Design of Nearly Linear-Phase Recursive Digital Filters by Constrained Optimization

by

David Guindon
B.Eng., University of Victoria, 2001

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

Masters Of Applied Science

in the Department of Electrical and Computer Engineering

University of Victoria

Design of Nearly Linear-Phase Recursive Digital Filters by Constrained Optimization

by

David Guindon
B.Eng., University of Victoria, 2001

Supervisory Committee

Dr. Andreas Antoniou, Supervisor (Dept. of Elec. and Comp. Engr.)

Dr. Dale J. Shpak, Co-Supervisor (Dept. of Elec. and Comp. Engr.)

Dr. Aaron Gulliver, Departmental Member (Dept. of Elec. and Comp. Engr.)

Dr. Sadik Dost, Outside Member (Dept. of Mechanical Engr.)

Supervisory Committee

Dr. Andreas Antoniou, Supervisor (Dept. of Elec. and Comp. Engr.)

Dr. Dale J. Shpak, Co-Supervisor (Dept. of Elec. and Comp. Engr.)

Dr. Aaron Gulliver, Departmental Member (Dept. of Elec. and Comp. Engr.)

Dr. Sadik Dost, Outside Member (Dept. of Mechanical Engr.)

Abstract

The design of nearly linear-phase recursive digital filters using constrained optimization is investigated. The design technique proposed is expected to be useful in applications where both magnitude and phase response specifications need to be satisfied. The overall constrained optimization method is formulated as a quadratic programming problem based on Newton's method. The objective function, its gradient vector and Hessian matrix as well as a set of linear constraints are derived. In this analysis, the independent variables are assumed to be the transfer function coefficients. The filter stability issue and convergence efficiency, as well as a 'real axis attraction' problem are solved by integrating the corresponding bounds into the linear constraints of the optimization method. Also, two initialization techniques for providing efficient starting points for the optimization are investigated and the relation between the zero and pole positions and the group delay are examined. Based on these ideas, a new objective function is formulated in terms of the zeros and poles of the transfer function expressed in polar form and integrated into the optimization process. The coefficient-based and polar-based objective functions are tested and compared and it is shown that designs using the polar-based objective function produce improved results. Finally, several other modern methods for the design of nearly linear-phase recursive filters are compared with the proposed method. These include an elliptic design combined with an

optimal equalization technique that uses a prescribed group delay, an optimal design method with robust stability using conic-quadratic-programming updates, and an unconstrained optimization technique that uses parameterization to guarantee filter stability. It was found that the proposed method generates similar or improved results in all comparative examples suggesting that the new method is an attractive alternative for linear-phase recursive filters of orders up to about 30.

Contents

List of Figures

List of Tables

Abbreviations

ATT	arc-tangent transformation
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BMT	balanced model truncation
CPU	central processing unit
CQP	conic-quadratic-programming
FIR	finite-duration impulse response
HTT	hyperbolic tangent transformation
IIR	infinite-duration impulse response
MBT	modified bilinear transformation
SDP	semidefinite programming

Acknowledgements

I would like to thank my supervisors, Dr. Andreas Antoniou and Dr. Dale Shpak for giving me the opportunity to work with some of the best in the field of digital signal processing. I also thank Dr. Antoniou for the valuable knowledge I have gained in computer programming in Delphi, Latex, and Matlab. This newfound knowledge has allowed me to pursue in the direction I have only dreamed and I do not think I would have found that path without the generous help from Dr. Antoniou.

Next, I would like to express my gratitude to Dr. Wu-Sheng Lu. I do not think I would have been able to produce the results in this thesis without his valuable advice. Furthermore, it was Dr. Lu's course in optimization that helped me formulate the methods proposed in this thesis. In addition, several aspects throughout this thesis were inspired by his publications.

I would also like to thank Vicky Smith, Lynne Barrett, and Catherine Chang, all of whom helped me along the way.

Lastly, I would like to thank my close friends, Ari Knazan and Rob Kliska for their patience and amazing support.

Chapter 1

Introduction

> As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.
>
> —Albert Einstein

Recursive or infinite-duration impulse response (IIR) digital filters have been used in numerous applications such as system identification, channel equalization, signal enhancement, and signal prediction [1]. In applications where moderate selectivity is required, nonrecursive or finite-duration impulse response (FIR) digital filters can easily provide the desired results. Unfortunately, when high selectivity is desired, the required order of an FIR filter can be very high, and in most cases, much higher than that required by a corresponding IIR filter. FIR filters can be easily designed to have a linear phase response. IIR filters with linear phase response are not possible. However, nearly linear-phase responses can be achieved by combining IIR delay equalizers with IIR filters.

In recent years new methods have been proposed that do not use the filter-plus-equalizer approach. Even though the design of equalizers is reasonably efficient, this technique requires adding more filter sections to provide the equalization and hence increases the filter order and ultimately the cost of implementation. Another approach to this problem is to design

for both magnitude and phase response specifications simultaneously. By using constrained optimization, linear constraints can be used to assure filter stability which is a common problem with unconstrained optimization techniques [1][2].

In this thesis, an objective function that can be optimized for both the magnitude and phase response will be developed along with a constrained optimization technique. It is assumed that the phase response is required to be linear only in the passband regions. The delay distortion in the stopband region is not considered since the phase angles of attenuated signal components are unimportant.

In Chapter 2, the objective function is derived in terms of the transfer function coefficients and used in the constrained optimization problem. The constrained optimization algorithm to be used encompasses Newton's method that minimizes the objective function using quadratic programming (QP). Furthermore, using constrained optimization, bounds can be placed on the transfer function coefficients to assure filter stability as well as to facilitate efficient movement toward a feasible solution. The overall method requires the gradient vector and Hessian matrix plus a set of linear constraints. The objective function is based on a weighted least-pth function. Real pole boundary constraints are integrated in the optimization problem to prevent the poles from reaching the real axis. Otherwise, as explained in Chapter 2, a so-called 'real-axis attraction' phenomenon produces undesired results.

Further research into the zero and pole positions related to the group delay is carried out in Chapter 3. This chapter also proposes a new objective function and provides reasons why this new objective function should be pursued. The new objective function is expressed in terms of the zeros and poles of the transfer function in polar representation. The gradient and Hessian are then determined along with a new set of linear constraints. Chapter 3 also deals with two initialization methods for obtaining good initial points for the constrained optimization.

In Chapter 4, the coefficient-based and polar-based objective functions are tested and com-

pared with respect to filter quality of the designed filters and computational cost. To provide a full understanding of how each function performs, several different initial points are used to design a nearly linear-phase lowpass IIR filter.

The proposed method is compared with three other modern design methods that optimize with respect to both the magnitude and phase responses. The three methods are: An elliptic design with an efficient equalization technique that achieves prescribed magnitude response and nearly linear-phase response in passbands, a minimax design with a prescribed stability margin formulated as a conic-quadratic-programming problem (CQP), and an unconstrained optimization method where parameterization is used to guarantee filter stability.

The innovations proposed in this thesis include the new polar-based objective function designed to satisfy prescribed specifications for both the magnitude and phase responses as well as several topics surrounding the optimization technique. More specifically, to ensure efficient convergence several linear constraints are developed, two efficient initialization methods are investigated, a nonuniform sampling technique is developed, and a dynamic step updating scheme is proposed. Additionally, several quality factors are proposed to facilitate comparisons between various filter designs. In particular, separate quality factors are proposed for both the magnitude and phase responses to efficiently monitor the design progress during optimization. Also, modified quality factors are proposed to provide a scale of how close the filter design satisfies the prescribed specifications. These modified quality factors not only provide additional insights during optimization but are also used to control the step limits for efficient convergence.

As shown in Chapter 2, the gradient and Hessian of the objective function and error function are required for the optimization problem. Unexpectedly, the Hessian of the error function turned out not to be block diagonal as it was for the equalizer sections in Ko's thesis [2] and, consequently, several closed-form equations had to be derived in order to construct the Hessian. This is also the case for the polar-based objective function derived in Chapter 3.

Inspired by Ko's thesis [2], a polar-based objective function is derived revealing two immediate advantages. Specifically, the real-axis attraction problem is largely avoided and the filter stability issue is handled with a simple linear constraint. This reduces the number of constraints by about half.

The polar-based objective function is used to obtain several designs using random initial points as well as points generated by two efficient initialization routines. With these initialization routines, it is shown that better designs can be obtained in terms of filter quality and number of iterations. However, improved results are achieved for lower filter order designs of less than 16 in the case where efficient initial points are used for the optimization. On the other hand, the coefficient-based objective function is more robust to random initial points and requires less computation for higher-order designs of orders over 16. For lowpass filters, the polar-based objective function was found to give better designs than the coefficient-based objective function for all of the design examples attempted in this thesis.

In Chapter 5, the methods developed in Chapters 2 and 3 as well as the modifications provided in Chapter 4 are used to design several lowpass filters and the results obtained are compared. The first design technique that is compared with the proposed method is the traditional equalization approach which utilizes an elliptic filter and an optimal equalizer. The proposed method produced a lower-order filter satisfying the required specifications but the equalizer technique required far less computational effort to complete the optimization. More importantly, the proposed method produced a better filter quality while retaining a lower filter order for a more stringent filter design. The second comparison was with a minimax design with a prescribed stability margin formulated as a CQP problem. The proposed method is shown to produce a filter with improved magnitude response as well as an improved filter quality. However, the CQP method requires fewer iterations to complete the design. Finally, the proposed method is compared with the unconstrained optimization method developed by Lu in [3]. For this design example, the proposed method was found to outperform the unconstrained optimization technique for all prescribed specifications demonstrating that

this method can successfully compete with other modern optimization techniques.

In Chapter 6, several issues that are significant factors in producing improved results are suggested. More specifically, a dynamic weighting scheme that can weigh the importance of either the magnitude response or group delay is discussed. Another suggestion is further research into improving the initial points with the BMT method and possibly obtaining a closed-form algorithm requiring no optimization. Additionally, the unconstrained optimization method used in the last comparison of Chapter 5 could be adapted for a polar-based objective function with parametrization along with the BMT initialization routine. Lastly, other filter types as well as odd-order filters can be investigated.

Since this optimization problem has several local minima, it is difficult to determine if any particular solution has converged to the global minimum. Therefore, throughout this thesis the term 'optimal solution' or 'optimal design' refers to an optimized solution that satisfies the prescribed design specifications.

The developed methods along with some additional research to improve a number of design aspects can serve as an attractive alternative approach for the design of lower-order nearly-linear phase IIR lowpass filters of orders up to about 30.

Chapter 2

Design Using Constrained Optimization

> It's a job that's never started that takes the longest to complete.
>
> —J. R. R. Tolkien

2.1 Introduction

In recent years several methods have been proposed for designing nearly linear-phase recursive digital filters. The method in [3] employs the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton unconstrained optimization method using a least-pth objective function to design the IIR filter. An alternative to the least-pth objective function that utilizes an iterative semidefinite programming (SDP) algorithm to design IIR filters with arbitrary magnitude and phase responses is presented in [4]. Like the second method, the techniques and methods developed in this chapter utilize a constrained optimization method.

A quadratic programming constrained optimization method is used to minimize the objective function. The digital filter design is then performed by minimizing an objective function that

represents the difference between the actual and desired frequency responses.

First, the problem is explained and then the constrained optimization method is applied. The method also requires a suitable set of linear constraints to assure filter stability, efficient solution convergence, and to avoid a real-axis attraction phenomenon. Lastly, two efficient initialization techniques are presented.

2.2 Problem Formulation

An nth-order recursive digital filters can be represented by the transfer function

$$H(\mathbf{x}_1, z) = H_0 \prod_{k=1}^{J} \frac{a_{0k} + a_{1k} z + z^2}{b_{0k} + b_{1k} z + z^2} = H_0 \prod_{k=1}^{J} \frac{N_k(z)}{D_k(z)} \tag{2.1}$$

where a_{0k}, a_{1k} and b_{0k}, b_{1k} are real coefficients, $J = n/2$ is the total number of filter sections, n is the filter order, and H_0 is a positive multiplier constant. For optimization, the parameter vector is

$$\mathbf{x}_1 = [a_{01} \quad a_{11} \quad b_{01} \quad b_{11} \quad \cdots \quad b_{0J} \quad b_{1J} \quad H_0]^T. \tag{2.2}$$

The frequency response can be found by substituting $z = e^{j\omega T}$ where ω is the frequency in rad/s and T is the sampling period in seconds.

For the design of a nearly linear-phase IIR digital filter, let $\Omega = \omega_i,\ 1 \leq i \leq M$, be the set of frequencies where the frequency response is evaluated. The error at each point in ω is defined as

$$e_i(\mathbf{x}) = F(\mathbf{x}_1, \omega_i) - F_0(\omega_i) \tag{2.3}$$

where $F(\mathbf{x}_1, \omega_i)$ is the digital filter transfer function evaluated at $z = e^{j\omega T}$ representing the actual frequency response and $F_0(\omega_i)$ represents the desired frequency response.

For linear phase, the desired frequency response is given as

$$F_0(\omega) = M_0(\omega_i)e^{-j\omega\tau} \tag{2.4}$$

where $M_0(\omega_i)$ is the desired magnitude response at ω_i and τ is the group delay. The actual frequency response is

$$F(\mathbf{x}_1, \omega_i) = H_0 \prod_{k=1}^{J} \frac{a_{0k} + a_{1k}e^{j\omega_i T} + e^{j2\omega_i T}}{b_{0k} + b_{1k}e^{j\omega_i T} + e^{j2\omega_i T}} = H_0 \prod_{k=1}^{J} \frac{N_k(e^{j\omega_i T})}{D_k(e^{j\omega_i T})} \tag{2.5}$$

and the parameter vector is updated as

$$\mathbf{x} = [\mathbf{x}_1^T \quad \tau]^T \tag{2.6}$$

to include the group delay.

2.3 Objective Function

The objective function is represented as a weighted least-pth function

$$E(\mathbf{x}) = \sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^p \tag{2.7}$$

where $p > 0$ is an even integer, and $\{w_i, 1 \leq i \leq M\}$ are weights at the frequencies defined by Ω.

2.4 Constrained Optimization Method

The objective function given in Eq. 2.7 is minimized using a quadratic programming method. These methods are used to minimize quadratic objective functions subject to linear constraints. Although Eq. 2.7 is not quadratic, an iterative approach is performed that eventually solves the general programming problem.

The overall constrained optimization method is based on Newton's method. The method finds an initial feasible solution by first solving a linear programming problem. Since the objective function is highly nonlinear, the optimization is broken down into subproblems where at each iteration the solution is approximated to establish a direction of search and used as the initialization for the subsequent iteration.

2.4.1 Newton's Method

Newton's method is used to solve the highly nonlinear problem by solving several smaller quadratic subproblems. First, the method is presented for a general linearly constrained problem of the form

$$\begin{aligned} &\min\ f(\mathbf{x}) \\ &\text{subject to } \mathbf{A}^T\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{2.8}$$

The optimization subproblems require the objective function, its gradient vector and Hessian matrix. Newton's method is a second-order method using a quadratic approximation to the Taylor series [5]. If $\boldsymbol{\delta}$ is a change in $\mathbf{x}$, $f(\mathbf{x}+\boldsymbol{\delta})$ is given by

$$f(\mathbf{x}+\boldsymbol{\delta}) \approx f(\mathbf{x}) + \sum_{j=1}^{n} \frac{\partial f}{\partial x_i}\delta_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}\partial_i\partial_j \tag{2.9}$$

Eq. 2.9 can be represented in matrix form as

$$f(\mathbf{x}+\boldsymbol{\delta}) \approx f(\mathbf{x}) + \boldsymbol{\delta}^T\mathbf{g} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{H}\boldsymbol{\delta} \tag{2.10}$$

where $\mathbf{g}$ and $\mathbf{H}$ are the gradient and Hessian of the objective function, respectively. Eq. 2.10 is used to approximate the solution of the subproblem at the current iteration; therefore, the objective function of the subproblem at the kth iteration is

$$f(\mathbf{x}^{(k)}+\boldsymbol{\delta}) \approx \mathbf{F}^{(k)}(\boldsymbol{\delta}) = f^{(k)} + \boldsymbol{\delta}^T\mathbf{g}^{(k)} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{H}^{(k)}\boldsymbol{\delta} \tag{2.11}$$

where $\mathbf{x}^{(k)}$ is the value of the parameter vector at the kth iteration, $f^{(k)}$ is the objective function evaluated at $\mathbf{x}^{(k)}$, $\mathbf{g}$ and $\mathbf{H}$ are the gradient and Hessian of the objective function, respectively. The value $\boldsymbol{\delta}$ represents the change in $\mathbf{x}^{(k)}$ in the neighborhood of $\mathbf{x}^{(k)}$.

Using Eq. 2.11 in Eq. 2.8, the quadratic programming problem can be rewritten as

$$\begin{aligned} &\text{minimize } F(\boldsymbol{\delta}) = \tfrac{1}{2}\boldsymbol{\delta}^T\mathbf{H}\boldsymbol{\delta} + \boldsymbol{\delta}^T\mathbf{g} \\ &\text{subject to } \mathbf{A}^T\boldsymbol{\delta} \leq \mathbf{b}. \end{aligned} \tag{2.12}$$

The quadratic approximation is then applied to the objective function where the current iteration generates the value of $\mathbf{x}$ to be used as the starting point for the next iteration. The solution is updated as

$$\mathbf{x} = \mathbf{x}^{(k)} + \boldsymbol{\delta} \tag{2.13}$$

and $\boldsymbol{\delta}$ provides a change in $\mathbf{x}$ in a descent direction toward a feasible solution.

Unfortunately, due to the highly nonlinear objective function to which this approximation is applied to, this procedure yields limited accuracy. In addition, as the number of sections are increased, the accuracy tends to decrease [2]. Therefore, another set of constraints is required to ensure that the step size resulting from each iteration is limited.

This additional constraint is a step limit or maximum change in $\boldsymbol{\delta}$ allowed from each suboptimization. This step limit is

$$|\delta_i| \leq \beta \tag{2.14}$$

where δ_i are the values of $\boldsymbol{\delta}$ representing the change in $\mathbf{x}$, and β is a predefined constant.

The step limits will later be modified to satisfy certain conditions that apply to the programming problem. These conditions are explained in more detail in section 2.6.

Newton's method will generate a solution if and only if the Hessian is nonsingular and the approximation in Eq. 2.9 is valid [5]. Since any function $f(x) \in C^2$ can be accurately represented in the neighbourhood of the solution vector x^* by the quadratic approximation of the Taylor series, the solution to Eq. 2.9 exists [5]. Furthermore, in order for the Hessian to be nonsingular the matrix must be positive definite and there is no guarantee that the Hessian will be positive definite. If the Hessian is not positive definite, it can be altered to become

$$\mathbf{H}^* = \mathbf{H} + \beta \mathbf{I}_n \tag{2.15}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\beta \geq -\lambda_n$ if $\lambda_n \leq 0$ or $\beta = 0$ if $\lambda_n > 0$ with λ_n being the minimum eigenvalue of the original Hessian $\mathbf{H}$ [5]. Since $\mathbf{I}$ is positive definite, the problem of a nonsingular $\mathbf{H}^*$ is eliminated. As the optimization problem approaches the solution, the value of β decreases resulting in a more accurate modified Hessian $\mathbf{H}^*$. Furthermore, if β is large, then $\mathbf{H}^* \approx \mathbf{I}$ and Newton's method is reduced to a steepest-descent method [5]. The Hessian tends to become nonpositive definite at points that are far from the solution and this is where the steepest-descent method is most effective. In effect, this modified Newton's method utilizes the benefits from both the steepest-descent and Newton's method. The preceding modified Newton's method is applied when the ratio of the smallest eigenvalue of $\mathbf{H}$ to the largest is less than 1×10^{-15}.

2.5 Gradient and Hessian

Two sets of matrices are required: the gradient and Hessian of the objective function and the gradient and Hessian of the error function in Eq. 2.7.

2.5.1 Gradient Vector of the Objective Function

The gradient of the objective function $E(\mathbf{x})$ can be evaluated as

$$\begin{aligned} \nabla E(\mathbf{x}) &= \frac{p}{2}\sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^{p-2}[\overline{e}_i(\mathbf{x})\nabla e_i(\mathbf{x}) + e_i(\mathbf{x})\nabla \overline{e}_i(\mathbf{x})] \\ &= p\sum_{i=1}^{M} w_i |e_i(\mathbf{x})|^{p-2} Re[\overline{e}_i(\mathbf{x})\nabla e_i(\mathbf{x})] \end{aligned} \tag{2.16}$$

where $\overline{e}_i(\mathbf{x})$ denotes the complex conjugate of $e_i(\mathbf{x})$, and $\nabla e_i(\mathbf{x})$ is the gradient of the error function.

2.5.2 Gradient of the Error Function

The gradient of the error function can be represented as

$$\nabla e_i(\mathbf{x}) = \left[\frac{\partial e_i(\mathbf{x})}{\partial a_{01}} \quad \frac{\partial e_i(\mathbf{x})}{\partial a_{11}} \quad \frac{\partial e_i(\mathbf{x})}{\partial b_{01}} \quad \frac{\partial e_i(\mathbf{x})}{\partial b_{11}} \quad \cdots \frac{\partial e_i(\mathbf{x})}{\partial b_{0J}} \quad \frac{\partial e_i(\mathbf{x})}{\partial b_{1J}} \quad \frac{\partial e_i(\mathbf{x})}{\partial H_0} \quad \frac{\partial e_i(\mathbf{x})}{\partial \tau}\right]^T \tag{2.17}$$

where

$$\frac{\partial e_i(\mathbf{x})}{\partial a_{0k}} = \frac{F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})} \tag{2.18}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial a_{1k}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})} \tag{2.19}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial b_{0k}} = -\frac{F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})} \tag{2.20}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial b_{1k}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})} \tag{2.21}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial H_0} = \frac{F(\mathbf{x},\omega_i)}{H_0} \tag{2.22}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \tau} = j\omega_i F_0(\omega_i). \tag{2.23}$$

Although the derivatives of the error function are complex qualities, Eq. 2.16 generates a real-valued gradient since the imaginary parts cancel out when the error and its conjugates are added, i.e.,

$$\overline{e}_i(\mathbf{x})\nabla e_i(\mathbf{x}) + e_i(\mathbf{x})\nabla\overline{e}_i(\mathbf{x}) = 2Re[\overline{e}_i(\mathbf{x})\nabla e_i(\mathbf{x})].$$

2.5.3 Hessian of the Objective Function

The optimization method requires the Hessian of the objective function as well as the Hessian of the error function. The Hessian of the objective function is

$$\nabla^2 E(\mathbf{x}) = p\sum_{i=1}^{M} w_i \mathbf{H}_i^{(E_x)}(\mathbf{x}) \tag{2.24}$$

where

$$\begin{aligned}\mathbf{H}_i^{(E_x)}(\mathbf{x}) = \quad & (p-2)|e_i(\mathbf{x})|^{p-4}Re[\overline{e}_i(\mathbf{x})\nabla e_i(\mathbf{x})]Re[\overline{e}_i(\mathbf{x})\nabla^T e_i(\mathbf{x})] \\ & +p|e_i(\mathbf{x})|^{p-2}Re[\overline{e}_i(\mathbf{x})\nabla^2 e_i(\mathbf{x}) + \nabla\overline{e}_i(\mathbf{x})\nabla^T e_i(\mathbf{x})]\end{aligned} \tag{2.25}$$

$\nabla e_i(\mathbf{x})$ is the gradient of the error function $e_i(\mathbf{x})$ with respect to the parameter vector $\mathbf{x}$ and $\nabla^2 e_i(\mathbf{x})$ represents the Hessian of the error function with respect to $\mathbf{x}$. The closed-form equations that give the Hessian of the error function are derived in the following section. An interesting observation for the least-pth optimization when $p = 2$ is that Eq. 2.24 reduces to

$$\nabla^2 E(\mathbf{x}) = 2p \sum_{i=1}^{M} w_i Re[\overline{e}_i(\mathbf{x}) \nabla^2 e_i(\mathbf{x}) + \nabla \overline{e}_i(\mathbf{x}) \nabla^T e_i(\mathbf{x})]. \tag{2.26}$$

2.5.4 Hessian of the Error Function

The Hessian of the error function $\nabla^2 e_i(\mathbf{x})$ is defined as

$$\nabla^2 e_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{01} \partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{11} \partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{01} \partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{11} \partial \tau} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{01}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{11}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \tau} \end{bmatrix}. \tag{2.27}$$

Unexpectedly, this matrix is not block diagonal as was the case in the optimization problem considered by Ko in [2]. This is due to the form of the transfer function given in Eq. 2.1 as opposed to the all-pass biquadratic form of the transfer function given in [2]. Also, the objective function presented in this chapter incorporates both the magnitude and phase responses whereas only the group delay is optimized in [2].

The evaluation of the matrix in Eq. 2.27 can be simplified by splitting the matrix into several block symmetric matrices and taking advantage of the symmetry property of the Hessian. The second-order partial derivatives with respect to only the transfer function coefficients are divided into two block matrix types, the diagonal 4×4 block matrices and the lower

triangular 4×4 block matrices. The diagonal block matrices are defined as

$$H_{kk}^{(\text{diag})} = \begin{bmatrix} 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0k}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0k}\partial b_{1k}} \\ 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1k}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1k}\partial b_{1k}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial b_{1k}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial b_{1k}} \end{bmatrix} \tag{2.28}$$

where $k = 1 \ldots J$ is the section number of the transfer function. The $H_{kk}^{(\text{diag})}$ block matrices are the symmetric blocks that are located along the diagonal of the Hessian and represent all of the second-order partial derivative combinations with respect to the transfer function coefficients for the same section. By virtue of the symmetric property of the Hessian, the upper triangular matrix is equal to the lower triangular matrix. Therefore, only the equations for the lower triangular matrix and the diagonal second-order partial derivatives need to be evaluated. It was also found that the upper 2×2 block diagonal submatrices of the $H_{kk}^{(\text{diag})}$ blocks are equal to zero. The equations for the $H_{kk}^{(\text{diag})}$ block matrices are as follows:

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial a_{0k}} = -\frac{F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.29}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial a_{0k}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.30}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial a_{1k}} = \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial a_{0k}} \tag{2.31}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial a_{1k}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_k(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.32}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0k}\partial b_{0k}} = \frac{2F(\mathbf{x},\omega_i)}{D_k^2(e^{j\omega_i T})} \tag{2.33}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial b_{0k}} = \frac{2e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_k^2(e^{j\omega_i T})} \tag{2.34}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1k}\partial b_{1k}} = \frac{2e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_k^2(e^{j\omega_i T})} \tag{2.35}$$

With further investigation, all of the 2×2 submatrices within the $H_{kk}^{(\text{diag})}$ blocks were also found to be block symmetric as can be seen from Eq. 2.31, therefore, only 6 equations are

needed to obtain all of the $H_{kk}^{(\text{diag})}$ blocks in the Hessian.

The lower triangular block types are given by

$$H_{jk}^{(\text{lower})} = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{1k}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{1k}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{1k}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{1k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{0k}} & \frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{1k}} \end{bmatrix} \tag{2.36}$$

where $j = 1 \ldots J, k = 1 \ldots J, j \neq k$ are the transfer function section numbers. Unlike the diagonal block matrices, the lower triangular block matrices are not symmetric and, therefore, there are 16 equations for each block. The $H_{jk}^{(\text{lower})}$ block matrix elements represent the second-order partial derivative combinations with respect to the transfer function coefficients for sections where $j \neq k$. The equations for the $H_{jk}^{(\text{lower})}$ block matrices are as follows.

The equations for the first column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{0k}} = \frac{F(\mathbf{x}, \omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.37}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1k}\partial a_{0k}} = \frac{e^{j\omega_i T}F(\mathbf{x}, \omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.38}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{0k}} = -\frac{F(\mathbf{x}, \omega_i)}{D_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.39}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{0k}} = -\frac{e^{j\omega_i T}F(\mathbf{x}, \omega_i)}{D_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.40}$$

The equations for the second column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial a_{1k}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.41}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial a_{1k}} = \frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.42}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial a_{1k}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.43}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial a_{1k}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{2.44}$$

The equations for the third column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{0k}} = -\frac{F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.45}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{0k}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.46}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{0k}} = \frac{F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.47}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{0k}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.48}$$

The equations for the fourth column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{0j}\partial b_{1k}} = -\frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.49}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial a_{1j}\partial b_{1k}} = -\frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.50}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{0j}\partial b_{1k}} = \frac{e^{j\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.51}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial b_{1j}\partial b_{1k}} = \frac{e^{j2\omega_i T}F(\mathbf{x},\omega_i)}{D_j(e^{j\omega_i T})D_k(e^{j\omega_i T})} \tag{2.52}$$

The remaining equations needed to construct the Hessian are the second-order partial deriva-

tives with respect to H_0 and τ. The second-order partial derivatives with respect to the multiplier constant H_0 are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{0j}} = \frac{F(\mathbf{x}, \omega_i)}{H_0 N_k(e^{j\omega_i T})} \tag{2.53}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial a_{1j}} = \frac{e^{j\omega_i T} F(\mathbf{x}, \omega_i)}{H_0 N_k(e^{j\omega_i T})} \tag{2.54}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{0j}} = -\frac{F(\mathbf{x}, \omega_i)}{H_0 D_k(e^{j\omega_i T})} \tag{2.55}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial b_{1j}} = -\frac{e^{j\omega_i T} F(\mathbf{x}, \omega_i)}{H_0 D_k(e^{j\omega_i T})} \tag{2.56}$$

The second-order partial derivatives with respect to the group delay constant τ are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{0j}} = 0 \tag{2.57}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial a_{1j}} = 0 \tag{2.58}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{0j}} = 0 \tag{2.59}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial b_{1j}} = 0 \tag{2.60}$$

and

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial H_0} = 0 \tag{2.61}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial H_0} = 0 \tag{2.62}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \tau} = -\omega_i^2 M_0(\omega_i) e^{-j\omega_i \tau}. \tag{2.63}$$

2.5.5 Building the Hessian

The Hessian is constructed by evaluating the $H_{kk}^{(\text{diag})}$ and $H_{jk}^{(\text{lower})}$ matrices at ω_i for each section. There are J matrix blocks for both the $H_{kk}^{(\text{diag})}$ and $H_{jk}^{(\text{lower})}$ matrices where J is the total number of biquadratic sections in the digital filter. Both matrix block types are evaluated to form the lower triangular part of the Hessian. Next, the second-order partial derivatives with respect to H_0 are added at the $(J-1)$th row and the second-order partial derivatives with respect to τ are added at the Jth row. Finally, the elements are reflected about the main diagonal to fill the upper triangular part to form the complete Hessian of the error function at ω_i.

