Order Number 9413992

# Optimal FIR filter design

Komodromos, Michael Zacharia, Ph.D.

Iowa State University, 1993



U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

# Optimal FIR filter design

by

Michael Zacharia Komodromos

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Electrical Engineering and Computer Engineering
Major: Electrical Engineering (Communications and Signal Processing)

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee

Signature was redacted for privacy.

Iowa State University
Ames, Iowa

1993

## DEDICATION

I dedicate this to my parents for their support during my studies and for teaching me that success is the fruit of hard work.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## ACKNOWLEDGEMENTS

I would like to take the opportunity to thank some people that generously supported me during my studies. First I thank my advisor, Professor Steve Russell for his valuable guidance and strength which made my work possible. I would also like to thank Professors John Basart, John Doherty, Steve Vardeman and Rajbir Dahiya for serving as members of my committee. I am grateful to Dr. Peter Tang of Argonne Labs for his cooperative manner and contributions to this work. My special thanks to Rockwell International and especially to Marvin Frerking for their financial support, technical discussions and help.

In addition to my professional contacts, I would like to thank some other important people. First I would like to thank my friends at Iowa State University. I am most grateful to my family and especially to my wonderful parents for their support during my education.

## ABSTRACT

The design of Finite Impulse Response (FIR) digital filters that considers both phase and magnitude specifications is investigated. This dissertation is divided into two parts. In Part I we present our implementation of an algorithm for the design of minimum-phase filters. In Part II we investigate the design of FIR filters in the complex domain and develop a new powerful design method for digital FIR filters with arbitrary specification of magnitude and phase.

Part I considers the design of minimum-phase filters. The method presented uses direct factorization of the transfer function of a companion Parks-McClellan linear-phase filter of twice the length of the desired minimum-phase filter. The minimum-phase filter is derived with excision of half the zeros of the companion linear-phase filter. The zeros of the prototype filter are found using Laguerre's method. We will present our implementation of the design method, and describe some practical aspects and problems associated with the design of minimum-phase filters.

Part II investigates the design of optimal Chebychev FIR filters in the complex domain. The design of FIR filters with arbitrary specification of magnitude and phase is formulated into a problem of complex approximation. The method developed is capable of designing filters with real or complex coefficients. Complex impulse response designs are

an extension of the real coefficient case based on a proper selection of the approximating basis functions.

The minimax criterion is used and the complex Chebychev approximation is posed as a linear optimization problem. The primal problem is converted to its dual and is solved using an efficient, quadratically convergent algorithm developed by Tang [14]. The relaxation of the linear-phase constraint results in a reduction of the number of coefficients compared to linear-phase designs. Linear-phase filters are a special case of our filter design approach. We examine the design of frequency selective filters with or without the conjugate symmetry, the design of one-sided, two-sided, narrowband and fullband Hilbert Transformers and differentiators.

# GENERAL INTRODUCTION

## Digital Filters

Digital filters are a major part of several systems where processing of data is required. Examples include communication systems, medical equipment, data acquisition systems, audio and video systems. The initial work on developing new and efficient methods to design digital filters started in the 1960's. The objective of the pioneers in this area was to use the vast amount of knowledge on the design of analog filters while inventing algorithms that can be implemented on a computer.

In general the digital filter design involves 1) specification of a desired frequency response, 2) actual design which involves approximation of the given specifications by a realizable filter, and 3) implementation of the resulting design. The first and third tasks are more dependent on the particular application. In this dissertation we only address the design of digital filters. Over the last thirty years there has been an abundance of filter design methods and algorithms ranging from analytical techniques to computation-bound techniques requiring the use of a computer to carry out their large number of computations. Analytical techniques are suitable for low-order designs with only minimal requirements. For high-order digital filter designs with stringent requirements, optimal designs which require computationally intensive algorithms are often sought.

In this dissertation we examine the design of minimum-phase Finite Impulse Response (FIR) filters, and the design of FIR filters in the complex domain. This dissertation is divided into two parts. Part I examines the design of optimal minimum-phase FIR filters. Minimum-phase filters are needed in applications where the large group delay of linear-phase filters cannot be tolerated. Part II presents a new design method for FIR Chebychev filters in the complex domain which approximates both the magnitude and phase responses. The method is efficient and powerful in that it allows design of digital filters not possible with existing design methods.

## Concept of a Digital Filter

A digital filter is a computational process of transforming an input sequence of numbers representing the input signal to another sequence of numbers representing the output signal. A digital filter can be implemented as software on a general purpose computer, or it can be a piece of hardware which is part of a special purpose computer. Even though the computational filtering process is performed in the time domain, the description of the filter is given in the frequency domain. The most familiar filters are the frequency selective filters which pass certain frequencies of the input signal and reject others. Examples are lowpass and bandpass filters. Besides these common types, the term filter includes any system that performs some frequency transformation on the input signal. For example a phase equalizer

might not reject any frequencies of the input signal but transforms the phase of the input signal to compensate the characteristics of a different system.

Digital filters have been divided into several classes depending on their time or frequency domain characteristics, their mathematical representation, and their implementation. One of the most important distinctions among digital filters is the length of their time domain sequence representation which is called impulse response of the filter. This characteristic results in two important classes; 1) Finite Impulse Response (FIR), and 2) Infinite Impulse Response (IIR) filters. A discussion on this classification is provided in the next section. Another major classification depends on whether the system parameters are fixed or changing during the filtering operation. The two classes derived from this distinction are 1) adaptive filters, and 2) non-adaptive filters. The design and implementation of the two types is different. This dissertation addresses only the class of non-adaptive filters.

The implementation of non-adaptive filters results in various classifications. The most important classes are 1) Recursive filters, and 2) Nonrecursive filters. Nonrecursive filters derive their output from a weighted sum of only past and present values of the input. Recursive filters derive their output from past and present values of the input as well as past values of the output.

Other classifications refer to the specific classes of FIR and IIR filters. It is not the purpose of this dissertation to include a thorough study of the subject. There is an abundance of material given in the bibliography section for the interested reader. Next we briefly

discuss FIR and IIR filters and note their major differences. Then we focus only on issues related to the design of FIR filters.

## FIR and IIR Filters

Probably the most important classification is the one related to the length of the filter. Finite Impulse response (FIR) filters are described with a finite length sequence in the time domain, and also by a complex polynomial in negative powers of $z$ as dictated by the z-transform in the frequency domain. A digital filter with infinite length impulse response (IIR) is described by a ratio of polynomials in the frequency domain.

The distinction between FIR and IIR filters is necessary since the properties, design methods, and implementation procedures are different for the two classes. The complex polynomial in negative powers of $z$ describing an FIR filter produces zeros that can be anywhere in the complex plane, and an equal number of poles which are all located at the origin. Therefore, the frequency response of an FIR filter is controlled entirely by its zeros. Also, no stability concerns exist since all the poles are inside the unit circle. On the other hand, the frequency response of an IIR filter is controlled by its zeros as well as the poles. This raises concerns of stability since, for stability, the poles must be inside the unit circle.

It is widely accepted that IIR filters can achieve much better magnitude response characteristics compared to FIR filters with the same number of taps. This performance

superiority comes with a price. First the presence of poles in the IIR transfer function raises stability concerns. Even though it can be argued that good stable IIR filters can be designed, the implementation of the filters with finite precision arithmetic causes some of the poles to move close to, or outside, the unit circle. This is no problem with FIR filters since the poles do not move from zero when the filter is implemented with finite point arithmetic. Other effects of finite precision arithmetic such as limit cycles and coefficient quantization effects are more severe for IIR filters than FIR filters [1] [2]. The last complication of infinite impulse response filters is that it is not possible to have linear phase because no symmetry can be imposed on the infinite impulse response. On the other hand, FIR filters can have exactly linear-phase. This makes both the design and implementation of FIR filters more appealing.

## Design of FIR Filters

In this section we discuss the basic classes of FIR filters and give some background on the most important design approaches. A large part of the research effort on FIR filters has been spent on the design of linear-phase filters. We also examine the design of minimum-phase filters. Finally we give an introduction to the design of FIR filters in the complex domain which constitutes the major part of this dissertation.

## Linear-Phase Filters

The design of linear-phase FIR filters received considerable attention at both the early stages of research on this subject as well as later. The reasons are 1) linear phase results in constant group delay and thus delay distortion is avoided, and 2) ease of design. Probably the easiest way to design an FIR filter is by the *Sampling method* [3]. An FIR filter can be uniquely defined by either the impulse response coefficients or by the Discrete Fourier Transform (DFT) of the coefficients. To approximate a continuous desired frequency response one could then sample in frequency, at equally spaced points, and evaluate the continuous frequency response as an interpolation of the sampled frequency points using the DFT. The approximation would result in no error at the specified points and finite error between the points. More specified points would result in a smoother approximation. Despite the ease of the design, the results are very poor compared to results that can be achieved with more sophisticated methods. An improvement to the technique can be achieved if some of the frequency points are left as unconstrained variables and an optimization procedure is applied to optimize their values. Usually linear programming is used for the optimization.

A full extension of the sampling design technique is to make all frequency response points unconstrained variables and optimize their position in order to minimize some measure of the error between the desired samples and the approximating frequency response. One such technique was developed by Rabiner [4], which designs equiripple FIR filters by

minimizing the maximum of the absolute value of the error between the desired response and the approximating FIR filter. This technique results in optimum Chebychev filters. Its drawback is that it is not very efficient and also does not converge rapidly compared to the well known Remez Exchange algorithm. Additionally the cutoff frequencies cannot be accurately controlled.

The most well-known linear-phase digital FIR filter design technique in use today is the Parks-McClellan algorithm [5]. The method minimizes a weighted-error function between a given, usually ideal response, and an FIR filter using the well-known Remez Exchange algorithm [6]. The method was developed by Parks and McClellan [5] as an algorithm and a computer program written in Fortran. The linear-phase FIR filter design is translated into a Chebychev approximation problem that tries to match an amplitude response with an FIR filter in such a way that the maximum value of the error is minimized. The frequency response of a realizable FIR filter is always a complex-valued function. The linearity of the phase function imposes symmetry on the impulse response of an FIR filter. Using this symmetry, the complex approximation problem is reduced to a real approximation problem. The phase function does not affect the approximation since it is a linear function of the frequency with a known slope for a given filter length. This slope is the negative of the group delay of the filter, and is always constant for a given filter length. It is this property of linear-phase filters that makes their design fairly easy and attractive since real approximation is much easier than complex approximation. Besides the ease in their design, linear-phase filters are preferred because they do not cause any delay distortion on the filtered

signal. The basics of the Parks-McClellan approach and the Remez Exchange algorithm are presented in Appendix A. The reason this material is included as an appendix of this dissertation is 1) the use of the Parks-McClellan algorithm and program for the design of the prototype filter used in the design of the minimum-phase filters presented in Part I, and 2) the results of some of the examples designed by the methods presented in this document are compared to the results obtained by the Parks-McClellan program.

**Minimum-Phase Filters**

The group delay of a linear-phase FIR filter is one half of the length of the filter. In applications requiring filtering with stringent requirements, a high order filter must be designed. If a linear-phase filter is used, the group delay might become prohibitively large. The group delay can be made shorter using minimum-phase filters. These filters give minimum delay in the passband that is much less than that of linear-phase filters for the same length but the delay is not constant for all frequencies. In addition to the minimum delay property, generally minimum-phase filters achieve the same magnitude specifications as linear-phase filters but with fewer coefficients at the expense of phase distortion. The main problem with these filters is that the group delay, other than minimum, is not predictable or controllable by any design technique.

The utilization of minimum-phase filters is an advantage for applications when the deviation of the group delay from a constant is not as important as the nominal value of the group delay. The other advantage these filters offer is that, even though no symmetry of the impulse response is used and the phase is not linear, the design process does not involve complex approximation. The most common design technique is also an extension of the linear-phase filter design, which has been studied extensively and the efficient Remez algorithm can be used. The design of minimum-phase FIR filters is the subject of Part I of this work.

**Complex Filters**

The major part of this dissertation considers the design of digital FIR filters in the complex domain. The motivation for this work was the need to develop an FIR filter design method that would account for both magnitude and phase response characteristics and generally design filters with non-symmetric frequency responses. In the case of linear-phase filters, only constant group delay is accommodated. In the design of minimum-phase filters there is no control over the phase function. The new design method proposed here relaxes the linearity of the phase function in order to get improvement in the magnitude response of the filter. The other major drive behind the effort is that a stronger control on the phase function will be achieved by using complex approximation. This is important when small

group delay deviations from a constant can be tolerated in order to improve the magnitude response, and also make the average value of the group delay smaller than that of a linear-phase filter of the same length. Thus, the design method would allow specification of the average group delay to be smaller than half of the length of the filter as is the case with linear-phase filters.

The design of FIR digital filters with separate magnitude and phase specification is viewed as an approximation of a complex-valued function by a complex polynomial on the unit circle. The symmetry of the filter impulse response is no longer necessary. The design of linear-phase filters is therefore a special case of the suggested filter design method. The difficulty introduced when complex approximation is used is that it is not as straightforward as real approximation. Also, only a few powerful theorems exist that characterize an optimal solution and in many cases these theorems are not powerful enough to suggest an easy and efficient algorithm for the approximation. Because of this difficulty, less work has been done on the subject of complex approximation compared to real approximation. The design of FIR filters in the complex domain is the subject of Part II of this work.

# Work of this Dissertation

## Research Objective

The primary objective of this research is to devise methods, and the corresponding computer software, for the design of digital FIR filters with phase not limited to be a linear function of frequency. The first step towards fulfilling this goal is the design of minimum-phase filters. These filters have minimum phase and delay among all FIR filters with the same magnitude response. The drawback of these filters is that even though the phase function is considered in the design, no control is possible over the phase function.

A more powerful approach, with separate specification and design control of the magnitude and phase functions, is required. This requirement calls for approximation of a desired complex-valued frequency response in the complex domain. The primary motivation for this work is to extend the design of digital FIR filters beyond the design of linear and minimum-phase filters. As a result, the design will be more powerful and versatile, thereby accommodating filters with special characteristics of both magnitude and phase which is not possible with existing filter design methods.

A trade-off between the quality of the phase and magnitude response of the filter is developed which will provide freedom to a designer to match exact system design requirements. When compared to linear-phase filters with the same length, the design method we propose results in filters with better magnitude response characteristics, if the phase is

allowed to be slightly nonlinear. Therefore, in applications where exact linear phase is not as important, computationally more efficient filters can be designed by our method. Furthermore, linear-phase designs are a special case of our design method.

The primary objective is accompanied with the requirement to develop the supporting theory behind the techniques. The supporting theoretical material should be documented to a level that is understandable by readers and at the same time provide sufficient theoretical background to support the validity of the results. In this dissertation, enough material to support a concept is provided or the proper references where additional material can be found are given.

**Organization of this Dissertation**

The remainder of the dissertation is divided in two parts. The first part elaborates on the design of minimum-phase filters. Initially a minimum-phase system is defined. Then the minimum-phase filter problem is formulated. Next we present the design algorithm. A discussion of the implementation of the design algorithm with filter design examples is the subject of Chapter 3 of Part I.

Part II investigates the design of FIR filters in the complex domain. We open this part by introducing the reader to the general problem of designing FIR filters with arbitrary specification of magnitude and phase. Then we discuss the formulation of the filter design

problem as a complex approximation problem. The formulation of the problem is done separately for real and complex coefficients. Then we present the algorithm that applies to both real and complex coefficients. In Chapter 4 of Part II we consider the implementation of the algorithm and we present several filter design examples. We consider the design of linear-phase filters, nearly-linear-phase filters, non-conjugate symmetric filters, Hilbert transformers, and differentiators.

Three appendices are included at the end of this document providing supporting material important to this research work. The first appendix presents the basics of the real Remez algorithm and its use in the linear-phase filter design developed by Parks and McClellan [5]. This material is important because in the design of minimum-phase filters, in Part I, this algorithm is used to design a prototype linear-phase filter which is converted to a minimum-phase filter. Linear-phase filters are a special case of the design method presented in Part II which makes them important to this work. Also, since this design method has been the dominant FIR filter design method for over two decades, it has been a tradition that the results of new design methods are compared to the results of this method.

The second appendix discusses the basics of semi-infinite linear optimization. The material is essential in the presentation of the algorithm for the design of FIR filters in the complex domain. The last appendix contains material related to the characterization of FIR filters with the transfer function and frequency response. Also, we discuss matters related to the phase and group delay functions and computational procedures are presented that were used during this research.

## Conclusion

In this chapter we discussed some of the basics of digital filters. The two major classes of digital filters are FIR and IIR filters. This dissertation considers only the design of FIR filters. In this framework we discussed the design of linear-phase filters, which is very efficient due to the use of the real Remez algorithm. We also discussed minimum-phase filters. Minimum-phase filter design is the subject of Part I. Finally, a design approach was suggested that considers phase, in addition to magnitude, in the design process. The design problem then becomes one of complex approximation. Part II examines the design of FIR filters in the complex domain and an efficient and powerful design method is developed.

# PART I : DESIGN OF MINIMUM PHASE FIR FILTERS

# 1. INTRODUCTION

## 1.1 Subject of PART I

This part of the dissertation describes the design of digital FIR minimum-phase filters. The design method uses direct factorization of the transfer function of a companion Parks-McClellan linear-phase filter. The minimum-phase filter is formed using half the zeros of the prototype filter. We will present our implementation of the design method and describe some practical aspects and problems associated with the design of minimum-phase filters. Typical filter design examples are given to illustrate the use of the computer design program.

In this chapter we give the basic theory of minimum-phase filters. We discuss the importance of the prototype linear-phase filters in the design of minimum-phase filters. The location of the zeros of the prototype linear-phase and the derived minimum-phase filter in complex plane as related to the unit circle is of importance to the design algorithm presented in Chapter 2. A short review on the research work done on minimum-phase filter design is also given in this chapter.

Chapter 2 begins with the formulation of the filter design problem. The minimum-phase filter is derived from a prototype linear-phase filter of degree that is twice the degree of the desired minimum-phase filter. Then the design algorithm is presented. The basic idea behind the algorithm is to factor the complex-valued polynomial describing the prototype

linear-phase filter and then select half of the factored zeros that will result in the minimum-phase filter.

In Chapter 3 we discuss the implementation of the design algorithm with a FORTRAN program. We give typical design examples of minimum-phase filters to show the use of the design program. We will see that the average group delay in the passband is much smaller when compared to the group delay of a linear-phase filter of the same length. However, the group delay is not constant, thereby causing group delay distortion.

## 1.2 Minimum-Phase FIR Systems

A minimum-phase system is defined as one that has all its zeros inside the unit circle. The name, though not trivial by the definition, results from the fact that a minimum-phase system has minimum phase and delay among all the filters with the same magnitude. To discuss minimum-phase systems we first need to look at the *all-pass system.* The transfer function of a first degree all-pass system is given by

$$H_{ap}(z) = \frac{z^{-1} - c^*}{1 - c\, z^{-1}} \tag{1.1}$$

where * denotes complex conjugation. The all-pass system has a zero at $z = 1/c^*$ and a pole at $z = c$. The importance of this system is that its magnitude response is one for all

frequencies and therefore independent of frequency. This can be seen by evaluating the transfer function on the unit circle to get the frequency response given by

$$H_{ap}(f) = \frac{e^{-j2\pi f} - c^*}{1 - c e^{-j2\pi f}} = e^{-j2\pi f} \frac{1 - c^* e^{j2\pi f}}{1 - c e^{-j2\pi f}} \tag{1.2}$$

The exponential factor has unity magnitude and since the numerator and denominator of the ratio are complex conjugates, the magnitude is exactly one for all frequencies. Higher order all-pass systems can be constructed by cascading more all-pass factors similar to the first order system.

Any stable system function can be expressed as a product of a minimum-phase function and an all-pass system function [1],

$$H(f) = H_{\min}(f)\ H_{ap}(f) \tag{1.3}$$

To prove this let an FIR system with transfer function $H(z)$ have all its zeros inside the unit circle except one that is outside the unit circle at $z^{-1} = c^*$ and $|c| < 1$. Then $H(z)$ can be written in the form

$$H(z) = H_1(z)\ (z^{-1} - c^*) \tag{1.4}$$

where $H_1(z)$ is a minimum-phase system since all its zeros are inside the unit circle. We can also write $H(z)$ in the form

$$H(z) = H_1(z)\ (1 - c z^{-1})\ \frac{(z^{-1} - c^*)}{(1 - c z^{-1})} \tag{1.5}$$

where we multiplied and divided by the factor $(1 - cz^{-1})$. The ratio of the last term in the equation above is an all-pass system and the product $H_1(z)(1 - cz^{-1})$ forms a minimum-phase system. Note that the magnitude of the minimum-phase portion of the right hand side, when evaluated on the unit circle, is the same as that of the original system since the all-pass system does not contribute to the magnitude response. The same procedure can be followed if more than one zero is outside the unit circle. Therefore a minimum-phase system can be constructed from a non-minimum-phase system by reflecting all the zeros outside the unit circle at the conjugate reciprocal locations inside the unit circle. Systems with zeros on the unit circle are not strictly minimum phase based on the definition given at the beginning of this discussion, but often they are considered as such since they have many of the properties of strictly minimum-phase systems [1].

Let us now examine why a system with all its zeros inside the unit circle has minimum phase and delay compared to all systems with the same magnitude response. From Equation (1.3) it can be seen that the continuous phase of a non-minimum-phase system is the sum of the phases of the minimum-phase and all-pass systems, i.e.,

$$arg[H(f)] = arg[H_{\min}(f)] + arg[H_{ap}(f)] \tag{1.6}$$

The phase of an all-pass system is always a negative function in the normalized frequency interval [0,0.5]. For a first order all-pass system with a pole at $re^{j\theta}$ the phase function can be obtained from Equation (1.1) given by

$$arg[H_{ap}(f)] = -2\pi f - 2 \arctan\left[\frac{r \sin(2\pi f - \theta)}{1 - r \cos(2\pi f - \theta)}\right] \tag{1.7}$$

To prove the negativity of the phase function of an all-pass system, we consider its group delay which is the negative frequency derivative of the phase function given by Equation (1.7). The group delay of a first order all-pass system with a pole at $re^{j\theta}$ is given by [1]

$$\tau_g(f) = \frac{1 - r^2}{1 + r^2 - 2r \cos(2\pi f - \theta)} = \frac{1 - r^2}{\left|1 - re^{j(\theta - 2\pi f)}\right|^2} \tag{1.8}$$

The group delay is always a positive function of frequency since $r<0$ for a stable system and the denominator is always positive. For higher order all-pass systems, since the phase function is the sum of the individual phases of the first order systems making up the system, the group delay is positive for any order all-pass stable system.

The phase function in $0 \le f \le 0.5$ can be obtained by the integral of the group delay given by

$$arg\,[H_{ap}(f)] = -2\pi \int_0^f \tau_g(\xi)\, d\xi + arg[H_{ap}(0)] \tag{1.9}$$

From Equation (1.7) it can be shown that the phase at zero is zero. Since the group delay is always a positive function, the phase is always a negative function in $0 \le f \le 0.5$. From Equation (1.6) we can see that the group delay of any system is the sum of the delay of the minimum-phase portion and the all-pass group delay. Thus,

$$\tau_g(f) = \tau_{g,\min}(f) + \tau_{g,ap}(f) \tag{1.10}$$

Since the group delay of the all-pass portion is always positive, and the magnitude response of the minimum phase portion is the same as the overall magnitude, a minimum-phase system has the *minimum group delay* among all systems with the same magnitude response.

Finally from Equation (1.6) we see that starting from a minimum-phase system and reflecting one of the zeros inside the unit circle to the conjugate, reciprocal location makes the phase more negative since the phase of an all-pass system is negative in $0 \le f \le 0.5$. On the other hand, the negative of the phase function, which is called *phase-lag* [1], increases.

Therefore the phase lag is minimum for the minimum-phase system compared to all other systems with the same magnitude response. A more precise terminology for such a system is *minimum-phase-lag* system.

## 1.3 Location of the Zeros

The minimum-phase filters are derived from prototype optimal Chebychev linear-phase filters using proper zero excision of their linear-phase complex polynomials. Therefore the layout of the zeros of the prototype linear-phase filters is important. Two properties of the impulse response coefficients dictate the layout of the zeros of a linear-phase filter in the complex plane. First, the coefficients are real numbers which restricts complex roots to come in conjugate pairs. This implies that all zeros above the real axis in the complex plane have mirror image zeros below the real axis. Second, the symmetric impulse response of the linear-phase filter requires symmetric zeros inside and outside the unit circle.

The symmetry of the impulse response implies that if there is a zero at $(r,\theta)$, there will also be a zero at $(1/r,\theta)$. Thus zeros not on the unit circle, nor on the real axis, come in groups of four. The four zeros of a group consist of the zero inside the unit circle, the conjugate zero on the other side of the real axis, and two more that are their reflections on the unit circle. It can also be shown [3] that zeros on the unit circle but not on the axis come

in pairs. Also, zeros on the axis but not on the unit circle come in pairs. Finally, zeros on the unit circle and the real axis, that is zeros located at (1,0) or $(1,\pi)$, are single zeros.

The zeros of a linear-phase FIR filter can also be classified as passband zeros, stopband zeros and extra zeros [3]. The passband zeros control the ripple in the passband while the stopband zeros control the ripple in the stopband. Each ripple in the passband is associated with a pair of passband zeros. In general, no passband zero lies on the unit circle. In contrast, all the stopband zeros lie on the unit circle. The position of a pair of zeros on the unit circle is associated with each valley in the stopband. Finally, the extra zeros control the cutoff frequencies of the filter.

The zeros of the minimum-phase filter are confined inside and on the unit circle. The stopband zeros are on the unit circle and control the attenuation of the filter. Since the coefficients are real, the zeros are in conjugate pairs. The passband zeros are inside the unit circle and control the position and the size of the passband ripple. No zeros exist outside the unit circle.

## 1.4 Previous Work

Originally, the design of optimal minimum-phase filters was proposed by Herrmann and Schuessler [7]. The algorithm uses a prototype linear-phase filter of double the length of the desired minimum-phase filter. The amplitude response of the linear-phase filter is

shifted and scaled in order to make the zeros on the unit circle double. Then, only the zeros inside the unit circle and one of the double zeros on the unit circle are used to compose the minimum-phase filter. The problem with this direct design approach is the difficulty to find the zeros of the prototype linear-phase filter when the order of the filter is high. In their paper [7], the authors do not suggest any particular method of locating the zeros of the complex-valued polynomial. The minimum-phase filter design method described in this dissertation is based on the Herrmann and Schuessler approach.

Other design methods of minimum-phase filters are based on the algorithm proposed in [7] with attempts to improve on the way the zeros of the prototype filter are found. Chen and Parks [8] use a prototype linear-phase filter designed by the Parks-McClellan method. To find the stopband zeros they use the fact that stopband zeros are on the unit circle and their frequency position is known from the final set of extremal frequencies provided by the Parks-McClellan program. The passband zeros are found by giving initial points of the zeros from the passband ripples and then use Newton's method to find the correct location of the zeros.

Boite et. al. [9] proposed a method that uses deconvolution of the complex cepstrum of the filter instead of direct factorization of the complex polynomial. Since the complex cepstrum is infinite and has to be truncated, the resulting phase is an approximation. Mian et. al. [10] proposed a similar procedure that uses FFT. Kamp and Wellkens [11] proposed a method of approximating the square magnitude of the minimum-phase filter using the Remez exchange algorithm. As the authors indicated in [11], the method becomes very

complicated for bandpass filters. Lately an adaptive procedure was proposed by Chit and Mason [12].

## 1.5 Conclusion

In this chapter we discussed some of the basics of minimum-phase systems. Initially minimum-phase systems were defined. The properties of minimum-phase lag and group delay were examined using the idea of all-pass systems. A minimum-phase FIR system has all its zeros inside or on the unit circle. This system has minimum group delay and phase lag among all systems with the same magnitude response.

The location of the zeros of linear-phase filters was considered since these filters are used as prototypes in the design of minimum-phase filters. The linearity of the phase function and the fact that the coefficients are real, constrain the complex zeros to possess certain symmetries. This makes the construction of the minimum-phase filter derived from the prototype linear-phase filter possible. Finally, we discussed some of the existing methods developed for the design of minimum-phase filters.

# 2. FORMULATION AND ALGORITHM

## 2.1 Introduction

This chapter describes the method we use to design FIR minimum-phase filters. The minimum-phase filters are derived from optimal Chebychev linear-phase filters designed using the real Remez algorithm and the Parks-McClellan program. A detailed discussion of the formulation and theory behind the design of the prototype linear-phase filter can be found in Appendix A. The material presented in this chapter assumes that the prototype linear-phase filter is available.

The approach we follow was first proposed by Herrmann and Schuessler [7]. We chose this approach since their method makes good intuitive sense for designs that use the location of transfer function zeros to describe filter performance. Secondly, the initial companion linear-phase filter can be designed using the Parks-McClellan algorithm which is very efficient, well tested and results in an optimal Chebychev filter for a given order.

The prototype linear-phase FIR filter design is an approximation problem that tries to match some ideal amplitude response with a linear combination of functions in such a way that a certain optimization criterion is satisfied. The Chebychev approximation is a minimax technique in that, it tries to minimize the maximum value of the error between the desired amplitude response and the approximating function. The optimal filter designed by the

Remez exchange algorithm has equiripple magnitude response in both the passband and stopband. In the following we discuss some of the preliminaries of the algorithm. Then we describe the design algorithm. The final step of the algorithm is to find the zeros of the prototype linear-phase filter. We discuss the polynomial factorization method we used in our implementation.

## 2.2 Minimum-Phase Design Algorithm

### 2.2.1 Shifting of the Amplitude Response

The location of the zeros of a Chebychev linear-phase filter designed by the Parks-McClellan algorithm was discussed in the previous chapter. Those ideas are used in the design procedure of a minimum-phase filter. There, we have mentioned that a minimum-phase filter can be constructed by a non-minimum-phase filter by reflecting all the zeros outside the unit circle at their reciprocal conjugate location inside the unit circle. Since the linear-phase filter has all its stopband zeros on the unit circle, we need to choose half of those zeros to construct the minimum-phase filter. We confine the following discussion of the design method only to the case of lowpass filters with a prototype lowpass linear-phase filter with symmetric impulse response and odd length. This is often called a "type 1", linear-phase filter. Other cases, such as even length of the prototype, antisymmetric impulse

response, and the remaining frequency selective filters can be treated similarly with minor adjustments to the design equations.

The transfer function of the prototype, linear-phase FIR filter is given by

$$H(z) = \sum_{n=0}^{N-1} h(n)\, z^{-n} \tag{2.1}$$

where $N$ is the length of the filter, and $h(n)$ is the impulse response. We define a new shifted transfer function

$$H_s(z) = H(z) + \delta_2\, z^{-\frac{N-1}{2}} = \sum_{n=0}^{N-1} h(n)\, z^{-n} + \delta_2\, z^{-\frac{N-1}{2}} \tag{2.2}$$

where $\delta_2$ is the stopband ripple. In effect, this amounts to adding the deviation in the stopband to the middle coefficient of the linear-phase filter. The addition of the extra term in the transfer function has the effect of shifting the amplitude response of the linear-phase filter by $\delta_2$ to make it positive for all frequencies. In Appendix A it is shown that the frequency response of a type 1, linear-phase filter can be separated in an exponential term and the real amplitude response as follows

$$H(f) = e^{-j2\pi fM}\left[\sum_{k=1}^{M} a(k)\cos 2\pi fk + a(0)\right] = e^{-j2\pi fM} A(f) \qquad (2.3)$$

where

$$a(0) = h(M) \qquad a(k) = 2h(M-k) \qquad k=1,\ldots,M \qquad M = \frac{N-1}{2} \qquad (2.4)$$

Since $a(0)=h(M)$, the addition of the stopband deviation to the middle coefficient of the impulse response is equivalent to adding $\delta_2$ to the amplitude response. Thus, the amplitude response becomes nonnegative.

The shifting of the amplitude response moves consecutive zeros on the unit circle (stopband zeros) to meet at the same point and, thus, make them double zeros. This can be seen using Figure 2.1. The filter with amplitude response shown in Figure 2.1 has 11 zeros in the stopband corresponding to the zero crossings of the response with the frequency axis. By gradually shifting the amplitude response upwards, the zero crossings move towards each other in pairs until they meet. For example the frequency crossings $f_1$, $f_2$ move towards each other until they meet on the frequency axis at $f_3$. Therefore, the zeros on the unit circle become double without changing the shape of the amplitude response.

