Department of Electrical and Computer Engineering

Optimal Design of Farrow Variable Fractional Delay Filters

Chuan-Wei Kevin Chu

This thesis is presented for the Degree of

Doctor of Philosophy

of

Curtin University

June 2017

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: ……

Date: 3rd July 2017 ……

Acknowledgements

I would like to thank my supervisor, Dr. Yee-Hong Leung, for the relentless support, encouragement, and guidance he has provided throughout my extended postgrad journey. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I would also like to thank Prof. Sven Nordholm and Prof. Yue Rong for providing me with valuable ideas and directions.

I must express my gratitude to my family and friends for their patient and continuous support to complete this research. Thanks for helping me believe in myself and motivating me to achieve this milestone.

Finally, to my beloved Li Sha; you help me to regain hope after despair, resume life after obstructions, restart journeys after detours, revive strength after defeat and resurrect dreams after rejection. Thank you for sticking with me and believing in me for this long. We did it!

Abstract

Variable fractional delay filters allow the delay of a filter to be varied continuously over an interval without the need to redesign the filter. They can be found in applications such as voice recognition, augmented reality and next generation data network. A popular structure to implement variable fractional delay filters is the Farrow structure. This structure allows variable filters to be implemented efficiently.

In the past two decades, a plethora of approaches have been proposed to design optimum variable fractional delay filters. The two most popular approaches are the weighted least squares and the weighted minimax. This thesis presents further insights into the design of weighted least squares variable fractional delay filters as well as an alternative method to design weighted minimax variable fractional delay filters.

Recent works on the design of weighted least squares variable fractional delay filters have exploited an assumption to simplify the design method. Specifically, it is claimed the coefficients of the filters will exhibit a certain symmetrical and anti-symmetrical property. However, no rigorous proof of this claim has ever been published. This thesis provides such a proof. Moreover, in the construction of the proof, firstly, sufficient conditions for the symmetry and anti-symmetry properties to hold are derived; and secondly, a numerical difficulty associated with the solution of the weighted least squares problem, namely, the inversion of a poorly conditioned matrix is resolved.

Guidelines for the weighted least squares design of variable fractional delay filters whose magnitude response is flat from zero to a certain frequency less than half the sampling frequency are then studied. In particular, the relationship between the size of the Farrow structure and the filter performances is investigated. In this study, an unexpected result where the error in the phase response does not decrease monotonically as the filter length is increased was observed. An explanation of this counterintuitive phenomenon is presented in this thesis.

In the case of the weighted minimax design of variable fractional delay filters, there exist many optimisation methods. The current state-of-the-art is the sequential quadratic programming method which has been implemented in most mathematical

software packages. In this thesis, a different method to solve the minimax design problem is explored. The proposed method is based on extending Lawson's algorithm for minimax design in one dimension (frequency) to two dimensions (frequency and delay). The salient feature of the proposed algorithm is that it provides a bridge between weighted least squares and weighted minimax designs. It also includes a proactive convergence scheme that significantly increases the convergence rate of the algorithm.

Contents

List of Figures

List of Tables

List of Abbreviations

AF	Almost-Flat
AP	All Pass
BP	Band Pass
BS	Band Stop
DFT	Discrete Fourier Transform
FD	Fractional Delay
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
HP	High Pass
IIR	Infinite Impulse Response
IRLS	Iteratively Reweighted Least Squares
LMA	Levenberg-Marquardt Algorithm
LP	Low Pass
LS	Least Squares
PCC	Pearson Correlation Coefficient
SOCP	Second-Order Cone Programming
SOPOT	Sum of Power of Two
SQP	Sequential Quadratic Programming
SVD	Single Value Decomposition
VFD	Variable Fractional Delay
WLS	Weighted Least Squares
WMM	Weighted Minimax

List of Symbols

D	Bulk delay, i.e. $\frac{K-1}{2}$
δ	Non-integer fractional delays
$\delta_{\max}$	Maximum range of fractional delay, i.e. $\delta = [-\delta_{\max}, \delta_{\max}]$
ω	Normalised frequency
$\otimes$	Kronecker product
$A_d(\omega)$	Desire amplitude response
$W(\omega, \delta)$	Non-negative weighting function
K	Length of finite impulse response filter
M	Farrow filter approximation order
$h(k,m)$	Filter coefficient
$\mathbb{Z}$	Integer
$\mathbb{R}$	Real number
$\lfloor x \rfloor$	Flooring x-value
$\lceil x \rceil$	Ceiling x-value
J_{wls}	The actual maximum WLS cost
J_{WLS}	The design guide maximum WLS cost
$\overline{J}_{WLS}$	The design guide average WLS cost
$\varepsilon_n(\omega, \delta)$	Error function at n^{th} iteration
r	Pearson-product-moment correlation coefficient
l_p	p^{th} norm approximation
L_p	p^{th} norm
τ	Homotopy convergence parameter
λ	Newton-Raphson step parameter
α_i	Cut-off frequencies for almost-flat filter
β_i	Cut-off frequencies for band-pass filter
γ_i	Cut-off frequencies for low-pass filter
ϕ_k	Orthonormal basis function

Publications

The following publications arose as a result of this research work:

- C. W. K. Chu and Y. H. Leung, "Further results on the WLS design of variable fractional delay filters," *Signal Processing and Communications Systems (ICSPCS'12), 2012 6th International Conference on*, Gold Coast, QLD, 2012, pp. 1-7.
- C. W. K. Chu, "Improved WLS design of variable fractional delay filters", *Postgraduate Electrical Engineering and Computing Symposium (PEECS'12)*, Bentley, WA, 2012, pp. 1-4.
- C. W. K. Chu, "Optimum design of variable fractional delay filters with Laguerre formulation", Postgraduate Electrical & Computing Symposium (PEECS'09) 2009 10th, Joondalup WA, 2009, pp. 1-5.
- C. W. K. Chu and Y. H. Leung, "Optimum design of variable fractional delay filters", *Curtin Engineering Faculty Research Colloquium (CEFRC'08), 2008 4th*, Bentley WA, 2008, pp. 1-4.

1 Introduction

1.1 Fractional Delay Digital Filters

Fractional Delay (FD) digital filters are filters that enable an input signal to be delayed by a fraction of the sampling period. These filters are often found in diverse digital signal processing applications including speech coding [1, 2, 3, 4], speech synthesis [4, 5, 6], arbitrary sampling-rate conversion [7, 8, 9], efficient antenna arrays [10, 11], digital beamforming [12, 13], Doppler effects in virtual reality [14, 15, 16], phase matching [17, 18, 19], synchronisation [20, 21, 22, 23], and musical instrument modelling [24, 25].

A challenge for the FD digital filter designer is in choosing a suitable design technique from the vast array of available tools. In 1996, a comprehensive review of the tools to design fractional delay filters was published in [26]. The paper summarised the common techniques used to solve the related problem of bandlimited interpolation between samples. It is the most cited paper on the design and application of FD digital filters.

Briefly, a digital delay filter, in discrete time, is expressed as

$$y(n) = x(n - D) \tag{1.1}$$

where the input signal $x(n)$ is delayed by D samples, and D is a positive integer. In the case of a fractional delay digital filter, the output is given by

$$y(n) = x(n - \delta) \tag{1.2}$$

where δ is not necessarily an integer as illustrated in Figure 1. The objective of the FD digital filter is to interpolate the input signal value between two sampling points.

Two common techniques for implementing digital filters are the Finite Impulse Response (FIR) filter and the Infinite Impulse Response (IIR) filter [27]. FIR filters are always stable and can exhibit linear phase. In contrast, IIR filters tend to have lower filter orders, however care must be exercised in the design since they can be unstable. Moreover, they can be numerically unstable [27]. This thesis will focus on FIR FD

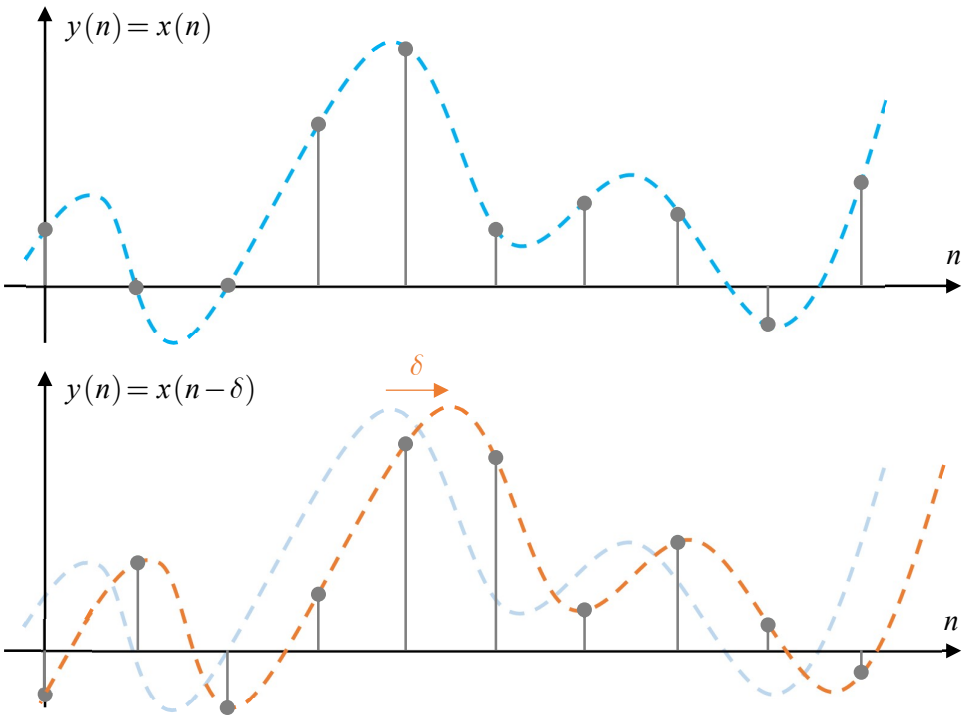


Figure 1 – A input signal that has been fractional delayed by δ

digital filter; and for a discussion on linear phase IIR approximations, the interested reader is referred to [28].

With reference to (1.2), it follows the frequency response of an ideal FD digital filter can be expressed as,

$$H_d(\omega,\delta) = A_d(\omega)e^{-j\omega\delta} \tag{1.3}$$

where $A_d(\omega) = 1$ is the ideal amplitude response, δ is the desired fractional delay, and ω is frequency in radians per sample. The key features observed in (1.3) are linear phase and a flat amplitude response. When transformed to the time domain the impulse response is given by a time-shifted sinc function. In other words, the ideal impulse response is non-causal and has infinite duration. The challenge then is to design an FIR filter to approximate the ideal impulse response as closely as possible in some sense.

1.2 Variable Fractional Delay FIR Filters

One specific branch of FD digital filters called Variable Fractional Delay (VFD) filters is the primary focus of this thesis. This filter allows a signal to be delayed by

some fraction of the sampling period where the amount of delay is adjustable in real-time.

One way to achieve the variable delay control is to adjust the impulse response of the VFD filter as the delay is being adjusted. This can be accomplished by storing the impulse responses of a number of fractional delay filters in a table where the fractional delay filters are selected from a limited set covering the range of delays to be implemented. An interpolation scheme is then applied to derive the impulse response of the fractional delay filter of interest [29, 30]. Alternatively, the impulse response of the fractional delay filter of interest can be computed on the fly as the delay is being varied [29]. Both methods have their drawbacks. The first method may require a large look up table which can result in memory storage issues. The second method requires on demand filter design which can be computationally intensive.

Another method is the Oetken method [31] which is essentially the frequency sampling method [32]. The drawback is that it requires the evaluation in real-time of K sine and K cosine functions, where K is the length of the VFD filter, as the delay is varied.

Based on the recent publications on VFD filters [33, 34, 35], it seems the most promising technique to implement VFD filters is the Farrow structure [20] shown in Figure 2. It consists of a bank of FIR filters whose coefficients are fixed, and a

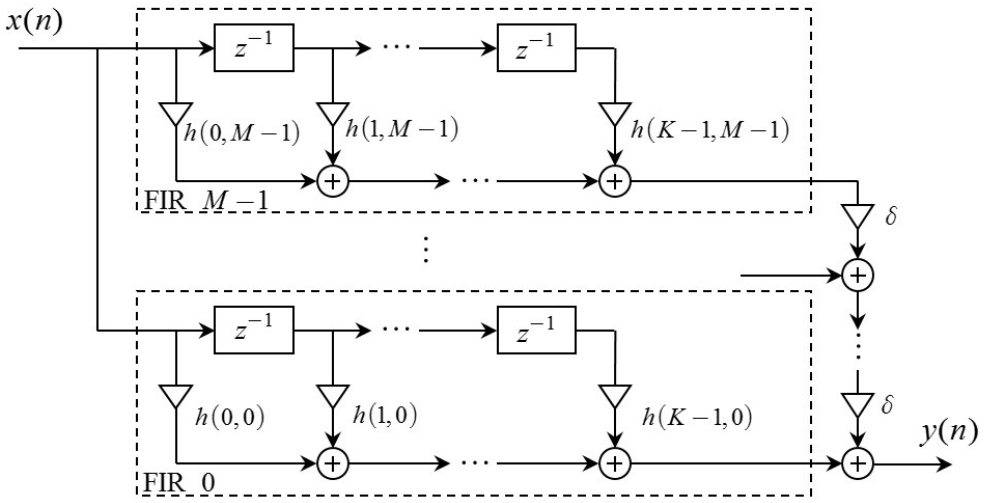


Figure 2 – Classic Farrow structure

multiplier-adder chain through which the delay can be controlled conveniently, on-line and in real-time.

The idea behind the Farrow structure can be explained as follow. Consider a length K FIR filter where the coefficients depend on the parameter θ. Denote the coefficients by $b(k,\theta)$ where $k \in [0, K-1]$. The frequency response of the filter is then given by

$$H(\omega,\theta) = \sum_{k=0}^{K-1} b(k,\theta) e^{-jk\omega} . \tag{1.4}$$

Suppose each filter coefficient can be approximated by an $(M-1)^{\text{th}}$ order polynomial as follows and as illustrated in Figure 3,

$$b(k,\theta) = \sum_{m=0}^{M-1} h(k,m)\theta^m , \tag{1.5}$$

where $h(k,m)$ are real coefficients. The overall frequency response is then given by

$$H(\omega,\theta) \cong \sum_{k=0}^{K-1} \left[\sum_{m=0}^{M-1} h(k,m)\theta^m \right] e^{-jk\omega} . \tag{1.6}$$

Rewriting (1.6) as follows,

$$H(\omega,\theta) \cong \sum_{m=0}^{M-1} \left[\sum_{k=0}^{K-1} h(k,m) e^{-jk\omega} \right] \theta^m \tag{1.7}$$

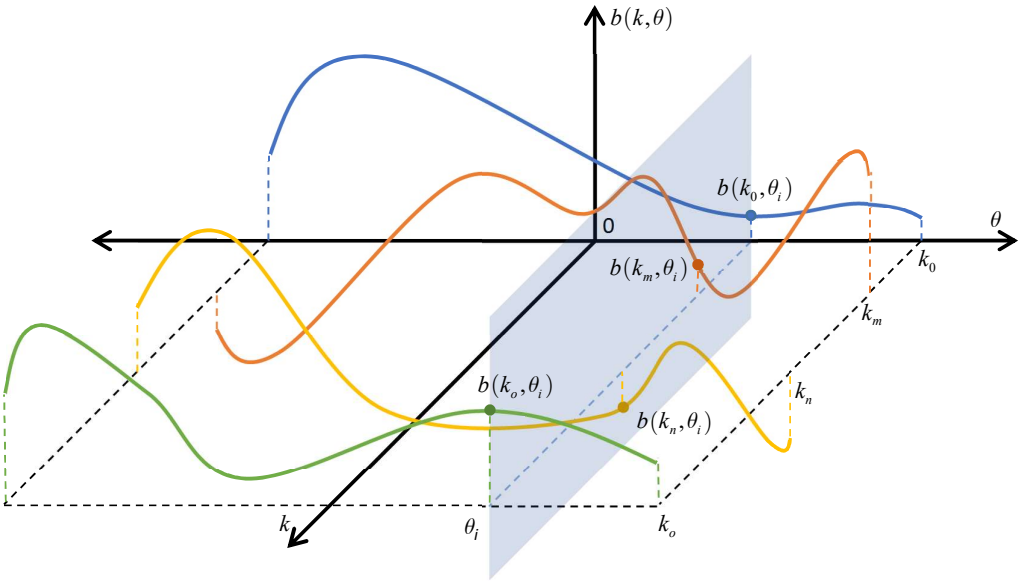


Figure 3 – The active delay approximation of filter coefficients

yields the Farrow structure shown in Figure 2. Note the Farrow structure is quite general and can be used to implement any filters whose frequency response depend on a parameter. In the case of a VFD filter, θ corresponds to the delay parameter δ, and in the context of the above discussion, $\delta^0 = 1$ for all δ including $\delta = 0$.

The Farrow structure can be efficiently implemented by streamlining the input signal into a cascaded ladder form as illustrated in Figure 4. Also, in Chapter 2 it will be proved that the coefficients of the WLS Farrow VFD filter will exhibit symmetry and anti-symmetry characteristics which can be further efficiently implemented using

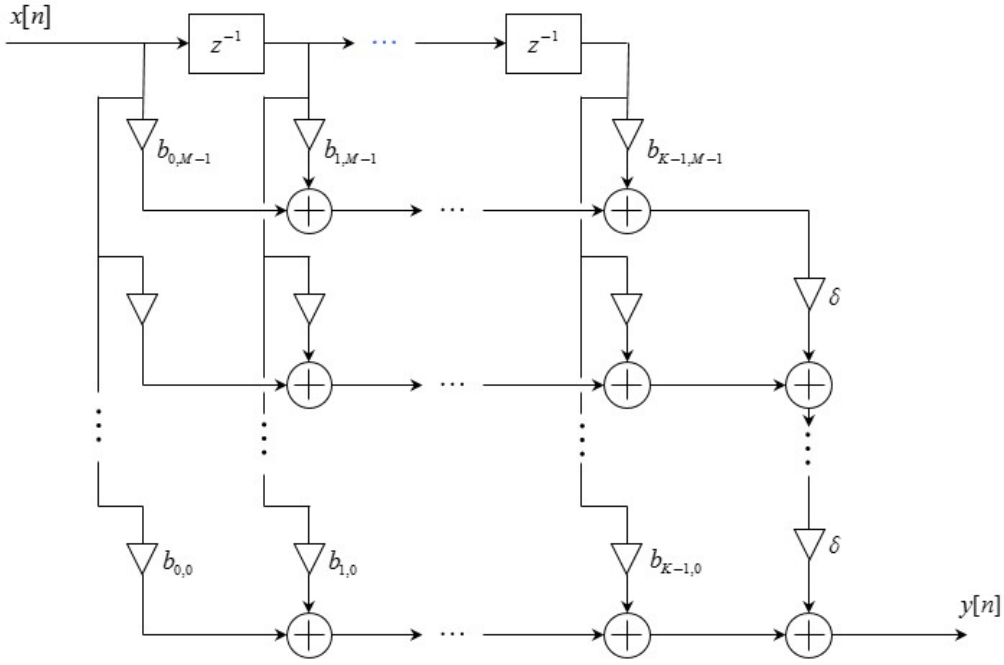


Figure 4 - Efficient implementation of Farrow structure

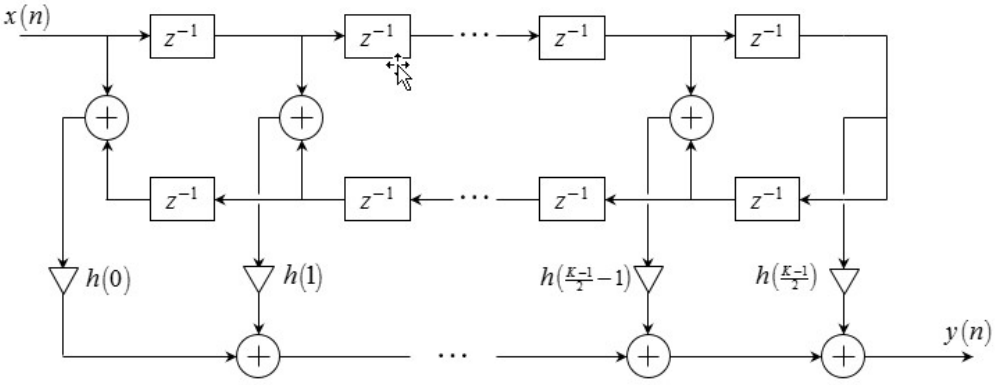


Figure 5 - Efficient Implementation of Farrow structure with coefficient symmetry

a loop-back structure. In the case of an even K length FIR filter, the filter coefficients can be halved as show in Figure 5. Similar figures can be drawn for other *K*s, odd or even.

1.3 Design of Variable Fractional Delay Filter

As discussed, the frequency response of a Farrow VFD filter can be written as

$$H(\omega,\delta) \cong \sum_{m=0}^{M-1}\left[\sum_{k=0}^{K-1} h(k,m)e^{-jk\omega}\right]\delta^m \quad . \tag{1.8}$$

The objective of the design process is to find $h(k,m)$ such that $H(\omega,\delta)$ will approximate $H_d(\omega,\delta)$ (1.3), for a range of δ, as closely as possible.

However, (1.8) has a fundamental limitation with the phase response at $\omega=\pi$. Specifically, $H(\omega,\delta)$ must be real at $\omega=\pm\pi$. Hence it is impossible to achieve the correct phase delay for any FD at $\omega=\pm\pi$. In this thesis, we propose the Almost-Flat (AF) filter, defined by $A_d(\omega)=1$ for $0\le\omega\le\alpha\pi$, and $\alpha<1$. This filter allows the design process to ignore a small region near π so that the actual frequency response can better approximate the desired frequency response over the rest of the frequency range.

At this point, it should be noted that, although the discussion in the remainder of this thesis is focussed mainly on the AF filter, the concepts and results can be easily transferred to other filter types such as Low Pass (LP), Band Pass (BP), and Band Stop (BS).

1.3.1 Weighted Least Squares

Weighted Least Square (WLS) is a popular method for solving the Farrow VFD filter design problem, as the solution has a closed-form expression [36]. The WLS design problem for the Farrow filter is defined by

$$\min_{\mathbf{h}} \int_{-\delta_{\max}}^{\delta_{\max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|H(\omega,\delta)-H_d(\omega,\delta)\right|^2 d\omega\, d\delta \tag{1.9}$$

where $W(\omega,\delta)$ is a non-negative weighting function, and $\delta_{\max}$ specifies the range of fractional delay to be implemented. There exist many methods to design WLS Farrow VFD filter, some notable ones are [37, 38, 39, 40, 41, 42, 43, 44].

Interestingly, a common assumption in the aforesaid works is that the filter coefficients exhibit a certain symmetry and anti-symmetry characteristic if the ideal frequency response of the desired filter is symmetrical in both frequency and delay. This symmetrical property was first claimed in [45] with many subsequent papers on WLS Farrow VFD filter exploiting the property and providing their take on the optimisation process [46, 47, 48, 49]. The problem with the symmetry assumption is that there are no rigorous proofs that it should hold. In this thesis, we provide the proof. In particular, we show that if the ideal frequency response and weightings are symmetrical, about frequency ω and delay δ, then the symmetry property will follow from the closed-form solution of the WLS Farrow VFD filter design problem.

1.3.2 Weighted Minimax

Another popular design method is the Weighted Minimax (WMM) [50, 51, 52, 53]. The WMM Farrow VFD filter design problem can be formulated as follows,

$$\min_{\mathbf{h}} \max_{\substack{-\pi\le\omega\le\pi \\ -0.5\le\delta\le 0.5}} W(\omega,\delta)\left|H(\omega,\delta)-H_d(\omega,\delta)\right|. \tag{1.10}$$

In the case of *normal* FIR filters, as discussed in many textbooks, the weighted least squares design is cost-effective and easy to implement, thanks to the availability of a closed-form solution [36]. However, the trade-off is the large peak errors at the band edges. WMM design [54], also as known as weighted Chebyshev approximation, provides good control over the frequency range where we wish to optimise, and results in smaller distributed error across the passband and stopband. The difficulty though is that there is no closed-form solution leading to various researchers employing complex game theory sequences to solve the design problem. Currently the most popular method is sequential quadratic programming (SQP) which has been implemented in most mathematical software packages. Other authors had proposed to solve the WMM problem via various other approaches such as the iterative approach [50], decoupling minimax [55], second-order cone programming (SOCP) [53], genetic algorithm [56] or sum-of-power-of-two (SOPOT) [57]. The overarching questions for these approaches are (i) whether they can achieve the actual WMM solution and if not, then (ii) how close is the approximated solution to the actual solution.

A heuristic discussion on the concept of using weighted L_p norm to approximate Chebyshev (L_∞) norm was presented in [58, 59]. Soon after, Lawson [60] published a study in his thesis developing the idea of best L_∞ approximation as the limit of a special sequence of WLS (L_2) problems. Lawson introduced the Iteratively Re-weighted Least Squares (IRLS) algorithm which defines a family of iterative methods that solve an otherwise complicated numerical optimisation problem by breaking it into a series of WLS problems. This establishes a relationship, or a bridge, between WLS FIR filter and WMM FIR filter. His idea was further expanded by Rice and Usow [61] to include the general L_p problem. Other authors [62, 63, 64, 65] also took the concept further by improving the convergence rate, implementing filter coefficient updates, and even introducing homotopy [66].

Today, there are many papers published utilising the IRLS technique [60]- [65] for finding the optimum coefficient of one-dimensional Finite Impulse Response (FIR) filter. In one dimension, IRLS converges exactly to minimax [60, 61] and it is an alternative to the Remez exchange algorithm. In relation to the design of two-dimension filters, the difficulty with the Remez exchange algorithm is that the alternation theorem solution in two dimensions are not unique and does not converge properly [67, 68, 69].

Algaz et al [70] exploited the WLS to Chebyshev relationship to construct a two-dimensional FIR filter through the use of Lawson's algorithm. In [71], Barreto and Burrus took the idea further and tackled a two-dimension FIR filter design problem by minimizing the magnitude of the error function using the L_p norm method. Chi and Chiou [72] introduced a modified IRLS method that can approximate any arbitrary two-dimensional complex frequency response. However, there are complicated matrix reshaping involved. To overcome this issue, Hsieh et al [73] proposed a method that allows the optimisation to retain the two-dimension matrix form which reduces the complexity. In this thesis, motivated by [74], we extend and improve the adaptive IRLS technique into two dimensions, in frequency and delay, to solve the WMM Farrow VFD filter design problem. We also introduce a novel way of calculating, in each iteration, the convergence parameter that is proactively evaluated for the homotopy direction and step size. This method overcomes the convergence issue

pointed out in [65] and eliminates the resource intensive tree-search of [74], thus generally improving the optimisation time and reduce the number of iterations. Furthermore, we will be incorporating the coefficient symmetry property that reduces the design complexity thus further improving the efficiency. Even though we have not proved the proposed algorithm will yield the optimum Chebyshev approximation in two dimensions, many design examples in [70, 71, 72, 73] show that it can lead to almost optimum approximation. We believe that by extending the algorithm to the two-dimensional Farrow VFD filter design problem, we can expect good results because, (i) for one dimensional FD filters the Lawson's algorithm can lead to the exact Chebyshev approximation, and (ii) the basis of one dimension to two dimensions extension are not influenced by the alternation theory as seen in Remez.

1.4 Design Guides for VFD Filter

Lastly, in this past decade there has been progressive development in the optimum design of VFD filters. The incremental improvements are almost reaching a plateau where the design cost improvement is now marginal from one method to the next. From a filter designers' point of view, having a simple tool that allows them to dimension their design with great accuracy is essential and desirable. In this thesis, we present three AF design-guides that allow the user to evaluate the three most common cost criteria, i.e. WLS cost, maximum magnitude response error, and maximum phase response error as functions of the filter dimensions M and K. There will be working examples on how these equations can be utilised to produce the design parameters for the desired filter. Most importantly, this numerical study sets a framework for other potential design guides while highlighting some noteworthy observations.

1.5 Aims and Objectives

This thesis focuses on the weighted least square and weighted minimax design of optimum Farrow VFD filter. The aims and objectives of this thesis are as follows.

a) Weighted Least Squares

The aim here is to revisit the WLS problem formulation. In particular, to address the following two unresolved issues.

1) It is commonly assumed that the optimum filter coefficients are symmetrical. However, it is the belief of the author that the proof published is not sufficiently rigorous. Specifically, the proof states that the optimum coefficients are symmetrical because the ideal response is symmetrical.
2) In the solution of the WLS problem a poorly condition matrix is required to be inverted. The standard technique to circumvent this issue is to use spectral decomposition techniques such as QR, Cholesky or SVD. It is the belief of the author, that by exploiting the structures that exists within the poorly conditioned matrix, a better method can be derived.
3) Another issue related to the design of WLS VFD filter is the lack of a set of guidelines to assist designers to dimension the size of their design. That is, the order of approximation, $M-1$, and the filter length, K. An extensive numerical study will be conducted.

b) Weighted Minimax

The aim here is to solve the WMM design problem, and in particular, to explore methods that are numerically more efficient than the state-of-the-art SQP method. A sub-aim of this investigation is to determine whether the coefficient symmetry property holds also for the WMM solution.

1.6 Contributions

The major contributions in this thesis are as follows.

- Proved that the coefficients of a WLS Farrow VFD filter are indeed symmetrical and anti-symmetrical.
- Presented a technique to overcome a known poorly conditioned numerical problem with inverting a matrix. This technique is based on a bootstrapping method to yield an analytical expression for the inversion of the said poorly conditioned matrix.
- Created three empirical design guides that enable the filter designer to easily evaluate the three cost criteria; that is, WLS cost, maximum magnitude response error cost, and maximum phase response error cost as a function of filter length K and polynomial order M.

- Explained an interesting observation on the maximum phase response error cost in the WLS design. It was noted that the maximum phase response error exhibits a saw-like pattern as the length of the filter increases.
- Presented an alternative method of solving the WMM Farrow VFD filter design problem. This method uses a modified and extended form of Lawson's algorithm. Our numerical study shows the proposed method converges faster than the popular SQP method.

1.7 Thesis Scope and Overview

Chapter 2: This chapter presents a brief review on the background of WLS Farrow VFD filter design. It also presents conclusive proof of symmetry characteristic in the filter design.

Chapter 3: This chapter presents some design examples showcasing the design method proposed in Chapter 2.

Chapter 4: This chapter introduce the WLS Farrow VFD filter design guides. These guides allow the user to calculate suitable starting parameters for their filter design.

Chapter 5: This chapter looks into an iterative method to solve the two-dimensional WMM Farrow VFD filter design problem along with improvements on the convergence rate.

Chapter 6: This chapter details the conclusion of this thesis and presents a number of notebook ideas for future work.

2 Weighted Least Squares Design

2.1 Introduction

Weighted least squares is a popular design method for Farrow VFD filter [37, 38, 39, 40, 41, 42, 43, 44]. The design method is easy to implement in software since it has a closed-form expression.

A feature of most of the papers on the WLS design of Farrow VFD filters is that they invoke some form of coefficient symmetry to improve computational efficiency and accuracy. Coefficient symmetry allows us to fold the filter coefficients by almost half. However, there is no conclusive proof that the coefficients should exhibit symmetry at all. Rather it was stated simply that symmetry results from the desired ideal frequency response being symmetrical in both frequency and delay [45].

In this chapter, a proof that the WLS coefficients will exhibit symmetry is presented. Additionally, the condition for the coefficients to exhibit symmetry are deduced, and an efficient way to invert a poorly conditioned matrix which exists in the solution of the WLS Farrow VFD filter design problem is derived.

2.2 The WLS Problem Formulations

In Chapter 1, it is shown that the frequency response of a Farrow VFD filter can be expressed as

$$H(\omega,\delta)=\sum_{m=0}^{M-1}\left[\sum_{k=0}^{K-1}h(k,m)e^{-jk\omega}\right]\delta^m . \tag{2.1}$$

The above expression can be put into vector-matrix notation as follows [75]

$$H(\omega,\delta)=\mathbf{h}^T(\boldsymbol{\delta}\otimes\boldsymbol{\omega}), \tag{2.2}$$

where $\otimes$ denotes Kronecker product,

$$\mathbf{h}^T=\begin{bmatrix}\mathbf{h}_0^T & \mathbf{h}_1^T & \cdots & \mathbf{h}_{M-1}^T\end{bmatrix}, \tag{2.3}$$

$$\boldsymbol{\delta}^T=\begin{bmatrix}\delta^0 & \delta^1 & \cdots & \delta^{M-1}\end{bmatrix}, \tag{2.4}$$

$$\boldsymbol{\omega}^T=\begin{bmatrix}1 & e^{-j\omega} & \cdots & e^{-j(K-1)\omega}\end{bmatrix}, \tag{2.5}$$

and for $m\in\{0,1,\ldots,M-1\}$,

$$\mathbf{h}_m^T = [h(0,m) \quad h(1,m) \quad \cdots \quad h(K-1,m)]. \tag{2.6}$$

Suppose the ideal frequency response of the VFD filter is given by

$$H_d(\omega,\delta) = A_d(\omega)\cdot e^{-j\omega\left(\frac{K-1}{2}+\delta\right)}, \tag{2.7}$$

where $A_d(\omega)$ is the desired amplitude response and $\frac{K-1}{2}$ is the bulk delay. The weighted least squares design problem is given by,

$$\min_{\mathbf{h}} \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|H(\omega,\delta) - H_d(\omega,\delta)\right|^2 d\omega\, d\delta, \tag{2.8}$$

where $W(\omega,\delta)$ is a non-negative weighting function, and δ_{max} specifies the range of fractional delay to be implemented.

