DESIGN AND IMPLEMENTATION OF COMPUTATIONALLY EFFICIENT DIGITAL FILTERS

YU JIANGHONG

(Master of Engineering, HIT)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgement

First, I would like to thank my thesis supervisor, Dr. Lian Yong, for his consistent support, advice and encouragement during my Ph.D. candidature. Dr. Lian's profound knowledge and abundant experience helped my research work go ahead smoothly.

I also want to thank all my colleagues and friends in VLSI & Signal Processing Laboratory, for helping me solve problems encountered in my research work. I enjoy the life in Singapore together with them. They are Yu Yajun, Cui Jiqing, Yang Chunzhu, Jiang Bin, Xu Lianchun, Liang Yunfeng, Luo Zhenying, Wang Xiaofeng, Cen Ling, Yu Rui, Wu Honglei, Wei Ying, Hu Yingping, Gu Jun, Tian Xiaohua and Pu Yu.

This thesis is dedicated to my deeply loved father Yu Jinghe and mother Zheng Yide. Their concern and expectation make me have confidence and perseverance in mind, and help me overcome all kinds of difficulties.

Contents

Summary

In this thesis, two computationally efficient structures of symmetric FIR filters are discussed: the parallel filter and the frequency-response masking (FRM) structure.

A basic parallel filter is composed of a parallel prefilter and an equalizer. A new design method based on weighted least square (WLS) is proposed to jointly optimize all subfilters in a parallel filter. New equations are developed to estimate the lengths of subfilters in a jointly optimized basic parallel filter.

When subfilters in a FRM filter are jointly optimized, lengths of two masking filters are reduced. This reduction makes some original design equations inaccurate for jointly optimized FRM filters. A new set of design equations are developed. These equations give accurate estimations of subfilter lengths and the interpolation factor in a jointly optimized FRM filter.

An even-length FIR filter is proposed to be utilized as the prototype filter in a FRM filter. New structures are proposed for the synthesis of FRM filters

with even-length prototype filters. Sequential quadratic programming (SQP) is utilized to jointly optimize all the subfilters in a FRM filter.

In addition, a new design method is proposed to improve design efficiency of jointly optimized FRM filters. This method is based on a dynamic frequency grid points allocation scheme, resulting in significant savings in memory and computing time.

List of Figures

List of Tables

List of Abbreviations

CP	Cyclotomic Polynomial
CPF	Cyclotomic Polynomial Filter
FIR	Finite Impulse Response
FRM	Frequency-Response Masking
IFIR	Interpolated Finite Impulse Filter
IIFOP	Inverse of Interpolated First Order Polynomial
IIR	Infinite Impulse Response
ISOP	Interpolated Second Order Polynomial
MILP	Mixed Integer Linear Programming
RRS	Recursive Running Sum
SDP	Semidefinite Programming
SoCP	Second-order Cone Programming
SPOT	Sum of Power Of Two
SQP	Sequential Quadratic Programming
SSS	Simple Symmetric Sharpening
WLS	Weighted Least Square

List of Symbols

L	An integer
ω	Radian frequency
f	Normalized frequency
$\boldsymbol{g}$	A matrix or vector $\boldsymbol{g}$
$\Delta\boldsymbol{g}$	Increase of vector $\boldsymbol{g}$
$\nabla f(\boldsymbol{g})$	Partial derivative of function f with respect to vector $\boldsymbol{g}$
$\boldsymbol{H}$	Hessian Matrix

Chapter 1

Introduction

In 1928, Harry Nyquist articulated his famous sampling theorem [1], and it was mathematically proven by Claude Shannon in 1949 [2]. The sampling theorem is a fundamental theorem of digital signal processing (DSP). In the past decades, more and more new DSP algorithms have been developed. These new algorithms effectively reduce the complexity and improve the performance of DSP systems. At the same time, with the development of integrated circuits and digital signal processors, DSP techniques are widely employed in fields of communications, satellite, radar, audio and image processing.

Digital filters play an important role in the field of DSP. Digital filter can be classified into two classes: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. FIR filters have the advantage of guaranteed stability, which is often a fatal problem that IIR filters have to face. Moreover, symmetric

FIR filter provides linear phase frequency response, which is important in many applications. However, an IIR filter generally has a lower complexity than a corresponding FIR filter. The arithmetic computation cost for every output sample of a FIR filter is higher, especially when the transition bandwidth is narrow. Recently, more and more computationally efficient digital filter structures have been proposed to reduce the filter complexity. Meanwhile, many new design methods were developed to shorten the design time, or improve the performance of digital filters. Digital filters have become more attractive than ever.

1.1 Literature Review

In the past decades, many new computationally efficient FIR structures have been proposed to reduce the number of multipliers and adders in a FIR filter. The number of multipliers and adders of a FIR filter is determined by the filter length. At the same time, design methods, such as the Remez iterative exchange design method [3, 7, 8], require that the filter length is known in advance. Therefore, it is needed to estimate the filter length accurately. The literature review begins with the length estimation of a FIR filter.

1.1.1 Low Pass FIR Filter Length Estimation

The length of a linear phase low pass FIR filter, N, is affected by four parameters: passband ripple δ_p, stopband ripple δ_s, passband edge f_p and stopband edge f_s. The relationship between N, δ_p, δ_s and transition bandwidth ΔF ($\Delta F = f_s - f_p$) is given in [9, 10, 12]. In [9] and [10], Herrmann *et al.* gave the filter length estimation as

$$\hat{N}_1(\Delta F, \delta_p, \delta_s) = \left\langle \frac{D_\infty(\delta_p, \delta_s)}{\Delta F} - f(\delta_p, \delta_s) \cdot \Delta F + 1 \right\rangle \tag{1.1}$$

where

$$\begin{aligned} D_\infty(\delta_p, \delta_s) &= \left[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3\right]\log_{10}\delta_s + \\ &\quad \left[a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6\right] \qquad (1.2) \\ f(\delta_p, \delta_s) &= b_1 + b_2(\log_{10}\delta_p - \log_{10}\delta_s) \qquad (1.3) \\ a_1 &= 5.309 \times 10^{-3}, \quad a_2 = 7.114 \times 10^{-2} \\ a_3 &= -4.761 \times 10^{-1}, \quad a_4 = -2.660 \times 10^{-3} \\ a_5 &= -5.941 \times 10^{-1}, \quad a_6 = -4.278 \times 10^{-1} \\ b_1 &= 11.01217, \qquad b_2 = 0.51244. \end{aligned}$$

In (1.1), $\langle a \rangle$ denotes the nearest odd integer from a (For example, $\langle 6.1 \rangle = 7$).

Besides the formula proposed in [9] and [10], Kaiser [12] independently proposed another formula to estimate FIR filter length, i.e.

$$\hat{N}_2(\Delta F, \delta_p, \delta_s) = \left\lceil \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{14.6\Delta F} + 1 \right\rceil \tag{1.4}$$

were, $\lceil a \rceil$ denotes the minimum integer greater than or equal to a. In (1.4), $\hat{N}_2$ is an odd integer.

Equations (1.1) and (1.4) consider the odd length FIR filter cases, but do not take into account of the effects of passband and stopband edges. Ichige, Iwaki and Ishii developed a new formula to estimate the length of a low pass FIR filter [64, 67, 73]. Their formula is much more accurate than Equation (1.1) and (1.4), because it takes into account of effects of bandedges. The filter length can be estimated by

$$N(f_p,\ \Delta F,\ \delta_p,\ \delta_s) = N_1(f_p,\ \Delta F,\ \delta_p) + DN(f_p,\ \Delta F,\ \delta_p,\ \delta_s) \tag{1.5}$$

where,

$$\begin{aligned}
N_1(f_p, \Delta F, \delta) &= \lceil N_c(\Delta F, \delta) \cdot \{ g(f_p, \Delta F, \delta) + \\
&\quad g(0.5 - \Delta F - f_p, \Delta F, \delta) + 1 \}/3 \rceil && (1.6)\\
N_c(\Delta F, \delta) &= \left\lceil \frac{1.101\{-\log_{10}(2\delta)^{1.1}\}}{\Delta F} + 1 \right\rceil && (1.7)\\
g(f_p, \Delta F, \delta) &= \frac{2}{\pi} \arctan \left\{ v(\Delta F, \delta) \left(\frac{1}{f_p} - \frac{1}{0.5 - \Delta F} \right) \right\} && (1.8)\\
v(\Delta F, \delta) &= 2.325(-\log_{10} \delta)^{-0.445} (\Delta F)^{-1.39} && (1.9)\\
DN(f_p, \Delta F, \delta_p, \delta_s) &= \lceil N_m(\Delta F, \delta_p, \delta_s) \cdot \big\{ h(f_p, \Delta F, 1.1) - \\
&\quad [h(0.5 - \Delta F - f_p, \Delta F, 0.29) - 1]/2 \big\} \rceil && (1.10)\\
N_m(\Delta F, \delta_p, \delta_s) &= 0.52 \cdot \frac{\log_{10}(\delta_p/\delta_s)}{\Delta F} (-\log_{10} \delta_p)^{0.17} && (1.11)\\
h(f_p, \Delta F, c) &= \frac{2}{\pi} \arctan \left\{ \frac{c}{\Delta F} \left(\frac{1}{f_p} - \frac{1}{0.5 - \Delta F} \right) \right\}. && (1.12)
\end{aligned}$$

Although formulas reviewed so far have different accuracy, all of them indicate

that the length of a FIR filter is approximately proportional to the reciprocal of the transition bandwidth ΔF. A very narrow transition bandwidth will result in a long FIR filter. To reduce the complexity of the FIR filter, many filter structures have been proposed which reduce the complexity of the filter.

In the following sections, these FIR structures will be reviewed one by one.

1.1.2 Prefilter-Equalizer Approach

The essence of the prefilter-equalizer approach is to compose a filter using two subfilters. It can be synthesized by following two steps [23]:

(1) Select an efficient "prefilter" for the given specification. The requirement for this prefilter is to have a minimum number of multipliers, while having a reasonably large stopband attenuation.

(2) Design an "equalizer" to compensate the frequency response of the prefilter, and cascade the prefilter with the "equalizer" to achieve the desired overall specifications.

With the help of the prefilter's stopband attenuation, the equalizer and the overall prefilter-equalizer structure will generally require less multipliers compared with a filter designed in a direct form

Much effort has been made to reduce the complexity of both the prefilter and the equalizer. It is preferred that the prefilter is a multiplier-free filter. In [22] and [23], Adams and Willson proposed to use the "recursive running sum" (RRS) [14] filter as the prefilter in the design of a low pass filter. A RRS FIR filter is a low pass filter, having equally spaced zeros on the unit circle, and provides about 13 dB attenuation in the stopband (with respect to the passband peak at normalized radian frequency $\omega = 0$). The implementation of a RRS filter requires L (L is an integer) delays and two adders, which is simple and efficient. If large stopband attenuation is required, several RRS prefilters can be cascaded to produce the desired stopband attenuation. As RRS filters are free from multiplication, the total number of multipliers required for an overall filter depends on the equalizer. Generally speaking, the equalizer requires less arithmetic operations than a traditional FIR filter does. Therefore, the prefilter-equalizer structure may reduce the number of multipliers. If a high-pass or bandpass prefilter is desired, the corresponding prefilter can be transformed from a low pass prototype RRS filter.

Another prefilter approach is the simple symmetric sharpening (SSS) structure [15]. The SSS structure improves the frequency response in both the passband and the stopband by using a filter repeatedly. Adams and Willson [27] modified the SSS structure to improve its performance. Each subfilter in the modified SSS structure is still a RRS filter, which leads to a multiplier-free prefilter.

Another prefilter proposed by Adams and Willson in [27] is Bateman-Liu filter [19], which comes from a communication technique called delta-modulation. All

the coefficients of Batenman-Liu filter have the values of 0, +1 or -1, which eliminate the multipliers in the filter. They noticed that the increased order of Batenman-Liu filter could only improve the performance of the filter slightly. Therefore, a short Batenman-Liu filter is adopted as a prefilter.

Vaidyanathan and Beitman proposed a new family of prefilters [31], based on the well-known Dolph-Chebyshev functions [5]. Therefore, the coefficients of low order Chebysev polynomial are often simple combinations of powers of two. The implementations of these prefilters are multiplier-free. Compared with RRS filters, the prefilters based on the Dolph-Chebysev functions have another advantage, i.e. the designers have more choice of the prefilter parameters because the coefficients of the prefilter is not limited to be all ones. This advantage makes the design of the prefilter become more flexible.

Lian and Lim [50] proposed a new prefilter structure based on the combination of two cosine functions. When the cosine function is negative and its square is positive, a stopband is formed. The sum of a cosine function and its square results in an acceptable stopband attenuation. Lian and Lim's prefilter can provide 18 dB stopband attenuation, which is higher than the stopband attenuation of a RRS prefilter.

Lian and Lim [53, 57, 65] also proposed a filter with a parallel structure . Here, it is referred to as a parallel prefilter. The parallel prefilter is composed of two subfilters in parallel: an odd-length filter and an even-length FIR filter. The

two subfilters are both interpolated M times and connected in parallel. One or more stopbands are formed at the frequencies where the passband magnitude of the odd-length filter is positive and the magnitude of the even-length filter is negative. The parallel prefilter can provide about 30 dB stopband attenuation.

Another approach for the prefilter-equalizer structure is to use cyclotomic polynomial (CP) filters as prefilters. Cyclotomic polynomials were originally used to simplify complex-valued computation [66], or in the development of minimum complexity circular algorithms [18]. In [33], Babic *et al.* described how to cascade cyclotomic polynomial filters (CPF's) to form a multiplier-free, linear phase filter. Kikuchi *et al.* [35,40] used CPF's to form efficient prefilters. They searched over a field of 24 eligible polynomial responses, and determined the prefilter according to the search result. However, their method is based on a trial-and-error approach leading to suboptimal designs.

To design more efficient prefilters, Hartnett and Boudreaux-Bartels [51] proposed a straightforward automated method to form efficient prefilters using CPF's. They chose the first 104 CP's which only contain the coefficients {0, 1, -1}, such that the prefilters are multiplier-free. As the root of each of these CP's are distinct, each CPF can provide unique stopband attenuation. By cascading several different CPF's, a wide range of CP prefilters are available. The RRS (used in [22,23,27]) filter is just a special case of the cyclotomic polynomial filter.

Oh and Lee proposed a mixed integer linear programming (MILP) method [29] to design CP prefilters [60, 61]. They first formulated the CP prefilter design problem as an optimization problem with a linear objective function, and solved the optimization problem by the MILP algorithm. A new approach for the design of prefilter-equalizer filter was introduced in [72], where equalizer is based on the interpolated second order polynomial (ISOP) [69] in the case of FIR or inverse of the interpolated first order polynomial (IIFOP) for the IIR. They selected optimal CP's for both the prefilter and equalizer by the method in [61].

All the materials reviewed above focus on the prefilter design. To reduce the number of multipliers in the equalizer, Cabezas and Diniz [42] introduced the concept of interpolation [28] into the design of equalizer. When an efficient prefilter is adopted, the prefilter provides enough stopband attenuation. The passband replicas of the interpolated equalizer in the stopband can be removed by the prefilter. The interpolated equalizer approach greatly reduces the number of required multipliers in the equalizer.

Another prefilter-equalizer filter proposed by Diniz and Cabezas [44] is based on the concept of "filter sharpening" developed by Kaiser and Hamming [15], which is generalized by Saramäki [34]. The equalizer is designed by sharpening identical subfilters, which are RRS filters or comb filters.

In this section, different design methods of the prefilter-equalizer filter have been reviewed. All these methods reduce the complexity of FIR filters. In next section,

another filter structure, interpolated finite impulse filter (IFIR), will be reviewed.

1.1.3 Interpolated Finite Impulse Filter Approach

The interpolated finite impulse filter (IFIR) [28] was introduced by Neuvo *et al.* . The IFIR approach yields significant savings in terms of multipliers and adders in both linear and nonlinear cases. Here, only the linear case is reviewed. The design of an IFIR filter involves two subfilters: interpolated impulse response filter $H_M(z^L)$, and the interpolator filter $G(z)$. For a given linear phase FIR filter $H_M(z)$ (this is called model filter), an interpolated impulse response filter $H_M(z^L)$ is formed by replacing each delay in $H_M(z)$ by L delays. Note that the period of $H_M(e^{jL\omega})$ is $2\pi/L$, and the passband replicas appear in the desired stopband. Any passband of $H_M(z^L)$ in $[0, \pi]$ can be used as the passband of the overall filter. The purpose of the interpolator $G(z)$ is to attenuate the undesired passband replicas of $H_M(z^L)$ to meet the desired stopband requirement. It is very important to note that the passband and transition bandwidth of the interpolated model filter are $1/L^{th}$ of the corresponding model filter $H_M(z)$. According to Equation (1.1)-(1.12), the length of a linear phase FIR filter is almost inversely proportional to the transition bandwidth. Compared with a direct design of the same transition bandwidth, the interpolated filter only requires about $1/L^{th}$ of the number of nonzero coefficients. Therefore, the numbers of required multipliers and adders are reduced approximately to $1/L^{th}$ of the original direct design. Meanwhile, the numbers of multipliers and adders in $G(z)$ are small due to $G(z)$'s relatively

wide transition bandwidth. Thus, the total number of nonzero coefficient of the IFIR filter can be reduced greatly, and thus the IFIR filter has a much lower complexity, compared with a direct conventional design.

Saramäki *et al.* [36] proposed two methods to design IFIR filters. The first method is based on Remez multiple exchange algorithm. The design method is applied to both single stage and multiple stage implementation of the interpolator $G(z)$. This method optimizes the model filter $H_M(z)$ and the interpolator $G(z)$ simultaneously. Therefore, the number of needed multipliers and adders can be reduced to the minimal value. Their second method is to derive a new interpolator structure based on RRS filters [14]. The new interpolator structure overcomes the limitations of RRS filters, such as moderate stopband attenuation. Meanwhile, the new interpolator can still keep the property of linear phase. Saramäki *et al.* [36] also analyzed the optimal conditions of these two methods.

To realize a multiplier-free interpolator, cyclotomic polynomials and B-spline functions were proposed to be used to design the interpolator. Kikuchi, *et al.* [37] proposed to design the interpolator by utilizing cyclotomic polynomials [18]. They selected 24 CP's and summarized the selected CP's as a chart diagram. The coefficients of selected CP's take the values of 0, 1 or -1 only. The maximum number of addition of each interpolator selected from their set will not exceed 10, and the number of delay elements will not exceed 12. Interpolators based on these 24 CP's can also be applied to bandpass and highpass filters.

Pang *et al.* [41,45] proposed to design the interpolator by utilizing B-spline functions. The m^{th} order interpolator is implemented by cascading m 1^{st} order interpolators. Coefficients of the interpolator take the values of 0 or 1 for low pass filter case, and 0, 1 or -1 for highpass case. For both cases, the interpolator only requires simple shifting and addition operations.

1.1.4 Frequency-Response Masking Approach

Although IFIR filters can effectively reduce the arithmetic operations of FIR filters with narrow transition bands, they are only suitable for the design of FIR filters with narrow passbands. To synthesize sharp FIR filters with arbitrary passband width, Lim [32] proposed the frequency-response masking (FRM) approach. The FRM approach utilizes a very sparse set of coefficients, and reduces the number of multipliers and adders tremendously. The price paid for the complexity reduction is a slight increase in the effective filter length. Compared with other low complexity FIR filter synthesis techniques, the FRM approach has the smallest group delay [38].

A FRM filter is composed of three subfilters: one odd-length prototype filter $H_a(z)$, and two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$. The prototype filter $H_a(z)$ is interpolated M times, namely each delay z^{-1} of $H_a(z)$ is replaced by M delays z^{-M}. The frequency response of the interpolated prototype filer $H_a(e^{jM\omega})$ is periodical with a period of $2\pi/M$. The transition bandwidth of the interpo-

lated prototype filter is reduced to $1/M^{th}$ of that of the original prototype filter. To realize any arbitrary passband width, the concept of complementary filter is utilized. The complementary filter $H_c(z^M)$ and the interpolated prototype filter $H_a(z^M)$ satisfy the complementary condition, which is $|H_a(e^{jM\omega})+H_c(e^{jM\omega})| = 1$. In actual implementation, $H_c(z^M)$ can be realized by subtracting the output of $H_a(z^M)$ from the delayed version of the input signal. Two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$ remove undesired passband replicas of $H_a(e^{jM\omega})$ and $H_c(e^{jM\omega})$, respectively in the stopband. At last, the outputs of $H_{Ma}(z)$ and $H_{Mc}(z)$ are added together, to form the output of the overall FRM filter.

Much effort has been made to reduce the complexity of FRM filters further. Based on the traditional design methods of linear programming in [32], Lim and Lian [48] analyzed the optimal conditions, where the total number of a FRM filter is the smallest is said to be optimal, for single stage and multiple stage FRM filters. They derived an equation to estimate the optimum interpolation factor M for the prototype filter, and analyzed the complexity of a K-stage design. The criterion for selecting the optimum value of K and a multistage ripple compensation technique were also presented in [48]. Chen and Lee [52,59] proposed another practical criterion to choose the interpolation factor M. As the filter length is approximately proportional to the reciprocal of the filter's transition bandwidth, Chen and Lee proposed to use the sum of reciprocals of three subfilters' transition bandwidths as the criterion to find the optimum interpolation factor M.

Besides the basic FRM structure proposed by Lim [32], many modified FRM

structures were proposed. When the interpolation factor M becomes high, the transition bandwidths of the two masking filters become sharper, which result in increasing the orders of the two masking filters. Yang *et al.* [39] introduced the concept of "frequency compression" in the design of two masking filters to overcome the difficulty mentioned above. In this approach, two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$ are interpolated N_M times, and a third low pass filter $E(z)$ removes the unwanted passband replicas of $H_{Ma}(z^{N_M})$ and $H_{Mc}(z^{N_M})$.

The masking filter factorization approach proposed by Lim and Lian [54] can further reduce the complexity of a FRM filter. The frequency responses of the two masking filters are quite similar except near the transition bands of the overall filter. This makes it possible to realize the two masking filters by a pair of relatively simple equalizers $H'_{Ma}(z)$ and $H'_{Mc}(z)$ cascaded by a common filter $H_x(z)$. The filter $H_x(z)$ realizes the common parts of the frequency responses of $H_{Ma}(z)$ and $H_{Mc}(z)$. Two equalizers $H'_{Ma}(z)$ and $H'_{Mc}(z)$ compensate the differences between the desired frequency responses of $H_{Ma}(z)$ and $H_{Mc}(z)$ and the frequency response of the common filter $H_x(z)$, respectively.

Lian *et al.* [77] proposed to replace the prototype filter by an IFIR filter. The drawback of this approach is that one masking filter may become very long. To overcome this drawback, Yang and Lian further improved the approach in [77] by introducing one more masking filter $M_c(z)$ in between the prototype filter and the two masking filters [81]. The masking filter $M_c(z)$ is interpolated L_c times, and it can extend the transition bandwidths of two masking filters.

Another important aspect of FRM filter design is to find a way to optimize FRM filters. Besides the traditional design method of utilizing linear programming in [32], Saramäki and Lim proposed a design method based on Remez exchange method [70, 83]. Chen and Lee [52, 59] proposed to use weighted least square (WLS) algorithm to design FRM filters. But, all the methods above design each subfilter separately, which can only result in suboptimum solutions.

A method which designs all the subfilters simultaneously is very likely to improve the design result of a FRM filter. Many new design methods based on nonlinear optimization techniques were proposed to jointly optimize the three subfilters in a FRM filter. These methods include Saramäki's two-step method [75,80,93], Yu and Lim's weighted least square (WLS) approach [79], semidefinite programming (SDP) design method [91] and second-order cone programming (SoCP) design method [92, 99] proposed by Lu and Hinamoto, and the WLS approach [101] proposed by Lee *et al.*. All these design methods mentioned above can result in the reduction of the numbers of multipliers and adders compared with the iterative design methods.

In this section, different structures and design methods of FRM filters have been reviewed. These structures and design methods further improve the efficiency of FRM filters.

1.2 Outline

The rest of this thesis is organized as follows.

Chapter 2: The parallel prefilter is first reviewed. A design method based on weighted least square (WLS) criterion is proposed to jointly optimize the parallel prefilters and its equalizers. Design examples show that the proposed design method yields more savings of multipliers and adders.

Chapter 3: The design problem of a basic parallel filter is first formulated as a goal attainment problem. By analyzing the effects of different parameters, formulas and a table are presented to estimate the lengths of subfilters in a basic parallel filter. Accuracy of the presented formulas and table are shown to be satisfactory.

Chapter 4: A suitable interpolation factor M can effectively reduce the complexity of a FRM filter. Based on some new observations, a new set of equations is derived to find subfilter lengths and the optimum value of M for jointly optimized FRM filters.

Chapter 5: Problems are first pointed out if the prototype filter in a FRM filter is even-length. A new design method is proposed to design FRM filters, including FRM filters utilizing even-length prototype filters. New structures are proposed to utilize odd interpolation factor M with an even-length prototype filter. Design examples show that an even-length prototype filter may result in more savings of multipliers and adders, compared with an odd-length prototype filter.

Chapter 6: Factors affecting the density of frequency grid points in the design of FRM filters are first analyzed. A new dynamic frequency grid point allocation scheme is proposed to save required memory and computation time. A design example shows the efficiency of the new frequency grid point allocation scheme.

It should be pointed out that all the filters discussed are all symmetric FIR filters in this thesis.

1.3 Statement of Originality

The following items are claimed to be original.

1. The weighted least square (WLS) design method for parallel filters (Chapter 2).

2. Formulas to estimate subfilter lengths in a jointly optimized basic parallel filter (Chapter 3).

3. Formulas to estimate subfilter lengths and the optimal interpolation factor M in a jointly optimized FRM filter (Chapter 4).

4. Utility of even-length filter as the prototype filter in a FRM filter and corresponding new filter structures (Chapter 5).

5. Dynamic frequency grid point allocation scheme for FRM filter design (Chapter 6).

1.4 List of Publications

1. Y. Lian and J. H. Yu, "VLSI implementation of multiplier-free low power baseband filter for CDMA systems," *Proceedings of IEEE Workshop on Signal Processing Systems*, pp. 111-115, Seoul, Korea, Aug. 2003.

2. J. H. Yu and Y. Lian, "Frequency-response masking based filters with even-length bandedge shaping filter," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 536-539, Vancouver, Canada, May 2004.

3. Y. Lian and J. H. Yu, "The synthesis of low power baseband filters for CDMA systems," *Proceedings of IEEE 6th CAS Workshop/Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, pp. 599-602, Shanghai, China, May 2004.

4. Y. Lian and J. H. Yu, "The reduction of noises in an ECG signal using frequency response masking based FIR filters," *Proceedings of IEEE Workshop on BioMedical Circuits and Systems*, pp. s2.4-17-20, Singapore, Dec. 2004.

5. J. H. Yu and Y. Lian, "Interpolation factor analysis for jointly optimized frequency-response masking filters," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 2016-2019, Kobe, Japan, May 2005.

6. Y. Lian and J. H. Yu, "A low power linear phase digital FIR filter for wearable ECG devices," *Proceedings of The 27th Annual International Confer-*

ence of the IEEE Engineering in Medicine and Biology Society, pp. 7357 - 7360, Singapore, Sep. 2005.

7. J. H. Yu and Y. Lian, "Design equations for jointly optimized frequency-response masking filters," *Circuits Systems And Signal Processing*, under revision.

8. Y. Lian and J. H. Yu, "The design of FIR filters based on parallel structure," under preparation.

9. J. H. Yu and Y. Lian, "Order estimation of the parallel structure FIR digital filters," under preparation.

10. J. H. Yu and Y. Lian, "A dynamic frequency grid point allocation scheme for the efficient design of frequency-response masking FIR filters," under preparation.

11. Y. Lian and J. H. Yu, "Optimal design of frequency-response-masking filters with even-length prototype filters using sequential quadratic programming," under preparation.

Chapter 2

Filter Design Based on Parallel Prefilter

In this chapter, FIR filter structures based on the parallel prefilter and the iterative design method are first reviewed. The weighted least square (WLS) method is presented for the design of parallel prefilter and its equalizer. This method optimizes the coefficients of subfilters simultaneously, and further improves the efficiency of the parallel prefilter and its equalizer.

2.1 Introduction

In the past decades, many methods have been proposed to reduce the complexity of digital filters [22, 23, 28, 32, 48, 58, 70, 83]. Among them, the prefilter-equalizer

structure [22, 23] and interpolated finite impulse response (IFIR) filter [28] are computationally efficient for the synthesis of sharp FIR filters with narrow passband. For a sharp filter with arbitrary passband bandwidth, frequency-response masking (FRM) approach is one of the most efficient techniques [32, 54, 58, 83], at the expense of slightly increased filter order.

The parallel FIR filter proposed in [53, 57, 65] is suitable for both narrow and moderately wide transition bandwidth, by utilizing the combination of an even-length and an odd-length symmetric FIR filter as a prefilter, which provides higher attenuation in the stopband while having less distortion in the passband, compared with an RRS prefilter in [22, 23].

To further improve the efficiency of a parallel filter, a joint optimization method is highly desired. In this chapter, a new design method based on WLS is proposed to jointly optimize the parallel prefilter and its equalizer. Design examples show that the proposed method leads to more savings in terms of arithmetical operations compared with the original iterative method.

The organization of this chapter is as follows. In Section 2.2, filter structures based on the parallel prefilter are first reviewed. The WLS design method is presented in Section 2.3 for the design of the parallel prefilter and its equalizer. Design examples are given in Section 2.4, and a conclusion is drawn in Section 2.5.

2.2 Structures Based on Parallel Prefilter

2.2.1 Parallel Prefilter

Consider a first order even-length linear-phase FIR filter with the z-transform transfer function $H_e(z) = 1 + z^{-1}$. Its zero-phase frequency response can be written as

$$H_e(e^{j\omega}) = \cos\left(\frac{\omega}{2}\right). \tag{2.1}$$

The linear phase term has been dropped for the sake of expository clarity throughout this chapter. Since the $H_e(z)$ is an even-length filter, its interpolated frequency response $H_e(e^{j2\omega})$ will have a positive value in the normalized frequency interval $[0, 0.5\pi]$ and a negative value in the interval $[0.5\pi, \pi]$ as shown in Figure 2.1(a). Let $H_o(z)$ to be another odd-length filter that is designed such that its interpolated frequency response $H_o(e^{j2\omega})$ approximates $H_e(e^{j2\omega})$ in the interval $[0, 0.5\pi]$, $H_o(e^{j2\omega})$ will have a positive frequency component in the interval $[0.5\pi,\ \pi]$ that approximates $|H_e(e^{j2\omega})|$, as shown in Figure 2.1(b). Since

$$\begin{aligned} H_o\left(e^{j2(\pi-\omega)}\right) &= h\left(\frac{N_o+1}{2}\right) + 2\sum_{n=1}^{(N_o-1)/2} h(n)\cos\left((N_o-2n+1)(\pi-\omega)\right) \\ &= h\left(\frac{N_o+1}{2}\right) + 2\sum_{n=1}^{(N_o-1)/2} h(n)\cos\left((N_o-2n+1)\omega\right) \\ &= H_o\left(e^{j2\omega}\right), \end{aligned} \tag{2.2}$$

$H_o(e^{j2\omega})$ is positive in $[0.5\pi,\ \pi]$. In a similar way, we can have that

$$H_e\left(e^{j2(\pi-\omega)}\right) = -H_e\left(e^{j2\omega}\right), \tag{2.3}$$

which means that $H_e(e^{j2\omega})$ is negative in the interval $[0.5\pi,\ \pi]$. Therefore, a stopband can be created in the interval $[0.5\pi,\ \pi]$ by connecting $H_e(z^2)$ and $H_o(z^2)$ in parallel, as shown in Figure 2.1(c). Note that additional delay elements must be added to $H_e(z^2)$ to make sure that the group delay of $H_e(z^2)$ and $H_o(z^2)$ are the same.

