}}	{{3,3},{0}}
{{1,1},{1}}	{{3,0},{0}}	{{3,1},{0}}	{{0,0},{1}}
{{1,0},{0}}	{{1,1},{0}}	{{2,2},{0}}	{{1,0},{1}}
{{2,1},{1}}	{{2,2},{1}}	{{3,3},{1}}	{{2,0},{1}}
{{2,0},{0}}	{{2,1},{0}}	{{3,2},{0}}	{{3,0},{1}}.

The number of terms clearly increases very rapidly with *order*, *max. delay* and number of inputs. The *canon* field was introduced in the expectation that a delayed terms of the canonical form might be more significant. As an example of a parameter entry using the *canon* field field, an entry in the parameters file of

$$\{2,3\},\{1,1\} \quad 9$$

results in the following terms:

$\{\{0,0\},\{0\}\}$	$\{\{7,7\},\{7\}\}$	$\{\{3,0\},\{0\}\}$	$\{\{3,3\},\{1\}\}$
$\{\{1,1\},\{1\}\}$	$\{\{8,8\},\{8\}\}$	$\{\{1,1\},\{0\}\}$	$\{\{3,2\},\{0\}\}$
$\{\{2,2\},\{2\}\}$	$\{\{9,9\},\{9\}\}$	$\{\{2,2\},\{1\}\}$	$\{\{3,3\},\{0\}\}$
$\{\{3,3\},\{3\}\}$	$\{\{1,0\},\{0\}\}$	$\{\{2,1\},\{0\}\}$	$\{\{0,0\},\{1\}\}$
$\{\{4,4\},\{4\}\}$	$\{\{2,1\},\{1\}\}$	$\{\{3,2\},\{1\}\}$	$\{\{1,0\},\{1\}\}$
$\{\{5,5\},\{5\}\}$	$\{\{2,0\},\{0\}\}$	$\{\{3,1\},\{0\}\}$	$\{\{2,0\},\{1\}\}$
$\{\{6,6\},\{6\}\}$	$\{\{3,1\},\{1\}\}$	$\{\{2,2\},\{0\}\}$	$\{\{3,0\},\{1\}\}$.

Once the nonlinear products have been calculated, the calculation of the matrix $\boldsymbol{A}(n)$ and the vector $\boldsymbol{v}(n)$ is performed using the copying optimization discussed in section 5.3.3. With the aid of Linpack library routines, the solution of the Wiener-Hopf equations is straight forward. Both QR and SVD algorithms were tried. See sections 6.4.3 and 6.5.2 for discussion on the suitability of the SVD algorithm for the particular case of the HFR experiments.

The Weight Calculation Program (Recursive Least Squares)

A functional implementation of the RLS algorithm has not been completed, however, as far as external inputs are concerned, it is designed to appear the same as the program implementing the direct solution method. One exception is the "pre-loading" feature which was easily implemented for the direct solution method (section 5.6.1). It is not straight forward to initialize the RLS method in the same way.

It is worth noting that the elements of the sub-vectors $\boldsymbol{\xi}_1$ can always be

ordered in such a way that the matrix $\grave{\boldsymbol{T}}_q$ is of the form

$$\grave{\boldsymbol{T}}_q = \left(\boldsymbol{I} \mid \boldsymbol{0} \right). \tag{5.34}$$

If $\grave{\boldsymbol{T}}_q$ is of the form of equation 5.34, expressions of the form

$$\grave{\boldsymbol{T}}_q^* \left(\grave{\boldsymbol{T}}_q \boldsymbol{\epsilon}_{m-1}^{f,q+1}(n-1) \grave{\boldsymbol{T}}_q^* \right)^{-1} \grave{\boldsymbol{T}}_q, \tag{5.35}$$

which appear in a number of equations in the RLS Lattice algorithm (section D.10), is equivalent to inverting a $p_{q+1} \times p_{q+1}$ sub-matrix of $\boldsymbol{\epsilon}_{m-1}^{f,q+1}$ and zeroing the elements outside this sub-matrix. This is clearly a much cheaper operation than a naive evaluation of expression 5.35. Indeed, it turns out that the $\boldsymbol{T}$ and $\grave{\boldsymbol{T}}$ don't need to ever be stored. All that needs to be stored are the values of p_{i+1} which are the number of rows of the $\boldsymbol{T}$ matrices.

5.7 Summary

In this chapter, we have discussed the theory and implementation of adaptive nonlinear digital filters. We have seen how the discrete time Volterra series can form the basis for a nonlinear digital filter. Nonlinear filters can be made to adapt to a required characteristic. The nonlinear adaptive filter can be used in the roles of estimators, predictors and equalizers in the same way as linear adaptive filters.

The LMS gradient algorithm and the method of directly solving the Wiener-Hopf equations used in linear adaptive filters are directly applicable to the nonlinear case. Other more efficient linear adaption algorithms such as the Recursive Least Squares algorithms are not directly applicable to the nonlinear case. The Recursive Least Squares algorithm has been extended to the nonlinear case in section 5.3.4. The LMS gradient algorithm and the RLS algorithm require $O(nm)$ operations, whereas the Direct Solution of the Wiener-Hopf equations requires $O(nm^2)$ operations assuming that the weights are updated infrequently

so that number of operations is dominated by the calculation of the correlation matrix $\boldsymbol{A}$. The RLS algorithms and direct solution of the Wiener-Hopf equations have zero convergence time.

RLS algorithms achieve their efficiency by partial orthogonalization of the nonlinear product terms. It is possible to explicitly orthogonalize the outputs of the filter's nonlinear combiner which has the potential to improve convergence rates. However, explicit orthogonalization has no computational advantage over direct solution of the Wiener-Hopf equations assuming that the weights do not have to updated for every input sample.

The single input nonlinear digital filter is simply extendible to two or more inputs. The adaption algorithms can also be extended to the multi-input case.

The performance of adaption algorithms in the presence of periodic signals can be improved by appropriate initialization of the filter. The implementation of a suite of programs for efficiently implementing and testing nonlinear adaptive filters has been discussed.

Chapter 6

HFR Receiver Experiments

6.1 Introduction

Workers in High Frequency Over the Horizon Radar (HF OTHR) have been interested in high linearity RF equipment for some time [2] [27] [17]. The performance of all the RF equipment is, of course, important, but the receivers are possibly the most critical in terms of nonlinear performance, since they are exposed to large signals (interference) over which the radar designer has no control. Anderson [2] notes the Volterra Series is a general way of representing the receiver output, but is forced to simplify the model considerably in his simulation. In particular, the system is assumed to consist of distortionless mixers, ideal filters and amplifiers which are memoryless nonlinearities. The nonlinear parameters were estimated by comparing the intermodulation distortion of the model and of a real receiver with a two tone test. As has been pointed out (section 3.7), a two tone test is not capable of identifying a general nonlinearity. Earl [27] used an even simpler model, the power series, in his study of cross-modulation effects.

Accurate models of receiver nonlinearities would allow better simulation of

the actual radar performance attainable in a particular HF environment. If such simulations reveal that the best receivers have distortion levels which limit radar performance, then it is of interest to know to what extent receiver nonlinearities can be improved by post processing. Modelling and correction correspond to estimation and equalization discussed in section 5.3.1.

The aim of the experiments of this chapter are to see how well, in practice, adaptive nonlinear digital filters can be used to estimate and equalize distortion in an HFR receiver. We are also able to verify experimentally the theoretical properties of multiple input systems derived in chapter 4.

Similar experiments which use adaptive nonlinear digital filters of the form described in chapter 5 to model or correct distortion in real[1] systems have been described in the literature [28] [47] [9] [1].

The HFR receiver has some properties which mean the design of the experiment must differ significantly to other experiments described in the literature. Specifically:

1. The receiver must be modelled as a two input device.
2. The receiver has slight, but important, nonlinear distortion in the presence of large linear distortion.
3. The receiver's inputs and outputs are at different frequencies.
4. The band width of the signals applied to the receiver's inputs should be very much greater than the band width of the receiver output.

These features, and the reason they affect the experiment are discussed in sections 6.2 and 6.3.

[1]In some cases the authors do not make it clear whether they are dealing with a real system or a simulation. In the case of simulations, there are simulations of real systems and simulations of hypothetical systems so the classification becomes fuzzy.

These experiments presented some non-trivial problems in the generation of test signals, selection of sampling rates and design and implementation of the analysis algorithms.

6.2 The HFR Receiver and Its Operating Conditions

The Jindalee[2] stage B receiver is a triple conversion heterodyne design manufactured by SRI International. The receiver used in the experiments of this chapter was not an actual stage B receiver, but an experimental version which had several modules replaced. Most notably, the output band width of the receiver had been increased. A block diagram of the receiver is shown in figure 6.1.

In normal operation, the first local oscillator is a continuous FM signal where the modulating signal is a linear sweep continuously repeated (a saw tooth). This results in a sequence of "chirps". We will refer to this signal as an FMCW signal. The other local oscillators are sinusoids. The receiver input consists of noise, interference, ground clutter, sea clutter and (moving) targets all transformed by reflection from the ionosphere which might include significant multipath effects. The signal from ground clutter, sea clutter and target signals must be of the form of an infinite sum of delayed (and possibly doppler shifted) versions of the transmitted signal which is also an FMCW signal. A discussion of the theory of operation of radars is outside the scope of this thesis, but see [46] [3]. The largest source of interference under normal circumstances is AM HF broadcast stations. Of course, in the case of deliberate interference (jamming) the nature of the interference can't be predicted and is likely to be of the most troublesome nature.

[2]The Australian OTHR facility [2].

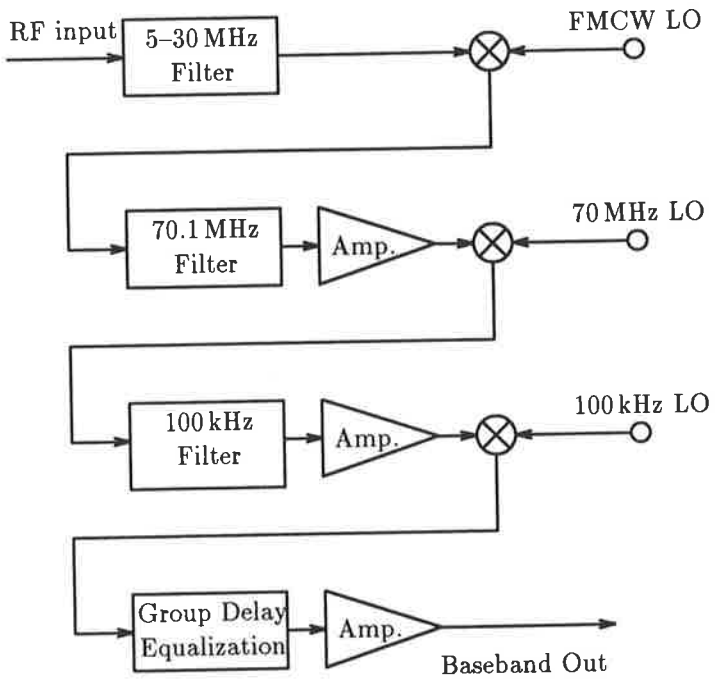


Figure 6.1: Experimental radar receiver block diagram.

6.3 Design

The basic strategy of the experiments can be summarized is as follows: Known signals are applied to all the receiver inputs. The output is measured. In the estimation case, we find the multi-input nonlinear digital filter which is best able to estimate the receiver output. In the corrector case we calculate the expected output from an ideal receiver subject to the same inputs and find the nonlinear filter which is best able to transform the real receiver output to the ideal receiver output.

This is basically the same as the traditional estimation and equalization processes of section 5.3.1 with some differences. The most notable difference is that the estimator must be a multi-input estimator (figure 6.2). As was discussed in section 4.3.2, the sinusoidal local oscillators can be ignored from the point of view of the distortion model. Since only the first local oscillator is an

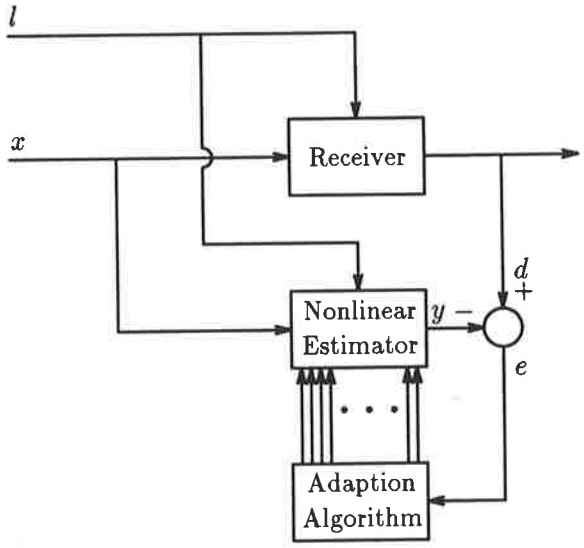


Figure 6.2: Estimation of a two input receiver.

FMCW signal, the receiver can be treated as a general two input nonlinearity. If the FMCW signal sweeps sufficiently slowly it may be possible to model the receiver as a single input single output nonlinear device followed by an *ideal* receiver (figure 6.3). The best way to tell whether the simpler model is adequate is to test both cases and compare the results.

In the case of the corrector it is less obvious that a two input filter could be employed since the receiver is a single output device. However in the case of a receiver, the local oscillator inputs are available as inputs to the corrector and if the local oscillator input has a significant effect on the receiver distortion, then it is at least possible that a two input corrector would provide better performance than a single input corrector. See figure 6.4.

The design of the experiments was constrained by the available equipment (section 6.3.1).

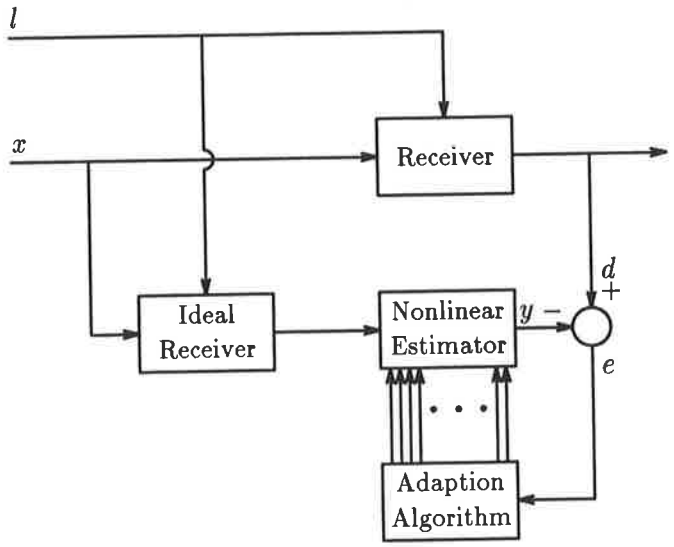


Figure 6.3: Approximate estimation of a two input receiver using a single input estimator.

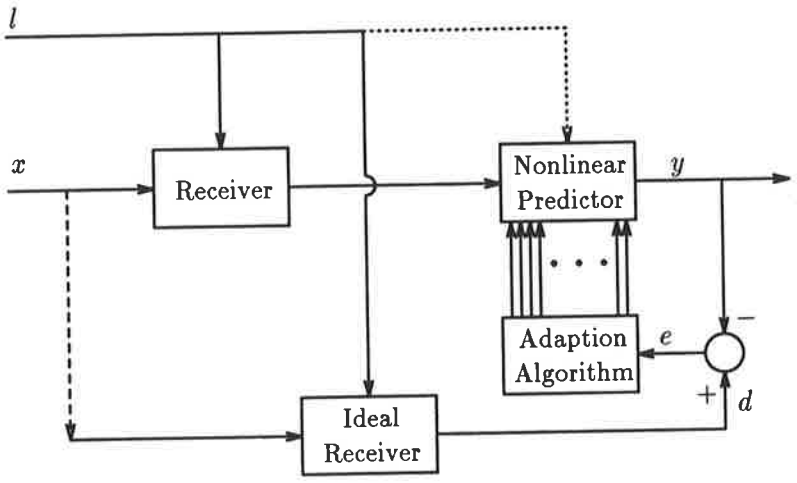


Figure 6.4: Equalization of a two input receiver. The extension to a two input corrector is by the dotted line.

6.3.1 Equipment

The general strategy was to generate known test signals in a general purpose computer off-line and down load the data into a real-time D/A facility designed for generating arbitrary wave forms. The outputs from the D/A converters would be applied to the receiver and the receiver output would be sampled with an A/D converter. The results would be transferred to a general purpose computer and analyzed off-line. In practice the equipment configuration is somewhat more complicated.

The D/A facility has a built in memory which it simply transfers sequentially to the 12 bit D/A converter. The sample rate is programmable and when the end of the block of memory in use is reached, the sequence repeats indefinitely. The memory of the system and the maximum sampling rate are bounded and so the period of the input signal(s) is bounded. There is a trade-off between these parameters (sampling rate, number of samples and signal period), but a D/A sampling rate of approximately 5 MHz was chosen. See sections 6.3.2 and 6.3.4 for a more detailed discussion of the selection of sampling rates.

The D/A sampling rate f_s is not sufficient to allow direct generation of the receiver input signal which is approximately 15 MHz and the local oscillator signal which is approximately 85 MHz. These number correspond approximately to the middle of the receivers frequency range. The output from the D/A converter, must, therefore go through a frequency translator before it is applied to the receiver. The frequency translators must contain filters to eliminate aliases of the desired signal. The least demanding specifications for the translator filters occur when the aliases are equally spaced which occurs when the spectrum of the data in the D/A memories is narrow compared with $f_s/2$ and centred on $f_s/4$. Consequently, the spectrum of the D/A output is centred at $f_s/4$.

The equipment configuration is illustrated in figure 6.5. All signal sources are derived from a single stable frequency reference.

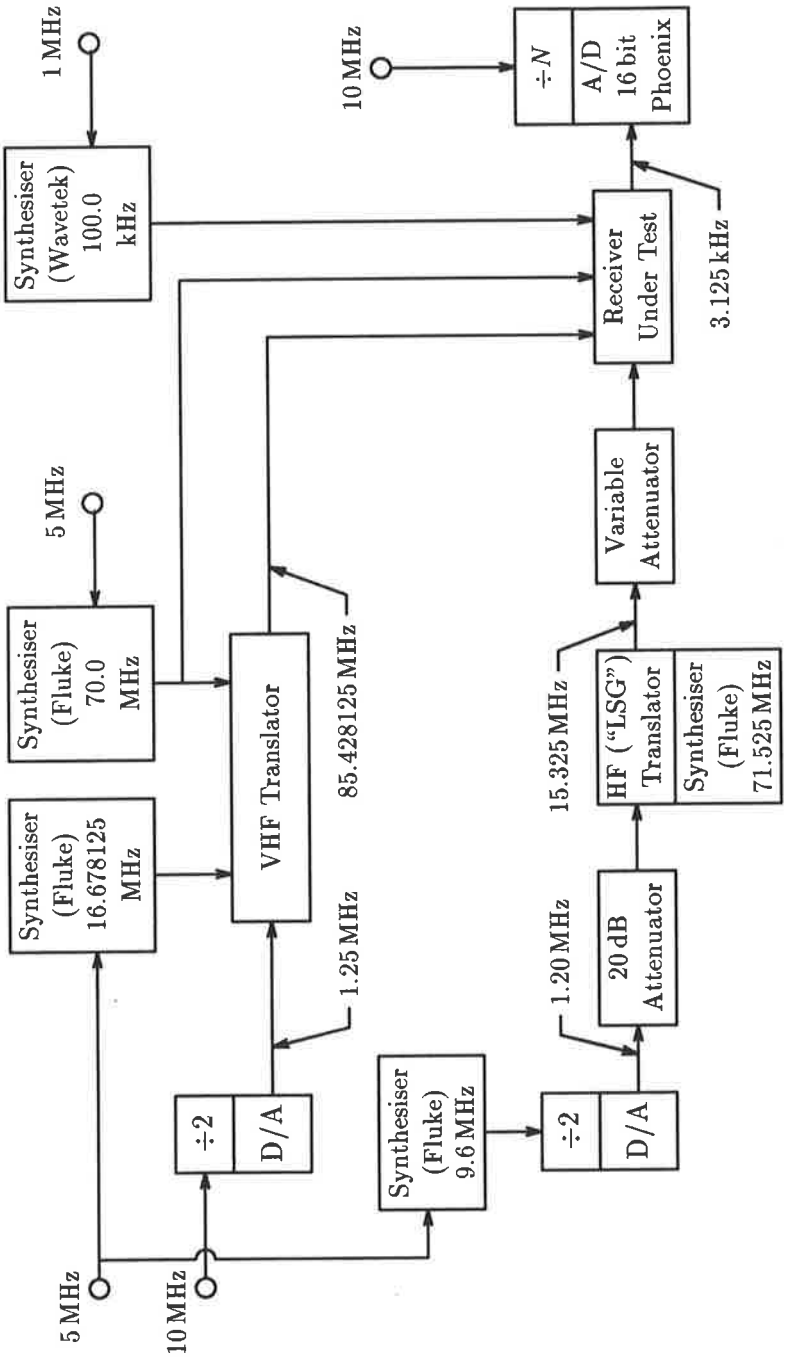


Figure 6.5: Equipment configuration for HFR receiver experiments. The design of the VHF translator is shown in figure 6.6. Signal frequencies shown are the nominal centre frequencies.

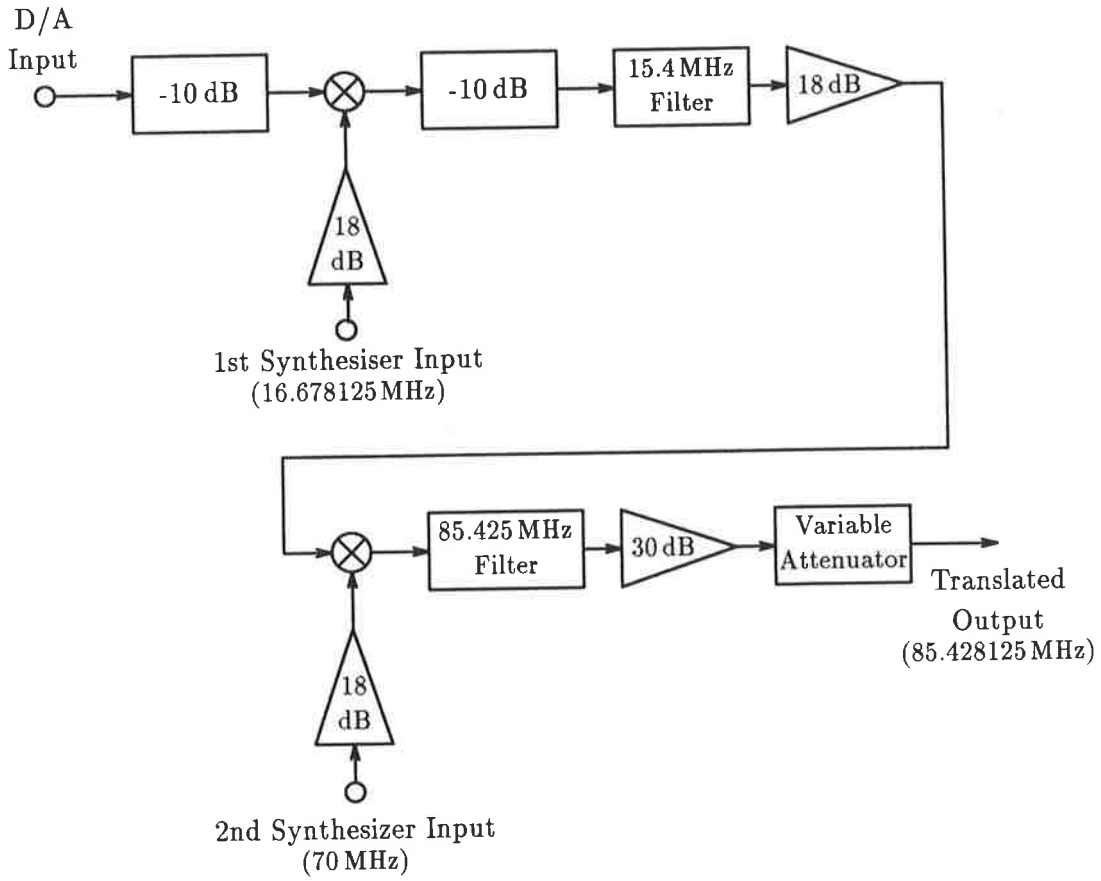


Figure 6.6: VHF frequency translator.

All the equipment was provided by the Australian Defence Science and Technology Organization (DSTO) who had most of the equipment to hand as part of a related project. The VHF translator was built to the authors specification. The rest of the equipment was either commercially available equipment or pre-existing, DSTO built equipment. Some minor modifications were required to suit this project. Specifically, the down-loading facility was unable to cope with the amount of data required and needed enhancing. Also, the D/A converters were originally completely free running. This project required that the D/A converters be synchronized with each other and with the A/D converters.

6.3.2 Test Signals

The choice of test signals is not entirely straight forward. More traditional methods of measuring distortion employing sinusoids have the advantage that it is relatively easy to isolate the nonlinear response because it is at a different frequency than the linear response. However, section 3.7 points out that a broad band signal is required to completely identify a general nonlinearity. Also, the optimum estimation of a system depends on the probability density function of the input signal (section 3.5), so it is desirable for the test signals to have a similar probability density function to the signals applied to the receiver in normal operation. The normal receiver input can be considered as a sum of the backscatter from an infinite sum of infinitesimal backscatterers. However, these components are not independent processes because the illuminating signal is strongly correlated and the central limit theorem is not strictly applicable. If the probability density function of the test signal is not exactly the same as the normal receiver input, an estimation will still be achieved, but it will be sub-optimal.

Band limited quasi-Gaussian pseudo-random noise was selected for these receiver experiments. The author does not claim that this signal is in any sense ideal, but its properties are a reasonable approximation to the properties mentioned in the last paragraph. The quasi-Gaussian distribution was achieved by summing 10 uniformly distributed random variables. The local oscillator signal was chosen to be a band limited FMCW ("chirp") signal with sweep rates and repetition rates typical of normal operation.

Both the FMCW and pseudo-random noise signals were band limited by Fourier transforming, zeroing frequencies outside the desired band and inverse transforming[3]. Doing this digitally allows the specification for the analogue

[3]Since the signal is generated by cycling through the set of samples, it is truly periodic and hence this method of bandpass filtering is exact.

filters in the frequency translators (figure 6.5) to be relaxed.

6.3.3 Chirp Generation by Phase Accumulation

Experience has shown that generating sinusoids on a general purpose computer is best done by incrementing the phase modulo 2π. If the obvious formula is used $x(t) = \sin(\omega t)$ for $t = 0, t_1, t_2, \ldots t_n$, then there is considerable phase noise if t_n is large. That is because, even with double precision arithmetic, the error in ωt for large t is a significant proportion of 2π. Consequently, the best algorithm is:

1. Initialize ϕ and $\Delta\phi$.
2. $x(t) = \cos(\phi)$.
3. Set $\phi \leftarrow \phi + \Delta\phi$.
4. If $\phi > \pi$ set $\phi \leftarrow \phi - 2\pi$.
5. Repeat from step 2.

This algorithm reduces phase noise by keeping $|\phi| \leq \pi$.

A similar algorithm can be used for a linear frequency sweep. First note that the instantaneous frequency can be defined by

$$f = \frac{1}{2\pi}\frac{d\phi}{dt}, \tag{6.1}$$

and a linear sweep can be defined

$$f(t) = f_0 + (f_1 - f_0)\frac{t}{t_{\text{wrf}}} \tag{6.2}$$

$$= f_0 + (f_1 - f_0) f_{\text{wrf}} t. \tag{6.3}$$

Sum of Sinusoid Signals				
No.	**File Name**	**Frequencies**	**Amplitudes**	**Comment**
1	single	32768	2047	A single sinusoid
2	doublet	32767 32769	1000 1000	Two closely spaced sinusoids
3	doublet2	32767 32769	1800 180	Two closely spaced spaced sinusoids
4	triplet	32758 32768 32778	150 450 1359	Three sinusoids

FMCW ("Chirp") Signals							
No.	**File Name**	**Sweep Start Freq.**	**Sweep Stop Freq.**	**Sweep Rep. Freq.**	**Filter Start Freq.**	**Filter Stop Freq.**	**Amp.**
5	lo-data1	32256	33279	2	31774	33791	2047
6	lo-data2	32256	33279	2	31744	33791	2047
7	lo-data2l	32256	33279	2	31744	33791	1024

Noise Signals					
No.	**File Name**	**Random No. Seed**	**Filter Start Freq.**	**Filter Stop Freq.**	**Amplitude**
8	input1	123	31774	33791	2047
9	input2	234	31774	33791	2047
10	input3	123	31744	33791	2047
11	input3l	123	31744	33791	1024
12	input4	234	31744	33791	2047

Table 6.1: Test signal parameters. All frequencies are in units of $f_s/131072$ Hz where f_s is the sample frequency (see table 6.2). All signals are bandpass filtered. The lower limits of the passband are tabulated. Note the anomalous filter start frequency of signals 5, 8 and 9 which was due to a typographical error in the script which generated the signals.

Substitute equation 6.3 in to equation 6.1 and integrate to get

$$\phi(t) = 2\pi \left(f_0 + (f_1 - f_0) f_{\text{wrf}} \frac{t^2}{2} + \phi_0 \right), \tag{6.4}$$

and therefore

$$\begin{aligned} \Delta\phi(t) &\equiv \phi(t + \Delta t) - \phi(t) \\ &= 2\pi \left(f_0 \Delta t + (f_1 - f_0) \frac{f_{\text{wrf}}}{2} \left[(t + \Delta t)^2 - t^2 \right] \right) \\ &= 2\pi \left(f_0 \Delta t + (f_1 - f_0) f_{\text{wrf}} \Delta t \, t \right) \end{aligned} \tag{6.5}$$

and

$$\Delta\phi(0) = 2\pi f_0 \Delta t. \tag{6.6}$$

Now we can define

$$\begin{aligned} \Delta\Delta\phi(t) &\equiv \Delta\phi(t + \Delta t) - \Delta\phi(t) \\ &= 2\pi (f_1 - f_0) f_{\text{wrf}} \Delta t \Delta t. \end{aligned} \tag{6.7}$$

When the wave form is generated in a general purpose computer, the frequencies are somewhat arbitrary since it depends on the eventual output sample rate. The expressions for $\Delta\phi$ (equation 6.5) and $\Delta\Delta\phi(t)$ (equation 6.7) can be simplified by specifying the frequency in units of $1/(n\Delta t)$Hz where n is the total number of samples. The frequency units are then the same size as an FFT "bin". We have

$$f_0 = \frac{i_0}{n\Delta t}; \qquad f_1 = \frac{i_1}{n\Delta t}; \qquad f_{\text{wrf}} = \frac{i_{\text{wrf}}}{n\Delta t}. \tag{6.8}$$

Now substitute equations 6.8 into equations 6.6 and 6.7 to get

$$\Delta\phi(0) = \frac{2\pi i_0}{n}, \tag{6.9}$$

and

$$\Delta\Delta\phi(t) = \frac{2\pi (i_1 - i_0) i_{\text{wrf}}}{n^2}. \tag{6.10}$$

The algorithm is then:

1. Initialize ϕ, $\Delta\phi$ and $\Delta\Delta\phi$.
2. $x(n) = \cos(\phi)$.
3. Set $\phi \leftarrow \phi + \Delta\phi$.
4. If $\phi > \pi$ set $\phi \leftarrow \phi - 2\pi$.
5. Set $\Delta\phi \leftarrow \Delta\phi + \Delta\Delta\phi$.
6. Repeat from step 2.

It is not normally necessary to ensure that $|\Delta\phi| \leq \pi$ since that would correspond to a maximum frequency above the Nyquist sampling rate.

6.3.4 Sampling Rates

The sample rates and frequencies have been chosen so the data contains an integer number of cycles. This allows manipulations to be done by Fast Fourier Transforms (FFT's) and Inverse Fast Fourier Transforms (IFFT's) without windowing (or with a rectangular window).

The problem of aliases of the desired signal has already been mentioned in section 6.3.1. However, with careful choice of sampling rates the requirements of the translator filters can be relaxed further. The presence of aliases results in energy centred at frequencies

$$f_x = nf_{s_x} \pm f_1, \tag{6.11}$$

$$f_l = mf_{s_l} \pm f_2, \tag{6.12}$$

for the input (x) and local oscillator (l) signals. If these two signals are translated and mixed (inside the receiver) the resultant signal has components centred at frequencies

$$f = nf_{s_x} \pm mf_{s_l} \pm f_1 \pm f_2 + k, \tag{6.13}$$

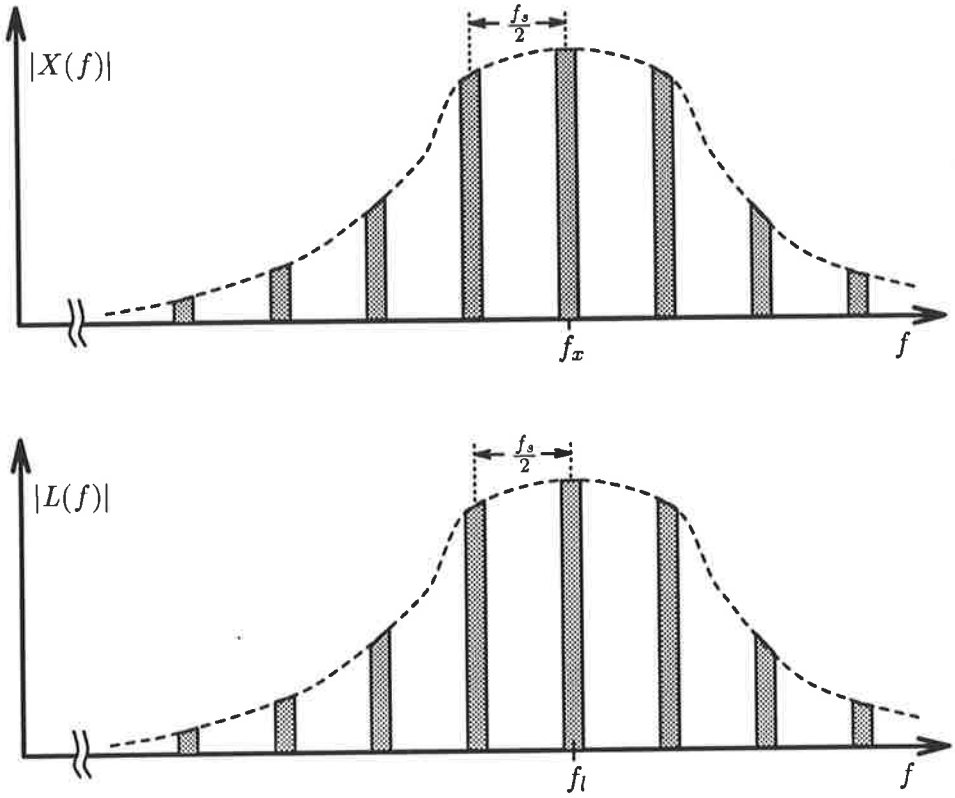


Figure 6.7: Illustration of mixing of aliased components (common sampling rate). The variation of the amplitudes is assumed to include the effects of an anti-aliasing filter. Components which contribute energy to the passband are shown shaded.

and if $f_{s_x} = f_{s_l} = f_s$,

$$f = (n \pm m)f_s \pm f_1 \pm f_2 + k. \tag{6.14}$$

where k is a constant to allow for the frequency translators. Now, suppose $n = m = 0$ and $f = f_2 - f_1$ is the desired component, which corresponds to the lower sideband. Extension to the case of the upper sideband is straight forward. Then there will also be contributions at this frequency due to all $\{\,(n,m) \mid n - m = 0\,\}$. See figure 6.7.

Reducing these aliased components so that they are smaller than the dis-

tortion components is difficult. The design target for these experiments was to be able to detect nonlinear distortion components 80 dB down from the linear components (based on two tone IMD measurements). At a sampling frequency of 5 MHz and assuming the signal is at a centre frequency of $f_s/4 = 1.25$ MHz, the requirement for the anti-aliasing filter is 80 dB rejection 2.5 MHz from the centre frequency at a centre frequency of 15 MHz. The passband needs to be approximately 100 kHz wide. This is a fall off of some 47 dB per decade which is certainly possible, but becomes more difficult to meet if there are also constraints on passband ripple and phase linearity.

Now suppose the sampling rates are different and $f_{s_x} = f_{s_l} - 2\Delta f$ where Δf is the band width of the signal. Referring to figure 6.8, the aliases of desired components do not have commensurate frequencies except at a distant frequency such that

$$\begin{aligned} m f_{s_l}/2 &= (m+1) f_{s_x}/2 \\ &= (m+1)\left(f_{s_l}/2 - \Delta f\right), \end{aligned}$$

$\Rightarrow$

$$m = \frac{f_{s_l}}{2\Delta f} - 1,$$

$\Rightarrow$

$$\Delta f_1 = \left(\frac{f_{s_l}}{2\Delta f} - 1\right) f_{s_l}/2, \tag{6.15}$$

where Δf_1 is the distance from the centre frequency. Using the previous numbers of 5 MHz for f_{s_l} and 100 kHz for Δf, $\Delta f_1 = 60$ MHz. This figure is greater than the centre frequency of 15 MHz and filtering of components this far from the centre frequency must be done by a filter on the D/A output, before the translation process. In practice, a post translation filter is still desirable because the band limited components illustrated in figures 6.7 and 6.8 become broadened by any distortion encountered before the receiver first mixer. However, if the first aliased components are reduced by 20 dB, then the 3rd and higher order distortion components are reduced by 60 dB or greater so if the anti-aliasing

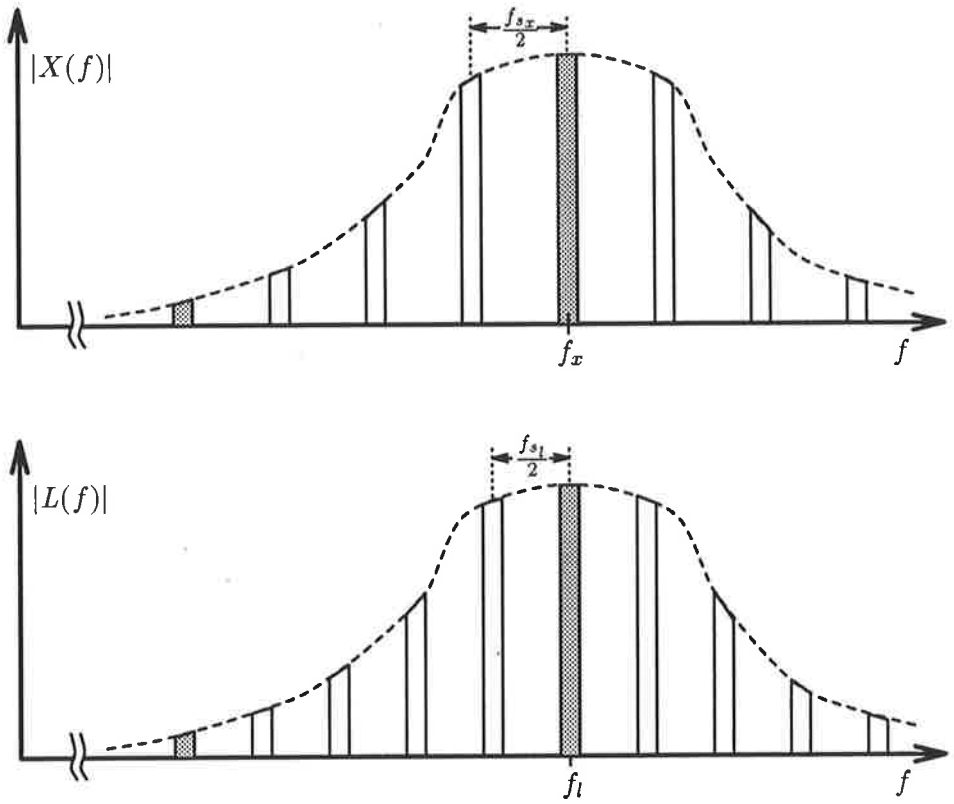


Figure 6.8: Illustration of mixing of aliased components (different sampling rates). The variation of the amplitudes is assumed to include the effects of an anti-aliasing filter. Components which contribute energy to the passband are shown shaded.

filter has 20 dB rejection by the first alias, then errors in the estimation of the nonlinearities should be negligible.

There is a second reason for using different sampling rates for the generation of the local oscillator signals and for the receiver input. The number of data samples in the signal generation was restricted to 131072 (128k) samples. At 5 MHz that is only 0.026 seconds. The receiver output is narrowband and sampling at 16 kHz (the Nyquist rate) we would only get 418 frequency estimates, which is not enough to identify the parameters in a model with more than 418

parameters even assuming zero noise. There is a mismatch in the quantity of information supplied in the input and the quantity of information in the output. A total of 256k data samples (taking into account the input signal and the local oscillator signal) have been input, but we have only got 418 independent samples out. The reason, of course, is that most of the information has been filtered out by the receivers frequency selectivity. By sampling the local oscillator data and the input data at slightly different rates, the combined sequence does not repeat until many cycles through the input data.

Because the input sequences are repeated, the spectra of the signals are in the form of a fine comb. The repetition rate of the RF input sequence, and hence the line spacing of the comb, is

$$\begin{aligned} f_{r_x} &= \frac{4.8 \times 10^6}{2^{17}} = 36.62109375\,\text{Hz} \\ &= \frac{5^5 \times 2^6}{2^{17}} \left(2^3 \times 3\right)\,\text{Hz}, \end{aligned} \tag{6.16}$$

and

$$\begin{aligned} f_{r_l} &= \frac{5.0 \times 10^6}{2^{17}} = 38.1469726563\,\text{Hz} \\ &= \frac{5^5 \times 2^6}{2^{17}} \left(5^2\right)\,\text{Hz}. \end{aligned} \tag{6.17}$$

The repetition rate of the combined input is the largest common factor of f_{r_x} and f_{r_l} which is

$$f_{r_o} = \frac{5^5}{2^{11}} = 1.52587890625\,\text{Hz}. \tag{6.18}$$

Note that the spacing of the lines in the output spectrum (equation 6.18) is very much finer than the spacing of the lines in the input spectrum (equation 6.16). Thus there is much more information (by a factor of 24) in the output spectrum than there would be if the local oscillator input and the RF input spectrum had a constant sampling rate of 4.8 MHz.