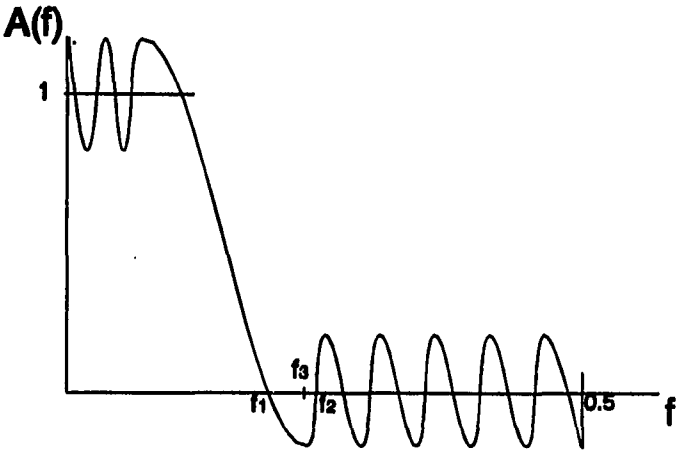


Figure 2.1. Amplitude response of a linear phase lowpass filter. There are 11 zeros on the unit circle corresponding to stopband zeros

The function $H_s(z)$ can be factored in a product of two polynomials of equal degree given by

$$H_s(z) = H_{mp}(z)\ H_{mp}(z^{-1}) \tag{2.5}$$

where $H_{mp}(z)$ consists of the zeros inside the unit circle and one of each of the double zeros on the unit circle. The function $H_{mp}(z^{-1})$ contains the remaining zeros. The two complex polynomials have identical stopband zeros on the unit circle while zeros that are not on the unit circle are symmetrical with respect to the unit circle because of the symmetry of the impulse response. Consequently the two functions are the same when evaluated on the unit

circle and their magnitude response is the square root of the magnitude of the original function. This is the main point of the technique since it separates the zeros that result in a minimum-phase filter.

### 2.2.2 Scaling of the Amplitude Response

The shifting of the frequency response results in an amplitude response that oscillates between $1+\delta_1+\delta_2$ and $1-\delta_1+\delta_2$ in the passband. We can scale the amplitude response of the minimum-phase function $A_{\min}(f)$ to oscillate between $1+\delta_{pnew}$ and $1-\delta_{pnew}$, where $\delta_{pnew}$ is the new passband ripple. Let $\delta_1$ denote the passband deviation, and $\delta_2$ the stopband deviation of the prototype, linear-phase filter. After shifting the amplitude response by $\delta_2$, the passband ripple varies between $1+\delta_1+\delta_2$ and $1-\delta_1+\delta_2$, and the stopband ripple varies between 0 and $2\delta_2$. If we let $s$ denote the scaling factor, and $\delta_{pnew}$ the new minimum-phase filter passband ripple, then the following relations hold:

$$1 + \delta_{pnew} = \sqrt{(1+\delta_1+\delta_2)\, s} \qquad 1 - \delta_{pnew} = \sqrt{(1-\delta_1+\delta_2)\, s} \tag{2.6}$$

If we combine the two equations and eliminate $\delta_{pnew}$, we obtain the required scaling factor which is given by

$$s = \frac{1}{\delta_1^2}\left(\sqrt{1 + \delta_1 + \delta_2} - \sqrt{1 - \delta_1 + \delta_2}\right)^2 \qquad (2.7)$$

Shifting and scaling the amplitude response of the linear prototype filter changes the location of the zeros without changing the shape of the amplitude response. The resulting zeros can be separated to give the minimum-phase filter. Next, we summarize the basic steps of the algorithm for the design of a minimum-phase filter.

### 2.2.3 Algorithm

The four basic steps required to design a minimum-phase filter of length $N$ using polynomial factorization are the following:

1. Design an optimal Chebychev linear-phase filter of length *(2N-1)* using the Remez algorithm and the Parks-McClellan program to obtain a filter with amplitude response $A(f)$, which has a passband ripple $\delta_1$, and a stopband ripple $\delta_2$.
2. Shift the resulting filter frequency response by adding the stopband deviation $\delta_2$ to the amplitude response so that $A(f) + \delta_2$ is positive for all frequencies. This can be done by adding the stopband ripple to the mid-power coefficient of the complex

linear-phase polynomial. The result is a shifting of the zeros on the unit circle towards each other until they meet, and thus become double zeros.

3. Scale the resulting response by the factor $s$, where

$$s = \frac{1}{{\delta_1}^2}\left[\sqrt{1 + \delta_1 + \delta_2} - \sqrt{1 - \delta_1 + \delta_2}\right]^2 \qquad (2.8)$$

to center the amplitude response at one in the passband.

4. Factor the transfer function of the scaled filter in step 3, keeping all the zeros inside the unit circle and one of each of the double zeros on the unit circle. These zeros constitute the minimum-phase system. Multiply these zeros to get the impulse response of the minimum-phase filter.

### 2.2.4 Polynomial Factorization

When long filters are being designed, step 4 becomes difficult. With a high degree polynomial, most of the root-finding techniques fail to give accurate results, or they do not converge. It was discussed before that the shifting of the amplitude response of the prototype linear-phase filter has the effect of moving pairs of consecutive zeros on the unit circle to meet at the same point. As it was seen in Figure 2.1, the frequency location of the double

zeros on the unit circle is an extreme frequency of the optimal linear-phase filter. Since the stopband zeros have magnitude one, and their phase is known from the Parks-McClellan program output, the locations of the stopband zeros are known.

The rest of the zeros can be located by deflation of the original polynomial. When the roots on the unit circle are found, the original polynomial is factored in two polynomials, one containing the unit circle roots and one containing the remaining zeros. Deflation can be used in cases where the degree of the polynomial is not very high. Since the location of each root is known with finite accuracy, deflation can cause large errors in the remaining coefficients, thus causing inaccurate results. Since this work deals only with moderate degree minimum-phase filters, this is not a very crucial problem to us.

To find the remaining prototype filter zeros we have used Laguerre's method [13]. This method guarantees convergence to a zero from any starting point [13]. Another reason we use this method is its easy implementation. Following, we give the basics of Laguerre's method [13]. Given a polynomial $P(x)$ of order $N$ with zeros $x_i$, $i = 1, \ldots, N$, the following relations hold:

$$P(x) = (x - x_1)\ (x - x_2) \ldots (x - x_N) \tag{2.9}$$

$$\ln|P(x)| = \ln|x - x_1| + \ln|x - x_2| + \ldots + \ln|x - x_N| \tag{2.10}$$

$$\frac{d\ln|P(x)|}{dx} = \frac{1}{x - x_1} + \frac{1}{x - x_2} + \ldots + \frac{1}{x - x_N} = \frac{P'}{P} \equiv G \tag{2.11}$$

$$-\frac{d^2\ln|P(x)|}{dx^2} = \frac{1}{(x - x_1)^2} + \ldots + \frac{1}{(x - x_N)^2} = \left(\frac{P'}{P}\right)^2 - \frac{P''}{P} \equiv H \tag{2.12}$$

The method assumes that the root $x_1$ is at distance $a$ from the current guess $x$, and the rest of the roots are at distance $b$. Using the relations above

$$\frac{1}{a} + \frac{n-1}{b} = G \qquad \frac{1}{a^2} + \frac{n-1}{b^2} = H \tag{2.13}$$

which result in the distance $a$ given by

$$a = \frac{n}{G \pm \sqrt{(n-1)\,(nH - G^2)}} \tag{2.14}$$

with the sign chosen to make the denominator the largest. The algorithm operates iteratively. At each iteration the value of $a$ is calculated for a trial value of $x$. The next trial uses the value $x$-$a$, and this continues until the value of $a$ becomes smaller than a predetermined value. In our implementation this value has been set at $10^{-14}$. This value has been found to provide

a balance between accuracy and inability to converge. The search for a zero begins at the origin of the complex plane. If the routine fails to locate the zero in some number of iterations, a random location inside the unit circle is chosen, and the procedure repeats. In our simulations, no more than two trial starting locations were needed for any zero.

## 2.3 Conclusion

We discussed the design of FIR minimum-phase filters using the Herrmann and Schuessler algorithm. The minimum-phase filter is derived from a prototype linear-phase filter using polynomial factorization. The prototype double-length, linear-phase filter is designed using the Parks-McClellan program. The amplitude response of this filter is shifted to make the zeros double on the unit circle. The response is then scaled to make the amplitude response equiripple at one in the passband.

The double-length polynomial is then factored into two equal degree polynomials. One of the polynomials represents the minimum-phase filter, and contains one of each of the double zeros on the unit circle and all the zeros inside the unit circle. The linear-phase polynomial is factored using Laguerre's method [13]. The method gives satisfactory results for minimum-phase filter designs with orders below 70. For higher order filters the method fails to give satisfactory results in most case because of the difficulty locating the zeros of high degree polynomials. This problem is common to most root-finding algorithms.

# 3. IMPLEMENTATION AND RESULTS

## 3.1 Implementation Notes

The minimum-phase filter design method was implemented with a computer program written in FORTRAN. The prototype, linear-phase filter is designed using the Parks-McClellan program [5]. For convenience, we incorporated all the design steps of the algorithm into one program. In our implementation we only examine the design of frequency selective filters. The input to the program is similar to the input of the Parks-McClellan program [5]. Specifically, the input consists of, 1) the order of the minimum-phase filter, 2) the frequency edges of each band, 3) the desired amplitude value in each band and, 4) the weight in each band. Usually, the desired amplitude value is one in the passbands and zero in the stopbands while no specification is given for the transition bands.

A large weight can be applied to the stopbands if a smaller error is desired in the stopbands compared to the error in the passband. It is found that a large weight in the stopband does not affect the error considerably in the passband as it does in the case of linear-phase filter designs. Therefore, a large stopband weight can be used to achieve better attenuation in the stopband.

The output of the program consists of 1) the impulse response coefficients of the minimum-phase filter, 2) the zeros, 3) the magnitude response and, 4) the group delay. It is

rather easy to calculate the frequency response of the filter when the zeros are available [2]. This amounts to calculating the effect of each zero at discrete points on the unit circle. The frequency response of the filter can be written as

$$H(f) = G\, e^{-j2\pi fN} \left(e^{j2\pi f} - z_0\right) \ldots \left(e^{j2\pi f} - z_{N-1}\right) \tag{3.1}$$

where $G$ is the gain, and $N$ is the length of the filter. Each factor can be written as

$$e^{j2\pi f} - z_k = V_k(f)\, e^{j\phi_k(f)} \tag{3.2}$$

where $V_k(f)$ is the magnitude and $\phi_k(f)$ is the phase of each factor on the unit circle. The magnitude and phase of the frequency response can be calculated at equally spaced points of frequency by

$$|H(f)| = |G|\, V_0(f)\, V_2(f) \ldots V_{N-1}(f) \tag{3.3}$$

and

$$\Phi(f) = -2\pi fN + \phi_0(f) + \ldots + \phi_{N-1}(f) \tag{3.4}$$

The group delay can be computed from the zeros, as indicated in Appendix C, and it is given by

$$\tau(f) = \sum_{k=0}^{N-1} \frac{r_k^2 - r_k \cos(2\pi f - \theta_k)}{1 + r_k^2 - 2r_k \cos(2\pi f - \theta_k)} \tag{3.5}$$

where $z_k = r_k e^{j\theta_k}$, $k = 0, \ldots, N-1$, are the zeros of the filter.

## 3.2 Design Examples

We present two design examples. The first example is a lowpass filter and the second example is a bandpass filter. Since the order of these filters is not very high, the execution time is very small. The total execution time that includes the design of the prototype, linear-phase filter, does not take more than a few seconds on a fast personal computer, e.g. an IBM-compatible 386-33 MHz computer.

### Example 3.1

This example is a lowpass minimum-phase filter of length 39. The linear-phase prototype filter has length 77. The passband is defined on [0,0.33], and the stopband on [0.375,0.5]. A weight of 1:10000 is chosen to give good attenuation in the stopband. The passband error for the linear-phase prototype is 3.858e-2 and the stopband error is 3.858e-6.

Using these values we can derive the expected theoretical ripple values for the minimum-phase filter which are 0.0193 (0.166 dB) in the passband, and 0.00278 (51.12 dB) in the stopband. The minimum-phase filter constructed from the proper zeros has a passband ripple of 0.285 (0.244 dB) and stopband attenuation of 0.0028 (51.06 dB). The magnitude response in dB is shown in Figure 3.1. Figure 3.2 shows the passband magnitude error. The maximum magnitude error is the same as the theoretical value in most of the passband except at the edge of the passband where the error attains a maximum of 0.285 (0.244 dB). The stopband attenuation is almost the same as the expected theoretical value.

Figure 3.3 shows the zeros of the minimum-phase design. The stopband zeros are on the unit circle and the passband zeros are inside the unit circle. A plot of the group delay is shown in Figure 3.4. The passband group delay is smaller than that of a linear-filter of the same length but it is not constant. The nonlinearity of the phase is stronger in the transition band where the delay jumps to a new value. In the passband the deviation of the delay from a constant is small.

For comparison, a linear-phase filter of the same length and cutoff frequencies has been designed using the Parks-McClellan program. A weight of 1:10 has been chosen for comparable results. A larger weight in the stopband gives a large ripple in the passband. The linear-phase filter has a passband ripple 0.4 dB, and stopband attenuation 46.7 dB. Thus, the minimum-phase filter gives an improvement of about 4.36 dB in the stopband attenuation compared to the linear-phase filter.

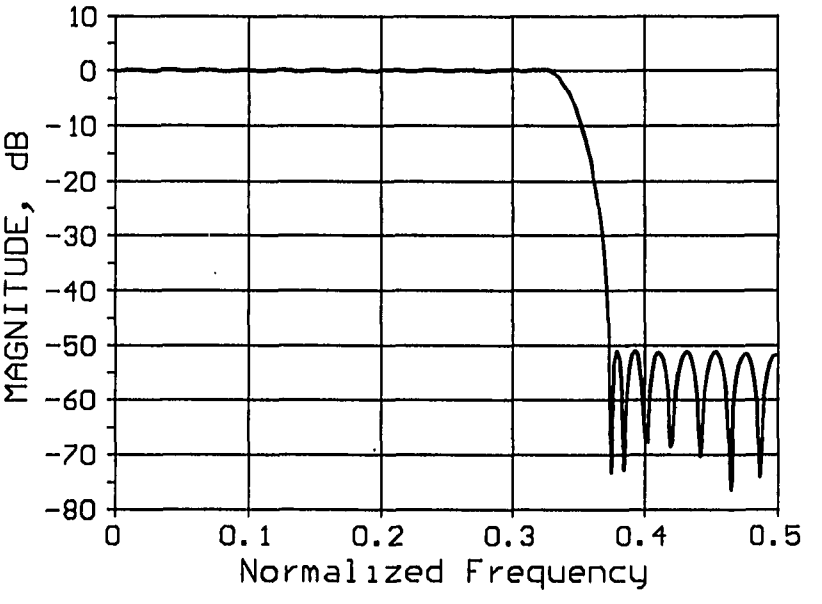


Figure 3.1. Magnitude response in dB of the minimum-phase filter in example 3.1

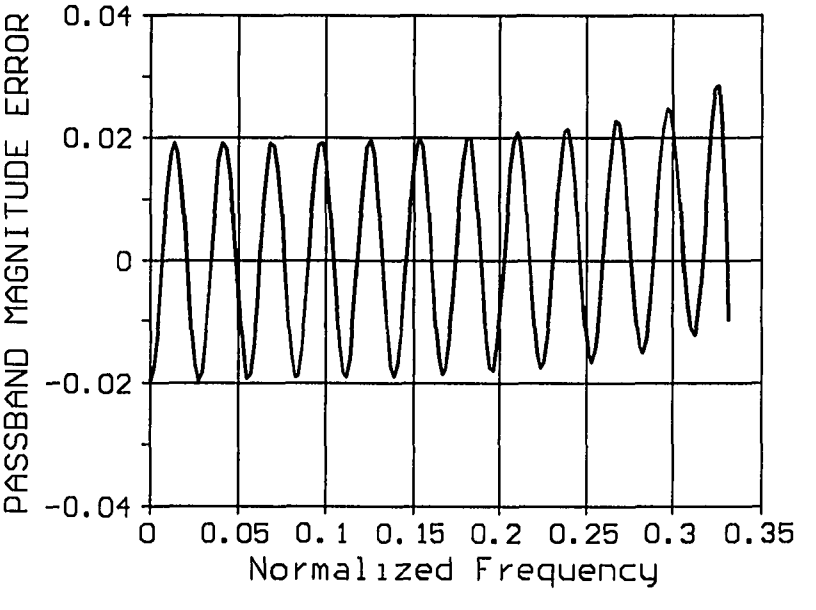


Figure 3.2. Passband magnitude error of the minimum-phase filter in example 3.1. The error is equiripple in most of the passband

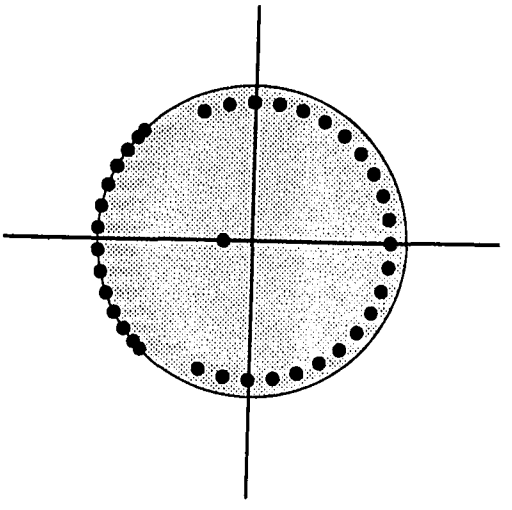

Figure 3.3: Zeros of the minimum-phase filter in example 3.1

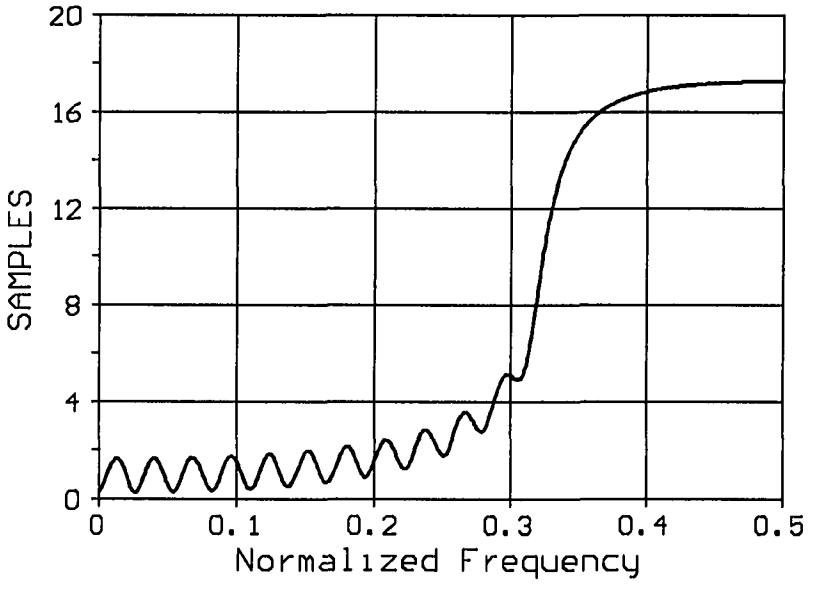


Figure 3.4: Group delay of the minimum-phase filter in example 3.1

**Example 3.2**

The second example, taken from [8], is a bandpass filter with stopbands [0 , .1] and [.33 , .5], and passband [.14 , .29]. The length of the filter is 50, implying that the prototype linear-phase filter required has length 99. The error weighing used is 3000:1:3000. The prototype linear-phase filter has a passband ripple of 0.01568 and stopband attenuation of 5.225e-6 (105.64 dB). Figure 3.5 shows the magnitude response and Figure 3.6 shows the group delay. The zeros of the filter are shown in Figure 3.7. The resulting minimum-phase filter has a passband ripple 0.01 (0.086 dB) and stopband attenuation 0.0033 (49.63 dB). These results are comparable to the results in [8]. It is seen again that a large stopband weight can used in the design of minimum-phase filters.

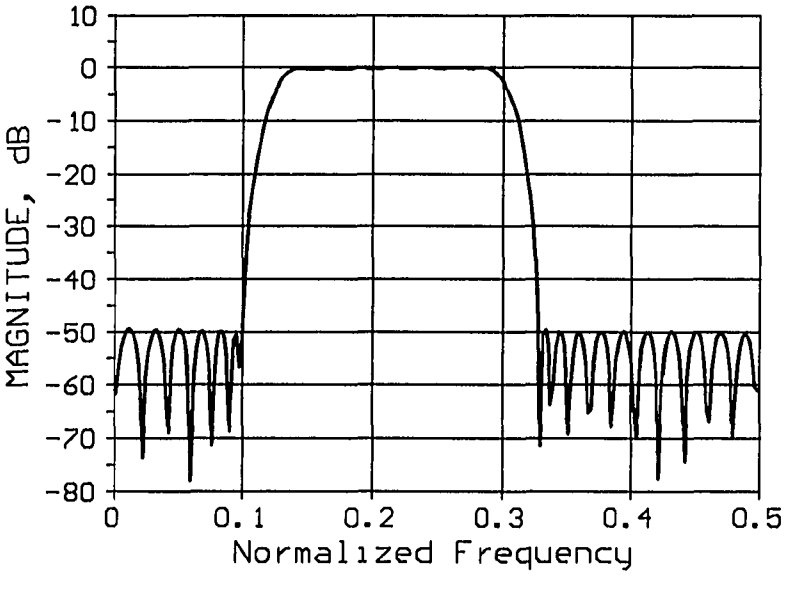


Figure 3.5: Magnitude in dB of the bandpass filter in example 3.2

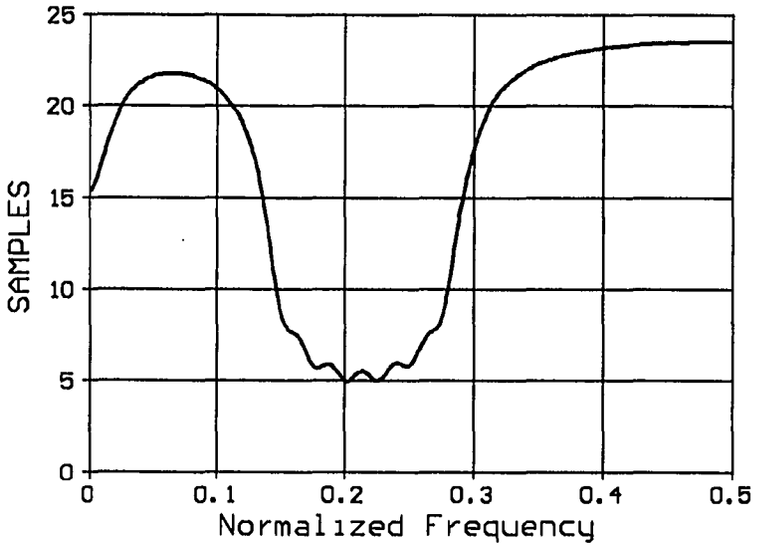


Figure 3.6 Group delay of the bandpass filter in example 3.2

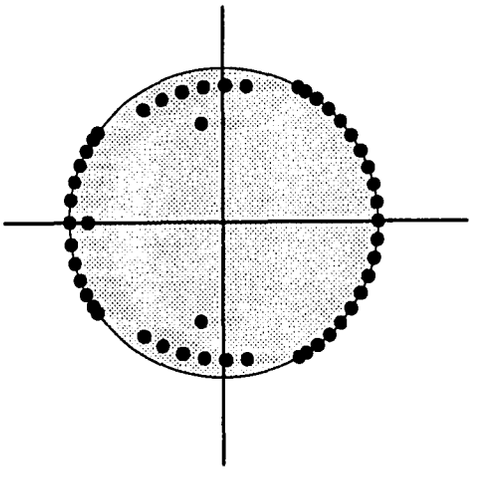

Figure 3.7 Zeros of the bandpass filter in example 3.2

## 3.3 Conclusion

We discussed our implementation of the FIR minimum-phase filter design method initially proposed by Herrmann and Schuessler [7]. For the design of a minimum-phase filter of a given length, a prototype double length linear-phase filter is designed using the Parks-McClellan program [5]. A large weight can make the stopband attenuation larger for minimum-phase filters without affecting the passband ripple severely. This can be used to design filters with same attenuation as linear-phase filters but shorter length.

The main problem associated with this method is the need to find the roots of a high degree polynomial when the order of the desired filter is large. For minimum-phase filter designs of length less than 70 our implementation is sufficient to produce good designs. For longer filters, a more detailed approach should be followed in the location of the zeros of the prototype linear-phase filter, since our implementation fails to give good results consistently. Also, the impulse response is obtained by multiplying the zeros of the minimum-phase filter. For higher order designs, this approach will introduce numerical error. A more appropriate method is to use a long inverse FFT. This approach has been used in [8] for the design of long filters.

Two minimum-phase design examples were presented. It was seen that the minimum-phase filters gave slightly better magnitude response characteristics compared to linear-phase filters of the same length. Also, the group delay in the passband is much smaller compared to the half length constant group delay of linear-phase filters. The disadvantage is that the

group delay is non-constant. This is undesirable when group delay distortion cannot be tolerated. Also, since no symmetry of impulse response exists, the computational benefit of using only half of the filter coefficients is lost.

# PART II : DESIGN OF FIR FILTERS IN THE COMPLEX DOMAIN

# 1. INTRODUCTION

## 1.1 Introduction

An FIR digital filter design method is presented that allows approximation to an arbitrary complex-valued frequency response using complex approximation. The minimax criterion is used with the complex Chebychev approximation being posed as a linear optimization problem. The primal minimization problem is converted to its dual equivalent maximization problem, and is solved using the efficient, quadratically convergent algorithm first developed by Tang [14]. This method results in optimal Chebychev designs without the need of discretization.

In several applications where FIR filters are used, the linear-phase constraint of the Parks-McClellan method imposes unnecessary limitations on the design. In general, the relaxation of the linear-phase requirement results in designs that achieve nearly the same magnitude characteristics as linear-phase filters but with less coefficients. One design method that relaxes the phase linearity constraint was examined in Part I, which results in minimum-phase filters. Minimum-phase filters are valuable in applications where severe delay distortion is not a problem and smaller delay is needed. When control over the phase function is required the technique presented here is useful. The most important reason FIR filter designs other than linear-phase are required is the need to obtain freedom in specifying

arbitrary filter magnitude and phase characteristics to meet specifications of a wide variety of applications.

The complication introduced by relaxing the linear-phase constraint is that the filter impulse response is no longer symmetric. Unlike the transfer function for linear-phase filter design, the transfer function of the nonlinear-phase filter cannot be separated in a product of a real analytic amplitude function and an exponential. Therefore, an approximation to an amplitude response is performed in conjunction with the phase characteristic using complex functions. Since the sum of the approximating functions must constitute the transfer function of an FIR filter, the approximating functions must be complex exponentials. This filter design is a complex approximation problem in contrast to a real approximation problem in the case of linear-phase FIR filter design.

In this chapter, we discuss in general terms the basic ideas behind the new design method for FIR filters that can approximate a complex desired frequency response. We discuss the benefits of the method as well as the difficulties implied by the extension of the filter design to account for the phase function. We also review some of the other FIR filter design methods that consider the phase function in the approximation.

The filter design method proposed here uses the Chebychev norm as the criterion of goodness, producing FIR filters with almost equiripple magnitude response. The FIR filter design problem with arbitrary specification of magnitude and phase is a Chebychev approximation on the complex plane. The Chebychev approximation problem is interpreted as a minimization of a linear optimization problem. The primal problem is converted to its

dual which is solved using an efficient algorithm based on the Remez Exchange algorithm. The formulation of the FIR filter design into a complex approximation problem is presented in Chapter 2. As we will see later, when the approximated desired frequency response is not conjugate symmetric, the impulse response of the filter must be a complex sequence. Initially, the problem formulations of the real and complex impulse responses are treated separately. When the primal complex approximation problem is developed, a common form is generated for the two problems allowing their simultaneous treatment.

The dual of the primal approximation problem is solved by a powerful quadratically convergent algorithm developed by Tang [14]. The algorithm is presented in Chapter 3. The advantage of solving the dual problem is that there is no need to replace the frequency and angle domains by finite sets. Also, the infinite number of constraints, due to the continuous frequency domain in the primal problem, becomes finite in the dual which makes its solution easier. This is an advantage of our method over existing design methods. More about the advantages of our design method is discussed in the next section which focuses on other methods capable of designing filters with nonlinear phase. Also, some of the benefits of our method are discussed with the presentation of the algorithm.

The implementation and results of the design method are the subject of Chapter 4. The design program is written in FORTRAN. We discuss the use of the program to design various types of FIR filters. We show that the design of linear-phase filters is a special case of our filter design method. Typical filter design examples and the adjustments required for each class of filters are discussed. The design examples presented include, 1) linear-phase

filters to show the generalization of the Parks-McClellan algorithm, 2) conjugate-symmetric FIR filters (real impulse response coefficients) with smaller group delay than that of linear-phase filters, 3) FIR filters approximating a non-conjugate symmetric response (complex impulse response coefficients), 4) Hilbert transformers, and 5) differentiators. The wide variety of examples is presented to show the versatility of the design algorithm and computer program.

## 1.2 Complex Filter Design Methods

The design methodologies and algorithms for FIR filters that approximate complex-valued frequency responses are more difficult in both theory and implementation when compared to linear and minimum-phase design techniques. In real approximation, the problem can be solved on a small number of points, called extremal points, which characterize the best polynomial approximation. The number of these points is known for each problem, and the real Remez algorithm can be used to locate these points that result in the best approximation in the Chebychev sense. The use of the real Remez algorithm in the design of FIR filters is discussed in Appendix A. While this is true in real approximation, the number of extremal points of the best polynomial approximation to a complex-valued function in the Chebychev sense is not known. Therefore, the alternation theorem and the real Remez algorithm cannot be used to find the best approximation.

The most basic form of the digital FIR filter design problem is the following. Given a desired frequency response $D$, representing a set of design specifications, and an approximating filter,with frequency response $H$, define the complex error function $E=D-H$. The filter design problem is to determine the set of impulse response coefficients that minimize a measure of the complex error magnitude, $|E|$. The most common measures used in the minimization are the 1-norm, 2-norm, and the Chebychev norm. The most effective methods use the 2-norm or the Chebychev norm. The use of the 2-norm is associated with minimization of the error magnitude in the least squares sense, which in effect minimizes a quadratic form of the error. The Chebychev norm is used in the minimax sense, meaning that the maximum of the complex error magnitude is minimized. The design method we develop here uses the Chebychev norm as the criterion of fitness for the approximation. The complete formulation of the filter design problem as a complex approximation problem is the subject of the next chapter.

The approximation of a complex frequency response is generally a problem of complex approximation. However, when linear-phase filters are designed, the linearity of the phase can be used to reduce the original complex approximation problem to one of real approximation. The real approximation problem and the design of linear-phase filters is discussed in Appendix A. Also, the design of minimum-phase filters is a real approximation problem since minimum-phase filters can be derived from prototype linear-phase filters. Because of the difficulty of the complex approximation problem due to the inability to use the alternation theorem, most of the design approaches approximating a complex frequency

response seek transformations of the complex problem to one of real approximation. Next, we discuss some of the existing methods of designing FIR filters with specification of magnitude and phase functions.

One of the methods for designing FIR filters in the complex domain was developed by Chen and Parks [15]. Their approach is based on a complex approximation method developed by Glashoff and Roleff [16], and Streit and Nuttall [17]. The basic idea behind this design method was to convert the complex approximation problem into a real approximation problem which is nearly equivalent to the complex problem. The original complex approximation problem using the Chebychev norm is a nonlinear optimization problem in the real domain since the magnitude of the complex error, which is minimized, is a nonlinear function of the approximation coefficients. The Chebychev approximation minimizes the magnitude of the complex error function $E = Re(E) + j\,Im(E)$. Although the functions $Re(E)$ and $Im(E)$ are linear functions of the approximation coefficients, the minimized function $|E| = [(Re\,E)^2 + (Im\,E)^2]^{1/2}$ is not. A transformation of the original problem is possible [16], converting the nonlinear problem to a linear optimization problem with a finite number of variables and an infinite number of constraints. The transformation introduces an angle variable in the approximation. This is called a semi-infinite problem. A discretization of this problem of both the frequency and angle variables results in a standard linear programming problem.

The solution of the real approximation problem is obtained using a standard linear programming algorithm for the Chebychev solution of overdetermined equations [18]. The

discretization of the semi-infinite problem results in solving an approximation to the original problem. Solutions near the optimal can be obtained when fine grids of angle and frequency are used in the discretization process. The problem becomes prohibitively large and complicated when the filter orders are high and large CPU time and storage requirements are necessary. A coarser grid results in solutions that are not as close to the optimal.

Even though good filter designs can be obtained when a fine grid is used, this solution to the problem is an approximation to the original problem because of the discretization. Our technique does not have this limitation. Because the problem of infinite constraints is transformed to its dual, the number of constraints becomes finite. Then, the problem no longer requires discretization. We will elaborate on this later. Also, Chen and Parks only consider the design of conjugate symmetric filters, which result in real impulse response coefficients. In our work, conjugate symmetric filters are a special case of the general complex filter design problem.

Preuss [19] [20] derived an approximation algorithm that resembles the real Remez algorithm. The difference of this method from other methods is that it deals directly with the complex error. The method is capable of producing filters with complex coefficients. First, a set of extremal points is assumed for the magnitude of the complex error. These points are used to interpolate an FIR filter using the Newton interpolation formula. Then the complex error is computed using the FFT, and the extremal frequencies of the error magnitude are found. For a filter of length $N$, $N+1$ frequencies with the largest error deviations are used to interpolate a new filter. The magnitudes of the error at the extremal

points are modified, with a controlling factor, before the interpolation. The angles of the error used in the interpolation are the same as in the previous iteration.