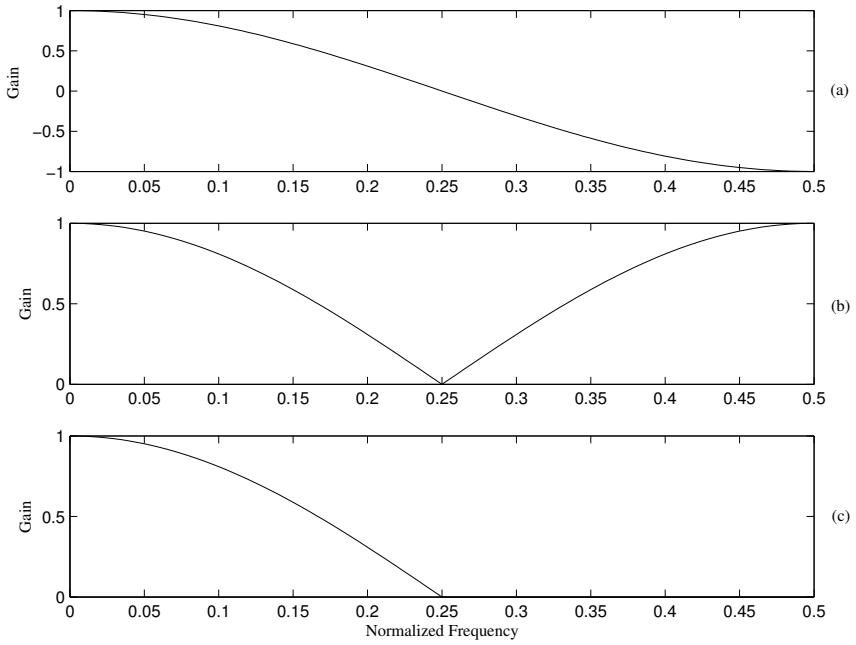

Figure 2.1: The frequency responses of (a) $H_e(z^2)$, (b) $H_o(z^2)$ and (c) $H_p(z)$

.

The z-transform transfer function of the new filter $H_p(z)$ is given by

$$H_p(z) = \frac{1}{2}\left[z^{-\frac{N_{oi}-3}{2}} H_e(z^2) + H_o(z^2)\right] \tag{2.4}$$

where N_{oi} is the filter length of $H_o(z^2)$. The new filter $H_p(z)$ is called as a parallel prefilter, because it consists of two subfilters in parallel.

With the parallel prefilter $H_p(z)$, a prefilter-equalizer based filter can be formed

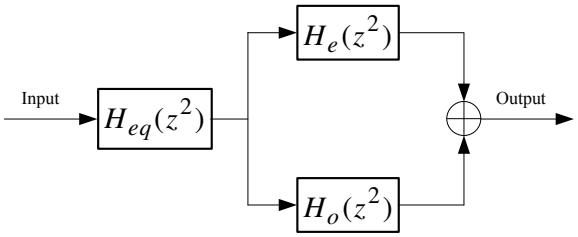

Figure 2.2: A realization structure for a basic parallel filter

as

$$H(z) = H_{eq}(z^2)\left[z^{-\frac{N_{oj}-3}{2}}H_e(z^2) + H_o(z^2)\right]. \tag{2.5}$$

The above z-transform transfer function has a gain of two. To normalize it, the coefficients of $H_{eq}(z)$ should be divided by 2. Note that the equalizer $H_{eq}(z)$ is interpolated by a factor of 2 in Equation (2.5). This is to reduce the complexity of $H_{eq}(z)$ as the stopband attenuation of $H_p(z)$ is large enough to get rid of the passband replica of $H_{eq}(z)$ in $[\pi-\omega_s, \pi]$, where ω_s is the stopband edge of $H_{eq}(z^2)$. Figure 2.2 shows one of the possible implementation structures for the proposed filter. It should be pointed out that additional delays should be added to $H_e(z^2)$ to keep the phase frequency responses of $H_e(z^2)$ and $H_o(z^2)$ the same.

From now on, a FIR filter composed of a parallel prefilter and its equalizer interpolated by a factor of 2 is referred as a basic parallel filter. In next section, an iterative design method will be reviewed, which is for the design of a basic parallel filter.

2.2.2 Iterative Design Method for Parallel Prefilter

To form a basic parallel filter, two subfilters are needed, i.e. $H_o(z)$ and the equalizer $H_{eq}(z)$. It is a well known fact that an even-length symmetric FIR filter contains a zero at $z = -1$, i.e. we can always decompose an even-length prototype filter into two filters: $H_e(z) = (1 + z^{-1})$ and $H_{eq}(z)$. It is noted that the frequency response of $H_p(z)$ is very close to that of $H_e(z^2)$ in the passband. Hence, the design of the basic parallel filter can start with the design of an even-length prototype filter $H_{pro}(z)$ with the passband and stopband edges at 2 times of desired passband and stopband edges. The passband and stopband ripples of $H_{pro}(z)$ are the same as the desired filter. Factorizing the $1 + z^{-1}$ from $H_{pro}(z)$, we have

$$H_{pro}(z) = (1 + z^{-1})H_{eq}(z) = H_e(z)H_{eq}(z). \tag{2.6}$$

The $H_e(z)$ and $H_{eq}(z)$ in (2.6) can be used as initial prefilters to obtain $H_o(z)$. Let the frequency response of $H_o(z)$ be expressed in the form of a Fourier series,

$$H_o(e^{j\omega}) = \sum_n \alpha(n) Trig(n, \omega) \tag{2.7}$$

where $\alpha(n)$ is the n^{th} Fourier series coefficient and $Trig(n, \omega)$ is a trigonometric function depending on the type of filter used. Substituting (2.7) into (2.5), we have

$$H(e^{j\omega}) = \frac{1}{2} H_{eq}(e^{j2\omega}) \left[\cos(\omega) + \sum_n \alpha(n) Trig(n, 2\omega) \right]. \tag{2.8}$$

Let $\delta(\omega)$ be the maximum allowable frequency response deviation at frequency ω. Two functions can be defined as follows:

$$\delta_p^{\pm} = 1 - \frac{1}{2}\cos(\omega)H_{eq}\left(e^{2j\omega}\right) \pm \delta(\omega) \tag{2.9}$$

$$\delta_s^{\pm} = -\frac{1}{2}\cos(\omega)H_{eq}\left(e^{2j\omega}\right) \pm \delta(\omega). \tag{2.10}$$

In the passband, we have

$$\delta_p^{-}(\omega) \leqslant \frac{1}{2}H_{eq}\left(e^{2j\omega}\right)\sum_n \alpha(n)Trig(n,2\omega) \leqslant \delta_p^{+}(\omega). \tag{2.11}$$

In the stopband, we have

$$\delta_s^{-}(\omega) \leqslant \frac{1}{2}H_{eq}\left(e^{2j\omega}\right)\sum_n \alpha(n)Trig(n,2\omega) \leqslant \delta_s^{+}(\omega). \tag{2.12}$$

Linear programming [25], modified Remez exchange method [11,23], or any other suitable technique may be used to solve (2.11) and (2.12). Once $H_o(z)$ is designed, $H_{eq}(z)$ should be optimized again to reduce the overall complexity further. This can be done by swapping the role of $H_o(z)$ with $H_{eq}(z)$ in the above design procedure.

2.2.3 Filter Structures Based on Parallel Prefilter

In a basic parallel filter, all subfilters are interpolated by 2. This structure can be generalized to any even interpolation factor. Assuming that all the subfilters in Figure 2.2 are interpolated by a factor of M, we have

$$H(z) = 0.5[H_o(z^M) + z^{-\frac{(No-Ne)M}{2}}H_e(z^M)]H_{eq}(z^M) \tag{2.13}$$

where N_e and N_o are the lengths of $H_e(z)$ and $H_o(z)$, respectively, M must be an even number to avoid half sample delay. At the same time, N_o should be larger than N_e. If $N_e > N_o$, we have

$$H(z) = 0.5[z^{-\frac{(Ne-No)M}{2}} H_o(z^M) + H_e(z^M)]H_{eq}(z^M). \tag{2.14}$$

From now on, we call this generalized filter and related filter structures as parallel filters. Note in the generalized parallel filters, $H_e(z)$ is not necessary a 1^{st}-order filter.

In (2.13) and (2.14), The frequency responses are shown in Figures 2.3(a), (b) and (c). To form the overall filter, a masking filter $H_{msk}(z)$ is needed to remove unwanted passband replicas in the stopband, as shown in Figure 2.3(c). The transfer function of the overall filter is

$$H(z) = 0.5[z^{-\frac{(Ne-No)M}{2}} H_o(z^M) + H_e(z^M)]H_{eq}(z^M)H_{msk}(z) \tag{2.15}$$

or

$$H(z) = 0.5[H_o(z^M) + z^{-\frac{(No-Ne)M}{2}} H_e(z^M)]H_{eq}(z^M)H_{msk}(z) \tag{2.16}$$

depending on the lengths of $H_o(z)$ and $H_e(z)$. Figure 2.4(a) shows one realization structure of the overall filter $H(z)$.

The structure shown in Figure 2.4(a) is suitable for the synthesis of the narrow passband case. If a wider passband is desired, Figure 2.4(b) suggests a possible solution. Now $H_o(z^M)$ and $H_e(z^M)$ are still in parallel, and the output of $H_e(z^M)$ is subtracted from the output of $H_o(z^M)$. If $H_{eq}(z)$ is odd-length, its output is

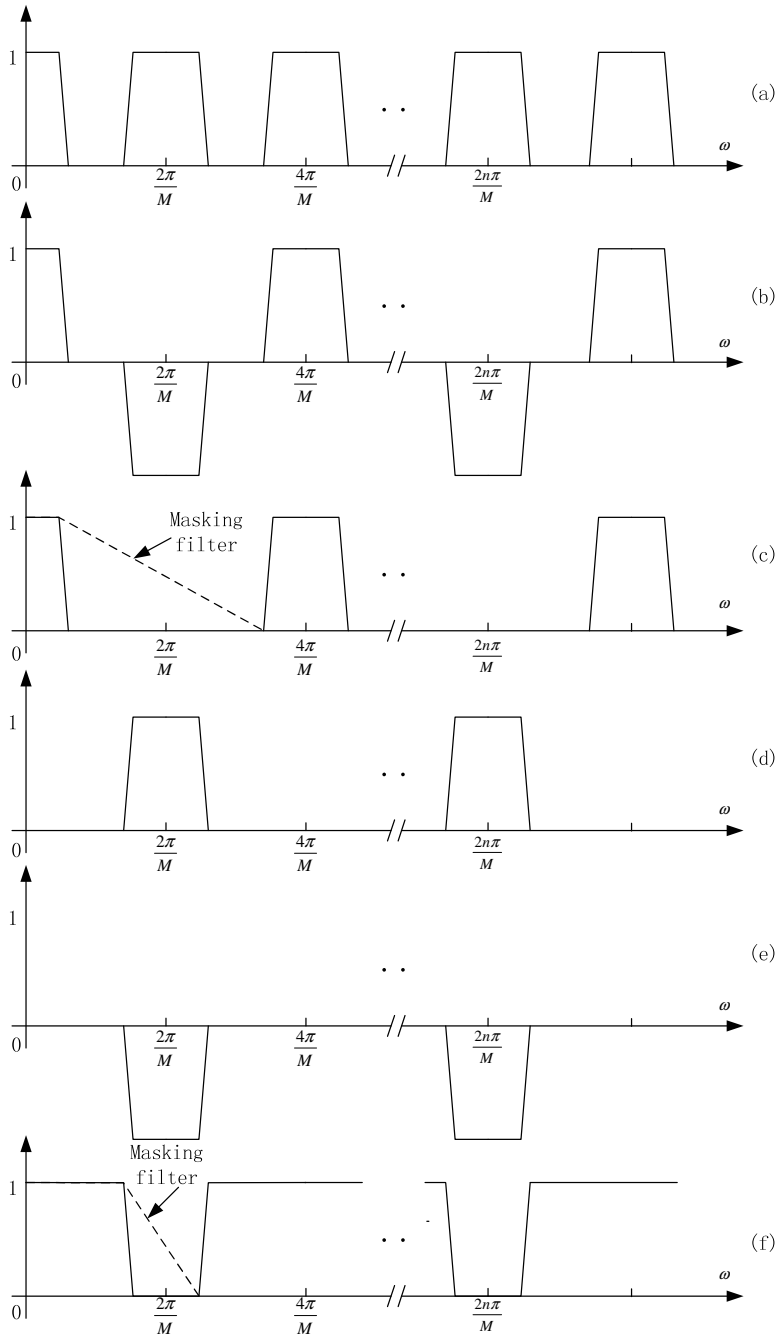

Figure 2.3: Frequency responses of (a) $H_o(e^{jM\omega})$, (b) $H_e(e^{jM\omega})$, (c) $0.5[H_o(e^{jM\omega}) + H_e(e^{jM\omega})]$, (d) $0.5[H_o(e^{jM\omega}) - H_e(e^{jM\omega})]$, (e) $0.5[H_e(e^{jM\omega}) - H_o(e^{jM\omega})]$ and (f) $1 - 0.5[(H_e(z^M) - H_o(z^M))]$

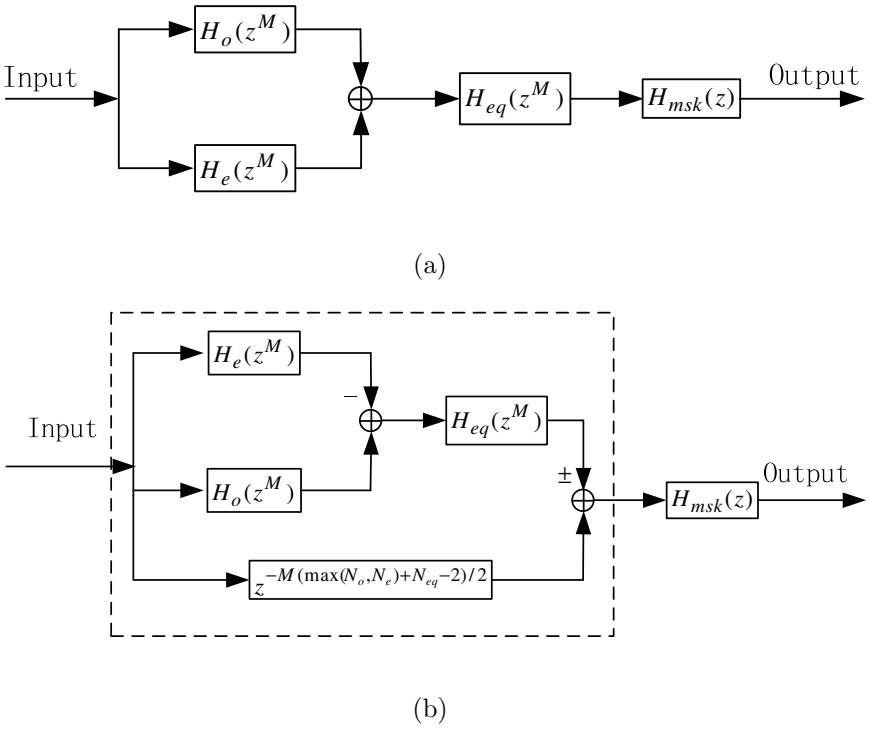

Figure 2.4: Realization structures for (a) Equation (2.15) and (2.16), and (b) Equation (2.17)

subtracted from the delayed input signal. Otherwise, its output is added with the delayed version of the input signal. The transfer function of the overall filter is

$$H(z) = [z^{-\frac{M(\max(No,Ne)+Neq-2)}{2}} \pm H_{pw}(z)H_{eq}(z^M)]H_{msk}(z) \tag{2.17}$$

where

$$H_{pw}(z) = 0.5[z^{-\frac{(Ne-No)M}{2}}H_o(z^M) - H_e(z^M)] \tag{2.18}$$

or

$$H_{pw}(z) = 0.5[H_o(z^M) - z^{-\frac{(No-Ne)M}{2}}H_e(z^M)] \tag{2.19}$$

depending on the lengths of $H_e(z)$ and $H_o(z)$, and N_{eq} is the length of the equalizer $H_{eq}(z)$. Note that additional delay elements must be added to $H_o(z^M)$ or $H_e(z^M)$ to make sure the group delay of $H_o(z^M)$ and $H_e(z^M)$ are the same in Figure 2.4.

In this section, different filter structures are reviewed. These structures are all based on the parallel prefilter. They can be utilized to design FIR filters with a narrow or moderately wide transition band. In next section, a WLS design method will be presented to optimize all the subfilters simultaneously.

2.3 Weighted Least Square Design Method for Parallel Prefilter-Equalizer

In this section, the discussion is limited to the design of a basic parallel filter reviewed in Section 2.2.1. Other structures reviewed in Section 2.2.3 can be designed in a similar way, only slight modifications are needed.

2.3.1 Design Problem Formulation

The parallel filter in Section 2.2.1 uses an even-length subfilter $H_e(z)$ of length 2. The design starts with an even-length prototype filter $H_{pro}(z)$ with passband and stopband bandedges set at two times of the given specifications. The equalizer $H_{eq}(z)$ is obtained by factorizing $H_e(z) = 1+z^{-1}$ from the prototype filter $H_{pro}(z)$. $H_o(z)$ is determined using $H_e(z)$ and $H_{eq}(z)$ as prototype filters. Once $H_o(z)$ is decided, an iterative procedure is employed to further optimize $H_e(z)$ and $H_{eq}(z)$ by minimizing the ripples in the passband and stopband, one at a time, in order to reduce the overall complexity. In each step of the above procedure, only one subfilter is designed, and linear programming [25] can handle this problem.

However, the iterative procedure does not fully explore the property of a parallel prefilter. The advantage of combining an even-length and an odd-length subfilter to form a prefilter $H_p(z)$ is the ability to create a stopband with sufficient attenuation. The attenuation depends largely on the similarity of $|H_o(e^{j2\omega})|$ and

$|H_e(e^{j2\omega})|$ in the normalized frequency interval $[0.5\pi, \pi]$. Because $H_e(z)$ is fixed to a first order linear phase FIR filter, the filter length of $H_o(z)$ goes up if a large stopband attenuation is required. The difference between lengths of $H_o(z)$ and $H_e(z)$ can be quite large in some designs.

It is easy to verify that frequency responses of the even and odd-length filters with close lengths can be very similar, except for the cases where the normalized band-edges are close to π. Based on this observation, it is reasonable to believe that the length of $H_o(z)$ is likely to be reduced if the length of $H_e(z)$ increases from 2 to some extent. This may result in complexity reduction and increased stopband attenuation for the prefilter. However, the increased order in $H_e(z)$ makes the design of overall filter more difficult, as the reviewed design procedures in Section 2.2.2 are no longer applicable to the new parallel prefilter. A global optimization of the overall filter requires the design of subfilters $H_e(z)$, $H_o(z)$ and the equalizer $H_{eq}(z)$ to be done simultaneously. One algorithm which can solve this problem is the WLS algorithm. In the rest of this section, a detailed WLS design method is presented.

Suppose N_e is the length of the even-length filter $H_e(z)$, and N_o is the length of the odd-length filter $H_o(z)$. When N_e is not fixed to be two, the transfer function of the prefilter $H_p(z)$ is represented by Equations (2.13) or (2.14). Correspondingly, the transfer function of the overall filter is

$$H(z) = H_{eq}(z^2)\left[z^{-(N_o-N_e)}H_e(z^2) + H_o(z^2)\right] \text{ if } N_o > N_e. \tag{2.20}$$

$$H(z) = H_{eq}(z^2)\left[H_e(z^2) + z^{-(N_e-N_o)}H_o(z^2)\right] \text{ if } N_e > N_o. \tag{2.21}$$

As in Section 2.2.1, the coefficients of $H_{eq}(z)$ should be divided by 2, to normalize the gain of $H(z)$ in (2.20) and (2.21).

When extra delay elements are added and the phase frequency responses of $H_e(z^2)$ and $H_o(z^2)$ are the same, we can drop the phrase terms, and the zero-phase frequency response of the overall filter is

$$H(e^{j\omega}) = H_{eq}(e^{2j\omega})[H_e(e^{2j\omega}) + H_o(e^{2j\omega})]. \tag{2.22}$$

Suppose the coefficient vector of $H_o(z)$ is $\boldsymbol{h}_o$. The frequency response of $H_o(z^2)$ can be expressed as

$$H_o\left(e^{2j\omega}\right) = h_o\left(\frac{N_o+1}{2}\right) + 2\sum_{i=1}^{\frac{No-1}{2}} h_o(i)\cos[(N_o-2i+1)\omega]. \tag{2.23}$$

If a vector $\boldsymbol{g}_o$ is defined as

$$\boldsymbol{g}_o = \left[2h_o(1)\ 2h_o(2)\ \cdots\ 2h_o\left(\frac{N_o-1}{2}\right)\ h_o\left(\frac{N_o+1}{2}\right)\right]^T, \tag{2.24}$$

we have

$$H_o\left(e^{2j\omega}\right) = \boldsymbol{g}_o{}^T[\cos((N_o-1)\omega)\ \cos((N_o-3)\omega)\ \cdots\ \cos(\omega)\ \ 1]^T. \tag{2.25}$$

Similarly, we can define another vector $\boldsymbol{g}_e$ as

$$\boldsymbol{g}_e = \left[2h_e(1)\ 2h_e(2)\ \cdots\ 2h_e\left(\frac{N_e}{2}\right)\right]^T \tag{2.26}$$

where $\boldsymbol{h}_e$ is the coefficient vector of $H_e(z)$. So

$$H_e(e^{2j\omega}) = \boldsymbol{g}_e{}^T[\cos((N_e-1)\omega)\ \cos((N_e-3)\omega)\ \cdots\ \cos(\omega)]^T. \tag{2.27}$$

If the equalizer $H_{eq}(z)$ is odd-length and its coefficient vector is $\boldsymbol{h}_{eq}$, then a vector $\boldsymbol{g}_{eq}$ is defined as

$$\boldsymbol{g}_{eq} = \left[2h_{eq}(1)\ 2h_{eq}(2)\ \cdots\ 2h_{eq}\left(\frac{N_{eq}-1}{2}\right)\ h_{eq}\left(\frac{N_{eq}+1}{2}\right)\right]^T. \tag{2.28}$$

So the frequency response of the interpolated equalizer $H_{eq}(z^2)$ is

$$H_{eq}\left(e^{2j\omega}\right) = \boldsymbol{g}_{eq}{}^T\left[\cos((N_{eq}-1)\omega)\ \cos((N_{eq}-3)\omega)\ \cdots\ \cos(\omega)\ \ 1\right]^T. \tag{2.29}$$

Substituting (2.25), (2.27) and (2.29) into (2.22), we have the frequency response of the overall system

$$\begin{aligned} H(e^{j\omega}) \ = \ & \{\boldsymbol{g}_o{}^T[\cos((N_o-1)\omega)\ \cos((N_o-3)\omega)\ \cdots\ \cos(\omega)\ 1]^T + \\ & \boldsymbol{g}_e{}^T[\cos((N_e-1)\omega)\ \cos((N_e-3)\omega)\ \cdots\ \cos(\omega)]^T\}\cdot \\ & \{\boldsymbol{g}_{eq}{}^T\left[\cos[(N_{eq}-1)\omega]\ \cos[(N_{eq}-3)\omega]^T\ \cdots\ \cos(\omega)\ 1\right]^T\}. \end{aligned} \tag{2.30}$$

If the equalizer $H_{eq}(z)$ is even-length, the definition of $\boldsymbol{g}_{eq}$ is

$$\boldsymbol{g}_{eq} = \left[2h_{eq}(1)\ 2h_{eq}(2)\ \cdots\ 2h_{eq}\left(\frac{N_{eq}}{2}\right)\right]^T. \tag{2.31}$$

Equation (2.29) is changed to

$$H_{eq}(e^{2j\omega}) = \boldsymbol{g}_{eq}{}^T[\cos((N_{eq}-1)\omega)\ \cos((N_{eq}-3)\omega)\ \cdots\ \cos(\omega)]^T. \tag{2.32}$$

Equation (2.30) is changed to

$$\begin{aligned} H(e^{j\omega}) \;=\; & \{\boldsymbol{g}_o{}^T[\cos((N_o-1)\omega)\ \cos((N_o-3)\omega)\ \cdots\ \cos(\omega)\ 1]^T + \\ & \boldsymbol{g}_e{}^T[\cos((N_e-1)\omega)\ \cos((N_e-3)\omega)\ \cdots\ \cos(\omega)]^T\}\cdot \\ & \{\boldsymbol{g}_{eq}{}^T\left[\cos[(N_{eq}-1)\omega]\ \cos[(N_{eq}-3)\omega]^T\ \cdots\ \cos(\omega)\right]^T.\} \end{aligned} \tag{2.33}$$

The design vector is defined as

$$\boldsymbol{g} = [\boldsymbol{g}_o{}^T\ \ \boldsymbol{g}_e{}^T\ \ \boldsymbol{g}_{eq}{}^T]^T. \tag{2.34}$$

Therefore, the frequency response of the overall FIR filter $H(e^{j\omega})$ is a function of $\boldsymbol{g}$.

The design of the parallel prefilter and its equalizer is carried out on a dense set of grid points in the passband and stopband. For any grid point $\omega_k \in [0,\ 2\pi f_p] \cup [2\pi f_s,\ \pi]$, the error function $E(\omega_k)$ can be defined as

$$E(\omega_k) = H\left(e^{j\omega_k}\right) - H_d\left(e^{j\omega_k}\right) \tag{2.35}$$

where

$$H_d\left(e^{j\omega_k}\right) = \begin{cases} 1 & \omega_k \in [0,\ 2\pi f_p] \\ 0 & \omega_k \in [2\pi f_s,\ \pi]. \end{cases} \tag{2.36}$$

The WLS objective function is defined as

$$f(\boldsymbol{g}) = \sum_k W(\omega_k)E^2(\omega_k) \tag{2.37}$$

where $W(\omega_k)$ is a weighting function. The design task is to minimize $f(\boldsymbol{g})$ in Equation (2.37) .

2.3.2 BFGS Iterative Procedure

The design begins with the determination of a set of frequency grid points. After a set of suitable grid points in the frequency domain is determined, the solution of the WLS objective function can be determined by any nonlinear unconstrained design method. There are two often used methods. One is the Davidon-Fletcher-Powell (DFP) method [26]. The solution of DFP method converges to the inverse of Hessian, where Hessian is a matrix in Taylor's expansion [78]. A more direct way is Broydon-Fletcher-Goldfarb-Shanno (BFGS) method [30] because the BFGS method converges to the Hessian itself. Here, the BFGS method is utilized to minimize $f(\boldsymbol{g})$ in (2.37).

The design procedure is given below:

Step 1: Determine the initial value of $\boldsymbol{g}_0$ by designing each subfilter separately using Remez exchange method [11], or linear programming [25]. Determine the maximum iteration limitation N, and tolerances for stopping criteria $\varepsilon_1, \varepsilon_2$ and ε_3. Set the initial matrix $\boldsymbol{A}_0$ to be a unit matrix $\boldsymbol{I}$, and the iteration counter i=0.

Step 2: Determine the search direction as

$$\boldsymbol{A}_i \boldsymbol{s}_i = -\nabla f(\boldsymbol{g_i}) \tag{2.38}$$

and update the design vector

$$\boldsymbol{g}_{i+1} = \boldsymbol{g}_i + \alpha_i \boldsymbol{s}_i \tag{2.39}$$

where $\boldsymbol{s}_i$ is the search direction, and α_i is the step size determined by

$$\min_{\alpha_i} f(\boldsymbol{g}_{i+1}) = f(\boldsymbol{g}_i) + \alpha_i \boldsymbol{s}_i. \tag{2.40}$$

Details will be presented in Section 2.3.3 about the determination of α_i.

Step 3: Stop if one of the conditions below is satisfied:

1. $\nabla f(\boldsymbol{g}_{i+1})^T \nabla f(\boldsymbol{g}_{i+1}) \leqslant \varepsilon_3$ (converged)

2. $|f(\boldsymbol{g}_{i+1}) - f(\boldsymbol{g}_i)| \leqslant \varepsilon_2$ (function not changing)

3. $(\boldsymbol{g}_{i+1} - \boldsymbol{g}_i)^T (\boldsymbol{g}_{i+1} - \boldsymbol{g}_i) \leqslant \varepsilon_1$ (design not changing)

4. $i > N$ (maximum iteration limitation exceeded)

Step 4: Update metric $\boldsymbol{A}_i$ and i by

$$\begin{aligned}
\boldsymbol{Y} &= \nabla f(\boldsymbol{g}_{i+1}) - \nabla f(\boldsymbol{g}_i) && (2.41)\\
\boldsymbol{B} &= \frac{\boldsymbol{Y}\boldsymbol{Y}^T}{\boldsymbol{Y}^T(\boldsymbol{g}_{i+1} - \boldsymbol{g}_i)} && (2.42)\\
\boldsymbol{C} &= \frac{\nabla f(\boldsymbol{g}_i) \nabla f(\boldsymbol{g}_i)^T}{\nabla f(\boldsymbol{g}_i)^T \boldsymbol{s}_i} && (2.43)\\
\boldsymbol{A}_{i+1} &= \boldsymbol{A}_i + \boldsymbol{B} + \boldsymbol{C} && (2.44)\\
i &= i + 1 && (2.45)
\end{aligned}$$

And go to Step 2.

When one iteration is converged, the solution is not equiripple. To obtain an equiripple solution, the weighting function $W(\omega_k)$ needs updating after each it-

eration. The initial value of $W(\omega_k)$ can be set to

$$W(\omega_k) = \begin{cases} 1 & \omega_k \in [0, 2\pi f_p] \\ \delta_p/\delta_s & \omega_k \in [2\pi f_s, \pi] \end{cases} \tag{2.46}$$

and $W(\omega_k)$ is updated by using the Lim-Lee-Chen-Yang algorithm in [47]. After $W(\omega_k)$ is updated, Step 1 to 4 are repeated until an equiripple solution is obtained.

2.3.3 Gold Section Method

In Equations (2.40) and (2.39), α_i is determined by the Gold Section Method. The detailed steps are given below [78]:

Step 1: Choose the lower and upper bounds of α_i, i.e. α_i^{low} and, tolerance $\varepsilon = (\Delta\alpha_i)_{final}/(\alpha_i^{up} - \alpha_i^{low})$,
$\tau = (\sqrt{5} - 1)/2$, and number of iteration $N = -2.078 \ln \varepsilon$. Let the iteration number i be equal to 1. α_i^{low} can be set to 0, and α_i^{up} can have the value of 1.

Step 2: $\alpha_1 = \tau\alpha_i^{low} + (1-\tau)\alpha_i^{up}; f_1 = f(\alpha_1);$
$\alpha_2 = (1-\tau)\alpha_i^{low} + \tau\alpha_i^{up}; f_2 = f(\alpha_2);$

Step 3: if $i > N$, quit;

Step 4: if $f_1 > f_2$,

$$\alpha_i^{low} = \alpha_1; \alpha_1 = \alpha_2; f_1 = f_2$$

$\alpha_2 = (1-\tau)\alpha_i^{low} + \tau\alpha_i^{up}; f_2 = f(\alpha_2)$

$i = i + 1$

Go to Step 3;

2.3.4 Analytical Calculation of Derivatives

In Equations (2.38), (2.41) and (2.43) the derivative of $\nabla f(\boldsymbol{g})$ is needed. To improve the efficiency of the iterative procedure in Section 2.3.2, the analytical calculation $\nabla f(\boldsymbol{g})$ is desired. This is because analytical calculation saves much more calculation time, compared with the numerical way. In this section, the analytical expression of $\nabla f(\boldsymbol{g})$ is derived.

