Design of Recursive Delay Equalizers by Constrained Optimization

by

Nelson Richard Ko
B.A., University of Toronto, 1990
B.Eng., University of Victoria, 1995

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
We accept this thesis as conforming
to the required standard

[redacted]

Dr. A. Antoniou, Supervisor (Dept. of Elec. and Comp. Eng.)

[redacted]

Dr. D. Shpak, Supervisor (Dept. of Elec. and Comp. Eng.)

[redacted]

Dr. N. Horspool, Outside Member (Dept. of Computer Science)

[redacted]

Dr. L. Trajkovic, External Examiner (School of Engineering Science, Simon Fraser University)

University of Victoria

Supervisors: Dr. A. Antoniou and Dr. D. Shpak

ABSTRACT

Current design methods for recursive delay equalizers use unconstrained optimization techniques that can lead to unstable designs. Several approaches are taken here to find a better algorithm. To begin this thesis, the unconstrained method is introduced and analyzed. Among the findings of the analysis are regular pole plots for the solutions, a tendency for the poles to be moved to the real axis during the optimization, a strong correlation between the starting point placement for the optimization and the optimization's success, and, due to the unconstrained nature of the algorithm, unstable equalizer designs. Although there are ways to deal with these problems (such as restarting a design with new initial values), there is no guarantee that a stable equalizer which meets prescribed specifications will be obtained. This thesis focuses on achieving a stable design and on making the optimization as efficient as possible. Examination of stable recursive delay equalizers leads to improved design techniques through two main improvements. First, the equalizers are designed using a constrained optimization technique incorporating a known objective function. The constrained technique involves four methods: the Newton method, the primal active set method, the Lagrange method, and the generalized elimination method. Modifications such as a different starting point algorithm were also made to the basic model to make the algorithm more efficient. This new method guarantees a stable design, but is found to require more computations per iteration due to the use of

the Hessian matrix. In order to further understand the problem, an analysis of the relationship of the pole positions to the group delay is performed. Delay magnitude and peak positions were found to relate to the pole angle and magnitude. Using this relationship, an explanation of the regular pole patterns found earlier in successful equalizer designs was developed. The objective function is finally reworked to be pole-position based in order to improve the performance and characteristics of the optimization. The new method is shown to be more effective in finding stable and efficient equalizers, and more capable in finding them for difficult filters. The computational complexity is much greater than the unconstrained coefficient-based method, and this thesis mentions some possible solutions to make the algorithm more efficient, including using a mix of unconstrained and constrained optimizations. Some directions on where to further investigate this topic are also given.

Examiners:

[redacted]

Dr. A. Antoniou, Supervisor (Dept. of Elec. and Comp. Eng.)

[redacted]

Dr. D. Shpak, Supervisor (Dept. of Elec. and Comp. Eng.)

[redacted]

Dr. N. Horspool, Outside Member (Dept. of Computer Science)

[redacted]

Dr. L. Trajkovic, External Examiner (School of Engineering Science, Simon Fraser University)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recursive filters are not commonly used in applications that require constant group delay [4] because they often have highly nonlinear phase responses. Instead, nonrecursive filters are generally used. Unfortunately, if the required filter selectivity is high, the minimum nonrecursive filter order can be very high, and usually much higher than that required by a corresponding recursive filter [4]. Savings can be achieved even when recursive delay equalizers are used in combination with these filters in order to satisfy constant group delay requirements. Current methods of designing recursive equalizers are based on unconstrained optimization methods. These methods can lead to unstable designs that can add much time and computational cost to a problem since repeated attempts may be required before a stable design is obtained. By exploring the design of equalizers and possibly making them easier to find in many applications, a recursive filter and equalizer combination may become an attractive alternative to the nonrecursive filter.

It should be mentioned that in this study, the equalizers are usually designed to

equalize elliptic filters, as they are a common and efficient type of filter. Both the filters and equalizers are analyzed and designed in terms of sections characterized by biquadratic transfer functions of the form

$$H(z) = \frac{a_1 + a_2 z^{-1} + z^{-2}}{1 + b_2 z^{-1} + b_1 z^{-2}}. \tag{1.1}$$

Finally, it is assumed that the phase response is required to be linear in the passband (where equalization is required), and that delay distortion is unimportant in the stopbands since signals in these bands are deemed to be noise.

An introduction to recursive delay equalizers is presented in Chapter 2 to provide a background to the topic and to serve as a stepping stone for methods pursued later in this study. A few methods are currently used to design recursive delay equalizers. Among them are the least-pth and Charalambous minimax algorithms that are based on unconstrained optimization methods. Unfortunately, with these techniques, the poles of the equalizer equation can move anywhere in the entire z-plane. Of course, if any of the resulting poles are outside of the unit circle, then the equalizer is unstable. There are also other negative aspects that will be discussed later. In this thesis, an investigation of current techniques will serve to identify areas of the optimization problem that can be improved.

Using constrained optimization, bounds can be placed on the coefficients that guarantee a stable equalizer, as shown in Chapter 3. The proposed constrained optimization method is composed of four different algorithms that are developed to produce stable designs. The four methods are arranged in a hierachical structure. Newton's method is used to simplify the problem into a series of quadratic opti-

mization sub-problems. The primal active set method and the Lagrange method are used to further simplify the problem. Finally, the generalized elimination method is used to solve the resulting series of sub-problems. The overall technique requires the objective function, its gradient, Hessian, and a set of constraints. The objective function and gradient are based on the least-pth objective function developed for the unconstrained techniques. The Hessian is developed in Chapter 3. Moreover, a set of linear constraints is developed, and modifications are made to the algorithm to adapt it to the recursive delay equalizer problem.

A more detailed analysis of the problem is undertaken in Chapter 4 in an attempt to find a better method of deriving a starting point, or even some sort of closed-form solution. To do this, an analysis of the group delay from the perspective of the pole positions of the equalizer is done.

Finally, an improvement to the objective function is proposed in Chapter 5. The least-pth objective function based on the polar equations for the poles is developed and analyzed. Using this new objective function, some solutions to problems identified in Chapter 3 by using the previous method are sought.

Using the new ideas developed for designing recursive delay equalizers, it may become simpler to design recursive delay equalizers for recursive filters, and they may become a viable alternative to nonrecursive filters currently used in applications requiring constant group delay.

Chapter 2

Design by Unconstrained Optimization

2.1 Introduction

Several methods are currently being used to design recursive delay equalizers, and almost without exception, they are based on unconstrained optimization [1],[4],[5]. The method in [1] employs a quasi-Newton method and the least-pth minimax objective function to design the equalizer. Charalambous [5] provides a method that offers an efficient alternative to the least-pth method. It is from the first method, however, that the constrained technique described later is developed.

This chapter provides some details of the method in [1]. The algorithm is initially discussed followed by the objective function used in the method. The remainder of the chapter contains observations of the characteristics of the method and points out specific undesired characteristics.

2.2 Design of Recursive Delay Equalizers using Unconstrained Methods

The algorithm described in [1] incorporates three distinct processes to design a recursive delay equalizer. At the lowest level, the algorithm attempts to grow the equalizer section by section starting with a single biquadratic equalizer section. The algorithm optimizes the first section equalizer and uses the obtained solution to determine a starting point for the second-section equalizer optimization. The algorithm continues to increase the number of biquadratic sections, using the previous solution to determine the next starting point, until the equalizer meets prescribed specifications. At the second level, the algorithm optimizes each of these stages using unconstrained optimization. The method used a quasi-Newton algorithm with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating formula in combination with the Fletcher inexact line search or a variation of it described by Antoniou [2]. At the highest level is a least-pth minimax algorithm based on the quasi-Newton algorithm. The objective function used is the basis for the constrained method developed later. Hence, we give here details of the equations involved.

2.2.1 'Growing' the Equalizer

At the highest level of the design algorithm, a method to grow equalizers is used. It starts with a single equalizer section, and optimizes it using a series of widely varying starting points until a stable solution is found. With a stable solution to this problem, another section is added using a heuristic scheme to generate a new

starting point, and another optimization is performed. This procedure of adding sections and generating new starting points continues until the equalizer meets a predefined specification. In this respect, the equalizer is 'grown'. The specification that the equalizer must meet is in terms of Q, a tolerance factor, which is defined in [1] as

$$Q = \frac{100(\tau_{\hat{FE}} - \tau_{\breve{FE}})}{\tau_{\hat{FE}} + \tau_{\breve{FE}}} \tag{2.1}$$

where $\tau_{\breve{FE}}$ and $\tau_{\hat{FE}}$ are the minimum and maximum total group delays of the filter-equalizer combination, respectively. By reducing Q, better results are achieved. In the case where the equalizer becomes unstable, a few alternative modification points to the solution of the previous optimization are available to obtain a new starting point. If none of the alternate starting points is successful, then the number of sections in the optimization is reduced. For instance, if no solution for a 7-section equalizer were found with the current solution for a 6-section equalizer, a new 6-section equalizer is designed using another starting point derived from the last 5-section equalizer that was designed.

The optimization is re-done using an alternative starting point for the reduced section equalizer. This process of growing and reducing the equalizer is repeated until a stable equalizer with a suitable Q is designed. Occasionally, the algorithm will not find a stable solution at a certain order, and will decrease the number of sections in the equalizer by many levels before finding a solution. For difficult filters the design may never converge to a stable solution that meets specifications.

2.2.2 Optimization Method

To optimize an equalizer with a specific number of biquadratic sections, a minimax approach is used. This is done using the least-pth algorithm, where repeated optimization for an increasing p is carried out. As p increases, this solution approaches the L_∞ or minimax solution. The value of p may start at 2, and iteratively increases by a factor of 1.5 to 2 after a solution is found. When the difference between the solutions of subsequent levels of p approaches a specified threshold, the optimization is considered to have reached the minimax solution. Empirical results have shown that if p increases beyond 1000, the optimization can also be considered as having converged.

The unconstrained optimization method used to find the solution of each least-pth sub-problem is the quasi-Newton algorithm using the BFGS updating matrix method [7] with either the Fletcher line search or its modified version [1].

Error Function

The error function $e_i(\mathbf{x})$ for the delay equalizer problem is assumed to be the difference between the actual delay and the average normalized passband delay τ_0 of the current filter equalizer combination. The error function and its gradient are

$$e_i(\mathbf{x}) = \frac{1}{T}\tau_{FE}(\mathbf{x}, \omega_i) - \tau_0, \tag{2.2}$$

and

$$\nabla e_i(\mathbf{x}) = \left[\frac{\partial e_i(\mathbf{x})}{\partial c_{11}} \frac{\partial e_i(\mathbf{x})}{\partial c_{12}} \ldots \frac{\partial e_i(\mathbf{x})}{\partial \tau_0} \right]^T. \tag{2.3}$$

Since parameter τ_0 is unknown, it can be treated as one of the problem variables.

Thus the parameter vector of the problem at hand can be expressed as

$$\mathbf{x} = [\mathbf{c}^T \tau_0]^T \tag{2.4}$$

where $\mathbf{c}$ is a vector whose elements are the coefficients of the equalizer transfer function

$$H_E(z) = \prod_{j=1}^{M} \frac{c_{j1} + c_{j2} z^{-1} + z^{-2}}{1 + c_{j2} z^{-1} + c_{j1} z^{-2}} \tag{2.5}$$

i.e.,

$$\mathbf{c} = [c_{11} \quad c_{12} \quad c_{21} \quad c_{22} \quad \cdots]^T . \tag{2.6}$$

In equation 2.5, M is the number of biquadratic equalizer sections being used in the optimization.

To obtain the error function, the delay of the filter-equalizer combination, τ_{FE}, must be found. It is the sum of the delay of the filter, τ_F, and the delay of the equalizer, τ_E. The delay of the filter is shown in [1] to be

$$\tau_F(\omega) = -T \sum_{j=1}^{J} \frac{\hat{N}_j(\omega)}{N_j(\omega)} + \sum_{j=1}^{J} \frac{\hat{D}_j(\omega)}{D_j(w)} \tag{2.7}$$

where

$$\hat{N}_j(\omega) = b_{j1}^2 - b_{j3}^2 + b_{j2}(b_{j1} - b_{j3}) \cos(\omega T) \tag{2.8}$$

$$N_j(\omega) = (b_{j1} - b_{j3})^2 + b_{j2}^2 + 2b_{j2}(b_{j1} + j_3) \cos(\omega T) + 4b_{j3} b_{j1} \cos^2(\omega T) \tag{2.9}$$

$$\hat{D}_j(\omega) = a_{j1}^2 - a_{j3}^2 + a_{j2}(a_{j1} - a_{j3}) \cos(\omega T) \tag{2.10}$$

$$D_j(\omega) = (a_{j1} - a_{j3})^2 + a_{j2} + 2a_{j2}(a_{j1} + a_{j3}) \cos(\omega T) + 4a_{j3} a_{j1} \cos^2(\omega T) \tag{2.11}$$

and T is the sampling period. The coefficients are the coefficients of the filter transfer function

$$H_F(z) = H_0 \prod_{j=1}^{J} \frac{b_{j1} + b_{j2}z^{-1} + b_{j3}z^{-2}}{a_{j1} + a_{j2}z^{-1} + a_{j3}z^{-2}}. \tag{2.12}$$

The delay of the equalizer can similarly be represented as

$$\tau_E(\mathbf{c}, \omega) = 2T \prod_{j=1}^{M} \frac{\hat{C}_j(\omega)}{C_j(\omega)} \tag{2.13}$$

where

$$\hat{C}_j(\omega) = 1 - c_{j1}^2 + c_{j2}(1 - c_{j1})\cos(\omega T) \tag{2.14}$$

$$C_j(\omega) = (1 - c_{j1})^2 + c_{j2}^2 + 2c_{j2}(1 - c_{j1})\cos(\omega T) + 4c_{j1}\cos^2(\omega T) \tag{2.15}$$

where the coefficient are the coefficients in equation 2.5.

The gradient of the error function is composed of the partial derivatives

$$\frac{\partial e_i(\mathbf{x})}{\partial c_{j1}} = \frac{U_{j0} + Uj1\cos(\omega_i T) + U_{j2}\cos^2(\omega_i T) + U_{j3}\cos^3(\omega_i T)}{[C_j(\omega_i)]^2} \tag{2.16}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial c_{j2}} = \frac{V_{j0} + Vj1\cos(\omega_i T) + V_{j2}\cos^2(\omega_i T) + V_{1j}\cos^3(\omega_i T)}{[C_j(\omega_i)]^2} \tag{2.17}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \tau_0} = -1 \tag{2.18}$$

where $1 \leq j \leq M$ and

$$U_{j0} = 4\left[(1 - c_{j1})^2 - c_{j1}c_{j12}^2\right] \tag{2.19}$$

$$U_{j1} = -2c_{j2}(1 + 6c_{j1} + c_{j1}^2 + c_{j2}^2) \tag{2.20}$$

$$U_{j2} = -8(1 + c_{j1}^2 + c_{j2}^2) \tag{2.21}$$

$$U_{j3} = -8c_{j2} \tag{2.22}$$

$$V_{j0} = -4c_{j2}(1 - c_{j1})(1 + c_{j1}) \tag{2.23}$$

$$V_{j1} = -2c_{j2}(1 - c_{j1})(1 + 6c_{j1} + c_{j1}^2 + c_{j2}^2) \tag{2.24}$$

$$V_{j2} = 0 \tag{2.25}$$

$$V_{j3} = 8(1 - cj1)c_{j1}. \tag{2.26}$$

2.2.3 Objective Function

The objective function as described in [1] is

$$\Psi(\mathbf{x}) = \hat{E}(\mathbf{x}) \left\{ \sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})} \right]^p \right\}^{1/p} \tag{2.27}$$

where K is the number of frequency samples used and

$$\hat{E}(\mathbf{x}) = \max_{1 \le i \le K} |e_i(\mathbf{x})| . \tag{2.28}$$

The gradient of equation 2.27 is given in [1] as

$$\nabla\Psi(\mathbf{x}) = \left\{ \sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})} \right] \right\}^{1/p-1} \sum_{i=1}^{K} \left\{ \left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})} \right]^{p-1} \nabla |e_i(\mathbf{x})| \right\} . \tag{2.29}$$

Note that the gradient of the objective function requires the error function as well as the gradient of the error function.

2.3 Observations

2.3.1 Objective Function

There are several characteristics that are noticeable during the course of the optimization. Some of these characteristics can be attributed to the objective function

and some to the optimization method. For instance, the solutions from the optimization for several designs were found to have pole plots that are very regular. Also, the poles tend to move toward the real axis during the course of the optimization. These characteristics will be later attributed directly to the objective function. There are also traits that can be attributed to the optimization method. One is the need for the value τ_0 to be close to the solution value for the starting point to work well. Also, the poles can move outside the unit circle causing the equalizer to become unstable. Moreover, sometimes the steps can be too large, and the optimization can miss a path that would lead to a better solution. Examination of these characteristics may lead to an algorithm that can avoid a trait or use it to its advantage.

Solution Structures

The most noticeable trait is that the solutions to the recursive delay equalizer problems are so predictable that one can tell if the current solution is the most efficient one or not. Generally, the poles are found to be most efficient (in terms of Q) when placed within a sector that includes the passband regions. This is under the assumption that the passband is the region where the equalizer is required to have a constant group delay. Moreover, the poles are generally placed evenly along an ellipse within these passbands. Figure 2.1 shows zero-pole plots for four examples where lowpass, highpass, bandpass, and bandstop filters have been equalized. A study of these equalizers, which is described later, explains why these structures might be the most efficient.