2.2.1 The Unconstrained Farrow Structure

By adapting the least squares derivations in [75], the cost WLS function of (2.8) can be expressed as follows

$$J(\mathbf{h}) = \mathbf{h}^T\mathbf{Q}\mathbf{h} - 2\mathbf{h}^T\mathbf{p} + r, \tag{2.9}$$

where
$$\mathbf{Q} = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left[(\boldsymbol{\delta}\otimes\boldsymbol{\omega})(\boldsymbol{\delta}\otimes\boldsymbol{\omega})^T\right] d\omega\, d\delta, \tag{2.10}$$

$$\mathbf{p} = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\boldsymbol{\delta}\otimes\mathbb{R}\left[\boldsymbol{\omega}\cdot H_d^*(\omega,\delta)\right] d\omega\, d\delta, \tag{2.11}$$

and
$$r = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|H_d(\omega,\delta)\right|^2 d\omega\, d\delta. \tag{2.12}$$

Note that the matrix $\mathbf{Q}$ can be split into two components,

$$\mathbf{Q} = \mathbf{Q}_\delta \otimes \mathbf{Q}_\omega \ , \tag{2.13}$$

where
$$\mathbf{Q}_\delta = \int_{-\delta_{max}}^{\delta_{max}} W(\omega,\delta)\boldsymbol{\delta}\boldsymbol{\delta}^T d\delta \ , \tag{2.14}$$

and
$$\mathbf{Q}_\omega = \int_{-\pi}^{\pi} W(\omega,\delta)\boldsymbol{\omega}\boldsymbol{\omega}^H d\omega \ . \tag{2.15}$$

As can be readily shown from (2.8), the solution of the WLS design problem is given by

$$\mathbf{h}_{opt} = \mathbf{Q}^{-1}\mathbf{p}. \tag{2.16}$$

2.2.2 The Constrained Farrow Structure

In many related papers on the Farrow VFD structure, there is a prevailing constraint that is imposed in the design of All-Pass (AP) Farrow VFD filter. That is, filters whose desired amplitude response $A_d(\omega)=1$ for $\forall\omega$ in (2.7). It was first stated in [20], where the author, Farrow, considered the following design specifications $A_d(\omega)=1$, $W(\omega,\delta)=1$ and K is even. Farrow imposed a pair of constraints to ensure the filter exhibits a flat amplitude response when $\delta=\pm 0.5$. i.e,

$$\sum_{m=0}^{M-1} h(k,m)0.5^m = \begin{cases} 0, & k \neq K/2 \\ 1, & k = K/2 \end{cases}, \tag{2.17}$$

and

$$\sum_{m=0}^{M-1} h(k,m)(-0.5)^m = \begin{cases} 0, & k \neq K/2-1 \\ 1, & k = K/2-1 \end{cases}. \tag{2.18}$$

Later the constraints were extended to odd-length filters. In [76], the following constraint was imposed to guarantee the filter exhibits a flat amplitude response when $\delta=0$. i.e,

$$h(k,0)=\delta\left(k-\tfrac{K-1}{2}\right). \tag{2.19}$$

The *flat amplitude response constraint* of (2.17) and (2.18), or (2.19), can be expressed in vector-matrix form as follows

$$\mathbf{Ah}=\mathbf{b}, \tag{2.20}$$

where, for K even

$$\mathbf{A}=\begin{bmatrix} 0.5^0 & 0.5^1 & \cdots & 0.5^{M-1} \\ (-0.5)^0 & (-0.5)^1 & \cdots & (-0.5)^{M-1} \end{bmatrix} \otimes \mathbf{I}_K, \tag{2.21}$$

$$\mathbf{b}=\left[\boldsymbol{\delta}^T\left\langle K,\tfrac{K}{2}+1\right\rangle \quad \boldsymbol{\delta}^T\left\langle K,\tfrac{K}{2}\right\rangle\right]^T, \tag{2.22}$$

and for K odd

$$\mathbf{A}=\left[\mathbf{I}_K \quad \mathbf{0}_{K,(K-1)\times M}\right], \tag{2.23}$$

$$\mathbf{b}=\boldsymbol{\delta}\left\langle K,\tfrac{K}{2}\right\rangle, \tag{2.24}$$

and where $\mathbf{I}_K$ is the $K\times K$ identity matrix, and the $K\times 1$ unit vector $\boldsymbol{\delta}\langle K,l\rangle$ is defined by

$$[\boldsymbol{\delta}\langle K,l\rangle]_k = \begin{cases} 1, & k=l \\ 0, & \text{otherwise} \end{cases} . \tag{2.25}$$

The WLS filter with linear constraints is given by [77]

$$\mathbf{h}_{opt} = \mathbf{Q}^{-1}\mathbf{A}^T\left(\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T\right)^{-1}\left[\mathbf{b}-\mathbf{A}\mathbf{Q}^{-1}\mathbf{p}\right]+\mathbf{Q}^{-1}\mathbf{p} . \tag{2.26}$$

Note, (2.26) is a general result and applies to any constraints that can be expressed in the form of (2.20). However, to the author's best knowledge, in the context of Farrow VFD filter design, no other constraints other than the flat amplitude response constraints of (2.17) and (2.18), or (2.19), have been proposed in the literature. In any event, constrained WLS design will not be considered in this thesis. Since, according to the author's experience, the frequency response of the designed filter can be controlled more effectively through the desired amplitude response $A_d(\omega,\delta)$ and the weighting function $W(\omega,\delta)$.

2.3 Coefficient Symmetry

In [78], the authors proved the solution of the WLS design problem, $\mathbf{h}_{opt}$, exhibit symmetry providing the weighting function is separable and symmetrical, i.e.,

$$W(\omega,\delta) = W_1(\omega)\cdot W_2(\delta) . \tag{2.27}$$

$$W_1(\omega) = W_1(-\omega) , \tag{2.28}$$

$$W_2(\delta) = W_2(-\delta) , \tag{2.29}$$

and the desired amplitude response is symmetrical, i.e.

$$A_d(\omega) = A_d(-\omega) . \tag{2.30}$$

Equations (2.27) to (2.29) impose a restriction on the weighting function. See Appendix A2 for an example of the above special case.

In this thesis, we consider a set of constraints that are less restrictive than (2.27) to (2.29). Specifically, we require the functions to be quadrantally symmetrical, i.e.

$$W(\omega,\delta) = W(\omega,-\delta) = W(-\omega,\delta) = W(-\omega,-\delta). \tag{2.31}$$

2.3.1 Preliminary Mathematical Results

We first state the following five preliminary mathematical results.

I. It follows from [79] that

$$\mathbf{Q}^{-1} = (\mathbf{Q}_\delta \otimes \mathbf{Q}_\omega)^{-1} = \mathbf{Q}_\delta^{-1} \otimes \mathbf{Q}_\omega^{-1}. \tag{2.32}$$

It was observed that large $\mathbf{Q}_\delta$ matrices are poorly conditioned.

II. Following on from the above result, the well-known block matrix inversion method [80] can be employed to invert $\mathbf{Q}_\delta$. Suppose the matrix $\mathbf{M}$ is partitioned as follows

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}. \tag{2.33}$$

$$\text{Then} \quad \mathbf{M}^{-1} = \begin{bmatrix} \mathbf{M}_{11}^{-1} + \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{S}^{-1}\mathbf{M}_{21}\mathbf{M}_{11}^{-1} & -\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{M}_{21}\mathbf{M}_{11}^{-1} & \mathbf{S}^{-1} \end{bmatrix}, \tag{2.34}$$

where $\mathbf{S}$ is the *Schur complement* of $\mathbf{M}$ and is expressed as

$$\mathbf{S} = \mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}. \tag{2.35}$$

III. A special case of the above block matrix inversion method, where $\mathbf{M}_{12} = \mathbf{M}_{21}^T$, and $\mathbf{M}_{12}$ is a column vector, can be written as follows

$$\begin{bmatrix} \mathbf{U} & \mathbf{v} \\ \mathbf{v}^T & w \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{U}^{-1} + \mathbf{t}\mathbf{t}^T/s & -\mathbf{t}/s \\ -\mathbf{t}^T/s & 1/s \end{bmatrix}, \tag{2.36}$$

where $$\mathbf{t} = \mathbf{U}^{-1}\mathbf{v}, \tag{2.37}$$

and $$s = w - \mathbf{v}^T\mathbf{U}^{-1}\mathbf{v} \neq 0. \tag{2.38}$$

This special inverse by partition allows us to recycle the known inverse of a matrix to find the inverse of another matrix formed by expanding the original matrix with an extra row and column.

IV. With reference to (2.36), suppose $\mathbf{U}$ and $\mathbf{U}^{-1}$ are known with size $2n \times 2n$, $n \in \mathbb{Z}$, and they both have a checkerboard pattern. Suppose, following (2.14), the appending row-column vector $\mathbf{v}$ has the alternating-zero pattern,

$$\mathbf{v} = [v_1 \quad 0 \quad v_3 \quad 0 \quad \cdots \quad v_{2n-1} \quad 0]^T, \tag{2.39}$$

and the bottom-right corner scalar w is given by,

$$w = \mathbf{U}_{2n+1,2n+1}, \tag{2.40}$$

where v_i and w are non-zero real numbers. It then follows from (2.37), that

$$\mathbf{t}=\mathbf{U}^{-1}\mathbf{v}=\begin{bmatrix} U_{1,1} & 0 & U_{1,3} & & 0 \\ 0 & U_{2,2} & 0 & \cdots & U_{2,2n} \\ U_{3,1} & 0 & U_{3,3} & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & U_{2n,2} & 0 & \cdots & U_{2n,2n} \end{bmatrix}\begin{bmatrix} v_1 \\ 0 \\ v_3 \\ \vdots \\ 0 \end{bmatrix}=\begin{bmatrix} \sum_{\substack{j=1 \\ j \text{ odd}}}^{2n} U_{1,j}v_j \\ 0 \\ \sum_{\substack{j=1 \\ j \text{ odd}}}^{2n} U_{3,j}v_j \\ \vdots \\ 0 \end{bmatrix}. \tag{2.41}$$

In the other words, $\mathbf{t}$ also has an alternating-zero pattern similar to $\mathbf{v}$.

V. A similar result can be obtained for the odd sized $(2n-1)\times(2n-1)$, $n\in\mathbb{Z}$ matrices $\mathbf{U}$ and $\mathbf{U}^{-1}$. Suppose,

$$\mathbf{v}=\begin{bmatrix} 0 & v_2 & 0 & v_4 & \cdots & v_{2n-2} & 0 \end{bmatrix}^T, \tag{2.42}$$

and

$$w=\mathbf{U}_{2n+1,2n+1}. \tag{2.43}$$

It can be shown that,

$$\mathbf{t}=\mathbf{U}^{-1}\mathbf{v}=\begin{bmatrix} U_{1,1} & 0 & U_{1,3} & & U_{1,2n-1} \\ 0 & U_{2,2} & 0 & \cdots & 0 \\ U_{3,1} & 0 & U_{3,3} & & U_{3,2n-1} \\ & \vdots & & \ddots & \vdots \\ U_{2n-1,1} & 0 & U_{2n-1,3} & \cdots & U_{2n-1,2n-1} \end{bmatrix}\begin{bmatrix} 0 \\ v_2 \\ 0 \\ v_4 \\ \vdots \\ 0 \end{bmatrix}=\begin{bmatrix} 0 \\ \sum_{\substack{j=2 \\ j \text{ even}}}^{2n-1} U_{2,j}v_j \\ 0 \\ \sum_{\substack{j=2 \\ j \text{ even}}}^{2n-1} U_{4,j}v_j \\ \vdots \\ 0 \end{bmatrix}. \tag{2.44}$$

2.3.2 The Q_δ Matrix and its Inverse

This section shows the inverse of $\mathbf{Q}_\delta$ is symmetric and has checkerboard form. The result is an important step in proving the coefficients of a VFD Farrow filter are symmetrical or anti-symmetrical.

Result 1: $\boldsymbol{Q_\delta}$ is Hankel and has checkerboard form

Proof: Suppose the weighting $W(\omega,\delta)$ is quadrantally symmetric, then the individual elements in $\mathbf{Q}_\delta$ can be represented by

$$Q_\delta|_{l,k}=\int_{-\delta_{max}}^{\delta_{max}} W(\omega,\delta)\delta^l\,\delta^k\,d\delta \tag{2.45}$$

$$= \int_{-\delta_{max}}^{\delta_{max}} W(\omega,\delta)\delta^{l+k}\, d\delta\,. \tag{2.46}$$

Hence $\mathbf{Q}_\delta$ is Hankel. Also, it follows from (2.46), that if $l+k$ is odd then $Q_\delta = 0$ and if $l+k$ is even then

$$Q_\delta|_{l,k} = \int_{-\delta_{max}}^{\delta_{max}} W(\omega,\delta)\delta^{l+k}\; d\delta \neq 0\,. \tag{2.47}$$

That is, $\mathbf{Q}_\delta$ also has checkerboard form. ■

Result 2: $\boldsymbol{Q\delta}^{-1}$ is symmetric and has checkerboard form

Proof: Suppose $\mathbf{Q}_\delta$ can be represented as follows

$$\mathbf{Q}_\delta = \begin{bmatrix} x_1 & 0 & x_3 & 0 & \\ 0 & x_3 & 0 & x_5 & \cdots \\ x_3 & 0 & x_5 & 0 & \\ 0 & x_5 & 0 & x_7 & \\ & \vdots & & & \ddots \end{bmatrix}. \tag{2.48}$$

The matrix $\mathbf{Q}_\delta^{-1}$ can be constructed analytically as follows. Starting with,

$$x_1^{-1} = \tfrac{1}{x_1}, \tag{2.49}$$

it follows from (2.36) to (2.38) that,

$$\begin{bmatrix} x_1 & 0 \\ 0 & x_3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{x_1} & 0 \\ 0 & \frac{1}{x_3} \end{bmatrix}, \tag{2.50}$$

$$\begin{bmatrix} x_1 & 0 & x_3 \\ 0 & x_3 & 0 \\ x_3 & 0 & x_5 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{x_5}{x_1x_5-x_3^2} & 0 & \frac{-x_3}{x_1x_5-x_3^2} \\ 0 & \frac{1}{x_3} & 0 \\ \frac{-x_3}{x_1x_5-x_3^2} & 0 & \frac{x_1}{x_1x_5-x_3^2} \end{bmatrix}, \tag{2.51}$$

$$\begin{bmatrix} x_1 & 0 & x_3 & 0 \\ 0 & x_3 & 0 & x_5 \\ x_3 & 0 & x_5 & 0 \\ 0 & x_5 & 0 & x_7 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{x_5}{x_1x_5-x_3^2} & 0 & \frac{-x_3}{x_1x_5-x_3^2} & 0 \\ 0 & \frac{x_7}{x_3x_7-x_5^2} & 0 & \frac{-x_5}{x_3x_7-x_5^2} \\ \frac{-x_3}{x_1x_5-x_3^2} & 0 & \frac{x_1}{x_1x_5-x_3^2} & 0 \\ 0 & \frac{-x_5}{x_3x_7-x_5^2} & 0 & \frac{x_3}{x_3x_7-x_5^2} \end{bmatrix}, \text{ etc…} \tag{2.52}$$

As can be seen $\mathbf{Q}_\delta^{-1}$ is symmetric and has checkerboard form. ■

2.3.3 The Q_ω Matrix and its Inverse

This section shows the inverse of $\mathbf{Q}_\omega$ is bisymmetric which contributes to the efficient implementation of the closed form solution. The result also assists in proving the coefficients of the optimum VFD filter exhibit symmetry and anti-symmetry.

Result 3: $\boldsymbol{Q_\omega}$ is Toeplitz and symmetric

Proof: It follows from (2.15), the lk^{th} elements of $\mathbf{Q}_\omega$ is given by

$$Q_\omega|_{l,k} = \int_{-\pi}^{\pi} W(\omega,\delta) e^{-jk\omega} e^{+jl\omega}\, d\omega, \tag{2.53}$$

$$= \int_{-\pi}^{\pi} W(\omega,\delta) e^{-j(k-l)\omega}\, d\omega, \tag{2.54}$$

and if $W(\omega,\delta)$ is quadrantally symmetrical, (2.54) can be simplified further as follows

$$Q_\omega|_{l,k} = 2\times\int_{0}^{\pi} W(\omega,\delta)\cos((l-k)\omega)\, d\omega. \tag{2.55}$$

Now, on the diagonal of $\mathbf{Q}_\omega$, where $l=k$, (2.55) yields

$$Q_\omega|_{k,k} = 2\times\int_{0}^{\pi} W(\omega,\delta)\, d\omega = f(\delta), \tag{2.56}$$

where $f(\delta)$ is some function of δ. As for the off-diagonal terms, i.e. when $l\neq k$, (2.55) yields

$$\begin{aligned} Q_\omega|_{l,k} &= 2\times\int_{0}^{\pi} W(\omega,\delta)\cos((l-k)\omega)\, d\omega \\ &= 2\times\int_{0}^{\pi} W(\omega,\delta)\cos((k-l)\omega)\, d\omega \end{aligned}. \tag{2.57}$$

In other words,

$$Q_\omega|_{l,k} = Q_\omega|_{k,l}\ , \tag{2.58}$$

and

$$Q_\omega|_{l+m,k+m} = Q_\omega|_{l,k}\ . \tag{2.59}$$

i.e. $\mathbf{Q}_\omega$ is Toeplitz and symmetric. ■

Result 4: $\boldsymbol{Q\omega^{-1}}$ is bisymmetric

Proof: Since $\mathbf{Q}_\omega$ is symmetric and Toeplitz, the proof follows from [81]. ■

2.3.4 The Q Matrix and its Inverse

This section shows the inverse of $\mathbf{Q}$ which is a block symmetric and has block checkerboard form with each block being bisymmetric.

Result 5: $\boldsymbol{Q}$ is block Hankel and has block checkerboard form where the blocks are bisymmetric

Proof: Using *Result 1*, *Result 2*, and *Result 4*, the Kronecker product in (2.13) will have the following structure

$$\mathbf{Q} = \mathbf{Q}_\delta \otimes \mathbf{Q}_\omega = \begin{bmatrix} \boldsymbol{\Gamma}_1 & \mathbf{0}_K & \boldsymbol{\Gamma}_3 & \cdots & \mathbf{0}_K \\ \mathbf{0}_K & \boldsymbol{\Gamma}_3 & \mathbf{0}_K & \cdots & \boldsymbol{\Gamma}_m \\ \boldsymbol{\Gamma}_3 & \mathbf{0}_K & \boldsymbol{\Gamma}_5 & & \mathbf{0}_K \\ \vdots & \vdots & & \ddots & \vdots \\ \mathbf{0}_K & \boldsymbol{\Gamma}_m & \mathbf{0}_K & \cdots & \boldsymbol{\Gamma}_M \end{bmatrix}, \tag{2.60}$$

where $\boldsymbol{\Gamma}_i$ are bisymmetric matrices and, $\mathbf{0}_K$ is the $K \times K$ zero matrix. ■

Result 6: $\boldsymbol{Q}^{-1}$ is block symmetric and has block checkerboard form where the blocks are bisymmetric

Proof: In *Result 2*, it is shown that $\mathbf{Q}_\delta^{-1}$ is symmetric and has checkboard form. In other words, it can be represented by

$$\mathbf{Q}_\delta^{-1} = \begin{bmatrix} \eta_{1,1} & 0 & \eta_{1,3} & 0 & \\ 0 & \eta_{2,2} & 0 & \eta_{2,4} & \cdots \\ \eta_{3,1} & 0 & \eta_{3,3} & 0 & \\ 0 & \eta_{4,2} & 0 & \eta_{4,4} & \\ & \vdots & & & \ddots \end{bmatrix}. \tag{2.61}$$

Next, in *Result 4*, it is shown that $\mathbf{Q}_\omega^{-1}$ is bisymmetric. It then follows from (2.32) that

$$\mathbf{Q}^{-1} = \mathbf{Q}_\delta^{-1} \otimes \mathbf{Q}_\omega^{-1} = \begin{bmatrix} \eta_{1,1} & 0 & \eta_{1,3} & 0 & \\ 0 & \eta_{2,2} & 0 & \eta_{2,4} & \cdots \\ \eta_{3,1} & 0 & \eta_{3,3} & 0 & \\ 0 & \eta_{4,2} & 0 & \eta_{4,4} & \\ & \vdots & & & \ddots \end{bmatrix} \otimes \mathbf{Q}_\omega^{-1} \quad . \tag{2.62}$$

i.e.,
$$\mathbf{Q}^{-1}=\begin{bmatrix}\eta_{1,1}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \eta_{1,3}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \\ \mathbf{0}_K & \eta_{2,2}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \eta_{2,4}\mathbf{Q}_{\omega}^{-1} & \cdots \\ \eta_{3,1}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \eta_{3,3}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \\ \mathbf{0}_K & \eta_{4,2}\mathbf{Q}_{\omega}^{-1} & \mathbf{0}_K & \eta_{4,4}\mathbf{Q}_{\omega}^{-1} & \\ & \vdots & & & \ddots \end{bmatrix}. \tag{2.63}$$

That is, $\mathbf{Q}^{-1}$ is block symmetric and has block checkerboard form where the blocks are bisymmetric. ■

2.3.5 The p Vector

This section shows the alternating symmetric and anti-symmetric nature of $\mathbf{p}$ vector which ultimately impart onto the filter coefficients.

Result 7: $\boldsymbol{p}$ is a stacked vector composes of alternating symmetric and anti-symmetric vectors

Proof: It follows from (2.11) that $\mathbf{p}$ can be expressed as follows,

$$\mathbf{p}=[\mathbf{p}_0 \quad \mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \quad \cdots \quad \mathbf{p}_{M-1}], \tag{2.64}$$

and
$$\mathbf{p}_m=[p_{0,m} \quad p_{1,m} \quad p_{2,m} \quad p_{3,m} \quad \cdots \quad p_{K-1,m}], \tag{2.65}$$

where $m\in[0,M-1]$, $k\in[0,K-1]$, and

$$p_{k,m}=\int_{-\delta_{\max}}^{\delta_{\max}}\delta^m\otimes\int_{-\pi}^{\pi}W(\omega,\delta)\cos\left(\omega\left(\left(\tfrac{K-1}{2}-k\right)+\delta\right)\right)d\omega\,d\delta. \tag{2.66}$$

Next, replacing δ with $-\theta$ in (2.66) it follows that,

$$p_{k,m}=\int_{\delta_{\max}}^{-\delta_{\max}}(-\theta)^m\otimes\int_{-\pi}^{\pi}W(\omega,-\theta)\cos\left(\omega\left(\left(\tfrac{K-1}{2}-k\right)-\theta\right)\right)d\omega\,d(-\theta). \tag{2.67}$$

But since $W(\omega,\theta)=W(\omega,-\theta)$ due to quadrantally weighting and $\cos(A)=\cos(-A)$, therefore,

$$p_{k,m}=(-1)^m\int_{-\delta_{\max}}^{\delta_{\max}}\theta^m\otimes\int_{-\pi}^{\pi}W(\omega,\theta)\cos\left(\omega\left(-\left(\tfrac{K-1}{2}-k\right)+\theta\right)\right)d\omega\,d\theta. \tag{2.68}$$

Finally, replacing θ with δ, we get from (2.68),

$$p_{k,m}=(-1)^m\int_{-\delta_{\max}}^{\delta_{\max}}\delta^m\otimes\int_{-\pi}^{\pi}W(\omega,\delta)\cos\left(\omega\left(-\left(\tfrac{K-1}{2}-k\right)+\delta\right)\right)d\omega\,d\delta. \tag{2.69}$$

Now, observe that

$$p_{K-1-k,m} = \int_{-\delta_{\max}}^{\delta_{\max}} \delta^m \otimes \int_{-\pi}^{\pi} W(\omega,\delta)\cos\left(\omega\left(\tfrac{K-1}{2}+\delta-(K-1-k)\right)\right) d\omega\, d\delta \qquad (2.70)$$

$$= \int_{-\delta_{\max}}^{\delta_{\max}} \delta^m \otimes \int_{-\pi}^{\pi} W(\omega,\delta)\cos\left(\omega\left(-\left(\tfrac{K-1}{2}-k\right)+\delta\right)\right) d\omega\, d\delta\,. \qquad (2.71)$$

Therefore, comparing (2.69) and (2.71), we see that if m is odd, then

$$p_{k,m} = -p_{K-1-k,m}\ , \qquad (2.72)$$

and if m is even, then

$$p_{k,m} = +p_{K-1-k,m}\ . \qquad (2.73)$$

In other words, $\mathbf{p}$ has the following stacked structure

$$\mathbf{p} = \begin{bmatrix} \mathbf{s}_1^T & \mathbf{a}_2^T & \mathbf{s}_3^T & \mathbf{a}_4^T & \cdots \end{bmatrix}^T, \qquad (2.74)$$

where $\mathbf{s}_i$ is a symmetric vector, $\mathbf{a}_i$ is an anti-symmetric vector, the length of $\mathbf{s}_i$ and $\mathbf{a}_i$ are K, and $i \in [1, M]$. ■

2.3.6 Symmetry in the Optimum Coefficients **h**

This section shows the final form for the optimum coefficients **h** which inherit the alternating symmetric and anti-symmetric vectors. This is the conclusive proof that the optimum WLS filter coefficients are indeed symmetrical.

Result 8: optimum coefficients, $\mathbf{h}_{opt}$, is a stacked vector composes of alternating symmetric and anti-symmetric vectors

Proof: Combining (2.63) and (2.74), we now prove the optimum coefficients $\mathbf{h}_{opt}$ exhibit the symmetry and anti-symmetry characteristics as claimed in [45]. We first establish the following result. Let $\mathbf{B}$ be a bisymmetric matrix, $\mathbf{s}$ a symmetric vector, and $\mathbf{a}$ an anti-symmetric vector. Then $\mathbf{B}$, $\mathbf{s}$, and $\mathbf{a}$ must satisfy the following conditions,

$$\mathbf{JB} = \mathbf{BJ}\ , \qquad (2.75)$$

$$\mathbf{Js} = \mathbf{s}\ , \qquad (2.76)$$

and

$$\mathbf{Ja} = -\mathbf{a}\ . \qquad (2.77)$$

Accordingly,

$$\mathbf{J}(\mathbf{Bs}) = \mathbf{JBs} = \mathbf{BJs} = \mathbf{Bs}\,, \qquad (2.78)$$

and $$\mathbf{J}(\mathbf{Ba}) = \mathbf{JBa} = \mathbf{BJa} = -\mathbf{Ba}\,. \tag{2.79}$$

In other words, the product of a bisymmetric matrix with a symmetric vector is a symmetric vector; and the product of a bisymmetric matrix with an anti-symmetric vector is an anti-symmetric vector.

With reference to (2.16), (2.63), and (2.74), it follows that $\mathbf{h}_{opt}$ is given by

$$\mathbf{h}_{opt} = \mathbf{Q}^{-1}\mathbf{p} = \begin{bmatrix} \mathbf{\Gamma}'_{1,1} & \mathbf{0}_K & \mathbf{\Gamma}'_{1,3} & \cdots & \mathbf{0}_K \\ \mathbf{0}_K & \mathbf{\Gamma}'_{2,2} & \mathbf{0}_K & \cdots & \mathbf{\Gamma}'_{2,M} \\ \mathbf{\Gamma}'_{3,1} & \mathbf{0}_K & \mathbf{\Gamma}'_{3,3} & & \mathbf{0}_K \\ \vdots & \vdots & & \ddots & \vdots \\ \mathbf{0}_K & \mathbf{\Gamma}'_{M,2} & \mathbf{0}_K & \cdots & \mathbf{\Gamma}'_{M,M} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{a}_2 \\ \mathbf{s}_3 \\ \mathbf{a}_4 \\ \vdots \end{bmatrix}, \tag{2.80}$$

where $$\mathbf{\Gamma}'_{i,j} = \eta_{i,j}\mathbf{Q}_{\omega}^{-1}\ . \tag{2.81}$$

Let $$\mathbf{h}_{opt} = \begin{bmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T & \cdots & \mathbf{h}_M^T \end{bmatrix}^T\ , \tag{2.82}$$

it then follows from (2.80) that

$$\mathbf{h}_m = \begin{cases} \mathbf{\Gamma}'_{m,1}\mathbf{s}_1 + \mathbf{\Gamma}'_{m,3}\mathbf{s}_3 + \mathbf{\Gamma}'_{m,5}\mathbf{s}_5 + \cdots & m \text{ odd} \\ \mathbf{\Gamma}'_{m,2}\mathbf{a}_2 + \mathbf{\Gamma}'_{m,4}\mathbf{a}_4 + \mathbf{\Gamma}'_{m,6}\mathbf{a}_6 + \cdots & m \text{ even} \end{cases}, \tag{2.83}$$

and by (2.78) and (2.79) $\mathbf{h}_m$ is symmetric if m is odd, and anti-symmetric if m is even. ■

2.4 Remarks on Coefficient Symmetry

1. From equations (2.82) and (2.83), the symmetry property of the optimum filter coefficients $\mathbf{h}_{opt}$ can be expressed as

$$h(k,m) = (-1)^m\, h(K-1-k,m)\ . \tag{2.84}$$

2. As can be seen from the above discussion, a sufficient condition for $\mathbf{h}_{opt}$ to exhibit the stated symmetry property is that $W(\omega,\delta)$ is quadrantally symmetrical and $A_d(\omega)$ is also symmetrical.

2.5 Closed-Form Solution with Coefficient Symmetry

In [45, 46, 47, 76, 78, 82] a number of design methods that make use of coefficient symmetry are reported. In this section, we give an alternative set of expressions that will lead to an efficient algorithm to calculate $\mathbf{h}_{opt}$.

Incorporating the coefficient symmetry (2.84) into (1.8), the frequency response of the Farrow VFD filter can now be re-written as follows.

For an even length filter,

$$H(\omega,\delta)=\sum_{m=0}^{M_e}\left[\sum_{k=1}^{K_c}h_c(k,2m)\cos(k\omega)\right]\delta^{2m}+\sum_{m=1}^{M_o}\left[\sum_{k=1}^{K_s}h_s(k,2m-1)\sin(k\omega)\right]\delta^{2m-1}\quad , \tag{2.85}$$

and for an odd length filter

$$H(\omega,\delta)=\sum_{m=0}^{M_e}\left[\sum_{k=0}^{K_c}h_c(k,2m)\cos(k\omega)\right]\delta^{2m}+\sum_{m=1}^{M_o}\left[\sum_{k=1}^{K_s}h_s(k,2m-1)\sin(k\omega)\right]\delta^{2m-1}\quad , \tag{2.86}$$

where

$$M_e=\left\lfloor\tfrac{M-1}{2}\right\rfloor\ ,\ M_o=\left\lceil\tfrac{M-1}{2}\right\rceil, \tag{2.87}$$

$$K_c=\left\lceil\tfrac{K}{2}\right\rceil\ ,\ K_s=\left\lfloor\tfrac{K}{2}\right\rfloor, \tag{2.88}$$

$$h_c(k,2m)=\begin{cases}h(0,2m) & \text{for } k=0\\ 2h(k,2m) & \text{for } k\neq 0\end{cases}, \tag{2.89}$$

and

$$h_s(k,2m-1)=2h(k,2m-1). \tag{2.90}$$

Equation (2.85) and (2.86) can be put into vector-matrix notation as follows.

$$H(\omega,\delta)=\mathbf{h}_c^T(\boldsymbol{\delta}_e\otimes\mathbf{c})+\mathbf{h}_s^T(\boldsymbol{\delta}_o\otimes\mathbf{s}) \tag{2.91}$$

where

$$\mathbf{h}_c^T=\begin{bmatrix}\mathbf{h}_{c,0}^T & \mathbf{h}_{c,1}^T & \cdots & \mathbf{h}_{c,M_e}^T\end{bmatrix}\ \text{for}\ m\in\{0,1,\ldots,M_e\}, \tag{2.92}$$

$$\mathbf{h}_{c,m}^T=\begin{cases}[h(0,m) & h(1,m) & \cdots & h(K_c,m)] & \text{for odd } K\\ [h(1,m) & h(2,m) & \cdots & h(K_c,m)] & \text{for even } K\end{cases}, \tag{2.93}$$

$$\mathbf{h}_s^T=\begin{bmatrix}\mathbf{h}_{s,0}^T & \mathbf{h}_{s,1}^T & \cdots & \mathbf{h}_{s,M_o}^T\end{bmatrix}\ \text{for}\ m\in\{1,2,\ldots,M_o\}, \tag{2.94}$$

$$\mathbf{h}_{s,m}^T = [h(1,m) \quad h(2,m) \quad \cdots \quad h(K_s,m)] \,, \tag{2.95}$$

$$\boldsymbol{\delta}_e^T = \left[\delta^0 \quad \delta^2 \quad \cdots \quad \delta^{2M_e}\right], \tag{2.96}$$

$$\boldsymbol{\delta}_o^T = \left[\delta^1 \quad \delta^3 \quad \cdots \quad \delta^{2M_o-1}\right], \tag{2.97}$$

$$\mathbf{c}^T = \begin{cases} [\cos(0\omega) \quad \cos(1\omega) \quad \cdots \quad \cos(K_c\omega)] & \text{for odd } K \\ [\cos(1\omega) \quad \cos(2\omega) \quad \cdots \quad \cos(K_c\omega)] & \text{for even } K \end{cases}, \tag{2.98}$$

and
$$\mathbf{s}^T = [\sin(1\omega) \quad \sin(2\omega) \quad \cdots \quad \sin(K_s\omega)] \,. \tag{2.99}$$

Next, the desired frequency response (2.7) can be expressed as

$$H_d(\omega,\delta) = A_d(\omega)\cdot e^{-j\omega\hat{k}} \tag{2.100}$$

where
$$\hat{k} = \begin{cases} \frac{K-1}{2} + \delta & \text{for odd } K \\ \frac{K-1}{2} + \delta - \frac{1}{2} & \text{for even } K \end{cases}. \tag{2.101}$$

Combining (2.91) and (2.100), the error function of the WLS design problem of (2.8) can be split into a symmetrical (ε_c) and an anti-symmetrical (ε_s) part.

$$H(\omega,\delta) - H_d(\omega,\delta) = \varepsilon_c + \varepsilon_s = \varepsilon(\omega,\delta) \tag{2.102}$$

where
$$\varepsilon_c(\omega,\delta) = \mathbf{h}_c^T(\boldsymbol{\delta}_e \otimes \mathbf{c}) - A_d(\omega).\cos\left(\omega\hat{k}\right) \tag{2.103}$$

and
$$\varepsilon_s(\omega,\delta) = \mathbf{h}_s^T(\boldsymbol{\delta}_o \otimes \mathbf{s}) - A_d(\omega).\sin\left(\omega\hat{k}\right). \tag{2.104}$$

The WLS design problem of (2.8) can thus be reformulated as

$$\min_{\mathbf{h}_c} \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)|\varepsilon_c|^2 \, d\omega \, d\delta + \min_{\mathbf{h}_s} \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)|\varepsilon_s|^2 \, d\omega \, d\delta \,. \tag{2.105}$$

By adopting the derivation in [83], the cost function (2.105) can be expressed as

$$J(\mathbf{h}) = \left(\mathbf{h}_c^T\mathbf{Q}_c\mathbf{h}_c - 2\mathbf{h}_c^T\mathbf{p}_c + r_c\right) + \left(\mathbf{h}_s^T\mathbf{Q}_s\mathbf{h}_s - 2\mathbf{h}_s^T\mathbf{p}_s + r_s\right), \tag{2.106}$$

where
$$\mathbf{Q}_c = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left(\left(\boldsymbol{\delta}_e\boldsymbol{\delta}_e^T\right) \otimes \left(\mathbf{c}\mathbf{c}^T\right)\right) d\omega \, d\delta \,, \tag{2.107}$$

$$\mathbf{Q}_s = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left(\left(\boldsymbol{\delta}_o\boldsymbol{\delta}_o^T\right) \otimes \left(\mathbf{s}\mathbf{s}^T\right)\right) d\omega \, d\delta \,, \tag{2.108}$$

$$\mathbf{p}_c = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left(\boldsymbol{\delta}_e \otimes \mathbb{R}\left[\mathbf{c}\cdot\cos\left(\omega\hat{k}\right)\right]\right) d\omega \, d\delta \,, \tag{2.109}$$

$$\mathbf{p}_s = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left(\boldsymbol{\delta}_o \otimes \mathbb{R}\left[\mathbf{s}\cdot\sin\left(\omega\hat{k}\right)\right]\right) d\omega \, d\delta \,, \tag{2.110}$$

$$r_c = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|\cos\left(\omega\hat{k}\right)\right|^2 d\omega\, d\delta\,, \tag{2.111}$$

and

$$r_s = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|\sin\left(\omega\hat{k}\right)\right|^2 d\omega\, d\delta\,. \tag{2.112}$$

The solution of (2.105) is given by [77],

$$\mathbf{h}_c = \mathbf{Q}_c^{-1}\mathbf{p}_c\ , \tag{2.113}$$

and

$$\mathbf{h}_s = \mathbf{Q}_s^{-1}\mathbf{p}_s\,, \tag{2.114}$$

where the coefficients in $\mathbf{h}_c$ and $\mathbf{h}_s$ are shown in (2.92) and (2.94) respectively.