Suppose the equalizer $H_{eq}(z)$ is odd-length. According to Equation (2.37), the derivative of $f(\boldsymbol{g})$ should be

$$\begin{aligned}
\nabla f(\boldsymbol{g}) &= 2\sum_k W(\omega_k)E(\omega_k)\nabla_{\boldsymbol{g}}E(\omega_k) \\
&= 2\sum_k W(\omega_k)\left[H(\omega_k) - H_d(\omega_k)\right]\nabla_{\boldsymbol{g}}\left[H(\omega_k) - H_d(\omega_k)\right] \\
&= 2\sum_k W(\omega_k)\left[H(\omega_k) - H_d(\omega_k)\right]\begin{pmatrix} \nabla_{\boldsymbol{g}_o}H(\omega_k) \\ \nabla_{\boldsymbol{g}_e}H(\omega_k) \\ \nabla_{\boldsymbol{g}_{eq}}H(\omega_k) \end{pmatrix}. \qquad (2.47)
\end{aligned}$$

According to Equations (2.25) and (2.30),

$$\nabla_{\boldsymbol{g}_o} H(\omega_k) = H_{eq}(e^{2j\omega_k}) \begin{pmatrix} \cos((N_o-1)\omega_k) \\ \cos((N_o-3)\omega_k) \\ \vdots \\ \cos(\omega_k) \\ 1 \end{pmatrix}. \tag{2.48}$$

Similarly,

$$\nabla_{\boldsymbol{g}_e} H(\omega_k) = H_{eq}(e^{2j\omega_k}) \begin{pmatrix} \cos((N_e-1)\omega_k) \\ \cos((N_e-3)\omega_k) \\ \vdots \\ \cos(\omega_k) \end{pmatrix}. \tag{2.49}$$

If the equalizer $H_{eq}(z)$ is odd-length,

$$\nabla_{\boldsymbol{g}_{eq}} H(\omega_k) = \left(H_o(e^{2j\omega_k}) + H_e(e^{2j\omega_k})\right) \begin{pmatrix} \cos((N_{eq}-1)\omega_k) \\ \cos((N_{eq}-3)\omega_k) \\ \vdots \\ \cos(\omega_k) \\ 1 \end{pmatrix}. \tag{2.50}$$

Otherwise,

$$\nabla_{\boldsymbol{g}_{eq}} H(\omega_k) = \left(H_o(e^{2j\omega_k}) + H_e(e^{2j\omega_k})\right) \begin{pmatrix} \cos((N_{eq}-1)\omega_k) \\ \cos((N_{eq}-3)\omega_k) \\ \vdots \\ \cos(\omega_k) \end{pmatrix}. \tag{2.51}$$

If the above WLS design method is applied to the filter structures in Section 2.2.3, the masking filter $H_{msk}(z)$ can be designed simultaneously with other subfilters by changing the design vector to be

$$\boldsymbol{g} = [\boldsymbol{g}_o^T \;\; \boldsymbol{g}_e^T \;\; \boldsymbol{g}_{eq}^T \;\; \boldsymbol{g}_{msk}^T]^T \tag{2.52}$$

where $\boldsymbol{g}_{msk}$ is a vector of $H_{msk}(z)$, similar to previously defined $\boldsymbol{g}_e$, $\boldsymbol{g}_o$ and $\boldsymbol{g}_{eq}$. Corresponding equations of the frequency response of the overall filter, the error function, the objective function, and the analytical derivative expressions need slight modifications. The modifications are easy, and not listed here. One design example of the filter structure in Figure 2.4(a) will be presented in next section.

2.4 Design Examples

In this section, three design examples are presented. The first one is a low pass baseband filter for the IS-95 CDMA system. The second one is a low pass filter with high attenuation, and the last design example is a sharp FIR filter with narrow passband.

In the CDMA systems, a baseband digital bandlimiting filter is required to achieve zero Inter-Symbol Interference (ISI) at the expense of as little excess bandwidth as possible. The IS-95 CDMA standard recommends a pair of 48 taps symmetric linear-phase FIR filter as the baseband filter. The passband ripple of the filter should be less than 1.5dB from DC to 590kHz and the stopband attenuation should be greater than 40dB from 740kHz onwards. The signal rate is

4.9152MHz (=4×1.2288MHz). The implementation of such a baseband filter in the transmitter has been reported in [49, 63, 71, 74, 76]. These implementations are very cost effective as they utilize simple combinational logic or table look-up techniques to replace the costly multiplier based on the fact that the word length of the input signal is 1 bit. But at the receiver side, the multiplier is unavoidable as the input signal samples consists of multiple bits. As a result, more power and large chip area are required.

Here, a basic parallel filter is utilized to implement the IS-95 low pass filter. If we use the iterative method in Section 2.2.2 to design the IS-95 filter, the lengths of $H_e(z)$, $H_o(z)$ and $H_{eq}(z)$ are 2, 5 and 23, respectively. By the methods in [24, 43, 82], $H_o(z)$ is quantized into 5 bits sum-of-power-of-two (SPOT) terms, and $H_{eq}(z)$ is of 6 bits SPOT terms. If the WLS method is utilized, the lengths of $H_e(z)$, $H_o(z)$ and $H_{eq}(z)$ are 4, 3 and 22, respectively. The quantization result is that the $H_e(z)$ and $H_o(z)$ are of 4 bits SPOT term, and $H_{eq}(z)$ are of 5 bits SPOT terms. If the IS-95 48 Tap filter is quantized in the same way, it is of 7 bits SPOT terms. Figure 2.5 shows the frequency responses of a RRS prefilter, and the parallel prefilters designed by the iterative and WLS methods. From Figure 2.5 we can see that the WLS design method results in the largest stopband attenuation. Figure 2.6 shows the frequency responses of the IS-95 48-tap filter, and two basic parallel filters designed by different methods. Figure 2.7 shows the frequency responses of each subfilter designed by the WLS design method.

In actual implementation, $H_{eq}(z)$ is cascaded with the parallel combination of

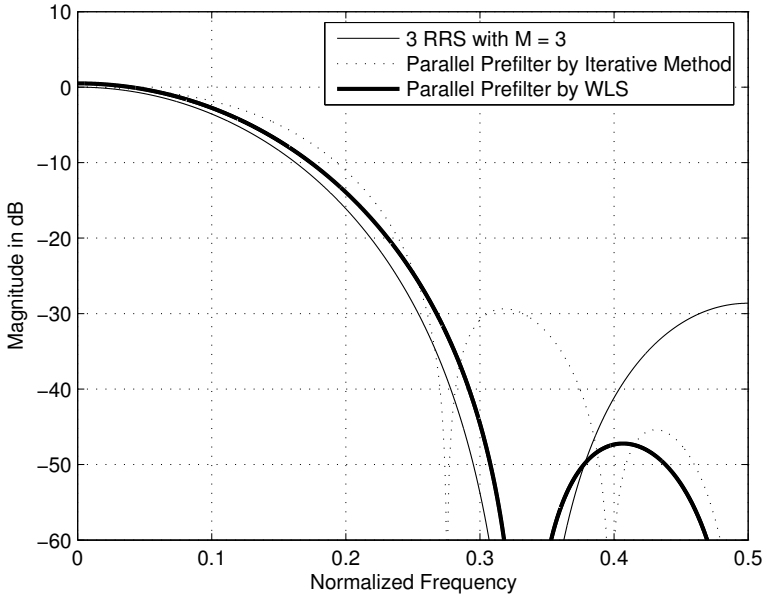

Figure 2.5: Frequency response of 3 prefilters

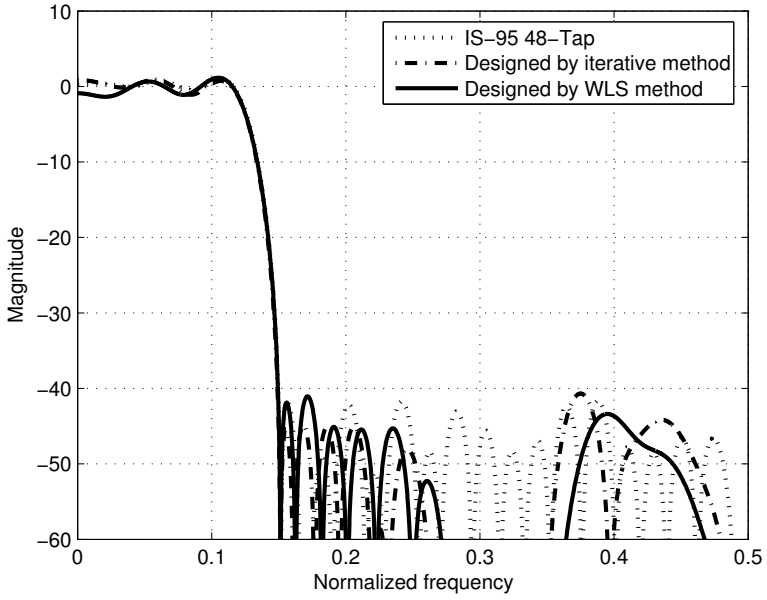

Figure 2.6: Frequency responses of overall filters

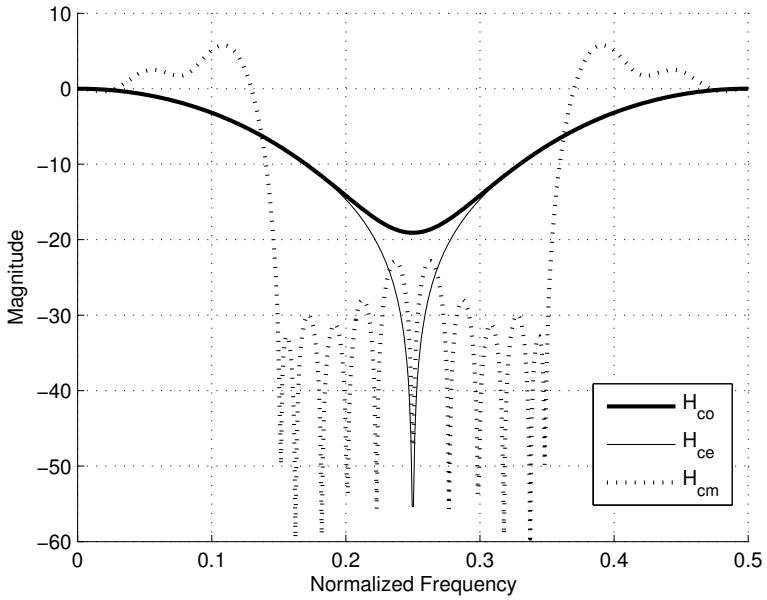

Figure 2.7: Frequency responses of each subfilter designed by WLS methods

$H_o(z)$ and $H_e(z)$ as shown in Figure 2.2. After comparing the various FIR implementation schemes, a direct form linear phase approach shown in Figure 2.8 is employed to implement the $H_{eq}(z)$, $H_o(z)$, and $H_e(z)$.

For a transmitter, it is noticed that the input to the IS-95 baseband filter is a 1 bit stream. The input goes into a delay chain and two delayed inputs are summed based on the coefficient symmetric property of a linear-phase FIR filter. This sum is then multiplied with the appropriate coefficient, which is shown as the dashed line box in Figure 2.8. Because the input to the filter is a 1 bit bipolar signal, the dashed box can be reduced to a decoding logic as shown in Table 2.1, assuming an input high represented by '1' and an input low by '-1'. Depending on the input signals of 'A' and 'B', the output of the dashed box takes values of 0, $-2h_k$ or $2h_k$. Table 2.1 suggests a simple and efficient way of implementing the coefficient

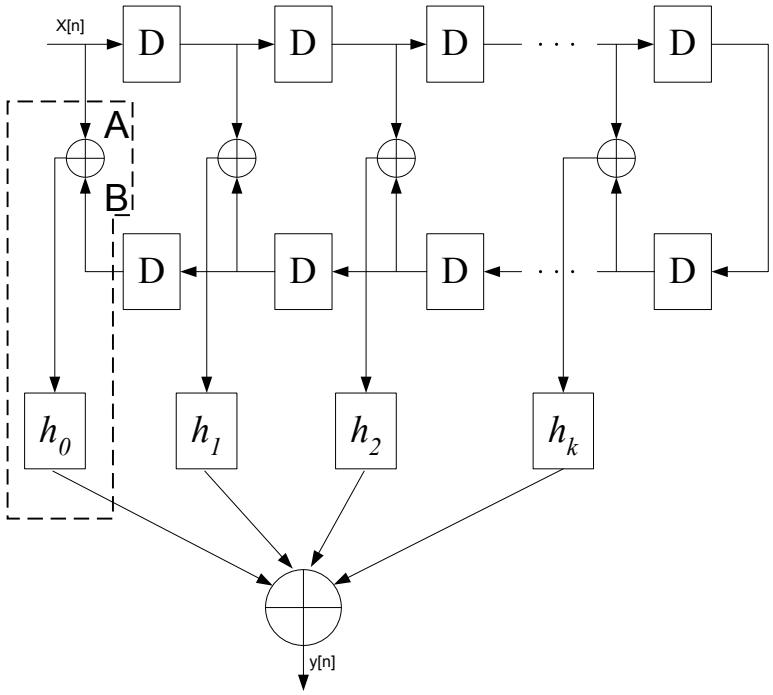

Figure 2.8: Direct form linear FIR structure for IS-95

multiplication. The fixed values , 0, $-2h_k$ and $2h_k$, can be implemented as the constant value in a look-up table approach, and applied to the inputs of a 3-1 multiplexer. The 'A' and 'B' can be used to create the selection signals for the multiplexer resulting in the output being the coefficient itself, the negated and zero. It is noted that 0, $2h_k$ and $-2h_k$ are all even integers, so the least significant bit of multiplexers' output will always be zero. As a result, the word-length of the accumulator can be reduced by 1 bit. The final output can be formed by padding a zero to the end of the accumulator's output.

For $H_o(z)$ and $H_e(z)$, and the IS-95 receiver, the input is no longer 1-bit stream. Therefore, the decoding logic in Table 2.1 can't be used. The multiplications are implemented in a shifter manner. Assume that the input of a receiver is

A	B	h_k
0	0	$-2h_k$
0	1	0
1	0	0
1	1	$2h_k$

Table 2.1: Decoding logic table used on the computation of sampled values

of 8 bits. $H_{eq}(z)$, $H_o(z)$ and $H_e(z)$ are implemented using Direct-form I. To compare the performance of the basic parallel filter designed by iterative method in Section 2.2.2, by WLS method in Section 2.3, and the original 48-tap IS-95 filter, ASIC designs are carried out. The above filters are coded in VHDL, and synthesized by Synopsys Design Compiler using 0.35 μm CMOS technology. Power consumption is analyzed by Synopsys Power Compiler. Tables 2.2 and 2.3 show the comparison between original IS-95 filter and the basic parallel filter used in both transmitter and receiver. It can be seen from Tables 2.2 and 2.3 that the basic parallel filter achieves considerable savings in terms of silicon area and power dissipation.

The second design example is a low pass filter. The normalized passband edge and stopband edge are 0.17 and 0.2, respectively. The passband ripple is 0.01, and the stopband ripple is 0.0001. The length of the minimax optimum design meeting this specification is 109. If the design method in Section 2.2.2 is utilized, the lengths of $H_e(z)$, $H_o(z)$, and $H_{eq}(z)$ are 2, 17 and 53, respectively. The total number of coefficients is 72, a 33% saving in the number of multipliers compared

with the minimax optimum design. If the WLS design method is utilized, the lengths of $H_e(z)$, $H_o(z)$ and $H_{eq}(z)$ are 6, 5 and 55, respectively. The total number of coefficients is 66, which saves 6 taps compared with the design by the iterative method. Meanwhile, the stopband attenuation of the filter designed by WLS method reaches -84.7dB, exceeding both the minimax optimum design and the result obtained by the iterative method in Section 2.2.2. Figure 2.9 shows the frequency response of the WLS designed overall filter.

The third design example is a low pass filter with narrow passband width, and sharp transition band. The passband and stopband edges are 0.05 and 0.06, respectively. The passband and stopband deviations are both 0.01. The length of minimax optimum design meeting this set of specifications is 200. If we use the parallel filter structure in Figure 2.4(a) with $M = 6$ and design the filter by the WLS method, the lengths of $H_e(z)$, $H_o(z)$, $H_{eq}(z)$ and $H_{msk}(z)$ are 2, 3, 35 and 6 respectively. The total number of taps is 46, which is only 23 per cent of that of the minimax optimum design. The group delay is 11.06% longer than the

	Number of SPOT Terms	Number of Adders (Receiver)	Number of Gates	
			Transmitter	Receiver
IS-95 48-Tap	44	66	1510	5569
Iterative design	27	38	1251	5087
Proposed WLS design	22	34	1103	4556

Table 2.2: Complexity comparison of IS-95 48-tap filter, iterative design, and proposed WLS design

	Power Consumption	
	Transmitter	Receiver
IS-95 48-tap	460.27μW	2.0mW
Iterative Design	360.95μW	1.5mW
Proposed WLS design	113.89μW	1.3mW

Table 2.3: Power consumption comparison of IS-95 48-tap filter, iterative design, and proposed WLS design

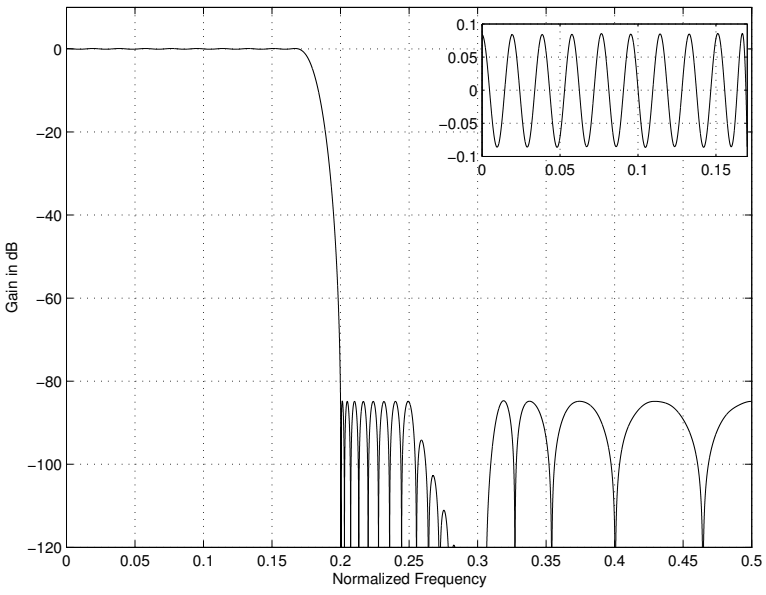

Figure 2.9: Frequency response of design example 2 by WLS method

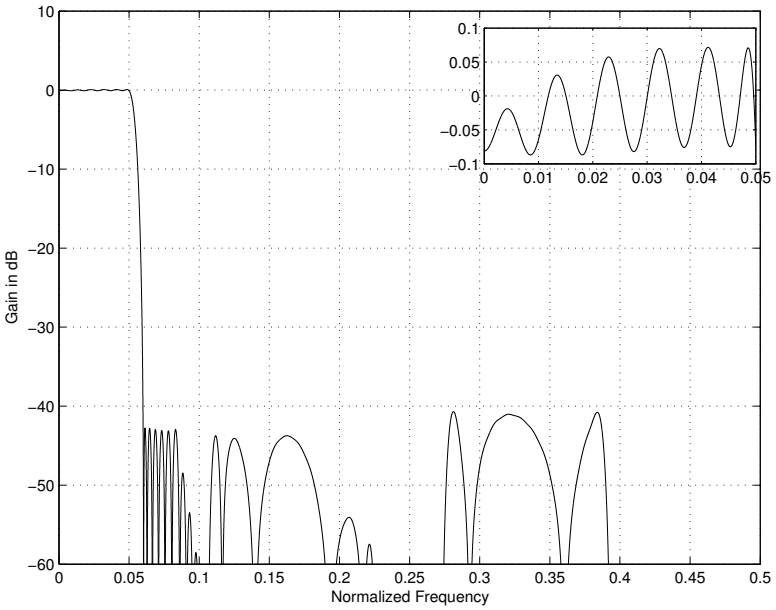

Figure 2.10: Frequency response of design example 3 by WLS method

minimax optimum design. The frequency response is shown in Figure 2.10.

2.5 Conclusion

In this chapter, the parallel prefilter, its iterative design method, and corresponding filter structures are first reviewed. A new WLS design method is proposed for the design of parallel prefilter and its equalizer. The success of the WLS method is due to the fact that it can optimize all the subfilters simultaneously. Design examples show that the WLS method improves the efficiency of a parallel prefilter for both moderately wide and narrow transition bandwidth cases.

Chapter 3

Length Estimation of Basic Parallel Filter

In last chapter, the parallel prefilter and its corresponding filter structures were reviewed. These filter structures can be used to design FIR filters with a narrow or moderate transition band. To further improve the efficiency of the parallel prefilter and its equalizer, a new WLS design method was proposed, which jointly optimizes all the subfilters.

Before an actual design begins, we need to estimate the lengths of the parallel prefilter and its equalizer. In this chapter, we will develop formulas and a table to estimate the lengths of a parallel prefilter and its equalizer.

3.1 Introduction

Filter length estimation plays an important role in the field of filter design. A filter design task can be regarded as finding a set of coefficients satisfying the given specifications. At the same time, the number of coefficients should be reduced as small as possible. Generally speaking, it is an iterative procedure to reach such a smallest number of coefficients. A good filter length estimation can effectively reduce the iteration times. Ichige *et al.* proposed a good length estimation equation in the design of a symmetric FIR filter [64, 67, 73]. Their equation is much more accurate than equations proposed by Herrmann *et al.* [9] and Kaiser [12].

However, the filter length estimation becomes more difficult when a filter consists of several subfilters. The ripple of one subfilter can be compensated by ripples of other subfilters. This compensation helps reduce the complexity of the overall filter. When nonlinear optimization techniques are introduced [75, 79, 91, 99] into filter designs, the estimated subfilter lengths according to [9, 12, 64, 67, 73] may be much larger than actual values. The reason behind this phenomena is because nonlinear optimization techniques can design all the subfilters simultaneously, and make the ripple compensation more effectively.

In this chapter, the length estimation of a low pass basic parallel filter will be exploited. A basic parallel filter refers to the parallel prefilter and its equalizer as shown in Figure 2.2 with $M = 2$, where M is the interpolation factor in a basic

parallel filter. A basic parallel filter is the base for the design of a multistage parallel filter [57].

The organization of this chapter is as follows. In Section 3.2, problems and corresponding solutions related to the length estimation of a basic parallel filter are presented. Section 3.3 analyzes the effects of different parameters to the filter length, and develops formulas and a table to estimate the length of each subfilter in a basic parallel filter. The accuracy of the proposed formulas and table are evaluated in Section 3.4. Section 3.5 makes some conclusions.

3.2 Problems and Solutions of Length Estimation of a Basic Parallel Filter

There exist two main problems related to developing length estimation formulas for a basic parallel filter: length combination and computing time. In this section, we present solutions for both of them. We also discuss the length relationship between the even-length and odd-length subfilters in a basic parallel filters.

3.2.1 Length Combination

Our target is to find formulas to estimate lengths of subfilters in a basic parallel filter. Estimating subfilter lengths of a basic parallel filter is more complex than

that of a single FIR filter. One problem is length combination. Due to the fact that a basic parallel filter consists of three subfilters, there may exist different length combinations satisfying the same set of filter specifications. That is to say two or more basic parallel filters have the same total number of coefficients, and they satisfy the same set of specifications. However, their subfilter lengths are different. For example, consider a low pass FIR filter with the following specifications: passband ripple $\delta_p = 0.01$, stopband ripple $\delta_s = 4.553 \times 10^{-4}$, normalized passband and stopband edges at $f_p = 0.221$ and $f_s = 0.244$, respectively. At least two set of filters satisfy the above specifications, i.e. lengths of $H_o(z)$, $H_e(z)$ and $H_{eq}(z)$ are 7, 8 and 65, or 9, 10 and 61. The total length for each set is the same.

According to our observations, shorter $H_o(z)$ and $H_e(z)$ result in smaller passband and stopband ripples. Therefore, the solution to the length combination problem is to find shorter $H_o(z)$ and $H_e(z)$ for a basic parallel filter, if there exists different length combinations. The reason is that the ripple of the overall parallel filter is mainly determined by the equalizer, and the longer equalizer will have smaller ripples [12].

3.2.2 Computing Time

To improve the efficiency of a basic parallel filter, a WLS design method has been proposed in Chapter 2. The WLS design method converts the design task to an unconstrained optimization problem. However, a converged solution of this

unconstrained optimization problem can not guarantee an equiripple solution, and the weighting function has to be updated several times to obtain an equiripple solution. Each time when the weighting function is updated, the optimization procedure has to be repeated to minimize the target function, until an equiripple solution is found.

The above procedure will be acceptable for the design of one basic parallel filter at a time. However, for length estimation of basic parallel filters, the WLS design method is too time-consuming to be acceptable. This is because thousands of basic parallel filters must be designed to collect enough data. The required computing time will be prohibitively long if the WLS method is utilized. At the same time, Remez [3, 7, 8] or linear programming [4, 25] can not be utilized, either. This is because Remez or linear programming can only design subfilters separately, which can't fully exploit the efficiency of a basic parallel filter. Therefore, a new design method must be found, which can optimize all the subfilters simultaneously with high convergence speed.

To reduce computing time, the design task is formulated as a goal attainment problem. A goal attainment problem can be solved very fast in the Matlab environment. Therefore, the difficulty of long computing time faced by the WLS design method can be overcome. Detailed formulation is as follows.

A low pass filter has four parameters: passband ripple δ_p, stopband ripple δ_s, passband edge f_p, and stopband edge f_s. No matter what specific structure the

actual filter utilizes, the frequency response of a symmetric FIR filter is fully determined by its coefficient vector $\boldsymbol{g}$ ($\boldsymbol{g}$ is used here to be consistent with the discussion in Chapter 2). As far as a basic parallel filter is concerned, it has three subfilters, namely an even-length filter $H_e(z)$, an odd-length filter $H_o(z)$, and an equalizer $H_{eq}(z)$. Their coefficient vectors $\boldsymbol{g}_e$, $\boldsymbol{g}_o$ and $\boldsymbol{g}_{eq}$ determine the frequency response of the overall basic parallel filter $H(z)$. A single vector $\boldsymbol{g}$ can be used to denote these three vectors. That is to say

$$\boldsymbol{g} = [\boldsymbol{g}_e^T \ \boldsymbol{g}_o^T \ \boldsymbol{g}_{eq}^T]^T \tag{3.1}$$

and $H(e^{j\omega})$ is a function of $\boldsymbol{g}$, or

$$H(e^{j\omega}) = F(\boldsymbol{g}). \tag{3.2}$$

Suppose $H_d(e^{j\omega})$ is the ideal frequency response of a filter. The task of designing a filter is to find a suitable $\boldsymbol{g}$ to satisfy

$$-\delta_p \leqslant H(e^{j\omega}) - H_d(e^{j\omega}) \leqslant \delta_p \tag{3.3}$$

in the passband, and to satisfy

$$-\delta_s \leqslant H(e^{j\omega}) - H_d(e^{j\omega}) \leqslant \delta_s \tag{3.4}$$

in the stopband.

Defining $w = \delta_s/\delta_p$, (3.4) is changed to be

$$-w\delta_p \leqslant H(e^{j\omega}) - H_d(e^{j\omega}) \leqslant w\delta_p. \tag{3.5}$$

For a given set of frequency grid points $\omega_i \in [\,0,\ 2\pi f_p\,]$ in the passband, where $i = 1, 2, \cdots, K$, (3.3) is equivalent to $2K$ conditions. The $(2i-1)^{th}$ and $(2i)^{th}$ conditions are

$$H(e^{j\omega_i}) - \delta_p \ \leqslant \ H_d(e^{j\omega_i}) \tag{3.6}$$

$$-H(e^{j\omega_i}) - \delta_p \ \leqslant \ -H_d(e^{j\omega_i}). \tag{3.7}$$

Similarly for $\omega_i \in [\,2\pi f_s,\ \pi\,]$ in the stopband, where $i = K+1, K+2, \cdots, N$, we can have

$$H(e^{j\omega_i}) - w\delta_p \ \leqslant \ H_d(e^{j\omega_i}) \tag{3.8}$$

$$-H(e^{j\omega_i}) - w\delta_p \ \leqslant \ -H_d(e^{j\omega_i}). \tag{3.9}$$

Therefore, for $\omega_i \in [\,0,\ 2\pi f_p\,] \cup [\,2\pi f_s,\ \pi\,]$, where $i = 1, 2, \cdots, N$, we have total $2N$ conditions. By setting

$$F_{2i-1}(\boldsymbol{g}) \ = \ H(e^{j\omega_i}) \tag{3.10}$$

$$F_{2i}(\boldsymbol{g}) \ = \ -H(e^{j\omega_i}) \tag{3.11}$$

$$F^*_{2i-1} \ = \ H_d(e^{j\omega}) \tag{3.12}$$

$$F^*_{2i} \ = \ -H_d(e^{j\omega}) \tag{3.13}$$

and $\boldsymbol{W} = [\underbrace{1, 1, \cdots, 1}_{2K}, \underbrace{w, w, \cdots, w}_{2(N-K)}]$, the filter design task can be formulated as

$$\min_{\boldsymbol{g}} \ \ \delta_p \tag{3.14}$$

$$\text{such that} \qquad F_i(\boldsymbol{g}) - \boldsymbol{W}_i\delta_p \leqslant F^*_i \quad i = 1, 2, \cdots, 2N$$

where $\boldsymbol{W}_i$=$\boldsymbol{W}(i)$.

Equation (3.14) is of the standard format of a goal attainment problem [13]. A Matlab function *fgoalattain* [68] is available for solving the goal attainment problem. In the actual implementation of function *fgoalattain*, sequential quadratic programming (SQP) [16, 17, 20, 21] is utilized. Due to the efficiency and high convergence speed of SQP, the function *fgoalattain* can return a solution for our design task very fast. Meanwhile, the weight vector $\boldsymbol{W}$ does not need updating during the design procedure, which makes the design a straightforward procedure.

Compared with the WLS design method in Section 2.3, the goal attainment is improves the design speed of one given design example. However, this improvement is at the expense of larger memory space. Because the goal attainment method is a constrained optimization method, more memory is needed to deal with those constraint conditions. For the same design task, the WLS method in Section 2.3 requires less memory because the WLS method is an unconstrained optimization method. As far as the design result is concerned, these two design method can return comparable design results. Design examples in Section 2.4 have been tried in both methods. These two methods require the same subfilter lengths, and result in almost the same passband and stopband ripples.

3.2.3 Length Relationship between $H_o(z)$ and $H_e(z)$

A basic parallel filter is suitable for the design of narrow and moderately wide low pass filters. Therefore, the passband width is selected in the range of $[0.02,\ 0.22]$. At the same time, the transition bandwidth is selected in the range of $[0.01,\ 0.1]$. Meanwhile, passband and stopband ripples can also affect the subfilter lengths in a basic parallel filter. To make sure our developed formula accurate for different passband and stopband ripples, we let passband and stopband ripples vary in the range of $[10^{-5},\ 0.316]$. The above range contains most designs for practical applications. The selected passband edges are evenly distributed in the range of $[0.02,\ 0.22]$. At the same time, selected transition bandwidth, passband and stopband ripples (dB) are also evenly distributed in the range mentioned above.

The filter length estimation of a basic parallel filter needs to determine three different subfilter lengths. If the exhaustive search is carried out for each design example, the computing time will be prohibitively long. Exhaustive search was carried out only for selected design examples in the range mentioned above. To save time, we must find a relationship between the lengths of these subfilters.

We tried about 400 design examples. According to these examples, it is found that the total coefficient number of a basic parallel filter is minimized when N_o, the length of odd-length filter $H_o(z)$, and N_e, the length of even-length filter

$H_e(z)$, have the following relationship:

$$N_o = \begin{cases} N_e - 1 & N_e > 2 \\ N_e + 1 & N_e = 2. \end{cases} \tag{3.15}$$

This conclusion is quite helpful to reduce computing time in searching the minimal total length of a basic parallel filter. This is because the above conclusion eliminates one searching variable, and make the problem simpler.

3.3 Length Estimation Formulas for Basic Parallel Filters

3.3.1 Equalizer Length Estimation

In the following subsections, we will analyze how different parameters affect the lengths of subfilters in a basic parallel filter. These parameters include passband ripple δ_p, stopband ripple δ_s, passband edge f_p, and stopband edge f_s. Generally speaking, N_{eq}, the length of the equalizer $H_{eq}(z)$, is much longer than N_e or N_o. Therefore, we will first develop a formula to estimate N_{eq}.