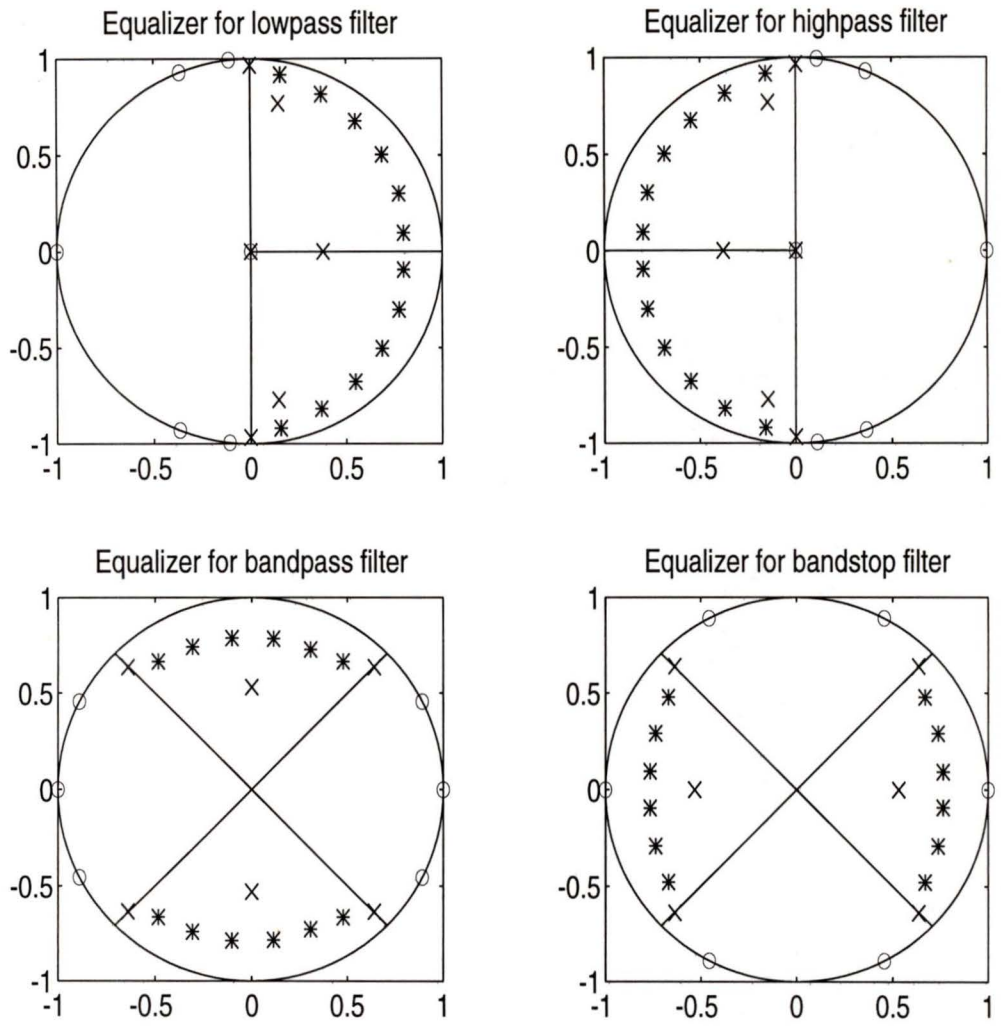

Figure 2.1: Ideal equalizers examples for lowpass, highpass, bandpass, and bandstop filters, x: filter pole, *: equalizer pole.

Real-Axis Attraction

One way an equalizer could become inefficient is for a pair of poles to wind up on the real axis. This happens often during the course of the optimization. Whenever a conjugate pair of poles of an equalizer section move close to the real axis, the poles may suddenly jump onto the axis as if there was a strong attraction to it. Usually,

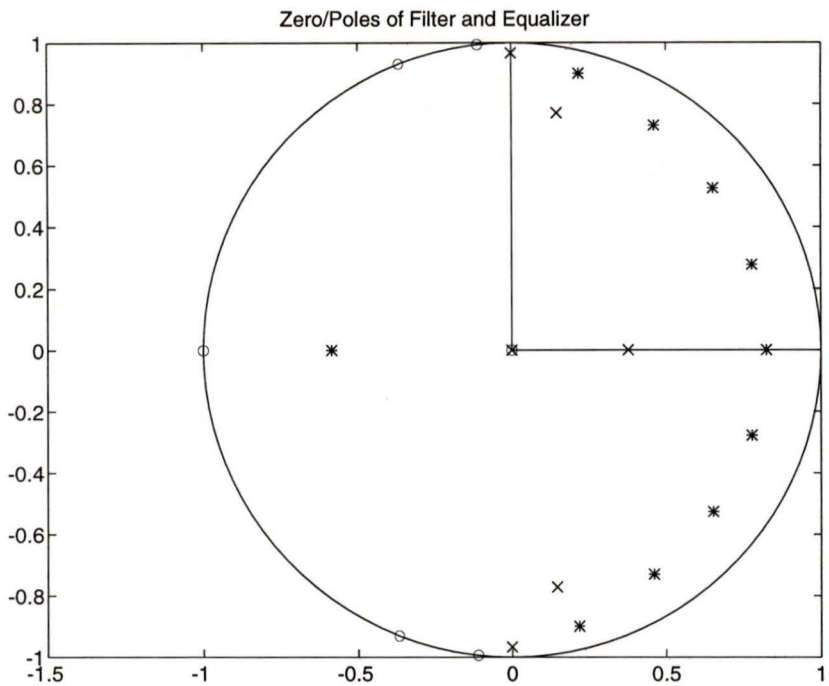

Figure 2.2: Poles of equalizer on real axis.

at this position, only one of the two poles is on the solution ellipse and is being efficiently used. The other pole often heads in the opposite direction of the first pole, toward the other end of the unit circle. It very often leaves the unit circle. An example of a pair of poles on the real axis is shown in Figure 2.2. By examining the equations involved, some insight may be revealed as to why this happens.

The poles are the roots of the denominator of equation 2.5. To be real, the coefficients must satisfy the inequality

$$c_{2j}^2 > 4c_{1j}. \tag{2.30}$$

Of course the poles must simultaneously be in the feasible region for the equalizer to be a stable one. Figure 2.3 shows the triangle representing the feasible region for

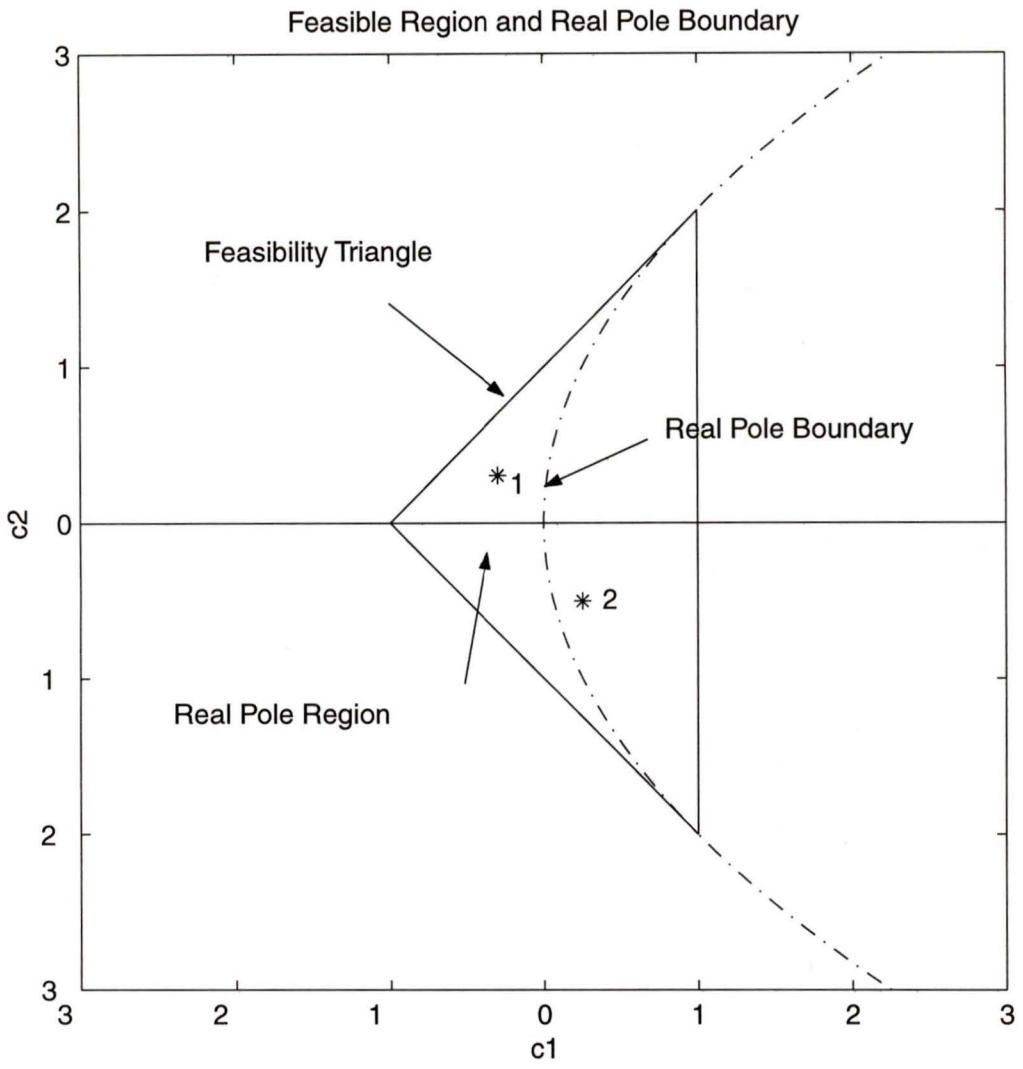

Figure 2.3: Region in coefficient plane where poles lie on the real axis in the imaginary plane.

the coefficients along with the curve representing equation 2.30. The latter defines the boundary of a region above which the poles are real (what will be called the real region). One can see that because the method was developed in the coefficient space (that is, the optimization is performed on the c_{j1},c_{j2} plane), it lends itself to developing real poles for the equalizer sections. The area in which poles are real is a large portion of the feasible region. As an example, if there is a pole within the

real feasible region (pole 1 in figure 2.3), and an optimization iteration occurs, the optimization step must be large enough to allow the pole to jump out of the real region, but hopefully still remain within the feasible region. Moreover, if a feasible pole is not within the real region, but close to it (pole 2 in figure 2.3), the pole has a good chance of being moved into the real region. This is not always the case, and even if the pole is in the real region, the optimization can still occasionally lead back into the part of the feasible region corresponding to complex-conjugate poles. Unfortunately, this does not usually happen, and the equalizer poles may end up on the real axis resulting in a stable solution that is locally but not globally optimal.

2.3.2 Optimization Method

The method of optimization also has certain traits that can be observed during the optimization.

Starting Points

It was found that good starting points tend to be less effective if the starting value of τ_0 is far from that required by the final solution. If the delay parameter must be changed by a large amount, the coefficient values also tend to change by large amounts because the gradient function is now dominated by the τ_0 component. Because the coefficient values are significantly smaller than the delay value, pole positions will be nowhere near previous positions after large steps. Therefore, good starting points for the poles are not effective if the value of τ_0 is not estimated correctly. The method described in [1] uses the value of τ_0 from the solution for the

equalizer with one less section. These values tend to be underestimations, but are not totally ineffective estimates for current optimizations. However, by finding a better starting value for τ_0 the starting points for the poles can be more effictively used.

Unstable Equalizers

In an unconstrained method, the optimization is performed over the entire coefficient plane. This allows the poles to wander outside the unit circle resulting in an unstable equalizer. Furthermore, the poles occasionally move in steps that are too large. This may be detrimental for the optimization in that there may be a more optimal path for the poles to follow and a large step could move the poles off this path. This could lead to a possibly ineffecient or unstable equalizer.

2.4 Conclusions

An unconstrained optimization method used to design recursive delay equalizers was described. The characteristics of the method and the results of the method were also described, most notably the regular pattern of the poles in the results, and the attraction of the poles to the real axis during optimization.

Unconstrained optimization methods such as the ones described are useful for designing equalizers having arbitrary specifications. However, because they are unconstrained, unstable equalizers may be obtained. The approach taken in the next chapter uses a constrained optimization technique to avoid these situations.

Chapter 3

Design by Constrained Optimization

The least-pth objective function from equation 2.27 can be used with a constrained optimization technique for the design of recursive delay equalizers. The main objective of this technique is to prevent the design solutions from becoming unstable. The overall method is composed of four techniques from [6], which are briefly described here. The method requires an objective function, its gradient, and Hessian. The objective function and its gradient have been obtained in Chapter 2, but the Hessian must be developed. The method also requires the development of a suitable set of constraints. Modifications are then made to the technique to adapt it to the problem at hand. Finally, the method is implemented and compared with the unconstrained method.

3.1 Development of Constrained Method

3.1.1 The Constrained Optimization Method

The constrained optimization method is based on Newton's method, the primal active set method, the Lagrange method, and the generalized elimination method. They are used in a hierarchical structure. Newton's method divides the problem into quadratic optimization sub-problems. Using the primal active set method, these sub-problems are changed so that the inequality constraints can be solved with equality programming techniques. The Lagrange method provides the parameters required by the primal active set method and, finally, the generalized elimination method is used to solve the quadratic equality problem.

Newton's Method

Newton's method is used to solve a highly nonlinear problem by solving a series of quadratic ones. The method is presented for a general linearly constrained problem of the form

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{subject to } \mathbf{A}^T\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{3.1}$$

and it will be approximated by a series of quadratic optimization problems. These approximations require an objective function, its gradient vector, and its Hessian matrix. When the solution to the optimization using the current approximation is found, the approximation is updated and used in another optimization. The iterations of these suboptimizations eventually solve the general programming problem.

The approximation represents the objective function as

$$f(\mathbf{x}^{(k)} + \delta) \approx F^{(k)}(\delta) = f^{(k)} + \delta^T \mathbf{g}^{(k)} + \frac{1}{2}\delta^T \mathbf{G}^{(k)} \delta \tag{3.2}$$

where $\mathbf{x}^{(k)}$ is the value of the parameter vector at the kth iteration, $f^{(k)}$ is the objective function evaluated at $\mathbf{x}^{(k)}$, $\mathbf{g}$ is the gradient, $\mathbf{G}$ is the Hessian of the objective function, and δ is a change in $\mathbf{x}^{(k)}$ in the neighbourhood of $\mathbf{x}^{(k)}$. Using this, equation 3.1 is rewritten into the quadratic programming problem

$$\begin{aligned} \min \mathbf{F}^{(k)}(\delta) &= \frac{1}{2}\delta^T \mathbf{G}^{(k)} \delta + \delta^T \mathbf{g}^{(k)} + f^{(k)} \\ \text{subject to } \mathbf{a}_i^T \delta &\geq b_i - \mathbf{a}_i^T \mathbf{x}^{(k)} \end{aligned} \tag{3.3}$$

where the variables $\mathbf{a}_i$ are the vectors in the constraint matrix $\mathbf{A}$ from equation 3.1.

The above approximation is of limited accuracy. Moreover, the neighbourhood in which the approximation is accurate has been observed to get smaller as the number of equalizer sections increases. Therefore, another set of constraints is required when using the Newton method. The constraints provide a step restriction in the form

$$\|\delta\|_\infty \leq h^{(k)} \tag{3.4}$$

although this is not the ultimate version of the step restriction used in this problem. The parameter $h^{(k)}$ is a step parameter that can be chosen to suit the problem (in the recursive delay equalizer problem, 1 would be satisfactory). A modification, described later, is made to it to make it more suited to this problem.

The Newton method generates a series of quadratic expressions for the next step to optimize, each having linear inequality constraints.

Primal Active Set Method

Linear inequalities cannot be handled by the final method in the constrained optimization technique, namely the generalized elimination method. To handle the inequalities in each of these quadratic approximations, the primal active set described in [6] is required. It allows problems with inequality constraints to be solved using equality programming techniques. Newton's method reduces the problem to equation 3.3 which can be rewritten as

$$\min \mathbf{f}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{x}^T\mathbf{g}$$

$$\text{subject to } \mathbf{a}_i^T\mathbf{x} \geq \mathbf{b}_i. \tag{3.5}$$

For the problem at hand, the constraints are all linear inequalities. Moreover, the recursive delay equalizer problem has two sets of inequality constraints. The first is the stability conditions, and the second is the step restriction in equation 3.4 imposed by the Newton method. The primal active set method, as inferred by its name, uses an active set of constraints to iteratively find a solution to the set of inequality constraints. The active set, $\mathcal{A}$, for the situation using $\mathbf{x}^{(k)}$ can be expressed as

$$\mathbf{a}_i^T\mathbf{x}^{(k)} = b_i, i \in \mathcal{A} \tag{3.6}$$

and, subsequently,

$$\mathbf{a}_i^T\mathbf{x}^{(k)} - b_i > 0, i \notin \mathcal{A}. \tag{3.7}$$

We rewrite equation 3.5 using $\mathbf{x} = \mathbf{x}^{(k)} + \delta$. The value $\mathbf{x}^{(k)}$ is the result from the kth iteration. We can then optimize δ in the equation

$$\min \mathbf{f}_k(\delta) = \frac{1}{2}\delta^T\mathbf{G}\delta + \delta^T\mathbf{g}_k$$

$$\text{subject to } \mathbf{a}_i^T \delta = 0, i \in \mathcal{A} \tag{3.8}$$

which is a quadratic programming problem with equality constraints where

$$\mathbf{g}_k = \mathbf{G}\mathbf{x}^{(k)} + \mathbf{g}. \tag{3.9}$$

The resulting quadratic equality problem is of the form

$$\begin{aligned} &\min f(x) = \tfrac{1}{2} x^T \mathbf{G} x + q^T x \\ &\text{subject to } \mathbf{A}^T x = b. \end{aligned} \tag{3.10}$$

Lagrange Method

To determine which constraints are active in the primal active set method, Lagrange multipliers are determined using the Lagrange method. The multipliers are found as

$$\lambda^* = (\mathbf{A}^T \mathbf{G}^{-1} \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{G}^{-1} \mathbf{q} + \mathbf{b}) \tag{3.11}$$

using a method in [6]. Though the method can also provide the solution to the quadratic problem, only the multipliers are used.

Generalized Elimination Method

Instead of solving the quadratic programming problem using the Lagrange method, the problem is solved using the generalized elimination method which solves problems with equality constraints. The method uses a closed-form equation to solve the problem, but requires the inverse of the Hessian. Therefore, for a convergent solution, the Hessian matrix, $\mathbf{G}$, must be positive definite.

The four methods are combined to form the constrained optimization algorithm used to solve the recursive delay equalizer problem presented in the following pages.

3.1.2 Hessian Matrices

The constrained method also requires the derivation of the Hessian matrix for minimizing the objective function. Again, there are two parts, the least-pth Hessian and the error function Hessian.