The relation between $\mathbf{h}_c$ and $\mathbf{h}_s$ to $\mathbf{h}_{opt}$ of (2.16) is given by

$$\mathbf{h}_{opt}^T = \begin{cases} \left[\overline{\mathbf{h}}_{c,0}^T \quad \overline{\mathbf{h}}_{s,1}^T \quad \overline{\mathbf{h}}_{c,1}^T \quad \overline{\mathbf{h}}_{s,2}^T \quad \cdots \quad \overline{\mathbf{h}}_{s,M_o}^T \quad \overline{\mathbf{h}}_{c,M_e}^T\right] & \text{for odd } M \\ \left[\overline{\mathbf{h}}_{c,0}^T \quad \overline{\mathbf{h}}_{s,1}^T \quad \overline{\mathbf{h}}_{c,1}^T \quad \overline{\mathbf{h}}_{s,2}^T \quad \cdots \quad \overline{\mathbf{h}}_{c,M_e}^T \quad \overline{\mathbf{h}}_{s,M_o}^T\right] & \text{for even } M \end{cases}, \tag{2.115}$$

where

$$\overline{\mathbf{h}}_{c,m}^T = \begin{cases} \left[\frac{h(K_c,m)}{2} \quad \cdots \quad \frac{h(1,m)}{2} \quad h(0,m) \quad \frac{h(1,m)}{2} \quad \cdots \quad \frac{h(K_c,m)}{2}\right] & \text{for odd } K \\ \left[\frac{h(K_c,m)}{2} \quad \cdots \quad \frac{h(1,m)}{2} \quad \frac{h(1,m)}{2} \quad \cdots \quad \frac{h(K_c,m)}{2}\right] & \text{for even } K \end{cases}, \tag{2.116}$$

$$\overline{\mathbf{h}}_{s,m}^T = \begin{cases} \left[\frac{-h(K_s,m)}{2} \quad \cdots \quad \frac{-h(1,m)}{2} \quad 0 \quad \frac{h(1,m)}{2} \quad \cdots \quad \frac{h(K_s,m)}{2}\right] & \text{for odd } K \\ \left[\frac{-h(K_s,m)}{2} \quad \cdots \quad \frac{-h(1,m)}{2} \quad \frac{h(1,m)}{2} \quad \cdots \quad \frac{h(K_s,m)}{2}\right] & \text{for even } K \end{cases}, \tag{2.117}$$

and their inner elements can be derived from (2.93) and (2.95).

As claimed in [45, 46] the separation of coefficients into $\mathbf{h}_c$ and $\mathbf{h}_s$ can lead to the efficient computation of $\mathbf{h}_{opt}$. This thesis supports this claim. In addition, a contribution of the thesis is the derivation of a more computational friendly expression for $\mathbf{h}_c$ and $\mathbf{h}_s$, as given by (2.113) and (2.114).

3 WLS Design Examples

3.1 Introduction

This section presents four examples demonstrating the efficiency of the proposed design method in solving the WLS Farrow VFD filter design problem: a small and a large Almost-Flat (AF) design and a small and large Band-Pass (BP) design where by "small", we mean $K < 10$, and by "large", we mean $K > 50$. The AF and BP characteristics are illustrated in Figure 6 and Figure 7. In the case of the small filter design examples, the filter coefficients are provided for readers' verification purposes.

The cost comparison between the proposed symmetry coefficients design (shown in Section 2.5) and the non-symmetrical method [75] (shown in equation (2.16)) are

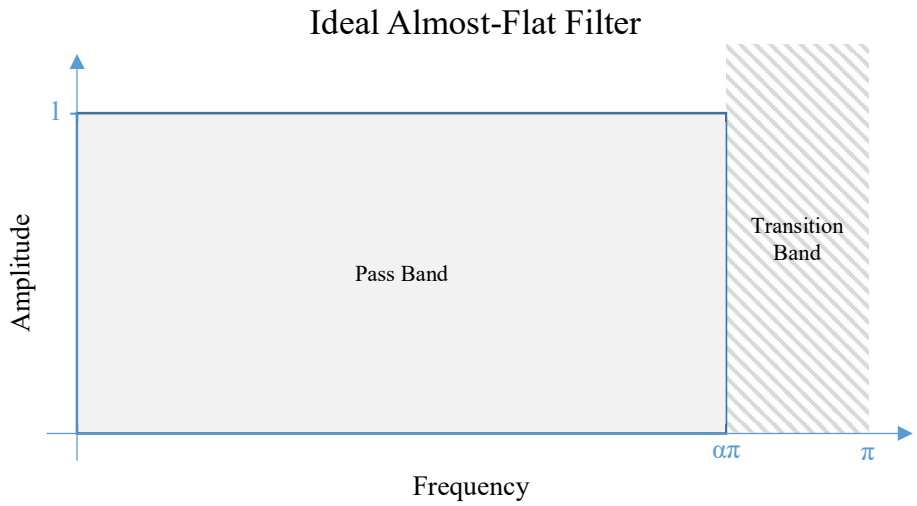


Figure 6 - The ideal almost-flat filter

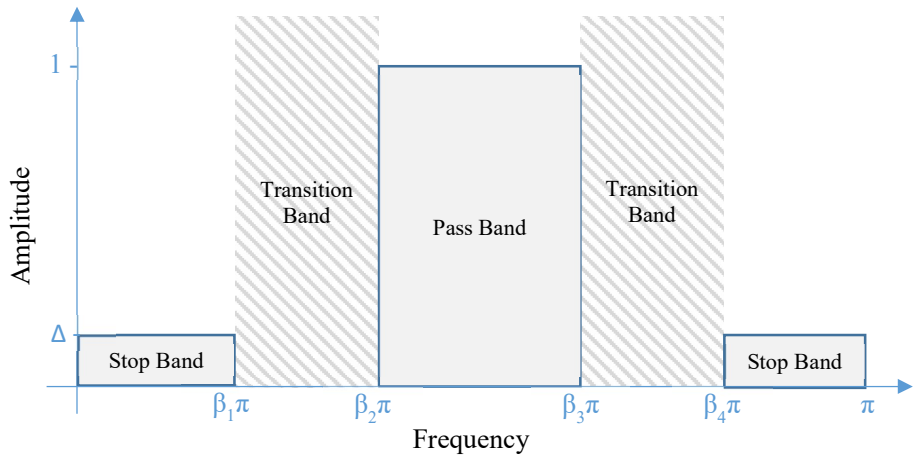


Figure 7 – The ideal band pass filter

highlighted in each design. The three cost parameters that are used are the weighted least squares error ε_{LS}, the maximum frequency error ε_{MM}, and the maximum phase response error ε_{MP} as defined by

$$\varepsilon_{LS} = \int_{-\delta_{\max}}^{\delta_{\max}} \int_{-\pi}^{\pi} \left(H(\omega,\delta) - H_d(\omega,\delta)\right)^2 d\omega\, d\delta\,, \tag{3.1}$$

$$\varepsilon_{MM} = \max\left|H(\omega,\delta) - H_d(\omega,\delta)\right|, \tag{3.2}$$

and

$$\varepsilon_{MP} = \max\left|\measuredangle H(\omega,\delta) - \measuredangle H_d(\omega,\delta)\right|. \tag{3.3}$$

Another parameter is the timer, which gives an indication of the resource utilisation during computation. The results are computed from MATLAB's `tictoc` function wrapped around the WLS Farrow VFD filter design construction and computation of the optimum coefficients. The computer used for computing these results has an Intel® i5 2.4GHz CPU with 3G RAM on 32-bit Windows10 and 32-bit MATLAB 2015a.

All results presented in this chapter, in both frequency ω and delay δ, are uniformly sampled at step size $\pi/2048$ and $1/128$ respectively.

3.2 AF Small Design

The advantage of a small design is the possibility for manually derivation of the optimum filter coefficients and compare against software computed filter coefficients. Hence, it allows the reader to compare and gain confidence in the result accuracy.

In the small AF design, consider these following parameters $M = 4$, $K = 8$, $\delta_{max} = 0.5$, $\alpha = 0.85$, and

$$W(\omega,\delta) = \begin{cases} 1, & \begin{array}{c} 0 \le \omega \le \alpha\pi \\ -\delta_{max} \le \delta \le \delta_{max} \end{array} \\ 0, & \text{elsewhere} \end{cases}. \tag{3.4}$$

The optimum coefficients are tabulated in Table 1 for user verification purposes.

Looking at the cost results in Table 2, we observe that the costs of the symmetrical and non-symmetrical design problem in [75] are similar. This should be no surprise as the computation complexity of 32 coefficients filter are trivial for any modern computer. The slight difference is due to the poorly conditioned **Q** matrix with a condition number of 2.082×10^4 in the non-symmetrical design problem. The

bootstrapping inversion method, of Section 2.3.1, overcame this problem to provide a favourable outcome. In the large design, the inversion issue will be more apparent.

Table 1 – Optimum coefficients of a M=4, K=8 and α=0.85 almost-flat filter

-0.0454154714707797	-0.00948869546118458	0.189845541372549	0.0389017366805486
0.0910791904492382	0.0315710340226714	-0.379169371972149	-0.129127239058728
-0.188305517176336	-0.119010718762338	0.771906095111603	0.482413107785682
0.628147458017961	1.24842857742149	-0.521902352298373	-1.00345492352191
0.628147458017961	-1.24842857742149	-0.521902352298373	1.00345492352191
-0.188305517176336	0.119010718762338	0.771906095111603	-0.482413107785682
0.0910791904492382	-0.0315710340226714	-0.379169371972149	0.129127239058728
-0.0454154714707797	0.00948869546118458	0.189845541372549	-0.0389017366805486

Table 2 – Results table for almost-flat small design (M=4, K=8 and α=0.85)

Design	ε_{LS}	ε_{MM}	ε_{MP}	Timer (s)
Symmetrical	0.002449597008	0.2029795967	0.03312357270	0.0153
Non-symmetrical	0.002449969034	0.2033511036	0.03381327316	0.0262

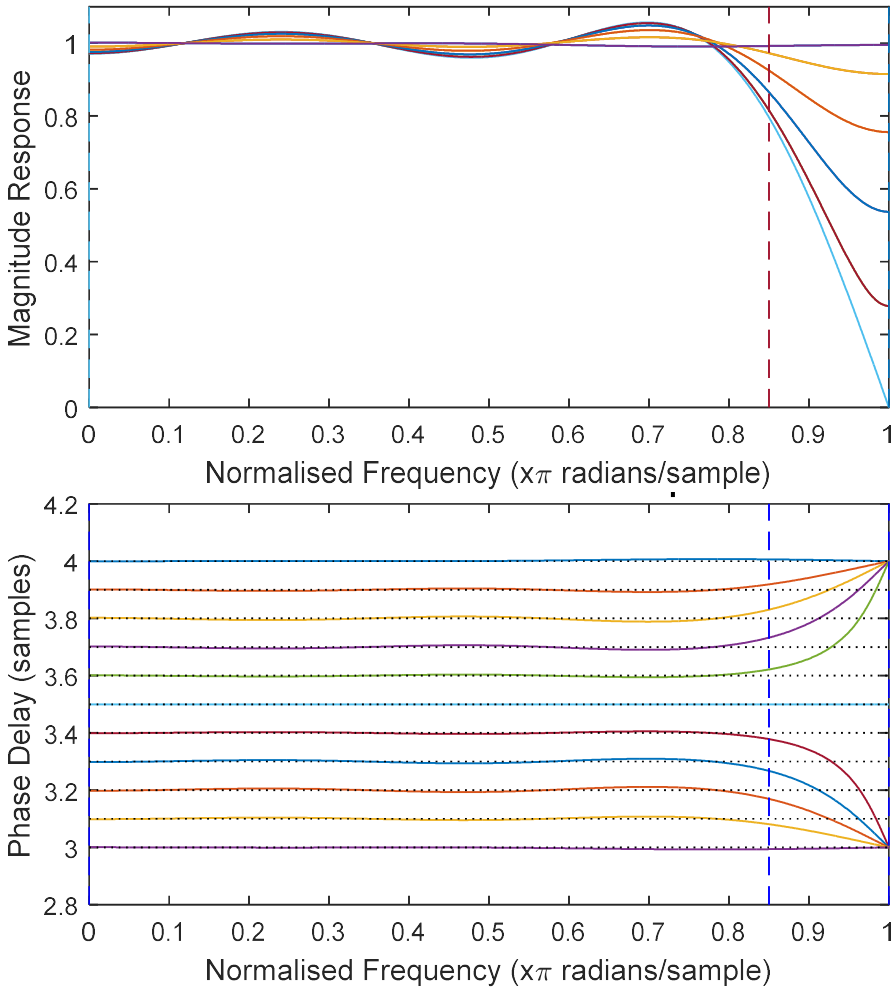


Figure 8 – Almost-flat small symmetry design (M=4, K=8 and α=0.85)

The timer output behaved as expected with the symmetrical design problem, using almost half of the computational and memory resources, thus performing faster than the non-symmetrical design problem.

The magnitude and phase response for the symmetrical design problem are shown in Figure 8. One can see that the phase response at higher frequency, as $\omega \to \pi$, that each fractional delay converges to the nearest integer group delay at π. That is because $H\left(e^{-j\pi}\right)$ must be real at π for the system to be stable.

3.3 AF Large Design

For the large design, an odd length filter will be presented. Consider these following parameters $M = 6$, $K = 51$, $\delta_{max} = 0.5$, $\alpha = 0.87$ and with same weighting as in Section 3.2.

Due to the increased computational complexity and increased number of filter coefficients, in this case with 306 coefficients, the non-symmetrical design problem produced a nonsensical cost outputs as seen in Table 3 and Figure 9. This is due to the high condition number for **Q** matrix at 4.384×10^{13}, resulting in poor inversion. However, the bootstrapping method for the symmetrical coefficient design was able to achieve satisfactory errors.

Table 3 – Results table for a M=6, K=51 and α=0.87 almost-flat large design

Design	ε_{LS}	ε_{MM}	ε_{MP}	Timer (s)
Symmetrical	$6.4340654725 \times 10^{-9}$	$1.9081036026 \times 10^{-4}$	$2.0874373214 \times 10^{-4}$	0.1750
Non-symmetrical	$3.6526394848 \times 10^{-2}$	$7.6531894198 \times 10^{-1}$	$3.0246139257 \times 10^{-1}$	0.3982

The magnitude and phase responses in Figure 10 show excellent response adherence in the desired range. The author recommends a tighter transition range, or higher cut-off frequency, on the AF filter to move it towards a closer approximation of an ideal AP filter.

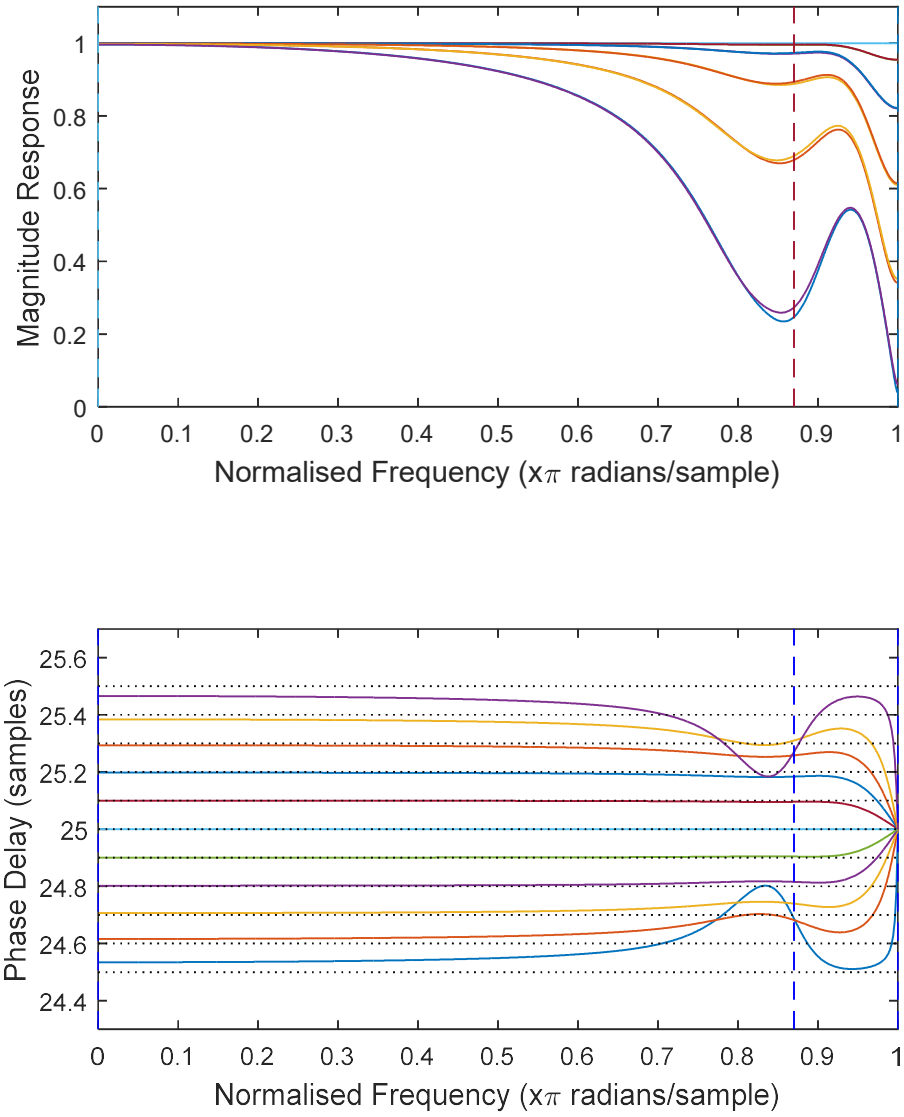


Figure 9 – Non-symmetrical Almost-flat large symmetry design (M=6, K=51 and α=0.87)

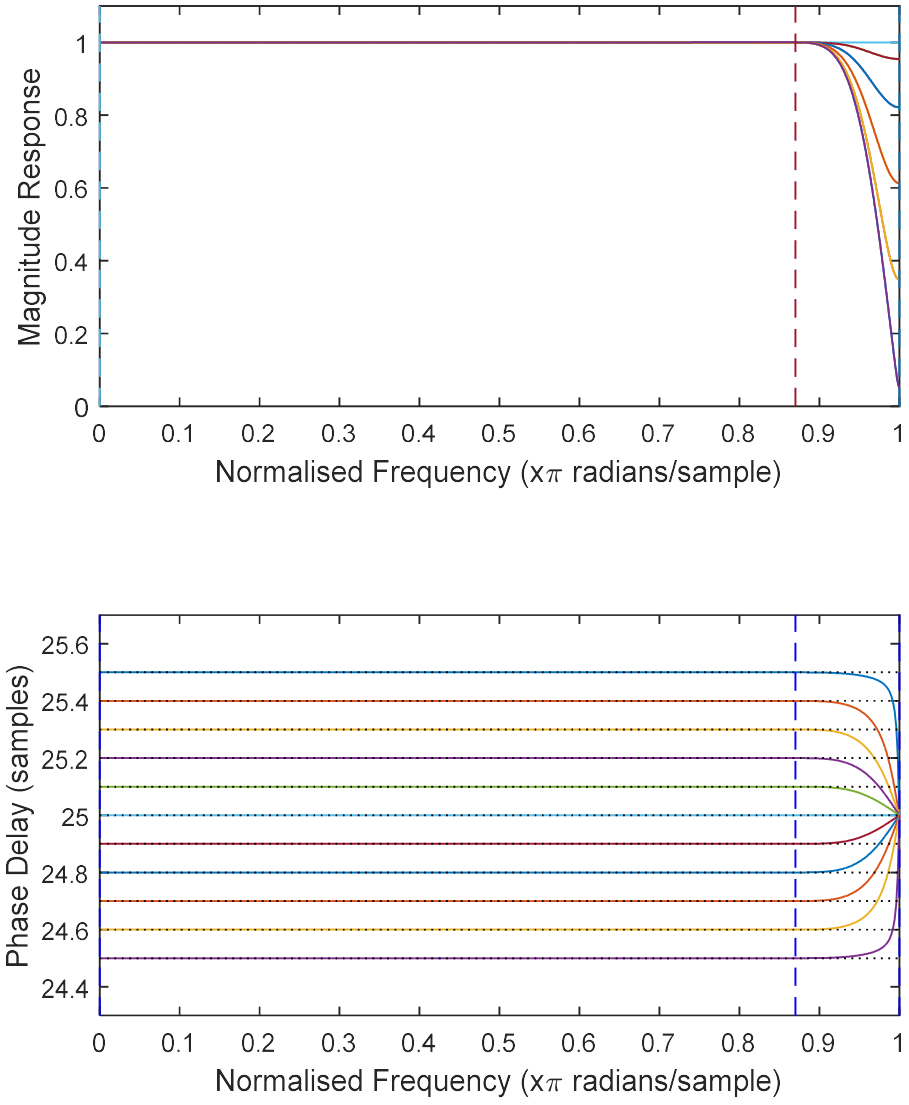


Figure 10 - Almost-flat large symmetry design (M=6, K=51 and α=0.87)

3.4 BP Small Design

An example for the small BP design, i.e. a bandpass filter that passes frequency between 4 kHz and 6 kHz, allowing transition bands from 3 kHz to 4 kHz and 6 kHz to 8 kHz with a 20 kHz sampling rate. In this BP design, consider these following parameters $M = 4$, $K = 9$, $\delta_{max} = 0.5$, $\beta_1 = 0.3$, $\beta_2 = 0.4$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and,

$$W(\omega,\delta) = \begin{cases} 1, & \begin{matrix} 0 \le \omega \le 0.3\pi, 0.4\pi \le \omega \le 0.6\pi, 0.8\pi \le \omega \le \pi \\ -\delta_{max} \le \delta \le \delta_{max} \end{matrix} \\ 0, & \text{elsewhere} \end{cases} . \quad (3.5)$$

Table 4 – Optimum coefficients of BP filter (M=4, K=9 and β=[0.3, 0.4, 0.6, 0.8])

0.110087829507077	-0.107661272912304	-0.100926769925639	0.0897052887744503	0.0136176406356583
0.0292463646466117	0.267629745022509	-0.130720434384436	-0.102933811476546	0.0461681630863495
-0.259057199243343	0.150816477642481	0.332639727239292	-0.107444848491442	-0.0723864967528343
-0.0198393848016551	-0.548576111315298	0.0781711756054175	0.237598384372763	-0.0271167161237145
0.333373238128079	0	-0.464405634766781	0	0.108351011577000
-0.0198393848016551	0.548576111315298	0.0781711756054175	-0.237598384372763	-0.0271167161237145
-0.259057199243343	-0.150816477642481	0.332639727239292	0.107444848491442	-0.0723864967528343
0.0292463646466117	-0.267629745022509	-0.130720434384436	0.102933811476546	0.0461681630863495
0.110087829507077	0.107661272912304	-0.100926769925639	-0.0897052887744503	0.0136176406356583

Table 5 – Results table for BP small design (M=4, K=9 and β=[0.3, 0.4, 0.6, 0.8])

Design	ε_{LS}	ε_{MM}	ε_{MP}	Timer (s)
Symmetrical	0.010962235650707	0.272243971707058	0.035266848637307	0.02134
Non-symmetrical	0.010963410322801	0.273835191216718	0.036660178489733	0.04740

The optimum coefficients for this BP WLS Farrow VFD filter design are tabulated in Table 4 for user verification purposes.

In Table 5, the symmetry design and non-symmetry design have a slight difference between the ε_{MM} cost and ε_{MP} cost due to the tight transition region and the bootstrapping inversion method.

The magnitude response in Figure 11 shows the general characteristic of a band-pass filter, but clearly it is not satisfactory. Similarly, for the phase response, it is encouraging to see the filter features good approximation within the desired range, however the performances are far from acceptable.

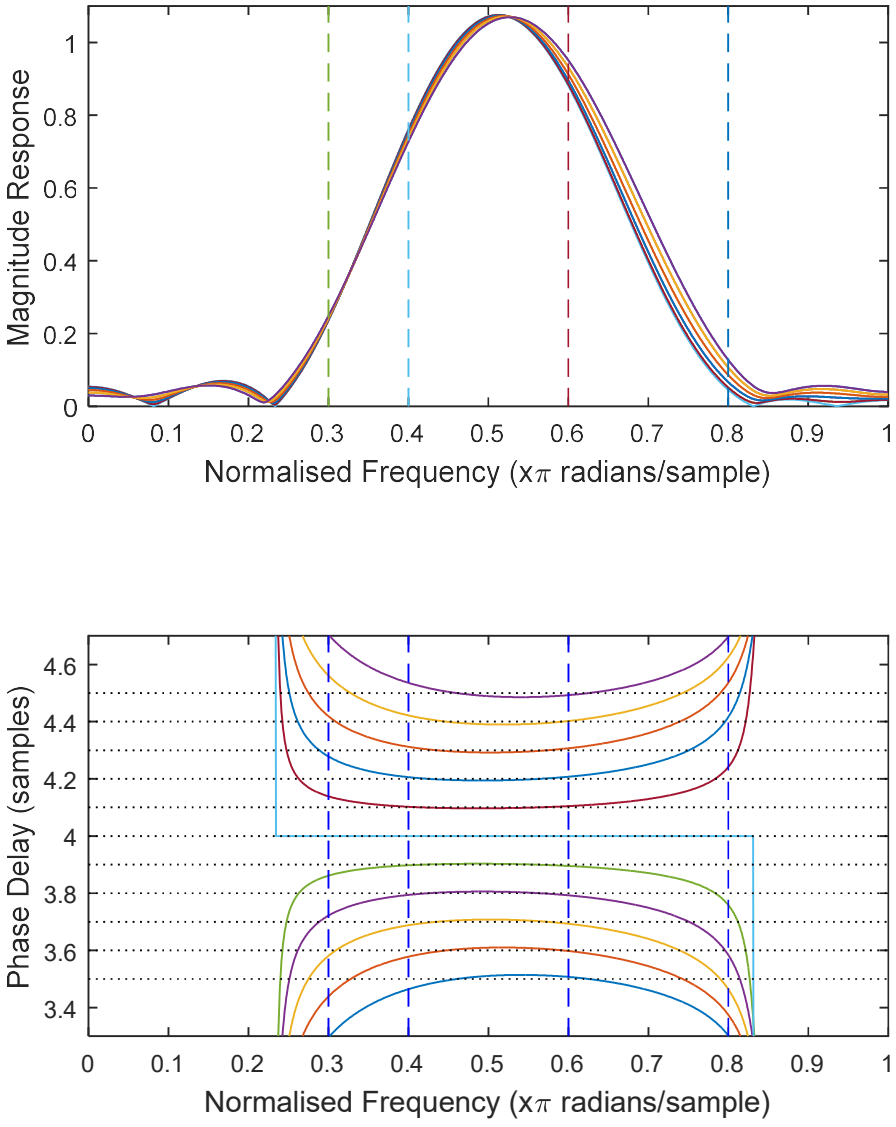


Figure 11 - Band pass small symmetry design (M=4, K=9 and β=[0.3, 0.4, 0.6, 0.8])

3.5 BP Large Design

In the previous section, the small BP design was deemed not suitable as it lacks the filter dimension for a reasonable BP filter design. This section will showcase a larger filter design comparable to real-world parameters.

Using the same BP and BS cut-off frequencies, as Section 3.4, we increase the filter size to $M = 7$ and $K = 62$. Due to the large matrix, consisting of 434×434 entries, MATLAB's `inv` function was not able to compute the inverse of this matrix which has a condition number of 5.3417×10^{14}. Thus, naturally, the non-symmetrical result is omitted. Whereas the symmetrical with bootstrapping method was able to produce adequate responses. Table 6 shows both methods with the non-symmetrical design output and symmetrical design output. Note that the output of non-symmetrical

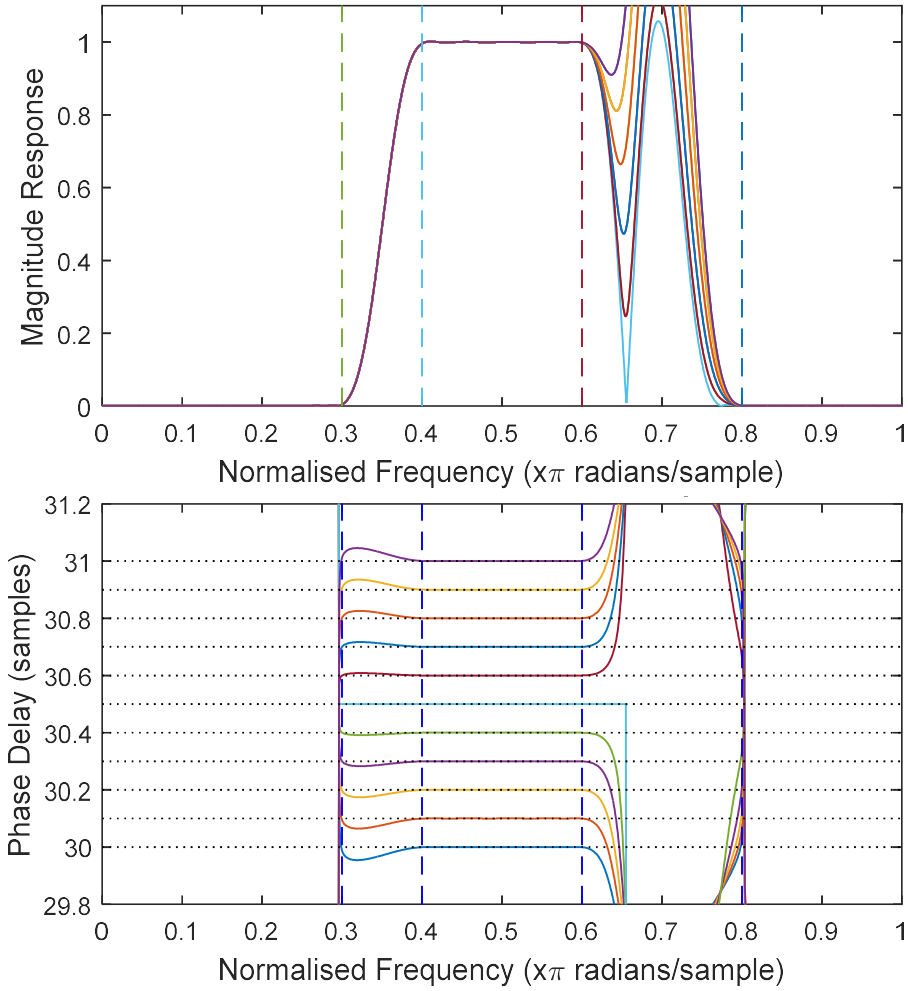


Figure 12 - Band pass large symmetry design (M=7, K=62 and β=[0.3, 0.4, 0.6, 0.8])

design is nonsensical and unreliable because of the fact that the inversion cannot be computed.

The frequency response, in Figure 12, shows a better BP design compared to the small design in Figure 11. The maximum error in the desired range being $\varepsilon_{MM} \cong 0.00416381$ and an almost-flat phase response in the desired range.

Table 6 – Results table for BP large design (M=7, K=62 and β=[0.3, 0.4, 0.6, 0.8])

Design	ε_{LS}	ε_{MM}	ε_{MP}	Timer (s)
Symmetrical	$5.543567929751\times10^{-7}$	0.004163805638	$5.609041334047\times10^{-4}$	0.2630
Non-symmetrical	$2.384114488155\times10^{+5}$	$2.19180745\times10^{+3}$	-	0.7697

3.6 Concluding Remarks

This chapter demonstrated that using symmetrical coefficients in the WLS Farrow VFD filter design produces more accurate design than using non-symmetrical coefficients. One contributor to the accuracy is the matrix bootstrapping inversion method used to calculate the poorly conditioned matrices. Through multiple examples shown in this Chapter, it was demonstrated that the built-in matrix inversion function in MATLAB cannot be relied on to invert a large matrix. A quick tutorial about the bootstrapping method can be found in Appendix A1. Whilst accuracy is an important factor, so is the computational time. It is established in this chapter that symmetrical coefficients filter design is quicker compared to non-symmetrical coefficients filter design.

A noteworthy observation in the AF WLS Farrow VFD filter is the maximum phase response of an even or odd K-length filter. It was pointed out earlier that all phase response will converge to an integer group delay at π because of the properties of an FIR filter. This new insight indicated that the location of the maximum phase response error for an even or odd length filter are consistently different. Looking at Figure 8, where K is even, the maximum phase response error occurs at the cut-off frequency $\alpha\pi$. Whereas in Figure 10, where K is odd, the maximum phase response error occurs at 0. This insight and explanation will be discussed in Chapter 4.

4 WLS Design Guide

4.1 Formulation of the AF WLS Design Guide

This chapter will showcase three useful design guides for digital filter designers to dimension their desired filter without complex algorithms. The three design guides are the WLS cost J_{wls}, the maximum magnitude response error J_m, and the maximum phase response error J_p. The guides were formed after extensive numerical studies on the design of AF WLS Farrow VFD filters whilst observing the influences of the three key design parameters *K*, *M*, and α.

The design parameters used to construct the design guides are as follows: $K \in [7,50]$ where $\Delta K = 1$, $M \in [3,8]$ where $\Delta M = 1$, $\delta = [-0.5, 0.5]$ where $\Delta\delta = 0.1$, and $\alpha \in [0.80, 0.95]$ where $\Delta\alpha = 0.25$. That is, the design guides were obtained after $44 \times 6 \times 11 \times 7 = 20,328$ designs have been performed. In all designs, the frequency ω and δ are uniformly sampled at a step size of $\pi/4096$ and $1/64$ respectively. Note this extensive numerical study is manageable only because of the efficient closed form solution method presented in Section 2.5.

The tool used to obtain the design guides is the `cftool` curve fitting package in MATLAB. The algorithms incorporated in `cftool` are Levenberg-Marquardt algorithm (LMA) [84] and Tukey's biweight [85]. The LMA is known to be robust and is able to work with extreme gradient fluctuations with the biweight function allows a better vigorousness in the regression [85].