A basic parallel filter is suitable for the design of narrow and moderately wide low pass filters. To develop a filter length formula, δ_p, δ_s, f_p and f_s are varied as described in Section 3.2.3 to represent a range of filters. We first consider the relationship between N_{eq} and δ_s.

a) Relationship between N_{eq} and δ_s

In the case of a single filter, the filter length is proportional to the logarithms of passband and stopband ripples. For a jointly optimized basic parallel filter, it is necessary to verify whether N_{eq} is still proportional to the logarithm of stopband ripple δ_s. Therefore, we let the stopband ripple vary from 1×10^{-2} to 1×10^{-5},

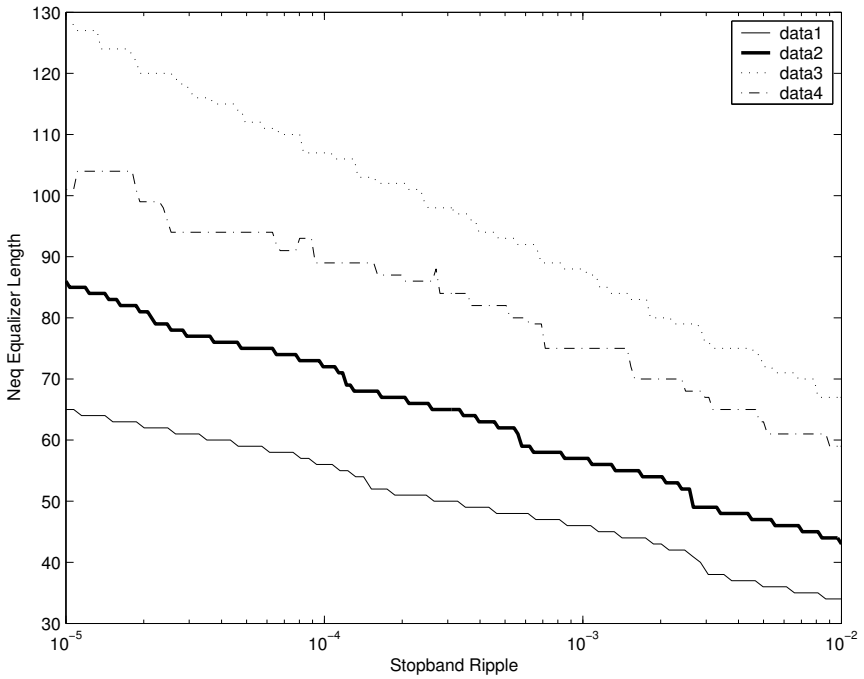

Figure 3.1: Relationship between N_{eq} and δ_s (logarithmic scale)

Legend	f_p	f_s	δ_p
Data1	0.05	0.08	0.01
Data2	0.068	0.091	0.01
Data3	0.124	0.139	0.01
Data4	0.216	0.234	0.01

Table 3.1: List of parameters used in Figure 3.1

and fix the other filter parameters which are listed in Table 3.1. The x-axis of Figure 3.1 is in logarithmic scale to observe the relationship between N_{eq} and $\log_{10}\delta_s$.

Observation 1: From Figure 3.1, it can seen that N_{eq} is approximately linear to $\log_{10}\delta_s$, for a given passband ripple, passband edge and stopband edge. Based on this observation, we can have the following equation,

$$N_{eq} = k_1 \log_{10}\delta_s + c_1 \tag{3.16}$$

where k_1 and c_1 are parameters to be determined.

b) Relationship between N_{eq} and δ_p

The same conclusion can be drawn for the relationship between N_{eq} and δ_p. Let us vary the passband ripple from 1×10^{-2} to 1×10^{-5} while keeping the other parameters fixed, and observe the variation of N_{eq}. The variation of N_{eq} is plotted in Figure 3.2, and the other filter parameters are listed in Table 3.2.

Observation 2: From Figure 3.2, it can be seen that N_{eq} is approximately proportional to $\log_{10}\delta_p$, for a given stopband ripple, passband edge and stopband edge. Therefore, we can have the following equation

$$N_{eq} = k_2 \log_{10}\delta_p + c_2 \tag{3.17}$$

where k_2 and c_2 are parameters to be determined.

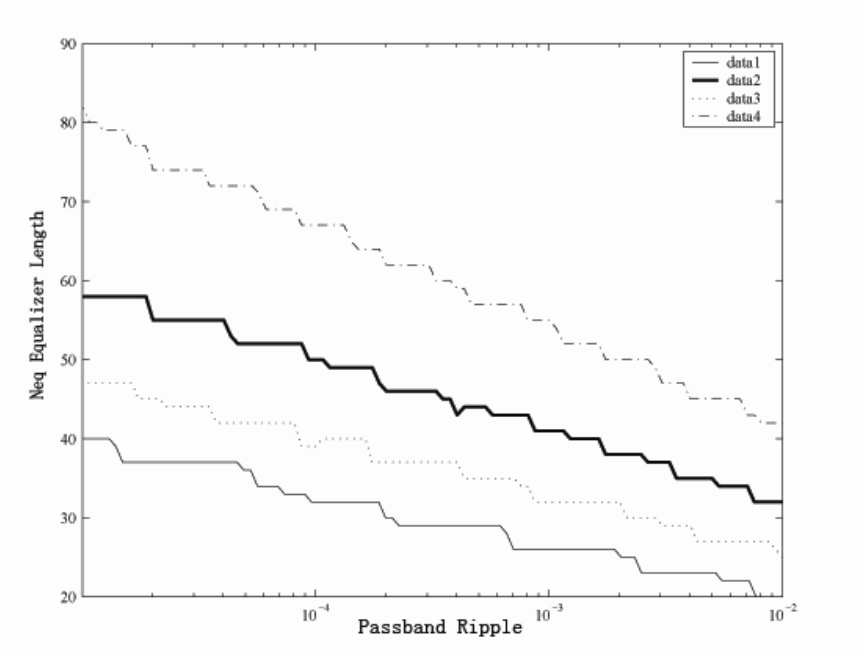

Figure 3.2: Relationship between N_{eq} and δ_p (logarithmic scale)

Legend	f_p	f_s	δ_s
Data1	0.086	0.1347	0.01
Data2	0.118	0.1506	0.01
Data3	0.135	0.1765	0.01
Data4	0.168	0.1928	0.01

Table 3.2: List of parameters used in Figure 3.2

c) Relationship between N_{eq} and ΔF

From now on, ΔF is used to denote the normalized transition bandwidth of the overall basic parallel filter. Here, the concept of central frequency is introduced. Central frequency of a low pass filter is defined as $f_c = (f_p + f_s)/2$. Then f_p and

f_s can be calculated using f_c and ΔF

$$f_p = f_c - \Delta F/2 \tag{3.18}$$

$$f_s = f_c + \Delta F/2. \tag{3.19}$$

When f_c and ΔF are determined, corresponding passband edge f_p and stopband edge f_s are determined as well. Therefore, N_{eq}'s relationships with f_c and ΔF are equivalent to its relationships with passband and stopband edges. The main reason for introducing the concept of central frequency f_c is because it is more convenient to study N_{eq}'s relationships with f_c and ΔF. Therefore, we try to find the relationship between N_{eq} and ΔF and f_c.

In the case of a single FIR filter, the filter length is inversely proportional to the transition bandwidth. For a jointly optimized basic parallel filter, it is possible that N_{eq} is proportional to the inverse of the transition bandwidth. Therefore, we fix the other filter parameters which are listed in Table 3.3, and observe the variation of N_{eq} when the transition bandwidth ΔF varies in the range of [0.01, 0.1]. Figure 3.3 shows the relationship between N_{eq} and ΔF. The x-axis of Figure 3.3 is the inverse of ΔF.

Observation 3: According to Figure 3.3, N_{eq} is proportional to the inverse of the transition bandwidth ΔF. we have

$$N_{eq} = \frac{k_3}{\Delta F} + c_3 \tag{3.20}$$

where k_3 and c_3 are to be determined.

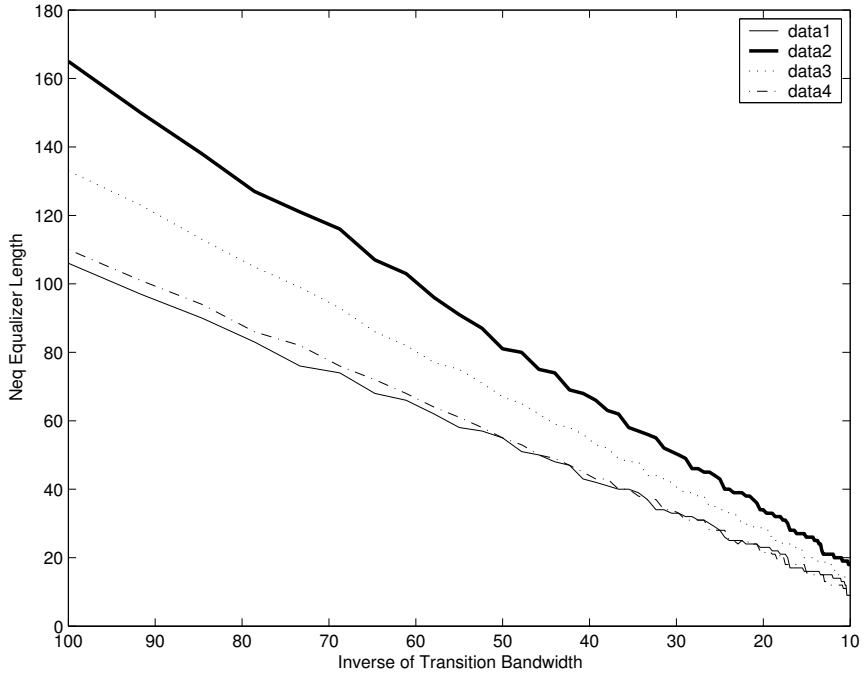

Figure 3.3: Relationship between N_{eq} and inverse of transition bandwidth

Legend	f_c	δ_p	δ_s
Data1	0.06	0.02	3.16×10^{-3}
Data2	0.09	0.02	3.16×10^{-5}
Data3	0.11	0.02	3.16×10^{-4}
Data4	0.17	0.05	5.62×10^{-4}

Table 3.3: List of parameters used in Figure 3.3

d) Relationship between N_{eq} and f_c

In the case of a single FIR digital filter, Ichige *et al.* [64, 67, 73] pointed out that f_c could affect the length of the FIR filter. To make sure our developed formulas are accurate, we need to observe the relationship between N_{eq} and f_c. The other parameters are fixed, and the variation of N_{eq} is observed when f_c

changes. Figure 3.4 shows the variation of N_{eq} with the change of f_c, while the other parameters are listed in Table 3.4.

Observation 4: From Figure 3.4, N_{eq} is larger when f_c is near 0, or 0.25.

Observation 5: N_{eq} is almost constant when f_c is in the interval [0.08 0.18].

According to Observations 4, a quadratic curve is used to fit the curves in Figure 3.4 while taking the effect of δ_s into account, i.e.

$$N_{eq} = k_4 \log_{10}(\delta_s) f_c^2 + k_5 \log_{10}(\delta_s) f_c^2 + c_4 \tag{3.21}$$

where k_4 and c_4 are to be determined.

e) Formula for N_{eq} Estimation

Combine (3.16) to (3.21), the following formula is heuristicly established to estimate N_{eq},

$$\begin{aligned} N_{eq} &= \left(\frac{A \log_{10} \delta_p + B \log_{10} \delta_s + C}{\Delta F} + D \right) \\ &\quad \cdot (E \log_{10}(\delta_s) f_c^2 + F \log_{10}(\delta_s) f_c + G) \end{aligned} \tag{3.22}$$

Least square (LS) criteria is utilized to determine parameters in equation (3.22). Suppose $\hat{N_{eq}}$ is the actual length of the equalizer. The parameters in equation (3.22) are determined by

$$\min \sum |\hat{N_{eq}} - N_{eq}|^2 \tag{3.23}$$

Approximately three thousand low pass basic parallel filters were designed. The selection of filter specifications is the same as described in Section 3.2.3. Normalized central frequency and transition bandwidth are evenly distributed in the

selected range. Passband and stopband ripples are selected to be logarithmically distributed, which is equivalent to even distribution in dB. For example, in Figure 3.4 each line corresponds to one hundred design examples. The central frequency f_c of these design examples are linearly distributed in the range of $[0.02,\ 0.21]$.

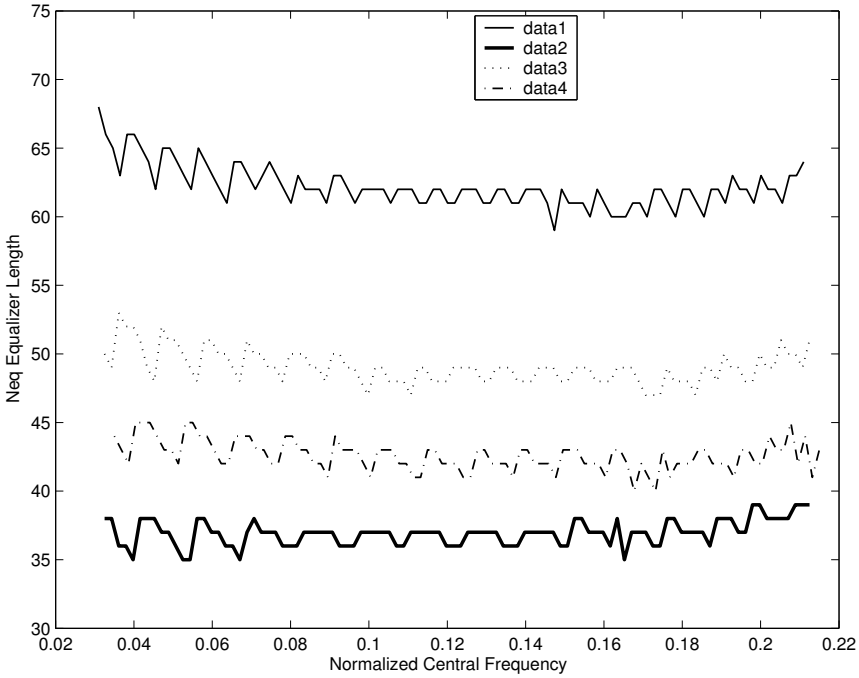

Figure 3.4: Relationship between N_{eq} and f_c

Legend	ΔF	δ_p	δ_s
Data1	0.022	0.02	3.16×10^{-4}
Data2	0.025	0.02	0.01
Data3	0.025	0.02	0.001
Data4	0.03	0.015	0.001

Table 3.4: List of parameters used in Figure 3.4

The parameters in Equation (3.22) is determined by using the actual lengths of equalizers $\hat{N_{eq}}$s of these design examples and the LS method of (3.23), and the following expression is obtained:

$$\begin{aligned} N_{eq} &= \left(\frac{-0.33\log_{10}\delta_p - 0.34\log_{10}\delta_s - 0.35}{\Delta F} + 0.7\right) \\ &\quad \cdot(-0.92\log_{10}(\delta_s)f_c^2 + 0.29\log_{10}(\delta_s)f_c + 1.04)\,. \end{aligned} \tag{3.24}$$

By trying different initial solutions, the global optimum solution is obtained.

According to Observation 5, N_{eq} can be regarded as having a constant value if $f_c \in [0.08, 0.18]$, i.e.

$$N_{eq} = c_5. \tag{3.25}$$

Therefore, (3.22) can be simplified to be

$$N_{eq} = \frac{A\log_{10}\delta_p + B\log_{10}\delta_s + C}{\Delta F} + D. \tag{3.26}$$

By least squares criterion, we obtain

$$N_{eq} = \frac{-0.3\log_{10}\delta_p - 0.33\log_{10}\delta_s - 0.26}{\Delta F} + 0.8. \tag{3.27}$$

3.3.2 Even and Odd-Length Filter Length Estimation

In this section, we do the filter length estimation for N_e. N_o can be determined by (3.15). The same procedure as that in the determination of N_{eq} is used.

a) Relationship between N_e and δ_s

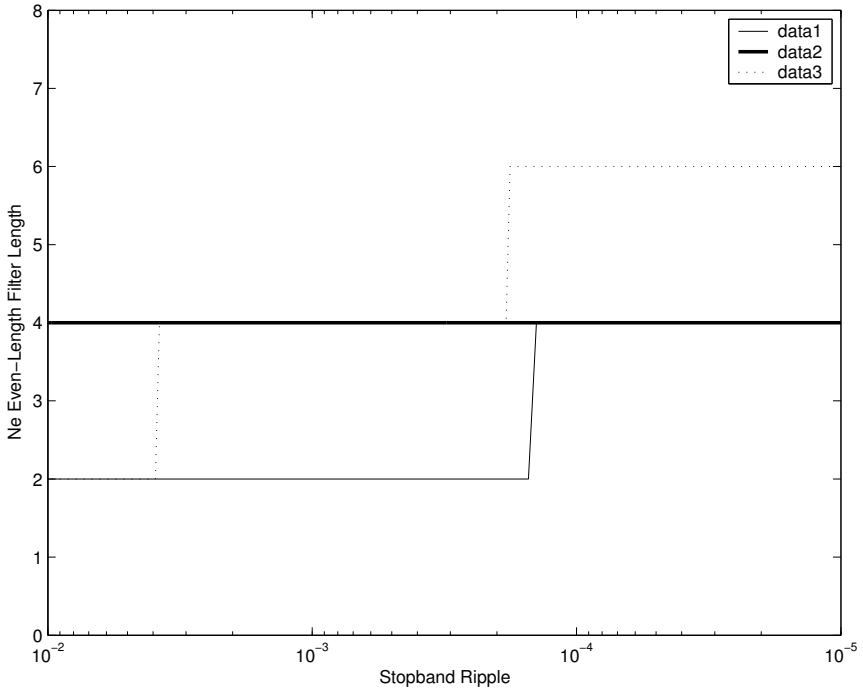

Figure 3.5: Relationship between N_e and δ_s

The relationship between N_e and δ_s is plotted in Figure 3.5 using the data listed in Table 3.1.

Observation 6: From Figure 3.5, it can be seen that N_e increases when stopband ripple decreases.

b) Relationship between N_e and δ_p

The relationship between N_e and δ_p is plotted in Figure 3.6 using the data in Table 3.2. The curves corresponding to data 1 and data 4 overlap with each other in Figure 3.6.

Observation 7: According to Figure 3.6, it can be seen that N_e increases when the passband ripple δ_p decreases.

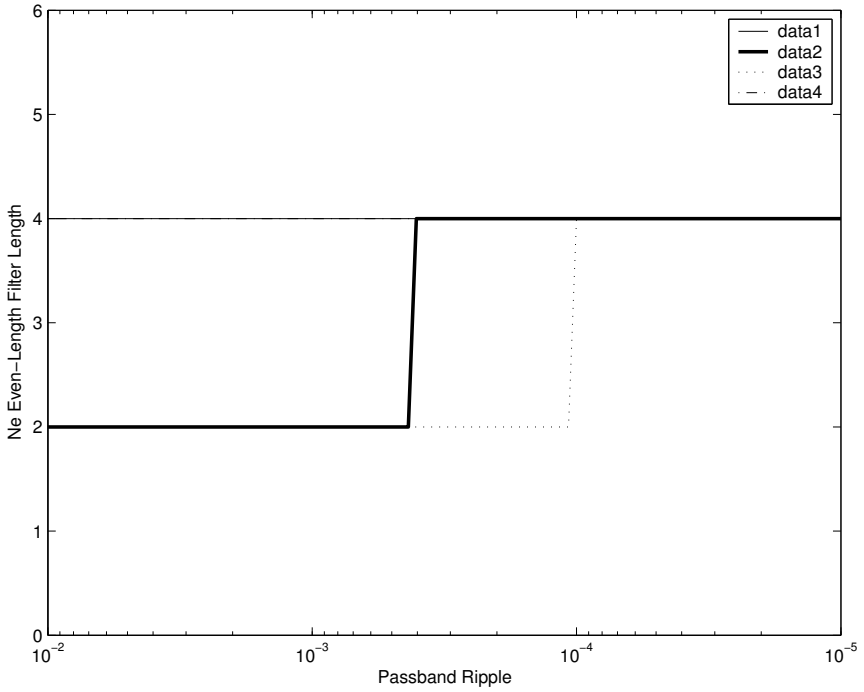

Figure 3.6: Relationship between N_e and δ_p

c) Relationships between N_e and Transition Bandwidth ΔF and Central Frequency f_c

To find the relationship between N_e and ΔF, the transition bandwidth of the overall filter, we use the data in Table 3.3 to obtain Figure 3.7. The curves corresponding to data 2 and data 3 overlap with each other in Figure 3.7.

Observation 8: N_e is independent of the transition bandwidth ΔF, according to Figure 3.7. We also note that N_e increases when the central frequency f_c increases, according to the listed f_c in Table 3.3.

d) N_e Estimation

According to collected data, N_e generally has a small value, in the range from

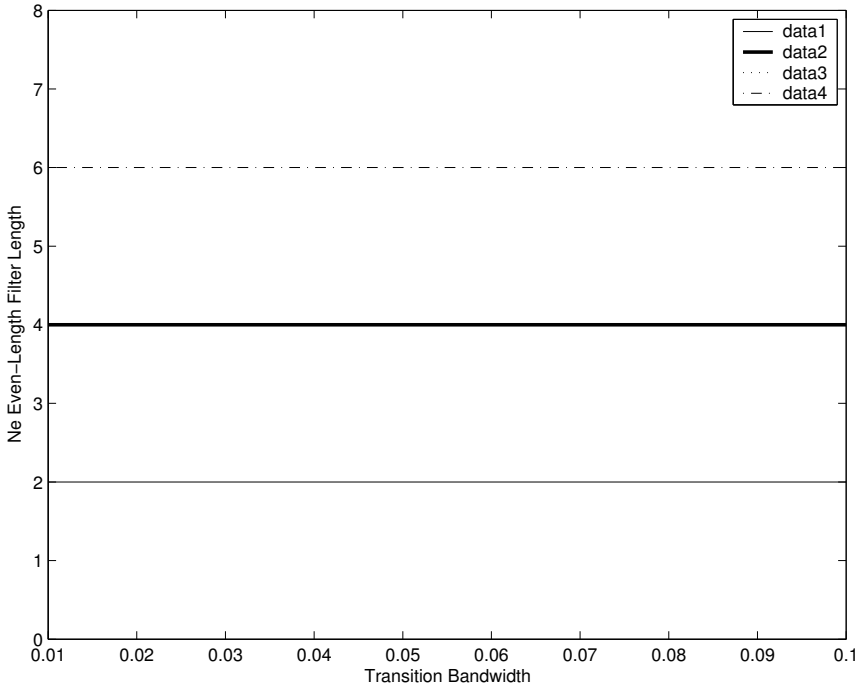

Figure 3.7: Relationship between N_e and transition bandwidth

2 to 14. The most suitable value of N_e, which will result in a simple basic parallel prefilter, is mainly determined by two parameters, central frequency f_c and stopband ripple δ_s. Although passband ripple can affect the value of N_e to some extent, the estimation of N_e is accurate enough if we ignore the effect from passband ripple δ_p. This will be verified in next section.

Because N_e is small, and its variation is much slower than N_{eq}, a look-up table strategy is more suitable for the estimation of N_e. According to the same set of design examples determining N_{eq} in Section 3.3.1, Table 3.5 lists possible estimation of N_e. If the passband ripple is quite small such as 0.001 or smaller than the stopband rippler, a larger value of N_e should be considered.

f_c	Stopband Attenuation (dB)												
	40	45	50	55	60	65	70	75	80	85	90	95	100
0.05	2	2	2	2	2	2	2	2	4	4	4	4	4
0.075	2	2	2	2	2	2	2	4	4	4	4	4	4
0.1	2	2	2	2	2	4	4	4	4	4	4	6	6
0.125	2	2	2	4	4	4	4	4	4	6	6	6	6
0.15	2	4	4	4	4	4	6	6	6	6	6	6	8
0.17	4	4	4	4	6	6	6	6	6	6	6	6	8
0.18	4	4	4	4	6	6	6	6	6	6	6	8	8
0.19	4	4	4	6	6	6	6	6	6	6	8	10	12
0.20	4	4	6	6	6	6	6	6	6	8	10	10	12
0.21	4	6	6	6	6	6	6	8	10	10	12	12	14
0.22	6	6	6	6	6	8	8	8	10	10	12	14	14

Table 3.5: N_e estimation table

3.4 Verification

To verify the accuracy of Equations (3.24) and (3.27), and Table 3.5, another approximately two thousand design examples are used. The selection methodology of specifications of new design examples is the same as what is described in Section 3.2.3. New design examples are selected to have different specifications from design examples in Section 3.3. The error distributions of Equations (3.24) and (3.27), and Table 3.5 are listed in Tables 3.6 to 3.8. These tables demonstrate that Equations (3.24) and (3.27) we developed , and Table 3.5 we summarized are accurate.

3.4.1 Accuracy Analysis of N_{eq} Estimation

When we analyze the accuracy of estimation equations and table, N_{eq} is used to denote the estimated value calculated, which is calculated according to the specifications of the design example and Equations (3.24) or (3.27). $\hat{N_{eq}}$ is the actual length of the corresponding equalizer. The error is defined as $|\hat{N_{eq}} - N_{eq}|$.

As shown in Table 3.6, the errors of 33.03% of all the design examples are 0, and 46.16% of all the design examples have an error of 1, if Equation (3.24) is used to estimate the length of the equalizer. As shown in Table 3.6, only 1.97% of all the design examples have an error greater than 3. Table 3.7 shows the corresponding error distribution of Equation (3.27). According to Tables 3.6 and 3.7, we can

see that Equations (3.24) and (3.27) can provide good estimation of the length of the equalizer. Tables 3.6 and 3.7 also show that Equation (3.24) gives a more accurate estimation of the length of the equalizer than Equation (3.27), which is expected.

Errors $\lvert\hat{N}_{eq} - N_{eq}\rvert$	Percentage of total design examples
$= 0$	33.03
$= 1$	46.16
$= 2$	13.68
$= 3$	5.16
>3	1.97

Table 3.6: N_{eq} error distribution for equation (3.24)

Errors $\lvert\hat{N}_{eq} - N_{eq}\rvert$	Percentage of total design examples
$= 0$	32.74
$= 1$	42.19
$= 2$	16.99
$= 3$	5.16
>3	2.92

Table 3.7: N_{eq} error distribution for equation (3.27)

3.4.2 Accuracy Analysis of N_e Estimation

Data collected in this section are also used to examine the accuracy of Table 3.5. When f_c or stopband attenuation falls between the values listed in Table 3.5, the smaller value is adopted for N_e. For example, if f_c is 0.1 and the stopband attenuation is 63 dB, N_e is estimated as 2. $\hat{N}_e$ is the actual length, and N_e is the estimated value. The resulting error distribution is shown in Table 3.8, similar to tables in Section 3.4.1.

From Table 3.8, it can been seen that Equation (3.15) and Table 3.5 can provide us satisfactory estimation of the length of the even-length subfilter.

Errors $\lvert\hat{N}_e - N_e\rvert$	Percentage of total design examples
$= 0$	69.05
$= 2$	30.95

Table 3.8: N_e error distribution

3.5 Conclusion

In this chapter, some problems facing the filter length estimation of a jointly optimized basic parallel filter are first pointed out. Solutions to these problems are presented. The effects of different parameters on the lengths of subfilters in a jointly optimized basic parallel filter are analyzed. Based on the collected data, two formulas are developed to estimate the length of equalizer $H_{eq}(z)$. A table is summarized to estimate the length of the even-length filter $H_e(z)$. The length

of the odd-length filter $H_o(z)$ is determined by an empirical relationship between N_e and N_o. Accuracy analysis shows that both the developed equations and the summarized table can provide satisfactory estimation of the subfilter lengths of a jointly optimized basic parallel filter. It should be pointed out that the equations and table are summarized according to some experiments, and not proved in mathematics. They are only a result of experience. At the same time, it is worth noting that these equations and tables are applicable to other optimization techniques. This is because the use of nonlinear optimization techniques is intended to produce a globally optimized design, e.g. a filter with minimum number of coefficients satisfying the given specifications. Hence, no matter what methods are used, the resulting FRM filters should approach a same destination, i.e. the final filter could be quite similar. Therefore, these equations and table can be used when other nonlinear optimization techniques are utilized.

Chapter 4

Design Equations for Jointly Optimized Frequency-Response Masking Filters

The introduction of nonlinear optimization techniques to the design of frequency-response masking (FRM) filters has changed the way how FRM filters are synthesized. It allows all subfilters in a FRM structure to be optimized jointly resulting in further savings in the number of arithmetic operations. Under the joint optimization, a new set of design equations is necessary not only for a more computationally efficient filter, but also for the simplification of design process and the reduction of design time. In this chapter, we present a set of design equations that helps to determine filter lengths and the optimum interpolation factor

in a FRM filter under joint optimization. It is shown, by means of examples, that the proposed design equations lead to a better estimation of the optimum interpolation factor and filter lengths for subfilters.

4.1 Introduction

Frequency-response masking (FRM) approach is one of the most computationally efficient techniques for the design of linear phase arbitrary bandwidth sharp FIR digital filters [32, 39, 48, 54–56, 59, 75, 77, 79–81, 83–102]. The structure of a basic FRM filter is shown in Figure 4.1, and Figure 4.2 shows the frequency responses of subfilters in a FRM structure. The creation of a FRM filter starts with a prototype filter $H_a(z)$ whose transition bandwidths are M times wider than that of the overall filter $H(z)$ as shown in Figure 4.2(a). A periodic narrow transition width filters $H_a(z^M)$ is created by interpolation, i.e. replacing each delay element in both $H_a(z)$ with M delay elements, as shown in Figure 4.2(b). The interpolation narrows the transition bandwidths of $H_a(z)$ by compressing their frequency responses, which produces extra passbands. The complementary part $H_c(z^M)$ is obtained by subtracting the output of $H_a(z^M)$ from a delayed version of the input signal, as shown in Figure 4.1. Parts of these extra passbands fall into the stopband region of the overall filter. Two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, as in Figures 4.2(c) and (e), are cascaded with $H_a(z^M)$ and $H_c(z^M)$, respectively, to remove the unwanted passbands. The outputs of $H_{Ma}(z)$ and

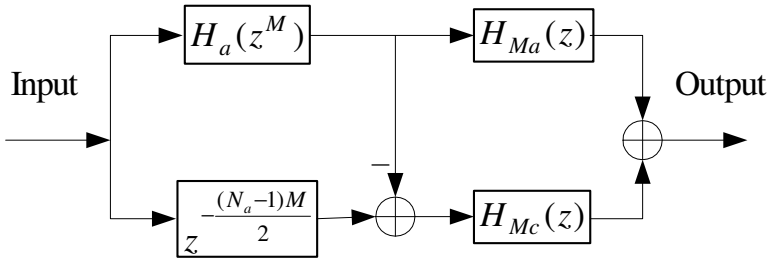

Figure 4.1: Basic FRM filter structure

$H_{Mc}(z)$ are combined to form the overall filter. Figures 4.2(c) and (d) show the situation of Case A, and Figures 4.2(e) and (f) show the situation of Case B. The design equations and the derivation of optimum interpolation factors for single and multiple stages FRM filters can be found in [32, 48].

In a FRM filter, the savings in terms of number of multipliers are closely related to the interpolation factor applied to the prototype filter. A larger interpolation factor reduces the complexity of prototype filter at the cost of longer masking filters. There exists an optimum interpolation factor M_{opt} for a given set of specifications that leads to a minimum total number of coefficients in a FRM filter. Such an optimum M_{opt} can be estimated using a formula given in [48]:

$$M_{opt} \approx \frac{1}{2\sqrt{\Delta F}} \tag{4.1}$$

where ΔF is the normalized transition bandwidth of a FRM filter. The derivation of (4.1) is based on an assumption that the complexity of a FRM filter is minimized when the transition bandwidths of the two masking filters are the same. Such an assumption may not hold for the FRM filters designed using an iterative design procedure [48]. Nevertheless, (4.1) gives a good estimation of the

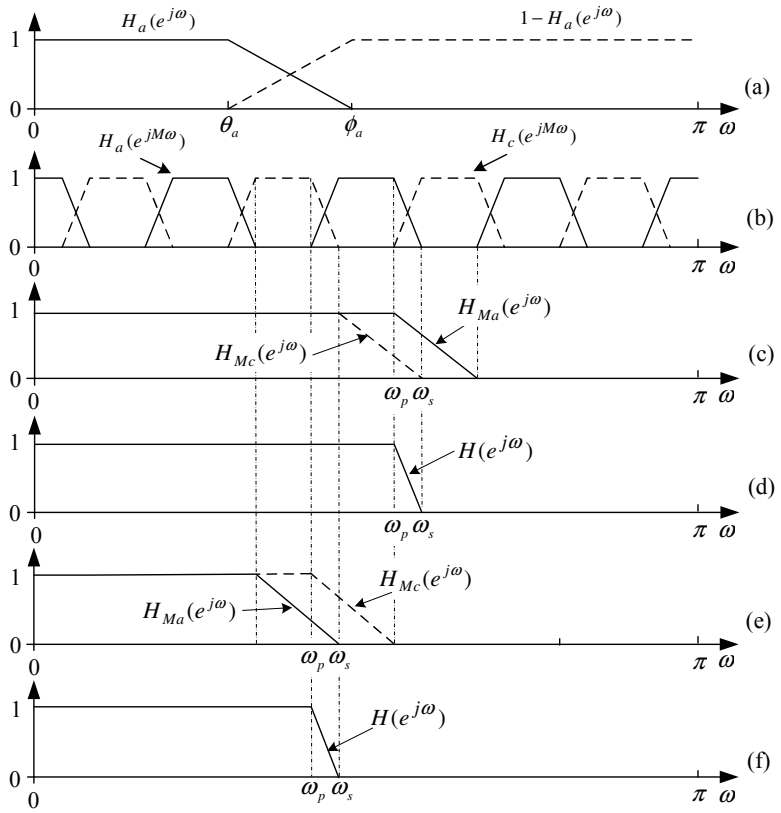

Figure 4.2: Frequency responses of subfilters in a FRM structure (a) Prototype filter and its complementary part (b) Interpolated prototype filter and its complementary part (c) Two masking filters for Case A (d) Overall FRM of Case A (e) Two masking filters for Case B (f) Overall FRM of Case B

optimum M and narrows the search region for the optimum M to a few integers.