The algorithm terminates when the magnitude deviations of the error at the extremal frequencies are equal within a prescribed tolerance. In Preuss' algorithm there appears to be no theoretical proof of convergence. Also, when it does converge, no proof is given that the approximation is optimal in some sense. As reported by the author [19], this algorithm has significant improvements in speed compared to the algorithm by Chen [15]. However, it is mentioned in [19] that the algorithm may diverge when the controlling factor is large.

Schulist [21] improved the convergence of Preuss' algorithm by modifying the way the set of extremal points is selected from the points where the magnitude of the complex error is maximum. He also proposed a modification on the way the interpolation of the error is made and replaced the Newton formula by a Gaussian relaxation algorithm. The author reports significant improvements in convergence of the algorithm as well as improvement of the numerical instabilities discussed by Preuss.

Chit and Mason [22] reported a new approach for the design of FIR filters with arbitrary specification of magnitude and phase. The method differs significantly from other methods in that it is an adaptive procedure, trying to find the suitable real-valued costs of the least mean square (LMS) approximation to give an optimum solution in the Chebychev sense. The procedure is called Double Adaptive Systems (DAS), and consists of two processes. The first procedure computes the coefficients of the filter under design using an LMS minimization, and the second is a weight adapting scheme designed to give the filter a

Chebychev characteristic with respect to the desired function. The coefficients of the filter design are adjusted depending on the initial costs and the frequency response of the current filter. The same procedure has been used by the authors to design linear and minimum-phase filters [23] [12].

A number of researchers approached the solution of the complex problem by approximating the magnitude and phase, or real and imaginary parts of the complex desired response separately. Holt et. al. [24], and Cuthbert [25] proposed a method of approximating the absolute value of the real and imaginary parts of the FIR filter to a desired response. The coefficients of the final filter are generated from the coefficients of the real and imaginary approximations. These techniques do not result in optimal Chebychev designs since special properties should be imposed on the coefficients. They consider the coefficients as being the sum of two symmetric sets, one possessing even symmetry and the other odd symmetry. Cortelazzo and Lightner [26], [27] applied a multiple criterion optimization technique to a specification of both gain and group delay. As the authors observe, their filter design method requires considerable computation time and it is only reliable for FIR filters with orders not higher than ten. Therefore, the technique is not practical for real applications.

## 1.3 Proposed FIR Filter Design Method

Our method of designing FIR filters, with arbitrary specification of magnitude and phase, produces optimal filters in the Chebychev sense, i.e., the maximum of the complex error magnitude is minimized. The method is capable of producing FIR filters approximating non-conjugate frequency responses. These filters require complex impulse response coefficients. Therefore, no special symmetries of the impulse response, or the desired frequency response, are necessary for our method.

The filter design problem is formulated as a Chebychev approximation in the complex domain. The approximation problem is a semi-infinite problem since there is a finite number of variables to be determined (the impulse response coefficients), subjected to an infinite number of constraints due to the continuous frequency. Our approach to solve the primal semi-infinite problem is to first derive its dual problem. In the dual problem the infinite number of constraints becomes finite. Handling a problem with finite number of constraints simplifies the solution of the problem. Also, in contrast to the method in [15], our method does not require discretization of the complex domain and optimal Chebychev solutions are obtained.

The development of the filter design method we present is strongly motivated by the work of Peter Tang [14]. His work was successful in developing a powerful algorithm to solve the complex Chebychev approximation problem without the need of discretization. Also, his algorithm, under certain conditions [28], converges quadratically. These conditions

require the magnitude of the complex error to exhibit N+1 extremal points, and the second derivatives of the error at these points to be nonzero. The algorithm serves as a generalization of the Remez algorithm in the complex domain. The thorough study of the subject performed by Peter Tang [14], [28] made possible the use of his algorithm for the development of a general, powerful FIR filter design method. Also, our personal contacts with Peter Tang were very important for the development of this filter design method.

The main advantages of the filter design method presented here are the following: 1) No discretization of the domain of approximation is required, i.e. it does not require replacement of the continuous frequency and angle intervals by finite sets. Also, when the frequency domain is discretized for easier implementation, no discretization of the angle domain is needed. 2) The complexity, and the memory requirements of the approximation do not increase considerably with increasing the order. As it will be clear later, the size of the matrices and vectors used in the approximation are linearly depended on the length of the filter. This is an advantage over the design method presented in [15]. 3) The algorithm does not suffer from numerical instabilities. This makes the method practical for design of high order filters. 4) The algorithm converges quadratically to the optimal solution. The last two points are advantages of our method over the method presented in [19].

We have adopted Tang's algorithm to fit our requirements for the design of FIR filters with arbitrary specification of magnitude and phase. As we will see in the next chapter, a specification of the magnitude and phase is done for each band of the filter. Since any meaningful desired complex frequency response can be specified, a large variety of design

applications is covered by the method. In this dissertation we show the design of the common frequency selective filters, e.g., lowpass and bandpass filters, the design of filters with non-conjugate symmetric frequency responses, one-sided and two sided Hilbert transformers, and differentiators. The method is also able to produce a system that performs a different function in each band, e.g., a system which is a bandpass filter for positive frequencies and a differentiator for negative frequencies. Also, since any specification of a desired frequency response is accommodated, the method can produce systems that have very specific requirements for magnitude and phase functions. Thus, the method can be employed for the design of phase equalizers.

## 1.4 Conclusion

We have provided some background of the design of FIR filters with separate specification of the magnitude and phase functions. In general, this is a complex approximation problem. The design of FIR filters in the complex domain is probably the most complicated approach to design FIR filters, but it offers freedom in designing filters that exactly fit various specifications. Also, in most cases the number of the required impulse response coefficients is generally smaller compared to exactly linear-phase designs for the same magnitude response. In applications that require a specific nonlinear-phase response such as phase equalizers, FIR filters designed in the complex domain are the best solution.

The method offers other possibilities such as the design of nearly linear-phase filters with a smaller group delay than that obtained with linear-phase filters.

We reviewed some of the existing methods of designing FIR filters with non-linear phase response and contrasted them with our proposed method. Our method offers several advantages over existing techniques. First, no discretization is required to perform the design. This allows solutions that are closer to the optimal without the need to increase the complexity of the problem. Also, long filters can be designed without the complexity of the problem increasing significantly. The algorithm is stable, and it guarantees a solution. Another very important advantage is that no symmetries of the impulse response, or the frequency response, need to be assumed. This allows the design of filters with non-conjugate responses. These filters necessarily have complex impulse responses. As we will see in the next chapter, the design problem is formulated such that complex coefficients are accommodated easily by only doubling the order of the approximation problem.

# 2. PROBLEM FORMULATION

## 2.1 Introduction

The design of FIR digital filters with specifications of both magnitude and phase responses is formulated into an approximation problem in the complex domain. The approximation uses the Chebychev norm as the optimality criterion. The resulting approximation problem is a nonlinear optimization problem with respect to the approximation coefficients. Using a simple transformation on the complex error function, the problem can be transformed into a linear semi-infinite optimization problem. This form of the primal optimization problem is converted to its dual equivalent since its solution is possible without discretization to the original problem. The dual problem is solved using an efficient generalized Remez Exchange algorithm.

First, we formulate the design of FIR filters that approximate conjugate symmetric frequency responses, but not necessarily linear phase. The conjugate symmetry of the frequency response results in real impulse response coefficients for the approximating filter. We then discuss the generalization of the filter problem to that of designing filters with non-symmetric frequency responses. This generalization requires filters with complex impulse response coefficients. The distinction between the two problems is necessary at this stage since, as we will see shortly, the approximating functions and the coefficients are defined

differently. The two problems are combined into an equivalent form that allows their simultaneous treatment, and also makes their conversion to the dual equivalent problem easier to interpret.

The dual of the primal approximation problem will be discussed next. In this section, no distinction will be made between real and complex coefficients since it will be clear that the approximation problem with complex coefficients is similar to the approximation problem with real coefficients. This is done by separating the complex coefficients into their real and imaginary parts. The dual problem is necessary since it transforms an optimization problem with infinite constraints to one with finite constraints. The dual problem, can be solved using an efficient algorithm based on the Remez algorithm, developed by Tang [14]. Some basic theory behind the transformation of the primal minimization problem to its dual equivalent maximization problem can be found in Appendix B.

## 2.2 Problem Formulation: Real Impulse Response

### 2.2.1 Problem Statement

We are required to approximate a complex frequency response with an FIR filter. The approximation will use the minimax criterion so that the maximum value of the complex error magnitude between the complex desired frequency response and the approximating filter is

minimized. The frequency response to be approximated must be conjugate symmetric in order that real coefficients are obtained. The desired response can be specified by a piecewise continuous function, on continuous disjoint normalized frequency intervals, or by complex data points. The specification of the desired response should provide transition bands between passbands and stopbands in order that meaningful filters are obtained. The discussion of the problem will assume a piecewise continuous desired frequency response since this development is more natural. In our implementation of the algorithm, the approximation of complex discrete data points was also realized.

### 2.2.2 Formulation

The complex approximation problem can be formulated as follows. We are required to approximate a complex-valued response $D(z)$, with a realizable FIR filter transfer function on the unit circle. In complex approximation theory, this translates to the problem of approximating a complex-valued function with a complex polynomial on the unit circle. An FIR filter is defined by its transfer function, given by the z-transform

$$H(z) = \sum_{k=0}^{n-1} h_k z^{-k} \tag{2.1}$$

and its frequency response, given by the Fourier transform

$$H(f) = \sum_{k=0}^{n-1} h_k \, e^{-j2\pi fk} \tag{2.2}$$

where the coefficients $\boldsymbol{h} = \{ h_0, \ldots, h_{n-1} \}$ represent the impulse response of the filter, $f$ is the normalized frequency in the interval [-0.5,0.5], and $n$ is the length of the filter. The frequency response is found by evaluating the transfer function on the unit circle.

Suppose the desired frequency response $D(z)$ is conjugate symmetric

$$D(z^*) = D^*(z) \tag{2.3}$$

The FIR filter design problem can be formulated into the following approximation problem. Find a set of optimal coefficients $\boldsymbol{h}' = (h_0', \ldots, h_{n-1}') \in \mathbb{C}^n$, where $\mathbb{C}^n$ denotes the $n$-dimensional vector space of complex numbers, such that

$$\| D(z) - H(\boldsymbol{h}', z) \| \leq \| D(z) - H(\boldsymbol{h}, z) \| \tag{2.4}$$

is satisfied for all $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbf{C}^n$, where $\|\bullet\|$ denotes the Chebychev norm, i.e., for a function $g(t)$ in [a,b], $\|g(t)\| = \max|g(t)|$ in [a,b]. The approximating function is given by

$$H(\boldsymbol{h}, z) = \sum_{k=0}^{n-1} h_k \, b_k(z) \qquad b_k(z) = z^{-k} \tag{2.5}$$

The functions $b_k(z)$, for $k=0,\ldots,n-1$, are the approximating basis functions. Because of the conjugate symmetry of the desired frequency response function, it can be proved [29] that the optimal coefficients are real and thus $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbf{R}^n$. Hence the filter impulse response is real. Moreover, the optimal impulse response $\boldsymbol{h}'$ satisfies

$$\|D(z) - H(\boldsymbol{h}', z)\| \leq \|D(z) - H(\boldsymbol{h}, z)\| \tag{2.6}$$

for all $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbf{R}^n$, where the Chebychev norm is taken on the unit circle on the upper half plane. The upper half unit circle corresponds to the normalized frequency range [0,0.5], which is the domain of approximation. Therefore the approximation of a conjugate symmetric frequency response requires a complex-valued polynomial with real coefficients, and only the normalized frequency interval [0,0.5] needs to be considered in the approximation.

## 2.3 Problem Formulation: Complex Impulse Response

### 2.3.1 Problem Statement

The complex-valued polynomial describing an FIR filter that approximates a non-conjugate complex-valued frequency response must have complex coefficients. The specification of the desired response will be given by a complex-valued function on piecewise continuous intervals, or by discrete complex data on a discrete frequency domain. The specification of the desired response should provide transition bands between passbands and stopbands in order that meaningful filters are obtained. The only requirement on the transition bands is that they exhibit a decreasing monotonic response. This is inherent in the approximation since no zeros exist in the transition bands.

### 2.3.2 Formulation

An FIR filter with complex impulse response is defined by its transfer function, given by the z-transform

$$H(z) = \sum_{k=0}^{n-1} h_{kc}\, z^{-k} \tag{2.7}$$

and its frequency response, given by its Fourier transform

$$H(f) = \sum_{k=0}^{n-1} h_{kc}\, e^{-j2\pi fk} \tag{2.8}$$

where $h_{kc}$ $k=0,\ldots,n-1$, are the complex impulse response coefficients of the filter, $f$ is the normalized frequency in the interval [-0.5,0.5], and $n$ is the length of the FIR filter. We separate the complex coefficients into their real and imaginary parts, and define them as follows:

$$h_{kc} = h_k + j\, h_{n+k} \qquad k=0\ ,\ 1\ ,\ \ldots\ ,\ n-1 \tag{2.9}$$

The transfer function can be put in the form

$$H(z) = \sum_{k=0}^{n-1} \left[ h_k\, (z^{-k}) + h_{n+k}\, (jz^{-k}) \right] = \sum_{k=0}^{2n-1} h_k\, b_k(z) \tag{2.10}$$

where the $2n$ basis functions are given by

$$b_k(z) = z^{-k} \qquad b_{k+n}(z) = j\, z^{-k} \qquad k = 0,\ \ldots\ ,\ n-1 \tag{2.11}$$

This representation allows the approximating polynomial to be expressed as a linear function of real-valued coefficients. The real coefficient vector is of order $2n$, defined as $\boldsymbol{h} = \{h_0, h_1, \ldots, h_{n-1}, h_n, \ldots, h_{2n-1}\}$. This vector will serve as the coefficient vector of the approximation problem. When the optimal solution is found, the complex impulse response of the filter is constructed from these coefficients according to Equation (2.9).

From this point, we can treat the problems of real and complex impulse responses simultaneously by replacing the upper index $n$ by $N$. We then need to remember that for the real case there are $N=n$ basis functions and coefficients, and for the complex coefficient case there are $N=2n$ basis functions and coefficients. The general form of the transfer function for the approximating FIR filter is

$$H(z) = \sum_{k=0}^{N-1} h_k \, b_k(z) \tag{2.12}$$

Since, in the complex coefficient case, the frequency response is not conjugate symmetric, the interval of approximation will be the normalized frequency interval [-0.5,0.5]. The frequency response of an FIR filter is the z-transform evaluated on the unit circle. This implies that the basis functions $b_k(z)$, $k=0,\ldots,N-1$, are complex exponentials that are functions of the normalized frequency $f$. The domain of approximation will be denoted by $F$, a finite set of disjoint intervals in the normalized frequency interval [-0.5,0.5]. When the desired frequency response is conjugate-symmetric, and the coefficients are real, the

frequency interval of approximation is $F \subset [0,.5]$. When the frequency response is non-conjugate symmetric the coefficients are complex, and the approximation interval is $F \subset [-.5,.5]$.

## 2.4 Another Form of the Primal Problem

The primal maximization problem is put in a form that makes its conversion to the dual problem possible. This equivalent formulation of the primal problem makes the constraints of the approximation linear with respect to the coefficients. The specification of the desired FIR filter is represented by a frequency response, a complex-valued function of the normalized frequency $f$. The approximating basis functions are complex exponentials on the unit circle.

Let $D(f)$ denote the desired complex valued frequency response on $F$, and let $H(f)$ be the approximating FIR filter. Define the weighted complex error by

$$E(f) = W(f)\,[D(f) - H(f)] \tag{2.13}$$

where $W(f)$ is a real-valued, positive weighting function introduced to control the relative magnitude of the complex error in the different frequency bands specified in the design. Find $N$ real parameters $h' = [h'_0, \ldots, h'_{N-1}]^T$ such that

$$\max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{N-1} h_k' \, b_k(f) \right| \leq \max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{N-1} h_k \, b_k(f) \right| \qquad (2.14)$$

for all $h$ in $\mathbb{R}^N$. The absolute error is smaller for the frequency bands with larger weight. Since the weighting function is positive, and can enter the magnitude factors of the expression above, we assume from this point in our discussion that the weight information can be carried with both the desired response and the approximating filter.

An equivalent formulation of the approximation problem is to find optimal variables $\delta, h_0, h_1, \ldots, h_{N-1}$ to minimize the linear function $0 \cdot h_0 + 0 \cdot h_1 + \ldots + 0 \cdot h_{N-1} + \delta$ (or $\delta$) subject to the constraints

$$\left| D(f) - \sum_{k=0}^{N-1} h_k \, b_k(f) \right| \leq \delta \qquad (2.15)$$

for all $f \in F$, and $h \in \mathbb{R}^N$. Note that, although the function to be minimized is linear with respect to the variables, the constraints are not. The constraints can be made linear by exploiting the fact [16] that for any complex number $z$

$$|z| = \max_{\alpha \in [0, 2\pi]} \left\{ Re\{ z \, e^{-j\alpha} \} \right\} \qquad (2.16)$$

This relation is used to transform the constraint equation into one that does not involve the magnitude operator. Using this relation, the constraint equation can be written in the form

$$\sum_{k=0}^{N-1} h_k \, Re\left\{ b_k(f) \, e^{-j\alpha} \right\} + \delta \geq Re\left\{ D(f) \, e^{-j\alpha} \right\} \tag{2.17}$$

for all $f \in F$, and for all $\alpha \in [0, 2\pi]$. If the sets $F$ and $[0,2\pi]$ were replaced by two finite sets, the problem would be transformed to a standard linear programming problem. This approach is called discretization of the semi-infinite optimization problem. It has been used and implemented by Chen and Parks [15] for the design of FIR filters. The solution of the problem can be obtained by applying the well-known Simplex method.

The above form of the constraint equation suggests that the approximation problem is an optimization problem in $\mathbf{R}^{N+1}$. This can be seen by relating the problem described above with the general form of the primal linear optimization problem discussed in Appendix B. We define the vectors $\beta$, $y$ and $a(f)$ in $\mathbf{R}^{N+1}$:

$$\beta = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \qquad y = \begin{bmatrix} h_0 \\ \cdot \\ \cdot \\ \cdot \\ h_{N-1} \\ \delta \end{bmatrix} \qquad a(f) = \begin{bmatrix} Re\{b_0(f) \, e^{-j\alpha}\} \\ \cdot \\ \cdot \\ \cdot \\ Re\{b_{N-1}(f) \, e^{-j\alpha}\} \\ 1 \end{bmatrix} \tag{2.18}$$

and the function

$$q(f) = Re\{D(f)\, e^{-j\alpha}\} \tag{2.19}$$

Then the primal minimization problem can be described as follows: Find the vector $y \in \mathbb{R}^{N+1}$, which minimizes the inner product $\beta^T \cdot y$, subject to the constraints

$$a^T(f) \cdot y \geq q(f) \tag{2.20}$$

over $f \in F$, and $\alpha \in [0,2\pi]$. This form of the problem is the same as that of the general primal optimization problem of order $N+1$ discussed in Appendix B.

The primal problem has a finite number of variables and an infinite number of constraints. Standard theory of linear optimization suggests that we should solve the dual of the primal problem since it is easier to confront a problem with fewer constraints than variables. Also, the solution of the dual does not involve any discretization of the problem. In the following section we formulate the dual approximation problem and show that there is no need to replace $F$ and $[0,2\pi]$ by finite sets.

## 2.5 Dual Problem

The primal complex approximation problem is converted to its dual equivalent. The dual form suggests an efficient algorithm for the determination of the optimal FIR filter in

the Chebychev sense. Supporting theory leading to the dual of the primal approximation problem can be found in Appendix B. The dual is derived because 1) the infinite constraints imposed by continuous frequency in the primal become finite in the dual, 2) an efficient algorithm can be used for its solution, and 3) the results are theoretically equivalent, i.e. no approximation is required for continuous frequency and angle. The *dual problem* of the primal approximation problem is stated below [14].

**Dual Problem:** Find $N+1$ frequencies $f_1, f_2, \ldots, f_{N+1} \in F$, $N+1$ angles $\alpha_1,\ldots,\alpha_{N+1} \in [0,2\pi]$, and $N+1$ non-negative weights $r_1, r_2, \ldots, r_{N+1}$ in [0,1] to maximize the inner product

$$d = \sum_{k=1}^{N+1} r_k \, Re\left\{D(f_k)\, e^{-j\alpha_k}\right\} \tag{2.21}$$

subject to the constraint:

$$\sum_{k=1}^{N+1} r_k = 1 \tag{2.22}$$

and

$$\sum_{k=1}^{N+1} r_k \, Re\left\{b_i(f_k)\, e^{-j\alpha_k}\right\} = 0 \qquad i = 0, 1, \ldots, N-1 \tag{2.23}$$

We can restate the problem in a more compact form using vector notation. We define the *(N+1)x1* vectors

$$f \in F^{N+1} \ , \ \alpha \in [0, 2\pi]^{N+1} \ , \ r \in [0, 1]^{N+1} \tag{2.24}$$

The *dual problem* can be expressed as follows. Find the vectors $f$, $\alpha$, $r$ to maximize the scalar

$$d = c^T \cdot r \tag{2.25}$$

subject to the constraints

$$A \cdot r = u_1 \tag{2.26}$$

where

$$c^T = \left[ c_1 \ c_2 \ \ldots \ c_{N+1} \right] = \left[ Re\left\{ D(f_1) e^{-j\alpha_1} \right\} \ldots \ Re\left\{ D(f_{N+1}) e^{-j\alpha_{N+1}} \right\} \right] \tag{2.27}$$

and the kth column of the *(N+1)x(N+1)* matrix $A$ is

$$\left[ 1 \ \ Re\left\{ b_0(f_k) \ e^{-j\alpha_k} \right\} \ \ Re\left\{ b_1(f_k) \ e^{-ja_k} \right\} \ \ldots \ Re\left\{ b_{N-1}(f_k) \ e^{-j\alpha_k} \right\} \right]^T \tag{2.28}$$

When the coefficients of the approximation are real, the elements of the matrix $A$ are cosine functions. When the coefficients are complex, the elements of the first half of the rows are cosines and the elements of the other half rows are sines. For the complex case the $(N+1)x(N+1)$ matrix $A$ is given by

$$A=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \cos(\alpha_1) & \cos(\alpha_2) & \cdots & \cos(\alpha_{N+1}) \\ \cos(2\pi f_1+\alpha_1) & \cos(2\pi f_2+\alpha_2) & \cdots & \cos(2\pi f_{N+1}+\alpha_{N+1}) \\ \vdots & \vdots & \cdots & \vdots \\ \cos[(n-1)2\pi f_1+\alpha_1] & \cos[(n-1)2\pi f_2+\alpha_2] & \cdots & \cos[(n-1)2\pi f_{N+1}+\alpha_{N+1}] \\ \sin[n2\pi f_1+\alpha_1] & \sin[n2\pi f_2+\alpha_2] & \cdots & \sin[n2\pi f_{N+1}+\alpha_{N+1}] \\ \vdots & \vdots & \cdots & \vdots \\ \sin[(N-1)2\pi f_1+\alpha_1] & \sin[(N-1)2\pi f_2+\alpha_2] & \cdots & \sin[(N-1)2\pi f_{N+1}+a_{N+1}] \end{bmatrix} \quad (2.29)$$

In the real impulse response case, the sine entries are not included and $N=n$. The column vector $u_1$ is the first $(N+1 \times 1)$ coordinate vector with a first element of one and the remaining zero.

The relation between the primal and dual problems is that the optimal (minimized) error $\|E\|$, in the primal problem, equals the optimal (maximized) dual variable, $d'$, in the dual problem. A relation exists among the optimal values of $d'$, $h'$ and the optimal set of extremal points $f'$, $\alpha'$ allowing the calculation of the optimal dual variable and the impulse response of the filter. When optimality is reached, the lower and upper bounds on the value of the primal problem meet and the following relation holds:

$$\beta^T \cdot y' = c^T \cdot r' \tag{2.30}$$

Also, it is proved in the next chapter that

$$\left[ d' \;\; h_0' \;\; \ldots \;\; h_{N-1}' \right] \cdot A' = c'^T \tag{2.31}$$

where $A'$ and $c'$ are the optimal values of the matrix $A$ and the vector $c$ respectively. This relation allows the calculation of the optimal coefficients from the optimal set $(f', \alpha')$.

The approximating functions must satisfy some constraints to qualify as basis functions for the complex approximation. These conditions are required to guarantee uniqueness of the approximation. The basis functions have to be linearly independent on the domain of approximation, $F$, and also satisfy the Haar condition [30], [31], [32]. If there is a domain of exactly $N$ complex points, and given their $N$ distinct locations, a set of $N$ basis functions satisfies the Haar condition if every function has a unique approximation on the domain, and the error at the points is zero. Existence and uniqueness of the best complex approximation is established in [32] and [29].

## 2.6 Comparison with Two Other Methods

We review in more detail the two most important methods capable of designing FIR filters in the complex domain. One of these methods was suggested by Chen and Parks [15], and was also discussed in Chapter 1 of Part II. Chen and Parks consider only approximation to conjugate symmetric responses. The constraint equation of the problem formulation by Chen and Parks is the same as Equation (2.17), repeated here for convenience. Using our notation, the constraint equation is

$$\sum_{k=0}^{N-1} h_k \, Re\left\{b_k(f)\, e^{-j\alpha}\right\} + \delta \geq Re\left\{D(f)\, e^{-j\alpha}\right\} \tag{2.32}$$

where $f \in F$ and $\alpha \in [0, 2\pi]$. The approximation problem in this form is a linear program with a finite number of variables and an infinite number of constraint equations, often called semi-infinite program. This problem is converted to a linear programming problem by discretizing both the angle and frequency domains. This is done as follows. The angle is sampled on $p$ points

$$\alpha_j = 2\pi t_j \qquad t_j \in T \qquad T = \{t_1, \ldots, t_p\} \tag{2.33}$$

and the frequency is sampled at $m$ points. The discretized version of the constraint equation is

$$\sum_{k=0}^{N-1} h_k \, Re\left\{ b_k(f_i) \, e^{-j\alpha_j} \right\} + \delta \geq Re\left\{ D(f_i) \, e^{-j\alpha_j} \right\} \tag{2.34}$$

for $i = 1, \ldots, m$, and $j = 1, \ldots, p$. This results in an overdetermined system of the form

$$\boldsymbol{A} \cdot \boldsymbol{h} = \boldsymbol{b} \tag{2.35}$$

where $\boldsymbol{h}$ contains the coefficients and it is of order $N$, and the matrix $A$ has dimension $(mp-1) \times (N-1)$. The complexity of the problem depends directly on how dense the discretized frequency and angle domains are. This method is more effective for small problems, because for large problems, good results require the solution of a large number of equations. This is not a limitation of our design method.

The Chebychev error $\|E\|_\infty$ is bounded by

$$\|E\|_\infty \leq \|E\|_{pm} \sec\left(\frac{\pi}{2p}\right) \tag{2.36}$$

where $\|E\|_{pm}$ is the maximum value of the error magnitude when the angle is sampled with $p$ points and the frequency is sampled with $m$ points in the Chen-Parks approach [15]. Thus, when $p=2$, the true error might be $\|E\|_{pm} \sec(\pi/4)$, or 1.414 times the minimized error, and when $p=8$, the true error is 1.02 times the minimized error.

Another significant approach is the one proposed by Preuss [19]. Preuss deals directly with the complex error. The complex error is

$$E(\Omega) = \left[ D(\Omega) - H(e^{j\Omega}) \right] W(\Omega) = |E(\Omega)| \, e^{j\alpha(\Omega)} \tag{2.37}$$

The magnitude error and the phase error do not have an equiripple behavior. Also, the number of extremal points $m$ of the magnitude error are at least $N+2$, and at most $2N+3$. The correct number is not known for each case. Therefore, the alternation theorem and the real Remez algorithm cannot be used. Since the magnitude of the optimal error is equiripple, the solution of the problem is described by $m$ equations

$$\left[D(\Omega_\nu) - H(e^{j\Omega_\nu})\right] W(\Omega_\nu) = \delta \, e^{\alpha_\nu} \qquad \nu=1,\ldots,m \tag{2.38}$$

where $\delta$, $\Omega_\nu$, and $\alpha_\nu$, $\nu=1,\ldots,m$ are not known. Also, the number of extremals, $m$, is not known.

The algorithm starts by constructing an initial solution assuming a set of $N+1$ points. The error is assumed to be zero at these points, and an initial filter is interpolated at these points using the Newton formula. The interpolation results in at least $N+2$ extremal points. At each iteration, only $N+1$ extremal points with the largest deviation values are chosen for the next iteration. The magnitude of the error deviations at these extremal points is adjusted using a control factor. In the interpolation of each iteration, the angles of the error function are kept from the previous iteration. The algorithm terminates when all the deviations are equal within a prescribed value.

The algorithm converges quickly when it will converge. But as noted by the author, the algorithm may diverge when the control factor used for the exchange is large [19]. Also, no theoretical proof of convergence is given, or any proof that the solution obtained is an optimal solution. Later, Schulist [21] improved the convergence and numerical instabilities of the algorithm by modifying the way the error magnitudes at the extremal points are calculated, and also by replacing the Newton interpolation formula by a Gaussian relaxation algorithm. Schulist [21] reports that the modifications show advantages if non-linear filters have to be designed, but the algorithm might still fail sometimes for filters with linear phase.

## 2.7 Conclusion

We presented the formulation of the design of FIR filters in the complex domain. The filters are designed using the minimax criterion. A filter designed by this method minimizes the Chebychev error between a complex-valued frequency response and the FIR filter. The filter design problem was put in the form of a semi-infinite optimization problem. This problem considers the minimization of an objective function subject to constraint equations.

In the original form, the objective function to be minimized is linear with respect to the variables but the constraint equations are not. The constraints can be made linear using a simple transformation from complex theory. This form results in a linear semi-infinite optimization problem. A solution can be obtained by discretizing this problem to obtain a linear program, which can be solved using the Simplex method. Our method of solving the problem is to first derive its dual, which transforms the infinite number of constraints to a finite number of constraints, and also allows the derivation of an efficient algorithm for its solution. This algorithm was developed by Tang [14], and it is described in the next chapter.

The formulation applies for the cases of real or complex filter impulse responses. When the frequency response to be approximated is conjugate symmetric, the approximating FIR filter will also have this property, and the coefficients of the best approximation will be real numbers. When the approximated frequency response is arbitrary, the coefficients of the best approximation must be complex numbers. This presents no problem since the complex coefficients can be separated into their real and imaginary parts, which are used to construct

a new set of approximating basis functions and coefficients of approximations. The approximation problem with complex coefficients can be treated in the same manner as the approximation problem with real coefficients. The development of the approximation problem using basis functions and coefficients is extremely efficient since it allows the treatment of problems similar to the FIR filter problem but with different basis functions and domain of approximation.

# 3. COMPLEX REMEZ ALGORITHM

The Remez algorithm solves the dual of the primal complex approximation problem. The chapter is organized as follows. After briefly describing the algorithm, we elaborate on the details of matters such as, the selection of an initial feasible extremal set of frequencies and angles, the optimality check and ending criterion of the algorithm, and the calculation of the optimal impulse response coefficients from the optimal approximation.

As previously discussed, real and complex coefficient filters correspond to approximations of conjugate and non-conjugate frequency responses, respectively. The formulation of the approximation problem was presented separately for real and complex coefficients, followed by discussion of a common form for the two cases. The presentation of the algorithm in this chapter applies to both cases. The differences for the two cases are, 1) the order of the approximation, and 2) the definition of the approximation functions and coefficients.

## 3.1 Complex Remez Algorithm

The complex Remez algorithm solves the dual problem by seeking the optimal set of extremal frequencies and angles maximizing the dual variable, and simultaneously minimizing

the maximum value of the complex error magnitude between a desired frequency response and an approximating FIR filter. The following theorem, adopted from [14] [28], suggests the algorithm.

**Theorem:** Suppose $H(\boldsymbol{h}^{/}, \bullet)$ is a best approximation in the Chebychev sense to a complex-valued desired frequency response $D(f)$ in $F$. Let $d^{/} \equiv \|E(\boldsymbol{h}^{/}, \bullet)\|$, where $\|E(\boldsymbol{h}^{/}, \bullet)\|$ is the optimal Chebychev error of the approximation, i.e., $\|E(\boldsymbol{h}^{/}, \bullet)\| = \max |E(\boldsymbol{h}^{/}, \bullet)|$ in $F$, and $\boldsymbol{h}^{/}$ is the optimal set of the approximating coefficients. Then $d^{/}$ is the optimal value of the following constrained maximization problem:

Maximize the inner product

$$d = \boldsymbol{c}^{T} \cdot \boldsymbol{r} \tag{3.1}$$

over all $f \in F^{N+1}$, $\alpha \in [0, 2\pi]^{N+1}$, and $\boldsymbol{r} \in [0, 1]^{N+1}$, subject to the constraint

$$\boldsymbol{A} \cdot \boldsymbol{r} = \boldsymbol{u}_1 \tag{3.2}$$

where $\boldsymbol{r}$ is a non-negative vector. The column vector $\boldsymbol{u}_1$ is the *(N+1 x 1)* coordinate vector, with the first element of one and the remaining zero. The vector $\boldsymbol{c}$ is given by

$$c^T = \left[ c_1 \; c_2 \; \ldots \; c_{N+1} \right] = \left[ Re\left\{ D(f_1) e^{-j\alpha_1} \right\} \; \ldots \; Re\left\{ D(f_{N+1}) e^{-j\alpha_{N+1}} \right\} \right] \tag{3.3}$$

The matrix $A$ was defined in the previous chapter. The theorem above suggests the following algorithm for the determination of the best FIR filter:

**STEP 1:** Choose an initial set of frequencies $f \in F^{N+1}$, and angles $\alpha \in [0, 2\pi]^{N+1}$, such that

$$r = A^{-1} \cdot u_1 \geq 0 \tag{3.4}$$

The initial set $(f, \alpha)$ is chosen in a way that the matrix $A$ is non-singular so its inverse exists. In our implementation this is the case when the initial set is chosen at random. The matrix $A$ calculated from a random set $(f, \alpha)$ is almost always non-singular and well-conditioned. Using the matrix $A$ calculate the corresponding inner product $d$, and a set of coefficients $h$.