With the gradient and Hessian of the error function for each ω_i known, the Hessian of the objective function can be evaluated using Eq. 2.24.

2.6 Constraints

In addition to the required objective function as well as its gradient and Hessian, the optimization method requires two primary sets of constraints, the step limits and the stability constraints. Furthermore, an additional set of constraints is integrated to restrict the poles from being attracted to the real axis.

2.6.1 Step Limits

The step limits are required to ensure that Newton's method continues to converge in a descent direction towards a feasible solution. Without step limits, any given iteration may produce undesirably large changes in $\mathbf{x}$. As a result, the objective function evaluation may increase instead of decrease. In such a case, the optimization would not be effective since

the goal is to minimize the objective function. This situation may occur due to numerical sensitivity of the highly nonlinear problem. To prevent this situation, optimization step limits are integrated to aid convergence. The step limits in Eq. 2.14 are modified to achieve an improved search direction. Through many simulations it was observed that the delay parameter, τ, generally assumes a much larger value than the coefficient parameters and changes in correspondingly larger steps. Using a common step limit as shown in Eq. 2.14 cannot restrain all the coefficients efficiently. If the common step limits are too large, the coefficient values may increase too rapidly and the optimization will not converge. This may also produce undesirable oscillations in the objective function evaluation thus causing the optimization to fail. If the common step limits are too small, the delay parameter will not change rapidly enough to produce a valid solution. Based on these observations, the step limit vector is defined as

$$\boldsymbol{\beta}^{(k)} = [\beta_{c_i}^{(k)} \quad \beta_{H_0}^{(k)} \quad \beta_{\tau}^{(k)}]^T \tag{2.64}$$

where $\beta_{c_i}^{(k)}$ represent the limits imposed on the coefficients of all the transfer function sections $i = 1 \ldots J$, $\beta_{H_0}^{(k)}$ represents the step limits for the transfer function multiplier constant, and $\beta_{\tau}^{(k)}$ is the step limit for the delay parameter during the current optimization iteration k.

In each iteration, the modified β values are determined as

$$\begin{aligned}
\beta_{c_i}^{(k)} &= \beta_{c_i}^{(k-1)} - R_\beta(\beta_{c_i}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)} \\
\beta_{H_0}^{(k)} &= \beta_{H_0}^{(k-1)} - R_\beta(\beta_{H_0}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)} \\
\beta_{\tau}^{(k)} &= \beta_{\tau}^{(k-1)} - \tfrac{1}{2}R_\beta(\beta_{\tau}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)}
\end{aligned} \tag{2.65}$$

where R_β is referred as the decrease ratio and $f^{(k)}$ is the objective function evaluated at $\mathbf{x}^{(k)}$ for the current optimization iteration k. Through extensive design simulations, a suitable value for the decrease ratio was found to be $R_\beta = 0.5$. In other words, when the value of the objective function increases, the step limits for the coefficient and transfer function constant parameters are restricted to $(1 - R_\beta)$ of their previous values and the step limit for the delay parameter is restricted to $(1 - \frac{1}{2}R_\beta)$ of its previous value.

Other modifications were tested, such as separate step limits for the numerator and denominator coefficient parameters but the optimization process produced better results when these step limits were equal. Also, an updating scheme was used to control the zero and pole positions based on the step limits. This scheme as well as further analysis and observations are presented in Chapter 4.

2.6.2 Filter Stability Constraints

The second set of optimization constraints are used to assure digital filter stability. The digital filter is considered stable if and only if all of the poles are located within the unit circle of the z plane [1]. The transfer function is represented as a product of biquadratic transfer functions where each has two poles. Therefore, the poles of each biquadratic transfer function must be located inside the unit circle to guarantee stability.

The denominator of each second-order section is said to be a Schur polynomial if the polynomial $p(z)$ contains real coefficients and if the roots of $p(z) = 0$ are located strictly inside the unit circle [3]. Consider the second-order monic polynomial

$$p_i(z) = b_{0i} + b_{1i}z + z^2 \tag{2.66}$$

where i is the section number and b_{0i}, b_{1i} are real valued coefficients. The polynomial $p_i(z)$ is a Schur polynomial if and only if [3][1]

$$b_{0i} < 1 \tag{2.67}$$

$$b_{1i} - b_{0i} < 1 \tag{2.68}$$

$$-b_{1i} - b_{0i} < 1 \tag{2.69}$$

As a result, the stability region for $p_i(z)$ is an open triangle in the b_{0i}, b_{1i} coefficient space

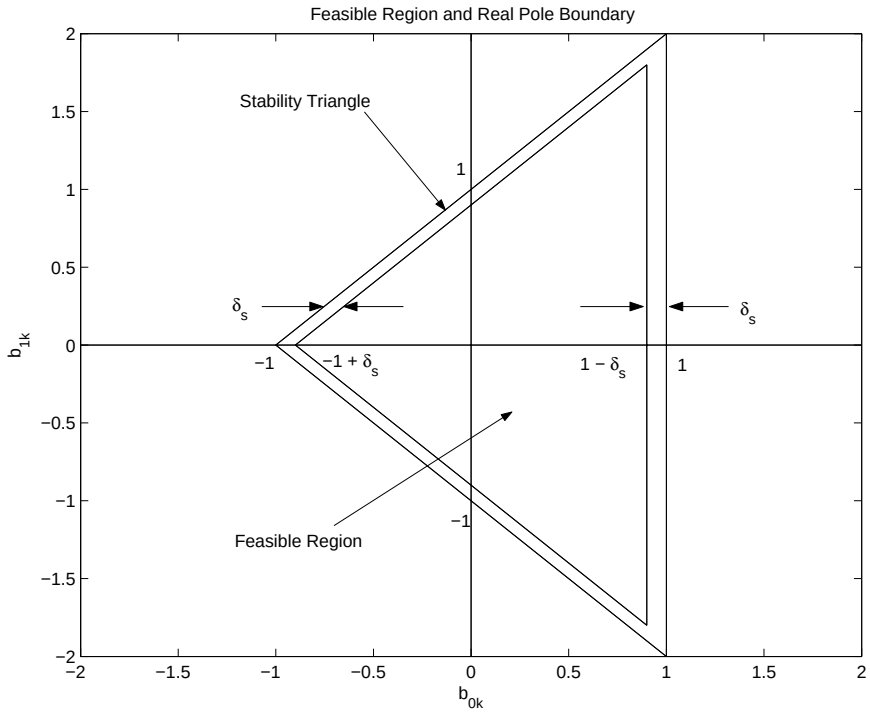

Figure 2.1: Stability region with a stability margin δ_s.

and shown in Figure 2.1.

During optimization, the denominator coefficients may assume values on or outside the stability triangle boundaries and would yield an unstable design. This problem can be eliminated by introducing a stability margin $0 < \delta_s < 1$ where $-1 + \delta_s \leq b_0 \leq 1 - \delta_s$ and $2(1 - \delta_s) \leq b_1 \leq -2(1 - \delta_s)$. The stability triangle and stability margin are shown in Figure 2.1.

Applying the stability margin to Eqs. 2.67-2.69, the following inequalities are obtained:

$$b_{0i} \leq 1 - \delta_s \tag{2.70}$$

$$b_{1i} - b_{0i} \leq 1 - \delta_s \tag{2.71}$$

$$-b_{1i} - b_{0i} \leq 1 - \delta_s. \tag{2.72}$$

If these inequalities are satisfied, filter stability is guaranteed.

Simulations have shown that when the stability margin, δ_s, assumes a small value of about 0.01, undesired peaks in the magnitude response can occur near the passband edge. These peaks can be reduced by slightly increasing the stability margin to about 0.05. Other methods to suppress undesirable peaks, such as nonuniform sampling and adjusting the weighting vector, are investigated in Chapter 3.

2.6.3 Real Pole Boundary Constraints

During many simulations it was observed that a pair of complex conjugate poles may move close to the real axis and when this happens the poles tend to be attracted to the real axis. In such a situation a poor design is obtained. This problem can be eliminated by further examining the second-order polynomial given in Eq. 2.66 and then defining an additional set of linear constraints.

The poles are the roots of the polynomial given in Eq. 2.66 and are real if

$$b_{0i} < \tfrac{1}{4} b_{1i}^2. \tag{2.73}$$

In effect, a nonlinear real-pole boundary is defined within the feasible region of the stability triangle. The real-pole boundary is shown in Figure 2.2 where the space located to the left of the boundary will be referred to as the real region and the right as the complex region. If a point in the coefficient space is located in the real region, the set of poles corresponding to the denominator polynomial will be located on the real axis inside the unit circle. Conversely, a point located in the complex region of the coefficient space corresponds to poles that are not located on the real axis but are somewhere inside the unit circle as a complex conjugate pair.

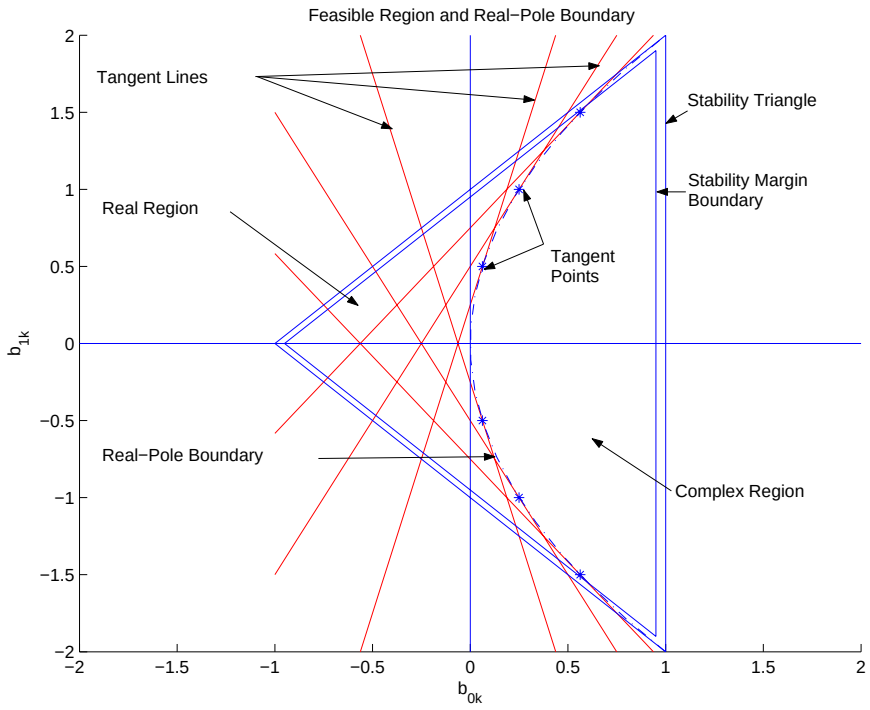

Figure 2.2: The feasible region and real-pole boundary.

The reason why the poles tend to be attracted to the real axis during many optimizations appears to be related to the fact that a large part of the stability region of the coefficient space pertains to poles located on the real axis.

In order to limit or control how close the poles can be allowed to approach the real axis, an additional set of linear constraints are defined within the coefficient space. These additional constraints restrict the solution point from moving into the real region. Furthermore, these constraints can be shifted using a margin control variable, δ_m.

Since the real-pole boundary cannot be represented by a linear equation it cannot be directly used as a constraint equation in the optimization algorithm. To overcome this problem, a set of lines that are tangent to the real-pole boundary are defined to provide an interpolated estimate of the boundary equation.

Several tangent lines located at 6 points along the real pole boundary as well as the b_1 axis were used to provide an approximation. The more points used, the closer the approximation approaches the shape of the real-pole boundary. To limit the algorithm computation, only 7 linear equations were used to provide the real-pole boundary constraint. The points were located at

$$b_{1k} = -1.5,\ \ -1,\ \ -0.5,\ 0,\ 0.5,\ 1,\ 1.5$$

The equation of the tangent lines located at the 6 points are

$$b_{1k} = (4/3)b_{0k} + 3/4 \tag{2.74}$$

$$b_{1k} = 2b_{0k} + 1/2 \tag{2.75}$$

$$b_{1k} = 4b_{0k} + 1/4 \tag{2.76}$$

$$b_{0k} = 0 \tag{2.77}$$

$$b_{1k} = -(4/3)b_{0k} - 3/4 \tag{2.78}$$

$$b_{1k} = -2b_{0k} - 1/2 \tag{2.79}$$

$$b_{1k} = -4b_{0k} - 1/4 \tag{2.80}$$

and are illustrated in Figure 2.2.

2.6.4 Building the Constraint Matrices

Using the three sets of constraint equations described, the required constraint matrices can be obtained. From Eq. 2.12, the linear constraint equations are represented in the matrix form as

$$\mathbf{A}\boldsymbol{\delta} \leq \mathbf{b} \tag{2.81}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

where $\mathbf{A}_1, \mathbf{b}_1$ correspond to the step limits, $\mathbf{A}_2, \mathbf{b}_2$ correspond to the filter stability constraints, and $\mathbf{A}_3, \mathbf{b}_3$ correspond to the real-pole boundary constraints.

Eq. 2.14 can be represented in terms of two linear equations for each element of $\boldsymbol{\delta}$ as

$$\delta_{a_{ji}} \leq \beta_{c_i} \tag{2.82}$$

$$-\delta_{a_{ji}} \leq \beta_{c_i} \tag{2.83}$$

$$\delta_{b_{ji}} \leq \beta_{c_i} \tag{2.84}$$

$$-\delta_{b_{ji}} \leq \beta_{c_i} \tag{2.85}$$

$$\delta_{H_0} \leq \beta_{H_0} \tag{2.86}$$

$$-\delta_{H_0} \leq \beta_{H_0} \tag{2.87}$$

$$\delta_{\tau} \leq \beta_{\tau} \tag{2.88}$$

$$-\delta_{\tau} \leq \beta_{\tau} \tag{2.89}$$

for $j = 0, 1$ and $i = 1 \dots J$. Therefore, the step limits can be expressed as

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & -1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta_{a_{01}}^{(k)} \\ \delta_{a_{11}}^{(k)} \\ \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \\ \vdots \\ \\ \\ \delta_{b_{0i}}^{(k)} \\ \delta_{b_{1i}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \beta_c \\ \beta_c \\ \beta_c \\ \beta_c \\ \vdots \\ \beta_c \\ \beta_c \\ \beta^{(H_0)} \\ \beta^{(H_0)} \\ \beta^{(\tau)} \\ \beta^{(\tau)} \end{bmatrix}. \tag{2.90}$$

Next, the stability constraint equations can be obtained in terms of $\boldsymbol{\delta}$. Using Eq. 2.13, the

denominator coefficients can be represented as

$$b_{0i}^{(k)} = b_{0i}^{(k-1)} + \delta_{b_{0i}}^{(k)} \tag{2.91}$$

$$b_{1i}^{(k)} = b_{1i}^{(k-1)} + \delta_{b_{1i}}^{(k)} \tag{2.92}$$

where k is the current optimization iteration. Substituting Eqs. 2.91-2.92 into Eq. 2.66, the denominator polynomials for the kth iteration can be obtained as

$$(b_{0i}^{(k-1)} + \delta_{0i}^{(k)}) + (b_{1i}^{(k-1)} + \delta_{1i}^{(k)})z + z^2 \tag{2.93}$$

and the stability equations become

$$\delta_{b_{0i}}^{(k)} \leq 1 - \delta_s - b_{0i}^{(k-1)} \tag{2.94}$$

$$\delta_{b_{1i}}^{(k)} - \delta_{b_{0i}}^{(k)} \leq 1 - \delta_s - b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \tag{2.95}$$

$$-\delta_{b_{1i}}^{(k)} - \delta_{b_{0i}}^{(k)} \leq 1 - \delta_s + b_{1i}^{(k-1)} + b_{0i}^{(k-1)}. \tag{2.96}$$

The stability equations can then be represented in the matrix form as

$$\begin{bmatrix} \mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \delta_{a_{01}}^{(k)} \\ \delta_{a_{11}}^{(k)} \\ \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{0i}}^{(k)} \\ \delta_{b_{1i}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_2^{(1)(k)} \\ \mathbf{b}_2^{(2)(k)} \\ \vdots \\ \mathbf{b}_2^{(J)(k)} \end{bmatrix} \tag{2.97}$$

where

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}, \quad \mathbf{b}_2^{(i)(k)} = \begin{bmatrix} 1 - \delta_s - b_{0i}^{(k-1)} \\ 1 - \delta_s - b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \\ 1 - \delta_s + b_{1i}^{(k-1)} + b_{0i}^{(k-1)} \end{bmatrix}$$

with i representing the section number and k the number of the current optimization iteration.

Next, the real-pole boundary constraints are obtained. Like the stability constraints, the real-pole boundary constraint equations are formulated by substituting Eqs. 2.74-2.80 into Eqs. 2.91-2.92. After substitution and rearranging, the equations become

$$\delta_{b_{1i}}^{(k)} - 2\delta_{b_{0i}}^{(k)} \leq 1/2 - \delta_m + 2b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{2.98}$$

$$\delta_{b_{1i}}^{(k)} - (4/3)\delta_{b_{0i}}^{(k)} \leq 3/4 - \delta_m + (4/3)b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{2.99}$$

$$\delta_{b_{1i}}^{(k)} - 4\delta_{b_{0i}}^{(k)} \leq 1/4 - \delta_m + 4b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{2.100}$$

$$-\delta_{b_{0i}}^{(k)} \leq -\delta_m + b_{0i}^{(k-1)} \tag{2.101}$$

$$-\delta_{b_{1i}}^{(k)} - 2\delta_{b_{0i}}^{(k)} \leq 1/2 - \delta_m + 2b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \tag{2.102}$$

$$-\delta_{b_{1i}}^{(k)} - (4/3)\delta_{b_{0i}}^{(k)} \leq 3/4 - \delta_m + (4/3)b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \tag{2.103}$$

$$-\delta_{b_{1i}}^{(k)} - 4\delta_{b_{0i}}^{(k)} \leq 1/4 - \delta_m + 4b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \tag{2.104}$$

where δ_m is a margin control variable that can be used to shift the position of the tangent lines along the b_0 axis shown in Figure 2.2. In other words, if δ_m is set to zero then the tangent lines are located along the real-pole boundary but if set to 0.5, then the tangent lines are shifted along the positive b_0 axis in the coefficient space.

The margin control variable can be used to restrain any poles from entering the real region. Due to the linear interpolation with the tangent lines, there are still small areas between the tangent lines and the real pole boundary that lie inside the real region. However, increasing the margin control variable will shift the tangent lines in the positive b_0-axis direction

eliminating these small areas. Another method to eliminate these small areas is to define additional tangent lines but this will increase the complexity of the constrained optimization problem.

If the margin control variable is increased, the feasible region is decreased making it more difficult for the optimization to converge to an optimal solution. More importantly, the real-axis attraction problem is eliminated and the resulting pole positions are all complex conjugates. Intuitively, this should produce better solutions but, unfortunately, it was found to do the opposite.

Through many simulations, it was found that if the margin control variable is decreased to -0.05 or -0.1 the tangent lines shift slightly into the real region of the coefficient space. The resulting feasible region is increased but part of the feasible region is now located in the real region. This tends to result in better solutions because the poles would travel into the real region and hit the tangent line boundary and then, in most cases, return to the complex region converging to an acceptable solution. The optimization tended to converge to a better solution when the feasible region was enlarged slightly.

The real-pole constraint equations can be represented in the matrix form as

$$\begin{bmatrix} \mathbf{A}_3 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \delta_{a_{01}}^{(k)} \\ \delta_{a_{11}}^{(k)} \\ \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{0i}}^{(k)} \\ \delta_{b_{1i}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_3^{(1)(k)} \\ \mathbf{b}_3^{(2)(k)} \\ \vdots \\ \mathbf{b}_3^{(J)(k)} \end{bmatrix} \tag{2.105}$$

where

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 & -2 & 1 \\ 0 & 0 & -4/3 & 1 \\ 0 & 0 & -4 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & -1 \\ 0 & 0 & -4/3 & -1 \\ 0 & 0 & -4 & -1 \end{bmatrix}, \quad \mathbf{b}_3^{(i)(k)} = \begin{bmatrix} 1/2 - \delta_m + 2b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ 3/4 - \delta_m + (4/3)b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ 1/4 - \delta_m + 4b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \\ -\delta_m + b_{0i}^{(k-1)} \\ 1/2 - \delta_m + 2b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \\ 3/4 - \delta_m + (4/3)b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \\ 1/4 - \delta_m + 4b_{0i}^{(k-1)} + b_{1i}^{(k-1)} \end{bmatrix}.$$

Finally, the constraint matrices given in Eqs. 2.90, 2.97 and 2.105 are combined to form the complete set of linear inequality constraint equations given in Eq. 2.81.

2.7 Initialization

The choice of an initialization point is very critical in achieving an acceptable solution in constrained optimization.

When considering the traditional method of designing linear-phase digital filters using equalization, the method consists of two main steps [1]. The first step consists of designing the filter based on the magnitude response specifications ignoring the group delay. The second step consists of designing a digital equalizer that will be cascaded with the filter to compensate for the variations in the group delay of the filter. In the first step, an optimal solution is obtained using the elliptic approximation. The group delay is not considered and, therefore, this type of initialization for the optimization presented in this chapter is not suitable. In the second step, a special type of initialization is used as described in [1].

Two suitable methods based on the group delay requirements are used to obtain the initial parameter vector for the optimization problem.

The first method was developed by Saab in [6] and generates the initial point based on numerous simulation trends. The second method applies the balanced truncation method to a higher-order FIR filter that approximates the desired frequency response [9].

2.7.1 Method by Trends

By further inspecting Eq. 2.1, we note that there are certain limitations to the zero-pole movements depending on the positions of the initial poles and zeros. If an optimal design requires a set of real zeros within the unit circle where one is negative and the other positive but the initial zeros are complex-conjugate pairs located outside the unit circle, then the optimal design will not be achieved. This occurs because the value of the objective function must increase as the zeros enter the unit circle but since Newton's method minimizes the objective function, the zeros will never reach the optimal point. A similar situation can arise for the pole locations within the unit circle. This problem can be eliminated by using an all-pass digital filter where the radius of the poles is 0.5 and the radius of the zeros is 2 and the angle difference between adjacent poles and zeros in an Nth-order filter with N even is $2\pi/N$. Through many simulations using the all-pass initialization an interesting zero/pole pattern emerged. It is this pattern that is developed to form a tailored initialization point that can yield lower errors, and potentially lower-order filters which satisfy the same specifications [6]. These tailored initial points contain specific zero positions that are tailored to the required passband phase and stopband attenuation. These two sets of zeros are called *phase zeros* and *magnitude zeros* [6]. The phase zeros are located outside of the unit circle and are used to obtain a linear phase. The magnitude zeros are located on the unit circle and are used to provide attenuation in the stopband. An example of this initialization strategy is illustrated in Figure 2.3 for a 10th-order lowpass filter where the passband edge is 0.2π rad/s and $T = 1$ s.

The algorithm for generating initial points for a lowpass or highpass digital filter can be

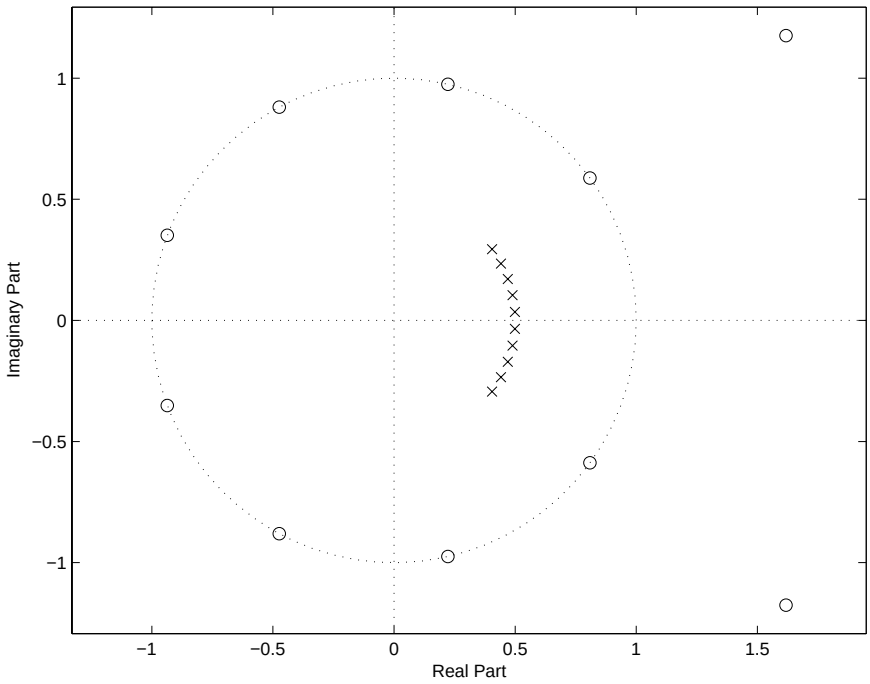

Figure 2.3: Initial pole/zero placement for a 10th-order lowpass filter.

found in [6] and it is also included in Appendix A.1 for the sake of convenience.

2.7.2 Balanced Model Truncation Method

The balanced model truncation method (BMT) is used to reduce the order of an FIR filter design to a IIR filter approximation [9]. The problem of converting FIR to IIR reduced-order filters has been discussed in the signal processing literature for several years [9][10]. In the early eighties, a new powerful method emerged that used a balanced form of a state-space representation for a dynamic system [11][12][13].

The main steps for this technique are to first convert the filter transfer function into a state-space balanced model, then reduce the system order, and finally convert the model back to a reduced-order transfer function.

The initial transfer function is obtained based on a linear-phase FIR digital filter satisfying the magnitude response specifications. Several different methods that design the initial FIR digital filter were investigated using available functions from the MATLAB Signal Processing Toolbox.

Through several simulations and tests with the available MATLAB FIR design methods, the constrained least-square filter design function `fircls1` produced the desired results.

This FIR design algorithm is a multiple exchange algorithm that uses Lagrange multipliers and Kuhn-Tucker conditions in each iteration [14]. This method is quite distinct from many other FIR design techniques such that it does not exclude the region around the cut-off frequency from the integral square error and thereby overcomes the Gibbs' phenomenon without using window functions [14].

A brief description of the algorithm is as follows:

1. Initialization
2. Minimization with Equality Constraints
3. Kuhn-Tucker Conditions
4. Multiple Exchange of the Constraint Set
5. Check for Convergence

A full description of this algorithm can be found in [14].

The FIR filter design required the cut-off frequency, ω_p, the maximum passband deviation from 1, δ_p, and the maximum stopband deviation from 0, δ_a. This design method produced better results than other MATLAB FIR design methods since in many cases the magnitude

specifications are satisfied after applying the BMT. Therefore, only the phase response specifications need to be optimized for linearity. Although, this may not occur in all cases, the design technique produces very good initialization parameters for the optimization problem.

BMT Algorithm

The FIR nth-order transfer function is represented as

$$F(z) = c_0 c_1 z^{-1} + c_2 z^{-2} + \cdots + c_n z^{-n}. \tag{2.106}$$

The Hankel matrix obtained from the FIR transfer function is

$$\mathbf{H}_{ha} = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \\ c_2 & c_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_n & 0 & \cdots & 0 \end{bmatrix} \tag{2.107}$$

Using the singular-value decomposition (SVD), the Hankel matrix can be represented as

$$\mathbf{H}_{ha} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{2.108}$$

where $\mathbf{U}^T\mathbf{V} = \mathbf{I}$, $\mathbf{I}$ is an $n \times n$ identity matrix, and $\mathbf{S}$ is a diagonal matrix containing the eigenvalues of $\mathbf{H}_{ha}$ [15].

Next, the kth-order balanced model truncation of a balanced or input-normal system is obtained by rejecting states associated with the $(n-k)$ smallest singular values. This can be implemented without explicitly computing the balanced (or even input-normal) realization

if the formulas for the state-space representation of the new kth-order truncated system, i.e.,

$$\mathbf{A}_t = \mathbf{V}(2:n, 1:k)^T \mathbf{V}(1:n-1, 1:k) \tag{2.109}$$

$$\mathbf{B}_t = \mathbf{V}(1, 1:k)^T \tag{2.110}$$

$$\mathbf{C}_t = \mathbf{U}(1, 1:k)\mathbf{S}(1:k, 1:k) \tag{2.111}$$

$$\mathbf{D}_t = c_0 \tag{2.112}$$

are used [10][15]. In the above equations, $\mathbf{V}(i:j, k:l)$ and $\mathbf{U}(i:j, k:l)$ denote the extraction of rows from i to j and columns from k to l from matrices $\mathbf{V}$ and $\mathbf{U}$, respectively. Finally, the reduced state-space representation is transformed back to the transform function as

$$F_r(z) = \mathbf{C}_t(z\mathbf{I} - \mathbf{A}_t)^{-1}\mathbf{B}_t + \mathbf{D}_t \tag{2.113}$$

where $F_r(z)$ is the balanced model truncation of the nth-order $F(z)$ given in Eq. 2.106. An in-depth analysis of the foundation of the BMT algorithm is given in [15].

2.8 Termination

For several designs, the optimization algorithm would not converge to a solution that would satisfy the prescribed specifications. To ensure that the algorithm terminates in such situations, a maximum number of 200 optimization iterations was used to terminate the algorithm. A more sophisticated termination criterion is to use the progress ratio which is defined as

$$r = \frac{|E_i(\mathbf{x}) - E_{i-1}(\mathbf{x})|}{E_i(\mathbf{x}) - E_{i-1}(\mathbf{x})} \tag{2.114}$$

where $E(\mathbf{x})$ is the value of the objective function at $\mathbf{x}$ and i is the optimization iteration. A suitable termination condition was found to be $r \leq 1 \times 10^{-7}$.

The progress ratio is also used to dynamically update the step limits during optimization as discussed in Chapter 4.

2.9 Conclusions

A constrained quadratic programming optimization method using a modified Newton's method was described. The derivation for the gradient and Hessian of the objective function required by the modified Newton's method was also presented.