To simplify FFT processing, the output sampling rate should be a power of two multiple of f_{r_o}. A/D sampling rates are derived by dividing a 10 MHz

clock, so the output sampling rate staisfies

$$f_{s_o} = \frac{10^7}{n}$$

and

$$f_{s_o} = m f_{r_o}, \tag{6.19}$$

which implies

$$mn = 2^{18} \times 5^2. \tag{6.20}$$

If $m = 2^{11}$, then from equation 6.18, $f_{s_o} = 5^5 = 3125\,\text{Hz}$. Since m must be a power of two, higher sampling rates must also be power of two multiples of 3125 Hz which is easier to picture. The other requirement on the A/D sampling rate is that it must satisfy the Nyquist rate for the output data.

In practice, components of the receiver output were present outside the nominal passband[4] and a higher sampling rate was used than would otherwise be expected. The actual sampling rates used are tabulated in table 6.2.

Port	Rate	No.	Period
LO Input	5.0 MHz	131072	0.0262
RF Input	4.8 MHz	131072	0.0273
Output	25 kHz	131072	5.24

Table 6.2: Sampling parameters.

6.4 Analysis Algorithms

6.4.1 Complex Envelope Processing

Whilst figures 6.2, 6.3 and 6.4 in principle show methods of estimating and correcting distortion in a receiver, the data rates at the inputs are much higher

[4]This was an experimental receiver and the filter determining the band width differed considerably from that actually used in the Radar.

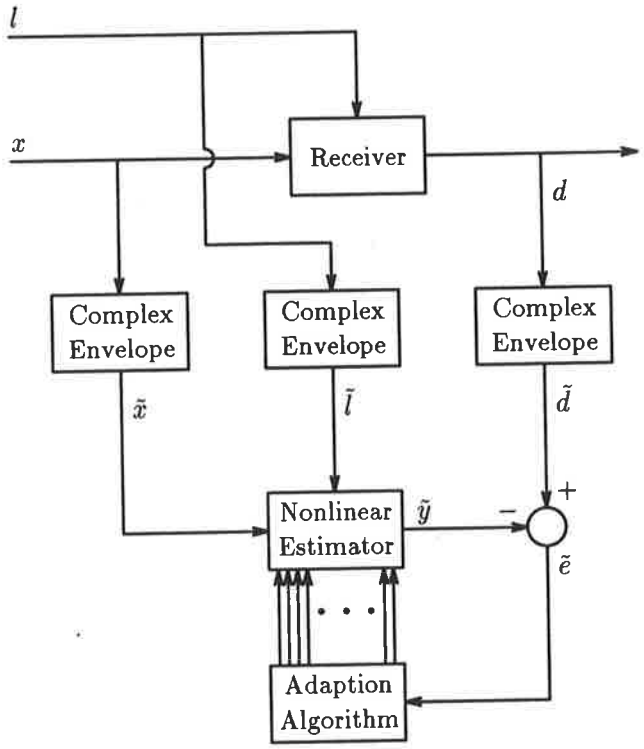


Figure 6.9: Estimation of a two Input receiver

than the information content makes necessary. In section 3.4, we showed that, in a bandpass context, distortion of a signal can be modelled as a distortion of the complex envelope of the signal.

The configuration for estimation of distortion of the complex envelope is not greatly different from estimation of distortion of the original signals. The modified method is shown in figure 6.9 (c.f. figure 6.2).

The only extra processing required is the calculation of the complex envelopes. There are some practical difficulties in calculating complex envelopes continuously as the definition of the complex envelope contains the Hilbert transform and the Hilbert transform of a continuous signal can only be approximated. In the case of a finite, batch-processed signal, however, it is easy to calculate to calculate the complex envelope. The process is [29]:

1. Take the Fourier Transform $\mathbf{X}(\omega) = \mathcal{F}(x(t))$.
2. Zero the negative frequencies.
3. Shift (ω) down by ω_0 .
4. Inverse Fourier Transform.

Operation 3, above, is slightly complicated by the way the output of an FFT is stored in an array[5].

6.4.2 The Ideal Receiver and Sample Interpolation

Figures 6.3 and 6.2 both have an "Ideal Receiver" in the block diagram. The HFR receiver performs two functions, one is the frequency translation function and the other is the "de-ramping" function. Doing these operations directly involves emulating the mixing and filtering processes in the real receiver (see figure 6.1) which is complex and numerically intensive[6]. It is much easier to emulate an ideal receiver by performing operations on the complex envelopes of the RF input and local oscillator signals. Essentially, the complex envelope of the ideal receiver output is simply the product of the complex envelopes of the RF and LO inputs. There are, in practice, two complications.

The receiver de-ramping process (the first mixer of figure fig-rx-block), involves mixing and selecting the *lower* sideband. Assuming the original signals were real, $\mathbf{Y}(\omega_0 - \omega) = \mathbf{Y}(\omega_0 + \omega)^*$, where $y(t)$ is the output of the first mixer prior to filtering. Therefore, the de-ramping can be modelled by swapping the positive and negative frequency components of $\tilde{\mathbf{X}}(\omega)$ prior to the multiplication with the local oscillator signal. This reversing of frequency components can be

[5]Positive frequencies first, then negative frequencies.

[6]Although simplifications can be made by reducing the number of conversions if ideal filtering operations are performed.

performed by taking the Fourier transform, doing the swapping, and taking the inverse Fourier transform, but it is more efficient to combine it as part of the process of finding the complex envelope. This modified complex envelope $\tilde{x}_l(t)$ can be found thus (c.f. section 6.4.1):

1. Take the Fourier Transform $\mathbf{X}(\omega) = \mathcal{F}(x(t))$.
2. Zero the *positive* frequencies.
3. Shift $\mathbf{X}(\omega)$ *up* by ω_0.
4. Inverse Fourier Transform.

The second complication in calculating the output of the ideal receiver, is due to the different sampling rates for the RF and LO inputs (see table 6.2). We need to interpolate to a common higher sampling rate. Further, since both inputs have the same number of samples but different sampling rates, both signals need to be replicated so they have a common duration. The lowest common multiple of the sample rates is $25f_{s_x} = 24f_{s_l} = 120\,\text{MHz}$ as can be seen from equations 6.16 and 6.17 and table 6.2. Interpolation in the time domain is the same as zero padding in the frequency domain so the straight forward method is (for the RF input):

1. Replicate to a length of $24 \times 128\text{k}$.
2. Take the FFT.
3. Pad with zeros to a length of $25 \times 24 \times 128\text{k}$.
4. Take the Inverse FFT.

The same process is used for the LO signal except using 25 instead of 24 and visa-versa. The problem with this approach is that it requires a 78643200 sample IFFT which is quite impractical.

A more efficient method was derived taking advantage of the fact that the input signals are band limited and the receiver output is band limited. For the purpose of comparison with the sampled receiver output, 128k samples is the maximum. In fact, because the receiver sampling rate is high enough to allow a guard band, we need less than 128k samples. The strategy is to perform the simulation in such a way as to minimize the number of samples to be stored at any point. One way of achieving this is to work with a complex envelope representation as early as possible. Another useful property is that replication in the time domain is equivalent to zero stuffing[7] in the frequency domain. The entire process (including calculation of the complex envelopes) is as follows (see also figure 6.10):

1. Take FFT's of the RF(x) and LO(l) inputs.
2. Zero the *positive* frequencies of $\mathbf{X}(\omega)$ and the *negative* frequencies of $\mathbf{L}(\omega)$.
3. Shift $\mathbf{X}(\omega)$ *down* by ω_0 and $\mathbf{L}(\omega)$ *up* by ω_0.
4. Stuff $\mathbf{X}(\omega)$ with (24 - 1) zeros between each sample, and $\mathbf{L}(\omega)$ with (25 - 1) zeros between each sample. Note that because $\mathbf{X}(\omega)$ and $\mathbf{L}(\omega)$ are narrowband this step can be preformed without increasing the number of samples. This step is equivalent to replication in the time domain.
5. Zero pad $\mathbf{X}(\omega)$ and $\mathbf{L}(\omega)$. In principle we zero pad to $24 \times 25 \times 128$k, but in practice, the tails of all zeros can be ignored. That is because the next two steps are equivalent to convolution in the frequency domain, and a long tail of zeros wider than the band width of the signal doesn't contribute to the convolution. In practice we zero pad to the smallest power of two greater than the sum of the band widths of $\mathbf{X}(\omega)$ and $\mathbf{L}(\omega)$.
6. Take the IFFTs of $\tilde{x}(t) = \mathcal{F}^{-1}\mathbf{X}(\omega)$ and $\tilde{l}(t) = \mathcal{F}^{-1}\mathbf{L}(\omega)$.

[7]Inserting zero samples between the existing samples.

7. Output $\tilde{x}_l(t)$ and $\tilde{l}(t)$ which are the complex envelopes of $x(t)$ and $l(t)$ converted to a common sampling rate.

8. Multiply $\tilde{x}_l(t)$ and $\tilde{l}(t)$ and output as the complex envelope of the output of an ideal receiver.

The measured output data must also be converted to a complex envelope at the same interpolated sampling rate. The interpolation (by zero padding in the frequency domain) is performed in the same way as for the RF and LO inputs. The sampling interval includes 8 repetitions[8] of the output data therefore, only 1 in 8 frequency estimates will be non zero. In practice noise and spurious signals mean this is not strictly true, however, setting them to zero effectively improves the signal to noise ratio of the sampled output. Once this has been done, there is little point in maintaining such a high frequency resolution and the zero samples can be removed in a process which is the inverse to the zero stuffing performed for the processing of the RF and LO data.

6.4.3 Adaptive filter Algorithms

The direct solution method of estimating the filter weights was used because it is simple, robust and has no convergence time. Since the nonlinear distortion was assumed to be of small energy compared to the linear components, the eigenvalue spread would be large and gradient algorithms would probably not converge sufficiently in a reasonable time to estimate the nonlinear weights. Unfortunately, the computation time for the direct solution method is excessive for a large number of taps which prompted the author to investigate the RLS methods (section D). The RLS methods have not been successfully implemented in software and this is a suggested topic for further research.

[8]In practice most results were obtained by discarding the first samples to avoid the startup transient. In those cases only 64k samples were processed and the relevant factor here was 4.

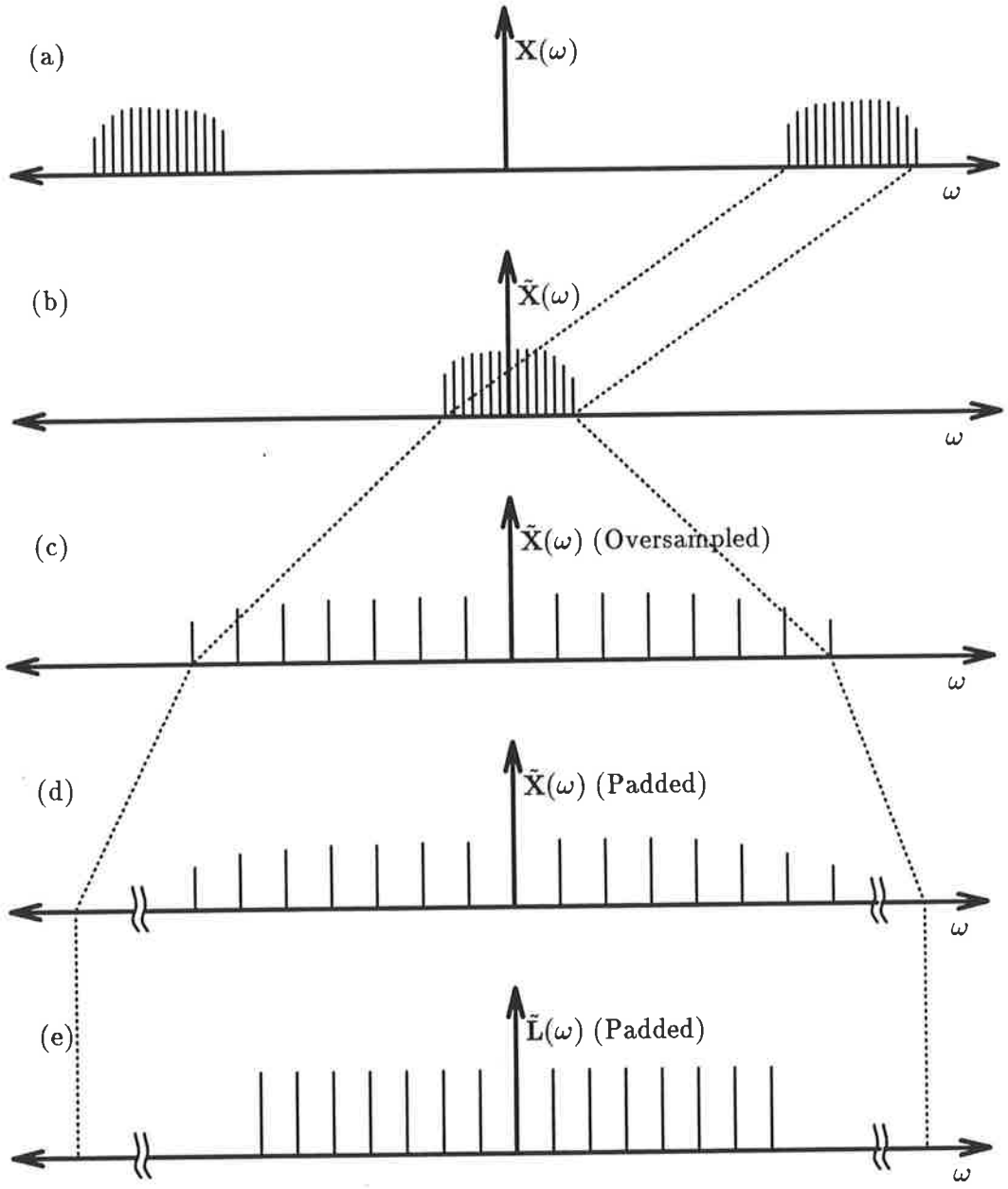


Figure 6.10: Illustration of ideal receiver simulation.

The direct solution method provides the best possible estimation of the weights for a given number of input samples. To that extent it places an upper bound on the performance achievable by any other methods.

Two versions of the program were written, one using the QR algorithm to solve the Wiener-Hopf equations and one using singular value decomposition (SVD). Whilst the auto-correlation matrix is guaranteed to be positive semi-definite, it is not necessarily positive definite. The QR algorithm is well behaved with poorly conditioned matrices, but if the correlation matrix is singular, a unique solution does not exist. In the linear case, it is known [42] that a singular correlation matrix results when the power spectrum is zero at some frequency. The input to the adaptive filter is, in the HFR experiments, band limited. In the estimation case, the input occupies almost the full band width but the number of frequency estimates was rounded up to a power of two when the complex envelopes were calculated and consequently the signal does not occupy the full band width. In the correction case, the band width is even narrower as it has been reduced by the receiver selectivity. If a single input corrector is used, this can be alleviated by reducing the effective sample rate of both the input and the desired output, but in the case of a two input corrector we are stuck with a narrowband width input. One solution is to add random noise to ensure that the spectrum never goes to zero. Another, possibly more elegant, method is to use singular value decomposition (SVD).

The SVD process returns the matrices $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Lambda} \mid \boldsymbol{A} = \boldsymbol{U}^{*-1}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}\}$ and consequently the solution of the Wiener-Hopf equation 5.21 is

$$\boldsymbol{c} = \boldsymbol{V}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{*}\boldsymbol{v}. \tag{6.21}$$

The matrix $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues λ_i in reducing order. Inverting $\boldsymbol{\Lambda}$ is performed simply by inverting each of the elements. The trick is to say that if any of the λ_i are zero (or less than a certain value) then replace $1/\lambda_i$ with zero. In equation 6.21, zeroing some of the λ_i corresponds to zeroing the corresponding columns of $\boldsymbol{U}$ and the corresponding rows of $\boldsymbol{V}$. Zeroing some

of the elements of $\mathbf{\Lambda}^{-1}$ reduces the norm of vector $\mathbf{c}$. This method can be summarized as detecting when a system is under determined and selecting the solution which has a minimum norm.

6.5 Results

6.5.1 Equipment and Procedure Verification

In order to gain confidence in both the hardware and the software some simple test signals were used initially. With signal 4 of table 6.1 as the RF input and signal 3 of table 6.1 as the LO input, the spectrum of the measured output is shown in figure 6.11. The strange looking noise floor was found to be due to the receiver start up transient. Omitting the first 1k samples from the analysis results in the spectrum in figure 6.12. The spectrum of the simulated output of an ideal receiver with the same inputs is shown in figure 6.13.

Observe that the simulated output of an ideal receiver (figure 6.13) correctly estimates the main output components of the real receiver (figure 6.12). This case is also simple enough to be able to predict how the simulated output should look. Figure 6.13, clearly shows a doublet shifted by each of the three sinusoids in the triplet. Observe in figure 6.12 the 3rd harmonic distortion clustered around 10 kHz and the rather larger odd order distortion components clustered around 0 Hz. Also observe a small amount of IMD distortion products observable as "sidebands" of the "ideal" signal. The shifted doublets of figure 6.13 appear to have been converted to shifted triplets in figure 6.12. This is attributable to the local oscillator input being overdriven to the point of clipping[9].

[9]This is the normal operating mode for such a mixer.

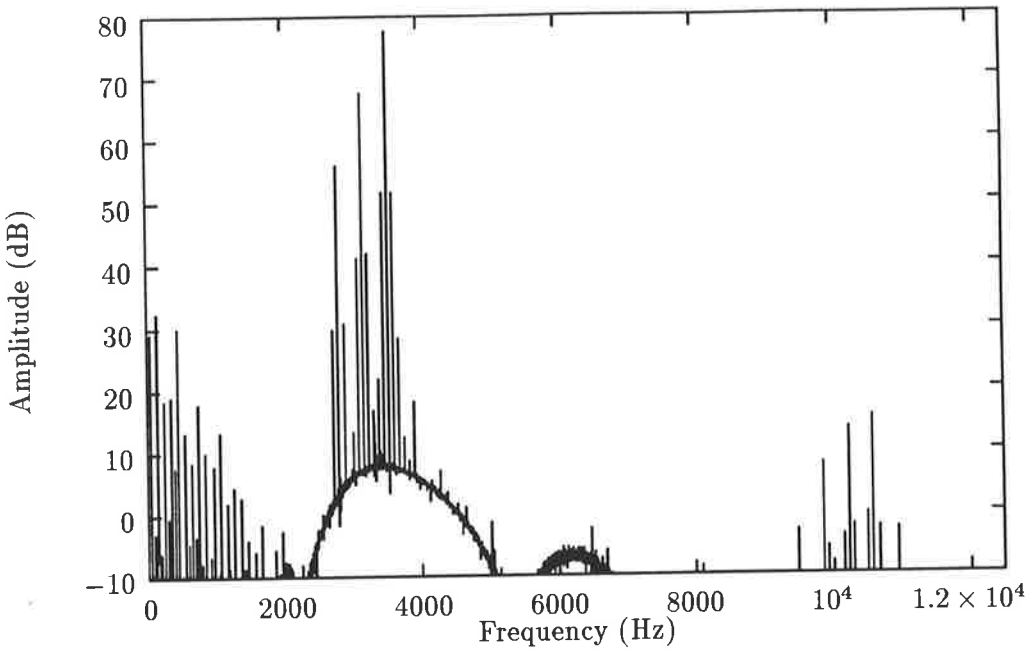


Figure 6.11: Measured output. Signal 4 of table 6.1 as the RF input and signal 3 of table 6.1 as the LO input.

The procedure was then repeated using more complex signals. With noise (signal 10 of table 6.1) as the RF input and a chirp (signal 6 of table 6.1) as the LO input: The full measured spectrum is shown in figure 6.14 and the full simulated spectrum is shown in figure 6.17. The zero amplitude points have not been plotted. A portion of the spectrum of the measured output is shown in figure 6.15. A portion of the spectrum of the simulated output of an ideal receiver with the same inputs is shown in figure 6.18. Finally, figures 6.16 and 6.19 show a portion of the measured and simulated spectra with the zero amplitude points omitted.

Observe the similarities in the peaks of the simulated (figures 6.18 and 6.19) and measured (figures 6.15 and 6.16) spectra. There are also noticeable differences, especially for the smaller peaks. There is also a large amplitude difference which is because the ideal receiver software does not attempt to estimate the

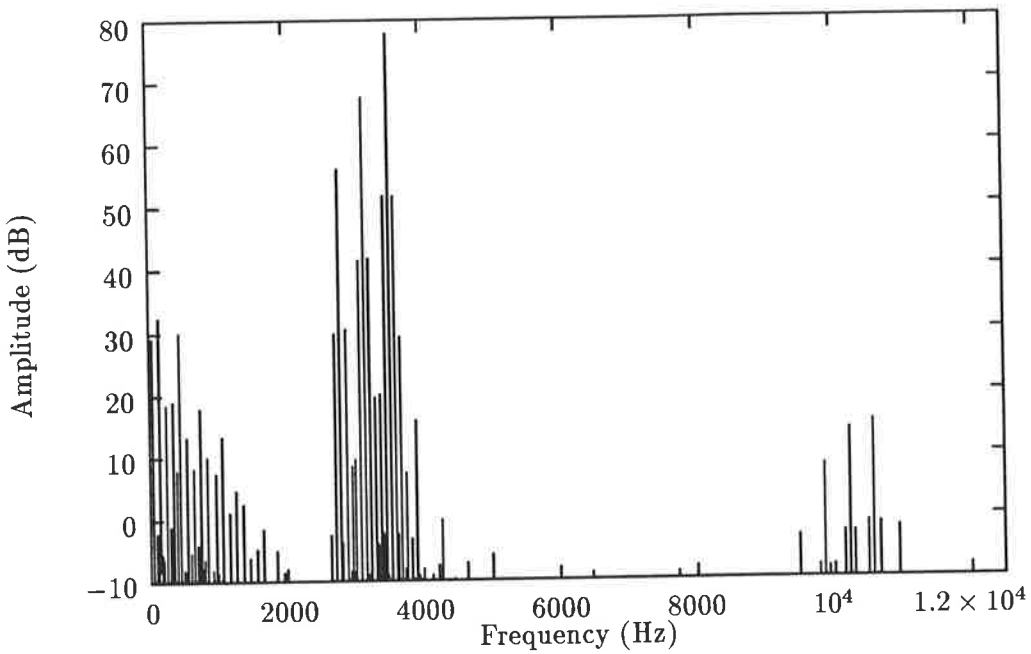


Figure 6.12: Measured output. Signal 4 of table 6.1 as the RF input and signal 3 of table 6.1 as the LO input and omitting the first 1k samples containing the start up transient.

gain of the receiver. A constant gain error is simply absorbed into the filter taps of an estimator or equalizer, so this is not a deficiency. The spectra in figures 6.19 and 6.16 are normalized in amplitude and plotted on the same graph in figure 6.20. The similarity of the curves in figure 6.20 is important because it demonstrates that the all the frequency shifting operations in the ideal receiver simulation have been done correctly. The difference between the measured and simulated curves on a dB (logarithmic) scale is equivalent to the receiver gain which is plotted in figure 6.21.

Figure 6.21 clearly shows the receiver passband. The "fuzz", especially in the passband, is due to nonlinearities. The phase part of figure 6.21 is somewhat complicated by "wrapping" from $-\pi$ to π so we will initially only discuss the amplitude. The author contends that the "fuzz" is due to nonlinearities because

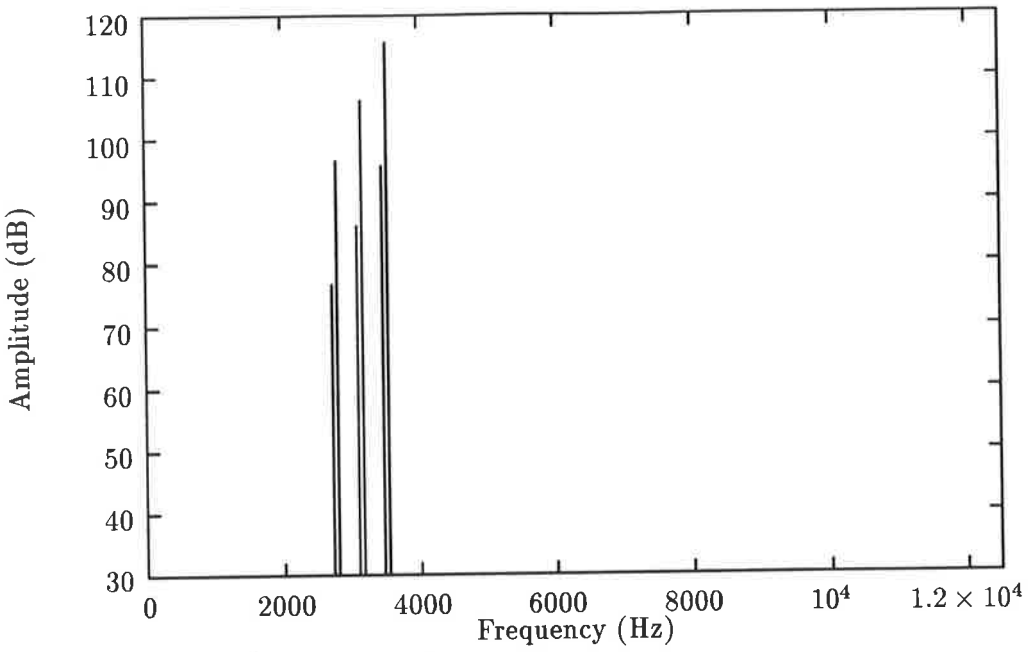


Figure 6.13: Simulated output. Signal 4 of table 6.1 as the RF input and signal 3 of table 6.1 as the LO input.

it is inconceivable that a practical linear system would have such a rapidly varying frequency response and the results of figure 6.12 indicate that the signal to noise ratio is very high.

The experiment was repeated and the results are in figure 6.22. Note that there should not be any correlation between thermal noise components in the two experiments. The amplitude response in figures 6.21 and 6.22 are virtually identical. The phase response is a little harder to compare due to the "wrapping" from $-\pi$ to π. None the less, if sections of the spectrum are avoided where wrapping may have occurred, the phase response compares very well *with an offset*. The phase offset is due to the inability to control the phase of the synthesizers in figure 6.5. The synthesizers are all phase locked to the reference, but there is no way of synchronizing the phases at the start of the sample period. This problem was anticipated in section 4.4 where it was shown that a

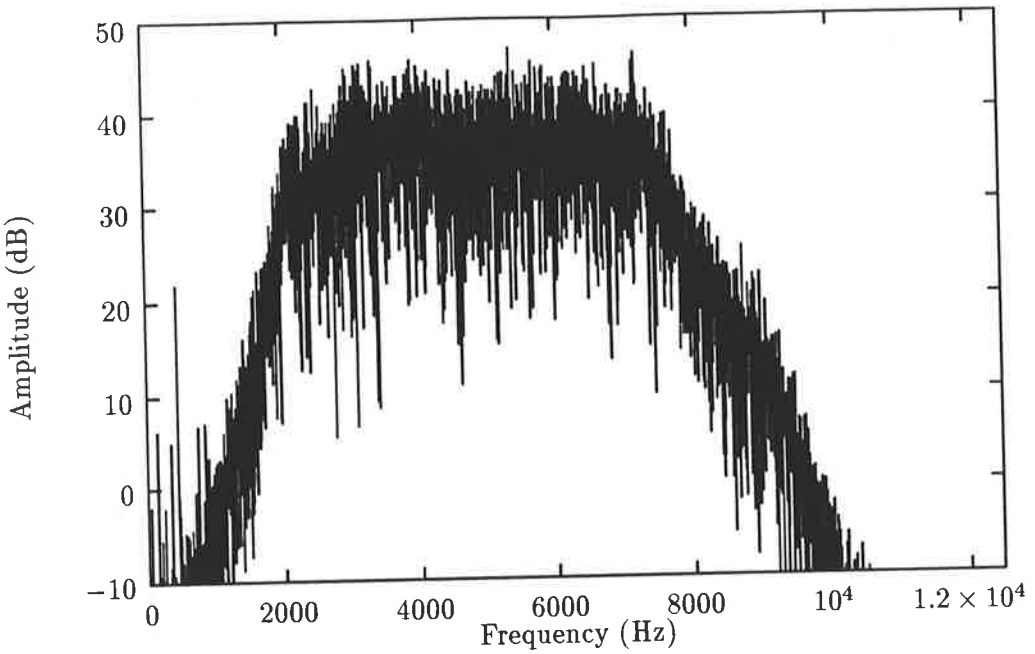


Figure 6.14: Measured output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input and omitting the first 32k samples containing the start up transient. The zero amplitude samples have been omitted.

constant phases shift of a local oscillator, results in a constant phase shift of all the components (linear and nonlinear) in the passband. The results in figures 6.21 and 6.22 confirm that theoretical result.

A potential problem with the experiment is ensuring that the test equipment could have distortion which over-shadows the distortion in the receiver[10]. For a single input, single output system, it is relatively straight forward to check for distortion in the test equipment by observing the distortion with, and without, the unknown system in the path. For a two input device, which performs

[10]A Receiver is, in effect a frequency translator and it is reasonable to ask why, if a distortionless test frequency translator can be made, then why can't a distortionless receiver be made? However, the design of low distortion translators is simplified because their noise figure is relatively unimportant compared to a receiver.

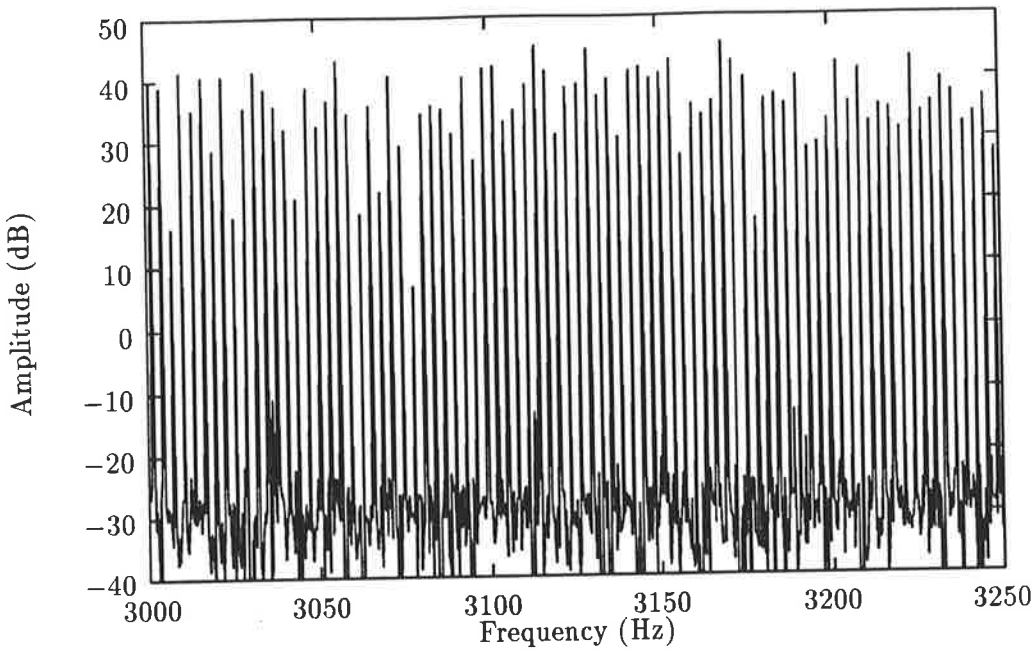


Figure 6.15: Measured output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input and omitting the first 32k samples containing the start up transient.

frequency translation, this simple strategy can not be used. However, a simple test is to reduce the magnitude of computer generated test signal and reduce the variable attenuation at the receiver input (see figure 6.5) by a corresponding amount. Since the amplitude of the distortion products are proportional to A^n, where A is the input amplitude and n is the order of the distortion, if significant distortion was occurring in the LSG translator, then the distortion observed in the receiver output should change. When this was done by using signal 11 in table 6.1 and reducing the attenuator by 6 dB, the output distortion was not noticeably changed. In principle, the same technique could be used to check for distortion in the VHF translator, but this was not done because there is no excess attenuation at the output of the VHF translator which can be removed. Note, however, that the design of the VHF translator is similar to the design of the HF translator.

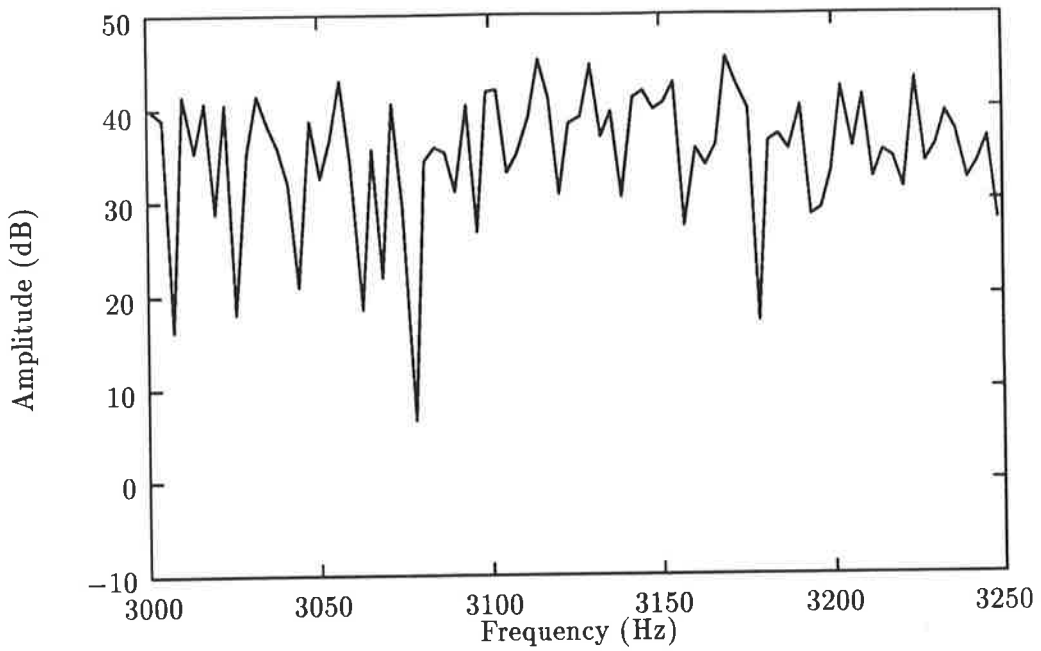


Figure 6.16: Measured output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input and omitting the first 32k samples containing the start up transient. The zero amplitude samples have been omitted.

6.5.2 Nonlinear Adaptive Filter Verification

It was considered desirable to verify that the nonlinear adaptive filter worked and could indeed estimate a known nonlinear system, before applying it to an unknown system.

When evaluating how good an estimation of a known system is, it is tempting to simply compare the weights. However, this can be misleading because the output may be much more sensitive to some weights than others. To complicate matters further, because the signals the weights are applied to are not orthogonal, the weights interact so that two quite different weight vectors can produce models with very similar outputs.

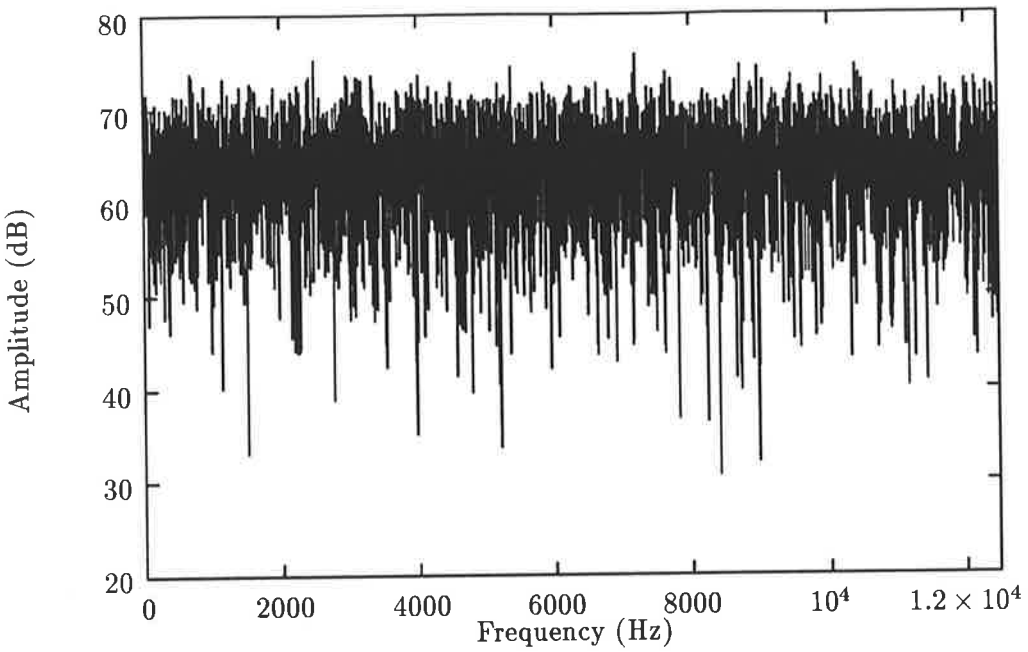


Figure 6.17: Simulated output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input. The zero amplitude samples have been omitted.

Consequently, a better way to evaluate how well a system has been estimated is to compare the output of a model based on that estimation with the output of the original system. This process is illustrated in figure 6.23. The input was pseudo random noise. The known weights were chosen in an *ad hoc* fashion covering linear, nonlinear terms and complex values. Several different sets of known weights were tried with no significant variation in results. The estimation of two input systems were also tried. If the unknown system is exactly representable by the model, and the measurements are not noisy, then the estimation should be precise even with a very small number of data samples. Indeed, in this case, increasing the number of data samples might actually increase the errors due to accumulation of rounding errors. The estimation error is plotted in figures 6.24, 6.25 and 6.26. Figures 6.25 and 6.26 represent the same data plotted so as to emphasize different aspects. Figure 6.24 is merely

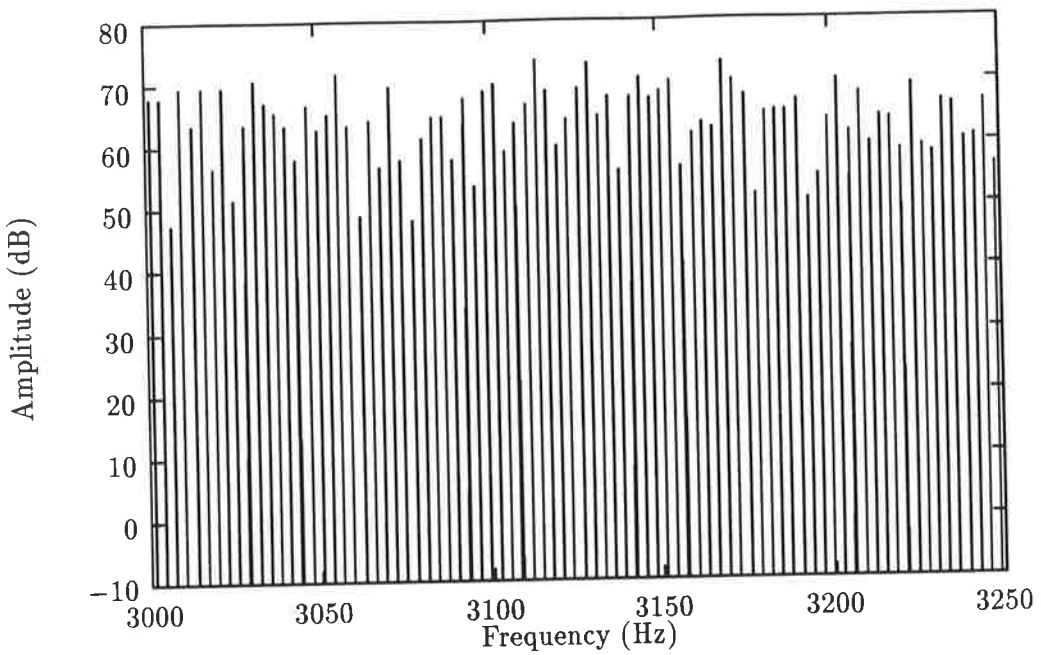


Figure 6.18: Simulated output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

to illustrate that the performance is far from being limited by numerical rounding errors. Figure 6.25 illustrates that in the presence of noise, the estimation error is inversely proportional to the number of samples processed and figure 6.26 illustrates that the estimation error is proportional to the noise. Figure 6.25 shows the number of samples needed to estimate a receiver. For example, thermal noise of $-40\,\mathrm{dB}$ relative to the signal implies $n = 10^{-4}$ and if the number of samples is 60 000, then the estimator output will be in error by less than $-70\,\mathrm{dB}$. This error doesn't take into account errors due to there being insufficient taps on the estimator.

As has been mentioned (section 6.4.3), input signals which don't occupy the full band width can cause ill-conditioning of the correlation matrix. Note that it is the spectrum of the signals at the output of the nonlinear filter (see figure 5.11) which is relevant. In the receiver two input case, there is no point

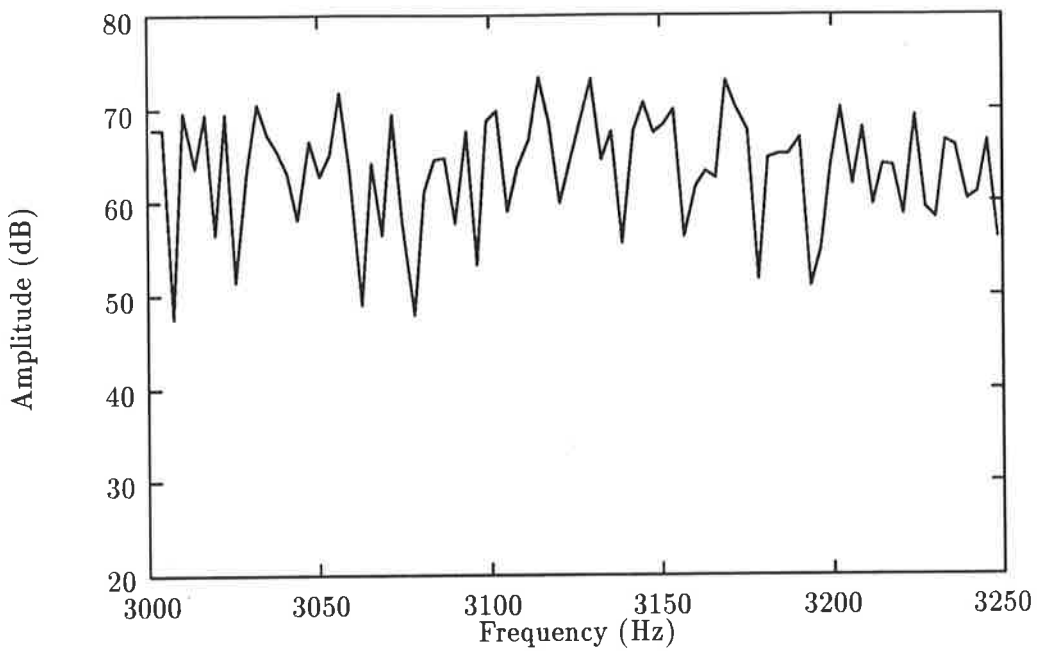


Figure 6.19: Simulated output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input. The zero amplitude samples have been omitted.

in generating terms which don't come out in the receiver passband. As was discussed in section 4.3.2, the terms of interest are those which are an odd power of the x_i times an odd power of the l_i. The narrowest band width case is simply the ideal receiver output.