Least-pth Hessian

To make the approximation for Newton's method, the objective function, its gradient, and Hessian matrix are required. The objective function and its gradient are the same as those used in the unconstrained optimization method (equations 2.27 and 2.29). The Hessian matrix for the least-pth objective function is given by

$$\begin{aligned}
\mathbf{H}_{\Psi}(\mathbf{x}) = p\left(\frac{1}{p}-1\right)&\left\{\sum_{i=1}^{K}\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]\right\}^{1/p-2} \\
&\cdot\sum_{i=1}^{K}\left\{\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]^{p-1}\frac{\nabla^T|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right\} \\
&\cdot\sum_{i=1}^{K}\left\{\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]^{p-1}\nabla|e_i(\mathbf{x})|\right\} \\
&+\left\{\sum_{i=1}^{K}\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]\right\}^{1/p-1}\cdot\sum_{i=1}^{K}\left\{\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]^{p-1}H\left(|e_i(\mathbf{x})|\right)\right. \\
&\left.+(p-1)\left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\right]^{p-2}\frac{\nabla^T|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})}\nabla|e_i(\mathbf{x})|\right\}. \qquad (3.12)
\end{aligned}$$

Error Function

Most of the parameters in the least-pth Hessian equation are already defined. The Hessian matrix of the error function $\mathbf{H}_{e_i}$ is defined as

$$\mathbf{H}_{e_i}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial c_{11}\partial c_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{11}\partial c_{12}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{11}\partial \tau_0} \\ \frac{\partial e_i^2(\mathbf{x})}{\partial c_{12}\partial c_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{12}\partial c_{12}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{12}\partial \tau_0} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_i^2(\mathbf{x})}{\partial \tau_0\partial c_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau_0\partial c_{12}} & \cdots & \frac{\partial e_i^2(\mathbf{x})}{\partial \tau_0\partial \tau_0} \end{bmatrix}. \tag{3.13}$$

The solution of the Hessian of the error function can be simplified by noticing that

$$\frac{\partial e_i^2(\mathbf{x})}{\partial c_{k1}\partial c_{l1}} = 0, \quad k \neq l \tag{3.14}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial c_{k1}\partial c_{l2}} = 0, \quad k \neq l \tag{3.15}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial c_{k2}\partial c_{l1}} = 0, \quad k \neq l \tag{3.16}$$

$$\frac{\partial e_i^2(\mathbf{x})}{\partial c_{k2}\partial c_{l2}} = 0, \quad k \neq l \tag{3.17}$$

and that any partial derivative including τ_0 is also 0. This makes the Hessian of the error function a block diagonal matrix of the form

$$\mathbf{H}_{e_i}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i^2(\mathbf{x})}{\partial c_{11}\partial c_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{11}\partial c_{12}} & 0 & 0 & \cdots & 0 \\ \frac{\partial e_i^2(\mathbf{x})}{\partial c_{12}\partial c_{11}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{12}\partial c_{12}} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{21}\partial c_{21}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{21}\partial c_{22}} & \cdots & 0 \\ 0 & 0 & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{22}\partial c_{21}} & \frac{\partial e_i^2(\mathbf{x})}{\partial c_{22}\partial c_{22}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \tag{3.18}$$

where

$$\frac{\partial e_i(\mathbf{x})}{\partial c_{j1}\partial c_{j1}} = \left\{\left[-8 + 8c_{j1} - 4c_{j2}^2\right.\right.$$

$$
\begin{aligned}
&+(-12c_{j2} - 4c_{j1}c_{j2})\cos(\omega_i T) \\
&+ (-16c_{j1})\cos^2(\omega_i T)\Big] C_j^2(\omega_i) \\
&- W_{j1} \cdot \frac{\partial C_j^2(\omega_i)}{\partial c_{j1}}\Bigg\} / \left[C_j(\omega_i)\right]^4 \qquad (3.19)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial e_i(\mathbf{x})}{\partial c_{j1}\partial c_{j2}} = \Big\{\Big[&-8c_{j1}c_{j2} + (-2 - 12c_{j1} - 2c_{j1}^2 - 6c_{j2}^2)\cos(\omega_i T) \\
&- 16c_{j2}\cos^2(\omega_i T) - 8\cos^3(\omega_i T)\Big] C_j^2(\omega_i) \\
&- W_{j1} \cdot \frac{\partial C_j^2(\omega_i)}{\partial c_{j2}}\Bigg\} / \left[C_j(\omega_i)\right]^4 \qquad (3.20)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial e_i(\mathbf{x})}{\partial c_{j2}\partial c_{j1}} = \Big\{\Big[&8c_{j1}c_{j2} + (-10 + 20c_{j1} + 6c_{j1}^2 + 2c_{j2}^2)\cos(\omega_i T) \\
&+ (8 - 16c_{j1})\cos^3(\omega_i T)\Big] C_j^2(\omega_i) \\
&- W_{j2} \cdot \frac{\partial C_j^2(\omega_i)}{\partial c_{j1}}\Bigg\} / \left[C_j(\omega_i)\right]^4 \qquad (3.21)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial e_i(\mathbf{x})}{\partial c_{j2}\partial c_{j2}} = \Big\{\Big[&-4 + c_{j1}^2 - (4c_{j2} - 4c_{j1}c_{j2})\cos(\omega_i T)\Big] C_j^2(\omega_i) \\
&- W_{j2}\frac{\partial C_j^2(\omega_i)}{\partial c_{j2}}\Bigg\} / \left[C_j(\omega_i)\right]^4 . \qquad (3.22)
\end{aligned}
$$

Most of the parameters in the above equations are found in [2], and the remaining ones are given by

$$
\begin{aligned}
W_{j1} &= U_{j0} + U_{j1}\cos(\omega_i T) \\
&\quad + U_{j2}\cos^2(\omega_i T) + U_{j3}\cos^3(\omega_i T) \qquad (3.23) \\
W_{j2} &= V_{j0} + V_{j1}\cos(\omega_i T) \\
&\quad + V_{j2}\cos^2(\omega_i T) + V_{j3}\cos^3(\omega_i T) \qquad (3.24)
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial C_j^2(\omega_i)}{\partial c_{j1}} &= -4(1 - c_{j1})^3 - 4(1 - c_{j1})c_{j2}^2 \\
&\quad +4\left[-c_{j2} - 2c_{j1}c_{j2} + 3c_{j1}^2 c_{j2} + c_{j2}^2\right]\cos(\omega_i T)
\end{aligned}
$$

$$+\left[8(1-4c_{j1}+3c_{j1}^2)+16c_{j2}^2+8c_{j1}c_{j2}^2\right]\cos^2(\omega_i T)$$

$$+\left[16c_{j2}+32c_{j1}c_{j2}\right]\cos^3(\omega_i T)$$

$$+32c_{j2}\cos^4(\omega_i T) \tag{3.25}$$

$$\frac{\partial C_j^2(\omega_i)}{\partial c_{j2}} = 4(1-c_{j1})^2 c_{j2} - 4c_{j2}^3$$

$$+4\left[(1+c_{j1})(1-c_{j1})^2+3c_{j2}^2(1+c_{j1})\right]\cos(\omega_i T)$$

$$+\left[16c_{j1}c_{j2}+8c_{j2}(1+c_{j1})^2\right]\cos^2(\omega_i T)$$

$$+\left[16c_{j1}(1+c_{j1})\right]\cos^3(\omega_i T). \tag{3.26}$$

It should be mentioned that even though both of the off-diagonal elements are described, the Hessian is a symmetric matrix. Calculating only one of the off diagonal matrix entries will save computations. Also, it should be noted that there is no guarantee that the Hessian is positive definite, even though it is a requirement of the algorithm. To remedy this, a technique described later is applied to the Hessian to achieve positive definiteness if necessary.

3.1.3 Constraints

Having described the optimization techniques used in the problem, the objective function, its gradient, and Hessian matrix, the constraints must be described. There are two sets of constraints.

Step Restrictions

The first set of constraints, the step constraints, are required by the Newton method. As mentioned, they are found in [1] and shown in equation 3.4. The value of $\mathbf{h}^{(k)}$ is

determined by the iterative decision

$$h^{(k+1)} = \|\delta^{(k)}\|/4, r^{(k)} < 0.25 \tag{3.27}$$

$$h^{(k+1)} = 2h^{(k)}, r^{(k)} > 0.75 \text{ and } h^{(k)} = \|\delta\| \tag{3.28}$$

where

$$r^{(k)} = \frac{\Delta f^k}{\Delta F^k} \tag{3.29}$$

and the remaining parameters are

$$\Delta f^k = f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \delta^{(k)}) \tag{3.30}$$

and

$$\Delta F^k = F^{(0)} - F^{(k)}(\delta^{(k)}). \tag{3.31}$$

Stability Constraints

The stability constraints are shown in [1] to be the linear inequalities

$$c_{j1} < 1 \tag{3.32}$$

$$c_{j2} - c_{j1} < 1 \tag{3.33}$$

$$c_{j2} + c_{j1} > -1 \tag{3.34}$$

which form the feasibility triangle shown in figure 2.3. These requirements add 3 constraints for each section in the equalizer. In the z-plane, they keep the poles of the equalizer within the unit circle $|z| = 1$. The boundaries are, in practice, buffered by a small value to ensure the values are still feasible after coefficient quantization.

3.1.4 Modifications

Some modifications to the method described are possible to make the constrained optimization more efficient for the recursive delay equalizer problem. The first modification assures a positive definite Hessian for the generalized elimination method. The second modification changes the step restriction to ensure that the optimization more appropriately handles the step sizes according to the characteristics of the optimization parameters. The final modification relates to the starting point method. A different technique was developed from that described in [1] using estimates based on successful solutions. These modifications streamline the algorithm to work better for designing equalizers.

Ensuring a Positive Definite Hessian Matrix

The generalized elimination method requires a positive definite Hessian. However, if $p > 2$ in the least-pth problem, there is no guarantee that the Hessian will be positive definite. If it is not positive definite, it can be made so by adding another matrix as

$$\mathbf{H}_2 = \mathbf{H}_1 + \beta\mathbf{I} \tag{3.35}$$

where $\beta \geq -\lambda_i$, λ_i being the minimum eigenvalue of the original Hessian $\mathbf{H}_1$ and $\mathbf{I}$ is the identity matrix [3]. As the solution is approached and the surface tends to become locally quadratic, β becomes increasingly smaller and $\mathbf{H}_2$ becomes a more accurate representation of the Hessian.

Modifying the Step Constraints

The second major modification to the method involves the step restriction constraints of the optimization. The original method in [1] has a restriction size that is equal for all parameters using equations 3.4 and 3.28. Since the characteristics of the parameters are known, this uniformity in the step restriction is not appropriate. The delay parameter generally has a much larger value than the coefficient values and moves in correspondingly larger steps. Using a common step restriction cannot restrain all the coefficients efficiently and simultaneously. If this uniform step restriction is too small, the delay parameter will not change quickly enough to facilitate rapid convergence. If the step restriction is too large, then the coefficients are not sufficiently restrained and they may move too far and ruin the optimization. This would negate any benefit of a good starting point (much like what can happen using unconstrained optimization). Therefore, the step restriction was altered so that the aspect pertaining to the delay parameter starts around 1, and the step restriction for the coefficients starts between 0.04 and 0.1. A suitable iterative determination of h^{k+1} was found to be as follows:

$$\begin{aligned} &\text{if } r^{(k)} < 0.25, && h^{(k+1)} = h^k/4 \\ &\text{if } r^{(k)} > 0.75 \text{ and } h^{(k)} = \|\delta\|, && h^{(k+1)} = 2h^{(k)}. \end{aligned} \tag{3.36}$$

In an additional change to the step restriction, a reduction of 1 to 10 percent was performed on $h^{(k+1)}$ on every iteration to prevent cycling. By constantly reducing the distance of the step restriction, cycling may be avoided by an increasingly accurate movement direction.

Choosing Better Starting Points

The final modification to the basic method was an algorithm to choose a good starting point. As in other optimization problems, a good starting point reduces design time. To be able to estimate good initial points, results were taken from unconstrained optimization problems and analyzed. A number of solutions from various filter specifications have been taken to make these estimates. As shown earlier in figure 2.1, the poles of the equalizer are usually within the passband region (the sector shown in figure 3.1). Moreover, they are almost evenly spaced along an ellipse. The ellipse becomes more circular (or less eccentric) as the number of sections in the equalizer increases. One can use these results by making the starting point of the optimization a set of poles inside the passband (or passbands in the case of bandstop filters), and evenly spacing them along the perimeter of a circle. With a properly determined radius for this circle, the poles will often be placed very close to the solution. The radius of this circle, however, must be determined. If the equalizer is 'grown' a section at a time as in [1], the radius can be approximated by the solution of the previous result. Either the average or the maximum of the previous pole distance from the centre can be used as the radius of the circle for the current approximation. As the number of sections increases, the approximate radius of the starting point circle becomes more accurate. This approximation gives a good starting point for the coefficient parameters that are dependent on the poles, but does not indicate how to approximate the delay parameter τ_0. If the equalizer has already been grown, then the value of τ_0 in the previous results can be taken to be unchanged. Another

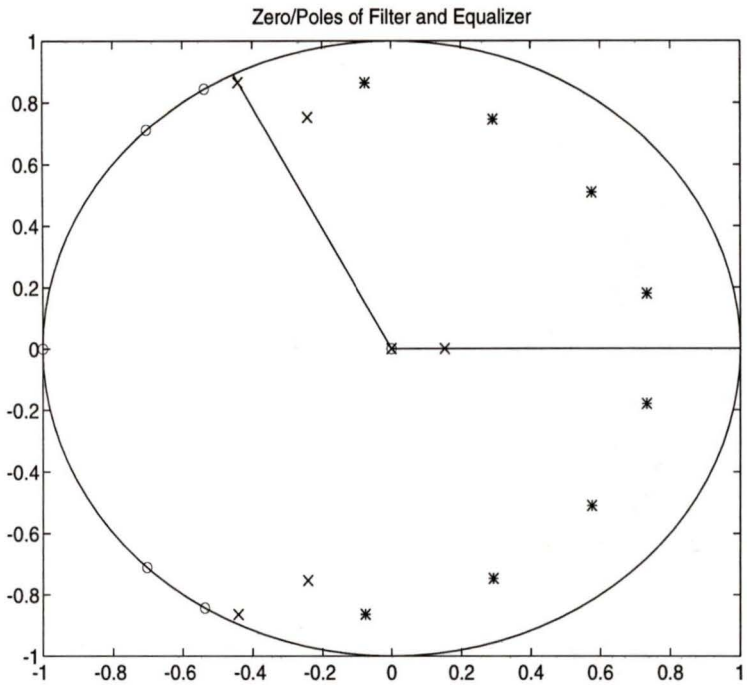

Figure 3.1: Pole placement within passband, o filter zeros, x filter poles, * equalizer poles.

method is to extrapolate a new value for τ_0 by looking at the previous results. These estimates for $\mathbf{c}$ and τ_0 provide good starting values for the optimization parameters.

3.2 An Example

An example illustrating the use of the algorithm for the design of an equalizer will now be described. The problem at hand is to equalize a filter with the following specifications:

5th-order elliptic lowpass filter

passband edge: 0.5 rad/s

passband ripple: 1 dB

minimum stopband attenuation: 30 dB

and the filter equalizer solution must have a Q of 1 or less.

The design was grown with an increasing number of sections, as suggested in [1], with the only modification being the new method of finding each subsequent starting point using the previous solution. At each growth level (with a specific number of equalizer sections), the method uses the constrained optimization technique with the nonuniform sampling technique described in [1] to achieve the desired delay characteristics. The algorithm was implemented in the MATLAB environment. The progress of the design was time consuming: using a Pentium Pro 150MHz personal computer it took about 12 hours to design a 20-section equalizer that satisfies the specifications. The progress in terms of Q is illustrated in figure 3.2. As can be seen, the progress in terms of Q decreases, with some suboptimal (but stable) results occuring along the way to the solution. The final solution is provided in appendix A.

3.3 Analysis of the Constrained Method

There are many similarities in the results using the constrained method with those obtained with the unconstrained method.

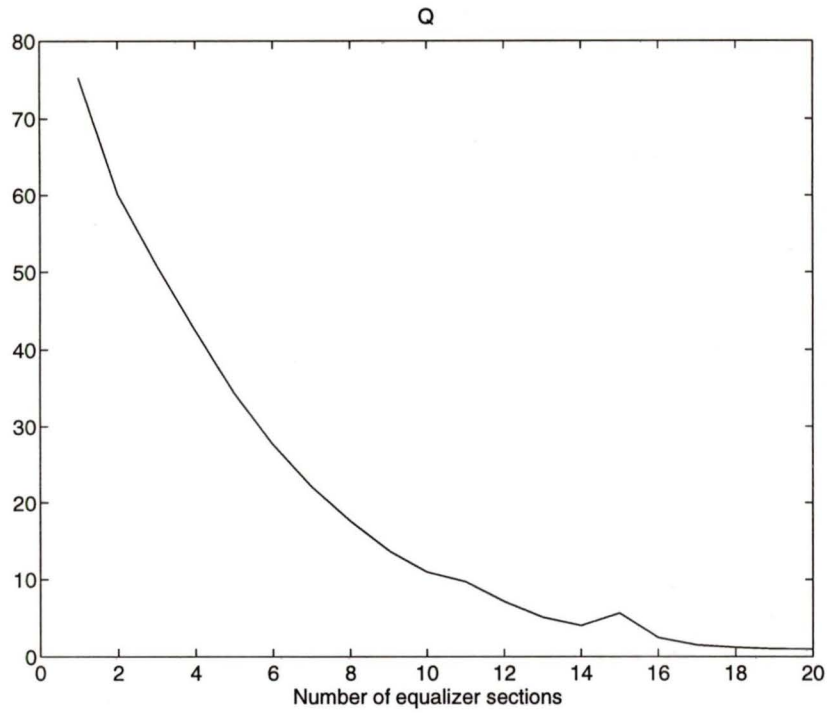

Figure 3.2: The progress of Q for a constrained optimization method.

3.3.1 Objective Function Characteristics

First, many of the traits relating to the objective function are carried over into constrained method, as one would expect. The poles of the designs still form regular structures. The movement of poles onto and along the real axis also remains. Finally, the effect of τ_0 on the optimization is still strong. Nevertheless, there are differences between the constrained and unconstrained method.

3.3.2 Optimization Method Characteristics

Flops

The constrained optimization method requires the calculation of the Hessian and its inverse. These calculations increase the amount of calculation. The number of floating point operations (FLOPS) for one iteration of the constrained and unconstrained optimization methods are compared for increasing sections in figure 3.3. The numbers are not for the most efficient version of the algorithms (the algorithms used had much extraneous code for graphing, etc.), but are for similarly designed ones so that the computational trends can be seen. One can see that the constrained method can get computationally very costly for many sections, although for equalizers with less than six sections, the number of flops for constrained optimizations can actually be less. Certainly, for complex filters, the number of equalizer sections can be high, and the amount of computation may get prohibitive. This will definitely be a consideration in the design process.