The constrained optimization method described is useful for designing nearly linear-phase recursive digital filters having arbitrary specifications. The optimization algorithm can be further investigated by looking into an alternative objective function to reduce the complexity associated with the large number of the linear constraints.

Three sets of constraints were developed to provide reasonable design results and two initialization techniques were discussed.

In the next chapter an alternative objective function is presented that addresses the real-axis attraction problem and provides a simplified set of linear constraints for the optimization problem.

Chapter 3

New Problem Formulation

> Genius without education is like silver in the mine.
>
> —Benjamin Franklin

3.1 Introduction

In Chapter 2, some issues concerning the development of the objective function that warrant further investigation were identified. For instance, a large number of complicated linear constraints are required and there is a need to control the real-axis attraction problem. In an attempt to alleviate these problems, the objective function is reformulated in terms of the zeros and poles of the transfer function in polar form following an approach in Ko's thesis [2]. This chapter also deals with the effects of the pole and zero positions on the filter's group delay and attempts to identify why certain zero/pole formations occur. The chapter deals, in addition, with the derivation of the gradient and Hessian of the new objective function.

Another issue concerns the amount of computation required. In order to achieve a reasonable solution a dense set of sample points is required. However, increasing the number of sample points increases the amount of computation. To mitigate this issue, a nonuniform variable

sampling technique based on certain ideas presented in [1] is investigated.

Lastly, to provide insight into the filter designs as well as the quality of the filter during the design process, several new filter quality factors are proposed.

3.2 Real-Axis Attraction

During many simulations, the poles located inside the unit circle tended to migrate toward the real axis resulting in suboptimal designs. In Chapter 2, a method was proposed for the control and possible elimination of this problem that involves introducing an additional set of linear constraints plus a margin control variable. Further investigation into the form of the digital filter's transfer function given in Eq. 2.1 revealed that there is an alternative approach for dealing with the real-axis attraction problem. Motivated by Ko's group delay analysis in [2], the transfer function can be represented in terms of its poles and zeros in polar form as

$$H(\mathbf{x}_1, z) = H_0 \prod_{k=1}^{J} \frac{(z - r_{ak}e^{j\theta_{ak}})(z - r_{ak}e^{-j\theta_{ak}})}{(z - r_{bk}e^{j\theta_{bk}})(z - r_{bk}e^{-j\theta_{bk}})} = H_0 \prod_{k=1}^{J} \frac{r_{ak}^2 - 2r_{ak}\cos(\theta_{ak})z + z^2}{r_{bk}^2 - 2r_{bk}\cos(\theta_{bk})z + z^2}. \quad (3.1)$$

The parameter vector can now be represented in terms of the radii and angles of the poles and zeros as

$$\mathbf{x}_1 = [r_{a0} \;\; \theta_{a0} \;\; r_{b0} \;\; \theta_{b0} \;\; \cdots \;\; r_{bJ} \;\; \theta_{bJ} \;\; H_0]^T. \quad (3.2)$$

Therefore, in order for the position of the kth pole to be located on the real axis within the unit circle, the angle θ_{bk} must be equal to zero or π as shown in Figure 3.1. Additionally, unlike the optimization using the coefficient-based objective function, it was observed that the pole pairs are not as easily attracted to the real axis and, further, the optimization path to the solution is more direct. This may be due to the fact that the unstable region in the polar parameter space is small compared to the stable region. More importantly, a simple

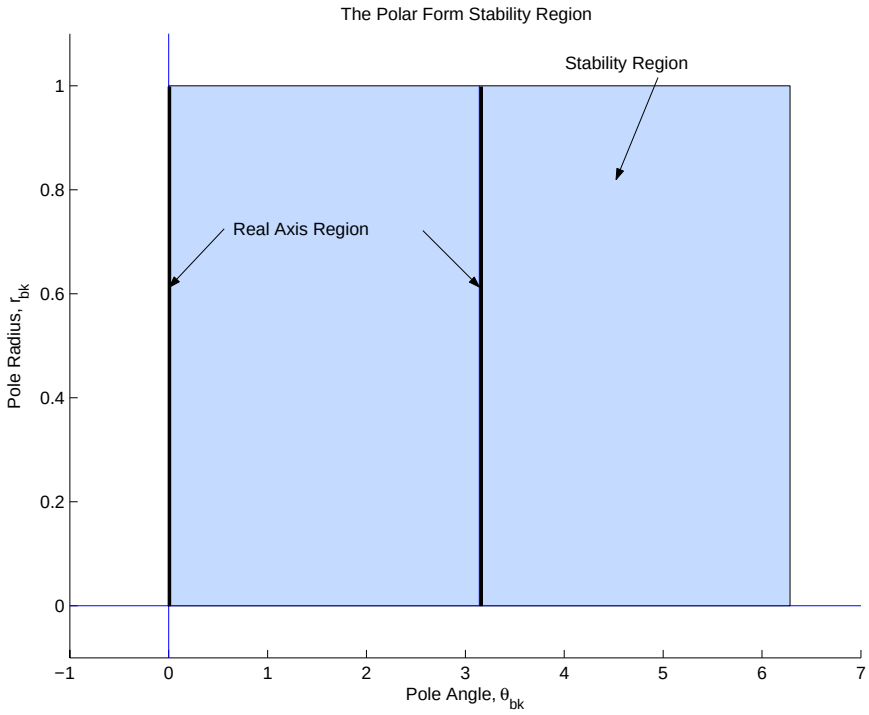

Figure 3.1: Stability region corresponding to the polar-based transfer function.

linear constraint equation can be employed to force the angles of the poles to be greater than zero to eliminate the possibility of poles moving to the positive real axis. Furthermore, if needed, an additional set of constraints can be added to maintain the poles within the passband sector of the z plane. Since the examples in this thesis are lowpass digital filters, the poles cannot reside on the negative real axis and, therefore, the possibility where $\theta_{bk} = \pi$ can be eliminated. For other filter designs requiring poles to be located in the left-hand z plane another linear constraint equation can be introduced to prevent poles from moving to the negative real axis. Since the filter designs investigated in this thesis require poles to be located in the right-hand z plane, this linear constraint equation was not implemented so as to not increase the amount of computations.

3.3 Zero/Pole Position Characteristics

An interesting aspect observed in many lowpass filter designs was the final locations of the poles and zeros within the passband sector of the z plane which is defined by the passpand of the filter. For example, for a lowpass filter with passband $0 \leq \omega \leq \omega_p$, where ω_p is the passband edge, the passband sector is defined by $-\omega_p T \leq \theta \leq \omega_p T$ where T is the sampling period.

For a lowpass filter design, the poles formed an ellipse within the passband sector and the zeros are split up into two formations. Typically, several zeros are located on or near the unit circle in the stopband sector (magnitude zeros) and several zeros are located outside of the unit circle (phase zeros). As mentioned in Chapter 2, the magnitude zeros contribute to the attenuation in the stopband and the phase zeros contribute to the linear phase in the passband.

With some additional investigation, the reason why this structure emerges can be explained. The group delay of the filter is given by

$$\tau_F(\omega) = -\frac{d\theta_F(\omega)}{d\omega} \tag{3.3}$$

where

$$\theta_F(\omega) = \arg\, F(\mathbf{x}_1, \omega) \tag{3.4}$$

and $F(\mathbf{x}_1, \omega)$ is given in Eq. 2.5. From Eqs. 2.1 and 3.3 the group delay can be represented as [1]

$$\tau_F(\omega) = -T \sum_{k=1}^{J} \frac{\tilde{N}_k(\omega)}{N_k(\omega)} + T \sum_{k=1}^{J} \frac{\tilde{D}_k(\omega)}{D_k(\omega)} \tag{3.5}$$

where T is the sampling period and

$$\begin{aligned}
\tilde{N}_k(\omega) &= 1 - a_{0k}^2 + a_{1k}(1 - a_{0k})\cos\,\omega T \\
N_k(\omega) &= (1 - a_{0k})^2 + a_{1k}^2 + 2a_{1k}(1 + a_{0k})\cos\,\omega T + 4a_{0k}\cos^2\,\omega T \\
\tilde{D}_k(\omega) &= 1 - b_{0k}^2 + b_{1k}(1 - b_{0k})\cos\,\omega T \\
D_k(\omega) &= (1 - b_{0k})^2 + b_{1k}^2 + 2b_{1k}(1 + b_{0k})\cos\,\omega T + 4b_{0k}\cos^2\,\omega T.
\end{aligned}$$

The group delay for the polar form can be reduced by comparing Eqs. 3.1 and 2.1 and then rearranging terms to obtain the following relationships:

$$\begin{aligned}
a_{0k} &= r_{ak}^2 \\
a_{1k} &= -2r_{ak}\cos(\theta_{ak}) \\
b_{0k} &= r_{bk}^2 \\
b_{1k} &= -2r_{bk}\cos(\theta_{bk}).
\end{aligned}$$

Substituting the preceding equations into Eq. 3.5, the group delay becomes

$$\begin{aligned}
\tau_F(\omega) &= -T\sum_{k=1}^{J}\frac{1 - r_{ak}^4 - 2r_{ak}\cos(\theta_{ak})(1 - r_{ak}^2)\cos(\omega T)}{(1 - r_{ak}^2)^2 + 4r_{ak}^2\cos(\theta_{ak}) - 4r_{ak}\cos(\theta_{ak})(1 + r_{ak}^2)\cos(\omega T) + 4r_{ak}^2\cos^2(\omega T)} \\
&\quad +T\sum_{k=1}^{J}\frac{1 - r_{bk}^4 - 2r_{bk}\cos(\theta_{bk})(1 - r_{bk}^2)\cos(\omega T)}{(1 - r_{bk}^2)^2 + 4r_{bk}^2\cos(\theta_{bk}) - 4r_{bk}\cos(\theta_{bk})(1 + r_{bk}^2)\cos(\omega T) + 4r_{bk}^2\cos^2(\omega T)} \\
&= -T\sum_{k=1}^{J}G_z^{(k)} + T\sum_{k=1}^{J}G_p^{(k)} \qquad (3.6)
\end{aligned}$$

where $G_z^{(k)}$ and $G_p^{(k)}$ are the contributions from the zeros and poles respectively for section k. One can immediately see that the contribution from the zeros for each section has the opposite effect relative to the contributions from the poles. Also, there is a common ratio in Eq. 3.6 that corresponds to the contribution from the poles and the contribution from the

zeros given by

$$G(\omega) = \frac{1 - r^4 - 2r\cos(\theta)(1 - r^2)\cos(\omega T)}{(1 - r^2)^2 + 4r^2\cos(\theta) - 4r\cos(\theta)(1 + r^2)\cos(\omega T) + 4r^2\cos^2(\omega T)}. \tag{3.7}$$

This will be referred to as the *polar group delay ratio*. The polar group delay ratio can be investigated graphically using a time-step 3-dimensional plot where a surface plot of $G(\omega)$ is generated for $0 \leq \theta \leq w_s/2$ for $0 \leq \omega \leq w_s/2$ and $0 \leq r \leq 2$. In other words, θ is incremented through the range $\{0 : w_s/2\}$ and a 3-dimensional surface plot is generated for $G(\omega)$ during each increment of θ. The result is quite interesting and provides insight as to why the optimization generates the elliptical formation of the poles in the passband and the elliptical formation of the phase zeros outside of the unit circle.

The surface plots for increasing values of θ displayed distinct peaks near $r = 1$ for several values of ω. The amplitude of the peaks grew larger near $\omega = 0$ and $\omega = \pi$. To illustrate the peaks, a single plot is shown in Figure 3.2 for the case $\theta = \pi/2$. One can see that there are two distinct peaks occurring one on each side of $r = 1$ with opposite values. Although, the peaks appear mirrored about $r = 1$, the peaks are actually slightly different due to a slight amplitude increase near $r = 0$ because of the $(1 - r^2)^2$ term in the denominator. More importantly, the optimization algorithm shifts the zeros into locations outside the unit circle producing group delay contributions to compensate for the contributions from the pole locations within the unit circle. As a result, the poles are positioned inside the unit circle so as to produce negative delay contributions near the negative value peak locations from the polar group delay ratio. Similarly, the zeros are positioned outside the unit circle so as to produce positive delay contributions near these distinct peak locations from the polar group delay ratio. For example, the case where $\theta = \pi/2$, the algorithm moved a pole so as to produce a delay τ near the delay value corresponding to the negative peak illustrated in Figure 3.2 for $r < 1$. At the same time, the algorithm also moved a zero so as to produce a delay τ near the delay value corresponding to the positive peak illustrated in Figure 3.2 for $r > 1$. This leads to the elliptical formation of the poles inside the unit circle and the

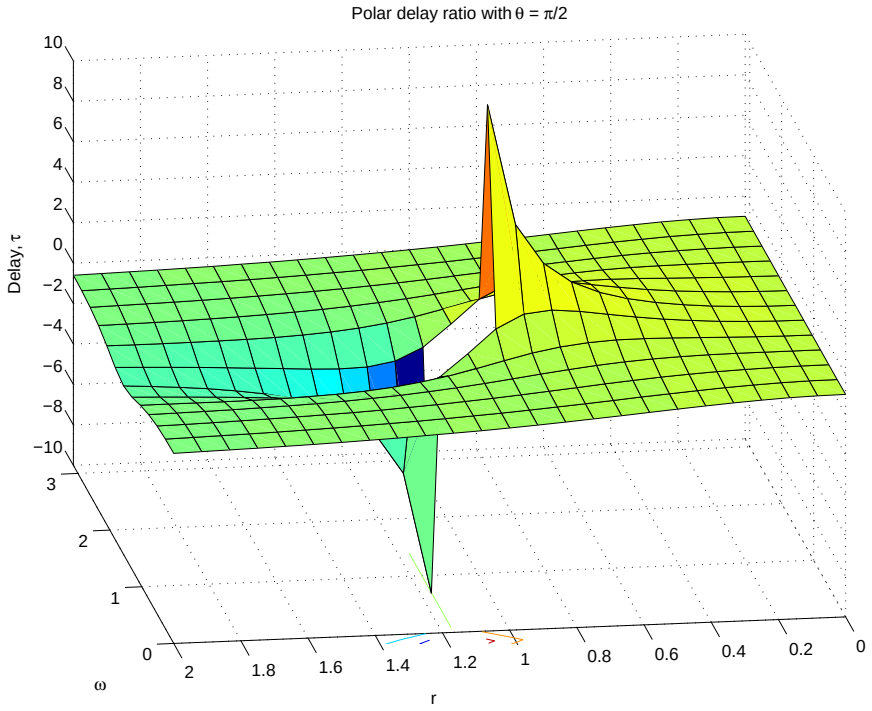

Figure 3.2: Plot of the polar delay ratio $G(\omega)$ for $\theta = \pi/2$.

elliptical formation of the zeros outside the unit circle. The reason for these formations is to create a constant polar delay ratio for all values of θ within the passband. An interesting observation is that one can possibly calculate the approximate locations of the poles and zeros before optimization is applied using the polar group delay equation. This is discussed in Chapter 6 as a possible improved initialization technique.

3.4 New Objective Function

In Chapter 2, the objective function given in Eq. 2.7 was derived from the actual and desired frequency response equations represented in terms of filter coefficients. As a result, several linear constraint equations are required to assure filter stability making the optimization more complicated. Furthermore, additional constraints were introduced to prevent the poles

from being attracted to the real axis resulting in suboptimal designs. By reformulating the transfer function in terms of its zeros and poles in polar form, a new objective function can be constructed. With the new objective function represented in terms of the zero/pole radii and angles, two immediate advantages are gained. First, the real axis-attraction problem can be controlled. Second, the stability constraint equations are reduced to $0 < r < 1$ where r is the radius of the pole for a given biquadratic factor.

3.4.1 Problem Formulation

The transfer function given in Eq. 2.1 can be represented in polar form as shown in Eq. 3.1 and the actual frequency response equation can be obtained as

$$F(\mathbf{x}_1, \omega_i) = H_0 \prod_{k=1}^{J} \frac{r_{ak}^2 - 2r_{ak}\cos(\theta_{ak})e^{j\omega_i T} + e^{j2\omega_i T}}{r_{bk}^2 - 2r_{bk}\cos(\theta_{bk})e^{j\omega_i T} + e^{j2\omega_i T}} \tag{3.8}$$

where the parameter vector $\mathbf{x}_1$ is given in Eg. 3.2. The error function is still given by

$$e_i(\mathbf{x}) = F(\mathbf{x}_1, \omega_i) - F_0(\omega_i)$$

and the desired frequency response is given in Eq. 2.4. As in Chapter 2, the complete parameter vector is given by

$$\mathbf{x} = [\mathbf{x}_1^T \quad \tau]^T$$

and the new objective function is the same as Eq. 2.7. The gradient and Hessian for the objective function will now be obtained in terms of the zero/pole radii and angles.

3.4.2 Gradient and Hessian

As in section 2.5, the constrained optimization method requires both the gradient and Hessian of the objective function. Again, there are two sets of gradients and Hessians to consider, the gradient and Hessian of the new objective function and the gradient and Hessian of the new error function.

The gradient and Hessian of the objective function are given by Eqs. 2.16 and 2.24, respectively but the gradient and Hessian of the new error function are different.

Gradient of the Error Function

The gradient of the error function is given by

$$\nabla e_i(\mathbf{x}) = \left[\frac{\partial e_i(\mathbf{x})}{\partial r_{a1}} \quad \frac{\partial e_i(\mathbf{x})}{\partial \theta_{a1}} \quad \frac{\partial e_i(\mathbf{x})}{\partial r_{b1}} \quad \frac{\partial e_i(\mathbf{x})}{\partial \theta_{b1}} \quad \cdots \frac{\partial e_i(\mathbf{x})}{\partial r_{bJ}} \quad \frac{\partial e_i(\mathbf{x})}{\partial \theta_{bJ}} \quad \frac{\partial e_i(\mathbf{x})}{\partial H_0} \quad \frac{\partial e_i(\mathbf{x})}{\partial \tau} \right]^T \tag{3.9}$$

where

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{ak}} = \left[2r_{ak} - 2\cos(\theta_{ak}) e^{j\omega_i T} \right] \frac{F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T})} \tag{3.10}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{ak}} = \left[2r_{ak}\sin(\theta_{ak}) e^{j\omega_i T} \right] \frac{F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T})} \tag{3.11}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{bk}} = \left[-2r_{bk} + 2\cos(\theta_{bk}) e^{j\omega_i T} \right] \frac{F(\mathbf{x}, \omega_i)}{D_k(e^{j\omega_i T})} \tag{3.12}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{bk}} = \left[-2r_{bk}\sin(\theta_{bk}) e^{j\omega_i T} \right] \frac{F(\mathbf{x}, \omega_i)}{D_k(e^{j\omega_i T})} \tag{3.13}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial H_0} = \frac{F(\mathbf{x}, \omega_i)}{H_0} \tag{3.14}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \tau} = j\omega_i F_0(\omega_i). \tag{3.15}$$

Hessian of the Error Function

The Hessian of the error function $\nabla^2 e_i(\mathbf{x})$ is given by

$$\nabla^2 e_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{a1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{a1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{b1}\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{b1}\partial \tau} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial H_0\partial \tau} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial r_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \theta_{a1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial r_{b1}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \theta_{b1}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial H_0} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau\partial \tau} \end{bmatrix} \tag{3.16}$$

Like the coefficient-based Hessian presented in Chapter 2, the matrix in Eq. 3.16 is not block diagonal and several equations are required to construct it.

The Hessian in Eq. 3.16 can be simplified by splitting it into several block symmetric matrices and taking advantage of the symmetry properties of the Hessian. The second-order partial derivatives with respect to only the zero/pole radii and angles are divided into two block matrix types, the diagonal 4×4 block matrices and the lower triangular 4×4 block matrices. The diagonal block matrices are given by

$$H_{kk}^{(\text{diag})} = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial r_{ak}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{ak}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{ak}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{ak}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk}\partial \theta_{bk}} \end{bmatrix} \tag{3.17}$$

where $k = 1 \ldots J$ is the number of sections. The $H_{kk}^{(\text{diag})}$ block matrices are symmetric blocks that are located along the diagonal of the Hessian and represent all of the second-order partial derivative combinations with respect to the zero/pole radii and angles for the same biquadratic factor. From the symmetry of the Hessian, the upper triangular matrix is the

reflection of the lower triangular matrix and, therefore, only the equations for the lower triangular matrix and the diagonal second-order partial derivatives are required.

Another interesting observation is that the upper 2×2 block diagonal submatrices of the $H_{kk}^{(\text{diag})}$ blocks are not equal to zero, unlike the blocks for the coefficient-based Hessian diagonal blocks. The equations for the $H_{kk}^{(\text{diag})}$ block matrices are as follows.

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{ak} \partial r_{ak}} = \frac{2F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T})} \tag{3.18}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak} \partial r_{ak}} = \frac{\left[2\sin(\theta_{ak})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T})} \tag{3.19}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk} \partial r_{ak}} = \frac{\left[2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right] \left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T}) D_k(e^{j\omega_i T})} \tag{3.20}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk} \partial r_{ak}} = \frac{\left[2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right] \left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T}) D_k(e^{j\omega_i T})} \tag{3.21}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{ak} \partial \theta_{ak}} = \frac{\left[2r_{ak}\cos(\theta_{ak})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T})} \tag{3.22}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk} \partial \theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right] \left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T}) D_k(e^{j\omega_i T})} \tag{3.23}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk} \partial \theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right] \left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{N_k(e^{j\omega_i T}) D_k(e^{j\omega_i T})} \tag{3.24}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk} \partial r_{bk}} = \frac{-2F(\mathbf{x}, \omega_i)}{D_k(e^{j\omega_i T})} + \frac{[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}][-4r_{bk} + 4\cos(\theta_{bk})e^{j\omega_i T}]F(\mathbf{x}, \omega_i)}{D_k^2(e^{j\omega_i T})} \tag{3.25}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk} \partial r_{bk}} = \frac{F(\mathbf{x}, \omega_i)}{D_k(e^{j\omega_i T})} \left(\left[-2\sin(\theta_{bk})e^{j\omega_i T}\right] + \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right] \left[-4r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]}{D_k(e^{j\omega_i T})} \right) \tag{3.26}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bk} \partial \theta_{bk}} = \frac{F(\mathbf{x}, \omega_i)}{D_k(e^{j\omega_i T})} \left(\left[-2\cos(\theta_{bk})e^{j\omega_i T}\right] + \frac{\left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right] \left[-4r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]}{D_k(e^{j\omega_i T})} \right) \tag{3.27}$$

The lower triangular blocks are given by

$$H_{jk}^{(\text{lower})} = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial \theta_{bk}} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{ak}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{bk}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial \theta_{bk}} \end{bmatrix} \tag{3.28}$$

where $j = 1 \ldots J, k = 1 \ldots J, j \neq k$. Unlike the diagonal block matrices, the lower-triangular block matrices are not symmetric and, therefore, there are 16 equations for each block. The $H_{jk}^{(\text{lower})}$ block matrix elements represent the second-order partial derivative combinations with respect to the transfer function coefficients for the cases where $j \neq k$. The equations for the $H_{jk}^{(\text{lower})}$ block matrices are given in equations (3.29)-(3.44) below.

The equations for the first column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{ak}} = \frac{\left[2r_{rak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right]\left[2r_{raj} - 2\cos(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.29}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{aj}\partial r_{ak}} = \frac{\left[2r_{rak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right]\left[2r_{raj}\sin(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.30}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bk}\partial r_{ak}} = \frac{\left[2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right]\left[-2r_{bj} + 2\cos(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.31}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \theta_{bj}\partial r_{ak}} = \frac{\left[2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right]\left[-2r_{bj}\sin(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.32}$$

The equations for the second column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial\theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right]\left[2r_{aj} - 2\cos(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.33}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{aj}\partial\theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right]\left[2r_{aj}\sin(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.34}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial\theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right]\left[-2r_{bj} + 2\cos(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.35}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{bj}\partial\theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right]\left[-2r_{bj}\sin(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.36}$$

The equations for the third column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial r_{bk}} = \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right]\left[2r_{aj} - 2\cos(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.37}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{aj}\partial r_{bk}} = \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right]\left[2r_{aj}\sin(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.38}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial r_{bk}} = \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right]\left[-2r_{bj} + 2\cos(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.39}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{bj}\partial r_{bk}} = \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right]\left[-2r_{bj}\sin(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.40}$$

The equations for the fourth column of the $H_{jk}^{(\text{lower})}$ block matrices are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{aj}\partial\theta_{bk}} = \frac{\left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]\left[2r_{aj} - 2\cos(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.41}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{aj}\partial\theta_{bk}} = \frac{\left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]\left[2r_{aj}\sin(\theta_{aj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_j(e^{j\omega_i T})N_k(e^{j\omega_i T})} \tag{3.42}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial r_{bj}\partial\theta_{bk}} = \frac{\left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]\left[-2r_{bj} + 2\cos(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.43}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial\theta_{bj}\partial\theta_{bk}} = \frac{\left[-2r_{bk}\sin(\theta_{bk})e^{j\omega_i T}\right]\left[-2r_{bj}\sin(\theta_{bj})e^{j\omega_i T}\right]F(\mathbf{x},\omega_i)}{N_k(e^{j\omega_i T})D_j(e^{j\omega_i T})} \tag{3.44}$$

The second-order partial derivatives with respect to the transfer function multiplier constant

H_0 are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial r_{ak}} = \frac{\left[2r_{ak} - 2\cos(\theta_{ak})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{H_0 N_k(e^{j\omega_i T})} \tag{3.45}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial \theta_{ak}} = \frac{\left[2r_{ak}\sin(\theta_{ak})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{H_0 N_k(e^{j\omega_i T})} \tag{3.46}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial r_{bk}} = \frac{\left[-2r_{bk} + 2\cos(\theta_{bk})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{H_0 D_k(e^{j\omega_i T})} \tag{3.47}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial \theta_{bk}} = \frac{\left[-2r_{bk}2\sin(\theta_{ak})e^{j\omega_i T}\right] F(\mathbf{x}, \omega_i)}{H_0 D_k(e^{j\omega_i T})}. \tag{3.48}$$

Finally, the second-order partial derivatives with respect to the group delay constant τ are

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial r_{ak}} = 0 \tag{3.49}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \theta_{ak}} = 0 \tag{3.50}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial r_{bk}} = 0 \tag{3.51}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \theta_{bk}} = 0 \tag{3.52}$$

and

$$\frac{\partial e_i^2(\mathbf{x})}{\partial H_0 \partial H_0} = 0 \tag{3.53}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial H_0} = 0 \tag{3.54}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial \tau \partial \tau} = -\omega_i^2 M_0(\omega_i) e^{-j\omega_i \tau}. \tag{3.55}$$

3.4.3 Constraints

Two sets of constraints are used in the new problem formulation; the step limits and filter stability constraints.

Step Limits

As for the coefficient-based objective function, the step limits are used to ensure that the search direction is a descent direction toward a feasible solution. The delay parameter generally assumes a much larger value than the zero/pole radii and angles and changes in corresponding larger steps. To avoid large steps with respect to the zero/pole radii and angles, separate step limits have been used, i.e.,

$$\beta^{(k)} = [\beta_{\hat{p}_i}^{(k)} \quad \beta_{H_0}^{(k)} \quad \beta_{\tau}^{(k)}]^T \tag{3.56}$$

where $\beta_{\hat{p}_i}^{(k)}$ represents the step limits for the zero/pole radii and angles for all biquadratic factors for $i = 1 \ldots J$. The modified β values are then determined in the same manner as for the coefficient-based objective function as

$$\begin{aligned}
\beta_{\hat{p}_i}^{(k)} &= \beta_{\hat{p}_i}^{(k-1)} - R_\beta(\beta_{\hat{p}_i}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)} \\
\beta_{H_0}^{(k)} &= \beta_{H_0}^{(k-1)} - R_\beta(\beta_{H_0}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)} \\
\beta_{\tau}^{(k)} &= \beta_{\tau}^{(k-1)} - \tfrac{1}{2} R_\beta(\beta_{\tau}^{(k-1)}) \quad , \text{ if } f^{(k)} > f^{(k-1)}
\end{aligned} \tag{3.57}$$

where R_β is the reduction ratio and $f^{(k)}$ is the value of the objective function at $\mathbf{x}^{(k)}$ for iteration k. A suitable value for the reduction ratio was found to be $R_\beta = 0.5$. Specific step limits for both forms of the objective function are investigated in Chapter 4.

Filter Stability Constraints

The digital filter is considered stable if and only if all the pole radii for each second-order biquadratic factor is less than one [1]. Since the transfer function is represented in polar form as shown in Eq. 3.1 only two constraint equations are required to limit the pole radii

within the unit circle. The new constraints are

$$0 < r_{bi} < 1 \tag{3.58}$$

where i is the biquadratic factor number and the lower bound of 0 will assure a positive pole radius. As a result, the number of stability constraints can be reduced from three to two per biquadratic factor.

During optimization, pole radii may approach unity leading to undesirable spikes in the frequency response near the passband edge, and in some cases, to an unstable design. These anomalies may occur because the zeros and poles may move into positions to balance the delay contributions as described in section 3.3 forcing the poles closer to the unit circle near the passband edges. Another constraint is used to prevent the pole radii from approaching zero. Although, a small pole radius may lead to an inefficient design it may not necessarily be a suboptimal design. The optimization tends to yield better results when the poles form an elliptical pattern to balance the delay contributions for the linear-phase specifications.

These two situations can be eliminated by introducing a stability margin δ_s in the range $0 < \delta_s < 1$. Incorporating this margin in the stability constraints of Eq. 3.58, the inequalities

$$r_{bi} \leq 1 - \delta_s \tag{3.59}$$

$$r_{bi} \geq \delta_s \tag{3.60}$$

can be obtained.

If these inequalities are satisfied, filter stability is guaranteed. It should be mentioned that using a good initialization technique such as that presented in section 2.7, the lower bound stability equation is not required because poles are less likely to move to the origin of the z plane. Through several simulations, it was observed that the additional constraint increases the complexity of the constrained optimization and did not result in better solutions. There-

fore, the number of constraint equations are reduced to one per section as opposed to three per section when using the coefficient-based objective function. However, in Chapter 4 the lower-bound constraint as well as a passband boundary constraint are integrated for the case of design examples using random initial points.