Much research effort has been made to improve the computational efficiency of FRM filters in recent years. A notable effort is to employ nonlinear optimization techniques to jointly optimize the three subfilters in a single stage FRM structure [75, 79, 80, 91–93, 99]. Such an approach leads to about 20% additional savings in terms of number of multipliers. However, it was reported in [75] that (4.1) is no longer accurate for an optimum M under the joint optimization. A new formula was given in [75]:

$$M_{opt} = \frac{1}{\sqrt{3.2\Delta F}}. \tag{4.2}$$

However, the authors of [75] did not show how accurate (4.2) is. In this chapter we present a new set of formulas for the estimation of filter lengths and the optimum M in a jointly optimized FRM filter. The new design equations are derived from the observations made based on a few hundreds of FRM filters jointly optimized by the sequential quadratic programming (SQP) algorithm [17]. The proposed formulas produce much more accurate results than those reported in [32, 48, 75].

The rest of chapter is organized as follows. In Section 4.2, we highlight the differences between the original and jointly optimized FRM filters, and discuss the impacts of joint optimization on the interpolation factor and subfilter lengths. In Section 4.3, we introduce a new filter length estimation equation for the prototype filter. In Section 4.4, further observations are made to masking filters and an equation is derived to estimate the total length of the two masking filters. A new expression for the estimation of the optimum M is presented in Section 4.5

together with design examples which verify the accuracy of proposed formulas. Conclusion remarks are drawn in Section 4.6.

4.2 Impacts of Joint Optimization on FRM Filters

A typical jointly optimized FRM filter employs three subfilters which have very different amplitude responses compared to those in an original FRM structure. This is because of the involvement of nonlinear optimization techniques that help to reshape all the subfilters in a FRM structure. As a result, the filter lengths of subfilters are reduced. The most obvious change in the amplitude responses is the prototype filter as shown in Figures 4.3(a) and (b). Its normalized DC gain is no longer 1 and the DC level is shifted. The ripple in the passband is considerably increased, especially near the edges of the transition bands. Such changes have little impact on the filter length of the prototype filter according to design examples [75, 79, 80, 91–93, 99]. But they affect the two masking filters greatly. The transition bands of the masking filters become very close to each other and are widened. The widening of transition bandwidths in both masking filters yields additional savings in the number of multipliers and allows a larger interpolation factor compared to the one used in a FRM filter designed by an iterative method. The authors of [75] first noticed the increase in the interpolation factor and proposed an estimation formula (4.2). Design examples show that (4.2)

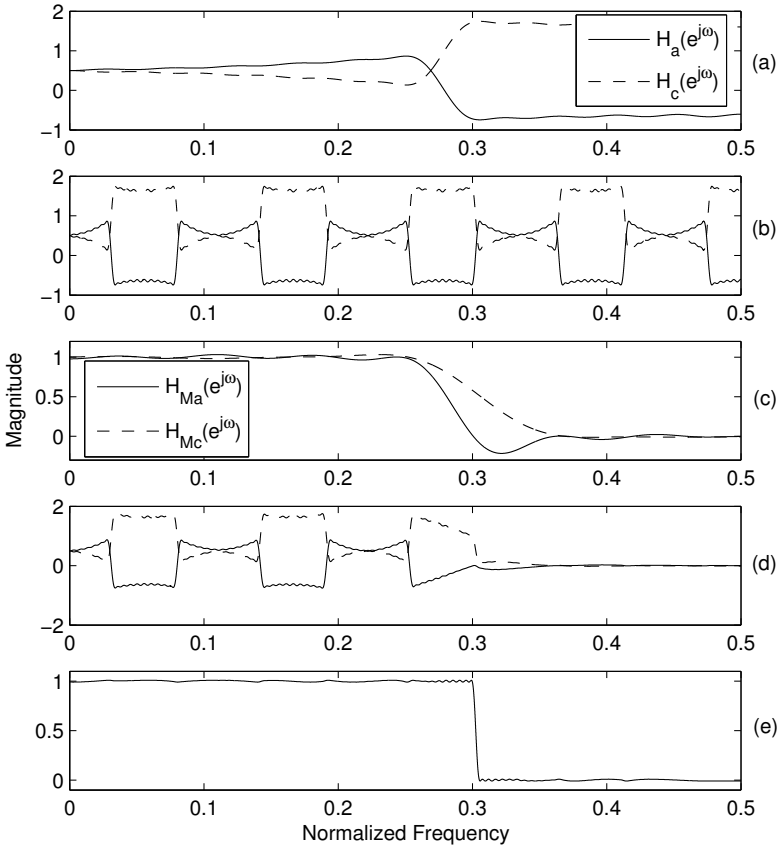

Figure 4.3: The frequency responses of various filters in jointly optimized FRM approach

gives a better estimation than (4.1). But the estimated value is still slightly away from the optimum M and requires a few iterations to reach an optimum design.

The question is whether there is a need to develop a more accurate formula for the interpolation factor. To answer this question, let us examine how a FRM filter is designed under the joint optimization methods. It starts with an initial solution that derives from one of the iterative design processes in [32, 48, 83], i.e. the interpolation factor is determined by a search method introduced in [48].

The initial filter lengths of prototype filter N_a , and the two masking filters N_{Ma} and N_{Mc}, can be determined by assuming that the ripple magnitudes are 85% of the allowed ripple magnitudes of the overall FRM filter. Nonlinear optimization techniques are employed to jointly optimize all subfilters leading to further reduction in the filter lengths. As the nonlinear optimization procedure does not include the interpolation factor M and filter lengths in the objective function, the resulting FRM filter is an improved version of the initial design. It is likely to find a better solution if a different M is used. In order to find the best design, an exhaustive search is necessory which involves an iterative procedure, i.e. vary the M, calculate the filter lengths, jointly optimize all subfilters, and gradually reduce the length of each subfilter to reach a minimum complexity for a given M. This is a very time consuming procedure. Thus an accurate formula for the optimum interpolation factor is necessary to simplify the design task.

To derive an interpolation factor estimation formula, we notice that the FRM filters designed with joint optimization methods coincide the assumption made in [48] about the masking filters, i.e. the transition bandwidths of the two masking filters are the same for a minimum complexity FRM filter. It is clear from examples in [75, 79, 80, 91–93] that the transition bandwidths are almost the same for the two masking filters. However, their lengths are not the same. This implies that the standard filter length estimation formula is no longer valid for subfilters in a FRM structure because of the shapes of all subfilters differ from the standard low pass filters. Thus new filter length estimation formulas are necessary

in the derivation of an optimum interpolation factor. Note from Figure 4.3 that the shape of the prototype filter differs from that of the two masking filters. We should develop the length estimations for the prototype filter and masking filters separately.

4.3 Filter Length Estimation for Prototype Filter

The filter length of a low pass filter is normally determined by a set of filter specifications, i.e. the normalized passband and stopband edges f_p and f_s, and the passband and stopband ripples δ_p and δ_s. The length estimation should be able to give a reasonable good match between the estimated length and actual filter length for a given set of specifications, i.e. varying the bandedges or ripples should not affect the accuracy of the formula. For the case of FRM filters, it is not necessary to develop the formula to cover all the normalized frequencies from 0 to 0.5. This is because the FRM approach is suitable for the synthesis of arbitrary bandwidth sharp transition bandwidth filters, mainly from moderate to wide bandwidth. For the cases of narrowband filters, the FRM approach simplifies to an IFIR structure [28, 48]. Therefore, we set the following restrictions for FRM filters designed in the rest of this chapter. The normalized passband edges are limited to in the region of $[0.1,\ 0.4]$. At the same time, the transition bandwidth of a FRM filter is limited to a range of $[0.002,\ 0.012]$. This region contains

most designs for a single stage FRM approach. To make sure that the developed formulas are accurate for different passband and stopband ripples, we vary the passband and stopband ripples from 10^{-1} to 10^{-5}. The interpolation factor M varies between 4 to 20 based on the given transition bandwidth of [0.002, 0.012].

Based on our experience, the length N_a of a jointly optimized prototype filter $H_a(z)$ is very close to the one designed by traditional iterative methods. The main reason behind this phenomenon could be that the transition bandwidth of the prototype filter remains the same for the traditional and jointly optimized designs. In [32], N_a is estimated by assuming that the ripples of $H_a(z)$ are 85% of the allowed, and by using Kaiser's equation [12]. Therefore,

$$N_a(\Delta F, \delta_p, \delta_s) = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 11.59}{14.6M\Delta F} + 1. \tag{4.3}$$

The above equation is not very accurate if the length of the prototype filter is long. To improve its accuracy, we modify the coefficients in Kaiser's equation, resulting in

$$N_a(\Delta F, \delta_p, \delta_s) = \frac{k_1\log_{10}\sqrt{\delta_p\delta_s} - k_2}{k_3M\Delta F} + k_4 \tag{4.4}$$

where k_1, k_2, k_3 and k_4 are parameters to be determined. A few hundreds of data points are collected from the FRM filters optimized with SQP technique. The bandedges, transition bandwidths, ripples, and interpolation factors are limited to ranges as described in the beginning of this section. The values of k_1, k_2, k_3 and k_4 are obtained from collected data points using the method of least squares

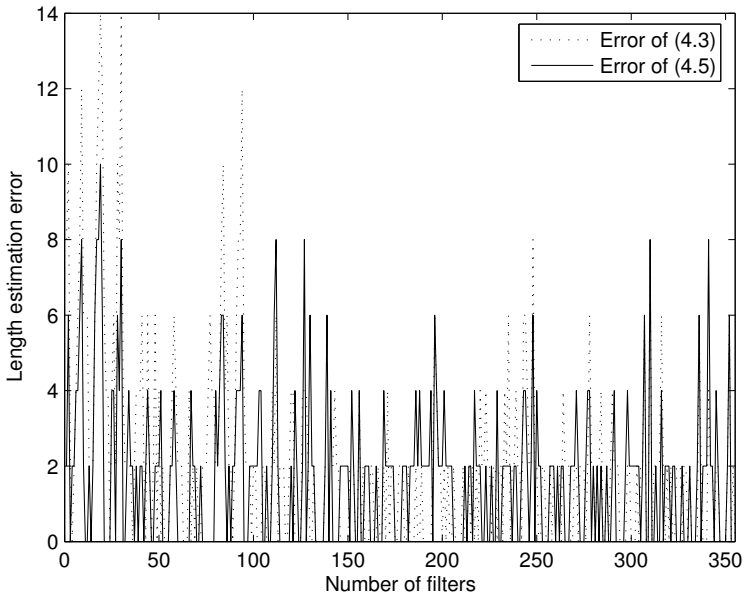

Figure 4.4: The absolute values of estimation errors of N_a

fitting. The modified formula is given by

$$N_a(\Delta F, \delta_p, \delta_s) = \frac{-9.9\log_{10}(\delta_p\delta_s) - 8.6}{16.3M\Delta F} + 3.33. \tag{4.5}$$

The absolute errors, i.e. the absolute values of the difference between actual and estimated lengths based on (4.3) and (4.5), are calculated and plotted in Figure 4.4. From the figure, it is clear that Equation (4.5) provides a better estimation for the length of prototype filters than Equation (4.3).

4.4 Masking Filter Length Estimation

The estimation of length for each masking filter is a very difficult task based on our experience. We notice that the lengths of two masking filters are exchangeable

for many designs while the resultant overall filters satisfy the same specifications. To show this, let us consider the design of a low pass filter with normalized passband and stopband edges at 0.191 and 0.197, respectively. The passband and stopband ripples are both 0.01. The optimum interpolation factor is 9 after exhaustive search. The lengths of 3 subfilters are $N_a = 39$, $N_{Ma} = 27$ and $N_{Mc} = 19$, where N_a is the length of the prototype filter, and N_{Ma} and N_{Mc} are the lengths of two masking filters. The frequency responses of subfilters and the overall filter are shown in Figure 4.5. For the same M value, another set of subfilters can be found which satisfies the given specifications and uses the same number of coefficients as in the first design. The filter lengths are $N_a = 39$, $N_{Ma} = 19$ and $N_{Mc} = 27$. The lengths of masking filters are swapped compared to the first design. The frequency responses of subfilters and the overall filter are shown in Figure 4.6. Many other design examples we tried show the same property.

This phenomena is very interesting and prompts us to find the sum of the lengths of masking filters instead of individual lengths. Recall from [32] that the sum of transition bandwidths of masking filters is related to M, i.e.

$$\Delta F_M = \Delta f_{Ma} + \Delta f_{Mc} = \frac{1}{M} \tag{4.6}$$

where Δf_{Ma} and Δf_{Mc} are the transition bandwidths of two masking filters. We are interested to know whether it is possible to use a modified Kaiser's formula for the estimation of the sum of the lengths of masking filters. Substituting (4.6)

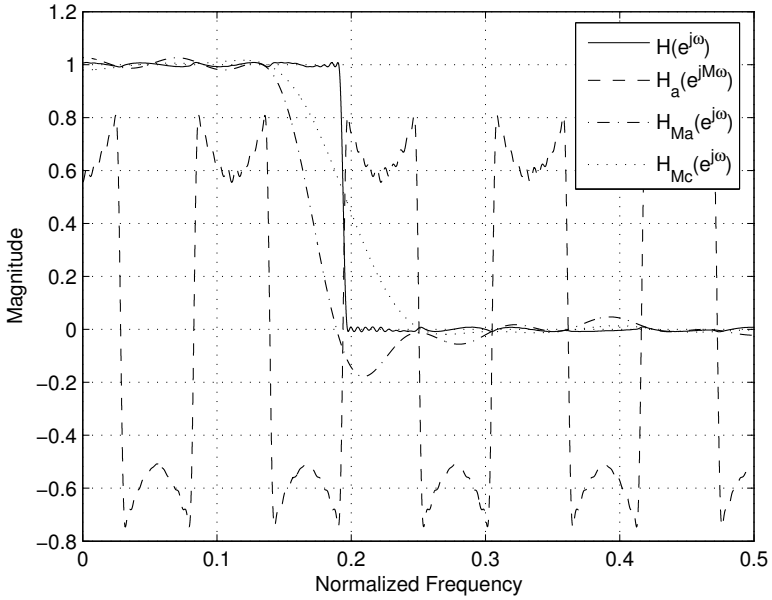

Figure 4.5: The frequency responses of subfilters and overall filter with $N_a = 39$, $N_{Ma} = 27$ and $N_{Mc} = 19$

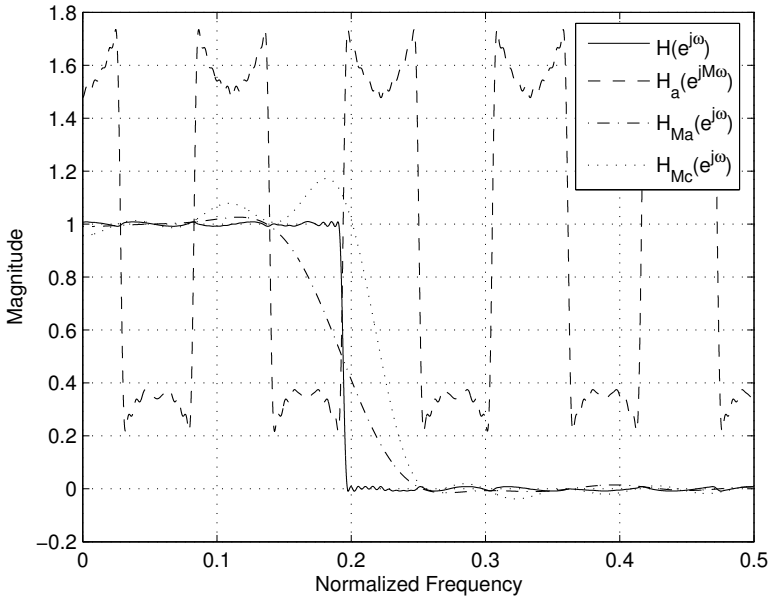

Figure 4.6: The frequency responses of subfilters and overall filter with $N_a = 39$, $N_{Ma} = 19$, and $N_{Mc} = 27$

into Kaiser's formula, we have

$$N_{Msum} = N_{Ma} + N_{Mc} = \frac{k_1 \log_{10} \sqrt{\delta_p \delta_s} - k_2}{k_3/M} + k_4. \tag{4.7}$$

After simplification, we have

$$N_{Msum} = N_{Ma} + N_{Mc} = kM \log_{10}(\delta_p \delta_s) - qM + c \tag{4.8}$$

where N_{Ma} and N_{Mc} are lengths of masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, respectively, δ_p and δ_s are the passband and stopband ripples, respectively, of the masking filters, and k, q, and c are variables yet to be determined. To verify the validity of (4.8), let us carry out a few experiments. In each experiment, the FRM filters are optimized by SQP. One of the parameters among bandedges, interpolation factor, and ripples is varied from one design to another, while the rest of the parameters are kept unchanged. The relationship between N_{Msum} and the selected parameter is analyzed.

It is interesting to note that (4.8) is not a function of transition bandwidth. To gauge the impact of transition bandwidth on N_{Msum}, let us fix the passband edge, ripples, and M as shown in Table 4.1 and vary the stopband edge. Figure 4.7 shows the relationship between N_{Msum} and the normalized transition bandwidth of FRM filters. It is clear that N_{Msum} remains a constant if the change in transition bandwidth is limited to a small range. For a large range from 0.002 to 0.012, N_{Msum} varies very little for each data set. Thus, the implication from (4.8) that N_{Msum} is independent of the transition bandwidth has been verified by these experiments.

To verify the relationship between N_{Msum} and δ_s, we fix f_p, f_s, δ_p and M as shown in Table 4.2. The δ_s is varied from 10^{-1} to 10^{-5}. The relationship between N_{Msum} and δ_s is plotted in Figure 4.8. It is obvious from Figure 4.8 that N_{Msum} changes almost linearly with the logarithm of stopband ripple δ_s or

$$N_{Msum} \approx k_1 \log_{10} \delta_s + c_1. \tag{4.9}$$

The same conclusion can be drawn for the passband ripple if we use the specifications listed in Table 4.3, i.e. varying δ_p from 10^{-1} to 10^{-5} while keeping the rest unchanged. Figure 4.9 confirms an approximate linear relationship between N_{Msum} and the logarithm of passband ripple δ_p, i.e.

$$N_{Msum} \approx k_2 \log_{10} \delta_p + c_2. \tag{4.10}$$

In (4.9) and (4.10) k_1, k_2, c_1, and c_2 are constants yet to be decided.

To show the relationship between N_{Msum} and M, we choose a set of low pass filters with different bandedges and passband and stopband ripples as shown in Table 4.4. For each set of specifications, we design the FRM filters using SQP by varying M from 4 to 20. The sums of masking filter lengths are calculated and shown in Figure 4.10. It is not difficult to conclude from Figure 4.10 that the sum of the lengths of the masking filters can be approximated by a linear relationship with M, i.e.

$$N_{Msum} \approx k_3 M + c_3. \tag{4.11}$$

where k_3 and c_3 are constants yet to be found.

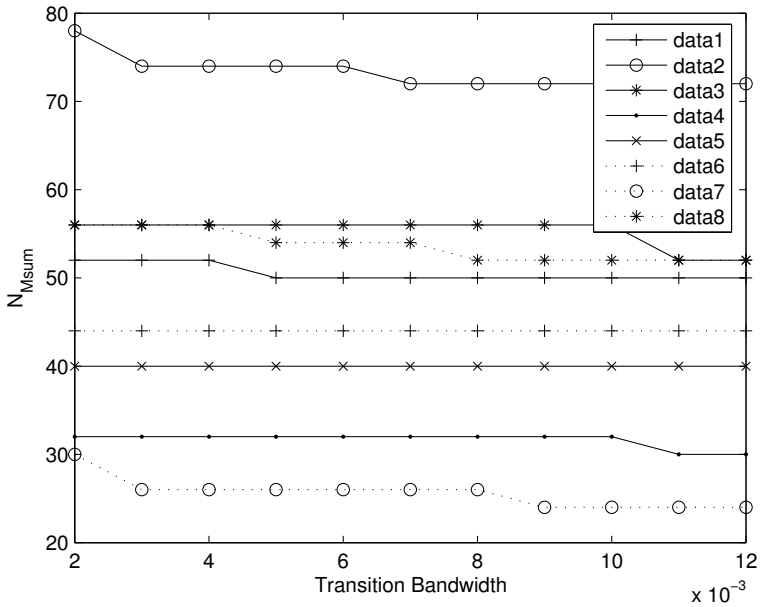

Figure 4.7: Relationship between N_{Msum} and transition bandwidth of the overall FRM filter

Legend	f_p	δ_p	δ_s	M
data1	0.11	0.05	0.005	11
data2	0.14	0.01	0.001	12
data3	0.16	0.03	0.001	10
data4	0.21	0.01	0.01	6
data5	0.24	0.08	0.005	9
data6	0.28	0.001	0.05	8
data7	0.33	0.005	0.03	5
data8	0.37	0.08	0.01	14

Table 4.1: Filter specifications used in Figure 4.7

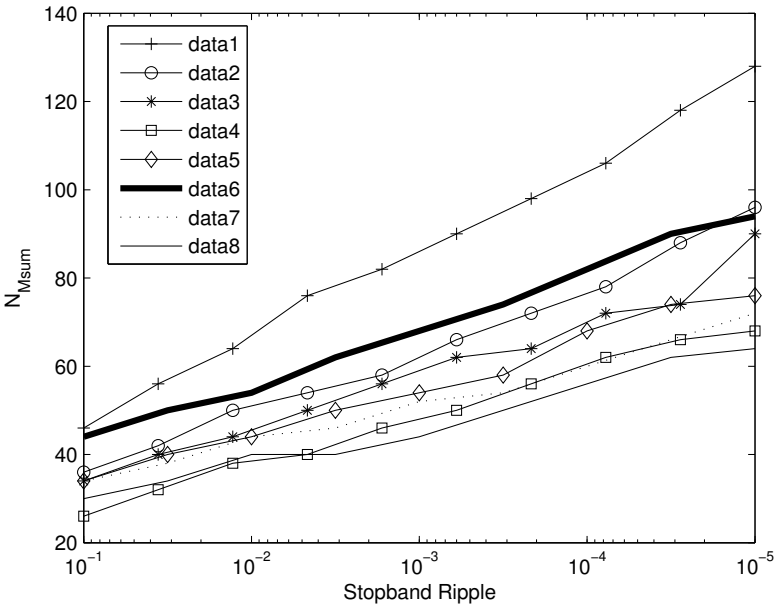

Figure 4.8: Relationship between N_{Msum} and stopband ripple

Legend	f_p	f_s	δ_p	M
data1	0.1	0.102	0.1	17
data2	0.14	0.142	0.08	12
data3	0.18	0.183	0.05	10
data4	0.22	0.224	0.03	8
data5	0.26	0.265	0.02	9
data6	0.3	0.306	0.01	11
data7	0.34	0.347	0.008	8
data8	0.38	0.388	0.005	7

Table 4.2: Filter specifications used in Figure 4.8

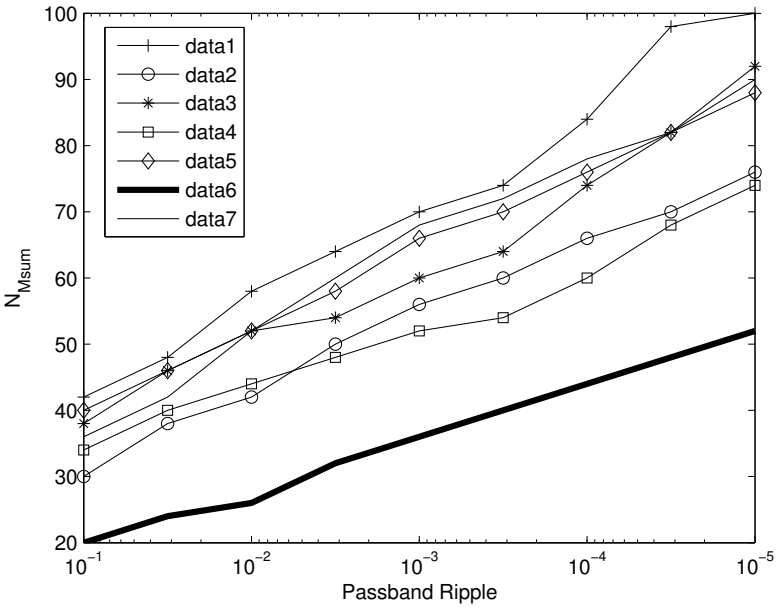

Figure 4.9: Relationship between N_{Msum} and passband ripple

Legend	f_p	f_s	δ_s	M
data1	0.12	0.122	0.1	15
data2	0.17	0.173	0.05	10
data3	0.21	0.214	0.03	11
data4	0.26	0.265	0.02	9
data5	0.32	0.326	0.01	10
data6	0.35	0.357	0.005	5
data7	0.39	0.4	0.001	8

Table 4.3: Filter specifications used in Figure 4.9

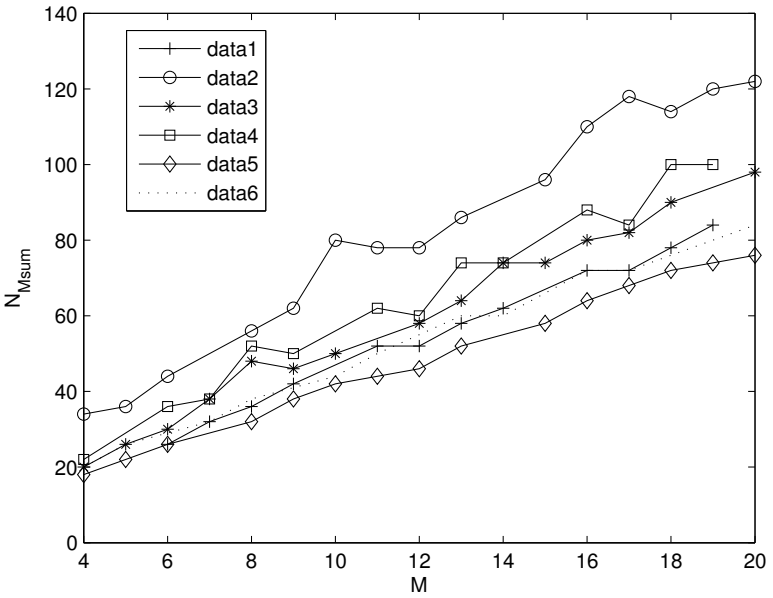

Figure 4.10: Relationship between the sum of the lengths of masking filters and interpolation factor M

Legend	f_p	f_s	δ_p	δ_s
data1	0.1	0.104	0.05	0.01
data2	0.14	0.146	3.16×10^{-3}	0.001
data3	0.18	0.185	0.01	0.01
data4	0.25	0.257	0.001	0.05
data5	0.285	0.288	0.001	0.001
data6	0.33	0.338	0.005	0.08

Table 4.4: Filter specifications used in Figure 4.10

Equations (4.9), (4.10) and (4.11) confirm that (4.8) is valid. The values of k, q and c in Equation (4.8) can be determined by the method of least squares fitting. A few hundreds of FRM filters with different specifications are designed by SQP. The obtained optimum interpolation factor and filter lengths for bandedges shaping filter and masking filters are used as data points for curve fitting. The final N_{Msum} is found to be

$$N_{Msum} = -1.15M \log_{10}(\delta_p \delta_s) + 0.02M + 5.24. \tag{4.12}$$

The second term in the above equation is much smaller than the other two terms. By ignoring it, we obtain the final filter length sum estimation formula:

$$N_{Msum} = -1.15M \log_{10}(\delta_p \delta_s) + 5.24. \tag{4.13}$$

4.5 Optimum Interpolation Factor

Based on Equations (4.5) and (4.13), the complexity of a FRM filter can be estimated by

$$\begin{aligned} N &= N_a + N_{Msum} \\ &= \frac{-9.9 \log_{10}(\delta_p \delta_s) - 8.6}{16.3 M \Delta F} - 1.15M \log_{10}(\delta_p \delta_s) + 8.57. \end{aligned} \tag{4.14}$$

An optimum M that minimizes the N can be easily found from the above equation if the partial derivative of N with respect to M is set to zero. This leads to the following equation.

$$M_{opt} = \sqrt{\frac{9.9 \log_{10}(\delta_p \delta_s) + 8.6}{18.75 \Delta F \log_{10}(\delta_p \delta_s)}}. \tag{4.15}$$

Generally speaking, $9.9\log_{10}(\delta_p\delta_s)$ is much larger than 8.6. If we ignore 8.6 and make a little compensation to it, a simplified estimation of M_{opt} is obtained as

$$M_{opt} \approx \frac{1}{\sqrt{2.25\Delta F}}. \tag{4.16}$$

To compare the accuracy of different estimations, two design examples are chosen for verification. The first example is taken from [32] with following specifications: $\delta_p = 0.01, \delta_s = 0.01, f_p = 0.3$ and $f_s = 0.305$. In [32], $M = 6$ and $M = 9$ are identified as optimum interpolation factors, where the total length of three subfilters are both 119. When nonlinear optimization technique is applied to this design, $M = 6$ leads to a design with the total length of 101 ($N_a = 67$, $N_{Ma} = 21$ and $N_{Mc} = 13$) [100]. For $M = 9$, the total length is reduced to 91 with $N_a = 45$, $N_{Ma} = 27$ and $N_{Mc} = 19$. Exhaustive search confirms that the optimum interpolation factor is 9 for this example. The frequency responses of subfilters and the overall filter are shown in Figures 4.11 and 4.12, respectively.

The second example is taken from [75] with following parameters: $\delta_p = 0.01, \delta_s = 0.001, f_p = 0.2$ and $f_s = 0.201$. In [75], two interpolation factors, $M = 16$ and $M = 21$, are used. $M = 16$ is calculated using (4.1), and $M = 21$ produces the best design. The total filter length is 267 for $M = 16$, with $N_a = 161$, $N_{Ma} = 48$ and $N_{Mc} = 58$. For $M = 21$, the total length is 257 with $N_a = 123$, $N_{Ma} = 56$ and $N_{Mc} = 78$.