Hessian Characteristics

Because the Hessian matrix is used in the constrained method and is required to be positive definite, the circumstances under which the Hessian is non-positive definite were examined. It was noticed that as the value of p is increased, the frequency of a non-positive definite matrix increases. Moreover, as the solution is approached, the frequency of positive definiteness increases. There seems to be a positive definite region around solutions that gets smaller as the value of p grows. This indicates the

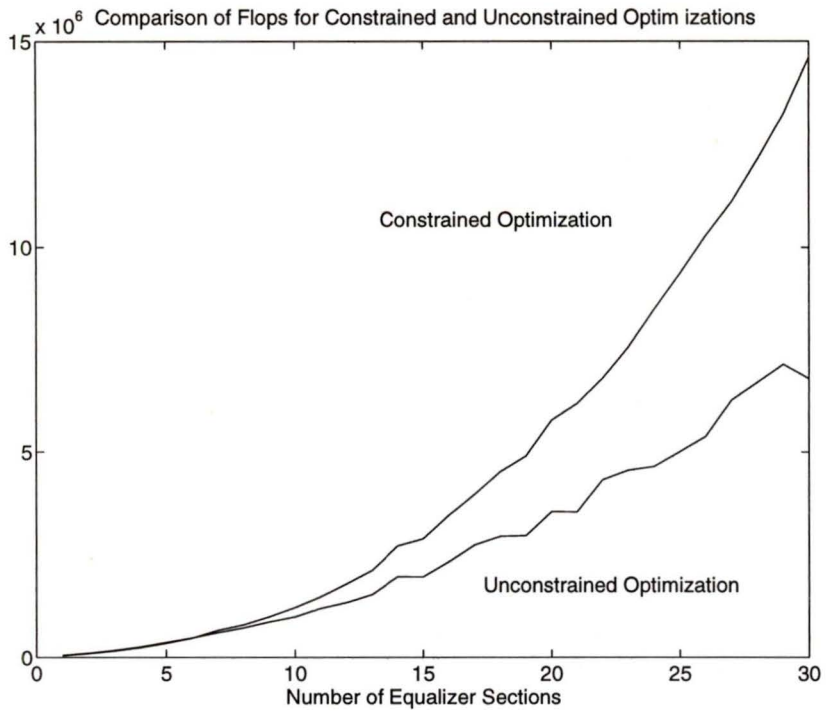

Figure 3.3: Comparison of flops for constrained and unconstrained methods.

problem is more quadratic near the solution.

Stability Constraints

The stability constraints prevent the poles from leaving the unit circle $|z| = 1$ and in doing so, prevent the equalizer from becoming unstable. This was not the ideal result for all situations, however. As mentioned, the poles will occasionally move to the real axis. If one of the poles leaves the unit circle along the real axis when using the unconstrained method (such as shown in figure 3.4), it will stop at the boundary when using the constrained method. The other pole generally heads in the opposite direction inward, or toward the other side of the unit circle. Sometimes, the pole on the boundary moves back off the boundary and may join its pair on the real axis.

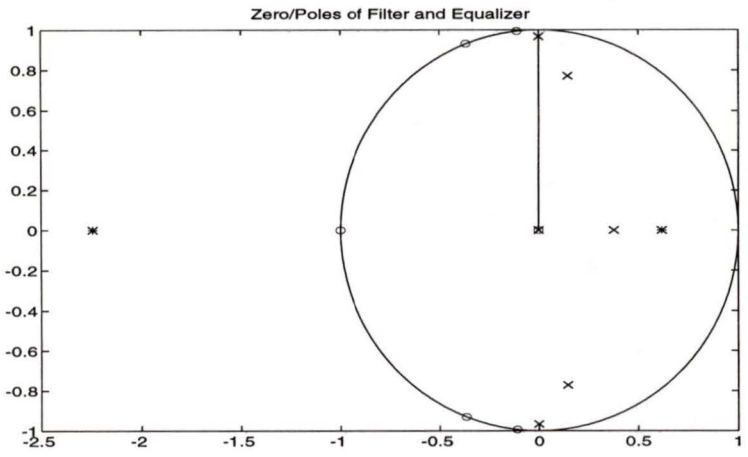

Figure 3.4: Equalizer solution where the real pole has left the unit circle.

The two may also join the other poles to make an efficient equalizer (as described earlier). Often, it does not do this, but remains against the boundary, sometimes not in the passband, and will not be used in an efficient manner. Moreover, when the pole gets close to the unit circle border, the algorithm has problems calculating values, as the condition of the Hessian approaches 0. However, this generally only occurs when starting points are random and never occurred when starting points generated with the method described earlier were used. Although there are cases where poles are at the border, one could conceivably remove them and place them back in the passband and continue the optimization. The stability constraint has shown to be an effective barrier, however, for poles that appear to be heading out of the unit circle. Therefore, despite the non-ideal occurrence of real poles, the constraints still did perform according to its original purpose.

Success Rate using Random Starting Points

Statistics also have shown that the constrained method has a higher success rate, and resulted in a higher number of stable, efficient equalizer designs. A sample of 20 equalizers were designed for equalizers with 1 to 10 biquadratic sections. A set of about 20 random starting points was used with both constrained and unconstrained methods to determine two things. First, a count of how many successes there were for each method was found. Second, an analysis of which method gave better Q values for a given equalizer order was done.

The number of successes meant different things for the constrained and the unconstrained methods. For the constrained method, the poles were always within the unit circle, so unstable filters were impossible. Instead, there were cases where a pair of poles would venture near the circle perimeter. They could not go on the unit circle, because of the buffer zone put on the stability constraints, but in some circumstances when the poles wandered into these zones, the algorithm ceased to continue. In these cases, the optimization was considered a failure. As mentioned earlier, this situation never occurred during normal optimizations using good starting points, but only when random starting points were used. For the unconstrained method, a failure was considered as an optimization that resulted in an unstable filter. Table 3.1 shows the successes for constrained and unconstrained optimization techniques using between 20 and 24 random starting points.

Table 3.1 indicates a substantially better rate of finding results for the constrained optimization method, but it does not indicate how good the results are. Table 3.2

Table 3.1: Success of optimization for random filters

Order	Constrained %	Unconstrained %
1	100	0
2	100	15
3	95.2	0
4	95.2	23.8
5	90.9	18.2
6	95.7	30.4
7	90.9	31.8
8	87.5	41.7
9	90.9	22.7
10	87.5	31.8

shows the minimum values of Q reached using constrained and unconstrained methods. As can be seen, the constrained method yields lower Q's. The value of the tolerance criterion aaplied to Q, ϵ, was the same in the two methods used. Since Q is based on the same criterion in both constrained and unconstrained methods, this comparison is meaningful and appropriate. What the table does not indicate is that the constrained method did not give good values of Q for all cases. One can get a better idea of the results by relating the designs in which the Q achieved is within 10 of the minimum value, as one can see in table 3.3. Even here, however, the constrained optimization technique was found to give better performace.

Table 3.2: Minimum Q values for different filters

Order	Constrained	Unconstrained
1	75.2411	N/A
2	60.0911	60.5061
3	50.8666	N/A
4	42.4805	43.0374
5	34.234	35.2485
6	27.4255	29.1962
7	22.1043	23.284
8	17.5994	17.931
9	13.7631	14.1292
10	10.6323	10.7497

Comparison Using Efficient Starting Points

With random starting points, the constrained optimization seems to be more successful in finding solutions. When incorporated into the full algorithm where the equalizer is grown, and the optimizations use good starting points, the two strategies can be compared in more practical terms. The comparison between the constrained and unconstrained optimizations was intended to compare only the optimization algorithms, as much as possible. That is, if any techniques, such as improved initial conditions, could be used by either algorithm, they were used by both. It was found that if the nonuniform variable sampling technique [1] used a high enough sampling density, then it would perform similarly to the uniform frequency sampling tech-

Table 3.3: Percentage of designs with Q within 10 of minimum Q

Order	Constrained	Unconstrained
1	65	0
2	60	15
3	42.9	0
4	61.9	23.8
5	63.6	18.2
6	60.9	30.4
7	63.6	31.8
8	66.7	41.7
9	86.4	22.7
10	79.2	29.2

nique in both the unconstrained and constrained problems. The benefit, however, was that the nonuniform variable sampling technique took less computations. The only noticeable difference was that the nonuniform variable sampling could occasionally result in a limit cycle within the optimization, that is, the method would not converge. This was solved by reducing the value of h by one to ten percent in every iteration. Having solved this problem, it became worthwhile to use the nonuniform variable sampling technique in both constrained and unconstrained optimizations. The starting points discussed earlier were also used in both the constrained and unconstrained methods, allowing a fair comparison between the algorithms.

When we compare a few designs using constrained and unconstrained methods,

we see several features. A fifth-order lowpass filter with a passband from 0 to 0.5 rad/s, is presented as an example.

The variation of Q for this problem for the unconstrained and constrained optimizations is shown in figure 3.5. The characteristics of the algorithms in terms of Q, according to the figure, are similar. First, one can see that suboptimal solutions are found by both methods. One can conclude this by noticing that although both algorithms have similar levels and rates of decent for Q, neither curve is monotonically decreasing. For the unconstrained method, this inconsistency is caused most often by a set of poles that do not 'fit' with the pattern of the rest of the poles. For instance, while most of the equalizer poles may be on an ellipse, one pair may be within the ellipse as shown in figure 3.6. These stable but suboptimal solutions cause the slight increases in Q noted in figure 3.5. For solutions using the constrained optimization, the higher Q's are more often the result of one or more of the pairs of poles becoming real. This is actually a problem for both the constrained and unconstrained methods, but with the unconstrained method, one of the real poles often leaves the unit circle, rendering the equalizer unstable. This is impossible with the constrained optimization, since the offending pole is very often trapped at the unit circle. Therefore a larger percentage of the suboptimal solutions for the constrained method are of this type.

The curve representing the unconstrained method in figure 3.5 only shows the stable solutions found by the algorithm. If it found these solutions without retrying any optimizations, then the unconstrained method should require much less computation. This assumption can be made because each unconstrained optimization

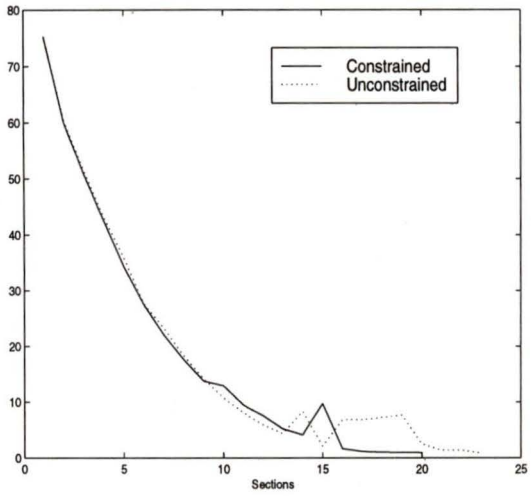

Figure 3.5: Convergence of Q using constrained and unconstrained optimization.

iteration involves reduced computational cost. However, the unconstrained optimization algorithm had many discarded unstable solutions before reaching the stable results shown in the graph. Consequently, the unconstrained method used much more computations.

Both methods are effective for equalizers of 15 sections or less. For 16 sections or more, the unconstrained method results tend to diverge at some levels, and could not meet specifications without retrying an earlier level. For example, it unsuccessfully optimized equalizers with as many as 23 sections before retrying smaller equalizers and finding a solution with a 21-section equalizer. The constrained method though, found a solution using a 20-section equalizer (with a final Q of 0.727%) without having to retry smaller section optimizations. Because of this direct approach, the constrained optimization requires less computation to complete than the unconstrained method for this design. Table 3.4 shows the results obtained for some other designs where the computations were comparable, if not better, using constrained rather

Table 3.4: Equalizer results for different filters

Filter Type	Order, Passband	Order		Flops	
		Con	Unc	Con	Unc
LP	5 0-0.5	20	21	1.1×10^{10}	2.2×10^{10}
LP	5 0-0.25	10	11	1.2×10^{9}	5.9×10^{9}
LP	5 0-0.75	15	16	1.2×10^{10}	3.3×10^{8}
LP	7 0-0.5	20	23	1.3×10^{10}	9.7×10^{8}

than unconstrained optimization. However, there are examples in the table where the specifications were easy enough so that the unconstrained optimization quickly found the solution. The constrained optimization, in these circumstances, still did find a more efficient solution, but at an enormous computational cost. Not all filters will be so simple, though, so the constrained method may still offer the fastest way of getting a stable solution.

Table 3.4 also shows that results using the constrained method are often more efficient than those designed by the unconstrained method. The equalizer entries in the table were obtained to different precisions (Q was between 1% and 10%, where 10% was used for more difficult filters) for each filter to prevent the number of sections from becoming excessive. This resulted in the number of sections for some designs in the table being higher for what should be for simpler filters. Despite the varying specifications, the table still shows that for many of the problems, the constrained solution often uses fewer sections than the solution found by the unconstrained method.

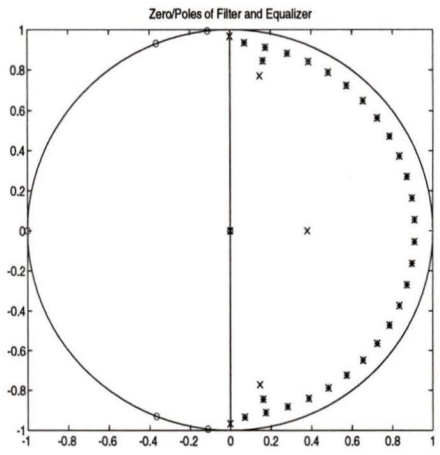

Figure 3.6: Example showing isolated equalizer poles, o: filter zeros, x: filter poles, *: equalizer poles.

3.4 Comments and Recommendations

Having described another method of designing recursive delay equalizers, one can perhaps develop some sort of composite algorithm that can be used to design the equalizers more efficiently. If one could determine when the equalizer optimization has a higher probability of becoming unstable, for example, when the number of equalizer sections is small, the constrained optimization method could be used. In this way, we can keep the amount of computation to a minimum. The unconstrained optimization, which uses less computation, can be used when the equalizer has reached a larger number of sections, and does not have as high a tendancy to become unstable during optimization. Perhaps when the order of the equalizer design is higher, and the precision of the unconstrained optimization method has been reached, the constrained method can be used again to find a lower Q value.

3.5 Conclusions

A constrained optimization technique was described, and used for the recursive delay equalizer problem. The problem extends the least-pth objective function used in the unconstrained optimization method and uses it with linear constraints and some modifications.

The constrained optimization method is found to be computationally less efficient per iteration, but when compared on the entire design process, can compete with the unconstrained method. This is due to the stable designs guaranteed with the constrained method. Fewer re-optimizations are required. The constrained optimization method also appears to give consistently better designs.

This method, or perhaps a composite method with the unconstrained optimization, can be a viable alternative to designing recursive delay equalizers. For practical systems, e.g. consumer products, the cost savings by using an efficient equalized recursive filter (thereby saving DSP MIPS) more than justify the use of an algorithm that yields better results.

Chapter 4

Further Investigations

There are a few issues concerning elliptic biquadratic filter sections and delay equalizers studied previously that warrant further investigation. For instance, the regular pole and zero positions achieved when the equalizer is optimized are particularly interesting. They indicate a relationship between the peaks of the group delay and the zero and pole positions of the equalizers. By examining the effect of individual sections on the group delay, and thereby the effect of conjugate zero and pole pairs, the reasoning behind the regular structures can be determined. The insight into why the delay responds to pole positions the way it does may allow the development of a better starting point or even a closed-form solution for the problem. Therefore, it is important to examine this issue for the many benefits it may have to offer.

4.1 Analysis of the Effect of Pole/Zero Position on the Group Delay

4.1.1 Pole/Zero Positions

To start the investigation, the group delay of the filter is examined. Figure 4.1 shows the zeros and poles of a fifth order filter, and figure 4.2 shows the group delay over the passband with the poles of the filter laid on the cartesian plane (based on its polar coordinates). One can discern a slight peak at $\omega = 0$, one roughly at $\omega = 0.25$ and a final one at the band edge, $\omega = 0.36$. Looking at the number of peaks and the position of these peaks, one can see a correlation to the pole positions of the filter. The group delay peak positions with respect to the frequency correspond to the angles of the poles of the filter. Moreover, the amplitude of the group delay relates to the radius of these same poles. A large group delay corresponds to a pole that is closer to the unit circle, or further from the origin. To be able to further investigate the filter response observations, an examination of individual biquadratic sections in terms of pole and zero positions is made.

The transfer function for biquadratic sections is therefore rewritten in polar form for the numerator and denominator to see how and why the relationship described earlier happens. The numerator and denominator zeros are defined as

$$\begin{aligned} &\text{zeros: } r_{aj}e^{j\theta_{aj}}, r_{aj}e^{-j\theta_{aj}}, j = 1 \ldots M \\ &\text{poles: } r_{bj}e^{j\theta_{bj}}, r_{bj}e^{-j\theta_{bj}}, j = 1 \ldots M \end{aligned} \tag{4.1}$$

where M is the number of biquadratic sections, r denotes the radius and θ the angle

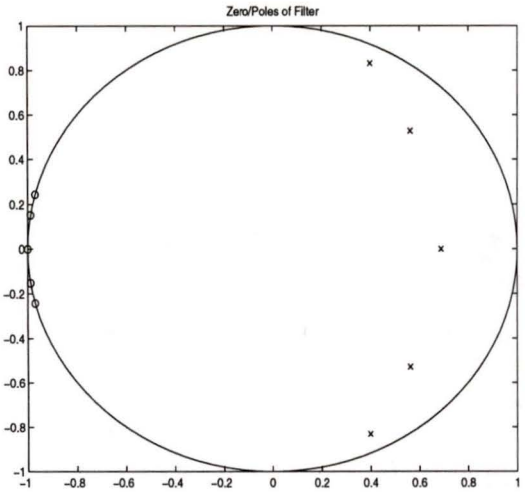

Figure 4.1: Fifth order lowpass filter, x = poles, o = zeros.