Building the Constraint Matrices

Using the two sets of constraint equations, the constraint matrices can be constructed. From Eq. 2.12, the linear constraint equations can be represented in the matrix form as $\mathbf{A}\boldsymbol{\delta} \leq \mathbf{b}$ or

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{\delta} \leq \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \tag{3.61}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

where $\mathbf{A}_1, \mathbf{b}_1$ and $\mathbf{A}_2, \mathbf{b}_2$ correspond to the step and stability constraints, respectively.

From Eq. 2.14, two linear equations for each element of $\boldsymbol{\delta}$ can be obtained as

$$\delta_{r_{ai}} \leq \beta_{\hat{p}_i} \tag{3.62}$$

$$-\delta_{r_{ai}} \leq \beta_{\hat{p}_i} \tag{3.63}$$

$$\delta_{\theta_{ai}} \leq \beta_{\hat{p}_i} \tag{3.64}$$

$$-\delta_{\theta_{ai}} \leq \beta_{\hat{p}_i} \tag{3.65}$$

$$\delta_{r_{bi}} \leq \beta_{\hat{p}_i} \tag{3.66}$$

$$-\delta_{r_{bi}} \leq \beta_{\hat{p}_i} \tag{3.67}$$

$$\delta_{\theta_{bi}} \leq \beta_{\hat{p}_i} \tag{3.68}$$

$$-\delta_{\theta_{bi}} \leq \beta_{\hat{p}_i} \tag{3.69}$$

$$\delta_{H_0} \leq \beta_{H_0} \tag{3.70}$$

$$-\delta_{H_0} \leq \beta_{H_0} \tag{3.71}$$

$$\delta_{\tau} \leq \beta_{\tau} \tag{3.72}$$

$$-\delta_{\tau} \leq \beta_{\tau} \tag{3.73}$$

for $i = 1 \ldots J$ where J is the total number of biquadratic factors. Therefore, the step limits

are given by $\mathbf{A}_1\boldsymbol{\delta} \leq \mathbf{b}_1$ or

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & -1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta_{r_{a1}}^{(k)} \\ \delta_{\theta_{a1}}^{(k)} \\ \delta_{r_{b1}}^{(k)} \\ \delta_{\theta_{b1}}^{(k)} \\ \\ \vdots \\ \\ \delta_{r_{bi}}^{(k)} \\ \delta_{\theta_{bi}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \beta_{\hat{p}} \\ \beta_{\hat{p}} \\ \beta_{\hat{p}} \\ \beta_{\hat{p}} \\ \vdots \\ \beta_{\hat{p}} \\ \beta_{\hat{p}} \\ \beta^{(H_0)} \\ \beta^{(H_0)} \\ \beta^{(\tau)} \\ \beta^{(\tau)} \end{bmatrix}. \tag{3.74}$$

Next, the stability constraints can be expressed in matrix form as follows. Using Eq. 2.13, the pole radii can be represented as

$$r_{bi}^{(k)} = r_{bi}^{(k-1)} + \delta_{b_{0i}}^{(k)} \tag{3.75}$$

where k is the number of the current iteration. Combining Eq. 3.58 with Eq. 3.75 and rearranging, the equation becomes

$$\delta_{r_{bi}}^{(k)} \leq 1 - \delta_s - r_{bi}^{(k-1)} \tag{3.76}$$

where δ_s is the stability margin and i is the number of the biquadratic factor. Therefore,

the stability constraints can be expressed as $\mathbf{A}_2\boldsymbol{\delta} \leq \mathbf{b}_2$ or

$$\begin{bmatrix} \mathbf{A}_2 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \delta_{r_{a1}}^{(k)} \\ \delta_{\theta_{a1}}^{(k)} \\ \delta_{r_{b1}}^{(k)} \\ \delta_{\theta_{b1}}^{(k)} \\ \vdots \\ \delta_{r_{bi}}^{(k)} \\ \delta_{\theta_{bi}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_2^{(1)(k)} \\ \mathbf{b}_2^{(2)(k)} \\ \vdots \\ \mathbf{b}_2^{(J)(k)} \end{bmatrix} \tag{3.77}$$

where

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{b}_2^{(i)(k)} = \begin{bmatrix} 1 - \delta_s - r_{bi}^{(k-1)} \end{bmatrix}$$

with i representing the biquadratic factor and k the current optimization iteration.

At this point, the constraint matrices given in Eqs. 3.74 and 3.77 can be combined to form the complete set of linear inequality constraint equations given in Eq. 3.61. Thus the new problem formulation entails $8J + 4$ step limits and J stability constraint equations yielding a total of $9J + 4$ equations. In the coefficient-based formulation, there are $8J + 4$ step limit equations, $3J$ stability constraint equations, and $7J$ real-pole boundary constraint equations yielding a total of $18J + 4$ equations. Therefore, $9J$ constraint equations are eliminated by using the new problem formulation. One immediate disadvantage is that the new objective function's gradient and Hessian contain several sine and cosine functions thereby increasing the computational complexity. More details involving computational comparisons are given in Chapter 4.

3.5 Nonuniform Variable Sampling

To achieve satisfactory results, the sampling of the error function, $e(\mathbf{x}, \omega)$, with respect to ω must be very dense otherwise large error spikes may occur in the intervals between the sampling points. Also, in this optimization problem, both the magnitude and phase response errors are minimized simultaneously making the overall optimization routine very complex. This necessitates the use of a dense set of sampling points for the optimization routine to converge to a good solution.

In a given design, there may be 30 optimization iterations that require several error function evaluations for the complete dense set of sampling points. The number of sampling points is usually in the order of five to ten times the number of variables in the parameter vector. For example, if there are 100 sampling points, each objective function evaluation for a given iteration will involve several hundred error function evaluations. Consequently, the amount of computation during the optimization is considerable.

A useful technique called *nonuniform variable sampling* described by Antoniou in [1] is used to suppress spikes in the error function while maintaining a low number of sampling points.

This technique identifies the locations where spikes are beginning to occur and these locations are used as the initial points for the next iteration in the optimization.

The technique presented here is slightly different from that in [1]. Here a set of sampling points is defined near the transition band where a weighting factor is applied. This reduces the possibility of missing error spikes near the transition band where spikes are more likely to occur. As in [1], the error function evaluation for the dense set of frequency points is done during each iteration. In [1], the ideal magnitude response is interpolated for arbitrary magnitude responses. Since the optimization problem involves designing a lowpass digital filter, interpolation is not required because the ideal magnitude response is one or zero.

Assume that a digital filter is required to have a passband, a transition band, and a stopband such as a typical lowpass or highpass filter. Let the dense set of sampling points be

$$\bar{\Omega} = \{\bar{\omega}_i, 1 \leq i \leq M_L\}$$

where M_L is the total number of sampling points in the order of $10K$ with K representing the total number of intervals in each band. The sampling points are evenly spaced throughout the active bands, i.e., the passband and stopband. The spacing is given by

$$\triangle\bar{\omega} = \frac{\hat{\omega}}{M_L + M_B} \tag{3.78}$$

where $\hat{\omega}$ is the total frequency range in all of the frequency bands and M_B is the total number of band edge points located on both sides of each transition band.

The band edge points are reserved for the weighting factor variable, w, in the objective function of Eq. 2.7. The weighting factor applies additional control over the band edge spikes. In addition, using the band edge points assures that the error at band edges is always minimized during the optimization.

The total frequency range is given by

$$\hat{\omega} = \sum_{j=1}^{N_B} \hat{\omega}_j, \qquad \hat{\omega}_j = \bar{\omega}_j^{\mathrm{e}} - \bar{\omega}_j^{\mathrm{s}} \tag{3.79}$$

where N_B is the total number of active bands, and $\bar{\omega}_j^{\mathrm{s}}$ and $\bar{\omega}_j^{\mathrm{e}}$ are the starting and ending frequencies for the band j, respectively.

To illustrate the frequency bands in more detail, consider a lowpass digital filter design containing a passband, transition band, and a stopband. The resulting frequency bands are

defined as

$$\begin{aligned}
\text{Passband}: \quad \bar{\Omega}_p &= \left\{\bar{\omega}_p : \bar{\omega}_p^{\mathrm{s}} \leq \bar{\omega}_p < \omega_p - \tfrac{1}{2} M_B \triangle\bar{\omega}\right\} \\
\text{Stopband}: \quad \bar{\Omega}_a &= \left\{\bar{\omega}_a : \omega_a + \tfrac{1}{2} M_B \triangle\bar{\omega} < \bar{\omega}_a \leq \tfrac{1}{2}\omega_s\right\} \\
\text{Transition band}: \quad \bar{\Omega}_b &= \left\{\bar{\Omega}_{b1} \quad \bar{\Omega}_{b2}\right\} \\
\text{Before transition band}: \quad \bar{\Omega}_{b1} &= \left\{\bar{\omega}_b : \omega_p - \tfrac{1}{2} M_B \triangle\bar{\omega} \leq \bar{\omega}_b \leq \omega_p\right\} \\
\text{After transition band}: \quad \bar{\Omega}_{b2} &= \left\{\bar{\omega}_b : \omega_a \leq \bar{\omega}_b \leq \omega_a + \tfrac{1}{2} M_B \triangle\bar{\omega}\right\}
\end{aligned}$$

where ω_p, ω_a, and ω_s are the passband edge, stopband edge and sampling frequencies respectively. Next, a set of intervals referred to as *sampling intervals*, are defined where each interval contains one sampling point that will eventually be used in the overall sampling frequency set, Ω.

The sampling intervals are

$$\begin{aligned}
\hat{\Omega}_p^k &= \left\{\omega : \bar{\omega}_i \leq \omega < \bar{\omega}_i + \tfrac{\hat{\omega}_j}{K_p}\right\} \\
\hat{\Omega}_a^k &= \left\{\omega : \bar{\omega}_i \leq \omega < \bar{\omega}_i + \tfrac{\hat{\omega}_j}{K_a}\right\}
\end{aligned}$$

where K_p and K_a are the total sampling intervals in the passband and stopband respectively, k is the sampling interval, $\bar{\omega}_i$ is the frequency value from the dense set, and $\hat{\omega}_j$ is the frequency range defined in Eq. 3.79.

Once the sampling intervals are defined, the frequency point inside each interval where the maximum error occurs is identified and used in the overall sampling frequency set. This set is then used in the next optimization iteration and a new set is defined in each subsequent iteration until a solution is reached.

3.6 Filter Quality

When comparing filter designs, it is preferable to monitor the quality of both the group delay and magnitude response instead of just one single quality factor for the entire design. To facilitate comparisons between various designs, three quality factors are defined which provide additional insight into the quality of filter designs.

The group delay quality factor is inversely related to the maximum variation of τ and is defined as [1]

$$Q_\tau = 100\left(\frac{\overset{\cap}{\tau} - \overset{\cup}{\tau}}{\overset{\cap}{\tau} + \overset{\cup}{\tau}}\right) \tag{3.80}$$

where $\overset{\cap}{\tau}$ and $\overset{\cup}{\tau}$ are the maximum and minimum passband group delays, respectively.

The magnitude error quality factor is inversely related to the maximum variation of the error in the magnitude response and is defined as

$$Q_M = 100\left(\frac{\overset{\cap}{M}_e - \overset{\cup}{M}_e}{\overset{\cap}{M}_e + \overset{\cup}{M}_e}\right) \tag{3.81}$$

where

$$\begin{aligned}\overset{\cap}{M}_e &= \max_{1\le i\le K}|e_i(\mathbf{x})| \\ \overset{\cup}{M}_e &= \min_{1\le i\le K}|e_i(\mathbf{x})|\end{aligned}$$

and $\overset{\cap}{M}_e$ and $\overset{\cup}{M}_e$ represent the maximum and minimum magnitude response errors, respectively.

Using the above quality factors, an overall composite filter quality can be constructed as

$$Q_e = 100\left(\frac{(\overset{\cap}{\tau} - \overset{\cup}{\tau})(\overset{\cap}{M}_e - \overset{\cup}{M}_e)}{(\overset{\cap}{\tau} + \overset{\cup}{\tau})(\overset{\cap}{M}_e + \overset{\cup}{M}_e)}\right) = \frac{1}{100}(Q_\tau Q_M). \tag{3.82}$$

By separating the filter quality factors, the group delay and magnitude response error can be monitored during a given optimization. The separate quality factors can also provide quality comparisons between separate designs.

3.7 Conclusions

The optimization problem described in Chapter 2 was reformulated in terms of the zeros and poles of the transfer function in polar form. This new approach revealed several benefits; filter stability can be achieved by simply constraining the pole radii for each biquadratic factor, and the real-axis attraction characteristic can be controlled.

The effect of the poles and zeros on the group delay has been examined and the reason for the elliptical formation of the poles and zeros has been identified. The zeros and poles are arranged into an elliptical formation due to the balancing of the delay contributions in Eq. 3.6.

The gradient and Hessian were derived. New linear constraints were introduced which lead to a reduction of $9J$ constraint equations from $18J + 4$ to $9J + 4$.

Next, the use of a nonuniform variable sampling technique was investigated. This technique was used to prevent the formation of error spikes during the optimization process with the goal of achieving a better solution. This technique takes advantage of the fact that error spikes usually occur near transition band edges.

Lastly, several filter quality factors have been defined to provide a way of comparing filter designs during a given optimization.

Chapter 4

Comparison of the Objective Functions

The secret of success is to know something nobody else knows.
—Aristotle Onassis

4.1 Introduction

The two objective functions presented in Chapters 2 and 3 provide satisfactory solutions to the design problem. In this chapter, the two objective function are tested using several different initial points for a lowpass filter design to reveal their main benefits and detriments. First, the nonuniform sampling technique described in Chapter 3 is investigated followed by detailed comparisons between designs using the coefficient-based and polar-based objective functions. It should be noted that the designs in Chapters 4 and 5 are carried out using a least-*pth* approach where $p = 2$ is used in the optimization algorithm.

4.2 Nonuniform vs Uniform Variable Sampling

Before the two proposed objective functions are compared, the application of the nonuniform sampling technique described in section 3.5 will be analyzed. The immediate advantage of using this technique is the reduction in the amount of computation required to complete the design. More importantly, this method may produce better results in terms of filter quality.

Firstly, a nearly linear-phase recursive digital filter that must satisfy the specifications given in Table 4.1 was designed using the coefficient-based objective function.

Using the design procedure explained in Chapter 2 for the coefficient-based objective function given in Eq. 2.7, two designs were considered; one using uniform sampling of the error function and one using the nonuniform sampling technique. Both designs were initialized using the balanced model truncation (BMT) method with the initial FIR filter order of 30 and reduced to 10. The stability margin variable, δ_s, was set to 0.06 and the real-pole boundary margin control variable, δ_m, was set to -0.05. When δ_m is set to a negative value, the real-pole boundary described in section 2.6.3 is shifted along the b_{0k} axis in the negative direction. In other words, the real-pole boundary is shifted slightly into the real region of the stability region shown in Figure 2.2. At first this may seem strange because the poles now have a greater chance to move in the real region and be positioned on the real axis, which would result in a suboptimal design. However, by observing the results of several

Parameter	Value
Sampling frequency, ω_s, rad/s	2π
Maximum passband ripple, A_p, dB	0.1
Minimum stopband attenuation, A_a, dB	45
Passband edge, ω_a, rad/s	0.2π
Stopband edge, ω_p, rad/s	0.3π
Maximum standard deviation of group delay in passband, %	6

Table 4.1: Prescribed specifications for the lowpass filter used to compare the nonuniform and uniform sampling schemes.

simulations and the behavior of the poles as they reach the real-pole boundary, it was noted that they tend to be repelled into the complex region if they get close to the boundary. More importantly, the feasible region is slightly enlarged because the real-pole boundary constraints are not as strict. In general, the smaller the feasible region, the more difficult it was for the optimization routine to converge to a solution that satisfies the specifications.

In the least-pth coefficient-based objective function, p was set to 2 and the weights were chosen as

$$w_i = \begin{cases} 1 & \text{for } 1 \leq i \leq (P-4), (P+5) \leq i \leq P_t \\ 0.4 & \text{for } i = P, (P+1) \\ 0.5 & \text{for } i = (P-1), (P+2) \\ 0.6 & \text{for } i = (P-2), (P+3) \\ 0.8 & \text{for } i = (P-3), (P+4) \end{cases} \tag{4.1}$$

where P is the index of the frequency point located at ω_p and P_t is the total number of sample points used to evaluate the error function. This weighting scheme helped reduce the formation of peaks in the magnitude response near the transition band edges.

4.2.1 Analysis and Comparisons

In the design with uniform sampling of the error function, 34 sample points were used in each of the passband and stopband totalling 68 sample points for the entire frequency range. A solution was achieved after 115 iterations taking 30.06 s of CPU time with a filter quality of $Q = 1.419$.

In the design with the nonuniform sampling technique, 30 sample intervals were used in each of the passband and stopband plus 4 band-edge points on each side of the transition band. Therefore, the total number of error samples was 68 which is the same as that used in the case of uniform sampling. The total number of samples was chosen to be the same for both designs to assure a fair comparison. Also, 1000 sample points were used in each band to

Computer Specifications	Value
Clock precision, s	5.0×10^{-8}
Clock speed, MHz	2000
RAM, MB	512
Operating system	Windows XP

Table 4.2: Specifications of the computer used to carry out the designs.

determine the position of the error spikes per sample interval. As a result, a solution was achieved after 44 iterations taking only 20.16 s with a filter quality of $Q = 1.2798$. The designs were performed with a laptop computer with the specifications given in Table 4.2. The results obtained are summarized in Table 4.3 and the magnitude response, passband ripple, and the group delay characteristic are plotted in Figure 4.1a to c.

As one can see, the design using nonuniform sampling resulted in 73 fewer optimization iterations and was about 33% faster than that using uniform sampling. Furthermore, the M-file profiler, a utility for debugging and optimizing M-file code in Matlab, was used to record three types of function evaluations. For each function, the profiler recorded information about the number of parent and child function calls referred to as M-function and M-subfunction evaluations, respectively. Additionally, the profiler recorded the number of Matlab executable file calls referred to as MEX-function evaluations. The total number of function evaluations for each design is given in Table 4.3. The function evaluations for the two designs were very similar suggesting that the nonuniform sampling technique is more efficient in terms of filter quality.

Another interesting observation relates to the quality of the filter after each optimization step. As can be seen in Figure 4.2a, the nonuniform sampling design improved the filter quality factor a lot more rapidly than the uniform sampling design and resulted in an improved quality factor at convergence.

There is also a slight increase in the minimum stopband attenuation with the design using

Sampling Technique	Uniform	Nonuniform
Total iterations	115	42
Total recorded time, s	30.06	20.16
Number of M-functions	207	206
Number of M-subfunctions	134	135
Number of MEX-functions	2	2
Maximum passband ripple, dB	0.032839	0.028853
Minimum stopband attenuation, dB	45.1145	45.2187
Standard deviation of group delay in passband, %	5.9972	5.8825
Composite filter quality	1.4190	1.2798
Objective function evaluation	9.267×10^{-5}	1.445×10^{-4}

Table 4.3: Design results for the uniform and nonuniform error sampling schemes.

nonuniform error sampling as can be seen in Figure 4.1a. Moreover, the uniform sampling design produced a peak in the magnitude response inside the transition band as can be seen in Figure 4.2b, which could be a problem in certain applications.

In conclusion, the design using nonuniform error sampling produced a better quality filter in less time using about the same number of function evaluations. For this reason, all subsequent design examples were designed using the nonuniform sampling technique for both the coefficient-based and polar-based objective functions.

4.3 Design Comparisons of the Proposed Objective Functions

In this section, the coefficient-based and polar-based objective functions are compared in terms of optimization efficiency and computational cost. The objective functions were tested using random initial points, initial points using the method by trends, and initial points using the BMT method.

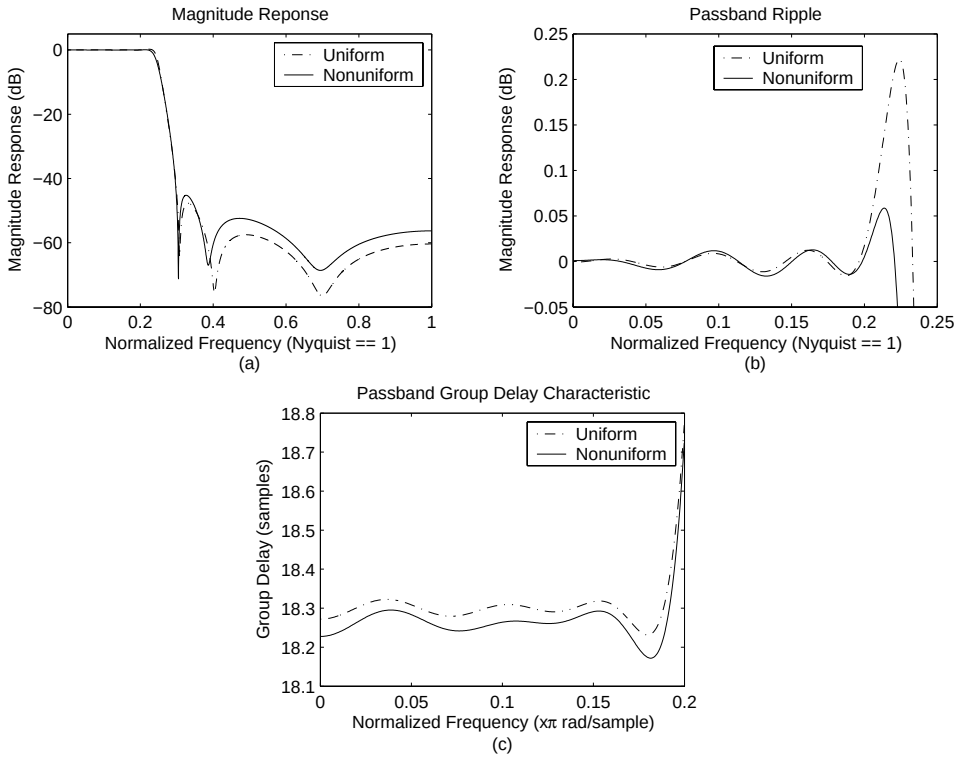

Figure 4.1: Magnitude response, passband ripple, and passband group delay characteristic for the lowpass filter.

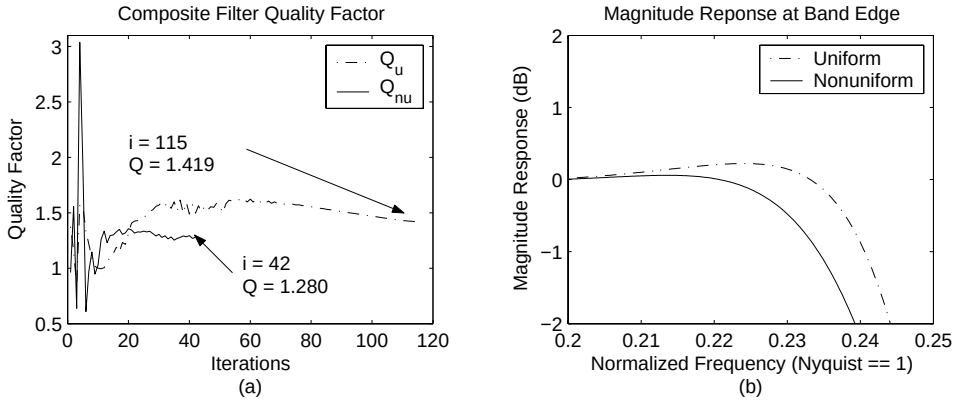

Figure 4.2: Plots of the composite filter quality factors vs iterations and magnitude response in the transition band for uniform and nonuniform variable sampling.

4.3.1 Coefficient-Based Objective Function

Several designs performed using the coefficient-based objective function with random initial points were deemed to be failed attempts since the optimization had to be terminated due to lack of progress. Although a satisfactory solution was not obtained for several designs, a suboptimal solution was usually found which would partially satisfy the magnitude response specifications but the group delay deviation at convergence was too large, suggesting that the phase response specifications are more difficult to satisfy. Also, these suboptimal designs resulted in poles that were not located in or near the passband sector of the z plane and, therefore, the phase zeros did not position themselves in an optimal formation as described in Chapter 3.

This observation suggests that the phase response becomes very difficult to optimize when zeros do not position themselves in the phase zero formation outside of the unit circle. Since the poles would move to suboptimal locations inside the unit circle, the zeros would not move to the formation that was observed to linearize the phase response. One solution for this problem would be to introduce another set of linear constraints that would force the poles into or near the passband sector. Under these circumstances the optimization would arrange zeros into the phase zero formation producing a better solution. Although this seems attractive, the additional constraints would increase the algorithm's complexity. However, investigating the passband boundary within the coefficient space, an easier solution was found.

From Eqs. 2.1 and 3.1

$$b_{0k} = r_{bk}^2, \quad b_{1k} = -2r_{bk}\cos(\theta_{bk})$$

Solving for b_{1k} at $\theta_{bk} = \omega_p$, the passband boundary for a lowpass filter design can be represented as

$$b_{1k} = -2(b_{0k})^{1/2}\cos(\omega_p) \tag{4.2}$$

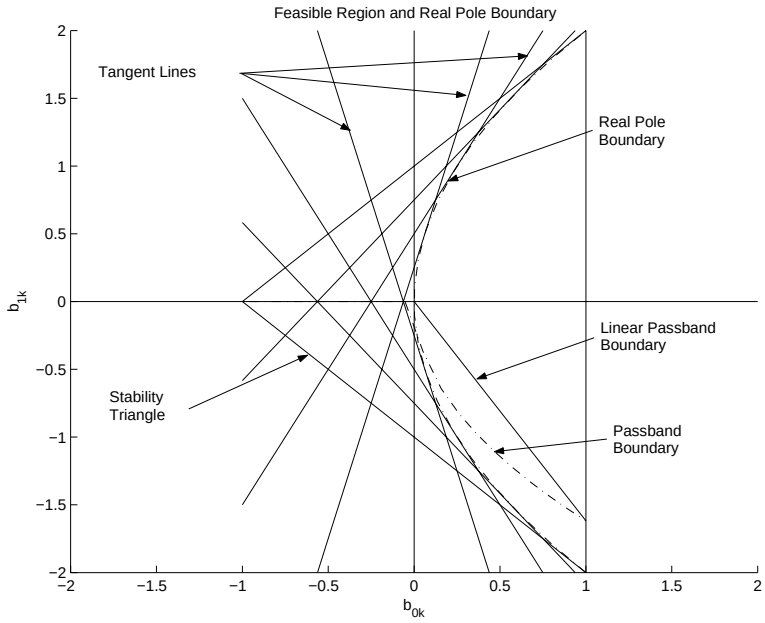

Figure 4.3: The passband boundary inside the coefficient space.

where k is the filter section. This equation represents a nonlinear curve similar to the real-pole boundary of Eq. 2.73 but is located farther into the complex region as shown in Figure 4.3. Therefore, by adding a single line constraint from the origin to the intersection of the passband boundary and the stability triangle, a linear passband boundary constraint can be constructed to force the poles to the vicinity of the passband sector. Although there is a large region under the linear passband boundary outside the passband sector, the poles were observed to move in the correct direction so as to avoid a suboptimal solution and produce satisfactory designs.

After substituting Eqs. 2.91-2.92 into Eq. 4.2, the new passband constraint equation becomes

$$2\cos(\omega_p)\delta_{b_{0i}}^{(k)} + \delta_{b_{1i}}^{(k)} \leq -2\cos(\omega_p)b_{0i}^{(k-1)} - b_{1i}^{(k-1)} \tag{4.3}$$

which is then represented in matrix form as $\mathbf{A}_4\boldsymbol{\delta} \leq \mathbf{b}_4$ or

$$\begin{bmatrix} \mathbf{A}_4 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_4 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_4 \end{bmatrix} \begin{bmatrix} \delta_{a_{01}}^{(k)} \\ \delta_{a_{11}}^{(k)} \\ \delta_{b_{01}}^{(k)} \\ \delta_{b_{11}}^{(k)} \\ \vdots \\ \delta_{b_{0i}}^{(k)} \\ \delta_{b_{1i}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_4^{(1)(k)} \\ \mathbf{b}_4^{(2)(k)} \\ \vdots \\ \mathbf{b}_4^{(J)(k)} \end{bmatrix} \tag{4.4}$$

where

$$\mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 2\cos(\omega_p) & 1 \end{bmatrix}, \quad \mathbf{b}_4^{(i)(k)} = \begin{bmatrix} -2\cos(\omega_p)b_{0i}^{(k-1)} - b_{b1i}^{(k-1)} \end{bmatrix}$$

i is the section number, and k is the number of the current iteration. The new constraint matrices are then combined with Eq. 2.81 to form the following complete set of linear constraints:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \end{bmatrix} \boldsymbol{\delta} \leq \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{bmatrix}. \tag{4.5}$$

4.3.2 Polar-Based Objective Function

As with the coefficient-based objective function, additional sets of linear constraints were required to force the poles into the passband sector when using random initial points. Furthermore, the poles and zeros were prevented from being located near the center of the unit circle thereby forcing the poles and zeros to move outward to a more optimal formation. The

required additional constraints can be represented in terms of the inequalities

$$\theta_{bi} \leq \omega_p$$
$$r_{ai} \geq \hat{\delta}$$
$$r_{bi} \geq \hat{\delta}$$

where i is the section number and $\hat{\delta}$ is the minimum radius variable. A suitable value for the design examples using random initial points was found to be $\hat{\delta} = 0.25$. Using Eq. 3.75, the constraint equations

$$\delta_{\theta_{bi}}^{(k)} \leq \omega_p - \theta_{bi}^{(k-1)} \tag{4.6}$$
$$-\delta_{r_{ai}}^{(k)} \leq r_{ai}^{(k-1)} - \hat{\delta} \tag{4.7}$$
$$-\delta_{r_{bi}}^{(k)} \leq r_{bi}^{(k-1)} - \hat{\delta} \tag{4.8}$$

can be formulated where k is the iteration and i is the section number. The new constraint equations can then be represented in the matrix form as $\mathbf{A}_3\boldsymbol{\delta} \leq \mathbf{b}_3$ or

$$\begin{bmatrix} \mathbf{A}_3 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \delta_{r_{a1}}^{(k)} \\ \delta_{\theta_{a1}}^{(k)} \\ \delta_{r_{b1}}^{(k)} \\ \delta_{\theta_{b1}}^{(k)} \\ \vdots \\ \delta_{r_{bi}}^{(k)} \\ \delta_{\theta_{bi}}^{(k)} \\ \delta_{H_0}^{(k)} \\ \delta_{\tau}^{(k)} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_3^{(1)(k)} \\ \mathbf{b}_3^{(2)(k)} \\ \vdots \\ \mathbf{b}_3^{(J)(k)} \end{bmatrix} \tag{4.9}$$

where

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad \mathbf{b}_2^{(i)(k)} = \begin{bmatrix} \omega_p - \theta_{bi}^{(k-1)} \\ r_{ai}^{(k-1)} - \hat{\delta} \\ r_{bi}^{(k-1)} - \hat{\delta} \end{bmatrix}.$$

These new constraint matrices can then be combined with Eq. 3.61 to form the following complete set of linear constraints

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \boldsymbol{\delta} \leq \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}. \tag{4.10}$$

4.3.3 Step Limit Modifications

Another problem occurred when the value of the objective function became larger than the previous value during several consecutive iterations. This tends to occur when the step limits are too large for a given iteration. For example, when the value of the objective function increases in a given iteration, the step limits are reduced according to section 2.6.1 for the next iteration. If this scenario happens several times, the step limits become very small rendering the optimization useless since the parameter vector values are not changing very much. A solution to this problem is to incorporate a dynamic step updating scheme which increases the step limits based on the objective function evaluation, the progress ratio, and the filter quality factor. In order to prevent cycling when the solution is near convergence, another condition was incorporated in the step constraint.