Table 4.5 compares the accuracy of length estimation formulas given by Kaiser's equation and the proposed equations (4.5) and (4.13). Table 4.6 lists the esti-

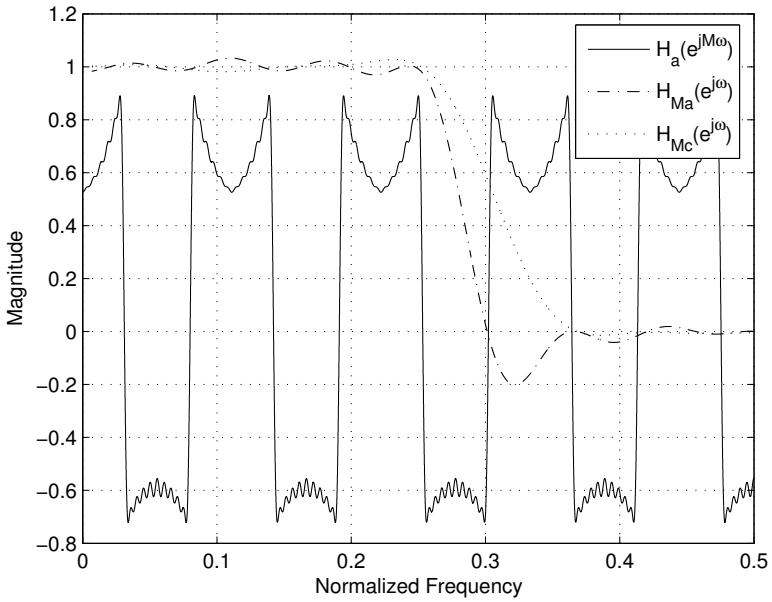

Figure 4.11: Frequency responses of various subfilters in example 1

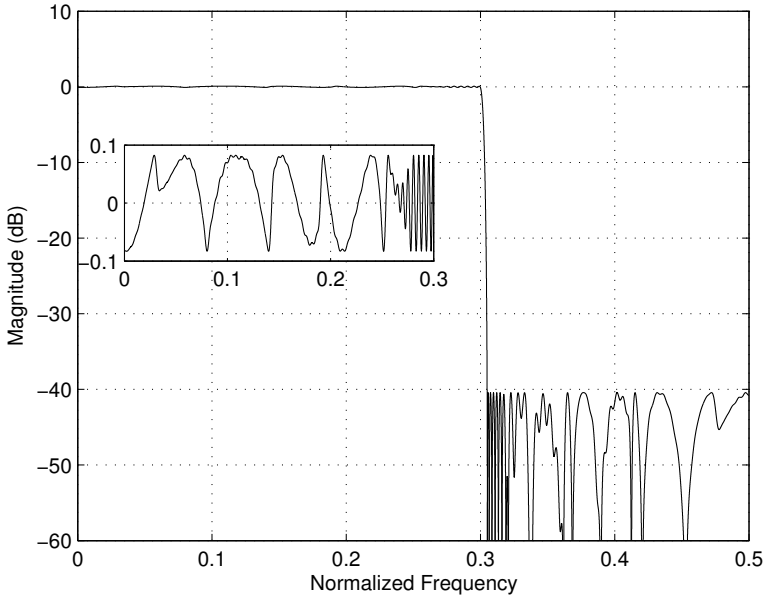

Figure 4.12: Frequency response of the overall filter in example 1

mation results of (4.1), (4.2), (4.15) and (4.16). It is clear from both tables that proposed design equations provide much better estimation.

Method	Design Example in [32]($M = 9$)		Design Example in [75]($M = 21$)	
	N_a	N_{Msum}	N_a	N_{Msum}
[32][a]	45	33+41=74	127	96+132=228
Proposed	45	47	123	126
Actual Value	45	46	123	134

Table 4.5: Comparison of filter length estimation

[a]Filter lengths are estimated by Kaiser's equation and ripples of each filter are set to 85% of the given specifications.

Equation	Design Example in [32]	Design Example in [75]
(4.1)	7	16
(4.2)	8	18
(4.15)	9	21
(4.16)	9	21
Actual Optimum M	9	21

Table 4.6: Comparison of interpolation factor

4.6 Conclusion

In this chapter, we have presented a set of new formulas for the estimation of the length of the prototype filter, the total length of two masking filters, and the optimum interpolation factor for a jointly optimized FRM filter. It is shown that an accurate optimum interpolation factor can be found if the Kaiser's filter length estimation formula is modified for the prototype filter and the total length of masking filters. About 200 design examples confirmed the accuracy of the proposed length estimation equations, and two design examples show the accuracy of the new interpolation factor estimation equations.

Chapter 5

Design of Frequency-Response Masking Filters With Even-Length Prototype Filters

The frequency-response masking (FRM) technique is one of the most computationally efficient techniques for the synthesis of narrow transition bandwidth FIR filters. In this chapter, we discuss problems faced in the original FRM approach when an even-length prototype filter is utilized. New FRM structures are proposed to make use of even-length prototype filters in the FRM approach. A design method based on sequential quadratic programming is presented. Design examples show that the proposed structures and design method lead to better results compared to original FRM filters with odd-length prototype filters.

5.1 Introduction

The frequency-response masking (FRM) technique proposed by Lim [32] has been proven to be one of the most efficient ways to design sharp FIR filters with arbitrary passband bandwidth. Much effort has been made to improve the FRM technique [48,54,70,75,77,79,85,86,91,92,99], including both theoretical analysis and practical design methods.

However, all reported FRM structures and optimization methods only employ an odd-length filter as the prototype filter. The usage of an even-length filter as the prototype filter in a FRM filter has never been explored. In fact, many problems surface if an even-length filter is used as the prototype filter. In this chapter, the detailed analysis will be first presented on why an even-length filter is not as good a candidate as an odd-length filter for the prototype filter. Solutions will be provided to overcome the shortcomings for an even-length prototype filter. Two new FRM structures suitable for even-length prototype filters are introduced in this chapter. Design examples show that the new FRM structures produce FRM filters better than that of original FRM structure.

This chapter is divided into five sections. Problems related to the usage of an even-length prototype filter in a FRM structure are revealed in Section 5.2. Section 5.3 develops the design method based on sequential quadratic programming (SQP). Modified FRM structures proposed for even-length prototype filters are introduced in Section 5.4. A conclusion is drawn in Section 5.5.

5.2 Ripple Analysis of FRM Using Even-length Prototype Filter

It is a well known fact that the zero phase passband magnitude of an interpolated low pass even-length symmetric FIR filter can be either 1 or -1 within the normalized frequency from 0 to 0.5. When such a filter is used as a prototype filter in a FRM structure, it produces a complementary prototype filter whose gain in the passband will be either 1 or 2, as shown in Figure 5.1(a). The z-transform transfer function of a FRM filter utilizing an even-length prototype filter is the same as the original FRM filters, and can be expressed as

$$H(z) = H_a\left(z^M\right) H_{Ma}(z) + \left[z^{-\frac{(Na-1)M}{2}} - H_a\left(z^M\right)\right] H_{Mc}(z). \tag{5.1}$$

Note that the interpolation factor M should be even when an even-length prototype filter is used in order to avoid the half sample delay. The zero phase frequency response of the overall filter is given by [32]

$$H(e^{j\omega}) = H_a\left(e^{jM\omega}\right) H_{Ma}\left(e^{j\omega}\right) + \left[1 - H_a\left(e^{jM\omega}\right)\right] H_{Mc}\left(e^{j\omega}\right). \tag{5.2}$$

The frequency responses of various subfilters in a FRM filter with an even-length prototype filter are shown in Figure 5.1. It is clear that there are a total of 4 transition bands that can be utilized to form the transition band of the overall filter as indicated in Figures 5.1(c), (e), (g) and (i) respectively. The cases shown in Figures 5.1(c) and (e) correspond to Cases A and B of the original FRM filter

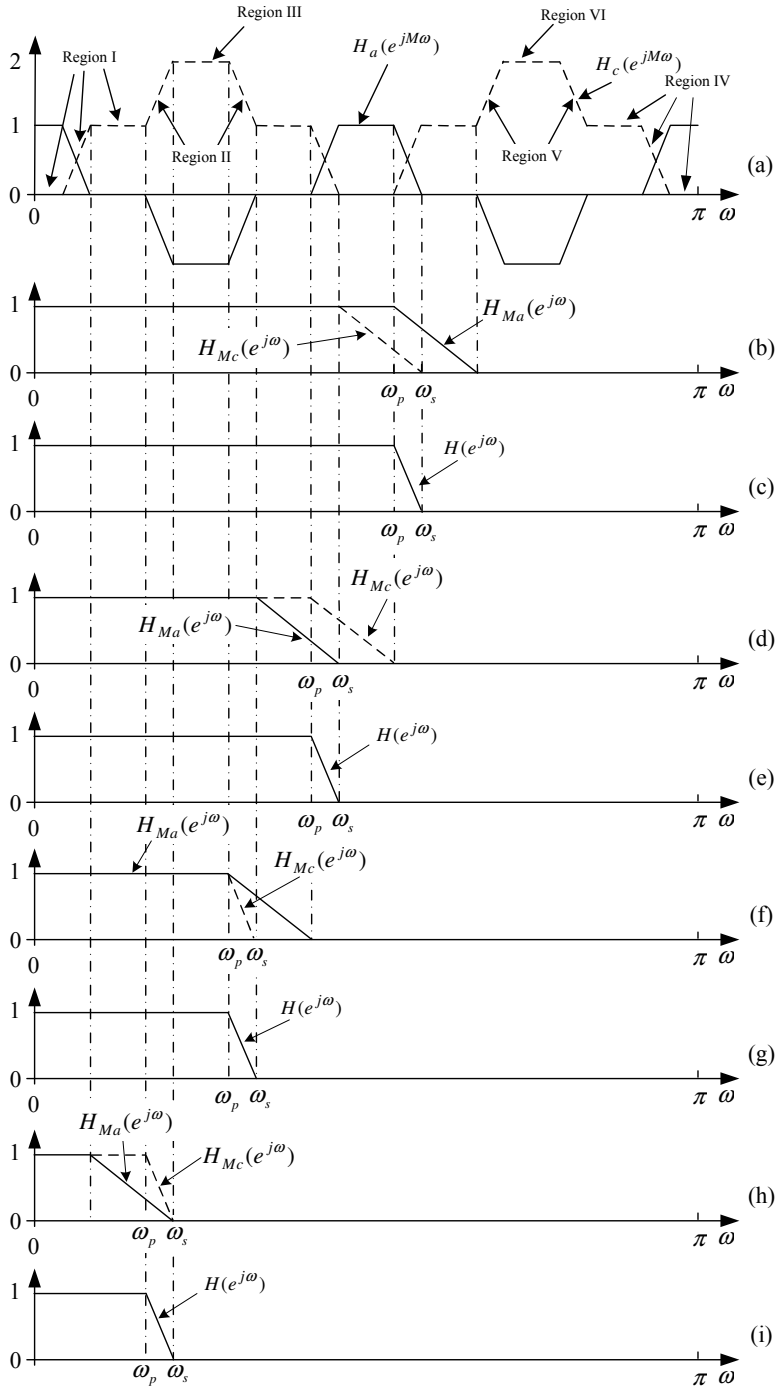

Figure 5.1: The frequency responses of subfilters and the overall FRM filter in the basic FRM filter with an even-length prototype filter (a) Interpolated prototype and complementary filter, (b) Two masking filters of Case A, (c) Overall FRM Filter of Case A, (d) and (e) Case B, (f) and (g) Case C, (h) and (i) Case D

with an odd-length prototype filter, respectively. The cases in Figures 5.1(g) and (i) exist only under the circumstance that an even-length prototype filter is employed. We denote the case in Figures 5.1(f) and (g) as Case C, and the case in Figures 5.1(h) and (i) as Case D. These two cases do not seem practical as they require the masking filter $H_{Mc}(z)$ having the same transition bandwidth as the overall filter, as shown in Figures 5.1(f) and (h).

To design a FRM filter based on an even-length prototype filter, it is interesting to know how the ripple of the overall filter is affected by the even-length prototype filter. Compared with the original FRM filter with an odd-length prototype filter, the protuberant shape of the complementary filter $H_c(e^{jM\omega})$ in frequency regions II, III, V and VI in Figure 5.1(a) may alter the ripple of the overall filter. A detailed analysis of ripples in frequency regions II, III, V and VI for Case A will be presented below. The same procedures can be applied to Cases B, C and D. For other frequency regions, the analysis is the same as in [32], which will not be repeated here.

Let us denote the desired value and deviation by $G(\omega)$ and $\delta(\omega)$, respectively, of the overall filter. Similarly, $G_p(\omega)$ and $\delta_p(\omega)$ are defined as the desired value and deviation, respectively, of the interpolated prototype filter, $G_{Ma}(\omega)$ and $\delta_{Ma}(\omega)$ are the desired value and deviation, respectively, of $H_{Ma}(e^{j\omega})$, and $G_{Mc}(\omega)$ and $\delta_{Mc}(\omega)$ are the desired value and deviation, respectively, of $H_{Mc}(e^{j\omega})$. Substitut-

ing the desired values and deviations into Equation (5.2), we have

$$\begin{aligned} G(\omega) + \delta(\omega) = &\left[G_p(\omega) + \delta_p(\omega)\right]\left[G_{Ma}(\omega) + \delta_{Ma}(\omega)\right] \\ &+ \left[1 - G_p(\omega) - \delta_p(\omega)\right]\left[G_{Mc}(\omega) + \delta_{Mc}(\omega)\right]. \end{aligned} \tag{5.3}$$

In Region II, which is the passband of the overall filter, $G(\omega) = G_{Ma}(\omega) = G_{Mc}(\omega) = 1$. Substituting this into Equation (5.3) and ignoring the second-order terms, we have

$$\delta(\omega) \approx G_p(\omega)\delta_{Ma}(\omega) + \left[1 - G_p(\omega)\right]\delta_{Mc}(\omega). \tag{5.4}$$

Note that we are only interested in the maximum value of $\delta(\omega)$. As $-1 < G_p(\omega) < 0$ in this region, the maximum ripple $\delta_M(\omega)$ is

$$\delta_M(\omega) = \left|\delta_{Ma}(\omega)\right| + 2\left|\delta_{Mc}(\omega)\right|. \tag{5.5}$$

Similarly, in the frequency region III, the ripple of the overall filter can be found by substituting $G_p(\omega) = -1$ into Equation (5.4),

$$\delta(\omega) = 2\delta_{Mc}(\omega) - \delta_{Ma}(\omega). \tag{5.6}$$

So the maximum ripple $\delta_M(\omega)$ should be

$$\delta_M(\omega)| = |\delta_{Ma}(\omega)| + 2|\delta_{Mc}(\omega)|. \tag{5.7}$$

In Region V, which is the stopband of the overall filter, $G(\omega)$, $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$ equal to 0. Substituting this into Equation (5.3) and ignoring the second-order terms, we have

$$\delta(\omega) \approx G_p(\omega)\delta_{Ma}(\omega) + \left[1 - G_p(\omega)\right]\delta_{Mc}(\omega). \tag{5.8}$$

We notice that Equation (5.8) is the same as Equation (5.4) and the value of $G_p(\omega)$ is equal to that in Region II. Thus, the maximum ripple can be expressed by Equation (5.5) for Region V. Following the same procedure, the maximum ripple $\delta_M(\omega)$ in Region VI is found to be the same as that in Region III.

According to Equations (5.5) and (5.7), the ripples in frequency regions II, III, V and VI can be much larger than the ripple of any one of the masking filters. If traditional design methods such as linear programming [32] or Remez method [70] are used to design each subfilter separately, the ripples of two masking filters must be no greater than half of the masking filter in the original FRM filter with an odd-length prototype filter. The reduced ripples of masking filters require longer masking filters. From the above analysis, it seems that an even-length prototype filter is not suitable for a FRM filter due to the large ripple in the frequency regions II, III, V and VI. However, an even-length prototype filter can still be utilized in a FRM filter if a nonlinear optimization technique is applied in the design of FRM filters. The design example in next section shows that an even-length prototype filter can result in less complexity in a FRM filter for the same interpolation factor M.

5.3 Design Method Based on Sequential Quadratic Programming

This section is divided into five subsections. The design task is first formulated as a minimax optimization problem. The details of the proposed optimization method are given in Sections 5.3.2 and 5.3.3. Based on the minimax optimization, Section 5.3.4 gives the design procedure of a FRM filter. In the last subsection, some design examples are presented.

5.3.1 Problem Formulation

According to (5.2), the frequency response of the overall FRM filter can be expressed as:

$$\begin{aligned} H(\omega, \boldsymbol{g}) &= [\boldsymbol{g}_a^T \boldsymbol{C}_a(\omega)][\boldsymbol{g}_{Ma}^T \boldsymbol{C}_{Ma}(\omega)] + \left[1 - \boldsymbol{g}_{\boldsymbol{a}}^T \boldsymbol{C}_a(\omega)\right][\boldsymbol{g}_{\boldsymbol{Mc}}^T \boldsymbol{C}_{Mc}(\omega)] \\ &= [\boldsymbol{g}_{\boldsymbol{a}}^T \boldsymbol{C}_a(\omega)][\boldsymbol{g}_{\boldsymbol{Ma}}^T \boldsymbol{C}_{Ma}(\omega) - \boldsymbol{g}_{\boldsymbol{Mc}}^T \boldsymbol{C}_{Mc}(\omega)] + \boldsymbol{g}_{\boldsymbol{Mc}}^T \boldsymbol{C}_{Mc}(\omega). \end{aligned} \tag{5.9}$$

If the prototype filter $H_a(z)$ is an odd-length FIR filter,

$$\begin{aligned} \boldsymbol{g}_{\boldsymbol{a}} = [&h_a((N_a+1)/2)\ 2h_a(1)\ 2h_a(2) \\ &\cdots 2h_a((N_a-3)/2)\ 2h_a((N_a-1)/2)]^T \end{aligned} \tag{5.10}$$

and the function $\boldsymbol{C}_a(\omega)$ should be

$$\boldsymbol{C}_a(\omega) = [1\ \cos((N_a-1)M\omega/2)\ \cdots\ \cos(2M\omega)\ \cos(M\omega)]^T. \tag{5.11}$$

If the prototype filter $H_a(z)$ is an even-length FIR filter,

$$\boldsymbol{g}_{\boldsymbol{a}} = [2h_a(1)\ 2h_a(2)\ \cdots\ 2h_a((N_a-2)/2)\ 2h_a(N_a/2)]^T \tag{5.12}$$

and the vector $\boldsymbol{C}_a(\omega)$ has the form

$$\boldsymbol{C}_a(\omega) = [\cos((N_a-1)M\omega/2) \ \cdots \ \cos(3M\omega/2) \ \cos(M\omega/2)]^T. \tag{5.13}$$

If two masking filters are of odd-length, then

$$\boldsymbol{g}_{Ma} = [h_{Ma}((N_{Ma}+1)/2) \ 2h_{Ma}(1) \ \cdots \ 2h_{Ma}((N_{Ma}-1)/2)]^T \tag{5.14}$$

$$\boldsymbol{C}_{Ma}(\omega) = [1 \ \cos((N_{Ma}-1)\omega/2) \ \cdots \ \cos(\omega)]^T \tag{5.15}$$

$$\boldsymbol{g}_{Mc} = [h_{Mc}((N_{Mc}+1)/2) \ 2h_{Mc}(1) \ \cdots \ 2h_{Mc}((N_{Mc}-1)/2)]^T \tag{5.16}$$

$$\boldsymbol{C}_{Mc}(\omega) = [1 \ \cos((N_{Mc}-1)\omega/2) \ \cdots \ \cos(\omega)]^T. \tag{5.17}$$

Otherwise, they are both of even-length, and

$$\boldsymbol{g}_{Ma}=[2h_{Ma}(1) \ 2h_{Ma}(2) \ \cdots \ 2h((N_{Ma}-2)/2) \ 2h(N_{Ma}/2)]^T \tag{5.18}$$

$$\boldsymbol{C}_{Ma}(\omega)=[\cos((N_{Ma}-1)M\omega/2) \ \cdots \ \cos(3\omega/2) \ \cos(\omega/2)]^T \tag{5.19}$$

$$\boldsymbol{g}_{Mc}=[2h_{Mc}(1) \ 2h_{Mc}(2) \ \cdots 2h((N_{Mc}-2)/2) \ 2h(N_{Mc}/2)]^T \tag{5.20}$$

$$\boldsymbol{C}_{Mc}(\omega)=[\cos((N_{Mc}-1)M\omega/2) \ \cdots \ \cos(3\omega/2) \ \cos(\omega/2)]^T. \tag{5.21}$$

The $\boldsymbol{g}$ in Equation (5.9) has the form

$$\boldsymbol{g} = [\boldsymbol{g}_a^T \ \boldsymbol{g}_{Ma}^T \ \boldsymbol{g}_{Mc}^T]^T. \tag{5.22}$$

Suppose the passband edge and stopband edge are at ω_p and ω_s, respectively, then the ideal frequency response of the overall filter is

$$H_d(\omega) = \begin{cases} 1 & \omega \in [0, \ \omega_p] \\ 0 & \omega \in [\omega_s, \ \pi]. \end{cases} \tag{5.23}$$

And the error function is defined as

$$Error(\omega, \boldsymbol{g}) = W(\omega)|H(\omega, \boldsymbol{g}) - H_d(\omega)| \tag{5.24}$$

where

$$W(\omega) = \begin{cases} 1 & \omega \in [0,\ \omega_p] \\ \delta_p/\delta_s & \omega \in [\omega_s,\ \pi] \end{cases} \tag{5.25}$$

The design is carried out over a selected set of dense grid points $\omega_k \in \boldsymbol{\Omega} = [\,0,\ \omega_p\,] \cup [\,\omega_s,\ \pi\,]$. Suppose there is a total of n grid points in $\boldsymbol{\Omega}$, and the maximum value of $E(\omega, \boldsymbol{g})$ over $\boldsymbol{\Omega}$ is γ. The design problem can be formulated as a minimax problem:

$$\begin{aligned} &\min_{\boldsymbol{g}}\ \gamma \\ &\text{subject to: } \boldsymbol{E} \leq \boldsymbol{\Gamma} \end{aligned} \tag{5.26}$$

where $\boldsymbol{g}$ is the coefficient vector yet to be determined to minimize the merit function $\phi(\boldsymbol{g}) \equiv \max(\boldsymbol{E})$, and $\boldsymbol{E}$ is given by

$$\boldsymbol{E} = \begin{bmatrix} Error(\omega_1, \boldsymbol{g}) \\ Error(\omega_2, \boldsymbol{g}) \\ \vdots \\ Error(\omega_n, \boldsymbol{g}) \end{bmatrix}$$

and $\boldsymbol{\Gamma} = [\,\gamma\ \gamma\ \cdots\ \gamma\,]^T_{n\times 1}$.

5.3.2 Design Method Based on SQP

To utilize SQP method to solve the minimax problem, (5.26) should be first converted to a quadratic programming problem. In [17], Brayton et. al. demonstrated that for a positive definite matrix $\tilde{\boldsymbol{H}}$, the step $\Delta \boldsymbol{g} \neq 0$ obtained by solving

$$\begin{aligned} \min_{\boldsymbol{g}} \quad & \tfrac{1}{2}\Delta \boldsymbol{g}^T \tilde{\boldsymbol{H}} \Delta \boldsymbol{g} + \Delta\gamma \\ \text{subject to: } & \boldsymbol{E} + \left(\nabla_{\boldsymbol{g}} \boldsymbol{E}^T\right) \Delta \boldsymbol{g} \leq \boldsymbol{\Gamma} + \Delta\boldsymbol{\Gamma} \end{aligned} \tag{5.27}$$

is in a direction of descent for the function $\phi(\boldsymbol{g}) \equiv \max(\boldsymbol{E})$. That is to say that the $\Delta \boldsymbol{g}$ obtained in (5.27) can make $\max(\boldsymbol{E})\big|_{\boldsymbol{g}=\boldsymbol{g}_0+\epsilon\Delta\boldsymbol{g}}$ less than $\max(\boldsymbol{E})\big|_{\boldsymbol{g}=\boldsymbol{g}_0}$, where $\boldsymbol{g}_0$ is a start point, and ϵ is a suitable small positive number.

Define a new variable $\boldsymbol{z} = [\,\gamma\ \boldsymbol{g}^T\,]^T$, and $\Delta \boldsymbol{z}^T = [\,\Delta\gamma\ \Delta\boldsymbol{g}^T\,]^T$. So we can transform (5.27) into

$$\begin{aligned} \min_{\boldsymbol{z}} \quad & \tfrac{1}{2}\Delta \boldsymbol{z}^T \boldsymbol{H} \Delta \boldsymbol{z} + \boldsymbol{c}^T \Delta \boldsymbol{z} \\ \text{subject to:} \quad & \boldsymbol{A}\Delta \boldsymbol{z} \leq \boldsymbol{b} \end{aligned} \tag{5.28}$$

where $\boldsymbol{H} = \begin{bmatrix} 0 & \mathbf{0}_{1\times m} \\ \mathbf{0}_{m\times 1} & \tilde{\boldsymbol{H}} \end{bmatrix}$, $\boldsymbol{c} = [1\ \mathbf{0}_{1\times m}]^T$, $\boldsymbol{A} = [-\mathbf{1}_{m\times 1}\ \nabla_{\boldsymbol{g}}\boldsymbol{E}^T]$ and $\boldsymbol{b} = \boldsymbol{\Gamma} - \boldsymbol{E}$, m is the length of $\boldsymbol{g}$, $\mathbf{0}_{i\times j}$ is a $i \times j$ zero matrix, and $-\mathbf{1}_{m\times 1}$ is a column vector whose elements are all -1. In (5.27) and (5.28), $\nabla_{\boldsymbol{g}}\boldsymbol{E}$ is calculated according to the parity of the length of each subfilter in a FRM filter, referring to Equations (5.9)-(5.21). The problem (5.28) can be solved by the method introduced in [6], or the Matlab function *quadprog* or *qp* [68].

5.3.3 Hessian Matrix Update

In (5.27), it is required that $\tilde{\boldsymbol{H}}$ is positive definite. The initial value can be the identity matrix $\boldsymbol{I}$. However, if $\tilde{\boldsymbol{H}}$ is fixed to be the identity matrix $\boldsymbol{I}$, the convergence speed will be slow. In [17], Brayton et al. pointed out that BFGS updating method works well for (5.28). Here, the Hessian matrix $\boldsymbol{H}$ is updated by the BFGS method [26].

The update of Hessian matrix $\boldsymbol{H}$ is according to the Lagrangian function of (5.26). The Lagrangian function is given by [17],

$$L = \sum_{i=1}^{n} \lambda_i Error(\omega_i, \boldsymbol{g}). \tag{5.29}$$

According to (5.29), the Lagrangian function is independent of γ. Therefore, the partial derivative $\partial L/\partial\gamma = 0$. For the t^{th} iteration, we have

$$\nabla_{\boldsymbol{z}} L^{(t)}(\boldsymbol{g}, \lambda^{(t)}) = \begin{bmatrix} 0 \\ \nabla_{\boldsymbol{g}} L^{(t)}(\lambda^{(t)}) \end{bmatrix}. \tag{5.30}$$

Define $\boldsymbol{d} = \nabla_{\boldsymbol{z}} L(\boldsymbol{g}^{(t+1)}, \lambda^{(t+1)}) - \nabla_{\boldsymbol{z}} L(\boldsymbol{g}^{(t)}, \lambda^{(t+1)})$. According to (5.30), we have

$$\begin{aligned} \boldsymbol{d} &= \begin{bmatrix} 0 \\ \nabla_{\boldsymbol{g}} L^{(t+1)}(\lambda^{(t+1)}) \end{bmatrix} - \begin{bmatrix} 0 \\ \nabla_{\boldsymbol{g}} L^{(t)}(\lambda^{(t+1)}) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \nabla_{\boldsymbol{g}} L^{(t+1)}(\lambda^{(t+1)}) - \nabla_{\boldsymbol{g}} L^{(t)}(\lambda^{(t+1)}) \end{bmatrix}. \end{aligned} \tag{5.31}$$

Calculate

$$\xi = \Delta \boldsymbol{g}^T \tilde{\boldsymbol{H}}^{(t)} \Delta \boldsymbol{g}$$

$$\psi = \Delta \boldsymbol{g}^T [\nabla_{\boldsymbol{g}} L^{(t+1)}(\lambda^{(t+1)}) - \nabla_{\boldsymbol{g}} L^{(t)}(\lambda^{(t+1)})]$$

$$\theta = \begin{cases} 1 & \text{if } \Delta \boldsymbol{z}^T \boldsymbol{d} \geq 0.2 \Delta \boldsymbol{g}^T \tilde{\boldsymbol{H}}^{(t)} \Delta \boldsymbol{g} \\ \dfrac{0.8\xi}{\xi - \psi} & \text{otherwise.} \end{cases} \tag{5.32}$$

Then calculate

$$\boldsymbol{w} = \theta \boldsymbol{d} + (1 - \theta) \boldsymbol{H}^{(t)} \Delta \boldsymbol{z} \tag{5.33}$$

where

$$\begin{aligned} \boldsymbol{H}^{(t)} \Delta \boldsymbol{z} &= \begin{bmatrix} 0 & \mathbf{0}_{1 \times m} \\ \mathbf{0}_{m \times 1} & \tilde{\boldsymbol{H}}^{(t)} \end{bmatrix} \begin{bmatrix} \triangle \gamma \\ \Delta \boldsymbol{g} \end{bmatrix} \\ &= \begin{bmatrix} 0 & (\tilde{\boldsymbol{H}}^{(t)} \Delta \boldsymbol{g})^T \end{bmatrix} .^T \end{aligned} \tag{5.34}$$

According to (5.31) to (5.34), the first component of $\boldsymbol{w}$ is zero, so $\boldsymbol{w}$ can be expressed as

$$\boldsymbol{w} = \begin{bmatrix} 0 \\ \tilde{\boldsymbol{w}} \end{bmatrix} . \tag{5.35}$$

The initial value of $\tilde{\boldsymbol{H}}$ is an identity matrix $\boldsymbol{I}$. The $\tilde{\boldsymbol{H}}$ is updated as

$$\tilde{\boldsymbol{H}}^{(t+1)} = \tilde{\boldsymbol{H}}^{(t)} - \frac{\tilde{\boldsymbol{H}}^{(t)} \Delta \boldsymbol{g} \Delta \boldsymbol{g}^T \tilde{\boldsymbol{H}}^{(t)}}{\Delta \boldsymbol{g}^T \tilde{\boldsymbol{H}} \Delta \boldsymbol{g}} + \frac{\tilde{\boldsymbol{w}} \tilde{\boldsymbol{w}}^T}{\Delta \boldsymbol{g}^T \tilde{\boldsymbol{w}}}. \tag{5.36}$$

5.3.4 Design Procedure

In Sections 5.3.2 and 5.3.3, we have presented an algorithm based on SQP for solving the minimax problem. When this algorithm is applied to the FRM filter design, the detailed design procedure is given as follows

1. Determine the initial solution $\boldsymbol{g}_0$ as in [32] or [70];

2. Set $\tilde{\boldsymbol{H}}$ to be an identity matrix $\boldsymbol{I}$, and construct matrix $\boldsymbol{H}$. Solve problem (5.28) by calling *quadprog* or *qp* function in Matlab, and obtain the search direction;

3. Determine the step size ϵ by

$$\min_{\epsilon}\{\max\{\boldsymbol{E}\big|_{\boldsymbol{g}=\boldsymbol{g}_0+\epsilon\Delta\boldsymbol{g}}\}\} \tag{5.37}$$

4. If $\left|\max\{\boldsymbol{E}\big|_{\boldsymbol{g}=\boldsymbol{g}_0+\epsilon\Delta\boldsymbol{g}}\}-\max\{\boldsymbol{E}\big|_{\boldsymbol{g}=\boldsymbol{g}_0}\}\right|\leqslant\varepsilon$, terminate the iterative process. Here, ε is a predetermined tolerance.

5. Update Hessian matrix $\boldsymbol{H}$ according to Equations (5.31)-(5.36).

6. Go back to Step 2.

It should be pointed out that the algorithm above can only result in a local minimum point over the design space. Different initial solution should be tried to find a near-global optimal solution.

5.3.5 Design Example

To illustrate the proposed SQP design method, let us redesign the example in [32], i.e. normalized passband and stopband edges are at 0.3 and 0.305, respectively. The ripples in both passband and stopband are 0.01. The interpolation factor is set to 9, and the subfilter lengths are $N_a = 45$, $N_{Ma} = 41$ and $N_{Mc} = 33$. The passband and stopband ripples of our design and those reported in [32] and [91] are listed in Table 5.1. It is obvious that our new design method results in the smallest passband and stopband ripples. The frequency response of our design is shown in Figure 5.2.