To observe the effect of a band limited signal on the input of the adaptive filter input, the complex envelope of the ideal receiver output was delayed exactly 13 samples and used as the desired signal for the weight generation process. This experiment was performed for the linear case only and the weights are numbered so that c_i is the weight which is applied to x_i. For a simple delay of 13, the correct estimation is for all the weights to be zero except for $c_{13} = 1$. When the input spectrum was white, weight generation program produced this result with a high degree of accuracy. However, when the input signal is band

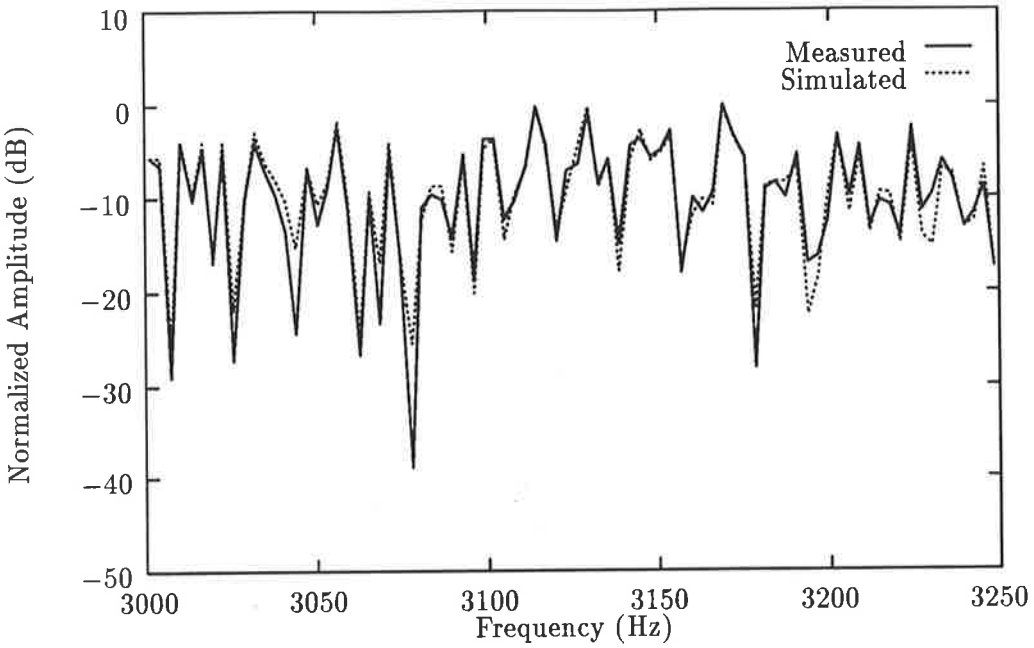


Figure 6.20: Comparison of simulated and measured output. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input. The zero amplitude samples have been omitted.

limited, the problem becomes ill conditioned and significant errors occur in estimating the weights.

The SVD algorithm has been implemented so as to discard eigenvalues less than the largest eigenvalue by a certain factor. We discard $\{ \lambda_i \mid \lambda_i < r\lambda_1 \}$. The estimated weights are plotted in figure 6.27 ($\lambda_{\min}/\lambda_1 = 10^{-6}$) and figure 6.28 ($\lambda_{\min}/\lambda_1 = 10^{-5}$).

Note the very considerable error evident in figure 6.27. A number of other cases for r up to 0.99 were also tested. Increasing r increases the theoretical error but may well reduce the numerical error[11]. There doesn't appear to be a

[11]In theory, it is only necessary to eliminate $\lambda_i = 0$, but retaining very small λ_i makes the problem ill-conditioned.

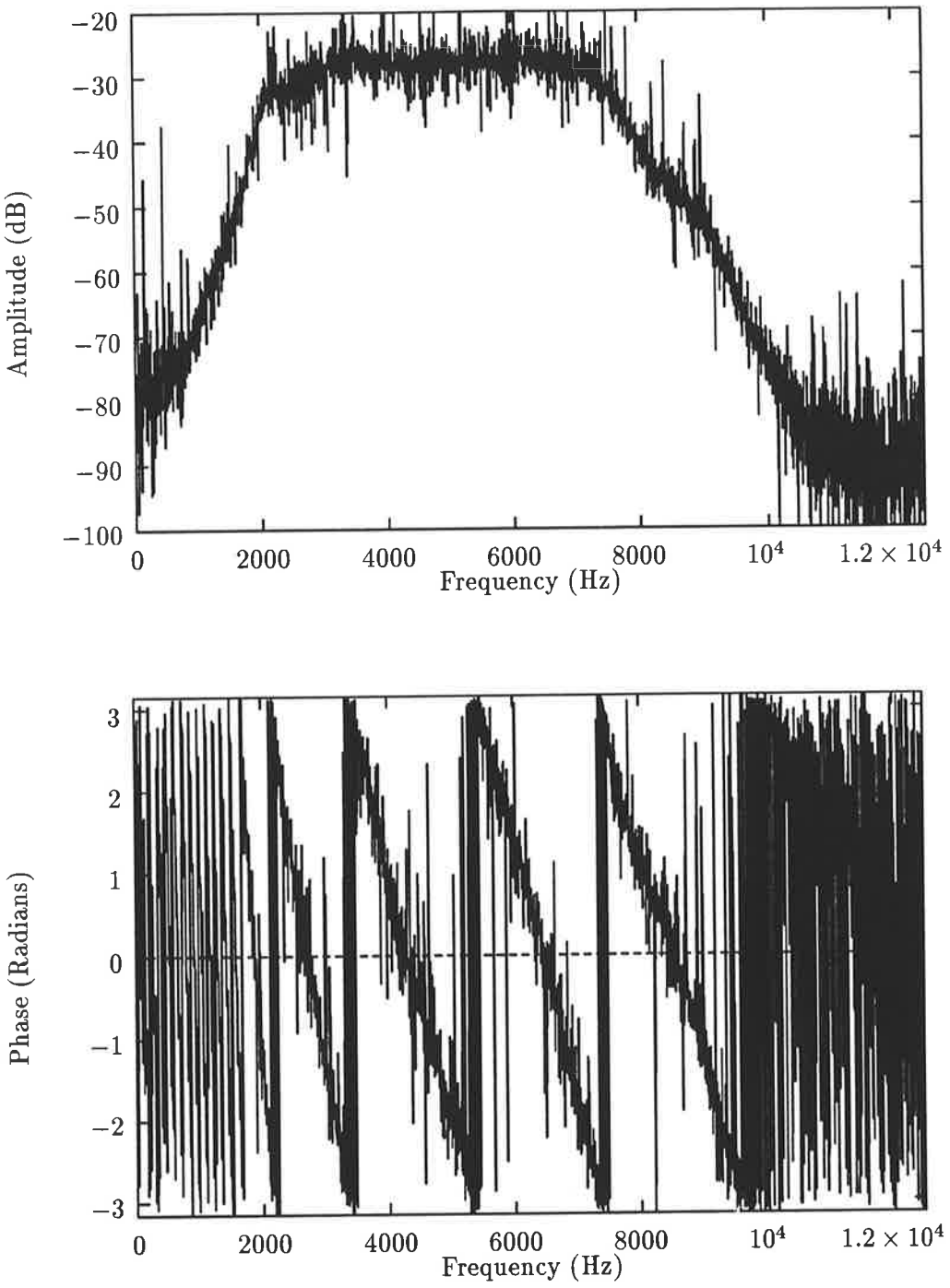


Figure 6.21: Receiver gain. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

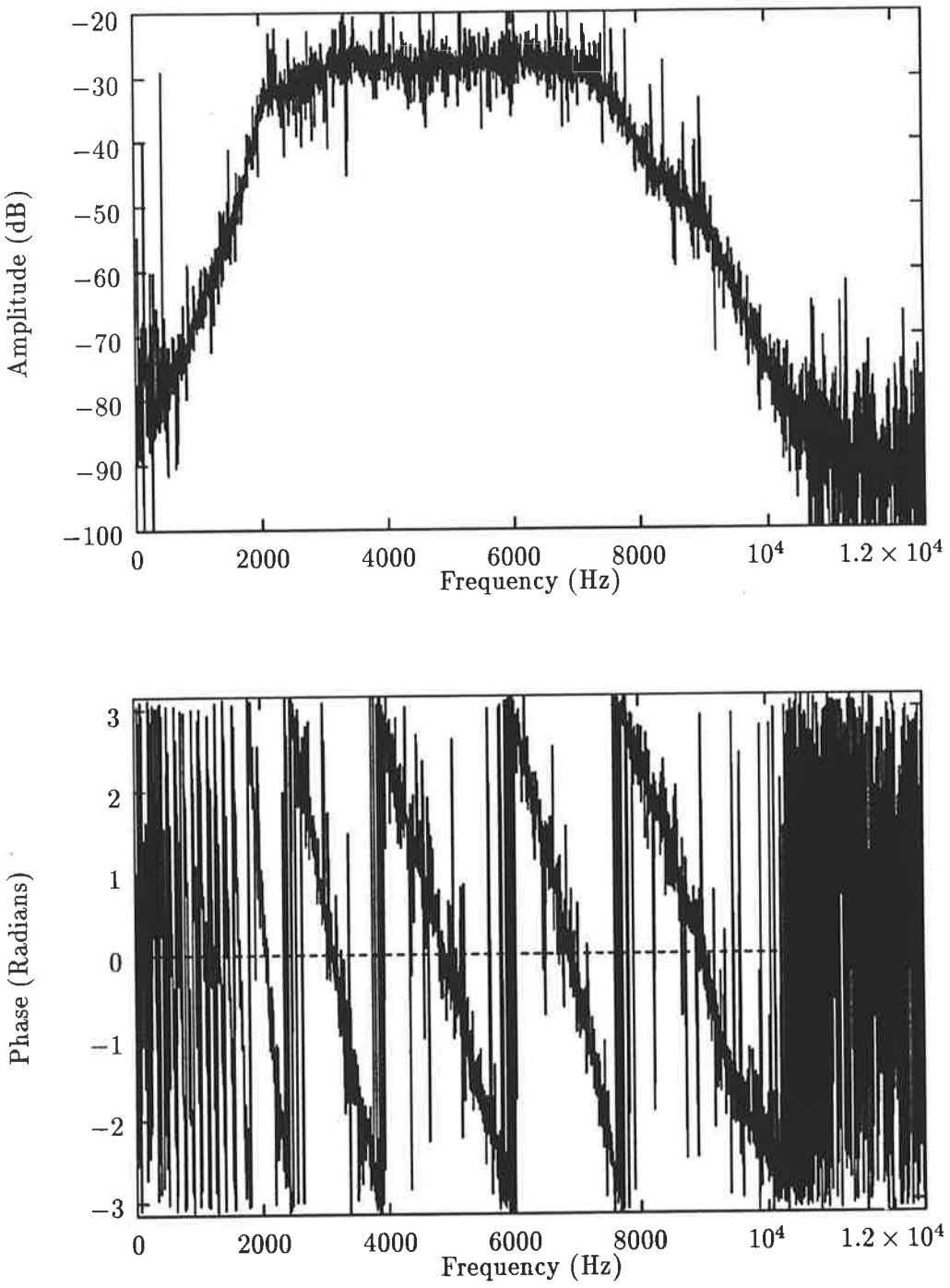


Figure 6.22: Receiver gain (repeat). Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

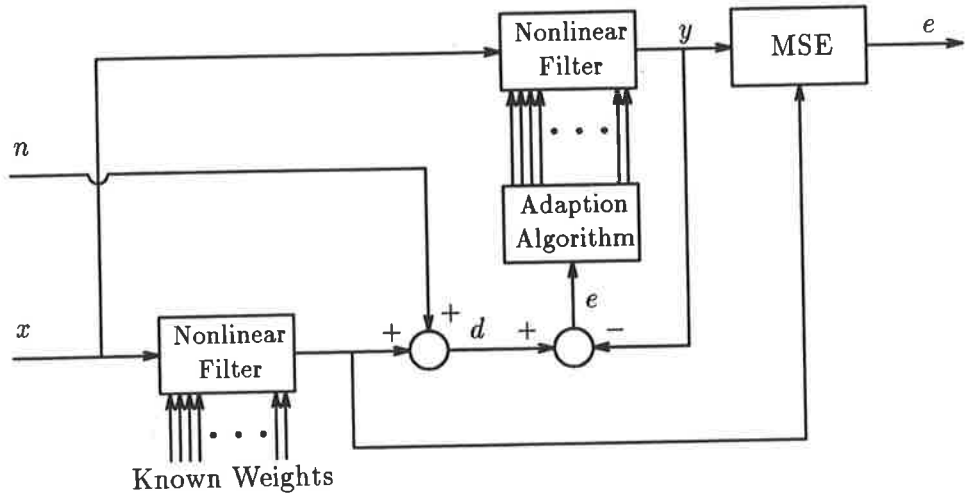


Figure 6.23: Evaluation of adaptive estimation in the presence of observation noise.

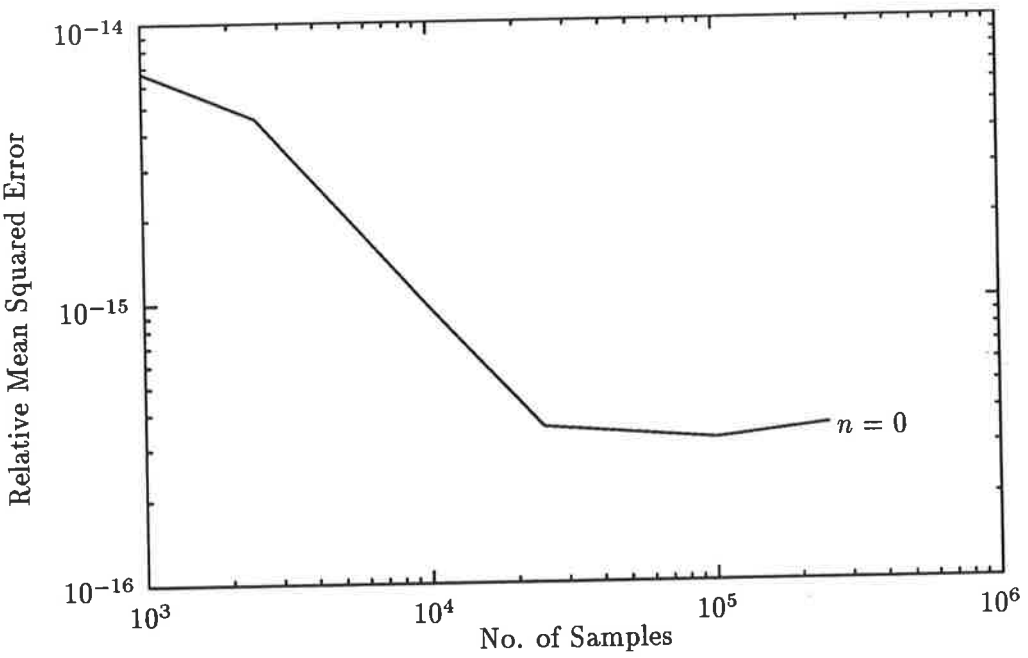


Figure 6.24: Estimation error due to rounding.

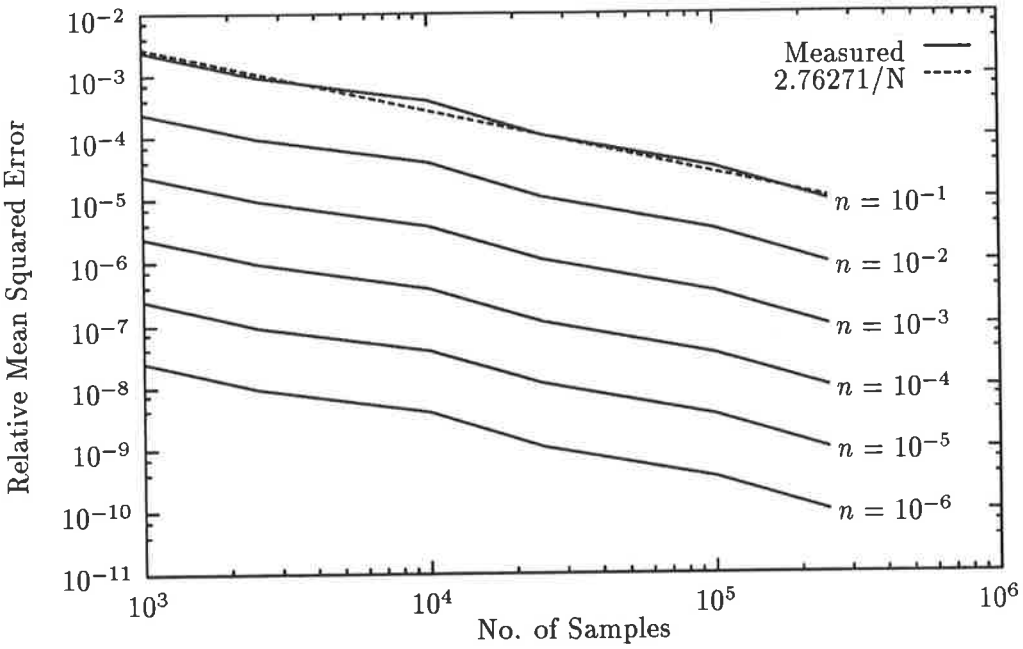


Figure 6.25: Estimation Error as a function of the number of samples processed for various noise levels (see figure 6.23).

way of determining an optimum value for the parameter r and so it remains a variable to be adjusted for best results. A value of $r = 10^{-5}$ has been used in subsequent work.

6.5.3 Estimation

An estimation of the receiver has been performed by the use of a two input nonlinear adaptive filter as is shown in figure 6.9. The weights of the nonlinear filter were restricted to terms of the form $x_i l_i$. This is a second order term, but it reduces reduces to the same form as estimating the linear weights in figure 6.3. These linear weights can be considered the (linear) impulse response of the receiver. The results of this estimation are shown in figure 6.29. The

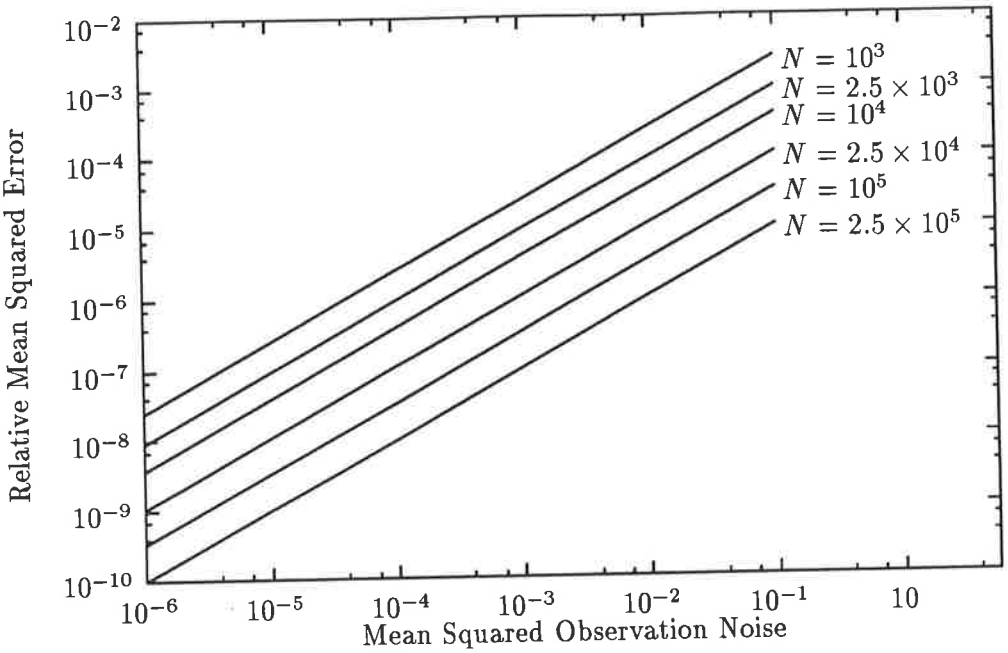


Figure 6.26: Estimation error as a function of the noise (see figure 6.23) for various number of samples.

Fourier transform of the linear impulse response is the system transfer function. The Fourier transform of the data in figure 6.29 was compared with the known (linear) transfer function and the agreement was good over most of the frequency range. In the range of frequencies where the power spectrum of the input to the estimator was zero, the error was large. It is not surprising, that the estimation has no information about the behaviour of the system at the frequencies which were not tested!

The execution time of the weight calculation program with 256 taps took over 10 hrs of CPU time on a Sun 4/280. A run with 512 weights takes over 40 hrs of CPU time on the same machine. The Sun 4/280 is not particularly well suited to this task since the arrays are much larger than the machines data cache. As was discussed in section 5.3.3, the execution time of this algorithm increases as $O(m^2)$ where m is the number of taps, so a faster machine does not allow the

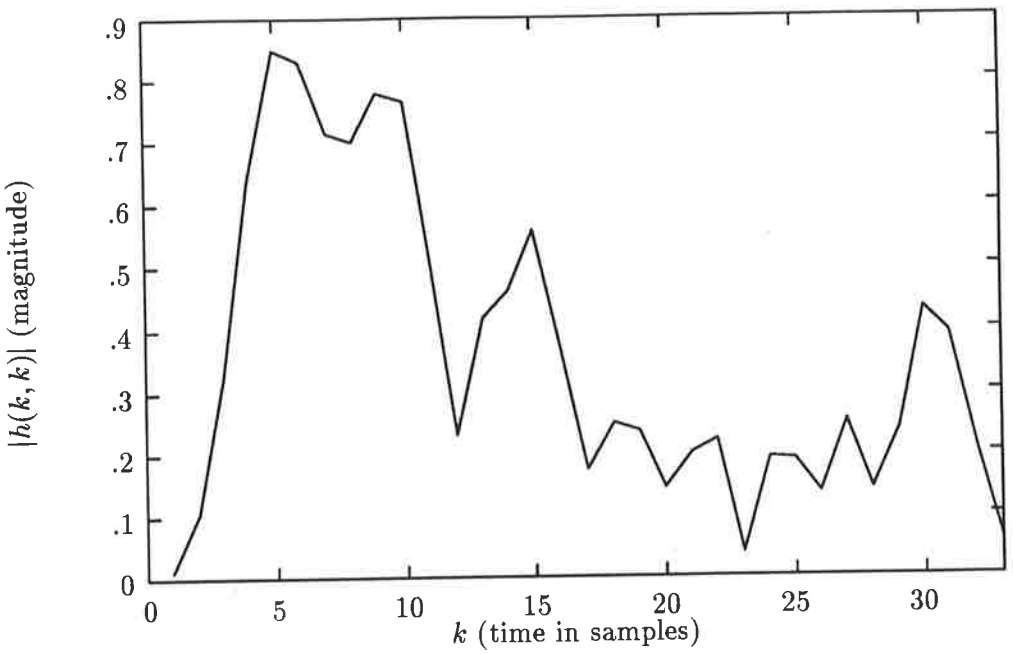


Figure 6.27: Estimation of a pure delay of 13 samples, $\lambda_{\min}/\lambda_1 = 10^{-6}$. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

number of taps to be increased very much. The total number of taps increases very quickly when higher order nonlinear terms are introduced, especially if a general system with memory is to be modelled. It is concluded, therefore, that this adaption algorithm is not practical for estimating this system. Note that the execution time of the model, once the weights have been calculated scales much more kindly as $O(m)$.

The difficulty arises because of the length of the impulse response of the receiver requires a lot of taps even for estimation of the linear terms. The long impulse response arises from the selective filters in the receiver. One possibility would be to pre-whiten the measured receiver output prior to estimation. This may not work directly, because in shortening the linear impulse response it is quite possible that the nonlinear response would be lengthened. This is because

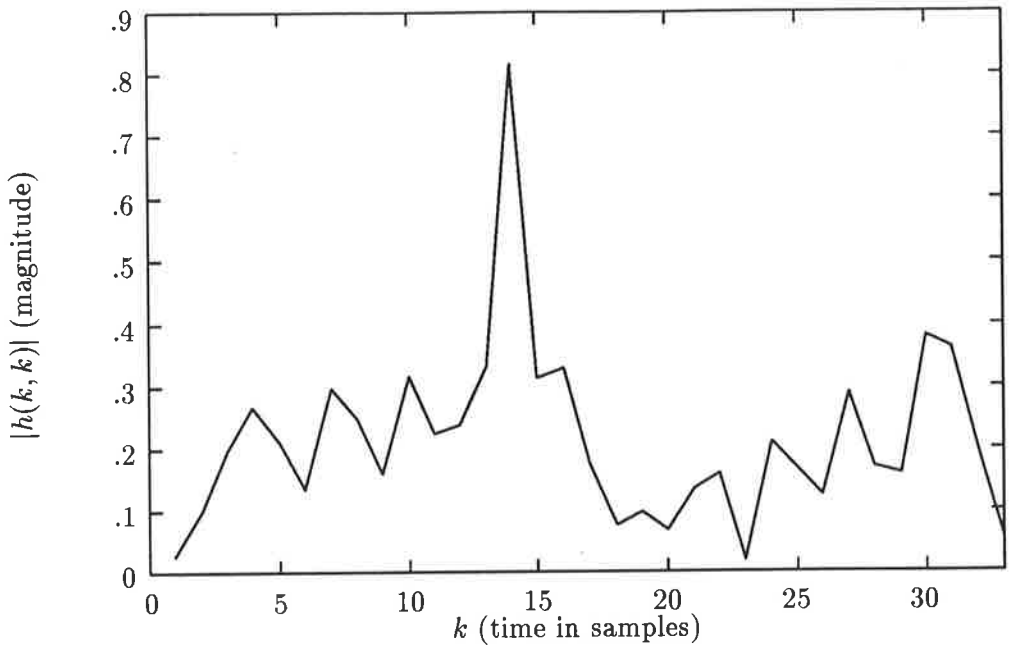


Figure 6.28: Estimation of a pure delay of 13 samples, $\lambda_{\min}/\lambda_1 = 10^{-5}$. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

there are nonlinearities both before and after the band width determining filters in the receiver. However, pre-whitening does reduce the eigenvalue spread which could make it practical to use the gradient algorithm (see section 5.3.2).

It is this problem which prompted the author to develop nonlinear versions of the Recursive Least Squares algorithms (see section 5.3.4 and appendix D). To date, these have not actually been implemented, however.

6.6 Summary

In this Chapter, we have examined the practical difficulties in estimating distortion in an HFR receiver. There are a number of features which make this

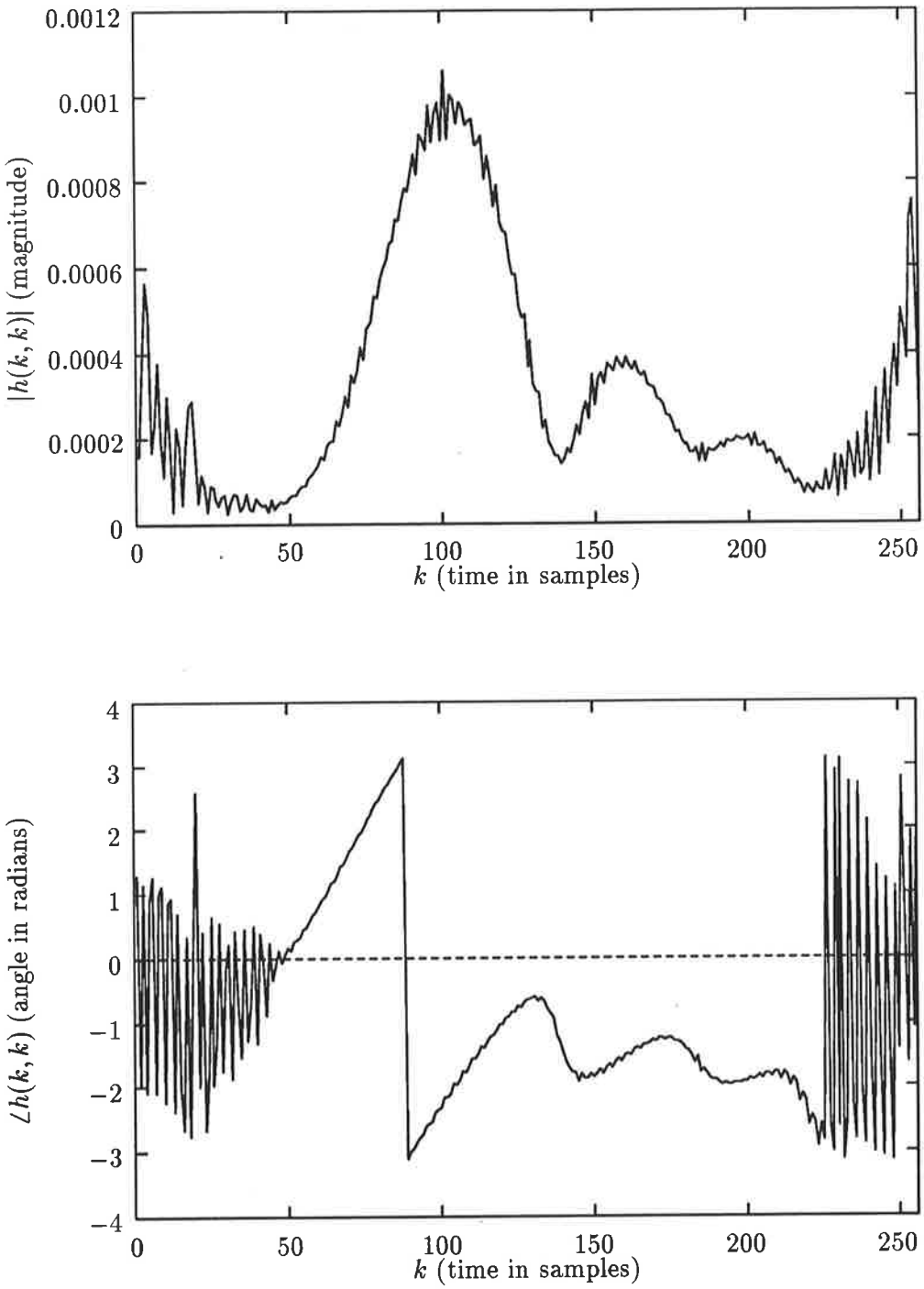


Figure 6.29: Estimation of the receiver. Noise (signal 10 of table 6.1) as the RF input and a "chirp" (signal 6 of table 6.1) as the LO input.

experiment different to other experiments which have been described in the literature. Specifically:

1. The receiver must be modelled as a two input device. The phase of the local oscillator input, relative to the RF input, can not be measured or controlled[12].

2. The receiver has slight, but important, nonlinear distortion in the presence of large linear distortion. The linear distortion is inherent in the high frequency selectivity of the receiver and implies a long impulse response. The small power of the nonlinear components relative to the linear components implies a large eigenvalue spread. The linear weights must be calculated accurately if the nonlinear distortion is to be seen.

3. The receiver's inputs and outputs are at different frequencies. This was dealt with by estimating the distortion in the complex envelope instead of estimating the distortion of the original signals.

4. The band width of the signals applied to the receiver's inputs should be very much greater than the band width of the receiver output to adequately probe the front end circuitry. Consequently, it is difficult to provide enough input samples so that enough information is available in the output for the estimation to be done.

Item 2 above has not been completely solved. The only algorithm implemented to date (direct solution of the Wiener-Hopf equations), has sufficient accuracy, but has excessive computation time. Gradient algorithms, are fast enough in terms of operations per step, but take too long to converge to a point where the small nonlinear distortion can be estimated. It may be possible to reduce the eigenvalue spread by pre-whitening. Nonlinear recursive least squares algorithms have been derived but not implemented to date. It is possible that

[12] All the signals are phase locked, but there is no way to set the initial phase.

other difficulties, such as stability in the face of rounding errors, might be a problem with the RLS algorithms. Finding a suitable adaption algorithm is a topic for future research.

Item 4, above, can be handled by sampling the input data at slightly different sampling rates. This complicates the processing of the data considerably, but the problem is tractable.

Experimental evidence has been gained for the theoretical result derived in section 4.4 where it was shown that a constant phase shift of a local oscillator, results in a constant phase shift of all the components (linear and nonlinear) in the passband (see item 1 above).

Chapter 7

Conclusion

One of the factors limiting the performance of many systems is nonlinear distortion. Traditional ways of correcting distortion such as feedback and feedforward have limitations for broad band high-frequency systems, especially systems which include frequency translators. Pre-distortion or post-distortion is an alternative way of reducing over all distortion which is useful in many circumstances where feedback or feedforward is impractical or impossible. One of the more demanding applications is High Frequency Radar (HFR) Receivers. In this thesis a number of theoretical and practical problems relevant to modelling and correcting nonlinear distortion in HFR Receivers have been solved. Designing a pre-distortion or post-distortion to preform nonlinear correction is equivalent to finding a model of the nonlinear inverse, thus modelling and correction are very similar processes. It is desirable to know whether a model is sufficiently general to model the system, or, in the case of a corrector, the system inverse.

In chapter 2, signal dependent ways of characterizing distortion were examined. "Harmonic Distortion", "Intermodulation Distortion", "Cross Modulation Distortion" and "Polyspectra" were discussed. These terms refer to measures of distortion rather than models of distortion.

It is possible (though not simple) to use these measures of distortion to evaluate the parameters of a model although, as we saw in chapter 3, a general model can not be evaluated with simple inputs consisting of only a few sinusoids.

In chapter 3, various parametric characterizations of distortion were considered to find one suitable for estimating or correcting bandpass systems. The distortion in systems modelled by power series, amplitude and phase describing functions and Volterra series was analyzed for the case of bandpass inputs and outputs. In these cases it was shown that distortion in a bandpass context can be considered as distortion of the complex envelope of the signal. This property is of significance for bandpass systems since it means that much less data need be processed to estimate or correct a nonlinear system. The Wiener model of non-linearities has also been derived but consideration of the bandpass case was deferred to chapter 4.

The power series model is a special case of the amplitude and phase describing function model. Both the power series model and the amplitude and phase describing function model are special cases of the Volterra series model and the form of the Volterra kernels has been deduced.

From consideration of the frequency domain representation of the Volterra kernels it was shown that, for band limited inputs, the Volterra kernels of the model and of the actual system need only match for ω within a hyper-cube of interest. This is analogous to requiring the model of a linear system to match the actual transfer function over a frequency range of interest.

It was shown that for a sufficiently narrowband input signal, any distorting system describable by a Volterra series could be approximated by the amplitude and phase describing function model. It was also shown that testing with a single sinusoid, even at a number of different frequencies, is inadequate to check whether amplitude and phase describing functions are an adequate model or whether a more general Volterra series model is necessary. Frequency indepen-

dence of the amplitude and phase describing functions, as measured by a single sinusoid is a necessary, but not sufficient condition for the amplitude and phase describing function model to be applicable.

Many bandpass systems, including HFR Receivers, perform frequency translation functions. This is done by "mixing" with one or more local oscillator signals. Thus, such systems are actually multiple input systems. In chapter 4 it was observed that mixer manufacturers' linearity specifications implicitly assume a rather simple model of distortion in a two input device. This assumption has been critically examined and found to be invalid if the mixer does not have a "pure"[1] sinusoidal local oscillator.

Most mixer applications employ a sinusoidal local oscillator but some modern mixer applications use non-sinusoidal local oscillator signals such as a linear frequency sweep (a "chirp") and caution must be exercised in applying performance measurements made with a sinusoidal local oscillator to these applications. It may be that a "chirp" changed frequency slowly enough to be considered a sinusoid, but this would depend on the mixer and the frequency sweep rate. The author speculates that useful relationships might arise from studying the case of constant envelope signals as local oscillators.

It was shown that in the case of a bandpass system, a change in phase of one of the input signals results in the same change in phase of all components of the output signal within the passband. This result is of practical importance in the modelling of distortion in a receiver where it is impossible or impracticable to measure the local oscillator signals. It was also shown, by example, that this is not the case for broad band systems.

The discrete time Volterra series is attractive to model or correct nonlinear

[1] A distorted sinusoid is satisfactory since the distortion can be considered part of the block N'' in figure figure 4.1. A "broadened" sinusoid (perhaps due to amplitude or phase modulation) is not satisfactory.

distortion, since it is capable of approximating a very large class of nonlinear functionals arbitrarily well. In chapter 5, it was seen how the discrete time Volterra series can form the basis for a nonlinear digital filter. Nonlinear filters can be made to adapt to a required characteristic. The nonlinear adaptive filter can be used in the roles of estimators, predictors and equalizers in the same way as linear adaptive filters.

The LMS gradient algorithm and the method of directly solving the Wiener-Hopf equations used in linear adaptive filters are directly applicable to the nonlinear case. Other more efficient linear adaption algorithms such as the Recursive Least Squares algorithms are not directly applicable to the nonlinear case. The Recursive Least Squares algorithm has been extended to the nonlinear case in section 5.3.4 and appendix D. The LMS gradient algorithm and the RLS algorithm require $O(nm)$ operations, whereas the Direct Solution of the Wiener-Hopf equations requires $O(nm^2)$ operations assuming that the weights are updated infrequently so that number of operations is dominated by the calculation of the correlation matrix $\boldsymbol{A}$. The RLS algorithms and direct solution of the Wiener-Hopf equations have zero convergence time.

The single input nonlinear digital filter is simply extendible to two or more inputs. The adaption algorithms can also be extended to the multi-input case. The implementation of a suite of programs for efficiently implementing and testing multi-input nonlinear adaptive filters has been discussed.

In Chapter 6, we examined in detail the practical difficulties in estimating distortion in an HFR receiver. There are a number of features which make this experiment different to other experiments which have been described in the literature. Specifically:

1. The receiver must be modelled as a two input device. The phase of the local oscillator input, relative to the RF input, can not be measured or

controlled[2].

2. The receiver has slight, but important, nonlinear distortion in the presence of large linear distortion. The linear distortion is inherent in the high frequency selectivity of the receiver and implies a long impulse response. The small power of the nonlinear components relative to the linear components implies a large eigenvalue spread. The linear weights must be calculated accurately if the nonlinear distortion is to be seen.

3. The receiver's inputs and outputs are at different frequencies. This was dealt with by estimating the distortion in the complex envelope instead of estimating the distortion of the original signals.

4. The band width of the signals applied to the receiver's inputs should be very much greater than the band width of the receiver output to adequately probe the front end circuitry. Consequently it is difficult to provide enough input samples so that enough information is available in the output for the estimation to be done.

Item 4, above, can be handled by sampling the input data at slightly different sampling rates. This complicates the processing of the data considerably, but the problem is tractable.

Experimental evidence has been gained for the theoretical result derived in section 4.4 where it was shown that a constant phase shift of a local oscillator, results in a constant phase shift of all the components (linear and nonlinear) in the passband (see item 1 above).

Item 2, above, places severe constraints on the adaption algorithms and prompted the development of the RLS algorithms. However, the only algorithm implemented to date (direct solution of the Wiener-Hopf equations), has sufficient accuracy, but has excessive computation time.

[2]All the signals are phase locked, but there is no way to set the initial phase.

There are a number of areas requiring further investigation.

Avenues for further theoretical work are as follows:

- The equivalence of distortion of a signal and distortion of its complex envelope needs to be extended to the two input case. We have assumed this equivalence in the experimental work.
- There may be useful insights to be gained looking at distortion in two input systems where one input is constant envelope. The work done in chapter 4 with sinusoidal oscillators is a special case.

There is plenty of scope for improvement in algorithms for nonlinear adaptive filters. Gradient algorithms, are fast enough in terms of operations per step, but take too long to converge to a point where the small nonlinear distortion can be estimated. It may be possible to reduce the eigenvalue spread by pre-whitening. Nonlinear recursive least squares algorithms have been derived in this thesis, but not implemented to date. Issues such as stability in the face of rounding errors need to be addressed and it may be possible to derive more efficient versions.

Because of the inefficiency of the algorithms available, the experimental work in this thesis has been curtailed. Ideally, much more extensive estimation experiments need to be performed, and the issue of nonlinear correction has not been tackled experimentally at all. If, with the aid of more efficient techniques, the discrete time Volterra kernels can be determined for both the estimation and equalization case, then it should be possible to determine experimentally whether simpler models are justified. For example, if amplitude and phase describing functions are an adequate model, then the Volterra kernels should be in the form of equation 3.27.

Appendix A

Complex Envelope Equivalent Distortion.

We show here that the distortion represented by a Volterra series is equivalent to distorting the complex envelope of the signal. The relationship appears to have been first noted by Benedetto et al [8]. The derivation here is somewhat more complete. Consider an input signal x of the form given in equation 3.3.