The dynamic step updating scheme was implemented as follows:

If $|Q| > Q_c$:

  If $[(E_i(\mathbf{x}) > 1)$ and $(r < 0.1)]$ or $[(E_i(\mathbf{x}) < 1)$ and $(r < 0.01)]$:

    $\beta_{c_i}^{(k)} = \beta_{c_i}^{(k-1)} + R_\beta(\beta_{c_i}^{(k-1)})$

$$\beta_{H_0}^{(k)} = \beta_{H_0}^{(k-1)} + R_\beta(\beta_{H_0}^{(k-1)})$$
$$\beta_{\tau}^{(k)} = \beta_{\tau}^{(k-1)} + \tfrac{1}{2}R_\beta(\beta_{\tau}^{(k-1)})$$

where Q_c is a predefined quality factor and r is the progress ratio from Eq. 2.114. A value in the range $2 \geq Q_c \geq 5$ can be used for Q_c depending on the filter order.

The additional condition to prevent cycling near convergence was implemented as follows:

If $(E_i(\mathbf{x}) < 1)$ and $(r < 0.01)$ and $(|Q| < Q_c)$:

$$\beta_{c_i}^{(k)} = \beta_{c_i}^{(k-1)} - R_c(\beta_{c_i}^{(k-1)})$$
$$\beta_{H_0}^{(k)} = \beta_{H_0}^{(k-1)} - R_c(\beta_{H_0}^{(k-1)})$$
$$\beta_{\tau}^{(k)} = \beta_{\tau}^{(k-1)} - R_c(\beta_{\tau}^{(k-1)})$$

where R_c is a predefined constant. A value $R_c = 0.01$ was found to give the desired results.

The above two conditions were found to stabilize the minimization progress as well as helped the optimization algorithm complete the design in cases where cycling near the solution point occurred.

4.3.4 Modified Quality Factors

In the proceeding sections, several designs are compared using the coefficient-based and polar-based objective functions with random initial points. Since a majority of the designs did not satisfy all of the prescribed specifications, four different quality factors are derived to facilitate a better comparison between the designs.

The quality factors are derived with respect to the filter specifications in the following manner; if the value of a given parameter is better than the corresponding specification parameter, the quality factor is set to zero; otherwise, it is assigned a value between 0 and 100.

The quality factor with respect to the magnitude response in the passband is defined as

$$\hat{Q}_p = 100 \left[\frac{(A_{pb} - A_p) + |A_{pb} - A_p|}{2(A_{pb} + A_p)} \right] \tag{4.11}$$

where A_{pb} is the maximum passband ripple of the designed filter and A_p is the prescribed maximum passband ripple. The quality factor with respect to the magnitude response in the stopband region is defined as

$$\hat{Q}_s = 100 \left[\frac{(A_a - A_{sb}) + |A_a - A_{sb}|}{2(A_a + A_{sb})} \right] \tag{4.12}$$

where A_{sb} is the minimum stopband attenuation of the designed filter and A_a is the prescribed minimum stopband attenuation. The quality factor for the group delay is defined as

$$\hat{Q}_\tau = 100 \left[\frac{(\tau_d - \tau_p) + |\tau_d - \tau_p|}{2(\tau_d + \tau_p)} \right] \tag{4.13}$$

where τ_d is the maximum group delay deviation in the passband of the designed filter, and τ_p is the prescribed maximum group delay deviation. Finally, a composite quality factor is derived that combines all three as

$$\hat{Q}_c = \frac{1}{3}[\hat{Q}_p + \hat{Q}_s + \hat{Q}_\tau]. \tag{4.14}$$

In Eq. 4.14 the separate quality factors are added instead of being multiplied because of the possible zero values. A zero quality factor means the corresponding parameter has satisfied the prescribed specification. For example, say the maximum passband rippled is set to 1 dB and the resulting passband ripple in the designed filter is 0.2 dB, then the corresponding quality factor is

$$\hat{Q}_p = 100 \left[\frac{(0.2 - 1) + |0.2 - 1|}{2(0.2 + 1)} \right] = 0.$$

Also, if the resulting composite quality factor $\hat{Q}_c = 0$, then the designed filter has satisfied all the prescribed specifications.

The above quality factors provide an insight as to how good the designs are with respect to the prescribed filter specifications and are used in the following section.

4.3.5 Design Examples Using Random Initial Points

It was of interest to investigate filter designs using completely random initial points for both objective functions. Intuitively, the use of random initial points should not work very well because the objective function is minimized for both the phase and magnitude response resulting in a highly nonlinear function. After running several simulations this was found to be the case and, indeed, the optimization encountered great difficulty converging to satisfactory solutions.

Several lowpass digital filters were designed to satisfy the specifications given in Table 4.4. There were 20 designs using a set of precalculated random initial points for each design. Furthermore, the 20 designs were carried out with filter orders of $n = 10, 12, 14, 16, 18, 20$ where the zeros and poles were randomly located inside the unit circle.

The weighting scheme given in Eq. 4.1 was used for the objective function and the nonuniform sampling technique described in Chapter 3 was implemented using 30 sample intervals per frequency band. The stability margin was set to $\delta_s = 0.05$, the real-pole boundary margin control variable was set to $\delta_m = 0.05$, and the maximum number of iterations was set to 300. Finally, the initial step limits were set to:

$$\beta_{c_i} = 0.03$$

$$\beta_{H_0} = 0.03$$

$$\beta_{\tau} = 0.1$$

where β_{c_i} represents the step limits for the coefficients in the parameter vector and i represents the filter section.

Parameter	Value
Sampling frequency, ω_s, rad/s	2π
Maximum passband ripple, A_p, dB	1
Minimum stopband attenuation, A_a, dB	30
Passband edge, ω_a, rad/s	0.2π
Stopband edge, ω_p, rad/s	0.3π
Maximum standard deviation in the group delay, %	10

Table 4.4: Specifications used for the lowpass digital filter with random initial points.

Results

As expected, the results were poor because of the highly nonlinear objective function. More specifically, the optimization seemed to have difficulty grouping the zeros into the magnitude zero and phase zero formations. This problem was overcome to some extent by incorporating the additional passband boundary constraints to force the poles to the vicinity of the passband sector. There were a total of 20 designs per filter order, where 6 different filter orders were tested for both the coefficient-based and polar-based objective functions. Therefore, a total of 240 designs were generated. After observing the design results, they were separated into 3 groups; failed, potential, and successful designs. A design is considered failed when the final value of the objective function was greater than 1, i.e. $E(\mathbf{x}) > 1$. A potential design produced the final value $E(\mathbf{x}) < 1$ and a successful design produced results that satisfied all of the prescribed specifications. The results obtained are summarized in Table 4.5.

One can see that there are fewer failed designs and more potential designs using the coefficient-based objective function as opposed to using the polar-based objective function. However, there was one more successful design when using the polar-based objective function. This suggests that when random initial points are used, the coefficient-based optimization algorithm is more robust than the polar-based algorithm. To get a better idea of how effective the separate designs are, the modified quality factors described in section 4.3.5 were calculated for each design and order. The average quality factor for all 20 designs for each order

Objective Function	Coefficient-Based			Polar-Based		
Order	Fails	Potentials	Successes	Fails	Potentials	Successes
10	4	16	0	9	9	2
12	4	16	0	6	14	0
14	2	15	3	8	10	2
16	9	10	1	13	6	1
18	9	11	0	13	7	0
20	11	9	0	15	5	0
Totals	39 16.25%	77 32.08%	4 1.67%	64 26.67%	51 21.25%	5 2.08%

Table 4.5: Number of failed, potential, and successful designs using random initial points.

is given in Table 4.6, where $\overline{Q}_p, \overline{Q}_s, \overline{Q}_\tau$, and $\overline{Q}_c$ are the passband, stopband, group delay, and composite quality factors, respectively. As previously mentioned, the quality factors are derived with respect to the prescribed specifications, where a quality factor equals zero if the corresponding specification is satisfied and the range can be $0 \leq \overline{Q} \leq 100$.

From Table 4.6 it can be seen that on average, the designs using the coefficient-based objective function produced better quality filters with respect to the specifications. Plots for each quality factor vs. the filter order are illustrated in Figure 4.4 where the dashed-dotted curve represents the designs using the coefficient-based objective function and the solid curve represents the designs using the polar-based objective function.

At this point it can be concluded that the coefficient-based objective function produced better results for all of the design groups from an average point of view. Alternatively, it was observed that the quality of the potential and successful designs using the polar-based objective function were better. This was due to the number of failed attempts because the quality factors for the failed attempts are substantially larger and, therefore, bias the averages. Moreover, since the optimization algorithm will be tested with efficient initial points, these failed attempts are less likely to occur. Therefore, by investigating the quality

Objective Function	Coefficient-Based				Polar-Based			
Order	$\overline{Q}_p$	$\overline{Q}_s$	$\overline{Q}_\tau$	$\overline{Q}_c$	$\overline{Q}_p$	$\overline{Q}_s$	$\overline{Q}_\tau$	$\overline{Q}_c$
10	18.39	4.24	58.86	27.16	37.78	6.07	50.85	31.56
12	16.34	9.53	54.18	26.68	24.47	16.15	49.02	29.88
14	1.25	12.21	40.80	18.09	37.53	12.32	54.69	34.85
16	39.87	6.95	68.69	38.50	57.40	8.98	69.58	45.32
18	35.76	8.10	49.39	31.08	48.18	14.64	57.16	39.99
20	49.26	14.40	69.14	44.27	70.10	14.69	81.10	55.30

Table 4.6: Average values of the modified quality factors obtained using random initial points.

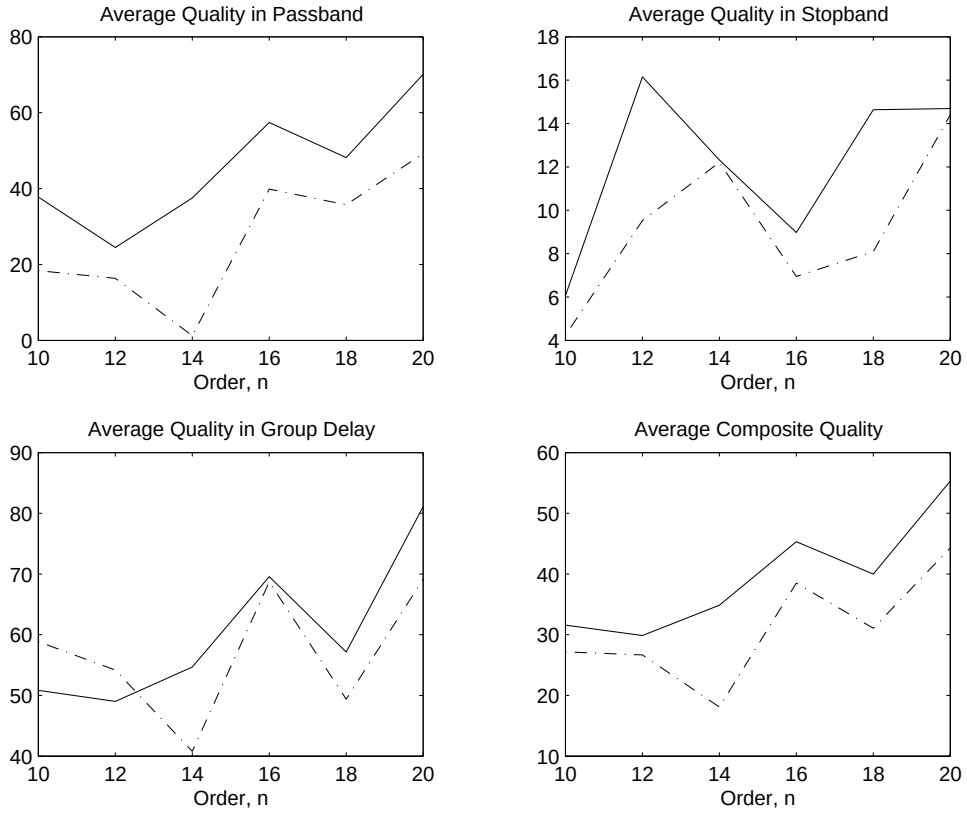

Figure 4.4: Plots of the data given in Table 4.6: $(\cdot - \cdot)$ coefficient-based, $(-)$ polar-based.

of only the potential and successful designs, a better idea as to how effective the optimization methods are can be obtained.

All of the average quality factors for the potential and successful designs are summarized in Table 4.7 and corresponding plots are shown in Figure 4.5. As one can see, the average quality factors for the designs using the polar-based objective function are usually better. Moreover, the average group delay quality factors are lower for all filter orders. Also, the passband specifications were satisfied for all orders except $n = 10$. This suggests that the polar-based objective function produces better results with respect to the passband.

Another interesting observation is the program execution time. At first, it was expected that the polar-form objective function would take longer to execute due to the nonlinearities introduced in the equation by the cosine and sine functions. After further investigation, this turned out not to be the case. The total number of iterations and execution times for the designs are given in Table 4.8. In this table, the average number of iterations and execution times as well as the execution time per iteration is given. It can be seen that the average number of iterations and average execution times for the polar-based objective function are lower than those for the coefficient-based objective function for all orders except $n = 18, 20$, and it is the same for $n = 16$. The execution time per iteration for the designs is plotted in Figure 4.6.

The polar-based objective function becomes marginally more computational expensive for higher-order filter designs suggesting that the time spent evaluating trigonometric functions becomes more dominant. For lower orders, the computational cost may be lower because of the additional linear constraints required for the coefficient-based objective function.

Objective Function	Coefficient-Based				Polar-Based			
Order	$\overline{Q}_p$	$\overline{Q}_s$	$\overline{Q}_\tau$	$\overline{Q}_c$	$\overline{Q}_p$	$\overline{Q}_s$	$\overline{Q}_\tau$	$\overline{Q}_c$
10	2.34	3.08	57.00	20.81	8.08	5.47	29.00	14.18
12	1.47	3.88	49.61	18.32	0.00	1.93	37.03	12.98
14	1.39	3.81	45.34	16.85	0.00	4.40	26.16	10.19
16	0.00	2.53	49.83	17.45	0.00	2.10	31.86	11.32
18	3.22	4.67	41.94	16.61	0.00	4.12	27.44	10.52
20	3.36	2.41	49.60	18.46	0.00	2.81	39.82	14.21

Table 4.7: Average values of the modified quality factors obtained for the potential and successful designs using random initial points.

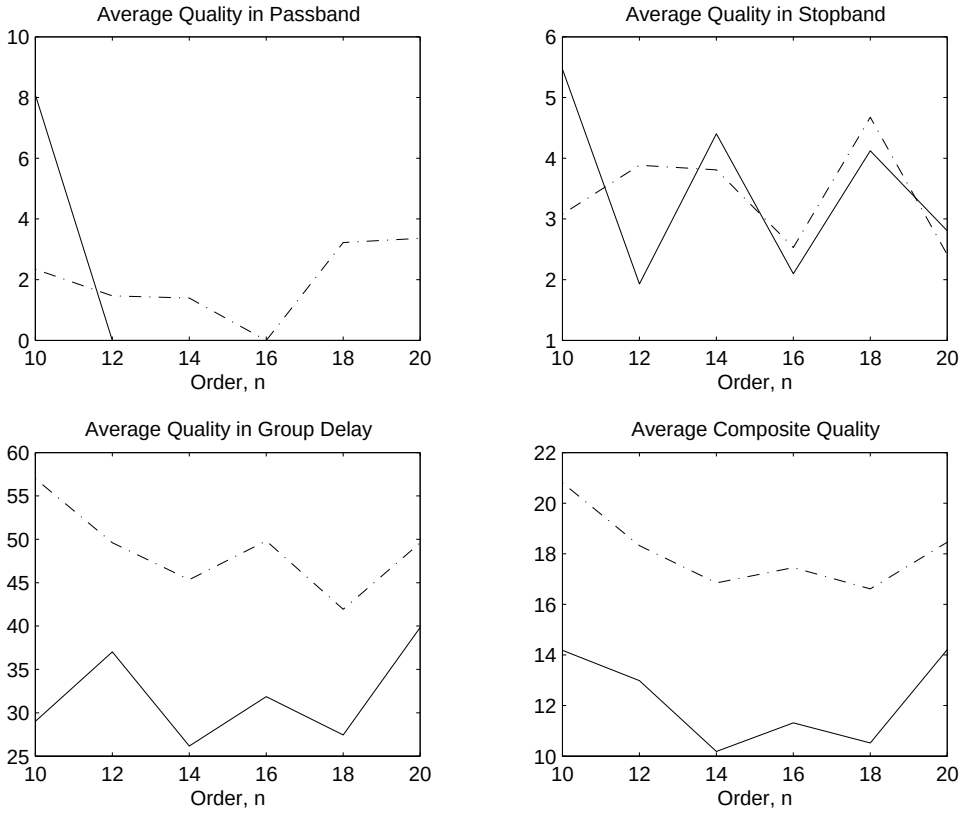

Figure 4.5: Plots of the data given in Table 4.7: $(\cdot - \cdot)$ coefficient-based, $(-)$ polar-based.

Objective Function	Coefficient-Based					Polar-Based				
Order	Iterations		Time, s		Execution Time per Iteration	Iterations		Time, s		Execution Time per Iteration
	Avg.	Total	Avg.	Total		Avg.	Total	Avg.	Total	
10	284.40	5688	102.52	2050.39	0.36	248.05	4961	80.99	1619.72	0.33
12	268.45	5369	111.07	2221.34	0.41	276.30	5526	109.00	2180.02	0.39
14	255.70	5114	121.35	2426.92	0.47	277.85	5557	127.23	2544.53	0.46
16	282.20	5644	151.15	3022.97	0.54	280.05	5601	151.12	3022.47	0.54
18	259.25	5185	155.27	3105.49	0.60	255.60	5112	160.66	3213.13	0.63
20	289.15	5783	194.60	3892.02	0.67	294.6	5892	212.85	4256.94	0.72

Table 4.8: The computational data for the designs.

Summarizing the results obtained, when using random initial points, the polar-based objective function produces better quality filters with respect to the passband and requires less computational cost for lower-order designs. On the other hand, the coefficient-based objective function seems to be more robust to random initial points and requires less computation for higher-order filter designs.

4.3.6 Design Example Using the Method by Trends

In light of the previous section, the optimization algorithm had difficulty shifting poles and zeros to the optimal formations and, therefore, the need for efficient initial points is required. The method proposed by Lu in [6] was developed by observing the zero/pole formations for several linear-phase recursive digital filter designs. Although, the optimization algorithm in [6] uses an unconstrained optimization technique, the final zero/pole formations are similar. The algorithm for generating these initial points for a lowpass or highpass digital filter is given in Appendix A.1.

A lowpass filter with the specifications given in Table 4.9 was designed using the coefficient-based and polar-based objective functions. In the following sections, several designs were carried out using each objective function with certain modifications as to produce the best design.

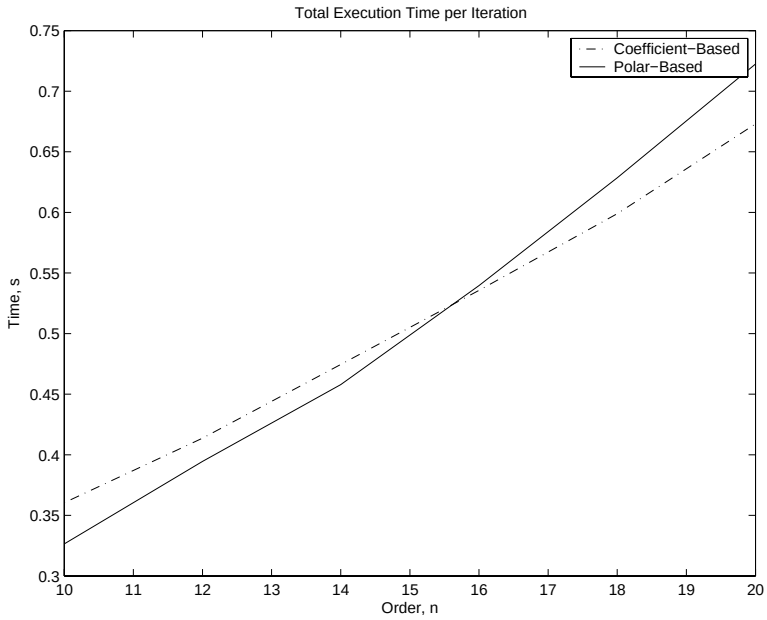

Figure 4.6: Execution time per iteration vs filter order for coefficient-based and polar-based objective functions.

Parameter	Value
Filter order, n	10
Sampling frequency, ω_s, rad/s	2π
Maximum passband ripple, A_p, dB	0.1
Minimum stopband loss, A_a, dB	40
Passband edge, ω_a, rad/s	0.2π
Stopband edge, ω_p, rad/s	0.3π
Maximum standard deviation in the group delay, %	6

Table 4.9: Lowpass digital filter specifications for the example using initial points from the method of trends.

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.05
Real pole boundary margin, δ_m	0.05
Step limit, β_{c_i}	0.03
Step limit, β_{H_0}	0.03
Step limit, β_τ	0.1
Initial delay, τ_0 (samples)	17

Table 4.10: Algorithm design parameters for the coefficient-based objective function using initial points from the method by trends.

Coefficient-Based Objective Function

For the coefficient-based objective function, the optimization algorithm seemed to be very sensitive to the initial delay parameter τ_0. If τ_0 was initially small, the resulting design exhibited a large spike in the transition band and in most cases a solution was not found. If τ_0 was initially large, the resulting design was not even close to the required design and barely resembled a lowpass filter. It was found that the initial delay parameter should be in the range $n \leq \tau_0 \leq 2n$. For this design example, $\tau_0 = 17$ was used. Even with a better initial delay parameter, a spike in the transition band still occurred but was not as prominent. Another way to suppress this spike is to increase the stability margin variable to force the poles away from the unit circle. As mentioned in [6], for constant delay filters this spike is caused by high sensitivity to one or more poles located close to the unit circle. Therefore, it is reasonable to assume that the amplitude of the spike decreases when δ_s increases. In fact, this was found to be the case but as a result the optimization would have difficulty satisfying the delay specification. Also, this transition spike can be suppressed by simply increasing the filter order. The algorithm parameters that produced the best design for this example are given in Table 4.10 and the weighting values were the same as in Eq. 4.1. Also, the step limit modifications explained in section 4.3.3 were applied for this example.

When using initial points generated with the method by trends, the passband boundary

constraints are no longer needed since the initial pole positions are already located inside the passband sector of the z plane. Therefore, these constraints were eliminated for this design example.

Another small modification that seemed to limit the transition band artifacts was using a slightly larger value for ω_p in step 2 of Algorithm 1 given in Appendix A.1. A value of $\omega_p = 0.22\pi$ was used for the method by trends and a value of $\omega_p = 0.2\pi$ was used in the optimization. This provided an overestimation for the passband cut-off frequency and resulted in a better magnitude response in the transition band.

Polar-Based Objective Function

Like the coefficient-based optimization, the polar-based optimization is sensitive to the initial delay parameter. If the initial delay parameter was far from the final delay, the resulting design produced a large peak in the magnitude response inside the transition band. The optimization algorithm seemed to have difficulty moving the zeros into their optimal formations. However, the optimization seemed to efficiently shift the poles into their correct formation or close to it. This suggests that the zero positions are more sensitive than the pole positions to the initial delay parameter. As in the coefficient-based design, the initial delay parameter produced the best results when $n \leq \tau_0 \leq 2n$. The algorithm parameters that produced a good design for this example are given in Table 4.11 and the weighting scheme given in Eq. 4.1 was used.

As one can see, the stability margin was slightly decreased which resulted in more flexibility in the optimization. As previously mentioned, when the poles move close to the unit circle a spike may form in the transition band or near the passband edge but for the polar-based objective function that was not the case. When the stability margin was decreased, the optimization had more flexibility to shift the poles and zeros to better positions so as to minimize error spikes in the transition band.

Another parameter modification involved the step limits. The step limits for the polar-based design were increased slightly compared to those in the coefficient-based design given in section 4.3.3 as follows:

If $|Q| > Q_c$:

If $[(E_i(\mathbf{x}) > 1)$ and $(r < 0.1)]$:

$$\beta_{c_i}^{(k)} = 2\beta_{c_i}^{(k-1)}$$
$$\beta_{H_0}^{(k)} = 2\beta_{H_0}^{(k-1)}$$
$$\beta_{\tau}^{(k)} = 2\beta_{\tau}^{(k-1)}$$

Else If $[(E_i(\mathbf{x}) < 1)$ and $(r < 0.01)]$:

$$\beta_{c_i}^{(k)} = \beta_{c_i}^{(k-1)} + R_{\beta}(\beta_{c_i}^{(k-1)})$$
$$\beta_{H_0}^{(k)} = \beta_{H_0}^{(k-1)} + R_{\beta}(\beta_{H_0}^{(k-1)})$$
$$\beta_{\tau}^{(k)} = \beta_{\tau}^{(k-1)} + \tfrac{1}{2}R_{\beta}(\beta_{\tau}^{(k-1)})$$

This modification reduced the number of iterations required to solve the problem. A suitable value for the predefined quality factor was $Q_c = 3$. This modification also contributed to a faster rate of convergence by doubling the step limits if the objective function evaluation was greater than 1 with little or no progress from the previous iteration.

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.04
Step limit, $\beta_{\hat{p}_i}$	0.05
Step limit, β_{H_0}	0.05
Step limit, β_{τ}	0.1
Initial delay, τ_0 (samples)	18

Table 4.11: Algorithm design parameters for the polar-based objective function using initial points from the method by trends.

Results

The design results are summarized in Table 4.12 and several magnitude response plots as well as the group delay plot are illustrated in Figure 4.7. As shown in Table 4.12, the polar-based design required 100 fewer optimization iterations than the coefficient-based design resulting in a savings of about 60% to complete the design. Furthermore, the resulting quality factor is much better as illustrated in Figure 4.9. In Figure 4.7, one can see that the coefficient-based design produced a slightly lower minimum stopband attenuation and a lower group delay across the passband. However, the coefficient-based design exhibited a large magnitude response spike in the transition band and the group delay deviation within the passband was larger than the polar-based value. This was due to the inefficient placement of the phase zeros in the passband region. As can be seen in Figure 4.8, one pair of the phase zeros from the coefficient-based design was placed on the real axis. Therefore, it was difficult for the algorithm to simultaneously reduce the error spike in the transition band and linearize the phase response near the passband edge. As explained in Chapter 3, the phase zeros are used to balance the polar delay ratios for both the zeros and poles so as to enforce a constant group delay in the passband region. This becomes difficult if the zeros are not placed where the group delay peaks occur from the polar delay ratio contributions. As can be seen in Figure 4.8, for the polar-based design, the algorithm placed the phase zeros in an evenly spaced formation located near these distinct peaks.

Another interesting test was letting the polar-based optimization proceed with a lower group delay deviation specification. This resulted in a lower group-delay standard deviation in the passband of around 1%, taking about 120 iterations. A similar test was done with the coefficient-based objective function but the standard deviation in the group delay did not improve. This suggests that the coefficient-based design produces slightly better results with respect to the magnitude response whereas the polar-based design produces better results with respect to the group delay.

Objective Function	Coefficient-Based	Polar-Based
Total optimization iterations	181	81
Total recorded time, s	99.52	39.42
Number of M-functions	50	50
Number of M-subfunctions	20	20
Number of MEX-functions	1	1
Maximum passband ripple, dB	0.068849	0.063749
Minimum stopband attenuation, dB	43.1489	40.2253
Standard deviation of group delay in passband, %	5.9619	3.2687
Composite filter quality	1.5997	0.9074
Objective function evaluation	7.4783×10^{-4}	1.0447×10^{-3}
Passband group delay, s	16.7773	17.7841

Table 4.12: Design results using initial points generated with the method by trends.

The filter coefficients for the design using the coefficient-based objective function are given in Table B.1 and the radii and angles for the design using the polar-based objective function are given in Table B.2 in Appendix B.

Summarizing the results obtained, the polar-based design produced a better filter in less time when compared with that obtained by using the coefficient-based design. Also, there seems to be a slight tradeoff between the two designs; the coefficient-based design produces a slightly better magnitude response and worse passband group delay response, whereas the polar-based design produces a slightly worse magnitude response and a better passband group delay response. Furthermore, this slight tradeoff is not as apparent if the polar-based design is allowed to continue and, in some cases, the design produced better results in both the magnitude and group delay responses.