	Passband Ripple (dB)	Stopband Attenuation (dB)
Design Result in [32]	0.0896	40.96
Design Result in [91]	0.0674	42.25
Our Design Result	0.0661	42.40

Table 5.1: Comparison of different design results

If an even-length prototype filter is used in the FRM structure of Figure 4.1 for the same specifications, the interpolation factor M must be even to avoid half sample delay. Therefore, M is chosen to be 6. Subfilter lengths, passband ripple δ_p, and stopband attenuation δ_s are listed in Table 5.2. The frequency response of the FRM filter with an even-length prototype filter is shown in Figure 5.3.

It is obvious that an even-length prototype filter results in less taps in the given

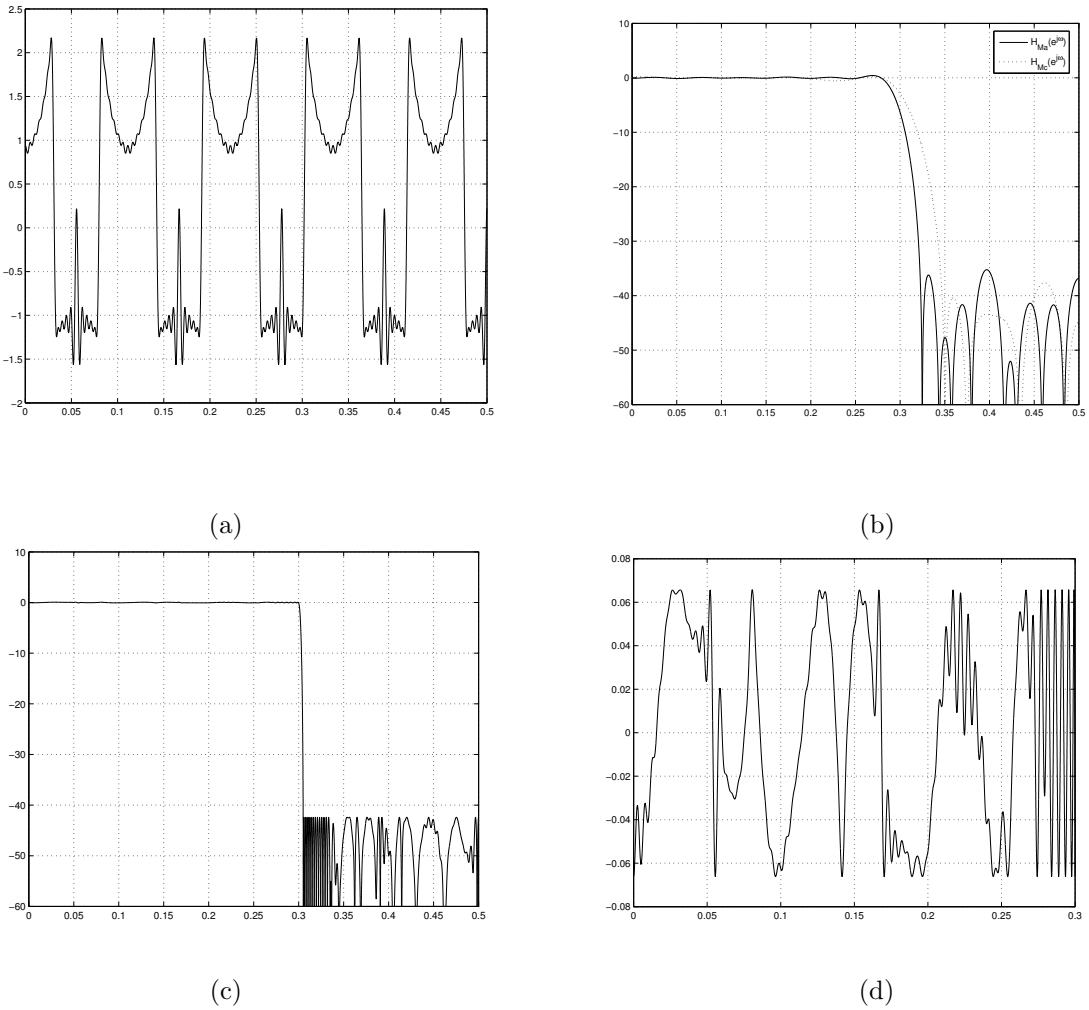

Figure 5.2: Frequency response of (a) prototype filter $H_a(z^9)$, (b) masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, (c) overall filter, and (d) passband ripples of the overall filter

example when the interpolation factor is 6. However, the interpolation factor is limited to an even integer. Such a constraint significantly reduces the design space and may lead to a suboptimum solution. For the given example, the best FRM design requires 47 taps ($N_a = 45,\ N_{Ma} = 27,\ N_{Mc} = 19$) with the interpolation factor of 9, which is 4 taps less than the FRM with an even-length prototype filter designed above whose interpolation factor M is 6. Therefore, it is necessary to find new FRM filter structures enabling odd interpolation factors in a FRM filter with an even-length prototype filter. In next section, new structures will be proposed to solve the problems mentioned above.

Filter structure	N_a	N_{Ma}	N_{Mc}	δ_p (dB)	δ_s (dB)
Even-length prototype filter	68	13	19	0.0843	40.31
Odd-length prototype filter	67	13	21	0.0818	40.57

Table 5.2: Ripple comparison of FRM filters with odd-length and even-length prototype filters

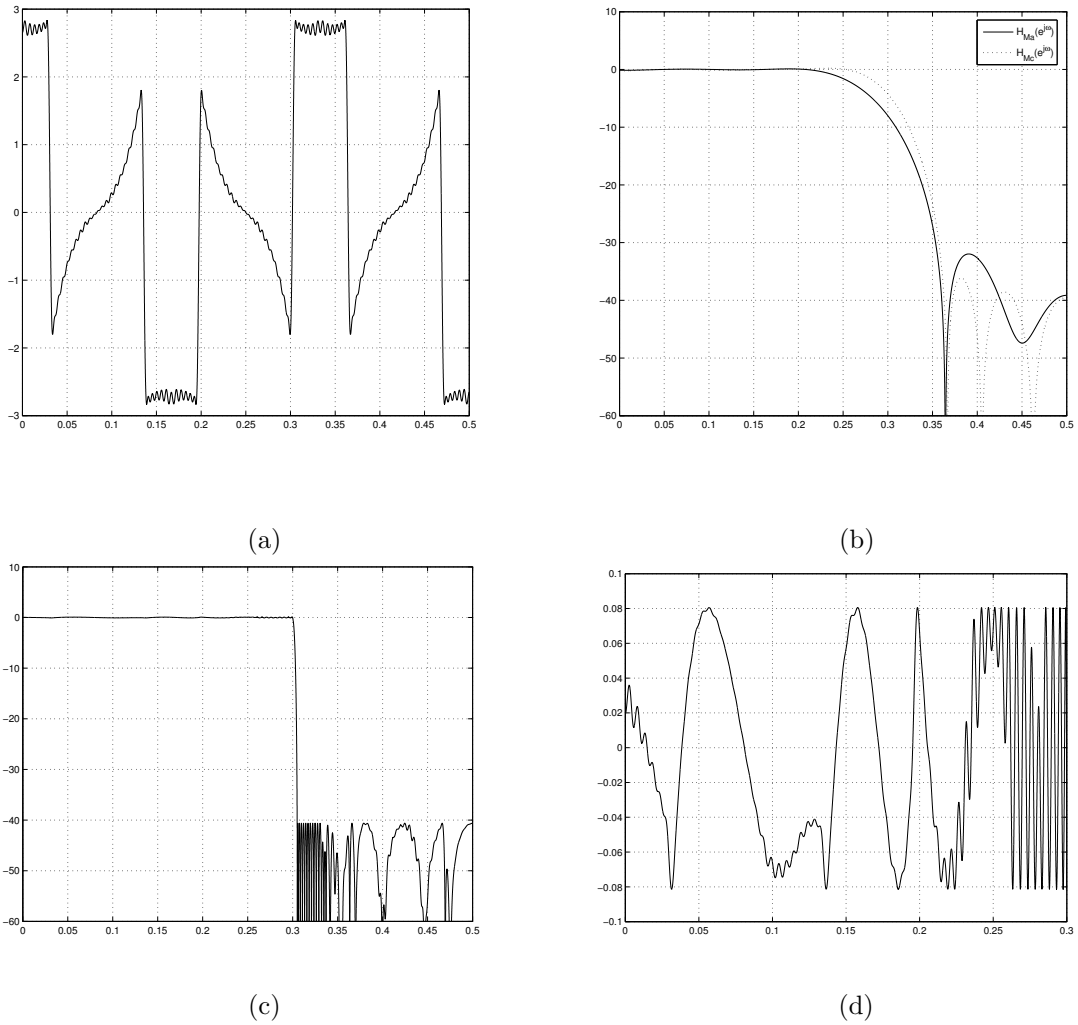

Figure 5.3: Frequency response of (a) prototype filter $H_a(z^6)$, (b) masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, (c) overall filter, and (d) passband ripples of the overall filter

5.4 Modified Structures for FRM with Even-Length Prototype Filters

Note from Figure 4.1 that the input signal passes through $(N_a - 1)/2$ delays in order to form a complementary output. If N_a is an odd number, $(N_a - 1)M/2$ is always a natural number. However, if N_a is an even number, M must be an even number to avoid half sample delay. This requirement prohibits the usage of an even-length prototype filter when M is odd.

To utilize an even-length prototype filter when M is odd, we modify the original FRM filter structure into one shown in Figure 5.4, which is suitable for Case A in Figures 5.1(b)-(c). We call this structure as modified FRM structure I. The frequency responses of various subfilters in the modified FRM structure I are shown in Figure 5.5. The frequency response of $F_a(z)$ now is

$$\begin{aligned} F_a(e^{j\omega}) &= H_a(z^M)H_{Ma}(z)\big|_{z=e^{j\omega}} \\ &= e^{-j\omega(MN_a+N_{Ma}-M-1)/2}R_a(M\omega)R_{Ma}(\omega) \end{aligned} \tag{5.38}$$

where $R_a(\omega)$ and $R_{Ma}(\omega)$ are the magnitude frequency response of the prototype filter $H_a(z)$ and the masking filter $H_{Ma}(z)$, respectively. The solid line in Figure 5.5(c) shows the shape of $F_a(e^{j\omega})$. The frequency response of its complement is

$$\begin{aligned} F_c(e^{j\omega}) &= z^{-\frac{(MNa+N_{Ma}-M-1)}{2}} - F_a(z)|_{z=e^{j\omega}} \\ &= e^{-j\omega(MN_a+N_{Ma}-M-1)/2} \cdot [1 - R_a(M\omega)R_{Ma}(\omega)]. \end{aligned} \tag{5.39}$$

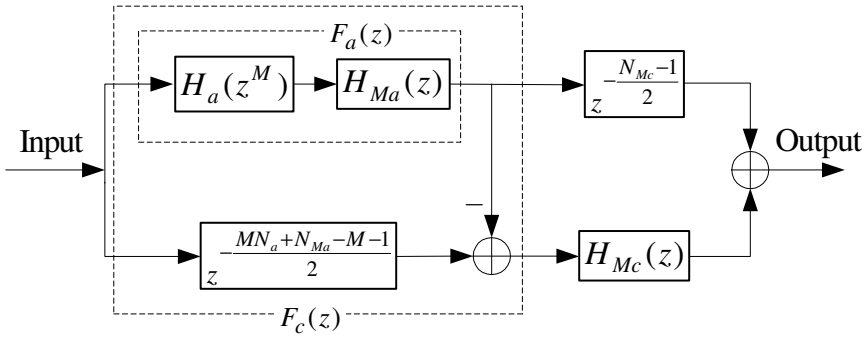

Figure 5.4: Modified FRM structure I

The dashed line in Figure 5.5(c) shows the frequency response of the complementary part. Masking filter $H_{Mc}(z)$, whose frequency response is the dashed line as shown in Figure 5.5(b), removes the additional passband beyond ω_s. The frequency response of the overall filter is shown in Figure 5.5(d). The z-transform transfer function of the modified FRM filter I is given by

$$\begin{aligned} H(z) = z^{-\frac{N_{Mc}-1}{2}} H_a(z^M) H_{Ma}(z) + \\ [z^{-\frac{MNa+N_{Ma}-M-1}{2}} - H_a(z^M) H_{Ma}(z)] H_{Mc}(z). \end{aligned} \tag{5.40}$$

In the modified FRM structure I, $(MN_a + N_{Ma} - M - 1)/2$ delay components are used to form the complement filter $F_c(z)$. It is easy to avoid half sample delay in the proposed structure, i.e. use an even-length masking filter $H_{Ma}(z)$ when M is odd, or an odd-length $H_{Ma}(z)$ when M is even.

The modified structure FRM I is suitable for any interpolation factor with an even-length prototype filter. It should be mentioned that the modified structure FRM I causes slight increase in group delay compared with the original FRM structure. The increase in group delay depends on the length of the masking

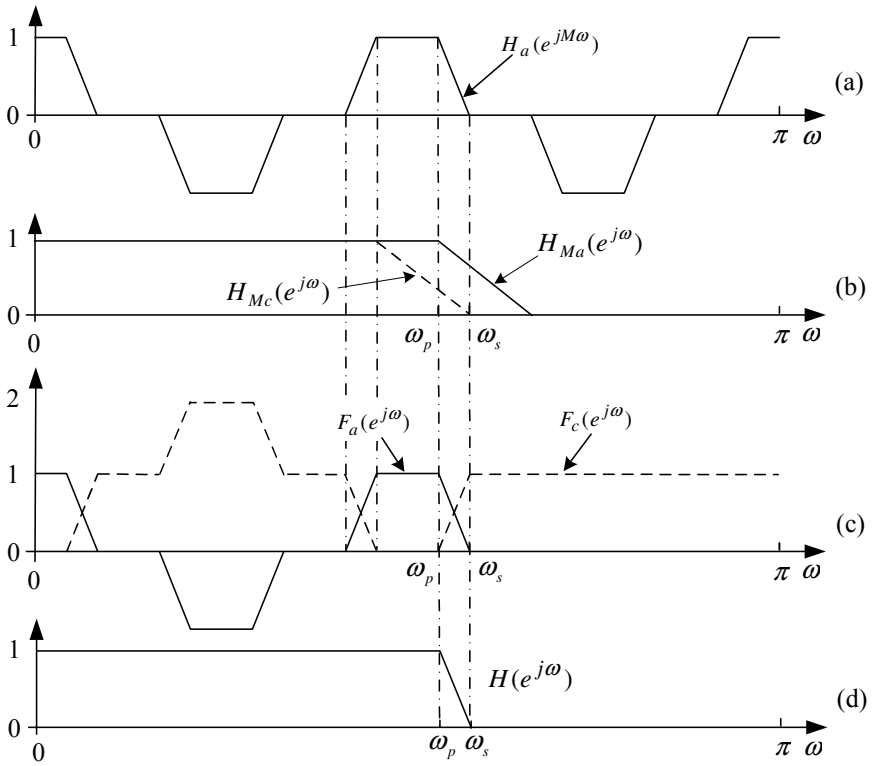

Figure 5.5: Frequency response of the modified FRM structure I

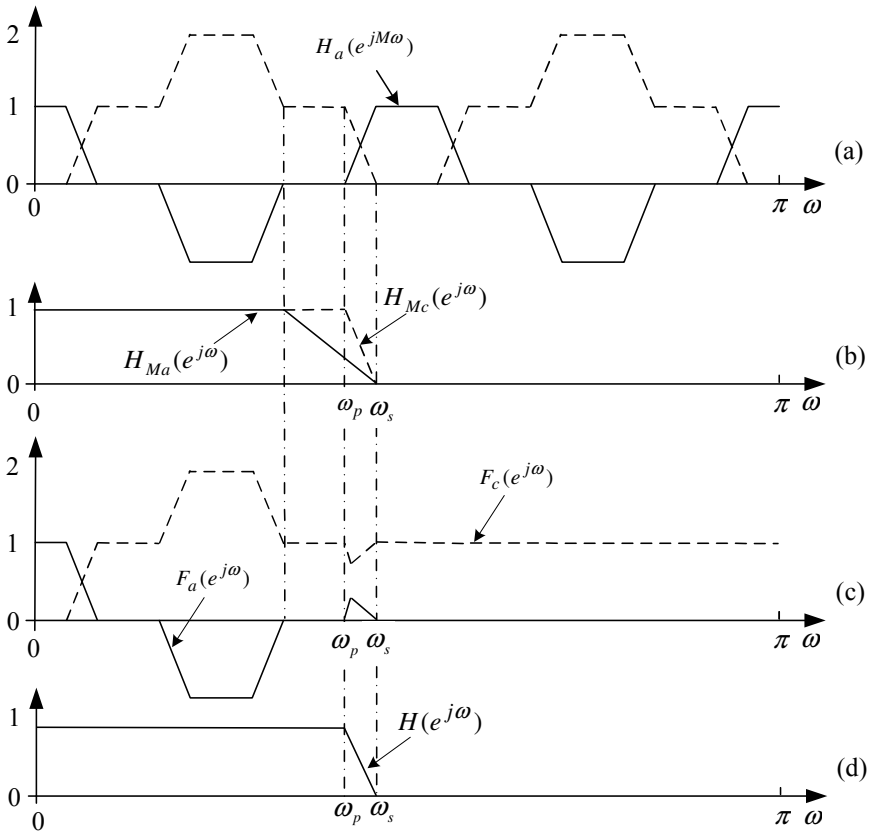

Figure 5.6: Frequency response of modified structure I for Case B

filter $H_{Ma}(z)$. However, the amount of increased group delay is not significant because of two reasons. First, nonlinear optimization techniques such as SQP helps reducing the length of $H_{Ma}(z)$. Moreover, the group delay of the overall filter is mainly determined by $H_a(z^M)$. The group delay of $H_{Ma}(z)$ contributes only to a small percentage of the overall filter's group delay.

The modified FRM structure I proposed above is only suitable for Case A. For Case B, the proposed structure is not very efficient because it requires the masking filter $H_{Mc}(z)$ to have the same transition bandwidth as the overall filter. Figure 5.6 shows the frequency response of the modified structure I for Case B.

To avoid the sharp transition band for the masking filter $H_{Mc}(z)$, we can change the masking filter $H_{Ma}(z)$ to a bandpass filter. The new FRM structure is shown in Figure 5.7. The frequency responses of various subfilters are shown in Figure 5.8. The frequency response of $F_a(z) = H_a(z^M)H_{Ma}(z)$ is now a bandpass signal with a sharp transition band. The transition bandwidth of the low pass masking filter $H_{Mc}(z)$ is extended compared with the one in Figure 5.6. This new structure is denoted as modified FRM structure II. The z-transform transfer function of is given by

$$H(z) = [z^{-\frac{MN_a+N_{Ma}-M-1}{2}} - H_a(z^M)H_{Ma}(z)]H_{Mc}(z). \tag{5.41}$$

The idea of using a bandpass masking filter $H_{Ma}(z)$ can be applied to Cases C and D of the FRM filter with an even-length prototype filter. For Cases C and D shown in Figures 5.1(f)-(i), it requires the masking filter $H_{Mc}(z)$ to have the same transition bandwidth as the overall filter. Such a requirement makes these cases unattractive, although SQP and other nonlinear optimization techniques can extend the transition bandwidth of $H_{Mc}(z)$ to some extent.

If $H_{Ma}(z)$ is chosen to be a bandpass filter with passband gain of -1 for Case C as shown in Figure 5.9(b), the frequency response of $F_a(z)$ will be a bandpass signal with sharp transition bandwidth. The transition bandwidth of $H_{Mc}(z)$ can be extended. Figure 5.9(b) shows the frequency response of $H_{Mc}(z)$ for Case C. The corresponding $F_a(e^{j\omega})$ and $F_c(e^{j\omega})$ are shown in Figure 5.9(c). The frequency response of the overall filter for Case C is shown in Figure 5.9(d). The realization

structure for Case C is the same as that in Figure 5.4. The difference between Case A and Case C is that the masking filter $H_{Ma}(z)$ in Case A is a low pass filter with passband gain equal to 1, while the masking $H_{Ma}(z)$ filter in Case C is a bandpass filter with passband gain of -1.

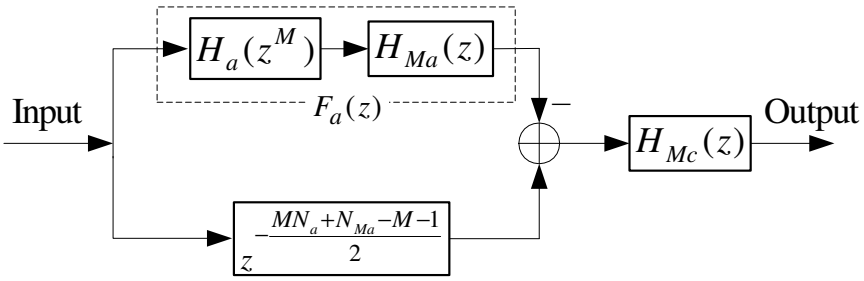

Figure 5.7: The realization structure of modified FRM II

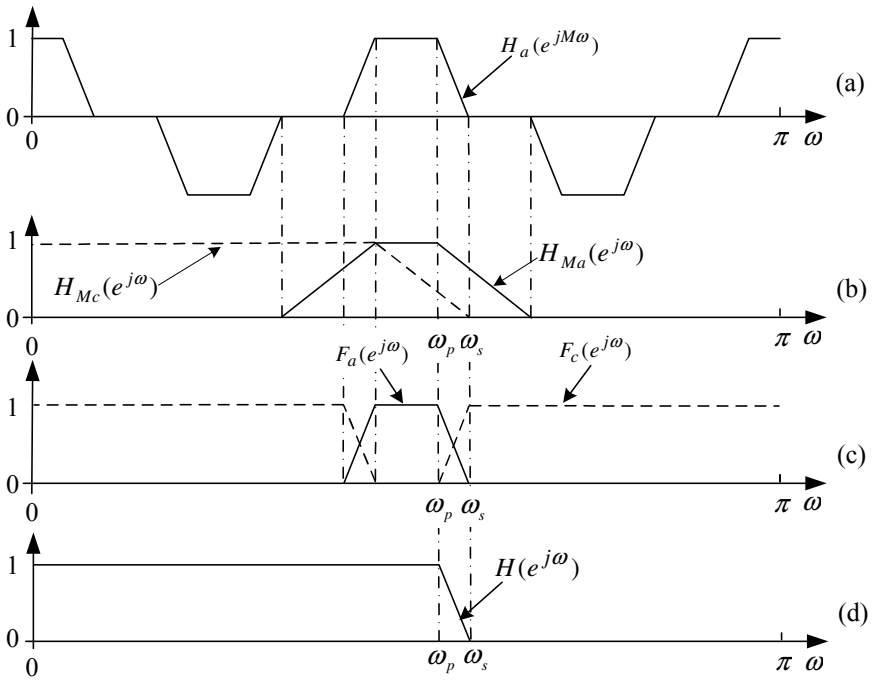

Figure 5.8: Frequency response of each subfilter in modified FRM structure II

For Case D, it can be implemented using the structure in Figure 5.7. The frequency responses of various subfilters are shown in Figures 5.9(e) and (f).

In both modified FRM structures I and II, the masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$ are not in parallel. Therefore, their lengths are not required to have the same parity to keep the same group delay.

To illustrate our new proposed structure for FRM filter with an even-length prototype filter, we redesign the low pass filter taken from [32]. The normalized passband and stopband edges are 0.3, and 0.305, respectively. The passband and stopband ripples are 0.01. To satisfy the given specification, the original FRM requires an interpolation factor of 9. Case D can be used. Table 5.3 lists designs using different approaches. It is easy to see that the even-length prototype filter requires the least total number of coefficients. The frequency responses of the FRM filter with an even-length prototype filter is drawn in Figure 5.10.

Another design example is taken from [75] and [91]. The normalized passband and stopband edges are at 0.2 and 0.201, respectively. The passband and stopband ripples are 0.01 and 0.001, respectively. The interpolation factor M is set to 21. It is of Case A of FRM filter with even-length prototype filter. Table 5.4 shows the comparison of different approaches. It is obvious that an even-length prototype filter results in further savings in terms of the number of coefficients. The frequency responses of the FRM filter with an even-length prototype filter are shown in Figure 5.11.

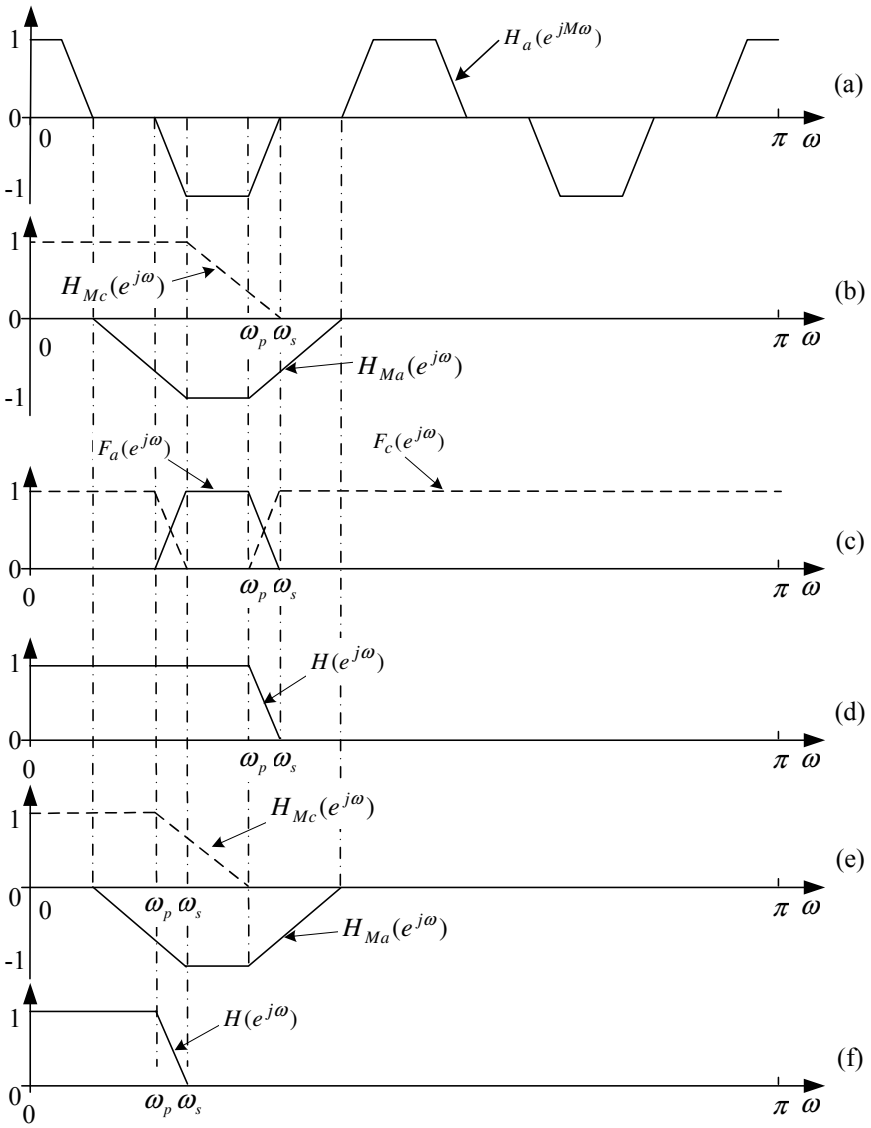

Figure 5.9: Frequency responses of each subfilter and the overall FRM filter in modified FRM structure II of Cases C and D (a) Prototype filter (b) Two masking filters for Case C (c) $F_a(e^{j\omega})$ and $F_c(e^{j\omega})$ (d) Overall FRM of Case C (e) Two masking filters for Case D (f) Overall FRM of Case D

Filter structure	N_a	N_{Ma}	N_{Mc}	δ_p (dB)	δ_s (dB)
Odd-length prototype filter (SQP approach)	45	27	19	0.086	-40.15
Odd-length prototype filter (WLS Approach [79])	45	28	22	≈ 0.087	$\approx -40^a$
Even-length prototype filter (SQP Approach)	46	20	21	0.084	-40.34

Table 5.3: Comparison of design results from different design methods

[a]Exact data not provided by the author.

Filter structure	N_a	N_{Ma}	N_{Mc}
Odd-length prototype filter in [75] and [91]	123	56	78
Even-length prototype filter	124	52	71

Table 5.4: Subfilter length comparison of different approaches

5.5 Conclusion

In this chapter, we first analyzed the problems in a FRM structure using an even-length FIR filter as the prototype filter. To overcome the difficulties caused by even-length prototype filters, the SQP optimization technique is utilized to design the three subfilters simultaneously. The given examples show that a FRM structure with an even-length prototype filter has a better performance than that

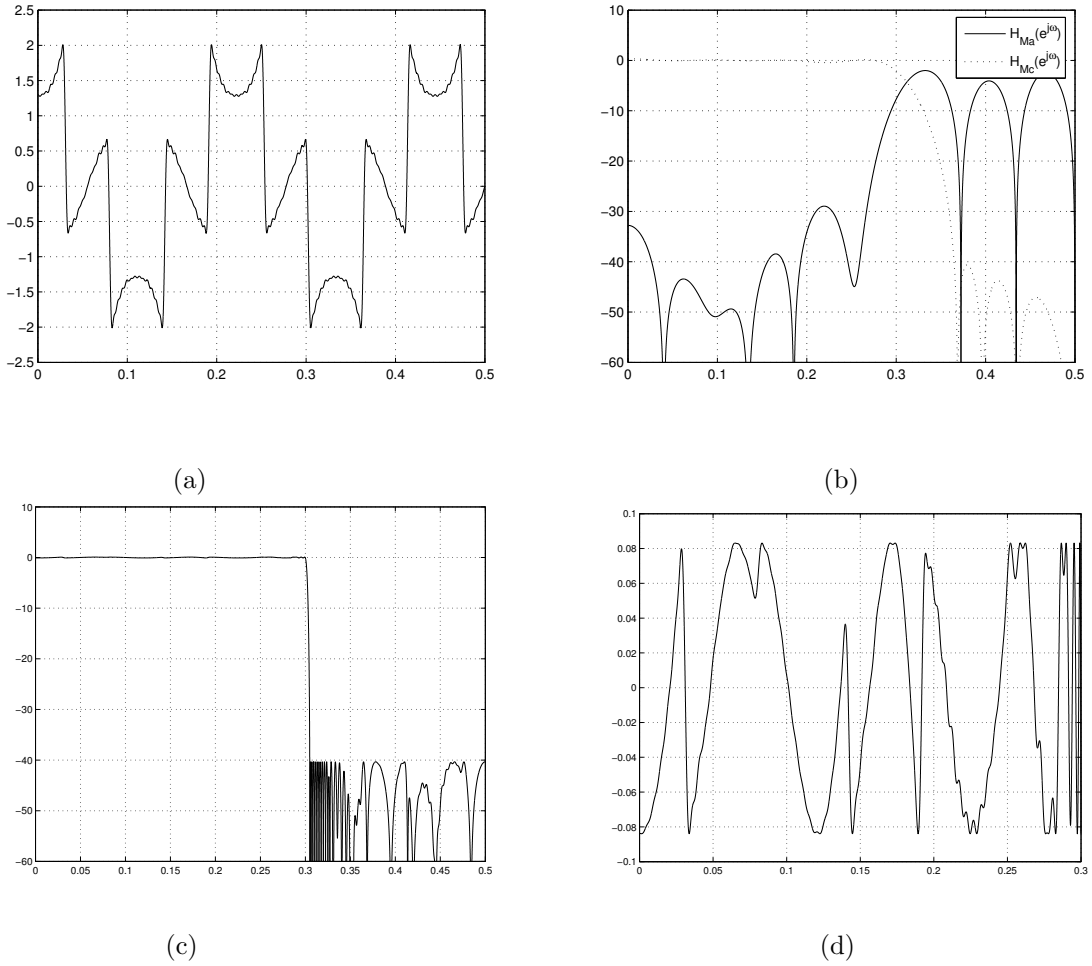

Figure 5.10: Frequency Response of (a) Prototype filter $H_a(z^9)$, (b) masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, (c) Overall filter, and (d) passband ripple of the overall filter

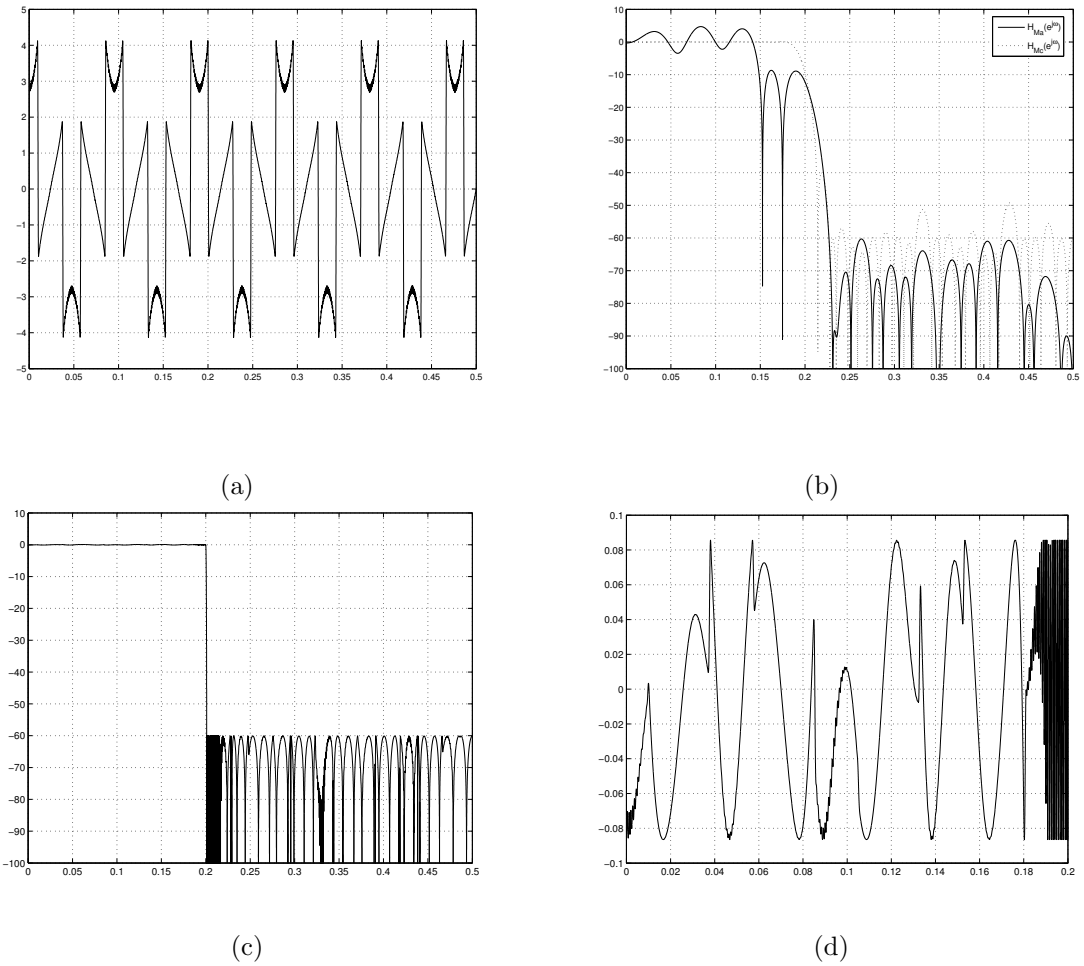

Figure 5.11: Frequency Response of (a) Prototype filter $H_a(z^{21})$, (b) masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$, (c) Overall filter, and (d) passband ripple of the overall filter

of a FRM with an odd-length prototype filter. New structures suitable for even-length prototype FRM filters were proposed. It was shown, by means of design examples, that the proposed FRM structures yield better results compared with the original FRM structures. For all the nonlinear optimization techniques, they all reshape parts of the interpolated prototype filter to have a negative magnitude. If the prototype is even-length, this reshaping will be easier. Therefore, an even-length prototype filter and further reduces the complexity in a FRM filter.