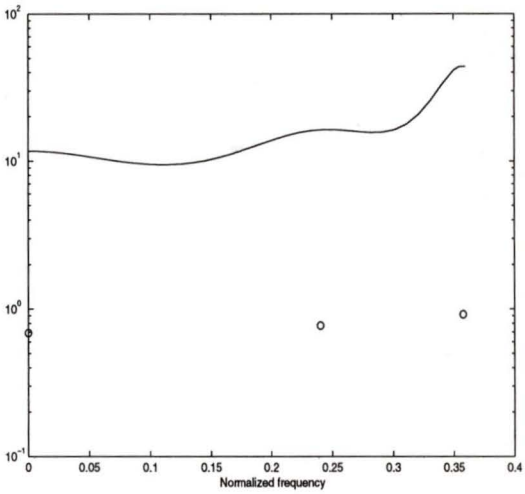

Figure 4.2: Fraction of group delay equation over passband for 5th-order lowpass filter.

for the poles and zeros. The transfer function of the filter is then

$$H_F(z) = \prod_{j=1}^{M} \frac{(z - r_{aj}e^{j\theta_{aj}})(z - r_{aj}e^{-j\theta_{aj}})}{(z - r_{aj}e^{j\theta_{bj}})(z - r_{aj}e^{-j\theta_{bj}})} = \prod_{j=1}^{M} \frac{z^2 - 2zr_{aj}\cos(\theta_{aj}) + r_{aj}^2}{z^2 - 2zr_{bj}\cos(\theta_{bj}) + r_{bj}^2}. \quad (4.2)$$

To quickly get the group delay, a comparison of equation 4.2 with equation 2.12 is made, revealing the relationships

$$a_{2j} = 1 \quad (4.3)$$

$$a_{1j} = -2r_{aj}\cos(\theta_{aj}) \tag{4.4}$$

$$a_{0j} = r_{aj}^2 \tag{4.5}$$

$$b_{2j} = 1 \tag{4.6}$$

$$b_{1j} = -2r_{bj}\cos(\theta_{bj}) \tag{4.7}$$

$$b_{0j} = r_{bj}^2. \tag{4.8}$$

When these relationships are substituted into equation 2.7, the group delay equation

$$\begin{aligned}\tau_F = &-T\sum_{j=1}^{M}\frac{1-r_{aj}^2+(-2r_{aj}\cos\theta_{aj})(1-r_{aj}^2)\cos\omega T}{(1-r_{aj}^2)^2+4r_{aj}^2\cos^2\theta_{aj}-4r_{aj}\cos\theta_{aj}(1+r_{aj}^2)\cos\omega T+4r_{aj}^2\cos^2\omega T}\\ &+T\sum_{j=1}^{M}\frac{1-r_{bj}^2+(-2r_{bj}\cos\theta_{bj})(1-r_{bj}^2)\cos\omega T}{(1-r_{bj}^2)^2+4r_{bj}^2\cos^2\theta_{bj}-4r_{bj}\cos\theta_{bj}(1+r_{bj}^2)\cos\omega T+4r_{bj}^2\cos^2\omega T}\end{aligned} \tag{4.9}$$

is obtained, where T is the sampling period.

One can immediately see the similarity between the first term which relates to the numerator, and the second term, which relates to the denominator of the transfer function. Examining the common form

$$\frac{1-r^2+(-2r\cos\theta)(1-r^2)\cos\omega T}{(1-r^2)^2+4r^2\cos^2\theta-4r\cos\theta(1+r^2)\cos\omega T+4r^2\cos^2\omega T} \tag{4.10}$$

will provide insight into how the radius and the angle of the zero of the numerator and denominator affect the group delay. Of course, equation 4.9 shows that the numerator zeros will have the opposite effect on the group delay with respect to the denominator zeros. The equalizer delay shown in the following chapter (equation 5.2) also contains this common fraction. In that case, however, the two delays due to the numerator and denominator are added, instead of the numerator delay being subtracted from the denominator delay. This difference is due to the reversal of the coefficients in the numerator of the equalizer, producing a similar, but negative version of the common

fraction. When the delay of the numerator and denominator of the equalizer are combined, the simplified result is essentially twice the common fraction.

4.1.2 Peak Position

First, an examination of the proximity of the peak of the group delay in equation 4.9 to the zero angle is made. The peak position can be resolved by examining the equations, but the reason for its placement is best described graphically. Both are attempted here.

The exact peak position can be found by finding where the derivative of the group delay with respect to angle is zero. The derivative is

$$\begin{aligned}\frac{d\tau_E}{d\omega} &= (2r_E\cos\theta_E)(1-r_E^2)((1-r_E^2)+4r_E^2\cos^2\theta_E)\sin\omega T \\ &\quad -4r_E\cos\theta_E(1+r_E^2)\sin\omega T \\ &\quad +4r_E^5\cos\theta_E(1+r_E^2)\sin\omega T \\ &\quad +(8r_E^2-r_E^6)\cos\omega T\sin\omega T \\ &\quad -8r_E^3\cos\theta_E(1-r_E^2)\cos^2\omega T\sin\omega T. \end{aligned} \tag{4.11}$$

Setting this to zero and rearranging, we get

$$\begin{aligned}0 &= 2*r_E\cos\theta_E((1-r_E^2)^3+4r_E^2(1-r_E^2)\cos^2\theta_E-2(1+r_E^2)+2r_E^4(1+r_E^2)) \\ &\quad +(8r_E^2-8r_E^6)\cos\omega T \\ &\quad +8r_E^3(1-r_E^2)\cos\theta_E\cos^2\omega T \end{aligned} \tag{4.12}$$

which can be solved for $\cos(\omega)$ by using the quadratic equation, and then we take the inverse cosine to give ω. This gives the exact position of the peak but it is not

very intuitive nor does it indicate why the peak occurs so close to the pole angle.

To show why there is a peak, a graphical approach is used assuming a sampling frequency of 2 (or T = π second). The single equalizer section being analyzed has a pole at position 1 in figure 4.3 with a positive imaginary part, and a conjugate pole at position 2. The group delay being examined is over the frequency range ω = 0 to 1. A frequency response at frequency ω can be obtained by evaluating the transfer function on the unit circle at a point w at angle θ, such as point w as shown in figure 4.3. If the phasor angle is normalized to 2, the frequency at point w will be equal to the angle of the phasor from the origin to point w. To calculate the group delay from ω = 0 to 1, point w is moved on the unit circle from the right side in the counterclockwise sense. The group delay at frequency ω is inversely related to the distances between that of point w and each of the two poles. If the distance between point w and either pole becomes very short, the amplitude of the group delay becomes very large. Naturally, of the two poles, the pole with the positive imaginary part will have the most effect on the group delay, since it will be closer to point w. If this pole is very close to the unit circle, then at some frequency ω_p the distance between the pole and the unit circle will be very short. This frequency value should be close to the pole angle value. The peak of the group delay, therefore, should be near this frequency, but the conjugate pole also affects the group delay. Because the distance to this second pole is normally much greater, the shift of the group delay peak away from this pole angle is usually slight. One will notice, though, that as the frequency gets close to 0 or 1, the effect of the conjugate pole becomes much stronger, and the peak in the group delay gets shifted much more. This happens

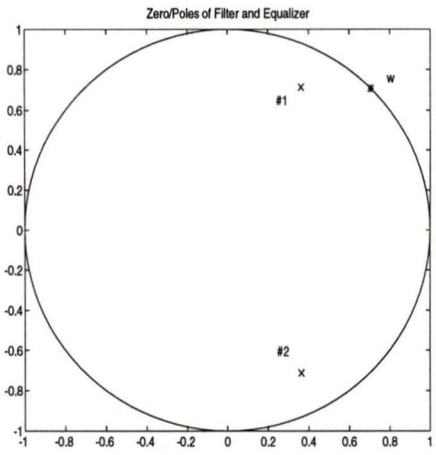

Figure 4.3: Zeros for single section showing relative distances to frequency angle.

because pole 2 becomes almost as close to point w as pole 1. Despite this interaction from the conjugate pole, the proximity of pole 1 to point w is the reason why group delay peak frequencies occur near the pole angle θ_p.

4.1.3 Peak Amplitude

To investigate the amplitude of the peak in the group delay, a few examples are examined. First, the effect of a pole with a radius of 0 can be computed immediately from equation 4.9 to have a constant value of 1 over the whole frequency range. This is naturally expected since a zero or pole of the biquadratic filter section at the origin would be expected to do nothing. Referring back to the graphical example, as the frequency response over ω (around the unit circle) is taken, the distance to the pole remains constant.

As the zero pair moves away from the origin, the group delay over the normalized frequency range forms a peak at roughly the zero angle. This is shown in figure 4.4

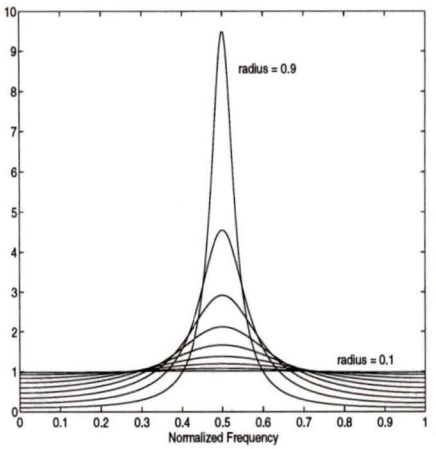

Figure 4.4: Equation 4.13 evaluated for poles of radii from 0.1 to 0.9.

for a set of zeros with varying radii from 0.1 to 0.9, and a normalized pole angle of 0.5. As the radius goes closer to 1, the magnitude of the peak increases exponentially. The lows of the expression over the frequency range also approach 0. An examination of the equation shows the actions in the equation which give these results. By making $\theta = 0.5$, $\cos \theta T = 0$, which simplifies equation 4.10 to

$$\frac{1 - r^2}{(1 - r^2)^2 + 4r^2 \cos^2 \omega T}. \tag{4.13}$$

The numerator, $1 - r^2$, is always less than 1. The denominator has the same term squared, plus another term. If $\omega = 0.5$, the other term also equals zero from the $\cos \omega T$ term, ensuring a peak that grows (since the denominator approaches 0 faster than the numerator). If the frequency term, ω is off the peak position ($\omega = 0.5$), the term $4r^2 \cos^2 \omega T$ becomes significant, and quickly reduces the expression closer to 0 as r is closer to 1.

If the zero is not at $\omega = 0.5$, the peak of the delay over the normalized frequency range changes. Figure 4.5 shows the variation of the delay over the baseband for

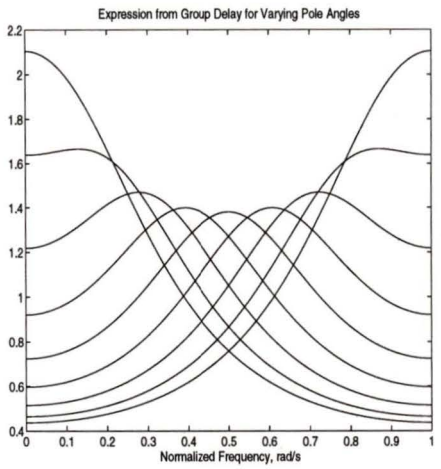

Figure 4.5: Equation 4.13 evaluated for poles having angles from 0.1 to 0.9 rad/s.

zeros with a constant radius of 0.3 but varying angle. As the pole angle θ moves to either side of $\omega = 0.5$, the peak amplitude increases.

Basically, the amplitude of the peak of the group delay is dominated by an inverse relation to $1 - r^2$.

4.1.4 Curve Shape

The pole's position also has an effect on the shape of the curve derived from equation 4.10. The radius of the pole affects the width of the bell on either side of the peak. The angle of the pole, θ, affects the skew of the bell.

As seen earlier in figure 4.4, a pole's radius close to the unit circle causes a very high peak amplitude. What one will also notice is that the width of the bell formed by the peak gets thinner as the radius gets closer to 1. If one compares the points at which the group delay drops below unity on either side one sees that the angle at which this happens gets closer to the peak angle as the pole gets closer to the unit

circle.

One will also notice in figure 4.5 that the bell of the expression is generally skewed. The bell is symmetric only when $\theta = 0.5$. If θ is to either side of 0.5 (in normalized frequency terms), the curve is skewed to either 0 or 1.

The width and skew can be precisely determined. The points where the group delay equals unity can be calculated by equating the expression in equation 4.10 to unity. We get

$$0 = (-2r^2 + 2r^4 + 4r^2\cos^2\theta) + (-2r\cos\theta)(1 + 3r^2)\cos\omega T + 4r^2\cos^2\omega T \quad (4.14)$$

that can again be solved for $\cos\omega T$. Although the value of unity has been chosen for the expression, one will notice that when the pole angle is close enough to either 0 or 1/s, the value in equation 4.10 does not reach unity on both sides of the peak. Other values can be substituted to observe the width behaviour or skew behaviour of the group delay curve.

For the case of the zeros of the biquadratic section, the radius is always 1 for the elliptic filter. And although the zero angle varies, this radius always leads to a constant group delay of 0. In computer simulations, numerical rounding may lead to some discontinuities around the pole angle, but by using L'Hôpital's rule, the group delay at $\omega = \theta$ can be shown to be 0. This is not usually an issue, though, since the zeros of the filter are in the stopband, the passband group delay is of interest in most cases.

4.1.5 Combining the Sections

The group delay equations show that the group delay curves for the individual sections that are added to yield the total equalizer group delay. There is always some group delay at any frequency (albeit sometimes very minute) for any section, and this explains why the solution values of τ_0 increase as the number of sections in the equalizer optimization are increased.

There are also differing effects on the curve shape that occur when multiple sections are combined in different ways. Generally, there are two different ways in which the sections can be combined. The first is if the pole angles of the sections are equal, and the second method is if the poles are at different angles.

Sections whose Poles are at the Same Angle

If the poles of two sections are at the same angle, the two peaks add amplitude to a single peak. When compared to a similar amplitude peak due to a single pole, the poles' radii do not have to be as large. This can be seen in figure 4.6, where a single pole nearest the band edge would have normally had a radius close to the largest filter pole radius. In this case, the two poles' radii are not as large. Note, however, that the bell formed by the group delay of the doubled poles (as shown in figure 4.7) is wider, as predicted earlier due to their smaller pole radius. This explains why these doubled poles occupy larger sections of the passband, as compared to the other poles of the solution. Solutions with poles in this configuration generally indicated that the convergence tolerance for the optimization was not small enough. This is observed during most optimization progressions, where, if the poles get in this situation, they

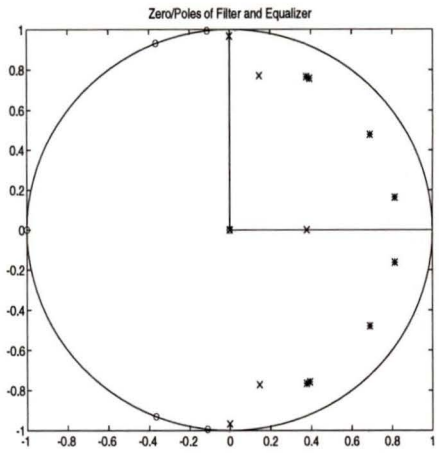

Figure 4.6: Two poles at the same angle.

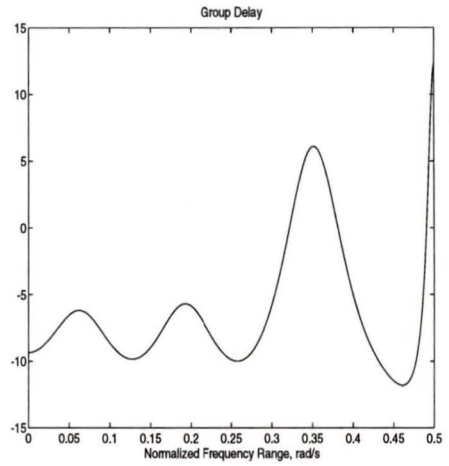

Figure 4.7: Group delay for two poles at the same angle.

usually move out of it into the regular structure described earlier as the "efficient equalizer structure".

Sections with Poles at Different Angles

If the poles of the two sections are at different angles and roughly of the same radii, the group delay has two distinct peaks. This is the way the poles are placed in the

efficient structure described in Chapter 2. Because the contributions from these poles are additive, the curve for one pole has two effects on the group delay of the other pole's curve. The group delays are increased, and the peaks are skewed and shifted. If, for instance, pole 1 is at one angle with a small radius (and therefore a smaller group delay, but wider bell), and pole 2 is near it with a larger radius, the peak in the vicinity of pole 2 will have its peak raised. It will also be slightly skewed (since one side of the contributing group delay is higher than the other). This skewing shifts the peak of the curve slightly toward the other peak. Depending on the radius of pole 2, the effect on pole 1 is very different. Since a pole with a radius close to 1 affects a very narrow region, it may not contribute to the shape of the other section's group delay significantly. However, if the radius is not too close to unity and the other section is close enough in frequency, the section containing pole 1 will have a similar, yet more severely skewed, group delay. Moreover, the absolute level of the group delay will increase, albeit slightly.

The height of the peaks in the group delay depend not only the peak of the individual sections, but also on the amount that the side lobe of the other section's group delay contributes. Also, if the two sections have poles that are both close to unity, and the poles are reasonably close together, the effect of the skew is much stronger. This is most prominently seen in a difficult filter design (an example of which is illustrated in figure 4.8) that has a highly-peaked group delay at its band edge. With elliptic filters, the pole often sits on or very close to the band edge and the peak for the curve is right on the band edge. When the equalizer has enough sections, eventually, a section has a pole with a radius approaching unity and it is

near a filter pole. The skew contributed by the equalizer pole on the filter pole is strong enough to bring the peak of the filter group delay right into the passband. Experimentation has shown that using two poles at different angles provides better results. That is, the difference between the minimum and maximum group delay is minimal. Moreover, if the poles are at different angles, the equalizer has better flexibility since some parts of the filter group delay may not need as much equalization as others.

Optimal Structures

The aforementioned reasons might explain why the optimal structures tend to be so regularly spaced along an ellipse. The even spacing is required to provide the peaks along the group delay where needed, and the poles are placed at radii that are required to level the group delay to the highest group delay peak of the filter and equalizer combination. Because the equalizer poles tend to be close to the unit circle, the effect of one section on others tends to be diminished (because of the thin bell shape), but as mentioned, the cumulative effect on the total group delay can still be present. For instance, the filter may have a peak group delay of about $\tau_F = 50$ but the combination of the filter and equalizer may have a group delay of $\tau_{FE} = 100$.