In this thesis, two measures to assess the accuracy of the design guides are presented. The first measure is the maximum differences between the cost predicted by the design guide against the actual cost. The second measure is more sophisticated and is based on the Pearson-product-moment correlation coefficient (PCC) [86] which evaluates the correlation between two matrices. PCC utilises the following equation,

$$r(\alpha) = \frac{\sum_m \sum_k \left(J_{a_{mk}}(\alpha) - \bar{J}_a(\alpha)\right)\left(J_{A_{mk}}(\alpha) - \bar{J}_A(\alpha)\right)}{\sqrt{\left(\sum_m \sum_k \left(J_{a_{mk}}(\alpha) - \bar{J}_a(\alpha)\right)^2\right)\left(\sum_m \sum_k \left(J_{A_{mk}}(\alpha) - \bar{J}_A(\alpha)\right)^2\right)}}, \tag{4.1}$$

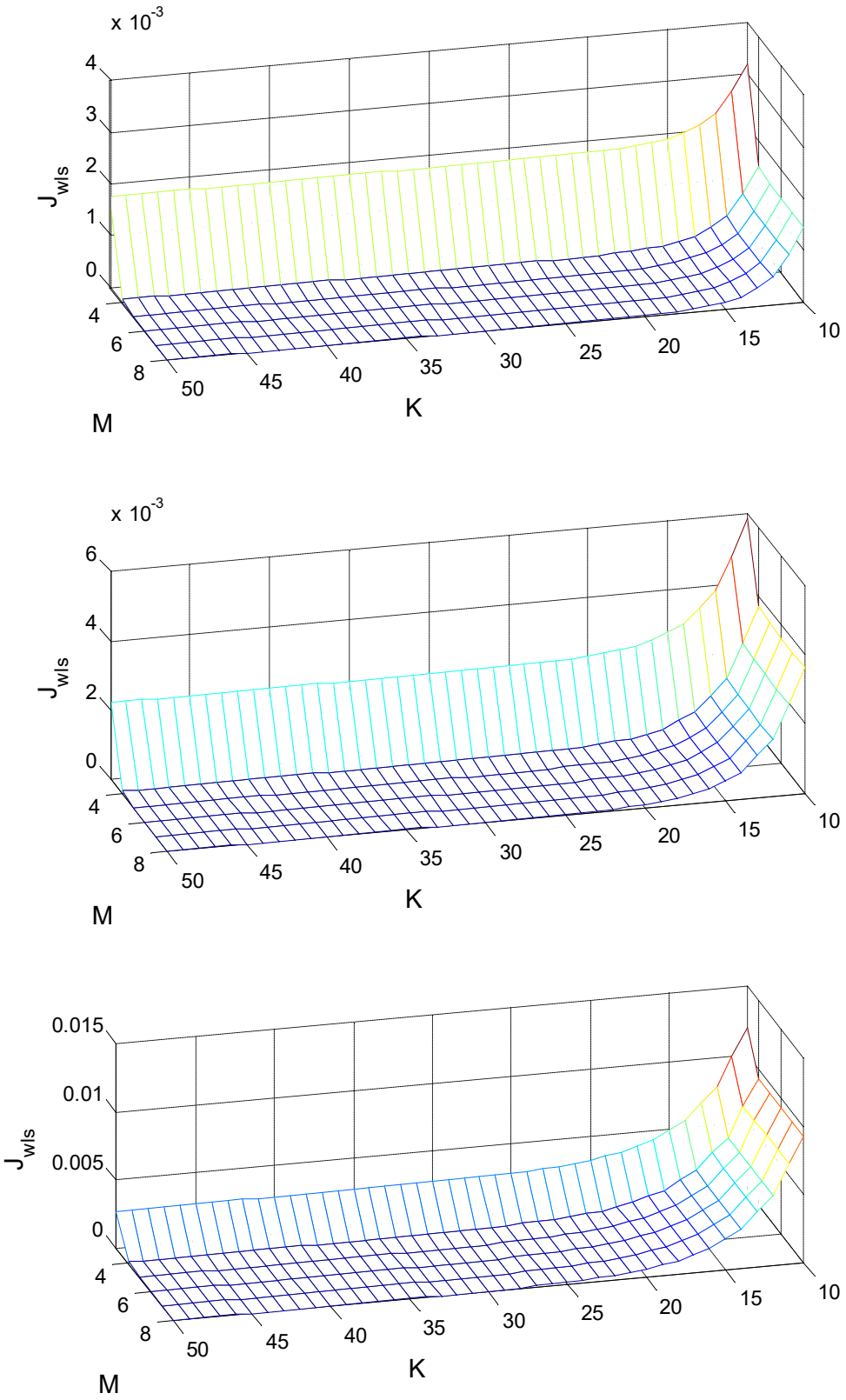


Figure 13 – Actual maximum WLS error surface for (top) α=0.87, (middle) α=0.90, and (bottom) α=0.93

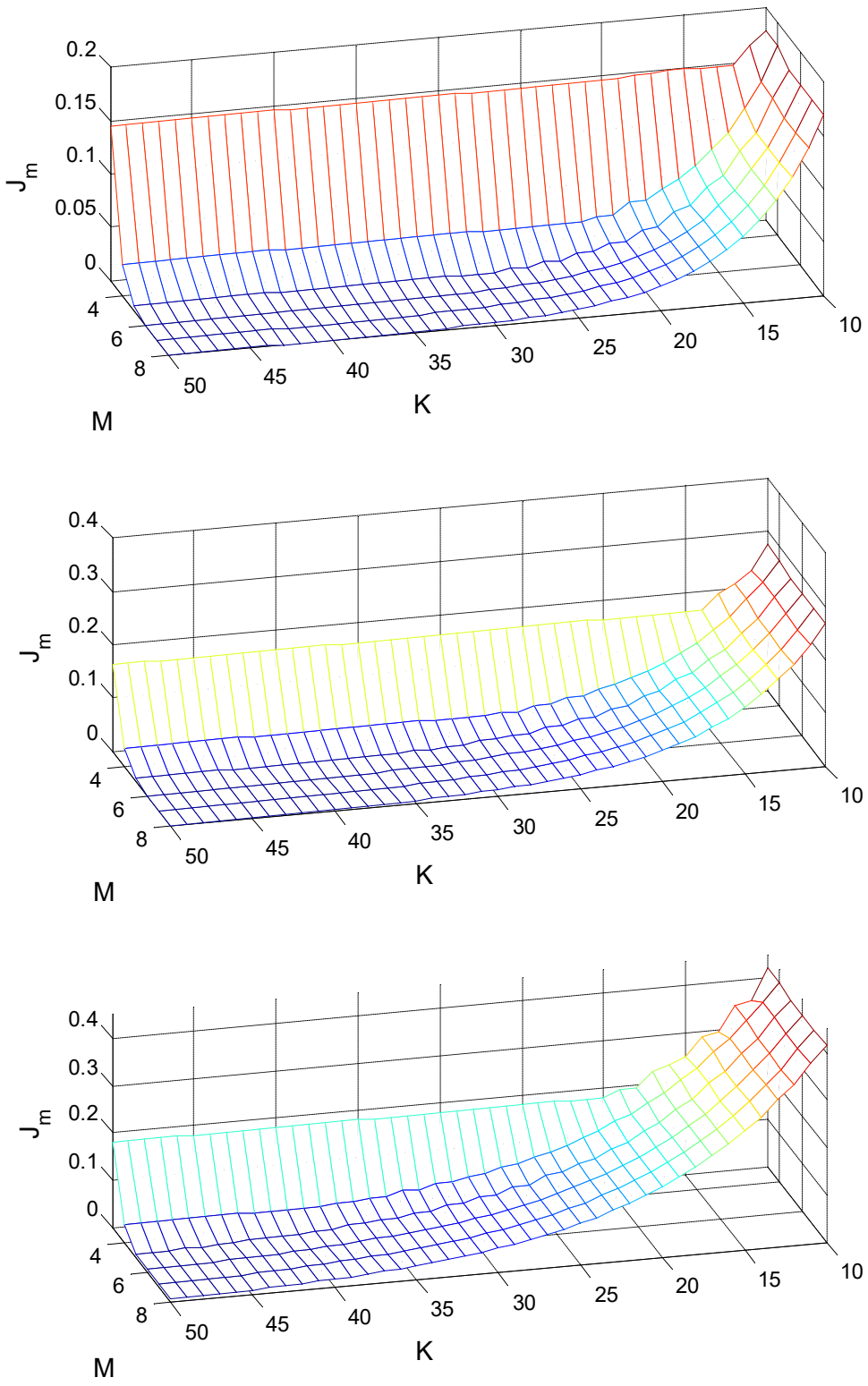


Figure 14 - Actual maximum magnitude error surface for (top) α=0.87, (middle) α=0.90, and (bottom) α=0.93

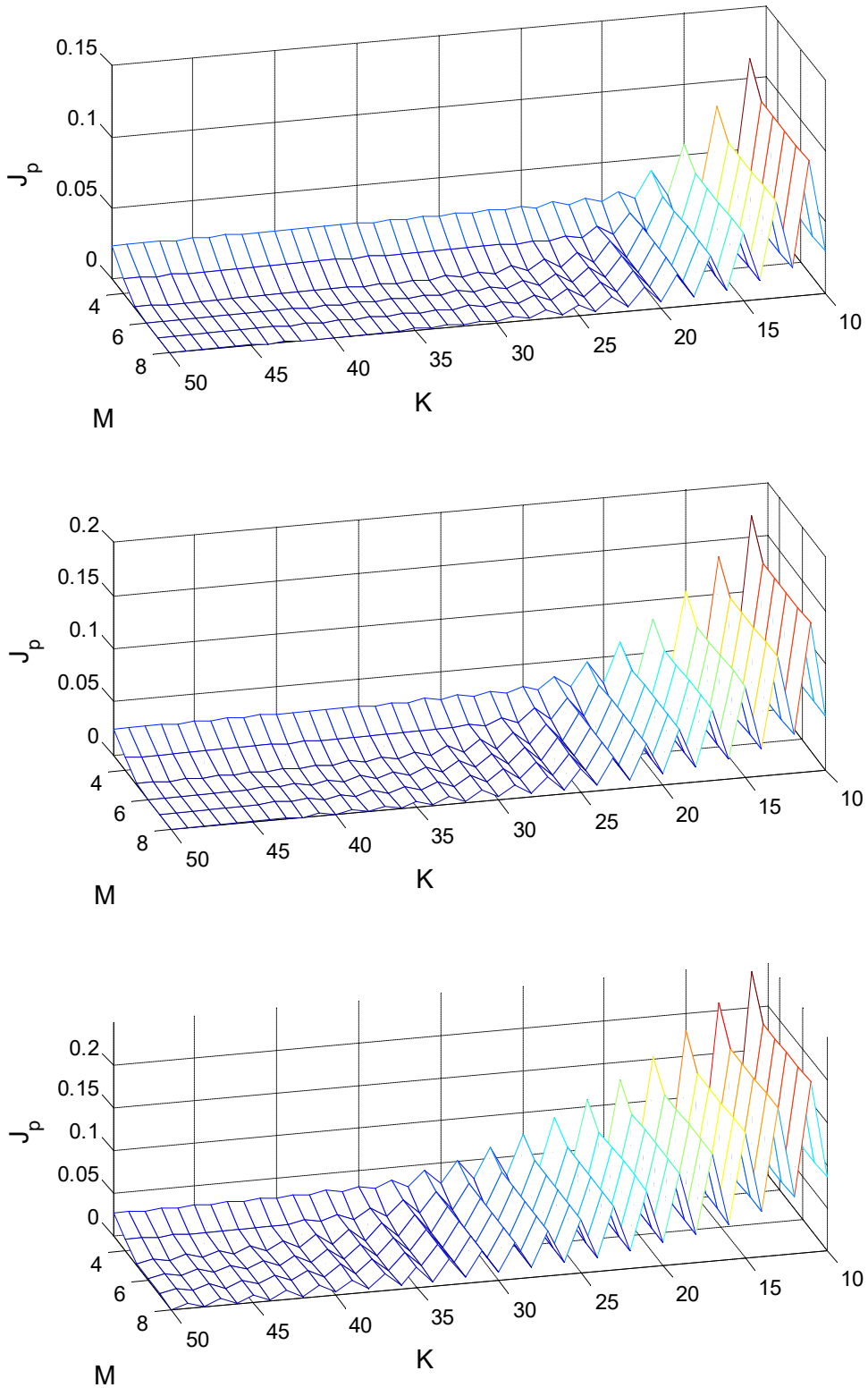


Figure 15 - Actual maximum phase error surface for (top) α=0.87, (middle) α=0.90, and (bottom) α=0.93

where $J_{a_{mk}}$ and $J_{A_{mk}}$ are the actual cost and the design guide cost, for a given M and K; and $\overline{J}_a$, and $\overline{J}_A$ are the mean of the actual and the mean of the design guide cost.

4.1.1 Results from the Numerical Study

Figure 13 plots the WLS error J_{wls} for $\alpha = \{0.87, 0.90, 0.93\}$ against M and K. Likewise, Figure 14 plots the maximum magnitude response error J_m for the same set of α, and Figure 15 plots the maximum phase response error J_p for the same set of α.

It appears from Figure 13 to Figure 15 that, due to the approximately exponential nature of the surfaces, J_{wls}, J_m, and J_p will have the following functional form

$$Ae^{-(a_1\alpha\pi K + a_2 K + a_3\alpha\pi)} + Be^{-(b_1\alpha\pi M + b_2 M + b_3\alpha\pi)} . \tag{4.2}$$

Using `cftool` to fit the results of the numerical study to the above proposed functional form, we obtain the design guides presented in the next subsections.

4.1.2 WLS Cost Design Guide

The WLS Farrow VFD filter cost is given by (2.8)

$$J_{wls} = \min_{\mathbf{h}} \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|H(\omega,\delta) - H_d(\omega,\delta)\right|^2 d\omega\, d\delta \tag{4.3}$$

and the corresponding design guide is given by

$$J_{WLS} = 0.976865e^{-(-1.048\alpha\pi K + 3.334K + 0.672\alpha\pi)} + 109.208e^{-(-0.792\alpha\pi M + 5.844M + 0.0014\alpha\pi)} . \tag{4.4}$$

Table 7 –Maximum WLS difference between J_{WLS} and actual cost J_{wls}

	M	***K***	α	***Value Difference***
Max	3	12	0.95	8.3844588×10^{-4}

Table 8 – PCC for WLS cost design guide

α	r
0.80	0.9898
0.82	0.9903
0.84	0.9966
0.86	0.9989
0.88	0.9996
0.90	0.9981
0.92	0.9987
0.94	0.9976
0.95	0.9923

The actual cost J_{wls} is compared against J_{WLS} using the maximum difference and the PCC criteria in Table 7 and Table 8.

4.1.3 Maximum Magnitude Response Error Design Guide

The maximum error in the magnitude response is given by

$$J_m = \max_{\omega,\delta} \left\| H(\omega,\delta) \right| - \left| H_d(\omega,\delta) \right\|^2 . \tag{4.5}$$

For AF WLS Farrow VFD filters, it was found the maximum magnitude response error only occurs at the band edge, i.e. $\alpha\pi$. The design guide is given by

$$J_M = 4.97257 e^{-(-0.525\alpha\pi K + 1.64K + 0.483\alpha\pi)} + 1.15\times10^{-8} e^{-(-2.625\alpha\pi M + 5.36M - 7.94\alpha\pi)} . \tag{4.6}$$

The accuracy of J_M is summarised in Table 9 and Table 10.

Table 9 – Maximum magnitude response differences between J_M and actual cost J_m

	M	***K***	α	**Value Difference**
Max	3	11	0.95	1.7969668×10^{-1}

Table 10 – PCC for maximum magnitude response error

α	r
0.80	0.9602
0.82	0.9761
0.84	0.9815
0.86	0.9768
0.88	0.9736
0.90	0.9851
0.92	0.9786
0.94	0.9716
0.95	0.9687

4.1.4 Maximum Phase Response Error Design Guide

The maximum error in the phase response is given by,

$$J_p = \max_{\omega,\delta} \left| \angle H(\omega,\delta) - \angle H_d(\omega,\delta) \right| . \tag{4.7}$$

There is an interesting behaviour in the maximum phase response error. Figure 16 shows a surface plot of the maximum phase response error for $\alpha = 0.88$. It is clear

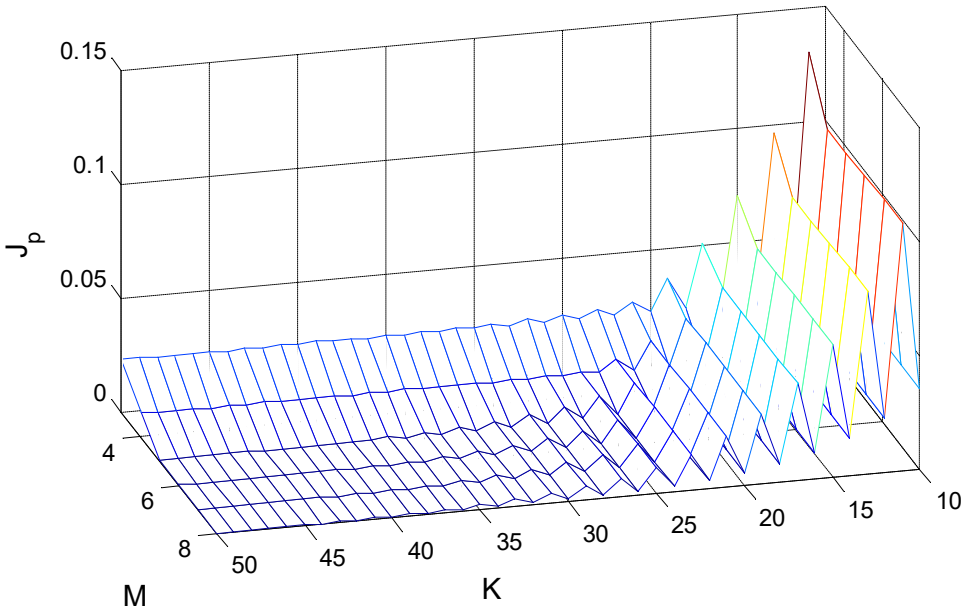


Figure 16 - Actual maximum phase error surface for α=0.88

from the plot that there is a saw-like fluctuation between even and odd *K*. Accordingly, we derived two design guides, one for even *K* and one for odd *K* as illustrated in Figure 17.

The even *K* design guide is given by

$$J_{P_e} = 0.0696865e^{-(-0.5\alpha\pi K - 1.58K - 0.46\alpha\pi)} + 342.301e^{-(0.571\alpha\pi M + 0.518M + 1.23\alpha\pi)}, \qquad (4.8)$$

while the odd *K* design guide is given by

$$J_{P_o} = 10.7599e^{-(-0.468\alpha\pi K + 1.454K + 1.02\alpha\pi)} + 12.3291e^{-(7.32\alpha\pi M + 0.86M)}. \qquad (4.9)$$

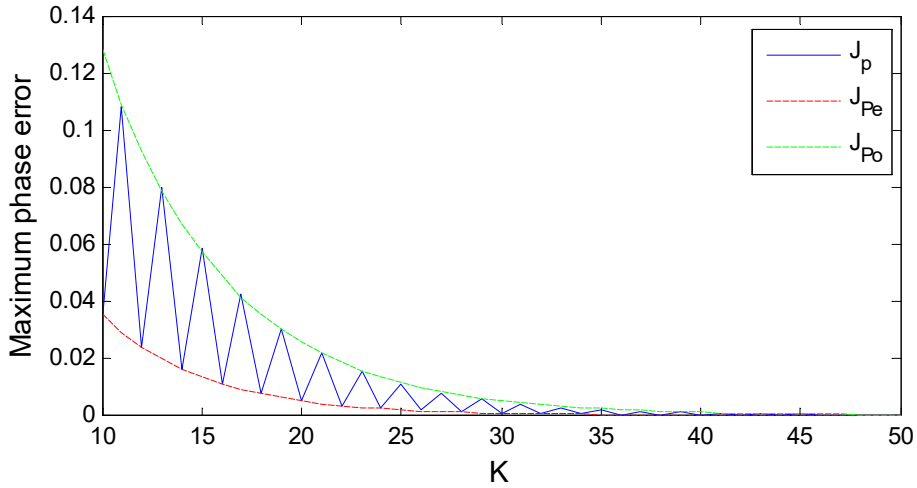


Figure 17 - Even and odd first-cut equation overlay for α=0.88 and M=7

Table 11 and Table 12 summarises the accuracy of the maximum phase response error design guide.

Table 11 –Maximum phase response differences between J_P and actual cost J_p

	M	***K***	α	**Value Difference**
Max	3	19	0.95	1.0536962×10^{-1}

Table 12 – PCC for maximum phase response error

α	r
0.80	0.97242
0.82	0.97921
0.84	0.99137
0.86	0.99658
0.88	0.98951
0.90	0.98781
0.92	0.97030
0.94	0.97152
0.95	0.98022

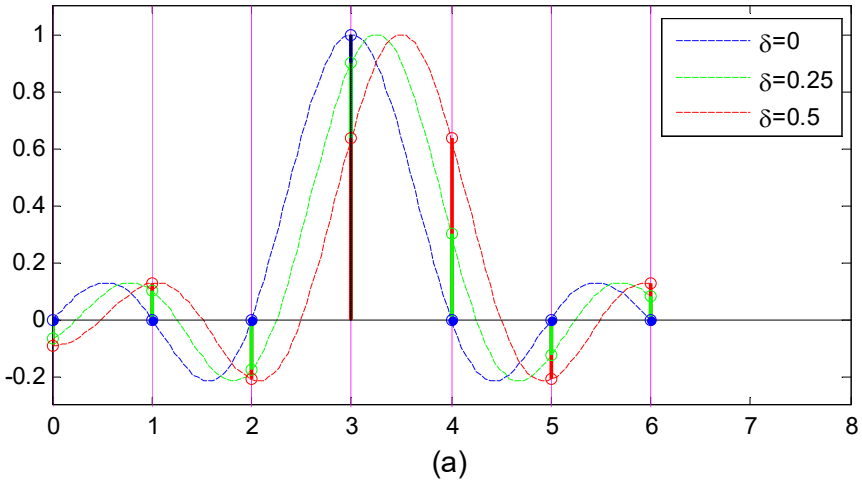
4.1.5 Noteworthy Observations

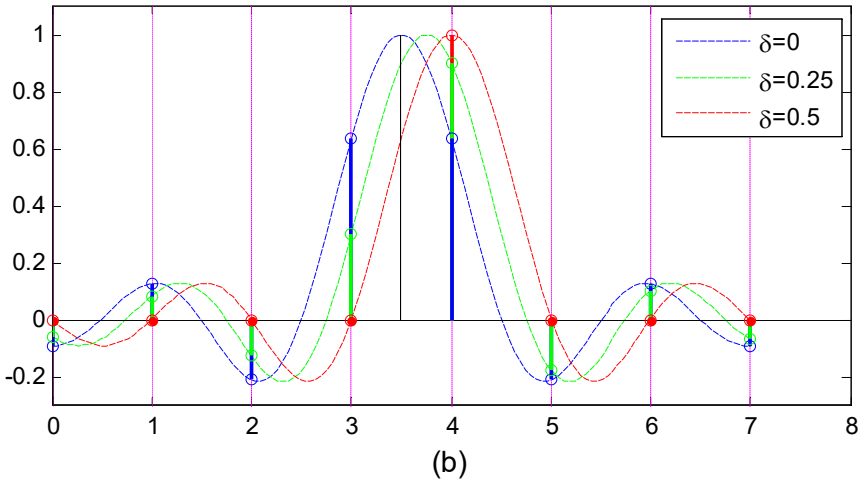
There are a few observations, in the area of maximum magnitude response and phase response error, that may help a designer to choose the right filter specifications or dimensions. These observations relate to when *K* is even or odd.

(I) In the maximum magnitude response error domain, the maximum error occurs only at the passband edge $\alpha\pi$. However, with regards to the fractional delay δ, there are two different outcomes depending on the filter length *K*. For even *K*, the maximum error occurs at $\delta = 0$, while for odd *K*, the maximum error occurs at $\delta = \pm 0.5$.

In the case of the maximum phase response error. When *K* is even, the maximum phase response error will occur at $\omega = \alpha\pi$ and $\delta = \pm 0.25$, whereas when *K* is odd, the maximum phase response error is found at $\omega = 0$ and $\delta = \pm 0.5$.

Table 13 provides a summary of these characteristics.





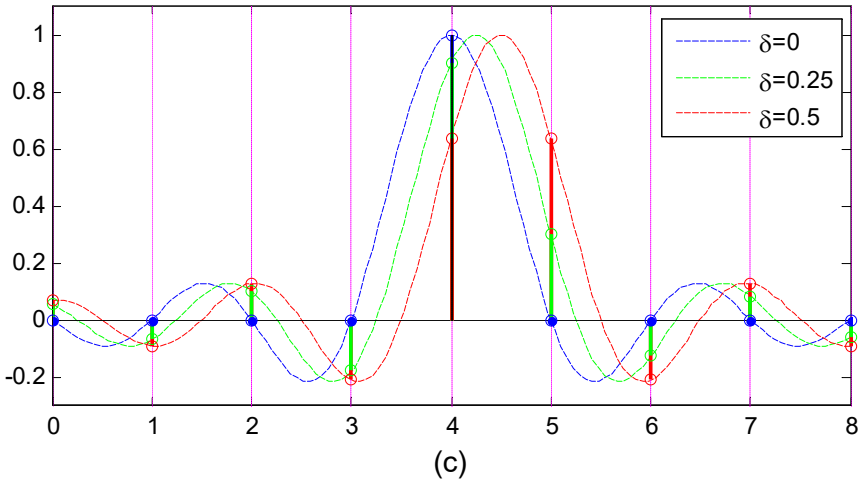


Figure 18 – Finite length impulse response of VFD AP filter when (a) K=7, (b) K=8 and (c) K=9

Table 13 – The maximum error locations for Farrow filters

Maximum Locations	Even	Odd
Magnitude error J_M	$\omega = \alpha\pi$ $\delta = 0$	$\omega = \alpha\pi$ $\delta = \pm 0.5$
Phase error J_P	$\omega = \alpha\pi$ $\delta = \pm 0.25$	$\omega = 0\pi$ $\delta = \pm 0.5$

(II) As pointed out in Figure 17, the maximum phase response error exhibits a saw-like characteristic. In particular, when the filter length K goes from even to odd, e.g. from $K = 2$ to $K = 3$, the maximum phase response error actually increases in contrast to the popular belief that it should be decreasing. The explanation to this counterintuitive pattern lies in $b(k,\delta)$ from equation (1.5),

$$b(k,\delta) = \sum_{m=0}^{M-1} h(k,m)\delta^m \tag{4.10}$$

which gives the impulse response of the VFD filter as a function of δ.

In Figure 18 we consider the design of *K*-tap all-pass (AP) filters. Figure 18(a) and Figure 18(c) show that when *K* is odd, and the desired bulk delay is set to $\frac{(K-1)}{2}$ samples, then impulse response will be ideal. However, when the desired bulk delay is increased or decreased by half-a-sample, that is to $\frac{K}{2}$ or $\frac{K}{2}-1$, then due to the truncation of the ideal impulse response, the resulting impulse response is not ideal.

In contrast, when *K* is even, for example $K = 8$ as shown in Fig 16(b), we see that for a desired bulk delay of $\frac{(K-1)}{2}$, we do not obtain the ideal impulse response. However, when the desired bulk delay is increased or decreased by half-a-sample, then we can obtain the ideal impulse response.

In short, when considering the frequency response of a VFD AP filter over the range of delay of ± 0.5 sample about the bulk delay of $\frac{(K-1)}{2}$ samples, one would expect the cost function of even length *K*-tap AP filters, as defined by (2.8), to be smaller than the neighbouring odd length $(K \pm 1)$-tap filters.

4.2 Assessment of the Design Guides

We take an AF WLS Farrow VFD filter design example from [82] where $K = 51$, $M = 8$ and $\alpha = 0.9$. This large design almost guarantees a flat response in both magnitude and phase as show in Figure 19.

It can be seen in Table 14, through substituting the design specifications into the respective equations, that the design guides give good approximations of the actual design cost.

Table 14 – Comparison between actual and design guide for a common AF filter design

	Actual design	Design guide
WLS Cost	2.1222674×10^{-9}	9.2524428×10^{-10}
Max Magnitude Error	4.6300434×10^{-4}	4.5855470×10^{-4}
Max Phase Error (odd)	5.2632254×10^{-4}	7.6397317×10^{-4}

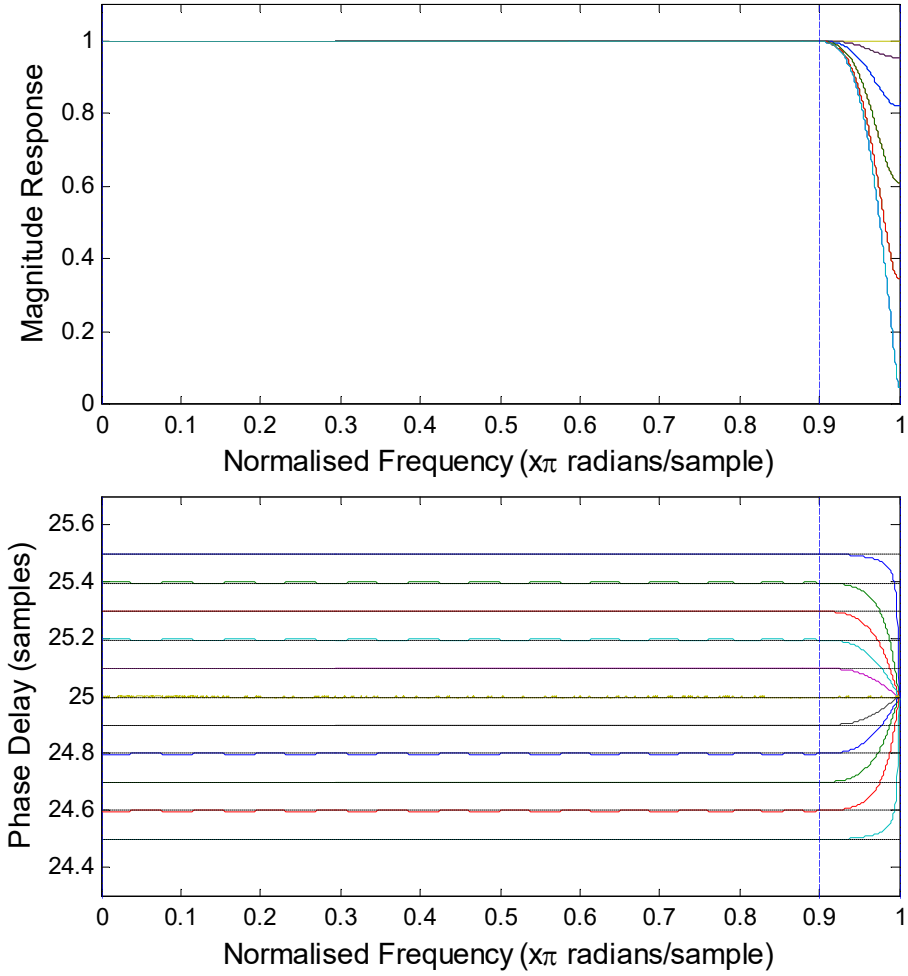


Figure 19 - Magnitude and phase response for an AF filter (M=8, K=51 and α=0.9)

4.3 Design Dimensioning

In this section, we present four examples to illustrate how the design guides (4.4), (4.6), (4.8), and (4.9) can be used to dimension the design of AF WLS Farrow VFD filters. That is, given a set of design requirements such as the tolerable WLS cost, the maximum magnitude response error and/or the maximum phase response error, how does one select *K*, *M*, and α.

Example 4.1: Suppose we have chosen $M = 5$ and $\alpha = 0.85$. To achieve a WLS cost of below 4×10^{-6}, from (4.4), we can rearrange the equation as follows

$$K \cong \left\lceil \frac{\ln\left(\frac{4\times10^{-6}-M_\alpha}{0.97686}\right)+0.672\alpha\pi}{-(-1.048\alpha\pi+3.334)} \right\rceil, \tag{4.11}$$

where $$M_\alpha = 109.208e^{-(-0.792\alpha\pi M+5.844M+0.0014\alpha\pi)}, \tag{4.12}$$

thus $$K \cong \lceil 20.2743355 \rceil = 21. \tag{4.13}$$

In Table 15, we see that $K = 21$ actually leads to an over-designed filter. By decreasing *K* to 20, we obtain a filter that just meets the design requirement.

Table 15 – Actual WLS cost with respective FIR length, K, between 18 to 21

K	J_{wls}
18	1.0373345×10^{-5}
19	6.6797861×10^{-6}
20	3.9794042×10^{-6}
21	2.6958064×10^{-6}

Example 4.2: Suppose we have chosen $K = 36$ and $\alpha = 0.94$. To achieve a maximum phase response error less than 8×10^{-3}, with *K* being even, we rearrange equation (4.8) as follows

$$M \cong \left\lceil \frac{\ln\left(\frac{8\times10^{-3}-K_\alpha}{342.301}\right)+1.23\alpha\pi}{-(0.571\alpha\pi+0.518)} \right\rceil, \tag{4.14}$$

where $$K_\alpha = 0.0696865e^{-(-0.5\alpha\pi K-1.58K-0.46\alpha\pi)}, \tag{4.15}$$

thus $$M \cong \lceil 3.9623045 \rceil = 4. \tag{4.16}$$

Comparing the recommended order with the actual maximum phase response error in Table 16, we see that an order of 5 is required.

Table 16 – Finding actual maximum phase error with varying order, M, between 3 to 6

M	J_p
3	2.7274753×10^{-2}
4	1.2928598×10^{-2}
5	7.5216660×10^{-3}
6	7.4652293×10^{-3}

Example 4.3: Suppose $M=6$, $K=40$, and the maximum magnitude response error J_m is required to be less than 0.003. To determine the cut-off frequency $\alpha\pi$, we can rearrange equation (4.6), similar to Example 4.1 and Example 4.2, or solve it graphically. As can be seen in Figure 20, J_M intersects J_m at $\alpha\approx0.9$.

Comparing J_M against $J_m=0.003$ for different cut-off frequencies, Table 17 suggests any cut-off frequency α below 0.9 will satisfy the constraint of $J_m\le0.003$.

Table 17 – Actual magnitude error by varying cut-off frequency, α, between 0.87 and 0.92

α	J_M
0.87	5.6770677×10^{-4}
0.88	9.2381978×10^{-4}
0.89	1.5852741×10^{-3}
0.90	2.8194952×10^{-3}
0.91	4.8797277×10^{-3}
0.92	9.0161390×10^{-3}

Example 4.4: Suppose one wants to find the design dimensions, *M* and *K*, given the cut-off frequency $\alpha=0.88$, maximum WLS tolerance $J_{wls}=4\times10^{-7}$ and maximum magnitude response tolerance $J_m=5\times10^{-3}$. The smallest *M* and *K* that will satisfy the design specification can be found as follows.