Let

$$\begin{aligned} w(\tau_1, \tau_2 \ldots \tau_n) &= \prod_{p=1}^{n} x(t-\tau_p) \\ &= \frac{1}{2^n} \prod_{p=1}^{n} \left(\tilde{x}(t-\tau_p) e^{j\omega_0(t-\tau_p)} + \tilde{x}^* e^{-j\omega_0(t-\tau_p)} \right). \end{aligned} \tag{A.1}$$

To enable us to manipulate equation A.1, define the set $A = \{1, 2, 3 \ldots n\}$ and $B_l \subset A$, such that $B_l \neq B_m$, for $m \neq l$. There are 2^n such B_l. Also define r_l to be the number of elements in B_l. Equation A.1 then becomes

$$\begin{aligned} w(\tau_1, \tau_2 \ldots \tau_n) &= \frac{1}{2^n} \sum_{l=1}^{2^n} \left[\prod_{i \in B_l} \tilde{x}(t-\tau_i) e^{j\omega_0(t-\tau_i)} \prod_{i \in B_l'} \tilde{x}^*(t-\tau_i) e^{-j\omega_0(t-\tau_i)} \right] \\ &= \frac{e^{j(2r_l - n)\omega_0 t}}{2^n} \sum_{l=1}^{2^n} \left[\prod_{i \in B_l} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \prod_{i \in B_l'} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \right]. \end{aligned} \tag{A.2}$$

Now applying the narrowband constraint, we discard all terms except those where $2r_l - n = \pm 1$. This requires n to be odd so setting $n = 2k+1$ implies $r_l = k+1$ or $r_l = k$. Therefore

$$w(\tau_1, \tau_2 \ldots \tau_{2k+1}) = w'(\tau_1, \tau_2 \ldots \tau_n) + w''(\tau_1, \tau_2 \ldots \tau_n),$$

where

$$w'(\tau_1, \tau_2 \ldots \tau_n) = \frac{e^{j\omega_0 t}}{2^{2k+1}} \sum_{l|r_l=k+1} \left[\prod_{i \in B_l} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \prod_{i \in B_l'} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \right]$$

and

$$w''(\tau_1, \tau_2 \ldots \tau_n) = \frac{e^{-j\omega_0 t}}{2^{2k+1}} \sum_{l|r_l=k} \left[\prod_{i \in B_l} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \prod_{i \in B_l'} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \right]. \tag{A.3}$$

We can require without, loss of generality, that the Volterra kernel be symmetric [68]. That is to say that $h_n(\tau_1, \tau_2 \ldots \tau_n)$ remains unchanged for any permutation of its arguments $\tau_1, \tau_2 \ldots \tau_n$ and

$$\begin{aligned} H_{2k+1} = &\int\int \ldots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) w'_{2k+1}(\tau_1 \ldots \tau_{2k+1})\, d\tau_1\, d\tau_2 \cdots d\tau_n \\ &+ \int\int \ldots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) w''_{2k+1}(\tau_1 \ldots \tau_{2k+1})\, d\tau_1\, d\tau_2 \cdots d\tau_n. \end{aligned} \tag{A.4}$$

Now since the value of H_{2k+1} is independent of the choice of the variables of integration and since $h_{2k+1}(\tau_1, \tau_2 \ldots \tau_{2k+1})$ is symmetric, it follows that H_{2k+1} is unaffected by the order of the arguments of $w'_{2k+1}(\tau_1, \tau_2 \ldots \tau_{2k+1})$ and $w''_{2k+1}(\tau_1, \tau_2 \ldots \tau_{2k+1})$. By reordering the arguments of w' and w'', A.3 can be expressed

$$w'(\ldots) = \frac{e^{j\omega_0 t}}{2^{2k+1}} \sum_{l|r_l=k+1} \left[\prod_{i=1}^{k} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \prod_{i=k+1}^{2k+1} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \right]$$

and

$$w''(\ldots) = \frac{e^{-j\omega_0 t}}{2^{2k+1}} \sum_{l|r_l=k} \left[\prod_{i=1}^{k} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \prod_{i=k+1}^{2k+1} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \right]. \tag{A.5}$$

There are $\binom{2k+1}{k}$ subsets B_l such that $r_l = k$, consequently there are $\binom{2k+1}{k}$ terms in the summations in equation A.5 and they are all equal. Thus

$$y(t) = \sum_{k=0}^{\infty} H_{2k+1}(x(t)),$$

where

$$\begin{aligned} H_{2k+1} = \binom{2k+1}{k} \frac{e^{j\omega_0 t}}{2^{2k+1}} \int\int \ldots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) \prod_{i=1}^{k} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \\ \times \prod_{i=k+1}^{2k+1} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \, d\tau_1 \, d\tau_2 \cdots d\tau_n \\ + \frac{e^{-j\omega_0 t}}{2^{2k+1}} \int\int \ldots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) \prod_{i=1}^{k} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \\ \times \prod_{i=k+1}^{2k+1} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \, d\tau_1 \, d\tau_2 \cdots d\tau_n. \end{aligned} \tag{A.6}$$

Equation A.6 is of the same form as 3.3 and so we can write the complex envelope of y as

$$\begin{aligned} \tilde{y}(t) = \sum_{k=0}^{\infty} \frac{\binom{2k+1}{k}}{2^{2k}} \int\int \ldots \int_{-\infty}^{\infty} h_{2k+1}(\tau_1 \ldots \tau_{2k+1}) \prod_{i=1}^{k} \tilde{x}^*(t-\tau_i) e^{-j\omega_0 \tau_i} \\ \times \prod_{i=k+1}^{2k+1} \tilde{x}(t-\tau_i) e^{j\omega_0 \tau_i} \, d\tau_1 \, d\tau_2 \cdots d\tau_n \end{aligned} \tag{A.7}$$

as required.

Appendix B

Multidimensional Orthogonal Functions

Here we show that the product of single dimensional orthonormal functions are multidimensional orthonormal functions. This relationship is noted in [67] but the derivation there is only for the two dimensional case.

First consider all the variables τ_m to be constant except τ_M. Then we represent the function $f(\tau_1, \tau_2, \ldots, \tau_M)$, as a sum of one dimensional orthogonal functions $l_m(\tau_M)$,

$$f(\tau_1, \tau_2, \ldots, \tau_M) = \sum_{m_M}^{\infty} a_{m_M}(\tau_1, \ldots, \tau_{M-1}) l_{m_M}(\tau_M). \tag{B.1}$$

Now $a_{m_M}(\tau_1, \ldots, \tau_{M-1})$ in B.1, is a function of $M-1$ variables and we can again consider all the variables τ_m to be constant except τ_{M-1}. Then we represent the function a_{m_M}, as a sum of one dimensional orthogonal functions $l_m(\tau_{M-1})$,

$$a_{m_M}(\tau_1, \tau_2, \ldots, \tau_{M-1}) = \sum_{m_{M-1}}^{\infty} a_{m_{M-1} m_M}(\tau_1, \ldots, \tau_{M-2}) l_{m_{M-1}}(\tau_{M-1}). \tag{B.2}$$

We repeat this process until we have a function of only one variable

$$a_{m_2 m_3 \ldots m_M}(\tau_1) = \sum_{m_M}^{\infty} a_{m_1 m_2 \ldots m_M} l_{m_1}(\tau_1). \tag{B.3}$$

Now we can recursively substitute for $a_{m_M}(\tau_1, \ldots, \tau_{M-1})$ in B.1 to get

$$\begin{aligned} f(\tau_1, \tau_2, \ldots, \tau_M) &= \sum_{m_1}^{\infty} \sum_{m_2}^{\infty} \cdots \sum_{m_M}^{\infty} a_{m_1 m_2 \ldots m_M} l_{m_1}(\tau_1) l_{m_2}(\tau_2) \ldots l_{m_M}(\tau_M) \\ &= \sum_{m_1}^{\infty} \sum_{m_2}^{\infty} \cdots \sum_{m_M}^{\infty} a_{m_1 m_2 \ldots m_M} \beta_{m_1, m_2, \ldots, m_M}(\tau_1, \tau_2 \ldots, \tau_M), \end{aligned} \tag{B.4}$$

where

$$\beta_{m_1 m_2 \ldots m_M}(\tau_1, \tau_2 \ldots, \tau_M) = l_{m_1}(\tau_1) l_{m_2}(\tau_2) \ldots l_{m_M}(\tau_M).$$

We can use a similar tactic to obtain an expression for $a_{m_1 m_2 \ldots m_M}$. Noting that since

$$\langle l_i(\tau) l_j(\tau) \rangle = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j; \end{cases}$$

then from B.1

$$a_{m_M}(\tau_1, \tau_2, \ldots, \tau_{M-1}) = \int_{-\infty}^{\infty} f(\tau_1, \tau_2, \ldots, \tau_M) l_{m_M}(\tau_M)\, d\tau_M, \tag{B.5}$$

and from B.2

$$a_{m_{M-1} m_M}(\tau_1, \tau_2, \ldots, \tau_{M-2}) = \int_{-\infty}^{\infty} a_{m_M}(\tau_1, \tau_2, \ldots, \tau_{M-1}) l_{m_{M-1}}(\tau_{M-1})\, d\tau_{M-1}, \tag{B.6}$$

and from B.3

$$a_{m_1 m_2 \ldots m_M} = \int_{-\infty}^{\infty} a_{m_2 m_3 \ldots m_M}(\tau_1) l_{m_1}(\tau_1)\, d\tau_1. \tag{B.7}$$

By recursive substitution

$$\begin{aligned} a_{m_1 m_2 \ldots m_M} &= \int\int \cdots \int_{-\infty}^{\infty} f(\tau_1, \tau_2, \ldots, \tau_M) \\ &\qquad \times l_{m_1}(\tau_1) l_{m_2}(\tau_2) \cdots l_{m_M}(\tau_M)\, d\tau_1\, d\tau_2 \ldots d\tau_M \\ &= \int\int \cdots \int_{-\infty}^{\infty} f(\tau_1, \tau_2, \ldots, \tau_M) \\ &\qquad \times \beta_{m_1 m_2 \ldots m_M}(\tau_1, \tau_2 \ldots, \tau_M)\, d\tau_1\, d\tau_2 \ldots d\tau_M. \end{aligned} \tag{B.8}$$

Equation B.4 shows that the $\beta_{m_1,m_2,\ldots,m_M}(\tau_1, \tau_2 \ldots, \tau_M)$ are a basis and equation B.8 shows that they are orthogonal.

Appendix C

Mixers with Sinusoidal Local Oscillators

Here it will be rigorously shown that a memoryless two input mixer with a sinusoidal local oscillator, in a bandpass context, is equivalent to a single input distortion followed by an ideal mixer. The distortion is, however, dependent on the amplitude of the sine wave.

Let the input be described by

$$\begin{aligned} x(t) &= \mathrm{Re}\left[\tilde{x}(t)e^{j\omega_0 t}\right] \\ &= \frac{1}{2}\left(\tilde{x}(t)e^{j\omega_0 t} + \tilde{x}^*(t)e^{-j\omega_0 t}\right), \end{aligned} \tag{C.1}$$

and the local oscillator by

$$\begin{aligned} l(t) &= A\cos(\omega t), \\ &= \frac{A}{2}\left(e^{j\omega_0 t} + e^{-j\omega_0 t}\right). \end{aligned} \tag{C.2}$$

Now let the output be defined by a polynomial

$$y(t) = \sum_{\substack{m=0 \\ n=0}}^{\infty} a_{m,n} x^m(t) l^n(t). \tag{C.3}$$

Substituting C.1 and C.2 in C.3 and discarding terms except those where $m - 2i = \pm 1$ and $n - 2k = \pm 1$ yields

$$\begin{aligned} y(t) = \frac{1}{4} \sum_{\substack{m=0 \\ n=0}}^{\infty} A^n a_{m,n} \Bigg[& \binom{m}{\frac{m-1}{2}} \binom{n}{\frac{n-1}{2}} \tilde{x}^{\frac{m+1}{2}}(t) \tilde{x}^{*\frac{m-1}{2}}(t) e^{(\omega_0+\omega_1)jt} \\ & + \binom{m}{\frac{m+1}{2}} \binom{n}{\frac{n-1}{2}} \tilde{x}^{\frac{m-1}{2}}(t) \tilde{x}^{*\frac{m+1}{2}}(t) e^{(-\omega_0+\omega_1)jt} \\ & + \binom{m}{\frac{m+1}{2}} \binom{n}{\frac{n+1}{2}} \tilde{x}^{\frac{m-1}{2}}(t) \tilde{x}^{*\frac{m+1}{2}}(t) e^{-(\omega_0+\omega_1)jt} \\ & + \binom{m}{\frac{m-1}{2}} \binom{n}{\frac{n+1}{2}} \tilde{x}^{\frac{m+1}{2}}(t) \tilde{x}^{*\frac{m-1}{2}}(t) e^{(\omega_0-\omega_1)jt} \Bigg]. \end{aligned}$$

Now $\binom{m}{i} = \binom{m}{m-i}$ which implies that $\binom{m}{\frac{m-1}{2}} = \binom{m}{\frac{m+1}{2}}$. Consequently

$$\begin{aligned} y(t) = \frac{1}{4} \sum_{\substack{m=0 \\ n=0}}^{\infty} & A^n a_{m,n} \binom{m}{\frac{m-1}{2}} \binom{n}{\frac{n-1}{2}} \\ & \times \Big[\tilde{x}^{\frac{m+1}{2}}(t) \tilde{x}^{*\frac{m-1}{2}}(t) e^{j\omega_0 t} e^{j\omega_1 t} \\ & + \tilde{x}^{\frac{m-1}{2}}(t) \tilde{x}^{*\frac{m+1}{2}}(t) e^{-j\omega_0} e^{j\omega_1 t} \\ & + \tilde{x}^{\frac{m+1}{2}}(t) \tilde{x}^{*\frac{m-1}{2}}(t) e^{j\omega_0} e^{-j\omega_1 t} \\ & + \tilde{x}^{\frac{m-1}{2}}(t) \tilde{x}^{*\frac{m+1}{2}}(t) e^{-j\omega_0 t} e^{-j\omega_1 t} \Big]. \end{aligned} \tag{C.4}$$

Equation C.4 can now be factored

$$\begin{aligned} y(t) = & \left(\frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2} \right) \sum_{\substack{m=0 \\ n=0}}^{\infty} A^n a_{m,n} \binom{m}{\frac{m-1}{2}} \binom{n}{\frac{n-1}{2}} \\ & \times [\tilde{x}(t)\tilde{x}^*(t)]^{\frac{m-1}{2}} \left[\frac{\tilde{x}(t)e^{j\omega_0 t} + \tilde{x}^*(t)e^{-j\omega_0 t}}{2} \right] \\ = & \, l(t) \sum_{\substack{m=0 \\ n=0}}^{\infty} A^n a_{m,n} \binom{m}{\frac{m-1}{2}} \binom{n}{\frac{n-1}{2}} |\tilde{x}(t)|^m x(t), \end{aligned} \tag{C.5}$$

as required. Equation C.5 shows the output in the form of the product of the local oscillator and the distorted signal. Note, however, that the nature of the distortion is dependent on the amplitude of the local oscillator A.

Appendix D

Recursive Least Squares Algorithms

In this appendix we will extend the theory of Recursive Least Squares (RLS) adaption algorithms so that they can be applied to nonlinear filters. Both the recursive least squares lattice filter and the recursive least squares transversal filters are derived in a common approach. It should be noted that these are true least squares algorithms as opposed to algorithms known as "least mean squares" algorithms.

A linear transversal filter can be divided into two sections. The first part is the tapped delay line and the second part performs a linear combination of those taps. A nonlinear filter can be defined where the inputs to the linear combiner are the terms of a nonlinear polynomial of the input samples. The cost function for selecting the weights in the nonlinear combiner is independent of how the inputs to the linear combiner where obtained. However, the linear RLS algorithms achieve computational efficiency by making use of *a priori* information about the nature of the inputs to the linear combiner. In particular, if the filter is defined by $y(n) = \boldsymbol{c}^*\boldsymbol{x}(n)$, then the linear RLS algorithms make use of the fact that the elements of $\boldsymbol{x}(n+1)$ are simply those of $\boldsymbol{x}(n)$ shifted

along, with the last element removed and a new first element.

To extend these algorithms to the nonlinear case, we need to similarly take advantage of any *a priori* information on the nature of the elements of $\boldsymbol{x}$. For example, the terms of a nonlinear filter might be arranged as column vectors: where x_i is the input $x(t-iT)$. At the next sampling instant, the values of all the

delay $\longrightarrow$

ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6
x_1	x_2	x_3	x_4	x_5	x_6
$x_1x_1x_1$	$x_2x_2x_2$	$x_3x_3x_3$			
$x_1x_1x_2$	$x_2x_2x_3$				
$x_1x_2x_2$	$x_2x_3x_3$				
$x_1x_2x_3$					

Figure D.1: Organization of nonlinear terms into unequal column vectors.

columns are shifted right one position. Mueller extended the RLS algorithms to fractionally spaced equalizers [51]. In his treatment, $\boldsymbol{x}$ is partitioned into equal length subvectors. The nonlinear case, when suitably arranged, is almost of this form if the columns in figure D.1 are regarded as the subvectors, except that the subvectors are not of equal length.

This formulation is useful for nonlinear systems but it is also more generally useful. For example, consider a linear filter modelling a multipath communication link. Suppose each of the modes has different dispersion. Such a system might be modelled acceptably with a minimum number of taps by arranging the inputs as in figure D.2.

delay $\longrightarrow$

ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6
x_1	x_2	x_3	x_4	x_5	x_6
x_{120}	x_{121}	x_{122}			

Figure D.2: Organization of linear terms into unequal column vectors.

D.1 The General Least Squares Problem

Let us define the vector of inputs to the linear combiner by

$$l\boldsymbol{x}_m(n)^* = \left(\boldsymbol{\xi}_1(n)^* \,\middle|\, \boldsymbol{\xi}_2(n)^* \,\middle|\, \cdots \,\middle|\, \boldsymbol{\xi}_m(n)^* \right). \tag{D.1}$$

For a conventional multiple input linear transversal filter $\boldsymbol{\xi}_1(j)$ is the vector of inputs at the jth time interval and $\boldsymbol{\xi}_i(j) = \boldsymbol{\xi}_{i-1}(j-1)$. In the example D.1, $\boldsymbol{\xi}_i(1)$ are the column vectors. For this general case where the $\boldsymbol{\xi}_i$ are not all the same length,

$$\boldsymbol{\xi}_i(j) = \boldsymbol{T}_{i-1}\boldsymbol{\xi}_{i-1}(j-1) \quad \text{for } i > 1$$

and

$$\boldsymbol{T}_i = \boldsymbol{I} \quad \text{for } i \leq 1. \tag{D.2}$$

The matrix $\boldsymbol{T}_{i-1}$ defined in equation D.2 is such that $\boldsymbol{\xi}_i(j)$ is the same as $\boldsymbol{\xi}_{i-1}(j-1)$ but with some elements removed and possibly some elements rearranged. We will derive the properties of $\boldsymbol{T}_i$ more precisely in section D.1.1. The method followed in this appendix is largely the same that of Mueller [51], however, the presence of $\boldsymbol{T}$ in equation D.2 complicates the derivation considerably. If equation D.2 is applied recursively

$$\begin{aligned}\boldsymbol{\xi}_i(j) &= \boldsymbol{T}_{i-1}\boldsymbol{T}_{i-2}\cdots\boldsymbol{T}_1\boldsymbol{\xi}_1(j-i+1) \\ &= \grave{\boldsymbol{T}}_{i-1}\boldsymbol{\xi}_1(j-i+1),\end{aligned} \tag{D.3}$$

where

$$\grave{\boldsymbol{T}}_{i-1} = \boldsymbol{T}_{i-1}\boldsymbol{T}_{i-2}\cdots\boldsymbol{T}_1. \tag{D.4}$$

We need to also define a row reduced version of $\boldsymbol{x}_m(j)$

$$\begin{aligned}\boldsymbol{x}_m^q(n) &= \hat{\boldsymbol{T}}_{q,m+q-1}\hat{\boldsymbol{T}}_{q-1,m+q-2}\cdots\hat{\boldsymbol{T}}_{1,m}\boldsymbol{x}_m(n) \\ &= \hat{\boldsymbol{T}}_{q,m+q-1}\boldsymbol{x}_m^{q-1}(n)\end{aligned} \tag{D.5}$$

and

$$\boldsymbol{x}_m^0(n) = \boldsymbol{x}_m(n), \tag{D.6}$$

where

$$\hat{\boldsymbol{T}}_{i,j} = \left(\begin{array}{c|c|c|c} \boldsymbol{T}_i & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{T}_{i+1} & \cdots & \mathbf{0} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{T}_j \end{array}\right). \tag{D.7}$$

Where only one subscript is used $\hat{\boldsymbol{T}}_j = \hat{\boldsymbol{T}}_{1,j}$. It follows from equations D.7, D.5 and D.3 that

$$\begin{aligned} \boldsymbol{x}_{m-q}^q(n) &= \boldsymbol{x}_{q+1,m}(n+q) \\ &= \left(\boldsymbol{\xi}_{q+1}(n+q)^* \,\middle|\, \boldsymbol{\xi}_{q+2}(n+q)^* \,\middle|\, \cdots \,\middle|\, \boldsymbol{\xi}_m(n+q)^* \right). \end{aligned} \tag{D.8}$$

That is to say $\boldsymbol{x}_{m-q}^q(n)$ denotes a vector made up of the last $m-q$ subvectors of $\boldsymbol{x}_m(n+q)$.

To keep track of the dimensions of matrices in the following derivation, let us define

$$p_m = \text{order}(\boldsymbol{\xi}_m) \tag{D.9}$$

and therefore

$$\begin{aligned} P_m &= \text{order}(\boldsymbol{x}_m) \\ &= \sum_{i=1}^{m} p_i. \end{aligned} \tag{D.10}$$

The dimensions of the major variables in this section are summarized in table D.1.

The processes of estimation, forward prediction and backward prediction are very similar and we will use a notation which allows the derivation of these three cases in parallel.

The linear combiner is defined by

$$\boldsymbol{y}_m^{s,q}(n) = \boldsymbol{c}_m^{s,q}(n-1)^* \boldsymbol{x}_m^q(n), \tag{D.11}$$

where $s \in \{e, b, f\}$ denotes the equalizer, backward predictor or forward predictor cases. The dimensions of $\boldsymbol{c}^s = \boldsymbol{c}^{s,0}$ and $\boldsymbol{y}^s = \boldsymbol{y}^{s,0}$ are in table D.1. The

Description	Symbol	Dimensions
Linear combiner input	$\boldsymbol{x}_m$	$P_m \times 1$
	$\boldsymbol{\xi}_m$	$p_m \times 1$
Correlation matrix	$\boldsymbol{A}_m$	$P_m \times P_m$
Transition matrix	$\boldsymbol{T}_m$	$p_{m+1} \times p_m$
	$\grave{\boldsymbol{T}}_m$	$p_{m+1} \times p_1$
Kalman Gain	$\boldsymbol{g}_m$	$P_m \times 1$
	$\boldsymbol{L}_m$	$P_m \times P_m$
Forward Prediction		
Prediction	$\boldsymbol{y}_m^f$	$p_1 \times 1$
Desired output	$\boldsymbol{d}^f$	$p_1 \times 1$
Coefficients	$\boldsymbol{c}_m^f$	$P_m \times p_1$
Prediction error	$\boldsymbol{e}_m^f$	$p_1 \times 1$
Error residual	$\boldsymbol{\epsilon}_m^f$	$p_1 \times p_1$
	$\boldsymbol{v}_m^f$	$P_m \times p_1$
	$\boldsymbol{E}_m^f$	$p_1 \times p_1$
Backward Prediction		
Prediction	$\boldsymbol{y}_m^b$	$p_1 \times 1$
Desired output	$\boldsymbol{d}^b$	$p_1 \times 1$
Coefficients	$\boldsymbol{c}_m^b$	$P_m \times p_1$
Prediction error	$\boldsymbol{e}_m^b$	$p_1 \times 1$
Error residual	$\boldsymbol{\epsilon}_m^b$	$p_1 \times p_1$
	$\boldsymbol{v}_m^b$	$P_m \times p_1$
	$\boldsymbol{E}_m^b$	$p_m \times p_1$
Equalizer		
Equalizer output	$\boldsymbol{y}_m^e$	1×1
Desired output	$\boldsymbol{d}^e$	1×1
Coefficients	$\boldsymbol{c}_m^e$	$P_m \times 1$
Prediction error	$\boldsymbol{e}_m^e$	1×1
Error residual	$\boldsymbol{\epsilon}_m^e$	1×1
	$\boldsymbol{v}_m^e$	$P_m \times 1$
	$\boldsymbol{E}_m^e$	1×1

Table D.1: Dimensions of variables.

q superscript denotes a member of a family of filters. We are ultimately interested in the filters corresponding to $q = 0$, but we need the partial filters corresponding to $q > 0$ in the recursive algorithms. Where q is omitted it is implied that q is zero.

The desired output of the filters are given by

$$\boldsymbol{d}_m^s(n) = \begin{cases} d(n), & \text{if } s = e; \\ \boldsymbol{\xi}_1(n+1), & \text{if } s = f; \\ \boldsymbol{\xi}_1(n-m), & \text{if } s = b. \end{cases} \tag{D.12}$$

Defining the error as

$$\boldsymbol{e}_m^{s,q}(n) = \boldsymbol{d}_m^s(n) - \boldsymbol{y}_m^{s,q}(n), \tag{D.13}$$

and substituting equation D.11, we get

$$\boldsymbol{e}_m^{s,q}(n) = \boldsymbol{d}_m^s(n) - \boldsymbol{c}_m^{s,q}(n-1)^* \boldsymbol{x}_m^q(n). \tag{D.14}$$

Note that the desired output, $\boldsymbol{d}^s$, is independent of q.

Equation D.14 defines the error before the coefficients have been updated. *After* the coefficients have been updated, we can define a new error

$$\tilde{\boldsymbol{e}}_m^{s,q}(n) = \boldsymbol{d}_m^s(n) - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(n) \tag{D.15}$$

and a variant form,

$$\tilde{\boldsymbol{e}}_m^{s,q}(n,k) = \boldsymbol{d}_m^s(k) - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(k). \tag{D.16}$$

We want to choose the value of $\boldsymbol{c}_m^{s,q}(n)$ so as to minimize the mean squared length of $\tilde{\boldsymbol{e}}_m^{s,q}(n,k), \quad 0 \le k \le n$. If $e_1, e_2, e_3 \ldots e_p$ are the elements of $\tilde{\boldsymbol{e}}_m^{s,q}(n,k)$, then the length squared of $\tilde{\boldsymbol{e}}_m^{s,q}(n,k)$ is $e_1 e_1^* + e_2 e_2^* + e_3 e_3^* \cdots e_p e_p^*$. Now,

$$\tilde{\boldsymbol{e}}_m^{s,q}(n,k)\tilde{\boldsymbol{e}}_m^{s,q}(n,k)^* = \begin{pmatrix} e_1 e_1^* & e_1 e_2^* & e_1 e_3^* & \cdots \\ e_2 e_1^* & e_2 e_2^* & e_2 e_3^* & \cdots \\ e_3 e_1^* & e_3 e_2^* & e_3 e_3^* & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Thus, the length squared of $\tilde{\boldsymbol{e}}_m^{s,q}(n,k)$ is $\mathrm{tr}(\tilde{\boldsymbol{e}}_m^{s,q}(n,k)\tilde{\boldsymbol{e}}_m^{s,q}(n,k)^*)$.

Let us define the error residual

$$\epsilon_m(n)^{s,q} = \sum_{k=0}^{n} \lambda^{n-k} \tilde{\boldsymbol{e}}_m^{s,q}(n,k)\tilde{\boldsymbol{e}}_m^{s,q}(n,k)^*. \tag{D.17}$$

The cost function which we wish to minimize is $\mathrm{tr}(\epsilon^{s,q})_m(n)$. The exponential factor λ is chosen to weight recent errors more heavily than past errors. If the system is stationary, λ can be set to 1. Note also that

$$\begin{aligned}\mathrm{tr}(\tilde{\boldsymbol{e}}_m^{s,q}(n)\tilde{\boldsymbol{e}}_m^{s,q}(n)^*) &= e_1 e_1^* + e_2 e_2^* + e_3 e_3^* \cdots \\ &= \tilde{\boldsymbol{e}}_m^{s,q}(n)^* \tilde{\boldsymbol{e}}_m^{s,q}(n).\end{aligned} \tag{D.18}$$

The value we wish to minimize is a scalar and it might seem circuitous to first define a matrix error residual and then define the cost as the trace of that matrix when we could have simply used the inner product of equation D.18. However, it turns out that there is insufficient information in the scalar cost function about the prior state of the system to allow a recursive algorithm.

If the derivative of a matrix is zero, the derivative of each of its elements must be zero and the derivative of the trace must be zero. Therefore the value of $\boldsymbol{c}_m^{s,q}(n)$ which minimizes $\mathrm{tr}(\boldsymbol{\epsilon}_m^{s,q}(n))$ can be found by found by differentiating equation D.17 with respect to each of the elements of $\boldsymbol{c}_m^{s,q}(n)$ and equating the derivatives to zero. For notational convenience we drop the m subscript and the s and q superscripts. Differentiating equation D.17 with respect to c_{ij},

$$\begin{aligned}\frac{\partial \boldsymbol{\epsilon}(n)}{\partial c_{i,j}(n)} &= \sum_{k=0}^{n} \lambda^{n-k} \frac{\partial}{\partial c_{i,j}(n)} [\tilde{\boldsymbol{e}}(n,k)\tilde{\boldsymbol{e}}(n,k)^*] \\ &= \sum_{k=0}^{n} \lambda^{n-k} \frac{\partial \tilde{\boldsymbol{e}}(n,k)}{\partial c_{i,j}(n)} \tilde{\boldsymbol{e}}(n,k)^* + \left[\frac{\partial \tilde{\boldsymbol{e}}(n,k)}{\partial c_{i,j}(n)} \tilde{\boldsymbol{e}}(n,k)^*\right]^*.\end{aligned} \tag{D.19}$$

Differentiate equation D.16,

$$\frac{\partial \tilde{\boldsymbol{e}}(n,k)}{\partial c_{i,j}(n)} = \frac{\partial \boldsymbol{c}(n)^*}{\partial c_{i,j}(n)} \boldsymbol{x}(k),$$

and substitute in equation D.19 to get

$$\frac{\partial \boldsymbol{\epsilon}(n)}{\partial c_{i,j}(n)} = \sum_{k=0}^{n} \lambda^{n-k} \left(\frac{\partial \boldsymbol{c}(n)^*}{\partial c_{i,j}(n)} \boldsymbol{x}(k)\tilde{\boldsymbol{e}}(n,k)^* + \left[\frac{\partial \boldsymbol{c}(n)^*}{\partial c_{i,j}(n)} \boldsymbol{x}(k)\tilde{\boldsymbol{e}}(n,k)^*\right]^* \right) \tag{D.20}$$

$$\begin{aligned}&= \frac{\partial \boldsymbol{c}(n)^*}{\partial c_{i,j}(n)} \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}(k)\tilde{\boldsymbol{e}}(n,k)^* + \left[\frac{\partial \boldsymbol{c}(n)^*}{\partial c_{i,j}(n)} \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}(k)\tilde{\boldsymbol{e}}(n,k)^*\right]^* \\ &= 0.\end{aligned} \tag{D.21}$$

Equation D.21 is satisfied if

$$\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}(k)\tilde{\boldsymbol{e}}(n,k)^* = 0, \tag{D.22}$$

and substituting equation D.16,

$$\sum_{k=0}^{n} \lambda^{n-k} \left[\boldsymbol{x}(k)\boldsymbol{x}(k)^*\boldsymbol{c}(n) - \boldsymbol{x}(k)\boldsymbol{d}(k)^*\right] = 0.$$

Therefore, returning to the complete notation,

$$\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}_m^q(k)\boldsymbol{x}_m^q(k)^*\boldsymbol{c}_m^{s,q}(n) = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}_m^q(k)\boldsymbol{d}_m^s(k)^*,$$

which is of the form

$$\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) = \boldsymbol{v}_m^{s,q}(n), \tag{D.23}$$

where

$$\boldsymbol{A}_m^q(n) = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}_m^q(k)\boldsymbol{x}_m^q(k)^* = \lambda \boldsymbol{A}_m^q(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^* \tag{D.24}$$

and

$$\boldsymbol{v}_m^{s,q}(n) = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{x}_m^q(k)\boldsymbol{d}_m^s(k)^* = \lambda \boldsymbol{v}_m^{s,q}(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^s(n)^*. \tag{D.25}$$

Equation D.23 (together with the definitions D.24 and D.25) is known as the Wiener-Hopf solution to the problem of minimizing D.17 [77].

D.1.1 Properties of the Transition Matrices

To proceed further, we need to establish some properties of the $\boldsymbol{T}_i$, $\hat{\boldsymbol{T}}_i$ and $\grave{\boldsymbol{T}}_i$ Matrices. Recall that $\boldsymbol{\xi}_{i+1}(n+1) = \boldsymbol{T}_i\boldsymbol{\xi}_i(n)$. Pre-multiplication by $\boldsymbol{T}_i$ selects certain rows of $\boldsymbol{\xi}_i(n)$ and transfers them to certain rows of $\boldsymbol{\xi}_{i+1}(n+1)$. This is achieved if $\boldsymbol{T}_i$ has the following properties:

1. $\boldsymbol{T}_i$ is a rectangular $p_{i+1} \times p_i$ matrix with $p_{i+1} \leq p_i$.
2. All the elements of $\boldsymbol{T}_i$ are zero except for at most one element of each column which is unity.
3. Exactly one element of each row is unity.

Since $\hat{\boldsymbol{T}}_m$ is a block diagonal matrix diag($\boldsymbol{T}_i$) it follows that $\hat{\boldsymbol{T}}$ also satisfies conditions 1–3 except the dimensions are $P_{i+1} \times P_i$, with $P_{i+1} \leq P_i$.

We will use the conditions 1–3 to establish some algebraic properties. Let S be the set of column numbers such that

$$\hat{\boldsymbol{T}}_m = \left(\hat{\boldsymbol{t}}_1 \,\middle|\, \hat{\boldsymbol{t}}_2 \,\middle|\, \cdots \,\middle|\, \hat{\boldsymbol{t}}_{P_m} \right), \tag{D.26}$$

$$\hat{\boldsymbol{t}}_i = 0, \quad i \in S,$$

and

$$\hat{\boldsymbol{T}}_m = \left(\begin{array}{c} \hat{\boldsymbol{r}}_1 \\ \hline \hat{\boldsymbol{r}}_2 \\ \hline \vdots \\ \hline \hat{\boldsymbol{r}}_{P_{m+1}} \end{array} \right). \tag{D.27}$$

Conditions 2 and 3 imply that rows of $\hat{\boldsymbol{T}}_m$ are orthogonal to each other and the non zero columns are orthogonal to each other

$$\hat{\boldsymbol{t}}_i^* \hat{\boldsymbol{t}}_j = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j \text{ and } i \notin S; \end{cases} \tag{D.28}$$

and

$$\hat{\boldsymbol{r}}_i \hat{\boldsymbol{r}}_j^* = \begin{cases} 0, & \text{if } i \neq j; \\ 1, & \text{if } i = j. \end{cases} \tag{D.29}$$

It follows then, that

$$\hat{\boldsymbol{T}}_m^* \hat{\boldsymbol{T}}_m = \underset{1 \leq i \leq P_m}{\text{diag}} \, (\hat{\boldsymbol{t}}_i^* \hat{\boldsymbol{t}}_i). \tag{D.30}$$

Also

$$\begin{aligned} \hat{\boldsymbol{T}}_m \hat{\boldsymbol{T}}_m^* &= \underset{1 \leq i \leq P_{m+1}}{\text{diag}} \, (\hat{\boldsymbol{r}}_i \hat{\boldsymbol{r}}_i) \\ &= \boldsymbol{I}. \end{aligned} \tag{D.31}$$

Thus $\hat{\boldsymbol{T}}_m^*$ is the postinverse of $\hat{\boldsymbol{T}}_m$.

D.2 Time Update Recursions

We have already derived recursive definitions for $\boldsymbol{A}_m^q(n)$ and $\boldsymbol{v}_m^{s,q}$ (equations D.24 and D.25). We now proceed to derive a number of other recursive relationships.

We find an alternative form for the error residual by substituting D.16 in D.17

$$\begin{aligned}\epsilon_m(n)^{s,q} &= \sum_{k=0}^{n} \lambda^{n-k} \left[\boldsymbol{d}_m^s(k) - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(k)\right] \left[\boldsymbol{d}_m^s(k) - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(k)\right]^* \\ &= \sum_{k=0}^{n} \lambda^{n-k} \Big[\boldsymbol{d}_m^s(k)\boldsymbol{d}_m^s(k)^* - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(k)\boldsymbol{d}_m^s(k)^* \\ &\qquad - \boldsymbol{d}_m^s(k)\boldsymbol{x}_m^q(k)^* \boldsymbol{c}_m^{s,q}(n) + \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{x}_m^q(k)\boldsymbol{x}_m^q(k)^* \boldsymbol{c}_m^{s,q}(n)\Big],\end{aligned} \tag{D.32}$$

and substituting D.24 and D.25,

$$\begin{aligned}\epsilon_m(n)^{s,q} = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{d}_m^s(k)\boldsymbol{d}_m^s(k)^* - \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{v}_m^{s,q}(n) \\ - \boldsymbol{v}_m^{s,q}(n)^* \boldsymbol{c}_m^{s,q}(n) + \boldsymbol{c}_m^{s,q}(n)^* \boldsymbol{A}_m^q \boldsymbol{c}_m^{s,q}(n),\end{aligned} \tag{D.33}$$

and substituting equation D.23 in the last term of equation D.33 and simplifying,

$$\epsilon_m(n)^{s,q} = \boldsymbol{E}_m^s(n) - \boldsymbol{v}_m^{s,q}(n)^* \boldsymbol{c}_m^{s,q}(n), \tag{D.34}$$

where

$$\boldsymbol{E}_m^s(n) = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{d}_m^s(k)\boldsymbol{d}_m^s(k)^* = \lambda \boldsymbol{E}_m^s(n-1) + \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^s(n)^*. \tag{D.35}$$

Now, substitute D.25(recursive form) into D.23 to give

$$\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) = \lambda \boldsymbol{v}_m^{s,q}(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^s(n)^*, \tag{D.36}$$

use equation D.23 to replace $\boldsymbol{v}_m^{s,q}(n-1)$ in equation D.36 to get

$$\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) = \lambda \boldsymbol{A}_m^q(n-1)\boldsymbol{c}_m^{s,q}(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^s(n)^*,$$

use D.24 to get

$$\begin{aligned}\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) &= \left[\boldsymbol{A}_m^q(n) - \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^*\right] \boldsymbol{c}_m^{s,q}(n-1) \\ &\qquad + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^s(n)^*, \\ &= \boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n-1) - \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^* \boldsymbol{c}_m^{s,q}(n-1) \\ &\qquad + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^s(n)^*,\end{aligned}$$

use D.13 to get

$$\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) = \boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n-1) - \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{s,q}(n-1) + \boldsymbol{x}_m^q(n)[\boldsymbol{e}_m^{s,q}(n) + \boldsymbol{y}_m^{s,q}(n)]^*,$$

use D.11 to get

$$\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n) = \boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{s,q}(n-1) - \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{s,q}(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{e}_m^{s,q}(n)^* + \boldsymbol{x}_m^q(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{s,q}(n-1),$$

and finally, simplify and premultiply by $\boldsymbol{A}_m^{q\,-1}$ to get

$$\boldsymbol{c}_m^{s,q}(n) = \boldsymbol{c}_m^{s,q}(n-1) + \boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q(n)\boldsymbol{e}_m^{s,q}(n)^*. \tag{D.37}$$

Equation D.37 is central to all the recursive least squares transversal algorithms. Unfortunately inverting $\boldsymbol{A}_m^q$ is computationally expensive. We note that $\boldsymbol{A}_m^{q\,-1}\boldsymbol{x}_m^q$ is a vector of the same order as $\boldsymbol{x}_m^q$. Let us define

$$\boldsymbol{g}_m^q(n) = \boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q(n). \tag{D.38}$$

The vector $\boldsymbol{g}_m^q(n)$ is known as the Kalman gain. The Kalman algorithm avoids inverting $\boldsymbol{A}_m^q(n)$ at each iteration by use of the matrix inversion lemma, but it still requires $O(P_m^2)$ operations as it manipulates $P_m \times P_m$ matrices. The fast Kalman algorithm algorithm is a method of updating $\boldsymbol{g}_m^q(n)$ without anywhere manipulating matrices larger than $P_m \times p_i, \quad 1 \leq i \leq m$. In the linear case $p_i = 1$, and in cases where the fast Kalman algorithm is of interest, $p_i \ll P_m$.