4.1.6 Filter and Equalizer Implications

The preceding analysis provides some insight into the equalizer problem. The filter has a fixed group delay, with peaks that depend on the position of the poles of the filter. For elliptic filters, the poles near the passband edges tend to have the largest

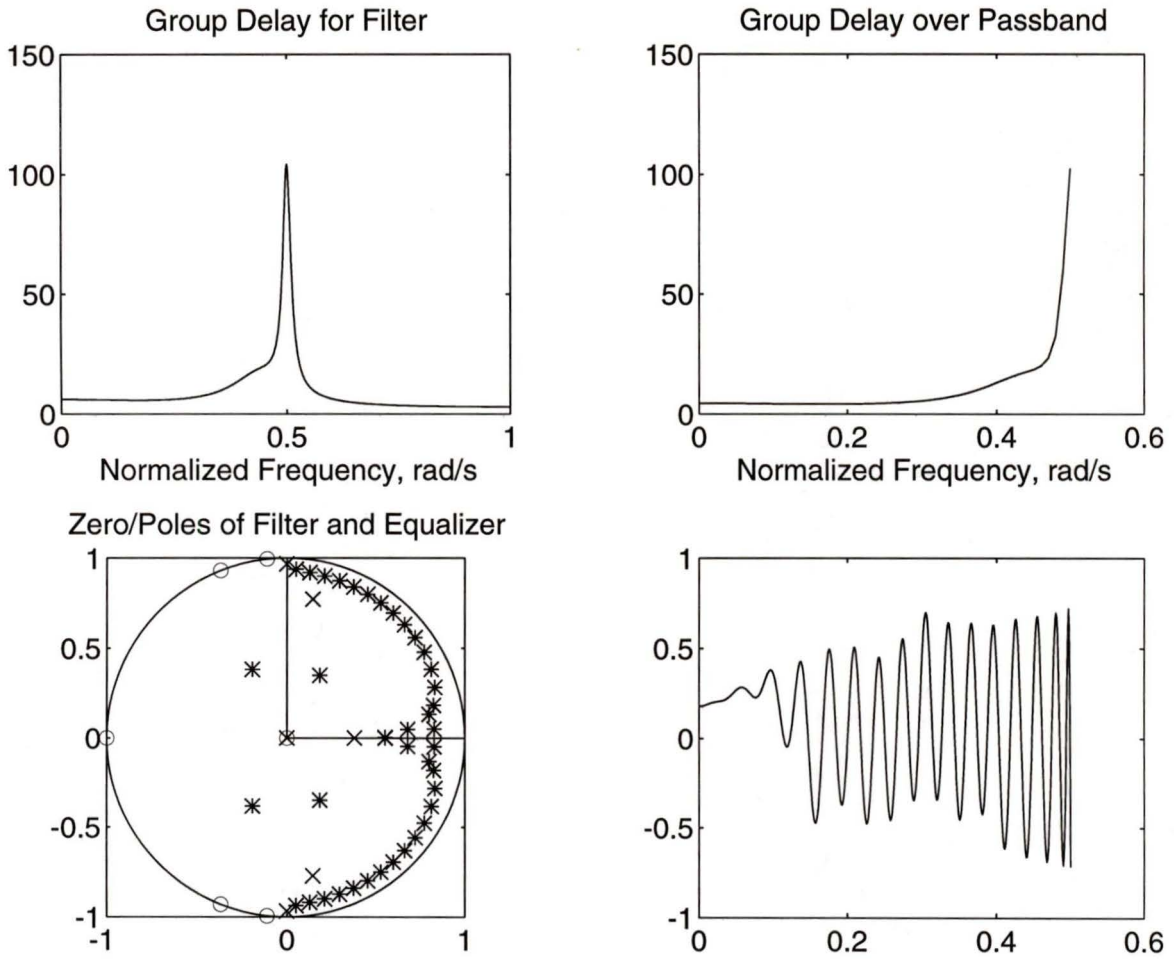

Figure 4.8: Example of shift in group delay peak. a) Group delay of filter, b) group delay of filter over passband, c) poles of filter and equalizer, d) group delay of filter and equalizer over passband.

radius. Moreover, the zeros of elliptic filters have no effect on the passband group delay. There are two reasons for this. The first is that all the zero angles are outside the band of interest. The second reason is that the zeros of elliptic filters lie on the unit circle. Because they do so, they have an infinitely thin bell within which they affect the group delay. The objective of the design of the equalizer, therefore, is to

place equalizer poles near the filter poles that form the maximum peak, and continue adding sections (and therefore poles) until the individual peaks yield a group delay response that meets specifications.

From the previous results, a new Q can be proposed. Moreover, they suggest that it may be possible to calculate a better starting point.

4.1.7 A New Q

The current Q from equation 2.1 is calculated as the maximum variation of the overall filter and equalizer group delay divided by the sum of the minimum and maximum group delays. However, as computations and the previous analysis have indicated, as the number of equalizer sections increases, the group delay of the equalizer tends to increase. Therefore, Q's denominator, which is the combination of the minimum and maximum group delay increases. This indicates a built-in decreasing of Q as the number of equalizer sections is increased. A better indication of how the equalizer is working would be to keep the numerator, but change the denominator to the sum of just the filter minimum and maximum group delay, or

$$Q_2 = \frac{100(\hat{\tau}_{FE} - \check{\tau}_{FE})}{\hat{\tau}_F + \check{\tau}_F} \tag{4.15}$$

which only compares the equalized to the original group delay. The Q_2 value tends to be higher as shown by the comparison in figure 4.9 for a seventh-order lowpass elliptic filter. The passband ripple of the elliptic filter is 1dB and the attenuation is 50dB. As the order of the equalizer increases, the new Q can be seen to increase significantly when compared to the old Q. This indicates how the group delay spread

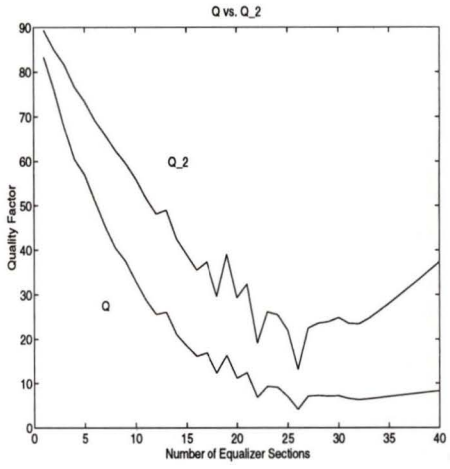

Figure 4.9: Comparison of old and new Q's.

is growing in absolute terms, rather than the relative terms indicated by the old Q. For consistency, however, the old Q will be used for the remainder of the comparisons in this thesis.

4.1.8 A Better Starting Point

On the basis of the analysis provided, a method may be developed that could either find a better starting point, or even find a solution to the recursive delay equalizer problem without optimization. By looking at the filter group delay, one can determine where the poles of the equalizer will be needed. Working with a Q_2 value from the specifications of the filter, one can get a required value of the difference in the total group delay using the equation

$$\hat{\tau}_{FE} - \breve{\tau}_{FE} = \frac{Q_2}{100}(\hat{\tau}_F + \breve{\tau}_F). \tag{4.16}$$

Using the required difference between the required group delay and the current group delay, one can get a point at which to modify the current group delay. For instance,

to add the first equalizer section, one would start at the peak with the highest group delay amplitude in the passband and find the point at which it dips below the peak minus the value from equation 4.16. Point A in figure 4.10 shows such a point. The first equalizer section can have its pole somewhere past this point. This point would be determined using the width and peak information already discussed. The difficulty of this process would be in balancing the two parameters with the group delay of the filter. Still, it may be possible to determine a proper angle and radius at which to add the equalizer section. This idea can be continued to force the group delay within the specified range over the entire required frequency range with more and more sections, and, eventually, a good starting point can be obtained. Extending this idea, one may obtain a solution of the equalization problem.

4.2 Recommendations

In some optimization problems, one may encounter a filter with a pole near the unit circle. As mentioned, this pole will contribute a large spike to the group delay. In elliptic filters, this pole also occurs near the passband edge. One might take advantage of this fact by altering the problem slightly. Instead of designing the filter to the specifications, and then finding an equalizer afterwards, one could find a filter with slightly different specifications, one that might have an easier equalizer to optimize. The strategy here would be to change the filter design so that the problem specifications for the magnitude response are stell met, but the group delay spike lies outside the area of interest for the optimization. For instance, if the problem requires

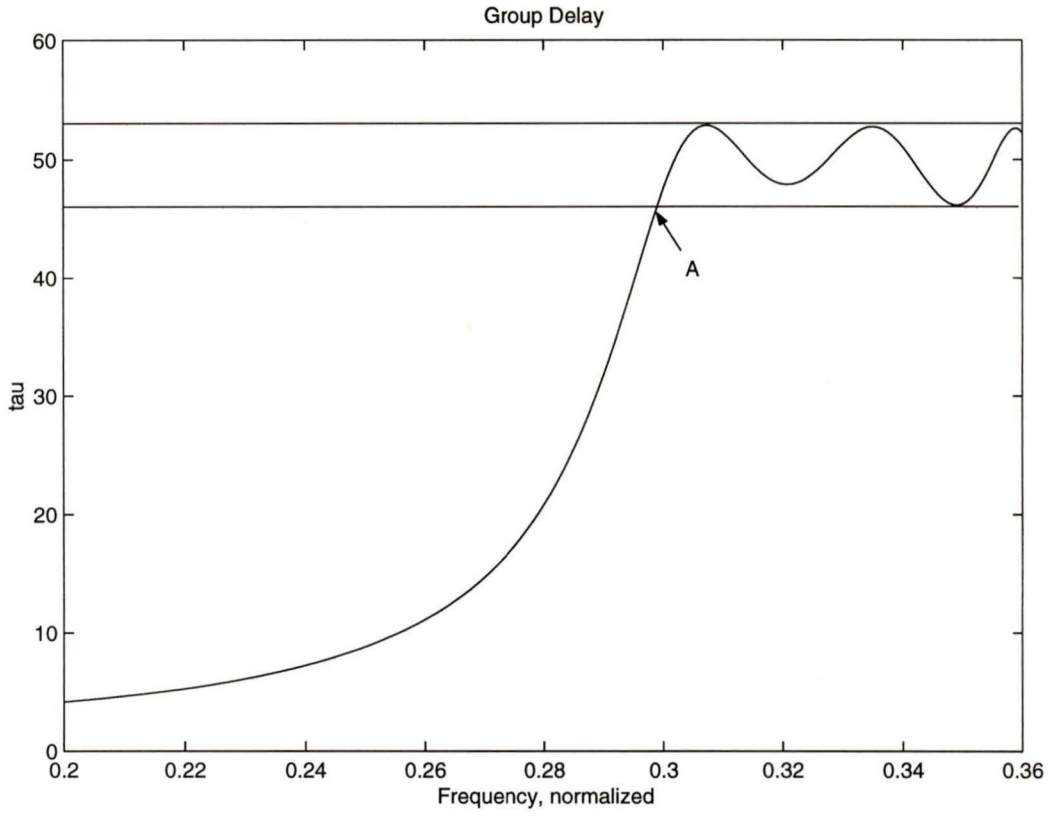

Figure 4.10: Point at which group delay drops below minimum required group delay.

a lowpass filter with a passband edge at $\omega = 0.5$, with linear phase also over that range, a lowpass filter whose passband is from $\omega = 0$ to $\omega = 0.55$ could be used. The filter then has its outermost pole outside the band of interest, and the group delay complexity is significantly reduced. As an example, a case of a fifth-order filter, that has a passband edge at $\omega = 0.5$ where the passband ripple is 1 dB and the minimum attenuation is 30 dB is used. Figure 4.11a shows the group delay for the original filter and figure 4.11b shows the group delay for the modified filter. The peak of the group delay is reduced from around 100 to around 18. In Chapter 3, it was shown that a 20 section equalizer was required to equalize the filter to a Q of 1%. This is a large overall system. Simply by changing the filter passband edge to $\omega = 0.55$, as

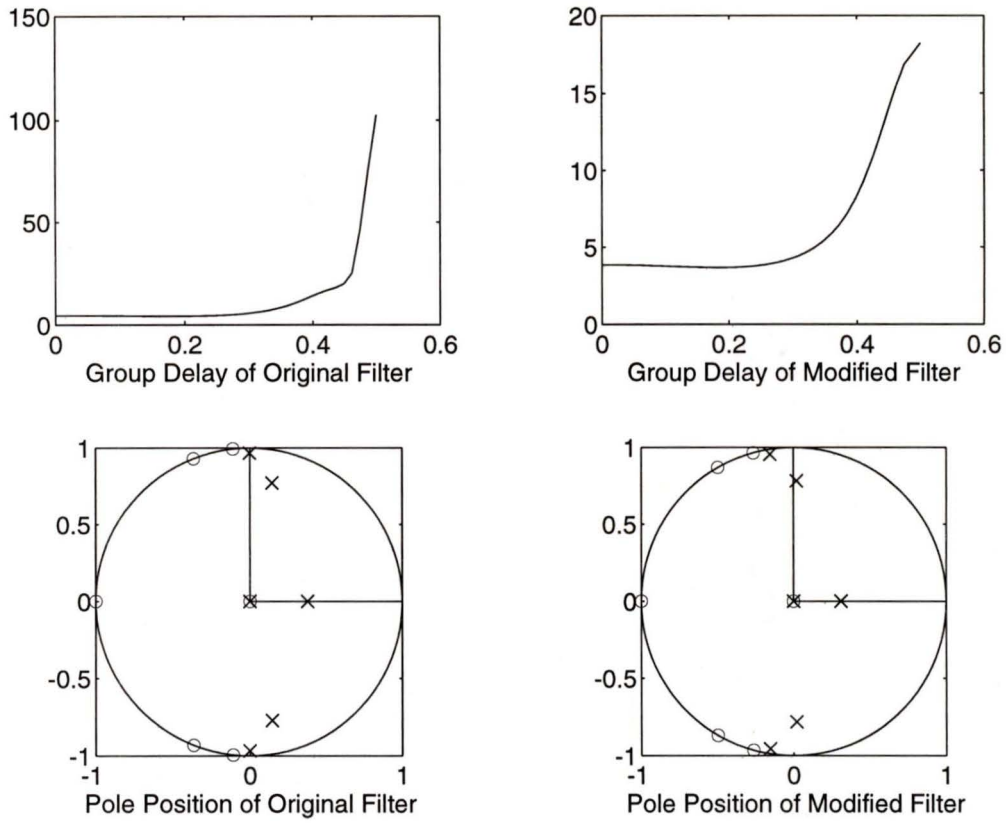

Figure 4.11: Comparison of group delays for original lowpass filter and modified lowpass filter.

shown in figure 4.11c and d, the equalizer size is reduced to 3 sections!

The exact amount that the altered poles of a filter needs to be shifted depends on the filter order, as higher-order filters tend to have more severe spikes, and need less shifting to give significant savings. A study into the combination may provide a strategy to reduce the overall size of the filter and equalizer.

4.3 Conclusions

The effect of the biquadratic transfer functions on the group delay has been analyzed in terms of the pole positions. The pole position was found to affect the group delay in many ways. The pole radius is directly related to the group delay amplitude peaks. The pole angle also strongly affects the position of the peak in the normalized frequency range. Moreover, the pole angle was found to affect the distribution and skew of the group delay curve.

On the basis of these observations, a new Q has been proposed that is not affected by an increasing value of τ_0. Also, a rationale why regular pole structures are solutions to the recursive delay equalizer problem has been developed. This analysis can be used to generate a better way of estimating starting points for the optimization or, perhaps, even obtaining a complete solution.

Chapter 5

New Objective Function

In Chapter 4, the importance of the pole positions of the equalizer on the group delay was emphasized. The characteristic of the attraction of the poles to the real axis using the coefficient-based objective function also warrants looking at an alternative objective function. This leads us to try reformulating the equations from being based on coefficients to being based on pole positions. By reformulating the error function of the objective function in terms of the pole radii and angles, some advantages are gained. First, the real axis attraction is eliminated. Second, the constraints are reduced to the form $0 < r < 1$ where r is the radius of the equalizer biquadratic section. One can also, if needed, easily add another set of constraints to maintain the poles within the passband of the equalizer.

5.1 Derivation of New Objective Function

The new approach to the optimization problem is to formulate the problem in terms of the radii and angles of the poles of the equalizer instead of the coefficients. The least-pth objective function, gradient and Hessian from equations 2.27, 2.29, and 3.12 are still used, but a different error function needs to be defined. To do this, the error function must use a polar representation of the transfer function. The transfer function for the recursive biquadratic equalizer is rewritten as

$$H_E(z) = \prod_{j=1}^{M} \frac{1 - (2r_{Ej}\cos\theta_{Ej})z + r_{Ej}^2 z^2}{r_{Ej}^2 - (2r_{Ej}\cos\theta_{Ej})z + z^2} \tag{5.1}$$

where M is the number of biquadratic sections. The group delay is

$$\tau_E(\omega) = \\ 2T\sum_{j=1}^{M} \frac{1-r_{Ej}^4-(2r_{Ej}\cos\theta_{Ej})(1-r_{Ej}^2)\cos\omega T}{(1-r_{Ej}^2)^2+4r_{Ej}^2\cos^2\theta_{Ej}-(4r_{Ej}\cos\theta_{Ej})(1+r_{Ej}^2)\cos\omega T+4r_{Ej}^2\cos^2\omega T} \tag{5.2}$$

which contains the expression of equation 4.10 from the last chapter and will consequently have the same characteristics. As mentioned, the group delay for the equalizer differs from the filter group delay in that it only has the pole parameters (no zero parameters), and the delay components are additive, and not subtractive, as they are in the filter expression.

The gradient, which will be required for both constrained and unconstrained optimization methods, is

$$\nabla e_i(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_i(\mathbf{x})}{\partial r_{E1}} \\ \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1}} \\ \vdots \\ \frac{\partial e_i(\mathbf{x})}{\partial \tau_0} \end{bmatrix}. \tag{5.3}$$

To express the gradient more simply, the group delay function from equation 5.2 is written as

$$\tau_E(\omega) = 2T \sum_{j=1}^{M} \frac{\tilde{C}_j}{C_j}. \tag{5.4}$$

This allows the partial derivative of the error function with respect to the radii of the poles to be written as

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{Ej}} = 2\frac{X_{0j} + X_{1j}\cos\omega T + X_{2j}\cos^2\omega T + X_{3j}\cos^3\omega T}{C_j^2} \tag{5.5}$$

where

$$X_{0j} = 4r_{Ej} - 8r_{Ej}^3 + 4r_{Ej}^5 - 8r_{Ej}\cos^2\theta_{Ej} - 8r_{Ej}^5\cos^2\theta_{Ej} \tag{5.6}$$

$$X_{1j} = (2 - 14r_{Ej}^2 + 14r_{Ej}^4 + 2r_{Ej}^6)\cos\theta_{Ej} + (8r_{Ej}^2 + 8r_{Ej}^4)\cos^3\theta_{Ej} \tag{5.7}$$

$$X_{2j} = -8r_{Ej} - 8r_{Ej}^5 - 32r_{Ej}^3\cos^2\theta_{Ej} \tag{5.8}$$

$$X_{3j} = 8r_{Ej}^2(1 + r_{Ej}^2)\cos\theta_{Ej}. \tag{5.9}$$

Similarly, the partial derivative with respect to θ_{Ej} becomes

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{Ej}} = 2\frac{Y_{0j} + Y_{1j}\cos\omega T + Y_{2j}\cos^2\omega T + Y_{3j}\cos^3\omega T}{C_j^2} \tag{5.10}$$

where

$$Y_{0j} = 8r_{Ej}^2\cos\theta_{Ej}\sin\theta_{Ej}(1 - r_{Ej}^4) \tag{5.11}$$

$$\begin{aligned} Y_{1j} = {} & -2r_{Ej}\sin\theta_{Ej} - 10r_{Ej}^3\sin\theta_{Ej} + 10r_{Ej}^5\sin\theta_{Ej} \\ & +2r_{Ej}^7\sin\theta_{Ej} - 8r_{Ej}^3\sin\theta_{Ej}\cos^2\theta_{Ej} + 8r_{Ej}^5\sin\theta_{Ej}\cos^2\theta_{Ej} \end{aligned} \tag{5.12}$$

$$Y_{2j} = 0 \tag{5.13}$$

$$Y_{3j} = 8r_{Ej}^3\sin\theta_{Ej} - 8r_{Ej}^5\sin\theta_{Ej}. \tag{5.14}$$

If one compares these equations to those for the coefficient-based method, these equations require more computations.