**STEP 2:** Terminate the algorithm if the value of the scalar product $d = c^T \cdot r$ is close to the optimal. By this we mean that $d$ is within a certain tolerance factor to $\| E(h, \bullet) \|$, the Chebychev norm on the unit circle of the error between the desired frequency response and the approximating filter. The smaller the tolerance factor, the closer the approximation is to the true optimal approximation. We will elaborate on the tolerance later.

**STEP 3:** Update the extremal set $(f,\alpha)$, and the weight vector $r$, without violating any of the constraints, to increase the value of the scalar product $d = c^T \cdot r$. This step corresponds to the exchange portion of the algorithm. Return to **STEP 2** and repeat the process until the algorithm terminates.

Let us discuss the steps of the algorithm in detail. The initial set of trial extremal points and angles is chosen at random. If the inverse of the resulting matrix $A$ is ill-conditioned, a different starting set is chosen until a well-conditioned matrix $A$ is found. In our implementation, the first matrix $A$ is usually sufficiently well-conditioned, thus allowing the algorithm to start without any problems. Since the matrix $A$ is invertible, it will be used as the initial matrix provided that

$$r = A^{-1} \cdot u_1 \geq 0 \tag{3.5}$$

If any of the elements of the initial vector $r$ are negative, an initial positive vector $r$ can be derived by constructing a vector of angles based on the following criterion [28]:

If $r_k < 0$ then $\alpha_k \rightarrow a_k + \pi$.

If $r_k \geq 0$ then $\alpha_k \rightarrow \alpha_k$.

The reason that the vector $r$ can be derived in this manner is the following. Let us express the matrix $A$ in the form $A = [a_1 \ . \ . \ . \ a_{N+1}]$, where the kth column of the matrix $A$ is given by

$$a_k = \left[1 \;\; Re\left\{b_0(f_k)\, e^{-j\alpha_k}\right\} \ldots Re\left\{b_{N-1}(f_k)\, e^{-j\alpha_k}\right\}\right]^T \tag{3.6}$$

Then, Equation (3.2) can be written in the form

$$\sum_{m=1}^{N+1} a_m\, r_m = u_1 \tag{3.7}$$

Suppose that $r_k < 0$. Then (3.7) can be written in the form

$$a_k\, r_k = u_1 - \sum_{\substack{m=1 \\ m \neq k}}^{N+1} a_m\, r_m \tag{3.8}$$

By changing $\alpha_k \rightarrow \alpha_k + \pi$, all the terms of the vector $a_k$ change sign and $a_k' = -a_k$, where $a_k'$ is the new vector. Then

$$a_k'\, r_k' = u_1 - \sum_{\substack{m=1 \\ m \neq k}}^{N+1} a_m\, r_m = a_k\, r_k \tag{3.9}$$

and therefore $r_k^{\prime} = -r_k$. Although there is no proof that the adjusted matrix $A$ will be invertible, we have never encountered problems with this adjustment in practice. This method of finding an initial $A$ is rather natural and was also used in the code published in [28].

In **STEP 2**, the value of $d$ is checked for optimality. For a given feasible set $(f, \alpha)$, the dual variable $d$, and the approximating coefficients $h$, can be calculated from the following relation:

$$[d\ ,\ h]^T \cdot A = c^T \tag{3.10}$$

*Proof:*

We can write the matrix $A$ as

$$A = \begin{bmatrix} a \\ B \end{bmatrix} \tag{3.11}$$

where the row vector $a$ has all its elements equal to 1. From the constraint equation $A \cdot r = u_1$, we can get the following identities:

$$a^T \cdot r = 1 \qquad B \cdot r = 0 \qquad h^T \cdot B \cdot r = 0 \tag{3.12}$$

The dual variable $d$ can be expressed as

$$d = d \cdot 1 + 0 = d \cdot a^T \cdot r + h^T \cdot B \cdot r = (d \cdot a^T + h^T \cdot B) \cdot r \tag{3.13}$$

and since $d = c^T \cdot r$,

$$d \cdot a^T + h^T \cdot B = c^T \qquad or \qquad [d\ ,\ h^T] \cdot A = c^T \tag{3.14}$$

For a given feasible set $(f, \alpha)$, it can be shown that the following relation holds [14]:

$$d \leq \|E(h^{/}\ ,\ \bullet)\| \leq \|E(h\ ,\bullet)\| \tag{3.15}$$

This relation suggests the stopping criterion of the algorithm. The value of $d$ is calculated at each iteration, and compared to the maximum value of the error $\|E(h,\bullet)\|$. If

$$\|E(h\ ,\ \bullet)\| - d \leq tolerance \cdot d \tag{3.16}$$

for some "tolerance" chosen beforehand, the algorithm terminates. A termination with tolerance chosen as 0.01 means that the approximation obtained at this point is within 1% of the optimal solution. A smaller tolerance can be specified if a closer approximation is desired. When the stopping criterion is satisfied, the optimal values $d'$ and $\boldsymbol{h}'$ are calculated from Equation (3.10).

**STEP 3** corresponds to the exchange portion of the algorithm. This step is repeated until the optimal solution is reached. The one point exchange is performed on the points and angles (extremal signature) of the current iteration. First, the error is calculated on the domain of approximation, and we find the frequency where the maximum value of the error magnitude occurs. This extremal frequency $f_{new}$, and the angle of the error function at that frequency are used to replace one element of the old extremal set. This process does not necessarily require discretization of the frequency domain $F$ because the derivative of the error magnitude can be expressed in closed form. A zero-finder can be used to find the zeros of the first derivative. At these points the error attains its maximum values.

To find the maximum of the error magnitude, we take its derivative. It is easier to take the derivative of the square magnitude [33]. The parametrization of the basis functions, and the desired response on the unit circle is given by $\gamma(f) = 2\pi f$. The complex error is

$$E(\boldsymbol{h}, \gamma(f)) = D(\gamma(f)) - \sum_{k=0}^{N-1} h_k \, b_k(\gamma(f)) \tag{3.17}$$

and the derivative of its squared magnitude is given by

$$\frac{d\,|E[\boldsymbol{h},\gamma(f)]\,|^2}{df} = \frac{d}{df}\left\{E[\boldsymbol{h},\gamma(f)]\;\; E^*[\boldsymbol{h},\gamma(f)]\right\} \tag{3.18}$$

The derivatives of the complex error and its conjugate are calculated using the chain rule. For the FIR filter $H(\boldsymbol{h},\gamma(f))$,

$$\frac{d}{df} H(\boldsymbol{h}\,,\,\gamma(f)) = \sum_{k=0}^{N-1} h_k \frac{d}{df} b_k(\gamma(f)) \tag{3.19}$$

$$\frac{d}{df} b_k(\gamma(f)) = \frac{d}{dz} b_k(z) \frac{d}{df} \gamma(f) \tag{3.20}$$

When the derivative of the error magnitude is found, a zero-finder routine can be used to find the zeros of the derivative and determine the maximum point.

After $f_{new}$ is determined, we determine $\alpha_{new}$, which is the angle of the complex error at the frequency $f_{new}$. Hence, there is no need to discretize the phase domain $[0,2\pi]$. When the frequency domain is discretized the phase domain still need not be discretized.

The pair $(f_{new}, \alpha_{new})$ is used to replace an appropriate pair of the old extremal set. The new extremal set should increase the dual variable $d$. Define a vector $\boldsymbol{v}$ of order N+1, as

$$v = \left[1 \;\; Re\left\{b_0(f_{new})\, e^{-j\alpha_{new}}\right\} \;\; \ldots \;\; Re\left\{b_{N-1}(f_{new})\, e^{-j\alpha_{new}}\right\}\right]^T \tag{3.21}$$

This is the same as a column of the matrix $A$ at the point $(f_{new}, \alpha_{new})$. We then define the *(N+1 x 1)* vector $p$ as

$$p \equiv A^{-1} \cdot v \tag{3.22}$$

**Lemma 1.** [28] At least one of the N+1 elements of the vector $p$ is strictly positive.

Since some of the elements of the vector $p = [p_1 \ldots p_{N+1}]^T$ are positive, one can find an index $m$, where $1 \le m \le N+1$, such that [28]

(i) $p_m > 0$ and,

(ii) $\dfrac{r_m}{p_m} \le \dfrac{r_j}{p_j}$ when $p_j > 0$.

The new pair $(f_{new}, \alpha_{new})$ is used to replace the frequency $f_m$ and angle $\alpha_m$, and thus

$$f_{new} = [f_1 \ldots f_{new} \ldots f_{N+1}] \qquad \alpha_{new} = [\alpha_1 \ldots \alpha_{new} \ldots \alpha_{N+1}] \tag{3.23}$$

This method has been used by Tang [28], and it is also the standard method for the Simplex algorithm. The matrix $A$ is evaluated using the new set of frequencies and angles to obtain $A_{new}$.

**Lemma 2.** [28] The matrix $A_{new}$ is non-singular.

*Proof:* The matrix $A_{new}$ and the old matrix $A$ are identical except for one column involving the new frequency and angle $(f_{new}, \alpha_{new})$. From the definition of the vector $p$ we get

$$A^{-1} \cdot v = A^{-1} \cdot (mth\ column\ of\ A_{new}) = p \tag{3.24}$$

Therefore,

$$A^{-1} \cdot A_{new} = [u_1 \ldots p \ldots u_{N+1}] = U \tag{3.25}$$

where for a vector $u_k$, only the kth element is non-zero and it is equal to 1, i.e. it is the kth coordinate vector. It is proved in [28] that $U$ is not singular and its inverse is

$$U^{-1} = [u_1 \ldots u_{m-1} \quad p' \quad u_{m+1} \ldots u_{N+1}] \tag{3.26}$$

where

$$p' = \frac{[-p_1 \cdots -p_{m-1} \quad 1 \quad -p_{m+1} \cdots -p_{N+1}]}{p_m} \qquad (3.27)$$

Therefore the inverse of $A_{new}$ is given by

$$A_{new}^{-1} = U^{-1} \cdot A^{-1} \qquad (3.28)$$

**Lemma 3.** [28] The new (N+1 x 1) vector $r_{new}$ is non-negative.

Note that $r_{new}$ is given by

$$r_{new} = A_{new}^{-1} \cdot u_1 = U^{-1} \cdot A^{-1} \cdot u_1 = U^{-1} \cdot r \qquad (3.29)$$

The previous lemmas show that the use of the new extremal set $(f_{new}, \alpha_{new})$ does not make the matrix $A$ singular and also the vector $r$ remains positive.

Now we need to show that the new extremal set is an improved set in that it forces the dual variable $d$ to increase. The following lemma suggests this.

**Lemma 4.** [28] Let $r_m$ denote the mth entry of the new primal solution $r_{new}$. Then the following relation holds:

$$d_{new} = (1 - r_m)\, d + r_m \, \|E(h, \bullet)\| \tag{3.30}$$

We know from Equation (3.15) that $d \leq \|E(h, \bullet)\|$. Combining this with the equation above it is seen that $d_{new} \geq d$.

Another way of looking at the exchange is to solve a linear program. Let us denote the new resulting matrix and vectors by $A_{new}$, $r_{new}$, $c_{new}$. Then the following relations hold:

$$r_{new} = A_{new}^{-1} \cdot \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \tag{3.31}$$

and

$$c_{new}^{T} \cdot r_{new} \geq d \tag{3.32}$$

The point of the old extremal set to be replaced is found by solving the following linear programming problem using the simplex method [14]: Find a nonnegative vector $r' \in \mathbb{R}^{N+2}$ to maximize

$$\left[ c^T \, , \, Re\left\{ D(f_{new})\, e^{-j\alpha_{new}} \right\} \right] \cdot r' \tag{3.33}$$

subject to the constraint

$$[A \mid v] \cdot \begin{bmatrix} r' \\ s \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.34}$$

with the vector $v$ was as defined in (3.21). The optimal basis for this problem, and the corresponding portion of $r'$, give $A_{new}$, and $r_{new}$. When the pivotal point is found and the exchange is performed, the algorithm returns to **STEP 2** and repeats.

Another issue to be discussed is the convergence of the algorithm. Tang [28] gives an extensive proof of quadratic convergence of the algorithm when certain assumptions are satisfied. The main question to be answered when convergence is discussed is if the quantity

$$\eta_k \equiv \| D - H(h^{(k)}, \bullet) \| - d^{(k)} \tag{3.35}$$

converges to 0 for a sufficiently large k. The function $H(h^{(k)}, \bullet)$ is the approximating filter at the *kth* iteration, $h^{(k)}$ is the set of approximating coefficients at the *kth iteration*, and $d^{(k)}$ is the value of the dual variable at the *kth* iteration. Tang [34], shows that if the vector $r^{(k)} > 0$ for all k = 1, 2, . . ., then

$$\lim_{k \to \infty} \inf \eta_k = 0 \tag{3.36}$$

This means that the algorithm terminates in a finite number of iterations for a positive stopping criterion. He also proves that convergence is quadratic. Quadratic convergence is described by the following theorem [28].

**Theorem:** Let the magnitude of the error function $|E(f)| = |D(f) - H(h',f)|$ have exactly N+1 extrema $f_j'$ $1 \le j \le N+1$, and angles of the complex error $\alpha_j'$, where $\alpha_j' = Arg\,[D(f_j') - H(h',f_j')]$ $1 \le j \le N+1$, and let $r' \ge 0$. Furthermore, let the second derivatives with respect to $f$ of $|D(f) - H(h',f)|$ be nonzero at each of the N+1 extrema points. Then there is exists a sequence $\{\delta_k\}$, where $\delta_k \ge \eta_k$ for all $k$, and two constants $M$, and $K$, such that

$$\delta_{N+1+k} \le M\, \delta_k^2 \qquad for \quad k \ge K \tag{3.37}$$

The complete proof of these statements can be found in [28] which holds an extensive portion of Tang's dissertation. In our applications we had no difficulty with convergence, and in most cases, convergence within 1 % of the optimal solution was obtained in a number of iterations that was four to five times the order of the filter, meeting expectations for problems of this kind. In other cases, a larger number of iterations is needed when stringent requirements are specified e.g. small transition bands.

The complexity of **STEP 1** is $O(N^3)$. This includes the first LU factorization and solution of the initial basis matrix to get the primal and dual vectors. Each iteration involves two main computational processes: (i) location of the maximum magnitude of the error function, and (2) update of the basis and primal and dual solutions. For (i), if the function evaluation at each frequency point is done in a straightforward manner the complexity is *O(N)*. If a discretized frequency domain is used, and the maximum magnitude is determined by evaluation of the magnitude at every single frequency point in a domain of $M$ points, say, $M = 4N$, then the complexity is $O(4N^2)$. Thus, for one sweep of $N$ iterations, the complexity is $O(4N^3)$. For (ii), if the updating is done by solving for the primal and dual variables with the new basis matrix, the complexity is $O(N^3)$. However, a simplification is possible by exploiting the fact that the new basis matrix differs from the previous one by only one column. In Tang [28], the implementation uses an $O(N^2)$ updating scheme which was based on well-known rank-one updating methods in computational linear algebra. Thus, for one sweep, the complexity is $O(N^3)$. To be numerically safe, in practice, we recompute the LU factorization of the basis matrix after every sweep, adding another $\frac{1}{3}N^3$ operations at every sweep. Summarizing then, every sweep requires less than $(5N^3)$.

## 3.2 Conclusion

We discussed the complex Remez algorithm used to solve the dual of the primal approximation problem. The solution of the dual problem is easier to obtain since the infinite constraints of the primal problem become finite in the dual. The optimal Chebychev solution is characterized by a set of frequencies and angles. The algorithm then tries to locate this extremal set which will minimize the Chebychev error.

The algorithm is given an initial random set of frequencies and angles to generate the necessary matrices and vectors. If the initial approximation matrix is ill-conditioned, a different random initial set is chosen. The complex error between the desired response and the approximating FIR filter is then formed using the initial set of points and angles, and is checked for optimality. The optimality criterion requires that the dual variable be equal to the maximum value of the error magnitude. In practice a tolerance is given between the maximum error and the dual variable to avoid unnecessary iterations. We saw that to find the maximum points of the error magnitude, no discretization of the frequency and angle intervals is required, since a closed form of the derivative of the error magnitude can be obtained. Then a zero-finder can be applied to locate the maxima and minima. Even when a discrete frequency domain is chosen, there is still no need to discretize the angle interval, since the angle used for the exchange is the angle of the complex error where it attains its maximum.

If the current iteration does not result in the optimal solution, a single exchange on a frequency and angle of the old extremal set is performed. The point to be exchanged is found by solving a linear program. The new frequency for the extremal set is the frequency with the maximum magnitude of the error, and the new angle is the angle of the complex error at this frequency. The exchange has the effect of increasing the value of the dual variable. Eventually, the dual variable approaches its upper bound, which is the Chebychev error of the approximation. One of the most important properties of the algorithm is that the approximation matrix remains non-singular throughout the duration of the algorithm.

# 4. IMPLEMENTATION AND RESULTS

## 4.1 Implementation

The design of FIR filters in the complex domain was implemented with programs written in FORTRAN. We implemented the algorithm with both continuous and discrete frequency domains. In the discrete case, we define a discrete frequency domain with grid density of approximately 10N, where N is the length of the filter. Note though that the angle interval $[0, 2\pi]$ is not discretized. The maximum value of the complex error magnitude is found by a search on the defined grid. This version of the program is useful in cases where specific values of the desired complex frequency response need to be specified at certain frequency points. Also, the extremal points of the magnitude error are easier to locate when a discrete frequency domain is used.

In the continuous frequency case, a continuous frequency domain $F$ is assumed, and the complex-valued frequency response is specified on $F$. The specification of the desired frequency response assumes transition bands between passbands and stopbands. The maximum of the error magnitude needed for the single exchange can be found by differentiating the squared magnitude, as explained in the previous chapter. Then a root finder routine is used to locate the zeros of the first derivative. We have used a routine developed by Brent [35] to find the maximum points. The method uses a combination of a

golden section search and successive parabolic interpolation. We divide the frequency domain $F$ into N+1 intervals and find the maximum in every interval. Then we choose the largest value from those local maximum values. The method of dividing the domain into intervals was utilized in [28].

Several of the routines in the design program were adapted from the code published in [28]. The matrix operations are performed using LINPACK [36]. The LINPACK package is an advanced set of FORTRAN subroutines for basic and advanced matrix operations. We use the four basic routines from LINPACK called DGECO, DGEFA, DGESL, and SGEDI. These routines use double precision arithmetic and operate on general square matrices. They are used in the complex filter design program to compute the inverse of the matrix $A$ by LU decomposition, calculate its condition number, and compute other vector and scalar quantities involving the matrix $A$.

### 4.1.1 Program Input and Output

The input to the program involves the following information about the desired frequency response and the approximating FIR filter:

1) *Type* of filter. The different choices are: a) Linear-phase filter, b) Conjugate-symmetric filter (nearly linear-phase filter), c) Non-conjugate symmetric FIR filter (complex impulse response), d) Hilbert transformer, and e) Differentiator.

2) The frequency *band edges* in the normalized frequency interval [-0.5,0.5]. These specify the passbands and stopbands of the filter and consequently the transition bands of the filter. If a conjugate symmetric response is approximated (real impulse response), only the interval [0.,0.5] needs to be specified. Specifying band edges for the approximation bands gives good control over the cutoff frequencies of the filter.

3) The *magnitude* values of the desired response in the passbands and stopbands. If a frequency selective filter is designed, the magnitude response is that of an ideal selective filter, which is one in the passbands and zero in the stopbands. Other responses besides those of the ideal selective responses can be specified.

4) The *weight* on the error function in each band. A larger weight in a band makes the magnitude error smaller compared to the other bands. On the other hand, the error is larger in the other bands when compared to using equal weights. Typical values of the weight are between 1 and 10. Usually error weighting is used in the stopbands to achieve better attenuation by allowing a larger ripple in the passbands.

5) The desired *group delay* of the filter. Usually a constant group delay is specified. We will see shortly that specifying a constant group delay gives an almost equiripple delay response in the passband. For a linear-phase filter, a group delay of half the filter length is specified. It will be shown by examples that in most cases, if a group delay other than half the filter length is specified, a better magnitude response results.

6) Percentage *tolerance* within the best approximation before the program stops. Usually 1% is a good stopping criterion. When a closer solution is desired, a smaller

percentage can be specified at the expense of more iterations. All examples discussed in this chapter use 1% tolerance.

## 4.2 Design Examples

In this section we discuss several filter design examples. We present a variety of examples to show the versatility of the algorithm and the design program. The most common designs are filters with real coefficients. We present frequency selective filters such as lowpass and bandpass filters. Usually a specified group delay which is less than half the length of the filter results in better magnitude characteristics compared to linear-phase filters with the same length.

A linear-phase filter design is also discussed. The filter can be designed by the complex algorithm when the basis functions are chosen appropriately. Only the first half of the coefficients are involved in the design. The remaining half of the coefficients can be obtained from the first half using the symmetry of the impulse response. The coefficients are compared to the impulse response coefficients obtained by the Parks-McClellan program [37], and as expected, they are in very close agreement. Note that the group delay in the linear-phase filter designs is specified to be half the length of the filter.

Hilbert transformers can also be designed by the complex algorithm. We show the design of narrow-band and wide-band transformers. In the case of wide-band transformers,

the desired frequency response is discontinuous at the domain endpoints, and a large error can occur in the design. When a non-integer group delay is specified, it is possible to get designs with small approximation error. In this section we also examine a one-sided Hilbert transformer. Since the frequency response is not conjugate symmetric, complex impulse response coefficients are required.

All filter designs approximating non-conjugate symmetric frequency responses require complex coefficients. We show a single-sideband complex filter approximating a response which is zero for negative frequencies. These filters are very important in the processing of analytical signals. Finally, we present the design of differentiators by two examples. One example deals with the design of narrow-band differentiators for which one or more stopbands are specified. Another example presents the design of a full-band differentiator. For the full-band differentiator designs, a non-integer group delay needs to be specified to make the desired response continuous at the endpoints of the frequency domain.

These examples were designed using the discretized version of the program. The discrete frequency domain used in these examples is between 10N to 15N points, where N is the order of the approximation. In all examples we give the CPU time of an IBM-compatible 486-66 MHz personal computer required by the algorithm to converge within 1% of the optimal solution.

### 4.2.1 Design of Filters with Real Coefficients

**Example 4.1:** *Lowpass Filter*

This design example is a lowpass filter with length N=35. The passband is defined on [0.,.13] and the stopband on [0.2,0.5]. Note that a transition band is implied by the specification of the passband and stopband. The desired magnitude value is 1 in the passband and 0 in the stopband. The desired group delay is specified to be 15 samples, since it was found by trial and error that it results in the least Chebychev error of the magnitude response. A weight ratio of 1:10 was given to make the relative magnitude error smaller in the stopband. For this example, the algorithm required 165 iterations and 2 minutes 42 seconds CPU time to converge within 1% of the best solution. The dual variable versus the number of iterations is plotted in Figure 4.1. The magnitude in dB is shown in Figure 4.2. This nearly-linear-phase filter has a passband ripple $\delta_p$=0.0145 (0.125 dB) and a stopband attenuation $\delta_s$=0.00145 (56.77 dB). Almost the same results are obtained when the group delay is specified to be 16 samples. The linear-phase design, as well as designs with group delay less than 15, give slightly poorer magnitude response. Figure 4.3 shows the group delay of the filter. The plot shows an equiripple behavior of the group delay in the passband with maximum deviation of 0.26 samples from the specified constant of 15 samples. Figure 4.4 shows the zeros of the lowpass filter.

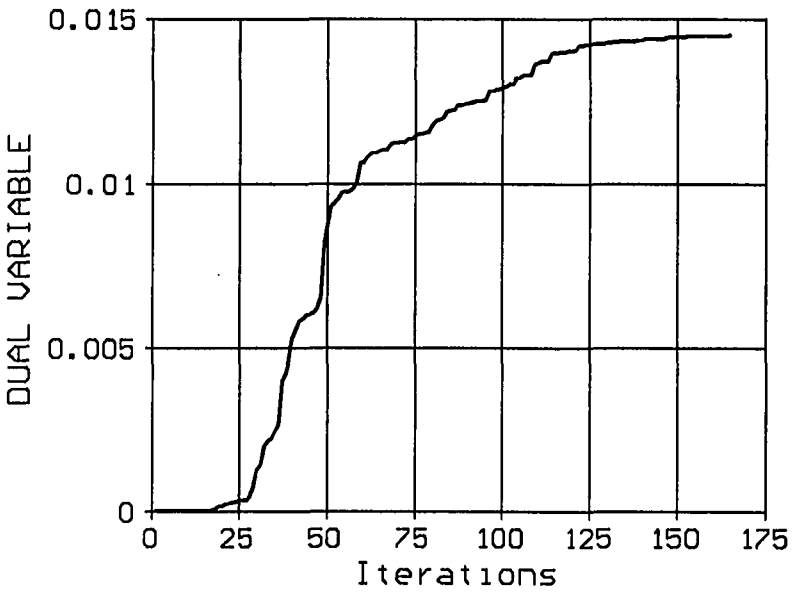


Figure 4.1: Dual variable plot for example 4.1

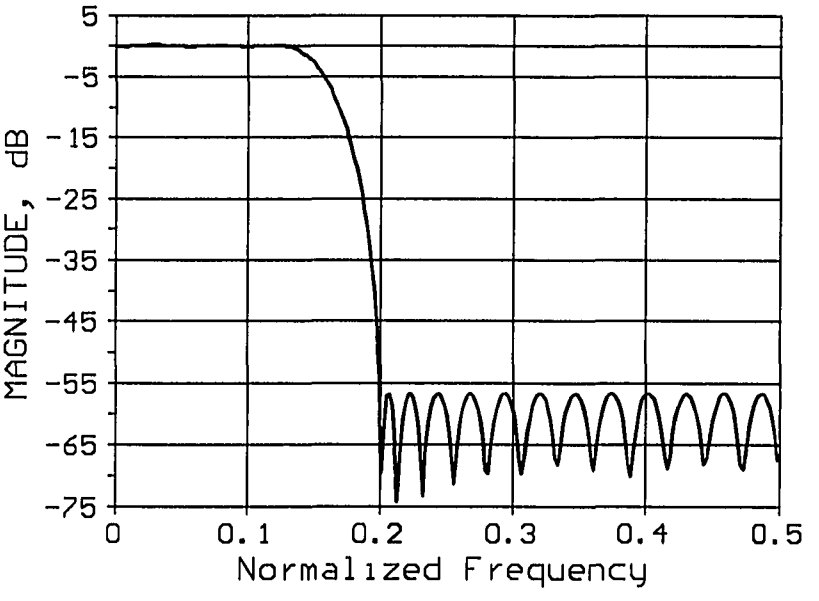


Figure 4.2: Magnitude response of the lowpass filter in example 4.1

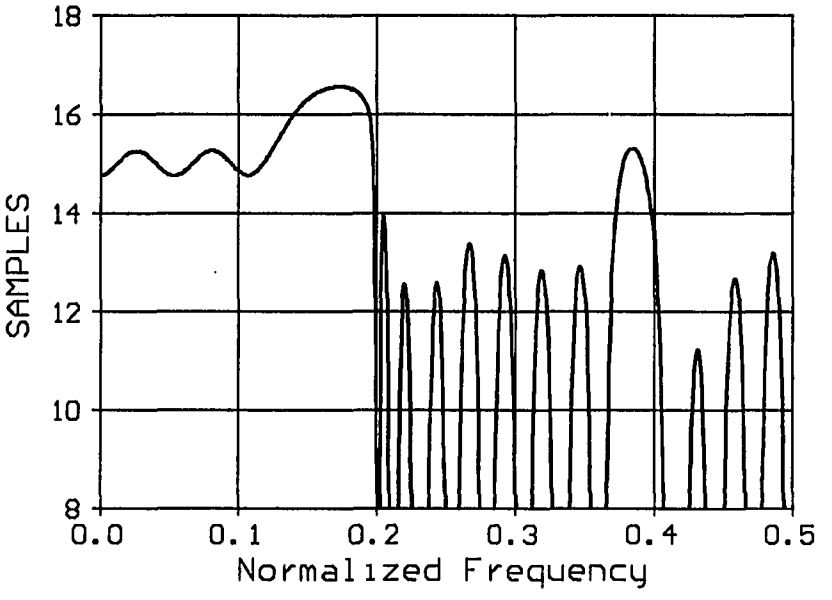


Figure 4.3: Group delay of the lowpass filter in example 4.1. The group delay is almost equiripple in the passband

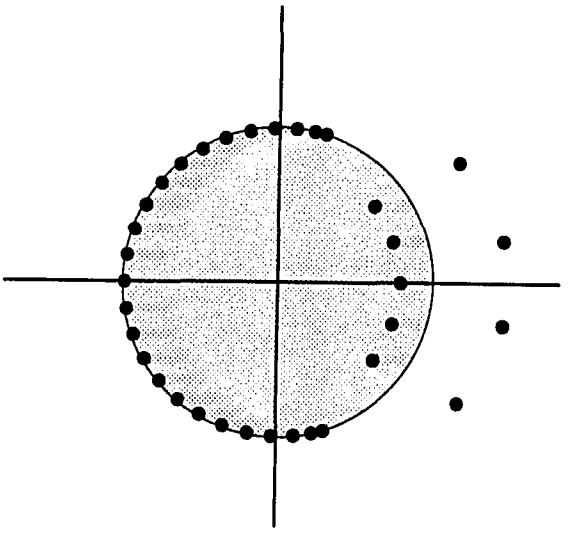

Figure 4.4: Zeros of the lowpass filter in example 4.1

**Example 4.2:** *Lowpass Filter*

This example compares the results of our design method with those of Example 1 in Chen and Parks [15]. We use their design specifications. The design is a lowpass filter with normalized cutoff frequencies $f_1$=*0.06* and $f_2$=*0.12* and a group delay $\tau_d = 12$ samples. The length of the filter is 31 and the weight used is 1:10. The algorithm converged in 196 iterations in 2 minutes and 24 seconds. The dual variable is shown in Figure 4.5. The resulting filter has a passband ripple $\delta_p = 0.0439$ (0.373 dB) and a stopband attenuation $\delta_s = 0.00439$ (47.15 dB). The filter magnitude, group delay, and zero plot are shown in Figures 4.6 through 4.8 respectively. In [15], the authors report $\delta_p = 0.0436$ (0.37 dB) and $\delta_s = 0.00436$ (47.21 dB). The results of the two methods are in close agreement. However, we note that, because of the discretization used in [15], the actual deviation could be as large as 0.00436 times a factor of 1.02. Thus, the 47.21 dB stopband attenuation can be, in reality, 47.04 dB. The group maximum group delay deviation in the passband from 12 samples is about 0.93 samples.

A Parks-McClellan linear-phase filter with the same specifications, results in $\delta_p = 0.0575$ (0.49 dB) and $\delta_s = 0.00575$ (44.81 dB). The filter has a slightly larger passband ripple and somewhat less stopband attenuation. Note that the group delay is assumed to be 15 samples when the Parks-McClellan program is used. When a delay of $\tau_d = 15$ samples is specified to our algorithm, a linear-phase filter results.