Rearranging (4.4) and (4.6), we obtain, respectively,

$$M_1\cong\left\lceil\frac{\ln\left(\frac{4\times10^{-7}-K_{\alpha_1}}{109.208}\right)+0.0014\alpha\pi}{-(-0.792\alpha\pi+5.844)}\right\rceil, \tag{4.17}$$

where $$K_{\alpha_1}=0.97686e^{-(-1.048\alpha\pi K+3.334K+0.672\alpha\pi)}, \tag{4.18}$$

and
$$M_2 \cong \left\lceil \frac{\ln\left(\frac{5\times10^{-3}-K_{\alpha_2}}{1.15\times10^{-8}}\right)-7.94\alpha\pi}{-(2.625\alpha\pi-5.36)} \right\rceil , \tag{4.19}$$

$$K_{\alpha_2} = 4.97257e^{-(-0.525\alpha\pi K+1.64K+0.483\alpha\pi)} . \tag{4.20}$$

Plotting (4.17) and (4.19) graphically, as shown in Figure 21, we obtain two possible solutions: $M=6$, $K=29$ and $M=6$, $K=31$.

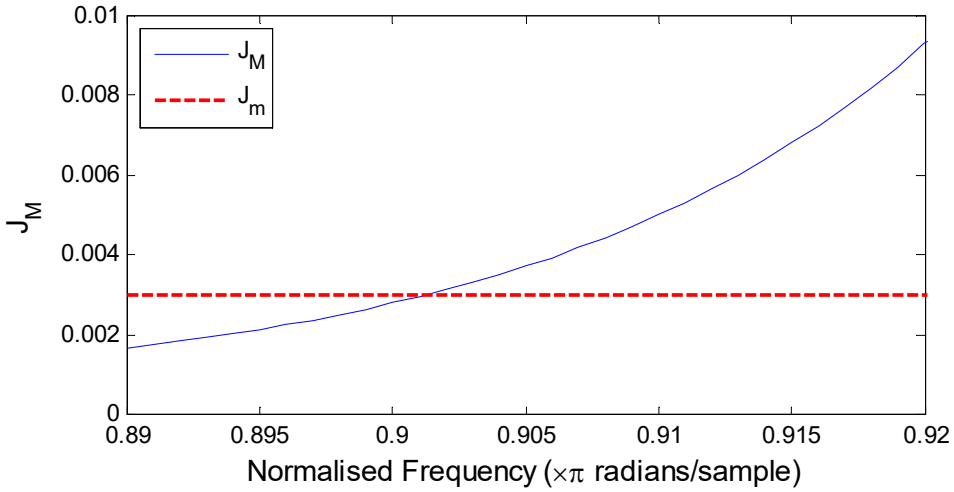


Figure 20 - Graphical analysis of the J_M function between α=0.89 and α=0.92

The actual costs J_{wls} and J_m are given in Table 18. The first solution, $M=6$

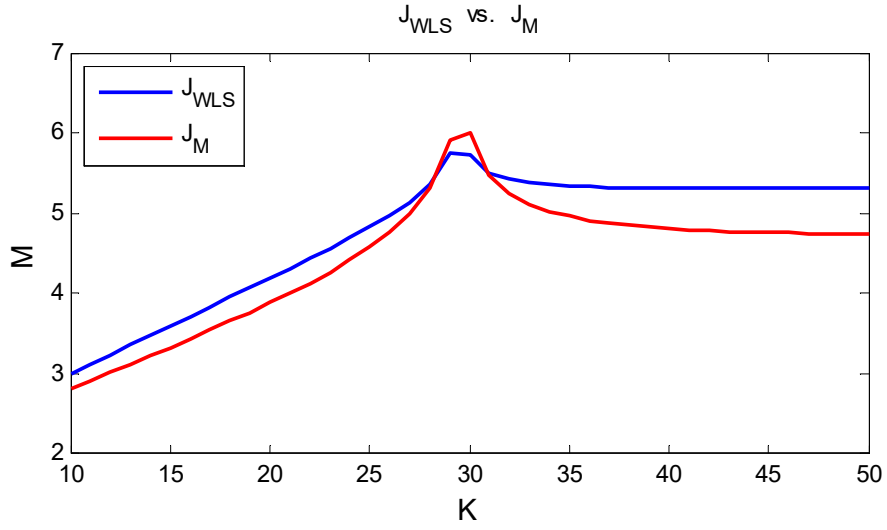


Figure 21 - Graphical analysis of solving two unknowns, M and K

and $K=29$, violates the design requirements but the second solution, $M=6$ and

$K = 31$, fits nicely. We can further tweak the design by reducing the FIR length to $K = 30$ and, as seen in Table 18, this length also satisfies the required specifications.

Table 18 – Actual design result for the two unknown variables example

K	M	J_{wls}	J_m
27	**6**	1.3395926×10^{-6}	8.0204638×10^{-3}
28		8.4460934×10^{-7}	7.1167376×10^{-3}
29		5.9209345×10^{-7}	5.4900656×10^{-3}
30		3.7622648×10^{-7}	4.9497102×10^{-3}
31		2.6582129×10^{-7}	3.7609515×10^{-3}
32		1.7121591×10^{-7}	3.4611624×10^{-3}

Through further investigation on the above design scenario we noticed that by reducing the filter order to $M = 5$, no designs that satisfy the design requirement, $J_{wls} \le 4\times10^{-7}$ and $J_m \le 5\times10^{-3}$, can be obtained. Therefore, it is safe to say that the design guide found an acceptable solution.

Furthermore, one might notice that there are two maximum phase response error equations, (4.8) for even *K* and (4.9) for odd *K*. Therefore, without knowing whether *K* should be even or odd, one will have to incorporate both equations and potentially end up solving 4 design problems.

4.4 Extrapolation of Results

The design guides are not limited to the fitted ranges, i.e. $K \in [7,50]$, $M \in [3,8]$, $\delta = [-0.5, 0.5]$, and $\alpha \in [0.80, 0.95]$. One can extrapolate the data outside the said range and still obtain a reasonable approximation of the actual cost. For example, take $M = 10$, $K = 70$ and with $\alpha = 0.93$. The comparison between the actual and the design guides are listed in Table 19 below.

Table 19 – Comparison between actual and guide results for M=10, K=70 and α=0.93

	Actual	**Guides**
WLS Cost	7.7750019×10^{-10}	7.3424586×10^{-10}
Max Magnitude Error	5.9949577×10^{-4}	7.2059379×10^{-4}
Max Phase Error	5.0781376×10^{-5}	6.3727962×10^{-5}

4.5 Design with Safety

Despite many successes at demonstrating the usefulness of the design guides, there are some limitations that should be mentioned as well. Throughout this chapter, sensible design constraints were deliberately chosen. Obviously, one can apply unrealistic or infeasible constraints on the design guides and they will still provide a solution. A useful result is that the solution itself will indicate whether or not the design specifications are infeasible. Specifically, if the solution is complex, then the design specifications are infeasible.

Example 4.5: Suppose $M = 4$, $\alpha = 0.85$, and we a require $J_{wls} \leq 2 \times 10^{-7}$. Following the method outlined in Example 4.1, we obtain $K \approx 15.7141 - 5.8669i$ which is not a real number. This indicates that the design failed. One can now decide either to increase *M*, decrease α, or increase the tolerance J_{wls}. Let us increase *M*. When $M = 5$, the recommended length is $K \approx 23.1564 - 5.8670i$ which is still complex. However, when $M = 6$, the recommended length is $K \approx \lceil 25.6172 \rceil = 26$. We can verify the actual design with $M = 6$ and $K = 26$ fits the specification, and the actual WLS cost is $J_{wls} \cong 1.61136 \times 10^{-7}$.

4.6 Closing Remark

This chapter is tailored to the design of AF WLS Farrow VFD filters within the ranges $K \in [7, 50]$, $M \in [3, 8]$, $\delta = [-0.5, 0.5]$, and $\alpha \in [0.80, 0.95]$. However, the reader can extend the idea towards any other filter designs, such as band-pass, and ranges. Using the framework described in this chapter, that is, by selecting an appropriate functional form and utilising MATLAB's `cftool` function, it is not difficult to create other design guides.

5 Weighted Minimax Design

5.1 Introduction

As discussed in Chapter 1, there exist many techniques to solve WMM design problems. These includes different classes of linear programming, quadratic programming or game theory. In this chapter, on the weighted minimax design of Farrow VFD filter, the sequential quadratic programming (SQP) method [87], as implemented in MATLAB'S `fminimax` function, is used as the reference since it is widely regarded as the state-of-the-art. The SQP methods have also been implemented in many other popular packages, including KNITRO [88], NPSOL [89], NLPQL [90], GNU Octave [91], and SciPy [92].

In this thesis, we present another technique to obtain the weighted minimax solution. It uses a series of weighted least square problem formulation to approximate the ∞-norm solution. The proposed technique is based on the Lawson's algorithm [60], where he established a link between WLS and WMM.

Specifically, our technique extends Lawson's one-dimension (frequency) algorithm to our two dimensions (frequency and delay) design problem. The significance of our technique is that we can then utilise the closed-form WLS solution, discussed in Chapter 2,

$$\mathbf{h}_c = \mathbf{Q}_c^{-1}\mathbf{p}_c \tag{5.1}$$

and

$$\mathbf{h}_s = \mathbf{Q}_s^{-1}\mathbf{p}_s \tag{5.2}$$

to achieve the best weighted minimax approximation via a sequence of carefully weighted closed-form solutions.

In addition, we introduce a new proactive homotopy convergence method that significantly decreases the optimisation time as well as provide additional features to efficiently solve the WMM Farrow VFD filter design problem.

5.2 The WMM Problem Formulation

As discussed in Section 1.3.2, the WMM VFD filter problem can be written as

$$L_\infty(\omega,\delta) = \min_{\mathbf{h}\in\mathbb{R}^{MK}} \max_{\substack{-\pi\le\omega\le\pi \\ -0.5\le\delta\le 0.5}} W(\omega,\delta)\left|H(\omega,\delta) - H_d(\omega,\delta)\right| \ . \tag{5.3}$$

Solving (5.3) requires creative methods as there are no analytical solutions. Typically, the Parks-McClellan method can be used to solve the WMM problem using Chebyshev approximation theory, but this method is applicable only to one dimension, that is, ω . In two dimensions, the Chebyshev theorem no longer applies, thus, nullifying the use of the Remez exchange algorithm and the aforementioned Parks-McClellan method.

As discussed in Section 1.3.2, there have been numerous exploration into the two-dimensional Chebyshev approximation space. However, these approaches only apply to specific optimisation problems. Up to date, there are no methods being developed for the Farrow VFD filter. In this chapter, a new two-dimensional VFD filter design technique will be presented, adapted from Lawson's one-dimension algorithm.

5.3 Review of Lawson's Algorithm

The general minimax formulation for FIR filter design consists of the minimisation over the set of coefficients $\mathbf{h}$ in the Chebyshev sense, i.e.,

$$L_\infty = \min_{\mathbf{h}} \max_{-\pi\le\omega\le\pi} \left|H(\omega) - H_d(\omega)\right|, \tag{5.4}$$

where

$$H(\omega) = \sum_{k=0}^{K-1} h(k) e^{-jk\omega} \ , \tag{5.5}$$

or by minimising

$$L_p = \left(\int_{-\pi}^{\pi} \left|H(\omega) - H_d(\omega)\right|^p \, d\omega \right)^{1/p} \ , \tag{5.6}$$

where $p \to \infty$. It can be approximated discretely by, for a sufficiently large p,

$$l_p = \sqrt[p]{\sum_i \left|H(\omega_i) - H_d(\omega_i)\right|^p} \ , \tag{5.7}$$

or since $\sqrt[p]{\bullet}$ is monotonic, it can be written as,

$$l_p = \sum_i \left|H(\omega_i) - H_d(\omega_i)\right|^p \ . \tag{5.8}$$

Minimax is one of many special cases of L_p approximation and solving the minimax problem has proved to be difficult since there is no analytical solution for L_∞. In fact, in the general L_p norm space, i.e.,

$$L_p = \min_{\mathbf{h}} \int_{-\pi}^{\pi} |H(\omega) - H_d(\omega)|^p \, d\omega \ , \tag{5.9}$$

only the *2*-norm, where $p = 2$, otherwise known as LS problem (2.8) has a closed-form solution. The LS closed-form solution, as presented in Chapter 2, can also be written as L_2.

5.3.1 Lawson's Minimax IRLS Algorithm in One Dimension

This basis of Lawson's algorithm is that there exists a weighting such that the solution of an L_p problem is also the solution of a weighted L_q problem [59, 93, 94], for arbitrary $p, q > 1$.

Lawson introduced the Iteratively Reweighted Least Squares (IRLS) technique to solve the minimax problem, in one dimension, by breaking the procedure into a series of WLS problems [60]. At every iteration, we find the weighting that will lead to a closer approximation of the known minimax solution. In other words, the algorithm seeks to find a weighting function $W^*(\omega)$ such that

$$\min_{\mathbf{h}} \int_{-\pi}^{\pi} W^*(\omega) |H(\omega) - H_d(\omega)|^2 \, d\omega \tag{5.10}$$

equals the minimax problem, that is,

$$\min_{\mathbf{h}} \int_{-\pi}^{\pi} |H(\omega) - H_d(\omega)|^\infty \, d\omega \equiv \min_{\mathbf{h}} \int_{-\pi}^{\pi} W^*(\omega) |H(\omega) - H_d(\omega)|^2 \, d\omega . \tag{5.11}$$

The solution of (5.10) can be obtained by solving the quadratic cost (2.9).

Lawson's algorithm can be summarised as follows:

1) Choose $W_0(\omega)$ arbitrarily such that $\sum_i W_0(\omega_i) = 1$, and $W_0(\omega_i) > 0$; and set $n = 0$.[1]
2) At the *n*-th iteration, solve the WLS problem to find the optimum h_n and compute a new $H_n(\omega)$.

[1] The frequency points, used by the author, are uniformly spaced in frequency between [0, π]. Although the original Lawson's thesis did not specify this.

$$\min_{\mathbf{h}} \int_{-\pi}^{\pi} W_n(\omega)\left|H_n(\omega)-H_d(\omega)\right|^2 d\omega \tag{5.12}$$

where $$H_n(\omega)=\sum_{k=0}^{K-1} h_n(k)e^{-jk\omega} \quad . \tag{5.13}$$

3) Update the weight for the next iteration as follows

$$W_{n+1}(\omega_i)=\frac{W_n(\omega_i)\left|H_n(\omega_i)-H_d(\omega_i)\right|}{\sum_i W_n(\omega_i)\left|H_n(\omega_i)-H_d(\omega_i)\right|} \quad . \tag{5.14}$$

4) Repeat step 2 and 3 until the WLS cost equals the known minimax cost.

The basic Lawson's algorithm described above is directly applicable to the design of one-dimensional digital filter. Since the basis functions are Chebyshev functions, the best Chebyshev approximation is unique and convergence is guaranteed by Lawson's algorithm [60].

The flow chart of Lawson's algorithm is presented in Appendix B1.

5.3.2 Rice and Usow's WMM IRLS Approach in One Dimension

The issue with Lawson's algorithm is that it requires prior knowledge of the end point, which is the minimax solution L_∞. On some occasions, it may be difficult to determine the L_∞ solution but a close l_∞ solution is sufficient.

In [95] Rice established a bootstrapping approach to approximate the desired L_p function by utilising lower order *p*-norm functions. Using Rice's technique, Rice and Usow built upon Lawson's algorithm and formulated an L_p approach that can be used to approximate L_∞ [61], for a sufficiently large *p*,

$$L_p = \min_{\mathbf{h}} \int_{-\pi}^{\pi} W(\omega)\left|H(\omega)-H_d(\omega)\right|^p d\omega \quad . \tag{5.15}$$

They demonstrated that one can solve *p*-norms, for *p* even, by evaluating a succession of weighted least squares problems. Starting from L_2, i.e.

$$L_2^2 = \int_{-\pi}^{\pi} W(\omega)\left|H_0(\omega)-H_d(\omega)\right|^2 d\omega \quad , \tag{5.16}$$

In the next iteration, we solve another WLS problem from the earlier solution as follows,

$$L_4^4 = \int_{-\pi}^{\pi} W_4(\omega)\left|W(\omega)\left|H_1(\omega)-H_d(\omega)\right|^2\right|^2 d\omega \quad , \tag{5.17}$$

$$L_6^6 = \int_{-\pi}^{\pi} W_6(\omega)\left|W(\omega)\left|H_2(\omega) - H_d(\omega)\right|^2\right|^3 d\omega \ , \tag{5.18}$$

$$\vdots$$

$$L_p^p = \int_{-\pi}^{\pi} W_p(\omega)\left|W(\omega)\left|H_{p/2-1}(\omega) - H_d(\omega)\right|^2\right|^{p/2} d\omega \ , \tag{5.19}$$

$$\vdots$$

where $W_p(\omega),\ p = 2,4,6,\ldots$, is defined later in (5.22). Sequentially stepping through the iteration n, where $n \in \mathbb{Z}$, until a desired $p = 2n$ is reached, we get

$$L_{2n}^{2n} = \int_{-\pi}^{\pi} W_{2n}(\omega)\left|W(\omega)\left|H_{n-1}(\omega) - H_d(\omega)\right|^2\right|^n d\omega \, . \tag{5.20}$$

The minimax approximation, l_∞ , can be achieved by choosing a p that is sufficiently large such that

$$l_p^* - l_{p-2} < \gamma \tag{5.21}$$

where γ is an acceptable tolerance for the approximation and l_p^* is the targeted p-norm where $L_p \cong L_\infty$. At l_p^* there exists a weighting, $W_p^*(\omega)$, that is the transformational bridge that links L_2 and L_p.

The Rice and Usow's method is similar to Lawson's algorithm with the exception of the weighting function. Their sequence of weighting functions are calculated as follows,

$$W_{2(n+1)}(\omega) = W_{2n}^{\alpha}(\omega)\left|H_{n-1}(\omega) - H_d(\omega)\right|^{\alpha/2\varphi} \tag{5.22}$$

where

$$\alpha = \frac{(p_n - 2)\cdot\varphi}{(p_n - 2)\cdot\varphi + 1}, \tag{5.23}$$

$p_n > 2$, and $\varphi \in \mathbb{R}$, $\varphi > 0$, is the convergence constant explained in [61].

5.3.3 Convergence Issue

Ideally the above methods should get us to the l_∞ solution. However, Lawson's algorithm has linear convergence issue with small residual weights which can lead to zero weighting and requires manual restart. Likewise, Rice and Usow's method also suffers from the same issue. Karlovitz [62] circumvents this issue by partially updating the filter coefficients in each iteration as follows

$$\hat{\mathbf{h}}_n = \mathbf{Q}_n^{-1}\mathbf{p}_n \ , \tag{5.24}$$

and using $\hat{\mathbf{h}}_n$ to calculate the incremental coefficient by

$$\mathbf{h}_n = \lambda_n \hat{\mathbf{h}}_n + (1-\lambda_n)\mathbf{h}_{n-1}, \tag{5.25}$$

where $\lambda_n \in [0,1]$. To find the optimum value of λ_n, Karlovitz proposed the use of line search in each iteration [62]. This recommendation suggests a separate optimisation problem nested within the IRLS algorithm.

Whilst the line search strategy is sound, the nested optimisation is computationally inefficient. Despite this drawback, Karlovitz had demonstrated that partially updating the coefficients can eliminate the zero-weighting issues in Lawson's, and Rice and Usow's IRLS algorithms. Kahng recognised the inefficiency of the Karlovitz method and applied Newton-Raphson's method to the line search optimisation and proposed a closed-form solution for λ_n [63] given by

$$\lambda_n = \frac{1}{p_n - 1} \tag{5.26}$$

resulting in reformulating (5.25) into

$$\mathbf{h}_n = \left(\tfrac{1}{p_n-1}\right)\hat{\mathbf{h}}_n + \left(\tfrac{p_n-2}{p_n-1}\right)\mathbf{h}_{n-1} \ . \tag{5.27}$$

One benefit of adopting Newton's method to find λ_n is that the algorithm converges at a quadratic rate. However, it has been noted that Newton-Raphson based algorithms are not guaranteed to converge globally; unless during the search the solution comes close enough to the optimum solution and within their radius of convergence[2] [96].

5.3.4 Active Homotopy Convergence

Burrus et al [65] revisited Kahng's original work and made several improvements on the convergence rate and the method's stability. The biggest change that Burrus proposed was to combine Lawson, Rice and Usow's algorithms with an alternative convergence scheme. Burrus proposed a homotopy-like solution in comparison to the original stepwise solution. The intention was to break away from the original belief that the initial solution, i.e., the un-weighted L_2, will be structurally similar to the L_p solution. This may not be true at all. Yet it is known that there exists a continuity of

[2] This limitation was also observed in the two-dimension Farrow design, thus the need to adopt a more resilient and adaptive convergence scheme.

solutions from L_p to L_q where $2 < p < q < \infty$. By slowly increasing p from one iteration to the next, one hopes to traverse the continuum of solutions from L_p to L_q. Also, note that p and q are not necessary integer powers of 2; in fact, they can be real. In other words, Burrus' method eliminated Rice and Usow's limitation of multiple-of-two p increments.

Burrus' homotopy technique enables the acceleration of the initial convergence to solve the Chebyshev problem through setting p_n at the n-th iteration with

$$p_n = \min(p_D, \tau \times p_{n-1}) \tag{5.28}$$

where $\tau \in [1,2]$ is a convergence parameter and p_D corresponds to the desired p_n^{th} solution. Note that in an L_∞ solution $p_D = \infty$. However, numerical studies show that for a sufficiently large p_D, e.g. $p_D = 400$, satisfactory approximation can be obtained as explained in equation (5.21).

The next natural step is to refine the convergence parameter, τ. Vargas, in his

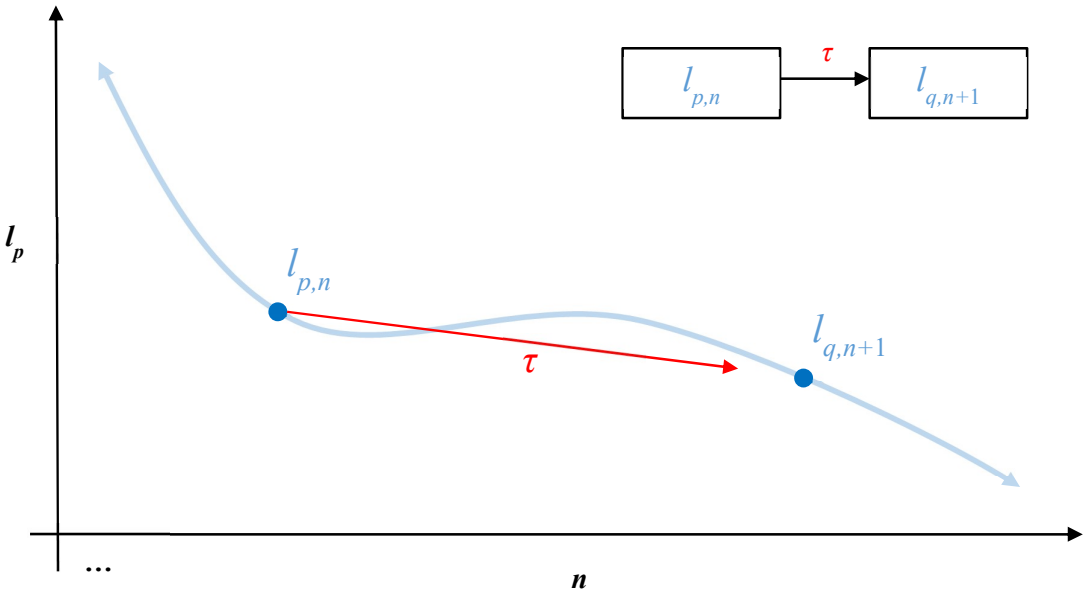


Figure 22 – Homotopy path from $l_{p,n}$ to $l_{q,n+1}$ with τ convergence rate

thesis [74], suggested the use of an adaptive algorithm that changes the value of τ such that the l_p cost (5.32) always decreases. His idea is to create a temporary new set of weights and filter coefficients that will be updated if its subsequent $l_{q,n+1}$ cost, where $q = \tau \times p_n$, is smaller than the current cost $l_{p,n}$ as illustrated in Figure 22.

If the current cost $l_{p,n}$ is smaller than the subsequent cost $l_{q,n+1}$, the algorithm will attempt to split the search direction by modifying τ into a high τ_H and a low τ_L,

$$\tau_L = \tau \times (1-\sigma) \quad \text{and} \quad \tau_H = \tau \times (1+\sigma) \tag{5.29}$$

where σ is the updating variable. The goal is to find a smaller $l^H_{q,n+1}$ or $l^L_{q,n+1}$ using τ_H or τ_L respectively in (5.28). From the two convergence parameters, temporary new set of weights and filter coefficients are then calculated. The smaller l_q cost of the two, $l^L_{q,n+1}$ or $l^H_{q,n+1}$, will be compared against the non-updated current cost l_p. If the new convergence τ produces a smaller cost, then the homotopy will select the smaller cost and update accordingly. Otherwise another split-search of τ will be initiated. This tree-like search will continue indefinitely until a smaller l_q cost is found[3]. Vargas claims that his tree search will find a lower cost if the search is deep enough. But in our extensive simulation studies, we observed a lower cost l_q may not exists in the τ homotopy range. In other word, Vargas' method is resource intensive since its expanding tree-search algorithm is a recursive loop, and moreover, it may not even converge. The expanding tree-branching is illustrated in Figure 23.

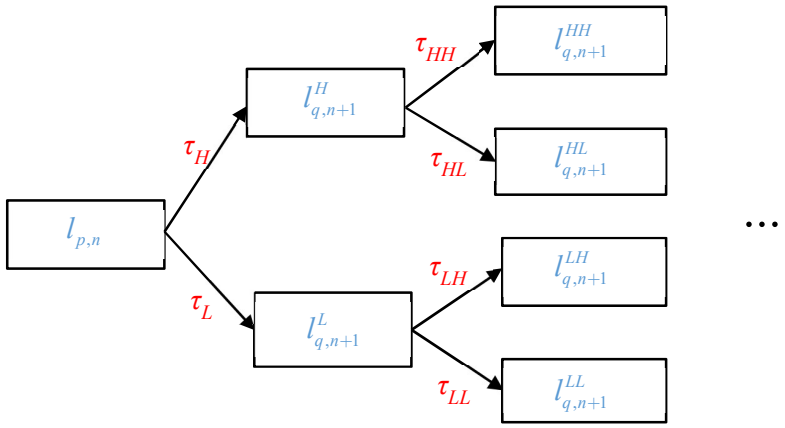


Figure 23 – Vargas adaptive τ algorithm

The algorithm above refines the updating variable τ which produces a different q and consequently l_q cost. The variable τ is to be adjusted as many times as necessary without modifying the last effective filter coefficients, the weights or p_n. If a τ value

[3] It was found in rare occasions that Vargas' algorithm can loop more than 50+ branches deep without a lower cost in sight. When this occurs, the algorithm can be resource intensive and time consuming.

of lower cost is found, Vargas recommends the user to continue with this new value until another update in τ is required.

As mentioned, Vargas' homotopy convergence scheme will occasionally fail to converge[4] and requires auto-restart. When a fail happens, we recommend that we take one step back to an earlier iteration $l_{p,n-1}$ and perform the Vargas' tree search again. This is to ensure that the current $l_{p,n}$ is the lowest cost in the n^{th} iteration. If during the tree search a lower cost solution was found, then update the coefficients, weights and p to the latest. Otherwise if it fails again, the algorithm will terminate as a potential minimum cost is found at $l_{p,n}$. An example of this multiple restart case is presented in Section 5.4.4.

5.4 Extension of the IRLS Algorithm for Farrow VFD Filters

In Section 5.3, a review of Burrus' method is presented. Burrus uses a *reactive* approach to determine the convergence parameter τ in each iteration. See equation (5.29). In this thesis, we propose a *proactive* approach to increase the convergence speed.

5.4.1 Extension of Rice, Usow, and Lawson's Algorithm to 2-D

Lawson, and Rice and Usow proposed a method that uses a sequence of WLS problems to approximate the WMM problem in one dimension. In order to utilise this method to the Farrow VFD filter design problem, it is necessary to extend their algorithms to the two-dimensional space, that is, in ω and δ.

The two-dimensional WMM VFD filter design problem (5.3) can be expressed by

$$L_p = \int_{-\delta_{max}}^{\delta_{max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left|\varepsilon(\omega,\delta)\right|^p \, d\omega \, d\delta \tag{5.30}$$

where $$\varepsilon(\omega,\delta) = H(\omega,\delta) - H_d(\omega,\delta) \ , \tag{5.31}$$

and approximated discretely by

$$l_p = \sum_i \sum_j W(\omega_i,\delta_j)\left|\varepsilon(\omega_i,\delta_j)\right|^p \ , \tag{5.32}$$

[4] A *fail* is defined as the inability to find a lower cost after a specific number of tree-branch searches.

where (ω_i, δ_j), $i, j \in \mathbb{Z}$, denotes the set of frequencies and delays in a two-dimension plane. For example,

$$\omega_i = [-\pi \quad -\pi + \Delta\omega \quad -\pi + 2\Delta\omega \quad \cdots \quad \pi], \tag{5.33}$$

and

$$\delta_j = [-\delta_m \quad -\delta_m + \Delta\delta \quad -\delta_m + 2\Delta\delta \quad \cdots \quad \delta_m]. \tag{5.34}$$

In [64, 65], Fletcher et al and Burrus et al indicate how an l_2 norm problem of two variables can be solved using a technique similar to that for solving an l_2 norm problem of one variable. In the context of the two-dimension Farrow VFD filter design problem, the solution method will be as follows.

1) Initialise the weight $W_0(\omega, \delta)$.
2) At the *n*-th iteration, solve for $\mathbf{h}_n$ given by (2.113) and (2.114), i.e.

$$\mathbf{h}_{c,n} = \mathbf{Q}_{c,n}^{-1}\mathbf{p}_{c,n} \quad \text{and} \quad \mathbf{h}_{s,n} = \mathbf{Q}_{s,n}^{-1}\mathbf{p}_{s,n} \tag{5.35}$$

   where the weight is $W_{n-1}(\omega, \delta)$.

3) Calculate H_n with $\mathbf{h}_{c,n}$ and $\mathbf{h}_{s,n}$ (2.91)
4) Update the weights with

$$W_{2(n+1)}(\omega, \delta) = W_{2n}^{\alpha}(\omega, \delta)\left|\varepsilon_{n-1}(\omega, \delta)\right|^{\alpha/2\varphi} \tag{5.36}$$

   where

$$\varepsilon_{n-1}(\omega, \delta) = H_{n-1}(\omega, \delta) - H_d(\omega, \delta) \tag{5.37}$$

   $p > 2$, and α is given by (5.23).

5) Repeat the last three steps, 2 to 4, until convergence is reached.

5.4.2 Extension of Burrus and Vargas' Homotopy Convergence Methods

In Section 5.3.4, we introduced Burrus' and Vargas' adaptive convergence methods. By using the homotopy approach, they achieved an accelerated convergence. But there is room for improvement with their recommendations. The difficulty with Burrus' selection of τ is that it, and hence the convergence rate, is static. Accordingly, the solution is susceptible to wander off the optimum path under certain parameters, see examples in [74]. To overcome this difficulty, Vargas proposed a variable τ which is adjusted when certain condition triggers it. That is, it is a *reactive* scheme.

In this thesis, a *proactive* adaptive technique is proposed. The technique works by re-evaluating τ in each iteration which not only guarantees rapid convergence but also take a necessary step towards the actual minimax solution. In particular, rather than the expanding tree-search of Vargas' method (5.29), shown in Figure 23, the proposed proactive method set τ which is required to satisfy $\tau \in [1,2]$, see (5.28), as follows.

$$\tau = \frac{4(\Delta J(\mathbf{h}))}{\sqrt[r]{\left(1+(\Delta J(\mathbf{h}))^{r}\right)}} + 1 \tag{5.38}$$

where $$\Delta J(\mathbf{h}) = (J(\mathbf{h}_{n-1}) - J(\mathbf{h}_{n}))/J(\mathbf{h}_{n-1}) \tag{5.39}$$

and $r > 4$, $r \in \mathbb{R}$, controls the slope of the sigmoid function (5.38). The reason for selecting the sigmoid function is because of its edge tapering feature which allows better control of the convergence rate τ.

The proactive adaptive convergence is achieved by acknowledging that in each iteration, p is affected by the convergence rate τ (5.28). Thus, τ can be interpreted as a *homotopy path confidence scale* between 1 to 2, where 1 is pessimistic and 2 is optimistic about the next point on the homotopy path. Therefore, when the system is optimistic one can increase the speed of decent, conversely when the system is pessimistic about the next step it can take steps back to the earlier confident points on the homotopy path. This means at the *n*-th iteration, the algorithm determines where the next point lies on the homotopy trail.

In summary, there are two possible outcomes with our proposed method (5.38):

- Positive $\Delta J(\mathbf{h})$ adjusts the homotopy convergence rate τ towards higher *p*-norm. A large positive $\Delta J(\mathbf{h})$ indicates a large step forward, while a small positive $\Delta J(\mathbf{h})$ indicates a small step meaning slower descent.
- Negative $\Delta J(\mathbf{h})$ means the algorithm is off-track and need to return to the homotopy path. The sigmoid function will return the calculation to an earlier point on the homotopy path and try again. A large negative $\Delta J(\mathbf{h})$ will result in a big step back, while a small negative $\Delta J(\mathbf{h})$ will take a small step back.

A comparison between our proposed technique and Vargas' method is discussed in Section 5.4.4. In particular, we observed (5.38) would lead to convergence where

others fail. In addition, our proposed technique yielded faster convergence compared to the Vargas' technique. Furthermore, our proactive adaptive IRLS design method eliminates the *fail and restart* issue. Also, the sigmoid function enables the algorithm to take the necessary precaution for the descent and not risk diverting too far away from the homotopy path.

5.4.3 Summary of the Proposed IRLS Farrow Design Method

In summary, the combined steps of our adaptive IRLS algorithm applied to the WMM design of Farrow VFD filters are given below.