4.3.7 Design Examples Using the BMT Method

In this section, the same filter is designed using the BMT method described in section 2.7.2. The method by trends is a fairly good initialization technique in terms of placing the zeros

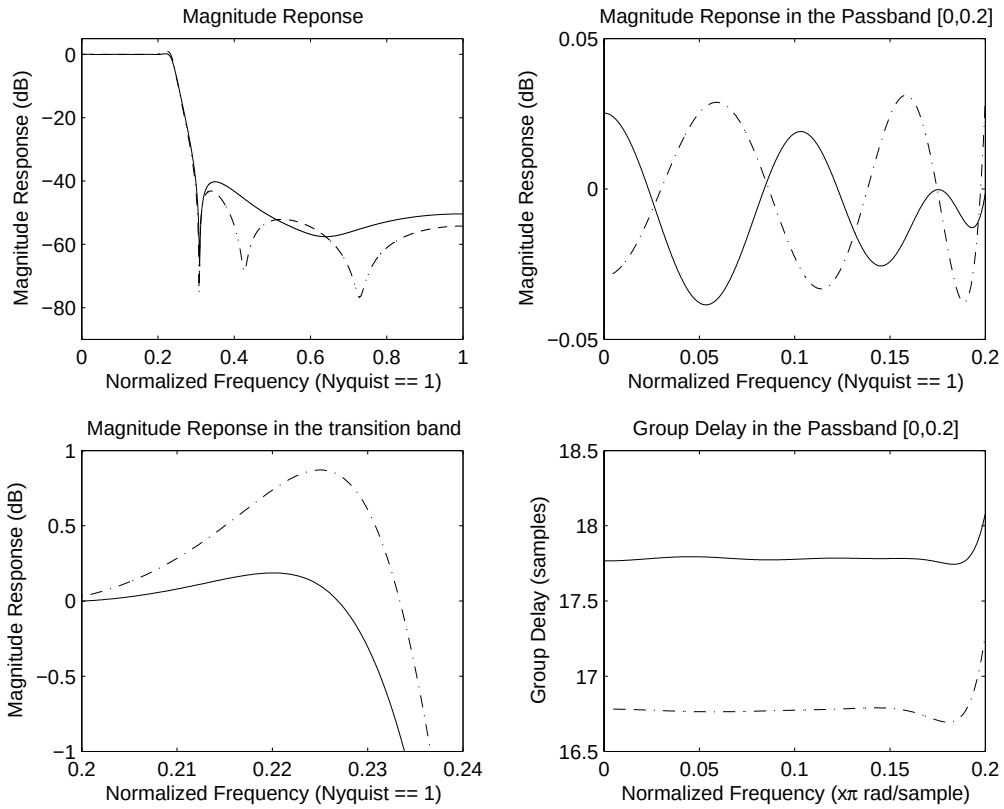

Figure 4.7: Magnitude response and group delay using initial points generated with the method by trends. $(\cdot - \cdot)$ coefficient-based, $(-)$ polar-based.

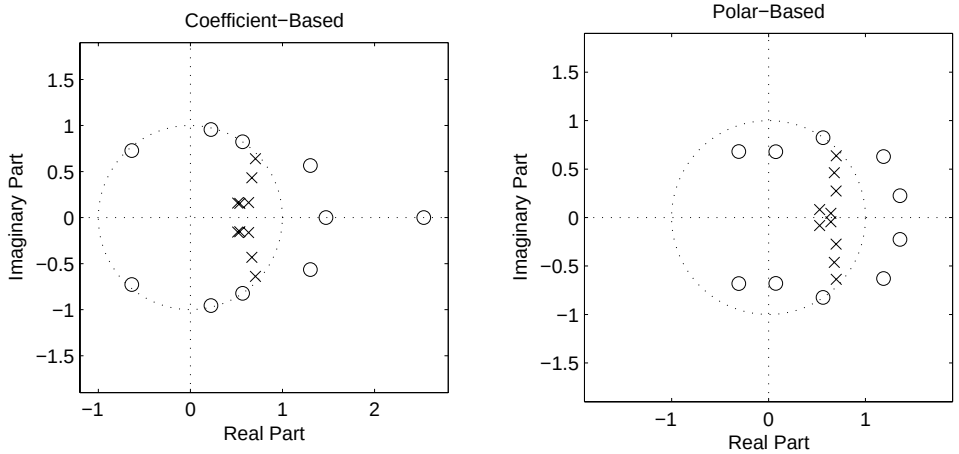

Figure 4.8: Zero and pole plots using initial points generated with the method by trends.

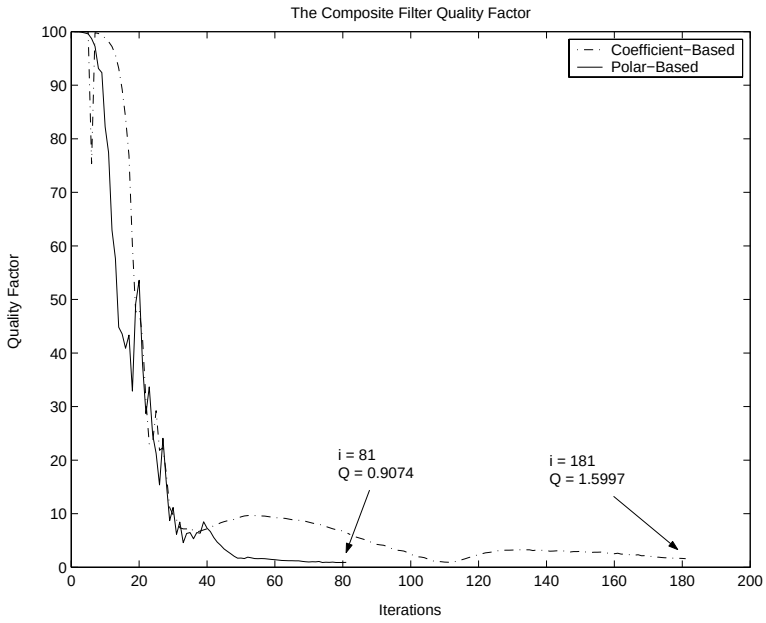

Figure 4.9: Composite quality factor vs optimization iteration using initial points generated with the method by trends.

and poles in the vicinity of their final optimal locations but the BMT method provides improved initial points.

In both the coefficient-based and polar-based designs, the initial delay parameter is assigned the average value of the group delay across the passband. This provides an excellent estimate because the group delay has already been semi-linearized by the BMT method and, therefore, the final delay will be fairly close to the optimal value.

For both designs, the passband boundary constraints were eliminated because the BMT method already places the poles within the passband sector of the z plane. Also, the weighting scheme given in Eq. 4.1 was used in both designs.

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.05
Real pole boundary margin, δ_m	-0.05
Step limit, β_{c_i}	0.03
Step limit, β_{H_0}	0.03
Step limit, β_τ	0.1
Initial linear-phase FIR order, n	34

Table 4.13: Algorithm design parameters for the coefficient-based objective function using initial points obtained with the BMT method.

Coefficient-Based Objective Function

When using the BMT method, the initial filter was designed as a high order linear-phase FIR filter. The FIR design algorithm used with the BMT method is described in section 2.7.2 and was used for several design attempts. Unfortunately, all these attempts produced an undesired magnitude response spike in the transition band. However, it was found that another linear-phase FIR design technique that uses least-squared error minimization produced better results with the coefficient-based design. The MATLAB function for this technique is `firls` and this function designs a linear-phase FIR filter that minimizes the weighted, integrated squared error between an ideal piecewise linear function and the magnitude response of the filter over a set of desired frequency bands. The theoretical approach on which `firls` is based is described in [16]. The algorithm parameters used for this example are given in Table 4.13 and the step limit modifications explained in section 4.3.3 were applied.

Polar-Based Objective Function

In the polar-based design, the initial linear-phase FIR filter was designed using the multiple exchange algorithm described in section 2.7.2. Also, the passband and stopband frequencies

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.05
Step limit, $\beta_{\hat{p}_i}$	0.03
Step limit, β_{H_0}	0.03
Step limit, β_τ	0.1
Initial linear-phase FIR order, n	30

Table 4.14: Algorithm design parameters for the polar-based objective function using initial points from the BMT method.

were slightly modified resulting in a better frequency response. The modification is as follows

$$\hat{\omega_p} = \omega_p + 0.25(\omega_a - \omega_p), \quad \hat{\omega_a} = \omega_p - 0.25(\omega_a - \omega_p)$$

where $\hat{\omega_p}$ and $\hat{\omega_a}$ are the passband and stopband frequencies used for the linear-phase FIR design technique respectively. The algorithm parameters used for this example are given in Table 4.14 and the step limit modifications explained in the previous section for the polar-based objective function were applied.

Results

The design results are summarized in Table 4.15 and several magnitude response plots as well as the group delay plot are illustrated in Figure 4.10. As for the previous example, the magnitude response using the coefficient-based objective function was slightly better. On the other hand, the group delay obtained using the polar-based objective function is about a full sampling period lower and has a smaller standard deviation as can be seen in Figure 4.10. Also the magnitude response spike that usually occurs in the transition band is not present in either of the designs. Furthermore, the zeros and poles are located in their optimal formations for both designs as illustrated in Figure 4.11. The variation in the filter quality vs iterations is plotted in Figure 4.12. As can be seen in Table 4.15, the polar-based design

Objective Function	Coefficient-Based	Polar-Based
Total optimization iterations	48	27
Total recorded time, s	25.95	13.70
Number of M-functions	76	78
Number of M-subfunctions	32	32
Number of MEX-functions	1	1
Clock precision, s	5.0×10^{-8}	5.0×10^{-8}
Clock Speed, MHz	2000	2000
Maximum passband ripple, dB	0.020761	0.079843
Minimum stopband attenuation, dB	44.3652	40.0546
Standard deviation of group delay, %	5.9474	5.1837
Composite filter quality	1.3221	0.56349
Objective function evaluation	3.4978×10^{-4}	2.0367×10^{-3}
Passband group delay, s	18.0908	16.8271

Table 4.15: Design results using initial points generated with the BMT method.

required 21 fewer iterations resulting in a savings of 12.25 s to satisfy the specifications.

The filter coefficients for the design using the coefficient-based objective function are given in Table B.3 and the radii and angles for the design using the polar-based objective function are given in Table B.4 in Appendix B.

Summarizing the results obtained, the design using the polar-based objective function produced a better quality filter in less time than the coefficient-based objective function. Furthermore, the group delay was decreased as well as the group delay deviation. Moreover, the BMT method produced improved initial points compared to the method by trends. Based on the results given in the previous three sections of this chapter, the polar-based objective function will be used in the examples and comparisons of Chapter 5.

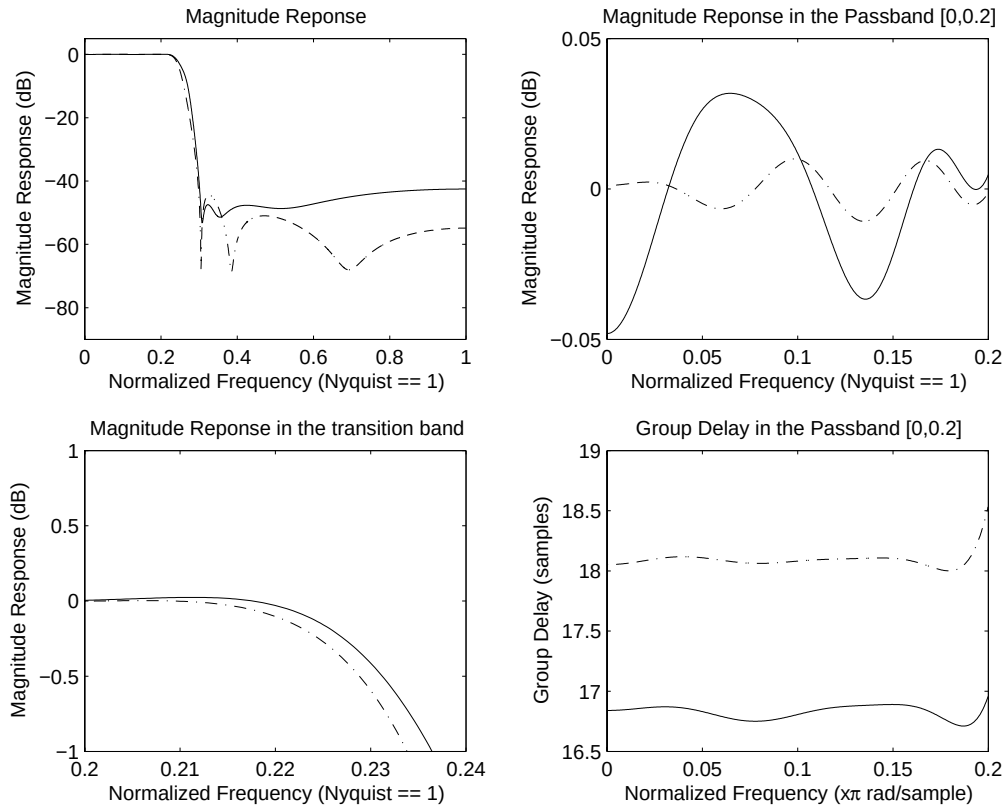

Figure 4.10: Magnitude response and group delay using initial points generated with the BMT method. $(\cdot - \cdot)$ coefficient-based, $(-)$ polar-based.

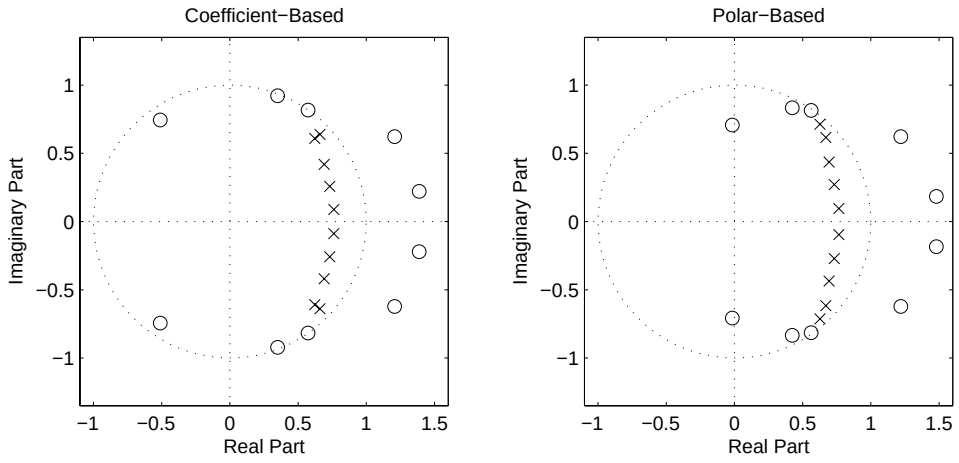

Figure 4.11: Zero and pole plots using initial points generated with the BMT method.

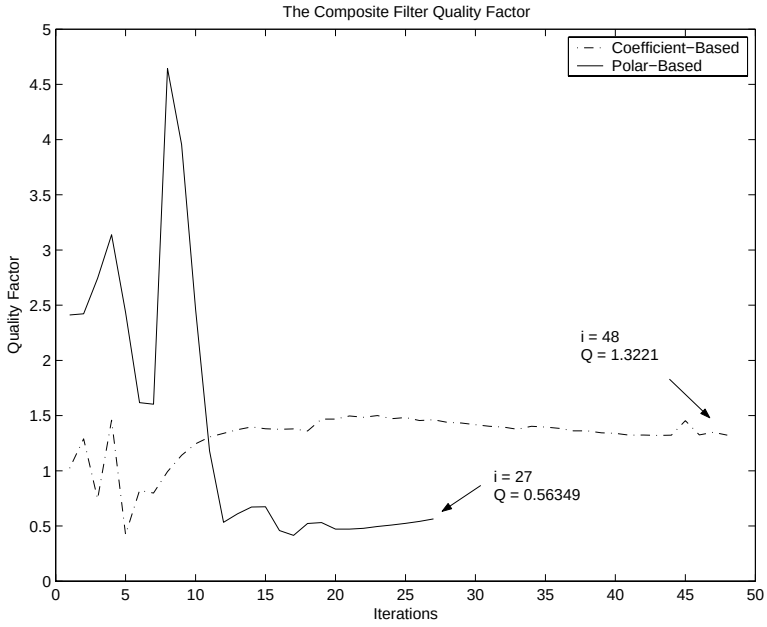

Figure 4.12: Composite quality factor vs optimization iteration using initial points generated with the BMT method.

4.4 Conclusions

The nonuniform sampling technique explained in section 3.5 was investigated. It was found that this technique produced the desired results in less time than when using uniform sampling. Furthermore, the artifacts in the transition band were slightly decreased and the overall composite quality factor was improved.

The two proposed objective functions were used to design several lowpass IIR filters and compared in terms of magnitude response and computational cost. Testing the objective functions with random initial points was particularly difficult due to the highly nonlinear characteristic of the objective functions. The polar-based objective function produced better filters for the potential and successful designs. On the other hand, the coefficient-based function required less computational cost for higher-order designs and seemed to be more robust to random initial points.

The two objective functions were also tested using a method based on observing several zero and pole formations for linear-phase IIR filter designs reported in [6] which were designed using the algorithm given in Appendix A.1. For this comparison, the filter designed using the polar-based objective function had a better quality factor and required less time to complete.

The last design comparison used initial points generated by a modern technique that converts a high-order linear-phase FIR filter into a nearly linear-phase IIR filter. The resulting designs revealed the benefits of using the polar-based objective function. The group delay deviation and the overall group delay were reduced; in addition, the design time was reduced and a better filter quality was achieved.

This chapter also dealt with several algorithm modifications that tend to improve the quality of the filter designed such as the step limit modifications, and the initial frequency modifications for the BMT method.

Chapter 5

Examples and Comparisons

> Never underestimate the power of an idea whose time has come.
>
> —Ari Knazan

5.1 Introduction

In this chapter, three modern techniques are investigated and compared with the proposed method. First, an optimal equalization technique that uses a prescribed group delay to equalize an elliptic filter design in the passband region is investigated. Next, a recent technique that can be used to design IIR filters with robust stability using conic quadratic programming (CQP) updates is compared with the proposed method. The last example uses parameterization of Schur polynomials to guarantee filter stability in an unconstrained optimization technique. All designs were performed on a laptop computer having the specifications given in Table 4.2.

5.2 Equalizer Design

In this section, filter designs using the proposed method are compared with filter designs using an efficient modern equalization technique. Nearly linear-phase digital filters can be achieved by initially designing an IIR filter that satisfies the magnitude response specifications ignoring the group delay. Then a recursive delay equalizer that is cascaded with the initial IIR filter to compensate for variations in the group delay of the passband is designed [1]. There are several optimization techniques that can be used to design equalizers and the method of interest is one implemented in terms of the program `iirgrpdelay` which is available in the MATLAB DSP Toolbox [17]. This program can be used to design an optimal allpass IIR filter with a prescribed group delay. The equalizer is then cascaded with an IIR filter to provide the overall linear-phase digital filter.

Filter requirements often call for highly selective filters, especially in bandpass filters designed to reject out-of-band carriers. If the cutoff-rate specification is stringent, the classical Butterworth and Chebyshev filters result in high orders. A higher order adds complexity where the resulting design is more difficult to tune. Furthermore, the sensitivity of the filter to its components also increase. For this reason, an elliptic filter is used for the initial IIR filter. This filter has the lowest-order of any of the classical filters for the same frequency and rejection requirements and, in addition, it has an equiripple magnitude response with respect to the passband and stopband [1].

The elliptic filter that satisfies the magnitude response specifications is designed using the command `ellip` in MATLAB. The design is performed with a given low-order and then the order is incremented until the rejection requirements are satisfied. Then the group delay of the elliptic filter is determined and used to create the desired group delay response for the `iirgrpdelay` function. This is done by creating group delay specifications which would complement the elliptic filter's group delay to obtain a constant group delay. Then the equalizer is designed using this mirrored copy of the group delay. In this way, when the

equalizer and elliptic filter are cascaded, the group delays of the filter and equalizer will be added yielding the desired equalized group delay. During this design process, the equalizer order is incremented until the required standard deviation of the group delay is achieved.

It should be noted that the value $p = 2$ was set in the least-pth algorithm of `iirgrpdelay` for a fair comparison with the proposed method.

5.2.1 Example 1

The first example will be the lowpass filter design investigated in the sections 4.3.6 and 4.3.7 where the specifications are given in Table 4.9. The results produced from the design using the polar-based objective function in section 4.3.7 are used for comparisons for this example.

Results

The design results are summarized in Table 5.1 and the magnitude response of the filter-equalizer combination is illustrated in Figure 5.1. As one can see, the equalizer method produced a slightly better quality factor than the proposed method. Furthermore, it required fewer function evaluations and was 62 times faster to design. On the other hand, the filter designed using the proposed method satisfied the specifications with a 10th-order transfer function whereas the equalizer method required a 15th-order transfer function. As a result, the group delay achieved with the proposed method was reduced by about 26 s as can be seen in Figure 5.1. The separate group delay plots are depicted in Figure 5.1, where the y-axis scales are the same. The zero/pole plots for the two designs are illustrated in Figure 5.2.

For this design example, the proposed method produced a lower error magnitude within the passband and about equal error in the stopband as shown in Figure 5.3. However, the equalizer design produced a better response in the transition band in terms of the error. To

Objective Function	Equalizer Method	Proposed Method
Elliptic filter order	5	NA
Equalizer filter order	10	NA
Overall filter order	15	10
Total recorded time, s	0.22	13.70
Iterations	N/A	27
Number of M-functions	52	78
Number of M-subfunctions	24	32
Number of MEX-functions	3	1
Maximum passband ripple, dB	0.099996	0.079843
Minimum stopband attenuation, dB	40.0025	40.0546
Standard deviation of group delay in passband, %	4.3431	5.1837
Composite filter quality	0.45705	0.56349
Passband group delay, s	42.9018	16.8271
L_2 norm of the error	4.1560	7.0478

Table 5.1: Design results for the equalizer and proposed methods of example 1.

get an idea for the measure of the error plotted in Figure 5.3, the L_2 norm of the error across all frequencies was calculated and is given in Table 5.1[1]. As a result, the overall frequency response error for the equalized filter was slightly lower than that in the proposed method design.

The filter radii and angles for the design using the proposed method are given in Table B.4 and the coefficients for the design using the equalizer method are given in Table B.5 of Appendix B.

Summarizing the results obtained, the equalizer design produced a slightly better filter compared to the proposed method at the cost of a significantly higher order. The goal of the proposed method was to explore the possibility of finding lower-order filters satisfying both the magnitude response and group delay specifications simultaneously when compared with using the equalizer design approach. Also, it is reasonable to say that when using the

[1]The L_2 norm of the error vector e was calculated by treating e as a matrix and then finding the largest singular value of e using the MATLAB command `max(svd(e))`.

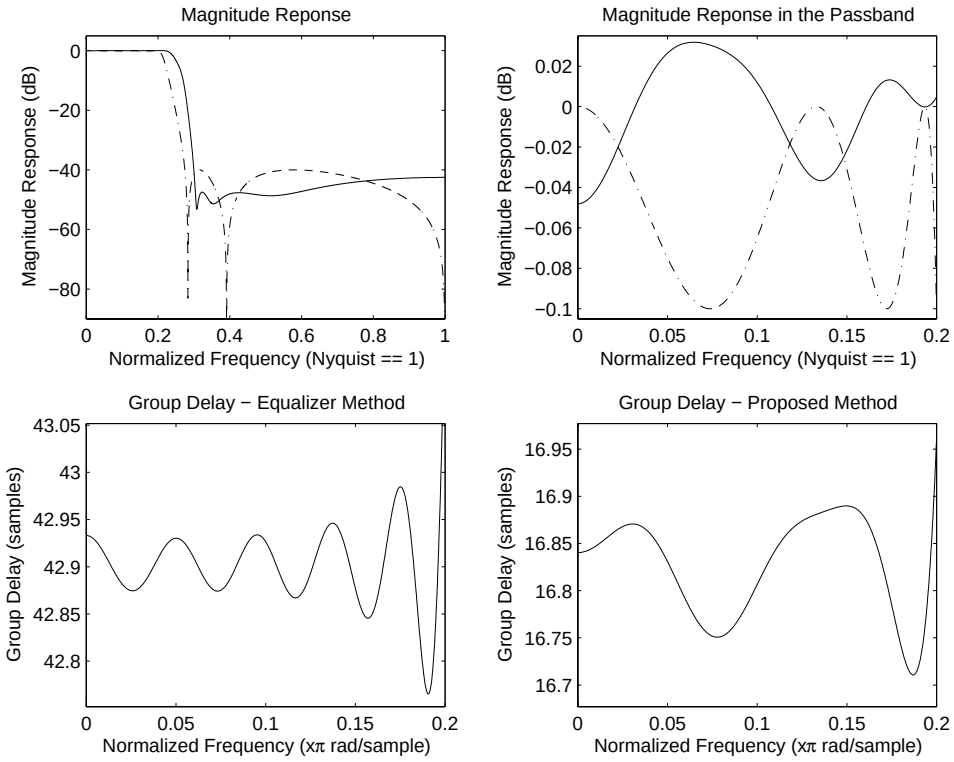

Figure 5.1: The magnitude response and group delay for the equalizer and proposed methods for example 1. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

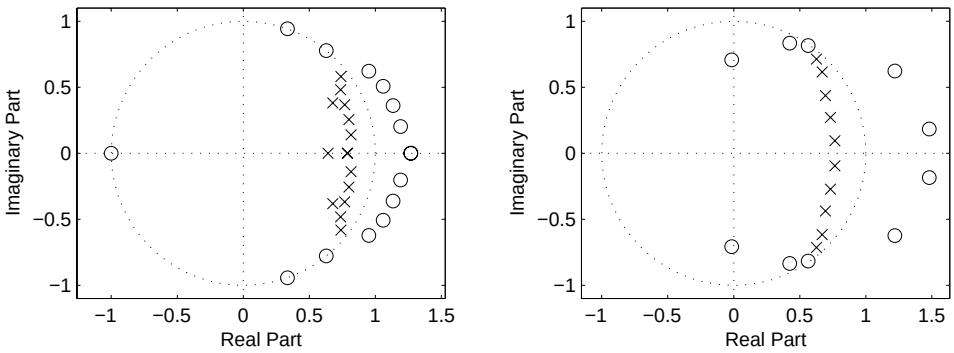

Figure 5.2: The zero and pole plots for the equalizer and proposed methods for example 1.

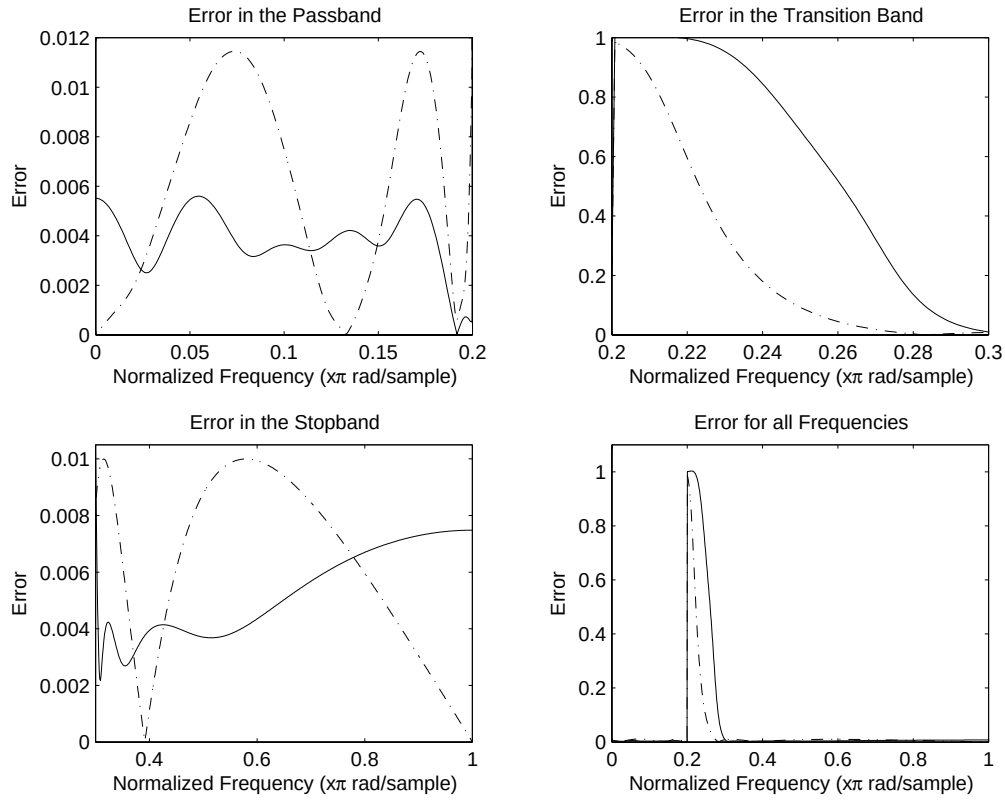

Figure 5.3: The magnitude of the error for the equalizer and proposed methods for example 1. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

proposed method, the quality of the filter will increase when the filter order is increased. Therefore, one could design a 12th or 14th-order filter using the proposed method and produce a better quality filter and still retain a lower order. In fact, this is the case as will be shown in example 3.

5.2.2 Example 2

To get a better idea of how effective the proposed method is, a much more stringent filter design was investigated. The filter specifications for this example are given in Table 5.2. As one can see, the passband is much larger than in the previous example and, therefore, it requires several more phase-zeros to linearize the phase response. Furthermore, the maximum passband ripple is two times smaller and the minimum stopband loss is 5 dB larger than the previous example.

Proposed Method Modifications

The initial passband and stopband frequencies used for the BMT method were slightly modified resulting in a better frequency response. The modification was

$$\hat{\omega_p} = \omega_p + 0.2(\omega_a - \omega_p), \quad \hat{\omega_a} = \omega_p - 0.2(\omega_a - \omega_p)$$

Parameter	Value
Sampling frequency, ω_s	2π rad/s
Maximum passband ripple, A_p	0.05 dB
Minimum stopband loss, A_a	45 dB
Passband edge, ω_a	0.6π rad/s
Stopband edge, ω_p	0.7π rad/s
Maximum standard deviation in the group delay	6%

Table 5.2: Lowpass digital filter specifications used for example 2.

where $\hat{\omega_p}$ and $\hat{\omega_a}$ are the passband and stopband frequencies used for the linear-phase FIR design technique, respectively. The algorithm parameters used for this example are given in Table 5.3 and the step limit modifications explained in section 4.3.6 for the polar-based objective function were applied.

Results

The design results are summarized in Table 5.4 and the magnitude response is illustrated in Figure 5.4. As with the previous example, the equalizer design produced a better quality filter and the design was a lot faster. To satisfy the specifications, the proposed and equalizer methods required a transfer function of order 20 and 28, respectively. This resulted in a reduction of about 18 s in the group delay as shown in Figure 5.4. This is quite substantial if a lower-delay nearly linear-phase digital filter is required for the application at hand.