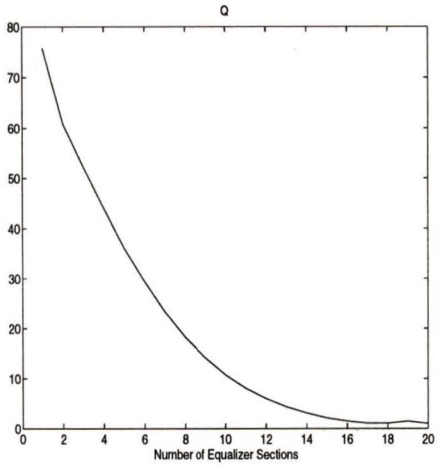

Figure 5.1: Convergence of Q for the pole-position-based objective function with unconstrained optimization.

5.2 Results with Unconstrained Optimization

5.2.1 An Example

This new formulation can be applied to the design of a filter satisfying specifications used in previous examples, namely, a fifth-order lowpass elliptic filter with a passband edge at $\omega = 0.5$ rad/s is equalized. The passband ripple for the filter was 1 dB and the stopband attenuation was 30 dB. Figure 5.1 illustrates the progress of the algorithm, and shows that if the tolerance level were a bit more strict ($\epsilon = 1 \times 10^{-2}$ for this example), the solution may have been an 18-section equalizer! However, to be able to compare the present with the previous algorithms, the tolerance level was kept the same as before. As a result, a 20 section equalizer was designed.

The algorithm was quite fast, as it took a very direct route to the solution at most levels, despite the increased amount of computation to get the objective function and

the gradient values.

5.2.2 Analysis

At this point, we see that the unconstrained method can be successfully used on the problem using the new version of the objective function. The real axis attraction that plagued the old objective function was eliminated. This let the optimization be more direct instead of being attracted to the real axis. The new method was also observed to be able to handle more difficult designs.

As mentioned in Chapter 2, the coefficient space in which the optimization took place lent itself to the poles of the equalizer being on the real axis. Moreover, pole pairs that were close to the real axis would often make large jumps onto the axis. Since the optimization is now done on the polar plane (using the radius and angle space) this diversion is no longer observed, and the optimization path is more direct to the solution.

There are benefits from not having this real axis attraction. Since the poles will not generally end up on the real axis, there are no unsatisfactory results caused by one pole of a pair lying in the opposite part of the unit circle to where the main group of poles are. Instead the poles are either both used efficiently, lying on the ellipse within the passband, or less efficiently off this structure but still as complex conjugate pairs. Whether the poles are efficiently or inefficiently used was observed to depend on the tolerance of the algorithm. As a second benefit, there were fewer unstable designs. When the poles went onto the axis using the old method, one of the real poles would often leave the unit circle, rendering the equalizer unstable.

This type of movement was never exhibited with the new method, though the pair of poles is still capable of leaving the unit circle. Nonetheless, the equalizer design is less likely to be unstable because the pole's real axis attraction is eliminated. The absence of the real axis attraction aided the design of the equalizer in both of these important ways.

The new method was also more robust with more difficult designs, and could optimize higher-order equalizers. With unconstrained optimization comparing the Q results from the coefficient-based objective function to the pole-position-based objective function (figure 5.2) for a lowpass filter problem, the difference is obvious. The filter is a 7th-order lowpass filter with a band edge at $\omega = 0.5$. The pole-position-based method seems to find a better solution using similar tolerances. Until about 25 sections, the two methods gave similar results, but after that, the coefficient-based objective function started to yield increased Q. The pole-position-based objective function method resulted in lower Q's, although not monotonically.

5.3 Constrained Method

The constrained method can also be used with the new polar formulation. A Hessian matrix must be provided for the new error function, and a new set of constraints must be used, but the underlying methods are essentially the same.

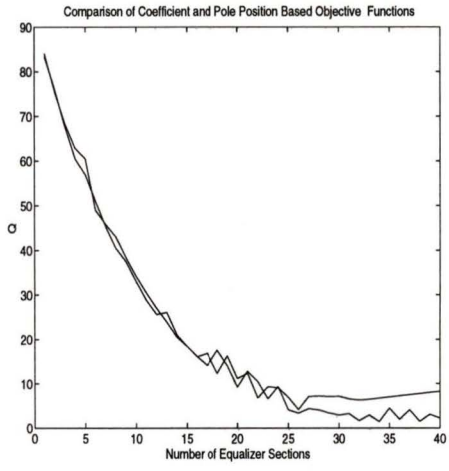

Figure 5.2: Comparison of the coefficient-based method to the pole-position-based method for a specific problem.

5.3.1 Hessian

The next step for the pole-position-based method is to develop the Hessian matrix for the error function so that it can be used in the constrained optimization technique described in Chapter 3. The Hessian matrix of the least-pth objective function is not changed and is still equation 3.12.

The Hessian matrix of the error function $\mathbf{H}_{e_i}$ is now given by

$$\mathbf{H}[e_i(\mathbf{x})] = \begin{bmatrix} \frac{\partial e_i(\mathbf{x})}{\partial r_{E1}\partial r_{E1}} & \frac{\partial e_i(\mathbf{x})}{\partial r_{E1}\partial \theta_{E1}} & \cdots & \frac{\partial e_i(\mathbf{x})}{\partial r_{E1}\partial \tau_0} \\ \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1}\partial r_{E1}} & \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1}\partial \theta_{E1}} & \cdots & \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1}\partial \tau_0} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_i(\mathbf{x})}{\partial \tau_0\partial r_{E1}} & \frac{\partial e_i(\mathbf{x})}{\partial \tau_0\partial \theta_{E1}} & \cdots & \frac{\partial e_i(\mathbf{x})}{\partial \tau_0\partial \tau_0} \end{bmatrix}. \tag{5.15}$$

Again, the Hessian can be simplified by noticing that

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{Ek}\partial r_{El}} = 0, \quad \text{for } k \neq l \tag{5.16}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial r_{Ek}\partial \theta_{El}} = 0, \quad \text{for } k \neq l \tag{5.17}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{Ek} \partial r_{El}} = 0, \quad \text{for } k \neq l \tag{5.18}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{Ek} \partial \theta_{El}} = 0, \quad \text{for } k \neq l \tag{5.19}$$

and, as for the previous objective function, any partial derivative including τ_0 is also 0. This causes the Hessian to be block diagonal matrix of the form

$$\mathbf{H}(e_i(\mathbf{x})) = \begin{bmatrix} \frac{\partial e_i(\mathbf{x})}{\partial r_{E1} \partial r_{E1}} & \frac{\partial e_i(\mathbf{x})}{\partial r_{E1} \partial \theta_{E1}} & 0 & 0 & \cdots & 0 \\ \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1} \partial r_{E1}} & \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E1} \partial \theta_{E1}} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \frac{\partial e_i(\mathbf{x})}{\partial r_{E2} \partial r_{E2}} & \frac{\partial e_i(\mathbf{x})}{\partial r_{E2} \partial \theta_{E2}} & \cdots & 0 \\ 0 & 0 & \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E2} \partial r_{E2}} & \frac{\partial e_i(\mathbf{x})}{\partial \theta_{E2} \partial \theta_{E2}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \tag{5.20}$$

where

$$\begin{aligned} \frac{\partial e_i(\mathbf{x})}{\partial r_{Ej} \partial r_{Ej}} = \Big\{ & 2 \Big[4 - 24 r_{Ej}^2 + 20 r_{Ej}^4 - 8 \cos^2 \theta_{Ej} - 40 r_{Ej}^4 \cos^2 \theta_{Ej} \\ & + ((28 r_{Ej} + 56 r_{Ej}^3 + 12 r_{Ej}^5) \cos \theta_{Ej} \\ & + (16 r_{Ej} + 32 r_{Ej}^3) \cos^3 \theta_{Ej}) \cos(\omega_i T) \\ & + (-8 - 40 r_{Ej}^4 - 96 r_{Ej}^2 \cos^2 \theta_{Ej}) \cos^2(\omega_i T) \Big] \\ & + (16 r_{Ej} \cos \theta_{Ej} + 32 r_{Ej}^3 \cos \theta_{Ej}) \cos^3(\omega_i T) \Big] C_j^2(\omega_i) \\ & - W_{j1} \cdot \frac{\partial C_j^2(\omega_i)}{\partial r_{Ej}} \Big\} / \left[C_j(\omega_i) \right]^4 \end{aligned} \tag{5.21}$$

$$\begin{aligned} \frac{\partial e_i(\mathbf{x})}{\partial r_{Ej} \partial \theta_{Ej}} = \Big\{ & 2 \Big[16 r_{Ej} \cos \theta_{Ej} \sin \theta_{Ej} + 16 r_{Ej}^5 \cos \theta_{Ej} \sin \theta_{Ej} \\ & + (-(2 + 14 r_{Ej}^2 + 14 r_{Ej}^4 + 2 r_{Ej}^6) \sin \theta_{Ej} \\ & - (24 r_{Ej}^2 + 24 r_{Ej}^4) \cos_{Ej}^2 \sin \theta_{Ej}) \cos \omega_i T \\ & + (64 r_{Ej}^3 \cos \theta_{Ej} \sin \theta_{Ej}) \cos^2 \omega_i T \end{aligned}$$

$$+ (-8r_{Ej}^2 \sin\theta_{Ej} + 8r_{Ej}^4 \sin\theta_{Ej}) \cos^3 \omega_i T \Big] C_j^2(\omega_i)$$

$$- W_{j1} \cdot \frac{\partial C_j^2(\omega_i)}{\partial \theta_{Ej}} \Big\} / [C_j(\omega_i)]^4 \tag{5.22}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{Ej} \partial r_{Ej}} = \Big\{ 2 \Big[16 r_{Ej} \cos\theta_{Ej} \sin\theta_{Ej} - 48 r_{Ej}^5 \cos\theta_{Ej} \sin\theta_{Ej}$$

$$+ (-2\sin\theta_{Ej} - 30 r_{Ej}^2 \sin\theta_{Ej} + 50 r_{Ej}^4 \sin\theta_{Ej} + 14 r_{Ej}^6 \sin\theta_{Ej}$$

$$-24 r_{Ej}^2 \sin\theta_{Ej} \cos^2\theta + 40 r_{Ej}^4 \cos^2\theta_{Ej} \sin\theta_{Ej}) \cos\omega_i T$$

$$+ (24 r_{Ej}^2 \sin\theta_{Ej} - 40 r_{Ej}^4 \sin\theta_{Ej}) \cos^3(\omega_i T) \Big] C_j^2(\omega_i)$$

$$- W_{j2} \cdot \frac{\partial C_j^2(\omega_i)}{\partial r_{Ej}} \Big\} / [C_j(\omega_i)]^4 \tag{5.23}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \theta_{Ej} \partial \theta_{Ej}} = \Big\{ 2 \Big[(8 r_{Ej}^5 - 8 r_{Ej}^6)(-\sin^2\theta_{Ej} + \cos^2\theta_{Ej})$$

$$+ (-2 r_{Ej} \cos\theta_{Ej} - 10 r_{Ej}^3 \cos\theta_{Ej} + 10 r_{Ej}^5 \cos\theta_{Ej} + 2 r_{Ej}^7 \cos\theta_{Ej}$$

$$+ (-8 r_{Ej}^3 + 8 r_{Ej}^5)(-2\cos\theta_{Ej} \sin^2\theta_{Ej} + \cos^3\theta_{Ej})) \cos(\omega_i T)$$

$$+ (8 r_{Ej}^3 \cos\theta_{Ej} - 8 r_{Ej}^5 \cos\theta_{Ej}) \cos^3 \omega_i T \Big] C_j^2(\omega_i)$$

$$- W_{j2} \frac{\partial C_j^2(\omega_i)}{\partial \theta_{Ej}} \Big\} / [C_j(\omega_i)]^4 \tag{5.24}$$

and

$$W_{j1} = 2(U_{j0} + U_{j1} \cos(\omega_i T)$$

$$+ U_{j2} \cos^2(\omega_i T) + U_{j3} \cos^3(\omega_i T)) \tag{5.25}$$

$$W_{j2} = 2(V_{j0} + V_{j1} \cos(\omega_i T)$$

$$+ V_{j2} \cos^2(\omega_i T) + V_{j3} \cos^3(\omega_i T)) \tag{5.26}$$

$$\frac{\partial C_j^2(\omega_i)}{\partial r_{Ej}} = 2C_j(-4 r_{Ej} + 4 r_{Ej}^3 + 8 r_{Ej} \cos^2\theta_{Ej}$$

$$-4\cos\theta_{Ej} \cos\omega T - 12 r_{Ej}^2 \cos\omega T + 8 r_{Ej} \cos^2\omega T) \tag{5.27}$$

$$\frac{\partial C_j^2(\omega_i)}{\partial \theta_{Ej}} = 2C_j(-8 r_{Ej}^2 \cos\theta_{Ej} \sin\theta_{Ej}$$

$$+4r_{Ej}\sin\theta_{Ej}\cos\omega T + 4r_{Ej}^3\sin\theta_{Ej}\cos\omega T). \tag{5.28}$$

Again, it can be seen that these equations require more calculations than their coefficient based counterparts.

5.3.2 Constraints

The stability constraints for the new method need only affect the radius parameters of the optimization. The new constraints are

$$\begin{aligned} r_{Ej} &< 1 \\ r_{Ej} &> 0 \end{aligned} \tag{5.29}$$

which means that the number of stability constraints has been reduced from three to two per section. The lower bound of 0, will assure a positive radius.

5.3.3 Other Modifications

Whereas the old objective function used a buffer for the coefficient constraints, the new method can directly relate the buffer to the radius. As Chapter 4 has shown, the group delay is dependant on the pole radius, and the group delays are additive for the filter and equalizer sections. Therefore, the maximum group delay that the equalizer should need to match is no higher than the highest group delay that the unequalized filter exhibits. Since the equalizer delay peak does not need to exceed that of the filter, the equalizer section radius, r_E, can be constrained to be less than the largest pole radius of the filter. This will keep the equalizer peak from needlessly exceeding the highest peak of the filter. In doing so, it will also prevent the pole from

getting too near the unit circle. This would help prevent computer overflow, as was occasionally witnessed when using the random starting points (mentioned later).

Another similar modification is to make the second radius constraint r_{Ej} greater than r_{min}, where r_{min} is some known lower bound of the radius. This can keep the poles from straying inward (toward the origin). An initial lower bound for the radius can be set and reset according to the smallest radius value of each subsequent optimization result. The value of r_{min} should increase with the number of sections, since the ellipse of an efficient solution tends to become more circular as the number of equalizer sections increases and, therefore, the smallest equalizer pole radius approaches the largest. By restricting the region in which solutions can be found, efficient solutions become more likely.

Finally, for a reason explained later, the step restriction is capped. A maximum value of $h = 2$ is placed on the algorithm.

5.4 Results and Comparisons

5.4.1 An Example

Using the fifth-order lowpass filter example used already for other techniques, the constrained optimization method with the pole-position-based objective function did not perform as well. The filter specifications were a passband edge of $\omega = 0.5$ rad/s, the passband ripple was 1 dB, and the minimum stopband attenuation was 50 dB.

Compared to the unconstrained method using the same objective function and filter specifications, the method took a longer time to complete, and was not as

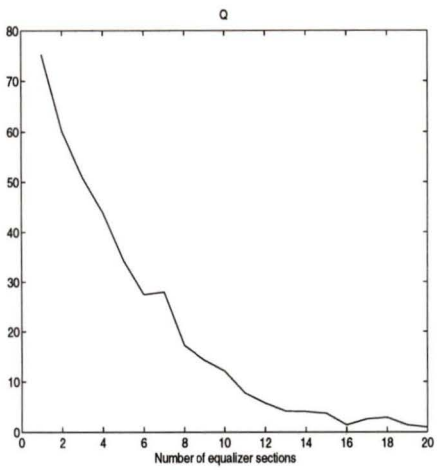

Figure 5.3: Convergence of Q for the pole position based objective function and the constrained optimization method.

consistent with the levels of Q as the unconstrained method was. The convergence can be seen in figure 5.3. One reason for this may be that the same constraints in terms of h were used for this method as were used for the coefficient-based objective function. They may not be relevant for the pole-position-based method.

5.4.2 Analysis

There are a few traits of this method that should be noted.

Because the constrained method allows the moderation of the movement of the values through h, the step restriction, the delay parameter did not play as dramatic a part in making the other parameters stray. In fact, the opposite effect happened. By limiting the size of h to less than 2, the equalizer poles do not move excessively, even when τ_0 is nowhere near the solution value. In the unconstrained method, if the value of τ_0 needed much movement, what usually happened was the poles would move

roughly to a good position, then the value of τ would start to change dramatically, either plummetting or skyrocketting, toward the solution value. In the meantime, the previously arranged poles sped off in random directions. This limiting of h would have less of a success in the coefficient based method because the main problem there is the real axis attraction. Because the pole-position-based method does not have this problem, the modification for the step restriction was very successful.

5.4.3 Comparison

Here, the constrained method is compared with the unconstrained method using the pole-position-based objective function. First, the number of computations per iteration is examined.