1) Set the initial weighting $W_0(\omega,\delta)$, homotopy $p_0 = 2$, and $\tau_0 = 1$.
2) Solve for the initial optimum coefficients $\mathbf{h}_0$ via (2.113) and (2.114).
3) Find the resulting error $\varepsilon_0(\omega,\delta)$ via (2.102). Update p by

$$p_1 = min\{p_D, \tau_0 \times p_0\} \ . \tag{5.40}$$

4) Update the new weighting by

$$W_1(\omega,\delta) = |\varepsilon_0(\omega,\delta)|^{p_1-2} \quad , \tag{5.41}$$

where $$\varepsilon_0(\omega,\delta) = H_0(\omega,\delta) - H_d(\omega,\delta) \ . \tag{5.42}$$

5) At each *n*-th iteration, the following four steps are required:
   I. Solve for the new interim coefficient, $\hat{\mathbf{h}}_n$ via (2.113) and (2.114) with the weighting $W_n(\omega,\delta)$.
   II. Update the new coefficient $\mathbf{h}_n$ with

$$\mathbf{h}_{c,n} = \left(\tfrac{1}{p_n-1}\right)\hat{\mathbf{h}}_{c,n} + \left(\tfrac{p_n-2}{p_n-1}\right)\mathbf{h}_{c,n-1} \ , \tag{5.43}$$

   and

$$\mathbf{h}_{s,n} = \left(\tfrac{1}{p_n-1}\right)\hat{\mathbf{h}}_{s,n} + \left(\tfrac{p_n-2}{p_n-1}\right)\mathbf{h}_{s,n-1} \ . \tag{5.44}$$

   III. Update the next *p*-value, i.e. p_{n+1}, via (5.28) using our proactive τ (5.38).
   IV. Calculate the new error $\varepsilon_n(\omega,\delta)$ via (2.102) and the new weighting with

$$W_{n+1}(\omega,\delta) = |\varepsilon_n(\omega,\delta)|^{p_{n+1}-2} . \tag{5.45}$$

6) Iterate step 5 until a desired convergence is reached.

The flowchart of the proposed adaptive IRLS algorithm is shown in Appendix B2.

5.4.4 A Numerical Study

Here we present some simulation results comparing the two methods: Vargas' reactive homotopy and our proposed proactive homotopy.

Example 5.1

Consider the following Almost-Flat (AF) filter specifications, $M = 5$, $K = 20$ and $\alpha = 0.9$. The proposed method achieved convergence shortly after 8 iterations, as seen in Figure 24, whereas Vargas' method reached a similar convergence after 291 iterations with multiple restarts, as seen in Figure 25. The exiting conditions for Vargas' method are (i) if the optimum l_p^* cost is less than or equal to our proposed l_p cost and (ii) the number of iterations reached 400.

In Figure 24, we can see that the time spent in each iteration by our proposed proactive method and Vargas' reactive method is similar in the first 8 iterations. In the same graph the maximum error of our proposed method started its steep descent immediately and reached the exiting condition on the 8th iteration. In contrast, Vargas' algorithm showed a slower descent. Examining the overall Vargas' optimisation progression in Figure 25, after 291 iterations, we see that Vargas' algorithm has restarted 3 times at iteration 11, 37, and 61 as indicated by the green sharp spikes. Also in Figure 25 we can see the time spent per iteration spiked during the expanded tree-

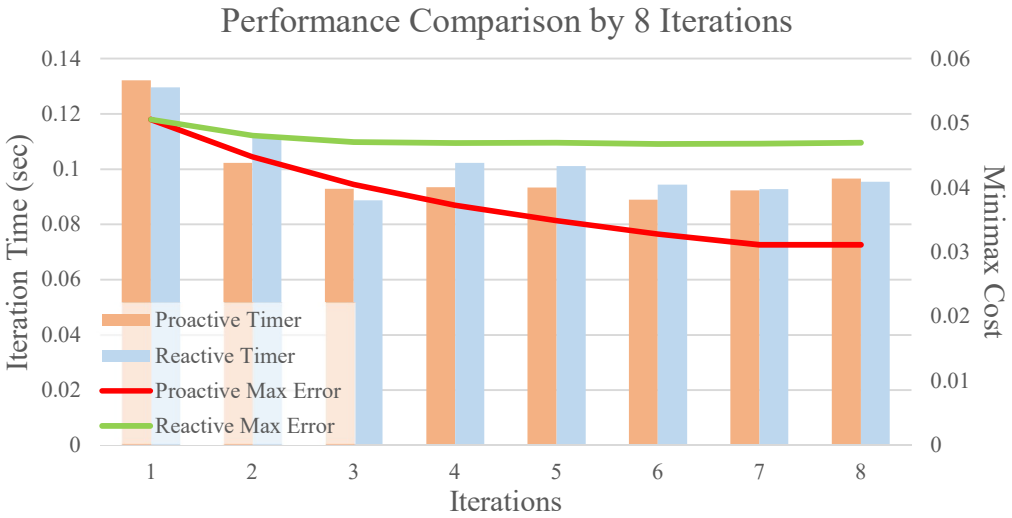


Figure 24 – Comparing Vargas' Method and Proposed Adaptive Method in 8 Iterations.

search stages. For Vargas' algorithm, the final convergence only took form at the last 2 iterations. Overall, our proposed method can achieve faster convergence as well as higher accuracy as summarised in the table below.

In the next section, we will introduce additional features that make the proactive IRLS more robust and accurate.

Table 20 – Summary of convergence performance comparison between our proposed and Vargas' methods

	Proposed Method	***Vargas' Method*** [74]
Iterations	8	291
WLS Cost	1.60017096×10^{-2}	1.52831944×10^{-2}
WMM Cost	2.11115855×10^{-2}	2.33101750×10^{-2}
Maximum Phase Error	4.14764534×10^{-3}	4.79489006×10^{-3}

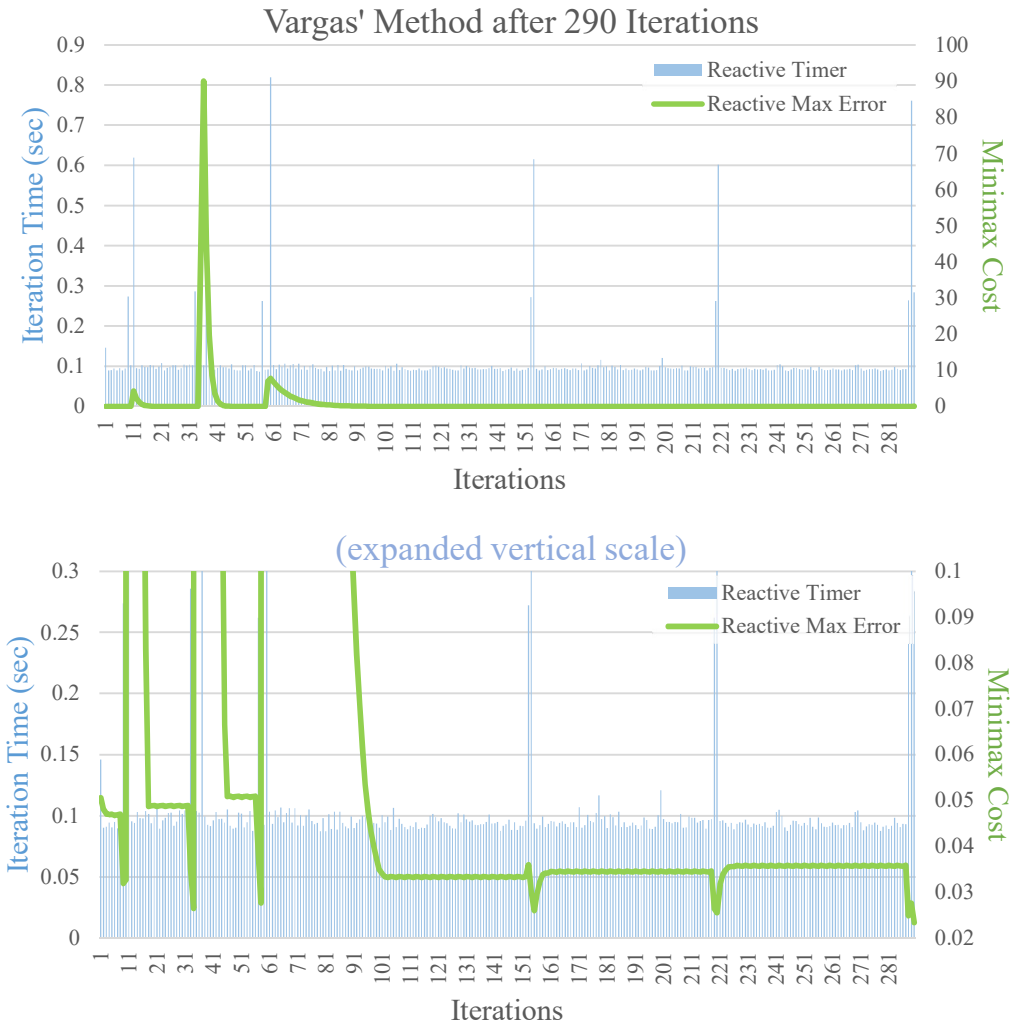


Figure 25 – Vargas' IRLS Performance after 290 Iterations

5.5 Further Enhancements on the Proactive IRLS Algorithm

In this section, we describe three enhancements to the proposed proactive IRLS algorithm. They are

i) a more appropriate method to calculate the error cost in each iteration,

ii) the inclusion of exiting strategies, and

iii) the introduction of multiple sets of frequency points to enhance convergence.

5.5.1 Two-Dimension Cost

A recurring theme in most papers on the IRLS cost function, in both one dimension and two dimensions, is the way the iterative l_p costs are calculated.

In one dimension the general *p*-norm cost equation is given by,

$$l_p = \sum_i W(\omega_i)\left|H(\omega_i) - H_d(\omega_i)\right|^p, \tag{5.46}$$

and similarly, in Farrow's two-dimension cost equation (in frequency and delay),

$$l_p = \sum_i \sum_j W(\omega_i, \delta_j)\left|\varepsilon(\omega_i, \delta_j)\right|^p. \tag{5.47}$$

The one-dimension cost (5.46) is relatively easy to implement in MATLAB through the function `norm`. However, when it comes to the two-dimension cost (5.47) there are no MATLAB tools available for *p*-norm where $p > 2$. Many authors overcome this limitation by representing $W(\omega_i, \delta_j)\left|\varepsilon(\omega_i, \delta_j)\right|^p$ as a matrix and then concatenating the matrix into a vector and calling `norm` to compute (5.47). Whilst this may adequately serve most applications, however, it is not necessary the most appropriate method to indicate that cost has decreased from one iteration to the next. Specifically, the concatenated vector method computes the ensemble average in both frequency ω_i and delay δ_j. However, in the design of VFD Farrow filter, there is an emphasis on the fractional delays δ_j. Therefore, rather than the ensemble average, we evaluate the norm of each delay δ_j across all frequency ω_i as follows

$$\mathbf{l}_{p_n} = \begin{bmatrix} \sum_i \left|\varepsilon_n(\omega_i, -\delta_m)\right|^{P_n} \\ \sum_i \left|\varepsilon_n(\omega_i, -\delta_m + \Delta\delta)\right|^{P_n} \\ \sum_i \left|\varepsilon_n(\omega_i, -\delta_m + 2\Delta\delta)\right|^{P_n} \\ \vdots \\ \sum_i \left|\varepsilon_n(\omega_i, \delta_m)\right|^{P_n} \end{bmatrix}, \tag{5.48}$$

where the error functions are defined by,

$$\varepsilon_n(\omega_i, \delta_j) = H_n(\omega_i, \delta_j) - H_d(\omega_i, \delta_j) , \tag{5.49}$$

$$H_n(\omega, \delta) = \sum_{k=0}^{K-1} \left[\sum_{m=0}^{M-1} h_n(k,m) \delta^m \right] e^{-jk\omega} , \tag{5.50}$$

and $h_n(k,m)$ is given by (5.43) and (5.44). We then compare the error cost function vector $\mathbf{l}_{p_n}$ against the error cost function vector from the previous iteration $\mathbf{l}_{p_{n-1}}$. To achieve the VFD Farrow filter minimax solution, we require $\mathbf{l}_{p_n} < \mathbf{l}_{p_{n-1}}$ for all fractional delays δ_j.

With reference to *Example 5.1*, it can be seen in Table 21 that the traditional concatenated l_p cost produces less accurate results, whereas the new proposed two-dimension cost of (5.48) yields a better WMM solution.

Table 21 – Comparison of traditional concatenated cost calculation vs new two-dimension cost calculation

	New 2-D method	***Concatenated Cost***
Iterations	8	2
WLS Cost	1.60017096×10^{-2}	9.39174885×10^{-3}
WMM Cost	2.11115855×10^{-2}	5.05579224×10^{-2}
Maximum Phase Error	4.14764534×10^{-3}	8.28369071×10^{-3}

Note that, in this example the number of iterations do not indicate IRLS has achieved convergence. Rather it has simply reached the exiting conditions as explained in Section 5.5.2.1.

The magnitude responses are compared in Figure 26, where the top graph is obtained from the new two-dimension cost and the bottom graph is obtained from the traditional concatenated cost. As can be seen from Figure 26 and Table 21, the

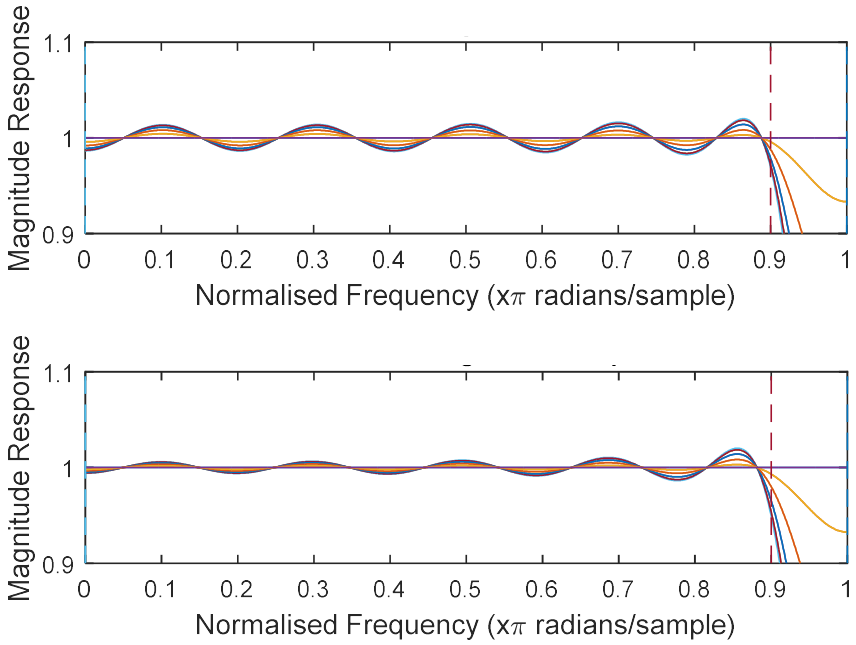


Figure 26 – 2-D Error Calculation Comparison. (top) new 2-D cost (bottom) concatenated cost

proposed two-dimension cost achieved better distribution of error compared to the traditional concatenated cost which still contains large error at $\alpha\pi$.

5.5.2 Exiting Strategies in the IRLS Algorithm

In any iterative environment, there must be some mechanism for it to exit. In our design, we proposed three methods for exiting the IRLS algorithm. They are the maximum weight tolerance, the *p*-approximation limit, and the loop iteration count. These methods are illustrated by means of the following example.

Example 5.2

Suppose the VFD filter is an AF filter with $M=5$, $K=18$ and $\alpha=0.93$ with frequency ω and delay δ uniformly sampled at a step size of $\pi/1800$ and $1/11$ respectively.

5.5.2.1 Maximum Weight Tolerance

In Chapter 2 Section 2.3.1, we saw the numerical issue with evaluating $\mathbf{Q}$ stems from the poorly conditioned matrix $\mathbf{Q}_\delta$. A similar issue can be seen in the IRLS algorithm, except now it is caused by the re-weighted weighting function $W_n(\omega,\delta)$ in (5.30). In the evaluation of the re-weighted weighting function (5.41), it is clear that

as l_p approaches L_∞ the maximum weighting should approach zero[5]. Due to the weighting being close to zero, the matrices $\mathbf{Q}_c$ (2.107) and $\mathbf{Q}_s$ (2.108) are poorly conditioned. This means the inverse of these two matrices will not be accurate, leading to inaccurate $\mathbf{h}_c$ (2.113) and $\mathbf{h}_s$ (2.114).

One method that was employed in our proposed adaptive IRLS algorithm is to ensure the maximum of the weight $W(\omega_i, \delta_j)$, i.e. $\max_{i,j} W(\omega_i, \delta_j)$, does not drop below a certain threshold. This will prevent $\mathbf{Q}_c$ and $\mathbf{Q}_s$ in the re-weighted LS from reaching a high condition number. In the case of *Example 5.2*, the weighting function tolerance limit is set to 1×10^{-17}. This corresponds to requiring the condition number of $\mathbf{Q}_c$ and $\mathbf{Q}_s$ to not exceed 1×10^{5}.

```
max(max(w_e + w_o)) >= 1e-17
```

Figure 27 plots $\max_{i,j} W(\omega_i, \delta_j)$ against IRLS iterations for *Example 5.2*. The IRLS algorithm satisfied the weighting function tolerance at the 26th iteration. As can be seen on the graph, the steepness of the maximum weight tapers off around the 12th iteration. The algorithm continues to refine the adaptive homotopy path until the maximum weight satisfies the weighting function tolerance.

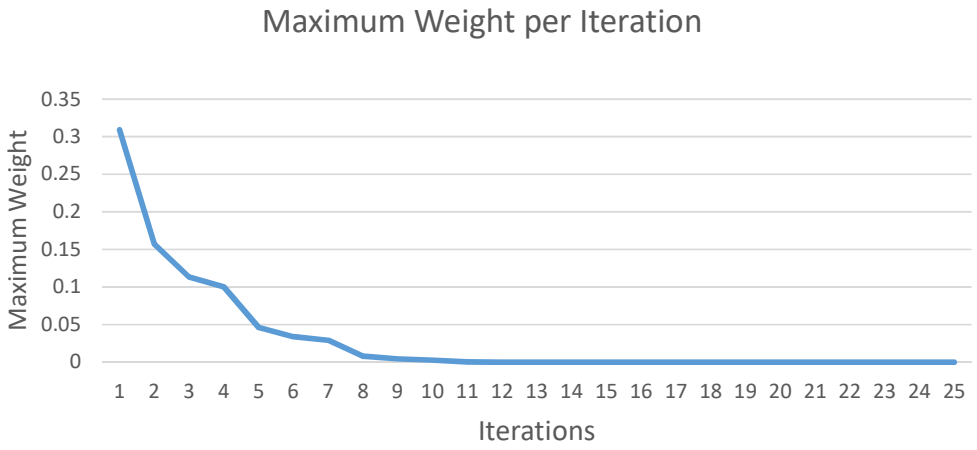


Figure 27 – Maximum Weight per Iteration

5.5.2.2 p-Approximation Limit

The IRLS algorithm solves the Chebyshev or L_∞ problem by approximating it as an l_p problem where p is sufficiently large.

[5] Similar weighting issue in Lawson's, and Rice and Usow's algorithm as discussed in Section 5.3

```
if (p >= 400)                % conclude loop
      …
end
```

The selection of the maximum p value depends on the users' preference. In *Example 5.2*, we set p to 400 since $p = 400$ will provide us with sufficient accuracy as the relative improvement of the maximum magnitude is 1.3×10^{-14} for p between 380 and 400[6]. In the case of WMM Farrow VFD filter designs of similar dimension, we recommend setting p to 500.

As it transpired in *Example 5.2*, the weighting function tolerance is satisfied when $p = 22$. The corresponding Chebyshev approximation error is $\varepsilon_{MM} \approx 0.096088$ (see Figure 28). For comparison, using MATLAB's function `fminimax`, an approximation error of $\varepsilon_{MM} \approx 0.0867795$ was achieved (See Table 22).

Table 22 – Combined 'exiting strategy' example results

	Proposed method with combined exiting strategy	***Minimax via MATLAB***
Iterations	26	318
WLS Cost	$5.17622484 \times 10^{-2}$	$8.23180391 \times 10^{-2}$
WMM Cost	$9.60881402 \times 10^{-2}$	$8.67795464 \times 10^{-2}$
Maximum Phase Error	$1.60846292 \times 10^{-2}$	$3.23538418 \times 10^{-2}$
Timer	4.3815	532.7223

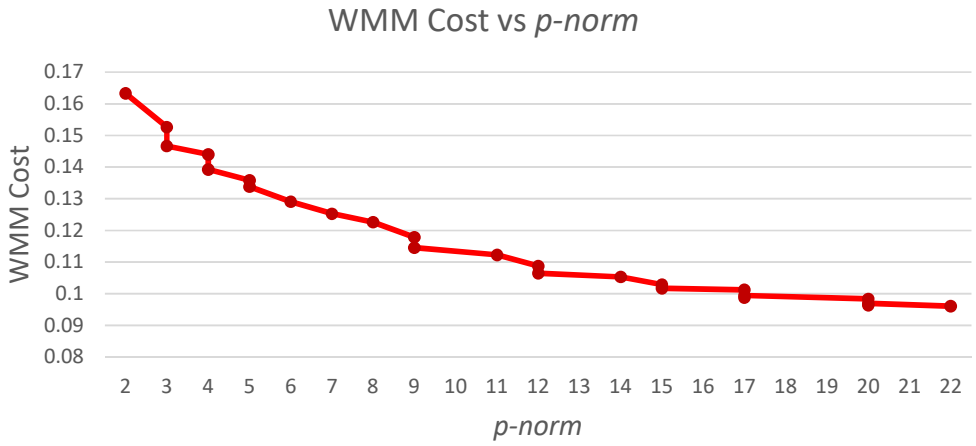


Figure 28 – WMM Cost vs p-Norm

[6] In fact, any further reduction in the maximum magnitude response error will be limited by the finite precision of the computer.

5.5.2.3 *Loop Iteration*

In general, the IRLS algorithm may continue to run without ever exiting through the weighting function tolerance or the *p*-approximation limit. Under this situation we should set an upper limit on the number of iterations.

```
c = 1 ;                    % Setup counter

while (c < 40)
      …
end
```

In *Example 5.2*, we observed that if $\alpha = 0.93$ the algorithm converges after 26 iterations due to the weighting function tolerance 1×10^{-17} being satisfied. We also observed for other values of α between 0.7 to 0.98, it can take up to 36 iterations. Accordingly, for WMM Farrow VFD filter designs of similar dimensions, we recommend setting the maximum loop iteration count to be 40.

Figure 29 shows the WMM cost against iteration. It can be seen that the decrease in WMM cost between iteration number 23 to 25 is 0.023%. Iterations beyond the 26th iteration exhibit infinitesimal change.

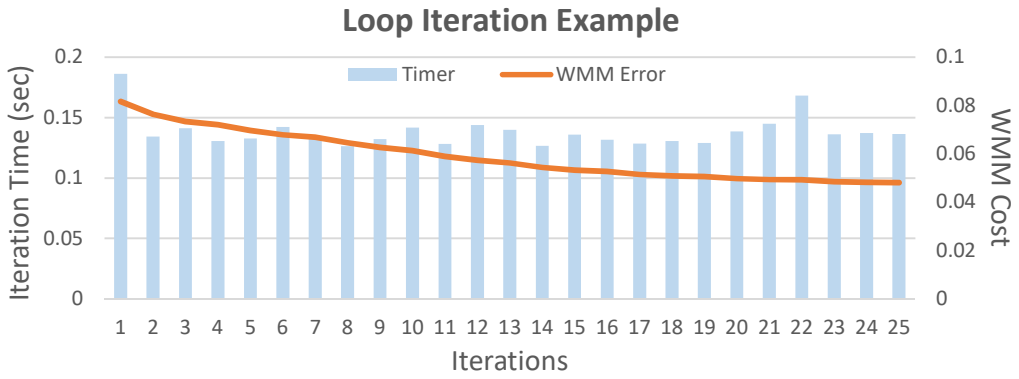


Figure 29 – Loop Iteration Example Showing WMM Cost Descend

5.5.3 Proactive IRLS Algorithm with Cyclical Sets of ω Points[7]

There exists a fundamental problem with the computational technique with the IRLS Farrow VFD filter algorithm. As seen in (5.32), the weighting function is discretised which lead to a suboptimal approximation relative to the error function defined in (5.30). The inaccuracies can be improved by increasing the number of

[7] This method can also be used for Lawson's restart routine. In fact, implementing this cyclical sets of frequency points will significantly reduce the chance of zero-weighting restart issue.

discretised sample points within ω and δ. However, an increase in sample points also lead to an exponential increase in resources used to compute the optimum coefficients.

In this thesis, we propose a remedy which switches the sample frequency points across a number of sets. Figure 30 illustrate the method for three interleaving sets of frequency points. The set are given by,

$$\boldsymbol{\omega}^{(0)} = [0 \quad \Delta\omega \quad 2\Delta\omega \quad \cdots \quad \pi - \Delta\omega], \tag{5.51}$$

$$\boldsymbol{\omega}^{(1)} = \left[\tfrac{\Delta\omega}{3} \quad \Delta\omega + \tfrac{\Delta\omega}{3} \quad 2\Delta\omega + \tfrac{\Delta\omega}{3} \quad \cdots \quad \pi - \tfrac{2\Delta\omega}{3}\right], \tag{5.52}$$

and

$$\boldsymbol{\omega}^{(2)} = \left[\tfrac{2\Delta\omega}{3} \quad \Delta\omega + \tfrac{2\Delta\omega}{3} \quad 2\Delta\omega + \tfrac{2\Delta\omega}{3} \quad \cdots \quad \pi - \tfrac{\Delta\omega}{3}\right]. \tag{5.53}$$

From one iteration to the next in the adaptive IRLS algorithm, the frequency points will be cycled through the sets $\boldsymbol{\omega}^{(0)}$, $\boldsymbol{\omega}^{(1)}$, and $\boldsymbol{\omega}^{(2)}$. The above example effectively triples the number of sample points. Clearly, methods involving more than 3 interleaving sets can also be employed.

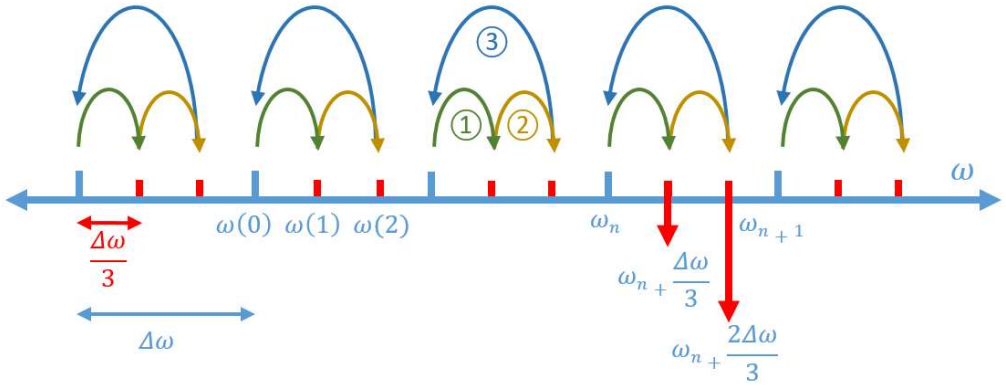


Figure 30 – Sliding Window Technique for IRLS Farrow

Adding the above feature to the existing proactive IRLS algorithm yields the following steps[8]:

1) Set the initial weighting $W_0(\omega,\delta)$, homotopy $p_0 = 2$, convergence rate $\tau_0 = 1$, and initialise the three sets of sampled frequency points.
2) Solve for the initial optimum coefficients $\mathbf{h}_0$ via (2.113) and (2.114) with $\boldsymbol{\omega}^{(0)}$.
3) Find the resulting error $\varepsilon_0(\omega,\delta)$ via (2.102). Update p by (5.40)

[8] Cyclic set can also be constructed for δ, but we found this is not necessary based on our numerical studies.

4) Update the new weighting $W_1(\omega,\delta)$ by (5.41)
5) At the *n*-th iteration, use (cyclically) the frequency points $\boldsymbol{\omega}^{(n \bmod 3)}$ with the following four steps:
    I. Solve for the new interim coefficient, $\hat{\mathbf{h}}_n$ via (2.113) and (2.114) with the weighting $W_n(\omega,\delta)$.
    II. Update the new coefficient $\mathbf{h}_n$ with (5.43) and (5.44).
    III. Update the next *p*-value via (5.28) using the proposed proactive τ (5.38).
    IV. Calculate the new error $\varepsilon_n(\omega,\delta)$ via (2.102) and the new weighting with (5.45)
6) Repeat step 5 until an exiting condition is met.

5.6 Design Examples of Adaptive IRLS

In this section, we present two WMM designs of the Farrow VFD filter using the IRLS method. These are the almost-flat filter and the low-pass filter.

In each design, we consider the following three optimisation methods.

- MATLAB's `fminimax` sequential quadratic programming (SQP) function,
- the proposed proactive IRLS, and
- the proposed proactive IRLS with cyclical frequency sets.

The designs are then compared under the following six criteria.

- WLS cost,
- WMM cost,
- maximum magnitude response error,
- maximum phase response error,
- computation time, and
- number of iterations.

The frequency ω and delay δ are uniformly sampled with a step size of $\Delta\omega = \pi/1800$ and $\Delta\delta = 1/11$ respectively.

The objective is to compare the three optimisation methods given the same starting points and exiting conditions.

5.6.1 Almost-Flat Design

For the AF design, we assume the following parameters $M = 5$, $K = 20$, $\delta_{max} = 0.5$, $\alpha = 0.83$ and the initial and desired weighting

$$W(\omega,\delta) = \begin{cases} 1, & \begin{matrix} 0 \le \omega \le \alpha\pi \\ -\delta_{max} \le \delta \le \delta_{max} \end{matrix} \\ 0, & \text{elsewhere} \end{cases} . \tag{5.54}$$

Setting MATLAB's SQP function with the following parameters, the program converges after 307 iterations.

```
iter = 500 ;
TolF = 1e-12 ;
TolX = 1e-12 ;
max_F_evals = 5000000 ;
```

For the adaptive IRLS and adaptive IRLS with three cyclical sets of frequency points they both converged after 7 iterations with the following parameters.

```
P = 400 ;
Hom = 1.7 ;
maxiter = 40 ;

w tol = 1e-12 ;
```

Table 23 – SQP, IRLS and IRLS (Cyclical Set) for the AF Design (M=5, K=20 and α=0.83)

	SQP	IRLS	IRLS (Cyclical Sets)
WLS Cost	2.3504454×10^{-3}	1.3368021×10^{-3}	1.3922748×10^{-3}
WMM Cost	1.9849030×10^{-3}	2.9964327×10^{-3}	2.6867123×10^{-3}
Max Magnitude Error	1.9849030×10^{-3}	2.9964327×10^{-3}	2.6867123×10^{-3}
Max Phase Error	2.6359851×10^{-3}	9.1870512×10^{-4}	8.8268237×10^{-4}
Computation Time (s)	511.6522	1.2407	1.4812
Iterations	307	7	7

Table 23 displays the results for the three optimisation methods. We can see the WMM cost of MATLAB's SQP is smaller than both our proactive IRLS methods. This is due to the nature of SQP where they scan the gradients around the proximity of the starting points and slowly move in the best direction. The choice of the "best" direction sometimes strikes jackpot leading to a great result such as in the example above. In other numerical studies, depending on the optimisation parameters, SQP occasionally achieved worse result compared to our proactive IRLS methods. Nonetheless the advantage of using IRLS is that it has fast convergence. Note that all three iterative

methods have a convergence tolerance of 1×10^{-12}. SQP took 307 iterations. In contrast, both our proposed adaptive IRLS took 7 iterations. Interestingly, the computational times and iteration ratios indicate that SQP takes an average of 1.6 seconds per iteration whereas the IRLS algorithm averages around 0.2 seconds per iteration. Figure 32 and Figure 33 shows the magnitude and phase response of the three design methods.

5.6.2 Low Pass Design

For the LP design, we assume the following parameters $M=6$, $K=19$, $\delta_{max}=0.5$, $\gamma_1=0.55$, $\gamma_2=0.68$ and the initial and desired weighting

$$W(\omega,\delta)=\begin{cases}1, & 0\le\omega\le\gamma_1\pi \\ & -\delta_{max}\le\delta\le\delta_{max} \\ 0, & \text{elsewhere}\end{cases}. \tag{5.55}$$

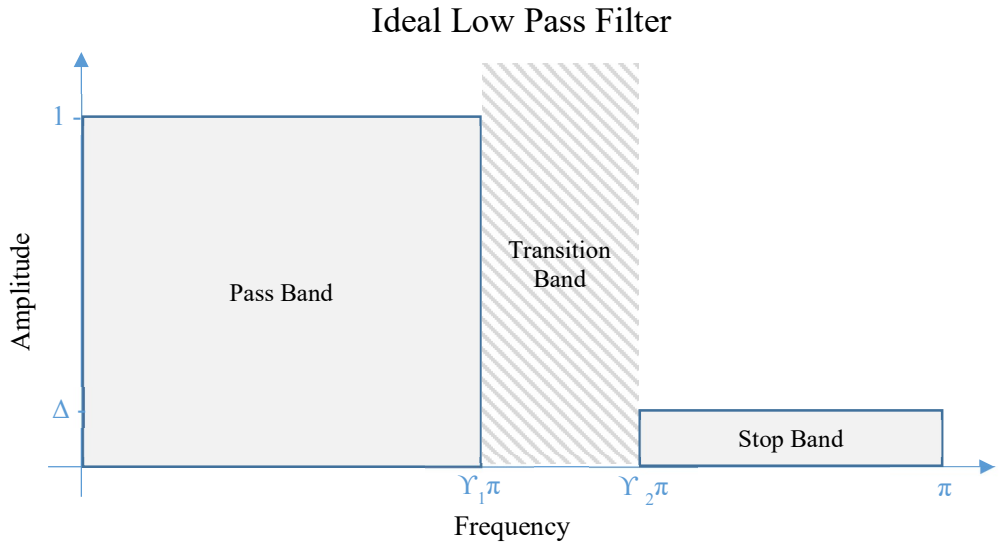


Figure 31 - The Ideal LP Filter

Setting MATLAB's SQP function with the following parameters, the program converges after 131 iterations.

```
P = 400 ;
Hom = 1.95 ;
maxiter = 40 ;

w_tol = 1e-18 ;
```

For the proposed adaptive IRLS and adaptive IRLS with three cyclical sets of frequency points they both converge after 11 iterations with the following parameters.

```
iter = 200 ;
TolF = 1e-18 ;
TolX = 1e-18 ;
max_F_evals = 5000000 ;
```

We found that for the given LP specification, the optimum homotopy is 1.95. The results in Table 24 show SQP has a smaller WMM cost compared to both the IRLS and IRLS with cyclical sets methods. In contrast, IRLS outperforms SQP in the category of computational speed despite the high homotopy value. i.e. smaller step size per iteration on the homotopy trail. The computation time and iteration ratio shows SQP averages 1.8 seconds per iteration whereas IRLS averages around 0.14 seconds per iteration.