The quality factor is better for the equalizer design and part of this factor is determined from the magnitude of the error between the filter response and the ideal response. Therefore, the errors within the transition band are taken into account in the composite quality factor. That being considered, if the application of interest does not require minimal error within the transition band, the proposed method design may be more attractive.

In this example, the passband magnitude response for the proposed method was slightly

Parameter	Value
Number of sampling intervals per band	50
Stability margin, δ_s	0.08
Step limit, $\beta_{\hat{p}_i}$	0.03
Step limit, β_{H_0}	0.03
Step limit, β_τ	0.1
Initial linear-phase FIR order, n	33

Table 5.3: Design parameters for the proposed method used in example 2.

Objective Function	Equalizer Method	Proposed Method
Elliptic filter order	6	NA
Equalizer filter order	22	NA
Filter order	28	20
Total recorded time, s	0.44	92.36
Iterations	N/A	65
Number of M-functions	52	80
Number of M-subfunctions	23	32
Number of MEX-functions	3	1
Clock precision, s	5.0×10^{-8}	5.0×10^{-8}
Clock speed, MHz	2000	2000
Maximum passband ripple, dB	0.050000	0.044392
Minimum stopband attenuation, dB	45.0034	46.2442
Standard deviation of group delay in passband, %	4.2565	5.4652
Composite filter quality	0.94123	1.8711
Passband group delay, s	34.8402	17.0818
L_2 norm of the error	4.6477	7.4099

Table 5.4: Design results for the equalizer and proposed methods for example 2.

better than that for the equalizer design, as illustrated in Figure 5.4. Furthermore, the group delay was better throughout the passband with the exception of a few samples near the passband edge. For comparative purposes, the zero/pole plots and error magnitude plots are shown in Figures 5.5 and 5.6, respectively.

An interesting observation involved the initial linear-phase FIR filter order when using the proposed method. When the order is increased, the resulting group delay is also increased. This is interesting because the final order is still the same, but the group delay increases. This may be due to an inherited trait from the BMT method. It is known that the group delay increases as the order increases for linear-phase FIR filters [1][2]. Therefore, since the initial delay parameter is approximated from the average group delay of the initial points generated from the BMT method, this value tends to be larger if the initial order is large. So, it would be beneficial to find the lowest initial order for the BMT method that will produce good initial points so as to minimize the resulting group delay of the designed filter.

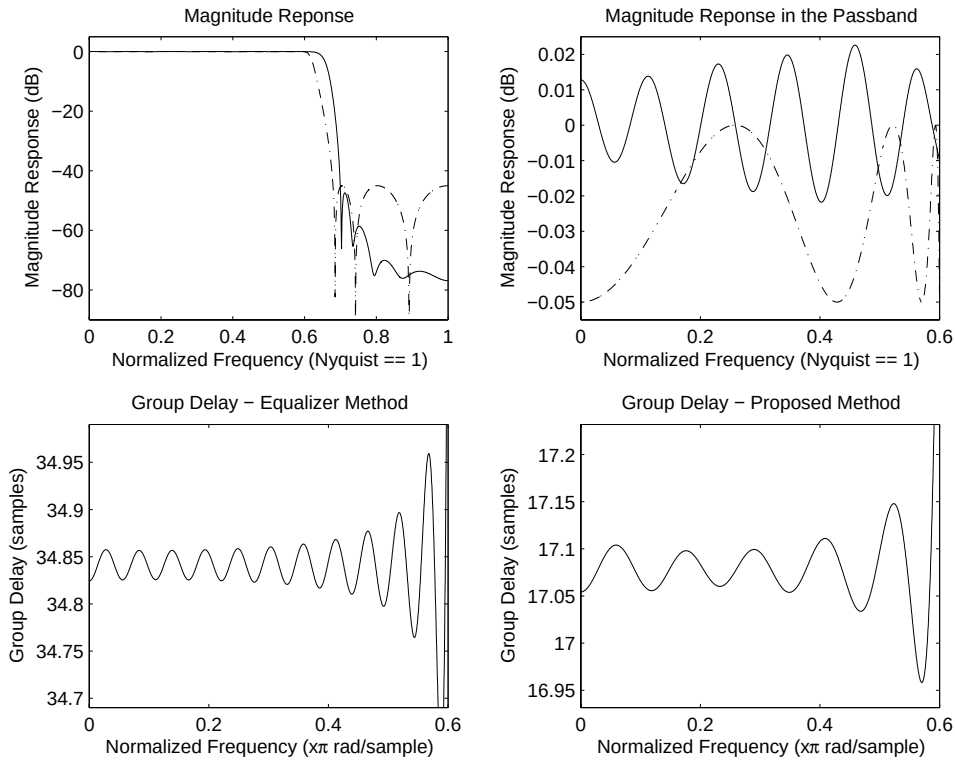

Figure 5.4: Magnitude response and group delay for the equalizer and proposed methods for example 2. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

The filter radii and angles for the design using the proposed method are given in Table B.6 and the coefficients for the design using the equalizer method are given in Table B.7 of Appendix B.

Summarizing the results obtained, the equalizer design produced a better quality filter in far less time than the design using the proposed method. However, the design using the proposed method required a 20th-order as opposed to a 28th-order transfer function in the case of the equalizer design approach. The lower order decreased the group delay from 35 to 17 s. Furthermore, the magnitude response in the passband and stopband is slightly better for the proposed method design.

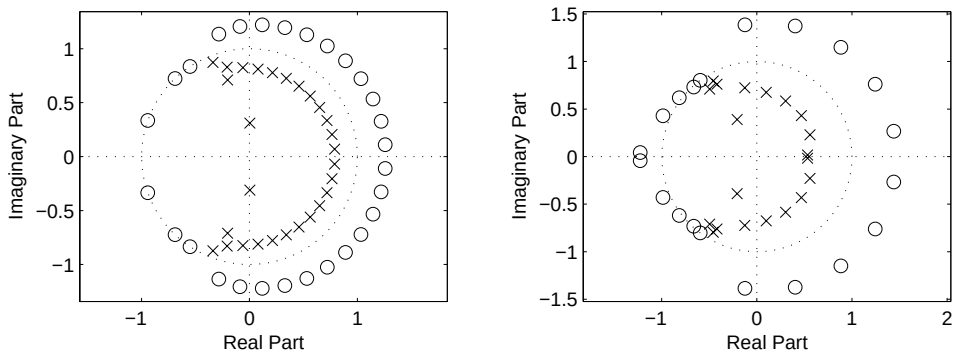

Figure 5.5: Zero and pole plots for the equalizer and proposed methods for example 2.

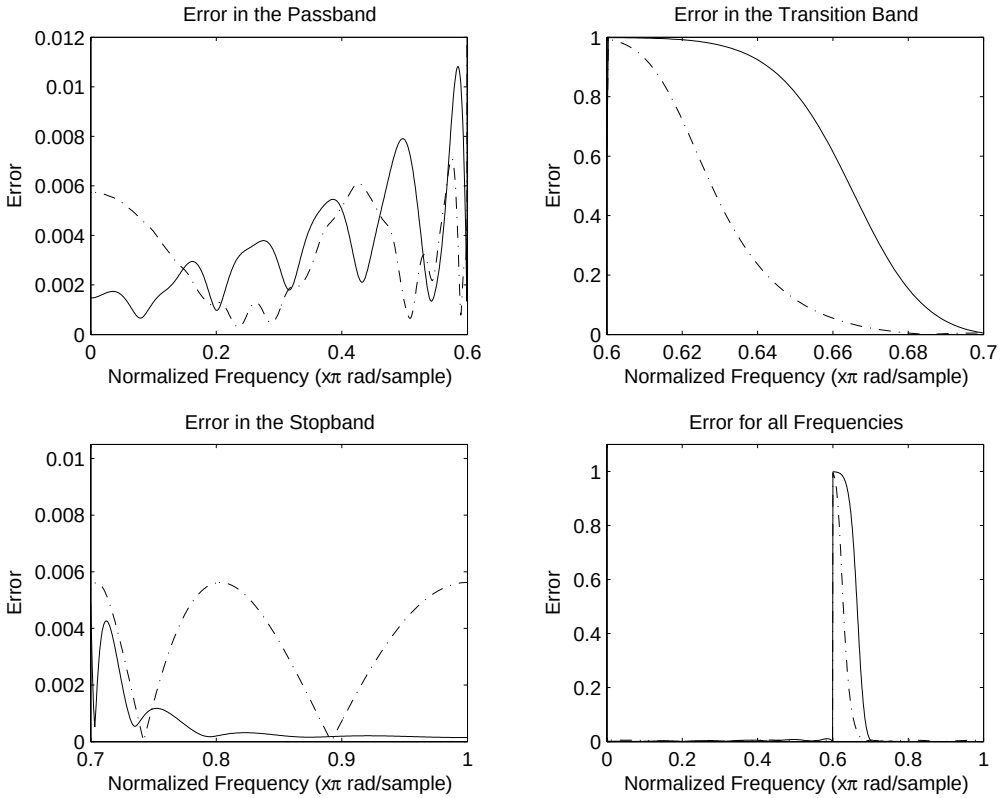

Figure 5.6: Magnitude of the error for the equalizer and proposed methods for example 2. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

5.2.3 Example 3

Example 3 is the same as the previous example except that the order of the transfer function designed for the proposed method was increased from 20 to 26. As mentioned from the case of example 1, if the order is increased, it is reasonable to expect that the quality of the filter should also increase. In this example, the goal is to show that the proposed method can produce a filter of equal or better quality with a lower order.

Some slight modifications in the design parameters were made in the BMT method before using the proposed method. First, there was difficulty satisfying the maximum passband ripple specification. To overcome this difficulty, a smaller value was used for the design of the linear-phase FIR filter to overcompensate, thereby making it easier for the algorithm to converge. Instead of $A_p = 0.05$ dB, $A_p = 0.01$ dB was used to design the FIR filter with the BMT method. In Addition, the passband frequency was set to $w_p = 0.65\pi$ for the initial FIR filter design to provide an overestimate which resulted in a better magnitude response near the passband edge. Also, the initial FIR filter order was set to $n = 56$.

Results

The results are given in Table 5.5, the magnitude response and group delay plots are shown in Figure 5.7, and the error plots are illustrated in Figure 5.8. This time, the proposed method designed a better quality filter with a better magnitude response in both the passband and stopband. Moreover, the group delay is much better in the passband with only 1.5591% standard deviation. Also, the group delay is still a full 6 s lower than for the equalizer design. One advantage of the equalizer design is that a steeper roll-off is achieved in the transition band producing a slight improvement with respect to the magnitude of the error as can be seen in Figure 5.8. Also, the equalizer method was much faster as before.

In several designs using the proposed method, it was observed that the algorithm seemed

Objective Function	Equalizer Method	Proposed Method
Elliptic filter order	6	NA
Equalizer filter order	22	NA
Filter order	28	26
Total recorded time, s	0.44	25.78
Iterations	N/A	14
Number of M-functions	52	78
Number of M-subfunctions	23	32
Number of MEX-functions	3	1
Clock precision, s	5.0×10^{-8}	5.0×10^{-8}
Clock Speed, MHz	2000	2000
Maximum passband ripple, dB	0.050000	0.044675
Minimum stopband attenuation, dB	45.0034	50.0182
Standard deviation of group delay in passband, %	4.2565	1.5591
Composite filter quality	0.94123	0.1373
Passband group delay, secs	34.8402	29.1634
L_2 norm of the error	4.6477	6.0445

Table 5.5: Design results for the equalizer and proposed methods for example 3.

to have more difficulty completing the design with higher-order transfer functions. This suggests that the proposed method is a great alternative for designing lower-order nearly linear-phase IIR filters and that the equalizer design method would be more effective for designing stringent filters requiring transfer functions of very high orders and high-selectivity bandpass and stopband filters.

The filter radii and angles for the design using the proposed method are given in Table B.8 and the coefficients for the design using the equalizer method are given in Table B.7 of Appendix B.

Summarizing the results obtained, when the filter order is increased, the proposed method has more difficulty converging to a solution that would satisfy the specifications. Also, slight modifications were required for the BMT method to produce viable initial points. On the other hand, the proposed method produced a filter of better quality with respect to both the passband and stopband. In effect, this example demonstrates that the proposed method

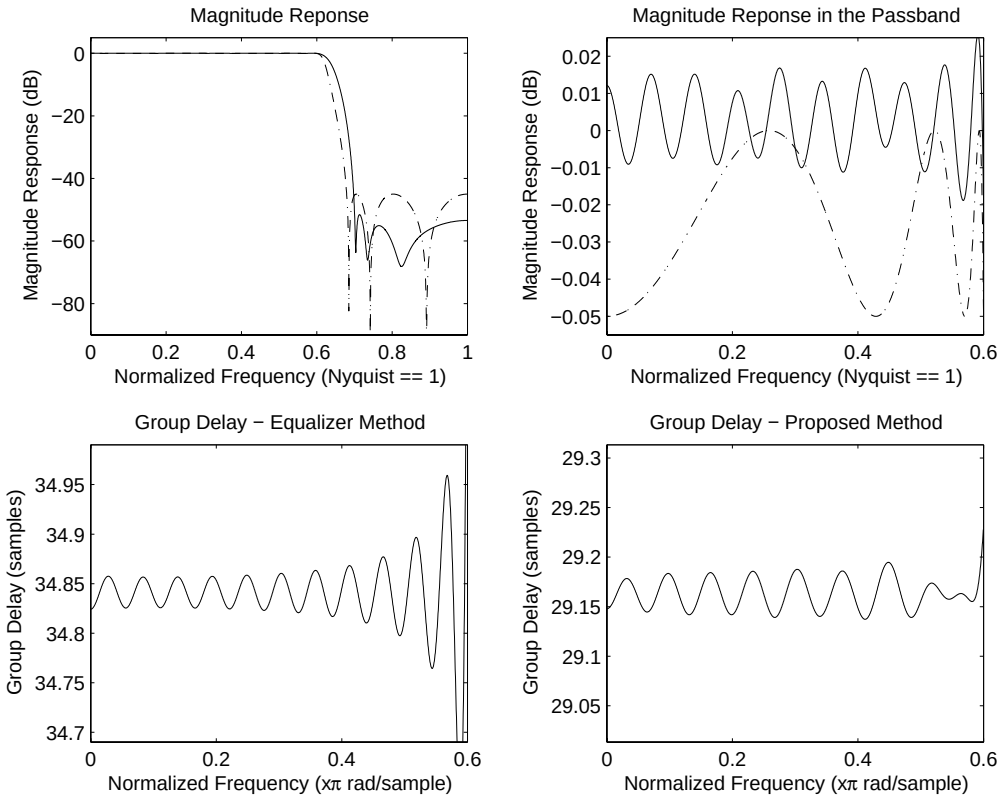

Figure 5.7: Magnitude response for the equalizer and proposed methods for example 3. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

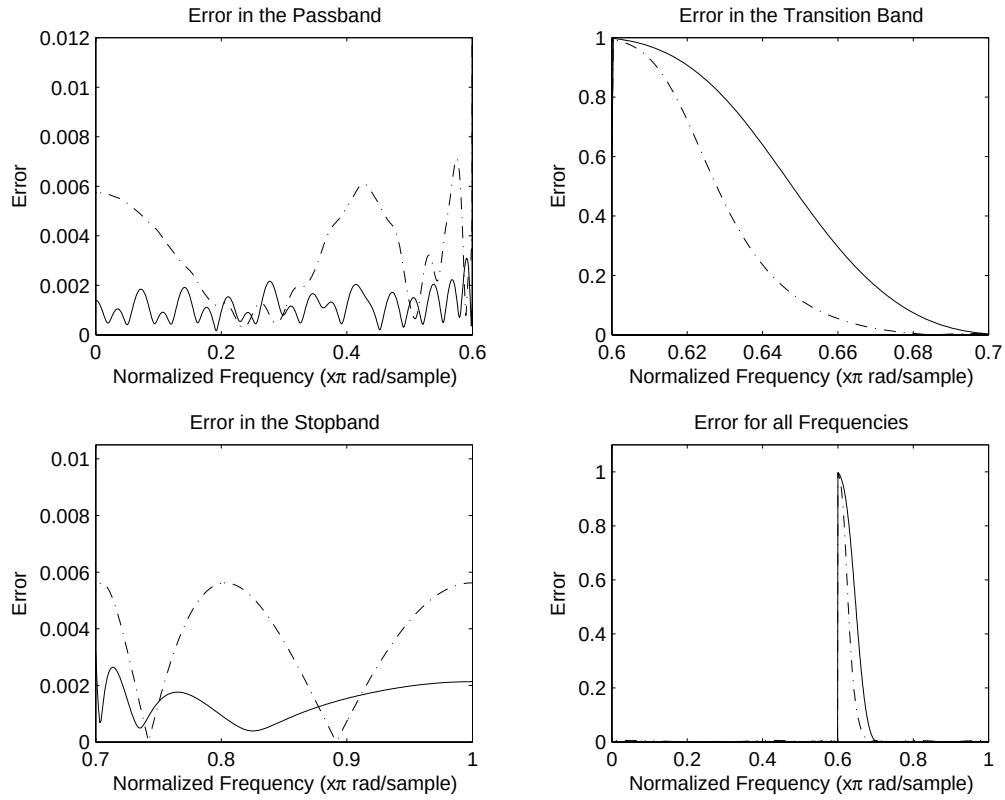

Figure 5.8: Magnitude of the error for the equalizer and proposed methods for example 3. $(\cdot - \cdot)$ Equalizer Method, $(-)$ Proposed Method

is capable of designing a lower-order filter of better quality but requires substantially more computation to complete the design when compared with designing an elliptic filter plus a delay equalizer.

5.3 Cited Designs

In this section, the proposed method is compared with other recent design techniques that do not use the equalization approach. The first example uses a minimax design method for IIR filters with a prescribed stability margin due to Lu and Hinamoto [7] that is formulated as a CQP problem. The second example uses an unconstrained optimization technique based on a parameterization approach proposed by Lu [3].

In the cited papers, a relative deviation of the group delay is used instead of the standard deviation of the group delay used in Chapter 4. For the sake of consistency, the relative deviation will be used in the following examples to provide a better comparison between the group delay responses. The relative deviation is defined as

$$\tau_{ri} = \frac{|\tau_i - \overline{\tau}|}{\overline{\tau}} \tag{5.1}$$

where τ_i is the group delay at frequency with index i and $\overline{\tau}$ is the average group delay across the passband.

5.3.1 CQP Example

Consider the IIR lowpass filter of order $n = 12$ presented by Deczky in Example 1 of [18], which has been used by many authors as a benchmark filter for comparison purposes. Deczky's filter was compared with a filter designed using the CQP method in [7]. For this example, the CQP results reported in [7] are compared with results using the method

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.1
Step limit, $\beta_{\hat{p}_i}$	0.02
Step limit, β_{H_0}	0.02
Step limit, β_τ	0.04
Initial linear-phase FIR order, n	25

Table 5.6: The algorithm design parameters for the proposed method used in the CQP example.

proposed in this thesis. The required passband and stopband frequencies are $\omega_p = 0.5\pi$ and $\omega_a = 0.6\pi$, respectively. The filter coefficients for the CQP design were taken directly from [7] for comparisons with the proposed method.

The goal in this example was to attempt to design a filter that produces better frequency response characteristics than the one designed in [7]. To achieve this goal, the design results from [7] were used as the filter specifications for the proposed method in this example. It was quickly realized that this filter design required many attempts to converge to a better solution. Therefore, certain algorithm parameters needed to be altered slightly to obtain the best solution. The algorithm parameters used for this example are given in Table 5.6 and the step limit modifications explained in section 4.3.7 for the polar-based objective function were applied.

Results

The design results are given in Table 5.7, the magnitude response and group delay plots are illustrated in Figure 5.9, and the zero/pole plots are shown in Figure 5.10. The design using the proposed method had a better magnitude response in both the passband and stopband. Moreover, the group delay in the passband was reduced by 4 s and the average relative deviation in the group delay was also substantially reduced as shown in Figure 5.9.

Objective Function	CQP Method	Proposed Method
Filter order	12	12
Iterations	16	71
Maximum passband ripple, dB	0.26584	0.12611
Minimum stopband attenuation, dB	36.1455	38.8629
Composite filter quality	4.5342	3.724
Passband group delay, s	15.89998	11.90938
L_2 norm of the error	5.2865	7.6329
Average relative deviation of group delay in passband	0.008741	0.003639
Maximum relative deviation of group delay in passband	0.069471	0.067940

Table 5.7: Design results for the CQP and proposed methods.

It is interesting to note that the zero and pole formations are very similar for the two designs. This may be because the CQP method described in [7] also uses an initial linear-phase FIR approximation along with the BMT method to obtain the initial points.

In Figure 5.11, one can see that the error was reduced throughout the passband with the exception of a few points near the passband edge where an error spike is evident. This error spike was easily reduced by allowing the optimization to continue beyond the required specifications. Furthermore, the error in the stopband was also reduced with the proposed method. On the other hand, the error within the transition band remained a problem because the magnitude response did not change fast enough. This suggests that the proposed method has more difficulty providing a sharp roll-off magnitude response than the CQP method.

Summarizing the results obtained, the proposed method produced a filter of better quality with respect to the magnitude response in both the passband and stopband. Also, the average and maximum relative group delay values were better than those for the filter design using the CQP method. Comparatively, the design results for the proposed method were slightly better demonstrating that the proposed method can compete with a well-known nearly linear-phase IIR filter design technique.

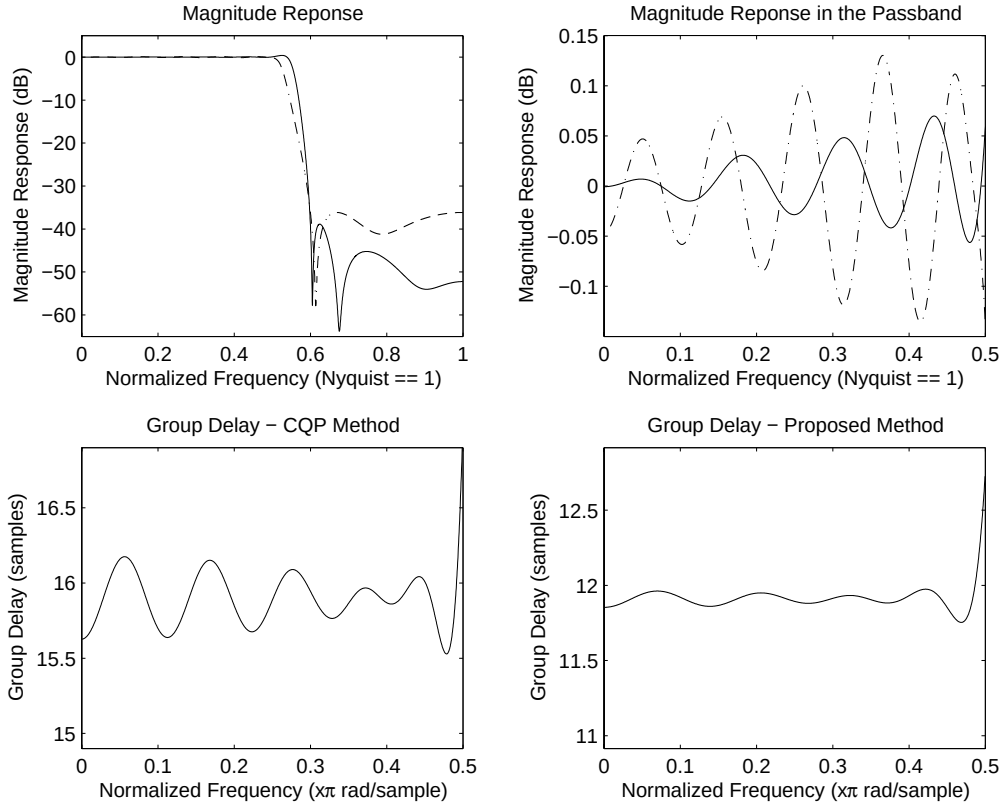

Figure 5.9: Magnitude response for the CQP and proposed methods. $(\cdot - \cdot)$ CQP Method, $(-)$ Proposed Method

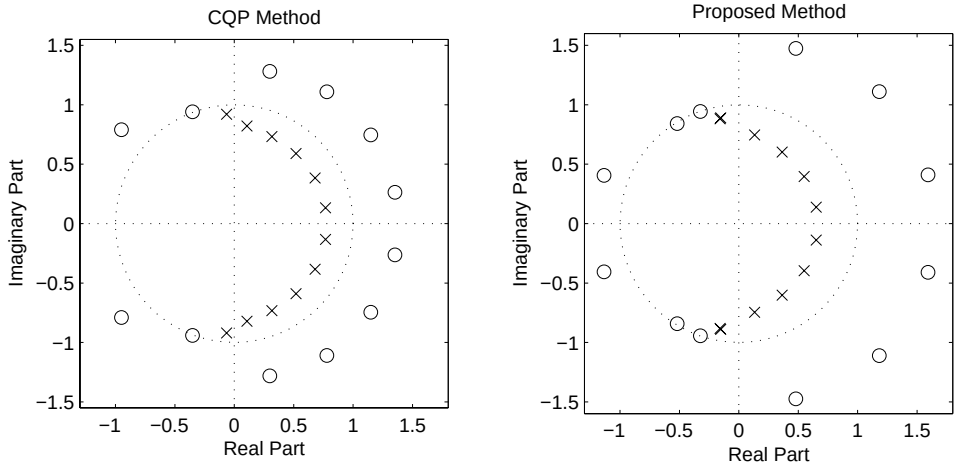

Figure 5.10: Zero and pole plots for the CQP and proposed methods.

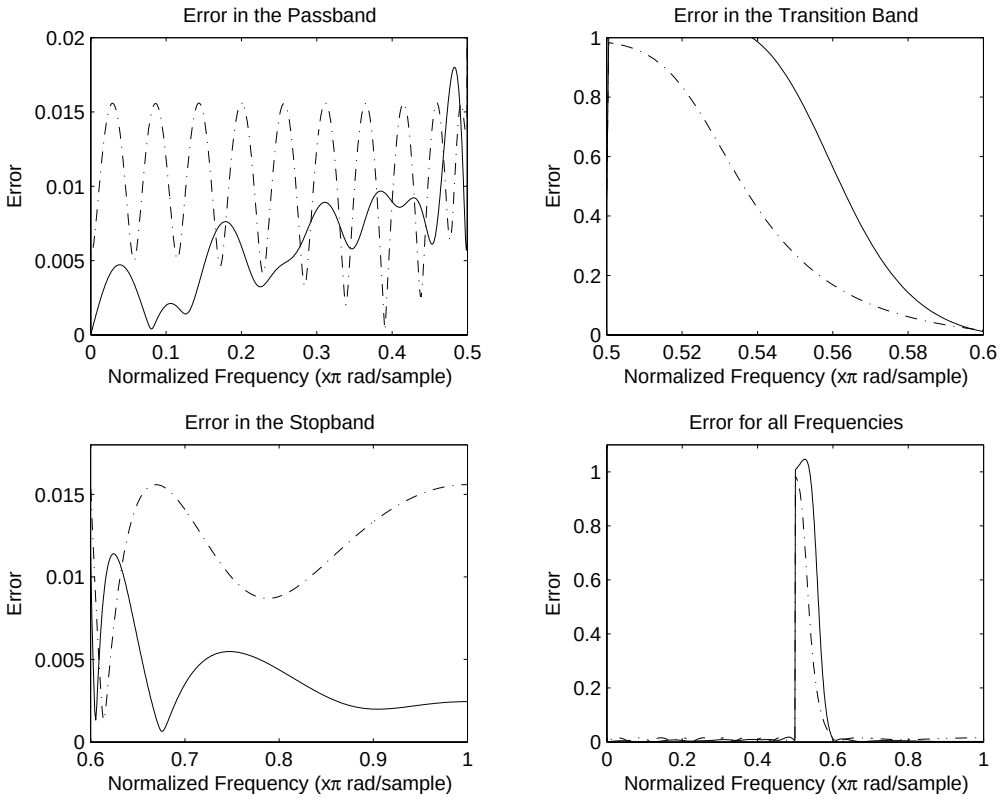

Figure 5.11: Magnitude of the error both the CQP and proposed methods. $(\cdot - \cdot)$ CQP Method, $(-)$ Proposed Method

5.3.2 Unconstrained Optimization Example

It is of interest to compare the results obtained with the proposed constrained optimization design presented in this thesis with those obtained with an unconstrained optimization approach. One major problem with designing IIR filters using an unconstrained optimization technique is satisfying the filter stability requirements [3][7]. Methods are available that can be used to stabilize the filter. One popular popular method is to replace poles that are located outside the unit circle by their reciprocals. However, this stabilization technique cannot be used if the phase response is part of the design problem because it would change the phase response of the stabilized filter. In [3], Lu presented an alternative approach by parameterizing the transfer function in such a way as to assure stability and then carry out unconstrained optimization over a set of possible transfer functions [3].

The filter specifications are summarized in Table 5.8 and the algorithm parameters used in the proposed method are given in Table 5.9.

Results

The design results obtained are summarized in Table 5.10. Since the transfer function coefficients were not given in [3], the results obtained using paramaterization cannot be pre-

Parameter	Value
Sampling frequency, ω_s, rad/s	2π
Maximum passband ripple, A_p, dB	0.1
Minimum stopband loss, A_a, dB	45 dB
Passband edge, ω_a, rad/s	0.4π
Stopband edge, ω_p, rad/s	0.6π
Maximum relative deviation of group delay in passband, %	5

Table 5.8: Lowpass digital filter specifications used for the unconstrained vs constrained optimization example.

Parameter	Value
Number of sampling intervals per band	40
Stability margin, δ_s	0.05
Step limit, $\beta_{\hat{p}_i}$	0.01
Step limit, β_{H_0}	0.01
Step limit, β_τ	0.02
Initial linear-phase FIR order, n	16

Table 5.9: Algorithm design parameters for the proposed method used in the parameterization example.

sented graphically. In Table 5.10, the abbreviation BFGS represents the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton optimization technique and Modified Newton refers to the modified Newton optimization technique mentioned in [3]. The abbreviations HTT, ATT, and MBT represent the hyperbolic tangent transformation, arc-tangent transformation, and modified bilinear transformation methods, respectively. These transformations are used to assure filter stability by applying parameterization to the second-order denominator polynomials of the transfer function. To illustrate the effectiveness of the proposed method, all of the design results from each transformation type are included in Table 5.10. As one can see, the proposed method designed a filter with the lowest maximum passband ripple, largest minimum stopband attenuation, and the lowest maximum relative deviation of the group delay in the passband. Moreover, the proposed method required only 37 iterations to obtain better results than all of the designs using the unconstrained optimization technique.