Note that $\boldsymbol{A}_m^{q\,-1^*} = \boldsymbol{A}_m^{q\,-1}$ since $\boldsymbol{A}_m^q$, and consequently $\boldsymbol{A}_m^{q\,-1}$, is Hermitian. We now take the conjugate-transpose of D.37,

$$\boldsymbol{c}_m^{s,q}(n)^* = \boldsymbol{c}_m^{s,q}(n-1)^* + \boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1},$$

post multiply by $\boldsymbol{v}_m^{t,q}(n), \quad t \in \{e, b, f\}$, to get

$$\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) = \boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n) + \boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1}\boldsymbol{v}_m^{t,q}(n). \tag{D.39}$$

Premultiply equation D.23 by $\boldsymbol{A}_m^q(n)^{-1}$ and substitute in the last term of equation D.39 and substitute equation D.25 (recursive form) into equation D.39,

$$\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) = \boldsymbol{c}_m^{s,q}(n-1)^*\left[\lambda\boldsymbol{v}_m^{t,q}(n-1) + \boldsymbol{x}_m^q(n)\boldsymbol{d}_m^t(n)^*\right] + \boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n),$$

add and subtract $\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*$,

$$\begin{aligned}\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) &= \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1)\\ &\quad+\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{x}_m^q(n)\boldsymbol{d}_m^t(n)^*+\boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n)\\ &\quad+\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*-\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*,\\ &= \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1)\\ &\quad+\left[\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{x}_m^q(n)-\boldsymbol{d}_m^s(n)\right]\boldsymbol{d}_m^t(n)^*\\ &\quad+\boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n)+\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*,\end{aligned} \tag{D.40}$$

and substitute equation D.14 to get

$$\begin{aligned}\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) &= \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1)-\boldsymbol{e}_m^{s,q}(n)\boldsymbol{d}_m^t(n)^*\\ &\quad+\boldsymbol{e}_m^{s,q}(n)\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n)+\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*,\\ &= \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1)\\ &\quad+\boldsymbol{e}_m^{s,q}(n)\left[\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n)-\boldsymbol{d}_m^t(n)\right]^*+\boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*.\end{aligned} \tag{D.41}$$

Put $s=t$ in equation D.15 and take the conjugate-transpose to get,

$$\tilde{\boldsymbol{e}}_m^{t,q}(n)^*=\boldsymbol{d}_m^t(n)^*-\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n), \tag{D.42}$$

substitute D.37 to get,

$$\tilde{\boldsymbol{e}}_m^{t,q}(n)^*=\boldsymbol{d}_m^t(n)^*-\boldsymbol{x}_m^q(n)^*\boldsymbol{c}_m^{t,q}(n-1)-\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q\boldsymbol{e}_m^{t,q}(n)^*,$$

substitute equation D.14 for the first two terms

$$\begin{aligned}\tilde{\boldsymbol{e}}_m^{t,q}(n)^* &= \boldsymbol{e}_m^{t,q}(n)^*-\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q\boldsymbol{e}_m^{t,q}(n)^*\\ &= \left[1-\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q\right]\boldsymbol{e}_m^{t,q}(n)^*\\ &= \left[1-\gamma_m^q(n)\right]\boldsymbol{e}_m^{t,q}(n)^*,\end{aligned} \tag{D.43}$$

where

$$\gamma_m^q(n)=\boldsymbol{x}_m^q(n)^*\boldsymbol{A}_m^q(n)^{-1}\boldsymbol{x}_m^q. \tag{D.44}$$

Substitute equation D.42 for the bracketed term of D.41 to get

$$\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) = \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1) - \boldsymbol{e}_m^{s,q}(n)\tilde{\boldsymbol{e}}_m^{t,q}(n)^* + \boldsymbol{d}_m^{s,q}(n)\boldsymbol{d}_m^t(n)^*,$$

and substitute D.43 to get,

$$\boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{t,q}(n) = \lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{t,q}(n-1) - [1-\gamma_m^q(n)]\,\boldsymbol{e}_m^{s,q}(n)\boldsymbol{e}_m^{t,q}(n)^* + \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^t(n)^*. \quad \text{(D.45)}$$

We are now in a position to derive a recursive form for the error residual. Substituting D.35 into equation D.34 results in

$$\boldsymbol{\epsilon}_m(n)^{s,q} = \lambda\boldsymbol{E}_m^s(n-1) + \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^s(n)^* - \boldsymbol{v}_m^{s,q}(n)^*\boldsymbol{c}_m^{s,q}(n),$$

and using D.34 again with n replaced by $n-1$,

$$\begin{aligned}\boldsymbol{\epsilon}_m^{s,q}(n) &= \lambda\left[\boldsymbol{\epsilon}_m^{s,q}(n-1) + \boldsymbol{v}_m^{s,q}(n-1)^*\boldsymbol{c}_m^{s,q}(n-1)\right] \\ &\quad + \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^s(n)^* - \boldsymbol{v}_m^{s,q}(n)^*\boldsymbol{c}_m^{s,q}(n), \\ &= \lambda\boldsymbol{\epsilon}_m^{s,q}(n-1) + \lambda\boldsymbol{v}_m^{s,q}(n-1)^*\boldsymbol{c}_m^{s,q}(n-1) \\ &\quad + \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^s(n)^* - \boldsymbol{v}_m^{s,q}(n)^*\boldsymbol{c}_m^{s,q}(n). \end{aligned} \quad \text{(D.46)}$$

Now rearrange equation D.45,

$$\lambda\boldsymbol{c}_m^{s,q}(n-1)^*\boldsymbol{v}_m^{s,q}(n-1) = \boldsymbol{c}_m^{s,q}(n)^*\boldsymbol{v}_m^{s,q}(n) + [1-\gamma_m^q(n)]\,\boldsymbol{e}_m^{s,q}(n)\boldsymbol{e}_m^{s,q}(n)^* - \boldsymbol{d}_m^s(n)\boldsymbol{d}_m^s(n)^*,$$

and substitute in D.46 to get

$$\boldsymbol{\epsilon}_m^{s,q}(n) = \lambda\boldsymbol{\epsilon}_m^{s,q}(n-1) + [1-\gamma_m^q(n)]\,\boldsymbol{e}_m^{s,q}(n)\boldsymbol{e}_m^{s,q}(n)^*. \quad \text{(D.47)}$$

D.3 The Nonlinear Lattice Filter

In this section we first show that a suitable lattice or ladder structure is equivalent to the transversal structure discussed so far. We then show how a lattice

filter can perform orthogonalization of $\boldsymbol{x}$ and we define some new variables necessary for the recursive lattice algorithm.

First consider a quite general nonlinear transversal filter

$$\begin{aligned}\boldsymbol{\chi}_m^f(n) = \boldsymbol{a}_0^*\boldsymbol{\xi}_1(n) + \boldsymbol{a}_1^*\grave{\boldsymbol{T}}_0\boldsymbol{\xi}_1(n-1) + \boldsymbol{a}_2^*\grave{\boldsymbol{T}}_1\boldsymbol{\xi}_1(n-2) \\ + \cdots + \boldsymbol{a}_i^*\grave{\boldsymbol{T}}_{i-1}\boldsymbol{\xi}_1(n-i) + \cdots + \boldsymbol{a}_m^*\grave{\boldsymbol{T}}_{m-1}\boldsymbol{\xi}_1(n-m). \quad \text{(D.48)}\end{aligned}$$

There is a corresponding filter whose impulse response has been reversed in time and conjugate-transposed. Disregard for the moment the physical significance of such a filter which is defined by

$$\begin{aligned}\boldsymbol{\chi}_m^b(n) = \grave{\boldsymbol{T}}_{m-1}^*\boldsymbol{a}_m\boldsymbol{\xi}_1(n) + \grave{\boldsymbol{T}}_{m-2}^*\boldsymbol{a}_{m-1}\boldsymbol{\xi}_1(n-1) \\ + \grave{\boldsymbol{T}}_{m-3}^*\boldsymbol{a}_{m-2}\boldsymbol{\xi}_1(n-2) \\ + \cdots + \grave{\boldsymbol{T}}_{m-i-1}^*\boldsymbol{a}_{m-i}\boldsymbol{\xi}_1(n-i) \\ + \cdots + \boldsymbol{a}_0\boldsymbol{\xi}_1(n-m). \quad \text{(D.49)}\end{aligned}$$

Now define reduced order versions of equations D.48 and D.49,

$$\begin{aligned}\boldsymbol{\chi}_{m-1}^f(n) = \boldsymbol{b}_0^*\boldsymbol{\xi}_1(n) + \boldsymbol{b}_1^*\grave{\boldsymbol{T}}_0\boldsymbol{\xi}_1(n-1) + \boldsymbol{b}_2^*\grave{\boldsymbol{T}}_1\boldsymbol{\xi}_1(n-2) \\ + \cdots + \boldsymbol{b}_i^*\grave{\boldsymbol{T}}_{i-1}\boldsymbol{\xi}_1(n-i) \\ + \cdots + \boldsymbol{b}_{m-1}^*\grave{\boldsymbol{T}}_{m-2}\boldsymbol{\xi}_1(n-m+1), \quad \text{(D.50)}\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\chi}_{m-1}^b(n) = \grave{\boldsymbol{T}}_{m-2}^*\boldsymbol{b}_{m-1}\boldsymbol{\xi}_1(n) + \grave{\boldsymbol{T}}_{m-3}^*\boldsymbol{b}_{m-2}\boldsymbol{\xi}_1(n-1) \\ + \grave{\boldsymbol{T}}_{m-4}^*\boldsymbol{b}_{m-3}\boldsymbol{\xi}_1(n-2) \\ + \cdots + \grave{\boldsymbol{T}}_{m-i}^*\boldsymbol{b}_{m-i+1}\boldsymbol{\xi}_1(n-i+1) \\ + \cdots + \boldsymbol{b}_0\boldsymbol{\xi}_1(n-m+1),\end{aligned}$$

and replacing n by $n-1$,

$$\begin{aligned}\boldsymbol{\chi}_{m-1}^b(n-1) = \grave{\boldsymbol{T}}_{m-2}^*\boldsymbol{b}_{m-1}\boldsymbol{\xi}_1(n-1) + \grave{\boldsymbol{T}}_{m-3}^*\boldsymbol{b}_{m-2}\boldsymbol{\xi}_1(n-2) \\ + \grave{\boldsymbol{T}}_{m-4}^*\boldsymbol{b}_{m-3}\boldsymbol{\xi}_1(n-3) \\ + \cdots + \grave{\boldsymbol{T}}_{m-i}^*\boldsymbol{b}_{m-i+1}\boldsymbol{\xi}_1(n-i) \\ + \cdots + \boldsymbol{b}_0\boldsymbol{\xi}_1(n-m). \quad \text{(D.51)}\end{aligned}$$

Now take equation D.51 multiplied by $\boldsymbol{K}_m^f$ and add to equation D.50

$$\begin{aligned}\boldsymbol{\chi}_{m-1}^f(n) + \boldsymbol{K}_{m-1}^f \boldsymbol{\chi}_m^b(n-1) \\ &= \boldsymbol{b}_0^* \boldsymbol{\xi}_1(n) + (\boldsymbol{b}_1^* \dot{\boldsymbol{T}}_0 + \boldsymbol{K}_m^f \dot{\boldsymbol{T}}_{m-2}^* \boldsymbol{b}_{m-1}) \boldsymbol{\xi}_1(n-1) \\ &\quad + (\boldsymbol{b}_2^* \dot{\boldsymbol{T}}_1 + \boldsymbol{K}_m^f \dot{\boldsymbol{T}}_{m-3}^* \boldsymbol{b}_{m-2}) \boldsymbol{\xi}_1(n-2) \\ &\quad + \cdots + (\boldsymbol{b}_i^* \dot{\boldsymbol{T}}_{i-1} + \boldsymbol{K}_m^f \dot{\boldsymbol{T}}_{m-i}^* \boldsymbol{b}_{m-i+1}) \boldsymbol{\xi}_1(n-i) \\ &\quad + \cdots + \boldsymbol{K}_m^f \boldsymbol{b}_0 \boldsymbol{\xi}_1(n-m).\end{aligned} \tag{D.52}$$

Equating coefficients of equations D.48 and D.52

$$\boldsymbol{a}_0^* = \boldsymbol{b}_0^*, \tag{D.53}$$

$$\boldsymbol{a}_m^* \dot{\boldsymbol{T}}_{m-1} = \boldsymbol{K}_m^f \boldsymbol{b}_0, \tag{D.54}$$

$$\boldsymbol{a}_i^* \dot{\boldsymbol{T}}_{i-1} = \boldsymbol{b}_i^* \dot{\boldsymbol{T}}_{i-1} + \boldsymbol{K}_m^f \dot{\boldsymbol{T}}_{m-i}^* \boldsymbol{b}_{m-i+1}, \tag{D.55}$$

and

$$\boldsymbol{\chi}_m^f(n) = \boldsymbol{\chi}_{m-1}^f(n) + \boldsymbol{K}_m^f \boldsymbol{\chi}_{m-1}^b(n-1). \tag{D.56}$$

Similarly, take equation D.51 and add to equation D.50 multiplied by $\boldsymbol{K}_m^b$

$$\begin{aligned}\boldsymbol{K}_m^b \boldsymbol{\chi}_{m-1}^f(n) + \boldsymbol{\chi}_{m-1}^b(n-1) \\ &= \boldsymbol{K}_m^b \boldsymbol{b}_0^* \boldsymbol{\xi}_1(n) + (\boldsymbol{K}_m^b \boldsymbol{b}_1^* \dot{\boldsymbol{T}}_0 + \dot{\boldsymbol{T}}_{m-2}^* \boldsymbol{b}_{m-1}) \boldsymbol{\xi}_1(n-1) \\ &\quad + (\boldsymbol{K}_m^b \boldsymbol{b}_2^* \dot{\boldsymbol{T}}_1 + \dot{\boldsymbol{T}}_{m-3}^* \boldsymbol{b}_{m-2}) \boldsymbol{\xi}_1(n-2) \\ &\quad + \cdots + (\boldsymbol{K}_m^b \boldsymbol{b}_i^* \dot{\boldsymbol{T}}_{i-1} + \dot{\boldsymbol{T}}_{m-i}^* \boldsymbol{b}_{m-i+1}) \boldsymbol{\xi}_1(n-i) \\ &\quad + \cdots + \boldsymbol{b}_0 \boldsymbol{\xi}_1(n-m).\end{aligned} \tag{D.57}$$

Equating coefficients of equations D.49 and D.57,

$$\boldsymbol{a}_0 = \boldsymbol{b}_0, \tag{D.58}$$

$$\dot{\boldsymbol{T}}_{m-1}^* \boldsymbol{a}_m = \boldsymbol{K}_m^b \boldsymbol{b}_0^*, \tag{D.59}$$

$$\dot{\boldsymbol{T}}_{m-i-1}^* \boldsymbol{a}_{m-i} = \boldsymbol{K}_m^b \boldsymbol{b}_i^* \dot{\boldsymbol{T}}_{i-1} + \dot{\boldsymbol{T}}_{m-i}^* \boldsymbol{b}_{m-i+1}, \tag{D.60}$$

and

$$\boldsymbol{\chi}_m^b(n) = \boldsymbol{K}_m^b \boldsymbol{\chi}_{m-1}^f(n) + \boldsymbol{\chi}_{m-1}^b(n-1). \tag{D.61}$$

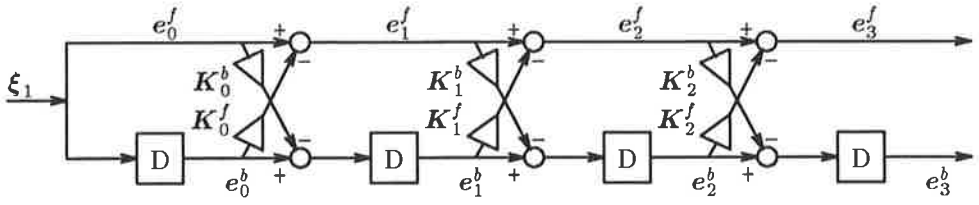


Figure D.3: The lattice filter.

Equations D.56 and D.61 are order recursive relationships which define a lattice filter (see figure D.3). Equations D.58 and D.59 define $\boldsymbol{K}_m^f$ and equations D.53 and D.54 define $\boldsymbol{K}_m^b$. Equations D.55 and D.60 are a pair of simultaneous equations which can be solved for the two unknowns $\boldsymbol{b}_i$ and $\boldsymbol{b}_{m-i+1}$. Writing equations D.55 and D.60 as one equation with partitioned matrices,

$$\left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{K}_m^b \\ \hline \boldsymbol{K}_m^f & \boldsymbol{I} \end{array}\right) \left(\begin{array}{c} \boldsymbol{b}_i^* \grave{\boldsymbol{T}}_{i-1} \\ \hline \grave{\boldsymbol{T}}_{m-i}^* \boldsymbol{b}_{m-i+1} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{a}_i^* \grave{\boldsymbol{T}}_{i-1} \\ \hline \grave{\boldsymbol{T}}_{m-i-1}^* \boldsymbol{a}_{m-i} \end{array}\right). \tag{D.62}$$

We now have a method of converting from a nonlinear transversal filter to a nonlinear lattice filter:

1. Use equations D.53 and D.54 to determine $\boldsymbol{K}_m^f$.
2. Use equation D.58 and D.59 to determine $\boldsymbol{K}_m^b$.
3. Calculate all the $\boldsymbol{b}_i^* \grave{\boldsymbol{T}}_{i-1}$ using equation D.62.
4. Copy all the $\boldsymbol{b}_i^* \grave{\boldsymbol{T}}_{i-1}$ to $\boldsymbol{a}_i^* \grave{\boldsymbol{T}}_{i-1}$.
5. Decrement m and repeat repeat the whole procedure until $m = 0$.

Up until now we have allowed the filter to be any linear combination of $\grave{\boldsymbol{T}}_i \boldsymbol{\xi}_1(n)$, where $0 \leq i \leq m$. For an adaptive lattice filter it is useful to consider the special case of predictors. A predictor is special because the desired value is one of the inputs delayed and so the prediction *error* is also a linear combination

of the inputs. Further, it is well known that for the case of a linear, real, scalar filter the coefficients of an optimal forward predictor taken in reverse order are the same as those for an optimal backward predictor [60].

We will now show that for a stationary *nonlinear* process the coefficients of an optimal forward predictor taken in reverse order and conjugate-transposed are the same as those for an optimal backward predictor. Further, assume we have access to the true correlation matrix $\hat{\mathbf{\Phi}}_m$. (The matrix $\boldsymbol{A}_m$ which we have been using up until now can be considered a best estimate of $\hat{\mathbf{\Phi}}_m$ based on the data processed). In this section we use $\mathcal{E}(x)$ to mean the expected value of x.

Define

$$\hat{\mathbf{\Phi}}_m = \begin{bmatrix} \mathcal{E}(\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k)^*) & \mathcal{E}(\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_2(k)^*) & \cdots & \mathcal{E}(\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_m(k)^*) \\ \mathcal{E}(\boldsymbol{\xi}_2(k)\boldsymbol{\xi}_1(k)^*) & \mathcal{E}(\boldsymbol{\xi}_2(k)\boldsymbol{\xi}_2(k)^*) & \cdots & \mathcal{E}(\boldsymbol{\xi}_2(k)\boldsymbol{\xi}_m(k)^*) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{E}(\boldsymbol{\xi}_m(k)\boldsymbol{\xi}_1(k)^*) & \mathcal{E}(\boldsymbol{\xi}_m(k)\boldsymbol{\xi}_2(k)^*) & \cdots & \mathcal{E}(\boldsymbol{\xi}_m(k)\boldsymbol{\xi}_m(k)^*) \end{bmatrix} \tag{D.63}$$

and from D.3,

$$\mathcal{E}(\boldsymbol{\xi}_i(k)\boldsymbol{\xi}_1(k)^*) = \grave{\boldsymbol{T}}_{i-1}\mathcal{E}(\boldsymbol{\xi}_1(k-i+1)\boldsymbol{\xi}_1(k)^*). \tag{D.64}$$

Note that since we have assumed a stationary process, the value of $\mathcal{E}(\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k+j)^*)$ is independent of k, and we can write

$$\mathcal{E}(\boldsymbol{\xi}_i(k)\boldsymbol{\xi}_1(k)^*) = \grave{\boldsymbol{T}}_{i-1}\boldsymbol{\phi}_{i-1}, \tag{D.65}$$

where

$$\boldsymbol{\phi}_j = \mathcal{E}(\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k+j)^*). \tag{D.66}$$

Combining equations D.63 and D.64,

$$\hat{\mathbf{\Phi}}_m = \hat{\grave{\boldsymbol{T}}}_{0,m-1}\mathbf{\Phi}\hat{\grave{\boldsymbol{T}}}^*_{0,m-1}, \tag{D.67}$$

where

$$\hat{\grave{\boldsymbol{T}}}_{i,j} = \operatorname*{diag}_{i\le k\le j}(\grave{\boldsymbol{T}}_k) \tag{D.68}$$

and

$$\mathbf{\Phi}_m = \begin{pmatrix} \phi_0 & \phi_1 & \cdots & \phi_{m-1} \\ \phi_1^* & \phi_0 & \cdots & \phi_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m-1}^* & \phi_{m-2}^* & \cdots & \phi_0 \end{pmatrix}. \tag{D.69}$$

Also define $\boldsymbol{\rho}_m^f$ and $\boldsymbol{\rho}_m^b$

$$\boldsymbol{\rho}_m^f = \begin{pmatrix} \mathcal{E}(\boldsymbol{\xi}_1(n)\boldsymbol{\xi}_1(n+1)^*) \\ \mathcal{E}(\boldsymbol{\xi}_2(n)\boldsymbol{\xi}_1(n+1)^*) \\ \vdots \\ \mathcal{E}(\boldsymbol{\xi}_m(n)\boldsymbol{\xi}_1(n+1)^*) \end{pmatrix} = \hat{\grave{\boldsymbol{T}}}_{0,m-1} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{pmatrix} \tag{D.70}$$

and

$$\boldsymbol{\rho}_m^b = \begin{pmatrix} \mathcal{E}(\boldsymbol{\xi}_1(n)\boldsymbol{\xi}_1(n-m)^*) \\ \mathcal{E}(\boldsymbol{\xi}_2(n)\boldsymbol{\xi}_1(n-m)^*) \\ \vdots \\ \mathcal{E}(\boldsymbol{\xi}_m(n)\boldsymbol{\xi}_1(n-m)^*) \end{pmatrix} = \hat{\grave{\boldsymbol{T}}}_{0,m-1} \begin{pmatrix} \phi_m^* \\ \phi_{m-1}^* \\ \vdots \\ \phi_1^* \end{pmatrix}. \tag{D.71}$$

We can now rewrite the Wiener-Hopf equation (D.23) replacing $\boldsymbol{A}$ with $\hat{\mathbf{\Phi}}$ and $\boldsymbol{v}$ with $\boldsymbol{\rho}$,

$$\hat{\grave{\boldsymbol{T}}}_{0,m-1} \mathbf{\Phi} \hat{\grave{\boldsymbol{T}}}^*_{0,m-1} \boldsymbol{c}_m^s = \boldsymbol{\rho}_m^s, \quad \text{for } s \in \{f, b\}. \tag{D.73}$$

The equation D.73 is true if

$$\mathbf{\Phi} \left(\hat{\grave{\boldsymbol{T}}}^*_{0,m-1} \boldsymbol{c}_m^f \right) = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{pmatrix} \tag{D.74}$$

and

$$\mathbf{\Phi} \left(\hat{\grave{\boldsymbol{T}}}^*_{0,m-1} \boldsymbol{c}_m^b \right) = \begin{pmatrix} \phi_m^* \\ \phi_{m-1}^* \\ \vdots \\ \phi_1^* \end{pmatrix}. \tag{D.75}$$

If we reverse the order of the subarrays of $(\hat{\grave{\boldsymbol{T}}}^*_{0,m-1} \boldsymbol{c}_m^b)$ and of the right hand side of equation D.75, then we must reverse the order of the rows and columns

of $\mathbf{\Phi}$ which is equivalent to taking the conjugate-transpose of each of the sub-matrices of $\mathbf{\Phi}$ (refer to equation D.69). Define $\boldsymbol{\alpha}_i^f$ and $\boldsymbol{\alpha}_j^b$ such that

$$\boldsymbol{c}_m^f(n) = \begin{pmatrix} \boldsymbol{\alpha}_1^f \\ \boldsymbol{\alpha}_2^f \\ \vdots \\ \boldsymbol{\alpha}_m^f \end{pmatrix} \quad \text{and} \quad \boldsymbol{c}_m^b(n) = \begin{pmatrix} \boldsymbol{\alpha}_1^b \\ \boldsymbol{\alpha}_2^b \\ \vdots \\ \boldsymbol{\alpha}_m^b \end{pmatrix}. \tag{D.76}$$

Consequently,

$$\mathbf{\Phi} \begin{pmatrix} \boldsymbol{\alpha}_m^{b\,*}\dot{\boldsymbol{T}}_{m-1} \\ \boldsymbol{\alpha}_{m-1}^{b\,*}\dot{\boldsymbol{T}}_{m-2} \\ \vdots \\ \boldsymbol{\alpha}_1^{b\,*}\dot{\boldsymbol{T}}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \vdots \\ \boldsymbol{\phi}_m \end{pmatrix}. \tag{D.77}$$

Comparing equations D.77 and D.74,

$$\dot{\boldsymbol{T}}_{i-1}^* \boldsymbol{\alpha}_i^f = \boldsymbol{\alpha}_{m+1-i}^{b\,*}\dot{\boldsymbol{T}}_{m-i}, \tag{D.78}$$

as required.

From equations D.14 and D.12

$$\begin{aligned} \boldsymbol{e}_m^f(n) &= \boldsymbol{\xi}_1(n+1) - \boldsymbol{c}_m^{f\,*}\boldsymbol{x}_m(n) \\ &= \boldsymbol{\xi}_1(n+1) + \boldsymbol{\alpha}_1^{f\,*}\boldsymbol{\xi}_1(n) + \boldsymbol{\alpha}_2^{f\,*}\boldsymbol{\xi}_2(n) \\ &\quad + \cdots + \boldsymbol{\alpha}_m^{f\,*}\boldsymbol{\xi}_m(n) \\ &= \boldsymbol{\xi}_1(n+1) + \boldsymbol{\alpha}_1^{f\,*}\dot{\boldsymbol{T}}_0\boldsymbol{\xi}_1(n) + \boldsymbol{\alpha}_2^{f\,*}\dot{\boldsymbol{T}}_1\boldsymbol{\xi}_1(n-1) \\ &\quad + \cdots + \boldsymbol{\alpha}_m^{f\,*}\dot{\boldsymbol{T}}_{m-1}\boldsymbol{\xi}_1(n-m+1), \\ \boldsymbol{e}_m^f(n-1) &= \boldsymbol{\xi}_1(n) + \boldsymbol{\alpha}_1^{f\,*}\dot{\boldsymbol{T}}_0\boldsymbol{\xi}_1(n-1) + \boldsymbol{\alpha}_2^{f\,*}\dot{\boldsymbol{T}}_1\boldsymbol{\xi}_1(n-2) \\ &\quad + \cdots + \boldsymbol{\alpha}_m^{f\,*}\dot{\boldsymbol{T}}_{m-1}\boldsymbol{\xi}_1(n-m), \end{aligned} \tag{D.79}$$

which is of the same form as D.48. Similarly

$$\begin{aligned} \boldsymbol{e}_m^b(n) &= \boldsymbol{\xi}_1(n-m) - \boldsymbol{c}_m^{b\,*}\boldsymbol{x}_m(n) \\ &= \boldsymbol{\xi}_1(n-m) + \boldsymbol{\alpha}_1^{b\,*}\boldsymbol{\xi}_1(n) + \boldsymbol{\alpha}_2^{b\,*}\boldsymbol{\xi}_2(n) \\ &\quad + \cdots + \boldsymbol{\alpha}_m^{b\,*}\boldsymbol{\xi}_m(n) \end{aligned}$$

$$= \boldsymbol{\alpha}_1^{b*}\dot{\boldsymbol{T}}_0\boldsymbol{\xi}_1(n) + \boldsymbol{\alpha}_2^{b*}\dot{\boldsymbol{T}}_1\boldsymbol{\xi}_1(n-1) + \cdots + \boldsymbol{\alpha}_m^{b\,*}\dot{\boldsymbol{T}}_{m-1}\boldsymbol{\xi}_1(n-m+1) + \boldsymbol{\xi}_1(n-m). \quad \text{(D.80)}$$

Provided we are predicting a stationary process and $\boldsymbol{c}_m^f$ and $\boldsymbol{c}_m^b$ contain the coefficients of the optimal forward and backward predictors, equation D.78 holds and equation D.80 can be rewritten

$$\boldsymbol{e}_m^b(n) = \dot{\boldsymbol{T}}_{m-1}^*\boldsymbol{\alpha}_m^f\boldsymbol{\xi}_1(n) + \dot{\boldsymbol{T}}_{m-2}^*\boldsymbol{\alpha}_{m-1}^f\boldsymbol{\xi}_1(n-1) + \cdots + \dot{\boldsymbol{T}}_0^{\,*}\boldsymbol{\alpha}_1^f\boldsymbol{\xi}_1(n-m+1) + \boldsymbol{\xi}_1(n-m), \quad \text{(D.81)}$$

which is of the same form as equation D.49. Thus if the filter described by D.48 has as its output the forward optimum prediction error $\boldsymbol{e}_m^f(n-1)$, then the filter described by equation D.49 has as its output the optimum backward prediction error $\boldsymbol{e}_m^b(n)$. Further more, $\boldsymbol{a}_0$ and $\boldsymbol{b}_0$ are always zero simplifying equations D.54 and D.59 so that

$$\boldsymbol{K}_m^f = \boldsymbol{a}_m^*\dot{\boldsymbol{T}}_{m-1} = \boldsymbol{K}_m^{b\,*}, \quad \text{(D.82)}$$

and equation D.62 becomes

$$\left(\begin{array}{c|c} \boldsymbol{I} & \dot{\boldsymbol{T}}_{m-1}^*\boldsymbol{a}_m \\ \hline \boldsymbol{a}_m^*\dot{\boldsymbol{T}}_{m-1} & \boldsymbol{I} \end{array}\right)\left(\begin{array}{c} \boldsymbol{b}_i^*\dot{\boldsymbol{T}}_{i-1} \\ \hline \dot{\boldsymbol{T}}_{m-i}^*\boldsymbol{b}_{m-i+1} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{a}_i^*\dot{\boldsymbol{T}}_{i-1} \\ \hline \dot{\boldsymbol{T}}_{m-i-1}^*\boldsymbol{a}_{m-i} \end{array}\right). \quad \text{(D.83)}$$

We have shown that a lattice filter to optimally estimate the forward and backward prediction errors has $\boldsymbol{K}_m^f = \boldsymbol{K}_m^{b\,*}$, however, a least squares adaptive filter may still be sub-optimal since the data samples are finite and the process being predicted may not be stationary. Some adaptive lattice algorithms make the assumption that $\boldsymbol{K}_m^f = \boldsymbol{K}_m^{b\,*}$, but in the true least squares algorithm which is the subject of this appendix $\boldsymbol{K}_m^f \neq \boldsymbol{K}_m^{b\,*}$.

We will now demonstrate that the least-squares backward prediction errors, suitably transformed, form an orthogonal basis for $\boldsymbol{x}_m(n)$. It follows straight away from equation D.22 that $\boldsymbol{x}_m$ is orthogonal to $\tilde{\boldsymbol{e}}_m^b(n)$. Conjugate equation

D.1 and substitute into equation D.22

$$\sum_{k=0}^{n} \lambda^{n-k} \left[\begin{array}{c} \boldsymbol{\xi}_1(k) \\ \hline \boldsymbol{\xi}_2(k) \\ \hline \vdots \\ \hline \boldsymbol{\xi}_m(k) \end{array}\right] \tilde{\boldsymbol{e}}_m^b(n,k)^* = 0,$$

which implies

$$\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_i(k) \tilde{\boldsymbol{e}}_m^b(n,k)^* = 0, \qquad \text{for } 1 \leq i \leq m. \tag{D.84}$$

That is, $\boldsymbol{\xi}_i$ is orthogonal to $\tilde{\boldsymbol{e}}_m^b$ for $1 \leq i \leq m$. Now consider the product of $\grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n,k)$ and $\tilde{\boldsymbol{e}}_j^b(n,k)^*$.

$$\begin{aligned} \sum_{k=0}^{n} \lambda^{n-k} \grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n,k) \tilde{\boldsymbol{e}}_j^b(n,k)^* &= \sum_{k=0}^{n} \lambda^{n-k} \Big[\grave{\boldsymbol{T}}_i \boldsymbol{\xi}_1(n-i) \\ &\qquad + \grave{\boldsymbol{T}}_i \sum_{l=1}^{i} \boldsymbol{\alpha}_l^{b*} \boldsymbol{\xi}_l(n) \Big] \tilde{\boldsymbol{e}}_j^b(n,k)^* \\ &= \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_{i+1}(n) \tilde{\boldsymbol{e}}_j^b(n,k)^* \\ &\qquad + \grave{\boldsymbol{T}}_i \sum_{l=1}^{i} \boldsymbol{\alpha}_l \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_l(n) \tilde{\boldsymbol{e}}_j^b(n,k)^* \end{aligned} \tag{D.85}$$

and from equation D.84

$$\sum_{k=0}^{n} \lambda^{n-k} \grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n,k) \tilde{\boldsymbol{e}}_j^b(n,k)^* = 0, \qquad \text{for } 1 \leq i < j. \tag{D.86}$$

Thus $\grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b$ is orthogonal to $\tilde{\boldsymbol{e}}_j^b$ for $1 \leq i < j$. If we conjugate equation D.86 then we get

$$\begin{aligned} &\sum_{k=0}^{n} \lambda^{n-k} \tilde{\boldsymbol{e}}_j^b(n,k) \tilde{\boldsymbol{e}}_i^b(n,k)^* \grave{\boldsymbol{T}}_i^* = 0, \qquad \text{for } 1 \leq i < j; \\ &\sum_{k=0}^{n} \lambda^{n-k} \tilde{\boldsymbol{e}}_i^b(n,k) \tilde{\boldsymbol{e}}_j^b(n,k)^* \grave{\boldsymbol{T}}_j^* = 0, \qquad \text{for } 1 \leq j < i. \end{aligned} \tag{D.87}$$

Post-multiplying equation D.86 by $\grave{\boldsymbol{T}}_j^*$ and pre-multiplying equation D.87 by $\grave{\boldsymbol{T}}_i$ implies

$$\sum_{k=0}^{n} \lambda^{n-k} \grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n,k) \tilde{\boldsymbol{e}}_j^b(n,k)^* \grave{\boldsymbol{T}}_j^* = 0, \qquad \text{for } i \neq j. \tag{D.88}$$

Now from equations D.15 and D.12,

$$\tilde{\boldsymbol{e}}_m^b(n) = \boldsymbol{\xi}_1(n-m) - \boldsymbol{c}_m^b(n)^* \boldsymbol{x}_m(n)$$

and

$$\begin{aligned}
\grave{\boldsymbol{T}}_m \tilde{\boldsymbol{e}}_m^b(n) &= \boldsymbol{\xi}_{m+1}(n) - \grave{\boldsymbol{T}}_m \boldsymbol{c}_m^b(n)^* \boldsymbol{x}_m(n) \\
&= \left(\begin{array}{c|c} -\grave{\boldsymbol{T}}_m \boldsymbol{c}_m^b(n)^* & \boldsymbol{I} \end{array} \right) \left(\begin{array}{c} \boldsymbol{\xi}_1(n) \\ \hline \boldsymbol{\xi}_2(n) \\ \hline \vdots \\ \hline \boldsymbol{\xi}_m(n) \\ \hline \boldsymbol{\xi}_{m+1}(n) \end{array} \right) \\
&= \left(\begin{array}{c|c} -\grave{\boldsymbol{T}}_m \boldsymbol{c}_m^b(n)^* & \boldsymbol{I} \end{array} \right) \boldsymbol{x}_{m+1}(n).
\end{aligned}$$

Therefore,

$$\left(\begin{array}{c} \grave{\boldsymbol{T}}_0 \tilde{\boldsymbol{e}}_0^b(n) \\ \hline \grave{\boldsymbol{T}}_1 \tilde{\boldsymbol{e}}_1^b(n) \\ \hline \vdots \\ \hline \grave{\boldsymbol{T}}_m \tilde{\boldsymbol{e}}_m^b(n) \end{array} \right) = \left(\begin{array}{c|c|c} -\grave{\boldsymbol{T}}_0 \boldsymbol{c}_0^b(n)^* & \boldsymbol{I} & \boldsymbol{0} \\ \hline -\grave{\boldsymbol{T}}_1 \boldsymbol{c}_1^b(n)^* & \boldsymbol{I} & \boldsymbol{0} \\ \hline \multicolumn{3}{c}{\vdots} \\ \hline \multicolumn{2}{c|}{-\grave{\boldsymbol{T}}_m \boldsymbol{c}_m^b(n)^*} & \boldsymbol{I} \end{array} \right) \boldsymbol{x}_{m+1}(n). \tag{D.89}$$

$$= \boldsymbol{L}_m(n) \boldsymbol{x}_m(n), \tag{D.90}$$

where

$$\boldsymbol{L}_m(n) = \left(\begin{array}{c|c|c} -\grave{\boldsymbol{T}}_0 \boldsymbol{c}_0^b(n)^* & \boldsymbol{I} & \boldsymbol{0} \\ \hline -\grave{\boldsymbol{T}}_1 \boldsymbol{c}_1^b(n)^* & \boldsymbol{I} & \boldsymbol{0} \\ \hline \multicolumn{3}{c}{\vdots} \\ \hline \multicolumn{2}{c|}{-\grave{\boldsymbol{T}}_{m-1} \boldsymbol{c}_{-1}^b(n)^*} & \boldsymbol{I} \end{array} \right). \tag{D.91}$$

Since the $\grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n)$ are a linear combination of the $\boldsymbol{\xi}_m(n)$, they span the same space so long as $\boldsymbol{L}_m(n)$ is of full rank. Thus pre-multiplication by $\boldsymbol{L}_m(n)$ has orthogonalized $\boldsymbol{x}_m(n)$. In a sense, this is only a partial orthogonalization. The subvectors $\grave{\boldsymbol{T}}_i \tilde{\boldsymbol{e}}_i^b(n)$ are orthogonal to each other, but the elements of $\tilde{\boldsymbol{\xi}}_i$ are not orthogonal to each other. This is, however, the best that can be achieved by the lattice structure.

Now define a transformed input vector

$$\tilde{\boldsymbol{x}}_m(n) = \boldsymbol{L}_m(n-1) \boldsymbol{x}_m(n), \tag{D.92}$$

where

$$\tilde{\boldsymbol{x}}_m(n) = \left[\begin{array}{c} \tilde{\boldsymbol{\xi}}_1(n) \\ \hline \tilde{\boldsymbol{\xi}}_2(n) \\ \hline \vdots \\ \hline \tilde{\boldsymbol{\xi}}_m(n) \end{array}\right], \tag{D.93}$$

and from equations D.91 and D.14,

$$\tilde{\boldsymbol{\xi}}_m(n) = \grave{\boldsymbol{T}}_{m-1} \boldsymbol{e}^b_{m-1}(n). \tag{D.94}$$

In this case, the sub-vectors of $\tilde{\boldsymbol{x}}_m(n)$, $\tilde{\boldsymbol{\xi}}_m$, are not truly orthogonal, though for a stationary process and sufficiently large n they will approach orthogonality. In the lattice based equalizer algorithm we define the output to be a linear combination of $\tilde{\boldsymbol{x}}$.