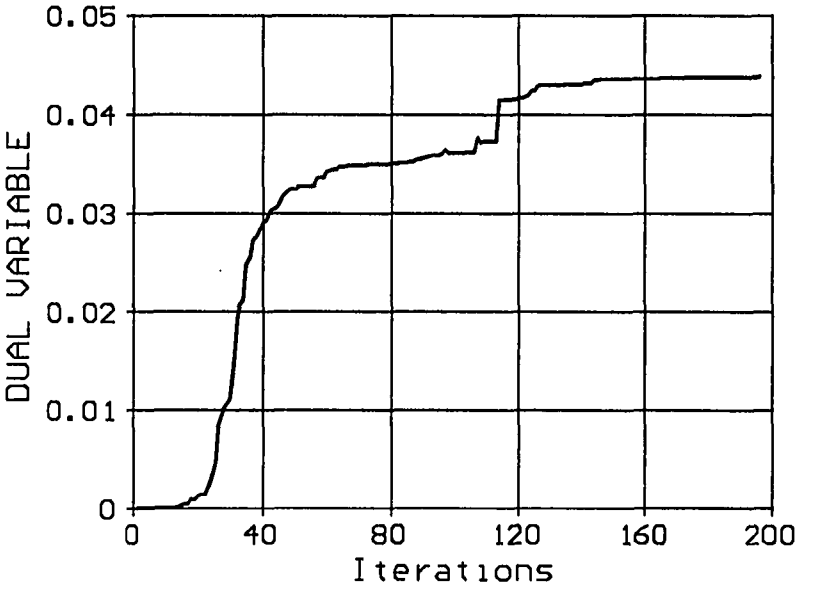


Figure 4.5: Dual variable plot for example 4.2

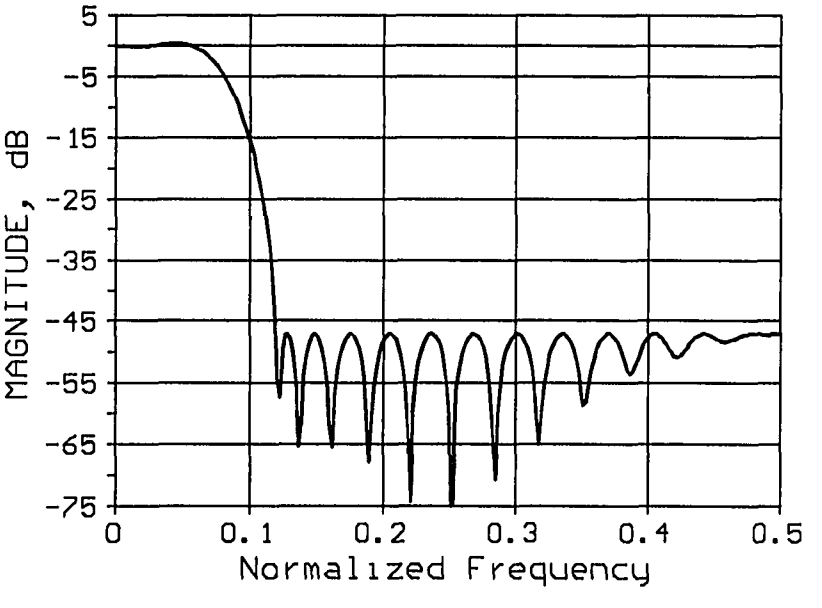


Figure 4.6: Magnitude in dB for the LPF in example 4.2. This example is taken from [15]

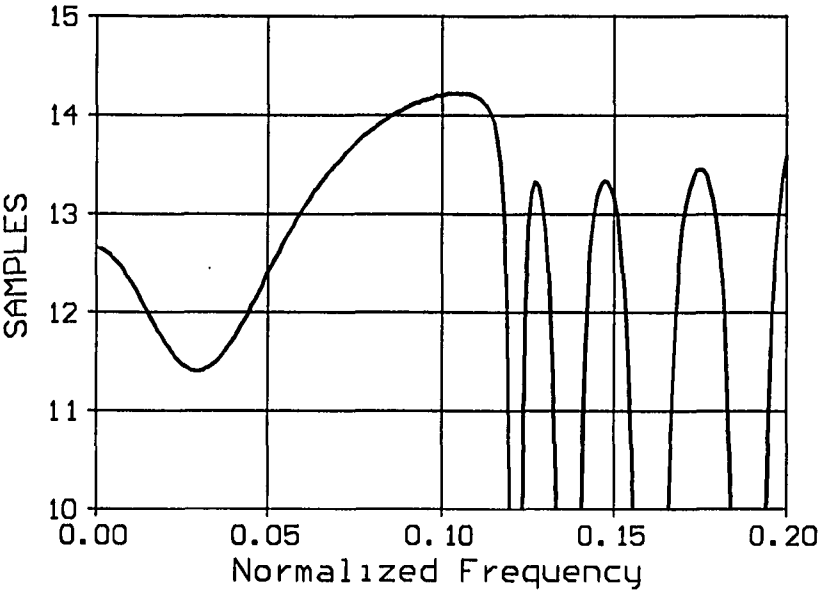


Figure 4.7: Group delay of the filter in example 4.2. The maximum deviation of the group delay in the passband from the constant 12 samples is about 0.93 samples

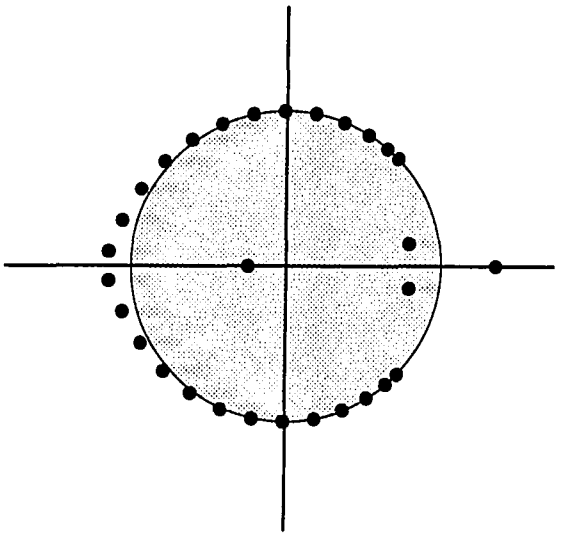

Figure 4.8: Zeros for the lowpass filter in example 4.2

**Example 4.3:** *Lowpass Filter*

This example is presented to show the ability of the algorithm to converge for rather long filters and narrow transition bands. The design example is a lowpass filter of length N=80, with passband [0.,0.1], and stopband [0.14,0.5]. The desired group delay is specified to be 30 samples and a weight ratio of 1:10 is used. The plot of the dual variable is shown in Figure 4.9. The magnitude response in dB is shown in Figure 4.10. The passband ripple is $\delta_p$=0.00449 (0.039 dB), and the stopband attenuation is $\delta_s$=0.000449 (66.96 dB). The group delay is shown in Figure 4.11. The maximum deviation from 30 samples is about 1 sample, and occurs at the edge of the passband.

### 4.2.2 Design of Linear-Phase Filters

**Example 4.4:** *Linear-Phase Filter*

The algorithm is capable of producing optimal Chebychev linear-phase filter designs equivalent to those produced by the Parks-McClellan algorithm. This example is a bandpass filter of length N=33, and a specified group delay of $\tau_d$=16 samples. The passband is defined on [0.2,0.35] and the stopbands are defined on [0.0,0.1] and [0.425,0.5]. A weight 10:1:10 is used. We define the approximating filter of odd order as follows:

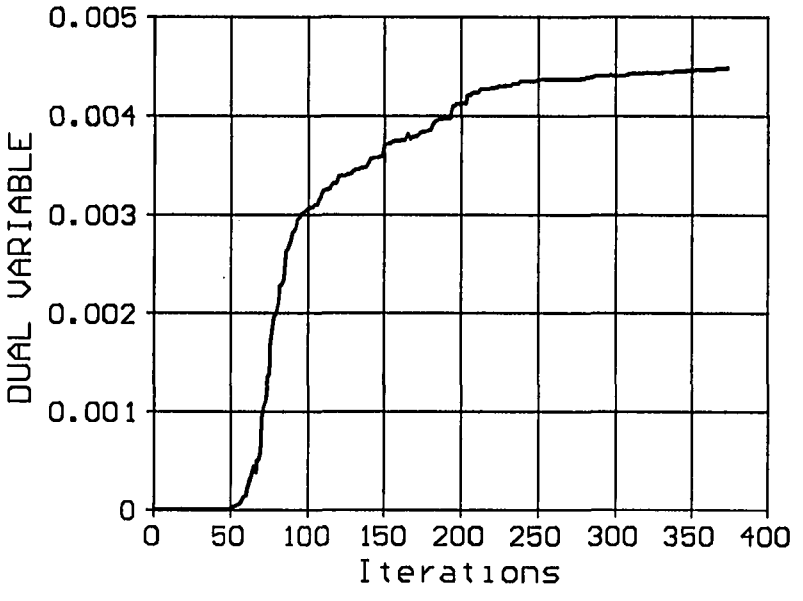


Figure 4.9: Dual variable for example 4.3

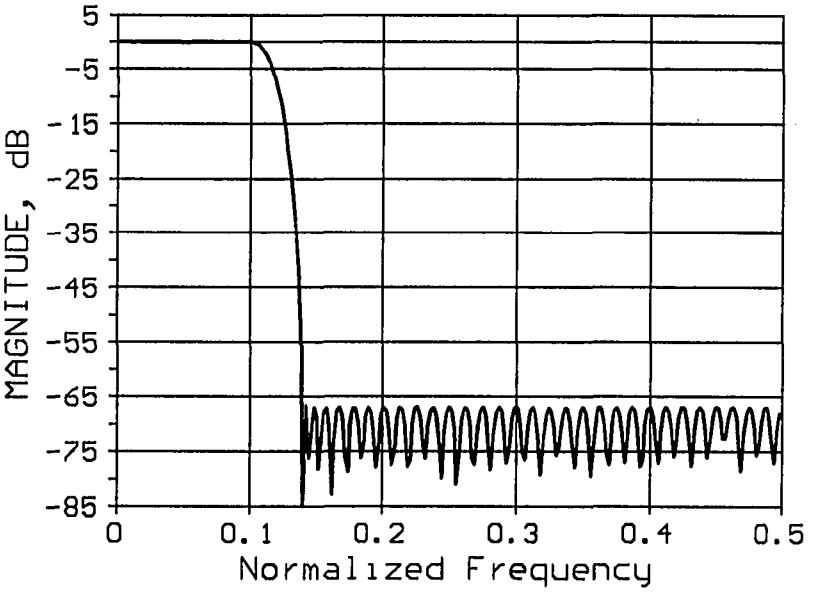


Figure 4.10: Magnitude in dB for the LPF of length 80 in example 4.3

$$H(f) \sum_{k=0}^{\frac{N-1}{2}} h_k \, b_k(f) \tag{4.1}$$

and

$$b_k(f) = e^{-j2\pi fk} + e^{-j2\pi f(N-1-k)} \tag{4.2}$$

The algorithm converged in 17 seconds and required 33 iterations. The dual variable plot is shown in Figure 4.12. The resulting filter has passband ripple $\delta_p = 0.016$ (.138 dB), and stopband attenuation $\delta_s = 0.0016$ (55.91 dB). The magnitude response in dB, and zero plot are shown in Figures 4.13 and 4.14 respectively. Table 4.1 shows a comparison of the impulse responses for the design obtained by the complex algorithm (a), and the Parks-McClellan program (b). Only half of the coefficients are shown. The remaining coefficients can be obtained using the symmetry of the impulse response.

### 4.2.3 Design of Filters with Complex Coefficients

**Example 4.5:** *Filter with Complex Coefficients*

In the processing of analytical signals, filters with non-symmetric frequency responses are frequently required. The algorithm is capable of determining an FIR filter, with complex coefficients, that approximates a non-conjugate symmetric frequency response. These filters

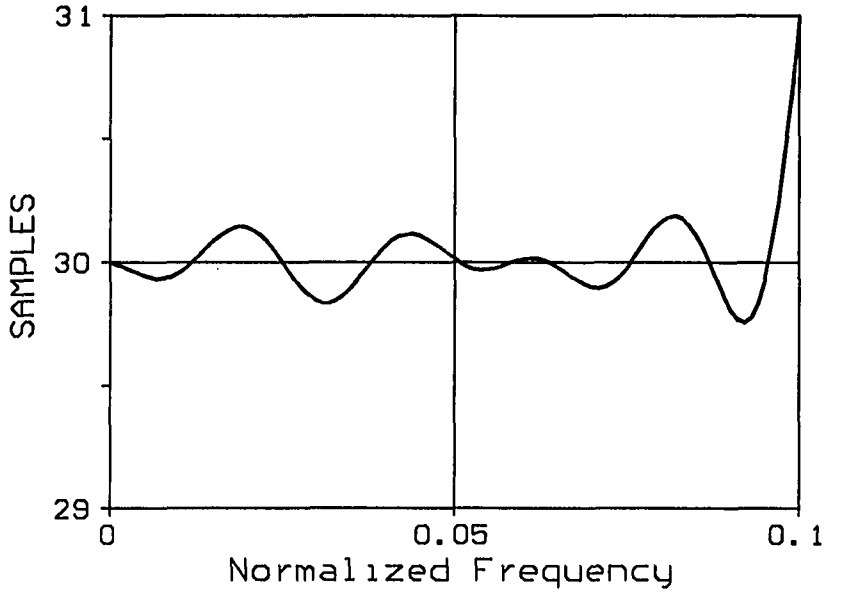


Figure 4.11: Passband group delay of the LPF of length 80 in example 4.3

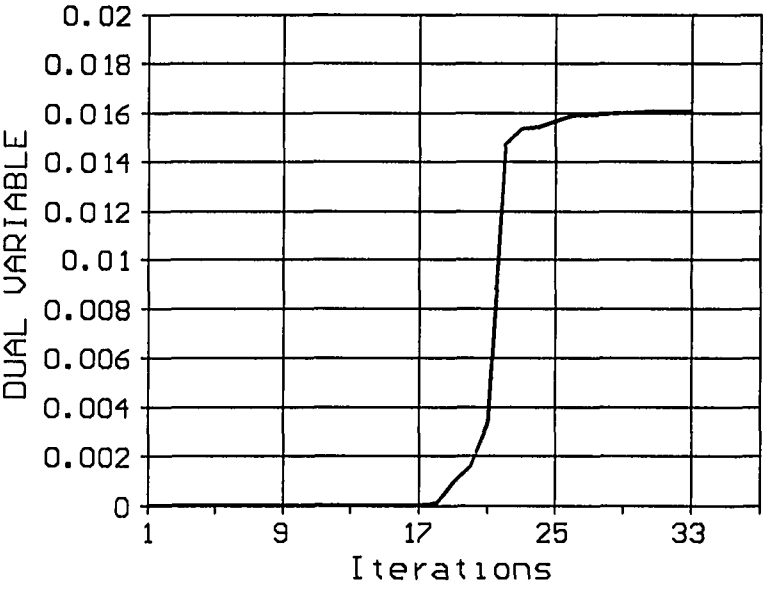


Figure 4.12: Dual variable plot for example 4.4

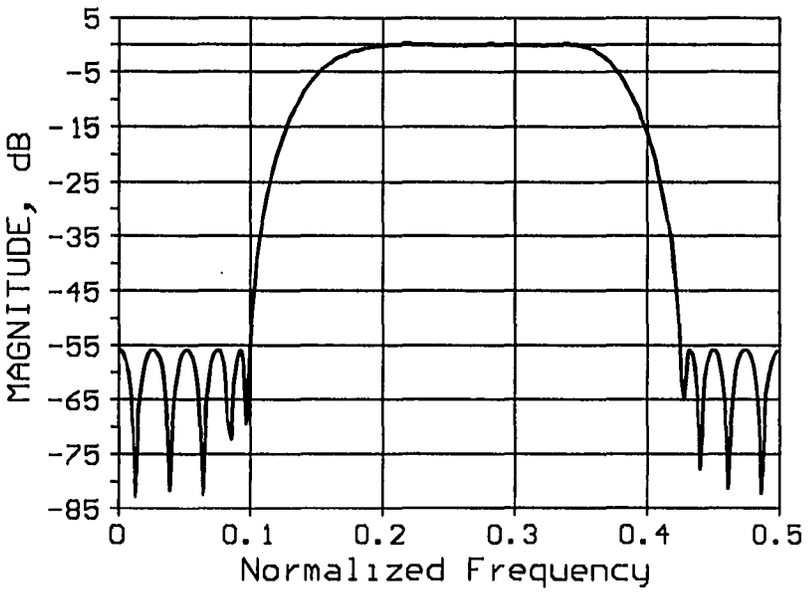


Figure 4.13: Magnitude response in dB of the linear-phase filter in example 4.4

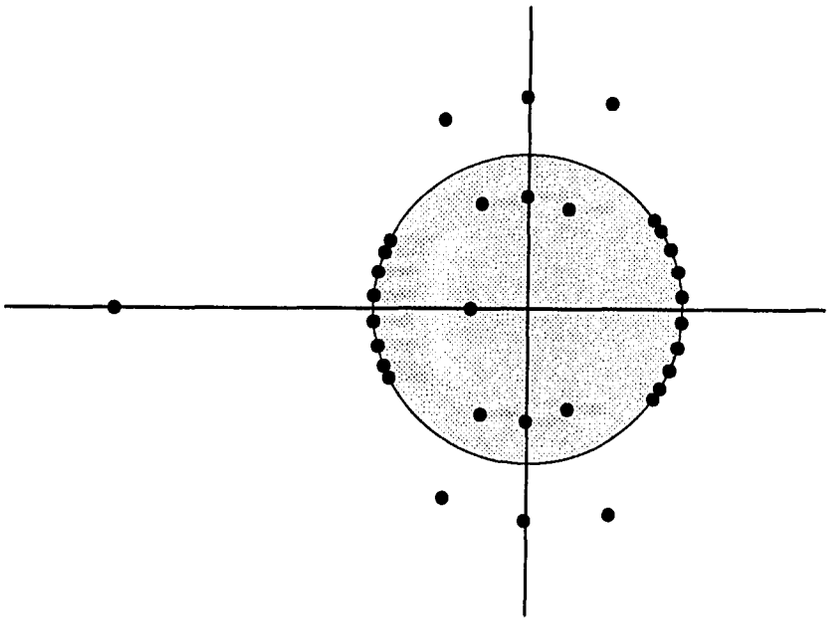

Figure 4.14: Zeros of the linear phase filter in example 4.4

Table 4.1: Impulse response coefficients for the linear phase filter of example 4.4. (a) Complex Remez algorithm, (b) Parks-McClellan program. Only half of the coefficients are shown

| (a) | (b) |
|---|---|
| -2.702875e-03 | -2.704665e-03 |
| -4.516627e-03 | -4.526726e-03 |
| 8.186741e-03 | 8.189951e-03 |
| -9.792692e-04 | -9.538837e-04 |
| -3.482344e-04 | -3.272372e-04 |
| 1.708382e-02 | 1.707943e-02 |
| -1.741194e-02 | -1.746195e-02 |
| -2.606369e-03 | -2.654066e-03 |
| -1.017401e-02 | -1.016346e-02 |
| -3.729828e-02 | -3.724241e-02 |
| 6.346552e-02 | 6.354181e-02 |
| 1.855073e-02 | 1.857321e-02 |
| 2.786971e-02 | 2.778989e-02 |
| 4.966967e-02 | 4.957702e-02 |
| -3.002305e-01 | -3.002637e-01 |
| -4.070676e-02 | -4.065302e-02 |
| 4.626913e-01 | 4.627986e-01 |

are very important in DSP implementations of single-sideband receivers. This example is a filter with length 35 (complex, i.e. N=70) with stopbands [-0.5,-0.04] and [.25,.5], and passband [.04,.2]. The desired group delay is =13 samples, and the weight ratio used is 10:1:5. The algorithm converged in 371 iterations and required 6 minutes and 10 seconds of CPU time. The dual variable plot is shown in Figure 4.15. The magnitude in dB is shown in Figure 4.16, with passband ripple $\delta_p = 0.03696$ (.315 dB), and attenuation in the first

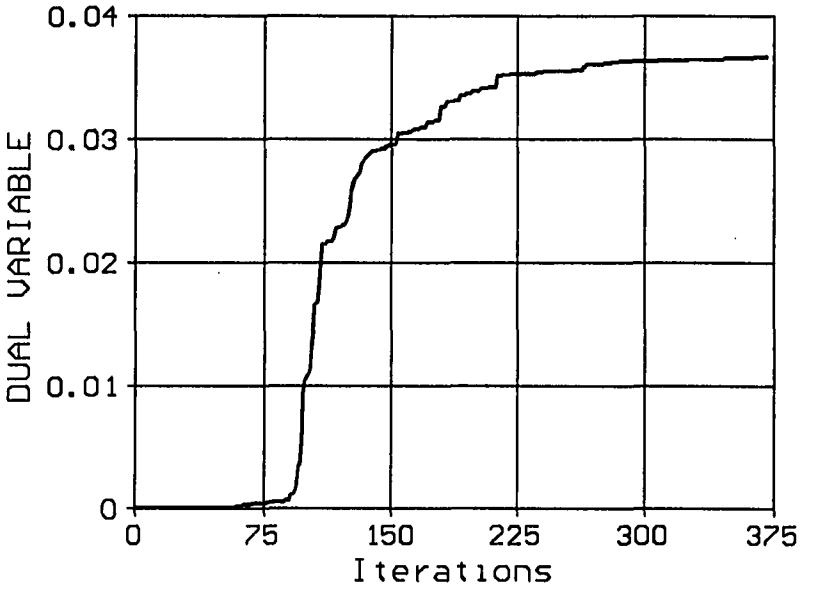


Figure 4.15: Dual variable plot for example 4.5

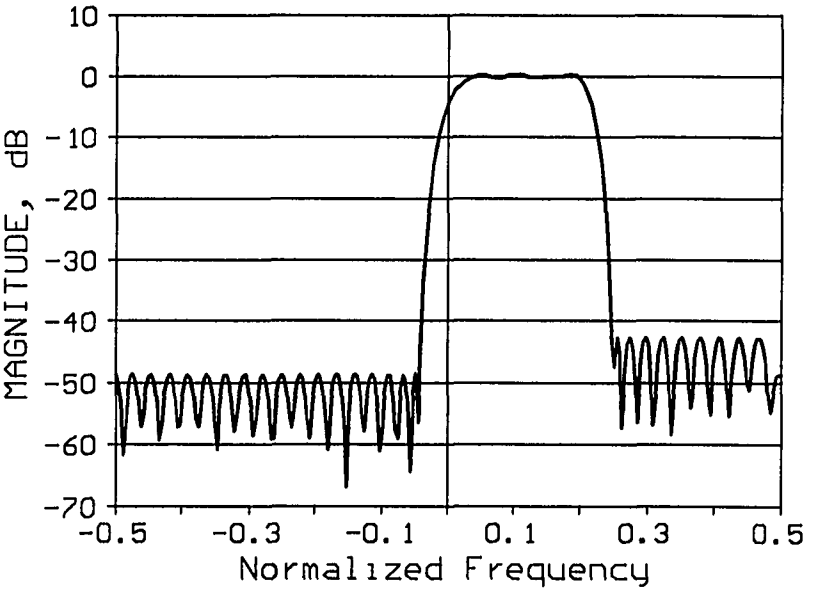


Figure 4.16: Magnitude response in dB for the complex filter in example 4.5

stopband $\delta_{s1} = 0.003696$ (48.645 dB), and attenuation in the second stopband $\delta_{s2} = 0.00739$ (42.624 dB). The filter group delay, and zero plot are shown in Figures 4.17 and 4.18 respectively. The group delay is nearly constant in the passband, with maximum deviation from the specified 13 samples, of about 1.24 samples occurring at the edge of the passband. It is seen in Figure 4.18 that there is no symmetry of zeros with respect to the real axis because the FIR filter is not conjugate symmetric. The complex impulse response coefficients are shown in Table 4.2.

### 4.2.4 Design of Hilbert Transformers

In this section we discuss the design of narrow-band and wide-band Hilbert transformers. The frequency response of an ideal Hilbert transformer is given by

$$H_{ideal}(f) = \begin{cases} -j & , \quad f \in B_p \quad f > 0 \\ j & , \quad f \in B_p \quad f < 0 \end{cases} \tag{4.3}$$

where $f \in [-.5, .5]$ and $B_p$ is the passband. The ideal Hilbert transformer is periodic in $f$, with period 1, and it is discontinuous at 0, 0.5, -0.5. To avoid large errors in both the magnitude and phase responses, a group delay must be specified. Also, since the frequency

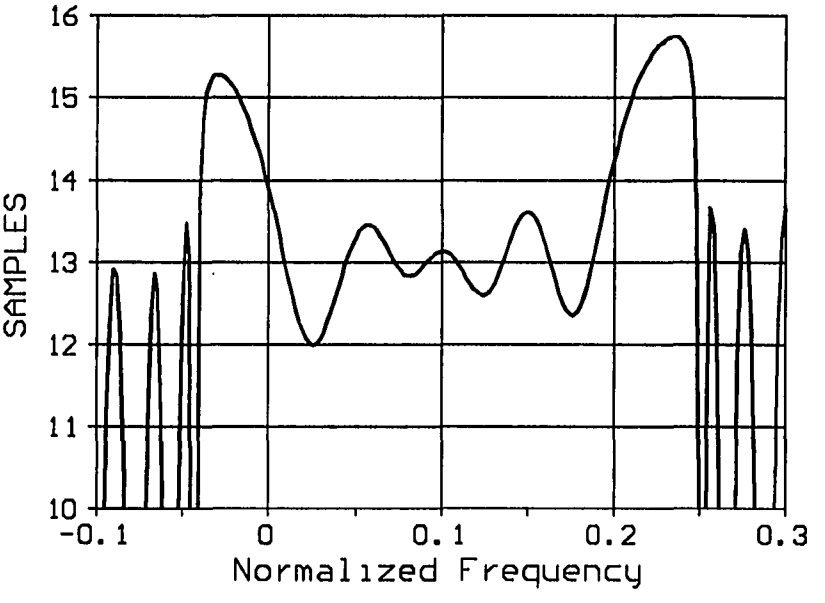


Figure 4.17: Group delay for the complex filter in example 4.5. The maximum group delay deviation from 13 samples is largest at the edge of the passband

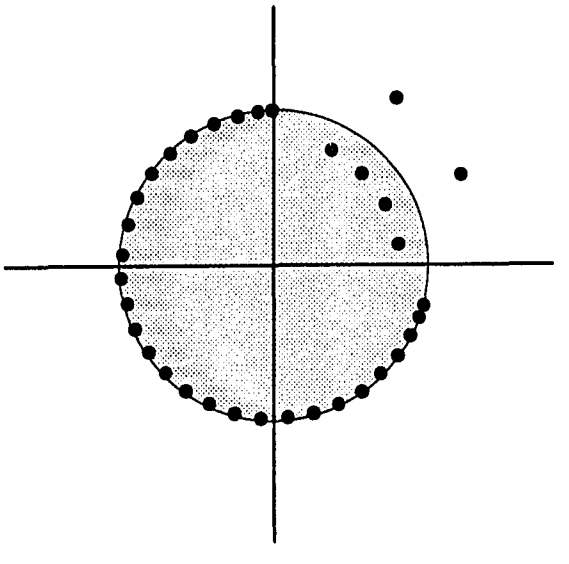

Figure 4.18: Zeros for the complex filter in example 4.5. As expected, the zeros are not conjugate symmetric since the coefficients are complex

Table 4.2: Complex impulse response coefficients of the filter in example 4.5

| | | | | |
|---|---|---|---|---|
| h(0) | = | -7.097166081553474E-3 | - j | 2.449869731874638E-3 |
| h(1) | = | -4.909853046017830E-3 | - j | 1.146660932435409E-2 |
| h(2) | = | 4.289096084777670E-3 | - j | 1.411606618853004E-2 |
| h(3) | = | 9.459599008774331E-3 | - j | 6.730734667903748E-3 |
| h(4) | = | 2.830157908894071E-4 | j | 1.308346148588824E-4 |
| h(5) | = | -1.230104170136316E-2 | - j | 1.187074729630058E-2 |
| h(6) | = | -3.935750914188269E-3 | - j | 3.205479320318927E-2 |
| h(7) | = | 1.844457967373864E-2 | - j | 3.082232910738495E-2 |
| h(8) | = | 1.692160060859027E-2 | - j | 7.369396214693416E-3 |
| h(9) | = | -2.157235295925872E-2 | - j | 1.095401199559976E-2 |
| h(10) | = | -3.974282244538216E-2 | - j | 7.513411918708487E-2 |
| h(11) | = | 2.711036534595051E-2 | - j | 1.480114410122990E-1 |
| h(12) | = | 1.532523016143514E-1 | - j | 1.315574837776295E-1 |
| h(13) | = | 2.226332010445301E-1 | - j | 5.331583938094072E-3 |
| h(14) | = | 1.617882670670987E-1 | j | 1.276860450452669E-1 |
| h(15) | = | 3.071642934059205E-2 | j | 1.534943866277171E-1 |
| h(16) | = | -4.588379078966925E-2 | j | 7.687168550597420E-2 |
| h(17) | = | -2.549668518504036E-2 | j | 2.487602594558106E-3 |
| h(18) | = | 2.317899049321154E-2 | - j | 8.562987328105028E-4 |
| h(19) | = | 2.644209543609863E-2 | j | 3.227911288109406E-2 |
| h(20) | = | -4.942770988777087E-3 | j | 3.641395875703429E-2 |
| h(21) | = | -1.722348926121762E-2 | j | 7.850610004098110E-3 |
| h(22) | = | 2.137993057059723E-3 | - j | 9.460697250750202E-3 |
| h(23) | = | 1.770032403466297E-2 | j | 2.434936734377863E-3 |
| h(24) | = | 8.116438266292256E-3 | j | 1.602770764935739E-2 |
| h(25) | = | -6.626594508985006E-3 | j | 8.854357157920349E-3 |
| h(26) | = | -3.885929788226283E-3 | - j | 5.840768270543552E-3 |
| h(27) | = | 8.102697976020187E-3 | - j | 6.450484978911618E-3 |
| h(28) | = | 9.797906032956867E-3 | j | 3.055376408744117E-3 |
| h(29) | = | 1.173547057601795E-3 | j | 5.663415429293717E-3 |
| h(30) | = | -2.728944766507320E-3 | - j | 1.820147222643994E-3 |
| h(31) | = | 2.862795208660757E-3 | - j | 6.351531955554874E-3 |
| h(32) | = | 7.990638546168480E-3 | - j | 2.719234171547787E-3 |
| h(33) | = | 6.507551565175226E-3 | j | 3.346413001817946E-3 |
| h(34) | = | 8.871053652445351E-4 | j | 4.223265457925279E-3 |

response for the two-sided Hilbert transformer is conjugate symmetric, its magnitude and phase only need to be specified for positive frequencies. The desired frequency response of a Hilbert transformer is given by

$$D(f) = \begin{cases} e^{-j\left(2\pi f \tau_d + \frac{\pi}{2}\right)} & , \quad f \in B_p \\ 0 & , \quad f \in B_s \end{cases} \tag{4.4}$$

When the group delay $\tau_d$ is an integer, the Hilbert transformer is not continuous at $f=0$, 0.5, and -0.5. This is because the phase function is discontinuous at these points. The desired frequency response can be made continuous at f=-0.5 and f=0.5 if we allow a specification of a non-integer group delay of the form $\tau_d = \tau_c + 0.5$, where $\tau_c$ is an integer. This is useful in the design of *wide-band* Hilbert transformers. When a group delay of this form is specified, our algorithm gives good results and converges rapidly. When an integer delay is specified, the algorithm sometimes does not converge and, when it does, the errors are usually large. Since the discontinuity at $f=0$ is an essential property of the Hilbert transform and cannot be avoided, a small stopband must be specified in the vicinity of zero to avoid large errors in the approximation. The Hilbert transformer is called *narrow-band* if the frequency response is nonzero only within a small band of the domain [0,0.5]. The discontinuities at $f=-0.5$ and $f=0.5$ are no problem in this case since a stopband is defined at these points.

**Example 4.6:** *Narrow-band Hilbert Transformer*

This example is a narrow-band Hilbert transformer of length N=42, with a passband [.04,.20], and stopbands [0.,0.005] and [.235,0.5]. A weight ratio of 1:1:1 is used. The group delay that gives the best chebychev error is $\tau_g = 14$ and is determined by trial-and-error. At this time we do not have a method to specify the group delay that gives the least Chebychev error other than trial-and-error. However, it has been observed that the smallest Chebychev error occurs at a specified delay that is less than half the filter length (linear-phase case). For this example the algorithm required 255 iterations and 5 minutes CPU time. The dual variable is plotted in Figure 4.19. The magnitude response in dB of the narrow-band Hilbert transformer is shown in Figure 4.20. The passband ripple is $\delta_p = 0.0297$ (0.254 dB) and stopband attenuation is $\delta_s = 0.0297$ (30.54 dB). The magnitude error in the passband between the desired response and the Hilbert transformer is shown in Figure 4.21 and the passband error in dB is shown in Figure 4.22. Figure 4.23 shows the group delay of the transformer which is nearly equiripple in most of the passband with the maximum deviation occurring at the edges of the passband. It is also observed that the minimum maximum deviation of the group delay from a constant occurs in the case of the least Chebychev error. Figure 4.24 shows the passband phase error in degrees with a maximum error of about 1.7 degrees. The zeros of the transformer are shown in Figure 4.25.