It was observed that when the optimization using the proposed method was allowed to continue, there was an improvement in both the minimum stopband attenuation and the maximum relative group delay in the passaband. However, a noticeable magnitude response spike would begin to form in the transition band. Therefore, the optimization was stopped just prior to the formation of this peak. Another way to suppress this peak is to simply increase the stability margin, but the design would become more difficult to complete. In fact, this slight peak can be seen in the magnitude response plot of [3] and suggests that

Optimization algorithm	BFGS			Modified Newton			Proposed
Parameterization	HTT	ATT	MBT	HTT	ATT	MBT	
Number of iterations	99	164	126	74	91	90	37
Maximum passband ripple, dB	0.0818	0.0663	0.0311	0.0773	0.0389	0.0785	0.0233
Minimum stopband attenuation, dB	47.5156	49.3101	46.5222	48.4584	45.7446	46.9811	50.945
Maximum relative deviation of group delay in passband	4.19%	3.63%	4.90%	1.89%	4.78%	1.80%	1.79%

Table 5.10: Design results for the CQP and proposed methods.

both the unconstrained and constrained optimization techniques have difficulty with this phenomenon. There seems to be a trade-off between the formation of this peak in the transition band and the filter specifications.

One noticeable advantage of the unconstrained method is a very slight reduction in the average group delay with respect to the passband. It can be seen in [3] that the average group delay is around 9 sampling periods whereas for the proposed method design the average group delay is 9.2 sampling periods.

Summarizing the results of this section, the proposed method produced a filter of better quality with fewer iterations when compared with the unconstrained method. This improvement is apparent for all the filter specifications: maximum passband ripple, minimum stopband attenuation, and maximum relative group delay in the passband.

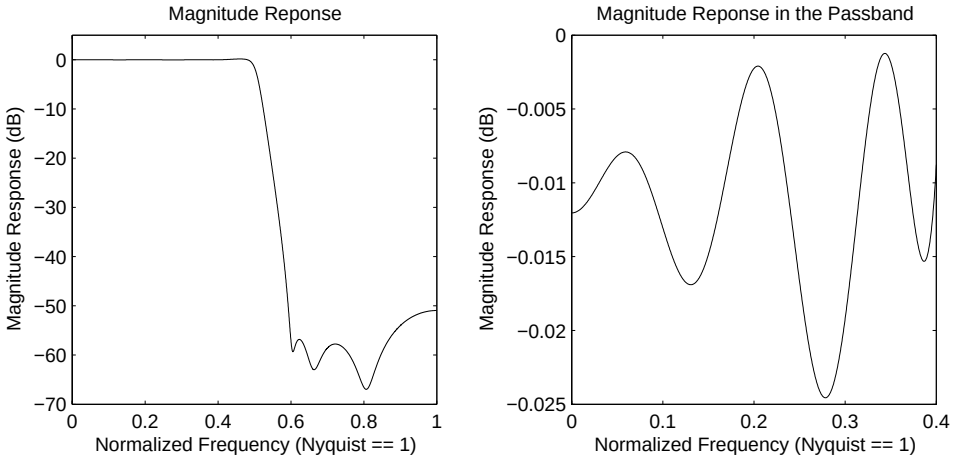

Figure 5.12: Magnitude response using the proposed method for the unconstrained vs constrained optimization example.

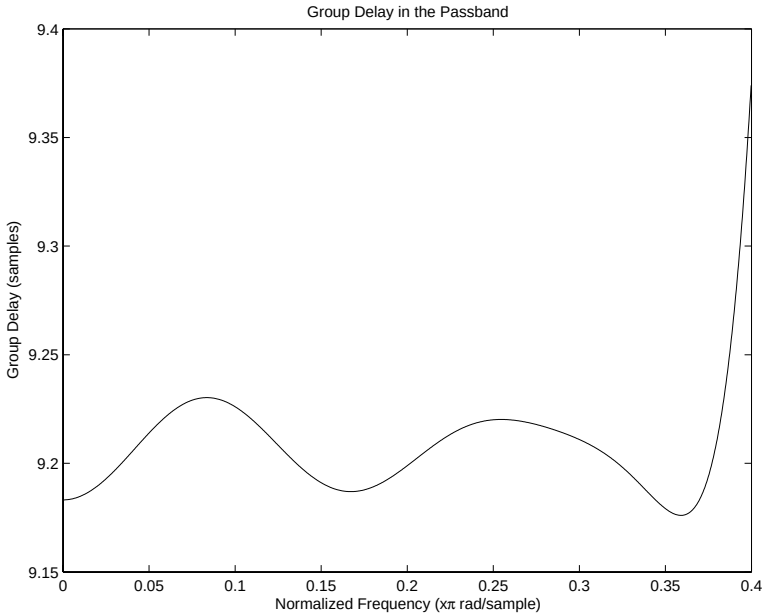

Figure 5.13: Group delay using the proposed method for the unconstrained vs constrained optimization example

5.4 Conclusions

To test the new method further, it was compared with three different modern techniques that can be used to design nearly linear-phase IIR filters. They include an optimal equalization technique, a minimax IIR filter design method formulated as a CQP problem, and an unconstrained optimization technique that uses parameterization to provide filter stability while retaining the desired phase response.

Several design examples were investigated and compared to the proposed method. Three design examples were investigated when comparing the proposed method and equalizer design method. In each example, the proposed method produced an equivalent or improved magnitude and phase response with a lower filter order. In example 3, a 26th-order filter was designed to illustrate that the new method can produce better results in terms of filter quality with a lower filter order compared with the equalizer design.

The proposed method produced improved results with respect to both the magnitude response and group delay deviation when compared to the design using the CQP method. However, the CQP method required fewer iterations and may have produced improved results if the optimization was allowed to continue. In the final example, the proposed method outperformed the unconstrained optimization technique in all characteristics except for a slightly higher average group delay in the passband.

Based on the results in this chapter, the new proposed method is deemed to be a viable alternative for the design of nearly linear-phase IIR digital filters for filter orders up to 30.

Chapter 6

Conclusions and Recommendations for Future Work

I am not young enough to know everything.

—Oscar Wilde

6.1 Conclusions

The design of recursive digital filters using constrained optimization was investigated. The problem required solutions designed for magnitude response specifications as well as a nearly linear-phase response in the passband. This problem was solved using a constrained optimization method subject to a set of linear constraints that utilizes a modified Newton's method. Following this, the required gradient and Hessian for the objective and error functions were derived. Unexpectedly, the Hessian of the problem investigated is not block-symmetric as was the case in the optimization problem considered in [2] and required several closed-form equations to facilitate its evaluation. Furthermore, since the objective function is highly nonlinear, the proposed method required a set of linear step constraints to provide

efficient convergence. The filter stability problem was solved by integrating several linear constraints which restricted the poles from moving on or outside the unit circle. However, another serious problem surfaced where the poles tended to be attracted to the real axis which generally yielded unsatisfactory results. This phenomenon was overcome by using a set of linear constraints that interpolate the real-pole boundary in the coefficient space. Two initialization methods were investigated: the method in [6] developed by observing the zero and pole formations for several nearly linear-phase recursive filter designs, and the balanced model truncation (BMT) method that converts a high-order linear-phase FIR filter into a lower-order nearly linear-phase IIR representation. As seen in Chapters 4 and 5, the BMT method provided the best initial points for each example where it was applied.

Inspired by Ko's thesis [2], further research was performed to find an alternative approach to the design problem. In particular, the zero/pole positions were investigated using a transfer function represented in terms of its zeros and poles in polar form. It was found that the observed zero and pole formations occurred because the optimization must balance all of the polar delay ratios to enforce a constant group delay in the passband. Using this new objective function, the number of linear constraints was substantially decreased and the possibility of poles being located on the real axis was also significantly reduced. In fact, the real-axis attraction problem was largely eliminated by integrating a linear constraint that controls the minimum allowable phase angle.

An analysis of a nonuniform error sampling technique was then carried out using the coefficient-based objective function and its application resulted in improved solutions. Using this technique, an in-depth comparison between several designs using the coefficient-based and polar-based objective functions was conducted. To fully understand each objective function's benefits and detriments, several different initial points were tested. It was found that the polar-based objective function produced better results in terms of filter quality and number of iterations. However, these improved designs were obtained only when the initial points were generated using a superior initialization method such as the BMT method. More impor-

tantly, several algorithm modifications were developed during the study of the two objective functions, namely, the step limit modifications and the initial frequency modifications for the BMT method.

Using the methods developed in Chapters 2 and 3 and applying the algorithm modifications and conclusions from Chapter 4, the proposed method was tested and compared with three existing design methods. First, the traditional equalization approach using an elliptic filter and an optimal equalizer was investigated. In three design examples, the proposed method was found to produce lower-order filters satisfying the specifications. Moreover, in the third example, the proposed method produced a filter of better quality while retaining a lower filter order. Second, the proposed method was compared with a method using a minimax approximation accomplished through a sequence of linear updates with each update carried out using a conic quadratic programming (CQP) technique as described in [7]. The resulting design using the proposed method had an improved magnitude response and quality factor. However, it required more iterations to complete the design compared with the approach in [7]. Lastly, the proposed method was compared to a parameterization approach for the design of IIR filters with a prescribed stability margin. The parameterization guaranteed filter stability and an unconstrained optimization method was used to design the filter. The proposed method outperformed the parameterization approach with respect to the magnitude and group-delay characteristic. Moreover, the proposed method outperformed all of the parameterization transformations investigated in [3].

On the basis of the comparisons carried out, the proposed method offers a number of advantages and is, therefore, considered a viable alternative in many situations. With additional research and better insight, the proposed method could replace known techniques for the design of IIR filters of orders up to 30 while preserving stability and obtaining nearly constant group delay with respect to the passband.

6.2 Recommendation for Future Work

As with many research projects, several issues and new ideas that may produce improved results have surfaced as summarized in the following subsections.

6.2.1 Dynamic Weighting Scheme

One interesting observation during several simulations was the order in which the algorithm optimized the specification parameters. The main trend was that the maximum passband ripple and minimum stopband attenuation parameters were reached before the minimum group delay deviation was achieved. This suggests that a dynamic weighting scheme may be attractive for this problem. The objective function could be reformulated to contain a weighting variable that can weigh the importance of either the magnitude response or the group delay. For example, when the magnitude response parameters are satisfied the algorithm could apply more weighting to the group delay characteristic and less to the magnitude in order to enhance the optimization of the phase response. This scheme could also be employed to achieve the opposite effect where the group delay requirements are satisfied before the magnitude response. In addition, by utilizing the separate quality factors described in Chapter 3, one could construct a conditional statement that updates the weighting variables based on the values of the separate quality factors.

Unfortunately, the gradient and Hessian are required for the objective and error functions. Furthermore, since the group delay and magnitude response equations are involved, the Hessian's closed-form equations may be difficult to derive. This possible approach gives rise to an important question: is it worth the extra research for the implementation of additional weighting factors? On the other hand, the modified objective function may provide the necessary control to force the optimization algorithm to focus on either the magnitude response or group delay. Furthermore, using such an objective function, the optimization

could be used to design filters with arbitrary group delay characteristics as opposed to just flat characteristics.

6.2.2 Improved Initial Points

Using the BMT method described in Chapter 2, the magnitude response specifications were usually satisfied before optimization was applied. Therefore, only the group delay response required correction when minimizing the objective function. This suggests that slightly moving the phase zeros so as to enforce a constant group delay could improve the quality of the initial filter parameters and may even solve the design problem without optimization for less stringent prescribed specifications. Furthermore, this may lead to a complete closed-form method that could be used to design lowpass recursive filters with nearly linear phase response. This method could initially begin with a high-order FIR filter that is designed using one of the many available techniques. The initial FIR filter can be represented by the transfer function in Eq. 2.106. Before the FIR filter is transformed into an IIR filter, the required approximation filter order must be determined. This is done by rejecting the negligible Hankel singular values [15]. After the approximation filter order is determined, the new kth-order truncated system representing an IIR filter is determined using Eqs. 2.109-2.112.

As noted in Chapter 3, the optimal zero positions for a lowpass filter are located in two formations along or near the unit circle in the stopband (magnitude zeros), and along an ellipse-like formation outside of the unit circle in the passband (phase zeros). It is the phase zeros that are of interest in these recommendations.

The phase zeros help to provide a nearly linear phase in the passband. The idea is to slightly shift these zeros with respect to their polar angles so as to reduce the group delay deviation. One way to determine where to move the phase zeros is to minimize the group delay error within the passband. However, by continuing the investigation from Chapter 3, another

solution may be more effective. Using Eq. 3.6, one can determine the phase angles for all the biquadratic transfer functions by finding the angles that keep the group delay constant in the passband. This can be achieved since the pole radii and angles as well as the zero radii are known values leaving only the zero angles to be determined. In fact, Eq. 3.6 can be reformulated in terms of several quadratic equations for each biquadratic transfer function.

At first, this scenario may seem fairly straightforward and easy to implement but the quadratic equations must be solved for all frequencies in the passband. To simplify this process one could solve these equations at frequencies where peaks occur in the group delay, which is typically near the passband edges.

Another approach would be to solve for the pole angles and shift them to enforce a constant group delay. Intuitively, this approach may be more difficult because the magnitude response will change and produce unsatisfactory results. This is because the poles must assume a specific formation in order to satisfy the magnitude response specifications whereas the phase zeros are used mainly to provide a linear phase in the passband and to balance the polar delay ratios for all biquadratic transfer functions.

6.2.3 Other Filter Types

In this thesis, only lowpass digital filter designs were investigated. Other types of filters should be investigated such as highpass, bandpass, bandstop, and possibly multiband filters.

6.2.4 Step Limit Updates

Another consideration when using the polar-based objective function is to separate the step limits for the angles and radii of the poles and zeros. With further investigation, one may find an efficient method to individually update the step limits during optimization. For

instance, since the phase zeros are located farther from the origin, the zero polar angle might be much smaller than the radius for a given optimization step whereas the poles are located closer to the origin and may not produce better convergence by separating the step limits. This idea could also be applied to the coefficient-based objective function for each individual coefficient. The integration of separate step limits was briefly tested but no apparent benefits were revealed. However, by further investigating the coefficient values and multiplier constant during several optimizations and understanding the step size for optimal convergence, one may find an efficient updating scheme.

6.2.5 Initialization of the BMT Method

When using the BMT method to generate the initial points for the optimization algorithm, the method required several inputs: the initial FIR filter order, the maximum passband ripple, the minimum stopband attenuation, and the passband and stopband frequencies. It was found that when these values were slightly adjusted, the resulting design is affected quite substantially. Also, as mentioned in section 5.2, the initial order for the FIR filter should be as low as possible but large enough to provide a good initial point. Furthermore, the initial passband and stopband frequencies were also slightly adjusted to provide a smaller transition band, which usually resulted in a better design. For example, the proposed algorithm sometimes had difficulty near the band edges and a simple modification to the passband and stopband frequencies usually fixed the problem.

Also, there seems to be a relation between the values of the passband and stopband edge frequencies with the order. For example, if the passband edge frequency value is slightly increased, the order of the FIR filter that produces the best initial points is different. This suggests that an efficient method could be devised to provide the best input for the BMT method to produce the best initial filter design with respect to the filter specifications.

6.2.6 Unconstrained Optimization Using the Polar-Form Objective Function with Parameterization

This was addressed in [3] by parameterizing the second-order denominator polynomials using several transformation functions. This idea can be easily applied to the polar-form objective function presented in this thesis. The immediate advantage of the proposed method is that the design can be accomplished using unconstrained optimization techniques [3].

The parameterization can be done using the hyperbolic tangent transformation (HTT) or the arc-tangent transformation (ATT). In these transformations, the number of parameters in the denominator of the transfer function remains the same and the parameters can vary over the entire parameter space without violating the stability of the transfer function.

To guarantee filter stability, the radius all the poles must be restricted to lie in the interval $0 \leq r_{bk} < 1$ for all transfer function sections. Therefore, the denominator of each filter section can be parameterized using either the HTT or ATT function. To ensure that the poles do not move close to the unit circle, a stability margin can be integrated.

Unfortunately, the closed-form equations for the gradient and Hessian must be derived. The objective function, its gradient, and Hessian can be used with an unconstrained optimization technique to design linear-phase IIR digital filter and the BFGS or Newton's method could be used as described in [3]. Intuitively, this new approach could be more computationally expensive due to the nonlinear contributions of the hyperbolic and cosine functions. However, since the problem is not using a quadratic approximation in a small neighborhood of $\mathbf{x}$, the rate of convergence per iteration might be improved compared with that achieved with a constrained optimization approach. By using this method along with the BMT method for initialization, it may be possible to achieve improved results compared with those obtained with the proposed method.

Bibliography

[1] A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw-Hill, New York, 2005.

[2] N. Ko, *Design of Recursive Delay Equalizers by Constrained Optimization*, Department of Electrical and Computer Engineering, Thesis, University of Victoria, 2001.

[3] W.-S. Lu, "Design of recursive digital filters with prescribed stability margin: a parameterization approach," *IEEE Trans. Circuits and Systems*, vol. 45, pp. 1289-1298, Sept. 1998.

[4] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, New York, 2007.

[5] A. Antoniou, *Optimization Theory and Practice*, Lecture Notes, Dept. of Electrical and Computer Engineering, University of Victoria, BC, Mar. 2000.

[6] S. Saab, A. Antoniou, and W.-S. Lu, "Design of linear-phase recursive digital filters by optimization," *Proc. Int. Symp. on Advances in Digital Filtering and Signal Processing*,

pp. 87-91, Jun. 1998.

[7] W.-S. Lu and T. Hinamoto, "Optimal design of IIR digital filters with robust stability using conic-quadratic-programming updates," *IEEE Trans. on Signal Processing*, vol. 51, pp. 1581-1592, Jun. 2003.

[8] R. Fletcher, *Practical Methods of Optimization, 2nd ed.*, Wiley, New York, NY, 1993.

[9] I. Kale, J. Gryka, G. D. Cain, and B. Beliczynski, "FIR filter order reduction: Balanced model truncation and Hankel-Norm optimal approximation," *IEEE Proc. Vis. Image Signal Processing*, vol. 141, pp. 168-174, Jun. 1994.

[10] B. Beliczynski, J. Gryka, and I. Kale, "Critical comparison of Hankel-norm optimal approximation and balanced model truncation algorithms as vehicles for FIR-to-IIR filter order reduction," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 593-596, Apr. 1994.

[11] B. C. Moore, "Principal component analysis in linear systems: controllability, observability and model reduction," *IEEE Trans. on Automatic Control*, vol. AC-26, pp. 17-31, Feb. 1981.

[12] L. Prenobo and L. M. Silverman, "Model reduction via balanced state space representation," *IEEE Trans. on Automatic Control*, vol. AC-27, pp. 382-387, Apr. 1982.

[13] S. Y. Kung and D. W. Lin, "Optimal Hankel-norm model reductions: Multivariable systems," *IEEE Trans. on Automatic Control*, vol. AC-26, pp. 832-852, Aug. 1981.

[14] I. W. Selesnick, M. Lang, and C.S. Burrus, "Constrained least square design of FIR filters without specified transition bands," *IEEE Trans. on Signal Processing*, vol. 44, pp. 1879-1892, Aug. 1996.

[15] B. Beliczynski, I. Kale, and G. D. Cain, "Approximation of FIR to IIR digital filters: an algorithm based on balanced model reduction," *IEEE Trans. on Signal Processing*, vol. 40, pp. 532-542, Mar. 1992.

[16] T. W. Parks and C.S. Burrus, *Digital Filter Design*, New York: Wiley, pp. 54-83, 1987.

[17] D.J. Shpak, "iirgrpdelay", in MATAB Filter Design Toolbox, The Mathworks.

[18] A. G. Deczky, "Equiripple and minimax (Chebyshev) approximation for recursive digital filters," *IEEE Trans. Acoust., Speech, Signal Procesing*, vol. ASSP-23, pp. 98-111, Aug. 1974.

Appendix A

Initialization by Trends

A.1 Algorithm 1: Lowpass and highpass filters

1. Set: n=filter order
2. Set: ω_p to passband edge
3. If lowpass: Set: $\Omega_p = \omega_p$
4. If highpass: Set: $\Omega_p = \pi/T - \omega_p$
5. Set $\Omega_a = \pi/T - \Omega_p$
6. Choose an even or odd number of phase zeros.

   If even: Set: $Z_p = 2\lfloor \Omega_p nT/(2\pi) + 1/2 \rfloor$

   If odd: Set: $Z_p = 2\lfloor \Omega_p nT/(2\pi) + 1 \rfloor$
7. Set: $Z_a = n - Z_p$
8. Call Algorithm 2 with the following parameters:

If lowpass:

(a) $n = \text{order}, \Omega = -\Omega_p, \omega_1 = \omega_p, m = 1/2$

(b) $n = Z_p, \Omega = -\Omega_p, \omega_1 = \omega_p, m = 2$

(c) $n = Z_a, \Omega = \Omega_a, \omega_1 = \omega_p, m = 1$

If highpass:

(a) $n = \text{order}, \Omega = \Omega_p, \omega_1 = \omega_p, m = 1/2$

(b) $n = Z_p, \Omega = \Omega_p, \omega_1 = \omega_p, m = 2$

(c) $n = Z_a, \Omega = -\Omega_a, \omega_1 = \omega_p, m = 1$

9. Assign H_0 to give an average gain of unity.

A.2 Algorithm 2: Pole and zero placement

1. Set: $\omega = \omega_1 + \Omega$

   If $\omega = 0$ or $\omega = \pi$: Set: $\Delta\omega = 2\Omega/(n-1)$

   Else: Set: $\Delta\omega = \Omega/(n-1)$

2. For $i = 1$ to $i = \lfloor (n+1)/2 \rfloor$ do:

   If $\omega_i = 0$ or $\omega_i = \pi$:

      If $m = 1/2$: Place a pole at $0.5e^{j\omega_i}$

      Else: Place a zero at $me^{j\omega_i}$

   Else:

      If $m = 1/2$: Place two poles at $0.5e^{\pm j\omega_i}$

      Else: Place two zeros at $me^{\pm j\omega_i}$

   Assign $\omega_{i+1} = \omega_i + \Delta\omega$

Appendix B

Design Examples

B.1 Tenth-Order Filters Obtained with Initial Points Using the Method by Trends

The coefficients of the 10th-order filters obtained with initial points generated by the method by trends using the coefficient-based and polar-based objective functions in section 4.3.6 are given in Tables B.1 and B.2, respectively.

B.2 Tenth-Order Filters Obtained with Initial Points Using the BMT Method

The coefficients of the 10th-order filters obtained with initial points generated by the balanced model truncation method using the coefficient-based and polar-based objective functions in section 4.3.7 are given in Tables B.3 and B.4, respectively.

B.3 Filters Obtained Using the Equalizer and Proposed Methods

The coefficients of the filters generated for the three examples in section 5.2 are given in Tables B.5 to B.7. Table B.5 gives the solution for example 1 using the equalizer method. Tables B.6 and B.7 give the solutions for example 2 using the proposed method and the equalizer method, respectively. Table B.8 gives the solution for example 3 using the proposed method.

Coefficient-Based Objective Function				
Section	Coefficients			
k	a_{k0}	a_{k1}	b_{k0}	b_{k1}
1	3.738040	-4.010070	0.313102	-1.075770
2	2.019180	-2.607780	0.289046	-1.028200
3	0.998513	-1.134030	0.427678	-1.266610
4	0.962355	-0.449304	0.632994	-1.335450
5	0.934984	1.273470	0.909390	-1.414390
$H_0 = 1.164619 \times 10^{-3}$ $\tau = 16.771953$				

Table B.1: Results for example in section 4.3.6.

Polar-Based Objective Function				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.375230	0.164649	0.749012	0.3737430
2	1.342720	0.487606	0.945468	0.7387690
3	0.998024	0.971923	0.821635	0.5987690
4	0.683881	1.461140	0.533826	-0.1540560
5	0.747657	1.994150	0.644794	0.0687978
$H_0 = 4.088575 \times 10^{-3}$ $\tau = 17.781171$				

Table B.2: Radii and angles for example in section 4.3.6.

Coefficient-Based Objective Function				
Section	Coefficients			
k	a_{k0}	a_{k1}	b_{k0}	b_{k1}
1	0.814732	1.021700	0.587557	-1.522850
2	1.975990	-2.776500	0.603455	-1.465090
3	1.846070	-2.415260	0.653866	-1.383160
4	0.973021	-0.698450	0.843569	-1.320200
5	0.996690	-1.146940	0.759085	-1.244940
$H_0 = 2.470408 \times 10^{-3}$ $\tau = 18.086577$				

Table B.3: Results for the example in section 4.3.7.

Polar-Based Objective Function				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.491250	0.123393	0.780374	0.354940
2	0.707742	1.592660	0.771273	0.123500
3	1.369320	0.472271	0.819884	0.561606
4	0.935502	1.100920	0.910005	0.743558
5	0.990347	0.966698	0.949277	0.850155
$H_0 = 5.379746 \times 10^{-3}$ $\tau = 16.829704$				

Table B.4: Results for the example in section 4.3.7.

Equalizer Method				
Index	Section 1		Section 2	
k	a_k	b_k	a_k	b_k
1	0.168930	1.000000	0.018338	1.000000
2	-1.893100	-7.819390	-0.017002	-3.474550
3	9.674460	27.962200	0.016776	5.308180
4	-29.700200	-60.194900	0.016776	-4.330120
5	60.678100	86.354400	-0.017002	1.875420
6	-86.230400	-86.230400	0.018338	-0.342712
7	86.354400	60.678100		
8	-60.194900	-29.700200		
9	27.962200	9.674460		
10	-7.819390	-1.893100		
11	1.000000	0.168930		
$H_0 = 1 \quad \tau = 42.901769$				

Table B.5: Results using the equalizer method for example 1 in section 5.2.

Proposed Method				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.226630	3.106340	0.533586	-0.0320582
2	1.077100	2.731540	0.442086	2.059150
3	1.463210	0.182845	0.603987	0.389271
4	1.459230	0.548701	0.637158	0.744003
5	1.449370	0.915195	0.658090	1.093270
6	1.430590	1.284700	0.682599	1.426340
7	1.391080	1.660850	0.734206	1.744780
8	1.025080	2.492640	0.868508	2.074530
9	0.998648	2.209040	0.866182	2.180810
10	0.988336	2.307980	0.918125	2.093170
$H_0 = 1.182254 \times 10^{-3} \quad \tau = 17.078690$				

Table B.6: Results using the proposed method for example 2 in section 5.2.

Equalizer Method				
Index	Section 1		Section 2	
k	a_k	b_k	a_k	b_k
1	0.008644	1.000000	0.132264	1.000000
2	-0.118852	-8.618270	0.577263	1.073050
3	0.838232	38.513100	1.215680	1.782140
4	-4.022610	-118.207000	1.532860	0.813227
5	14.706800	278.623000	1.215680	0.634328
6	-43.494300	-534.859000	0.577263	0.065641
7	107.901000	866.148000	0.132264	0.045967
8	-229.914000	-1210.280000		
9	427.538000	1481.210000		
10	-701.378000	-1603.530000		
11	1022.310000	1545.100000		
12	-1329.480000	-1329.480000		
13	1545.100000	1022.310000		
14	-1603.530000	-701.378000		
15	1481.210000	427.538000		
16	-1210.280000	-229.914000		
17	866.148000	107.901000		
18	-534.859000	-43.494300		
19	278.623000	14.706800		
20	-118.207000	-4.022610		
21	38.513100	0.838232		
22	-8.618270	-0.118852		
23	1.000000	0.008644		
$H_0 = 1 \quad \tau = 34.840205$				

Table B.7: Results using the equalizer method for example 2 in section 5.2.

Proposed Method				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.482020	2.765520	0.710447	-0.030086
2	1.270990	0.107324	0.757375	0.253646
3	1.039980	2.590040	0.732662	1.653980
4	1.270270	0.322579	0.769314	0.468430
5	1.260030	1.398970	0.775517	1.479630
6	1.269950	0.537754	0.774110	0.678557
7	1.268180	0.751935	0.776275	0.884379
8	1.262420	1.183920	0.775930	1.283650
9	1.264920	0.967829	0.776393	1.086400
10	1.252060	1.609560	0.790131	1.738620
11	1.220500	1.820170	0.841542	1.930550
12	1.012150	2.308620	0.885177	2.021490
13	1.003180	2.210250	0.910688	2.145140
$H_0 = 7.578389 \times 10^{-4}$ $\tau = 29.163123$				

Table B.8: Results using the proposed method for example 3 in section 5.2.

B.4 Filter Obtained for Example Using the CQP and Proposed Methods

The coefficients of the 12th-order filter generated by the proposed method in section 5.3.1 is given in Table B.9,

Proposed Method				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.204470	2.798140	0.666185	0.209707
2	1.643560	0.252086	0.677546	0.625413
3	1.622610	0.754500	0.703612	1.026360
4	1.550300	1.256020	0.757564	1.394910
5	0.989388	2.122800	0.902866	1.741360
6	0.997449	1.901180	0.895292	1.747590
$H_0 = 3.779753 \times 10^{-3}$ $\tau = 11.909379$				

Table B.9: Results using the proposed method for the example in section 5.3.1.

B.5 Filter Obtained from Example using the Unconstrained Optimization and Proposed Methods

The coefficients of the 12th-order filter generated by the proposed method in section 5.3.2 is given in Table B.10.

Proposed Method				
Section	Radii and Angles			
k	r_{ak}	θ_{ak}	r_{bk}	θ_{bk}
1	1.931050	-0.311310	0.586778	0.247106
2	1.823580	0.939245	0.601833	0.720966
3	1.058620	2.078650	0.656699	1.136790
4	0.938980	2.535370	0.796837	1.503860
5	0.977925	1.894980	0.923888	1.586380
$H_0 = 3.752166 \times 10^{-3}$ $\tau = 9.206966$				

Table B.10: Results using the proposed method for the example in section 5.3.2.