Table 24 – SQP, IRLS and IRLS (Cyclical Set) for the LP Design (hom=1.95)

	SQP	IRLS	IRLS (Cyclical Sets)
WLS Cost	5.6942917×10^{-2}	4.2793499×10^{-2}	4.6123144×10^{-2}
WMM Cost	4.4570838×10^{-2}	4.9753558×10^{-2}	4.7527757×10^{-2}
Max Magnitude Error	4.4570838×10^{-2}	4.9753558×10^{-2}	4.7527757×10^{-2}
Max Phase Error	3.4544427×10^{-1}	2.8820495×10^{-1}	2.9709868×10^{-2}
Computation Time (s)	243.5291	2.9724	3.2716
Iterations	131	21	23

Figure 34 and Figure 35 show the magnitude and phase responses for the three designs methods, SQP, adaptive IRLS, and adaptive IRLS with cyclical frequency sets.

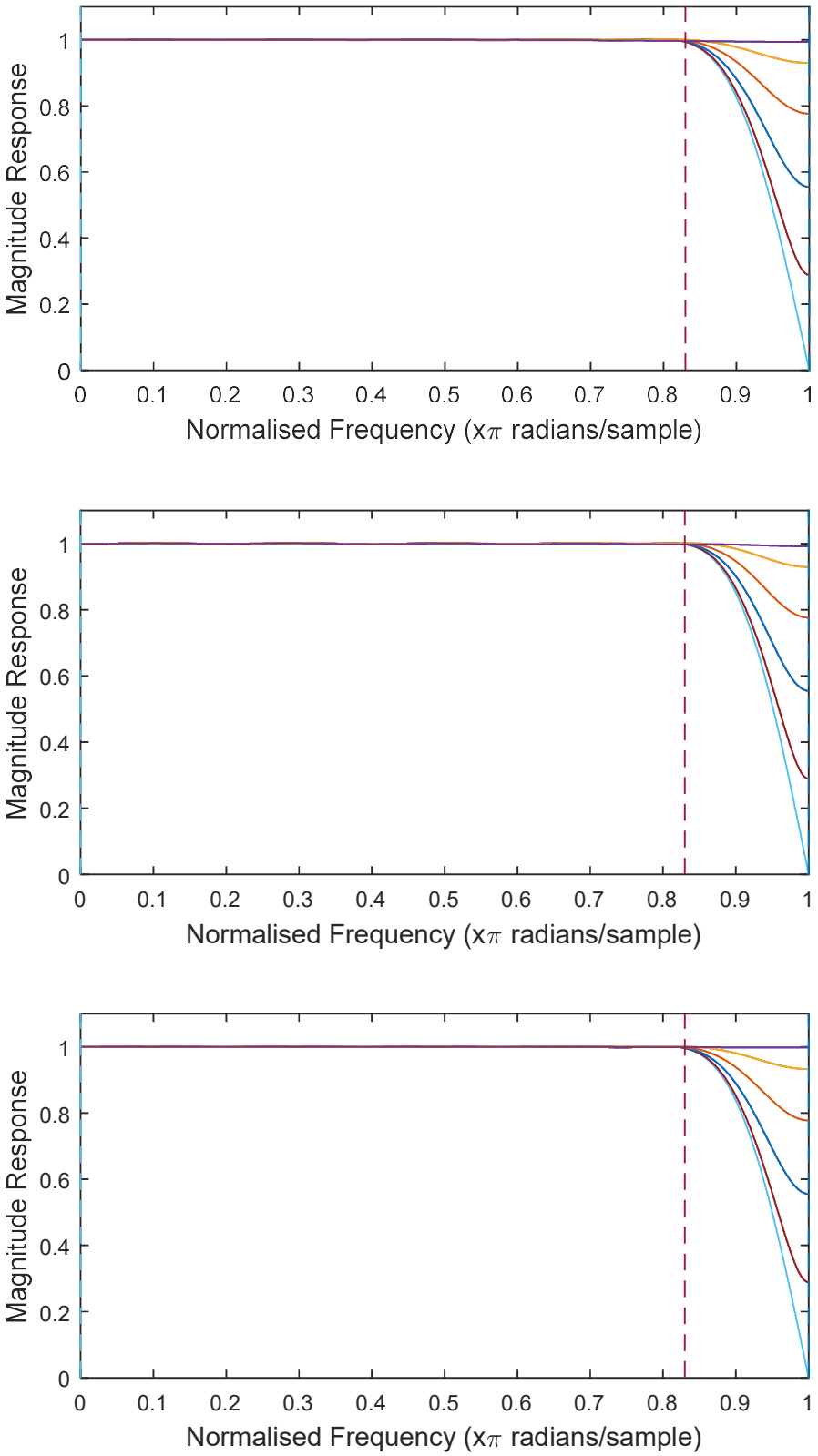


Figure 32 – SQP (top), IRLS (mid) and IRLS with cyclical frequencies sets (low) magnitude response for AF design (M=5, K=20 and α=0.83)

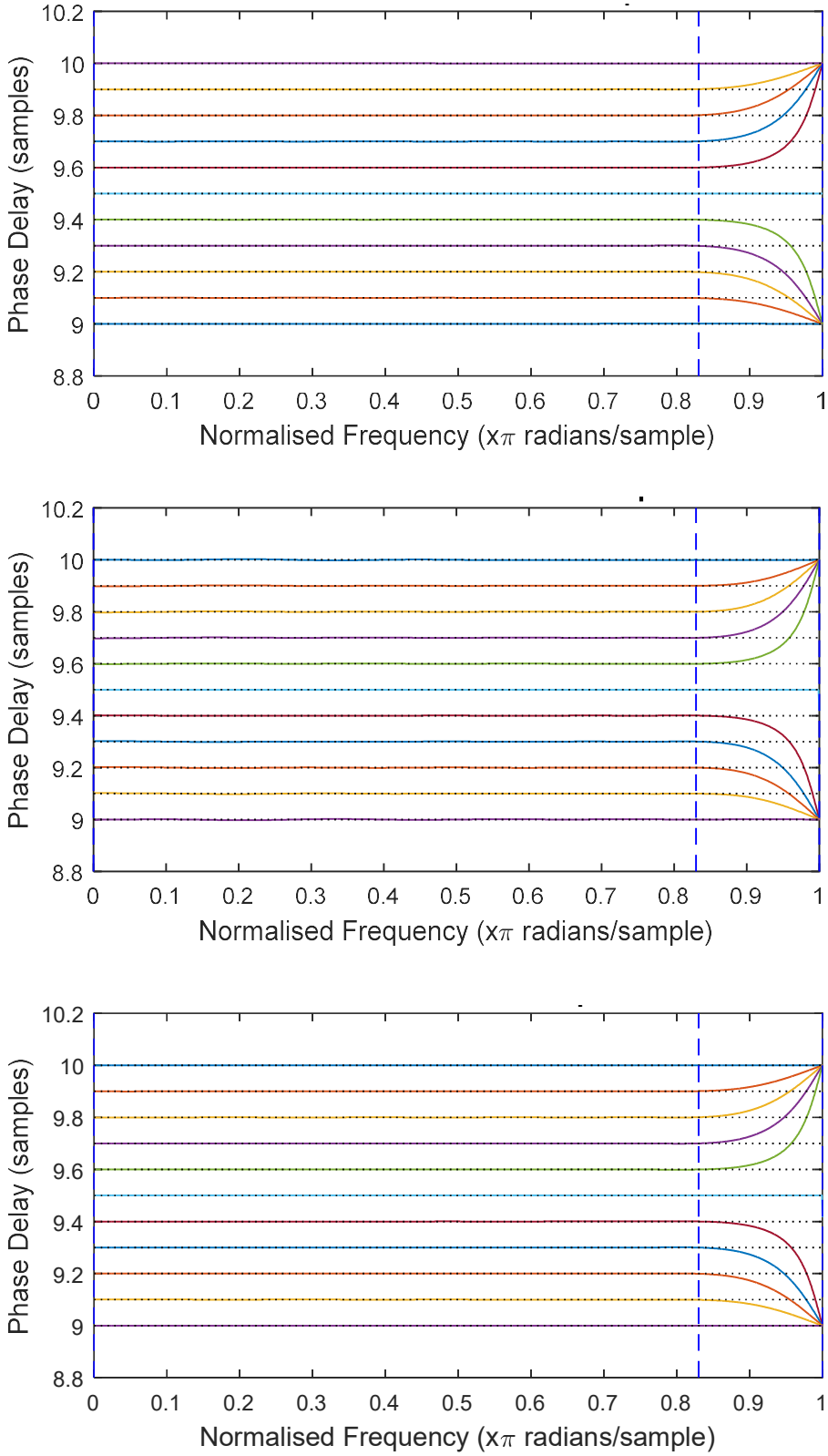


Figure 33 - SQP (top), IRLS (mid) and IRLS with cyclical frequencies sets (low) phase response for AF design (M=5, K=20 and α=0.83)

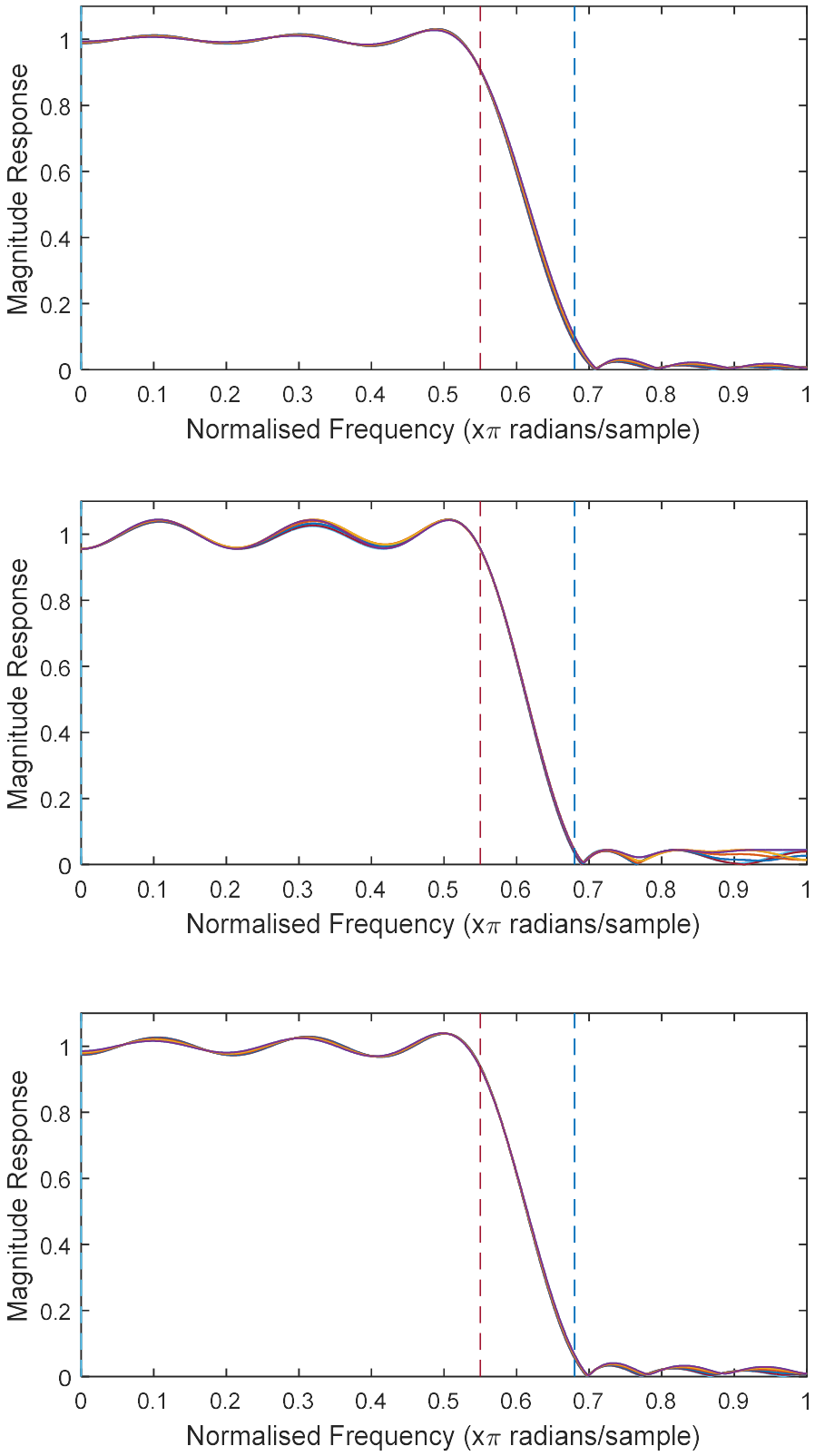


Figure 34 – SQP (top), IRLS (mid) and IRLS with cyclical frequencies sets (low) magnitude response for LP design.

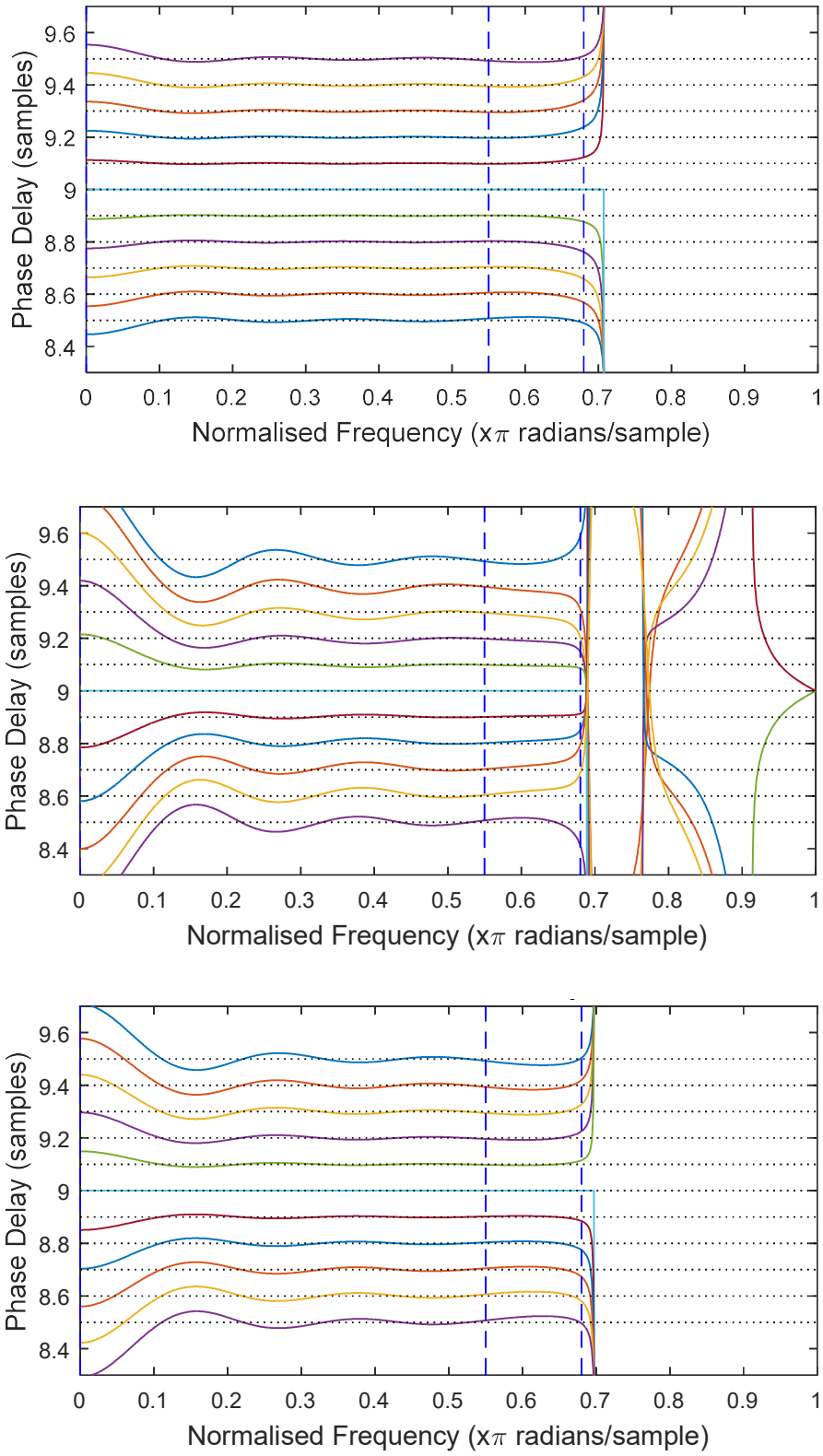


Figure 35 - SQP (top), IRLS (mid) and IRLS with cyclical frequencies sets (low) phase response for LP design.

6 Conclusion and Future Work

6.1 Summary

This thesis contains three major contributions.

Firstly, further insights on the weighted least squares design of Farrow variable fractional delay filters have been presented. In particular, the author provides a rigorous proof that the commonly held assumption that the filter coefficients are symmetrical is valid. The original discussion simply claimed that the symmetry characteristic exists because the ideal response is symmetrical. An important outcome from our proof is that sufficient conditions for the coefficients to be symmetrical are derived. Furthermore, a numerical problem that arises from the weighted least squares closed-form solution was resolved. The technique utilises a well-known matrix inversion bootstrapping method to yield an analytical expression for the inverse of a poorly conditioned matrix.

Secondly, an extensive numerical study on the design of almost-flat variable fractional delay filter has been conducted. From this study, three empirical design guides are derived. They relate the weighted least squares cost, the maximum magnitude response error cost, and the maximum phase response error cost to the Farrow design parameters, that is, the approximation order M, the filter length K and the frequency cut-off α. In our study, a peculiar behaviour in the maximum phase response error where the error does not decrease monotonically as the filter length increases was noticed. An explanation for this phenomenon was provided which leads to the conclusion that, for low phase response errors, even-length filters are preferred over odd-length filters, even if the odd-length filter is shorter.

Thirdly, we present a novel method to solve weighted minimax Farrow variable fractional delay filter design problems. The method is based on an algorithm proposed by Lawson and further developed by other researchers to solve the classic one-dimension finite impulse response filter design problem. This thesis extends the Lawson algorithm to the two-dimension Farrow variable fractional delay filter design problem. An adaptive convergence was also developed to overcome the manual restart

issue, which exists in the original Lawson and the later Vargas algorithm. Our proposed algorithm also increases the convergence rate.

6.2 Future Works

6.2.1 Decoupling and Constrained Formulations

One way to overcome fluctuating phase response errors, often seen in minimax design, is to optimise magnitude and phase separately. Decoupling the magnitude and phase is gaining momentum in the design of WMM VFD filter via various design methods [97, 98]. The justification behind this split is that it offers additional control and flexibility in the optimisation problem formulation. This contrasts with the classical method, which treats complex variables as a whole and attempts to find a balance between the magnitude and phase responses.

In addition to splitting the magnitude and phase in the optimisation process, one can further enhance the control by introducing a mask on either the magnitude and phase response. The construction of such masks allows one response to roam freely inside its error constraint while the optimisation algorithm attempts to find the optimum fit of the other response.

An example of a minimax problem that minimises the phase response error subject to magnitude response error masking is as follows. This design gives better, i.e. more linear, phase response while relaxes on the magnitude response.

$$\min_{\mathbf{h}} \max_{\substack{-\pi\le\omega\le\pi \\ -\delta_{max}\le\delta\le\delta_{max}}} W(\omega,\delta)\left|\ \angle H(\omega,\delta)-\angle H_d(\omega,\delta)\ \right|, \tag{6.1}$$

subject to $$\left|\ |H(\omega,\delta)|-|H_d(\omega,\delta)|\ \right|<\varepsilon_M(\omega)\ . \tag{6.2}$$

Reversing the role of magnitude and phase responses, the constrained WMM problem can be posed as follows.

$$\min_{\mathbf{h}} \max_{\substack{-\pi\le\omega\le\pi \\ -\delta_{max}\le\delta\le\delta_{max}}} W(\omega,\delta)\left|\ |H(\omega,\delta)|-|H_d(\omega,\delta)|\ \right|, \tag{6.3}$$

subject to $$\left|\ \angle H(\omega,\delta)-\angle H_d(\omega,\delta)\ \right|<\varepsilon_P(\omega)\ . \tag{6.4}$$

Alternatively, the minimax criteria of (6.1) and (6.3) can be replaced, respectively, by their WLS counterparts

$$\min_{\mathbf{h}} \int_{-\delta_{\max}}^{\delta_{\max}} \int_{-\pi}^{\pi} W(\omega,\delta)\left(|H(\omega,\delta)| - |H_d(\omega,\delta)|\right)^2 \, d\omega \, d\delta \ , \tag{6.5}$$

and

$$\min_{\mathbf{h}} \int_{-\delta_{\max}}^{\delta_{\max}} \int_{-\pi}^{\pi} \left(\measuredangle H(\omega,\delta) - \measuredangle H_d(\omega,\delta)\right)^2 \, d\omega \, d\delta \ . \tag{6.6}$$

6.2.2 Shortcomings of the Proposed IRLS Algorithm

In this thesis, the author had laid out the foundations of the proposed IRLS Farrow VFD filter design method. There are areas of possible research to further improve this design method.

6.2.2.1 Complex Optimisation

One obvious drawback with the proposed IRLS design method is its reliance on the weighted least squares design. Specifically, when it comes to the WLS computation, the algorithm will strive to achieve the optimum balance between magnitude and phase. However, as can be seen in (5.41), the adaptive weighting, $W(\omega,\delta)$, primarily focusses on the magnitude error. Thus, it is expected the resultant design will have a better magnitude response at the expense of the phase response. It is observed through many numerical studies of proposed IRLS Farrow VFD filter design that as the iteration increases the phase error worsens, regardless of the design or optimisation parameters. Remedies such as complex weighting or decoupling the optimisation of the magnitude and phase responses, seen in Section 6.2.1, are some of the recommended approaches.

6.2.2.2 Convergence Tuning

In the proposed IRLS Farrow VFD filter algorithm, there are three optimisation parameters that determines convergence, that is, the homotopy rate, the homotopy updating factor, and the desired *p*-norm. A difficulty is that there are no guidelines on how to set these parameters. An area of further investigation is to establish whether there exists a correlation, if any, between the design parameters and optimisation parameters.

6.2.2.3 True Minimax Solution

In one dimension, it has been proved that the IRLS algorithm will converge to the WMM solution [60]. However, in two dimensions, as stated in Section 1.3, there is currently no proof that the proposed IRLS algorithm will yield the optimum L_∞

approximation and achieve the optimum WMM Farrow solution. However, through multiple design examples and related works from other researchers, it can be seen that the results lead to almost optimum approximation.

Therefore, a challenging area of future work is to prove or disprove that the proposed IRLS algorithm will lead to the true WMM solution.

6.2.3 Laguerre-Farrow and Kautz-Farrow

It has been shown in [44] that the conventional Farrow structure can be generalized to incorporate Laguerre and Kautz filters which bestow additional degrees of freedom for a better design.

The idea is to simply replace the delay element $e^{-jk\omega}$ in the Farrow structure with an alternative orthonormal set of basis functions $\phi_k(\omega)$ in frequency, i.e.

$$H(\omega,\delta)=\sum_{k=0}^{K=1}\left[\sum_{m=0}^{M-1}h(k,m)\delta^m\right]\phi_k(\omega)\ . \tag{6.7}$$

where $\phi_k(\omega)$ can be the Laguerre or Kautz functions.

6.2.3.1 Laguerre-Farrow

The all-pass Laguerre function is represented as

$$\phi_k(\omega,\alpha)=\sqrt{1-\alpha^2}\cdot\frac{\left(z^{-1}-\alpha\right)^k}{\left(1-\alpha z^{-1}\right)^{k+1}}\ . \tag{6.8}$$

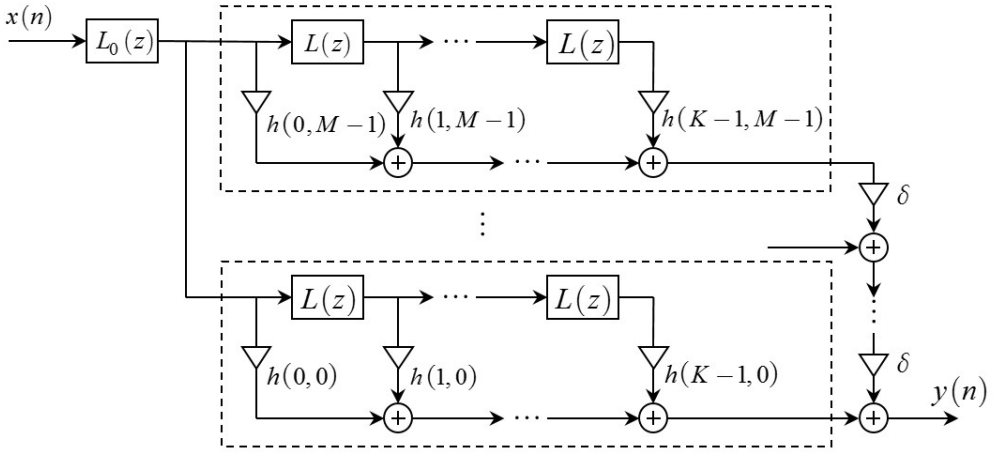


Figure 36 – The Laguerre-Farrow Structure

where $z = e^{j\omega}$ and $-1 < \alpha < 1$. Integrating the Laguerre function into the Farrow structure, one obtains the block diagram shown in Figure 36, where

$$L_0(z,\alpha) = \sqrt{1-\alpha^2} \cdot \frac{1}{1-\alpha z^{-1}} \tag{6.9}$$

and

$$L(z,\alpha) = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}} \ . \tag{6.10}$$

6.2.3.2 Kautz-Farrow

The all-pass Kautz function is denoted by

$$\phi_{2k}(\omega,\beta) = \sqrt{\tfrac{1}{2}(1+\beta)(1+\beta^*)(1-\beta\beta^*)} \cdot \frac{(1-z^{-1})\left[(z^{-1}-\beta)(z^{-1}-\beta^*)\right]^k}{\left[(1-\beta z^{-1})(1-\beta^* z^{-1})\right]^{k+1}} \tag{6.11}$$

$$\phi_{2k+1}(\omega,\beta) = \sqrt{\tfrac{1}{2}(1-\beta)(1-\beta^*)(1-\beta\beta^*)} \cdot \frac{(1+z^{-1})\left[(z^{-1}-\beta)(z^{-1}-\beta^*)\right]^k}{\left[(1-\beta z^{-1})(1-\beta^* z^{-1})\right]^{k+1}} \tag{6.12}$$

where $z = e^{j\omega}$ and $|\beta| < 1$. Integrating the Kautz function into the Farrow structure yields the block diagram show in Figure 37, where

$$K_0(z,\beta) = \sqrt{\tfrac{1}{2}(1+\beta)(1+\beta^*)(1-\beta\beta^*)} \cdot \frac{(1-z^{-1})}{(1-\beta z^{-1})(1-\beta^* z^{-1})} \ , \tag{6.13}$$

$$K_1(z,\beta) = \sqrt{\tfrac{1}{2}(1-\beta)(1-\beta^*)(1-\beta\beta^*)} \cdot \frac{(1+z^{-1})}{(1-\beta z^{-1})(1-\beta^* z^{-1})} \ , \tag{6.14}$$

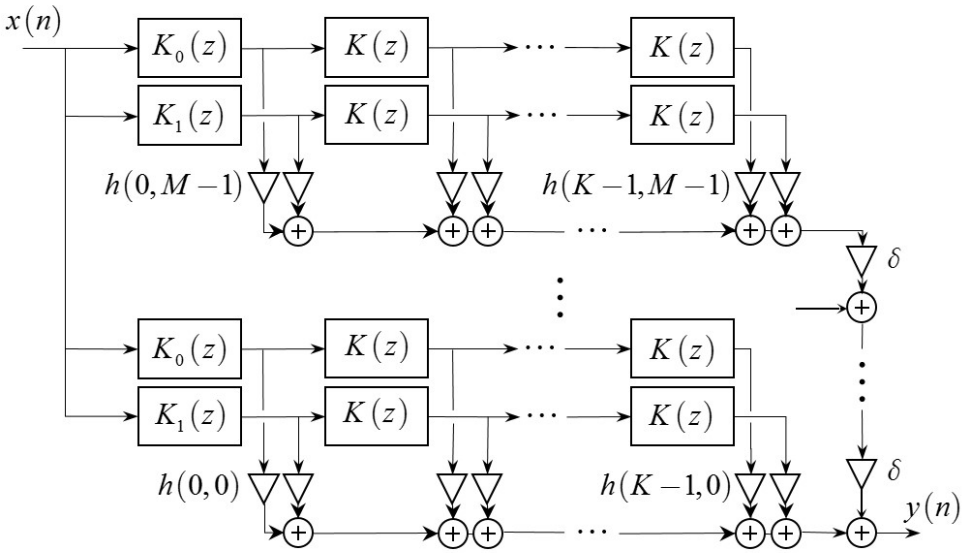


Figure 37 – The Kautz-Farrow structure

and
$$K(z,\beta)=\frac{\left(z^{-1}-\beta\right)\left(z^{-1}-\beta^{*}\right)}{\left(1-\beta z^{-1}\right)\left(1-\beta^{*}z^{-1}\right)}. \tag{6.15}$$

6.2.3.3 Potential Explorations into Laguerre and Kautz

It is highly unlikely that the coefficients $h(k,m)$ in the Laguerre-Farrow and Kautz-Farrow filter will be symmetrical. This can lead to challenging numerical issues in the optimum design of these filter in both the WLS and WMM sense.

Another open issue with the design of the Laguerre-Farrow and Kautz-Farrow filters is in the choice of the pole positions. Specifically, in the case of the Laguerre-Farrow filter, the $(k,m)^{\text{th}}$ Laguerre filter is given by

$$\phi_{k,m}(\omega,\alpha)=\sqrt{1-\alpha_{k,m}^{2}}\cdot\frac{\left(z^{-1}-\alpha_{k,m}\right)^{k}}{\left(1-\alpha_{k,m}z^{-1}\right)^{k+1}}\ . \tag{6.16}$$

The new filter structure (6.9) and (6.10) then becomes

$$\bar{L}_{k,m}(z,\alpha)=\sqrt{1-\alpha_{k,m}^{2}}\cdot\frac{1}{1-\alpha_{k,m}z^{-1}} \tag{6.17}$$

$$L_{k,m}(z,\alpha)=\frac{z^{-1}-\alpha_{k,m}}{1-\alpha_{k,m}z^{-1}} \tag{6.18}$$

as illustrated in Figure 38.

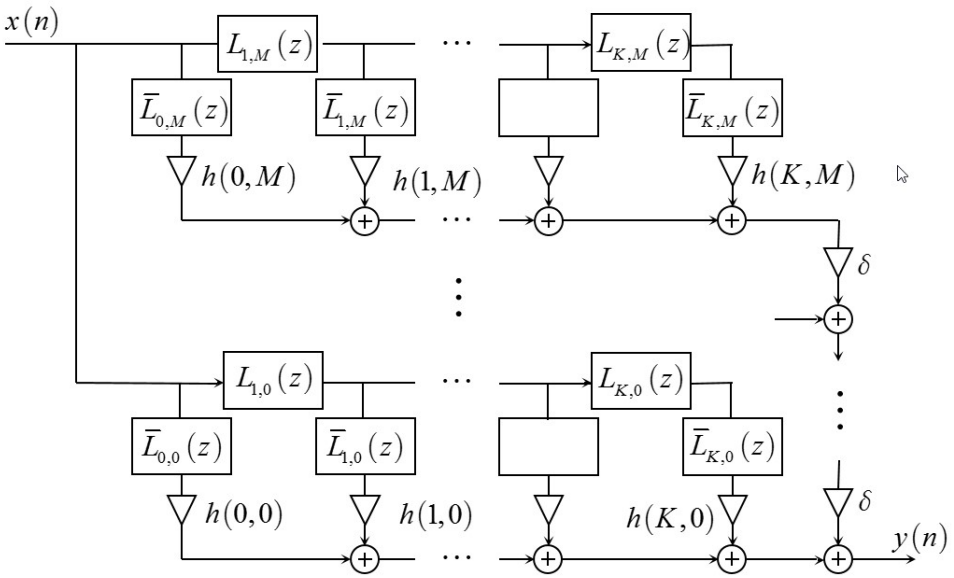


Figure 38 – The extended Laguerre-Farrow filter structure

The design problem can be simplified by requiring all the poles $\alpha_{k,m}$ to be the same. Alternatively, one can require all the poles in a row to be the same, or all the poles in a column to be the same. The optimum design problem will then include finding the optimum set of poles.

A similar consideration applies also to the design of Kautz-Farrow filters.

6.2.4 Other Notebook Ideas

- Farrow VFD filter with Hilbert responses (Type III and Type IV)
- Introduce minimax constraints on the LS design, i.e.(2.17), (2.18) and (2.19).
- Find the necessary conditions for the WLS Farrow VFD filter to exhibit coefficient symmetry.
- Formulate design guides for low-pass and band-pass WLS/WMM Farrow VFD filters.

Appendix A

A1 Bootstrapping Matrix Inversion (Inversion by Partitioning)

Let the matrix $A=[a_{ij}]$ of order n and its inverse $B=[b_{ij}]$ be portioned into submatrices of indicated orders:

$$\begin{bmatrix} A_{11} & A_{12} \\ (p\times p) & (p\times q) \\ A_{21} & A_{22} \\ (q\times p) & (q\times q) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} B_{11} & B_{12} \\ (p\times p) & (p\times q) \\ B_{21} & B_{22} \\ (q\times p) & (q\times q) \end{bmatrix} \quad \text{where} \quad p+q=n$$

Since $AB=BA=I_n$, we have

$$\begin{cases} A_{11}B_{11}+A_{12}B_{21}=I_p & A_{21}B_{11}+A_{22}B_{21}=0 \\ A_{11}B_{12}+A_{12}B_{22}=0 & A_{21}B_{12}+A_{22}B_{22}=I_q \end{cases} \tag{A.1}$$

Then, provided A_{11} is non-singular,

$$\begin{cases} B_{11}=A_{11}^{-1}+\left(A_{11}^{-1}A_{12}\right)\xi^{-1}\left(A_{21}A_{11}^{-1}\right) & B_{21}=-\xi^{-1}\left(A_{21}A_{11}^{-1}\right) \\ B_{12}=-\left(A_{11}^{-1}A_{12}\right)\xi^{-1} & B_{22}=\xi^{-1} \end{cases} \tag{A.2}$$

where $\xi=A_{22}-A_{21}\left(A_{11}^{-1}A_{12}\right)$.

In practice, A_{11} is usually taken of order $n-1$. To obtain A_{11}^{-1}, the following procedure is used. Let

$$G_2=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

$$G_3=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

$$G_4=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad \cdots$$

After computing G_2^{-1}, partition G_3 so that $A_{22}=[a_{33}]$ and use (A.2) to obtain G_3^{-1}. Repeat the process on G_4 after portioning it so that $A_{22}=[a_{44}]$ and so on.