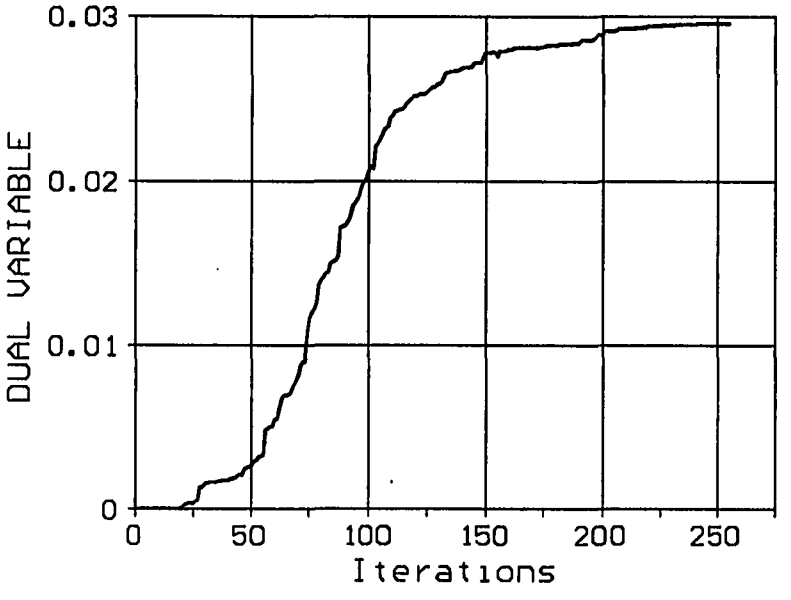


Figure 4.19: Dual variable plot for example 4.6

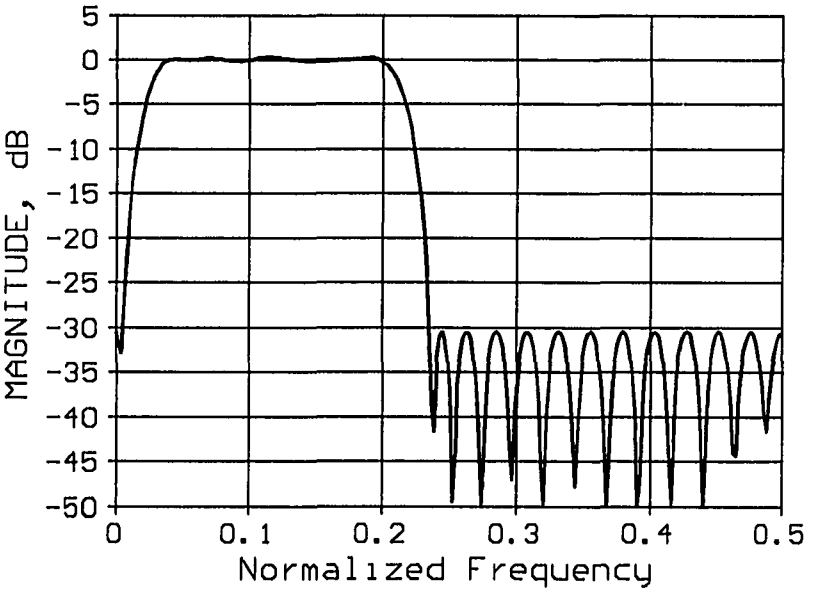


Figure 4.20: Magnitude in dB of the narrow-band Hilbert transformer in example 4.6

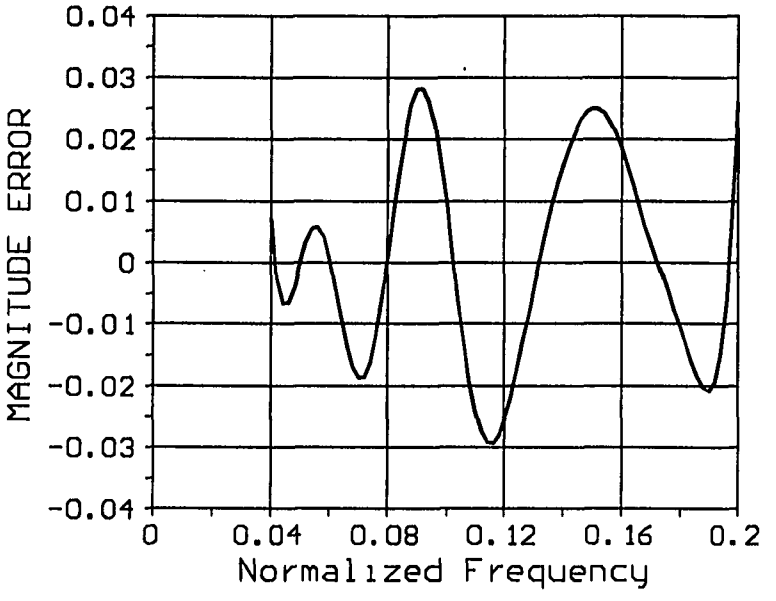


Figure 4.21: Magnitude error in the passband of the narrow-band Hilbert transformer in example 4.6

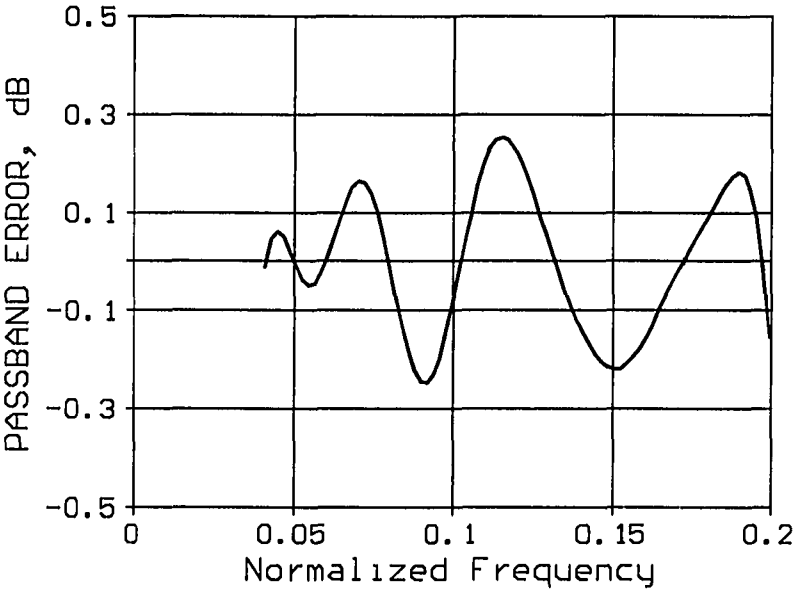


Figure 4.22: Passband magnitude error in dB in example 4.6

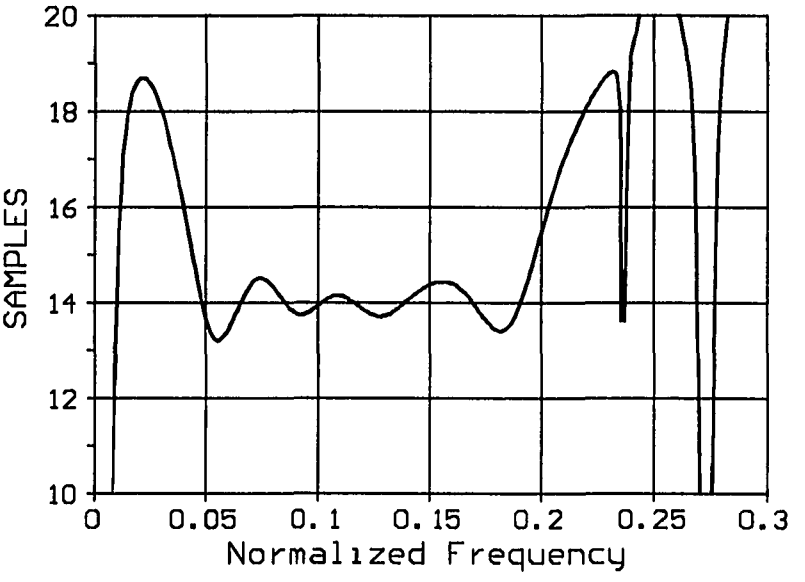


Figure 4.23: Group delay in the passband for the narrow-band Hilbert transformer in example 4.6. The deviation in the passband is largest at the edges of the passband

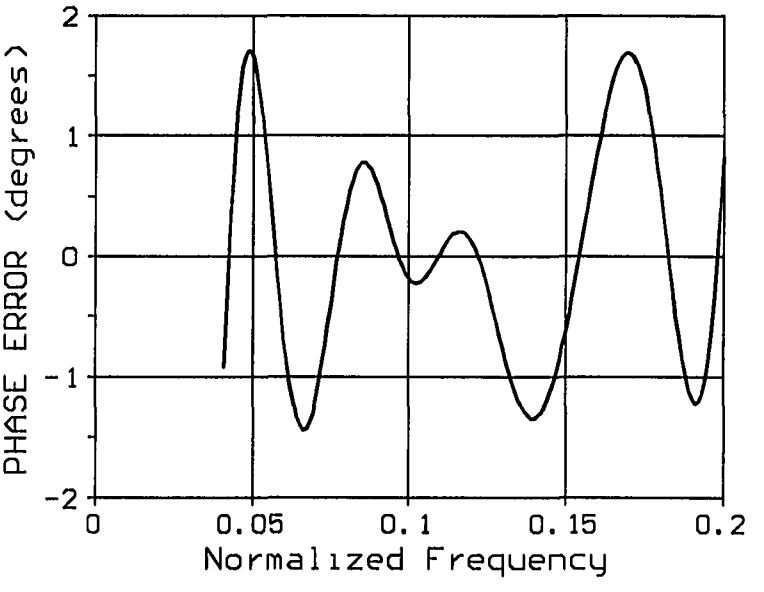


Figure 4.24: Passband phase error in degrees for the Hilbert transformer in example 4.6. The maximum error is about 1.7 degrees

**Example 4.7:** *Wide-band Hilbert Transformer*

As discussed previously, a small stopband in the vicinity of zero is specified in the design of a wide-band Hilbert transformer to avoid the discontinuity at that point. This stopband should be small enough to prevent a zero from occurring in that region and avoid the production of a ripple. A small stopband will result in a smooth monotonically increasing transition band. Note that in the design of a linear-phase Hilbert transformer no stopband needs to be specified since its magnitude response is inherently zero at zero.

This wide-band Hilbert transformer example has a desired passband defined in [0.04,0.5] and a small stopband in [0.0,0.002]. The length is N=42 and the weight ratio is 1:1. An exhaustive test was made to determine which group delay value would produce the smallest optimal Chebychev error. The plot in Figure 4.26 shows that the minimum error occurs at 10.5 and 30.5 samples. The curve is symmetric around the midpoint. We have not investigated efficient methods of finding what specification of group delay results in the minimum Chebychev error for a given approximation. Using repeated trials, we have observed that a minimum occurs at two points, as shown in Figure 4.26, and these points are quite remote from the half-length point. The algorithm converged in 138 iterations and required 2 minutes of CPU time. The dual variable is shown in Figure 4.27. The magnitude response of the transformer is shown in Figure 4.28 and the magnitude in dB is shown in Figure 4.29. The Chebychev error is 0.0146. The magnitude error in the passband is shown in Figure 4.30 and the magnitude error in dB is shown in Figure 4.31. The error is not quite equiripple in the passband. Figure 4.32 shows the group delay which has a small deviation

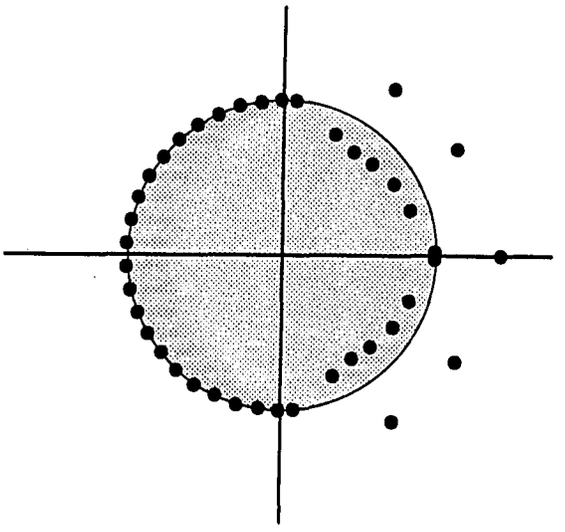

Figure 4.25: Zeros for the narrow-band Hilbert transformer in example 4.6

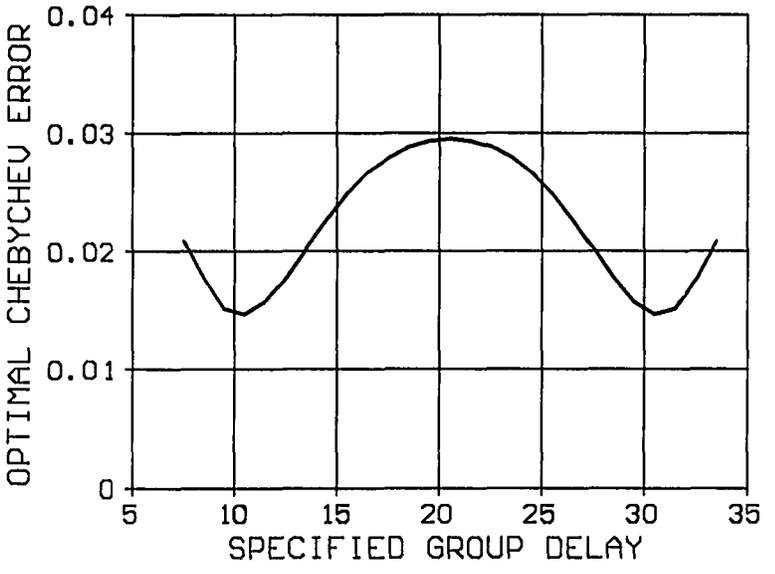


Figure 4.26: Magnitude of the optimal Chebychev error versus specified group delay for example 4.7. The minimum maximum magnitude error occurs when 10.5 or 30.5 samples is specified as the desired group delay

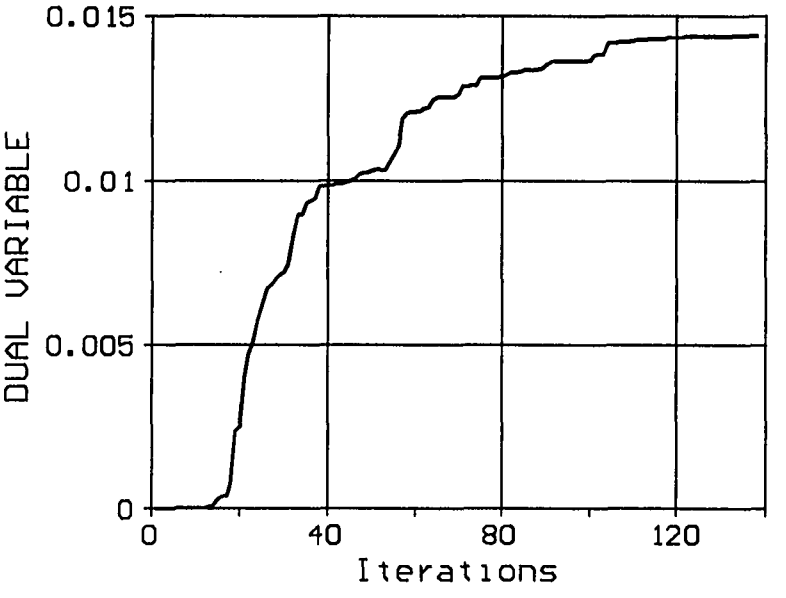


Figure 4.27: Dual variable plot for example 4.7

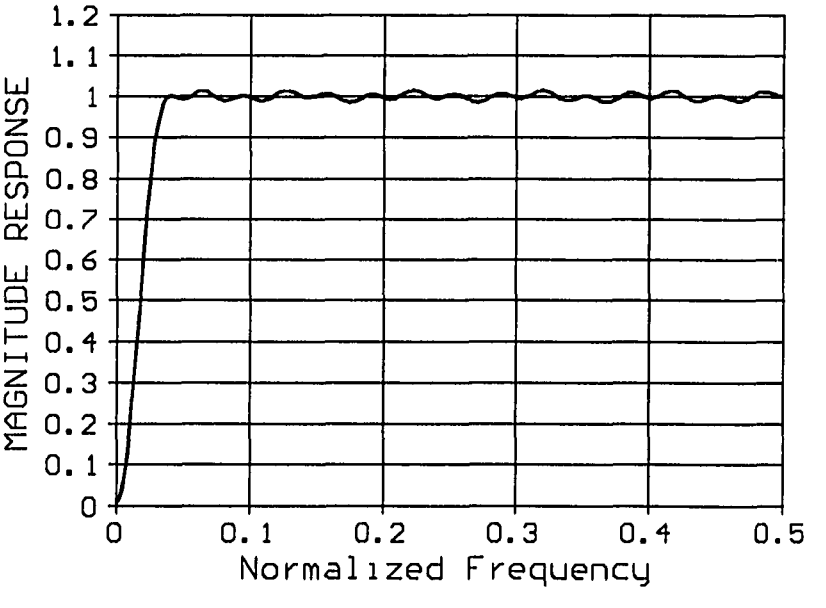


Figure 4.28: Magnitude of the wide-band Hilbert transformer in example 4.7

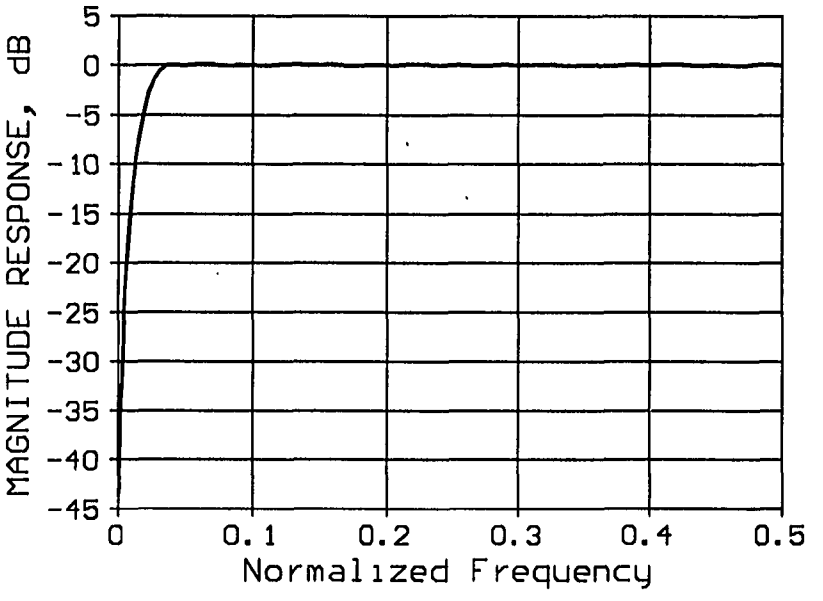


Figure 4.29: Magnitude in dB of the Hilbert transformer in example 4.7

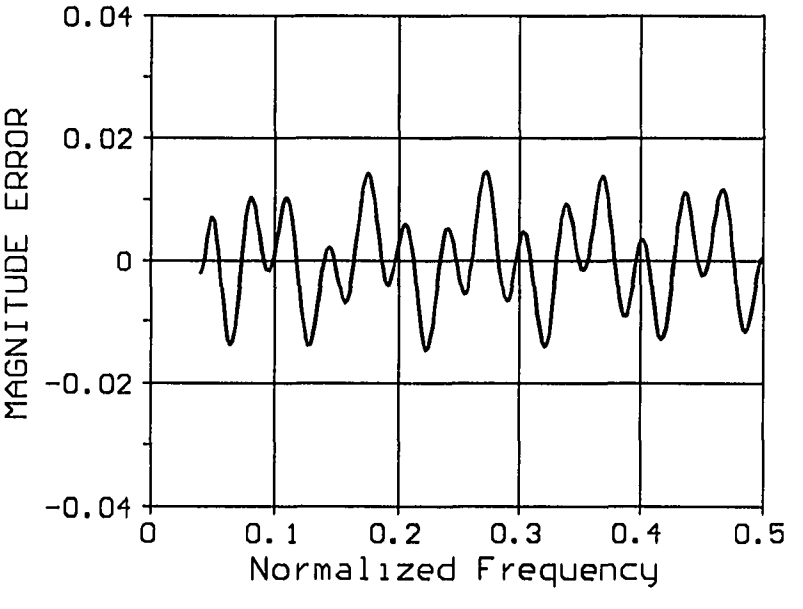


Figure 4.30: Magnitude error in the passband for the wide-band Hilbert transformer in example 4.7

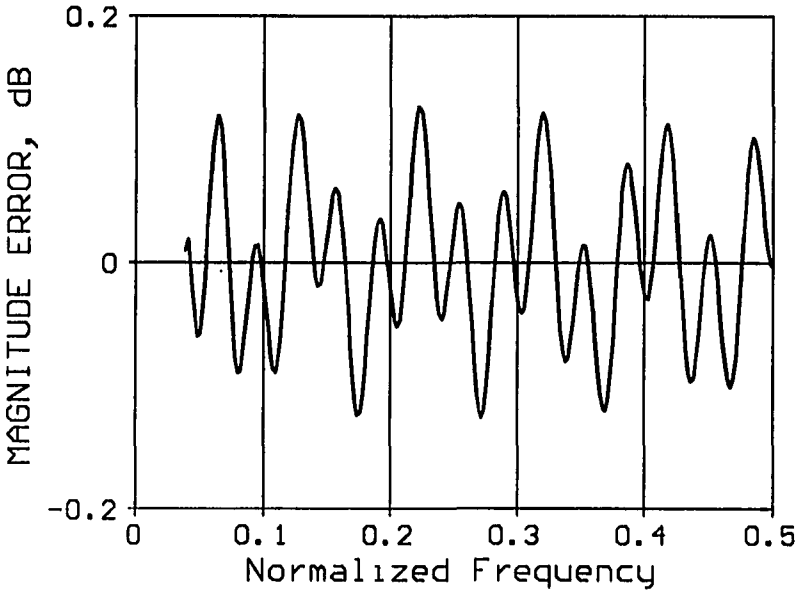


Figure 4.31: Passband magnitude error in dB in example 4.7

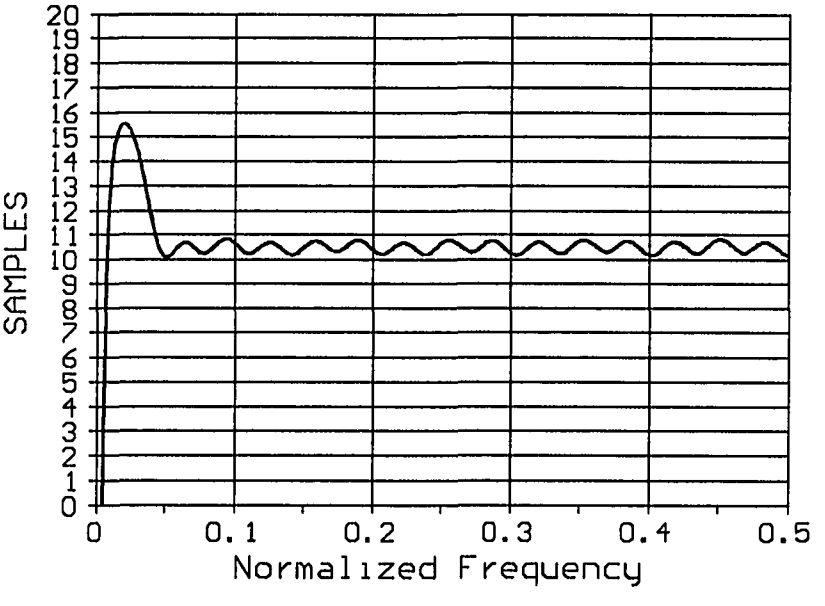


Figure 4.32: Group delay of the wide-band Hilbert transformer in example 4.7. The deviation from the specified delay of 10.5 samples is within one half sample

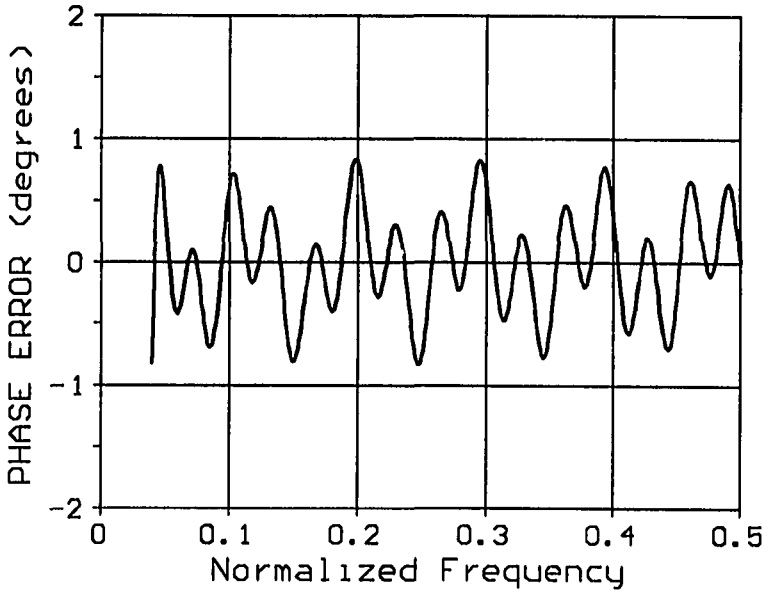


Figure 4.33: Passband phase error for the Hilbert transformer in example 4.7

from a constant in the passband and a large ripple in the transition band. The passband phase error is shown in Figure 4.33. The maximum phase error is about 0.82 degrees.

**Example 4.8:** *One-Sided Hilbert Transformer*

It is sometimes required that a Hilbert transformer be one-sided. In this case the Hilbert transformer should be specified in the whole interval [-0.5,0.5] since it is no longer conjugate symmetric. The resulting impulse response coefficients are complex numbers. The desired response of the one-sided Hilbert with non-zero response only for positive frequencies is defined as

$$D(f) = \begin{cases} e^{-j\left(2\pi f \tau_d + \frac{\pi}{2}\right)} & , \quad f \in B_p \\ 0 & , \quad f \in B_s \\ 0 & , \quad f < 0 \end{cases} \tag{4.5}$$

Since the Hilbert transformer is periodic, and a stopband is defined for negative frequencies, a small stopband should be defined in the vicinity of $f=0.5$ to avoid the discontinuity at this point. The example we present here has stopbands in the intervals [-0.5,0.002] and [0.498,0.5] and a passband in the interval [0.04,0.46]. The length of the complex impulse response is 22 (N=44) and the weight ratio is 1:1:1. A group delay of 10 samples is specified. For this example, the algorithm required 243 iterations and 2 minutes and 5 seconds CPU time. The dual variable plot is shown in Figure 4.34. The magnitude in dB is shown in Figure 4.35. The optimal error is 0.0891 which results in a passband ripple of 0.741 dB and stopband attenuation of 21 dB. The group delay is shown in Figure 4.36. In the passband, the deviation of the group delay from a constant is small. Figure 4.37 shows the zeros of the one-sided Hilbert transformer. The complex impulse response coefficients are shown in Table 4.3.

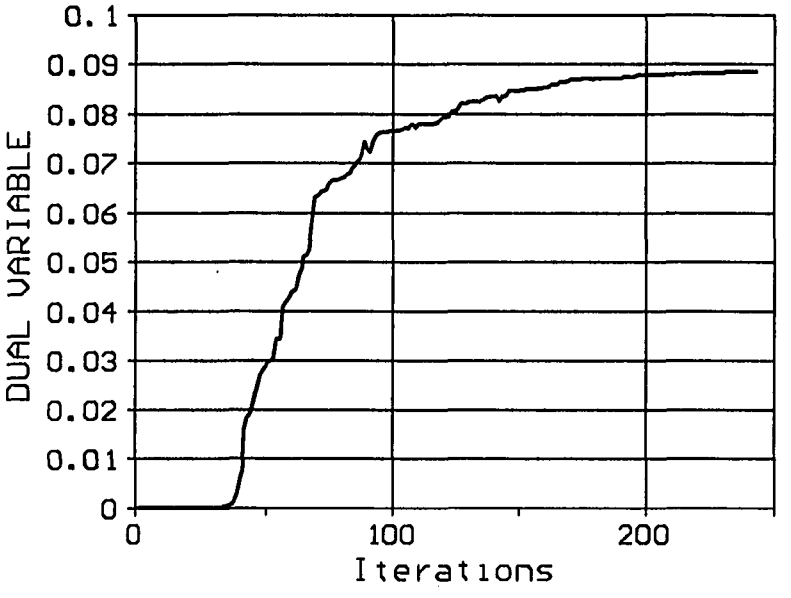


Figure 4.34: Dual variable plot for example 4.8

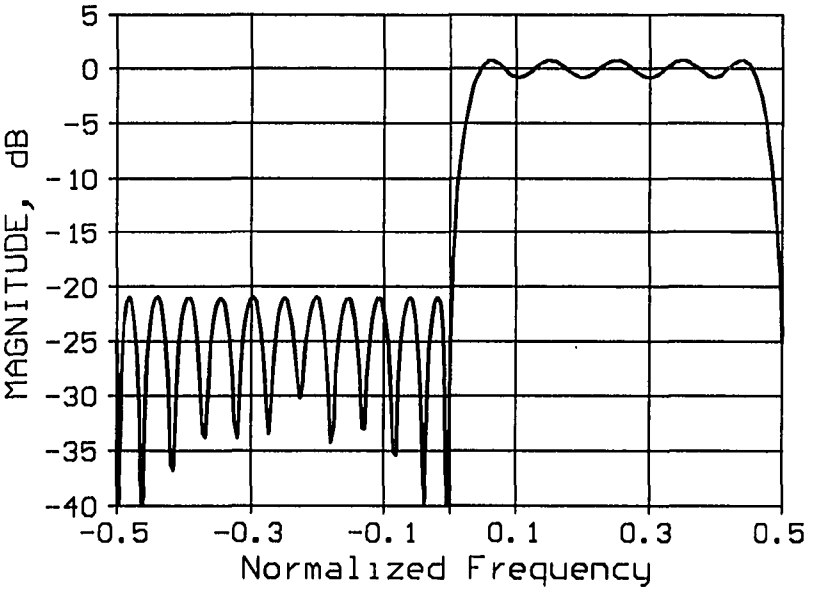


Figure 4.35: Magnitude in dB of the one-sided Hilbert transformer in example 4.8

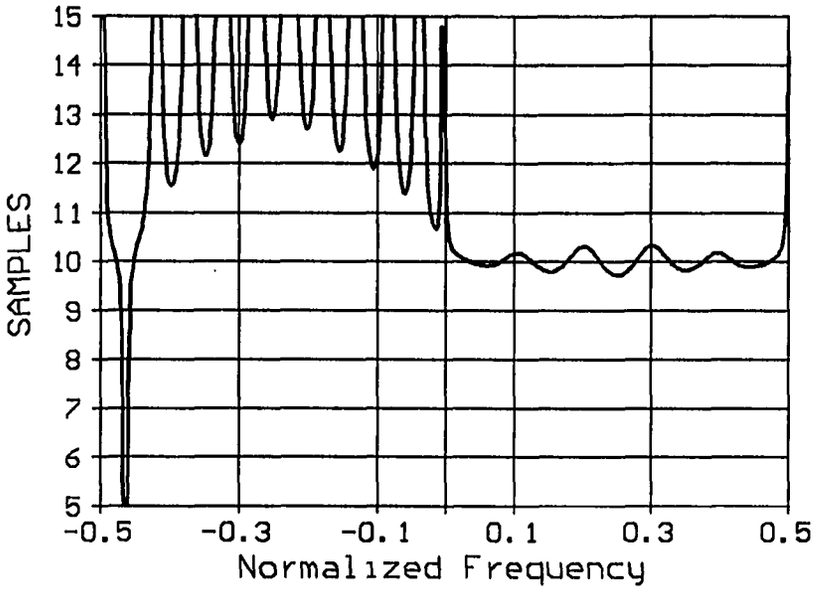


Figure 4.36: Group delay of the one-sided Hilbert transformer in example 4.8. The group delay is almost constant in the passband

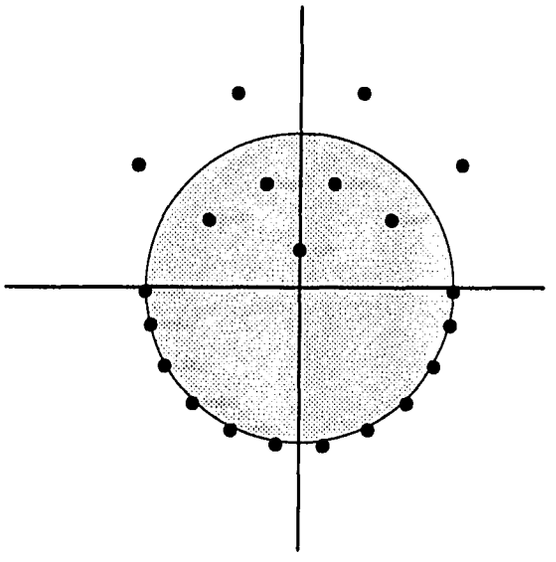

Figure 4.37: Zeros of the one-sided Hilbert transformer in example 4.8

Table 4.3: Complex impulse response of the Hilbert Transformer of example 4.8

| | | |
|---|---|---|
| h(0) | = 5.024054399750122E-004 | + j 5.803458489559261E-002 |
| h(1) | = -2.109919480117686E-002 | - j 7.491866424560872E-004 |
| h(2) | = -3.052826714313450E-004 | + j 2.859926602396223E-002 |
| h(3) | = -2.358623256517073E-002 | - j 7.324751903549087E-004 |
| h(4) | = -5.199349793508223E-005 | + j 3.507186188646441E-002 |
| h(5) | = -4.784747706850923E-002 | - j 1.064632616731648E-003 |
| h(6) | = -2.224732845440835E-004 | + j 3.888075831197935E-002 |
| h(7) | = -9.538333287373696E-002 | + j 5.630061007205200E-004 |
| h(8) | = -5.181869186081101E-004 | + j 3.993434652363943E-002 |
| h(9) | = -3.142588562125434E-001 | - j 1.573334097675440E-003 |
| h(10) | = 2.398180093382760E-004 | - j 4.579256903041351E-001 |
| h(11) | = 3.153266271365550E-001 | + j 1.683240508608685E-003 |
| h(12) | = -5.206936034853893E-004 | + j 4.214473903259816E-002 |
| h(13) | = 9.607820548626876E-002 | - j 4.901288804872728E-004 |
| h(14) | = -2.074358319945069E-004 | + j 3.809553641787688E-002 |
| h(15) | = 4.677345640464542E-002 | + j 1.085878024139619E-003 |
| h(16) | = -1.556128007118990E-004 | + j 3.403622719448390E-002 |
| h(17) | = 2.510061774435768E-002 | + j 7.063378367792167E-004 |
| h(18) | = -4.246270913411225E-004 | + j 2.987655807860095E-002 |
| h(19) | = 8.534397986236231E-003 | + j 7.593297322360248E-004 |
| h(20) | = 4.508743389454450E-004 | + j 5.671294496341162E-002 |
| h(21) | = 1.371124055960415E-002 | + j 1.260663140062110E-004 |

### 4.2.5 Design of Differentiators

Differentiators are used to obtain samples of the derivative of a bandlimited signal from the samples of that signal. For a signal $s(t)$ and its Fourier transform $S(f)$ the following relation exists:

$$\frac{d\,s(t)}{dt} \quad \leftrightarrow \quad (j2\pi f)\; S(f) \tag{4.6}$$

Thus, the Fourier transform of the derivative of a signal is $(j2\pi f)$ times the Fourier transform of the signal. The desired frequency response of the differentiator to be used in our design algorithm is given by

$$D(f) = \begin{cases} 2\pi f\, e^{-j\left(2\pi f \tau_d - \frac{\pi}{2}\right)} & , \quad f \in B_p \\ 0 & , \quad f \in B_s \end{cases} \tag{4.7}$$

For the full-band differentiator, the desired response is not continuous at the endpoints of the frequency domain $f$=-0.5 and $f$=0.5 when the group delay is an integer. As before, it can be made continuous by specifying a group delay of the form $\tau_d = \tau_c + 0.5$ where $\tau_c$ is an integer.