Let us define $\tilde{\boldsymbol{A}}(n)$ as

$$\tilde{\boldsymbol{A}}_m(n) = \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{L}_m(n) \boldsymbol{x}_m(k) \boldsymbol{x}_m(k)^* \boldsymbol{L}_m(n)^* \tag{D.95}$$

$$= \boldsymbol{L}_m(n) \boldsymbol{A}_m(n) \boldsymbol{L}_m(n)^*. \tag{D.96}$$

We can partition $\tilde{\boldsymbol{A}}(n)$

$$\tilde{\boldsymbol{A}}_m(n) = \left[\begin{array}{cccc} \boldsymbol{a}_{1,1} & \boldsymbol{a}_{1,2} & \cdots & \boldsymbol{a}_{1,m} \\ \boldsymbol{a}_{2,1} & \boldsymbol{a}_{2,2} & \cdots & \boldsymbol{a}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{m,1} & \boldsymbol{a}_{m,2} & \cdots & \boldsymbol{a}_{m,m} \end{array}\right],$$

where (from equations D.90 and D.96),

$$\boldsymbol{a}_{i,j} = \sum_{k=0}^{n} \lambda^{n-k} \grave{\boldsymbol{T}}_{i-1} \tilde{\boldsymbol{e}}^b_{i-1}(n,k) \tilde{\boldsymbol{e}}^b_{j-1}(n,k)^* \grave{\boldsymbol{T}}^*_{j-1},$$

and from equation D.88,

$$\boldsymbol{a}_{i,j} = 0, \qquad \text{for } i \neq j. \tag{D.97}$$

Thus,

$$\tilde{\boldsymbol{A}}_m(n) = \underset{1 \leq i \leq m-1}{\text{diag}} \left(\grave{\boldsymbol{T}}_{i-1} \left[\sum_{k=0}^{n} \lambda^{n-k} \tilde{\boldsymbol{e}}^b_{i-1}(n,k) \tilde{\boldsymbol{e}}^b_{i-1}(n,k)^* \right] \grave{\boldsymbol{T}}^*_{i-1} \right),$$

and from equation D.17,

$$\tilde{\boldsymbol{A}}_m(n) = \underset{0 \leq i \leq m-1}{\text{diag}} \left(\grave{\boldsymbol{T}}_i \boldsymbol{\epsilon}^b_i(n) \grave{\boldsymbol{T}}^*_i \right). \tag{D.98}$$

Equate the right half sides of equations D.95 and D.98

$$\boldsymbol{L}_m(n)\boldsymbol{A}_m(n)\boldsymbol{L}_m(n)^* = \underset{0\le i\le m-1}{\text{diag}}\left(\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n)\grave{\boldsymbol{T}}_i^*\right),$$

pre-multiply by $\boldsymbol{L}_m(n)^{-1}$ and post-multiply by $[\boldsymbol{L}_m(n)^*]^{-1}$,

$$\boldsymbol{A}_m(n) = \boldsymbol{L}_m(n)^{-1}\underset{0\le i\le m-1}{\text{diag}}\left(\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n)\grave{\boldsymbol{T}}_i^*\right)[\boldsymbol{L}_m(n)^*]^{-1},$$

and invert,

$$\boldsymbol{A}_m(n)^{-1} = \boldsymbol{L}_m(n)^*\underset{0\le i\le m-1}{\text{diag}}\left(\left[\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n)\grave{\boldsymbol{T}}_i^*\right]^{-1}\right)\boldsymbol{L}_m(n). \tag{D.99}$$

Now the equalizer output is a linear combination of $\tilde{\boldsymbol{x}}$

$$\boldsymbol{y}_m^e(n) = \tilde{\boldsymbol{c}}_m^e(n-1)^*\tilde{\boldsymbol{x}}_m(n), \tag{D.100}$$

and from equation D.11,

$$\begin{aligned}\boldsymbol{c}_m^e(n-1)^*\boldsymbol{x}_m(n) &= \tilde{\boldsymbol{c}}_m^e(n-1)^*\tilde{\boldsymbol{x}}_m(n)\\ &= \tilde{\boldsymbol{c}}_m^e(n-1)^*\boldsymbol{L}_m(n-1)\tilde{\boldsymbol{x}}_m(n).\end{aligned} \tag{D.101}$$

Equation D.101 is satisfied if

$$\begin{aligned}\boldsymbol{c}_m^e(n-1)^* &= \tilde{\boldsymbol{c}}_m^e(n-1)^*\boldsymbol{L}_m(n-1)\\ \tilde{\boldsymbol{c}}_m^e(n) &= \boldsymbol{L}_m(n)^{*-1}\boldsymbol{c}_m^e(n).\end{aligned}$$

Apply equation D.23,

$$\tilde{\boldsymbol{c}}_m^e(n) = \boldsymbol{L}_m(n)^{*-1}\boldsymbol{A}_m(n)^{-1}\boldsymbol{v}_m^e(n),$$

and substitute equation D.99,

$$\begin{aligned}\tilde{\boldsymbol{c}}_m^e(n) &= \boldsymbol{L}_m(n)^{*-1}\boldsymbol{L}_m(n)^*\underset{0\le i\le m-1}{\text{diag}}\left(\left[\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n)\grave{\boldsymbol{T}}_i^*\right]^{-1}\right)\boldsymbol{L}_m(n)^*\boldsymbol{v}_m^e(n)\\ &= \underset{0\le i\le m-1}{\text{diag}}\left(\left[\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n)\grave{\boldsymbol{T}}_i^*\right]^{-1}\right)\boldsymbol{L}_m(n)\boldsymbol{v}_m^e(n).\end{aligned} \tag{D.102}$$

Take the conjugate-transpose of equation D.102, replace n with $n-1$, and substitute into equation D.100,

$$\begin{aligned}\boldsymbol{y}_m^e(n) = &(\boldsymbol{L}_m(n-1)\boldsymbol{v}_m^e(n-1))^*\\ &\times\underset{0\le i\le m-1}{\text{diag}}\left(\left[\grave{\boldsymbol{T}}_i\boldsymbol{\epsilon}_i^b(n-1)\grave{\boldsymbol{T}}_i^*\right]^{-1}\right)\tilde{\boldsymbol{x}}_m(n).\end{aligned} \tag{D.103}$$

Let

$$[\boldsymbol{L}_m(n)\boldsymbol{v}_m^e(n)]^* = \left(\boldsymbol{z}_1(n)^* \,\middle|\, \boldsymbol{z}_2(n)^* \,\middle|\, \ldots \,\middle|\, \boldsymbol{z}_m(n)^* \right) \tag{D.104}$$

and substitute into D.103 to get

$$\begin{aligned}\boldsymbol{y}_m^e(n) = &\left(\boldsymbol{z}_1(n)^* \,\middle|\, \boldsymbol{z}_2(n)^* \,\middle|\, \ldots \,\middle|\, \boldsymbol{z}_m(n)^* \right) \\ &\times \operatorname*{diag}_{0\le i\le m-1} \left(\left[\grave{\boldsymbol{T}}_i \boldsymbol{\epsilon}_i^b(n-1) \grave{\boldsymbol{T}}_i^* \right]^{-1} \right) \tilde{\boldsymbol{x}}_m(n),\end{aligned}$$

and substitute D.93 to get

$$\begin{aligned}\boldsymbol{y}_m^e(n) &= \left(\boldsymbol{z}_1(n-1)^* \,\middle|\, \boldsymbol{z}_2(n-1)^* \,\middle|\, \ldots \right) \left(\begin{array}{c} \left[\grave{\boldsymbol{T}}_0 \boldsymbol{\epsilon}_0^b(n-1) \grave{\boldsymbol{T}}_0^* \right]^{-1} \tilde{\boldsymbol{\xi}}_1(n) \\ \hline \left[\grave{\boldsymbol{T}}_1 \boldsymbol{\epsilon}_1^b(n-1) \grave{\boldsymbol{T}}_1^* \right]^{-1} \tilde{\boldsymbol{\xi}}_2(n) \\ \hline \vdots \end{array} \right) \\ &= \sum_{i=1}^{m} \boldsymbol{z}_i(n-1)^* \left[\grave{\boldsymbol{T}}_{i-1} \boldsymbol{\epsilon}_{i-1}^b(n-1) \grave{\boldsymbol{T}}_{i-1}^* \right]^{-1} \tilde{\boldsymbol{\xi}}_i(n),\end{aligned}$$

and from equation D.94

$$\begin{aligned}\boldsymbol{y}_m^e(n) &= \sum_{i=1}^{m} \boldsymbol{z}_i(n-1)^* \left[\grave{\boldsymbol{T}}_{i-1} \boldsymbol{\epsilon}_{i-1}^b(n-1) \grave{\boldsymbol{T}}_{i-1}^* \right]^{-1} \grave{\boldsymbol{T}}_{i-1} \boldsymbol{e}_{i-1}^b(n) \\ &= \sum_{i=1}^{m} \boldsymbol{z}_i(n-1)^* \left[\grave{\boldsymbol{T}}_{i-1} \boldsymbol{\epsilon}_{i-1}^b(n-1) \grave{\boldsymbol{T}}_{i-1}^* \right]^{-1} \grave{\boldsymbol{T}}_{m-1} \boldsymbol{e}_{m-1}^b(n) \\ &= \boldsymbol{y}_{m-1}^e(n) + \boldsymbol{z}_m(n-1)^* \left[\grave{\boldsymbol{T}}_{m-1} \boldsymbol{\epsilon}_{m-1}^b(n-1) \grave{\boldsymbol{T}}_{m-1}^* \right]^{-1} \\ &\qquad \times \grave{\boldsymbol{T}}_{m-1} \boldsymbol{e}_{m-1}^b(n).\end{aligned} \tag{D.105}$$

Equation D.105 is an order recursive form for calculating the equalizer output $\boldsymbol{y}_m^e(n)$. It depends however on being able to calculate $\boldsymbol{z}_m(n-1)^*$, $\boldsymbol{\epsilon}_{m-1}^b(n-1)$ and $\boldsymbol{e}_{m-1}^b(n)$. We proceed now to find a recursive form for $\boldsymbol{z}_m(n-1)^*$.

From equations D.104 and D.91,

$$\boldsymbol{z}_m(n) = \left(-\grave{\boldsymbol{T}}_{m-1} \boldsymbol{c}_{m-1}^b(n)^* \,\middle|\, \boldsymbol{I} \,\middle|\, \boldsymbol{0} \right) \boldsymbol{v}_m^e(n).$$

From equations D.1 and D.25,

$$\boldsymbol{z}_m(n) = \left(-\grave{\boldsymbol{T}}_{m-1} \boldsymbol{c}_{m-1}^b(n)^* \,\middle|\, \boldsymbol{I} \,\middle|\, \boldsymbol{0} \right) \left(\frac{\boldsymbol{v}_{m-1}^e(n)}{\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_m(k) \boldsymbol{d}_m^e(k)^*} \right),$$

and substitute equation D.45 to get

$$\begin{aligned}
\boldsymbol{z}_m(n) &= -\lambda \grave{\boldsymbol{T}}_{m-1}\boldsymbol{c}^b_{m-1}(n-1)^*\boldsymbol{v}^e_{m-1}(n-1) \\
&\quad + [1-\gamma_{m-1}(n)]\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^* \\
&\quad - \grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n)\boldsymbol{d}^e_{m-1}(n)^* + \sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_m(k)\boldsymbol{d}^e_m(k)^* \\
&= \lambda\left(-\grave{\boldsymbol{T}}_{m-1}\boldsymbol{c}^b_{m-1}(n-1)^*\boldsymbol{v}^e_{m-1}(n-1) + \sum_{k=0}^{n-1}\lambda^{n-1-k}\boldsymbol{\xi}_m(k)\boldsymbol{d}^e_m(k)^*\right) \\
&\quad + \boldsymbol{\xi}_m(k)\boldsymbol{d}^e_m(k)^* + [1-\gamma_{m-1}(n)]\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^* \\
&\quad - \grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n)\boldsymbol{d}^e_{m-1}(n)^* \\
&= \lambda\boldsymbol{z}_m(n-1) + [1-\gamma_{m-1}(n)]\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^* \\
&\quad + \boldsymbol{\xi}_m(k)\boldsymbol{d}^e_m(k)^* - \grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n)\boldsymbol{d}^e_{m-1}(n)^*,
\end{aligned}$$

and note that $\boldsymbol{d}^e_m(n) = \boldsymbol{d}^e_{m-1}(n)$ (equation D.12) to get,

$$\begin{aligned}
\boldsymbol{z}_m(n) &= \lambda\boldsymbol{z}_m(n-1) + [1-\gamma_{m-1}(n)]\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^* \\
&\quad + \left[\boldsymbol{\xi}_m(n) - \grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n)\right]\boldsymbol{d}^e_m(n)^* \\
&= \lambda\boldsymbol{z}_m(n-1) + (1-\gamma_{m-1}(n))\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^* \\
&\quad + \left[\boldsymbol{\xi}_m(n) - \grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n)\right]\boldsymbol{d}^e_m(n)^*. \qquad \text{(D.106)}
\end{aligned}$$

From equation D.12,

$$\boldsymbol{d}^b_{m-1}(n) = \boldsymbol{\xi}_1(n-m+1)$$

and, applying equation D.4,

$$\grave{\boldsymbol{T}}_{m-1}\boldsymbol{d}^b_{m-1}(n) = \boldsymbol{\xi}_m(n)$$

and substitute in equation D.106 to get

$$\boldsymbol{z}_m(n) = \lambda\boldsymbol{z}_m(n-1) + [1-\gamma_{m-1}(n)]\,\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}^b_{m-1}(n)\boldsymbol{e}^e_{m-1}(n)^*. \qquad \text{(D.107)}$$

D.4 Order Update Recursions

In this section we derive order update recursive relationships, effectively describing a filter of length m in terms of a filter of length $m-1$. These relationships are clearly necessary for a lattice filter. They will also be necessary for the fast RLS transversal filters.

Define

$$\hat{\boldsymbol{x}}_m(n) = \begin{bmatrix} \boldsymbol{\xi}_1(n) \\ \boldsymbol{\xi}_1(n-1) \\ \vdots \\ \boldsymbol{\xi}_1(n-m+1) \end{bmatrix}. \tag{D.108}$$

From equations D.1, D.3 and D.68 it follows that

$$\boldsymbol{x}_m(n) = \hat{\boldsymbol{T}}_{0,m-1}\hat{\boldsymbol{x}}_m(n). \tag{D.109}$$

Now

$$\hat{\boldsymbol{x}}_m(n) = \begin{bmatrix} \hat{\boldsymbol{x}}_{m-1}(n) \\ \hline \boldsymbol{\xi}_1(n-(m-1)) \end{bmatrix},$$

and from equation D.12,

$$\hat{\boldsymbol{x}}_m(n) = \begin{bmatrix} \hat{\boldsymbol{x}}_{m-1}(n) \\ \hline \boldsymbol{d}^b_{m-1}(n) \end{bmatrix}. \tag{D.110}$$

Similarly

$$\hat{\boldsymbol{x}}_m(n) = \begin{bmatrix} \boldsymbol{\xi}_1(n) \\ \hline \hat{\boldsymbol{x}}_{m-1}(n-1) \end{bmatrix}$$

and from equation D.12,

$$\hat{\boldsymbol{x}}_m(n) = \begin{bmatrix} \boldsymbol{d}^f_{m-1}(n-1) \\ \hline \hat{\boldsymbol{x}}_{m-1}(n-1) \end{bmatrix}. \tag{D.111}$$

Also define

$$\hat{\boldsymbol{A}}_m(n) = \sum_{k=0}^{n} \lambda^{n-k}\hat{\boldsymbol{x}}_m(k)\hat{\boldsymbol{x}}_m(k)^* \tag{D.112}$$

and

$$\hat{\boldsymbol{v}}_m^s(n) = \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_m(k) \boldsymbol{d}_m^s(k)^*. \tag{D.113}$$

From equations D.24, D.109 and D.112

$$\boldsymbol{A}_m = \hat{\dot{\boldsymbol{T}}}_{0,m-1} \hat{\boldsymbol{A}}_m \hat{\dot{\boldsymbol{T}}}_{0,m-1}^*. \tag{D.114}$$

Similarly, from equations D.25, D.109 and D.113

$$\boldsymbol{v}_m = \hat{\dot{\boldsymbol{T}}}_{0,m-1} \hat{\boldsymbol{v}}_m. \tag{D.115}$$

From equation D.110,

$$\hat{\boldsymbol{x}}_{m+1}(k) \hat{\boldsymbol{x}}_{m+1}(k)^* = \left(\begin{array}{c|c} \hat{\boldsymbol{x}}_m(k) \hat{\boldsymbol{x}}_m(k)^* & \hat{\boldsymbol{x}}_m(k) \boldsymbol{d}_m^b(k)^* \\ \hline \boldsymbol{d}_m^b(k) \hat{\boldsymbol{x}}_m(k)^* & \boldsymbol{d}_m^b(k) \boldsymbol{d}_m^b(k)^* \end{array} \right), \tag{D.116}$$

and substituting in D.112,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|c} \hat{\boldsymbol{A}}_m(n+1) & \sum_{k=0}^{n+1} \lambda^{n+1-k} \hat{\boldsymbol{x}}_m(k) \boldsymbol{d}_m^b(k)^* \\ \hline \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{d}_m^b(k) \hat{\boldsymbol{x}}_m(k)^* & \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{d}_m^b(k) \boldsymbol{d}_m^b(k)^* \end{array} \right),$$

and applying equations D.113 and D.35,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|c} \hat{\boldsymbol{A}}_m(n+1) & \hat{\boldsymbol{v}}_m^b(n+1) \\ \hline \hat{\boldsymbol{v}}_m^b(n+1)^* & \boldsymbol{E}_m^b(n+1) \end{array} \right). \tag{D.117}$$

Similarly, using equation D.111,

$$\hat{\boldsymbol{x}}_{m+1}(k) \hat{\boldsymbol{x}}_{m+1}(k)^* = \left(\begin{array}{c|c} \boldsymbol{d}_m^f(k-1) \boldsymbol{d}_m^f(k-1)^* & \boldsymbol{d}_m^f(k-1) \hat{\boldsymbol{x}}_m(k-1)^* \\ \hline \hat{\boldsymbol{x}}_m(k-1) \boldsymbol{d}_m^f(k-1)^* & \hat{\boldsymbol{x}}_m(k-1) \hat{\boldsymbol{x}}_m(k-1)^* \end{array} \right). \tag{D.118}$$

Consider an arbitrary function $f(m,k)$,

$$\sum_{l=0}^{n+1} \lambda^{n+1-l} f(m, l-1) = \sum_{k=-1}^{n} \lambda^{n-k} f(m,k)$$

by setting $k = l - 1$,

$$\sum_{k=0}^{n+1} \lambda^{n+1-k} f(m, k-1) = \lambda^{n+1} f(m,-1) + \sum_{k=0}^{n} \lambda^{n-k} f(m,k). \tag{D.119}$$

Setting $f(m, k-1) = \hat{\boldsymbol{x}}_{m+1}(k) \hat{\boldsymbol{x}}_{m+1}(k)^*$ and applying equation D.112,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \lambda^{n+1} \hat{\boldsymbol{x}}_{m+1}(0) \hat{\boldsymbol{x}}_{m+1}(0)^* + \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_{m+1}(k+1) \hat{\boldsymbol{x}}_{m+1}(k+1)^*$$

and, provided $\hat{\boldsymbol{x}}_{m+1}(0) = \mathbf{0}$,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_{m+1}(k+1) \hat{\boldsymbol{x}}_{m+1}(k+1)^*.$$

Substitute equation D.118 with k replaced by $k+1$ to get

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|c} \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{d}_m^f(k) \boldsymbol{d}_m^f(k)^* & \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{d}_m^f(k) \hat{\boldsymbol{x}}_m(k)^* \\ \hline \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_m(k) \boldsymbol{d}_m^f(k)^* & \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_m(k) \hat{\boldsymbol{x}}_m(k)^* \end{array} \right),$$

and applying equations D.112, D.113 and D.35,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|c} \boldsymbol{E}_m^f(n)^* & \hat{\boldsymbol{v}}_m^f(n)^* \\ \hline \hat{\boldsymbol{v}}_m^f(n) & \hat{\boldsymbol{A}}_m(n) \end{array} \right). \tag{D.120}$$

We now use equation D.120 with m replaced by $m-1$ to substitute for $\hat{\boldsymbol{A}}_m(n+1)$ in equation D.117,

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{cc|c} \boldsymbol{E}_{m-1}^f(n)^* & \hat{\boldsymbol{v}}_{m-1}^f(n)^* & \\ \hat{\boldsymbol{v}}_{m-1}^f(n) & \hat{\boldsymbol{A}}_{m-1}(n) & \hat{\boldsymbol{v}}_m^b(n+1) \dot{\boldsymbol{T}}_m^* \\ \hline \multicolumn{2}{c|}{\hat{\boldsymbol{v}}_m^b(n+1)^*} & \boldsymbol{E}_m^b(n+1) \end{array} \right). \tag{D.121}$$

From the definition of $\hat{\boldsymbol{v}}_m$ (equation D.113), and equation D.111

$$\hat{\boldsymbol{v}}_m^b(n+1) = \sum_{k=0}^{n+1} \lambda^{n+1-k} \left(\begin{array}{c} \boldsymbol{d}_{m-1}^f(k-1) \\ \hline \hat{\boldsymbol{x}}_{m-1}(k-1) \end{array} \right) \boldsymbol{d}_m^b(k)^*,$$

and from equation D.12,

$$\hat{\boldsymbol{v}}_m^b(n+1) = \left(\begin{array}{c} \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \boldsymbol{\xi}_1(k-m)^* \\ \hline \sum_{k=0}^{n+1} \lambda^{n+1-k} \hat{\boldsymbol{x}}_{m-1}(k-1) \boldsymbol{d}_m^b(k)^* \end{array} \right). \tag{D.122}$$

Note from equation D.12 that $\boldsymbol{d}_m^b(k) = \boldsymbol{d}_{m-1}^b(k-1)$, and also apply equation D.119 to get

$$\hat{\boldsymbol{v}}_m^b(n+1) = \left(\begin{array}{c} \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \boldsymbol{\xi}_1(k-m)^* \\ \hline \lambda^{n+1} \hat{\boldsymbol{x}}_{m-1}(0) \boldsymbol{d}_{m-1}^b(0)^* + \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_{m-1}(k) \boldsymbol{d}_{m-1}^b(k)^* \end{array} \right),$$

and provided $\hat{\boldsymbol{x}}_{m-1}(0) = \mathbf{0}$ and using equation D.113

$$\hat{\boldsymbol{v}}^b_m(n+1) = \left(\frac{\sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \boldsymbol{\xi}_1(k-m)^*}{\hat{\boldsymbol{v}}^b_{m-1}(n)} \right). \tag{D.123}$$

Substitute equation D.123 into D.121 to get

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|c|c} \boldsymbol{E}^f_{m-1}(n)^* & \hat{\boldsymbol{v}}^f_{m-1}(n)^* & \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \\ & & \boldsymbol{\xi}_1(k-m)^* \\ \hline \hat{\boldsymbol{v}}^f_{m-1}(n) & \hat{\boldsymbol{A}}_{m-1}(n) & \hat{\boldsymbol{v}}^b_{m-1}(n) \\ \hline \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k-m) & \hat{\boldsymbol{v}}^b_{m-1}(n)^* & \boldsymbol{E}^b_m(n+1) \\ \boldsymbol{\xi}_1(k)^* & & \end{array} \right). \tag{D.124}$$

We now perform a similar process to that done in equations D.121–D.124, but this time starting by substituting equation D.117, with m replaced by $m-1$, and n replaced by $n-1$, for $\hat{\boldsymbol{A}}_m(n_1)$ in equation D.120, which results in

$$\hat{\boldsymbol{A}}_{m+1}(n+1) = \left(\begin{array}{c|cc} \boldsymbol{E}^f_m(n)^* & \multicolumn{2}{c}{\hat{\boldsymbol{v}}^f_m(n)^*} \\ \hline \hat{\boldsymbol{v}}^f_m(n) & \hat{\boldsymbol{A}}_{m-1}(n) & \hat{\boldsymbol{v}}^b_{m-1}(n) \\ \hline & \hat{\boldsymbol{v}}^b_{m-1}(n)^* & \boldsymbol{E}^b_{m-1}(n) \end{array} \right). \tag{D.125}$$

From equation D.113

$$\hat{\boldsymbol{v}}^f_m(n) = \sum_{k=0}^{n} \lambda^{n-k} \hat{\boldsymbol{x}}_m(k) \boldsymbol{d}^f_m(k)^*,$$

and substitute equation D.110 to get

$$\hat{\boldsymbol{v}}^f_m(n) = \sum_{k=0}^{n} \lambda^{n-k} \left(\frac{\hat{\boldsymbol{x}}_{m-1}(k)}{\boldsymbol{\xi}_1(k-m+1)} \right) \boldsymbol{d}^f_m(k)^*,$$

and from equation D.12 and D.113,

$$\hat{\boldsymbol{v}}^f_m(n) = \left(\frac{\hat{\boldsymbol{v}}^f_{m-1}(n)}{\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k-m+1) \boldsymbol{\xi}_1(k+1)^*} \right). \tag{D.126}$$

Substitute equation D.126 into equation D.125

$$\hat{\boldsymbol{A}}_{m+1}(n+1) \tag{D.127}$$

$$= \left(\begin{array}{c|c|c} \boldsymbol{E}_m^f(n)^* & \hat{\boldsymbol{v}}_{m-1}^f(n)^* & \sum_{k=0}^{n} \lambda^{n-k}\boldsymbol{\xi}_1(k+1) \times \boldsymbol{\xi}_1(k-m+1)^* \\ \hline \hat{\boldsymbol{v}}_{m-1}^f(n) & \hat{\boldsymbol{A}}_{m-1}(n) & \hat{\boldsymbol{v}}_{m-1}^b(n) \\ \hline \sum_{k=0}^{n} \lambda^{n-k}\boldsymbol{\xi}_1(k-m+1) \times \boldsymbol{\xi}_1(k+1)^* & \hat{\boldsymbol{v}}_{m-1}^b(n)^* & \boldsymbol{E}_{m-1}^b(n) \end{array}\right) \tag{D.128}$$

Note from equations D.5 and D.24 that

$$\boldsymbol{A}_m^q(n) = \hat{\boldsymbol{T}}_{q,m+q-1}\boldsymbol{A}_m^{q-1}(n)\hat{\boldsymbol{T}}_{q,m+q-1}^*, \tag{D.129}$$

and from equations D.5 and D.25 that

$$\boldsymbol{v}_m^{s,q}(n) = \hat{\boldsymbol{T}}_{q,m+q-1}\boldsymbol{v}_m^{s,q-1}(n). \tag{D.130}$$

D.4.1 Forward Prediction Coefficient Update

Define

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1) = \left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\boldsymbol{T}}_{q,m+q-1} \end{array}\right) \times \check{\boldsymbol{A}}_{m+1}^{f,q-1}(n+1) \left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\boldsymbol{T}}_{q,m+q-1}^* \end{array}\right) \tag{D.131}$$

and

$$\check{\boldsymbol{A}}_{m+1}^{f,0}(n+1) = \left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\hat{\boldsymbol{T}}}_{0,m-1} \end{array}\right) \hat{\boldsymbol{A}}_{m+1}(n+1) \left(\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\hat{\boldsymbol{T}}}_{0,m-1}^* \end{array}\right), \tag{D.132}$$

and substituting equation D.120 gives

$$\check{\boldsymbol{A}}_{m+1}^{f,0}(n+1) = \left(\begin{array}{c|c} \boldsymbol{E}_m^f(n) & \boldsymbol{v}_m^f(n)^* \\ \hline \boldsymbol{v}_m^f(n) & \boldsymbol{A}_m(n) \end{array}\right). \tag{D.133}$$

Equations D.133 and D.131 imply

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1) = \left(\begin{array}{c|c} \boldsymbol{E}_m^f(n) & \boldsymbol{v}_m^{f,q}(n)^* \\ \hline \boldsymbol{v}_m^{f,q}(n) & \boldsymbol{A}_m^q(n) \end{array}\right). \tag{D.134}$$

Now, from equation D.134,

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\begin{pmatrix}\boldsymbol{I}\\ \hline -\boldsymbol{c}_m^{f,q}(n)\end{pmatrix}=\begin{pmatrix}\boldsymbol{E}_m^f(n)-\boldsymbol{v}_m^{f,q}(n)^*\boldsymbol{c}_m^{f,q}(n)\\ \hline \boldsymbol{v}_m^{f,q}(n)-\boldsymbol{A}_m^q(n)\boldsymbol{c}_m^{f,q}(n)\end{pmatrix}. \quad \text{(D.135)}$$

Use equation D.23 to substitute for $\boldsymbol{v}_m^{f,q}(n)$ in the lower partition of equation D.135 and use equation D.34 to substitute for $\boldsymbol{E}_m^f(n)$ in the upper partition of equation D.135,

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\begin{pmatrix}\boldsymbol{I}\\ \hline -\boldsymbol{c}_m^{f,q}(n)\end{pmatrix}=\begin{pmatrix}\boldsymbol{\epsilon}_m^{f,q}(n)\\ \hline \boldsymbol{0}\end{pmatrix}. \quad \text{(D.136)}$$

Repartition equations D.131 and D.132 to get

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)=\left(\begin{array}{c|c|c}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\hat{\boldsymbol{T}}_{q,m+q-2}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\boldsymbol{0}&\boldsymbol{T}_{m+q-1}\end{array}\right)\times\check{\boldsymbol{A}}_{m+1}^{f,q-1}(n+1)\left(\begin{array}{c|c|c}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\hat{\boldsymbol{T}}^*_{q,m+q-2}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\boldsymbol{0}&\boldsymbol{T}^*_{m+q-1}\end{array}\right), \quad \text{(D.137)}$$

and

$$\check{\boldsymbol{A}}_{m+1}^{f,0}(n+1)=\left(\begin{array}{c|c|c}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\grave{\hat{\boldsymbol{T}}}_{0,m-2}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\boldsymbol{0}&\grave{\boldsymbol{T}}_{m-1}\end{array}\right)\times\hat{\boldsymbol{A}}_{m+1}(n+1)\left(\begin{array}{c|c|c}\boldsymbol{I}&\boldsymbol{0}&\boldsymbol{0}\\ \hline \boldsymbol{0}&\grave{\hat{\boldsymbol{T}}}_{0,m-2}{}^*&\boldsymbol{0}\\ \hline \boldsymbol{0}&\boldsymbol{0}&\grave{\boldsymbol{T}}_{m-1}{}^*\end{array}\right), \quad \text{(D.138)}$$

and substituting equation D.124 into equation D.138 gives

$$\check{\boldsymbol{A}}_{m+1}^{f,0}(n+1)=\left(\begin{array}{c|c|c}\boldsymbol{A}_{1,1}&\boldsymbol{A}_{1,2}&\boldsymbol{A}_{1,3}\\ \hline \boldsymbol{A}_{2,1}&\boldsymbol{A}_{2,2}&\boldsymbol{A}_{2,3}\\ \hline \boldsymbol{A}_{3,1}&\boldsymbol{A}_{3,2}&\boldsymbol{A}_{3,3}\end{array}\right), \quad \text{(D.139)}$$

where

$$
\begin{aligned}
\boldsymbol{A}_{1,1} &= \boldsymbol{E}^f_{m-1}(n)^* \\
\boldsymbol{A}_{1,2} &= \boldsymbol{v}^f_{m-1}(n)^* \\
\boldsymbol{A}_{1,3} &= \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \boldsymbol{\xi}_1(k-m)^* \dot{\boldsymbol{T}}^*_{m-1} \\
\boldsymbol{A}_{2,1} &= \boldsymbol{v}^f_{m-1}(n) \\
\boldsymbol{A}_{2,2} &= \boldsymbol{A}_{m-1}(n) \\
\boldsymbol{A}_{2,3} &= \boldsymbol{v}^b_{m-1}(n) \dot{\boldsymbol{T}}^*_{m-1} \\
\boldsymbol{A}_{3,1} &= \dot{\boldsymbol{T}}_{m-1} \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k-m) \boldsymbol{\xi}_1(k)^* \\
\boldsymbol{A}_{3,2} &= \dot{\boldsymbol{T}}_{m-1} \boldsymbol{v}^b_{m-1}(n)^* \dot{\boldsymbol{T}}_{m-1} \boldsymbol{E}^b_m(n+1) \dot{\boldsymbol{T}}^*_{m-1}.
\end{aligned}
$$

Equations D.139 and D.137 imply

$$
\check{\boldsymbol{A}}^{f,q}_{m+1}(n+1) = \left(\begin{array}{c|c|c} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3} \\ \hline \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3} \\ \hline \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3} \end{array} \right), \tag{D.140}
$$

where

$$
\begin{aligned}
\boldsymbol{A}_{1,1} &= \boldsymbol{E}^f_{m-1}(n)^* \\
\boldsymbol{A}_{1,2} &= \boldsymbol{v}^{f,q}_{m-1}(n)^* \\
\boldsymbol{A}_{1,3} &= \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k) \boldsymbol{\xi}_1(k-m)^* \dot{\boldsymbol{T}}^*_{m-1} \\
\boldsymbol{A}_{2,1} &= \boldsymbol{v}^{f,q}_{m-1}(n) \\
\boldsymbol{A}_{2,2} &= \boldsymbol{A}^q_{m-1}(n) \\
\boldsymbol{A}_{2,3} &= \boldsymbol{v}^{b,q}_{m-1}(n) \dot{\boldsymbol{T}}^*_{m+q-1} \\
\boldsymbol{A}_{3,1} &= \dot{\boldsymbol{T}}_{m+q-1} \sum_{k=0}^{n+1} \lambda^{n+1-k} \boldsymbol{\xi}_1(k-m) \boldsymbol{\xi}_1(k)^* \\
\boldsymbol{A}_{3,2} &= \dot{\boldsymbol{T}}_{m+q-1} \boldsymbol{v}^{b,q}_{m-1}(n)^* \\
\boldsymbol{A}_{3,3} &= \dot{\boldsymbol{T}}_{m+q-1} \boldsymbol{E}^b_m(n+1) \dot{\boldsymbol{T}}^*_{m+q-1}.
\end{aligned}
$$

Therefore,

$$\tilde{\boldsymbol{A}}^{f,q}_{m+1}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\\hline -\boldsymbol{c}^{b,q}_{m-1}(n)\dot{\boldsymbol{T}}^{*}_{m+q-1}\\\hline \boldsymbol{I}\end{array}\right)$$

$$=\left(\begin{array}{c}\left[-\boldsymbol{v}^{f,q}_{m-1}(n)^{*}\boldsymbol{c}^{b,q}_{m-1}(n)+\sum_{k=0}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^{*}\right]\dot{\boldsymbol{T}}^{*}_{m+q-1}\\\hline \left[-\boldsymbol{A}^{q}_{m-1}(n)\boldsymbol{c}^{b,q}_{m-1}(n)+\boldsymbol{v}^{b,q}_{m-1}(n)\right]\dot{\boldsymbol{T}}^{*}_{m+q-1}\\\hline \dot{\boldsymbol{T}}_{m+q-1}\left[-\boldsymbol{v}^{b,q}_{m-1}(n)^{*}\boldsymbol{c}^{b,q}_{m-1}(n)+\boldsymbol{E}^{b}_{m}(n+1)\right]\dot{\boldsymbol{T}}^{*}_{m+q-1}\end{array}\right). \tag{D.141}$$

We can put equation D.141 into a more tractable form if we define

$$\boldsymbol{k}^{b,q}_{m-1}(n)=\sum_{k=1}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^{*}-\boldsymbol{v}^{f,q}_{m-1}(n)^{*}\boldsymbol{c}^{b,q}_{m-1}(n). \tag{D.142}$$

From equations D.35 and D.12

$$\boldsymbol{E}^{b}_{m}(n+1)=\sum_{k=0}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_1(k-m)\boldsymbol{\xi}_1(k-m)^{*},$$

and using D.119

$$\boldsymbol{E}^{b}_{m}(n+1)=\lambda^{n+1}\boldsymbol{\xi}_1(-m)\boldsymbol{\xi}_1(-m)^{*}+\sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_1(k-m+1)\boldsymbol{\xi}_1(k-m+1)^{*},$$

and provided $\boldsymbol{\xi}_1(n)=0$; for $n<0$,

$$\boldsymbol{E}^{b}_{m}(n+1)=\boldsymbol{E}^{b}_{m-1}(n),$$

and using equation D.34

$$\boldsymbol{E}^{b}_{m}(n+1)=\boldsymbol{\epsilon}^{b,q}_{m-1}(n)+\boldsymbol{v}^{b,q}_{m-1}(n)^{*}\boldsymbol{c}^{b,q}_{m-1}(n). \tag{D.143}$$

Substitute equations D.23, D.142 and D.143 in equation D.141,

$$\tilde{\boldsymbol{A}}^{f,q}_{m+1}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\\hline -\boldsymbol{c}^{b,q}_{m-1}(n)\dot{\boldsymbol{T}}^{*}_{m+q-1}\\\hline \boldsymbol{I}\end{array}\right)=\left(\begin{array}{c}\boldsymbol{k}^{b,q}_{m-1}(n)\dot{\boldsymbol{T}}^{*}_{m+q-1}\\\hline \boldsymbol{0}\\\hline \dot{\boldsymbol{T}}_{m+q-1}\boldsymbol{\epsilon}^{b,q}_{m-1}(n)\dot{\boldsymbol{T}}^{*}_{m+q-1}\end{array}\right). \tag{D.144}$$

Now substitute equation D.128 into equation D.138 to get

$$\tilde{\boldsymbol{A}}^{f,0}_{m+1}(n+1)=\left(\begin{array}{c|c|c}\boldsymbol{A}_{1,1}&\boldsymbol{A}_{1,2}&\boldsymbol{A}_{1,3}\\\hline \boldsymbol{A}_{2,1}&\boldsymbol{A}_{2,2}&\boldsymbol{A}_{2,3}\\\hline \boldsymbol{A}_{3,1}&\boldsymbol{A}_{3,2}&\boldsymbol{A}_{3,3}\end{array}\right), \tag{D.145}$$

where

$$
\begin{aligned}
\boldsymbol{A}_{1,1} &= \boldsymbol{E}_m^f(n)^* \\
\boldsymbol{A}_{1,2} &= \boldsymbol{v}_{m-1}^f(n)^* \\
\boldsymbol{A}_{1,3} &= \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k+1)\boldsymbol{\xi}_1(k-m+1)^* \grave{\boldsymbol{T}}_{m-1}^* \\
\boldsymbol{A}_{2,1} &= \boldsymbol{v}_{m-1}^f(n) \\
\boldsymbol{A}_{2,2} &= \boldsymbol{A}_{m-1}(n) \\
\boldsymbol{A}_{2,3} &= \boldsymbol{v}_{m-1}^b(n) \grave{\boldsymbol{T}}_{m-1}^* \\
\boldsymbol{A}_{3,1} &= \grave{\boldsymbol{T}}_{m-1} \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k-m+1)\boldsymbol{\xi}_1(k+1)^* \\
\boldsymbol{A}_{3,2} &= \grave{\boldsymbol{T}}_{m-1} \boldsymbol{v}_{m-1}^b(n)^* \\
\boldsymbol{A}_{3,3} &= \grave{\boldsymbol{T}}_{m-1} \boldsymbol{E}_{m-1}^b(n) \grave{\boldsymbol{T}}_{m-1}^*.
\end{aligned}
$$

Equations D.145 and D.137 imply

$$
\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1) \left(\begin{array}{c|c|c} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3} \\ \hline \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3} \\ \hline \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3} \end{array} \right), \tag{D.146}
$$

where

$$
\begin{aligned}
\boldsymbol{A}_{1,1} &= \boldsymbol{E}_m^f(n)^* \\
\boldsymbol{A}_{1,2} &= \boldsymbol{v}_{m-1}^{f,q}(n)^* \\
\boldsymbol{A}_{1,3} &= \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k+1)\boldsymbol{\xi}_1(k-m+1)^* \grave{\boldsymbol{T}}_{m+q-1}^* \\
\boldsymbol{A}_{2,1} &= \boldsymbol{v}_{m-1}^{f,q}(n) \\
\boldsymbol{A}_{2,2} &= \boldsymbol{A}_{m-1}^q(n) \\
\boldsymbol{A}_{2,3} &= \boldsymbol{v}_{m-1}^{b,q}(n) \grave{\boldsymbol{T}}_{m+q-1}^* \\
\boldsymbol{A}_{3,1} &= \grave{\boldsymbol{T}}_{m+q-1} \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k-m+1)\boldsymbol{\xi}_1(k+1)^* \grave{\boldsymbol{T}}_{m+q-1}^* \\
\boldsymbol{A}_{3,2} &= \grave{\boldsymbol{T}}_{m+q-1} \boldsymbol{v}_{m-1}^{b,q}(n)^* \\
\boldsymbol{A}_{3,3} &= \grave{\boldsymbol{T}}_{m+q-1} \boldsymbol{E}_{m-1}^b(n) \grave{\boldsymbol{T}}_{m+q-1}^*.
\end{aligned}
$$

Therefore,

$$\tilde{\boldsymbol{A}}^{f,q}_{m+1}(n+1)\left(\begin{array}{c}\boldsymbol{I}\\ \hline -\boldsymbol{c}^{f,q}_{m-1}(n)\\ \hline \boldsymbol{0}\end{array}\right) = \left(\begin{array}{c}\boldsymbol{E}^{f}_{m}(n)^{*} - \boldsymbol{v}^{f,q}_{m-1}(n)^{*}\boldsymbol{c}^{f,q}_{m-1}(n)\\ \hline \boldsymbol{v}^{f,q}_{m-1}(n) - \boldsymbol{A}^{q}_{m-1}(n)\boldsymbol{c}^{f,q}_{m-1}(n)\\ \hline \dot{\boldsymbol{T}}_{m+q-1}\Big[\sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_{m}(k-m+1)\boldsymbol{\xi}_{1}(k+1)^{*}\\ \qquad - \boldsymbol{v}^{b,q}_{m-1}(n)^{*}\boldsymbol{c}^{f,q}_{m-1}(n)\Big]\end{array}\right). \tag{D.147}$$

Define

$$\boldsymbol{k}^{f,q}_{m-1}(n) = \sum_{k=1}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_{1}(k-m)\boldsymbol{\xi}_{1}(k)^{*} - \boldsymbol{v}^{b,q}_{m-1}(n)^{*}\boldsymbol{c}^{f,q}_{m-1}(n). \tag{D.148}$$

Also,

$$\sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_{1}(k-m+1)\boldsymbol{\xi}_{1}(k+1)^{*} = \sum_{l=1}^{n+1}\lambda^{n+1-l}\boldsymbol{\xi}_{1}(l-m)\boldsymbol{\xi}_{1}(l)^{*}$$

by putting $l = k+1$. From D.148,

$$\sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_{1}(k-m+1)\boldsymbol{\xi}_{1}(k+1)^{*} = \boldsymbol{k}^{f,q}_{m-1}(n) + \boldsymbol{v}^{b,q}_{m-1}(n)^{*}\boldsymbol{c}^{f,q}_{m-1}(n). \tag{D.149}$$

From equation D.35,

$$\boldsymbol{E}^{f}_{m}(n) = \sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{d}^{f}_{m}(k)\boldsymbol{d}^{f}_{m}(k)^{*},$$

and since $\boldsymbol{d}^{f}_{m}(n) - \boldsymbol{d}^{f}_{m-1}(n)$ (equation D.12),

$$\boldsymbol{E}^{f}_{m}(n) = \boldsymbol{E}^{f}_{m-1}(n),$$

and from equation D.33,

$$\boldsymbol{E}^{f}_{m}(n) = \boldsymbol{\epsilon}^{f,q}_{m-1}(n) + \boldsymbol{v}^{f,q}_{m-1}(n)^{*}\boldsymbol{c}^{f,q}_{m-1}(n). \tag{D.150}$$

Substitute equations D.23, D.149 and D.150 in equation D.147 to get

$$\tilde{\boldsymbol{A}}^{f,q}_{m+1}(n+1)\left(\begin{array}{c}\boldsymbol{I}\\ \hline -\boldsymbol{c}^{f,q}_{m-1}(n)\\ \hline \boldsymbol{0}\end{array}\right) = \left(\begin{array}{c}\boldsymbol{\epsilon}^{f,q}_{m-1}\\ \hline \boldsymbol{0}\\ \hline \dot{\boldsymbol{T}}_{m+q-1}\boldsymbol{k}^{f,q}_{m-1}(n)\end{array}\right). \tag{D.151}$$

In order to manipulate equations D.144 and D.151 more easily, define

$$\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n) = \grave{\boldsymbol{T}}_{m+q-1}\boldsymbol{\epsilon}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}_{m+q-1}^{*}, \tag{D.152}$$

$$\hat{\boldsymbol{k}}^{f,q}(n) = \grave{\boldsymbol{T}}_{m+q-1}\boldsymbol{k}_{m-1}^{f,q}(n) \tag{D.153}$$

and

$$\hat{\boldsymbol{k}}^{b,q}(n) = \boldsymbol{k}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}_{m+q-1}^{*}, \tag{D.154}$$

and rewrite equation D.144,

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\ \hline -\boldsymbol{c}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}_{m+q-1}^{*}\\ \hline \boldsymbol{I}\end{array}\right) = \left(\begin{array}{c}\hat{\boldsymbol{k}}_{m-1}^{b,q}(n)\\ \hline \boldsymbol{0}\\ \hline \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)\end{array}\right), \tag{D.155}$$

and equation D.151,

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\left(\begin{array}{c}\boldsymbol{I}\\ \hline -\boldsymbol{c}_{m-1}^{f,q}(n)\\ \hline \boldsymbol{0}\end{array}\right) = \left(\begin{array}{c}\boldsymbol{\epsilon}_{m-1}^{f,q}(n)\\ \hline \boldsymbol{0}\\ \hline \hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\end{array}\right). \tag{D.156}$$

Multiply equation D.155 by $\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)$

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\ \hline -\boldsymbol{c}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}_{m+q-1}^{*}\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\\ \hline \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\end{array}\right)$$

$$= \left(\begin{array}{c}\hat{\boldsymbol{k}}_{m-1}^{b,q}(n)\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\\ \hline \boldsymbol{0}\\ \hline \hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\end{array}\right). \tag{D.157}$$

Subtract equation D.157 from equation D.156

$$\check{\boldsymbol{A}}_{m+1}^{f,q}(n+1)\left(\begin{array}{c}\boldsymbol{I}\\ \hline -\boldsymbol{c}_{m-1}^{f,q}(n)+\boldsymbol{c}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}_{m+q-1}^{*}\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\\ \hline -\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\end{array}\right)$$

$$= \left(\begin{array}{c}\boldsymbol{\epsilon}_{m-1}^{f,q}(n)-\hat{\boldsymbol{k}}_{m-1}^{b,q}(n)\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n)\\ \hline \boldsymbol{0}\\ \hline \boldsymbol{0}\end{array}\right). \tag{D.158}$$