When A is symmetric, $a_{ij} = a_{ji}$, and only $\frac{1}{2}n(n+1)$ elements need to be computed instead of the usual n^2 in obtaining A^{-1} from *adj* A. The form (A.2) is then reduced to

$$\begin{cases} B_{11} = A_{11}^{-1} + \left(A_{11}^{-1}A_{12}\right)\xi^{-1}\left(A_{11}^{-1}A_{12}\right)^T & B_{21} = B_{12}^T \\ B_{12} = -\left(A_{11}^{-1}A_{12}\right)\xi^{-1} & B_{22} = \xi^{-1} \end{cases} \tag{A.3}$$

where $\xi = A_{22} - A_{21}\left(A_{11}^T A_{12}\right)$.

When A is not symmetric, the above procedure may be used to find the inversion of $A^T A$ which is symmetric, and then the inverse of A is found by

$$A^{-1} = \left(A^T A\right)^{-1} A^T \ . \tag{A.4}$$

Example

Q: Find the inverse of $A = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$, using partitioning.

A: Take $A_{11} = \begin{bmatrix} 1 & 3 \\ 1 & 4 \end{bmatrix}$, $A_{12} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $A_{21} = [1 \quad 3]$, and $A_{22} = [4]$.

Now $A_{11}^{-1} = \begin{bmatrix} 4 & -3 \\ -1 & 1 \end{bmatrix}$, $\quad A_{11}^{-1}A_{12} = \begin{bmatrix} 4 & -3 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$,

$A_{21}A_{11}^{-1} = [1 \quad 3]\begin{bmatrix} 4 & -3 \\ -1 & 1 \end{bmatrix} = [1 \quad 0]$,

$\xi = A_{22} - A_{21}\left(A_{11}^{-1}A_{12}\right) = [4] - [1 \quad 3]\begin{bmatrix} 3 \\ 0 \end{bmatrix} = [1]$, and $\xi^{-1} = [1]$.

Then

$$B_{11} = A_{11}^{-1} + \left(A_{11}^{-1}A_{12}\right)\xi^{-1}\left(A_{21}A_{11}^{-1}\right) = \begin{bmatrix} 4 & -3 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \end{bmatrix}[1]\cdot[1 \quad 0]$$

$$= \begin{bmatrix} 4 & -3 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 7 & -3 \\ -1 & 1 \end{bmatrix}$$

$$B_{12} = -\left(A_{11}^{-1}A_{12}\right)\xi^{-1} = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$$

$$B_{21} = -\xi^{-1}\left(A_{21}A_{11}^{-1}\right) = [-1 \quad 0]$$

$$B_{22} = \xi^{-1} = [1]$$

and $$A^{-1}=\begin{bmatrix}B_{11} & B_{12}\\ B_{21} & B_{22}\end{bmatrix}=\begin{bmatrix}7 & -3 & -3\\ -1 & 1 & 0\\ -1 & 0 & 1\end{bmatrix}.$$

A2 A Special Case of Bootstrapping Symmetry

In the paper [78], the author presented a special case of the symmetry proof starting with the assumption that the weighting function is separable,

$$W(\omega,\delta)=W_1(\omega)\cdot W_2(\delta)\ , \tag{A.5}$$

and that a WLS closed-from solution exists by solving

$$\mathbf{h}_{\text{opt}}=\mathbf{Q}^{-1}\mathbf{p}\ , \tag{A.6}$$

whereby

$$\mathbf{Q}=\mathbf{Q}_{\delta}\otimes\mathbf{Q}_{\omega}\ , \tag{A.7}$$

$$\mathbf{Q}_{\delta}=\int_{-\delta_{max}}^{\delta_{max}}W_2(\delta)\boldsymbol{\delta}\boldsymbol{\delta}^T\,d\delta\ , \tag{A.8}$$

$$\mathbf{Q}_{\omega}=\int_{-\pi}^{\pi}W_1(\omega)\boldsymbol{\omega}\boldsymbol{\omega}^T\,d\omega\ , \tag{A.9}$$

and

$$\mathbf{p}=\int_{-\delta_{max}}^{\delta_{max}}\int_{-\pi}^{\pi}W(\omega,\delta)\boldsymbol{\delta}\otimes\mathbb{R}\left[\boldsymbol{\omega}\cdot H_d^*(\omega,\delta)\right]d\omega\,d\delta\ . \tag{A.10}$$

It was noted that there is an inherent issue with solving (A.6) as the inversion of the matrix $\mathbf{Q}$ which is often poorly conditioned. The usual approach to overcome this problem is to first express $\mathbf{Q}^{-1}$ as follows [76]

$$\mathbf{Q}^{-1}=\mathbf{Q}_{\delta}^{-1}\otimes\mathbf{Q}_{\omega}^{-1}\ , \tag{A.11}$$

and then find $\mathbf{Q}_{\delta}^{-1}$ and $\mathbf{Q}_{\omega}^{-1}$ using numerical methods such as Cholesky factorization [75]. In the study, we found that $\mathbf{Q}_{\omega}$ though not well-conditioned, can nevertheless be inverted in MATLAB with good precision. In contrast, we can confirm that $\mathbf{Q}_{\delta}$ is indeed badly conditioned.

Here we will describe a procedure to derive a closed-form expression or $\mathbf{Q}_{\delta}^{-1}$. Firstly, for simplicity we can will assume that $W_2(\delta)=1$ and the integration of (A.8) becomes

$$\mathbf{Q}_\delta = \begin{bmatrix} \delta_{\max} & 0 & \frac{\delta_{\max}^3}{3} & 0 & \\ 0 & \frac{\delta_{\max}^3}{3} & 0 & \frac{\delta_{\max}^5}{5} & \cdots \\ \frac{\delta_{\max}^3}{3} & 0 & \frac{\delta_{\max}^5}{5} & 0 & \\ 0 & \frac{\delta_{\max}^5}{5} & 0 & \frac{\delta_{\max}^7}{7} & \\ & \vdots & & & \ddots \end{bmatrix}, \tag{A.12}$$

which is symmetric and checkerboard patterned.

Using the block-matrix inversion technique from [99]

$$\begin{bmatrix} \mathbf{U} & \mathbf{v} \\ \mathbf{v}^{-1} & w \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{U}^{-1} + \mathbf{t}\mathbf{t}^T/s & -\mathbf{t}/s \\ -\mathbf{t}^T/s & 1/s \end{bmatrix}, \tag{A.13}$$

where $$s = w - \mathbf{v}^T\mathbf{U}^{-1}\mathbf{v}\ , \tag{A.14}$$

and $$\mathbf{t} = \mathbf{U}^{-1}\mathbf{v}\ . \tag{A.15}$$

It can be used to find the inverse of a certain matrix via piecemeal approach by building upon an already known inverse.

Therefore, starting with

$$\delta_{\max}^{-1} = \frac{1}{\delta_{\max}}\ , \tag{A.16}$$

we can bootstrap the following results:

$$\begin{bmatrix} \delta_{\max} & 0 \\ 0 & \frac{\delta_{\max}^3}{3} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\delta_{\max}} & 0 \\ 0 & \frac{3}{\delta_{\max}^3} \end{bmatrix}, \tag{A.17}$$

$$\begin{bmatrix} \delta_{\max} & 0 & \frac{\delta_{\max}^3}{3} \\ 0 & \frac{\delta_{\max}^3}{3} & 0 \\ \frac{\delta_{\max}^3}{3} & 0 & \frac{\delta_{\max}^5}{5} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{9}{4\delta_{\max}} & 0 & \frac{-15}{4\delta_{\max}^3} \\ 0 & \frac{12}{4\delta_{\max}^3} & 0 \\ \frac{-15}{4\delta_{\max}^3} & 0 & \frac{45}{4\delta_{\max}^5} \end{bmatrix}, \tag{A.18}$$

$$\begin{bmatrix} \delta_{\max} & 0 & \frac{\delta_{\max}^3}{3} & 0 \\ 0 & \frac{\delta_{\max}^3}{3} & 0 & \frac{\delta_{\max}^5}{5} \\ \frac{\delta_{\max}^3}{3} & 0 & \frac{\delta_{\max}^5}{5} & 0 \\ 0 & \frac{\delta_{\max}^5}{5} & 0 & \frac{\delta_{\max}^7}{7} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{9}{4\delta_{\max}} & 0 & \frac{-15}{4\delta_{\max}^3} & 0 \\ 0 & \frac{75}{4\delta_{\max}^3} & 0 & \frac{-105}{4\delta_{\max}^5} \\ \frac{-15}{4\delta_{\max}^3} & 0 & \frac{45}{4\delta_{\max}^5} & 0 \\ 0 & \frac{-105}{4\delta_{\max}^5} & 0 & \frac{175}{4\delta_{\max}^7} \end{bmatrix}, \tag{A.19}$$

etc.

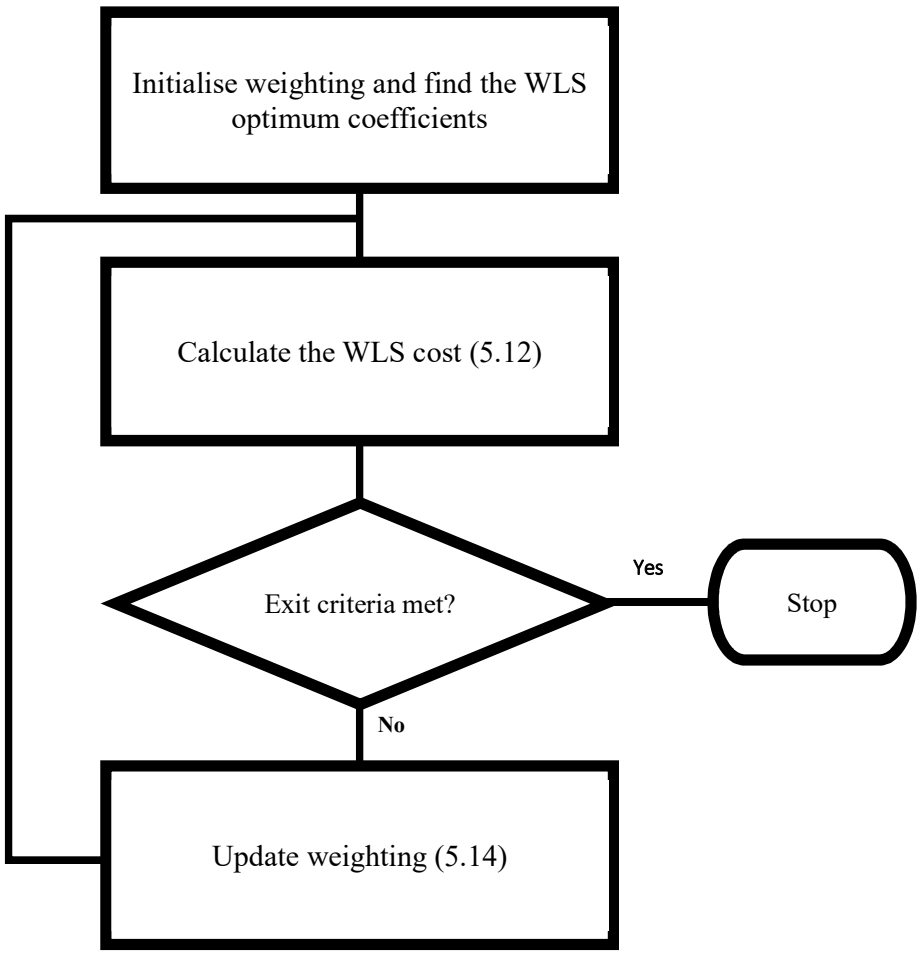
Observe that $\mathbf{Q}_\delta^{-1}$ is also symmetric and checkerboard patterned. Observe also that the above results pertain only to $W_2(\delta)=1$. For other weightings, we consider only those that are even about $\delta=0$ such that it can be expressed generally as

$$W_2(\delta)=\sum_{i=0}^{\infty} a_i \delta^{2i} \quad . \tag{A.20}$$

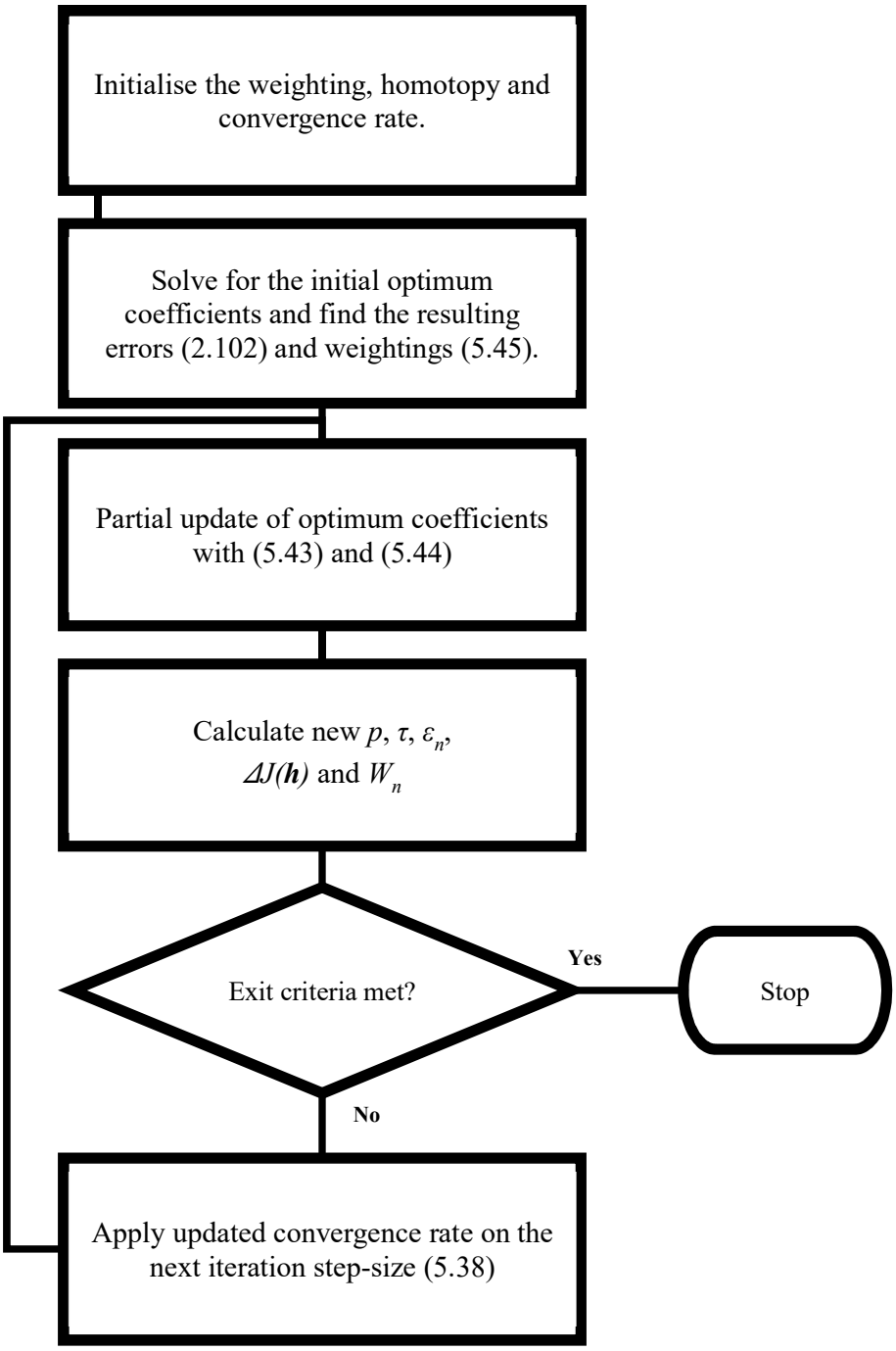
As can see from (A.8), if (A.20) holds, then closed-form expression for $\mathbf{Q}_\delta$, similar in structure to (A.12), can be developed. Also, using the block-matrix inversion technique, closed-form expression for $\mathbf{Q}_\delta^{-1}$ can be found, albeit in a more complex manner. In any event, $\mathbf{Q}_\delta$ and $\mathbf{Q}_\delta^{-1}$ will both be symmetric and checkerboard patterned.

Appendix B

B1 Original Lawson's Algorithm Flowchart



B2 Extended Adaptive IRLS flowchart



References

[1] J. S. Marques, I. M. Trancoso, J. M. Tribolet and L. B. Almeida, "Improved pitch prediction with fractional delays in CELP coding," in *Proc. 1990 IEEE Int. Conf. Acoust., Signal Processing (ICASSP'90)*, Albuquerque, 1990.

[2] J. S. Marques, J. M. Tribolet, I. M. Trancoso and L. B. Almeida, "Pitch prediction with fractional delays in CELP coding," *Proc. European Conf. Speech Communications and Technology (EUROSPEECH'89),* vol. 2, pp. 509-512, 1989.

[3] Y. Medan, "Using super resolution pitch in waveform speech coders," *Proc. 1991 IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'91),* vol. 1, pp. 633-636, 1991.

[4] M. Cernak, P. Motlicek and P. N. Garner, "On the (UN) importance of the contextual factors in HMM-based speech synthesis and coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP'13)*, Vancouver, Canada, 2013.

[5] K. Matsui, S. D. Pearson, K. Hata and T. Karnai, "Improving naturalness of text-to-speech synthesis using natural glottal source," in *IEEE Int. Conf. Acoust. Speech, SIgnal Processing (ICASSP'91)*, Toronto, 1991.

[6] T. Raition, A. Suni, J. Yamagishi, H. Pulakka, J. Numinen, M. Vainion and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *Audio, Speech, and Language Processing,* vol. 19, no. 1, pp. 153-165, 2011.

[7] A. Tarczynski and G. D. Cain, "Design of IIR fractional-sample delay filters," in *DSP for Communications Systems (DSPCS'94)*, Adelaide, 1994.

[8] A. Tarczynski, W. Kozinski and G. D. Cain, “Sampling rate conversion using fractional-sample delay,” in *IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'94)*, Adelaide, 1994.

[9] D. Massie, D. P. Rossum, B. Clark, L. Rub and J. Laroche, “Sample rate conversion using infinite impulse response filters”. United States of America Patent 8,618,961, 31 December 2013.

[10] C. C. Ko and Y. C. Lim, “Approximation of a variable-length delay line by using tapped delay line processing,” *Signal Processing,* vol. 14, no. 4, pp. 363-369, 1988.

[11] C. U. Edussooriya, L. T. Bruton, M. A. Naeini and P. Agathoklis, “Using 1-D variable fractional-delay filters to reduce the computational complexity of 3-D broadband multibeam beamformers,” *Circuits and Systems II: Express Briefs,* vol. 64, no. 4, pp. 279-283, 2014.

[12] K. Wu, T. Su and J. Pang, “Design of wideband digital array beamformer using the variable fractional delay allpass filter,” *Applied Mechanics & Materials,* vol. 719, pp. 826-832, 2014.

[13] C. Cheung, R. Shah and M. Parker, “The delay digital beamforming for wideband pulsed radar implementation,” in *Phased Array Systems & Technology, IEEE International Symposium*, Massachusetts, 2013.

[14] V. Valimaki and T. I. Laakso, “Fractinoal delay filters - design and applications,” *Theory and Applications of Nonuniform Sampling,* pp. 835-895, 2000.

[15] H. Strauss, “Implementing Doppler Shifts for Virtual Auditory Environments,” in *Audio Eng. Soc. 104th Conv.*, Amsterdam, 1998.

[16] V. Valimaki, A. Franck, J. Ramo, H. Gamper and L. Savioja, “Assisted Listening Using a Headset: Enhancing audio perception in real, augmented, and virtual environments,” *Signal Processing Magazine,* vol. 32, no. 2, pp. 92-99, 2015.

[17] J. O. Smith and B. Friedlander, “Adaptive interpolated time-delay estimation,” *IEEE Trans. Aerospace and Electronic Systems,* vol. 21, no. 3, pp. 180-199, 1985.

[18] S. Chin and L. Thevenaz, “Tunable photonic delay lines in optical fibres,” *Laser & Photonics Reviews,* vol. 6, no. 6, pp. 724-738, 2012.

[19] Z. Zhou, J. Jing, Z. Qin, Y. Fang, R. Glasser, U. Vogl and W. Zhang, “Single-Photon Level Induced Nonlinear Effectsin Parametric Self-Oscillation Regime,” *Quantum Information and Measurement,* p. W6.41, 2013.

[20] C. W. Farrow, “A continuously variable digital delay element,” *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'88),* vol. 3, pp. 2641-2645, 1988.

[21] J. Armstrong and D. Strickland, “Symbol synchronization using signal samples and interpolation,” *IEEE Trans. Communications,* vol. 41, no. 2, pp. 318-321, 1993.

[22] F. M. Gardner, “Interpolation in digital modems - part I: fundamentals,” *IEEE Trans. Communications,* vol. 41, no. 3, pp. 501-507, 1993.

[23] L. Erup, F. M. Gardner and R. A. Harris, “Interpolations in digital modems - part II: implementations and performance,” *IEEE Trans. Communications,* vol. 41, no. 6, pp. 998-1008, 1993.

[24] L. Gabrielli, V. Valimaki, H. Penttinen, S. Squartini and S. Bilbao, “A digital waveguide-based approach for Clavinet modeling and synthesis,” *EURASIP Journal on Advances in Signal Processing,* vol. 1, pp. 1-14, 2013.

[25] V. Valimaki, J. Pakarinen, C. Erkut and M. Karjalainen, “Discrete-time modelling of musical instruments,” *Reports on progress in physics,* vol. 69, no. 1, p. 1, 2005.

[26] T. L. Laakso, V. Valimaki, M. Karjalainen and U. K. Laine, "Splitting the unit delay: tools for fractional delay filter design," *IEEE Signal Process,* vol. 1, no. 13, pp. 30-50, 1996.

[27] A. V. Oppenheim and R. W. Schafer, Discrete Time Signal Processing, Prentice-Hall, 1989.

[28] S. R. Powell and P. M. Chau, "A Technique for Realizing Linear-Phase IIR Filters," *IEEE Trans. Signal Processing,* vol. 39, no. 11, pp. 2425-2435, 1991.

[29] A. Papoulis, Signal Analysis, New York: McGraw-Hill, 1977.

[30] J. W. Adams, "Alternate approach to digital phase shift filter," in *Proc. Int. Symp. Signal Processing: Theories, Implementations and Applications*, Brisbane, 1987.

[31] G. Oetken, "A new approach for the design of digital interpolating filters," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 27, no. 6, pp. 637-643, 1979.

[32] T. W. Parks and C. S. Burrus, Digital Filter Design, New York: Wiley-Interscience, 1987 .

[33] H. Johansson and E. Hermanowicz, "Two-Rate Based Low-Complexity Variable Fractional-Delay FIR Filter Structures," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 60, no. 1, pp. 136-149, 2013.

[34] M. Abbas, O. Gustaffson and H. Johansson, "On the Fixed-Point Implementation of Fractional-Delay Filters Based on the Farrow Structure," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 60, no. 4, pp. 926-937, 2013.

[35] C. Luo, L. Zhu and J. H. McClellan, "A general structure for the design of adjustable FIR filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, 2013.

[36] A. Tarczynski, G. D. Cain, E. Hermanowicz and M. Rojewski, "WLS design of variable frequency response FIR filters," in *Proceedings of 1997 IEEE International Symposium Circuits and Systems (ISCAS '97)*, Hong Kong, 1997.

[37] S. C. Pei and C. C. Tseng, “A comb filter design using fractional-sample delay,” in *IEEE International Symposium on Circuits and Systems (ISCAS '97)*, Hong Kong, 1997.

[38] S. U. Ahmad and A. Antoniou, “A Genetic Algorithm for the Design of Tunable Fractional-Delay Allpass IIR Filter Structures,” in *Canadian Conference on Electrical and Computer Engineering (CCECE'07)*, Vancouver, 2007.

[39] T. B. Deng, “Linear programming-based minimax design of odd-order variable fractional-delay filters,” in *17th Asia-Pacific Conference on Communications (APCC'11)*, Sabah, 2011.

[40] C. C. Tseng and S. L. Lee, “Closed-form design of variable fractional delay filter using discrete Fourier transform,” in *Signal Processing Conference, 2009 17th European*, Glasgow, 2009.

[41] C. C. Tseng and S. L. Lee, “Design of variable fractional delay filter using expansions of hyperbolic functions,” in *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, Tokyo, 2010.

[42] H. H. Dam, “Design of Allpass Variable Fractional Delay Filter with Powers-of-Two Coefficients,” *IEEE Signal Processing Letters,* vol. 22, no. 10, pp. 1643-1646, 2015.

[43] S. C. Pei and C. C. Tseng, “An efficient design of a variable fractional delay filter using a first-order differntiator,” *IEEE Signal Processing Letters,* vol. 10, no. 10, pp. 307-310, 2003.

[44] G. J. Cook, Y. H. Leung and Y. Liu, “On the design of variable fractional delay filters with Laguerre and Kautz filters,” *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03),* vol. 6, pp. VI-281-4, 2003.

[45] C. C. Tseng, “Design of variable fractional delay FIR filters using symmetry,” *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'04),* vol. III, pp. 477-480, 2004.

[46] T. B. Deng and Y. Lian, "Weighted-Least-Squares Design of Variable Fractional-Delay FIR Filters Using Coefficient Symmetry," *IEEE Transactions on Signal Processing,* vol. 54, no. 8, pp. 3023-3038, 2006.

[47] T. B. Deng, "Symmetric Structures for Odd-Order Maximally Flat and Weighted-Least-Squares Variable Fractional-Delay Filters," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 54, no. 12, pp. 2718-2732, 2007.

[48] J. J. Shyu and S. C. Pei, "A generalized approach to the design of variable fractional-delay FIR digital filters," *Signal Processing,* vol. 88, no. 6, pp. 1428-1435, 2008.

[49] H. Johansson and P. Lowerborg, "On the design of adjustable fractional delay FIR filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 50, no. 4, pp. 164-169, 2003.

[50] J. J. Shyu, S. C. Pei, C. H. Chan and Y. D. Huang, "Minimax design of variable fractional-delay FIR digital filters by iterative weighted least-squares approach," *Signal Processing,* vol. 89, no. 9, p. 1774–1781, 2009.

[51] T. B. Deng, "Minimax Design of Low-Complexity Allpass Variable Fractional-Delay Digital Filters," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 8, pp. 2075-2086, 2010.

[52] E. Hermanowicz and H. Johansson, "On designing minimax adjustable wideband fractional delay FIR filters using two-rate approach," in *Proceedings of the 2005 European Conference Circuit Theory and Design (ECCTD'05)*, Cork, 2005.

[53] T. B. Deng, "Minimax Design of Low-Complexity Even-Order Variable Fractional-Delay Filters Using Second-Order Cone Programming," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 58, no. 10, pp. 692-696, 2011.

[54] M. Willem, Minimax theorems, Boston: Springer Science & Business Media, 1997.

[55] T. B. Deng, "Decoupling Minimax Design of Low-Complexity Variable Fractional-Delay FIR Digital Filters," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 58, no. 10, pp. 2398-2408, 2011.

[56] K. E. Khamei, A. Nabavi and S. Hessabi, "Design of variable fractional delay FIR filters using genetic algorithm," in *10th IEEE International Conference on Electronics, Circuits and Systems (ICECS'03)*, Miami, 2003.

[57] C. K. S. Pun, Y. C. Wu, S. C. Chan and K. L. Ho, "On the design and efficient implementation of the Farrow structure," *IEEE Signal Processing Letters,* vol. 10, no. 7, pp. 189-192, 2003.

[58] T. S. Motzkin and J. L. Walsh, "Polynomials of best approximation on an interval," in *Proceedings of the National Academy of Sciences*, California, 1959.

[59] T. S. Motzkin and J. L. Walsh, "Polynomials of best approximation on a real finite point set I," in *Proceedings of the National Academy of Sciences of the United States of America*, Los Angeles, 1957.

[60] C. L. Lawson, Contributions to the Theory of Linear Least Maximum Approximations, Ph. D. Thesis, UCLA, 1961.

[61] J. R. Rice and K. H. Usow, "The Lawson algorithm and extensions," *Mathematics of Computation,* vol. 22, no. 101, pp. 118-127, 1968.

[62] L. A. Karlovitz, "Construction of nearest points in the Lp, p even, and L∞ norms. I," *Journal of Approximation Theory,* vol. 3, no. 2, pp. 123-127, 1970.

[63] S. W. Kahng, "Best Lp approximation," *Math. Comp.,* vol. 26, no. 118, pp. 505-508, 1972.

[64] R. Fletcher, J. A. Grant and M. D. Hebden, "The calculation of linear best Lp approximations," *The Computer Journal,* vol. 14, no. 3, pp. 276-279, 1971.

[65] C. S. Burrus, J. A. Barreto and I. W. Selesnick, “Iterative reweighted least-squares design of FIR filters,” *IEEE Transactions on Signal Processing,* vol. 42, no. 11, pp. 2926-2936, 1994.

[66] J. Nocedal and S. J. Wright, Numerical Optimization, New York: Springer Science & Business Media, 1999.

[67] E. W. Cheney, Introduction to Approximation Theory, New York: McGraw-Hill, 1966.

[68] Y. Kamp and J. P. Thiran, “Chebyshev approximation for two-dimensional nonrecursive digital filters,” *IEEE Trans. Circuits Syst.,* Vols. CAS-22, pp. 208-218, 1975.

[69] D. B. Harris and R. M. Mersereau, “A comparison of iterative methods for optimal two-dimensional filter design,” *ICASSP '77. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1977,* pp. 527-530, 1977.

[70] V. Algazi, M. S. Suk and C. S. Rim, “Design of almost minimax FIR filters in one and two dimensions by WLS techniques,” *IEEE Transactions on Circuits and Systems,* vol. 33, no. 6, pp. 590-596, 1986.

[71] J. A. Barreto and C. S. Burrus, “Iterative reweighted least squares and the design of two-dimensional FIR digital filters,” in *IEEE International Conference Image Processing (ICIP'94)*, Austin, 1944.

[72] C. Y. Chi and S. L. Chiou, “A new self-initiated WLS approximation method for the design of two-dimensional equiripple FIR digital filters,” in *IEEE International Symposium on Circuits and Systems (ISCAS'92)*, San Diego, 1992.

[73] C. H. Hsieh, C. M. Kuo, Y. D. Jou and Y. L. Han, “Design of two-dimensional FIR digital filters by a two-dimensional WLS technique,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 44, no. 5, pp. 348-357, 1997.

[74] R. A. Vargas, Iterative Design of Digital Filters, Ph. D. Thesis, Rice University, 2008.

[75] T. B. Deng, "Discretization-free design of variable fractional-delay FIR digital filters," *IEEE Trans Circuits Syst. II, Analog Digit. Singal Process.,* vol. 48, no. 6, pp. 637-644, 2001.

[76] Y. D. Huang, S. C. Pei and J. J. Shyu, "WLS design of variable fractional-delay FIR filters using coefficient relationship," *IEEE Trans. Circuits Syst. II, Express Briefs,* vol. 56, no. 3, pp. 220-224, 2009.

[77] D. G. Luenberger, Linear and Nonlinear Programming, 2nd ed., New York: Addison-Wesley, 1984.

[78] C. W. K. Chu and Y. H. Leung, "Further results on the WLS design of variable fractional delay filters," in *6th International Conference on Signal Processing and Communication Systems (ICSPCS'12)*, Gold Coast, 2012.

[79] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems,* vol. 25, no. 9, pp. 772-781, 1978.

[80] T. K. Moon and W. C. Stirling, Mathematical Methods and Algorithms for Signal Processing, New Jersey: Prentice Hall, 2000.

[81] J. Makhoul, "On the eigenvectors of symmetric Toeplitz matrices," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 29, no. 4, pp. 868-872, 1981.

[82] J. J. Shyu, S. C. Pei, C. H. Chan, Y. D. Huang and S. H. Lin, "A new criterion for the design of variable fractional-delay FIR digital filters," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 57, no. 2, pp. 368-377, 2010.

[83] T. B. Deng, "Generalized WLS method for design all-pass variable fractional-delay digital filters," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 56, no. 10, pp. 2207-2220, 2009.

[84] K. Levenberg, "A method for the solution of certain non-linear problems in lease squares," *Quarterly of Applied Mathematics 2,* pp. 164-168, 1994.

[85] F. Mosteller and J. W. Tukey, Data analysis and regression: a second course in statistics, Addison-Wesley Pub. Co., 1977.

[86] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician,* vol. 42, no. 1, pp. 59-66, Feburary 1988.

[87] R. K. Brayton, S. W. Director, G. D. Hachtel and L. Vidigal, "A New Algorithm for Statistical Circuit Design Based on Quasi-Newton Methods and Function Splitting," *IEEE Trans. Circuits and Systems,* Vols. CAS-26, pp. 784-794, 1979.

[88] R. H. Byrd, J. Nocedal and R. A. Waltz , "KNITRO: An Integrated Package for Nonlinear Optimization," 06 02 2006. [Online]. Available: http://www.artelys.com/downloads/pdf/composants-numeriques/knitro/papers/integratedpackage.pdf. [Accessed 12 03 2016].

[89] P. Gill, W. Murray, M. Saunders and M. Wright, "User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming," 04 06 2001. [Online]. Available: http://www.ccom.ucsd.edu/~peg/papers/npdoc.pdf. [Accessed 23 02 2016].

[90] K. Schittkowski, "NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search," 07 2015. [Online]. Available: http://www.klaus-schittkowski.de//nlpqlp_rep.htm. [Accessed 07 03 2016].

[91] J. S. Hansen, GNU Octave Beginner's Guide, Birmingham: Packt Publishing, 2011.

[92] T. E. Oliphant, "Guide to NumPy," 7 12 2006. [Online]. Available: http://csc.ucdavis.edu/~chaos/courses/nlp/Software/NumPyBook.pdf. [Accessed 14 02 2016].

[93] A. K. Cline, “Rate of Convergence of Lawson's Algorithm,” *Mathematics of Computation,* vol. 26, no. 117, pp. 167-176, 1972.

[94] J. L. Walsh and T. S. Motzkin, “Polynomials of Best Approximation on an Interval,” *Proceedings of National Academy of Sciences,* vol. 45, pp. 1523-1528, October 1959.

[95] J. R. Rice, The Approximation of Functions Vol. 2: Nonlinear and multivariate theory, Massachusetts: Addison-Wesley, 1969.

[96] M. Aoki, Introduction to Optimization Techniques, The Macmillan Company, 1971.

[97] T. B. Deng, “Iteratively weighted minimax design of variable fractional-delay FIR digital filters,” in *4th International Conference on Intelligent and Advanced Systems (ICIAS'12)*, Kuala Lumpur, 2012.

[98] H. H. Dam, “Design of Variable Fractional Delay Filters with Fractional Delay Constraints,” *IEEE Signal Processing Letters,* vol. 21, no. 11, pp. 1361-1364, 2014.

[99] F. Ayres, Schaum's Outline of Theory and Problems of Matrices, New York: McGraw-Hill Education, 1962.