To avoid a large group delay error in the vicinity of $f$=0, an inverse weighting function of the form

$$W(f) = \frac{1}{f+\Delta f} \qquad f \in F \tag{4.8}$$

is used to force the error and filter response to be zero at $f$=0. The value of $\Delta f$ is small and is used to avoid dividing by zero at $f$=0. The use of this inverse error weight causes the Chebychev error to increase starting at $f$=0 and increase to the largest value at $f$=0.5. In the case of a narrow-band differentiator, this is not necessary.

**Example 4.9:** *Narrow-band Differentiator*

This example is a narrow-band differentiator of length 42 with a passband defined on [0.04,0.2] and stopbands on [0.,0.005] and [0.24,0.5]. A group delay of 16 samples is defined since it results in the least magnitude error. The algorithm required 254 iterations and 5 minutes of CPU time. The dual variable is plotted in Figure 4.38. The magnitude response of the differentiator is shown in Figure 4.39. The optimal Chebychev error obtained by the design program is 0.02548 which results in a passband ripple of 0.2185 dB and stopband attenuation of 31876 dB. The magnitude error in the passband is shown in Figure 4.40. The magnitude error is not equiripple. Figure 4.41 shows the group delay in the passband. Figure

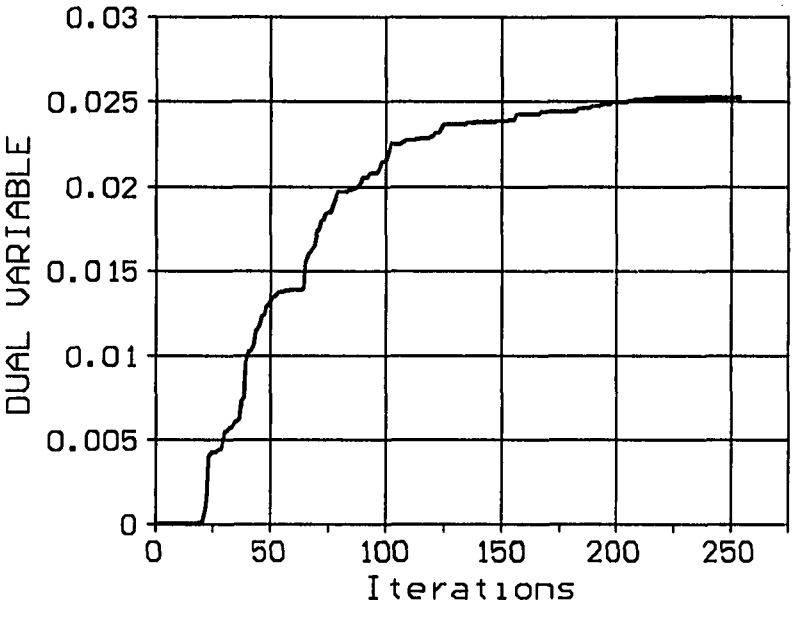


Figure 4.38: Dual variable plot for example 4.9

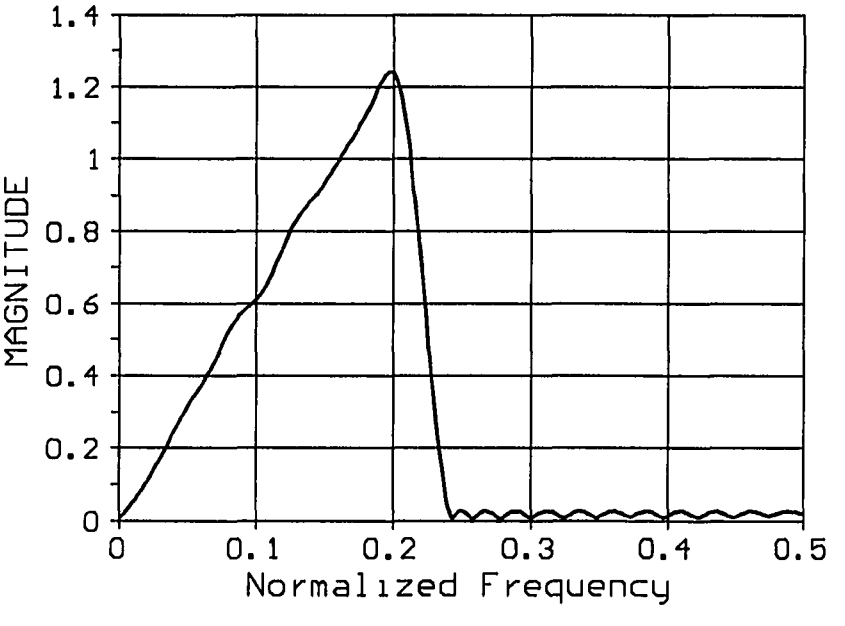


Figure 4.39: Magnitude response of the narrow-band differentiator of example 4.9

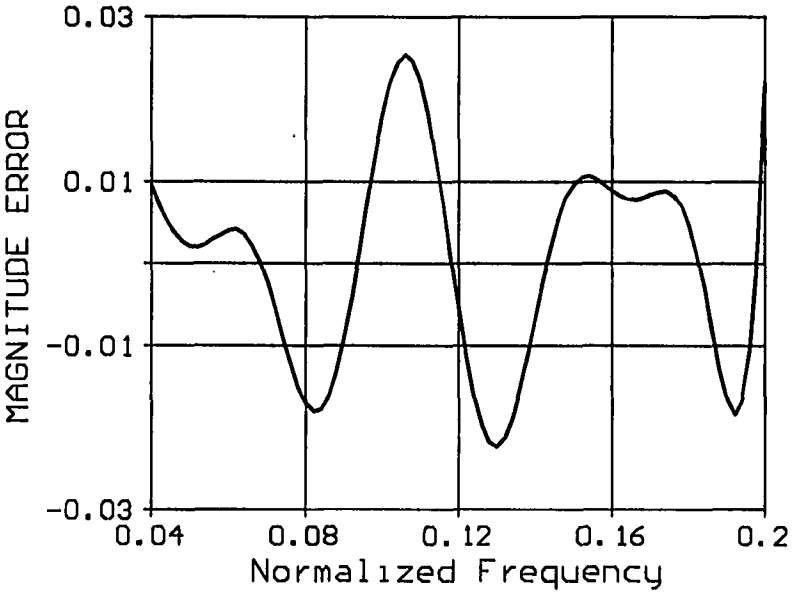


Figure 4.40: Magnitude error in the passband for the narrow-band differentiator of example 4.9

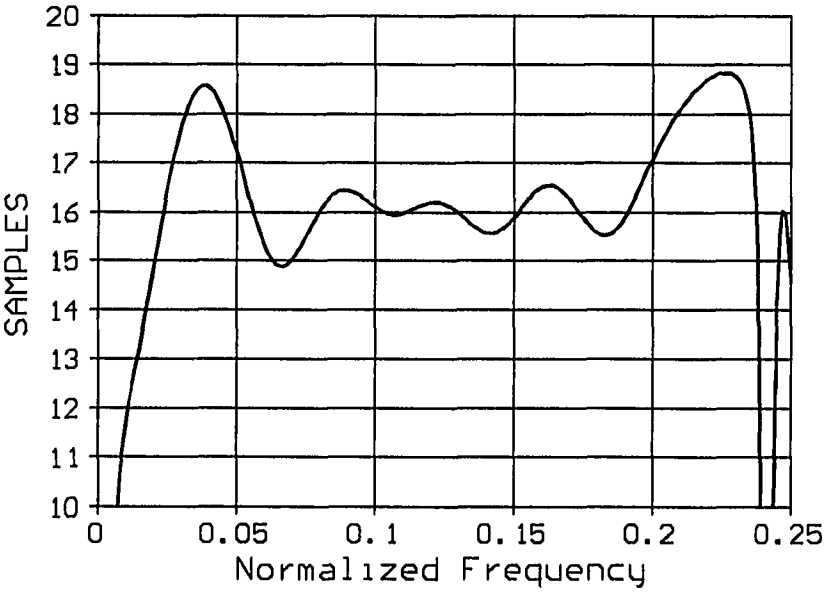


Figure 4.41: Group delay in the passband for the differentiator of example 4.9

4.42 shows the phase error in degrees with the largest error occurring at the lower edge of the passband. Finally, Figure 4.43 shows the zeros of the narrow-band differentiator.

**Example 4.10:** *Full-band Differentiator*

This example shows the design of a full-band differentiator of length 46. The passband is defined on [0,0.5] and the group delay is chosen to be 14.5 samples. The algorithm required 208 iterations and 3 minutes and 25 seconds CPU time. The dual variable plot is shown in Figure 4.44. The chebychev error of the approximation is 0.014865. The magnitude response is shown in Figure 4.45 and the magnitude error is shown in Figure 4.46. Note that the error is smallest at zero frequency and increases with frequency until the largest error occurs at f=0.5, since an inverse weight function has been used. The group delay is shown in Figure 4.47 with maximum deviation of less than 0.1 samples from the specified constant of 14.5 samples. The phase error in degrees is shown in Figure 4.48. The zeros of the full-band differentiator are shown in Figure 4.49.

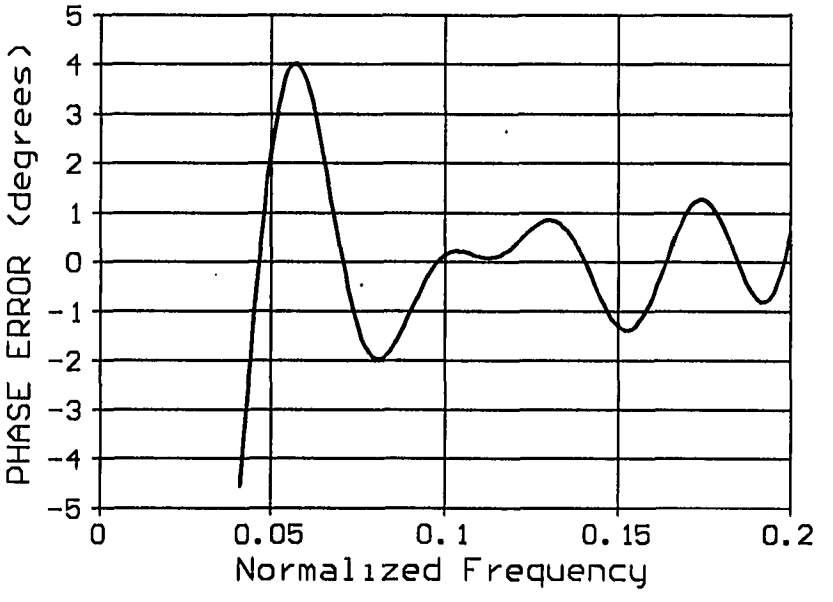


Figure 4.42: Phase error in degrees for the narrow-band differentiator in example 9. The phase error is largest at the lower edge of the passband

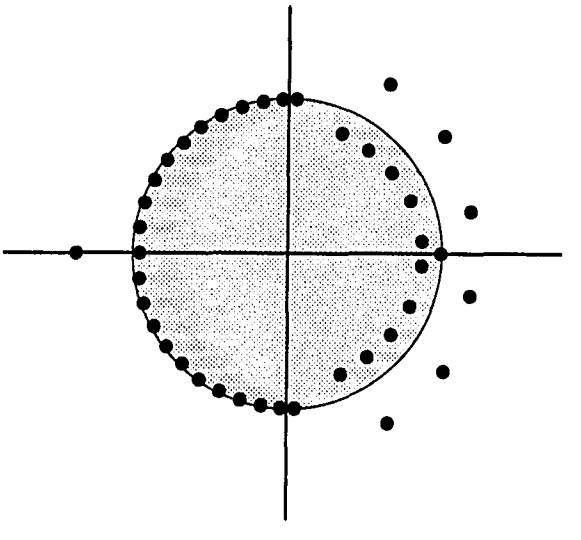

Figure 4.43: Zeros for the differentiator of example 4.9

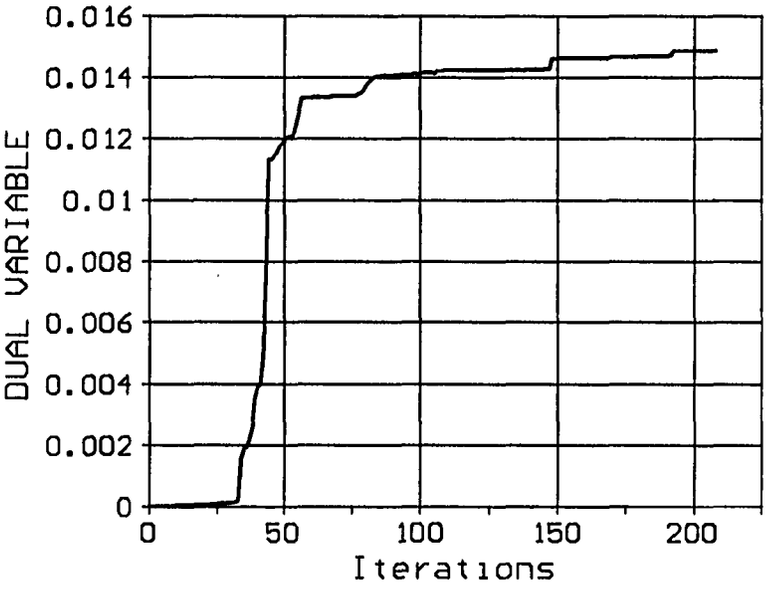


Figure 4.44: Dual variable plot for example 4.10

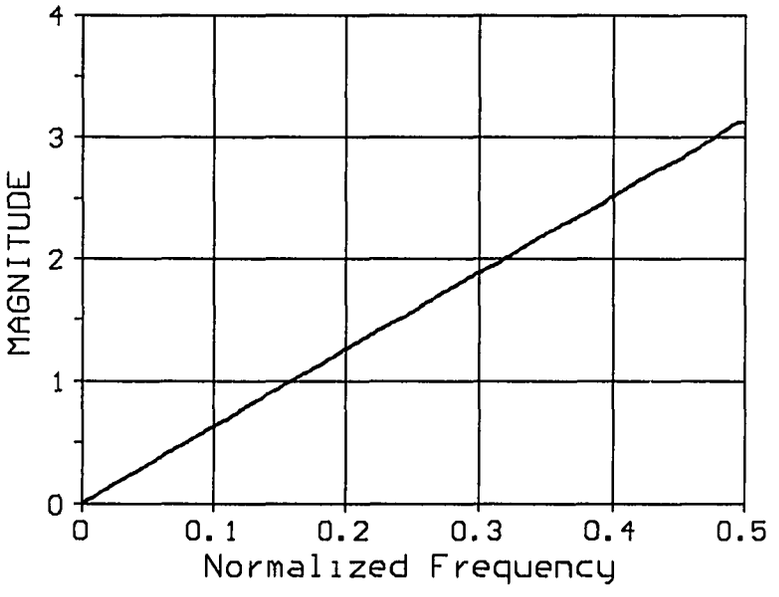


Figure 4.45: Magnitude of the full-band differentiator of example 4.10

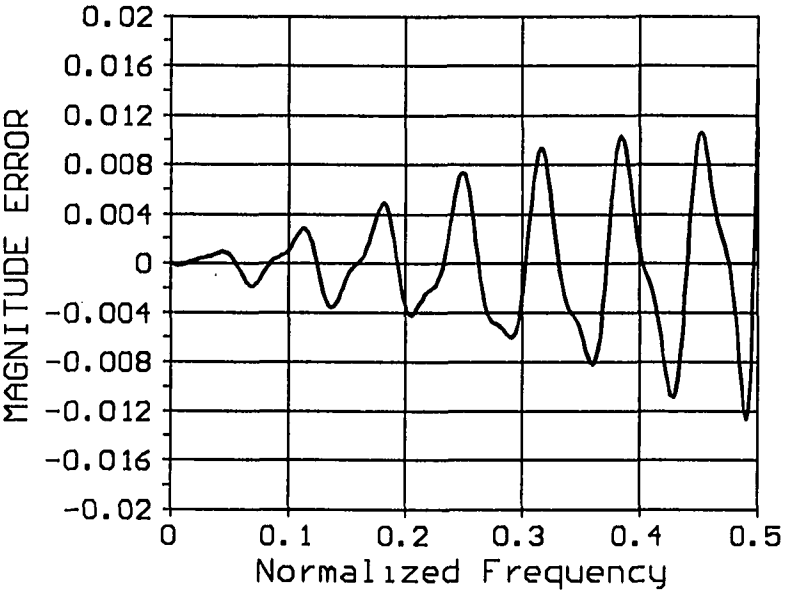


Figure 4.46: Magnitude error of the full-band differentiator of example 4.10

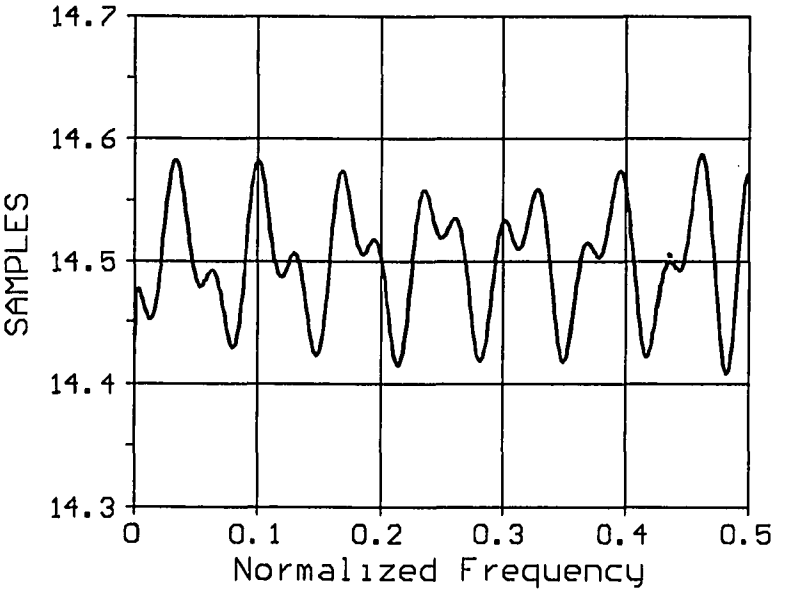


Figure 4.47: Group delay of the full-band differentiator in example 4.10. The maximum group delay error is less than 0.1 sample

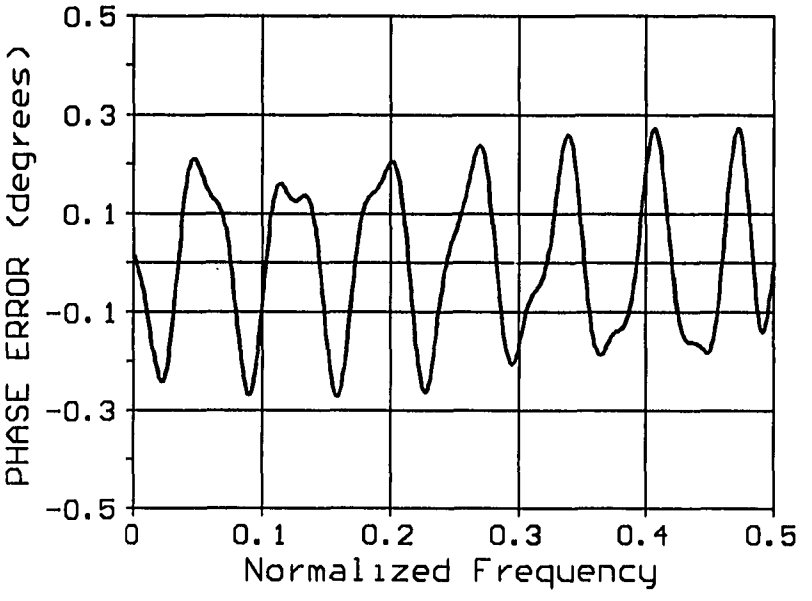


Figure 4.48: Phase error in degrees for the full-band differentiator in example 4.10

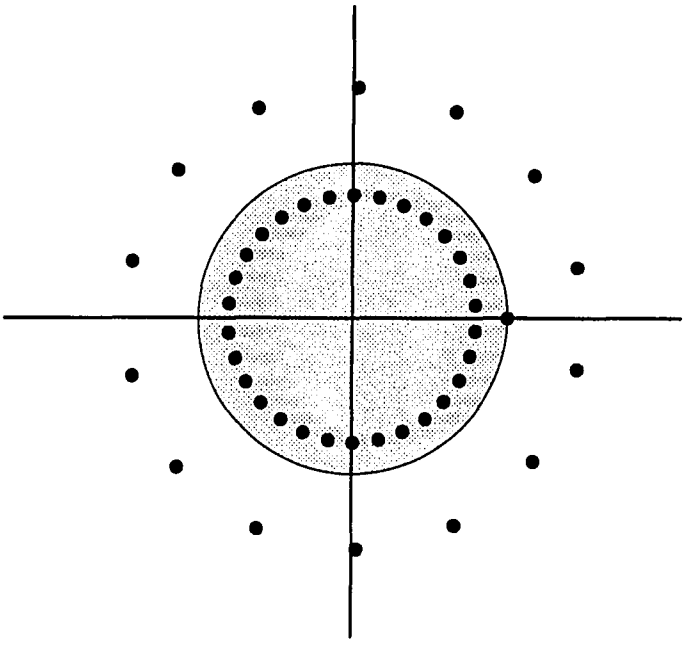

Figure 4.49: Zeros for the full-band differentiator of example 4.10

## 4.3 Conclusion

Design examples were presented to show the use of the filter design algorithm. Initially, we presented the design of frequency selective filters with real coefficients and nearly linear phase in the passband. The minimum maximum value of the absolute error occurs when the specified group delay is smaller than half the length of the filter. The algorithm converges for rather large order and stringent transition band requirements as was demonstrated by example 4.3.

We presented the design of linear-phase filters which is a special case of our algorithm. Because of the symmetry of the impulse response only half of the coefficients of the filter need to be determined. The example has shown that our algorithm results in linear-phase filters very close to the filters produced by the Parks-McClellan program.

One of the most powerful features of the design algorithm is its ability to design filters with frequency responses that carry no special symmetries. When the frequency response is conjugate symmetric the filter impulse response is a real sequence. When the desired frequency response does not have this symmetry the impulse response is a complex sequence. Non-conjugate symmetric filters are useful in single-sideband modulation and quadrature modulation. The complex coefficient designs present no problem to the algorithm since this problem is an extension of the real coefficient case with double the order. This is achieved by separating the complex coefficients and the basis functions in their real and imaginary parts.

Hilbert transformers are an important part of filter design. In this chapter we discussed the design of both narrow-band and wide-band Hilbert transformers using our complex approximation algorithm. It was shown, by examples, that in most cases the minimum Chebychev error of the magnitude response does not occur for a specified group delay of half the transformer length but at two other locations of the group delay. A one-sided Hilbert transformer was also designed. The impulse response of this transformer is a complex sequence.

Finally we presented the design of differentiators. Differentiators are used to obtain the derivative of a bandlimited signal from the samples of that signal. The design of full-band differentiators requires a non-integer specification of the group delay in order to avoid the discontinuity at the endpoints of the frequency domain. The freedom to specify non-integer group delays allows the design of a truly full-band differentiator without the need to include a small stopband at the endpoints. Also, designs with group delay lower than that of linear-phase designs can be achieved at the expense of a small group delay distortion in the passband.

# GENERAL SUMMARY

Until recently, linear-phase FIR filters have commonly been used in most applications. Linear-phase filters are popular since they offer linear phase, and thus constant group delay. Also, the design is relatively easy since it is a real approximation problem, and can be efficient using the Remez algorithm and the Parks-McClellan design program. Although the group delay is constant, it can be very large for long filters since it is always one half of the filter length.

The objective of this research work is to produce a practical design method for FIR filters that would allow the specification of both the magnitude and phase responses. This allows the design of filters with a smaller group delay than half the filter length. Also, for applications requiring filters with specific phase response requirements, other than linear phase, a design method is needed that considers the phase function in the approximation. Another motivation for developing the new approach was to examine FIR filters with nonlinear phase for better results in magnitude response characteristics compared to linear phase filters.

The first step in our research was the implementation of a design method for minimum-phase filters. The method uses a linear-phase prototype filter of twice the length of the desired minimum-phase filter. The minimum-phase filter is derived from the prototype linear-phase filter by direct factorization of the complex-valued polynomial representing the

linear-phase filter. This is possible because the prototype filter can be separated into a minimum-phase system and a non-minimum-phase system. The filters produced have minimum phase and group delay among all FIR filters with the same magnitude response. Also in most cases, the magnitude response of a minimum-phase filter is better than the magnitude response of a linear-phase filter of the same length and cutoff frequencies.

This method is efficient for filters with short lengths. For longer filters, a difficulty arises in the factorization of high order degree polynomials. Most common zero-finder methods fail to give accurate results. Our implementation is sufficiently accurate for filters with lengths of less than 70. Although the phase is minimum, it is highly nonlinear, resulting in a non-constant group delay which introduces group delay distortion on an input signal. The group delay cannot be controlled or predicted in the design of minimum-phase filters.

A more powerful FIR filter design method is developed using complex approximation. This method allows specification of the phase function of the filter. The filter design problem is expressed as a minimization of a linear optimization problem. The primal minimization problem is a semi-infinite problem because of the continuous frequency domain. This means, that there is a finite number of variables to be found subject to an infinite number of constraints.

One method of solving the maximization problem is to discretize the frequency and angle domains and derive the corresponding linear program. The linear program can be solved using the Simplex method. The solution is close to the optimal when fine grids of

angle and frequency are used. Because of the discretization, the number of equations to be solved is large. For large order problems, the CPU and memory requirements increase considerably.

A more efficient method to solve the primal semi-infinite problem is to derive the dual maximization problem. This approach has been considered in this dissertation. The conversion of the primal problem into its dual results in a finite number of constraint equations. The solution of the problem involves the determination of finite sets of frequencies and angles that characterize the optimal Chebychev solution. The solution of the dual problem is accomplished using a generalization of the Remez algorithm in the complex domain developed by Tang [14]. The algorithm seeks the set of points and angles that produce the optimal approximating polynomial in the Chebychev sense. The method is very efficient and guarantees a solution.

This method has been developed into an FIR filter design method that allows separate specification of the magnitude and phase responses of the filter. We have examined the design of filters with real coefficients and group delay that is lower than half the filter length. These filters have a slightly non-constant group delay, and when compared to linear-phase filters of the same length, they have somewhat better magnitude responses. When the group delay is specified to be half the filter length, the algorithm produces linear-phase filters, exactly the same as those produced by the Parks-McClellan method. Therefore, the design of optimal Chebychev linear-phase FIR filters is a special case of our design method.

We have examined the design of FIR filters that approximate a non-conjugate symmetric frequency response. These filters have complex impulse responses. This problem can be reduced to a problem with real coefficients using appropriate definition of the approximating basis functions and coefficients. The design of Hilbert transformers and differentiators has also been examined.

The method proposed in this dissertation is a generalization of the design of FIR filters. The generalization includes the design of linear and nonlinear phase filters. It is also capable of approximating non-conjugate frequency responses. The method has implemented with a general design program in FORTRAN.

## REFERENCES


[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[2] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*, Macmillan, New York, 1988.

[3] L. R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1975.

[4] L. R. Rabiner, "The Design of Finite Impulse Response Digital Filter Using Linear Programming Techniques," *Bell Syst. Tech. J.*, Vol. 51, pp. 1177-1198, July-Aug. 1972.

[5] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-21, No. 6, pp. 506-526, December 1973.

[6] E. Ya. Remez, *General Computational Methods of Chebychev Approximation*, Atomic Energy Translation 4491, Kiev, USSR, 1957.

[7] O. Herrmann and W. Schuessler, "Design of Nonrecursive Digital Filters with Minimum Phase," *Electron. Lett.*, Vol. 6, No. 11, pp. 329-330, 1970.

[8] X. C. Chen and T. W. Parks, "Design of Optimal Minimum Phase FIR Filters by Direct Factorization," *Signal Processing*, Vol. 10, No. 4, pp. 369-383, June 1986.

[9] R. Boite and H. Leich, "A New Procedure for the Design of High Order Minimum-Phase FIR Digital or CCD Filters," *Signal Processing*, Vol. 3, No. 2, pp. 101-108, April 1981.

[10] G. A. Mian and A. P. Nainer, "A Fast Procedure to Design Equiripple Minimum-Phase FIR Filters," *IEEE Trans. Circuit and Syst.*, Vol. CAS-29, No. 5, pp. 327-331, May 1982.

[11] Y. Kamp and C. J. Wellkens, "Optimal Design of Minimum-Phase FIR Filters," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. ASSP-31, pp. 922-926, August 1983.

[12] N. N. Chit and J. S. Mason, "Design of Minimum Phase FIR Digital Filters," *IEE Proceedings*, Vol. 135, Pt. G, No. 6, pp. 258-264, December 1988.

[13] W. H. Press and B. P. Flannery and S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, New York, 1990.

[14] P. T. P. Tang, "A fast algorithm for Linear Complex Chebychev Approximations," *Mathematics of Computation*, 51(184), pp. 721-739, October 1988.

[15] X. Chen and T. W. Parks, "Design of FIR Filters in the Complex Domain," *IEEE Transactions on ASSP*, Vol. ASSP-35, No. 2, pp. 144-153, February 1987.

[16] K. Glashoff and K. Roleff, "A New Method for Chebychev Approximation of Complex-valued Functions," *Mathematics of Computation*, 36(153), pp. 233-239, January 1981.

[17] R. L. Streit and A. H. Nuttall, "A General Chebychev Complex Function Approximation Procedure and an Application to Beamforming," *Journal of Acoustical Society of America*, 72, pp. 181-190, July 1982.

[18] I. Barrodale and C. Phillips, "Solution of an Overdetermined System of Linear Equations in the Chebychev Norm" (Algorithm 495), *ACM Trans. Math. Software*, Vol. 1, pp. 264-270, 1975.

[19] K. Preuss, "On the Design of FIR Filters by Complex Chebychev Approximation," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. ASSP-37, No. 5, pp. 702-712, May 1989.

[20] K. Preuss, "A Novel Approach for Complex Chebychev Approximation with FIR Filters Using the Remez Exchange Algorithm," *Proc. IEEE ICASSP*, pp. 872-875, 1987.

[21] M. Schulist, "Improvements of a Complex FIR Filter Design Algorithm," *Signal Processing*, Vol. 20, No. 1, pp. 81-90, May 1990.

[22] N. N. Chit and J. S. Mason, "Complex Chebychev Approximation for FIR Digital Filters," *IEEE Transactions on Signal Processing*, Vol. 39, No. 1, pp. 49-54, January 1991.

[23] J. S. Mason and N. N. Chit, "New Approach to the Design of FIR Digital Filters," *IEE Proceedings*, Vol. 134. Pt. G, No. 4, pp. 167-180, August 1987.

[24] A. G. Holt and J. Attikiouzel and R. Bennett, "Iterative Technique for Designing Non-recursive Digital Filter Non-linear Phase Characteristics," *Radio Electr. Eng.*, Vol. 46, No. 12, pp. 589-592, December 1976.

[25] L. G. Cuthbert, "Optimizing Non-recursive Digital Filters to Non-linear Phase Characteristics," *Radio Electr. Eng.*, Vol. 44, No. 12, pp. 645-651, December 1974.

[26] G. Cortelazzo and M. R. Lightner, "Simultaneous Design in Both Magnitude and Group-Delay of IIR and FIR filters: Problems and Results," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. 1, pp. 201-204, April 1983.