Computational Cost

There are several things to be said about the computational cost for the pole-position-based objective function when it is used with the constrained and unconstrained methods. Figure 5.4 shows the number of flops for one iteration of the methods using the pole-position-based objective function. Figure 5.4 b) shows the number of flops against the number of sections. Figure 5.4 a) is a detail showing the number of flops with less than ten sections. Both the curves representing the constrained and unconstrained methods have irregular shapes, and one can assume from this that neither method has a closed form for the number of flops for each iteration. The unconstrained optimization uses the Fletcher line search or a modified version found in [1], and the constrained method searches the constraints using the primal active

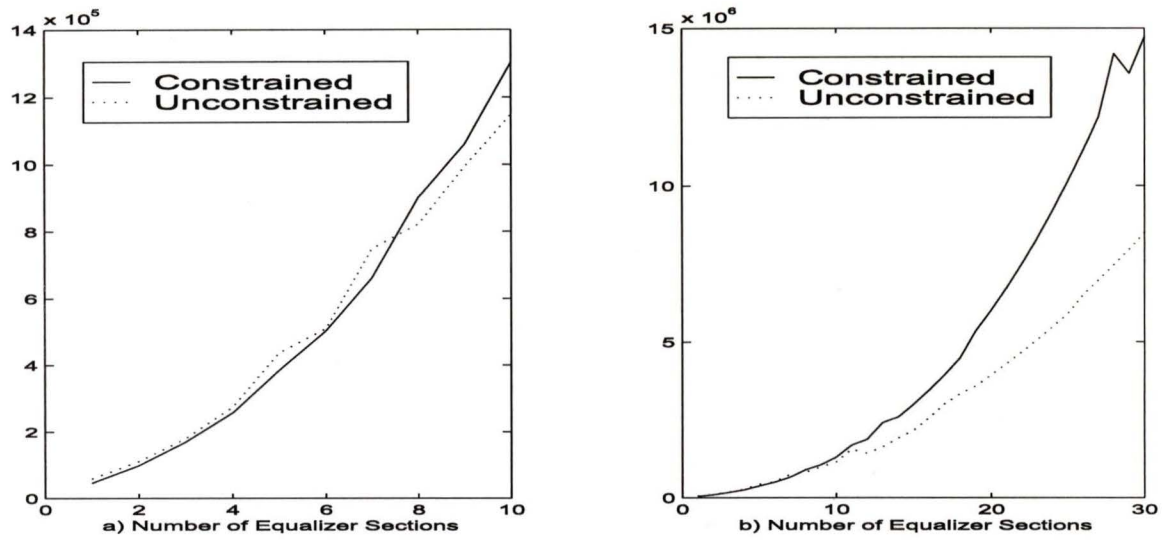

Figure 5.4: Comparison of flops for one iteration of unconstrained and constrained optimization methods.

set method. Because of these algorithms, the number of calculations per iteration will vary. Figure 5.4 a) shows the constrained optimization, in this case, uses less flops for equalizers with less than seven sections.

When compared with the coefficient based methods (figure 3.3), both constrained and unconstrained methods using the pole-position-based objective function are computationally more expensive. This seems to indicate that the increased number of computations to calculate the various expressions seems to outweigh the savings of fewer constraints.

Comparison of Results Using Random Starting Points

In order to show the effectiveness of the algorithms on random starting points, comparisons similar to those made in Chapter 3 are made. Between 20 and 24 optimizations were done for equalizers from one to ten sections in size. Table 5.1 shows

the success rate of the two algorithms using the constrained and the unconstrained methods. The minimum Q's are then compared, and finally, the percentage of results where the Q are within 10 of the minimum.

The rate of success was based on different things for each method as in the earlier comparison. For the unconstrained method, an unstable design was considerred unsuccessful. For the constrained method, the method would occasionally cease to continue (again, possibly due to a difficult inverse calculation), and of course, this was counted as an unsuccessful result. Again, the method never halted the algorithm using the controlled starting points. Using this gauge, it was found that the pole-position based objective function improved the success rate for both the unconstrained and constrained optimization methods.

Table 5.1 only indicates how successful the algorithms are at finding results, but it does not indicate how good the results are. Table 5.2 shows the minimum values of Q, which shows that all the minimum values are found using the constrained method. What the table does not indicate is that the constrained method did not give such good values of Q for all examples. One can get a better idea of the situation using the results that were within 10 of the minimum value, as one can see in table 5.3. Again, the constrained optimization technique was found to give better performace than the unconstrained technique. Compared to the methods using the coefficient-based objective function, the unconstrained method was found to more successful, but only in the sense that it could find stable solutions more successfully, not in the sense that the good solutions were any better. Otherwise, the numbers are roughly the same.

Table 5.1: Success of optimization for random filters

Order	Constrained (%)	Unconstrained (%)
1	95.2	55
2	87	35
3	85.7	55.6
4	95	15.8
5	100	35
6	100	40
7	100	40
8	100	50
9	95.2	50
10	86.4	31.8

Table 5.3 shows that the number of good solutions using the pole-position-based method have improved for both constrained and unconstrained methods relative to the coefficient-based method.

One thing that was observed with the random starting points was that with some equalizers, mainly the smaller ones, the solution would put the poles of the equalizer on an ellipse, but not over the entire passband. The ellipse seemed to avoid the area around the outermost pole. For this to occur, the starting value of τ_0 generally was quite far from the solution. When the value of h, the step restriction, became large, the poles would move in large steps, negating any benefit of a good starting point. The poles, having moved from the regular structure of the good starting point, move

Table 5.2: Minimum Q values for different filters

Order	Constrained	Unconstrained
1	75.2484	76.3234
2	60.0892	60.8234
3	50.8447	52.1553
4	42.4806	42.6502
5	34.3704	35.9539
6	27.4724	29.3343
7	21.8564	23.2628
8	17.2539	18.2579
9	13.4114	14.1329
10	10.306	10.7513

toward the area of the passband where the filter has its poles with the smallest radii, and move to form the regular structure over this smaller region. The group delay for this structure has the group delay of the equalizer matching that of the filter's peak, but then a gulf exists between the two, allowing a big drop in the group delay. These solutions have higher values for Q than the solutions whose equalizer poles are over the entire passband. When there are enough equalizer sections, this gulf is "filled in", and this structure ceases to occur. Moreover, when the algorithm is used, the starting values are not entirely random and the value of τ_0 tends to be close to the solution value, so these results occur less often.

By using the numbers in these tables, it can be said that the pole-position-based

Table 5.3: Percentage of results within 10 of minimum Q

Order	Constrained	Unconstrained
1	52.3	42.9
2	60.9	30.4
3	52.4	47.6
4	65	15
5	100	35
6	80	40
7	80	35
8	95	50
9	85.7	47.6
10	81.8	31.8

method improves the results when designing recursive delay equalizers using both the constrained and unconstrained optimization methods.

5.5 Recommendations

The method in this section placed two constraints in the problem. The first is the stability constraint, where the radius is kept less than one (or a very small amount less than that, actually). The second constraint is to keep the radius value positive. Chapter 4 showed that equalizer poles are best used within the passbands of the problem, so one could also put constraints on the pole angles. These constraints

would keep the poles within a very tight region during optimization. Moreover, one could keep the poles from 'crossing' one another by making an angle constraint dependant on the position of the pole's neighbouring equalizer poles. The code to accomplish this would not add much to the optimization complexity if the angle boundaries are already being enforced. It would, however, reduce the straying done by the poles during the course of the optimization. Constraining the pole angles may prove useful if there is difficulty in keeping the poles within the passbands during optimization.

5.6 Conclusions

Reformatting the objective function in terms of the pole positions of the equalizer eliminates one of the major problems of the coefficient-based objective function, the real axis attraction. The results tend to be an improvement in the success rate of the algorithms, both unconstrained and constrained.

For the unconstrained method, the gradient of the error function had to be defined. The constrained method required the Hessian matrix of the error function and the constraints to be defined.

It was found that with random starting points, constrained optimization is very effective at finding a solution to the recursive delay equalizer problem. But with the starting points chosen by the algorithm stated in Chapter 3, the unconstrained method was almost as good, and yielded results much more quickly. The constrained method mainly offers the benefit of providing constraints when designing the filter

to ensure that the poles are in the stable region, specifically if there is a problem finding a stable equalizer design.

Chapter 6

Conclusions and Recommendations for Future Work

6.1 Conclusions

An unconstrained optimization method for the design of recursive delay equalizers has been described and analyzed. A constrained optimization method was then applied to the problem. Following this, an analysis of the properties and characteristics of the group delay due to complex conjugate poles was made. This helped develop a new objective function based on the pole positions rather than the coefficients of the transfer function.

In describing and analyzing the current unconstrained method, several characteristics as well as certain problems were identified. First, good solutions were found to have well defined pole position patterns. Also, during the course of the optimization, there seemed to be a trend of the poles becoming real. It was found that this trend

was a result of the optimization being conducted in the coefficient space. Another two traits were dependant on the optimization method. Firstly, a good starting point is necessary. It was observed that the optimization was much more effective with a good starting point rather than a random point, or even one whose τ_0 was not close to that of the ultimate design. Secondly, and perhaps the main reason why this thesis was pursued, the optimization method often resulted in unstable equalizers.

The constrained optimization method developed employs the Newton method, the primal active set method, the Lagrange method, and the generalized elimination method. It uses the same objective function as the unconstrained optimization method, but the Hessian matrix had to be obtained. A step constraint and the stability constraint were created, and with some minor modifications, the algorithm was ready. With a constrained optimization method, it is possible to always guarantee a stable equalizer design. It has been shown that with the same objective function, this method is more likely to provide a good solution to the problem than the unconstrained optimization method, using both predetermined and random starting points. Also, it was shown that despite being more computationally expensive per iteration of optimization, using this method can yied a solution in a shorter time.

An analysis of the recursive delay equalizer problem was then carried out with respect to the effect of pole positions on the group delay. By studying individual biquadractic sections, it was found that the group delay peak was strongly related to the pole's radius and angle. The shape of the group delay was also strongly related to it. With these results, the effect of cascading several sections was studied. Specifically, the effects of poles at the same angle and at different angles were studied,

and it was found that poles at different angles contribute more effectively to adjusting the group delay. This lead to some reasoning behind the optimal pole structures. From this analysis, a new Q was proposed to aid the design of equalizers. Also, possible methods of finding either a better starting point or a better method of finding a solution to the equalizer problem have been postulated.

Using the insight provided by the analysis of Chapter 4, a new error function using a least-pth objective function was proposed and its gradient was derived. This new error function is based on the polar pole positions, and it eliminates the real axis attraction inherent in the method based on coefficients. It was found to be a significant improvement when used with the unconstrained method. The same objective function was applied to the constrained method, and the associated Hessian matrix was derived. The method proved computationally very intensive and should, perhaps, be better used for more difficult filters.

With new methods and better insight on the design of recursive delay equalizers, equalized recursive filters could prove a more economical alternative to nonrecursive filters in applications that require linear phase response in the passbands, particularly if the application also calls for a high filter selectivity.

6.2 Recommendations for Future Work

The research undertaken in this thesis may be continued in a number of directions. The first is to estimate a solution and to reserve as much as possible of the computation effort for the final design solution. Chapter 4 indicated patterns that can be

used, so essentially all that is needed is an idea of the size of the equalizer required to meet specifications. With that, a good starting point can be derived, and the optimization can begin there.

In the optimization methods described, a large part of the computational cost is caused by the need to find intermediate solutions leading up the the final solution (growing the equalizer). If there was a way to predict the size of the equalizer that can meet specifications, even approximately, much of this effort could have been saved. There are two possible ways that this might be done. The first is to obtain the solutions for the initial equalizers, i.e., equalizers with $M = 1,\ 2,$ and 3. By looking at their values of Q, one can determine where, by continuing the trend, the line goes below the required Q_{max} (the specification for the equalizer) and thereby determine the number of sections required for a potential solution. Because the value of Q does not tend to decrease according to a well defined law, it will be likely that even if one gets an efficient solution, further extrapolation of Q values will need to be made. If this method is used, one must guarantee that the solutions used in the trend are efficient or else the extrapolation may be completely inaccurate. Assuming that the solutions are good, though, one can skip optimizations for potentially quite a few equalizers.

Another potential method would be to start with the filter group delay, and to add equalizer sections, and therefore group delay peaks, until the filter specifications are met over the entire passband. This would resemble the method for finding a starting point or perhaps a solution that was described in Chapter 4. The first equalizer section is added so its peak occurs at the point where the group delay of

the filter falls below a minimum group delay value (determined by Q). In fact, the peaks would probably be best put a little after that point, since the group delay at the bells of each peak contributes to the overall group delay. Each subsequent section is added so that their group delay peaks are at the point where the last drops below the required group delay minimum (redetermined by Q), until the specifications are met. The difficulty with this, much like the difficulty in finding a starting point or solution with this method, is that the effect of each additional section on the existing group delay is not easily predictable without some optimization, and that the new section placement depends on the shape of the current group delay curve and the curve shape for the new section. For instance, if a new section must be added with a certain group delay amplitude in mind, the amplitude depends on the placement within the current group delay where the peak will be placed and that is dependant on the width of the bell formed by the section. The width, of course, is dependant on the radius and the problem continues. Some work to determine the appropriate equalizer size would save much computation since for each section or sections added, optimization is required. As mentioned, this method comes from observations from Chapter 4 and may be easier to see from graphs and intuitive arguments than to implement in an algorithm.

As mentioned in the analysis in Chapter 4, a closed form solution to the recursive delay equalizer problem might be possible. With the results of that analysis, one can imagine that a method perhaps similar to that used for elliptic filters can be devised to obtain a very good solution without any optimization.

An interesting topic more directly related to the algorithms discussed in this

thesis might be the varying settings of the tolerance values required for equalizers of different orders, and varying values of p for the least-pth algorithm. It has been found that increasingly small values for the tolerance ϵ are required as the order of the equalizer or the value of p grows. The exact increments by which the value of ϵ is modified has an effect on how many calculations are required. For instance, it has been observed that an optimization for a particular equalizer order will reach a solution, but will continue well after what seems visually reasonable, just to meet a prescribed tolerance. Very often, the optimization is stopped only due to an iteration limit installed in the algorithm. This is a waste of computation time, especially if the number of equalizer sections is nowhere near the number required for a satisfactory solution. That computational effort would be better used toward the actual solution. A critical aspect, then, to minimizing the amount of computation, is to determine a more efficient algorithm for determining the tolerances.

If the work in this thesis is continued, either a 'higher level' or 'lower level' approach can be taken. At a higher level, one can still use the optimization techniques and objective functions described here, but alter the method in which the optimizations are used, either by estimating solution sizes, or by individually adding equalizer sections. At a lower level, one can determine the tolerances that tweak each iteration of the optimization. Both would lead to interesting research that is beyond the scope of this thesis.

Appendix A

Sample Solutions

A.1 Solution for Coefficient Based Objective Function Using Constrained Optimization

The 20th-order equalizer solution for the filter described in Chapter 3, using the constrained optimization method with the coefficient based objective function, is given in Table A.1

Table A.1: Equalizer solution for the example in Chapter 3

i	c_{i1}	c_{i2}	i	c_{i1}	c_{i2}
1	0.5958	-1.5449	11	0.7773	-1.1816
2	0.6291	-1.5785	12	0.7769	-1.0633
3	0.6544	-1.6043	13	0.7950	-0.9682
4	0.7021	-1.6313	14	0.8264	-0.6651
5	0.7046	-1.5913	15	0.8213	-0.8338
6	0.7155	-1.5582	16	0.8263	-0.4421
7	0.7322	-1.5142	17	0.8547	-0.2616
8	0.7508	-1.4638	18	0.7417	-0.4844
9	0.7774	-1.3943	19	0.8788	-0.1046
10	0.7732	-1.2789	20	0.9467	0.4949

$\tau_0 = 72.1768$

A.2 Solution for Pole Position Based Objective Function Using Unconstrained Optimization

The 20th-order equalizer solution for the filter described in Chapter 5, using the unconstrained optimization method with the pole position based objective function, is given in Table A.2.

Table A.2: Equalizer solution for the example in Chapter 5

i	r_i	ω_i
1	0.8007	0.1118
2	0.8589	0.0615
3	0.8589	0.1822
4	0.8611	0.2840
5	0.8675	0.3696
6	0.8711	0.4922
7	0.7866	0.6033
8	0.8444	0.6104
9	0.6296	0.6289
10	0.8210	0.7369
11	0.7850	0.7247
12	0.8617	0.8018
13	0.8814	0.9082
14	0.8979	1.0039
15	0.9048	1.1020
16	0.9066	1.1942
17	0.9126	1.2886
18	0.9005	1.3852
19	0.9051	1.4352
20	0.9346	1.5157

$\tau_0 = 75.3461$

A.3 Solution for Pole Position Based Objective Function Using Constrained Optimization

The 20th-order equalizer solution for the filter described in Chapter 5, using the constrained optimization method with the pole position based objective function, is given in Table A.3.

Table A.3: Equalizer solution for the example in Chapter 5

i	r_i	ω_i
1	0.5514	-0.0032
2	0.8283	0.0608
3	0.6808	0.0699
4	0.8097	0.1636
5	0.8443	0.2184
6	0.8786	0.3287
7	0.8981	0.4420
8	0.9079	0.5542
9	0.9121	0.6596
10	0.9145	0.7616
11	0.9164	0.8608
12	0.9182	0.9581
13	0.9193	1.0541
14	0.9202	1.1497
15	0.9214	1.2441
16	0.9235	1.3391
17	0.9280	1.4313
18	0.9380	1.5151
19	0.3951	1.0848
20	0.4275	2.0351

$\tau_0 = 73.0102$

References

[1] A. Antoniou, *Digital Filters: Analysis, Design and Applications, 2nd ed.,* McGraw-Hill, New York, NY, 1993.

[2] A. Antoniou, 'Improved minimax optimisation algorithms and their application in the design of recursive digital filters,' IEE Proc., vol. 138, pt. G, pp. 724-730, Dec. 1991.

[3] A. Antoniou, *Optimization: Theory and Practice*, Chapter 5, University of Victoria, 1999.

[4] A. Antoniou and R. Howell, 'Design of phase equalizers for recursive filters,' in *Proc. 1993 IEEE Pacific Rim.* pp. 104-107.

[5] C. Charalambous and A. Antoniou, 'Equalization of recursive digital filters,' *Proc. Inst. Elec. Eng..* vol. 127. pt. G, no. 4. pp. 219-225, 1980.

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

[7] D. G. Luenberger, *Linear and Nonlinear Programming, 2nd ed.,* Addison-Wesley, Reading, Ma, 1984.

VITA

Surname: Ko ***Given Names:*** Nelson Richard
Place of Birth: Victoria, British Columbia, Canada ***Date of Birth:*** Jan. 21, 1969

Educational Institutions Attended

University of Victoria	1990 to 2000
University of Toronto, Trinity College	1986 to 1990

Degrees Awarded

B.Eng.	University of Victoria	1995
B.A.	University of Toronto, Trinity College	1990

Honors and Awards

NSERC Undergraduate Student Research Award	Sept. to Dec. 1994
NSERC Undergraduate Student Research Award	Jan. to Apr. 1994
British Columbia Graduating Scholarship	Sept. 1986 to Apr. 1987

Conference Publications

A. Antoniou and N. Ko, "A software package for the analysis and design of digital filters", 1995 IEEE Pacific Rim Conference on Communications, Computers, Visualization, and Signal Processing.

N. Ko, D. Shpak and A. Antoniou, "Design of Recursive Delay Equalizers Using Constrained Optimization", 1997 IEEE Pacific Rim Conference on Communications, Computers, Visualization, and Signal Processing.

UNIVERSITY OF VICTORIA PARTIAL COPYRIGHT LICENSE

Title of Thesis: DESIGN OF RECURSIVE DELAY EQUALIZERS BY CONSTRAINED OPTIMIZATION.

Author: [redacted]

Nelson Ko
December 22, 2000