We can use equations D.158 and D.136 to produce order recursive relationships for $\epsilon^{f,q}$ and $\boldsymbol{c}^{f,q}$. Note that these equations are of the form

$$\left(\begin{array}{c|c} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \hline \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} \end{array}\right)\left(\begin{array}{c} \boldsymbol{I} \\ \hline \boldsymbol{X}_1 \end{array}\right) = \left(\begin{array}{c} \boldsymbol{Y}_1 \\ \hline \boldsymbol{0} \end{array}\right) \tag{D.159}$$

and

$$\left(\begin{array}{c|c} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \hline \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} \end{array}\right)\left(\begin{array}{c} \boldsymbol{I} \\ \hline \boldsymbol{X}_2 \end{array}\right) = \left(\begin{array}{c} \boldsymbol{Y}_2 \\ \hline \boldsymbol{0} \end{array}\right), \tag{D.160}$$

which can be expanded into 4 equations

$$\boldsymbol{A}_{1,1} + \boldsymbol{A}_{1,2}\boldsymbol{X}_1 = \boldsymbol{Y}_1, \tag{D.161}$$

$$\boldsymbol{A}_{2,1} + \boldsymbol{A}_{2,2}\boldsymbol{X}_1 = \boldsymbol{0}, \tag{D.162}$$

$$\boldsymbol{A}_{1,1} + \boldsymbol{A}_{1,2}\boldsymbol{X}_2 = \boldsymbol{Y}_2 \tag{D.163}$$

and

$$\boldsymbol{A}_{2,1} + \boldsymbol{A}_{2,2}\boldsymbol{X}_2 = \boldsymbol{0}. \tag{D.164}$$

Subtract equation D.162 from equation D.164 to get

$$\boldsymbol{A}_{2,2}\left(\boldsymbol{X}_2 - \boldsymbol{X}_1\right) = \boldsymbol{0},$$

which, provided $\boldsymbol{A}_{2,2}$ is non-singular, implies

$$\boldsymbol{X}_2 = \boldsymbol{X}_1. \tag{D.165}$$

Now subtract equation D.161 from equation D.163 to get

$$\boldsymbol{A}_{1,2}\left(\boldsymbol{X}_2 - \boldsymbol{X}_1\right) = \boldsymbol{Y}_2 - \boldsymbol{Y}_1,$$

and substituting equation D.165 yields

$$\boldsymbol{Y}_2 = \boldsymbol{Y}_1. \tag{D.166}$$

Equations D.165 and D.166 show that we can equate the left and right halves of equations D.158 and D.136 to get

$$\boldsymbol{c}_m^{f,q}(n) = \left(\begin{array}{c} \boldsymbol{c}_{m-1}^{f,q}(n) - \boldsymbol{c}_{m-1}^{b,q}(n)\grave{\boldsymbol{T}}^{*}_{m+q-1}\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n) \\ \hline \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n) \end{array}\right) \tag{D.167}$$

and

$$\epsilon_m^{f,q}(n) = \epsilon_{m-1}^{f,q}(n) - \hat{\boldsymbol{k}}_{m-1}^{b,q}(n)\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1}\hat{\boldsymbol{k}}_{m-1}^{f,q}(n). \tag{D.168}$$

D.4.2 Backward Prediction Coefficient Update

The backward prediction coefficients can be derived similarly to the forward prediction coefficients. Define

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) = \left(\begin{array}{c|c} \hat{\boldsymbol{T}}_{q,m+q-1} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right) \check{\boldsymbol{A}}_{m+1}^{b,q-1}(n+1) \left(\begin{array}{c|c} \hat{\boldsymbol{T}}_{q,m+q-1}^{*} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right) \tag{D.169}$$

and

$$\check{\boldsymbol{A}}_{m+1}^{b,0}(n+1) = \left(\begin{array}{c|c} \hat{\hat{\boldsymbol{T}}}_{0,m-1} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right) \hat{\boldsymbol{A}}_{m+1}(n+1) \left(\begin{array}{c|c} \hat{\hat{\boldsymbol{T}}}_{0,m-1}^{*} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right), \tag{D.170}$$

and substituting equation D.117 gives

$$\check{\boldsymbol{A}}_{m+1}^{b,0}(n+1) = \left(\begin{array}{c|c} \boldsymbol{A}_m(n+1) & \boldsymbol{v}_m^b(n+1) \\ \hline \boldsymbol{v}_m^b(n+1)^* & \boldsymbol{E}_m^b(n+1) \end{array}\right). \tag{D.171}$$

Equations D.171 and D.169 imply

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) = \left(\begin{array}{c|c} \boldsymbol{A}_m^q(n+1) & \boldsymbol{v}_m^{b,q}(n+1) \\ \hline \boldsymbol{v}_m^{b,q}(n+1)^* & \boldsymbol{E}_m^b(n+1) \end{array}\right). \tag{D.172}$$

Post-multiplication of equation D.172 yields

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} -\boldsymbol{c}_m^{b,q}(n+1) \\ \hline \boldsymbol{I} \end{array}\right) = \left(\begin{array}{c} -\boldsymbol{A}_m^q(n+1)\boldsymbol{c}_m^{b,q}(n+1) \\ +\boldsymbol{v}_m^{b,q}(n+1) \\ \hline -\boldsymbol{v}_m^{b,q}(n+1)^*\boldsymbol{c}_m^{b,q}(n+1) \\ +\boldsymbol{E}_m^b(n+1) \end{array}\right),$$

and using equation D.23,

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} -\boldsymbol{c}_m^{b,q}(n+1) \\ \hline \boldsymbol{I} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{0} \\ \hline \boldsymbol{E}_m^b(n+1) \\ -\boldsymbol{v}_m^{b,q}(n+1)^*\boldsymbol{c}_m^{b,q}(n+1) \end{array}\right),$$

and from D.34,

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} -\boldsymbol{c}_m^{b,q}(n+1) \\ \hline \boldsymbol{I} \end{array}\right) = \left(\begin{array}{c} \boldsymbol{0} \\ \hline \epsilon_m^{b,q}(n+1) \end{array}\right). \tag{D.173}$$

Repartition equations D.169 and D.170 to get

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) = \left(\begin{array}{c|c|c} \boldsymbol{T}_q & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\boldsymbol{T}}_{q+1,m+q-1} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right) \times \check{\boldsymbol{A}}_{m+1}^{b,q-1}(n+1) \left(\begin{array}{c|c|c} \boldsymbol{T}_q^* & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\boldsymbol{T}}_{q+1,m+q-1}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right), \quad \text{(D.174)}$$

and

$$\check{\boldsymbol{A}}_{m+1}^{b,0}(n+1) = \left(\begin{array}{c|c|c} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\hat{\boldsymbol{T}}}_{1,m-1} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right) \hat{\boldsymbol{A}}_{m+1}(n+1) \left(\begin{array}{c|c|c} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \hat{\hat{\boldsymbol{T}}}_{1,m-1}^* & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right). \quad \text{(D.175)}$$

Note, from equations D.7 and D.68, that $\hat{\hat{\boldsymbol{T}}}_{1,m-1} = \hat{\boldsymbol{T}}_{1,m-1}\hat{\hat{\boldsymbol{T}}}_{0,m-2}$ and substitute equation D.124 in equation D.175 to get

$$\check{\boldsymbol{A}}_{m+1}^{b,0}(n+1) = \left(\begin{array}{c|c|c} \boldsymbol{E}_{m-1}^{f}(n)^* & \boldsymbol{v}_{m-1}^{f,1}(n)^* & \sum_{k=0}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k) \times \boldsymbol{\xi}_1(k-m)^* \\ \hline \boldsymbol{v}_{m-1}^{f,1}(n) & \boldsymbol{A}_{m-1}^{1}(n) & \boldsymbol{v}_{m-1}^{b,1}(n) \\ \hline \sum_{k=0}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k-m) \times \boldsymbol{\xi}_1(k)^* & \boldsymbol{v}_{m-1}^{b,1}(n)^* & \boldsymbol{E}_{m}^{b}(n+1) \end{array}\right). \quad \text{(D.176)}$$

Equations D.176 and D.174 imply

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) = \left(\begin{array}{c|c|c} \dot{\boldsymbol{T}}_q \boldsymbol{E}_{m-1}^{f}(n)^* \dot{\boldsymbol{T}}_q^* & \dot{\boldsymbol{T}}_q \boldsymbol{v}_{m-1}^{f,q+1}(n)^* & \dot{\boldsymbol{T}}_q \sum_{k=0}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k) \times \boldsymbol{\xi}_1(k-m)^* \\ \hline \boldsymbol{v}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* & \boldsymbol{A}_{m-1}^{q+1}(n) & \boldsymbol{v}_{m-1}^{b,q+1}(n) \\ \hline \sum_{k=0}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k-m) \times \boldsymbol{\xi}_1(k)^* \dot{\boldsymbol{T}}_q^* & \boldsymbol{v}_{m-1}^{b,q+1}(n)^* & \boldsymbol{E}_{m}^{b}(n+1) \end{array}\right). \quad \text{(D.177)}$$

Therefore

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\\hline -\boldsymbol{c}_{m-1}^{b,q+1}(n)\\\hline \boldsymbol{I}\end{array}\right)$$
$$=\left(\begin{array}{c}\dot{\boldsymbol{T}}_q\left[-\boldsymbol{v}_{m-1}^{f,q+1}(n)^*\boldsymbol{c}_{m-1}^{b,q+1}(n)+\sum_{k=0}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^*\right]\\\hline -\boldsymbol{A}_{m-1}^{q+1}(n)\boldsymbol{c}_{m-1}^{b,q+1}(n)+\boldsymbol{v}_{m-1}^{b,q+1}(n)\\\hline -\boldsymbol{v}_{m-1}^{b,q+1}(n)^*\boldsymbol{c}_{m-1}^{b,q+1}(n)+\boldsymbol{E}_m^b(n+1)\end{array}\right). \tag{D.178}$$

Substitute equations D.23, D.142 and D.143 in equation D.178 to get,

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)\left(\begin{array}{c}\boldsymbol{0}\\\hline -\boldsymbol{c}_{m-1}^{b,q+1}(n)\\\hline \boldsymbol{I}\end{array}\right)=\left(\begin{array}{c}\dot{\boldsymbol{T}}_q\boldsymbol{k}_{m-1}^{b,q+1}(n)\\\hline \boldsymbol{0}\\\hline \epsilon_{m-1}^{b,q+1}(n)\end{array}\right). \tag{D.179}$$

Now substitute equation D.128 into equation D.175 to get

$$\check{\boldsymbol{A}}_{m+1}^{b,0}(n+1)$$
$$=\left(\begin{array}{c|c|c}\boldsymbol{E}_m^f(n)^* & \boldsymbol{v}_{m-1}^{f,1}(n)^* & \sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_1(k+1)\times\boldsymbol{\xi}_1(k-m+1)^*\\\hline \boldsymbol{v}_{m-1}^{f,1}(n) & \boldsymbol{A}_{m-1}^{1}(n) & \boldsymbol{v}_{m-1}^{b,1}(n)\\\hline \sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_1(k-m+1)\times\boldsymbol{\xi}_1(k+1)^* & \boldsymbol{v}_{m-1}^{b,1}(n)^* & \boldsymbol{E}_{m-1}^b(n)\end{array}\right). \tag{D.180}$$

Equations D.180 and D.174 imply

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)=\left(\begin{array}{c|c|c}\boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \boldsymbol{A}_{1,3}\\\hline \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \boldsymbol{A}_{2,3}\\\hline \boldsymbol{A}_{3,1} & \boldsymbol{A}_{3,2} & \boldsymbol{A}_{3,3}\end{array}\right), \tag{D.181}$$

where

$$\boldsymbol{A}_{1,1}=\dot{\boldsymbol{T}}_q\boldsymbol{E}_m^f(n)^*\dot{\boldsymbol{T}}_q^*$$
$$\boldsymbol{A}_{1,2}=\dot{\boldsymbol{T}}_q\boldsymbol{v}_{m-1}^{f,q+1}(n)^*$$
$$\boldsymbol{A}_{1,3}=\dot{\boldsymbol{T}}_q\sum_{k=0}^{n}\lambda^{n-k}\boldsymbol{\xi}_1(k+1)\boldsymbol{\xi}_1(k-m+1)^*$$
$$\boldsymbol{A}_{2,1}=\boldsymbol{v}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^*$$
$$\boldsymbol{A}_{2,2}=\boldsymbol{A}_{m-1}^{q+1}(n)$$

$$\begin{aligned}
\boldsymbol{A}_{2,3} &= \boldsymbol{v}_{m-1}^{b,q+1}(n) \\
\boldsymbol{A}_{3,1} &= \sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_1(k-m+1)\boldsymbol{\xi}_1(k+1)^* \dot{\boldsymbol{T}}_q^* \\
\boldsymbol{A}_{3,2} &= \boldsymbol{v}_{m-1}^{b,q+1}(n)^* \\
\boldsymbol{A}_{3,3} &= \boldsymbol{E}_{m-1}^{b}(n).
\end{aligned}$$

Therefore

$$\begin{aligned}
&\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} \boldsymbol{I} \\ \hline -\boldsymbol{c}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* \\ \hline \boldsymbol{0} \end{array} \right) \\
&\qquad = \left(\begin{array}{c} \dot{\boldsymbol{T}}_q \left[\boldsymbol{E}_m^f(n)^* - \boldsymbol{v}_{m-1}^{f,q+1}(n)^* \boldsymbol{c}_{m-1}^{f,q+1}(n) \right] \dot{\boldsymbol{T}}_q^* \\ \hline \left[\boldsymbol{v}_{m-1}^{f,q+1}(n) - \boldsymbol{A}_{m-1}^{q+1}(n)\boldsymbol{c}_{m-1}^{f,q+1}(n) \right] \dot{\boldsymbol{T}}_q^* \\ \hline \left[\sum_{k=0}^{n} \lambda^{n-k} \boldsymbol{\xi}_m(k-m+1)\boldsymbol{\xi}_1(k+1)^* \right. \\ \qquad \left. -\boldsymbol{v}_{m-1}^{b,q+1}(n)^* \boldsymbol{c}_{m-1}^{f,q+1}(n) \right] \dot{\boldsymbol{T}}_q^* \end{array} \right).
\end{aligned} \tag{D.182}$$

Substitute equations D.23, D.149 and D.150 in equation D.182 to get

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} \boldsymbol{I} \\ \hline -\boldsymbol{c}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* \\ \hline \boldsymbol{0} \end{array} \right) = \left(\begin{array}{c} \dot{\boldsymbol{T}}_q \boldsymbol{\epsilon}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* \\ \hline \boldsymbol{0} \\ \hline \boldsymbol{k}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* \end{array} \right). \tag{D.183}$$

In order to manipulate equations D.179 and D.183 more easily define

$$\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n) = \dot{\boldsymbol{T}}_q \boldsymbol{\epsilon}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^*, \tag{D.184}$$

$$\hat{\hat{\boldsymbol{k}}}_{m-1}^{f,q+1}(n) = \boldsymbol{k}_{m-1}^{f,q+1}(n)\dot{\boldsymbol{T}}_q^* \tag{D.185}$$

and

$$\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n) = \dot{\boldsymbol{T}}_q \boldsymbol{k}_{m-1}^{b,q+1}(n). \tag{D.186}$$

Using equations D.184, D.185 and D.186, rewrite equation D.179

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1) \left(\begin{array}{c} \boldsymbol{0} \\ \hline -\boldsymbol{c}_{m-1}^{b,q+1}(n) \\ \hline \boldsymbol{I} \end{array} \right) = \left(\begin{array}{c} \hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n) \\ \hline \boldsymbol{0} \\ \hline \boldsymbol{\epsilon}_{m-1}^{b,q+1}(n) \end{array} \right) \tag{D.187}$$

and rewrite equation D.183,

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)\left(\begin{array}{c}\boldsymbol{I}\\ \hline -\boldsymbol{c}_{m-1}^{f,q+1}(n)\grave{\boldsymbol{T}}_q^{*}\\ \hline \boldsymbol{0}\end{array}\right)=\left(\begin{array}{c}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)\\ \hline \boldsymbol{0}\\ \hline \hat{\hat{\boldsymbol{k}}}_{m-1}^{f,q+1}(n)\end{array}\right). \tag{D.188}$$

Multiply equation D.188 by $\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)$ to get

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)\left(\begin{array}{c}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\\ \hline -\boldsymbol{c}_{m-1}^{f,q+1}(n)\grave{\boldsymbol{T}}_q^{*}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\\ \hline \boldsymbol{I}\end{array}\right)$$

$$=\left(\begin{array}{c}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\\ \hline \boldsymbol{0}\\ \hline \hat{\hat{\boldsymbol{k}}}_{m-1}^{f,q+1}(n)\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\end{array}\right), \tag{D.189}$$

and subtract equation D.189 from equation D.187 to get

$$\check{\boldsymbol{A}}_{m+1}^{b,q}(n+1)\left(\begin{array}{c}-\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\\ \hline -\boldsymbol{c}_{m-1}^{b,q+1}(n)+\boldsymbol{c}_{m-1}^{f,q+1}(n)\grave{\boldsymbol{T}}_q^{*}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\\ \hline \boldsymbol{I}\end{array}\right)$$

$$=\left(\begin{array}{c}\boldsymbol{0}\\ \hline \boldsymbol{0}\\ \hline \boldsymbol{\epsilon}_{m-1}^{b,q+1}(n)-\hat{\hat{\boldsymbol{k}}}_{m-1}^{f,q+1}(n)\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n)\end{array}\right). \tag{D.190}$$

Equations D.190 and D.173 are of similar form to equations D.158 and D.136 and we can equate the left and right hand sides of equations D.190 and D.173 to yield two new equations,

$$\boldsymbol{c}_m^{b,q+1}(n+1)=\left(\begin{array}{c}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q}(n)\\ \hline \boldsymbol{c}_{m-1}^{b,q+1}(n)-\boldsymbol{c}_{m-1}^{f,q+1}(n)\grave{\boldsymbol{T}}_q^{*}\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q}(n)\end{array}\right) \tag{D.191}$$

and

$$\boldsymbol{\epsilon}_m^{b,q}(n+1)=\boldsymbol{\epsilon}_{m-1}^{b,q+1}(n)-\hat{\hat{\boldsymbol{k}}}_{m-1}^{f,q+1}(n)\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n). \tag{D.192}$$

Equations D.167 and D.191 are order update equations for the forward and backward predictor coefficients. Equations D.192 and D.168 are used to update $\epsilon^{f,q}$ and $\epsilon^{b,q}$.

D.5 Error Update

Using equations D.1, D.12, D.3 and D.5,

$$\boldsymbol{x}_m^q(n+1) = \left(\frac{\boldsymbol{x}_{m-1}^q(n+1)}{\grave{\boldsymbol{T}}_{m+q-1}\boldsymbol{d}_{m+q-1}^b(n+1)} \right). \tag{D.193}$$

Pre-multiply equation D.167 by equation D.193 to get

$$\begin{aligned}\boldsymbol{x}_m^q(n+1)^* \boldsymbol{c}_m^{f,q}(n) = \boldsymbol{x}_{m-1}^q(n+1)^* \boldsymbol{c}_{m-1}^{f,q}(n) \\ - \left[\boldsymbol{x}_{m-1}^q(n+1)^* \boldsymbol{c}_{m-1}^{b,q}(n) - \boldsymbol{d}_{m-1}^b(n+1)\right] \\ \times \grave{\boldsymbol{T}}_{m+q-1}^* \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1} \hat{\boldsymbol{k}}_{m-1}^{f,q}(n),\end{aligned}$$

and from equation D.14,

$$\begin{aligned}\boldsymbol{x}_m^q(n+1)^* \boldsymbol{c}_m^{f,q}(n) = \boldsymbol{x}_{m-1}^q(n+1)^* \boldsymbol{c}_{m-1}^{f,q}(n) \\ + \boldsymbol{e}_{m-1}^{b,q}(n+1)^* \grave{\boldsymbol{T}}_{m+q-1}^* \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1} \hat{\boldsymbol{k}}_{m-1}^{f,q}(n),\end{aligned}$$

and applying equation D.14 again,

$$\begin{aligned}\boldsymbol{d}_m^f(n+1)^* - \boldsymbol{e}_m^{f,q}(n+1)^* = \boldsymbol{d}_{m-1}(n+1)^* - \boldsymbol{e}_{m-1}(n+1)^* \\ + \boldsymbol{e}_{m-1}^{b,q}(n+1)^* \grave{\boldsymbol{T}}_{m+q-1}^* \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1} \hat{\boldsymbol{k}}_{m-1}^{f,q}(n),\end{aligned} \tag{D.194}$$

and $\boldsymbol{d}_m^f(n+1) = \boldsymbol{d}_{m-1}^f(n+1)$ from equation D.12,

$$\boldsymbol{e}_m^{f,q}(n+1)^* = \boldsymbol{e}_{m-1}^{f,q}(n+1)^* - \boldsymbol{e}_{m-1}^{b,q}(n+1)^* \grave{\boldsymbol{T}}_{m+q-1}^* \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1} \hat{\boldsymbol{k}}_{m-1}^{f,q}(n). \tag{D.195}$$

Similarly, use equations D.1, D.12, D.3 and D.5 get

$$\boldsymbol{x}_m^q(n+2) = \left(\frac{\grave{\boldsymbol{T}}_q \boldsymbol{d}_{m-1}^f(n+1)}{\boldsymbol{x}_{m-1}^{q+1}(n+1)} \right), \tag{D.196}$$

and pre-multiply equation D.191 by equation D.196 to get

$$\begin{aligned}\boldsymbol{x}_m^q(n+2)^*\boldsymbol{c}_m^{b,q}(n+1) = \boldsymbol{x}_{m-1}^{q+1}(n+1)^*\boldsymbol{c}_{m-1}^{b,q+1}(n) \\ - \left[\boldsymbol{x}_{m-1}^{q+1}(n+1)^*\boldsymbol{c}_{m-1}^{f,q+1}(n) - \boldsymbol{d}_{m-1}^{f}(n+1)^*\right] \\ \times \grave{\boldsymbol{T}}_q^*\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n),\end{aligned}$$

and applying equation D.14 we get

$$\begin{aligned}\boldsymbol{d}_m^b(n+2)^* - \boldsymbol{e}_m^{b,q}(n+2)^* = \boldsymbol{d}_{m-1}^b(n+1)^* - \boldsymbol{e}_{m-1}^{b,q+1}(n+1)^* \\ + \boldsymbol{e}_{m-1}^{f,q+1}(n+1)^*\grave{\boldsymbol{T}}_q^*\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n),\end{aligned}$$

and since $\boldsymbol{d}_m^b(n+2) = \boldsymbol{d}_{m-1}^b(n+1)$ (from equation D.12),

$$\begin{aligned}\boldsymbol{e}_m^{b,q}(n+2)^* = \boldsymbol{e}_{m-1}^{b,q+1}(n+1)^* \\ - \boldsymbol{e}_{m-1}^{f,q+1}(n+1)^*\grave{\boldsymbol{T}}_q^*\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n)^{-1}\hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n).\end{aligned} \quad \text{(D.197)}$$

D.6 Auxiliary Variables Update

We have introduced the Auxiliary variables $\boldsymbol{k}^{b,q}$, $\boldsymbol{k}^{f,q}$, which can be used to calculate the other quantities, but have no particular physical significance. We need to derive some relationships which will allow these variables to be efficiently calculated.

Take the conjugate-transpose of equation D.45 and substitute in equation D.148,

$$\begin{aligned}\boldsymbol{k}_{m-1}^{f,q}(n) &= \sum_{k=1}^{n+1}\lambda^{n+1-k}\boldsymbol{\xi}_1(k-m)\boldsymbol{\xi}_1(k)^* - \lambda\boldsymbol{v}_{m-1}^{b,q}(n-1)^*\boldsymbol{c}_{m-1}^{f,q}(n-1) \\ &\quad + [1-\gamma_{m-1}(n)]\,\boldsymbol{e}_{m-1}^{b,q}(n)\boldsymbol{e}_{m-1}^{f,q}(n)^* - \boldsymbol{d}_{m-1}^b(n)\boldsymbol{d}_{m-1}^f(n)^* \\ &= \lambda\left(\sum_{k=1}^{n}\lambda^{n-k}\boldsymbol{\xi}_1(k-m)\boldsymbol{\xi}_1(k)^* - \boldsymbol{v}_{m-1}^{b,q}(n-1)^*\boldsymbol{c}_{m-1}^{f,q}(n-1)\right) \\ &\quad + \boldsymbol{\xi}_1(n+1-m)\boldsymbol{\xi}_1(n+1)^* + [1-\gamma_{m-1}(n)]\,\boldsymbol{e}_{m-1}^{b,q}(n)\boldsymbol{e}_{m-1}^{f,q}(n)^* \\ &\quad - \boldsymbol{d}_{m-1}^b(n)\boldsymbol{d}_{m-1}^f(n)^*\end{aligned}$$

$$= \lambda \boldsymbol{k}^{f,q}_{m-1}(n-1) + [1 - \gamma^q_{m-1}(n)]\, \boldsymbol{e}^{b,q}_{m-1}(n)\boldsymbol{e}^{f,q}_{m-1}(n)^* + \boldsymbol{\xi}_1(n+1-m)\boldsymbol{\xi}_1(n+1)^* - \boldsymbol{d}^b_{m-1}(n)\boldsymbol{d}^f_{m-1}(n)^*,$$

and from equation D.12,

$$\begin{aligned}\boldsymbol{k}^{f,q}_{m-1}(n) &= \lambda \boldsymbol{k}^{f,q}_{m-1}(n-1) + [1 - \gamma^q_{m-1}(n)]\, \boldsymbol{e}^{b,q}_{m-1}(n)\boldsymbol{e}^{f,q}_{m-1}(n)^* \\ &\quad + \boldsymbol{\xi}_1(n+1-m)\boldsymbol{\xi}_1(n+1)^* - \boldsymbol{\xi}_1(n+1-m)\boldsymbol{\xi}_1(n+1)^* \\ &= \lambda \boldsymbol{k}^{f,q}_{m-1}(n-1) + [1 - \gamma^q_{m-1}(n)]\, \boldsymbol{e}^{b,q}_{m-1}(n)\boldsymbol{e}^{f,q}_{m-1}(n)^*. \qquad \text{(D.198)}\end{aligned}$$

Equation D.198 is the recursive form we require for $\boldsymbol{k}^{f,q}$.

We will now show that $\boldsymbol{k}^{b,q}_m$ is the conjugate transpose of $\boldsymbol{k}^{f,q}_m$. Taking the conjugate-transpose of equation D.148,

$$\boldsymbol{k}^{f,q}_{m-1}(n)^* = \sum_{k=1}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^* - \boldsymbol{c}^{f,q}_{m-1}(n)^* \boldsymbol{v}^{b,q}_{m-1}(n),$$

applying equation D.23,

$$\boldsymbol{k}^{f,q}_{m-1}(n)^* = \sum_{k=1}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^* - \boldsymbol{c}^{f,q}_{m-1}(n)^* \boldsymbol{A}^q_{m-1}(n)\boldsymbol{c}^{b,q}_{m-1}(n),$$

and since $\boldsymbol{A}$ is Hermitian,

$$\boldsymbol{k}^{f,q}_{m-1}(n)^* = \sum_{k=1}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^* - \left[\boldsymbol{A}^q_{m-1}(n)\boldsymbol{c}^{f,q}_{m-1}(n)\right]^* \boldsymbol{c}^{b,q}_{m-1}(n),$$

and applying equation D.23 again,

$$\boldsymbol{k}^{f,q}_{m-1}(n)^* = \sum_{k=1}^{n+1} \lambda^{n+1-k}\boldsymbol{\xi}_1(k)\boldsymbol{\xi}_1(k-m)^* - \boldsymbol{v}^{f,q}_{m-1}(n)^* \boldsymbol{c}^{b,q}_{m-1}(n),$$

and comparing with equation D.142

$$\boldsymbol{k}^{f,q}_{m-1}(n)^* = \boldsymbol{k}^{b,q}_{m-1}(n). \qquad \text{(D.199)}$$

D.7 Kalman Gain Update

We now proceed to find recursive relationships for the Kalman gain. Substituting equations D.117 and D.68 in equation D.114 and replacing $n+1$ with n gives

$$\boldsymbol{A}_{m+1}(n) = \left(\begin{array}{c|c} \boldsymbol{A}_m(n) & \boldsymbol{v}^b_m(n)\dot{\boldsymbol{T}}^*_m \\ \hline \dot{\boldsymbol{T}}_m \boldsymbol{v}^b_m(n)^* & \dot{\boldsymbol{T}}_m \boldsymbol{E}^b_m(n)\dot{\boldsymbol{T}}^*_m \end{array}\right), \qquad \text{(D.200)}$$

and applying equations D.7, D.129 and D.130 gives

$$\boldsymbol{A}_{m+1}^{q}(n) = \left(\begin{array}{c|c} \boldsymbol{A}_{m}^{q}(n) & \boldsymbol{v}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \dot{\boldsymbol{T}}_{m+q}\boldsymbol{v}_{m}^{b,q}(n)^{*} & \dot{\boldsymbol{T}}_{m+q}\boldsymbol{E}_{m}^{b}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \end{array}\right). \tag{D.201}$$

Post-multiplication of equation D.201 gives

$$\boldsymbol{A}_{m+1}^{q}(n)\left(\begin{array}{c} -\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \boldsymbol{I} \end{array}\right) = \left(\begin{array}{c} -\boldsymbol{A}_{m}^{q}(n)\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ +\boldsymbol{v}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline -\boldsymbol{v}_{m}^{b,q}(n)^{*}\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ +\dot{\boldsymbol{T}}_{m+q}\boldsymbol{E}_{m}^{b}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \end{array}\right),$$

and using equation D.23 gives

$$\boldsymbol{A}_{m+1}^{q}(n)\left(\begin{array}{c} -\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \boldsymbol{I} \end{array}\right) = \left(\begin{array}{c} \mathbf{0} \\ \hline \dot{\boldsymbol{T}}_{m+q}[\boldsymbol{E}_{m}^{b}(n) \\ \qquad -\boldsymbol{v}_{m}^{b,q}(n)^{*}\boldsymbol{c}_{m}^{b,q}(n)]\dot{\boldsymbol{T}}_{m+q}^{*} \end{array}\right),$$

and from D.34,

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)\left(\begin{array}{c} -\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \boldsymbol{I} \end{array}\right) &= \left(\begin{array}{c} \mathbf{0} \\ \hline \dot{\boldsymbol{T}}_{m+q}\boldsymbol{\epsilon}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \end{array}\right) \\ &= \left(\begin{array}{c} \mathbf{0} \\ \hline \hat{\boldsymbol{\epsilon}}_{m}^{b,q}(n) \end{array}\right), \end{aligned} \tag{D.202}$$

and by post-multiplication

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)\left(\begin{array}{c} -\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \boldsymbol{I} \end{array}\right)\hat{\boldsymbol{\epsilon}}_{m}^{b,q}(n)^{-1}\dot{\boldsymbol{T}}_{m+q}\left[\boldsymbol{d}_{m}^{b}(n) - \boldsymbol{v}_{m}^{b,q}(n)^{*}\boldsymbol{g}_{m}^{q}(n)\right] \\ = \left(\begin{array}{c} \mathbf{0} \\ \hline \dot{\boldsymbol{T}}_{m+q}\left[\boldsymbol{d}_{m}^{b,q}(n) - \boldsymbol{v}_{m}^{b,q}(n)^{*}\boldsymbol{g}_{m}^{q}(n)\right] \end{array}\right). \end{aligned} \tag{D.203}$$

Substitute equation D.38 in the left hand side of equation D.203 and apply equation D.15 to get

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)\left(\begin{array}{c} -\boldsymbol{c}_{m}^{b,q}(n)\dot{\boldsymbol{T}}_{m+q}^{*} \\ \hline \boldsymbol{I} \end{array}\right)\hat{\boldsymbol{\epsilon}}_{m}^{b,q}(n)^{-1}\dot{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}_{m}^{b,q}(n) \\ = \left(\begin{array}{c} \mathbf{0} \\ \hline \dot{\boldsymbol{T}}_{m+q}\left[\boldsymbol{d}_{m}^{b,q}(n) - \boldsymbol{v}_{m}^{b,q}(n)^{*}\boldsymbol{g}_{m}^{q}(n)\right] \end{array}\right). \end{aligned} \tag{D.204}$$

From equations D.201 and D.38

$$\boldsymbol{A}^q_{m+1}(n)\left(\frac{\boldsymbol{g}^q_m(n)}{\mathbf{0}}\right)=\left(\frac{\boldsymbol{x}^q_m(n)}{\grave{\boldsymbol{T}}_{m+q}\boldsymbol{v}^{b,q}_m(n)^*\boldsymbol{g}^q_m(n)}\right). \tag{D.205}$$

Add equations D.205 and D.204

$$\begin{aligned}\boldsymbol{A}^q_{m+1}(n)\left(\left(\frac{\boldsymbol{g}^q_m(n)}{\mathbf{0}}\right)+\left(\frac{-\boldsymbol{c}^{b,q}_m(n)\grave{\boldsymbol{T}}^*_{m+q}}{\boldsymbol{I}}\right)\hat{\epsilon}^{b,q}_m(n)^{-1}\grave{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}^{b,q}_m(n)\right)&\\ =\left(\frac{\boldsymbol{x}^q_m(n)}{\grave{\boldsymbol{T}}_{m+q}\boldsymbol{d}^b_m(n)}\right)&\\ =\boldsymbol{x}^q_{m+1}(n).&\end{aligned} \tag{D.206}$$

Comparing equation D.206 with equation D.38 we obtain

$$\boldsymbol{g}^q_{m+1}(n)=\left(\frac{\boldsymbol{g}^q_m(n)}{\mathbf{0}}\right)+\left(\frac{-\boldsymbol{c}^{b,q}_m(n)\grave{\boldsymbol{T}}^*_{m+q}}{\boldsymbol{I}}\right)\hat{\epsilon}^{b,q}_m(n)^{-1}\grave{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}^{b,q}_m(n). \tag{D.207}$$

Similarly, substituting equations D.120 and D.68 in equation D.114 and replacing $n+1$ with n gives

$$\boldsymbol{A}_{m+1}(n)=\left(\begin{array}{c|c}\boldsymbol{E}^f_m(n-1) & \hat{\boldsymbol{v}}^f_m(n-1)^*\hat{\grave{\boldsymbol{T}}}^*_{1,m}\\ \hline \hat{\grave{\boldsymbol{T}}}_{1,m}\hat{\boldsymbol{v}}^f_m(n-1) & \hat{\grave{\boldsymbol{T}}}_{1,m}\hat{\boldsymbol{A}}_m(n-1)\hat{\grave{\boldsymbol{T}}}^*_{1,m}\end{array}\right). \tag{D.208}$$

Note that $\hat{\grave{\boldsymbol{T}}}_{1,m}=\grave{\boldsymbol{T}}_{1,m}\hat{\grave{\boldsymbol{T}}}_{0,m-1}$ and apply equations D.129 and D.130

$$\boldsymbol{A}^0_{m+1}(n)=\left(\begin{array}{c|c}\boldsymbol{E}^f_m(n-1) & \boldsymbol{v}^{f,1}_m(n-1)^*\\ \hline \boldsymbol{v}^{f,1}_m(n-1) & \boldsymbol{A}^1_m(n-1)\end{array}\right). \tag{D.209}$$

Equations D.209 and D.129 and D.130 imply

$$\boldsymbol{A}^q_{m+1}(n)=\left(\begin{array}{c|c}\grave{\boldsymbol{T}}_q\boldsymbol{E}^f_m(n-1)\grave{\boldsymbol{T}}^*_q & \grave{\boldsymbol{T}}_q\boldsymbol{v}^{f,q+1}_m(n-1)^*\\ \hline \boldsymbol{v}^{f,q+1}_m(n-1)\grave{\boldsymbol{T}}^*_q & \boldsymbol{A}^{q+1}_m(n-1)\end{array}\right). \tag{D.210}$$

Therefore,

$$\boldsymbol{A}^q_{m+1}(n)\left(\frac{\boldsymbol{I}}{-\boldsymbol{c}^{f,q+1}_m(n-1)\grave{\boldsymbol{T}}^*_q}\right)=\left(\begin{array}{l}\grave{\boldsymbol{T}}_q\boldsymbol{E}^f_m(n-1)\grave{\boldsymbol{T}}^*_q\\ \quad-\grave{\boldsymbol{T}}_q\boldsymbol{v}^{f,q+1}_m(n-1)^*\boldsymbol{c}^{f,q+1}_m(n-1)\grave{\boldsymbol{T}}^*_q\\ \hline \boldsymbol{v}^{f,q+1}_m(n-1)\grave{\boldsymbol{T}}^*_q\\ \quad-\boldsymbol{A}^{q+1}_m(n-1)\boldsymbol{c}^{f,q+1}_m(n-1)\grave{\boldsymbol{T}}^*_q\end{array}\right)$$

and using equations D.23, D.17 and D.184,

$$\boldsymbol{A}_{m+1}^{q}(n)\left(\frac{\boldsymbol{I}}{-\boldsymbol{c}_{m}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_{q}^{*}}\right)=\left(\frac{\hat{\hat{\boldsymbol{\epsilon}}}^{f,q+1}}{\boldsymbol{0}}\right), \tag{D.211}$$

and by post-multiplication,

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)&\left(\frac{\boldsymbol{I}}{-\boldsymbol{c}_{m}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_{q}^{*}}\right)\hat{\hat{\boldsymbol{\epsilon}}}_{m}^{f,q+1}(n-1)^{-1}\grave{\boldsymbol{T}}_{q}\\&\times\left[\boldsymbol{d}_{m}^{f}(n-1)-\boldsymbol{v}_{m}^{f,q+1}(n-1)^{*}\boldsymbol{g}_{m}^{q+1}(n-1)\right]\\&=\left(\frac{\grave{\boldsymbol{T}}_{q}\left[\boldsymbol{d}_{m}^{f,q+1}(n-1)-\boldsymbol{v}_{m}^{f,q+1}(n-1)^{*}\boldsymbol{g}_{m}^{q+1}(n-1)\right]}{\boldsymbol{0}}\right).\end{aligned} \tag{D.212}$$

Substitute equation D.38 in the left hand side of equation D.212 and apply equation D.15 to get

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)&\left(\frac{-\boldsymbol{c}_{m}^{f,q}(n)\grave{\boldsymbol{T}}_{m+q}^{*}}{\boldsymbol{I}}\right)\hat{\boldsymbol{\epsilon}}_{m}^{f,q}(n)^{-1}\grave{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}_{m}^{f,q}(n)\\&=\left(\frac{\boldsymbol{0}}{\grave{\boldsymbol{T}}_{m+q}\left[\boldsymbol{d}_{m}^{f,q}(n)-\boldsymbol{v}_{m}^{f,q}(n)^{*}\boldsymbol{g}_{m}^{q}(n)\right]}\right).\end{aligned} \tag{D.213}$$

From equations D.210 and D.38

$$\boldsymbol{A}_{m+1}^{q}(n)\left(\frac{\boldsymbol{0}}{\boldsymbol{g}_{m}^{q+1}(n-1)}\right)=\left(\frac{\grave{\boldsymbol{T}}_{q}\boldsymbol{v}^{f,q+1}(n-1)\boldsymbol{g}_{m}^{q+1}(n-1)}{\boldsymbol{x}_{m}^{q+1}(n-1)}\right). \tag{D.214}$$

Add equations D.214 and D.213

$$\begin{aligned}\boldsymbol{A}_{m+1}^{q}(n)&\left(\left(\frac{\boldsymbol{0}}{\boldsymbol{g}_{m}^{q+1}(n-1)}\right)\right.\\&\left.+\left(\frac{\boldsymbol{I}}{-\boldsymbol{c}_{m}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_{q}^{*}}\right)\hat{\hat{\boldsymbol{\epsilon}}}_{m}^{f,q+1}(n-1)^{-1}\grave{\boldsymbol{T}}_{q}\tilde{\boldsymbol{e}}_{m}^{f,q}(n-1)\right)\\&=\left(\frac{\grave{\boldsymbol{T}}_{q}\boldsymbol{d}_{m}^{f,q+1}(n-1)}{\boldsymbol{x}_{m}^{q+1}(n-1)}\right)\\&=\boldsymbol{x}_{m+1}^{q}(n).\end{aligned} \tag{D.215}$$

Comparing equation D.215 with equation D.38 we obtain

$$\boldsymbol{g}_{m+1}^{q}(n) = \left(\frac{\mathbf{0}}{\boldsymbol{g}_{m}^{q+1}(n-1)}\right) + \left(\frac{\boldsymbol{I}}{-\boldsymbol{c}_{m}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_{q}^{*}}\right) \hat{\tilde{\boldsymbol{\epsilon}}}_{m}^{f,q+1}(n-1)^{-1}\grave{\boldsymbol{T}}_{q}\tilde{\boldsymbol{e}}_{m}^{f,q}(n-1). \tag{D.216}$$

D.8 Gamma Update

From equations D.44 and D.38

$$\gamma_{m}^{q}(n) = \boldsymbol{x}_{m}^{q}(n)^{*}\boldsymbol{g}_{m}^{q}(n). \tag{D.217}$$

Substitute equations D.207 and D.193 into equation D.217 to get

$$\gamma_{m}^{q}(n) = \boldsymbol{x}_{m-1}^{q}\boldsymbol{g}_{m-1}^{q}(n) + [\boldsymbol{x}_{m-1}^{q}\boldsymbol{c}_{m-1}^{b,q}(n) - \boldsymbol{d}_{m+q-1}^{b}{}^{*}] \times \grave{\boldsymbol{T}}_{m+q}^{*}\hat{\tilde{\boldsymbol{\epsilon}}}_{m-1}^{b,q}(n)^{-1}\grave{\boldsymbol{T}}_{m+q-1}\tilde{\boldsymbol{e}}_{m-1}^{b,q}(n),$$

and substitute equations D.217 and D.15 to get

$$\gamma_{m}^{q}(n) = \gamma_{m-1}^{q}(n) + \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n)^{*}\grave{\boldsymbol{T}}_{m+q-1}^{*}\hat{\tilde{\boldsymbol{\epsilon}}}_{m-1}^{b,q}(n)^{-1}\grave{\boldsymbol{T}}_{m+q-1}\tilde{\boldsymbol{e}}_{m-1}^{b,q}(n). \tag{D.218}$$

Similarly, substitute equations D.216 and D.196 into equation D.217 to get

$$\gamma_{m}^{q}(n) = \boldsymbol{x}_{m-1}^{q+1}{}^{*}\boldsymbol{g}_{m-1}^{q+1}(n-1) + \left[\boldsymbol{d}_{m-1}^{f}(n-1)^{*} - \boldsymbol{x}_{m-1}^{q+1}(n-1)^{*}\boldsymbol{c}_{m}^{f,q+1}(n-1)\right] \times \grave{\boldsymbol{T}}_{q}^{*}\hat{\tilde{\boldsymbol{\epsilon}}}_{m}^{f,q+1}(n-1)^{-1}\grave{\boldsymbol{T}}_{q}\tilde{\boldsymbol{e}}_{m}^{f,q}(n-1),$$

and substitute equations D.217 and D.15 to get

$$\gamma_{m}^{q}(n) = \gamma_{m-1}^{q+1}(n-1) + \tilde{\boldsymbol{e}}_{m-1}^{f,q+1}(n-1)^{*}\grave{\boldsymbol{T}}_{q}^{*}\hat{\tilde{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n-1)^{-1}\grave{\boldsymbol{T}}_{q}\tilde{\boldsymbol{e}}_{m-1}^{f,q+1}(n-1). \tag{D.219}$$

D.9 The Generalized Fast Kalman Algorithm

We now have the recursive relationships we need for a fast Kalman filter. This section will chiefly consist of organizing the work of previous sections into a suitable order.