[27] G. Cortelazzo and M. R. Lightner, "Simultaneous Design in Both Magnitude and Group- Delay of IIR and FIR Filters Based on Multiple Criterion Optimization," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. ASSP-32, No. 5, pp. 949-967, October 1984.

[28] P. T. P. Tang, *Chebychev Approximation on the Complex Plane*, PhD Thesis, Department of Mathematics, University of California at Berkeley, May 1987.

[29] G. Meinardus, *Approximations of Functions: Theory and Numerical Methods*, (translated by L. L. Schumaker), Springer-Verlag, New York, 1967.

[30] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, 1981.

[31] E. W. Cheney, *Introduction to Approximation Theory*, Chelsea, New York, 1986.

[32] G. G. Lorentz, *Approximation of Functions*, Chelsea, New York, 1986.

[33] Personal Communication with Peter Tang.

[34] P. T. P. Tang, "A fast Algorithm for Linear Complex Chebychev Approximation," in *Algorithms for Approximation II*, Edited by J. C. Mason and M. G. Cox, Chapman and Hall Mathematics, New York, 1988.

[35] P. R. Brent, *Algorithms for Minimization without Derivatives*, Prentice Hall, Englewood Cliffs, NJ, 1973.

[36] Dongarra J. J. *LINPACK: user's guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[37] DSP Committee, IEEE ASSP Eds., *Programs for Digital Signal Processing*, IEEE Press, New York, 1979.

[38] K. Glashoff and S. Gustafson, *Linear Optimization and Approximation*, Springer-Verlag, New York, 1983.

## ADDITIONAL BIBLIOGRAPHY


1. K. Steiglitz, T. W. Parks, and J. F. Kaiser, "METEOR: A Constraint-Based FIR Filter Design Program," *Signal Processing,* Vol. 40, No. 8, pp. 1901-1909, August 1992.

2. J. K. Liang and R. J. P. Defigueiredo, "A Design Algorithm for Optimal Low-pass Nonlinear Phase FIR Digital Filters," *Signal Processing,* Vol. 8, No. 1, pp. 3-21, February 1985.

3. S. Ebert and U. Heute, "Accelerated Design of Linear or Minimum Phase FIR Filters with a Chebychev Magnitude Response," *IEE Proceedings,* Vol. 130, Pt. G, No. 6, pp. 267-270, December 1983.

5. S. Ebert and U. Heute, "Accelerated Design of Linear or Minimum Phase FIR Filters with a Chebychev Magnitude Response," *IEE Proceedings,* Vol. 130, Pt. G, No. 6, pp. 267-270, December 1983.

6. G. F. Boudreaux and T. W. Parks, "Thinning Digital Filters: A Piecewise-Exponential Approximation Approach," *IEEE Transactions on ASSP,* Vol. ASSP-31, No. 1, pp. 105-113, February 1983.

7. E. Goldberg and R. Kurshan and D. Malah, "Design of Finite Impulse Response Digital Filters with Nonlinear Phase Response," *IEEE Transactions on ASSP,* Vol. ASSP-29, No. 5, pp. 1003-1010, October 1981.

8. K. Steiglitz, "Design of FIR Digital Phase Networks," *IEEE Transactions on ASSP,* Vol. ASSP-29, No. 2, pp.171-176, April 1981.

9. C. E. Schmidt and L. R. Rabiner, " A study of Techniques for Finding the Zeros of Linear Phase FIR Digital Filters," *IEEE Transactions on ASSP,* Vol. ASSP-25, pp. 96-98, February 1977.

10. L. R. Rabiner J. H. McClellan and T. W. Parks, "FIR Digital Filter Design Techniques Using Weighted Chebychev Approximation," *Proc. IEEE,* Vol. 63, No. 4, pp. 595-610, April 1975.

11. J. H. McClellan and T. W. Parks, "A Unified Approach to the Design of Optimum FIR Linear Phase Digital Filters," *IEEE Trans. on Circuit Theory,* Vol. CT-20, pp. 697-701, November 1973.

12. L. R. Rabiner, "Approximate Design Relationships for Lowpass FIR Digital Filters," *IEEE Trans. Audio Electroacoust.*, Vol. AU-21, pp. 456-460, October 1973.

13. O. Herrmann and L. R. Rabiner and D. S. K. Chan, "Practical Design Rules for Optimum Finite Impulse Response Lowpass Digital Filters," *Bell Syst. Tech. J.*, Vol. 52, pp. 769-799, July-Aug. 1973.

14. T. W. Parks L. R. Rabiner and J. H. McClellan, "On the Transition Width of Finite Impulse Response Digital Filters," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, pp. 1-4, February 1973.

15. T. W. Parks and J. H. McClellan, "Chebychev Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. on Circuit Theory,* Vol. CT-19, pp. 189-194, March 1972.

16. O. Herrmann, "Design of Nonrecursive Digital Filters with Linear Phase," *Electron. Lett.*, Vol. 6, No. 11, pp. 328-329, 1970.

# APPENDIX A. REAL REMEZ EXCHANGE ALGORITHM AND LINEAR-PHASE FIR FILTER DESIGN

In this appendix we review the design of linear-phase FIR filters using the Parks-McClellan algorithm. This method is the most well-known digital FIR filter design method in use today [1] [5]. Linear-phase FIR filters appeal to many firmware designers because, 1) they have constant group delay and, 2) efficient design. The development of the real Remez Exchange algorithm is based on the powerful Alternation theorem, which we examine later. The material presented here is important in understanding 1) the design of linear-phase filters, 2) the design of minimum-phase, and 3) the design of FIR filters in the complex domain. The design of minimum phase filters requires a prototype linear-phase filter designed by this method. Linear-phase filters are a special case of the design of filters in the complex domain. Also, since this method has been the best method to design linear-phase filters, results of new filter design methods are compared to results obtained by the Parks-McClellan method. Therefore, a solid understanding of this material is essential.

## FIR Filters

The design of FIR filters received considerable attention in the early 70's. The early efforts focused on the design of FIR filters that have linear phase i.e., constant group delay.

There are two popular ways currently used to design linear-phase filters. The first uses linear programming and is described by Rabiner [4]. The second method minimizes a weighted error function between a given, usually ideal response, and a sum of real approximating functions, using the minimax criterion and the well-known Remez Exchange algorithm [6]. This method was developed by Parks and McClellan [5] as an algorithm and a computer design program written in Fortran. The linear programming method gives good results but its convergence is slow when compared to the Remez Exchange algorithm.

To avoid delay distortion in any filter, the group delay must be constant [1]. Since the group delay is defined as the frequency derivative of the phase response, constant group delay requires linear phase. The group delay of a linear-phase filter is directly dependent on the filter length as we will see shortly. FIR filters with sharp cutoff and linear phase require a large length, which makes group delay too large for some applications.

The group delay can be made shorter using a minimum-phase filter design. These filters give less delay than linear-phase filters for the same length, but the delay is not constant for all frequencies. In general, minimum-phase filters achieve the same magnitude specifications as the linear-phase filters but with fewer coefficients at the expense of phase distortion. The design of minimum-phase FIR filters is the subject of Part I.

A more powerful method is the design of filters in the complex domain. The method relaxes the symmetry of the filter impulse response and the conjugate symmetry of the frequency response. The approximation to a desired frequency response is a problem of complex approximation. The design of complex filters is discussed in Part II.

# Linear-Phase Filters

## Problem Formulation

The design of optimal equiripple linear-phase FIR filters is a real approximation problem of an ideal amplitude response with a linear combination of functions in such a way that the Chebychev criterion is satisfied. The Chebychev approximation is a minimax technique that minimizes the maximum value of the error function between the desired, usually ideal, amplitude response and the approximating function. Practically, the design problem translates to determining the impulse response coefficients of the optimal filter (best approximation) for a given set of filter specifications.

For a causal FIR filter of length $N$, the impulse response $\boldsymbol{h} = \{h_0, \ldots, h_{N-1}\}$ is finite, and nonzero only over the interval $0 \le k \le N-1$. An FIR filter can also be described by the transfer function, given by the z-transform of $\boldsymbol{h}$,

$$H(z) = \sum_{k=0}^{N-1} h_k \, z^{-k} \tag{A.1}$$

and the frequency response which is the Fourier Transform of $\boldsymbol{h}$,

$$H(f) = \sum_{k=0}^{N-1} h_k \, e^{-j2\pi fk} \tag{A.2}$$

The linear-phase FIR filter requires a phase that is a linear function of the frequency. The frequency response can be written in the form

$$H(f) = G(f) \, e^{j(\beta - \alpha 2\pi f)} \tag{A.3}$$

where $G(f)$ is the real-valued continuous amplitude function, and $\alpha$ and $\beta$ are real constants. This description of the frequency response gives the general form of a linear-phase FIR filter. It can be shown [1] that the only solutions for the constants $\alpha$ and $\beta$ are

$$i) \quad \alpha = \frac{N-1}{2}, \ \beta = 0, \qquad ii) \quad \alpha = \frac{N-1}{2}, \ \beta = \frac{\pi}{2} \tag{A.4}$$

When $\beta = 0$, the impulse response $\boldsymbol{h}$ is *symmetric* about its midpoint, i.e. $h_k = h_{N-1-k}$ for $k=0,1,...,N-1$. When $\beta = \frac{\pi}{2}$, $\boldsymbol{h}$ is *anti-symmetric*, i.e. $h_k = -h_{N-1-k}$. Since $N$ can be even or odd, and $\boldsymbol{h}$ can be symmetric or anti-symmetric, linear-phase FIR filters can be classified into four categories. These are:

(1) $N$ odd, $\boldsymbol{h}$ symmetric

(2) $N$ even, $\boldsymbol{h}$ symmetric

(3) $N$ odd, $\boldsymbol{h}$ anti-symmetric, and

(4) $N$ even, $\boldsymbol{h}$ anti-symmetric.

Cases (2), (3), and (4) can be reduced to case (1) using simple trigonometric identities. Here, we only discuss case (1).

When the symmetry $h_k = h_{N-1-k}$ is used, the amplitude function of Equation (A.3) can be written as a linear combination of cosine functions, that is,

$$G(f) = \sum_{k=0}^{\frac{N-1}{2}} a_k \cos 2\pi f k \tag{A.5}$$

where,

$$a_0 = h_{\frac{N-1}{2}} \quad , \quad a_k = 2\, h_{\frac{N-1}{2}-k} \qquad k=1 \, , \, \ldots \, , \, \frac{N-1}{2} \tag{A.6}$$

The coefficients $a_k$ are directly related to the impulse response of the filter and are used for notational convenience.

Let us pause here and discuss the importance of these results. The frequency response of a realizable FIR filter is always a complex-valued function. Using the linear-phase constraint and the impulse response symmetry, we are able to separate the complex function into two terms. A real function describing the amplitude response of the filter, and an exponential term relating to the phase response. The importance of this analysis is that the amplitude function is real and continuous. Note that since the frequency response of an FIR filter is a complex-valued function, it can also be put in a product form of its magnitude times an exponential. The difference between the two forms is that the magnitude and phase functions in the latter form are not continuous functions, while the amplitude and its

associated phase are continuous functions. Another point to be emphasized is that by using the symmetry of the impulse response, the complex approximation problem is reduced to a real approximation problem. It is this property of linear-phase filters that makes their design fairly easy and attractive.

To develop the approximation criterion, we define the real-valued error function between a desired response and the approximating FIR filter amplitude as

$$E(f) = W(f)\,[D(f) - G(f)] \tag{A.7}$$

where $W(f)$ is a real positive weight function used to control the relative error in the different bands. The real function $D(f)$ represents the desired amplitude response to be approximated. For the design of a lowpass filter, a desired response is usually defined to be the response of an ideal lowpass filter, given by

$$D(f) = \begin{cases} 1\,, & f \in B_p \\ 0\,, & f \in B_s \end{cases} \tag{A.8}$$

where $B_p$, and $B_s$ denote the passband and stopband respectively. As seen, the transition band is not specified. The domain of approximation is a finite set of disjoint bands in the normalized interval [0,0.5], and will be denoted by $F$. Other filters can be specified similarly.

The linear-phase FIR filter design with equiripple amplitude response can be put in the framework of the general Chebychev approximation problem, which can be stated as follows: Find the coefficients $a(k) \in \mathbf{R}$ that minimize $\|E(f)\|$ over $F$, where

$$\|E(f)\| = \max |E(f)| \qquad f \in F \tag{A.9}$$

In the following section we describe the powerful Alternation theorem which is the first step towards the approximation algorithm. Based on this theorem, the real Remez Exchange algorithm is developed, which is the basis of the Parks-McClellan program.

**Alternation Theorem**

The Alternation theorem is powerful because it guarantees a unique best solution to the approximation problem, and also gives important characteristics of the approximating function that make the determination of the best solution possible.

*Alternation Theorem:* If $G(f)$ is a linear combination of $r$ cosine functions, i.e.

$$G(f) = \sum_{k=0}^{r-1} a_k \cos 2\pi f k \tag{A.10}$$

then, the *necessary* and *sufficient* condition that $G(f)$ be the *unique* best weighted Chebychev approximation to a continuous function $D(f)$ on $F$, is that the weighted error function $E(f)$ exhibit at least $r+1$ extremal frequencies on $F$, i.e. there must exist $r+1$ points $f_i$ $i=0,...,r$ on $F$ such that

$$f_0 < f_2 < \ldots < f_{r-1} < f_r \tag{A.11}$$

and such that

$$E(f_m) = -E(f_{m+1}) = \|E(f)\| \qquad m = 0, 1, \ldots, r-1 \tag{A.12}$$

where

$$\|E(f)\| = \underset{f \in F}{\text{maximum}} \; |E(f)| \tag{A.13}$$

The extremal frequencies are the points in $F$ where the error function attains a maximum value. The absolute values of the optimum error function at the extremal frequencies are equal, resulting in an equiripple filter amplitude response. The theorem implies that the best Chebychev approximation must necessarily have an equiripple error function. The minus sign in Equation (A.12) implies that the maximum deviations of the error function will have alternating signs on the successive extremal frequencies.

From the Alternation theorem we get the necessary and sufficient conditions on the weighted error function such that the solution is the unique best approximation to the desired amplitude response $D(f)$. Note that the number of extremal frequencies is at least $r+1$, while there are cases that this number is greater than $r+1$ for the optimum approximation. But the optimum solution is unique, implying that no two solutions with different number of extremal frequencies will both be optimum for the same approximation problem.

The Alternation theorem guarantees a unique solution and also gives the characteristics of the best approximation, but it does not suggest a way to determine the extremal frequencies. If the location of the extremal frequencies was known for the optimum solution, the filter could be derived by simple interpolation on these points, as suggested by the frequency sampling method. The Remez Exchange algorithm is an efficient method that locates the extremal frequencies which define the best approximation.

**Remez Exchange Algorithm**

The Remez Exchange algorithm is an efficient method to find the unique set of extremal frequencies which will make the error function exhibit an equiripple response with minimum maximum deviation. The algorithm is an iterative procedure, and proceeds as follows. At the beginning, a set of $r+1$ extremal frequencies is chosen ($r$ is the number of cosine functions used in the approximation). It is known by the alternation theorem that the error function of the best approximation attains the same maximum value at the extremal frequencies with alternating sign. Let us denote this value by $\delta$. We can write $r+1$ equations involving the error function at the extremal frequencies as

$$W(f_m)\left[D(f_m) - G(f_m)\right] = (-1)^m \delta \qquad m = 0, 1, \ldots, r \qquad \text{(A.14)}$$

The alternating sign on the right-hand side accounts for the maximum and minimum values of the error function. It is important to note here that we do not yet know the position of the extremal frequencies nor the value of $\delta$. The things we know at this point are 1) the number of the extremal frequencies *(r+1)*, and 2) the magnitude of the optimal error function attains the same maximum values at the extremal frequencies.

The approximating amplitude response of the FIR filter, $G(f)$, can be substituted in Equation (A.14) and the result is

$$W(f_m)\left[D(f_m) - \sum_{k=0}^{r-1} a_k \cos 2\pi f_m k\right] = (-1)^m \delta \qquad m=1,\ldots, r \tag{A.15}$$

These equations can be written in matrix form. The matrix is of order *(r+1) x (r+1)*, and the vector of unknowns contains the coefficients $a_k$, *k=0,...,r-1*, and the maximum value of the error $\delta$. The right-hand side contains the values of the desired response at the extremal frequencies. The matrix equation is given by (A.16).

The *(r+1)x(r+1)* matrix contains cosine functions of the extremal frequencies. Note that we do not yet know those extremal points. So, in essence we have $r$ unknown coefficients $a_k$ related to the impulse response of the filter, the maximum absolute value of the error $\delta$, and *r+1* frequencies where the error is maximum. The basic idea behind the Remez algorithm is to find this set of extremal frequencies. This is done iteratively by initially selecting an arbitrary set of frequencies and solving the Equation (A.16). If the solution is not optimum, a different set of frequencies is chosen.

$$\begin{bmatrix} 1 & \cos 2\pi f_0 & \cdots & \cos(r-1)2\pi f_0 & \frac{1}{W(f_0)} \\ 1 & \cos 2\pi f_1 & \cdots & \cos(r-1)2\pi f_1 & \frac{1}{W(f_1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 1 & \cos 2\pi f_r & \cdots & \cos(r-1)2\pi f_r & \frac{(-1)^r}{W(f_r)} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{r-1} \\ \delta \end{bmatrix} = \begin{bmatrix} D(f_0) \\ D(f_1) \\ D(f_2) \\ \vdots \\ D(f_{r-1}) \\ D(f_r) \end{bmatrix} \tag{A.16}$$

The beauty of the Remez algorithm is its ability to choose the new set of frequencies in such a way that the optimum solution is reached rapidly. The matrix Equation (A.16) can be quite complicated and large for long filters. Its direct solution for a trial set of extremal frequencies is both slow and difficult. It is also unnecessary in some sense, because the coefficients $a_k$ relating to the impulse response are not needed until the last iteration is performed. A more efficient way to solve the problem is to solve for $\delta$ analytically. It can be shown [5] that $\delta$ is given by

$$\delta = \frac{\alpha_0 D(f_0) + \alpha_1 D(f_1) + \cdots + \alpha_r D(f_r)}{\frac{\alpha_0}{W(f_0)} - \frac{\alpha_1}{W(f_1)} + \cdots + (-1)^r \frac{\alpha_r}{W(f_r)}} \tag{A.17}$$

where

$$\alpha_k = \prod_{\substack{i=0 \\ i \neq k}}^{r} \frac{1}{(x_k - x_i)} \qquad x_i = \cos 2\pi f_i \qquad \text{(A.18)}$$

The algorithm starts with a guessed initial set of extremal frequencies. The Parks-McClellan program gives a starting initial set of $r+1$ equally spaced frequencies in the intervals of approximation. After $\delta$ is calculated, the Langrange formula in the barycentric form [5] is used to interpolate $G(f)$ on the $r$ points $f_k$, $k=0, \ldots, r$. The interpolated $G(f)$ is required in the estimation of the error function.

The next step in the algorithm is to evaluate the error $E(f)$ on a dense grid of the frequency axis. The error function is given by

$$E(f) = W(f)\,[D(f) - G(f)] \qquad \text{(A.22)}$$

Note that the error is a piecewise continuous function. Calculation of the error on a dense grid of frequencies though is sufficient to determine the maximum and minimum values of the error.

Next, the maximum values of the absolute error are compared with $\delta$ calculated from Equation (A.17). If $|E(f)| \leq \delta$ for all the frequencies on the grid, then the best solution is found. If $|E(f)| > \delta$ at any frequency, the best solution of the approximation has not yet been found and another set of extremal frequencies should be chosen. The new $r+1$ extremal frequencies for the next iteration are chosen to be the frequencies where the error function has maximum and minimum values. The replacement of the old set of frequencies with the

new set increases the value of $\delta$, and at some point converges to its upper bound. In case there are more than $r+1$ peaks in $E(f)$, only the first $r+1$ points that $|E(f)|$ is largest are kept for the next iteration.

The new set of frequencies is used to solve for a new value $\delta$ and error function. The new value of $\delta$ will be larger than the one from the previous iteration. This procedure repeats until the best approximation is found, based on the criteria discussed above. The error of the best approximation will have an equiripple behavior as dictated by the Alternation theorem. Note that the maximum of the absolute value of the error at an intermediate iteration is always larger than the value of $\delta$. At every new iteration the value of $\delta$ increases while the value of the absolute error decreases. The optimum solution is found when the two quantities meet. At that point the maximum absolute error attains its minimum value and thus the minimax criterion is met.

The operations described above are the basis of the real Remez Exchange algorithm. At an intermediate iteration of the algorithm the maximum values of the error are not all equal if the best approximation has not been reached. The alternation theorem assures that when the best approximation has been found, the error will have the same absolute maximum values. The frequency points these maximum values occur will correspond to the *unique optimum* extremal frequency set. In addition to the uniqueness of the approximation, the alternation guarantees a solution.

## Conclusion

We discussed the design of linear-phase FIR filters using the Remez Exchange algorithm. The design problem is translated into a real approximation problem using the linearity condition of the phase function. The Alternation theorem is the basis to the derivation of the Remez Exchange algorithm. The Remez algorithm is an efficient iterative procedure that locates the optimal set of frequencies that characterize the error function of the best approximation. When this set of points is found, the best approximation in the Chebychev sense is fully defined.

# APPENDIX B. LINEAR OPTIMIZATION PROBLEM

In this appendix we discuss some of the basics of the linear optimization problem. This material is essential in understanding the derivation of the dual of the primal complex approximation problem presented in Part II of this dissertation. First we present the general optimization problem. In this presentation we use notation that fits our problem of complex approximation rather than traditional notation. After the primal problem is presented, the dual equivalent problem is derived. The relation between the primal and dual problems and the importance of this transformation are discussed.

## Primal Linear Optimization Problem

Define the vectors $\beta \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, and an index set $F$, which might be infinite or finite. Associate with each $f \in F$ a vector $a(f) \in \mathbb{R}^n$, and a real number $q(f)$. The preference function is defined as

$$g(y) = \beta^T \cdot y = \sum_{l=1}^{n} \beta_l y_l \tag{B.1}$$

The primal linear optimization problem is the following. Given the vectors $\beta \in \mathbf{R}^n, a(f) \in \mathbf{R}^n$ and the numbers $q(f)$ for $f \in F$, find a vector $y' \in \mathbf{R}^n$ which solves the following minimization problem:

$$\underset{f \in F}{\textit{minimize}} \quad \beta^T \cdot y \tag{B.2}$$

subject to the constraint

$$a^T(f) \cdot y \geq q(f) \tag{B.3}$$

Often, this problem is called a *semi-infinite* problem since there is a finite number of variables to be determined (the vector $y'$), and an infinite number of constraints given by equation (B.3). When the index set $F$ is finite, the general linear optimization problem reduces to the known *linear programming problem.* Since there is a finite number of elements in the index set $F = \{f_1, \ldots, f_m\}$, the constraint equation can be written in matrix form. This is possible since there are only $m$ different vectors $a$ in the constraint equation (B.3). The matrix constraint equation in the linear programming case is given by

$$A^T \cdot y \geq q \tag{B.4}$$

The general optimization problem can be considered to be a linear program with a matrix $A$ that has an infinite number of columns and a finite number of rows.

One method to solve the minimization problem is by *discretization* of the semi-infinite problem. This method requires the selection of a finite set of points in the index set $F$, and derivation of the corresponding linear program. The resulting linear program is called *Discretization of the Linear Optimization Problem (LOP).* This is an approximate solution to the original problem with satisfactory results in most cases. This method has been used by Chen and Parks [15] to solve the filter design problem in the complex domain. Another approach to solve the problem is to look at its dual which we discuss next. The derivation of the dual provides a way to solve the minimization problem without using an approximation.

## Dual Problem

Let us denote the value of the preference function in the primal problem by $v(P)$. Then an upper bound of $v(P)$ is

$$v(P) \leq \beta^T \cdot y \tag{B.5}$$

when a feasible vector $y$ is found. A feasible vector $y$ is one that satisfies the constraints. The following lemma is important in the derivation of the dual problem.

**Duality Lemma:**[38] Let a finite subset $\{f_1, \ldots, f_K\} \subset F$ where $K \geq 1$, and a nonnegative set of numbers $r_1, \ldots, r_K$ such that

$$\beta = r_1\, a(f_1) + \ldots + r_K\, a(f_K) \tag{B.6}$$

Then the following inequality holds for every feasible vector $y$:

$$r_1\, q(f_1) + r_2\, q(f_2) + \ldots + r_K\, q(f_K) \le \beta^T \cdot y \tag{B.7}$$

The above lemma gives a lower bound for the primal problem which is

$$\sum_{k=1}^{K} r_k\, q(f_k) \le v(P) \tag{B.8}$$

*Proof:*

Since we assumed that the vector $y$ is a feasible solution then

$$a^T(f_k)\, y \ge q(f_k) \qquad k = 1, \ldots, K \tag{B.9}$$

Since $r_k \ge 0$ for $k = 1, \ldots, K$, then

$$\sum_{k=1}^{K} r_k\, q(f_k) \le \sum_{k=1}^{K} r_k\, [a^T(f_k) \cdot y] = \left( \sum_{k=1}^{K} r_k\, a^T(f_k) \right) \cdot y = \beta^T \cdot y \tag{B.10}$$

The following lemma describes the optimum solution:

**Lemma:**[38] Let $y = (y_1, \ldots, y_n)$ be feasible for the primal problem. Assume also that the subset $\{f_1, \ldots, f_K\} \subset F$ and the nonnegative numbers $r_1, \ldots, r_K$ satisfy Equation (B.6). If the relation

$$\sum_{k=1}^{K} r_k \, q(f_k) = \sum_{l=1}^{n} \beta_l \, y_l \tag{B.11}$$

is satisfied, then the vector $y$ is an optimal solution of the primal problem. The lower and upper bounds of the value of the primal problem meet. Thus, our goal is to choose the subset $\{f_1, \ldots, f_K\} \subset F$, and the nonnegative numbers $r_1, \ldots, r_K$ such that the lower bound obtained by the duality lemma is maximized. At this point we are ready to give the dual problem.

**Dual Problem:** Determine a finite subset $\{f_1, \ldots, f_K\} \subset F$ and real numbers $r_1, \ldots, r_K$ to maximize $\sum_{k=1}^{K} q(f_k) \, r_k$, subject to the constraints

$$\sum_{k=1}^{K} r_k \, a(f_k) = \beta_l \qquad l = 1, \ldots, n \tag{B.12}$$

and

$$r_k \geq 0 \qquad k = 1, \ldots, K \tag{B.13}$$

The number of points $K$ may be arbitrarily large. In the complex approximation problem it is proved by Tang [14] that the number of points required is $K=N+1$, where $N$ is the order of the approximation.

A more compact form of the dual problem can be obtained using vector notation. Define the vectors

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ f_K \end{bmatrix} \quad r = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_K \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix} \quad q = \begin{bmatrix} q(f_1) \\ q(f_2) \\ \cdot \\ \cdot \\ \cdot \\ q(f_K) \end{bmatrix} \tag{B.14}$$

and the matrix $A = \left[a(f_1) \ldots a(f_K)\right]$. The dual problem can be expressed as follows. Determine the vectors $f$ and $r$ $(r \geq 0)$ to maximize the inner product $q^T \cdot r$, subject to the constraint

$$A \cdot r = \beta \tag{B.15}$$

The relation between the primal and dual problems is that the *minimized* value of the primal problem is the same as the *maximized* of the dual problem. That is, for the optimal vector $y'$ the relation of Equation (B.11) holds:

$$\sum_{k=1}^{K} r_k \, q(f_k) = \sum_{l=1}^{n} \beta_l \, y_l \tag{B.16}$$

## Conclusion

We have examined the basics of the general optimization problem. A general optimization problem is usually formulated into the primal problem. This problem involves a preference function to be minimized subject to constraints. If the number of constraints is infinite, the problem is called a semi-infinite optimization problem. When the number of constraints is finite, the problem is a linear programming problem. A linear program can be solved using the well-known Simplex method.

The solution of a semi-infinite optimization problem can be approached in two ways. One way is to obtain an approximate solution by deriving the corresponding linear program. This can be done by discretization of the original problem. A more efficient method is to derive the dual problem. In the dual problem, the infinite number of constraints of the primal problem become finite. This simplifies the solution. More importantly, the solution of the dual does not involve any discretization, and therefore the solution is not an approximation.

# APPENDIX C. TRANSFER FUNCTION, PHASE, AND DELAY

In this appendix we discuss some of the functions used to describe FIR filters, such as the transfer function, magnitude response, and phase and group delay functions. In most applications, emphasis is given on the magnitude response of the filter while the phase and delay functions are of less importance. This has been the case until recently since usually linear-phase filters have been designed.

Recently, several digital filter design techniques, besides the one presented in this document, have been developed to allow non-linearities in the phase. The main reasons for the development of these techniques are 1) reduction of filter length for the same magnitude performance, 2) reduction of the nominal group delay value by allowing small non-linearities in the filter phase function, and most importantly 3) arbitrary specification of magnitude and phase or group delay functions. In this appendix we give definitions of these functions.

An FIR filter can be described by its *z-transform*, given by

$$H(z) = \sum_{k=0}^{N-1} h_k \, z^{-k} \qquad \text{(C.1)}$$

where $N$ is the filter length, and $h_k$ is the filter impulse response sequence. The transfer function is a complex function of the complex variable $z$, in negative powers of $z$. Note also that the impulse response is not restricted to be a real sequence.

The *frequency response* of the filter is the z-transform evaluated on the unit circle, i.e.

$$H(f) = H(z)_{|z=e^{j2\pi f}} = \sum_{k=0}^{N-1} h_k \, e^{-j2\pi fk} \qquad \text{(C.2)}$$

where $f$ is the normalized frequency in the interval [-0.5,0.5]. The frequency response is a complex-valued function of the normalized frequency, and it is periodic in $f$ with period 1. Therefore, specification of the frequency response for one period, in the interval [-0.5,0.5], is sufficient to describe the response for all frequencies.

Since the frequency response is a complex-valued function, it can be expressed in the form of its real and imaginary parts, or its magnitude and phase functions. The latter is preferred in most cases because filter specifications and performance are often given in the form of magnitude and phase. The frequency response can be written in the form

$$H(f) = |H(f)| \, e^{\phi(f)} \qquad \text{(C.3)}$$

where $\phi(f)$ is the phase function of the filter.

The transfer function of an FIR filter, with length $N$, can be written as

$$H(z) = \sum_{k=0}^{N-1} h_k \, z^{-k} = \frac{1}{z^{N-1}} \sum_{k=0}^{N-1} h_k \, z^{N-1-k} = \frac{1}{z^{N-1}} \prod_{k=0}^{N-1} (z - z_k) \qquad \text{(C.4)}$$

where $z_k$ are the zeros of the transfer function. If we let $z_k = r_k \, e^{j\theta_k}$, then the frequency response can be determined from the zeros of the transfer function by

$$H(f) = e^{-j2\pi f(N-1)} \left(e^{j2\pi f} - r_1 e^{j\theta_1}\right) \ \ldots \ \left(e^{j2\pi f} - r_{N-1} e^{j\theta_{N-1}}\right) \quad \text{(C.5)}$$

The phase response can also be obtained from the zeros, and is given by

$$\angle H(f) = -2\pi f(N-1) + \sum_{k=0}^{N-1} \angle\left(e^{j2\pi f} - r_k e^{j\theta_k}\right) \quad \text{(C.6)}$$

The group delay is defined as the negative frequency derivative of the phase function, given by

$$\tau(f) = -\frac{1}{2\pi}\frac{d}{df}\angle H(f) = N-1 - \frac{1}{2\pi}\frac{d}{df}\sum_{k=0}^{N-1}\angle\left(e^{-j2\pi f} - r_k e^{j\theta_k}\right) \quad \text{(C.7)}$$

Note that the phase function used above is the continuous phase of the transfer function. The frequency derivative of the *kth* summation term is given by

$$-\frac{1}{2\pi}\frac{d}{df}\angle\left(e^{j2\pi f} - r_k e^{j\theta_k}\right) = \frac{1 - r_k\cos(2\pi f - \theta_k)}{1 + r_k^2 - 2r_k\cos(2\pi f - \theta_k)} \quad \text{(C.8)}$$

and the total group delay of the FIR filter is given by

$$\tau(f) = N-1 - \sum_{k=0}^{N-1} \frac{1 - r_k \cos(2\pi f - \theta_k)}{1 + r_k^2 - 2 r_k \cos(2\pi f - \theta_k)} \qquad \text{(C.9)}$$

It can also be proved that the group delay is given by

$$\tau(f) = \sum_{k=0}^{N-1} \frac{r_k^2 - r_k \cos(2\pi f - \theta_k)}{1 + r_k^2 - 2 r_k \cos(2\pi f - \theta_k)} \qquad \text{(C.10)}$$

This expression can be obtained with the same method described above, by distributing the exponential term to the remaining factors in Equation (C.5). In the derivation of the group delay we used the continuous phase. One can also use the principal argument of the phase $ARG\,[H(f)]$, which will give the same result except at the discontinuities of $ARG\,[H(f)]$ at $\pi$ and $-\pi$.

Another related quantity is the *phase delay*. The phase delay is defined as

$$\tau_{pd}(f) = -\frac{\angle H(f)}{2\pi f} \qquad \text{(C.11)}$$

The phase delay is equal to the group delay *only* when the phase function is a linear function of frequency without any constant term.