From equations D.12 and D.14

$$\boldsymbol{e}_m^{f,q}(n-1) = \boldsymbol{\xi}_1^q(n) - \boldsymbol{c}_m^{f,q}(n-2)^* \boldsymbol{x}_m^q(n-1). \tag{D.220}$$

From equations D.37 and D.38

$$\boldsymbol{c}_m^{f,q}(n-1) = \boldsymbol{c}_m^{f,q}(n-2) + \boldsymbol{g}_m^q(n-1)\boldsymbol{e}_m^{f,q}(n-1)^*. \tag{D.221}$$

Put

$$\boldsymbol{f}^q(n) = \left[1 - \boldsymbol{x}_m^q(n-1)^* \boldsymbol{g}_m^q(n-1)\right] \boldsymbol{e}_m^{f,q}(n-1). \tag{D.222}$$

Substitute equation D.217 in equation D.222 and then substitute into equation D.47 to get

$$\boldsymbol{\epsilon}_m^{f,q}(n-1) = \lambda \boldsymbol{\epsilon}_m^{f,q}(n-2) + \boldsymbol{f}^q(n)\boldsymbol{e}_m^{f,q}(n-1)^*. \tag{D.223}$$

Substitute equation D.184 into equation D.216 to get

$$\boldsymbol{g}_{m+1}^q(n) = \left(\frac{\left[\grave{\boldsymbol{T}}_q \boldsymbol{\epsilon}_m^{f,q+1}(n-1)\grave{\boldsymbol{T}}_q^*\right]^{-1} \grave{\boldsymbol{T}}_q \tilde{\boldsymbol{e}}_m^{f,q}(n-1)}{\begin{array}{l}\boldsymbol{g}_m^{q+1}(n-1) - \boldsymbol{c}_m^{f,q+1}(n-1)\grave{\boldsymbol{T}}_q^* \\ \qquad \times \left[\grave{\boldsymbol{T}}_q \boldsymbol{\epsilon}_m^{f,q+1}(n-1)\grave{\boldsymbol{T}}_q^*\right]^{-1} \\ \qquad \times \grave{\boldsymbol{T}}_q \tilde{\boldsymbol{e}}_m^{f,q}(n-1)\end{array}} \right). \tag{D.224}$$

Re-partition equation D.224 and put

$$\left(\frac{\hat{\boldsymbol{g}}(n)}{\boldsymbol{\mu}(n)} \right) = \boldsymbol{g}_{m+1}^q(n). \tag{D.225}$$

From equations D.12 and D.14

$$\boldsymbol{e}^{b,q}(n) = \boldsymbol{\xi}_m(n-1) - \boldsymbol{c}_m^{b,q}(n-1)^* \boldsymbol{x}_m^q(n). \tag{D.226}$$

From equations D.37 and D.38

$$\boldsymbol{c}_m^{b,q}(n) = \boldsymbol{c}_m^{b,q}(n-1) + \boldsymbol{g}_m^q(n)\boldsymbol{e}_m^{b,q}(n)^*. \tag{D.227}$$

Unfortunately, we haven't calculated $\boldsymbol{g}(n)$ yet, and we can't simply change the order of the calculations as we would like to use $\boldsymbol{c}_m^{b,q}(n)$ to calculate $\boldsymbol{g}(n)$. So we need to be a little devious. Equate equations D.207 and D.225 to get

$$\boldsymbol{\mu}(n) = \hat{\boldsymbol{\epsilon}}_m^{b,q}(n)^{-1}\grave{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}_m^{b,q}(n) \tag{D.228}$$

$$\begin{aligned}\hat{\boldsymbol{g}}(n) &= \boldsymbol{g}_m^q(n) - \boldsymbol{c}_m^{b,q}(n)\grave{\boldsymbol{T}}_{m+q}^*\hat{\boldsymbol{\epsilon}}_m^{b,q}(n)^{-1} \\ &\qquad \times \grave{\boldsymbol{T}}_{m+q}\tilde{\boldsymbol{e}}_m^{b,q}(n) \\ &= \boldsymbol{g}_m^q(n) - \boldsymbol{c}_m^{b,q}(n)\grave{\boldsymbol{T}}_{m+q}^*\boldsymbol{\mu}(n).\end{aligned} \tag{D.229}$$

Use equation D.229 to substitute for $\boldsymbol{g}_m^q(n)$ in equation D.227

$$\boldsymbol{c}_m^{b,q}(n) = \boldsymbol{c}_m^{b,q}(n-1) + \left[\hat{\boldsymbol{g}}(n) - \boldsymbol{c}_m^{b,q}(n)\grave{\boldsymbol{T}}_{m+q}^*\boldsymbol{\mu}(n)\right]\boldsymbol{e}_m^{b,q}(n)^*,$$

and collect all terms in $\boldsymbol{c}_m^{b,q}(n)$ as follows

$$\boldsymbol{c}_m^{b,q}(n)\left[\boldsymbol{I} + \grave{\boldsymbol{T}}_{m+q}^*\boldsymbol{\mu}(n)\boldsymbol{e}_m^{b,q}(n)^*\right] = \boldsymbol{c}_m^{b,q}(n-1) + \hat{\boldsymbol{g}}(n)\boldsymbol{e}_m^{b,q}(n)^*.$$

Therefore,

$$\boldsymbol{c}_m^{b,q}(n) = \left[\boldsymbol{c}_m^{b,q}(n-1) + \hat{\boldsymbol{g}}(n)\boldsymbol{e}_m^{b,q}(n)^*\right]\left[\boldsymbol{I} + \grave{\boldsymbol{T}}_{m+q}^*\boldsymbol{\mu}(n)\boldsymbol{e}_m^{b,q}(n)^*\right]^{-1}. \tag{D.230}$$

From equation D.229

$$\boldsymbol{g}_m^q(n) = \hat{\boldsymbol{g}}(n) + \boldsymbol{c}_m^{b,q}(n)\grave{\boldsymbol{T}}_{m+q}^*\boldsymbol{\mu}(n). \tag{D.231}$$

The equalizer calculations are now simple. From equation D.11

$$\boldsymbol{y}_m^e(n) = \boldsymbol{c}_m^e(n-1)^*\boldsymbol{x}(n), \tag{D.232}$$

and from equation D.13

$$\boldsymbol{e}_m^e = \boldsymbol{d}_m^e(n) - \boldsymbol{y}_m^e(n), \tag{D.233}$$

and finally, from equations D.37 and D.38,

$$\boldsymbol{c}_m^e(n) = \boldsymbol{c}_m^e(n-1) + \boldsymbol{g}_m(n)\boldsymbol{e}_m(n)^*. \tag{D.234}$$

The algorithm consists of equations D.220, D.221, D.222, D.223, D.224, D.225, D.226, D.230, D.231, D.232, D.233 and D.234. We only need the results for $q = 0$, but unfortunately, the calculation of $\boldsymbol{g}_{m+1}^0(n)$ in equation D.224 requires $\boldsymbol{g}_m^1(n-1)$, which in turn requires $\boldsymbol{g}_{m-1}^2(n-2)$ and so on. In general, therefore, we need to maintain filters with values of m and q of $(1, m_1 - 1), (2, m_1 - 1), (3, m_1 - 2) \ldots (m_1, 0)$ in order for the final filter to have $q = 0$ and $m = m_1$. The equalizer equations (D.232, D.233 and D.234) can be omitted for all except the final filter.

In general this makes the number of calculations increase as $O(m_1^2)$ since we have m_1 filters whose length is proportional to m_1. However, savings are possible if $\boldsymbol{g}_m^q = \boldsymbol{g}_m^{q+1}$ at any point. In particular, this happens in the linear case for $q = 0$ and so the number of operations in the linear case increases as $O(m)$.

D.10 The Generalized LS Lattice Filter

The Lattice Filter is fundamentally order recursive instead of time recursive. As each sample is processed, the filter is reset to zero length and it is then recursively "grown" to the desired length.

For each n initialize to $m = 0$. From equation D.11

$$\boldsymbol{y}_0^{s,q}(n) = \mathbf{0}, \quad s \in \{e, f, b\}, \tag{D.235}$$

and from equation D.44

$$\gamma_0^q(n) = 0, \tag{D.236}$$

and from equations D.13 and D.11 and D.12

$$\boldsymbol{e}_0^e(n) = \boldsymbol{d}^e(n), \tag{D.237}$$

and from equations D.13 and D.11 and D.12

$$\boldsymbol{e}_0^{f,q}(n-1) = \tilde{\boldsymbol{e}}_0^{b,q}(n) = \boldsymbol{e}_0^{b,q}(n) = \boldsymbol{\xi}_1(n). \tag{D.238}$$

From equations D.47, D.236 and D.238

$$\epsilon_0^{f,q}(n-1) = \lambda\epsilon_0^{f,q}(n-2) + \boldsymbol{\xi}_1(n)\boldsymbol{\xi}_1{}^*. \tag{D.239}$$

Similarly

$$\epsilon_0^{b,q}(n) = \lambda\epsilon_0^{b,q}(n-1) + \boldsymbol{\xi}_1(n)\boldsymbol{\xi}_1{}^*. \tag{D.240}$$

Since equation D.240 is of the same as equation D.239

$$\epsilon_0^{b,q}(n) = \epsilon_0^{f,q}(n-1). \tag{D.241}$$

Substitute equation D.43 in equation D.198 and replace n with $n-1$ to get

$$\boldsymbol{k}_{m-1}^{f,q}(n-1) = \lambda\boldsymbol{k}_{m-1}^{f,q}(n-2) + \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n-1)\boldsymbol{e}_{m-1}^{f,q}(n-1)^*. \tag{D.242}$$

Put

$$\boldsymbol{K}_m^{f,q}(n) = \hat{\boldsymbol{k}}_{m-1}^{f,q}(n-1)^*\hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n-1)^{-1}\dot{\boldsymbol{T}}_{m+q-1}$$

and substitute equations D.153 and D.152 to get

$$\boldsymbol{K}_m^{f,q}(n) = \boldsymbol{k}_{m-1}^{f,q}(n-1)^*\dot{\boldsymbol{T}}_{m+q-1}^*\left(\dot{\boldsymbol{T}}_{m+q-1}\boldsymbol{\epsilon}_{m-1}^{b,q}(n-1)\dot{\boldsymbol{T}}_{m+q-1}^*\right)^{-1}\dot{\boldsymbol{T}}_{m+q-1}. \tag{D.243}$$

Put

$$\boldsymbol{K}_m^{b,q}(n) = \hat{\hat{\boldsymbol{k}}}_{m-1}^{b,q+1}(n-1)^*\hat{\hat{\boldsymbol{\epsilon}}}_{m-1}^{f,q+1}(n-1)^{-1}\dot{\boldsymbol{T}}_q,$$

and substitute equations D.186 and D.184 to get

$$\boldsymbol{K}_m^{b,q}(n) = \boldsymbol{k}_{m-1}^{b,q+1}(n-1)^*\dot{\boldsymbol{T}}_q^*\left(\dot{\boldsymbol{T}}_q\boldsymbol{\epsilon}_{m-1}^{f,q+1}(n-1)\dot{\boldsymbol{T}}_q^*\right)^{-1}\dot{\boldsymbol{T}}_q,$$

and applying equation D.199,

$$\boldsymbol{K}_m^{b,q}(n) = \boldsymbol{k}_{m-1}^{f,q+1}(n-1)\dot{\boldsymbol{T}}_q^*\left(\dot{\boldsymbol{T}}_q\boldsymbol{\epsilon}_{m-1}^{f,q+1}(n-1)\dot{\boldsymbol{T}}_q^*\right)^{-1}\dot{\boldsymbol{T}}_q. \tag{D.244}$$

From equation D.243 and equation D.195

$$\boldsymbol{e}_m^{f,q}(n-1) = \boldsymbol{e}_{m-1}^{f,q}(n-1) - \boldsymbol{K}_m^{f,q}(n-1)\boldsymbol{e}_{m-1}^{b,q}(n-1). \tag{D.245}$$

From equation D.244 and equation D.197

$$\boldsymbol{e}_m^{b,q}(n) = \boldsymbol{e}_{m-1}^{b,q+1}(n-1) - \boldsymbol{K}_m^{b,q}(n-1)\boldsymbol{e}_{m-1}^{f,q+1}(n-1). \tag{D.246}$$

From equations D.168, D.153, D.154 and D.152

$$\begin{aligned}\boldsymbol{\epsilon}_m^{f,q}(n-1) = \boldsymbol{\epsilon}_{m-1}^{f,q}(n-1) - \boldsymbol{k}_{m-1}^{f,q}(n-1)^* \grave{\boldsymbol{T}}^*_{m+q-1} \\ \times \left(\grave{\boldsymbol{T}}_{m+q-1}\boldsymbol{\epsilon}_{m-1}^{b,q}(n-1)\grave{\boldsymbol{T}}^*_{m+q-1}\right)^{-1} \\ \times \grave{\boldsymbol{T}}_{m+q-1}\boldsymbol{k}_{m-1}^{f,q}(n-1),\end{aligned}$$

and from equation D.243

$$\boldsymbol{\epsilon}_m^{f,q}(n-1) = \boldsymbol{\epsilon}_{m-1}^{f,q}(n-1) - \boldsymbol{K}_m^{f,q}(n)\boldsymbol{k}_{m-1}^{f,q}(n-1). \tag{D.247}$$

From equations D.192, D.185, D.186 and D.184

$$\begin{aligned}\boldsymbol{\epsilon}_m^{b,q}(n) = \boldsymbol{\epsilon}_{m-1}^{b,q+1}(n-1) - \boldsymbol{k}_{m-1}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_q^* \\ \times \left(\grave{\boldsymbol{T}}_q\boldsymbol{\epsilon}_{m-1}^{f,q+1}(n-1)\grave{\boldsymbol{T}}_q^*\right)^{-1} \\ \times \grave{\boldsymbol{T}}_q\boldsymbol{k}_{m-1}^{b,q+1}(n-1),\end{aligned}$$

and from equation D.244

$$\boldsymbol{\epsilon}_m^{b,q}(n) = \boldsymbol{\epsilon}_{m-1}^{b,q+1}(n-1) - \boldsymbol{K}_m^{b,q}(n)\boldsymbol{k}_{m-1}^{b,q+1}(n-1),$$

and applying equation D.199

$$\boldsymbol{\epsilon}_m^{b,q}(n) = \boldsymbol{\epsilon}_{m-1}^{b,q+1}(n-1) - \boldsymbol{K}^b\grave{\boldsymbol{T}}_q\boldsymbol{k}_{m-1}^{f,q+1}(n-1)^*. \tag{D.248}$$

We copy equation D.105 here for convenience;

$$\boldsymbol{y}_m^e(n) = \boldsymbol{y}_{m-1}^e(n) + \boldsymbol{z}_m(n-1)^*\left[\grave{\boldsymbol{T}}_{m-1}\boldsymbol{\epsilon}_{m-1}^b(n-1)\grave{\boldsymbol{T}}^*_{m-1}\right]^{-1}\grave{\boldsymbol{T}}_{m-1}\boldsymbol{e}_{m-1}^b(n). \tag{D.249}$$

From equation D.13

$$\boldsymbol{e}_m^e = \boldsymbol{d}_m^e(n) - \boldsymbol{y}_m^e(n). \tag{D.250}$$

From equation D.43 with $t = b$,

$$\tilde{\boldsymbol{e}}_{m-1}^{b,q}(n)^* = \left[1 - \gamma_{m-1}^q(n)\right]\boldsymbol{e}_{m-1}^{b,q}(n)^*. \tag{D.251}$$

From equation D.218

$$\gamma_m^q(n) = \gamma_{m-1}^q(n) + \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n)^* \grave{\boldsymbol{T}}_{m+q-1}^* \hat{\boldsymbol{\epsilon}}_{m-1}^{b,q}(n)^{-1} \grave{\boldsymbol{T}}_{m+q-1} \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n),$$

and substituting equation D.152

$$\begin{aligned}\gamma_m^q(n) = \gamma_{m-1}^q(n) + \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n)^* \grave{\boldsymbol{T}}_{m+q-1}^* \\ \times \left[\grave{\boldsymbol{T}}_{m+q-1} \boldsymbol{\epsilon}_{m-1}^{b,q}(n) \grave{\boldsymbol{T}}_{m+q-1}^*\right]^{-1} \\ \times \grave{\boldsymbol{T}}_{m+q-1} \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n).\end{aligned} \tag{D.252}$$

Finally substitute equation D.43 into equation D.107

$$\boldsymbol{z}_m(n) = \lambda \boldsymbol{z}_m(n-1) + \grave{\boldsymbol{T}}_{m-1} \tilde{\boldsymbol{e}}_{m-1}^{b,q}(n) e_{m-1}^e(n)^*. \tag{D.253}$$

The lattice algorithm consists initializing at each n to $m = 0$ using equations D.235, D.236, D.237, D.238, D.239 and D.241. Then equations D.242, D.243, D.243, D.244, D.245, D.246, D.247, D.248, D.249, D.250, D.251, D.252 and D.253 are repeated m_1 times with successively increasing m. Equations D.249, D.250 and D.253 are only required if an equalizer is being implemented. As for the case of the Fast Kalman algorithm, we only need results for $q = 0$ but because equations D.248 D.246 and D.244 are in terms of parameters of the filter $q + 1$, we must, in general, maintain filters with values of m and q of $(1, m_1 - 1), (2, m_1 - 1), (3, m_1 - 2) \ldots (m_1, 0)$ in order for the final filter to have $q = 0$ and $m = m_1$. The process is illustrated in figure D.4

In general this makes the number of calculations increase as $O(m_1^2)$ since we have m_1 filters whose length is proportional to m_1. However, savings are possible if $\boldsymbol{x}_m^q = \boldsymbol{x}_m^{q+1}$ at any point, in which case $\boldsymbol{e}^{b,q+1} = \boldsymbol{e}^{b,q}$. In particular, this happens in the linear case for $q = 0$ when the filter takes on the form of figure D.3 and the number of operations increases as $O(m)$.

If $\boldsymbol{T}_i = \boldsymbol{I}$ for $i > i_1$, then the computational burden increases as $O(i_1 m)$ for $m \gg i_1$. Let us consider a concrete example. In the following discussion,

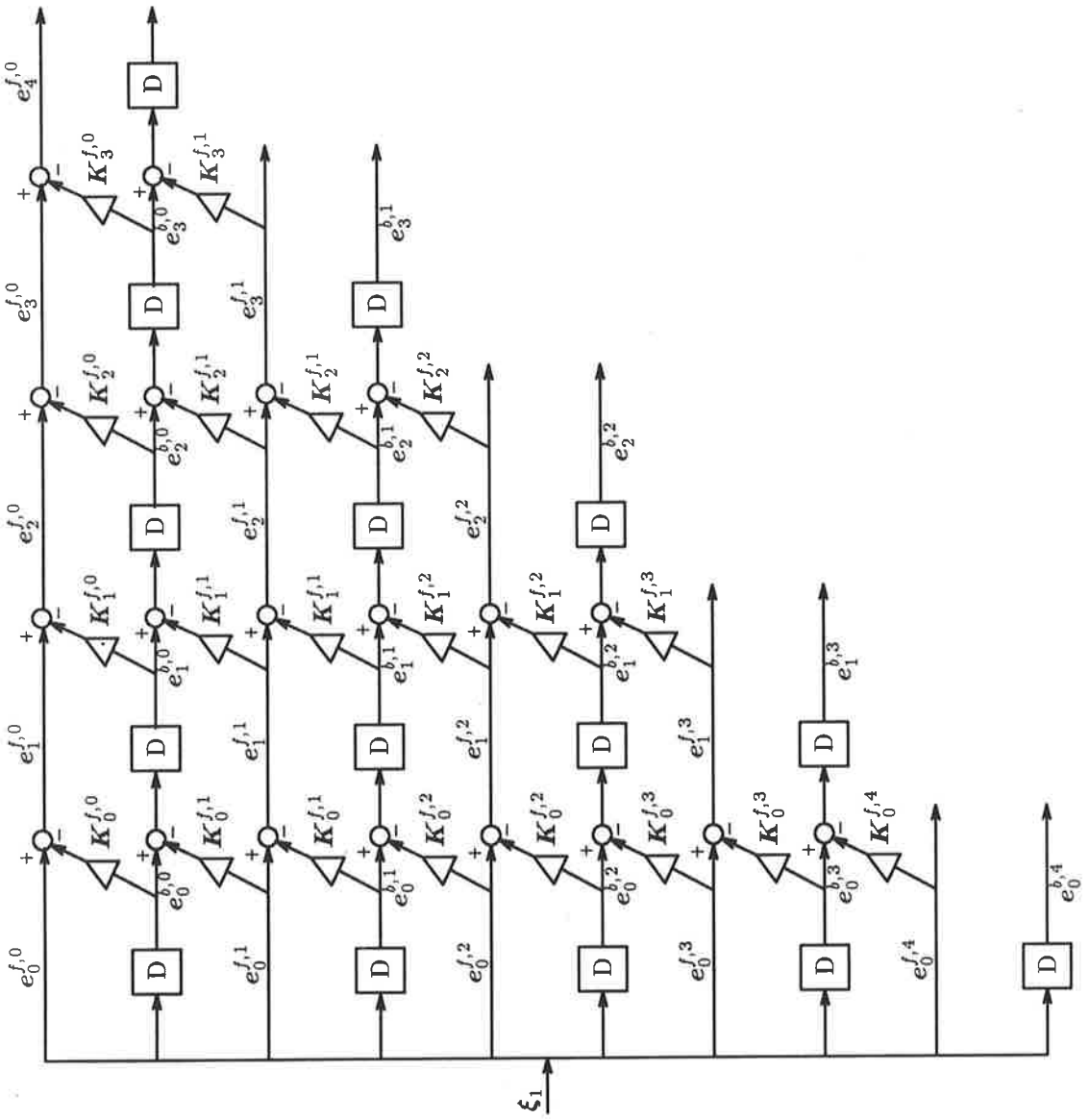


Figure D.4: The generalized lattice filter.

$\boldsymbol{I}_i$, is the identity matrix of rank i, and $\mathbf{0}_{i,j}$ is a zero matrix with i rows and j columns. Now consider a case where the $\boldsymbol{T}_i$ are given by

$$\begin{aligned}
\boldsymbol{T}_1 &= \boldsymbol{I}_3 \\
\boldsymbol{T}_2 &= \boldsymbol{I}_3 \\
\boldsymbol{T}_3 &= \boldsymbol{I}_3 \\
\boldsymbol{T}_4 &= \left(\boldsymbol{I}_2 \,\middle|\, \mathbf{0}_{2,1} \right) \\
\boldsymbol{T}_4 &= \boldsymbol{I}_2 \\
\boldsymbol{T}_5 &= \boldsymbol{I}_2.
\end{aligned} \tag{D.254}$$

Now from the definition (equation D.7), $\hat{\boldsymbol{T}}_{q,m+q-1}$ is the identity matrix if $q > 4$ or $m + q - 1 < 4$ and it follows that $\boldsymbol{x}_m^q = \boldsymbol{x}_m^{q-1}$ for those values of m and q (refer to equation D.5). We can construct table D.2.

				q			
m	0	1	2	3	4	5	6
1	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I}$		$\boldsymbol{I}$	$\boldsymbol{I}$
2	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I}$			$\boldsymbol{I}$	
3	$\boldsymbol{I}$	$\boldsymbol{I}$					
4	$\boldsymbol{I}$						
5							
6							

Table D.2: This table illustrates the values of m and q for which $\hat{\boldsymbol{T}}_{q,m+q-1} = \boldsymbol{I}$ in the example.

Where $\hat{\boldsymbol{T}}_{q,m+q-1} = \boldsymbol{I}$, then $\boldsymbol{x}_m^q = \boldsymbol{x}_m^{q-1}$ and hence all the variables associated with the qth filter are equal to the corresponding variable in the $(q-1)$th filter. In particular $\boldsymbol{e}_m^{b,q} = \boldsymbol{e}_m^{b,q-1}$. We can use this equality, to simplify figure D.4. The result is illustrated in figure D.5.

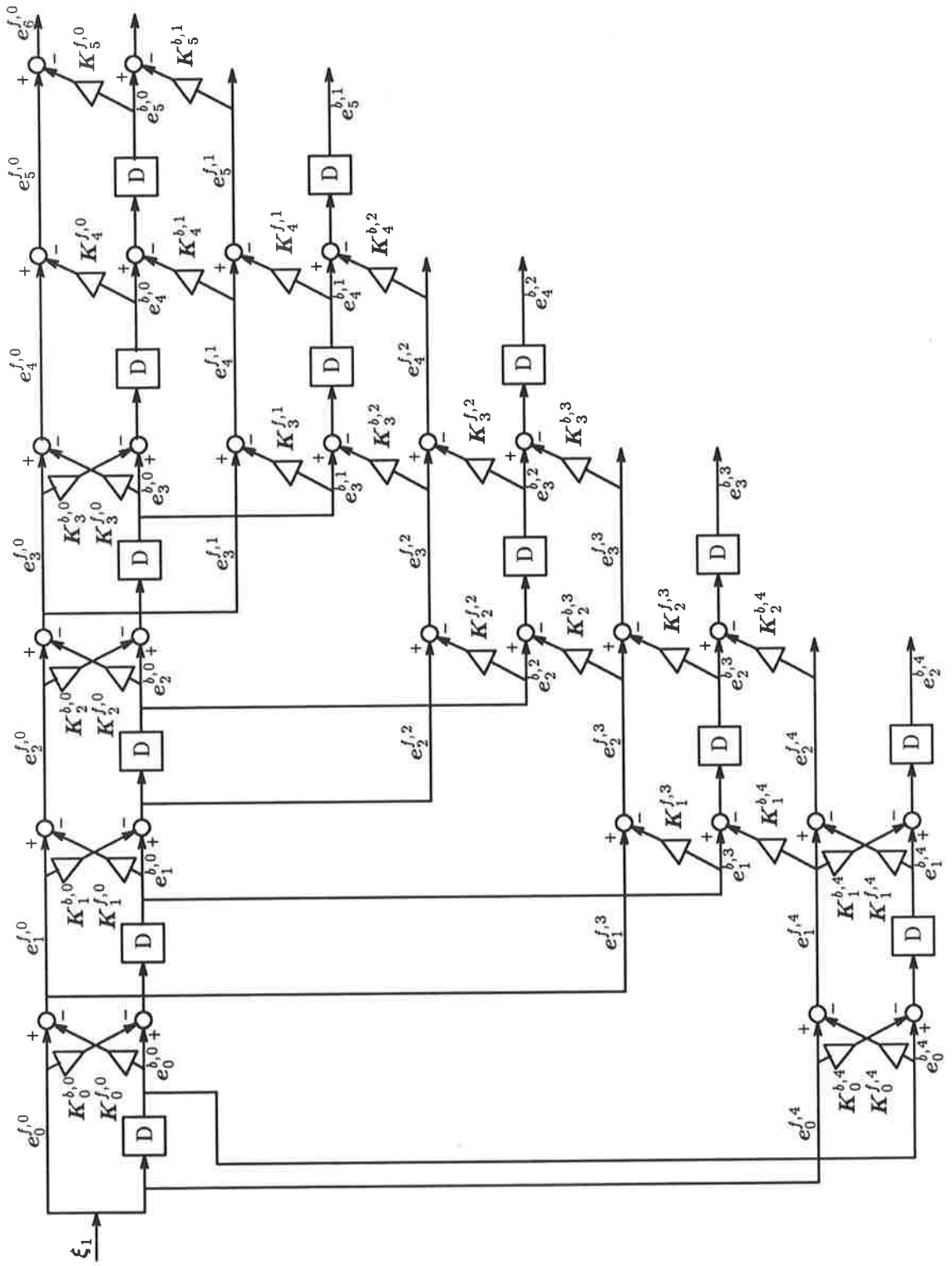


Figure D.5: Reduced generalized lattice filter.

Bibliography

[1] Oscar Agazzi, David G. Messerschmitt, and David A. Hodges. Nonlinear echo cancellation of data systems. *IEEE Transactions on Communications*, 30(11):2421–2433, 1982.

[2] S. J. Anderson. Simulation and modelling for the Jindalee over-the-horizon radar. *Mathematics and Computers in Simulation*, 27:241–248, 1985.

[3] D. E. Barrick. FM/CW signals and digital processing. NOAA Technical Report ERL 283-WPL 26, U.S. Department of Commerce National Oceanic and Atmospheric Administartion, Boulder, Colarado, July 1973.

[4] E. Bedrosian and S. O. Rice. The output properties of Volterra systems (nonlinear systems with memory) driven by harmonic and Gaussian inputs. *Proceedings of the IEEE*, 59(12):1688–1707, December 1971.

[5] M. Bellafemina and S. Benedetto. Identification of nonlinear channels for digital transmission. In *Proc. ISCAS 85*, pages 1477–1490, 1985.

[6] Julius S. Bendat. *Nonlinear System Analysis and Identification*, chapter 3.4, pages 110–113. John Wiley and Sons, 1990.

[7] Julius S. Bendat. *Nonlinear System Analysis and Identification.* John Wiley and Sons, 1990.

[8] S. Benedetto, E. Biglieri, and R. Daffara. Modeling and performance evaluation of nonlinear satellite links—a Volterra series approach. *IEEE Transactions on Aerospace Electronic Systems*, 15(4):494–507, July 1979.

[9] Sergio Benedetto and Ezio Biglieri. Nonlinear equalization of digital satellite channels. *IEEE Journal on Selected Areas in Communications*, 1(1):57–62, January 1983.

[10] A. L. Berman and C. E. Mahle. Nonlinear phase shift in traveling-wave tubes as applied to multiple access communication satellites. *IEEE Transactions on Communication Technology*, 18(1):37–48, February 1970.

[11] Ezio Biglieri, Allen Gersho, Richard D. Gitlin, and Tong Leong Lim. Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission. *IEEE Journal on Selected Areas in Communications*, 2(5):765–777, September 1984.

[12] Nelson M. Blachman. Detectors, bandpass nonlinearities, and their optimization: Inversion of the Chebyshev transform. *IEEE Transactions on Information Technology*, 17(4):398–404, July 1971.

[13] H. S. Black. U.S. Patent 1,686,792, October 1929.

[14] H.S. Black. Stabilized feedback amplifiers. *The Bell System Technical Journal*, 13:1–18, January 1934.

[15] Boualem Boashash, editor. *Proceedings of the 1st IASTED International Symposium on Signal Processing and its Applications*, Brisbane, August 1987. IE Aust.

[16] Hendrik W. Bode. *Network Analysis and Feedback Design.* Robert E. Kreiger Publishing Company, Huntington, New York, 1975. Original edition, 1945.

[17] Robert E. Bogner, Ian Dall, and Richard G. Lane. Distortion effects in over the horizon radar. In Boualem Boashash and Peter Boles, editors, *Proceedings of the 1990 IASTED International Symposium on Signal Processing and its Applications*, pages 391–396, Gold Coast — Australia, August 1990.

[18] K. G. Budden. *The Propagation of Radio Waves*, chapter 13.11–13.12. Cambridge University Press, 1985.

[19] A. Bruce Carlson. *Communication Systems*, chapter 4, pages 145–146. McGraw-Hill, New York, second edition, 1975.

[20] Ph Charas and J. D. Rogers. Improvements in on-board TWTA performance using amplitude and phase predistortion. In *IEE International Conference on Satellite Communication System Technology*, pages 270–280, April 1975.

[21] D.L. Cheadle. Selecting mixers for best intermod performance. *Microwaves*, 12(12):58–62, December 1973.

[22] John M. Cioffi and Thomas Kailath. Fast recursive-least-squares transversal filters for adaptive filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):304–337, April 1984.

[23] Ian W. Dall. Modelling nonlinear distortion in mixers. In R.F. Barrett, editor, *Proceedings of Australian Symposium on Signal Processing and Applications*, pages 118–122, Adelaide, April 1989. IEEE South Australian Section.

[24] Ian W. Dall and Robert E. Bogner. Modeling nonlinear communication systems. In Boashash [15], pages 357–362.

[25] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *Linpack Users' Guide*, chapter 9. Siam, Philadelphia, 1979.

[26] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *Linpack Users' Guide*, chapter 11. Siam, Philadelphia, 1979.

[27] George Frederick Earl. The influence of receiver cross-modulation on attainable HF radar dynamic range. *IEEE Transactions on Instrumentation and Measurement*, 36(3):776–782, September 1987.

[28] D. D. Falconer. Adaptive equalization of channel nonlinearities in QAM data transmission systems. *The Bell System Technical Journal*, 57(7):2589–2611, 1978.

[29] L.E. Franks. *Signal Theory*, chapter 4.4, pages 79–81. Prentice Hall Inc., 1969.

[30] J. C. Fuenzalida, O. Shimbo, and W. L. Cook. Time domain analysis of intermodulation effects caused by nonlinear amplifers. *COMSAT Technical Review*, 3(1):89–141, spring 1973.

[31] Denis Gabor. Communication theory and cybernetics. *IRE Transactions — Circuit Theory*, pages 19–31, December 1954.

[32] Denis Gabor, W. P. L. Wilby, and R. Woodcock. A universal non-linear filter, predictor and simulator which optimizes itself by a learning process. *The Proceedings of The Institute of Electrical Engineers*, pages 422–440, July 1960.

[33] Allen Gersho and Ezio Biglieri. Adaptive cancellation of channel nonlinearities for data transmission. In P. Dewilde and C. A. May, editors, *Links for the Future*, pages 1239–1242, North Holland, 1984. IEEE, Elsevier Science Publishers B.V.

[34] Georgios B. Giannakis. Cumulants: A powerful tool in signal processing. *Proceedings of the IEEE*, 75(9):1333–1334, September 1987.

[35] Georgios B. Giannakis and Jerry M. Mendel. Identification of nonminimum phase systems using higher order statistics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):360+, March 1989.

[36] Henri Girard and Kamilo Feher. A new baseband linearizer for more efficient utilization of earth station amplifiers used for QPSK transmission. *IEEE Journal on Selected Areas in Communications*, 1(1):46–56, January 1983.

[37] P. Hetrakul and D. P. Taylor. Compensators for bandpass nonlinearities in satellite communications. *IEEE Transactions on Aerospace Electronic Systems*, 12(4):509–514, July 1975.

[38] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications.* Kluwer, 1984.

[39] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications*, chapter 3.1, page 43. Kluwer, 1984.

[40] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications*, chapter 4, pages 85–123. Kluwer, 1984.

[41] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications*, chapter 6, pages 144–242. Kluwer, 1984.

[42] Michael L. Honig and David G. Messerschmitt. *Adaptive Filters: Structures, Algorithms and Applications*, chapter 3.1, page 46. Kluwer, 1984.

[43] E. Imboldi and G. R. Stette. AM-to-PM conversion and intermodulation in nonlinear devices. *Proceedings of the IEEE*, 61(6):796–797, July 1973.

[44] Kyoung Il Kim and Edward J. Powers. A digital method of modeling quadratically nonlinear systems with a general random input. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(11):1758–1769, November 1988.

[45] Young C. Kim and Edward J. Powers. Digital bispectral analysis and its applications to nonlinear wave interactions. *IEEE Transactions on Plasma Science*, 7(2):120–131, June 1979.

[46] A. C. Klauder, A. C. Price, S. Darlington, and W. J. Albersheim. The theory and design of chirp radars. *The Bell System Technical Journal*, 39(4):745–808, July 1960.

[47] Taiho Koh and Edward J. Powers. Second order Volterra filtering and its application to nonlinear system identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(6):1445–1455, December 1985.

[48] Daniel T. L. Lee, Martin Morf, and Benjamin Friedlander. Recursive least squares ladder estimation algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(3):627–641, June 1981.

[49] Mohammad Maqusi. Analysis and modeling of intermodulation distortion in wide-band cable TV channels. *IEEE Transactions on Communications*, 35(5):568–572, May 1987.

[50] Robert Aram Minasian. Intermodulation distortion analysis of MESFET amplifiers using the Volterra series representation. *IEEE Transactions on Microwave Theory and Techniques*, 28(1):1–8, January 1980.

[51] M. S. Mueller. Least-squares algorithms for adaptive equalizers. *The Bell System Technical Journal*, 60(8):1905–1925, October 1981.

[52] M. S. Mueller. On the rapid initial convergence of least-squares equalizer adjustment algorithms. *The Bell System Technical Journal*, 60(10):2345–2358, December 1981.

[53] M. Nashed and Robert E. Bogner. Harmonic generation with specified amplitude transformation. *Australian Telecommunication Research*, 20(1):31–39, December 1985.

[54] Chrysostomos L. Nikias and Mysore R. Raghuveer. Bispectrum estimation: A digital signal processing framework. *Proceedings of the IEEE*, 75(7):869–891, July 1987.

[55] Phillip F. Panter. *Communication Systems Design*, chapter 7.4.1,7.4.2, pages 186,187. McGraw-Hill, New York, 1972.

[56] Phillip F. Panter. *Communication Systems Design*, chapter 7,8, pages 181–252. McGraw-Hill, New York, 1972.

[57] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*, chapter 5-6, pages 115–120. McGraw-Hill Series in Electrical Engineering. McGraw-Hill, second edition, 1984.

[58] J. A. Richards. *Analysis of Periodically Time-Varying Systems*, chapter 1.2, pages 7–9. Springer-Verlag, 1983.

[59] Ch. P. Ritz, E. J. Powers, and C. K. An. Applications of digital bispectral analysis to nonlinear wave phenonema. In Boashash [15], pages 352–356.

[60] Enders A. Robinson and Manuel T. Silva. *Digital Signal Processing and Time Series Analysis*, page 364. Holden-Day, pilot edition, 1978.

[61] Rob J. Roy and James Sherman. A learning technique for Volterra series representation. *IEEE Transactions on Automatic Control*, pages 761–764, December 1967.

[62] A. A. M. Saleh. Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers. *IEEE Transactions on Communications*, 29(11):1715–1720, November 1981.

[63] A. A. M. Saleh. Intermodulation analysis of FDMA satellite systems employing compensated and uncompensated TWT's. *IEEE Transactions on Communications*, 30(5):1233–1242, May 1982.

[64] A. A. M. Saleh and J. Salz. Adaptive linearization of power amplifiers in digital radio systems. *The Bell System Technical Journal*, 62(4):1019–1033, April 1982.

[65] Gunkicki Satoh and Toshio Mizuno. Impact of a new TWTA linearizer upon QPSK/TDMA transmission performance. *IEEE Journal on Selected Areas in Communications*, 1(1):39–45, January 1983.

[66] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*, pages 123–141. Wiley, New York, 1980.

[67] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems.* Wiley, New York, 1980.

[68] M. Schetzen. Nonlinear system modeling based on the Wiener theory. *Proceedings of the IEEE*, 69(1):1557–1573, December 1981.

[69] Ronald G. Sea and André G. Vacroux. On the computation of intermodulation products for a power series nonlinearity. *Proceedings of the IEEE*, pages 337–338, March 1969.

[70] H. Seidel. A microwave feed-forward experiment. *The Bell System Technical Journal*, 50(9):2879–1916, November 1971.

[71] H. Seidel, H. R. Beurrier, and A. N. Friedman. Error-controlled high power linear amplifiers at VHF. *The Bell System Technical Journal*, 47(5):651–722, June 1968.

[72] O. Shimbo. Effects of intermodulation, AM–PM conversion and additive noise in multicarrier TWT systems. *Proceedings of the IEEE*, 59(2):230–238, February 1971.

[73] G.R. Stette. Calculation of intermodulation from a single carrier amplitude characteristic. *IEEE Transactions on Communications*, 22(3):319–323, March 1974.

[74] B. D. H. Tellegen. Interaction between radio waves. *Nature*, 131(3319):840, June 1933.

[75] Jitenda K. Tugnait. Comments on "Cumulants: A powerful tool in signal processing". *Proceedings of the IEEE*, 77(3):491–492, March 1989.

[76] Ronald E. Walpole and Raymond H. Myers. *Probability and Statistics for Engineers and Scientists*, chapter 5.2, pages 145–148. Macmillan Publishing Co., Inc., 1972.

[77] Bernard Widrow, John M. McCool, Michael G. Larimore, and C. Richard Johnson, Jr. Stationary and nonstationary learning characteristics of the

LMS adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, August 1976.

[78] Norbet Wiener. *Nonlinear Problems in Random Theory*, chapter 10,11, pages 88–100. The Technology Press, M.I.T. and John Wiley and Sons, Inc, New York, 1958.

[79] Norbet Wiener. *Nonlinear Problems in Random Theory.* The Technology Press, M.I.T. and John Wiley and Sons, Inc, New York, 1958.