Chapter 6

A Dynamic Frequency Grid Point Allocation Scheme for Efficient Design of Frequency-Response Masking FIR Filters

6.1 Introduction

In the past few years, various new methods have been proposed for the design of optimum frequency-response masking (FRM) filters [32]. Besides traditional linear programming [32] and Remez exchange methods [70, 83], several nonlinear optimization techniques have been utilized to design FRM filters, including the

two-step optimization method [75], the weighted least square (WLS) approach [79,101], the semi-definite programming (SDP) method [91], and the second-order cone programming (SoCP) method [92, 99].

No matter which design method is utilized, the frequency response of a FRM filter is optimized on a dense set of frequency grid points. Obviously, the density of the frequency grid points must be sufficient enough so that the given specifications are not violated at frequencies in between any two frequency grid points. However, there has been no report on how to determine such a dense frequency grid for the design of a FRM filter until now. For the case of a single FIR filter, Yang and Lim [46] proposed that the frequency grid spacing λ should be

$$\lambda = \frac{0.5 - \sum_i \Delta_i}{k \cdot N} \tag{6.1}$$

where $\sum_i \Delta_i$ is the total normalized transition bandwidth, N is the filter length, and k is a grid spacing factor to be determined. Here, deviation of a filter is defined as the maximum value of ripple. A larger value k makes the actual deviation $\hat{\delta}$ closer to δ, where δ is the deviation detected by the frequency grid points, at the expense of more memory space and CPU time [46]. To address this issue, Yang and Lim proposed a dynamic frequency grid point allocation scheme for the efficient design of equiripple FIR filters [62]. However, the methods in [46] and [62] are not well suited for FRM filters. The reasons are two-fold. First, the parameters in Equation (6.1) are hard to determine in case of FRM filters, especially when a nonlinear optimization technique is utilized to design a FRM filter. According to our experience, the grid scheme determined according to

Equation (6.1) is too sparse for a FRM filter design when a nonlinear optimization technique is utilized.

Second, the dynamic allocation scheme in [62] is based on the fact that the frequency positions of local extrema of a symmetrical FIR filter do not shift a lot between two consecutive optimization iterations. Therefore, the algorithm in [62] could only take into account frequency grid points corresponding to local ripple extrema. For a FRM filter that involves a few subfilters, each iteration during the optimization may lead to a local extremum that is quite different from the previous iteration. This phenomena makes dynamic allocation scheme fail to guarantee convergence under WLS method for a FRM filter. Moreover, the convergence of a single WLS iteration does not result in an equiripple solution as reported in [47, 79]. The weighting function must be updated (such as the algorithm in [47]) several times before a quasi-equiripple solution is found. This results in a two-loop algorithm in [62]. For FRM filters, the WLS approach such as [79] generally requires much more computations than a single FIR filter does, due to the nonlinear relationship between the ripple of the overall filter and the ripple of each subfilter. In this case, the total computation load will become very high for the design of a very sharp FRM filter.

In this chapter, a new dynamic frequency grid point allocation scheme is proposed. The new allocation scheme is suitable for the efficient design of a FRM filter. The organization of this chapter is as follows: in Section 6.2, the ripple of FRM filters designed by nonlinear optimization techniques is analyzed first.

In Section 6.3, a two-stage design method is proposed, and the detailed design procedure is given. Section 6.4 discusses how the sparse and dense frequency grid points are allocated. Section 6.5 gives different convergence criteria for sparse and dense frequency grid point allocation scheme. A design example is shown in Section 6.6 to demonstrate the effectiveness of the new frequency grid point allocation scheme, and conclusion is drawn in Section 6.7.

6.2 Ripple Analysis for Jointly Optimized FRM Filters

A basic FRM structure is shown in Figure 4.1. In this section, we only analyze the ripples of low pass FRM filters. Suppose the passband and stopband edges of a low pass prototype filter $H_a(z)$ are θ and ϕ, respectively, as shown in Figure 6.1(a). When nonlinear optimization techniques are utilized in the design of low pass FRM filters, the bandwidths of two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$ can be extended up to $(2\pi-\phi+\theta)/M$ for Case A, or $(2\pi+\phi-\theta)/M$ for Case B, where M is the interpolation factor of the prototype filter. For a low pass FRM filter, the transition bands of two masking filters are assumed to be in the frequency regions $[(2m\pi-\theta)/M,\ (2(m+1)\pi-\phi)/M]$ for Case A (see Figure 6.1(c)), and $[(2(m-1)\pi+\theta)/M,\ (2m\pi+\phi)/M]$ for Case B (see Figure 6.1 (e)). Design examples in [75, 79, 91, 92, 99] support the above assumption. m is an integer less in M [32]. Figure 6.2 shows the frequency responses of three subfilters of

a low pass FRM filter designed by a nonlinear optimization technique. As the distribution of extrema of a FRM filter determines the density of the frequency grid points, let us first analyze the ripple of a FRM filter jointly optimized by a nonlinear optimization technique before the presentation of a dynamic frequency grid point allocation scheme.

For the sake of expository clarity, we drop the phase frequency response in the rest of this section. Let $G_a(\omega)$ and $\delta_a(\omega)$ be the desired value and deviation of the frequency response of the interpolated prototype filter, $H_a(e^{jM\omega})$. Similarly, let $\{G_{Ma}(\omega), \delta_{Ma}(\omega)\}$ and $\{G_{Mc}(\omega)$, and $\delta_{Mc}(\omega)\}$ be the desired value and deviation of the frequency responses of the two masking filters, $H_{Ma}(e^{j\omega})$ and $H_{Mc}(e^{j\omega})$, respectively. Now we can express the desired value $G(\omega)$ and deviation $\delta(\omega)$ of the overall FRM filter as

$$\begin{aligned} G(\omega) + \delta(\omega) = {} & [G_a(\omega) + \delta_a(\omega)][G_{Ma}(\omega) + \delta_{Ma}(\omega)] \\ & +[1 - G_a(\omega) - \delta_a(\omega)][G_{Mc}(\omega) + \delta_{Mc}(\omega)]. \end{aligned} \tag{6.2}$$

Due to different values of $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$, we can divide the whole frequency region into two regions as indicated in Figures 6.1(c) and (e): in Region I $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$ have constant value, and in Region II $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$ drop from 1 to 0. The effects of $H_a(e^{jM\omega})$, $H_{Ma}(e^{j\omega})$ and $H_{Mc}(e^{j\omega})$ on the ripple of the overall FRM filter will be examined in different frequency regions one by one.

Frequency Region 1: In this region, $G_{Ma}(e^{j\omega}) = G_{Mc}(e^{j\omega}) = G(\omega) = C$. This region includes the passband and stopband of two masking filters: $C = 1$ in the

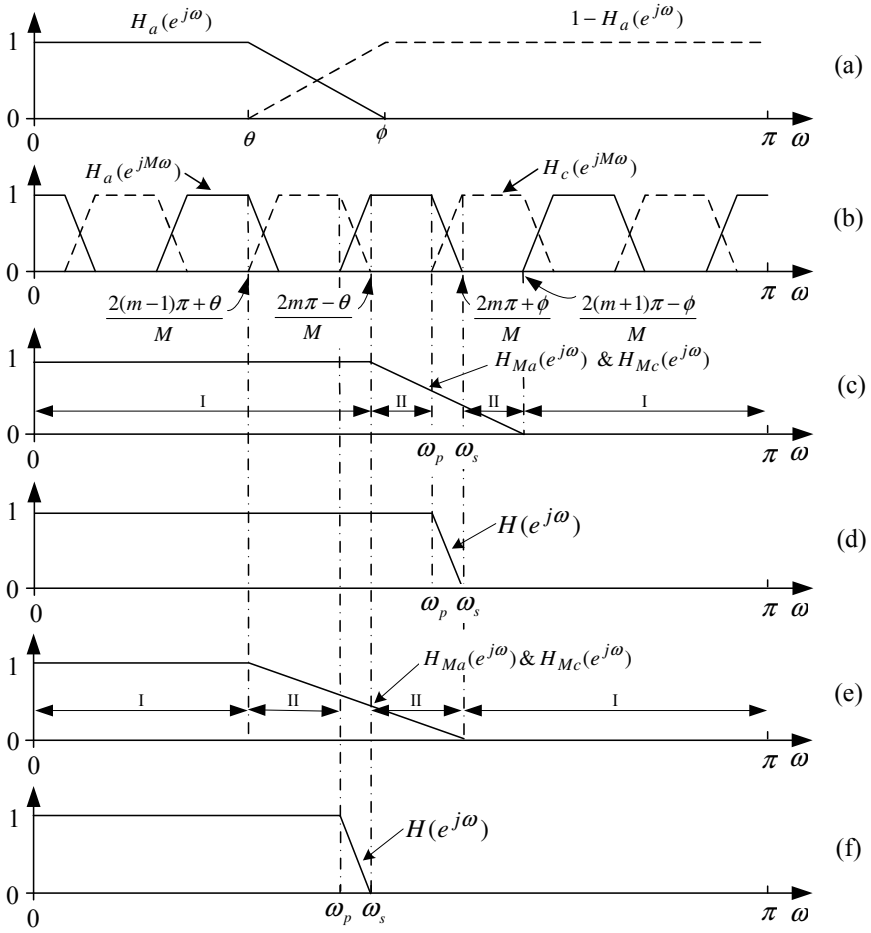

Figure 6.1: Frequency response of each subfilter when nonlinear optimization methods utilized (a) prototype filter and complementary filter, (b) interpolated prototype filter and complementary filter, (c) two masking filters for Case A, (d) Overall FRM filter for Case A, (e) two masking filters for Case B and (e) Overall FRM filter for Case B

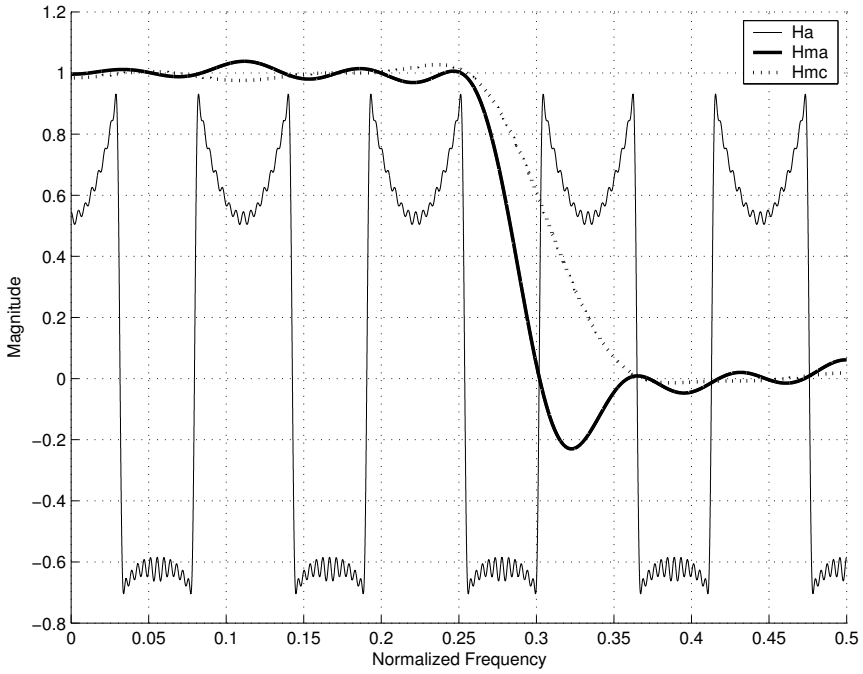

Figure 6.2: Design example of a FRM filter designed by SQP

passband or $C = 0$ in the stopband. For a low pass FRM filter, it is the region $[0,\ (2m\pi-\theta)/M]\cup[(2(m+1)\pi-\phi)/M,\pi]$ for Case A, or $[0,\ (2(m-1)\pi+\theta)/M]\cup[(2m\pi+\phi)/M,\pi]$ for Case B. Substituting the value of C into Equation (6.2) and ignoring second order terms, Equation (6.2) can be simplified to

$$\delta(\omega) = G_a(\omega)[\delta_{Ma}(\omega) - \delta_{Mc}(\omega)] + \delta_{Mc}(\omega). \tag{6.3}$$

In this region, $G_a(\omega)$ is no longer a constant value when a nonlinear optimization method is utilized. The value of $\delta(\omega)$ is determined by $G_a(\omega)$, $\delta_{Ma}(\omega)$ and $\delta_{Mc}(\omega)$.

An important problem we must consider is what is the number of extrema in a FRM filter. The number of extrema of two masking filters are determined by the lengths of $H_{Ma}(z)$ and $H_{Mc}(z)$, which are the same as the case of a single FIR filter. For the interpolated prototype filter, $G_a(\omega)$ is no longer a constant

value of 1 or 0 in the passband and stopband. Actually it has arch-shape as shown in Figure 6.2. At each bandedge of $H_a(z^M)$, $G_a(\omega)$ has an extremum. At $\omega = k\pi/M$ where k is an integer, $G_a(\omega)$ has another extremum. Therefore, in the interval $[2(m-1)\pi/M,\ 2m\pi/M)$, there are total 6 extrema for $G_a(\omega)$. Meanwhile, there are totally $6M+1$ extrema in $[0,\ \pi]$. From Figure 6.1(b), it is clear that the number of extrema of $G_a(\omega)$ in the passband is $6m$ for Case A, or $6m-4$ for Case B. Figures 6.1(b), (c) and (e) reveal that there are 4 extrema for $G_a(\omega)$ in the transition bands of two masking filters for Case A and 6 extrema for Case B. Therefore, in the stopband of two masking filters, $G_a(\omega)$ has $6M-6m-3$ extrema for Case A, or $6M-6m-1$ extrema for Case B. These extrema of $G_a(\omega)$ increase the number of extrema in the FRM filters. Moreover, the nonconstant $G_a(\omega)$ makes $\delta(\omega)$ a nonlinear combination of $\delta_{Ma}(\omega)$ and δ_{Mc}. This also increases the number of extrema in Frequency Region 1. The number of extrema of $\delta(\omega)$ can now be classified into 2 groups: one group is caused by the ripples of two masking filters, and another group is caused by the nonconstant gain of the prototype filter. Therefore, more frequency grid points are needed to deal with the extrema caused by $G_a(\omega)$ besides the frequency grid points needed by $H_{Ma}(z)$ and $H_{Mc}(z)$.

Frequency Region 2: This region is the transition bands of two masking filters $H_{Ma}(z)$ and $H_{Mc}(z)$. For Case A, the region is $((2m\pi-\theta)/M, (2(m+1)\pi-\phi)/M)$. For Case B, it is $((2(m-1)\pi+\theta)/M, (2m\pi+\phi)/M)$. In this region, $\delta_{Ma}(\omega) = \delta_{Mc}(\omega) = 0$ as $H_{Ma}(z)$ and $H_{Mc}(z)$ are both in their transition band. Therefore,

Equation (6.2) can be simplified to

$$\begin{aligned} G(\omega) + \delta(\omega) \ &= G_{Mc}(\omega) + G_a(\omega)[G_{Ma}(\omega) - \\ &\quad G_{Mc}(\omega)] + \delta_a(\omega)[G_{Ma}(\omega) - G_{Mc}(\omega)]. \end{aligned} \tag{6.4}$$

Equation (6.4) can be decomposed into two equations:

$$G(\omega) = G_{Mc}(\omega) + G_a(\omega)[G_{Ma}(\omega) - G_{Mc}(\omega)] \tag{6.5}$$

and

$$\delta(\omega) = \delta_a(\omega)[G_{Ma}(\omega) - G_{Mc}(\omega)]. \tag{6.6}$$

Here, $G(\omega) = 1$ in the passband of the overall FRM filter, and $G(\omega) = 0$ in the stopband of the overall FRM filter.

According to Equation (6.6), the deviation $\delta(\omega)$ in Frequency Region 2 is determined by the deviation of the interpolated prototype filter $H_a(z^M)$, and the gain of two masking filter $H_{Ma}(z)$ and $H_{Mc}(z)$. Because $\delta(\omega)$ is dependent on $\delta_a(\omega)$, the number of local extrema of $\delta(\omega)$ in this region is related to the filter length of the prototype filter $H_a(z)$ and its interpolation factor M. In this region, the two masking filters are both in their transition bands, and their gains drop from 1 to 0 gradually. As the two endpoints of this region are not included in this region, 1 and 0 are two limits of both $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$, but not their extrema. One of $G_{Ma}(\omega)$ and $G_{Mc}(\omega)$ may have an extremum in this region, to guarantee the tenableness of (6.5), as shown in Figure 6.2 and design examples in [75, 79, 91, 92, 99]. As a result, the number of extrema of $\delta(\omega)$ in Frequency Region 2 is mainly determined by the extrema of $\delta_a(\omega)$. In other words, the

number of frequency grid points in Frequency Region 2 are determined by the length of the prototype filter $H_a(z)$ and the interpolation factor M.

With the above analysis, we are ready to propose a new set of conditions to determine the frequency grid spacing based on Equation (6.1). In Frequency Region 1, it has been shown that the number of extrema of a FRM filter is determined by all three subfilters. N in Equation (6.1) takes the value of longer filter length between $H_{Ma}(z)$ and $H_{Mc}(z)$. k in Equation (6.1) should have a larger value than the value in the case of a single FIR filter, to provide extra frequency grid points needed by $G_a(\omega)$. For the case of a single FIR filter, k takes a value of $8 \sim 10$ [46]. We recommend that k takes value of $15 \sim 20$ for a FRM filter. In Frequency Region 2, the number of extrema is mainly determined by the filter length of $H_a(z)$ and its interpolation factor M. N is M times of the length of filter $H_a(z)$. k can be slightly larger than that in the case of a single FIR filter. We suggest that k takes a value of $10 \sim 15$.

6.3 A New Two-Stage Design Method Based on Sequential Quadratic Programming

It is possible to jointly optimize all the subfilters in a FRM structure using the WLS approach [79]. However, the WLS approach requires updating the weighting function after each convergence of the WLS design vector. The fact that a local

extremum may move makes a dynamic frequency grid point scheme not suitable for WLS approach. In practice, the above fact make it almost impossible for a WLS approach to obtain an equiripple solution for a FRM filter if a sparse frequency grid point scheme such as the scheme in [62] is used.

When the sequential quadratic programming (SQP) technique [17] is used, weighting function does not need updating no matter what frequency grid point allocation scheme is adopted. Details of the design procedure of the SQP technique can be found in Section 5.3, and will not be repeated here.

Although SQP is efficient for jointly optimizing all the subfilters in a FRM structure, the high density of the required frequency grid point increases the computing amount, and reduces the design efficiency. To further improve the efficiency, a sparse frequency grid point allocation scheme is highly desired. This leads to the two-stage design method described below, which uses a sparse dynamic frequency grid point allocation scheme in the first stage.

The new two-stage design method is based on the SQP optimization technique. Figure 6.3 shows the design procedure of our new method. During initialization, the initial value of the design vector $\boldsymbol{h} = [\boldsymbol{h}_a^T,\ \boldsymbol{h}_{Ma}^T,\ \boldsymbol{h}_{Mc}^T]^T$ is determined as described in [32] or [70], where $\boldsymbol{h}_a,\ \boldsymbol{h}_{Ma},\ \boldsymbol{h}_{Mc}$ are the coefficients of $H_a(z)$, $H_{Ma}(z)$, and $H_{Mc}(z)$. An initial set of sparse frequency grid points is also determined during initialization. There are two stages in the proposed method: a sparse frequency grid is first used followed by a much higher dense frequency grid. In Stage

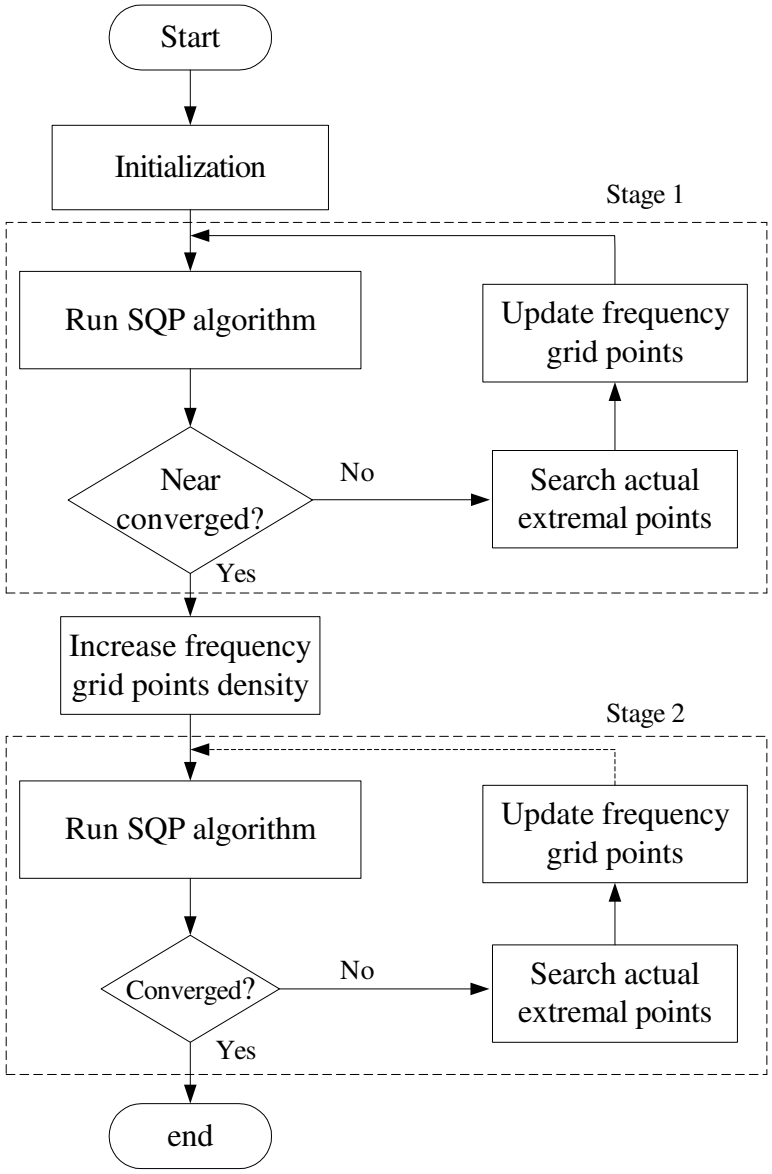

Figure 6.3: Flowchart for the dynamic frequency grid point scheme

1, the SQP iterative procedure is terminated after several iterations, and the optimization convergence status is analyzed. If the optimization result does not meet the condition of "near convergence", which will be defined in Section 6.5, actual extremal points are searched, and a new sparse set of frequency grid points is formed. A new SQP iterative procedure is carried out again. If the condition of "near convergence" is satisfied, a dense set of frequency grid points is formed, and steps in Stage 2 are followed. The convergence condition in Stage 2 is different from the decision in Stage 1. The difference between convergence conditions in the two stages will be discussed later in Section 6.5. Other steps in Stage 2 are the same as that in Stage 1.

The purpose of using a sparse frequency grid in Stage 1 is to save memory and CPU time. However, a problem faced by the sparse frequency grid is that the actual extrema may never be detected by the sparse frequency grid. The actual deviation $\hat{\delta}$ which is detected by a very dense frequency grid points will be larger than the deviation δ detected by the sparse frequency grid points. This problem also exists in a fixed dense frequency grid scheme, which can be solved by increasing the number of frequency grid points at the expense of increased memory requirement, more CPU time, and reduced efficiency. However, increasing the number of frequency grid points is not suitable for the sparse frequency grid point scheme. To detect the actual deviation $\hat{\delta}$ while not reducing the efficiency, the sparse frequency grid is updated regularly and a dense frequency grid is used in Stage 2 to guarantee the given specifications are satisfied everywhere in the

frequency domain. Although the memory and CPU time requirement for each iteration of SQP algorithm in Stage 2 is high, the needed iteration numbers in Stage 2 can be controlled to be within an acceptable level. To minimize the number of optimization iterations in Stage 2, it becomes very important in Stage 1 to decide on the convergence of $\boldsymbol{h}$, which will be discussed in Section 6.5.

6.4 Dynamic Frequency Grid Point Allocation Scheme

During initialization, a sparse frequency grid is formed by choosing N in Equation (6.1) to be the length of the longer filter between $H_{Ma}(z)$ and $H_{Mc}(z)$ in Frequency Region 1. In Frequency Region 2, N should be M times the length of $H_a(z)$. Because the initial frequency grid points are used during the first iteration in Stage 1, k in Equation (6.1) can be as small as 3 in both Frequency Regions 1 and 2. The frequency grid points are allocated evenly in Frequency Regions 1 and 2.

To solve the problem that the frequency positions of local extrema may move during two consecutive iterations of SQP algorithm, the most direct way is to insert several extra frequency grid points between two consecutive extrema. At the same time, all the extrema of the current iteration are also included in the set of updated frequency grid points for next iteration. The purpose of inserting

these extra frequency grid points is to guarantee that the ripples at most local extrema can be reduced to some extent. In fact, these inserted extra frequency grid points and the regular updating of the sparse frequency grid turn out to be able to realize such a purpose. At the same time, the computing speed is improved because the frequency grid point scheme is sparse which reduces the computing amount.

As analyzed in Section 6.2, the number of extrema for a FRM filter in Frequency Regions 1 and 2 is determined by different terms. It is reasonable to insert different numbers of frequency grid points between two consecutive extrema in the two regions. We might insert L_1 frequency grid points between two consecutive extrema in Frequency Region 1, and L_2 frequency grid points in Frequency Region 2. Due to the fact that the ripple in Frequency Region 1 is determined by three ripple terms, L_1 should have a larger value than L_2. According to our experience, L_1 can be in the range from 4 to 8, and L_2 can be in the range from 1 to 4 in Stage 1. In Stage 2, L_1 and L_2 should be 3 to 7 times the values in Stage 1. In both Stages 1 and 2, the frequency grid points are evenly distributed between two consecutive extrema.

To update the frequency grid points, actual extremal points are searched after deciding whether $\boldsymbol{h}$ has converged in both Stages 1 and 2. The search is done on a fixed set of frequency grid points. The fixed frequency grid point scheme is determined according to (6.1). N is the same as described in Section 6.2. In [62], Yang and Lim suggested $k = 40$ to search for the extremal points. When the

effects of nonconstant $G_a(\omega)$ are taken into account, this k value is too small to detect the actual extremal points. According to our experience, k should be in the range of $80-160$ in Frequency Region 1 depending on the value of interpolation factor M. If needed, k can have a larger value for a large M. In Frequency Region 2, $60M-100M$ will be enough to detect all the actual extremal points. Generally speaking, k increases with M.

6.5 Convergence Criteria for Dynamic Grid Points Allocation Scheme

Convergence criteria are always important to any optimization technique, because they determine the termination of an optimization procedure. Generally speaking, an optimization procedure is terminated if one of following requirements is satisfied.

1. The change in the objective function value is less than the predefined difference tolerance ε_1.

2. The change in design vector $\boldsymbol{h}$ is less than a predefined difference tolerance value ε_2.

3. The predefined maximum iteration number is reached.

The predefined maximum iteration number is always a large number. If any among the first two conditions above are satisfied, $\boldsymbol{h}$ is said to have converged.

However, it is unnecessary and inefficient to wait until $\boldsymbol{h}$ has converged to terminate the SQP algorithm in Stage 1. Since the frequency grid is very sparse in Stage 1, the actual extrema are not likely to be detected by the sparse frequency grid. When the design vector $\boldsymbol{h}$ converges in Stage 1, the given specifications are satisfied only at these sparse grid points. Actual ripples in between these grid points may still be larger than the given specifications. So the convergence of $\boldsymbol{h}$ makes no sense.

To improve efficiency, the SQP algorithm is terminated after just several iterations. The maximum iteration number of the SQP algorithm in Stage 1 is chosen to be a small number, such as five. When this maximum iteration number is reached, ripples at most local extrema can be reduced to some extent. It is possible that the maximum deviation may increase, especially during iterations right after the initialization. But the maximum deviation will decrease after the frequency grid points are updated and new SQP iterations are carried out.

It is not likely for $\boldsymbol{h}$ to converge within just several iterations in Stage 1. So the concept of "near convergence" is introduced to provide an exit for Stage 1. The design vector $\boldsymbol{h}$ is considered to be near converged when the following two conditions are satisfied.

1. The change of objective function value between two consecutive frequency grid point updates is less than $n \cdot \varepsilon_1$, where ε_1 is the predefined error tolerance, and n is greater than 1.

2. The maximum deviation detected by the sparse frequency grid points is less than the given specifications.

A suitable value of n can effectively reduce the number of iterations in Stage 2. According to our experience, n should be in the range from 1.5 to 3. The second condition is to guarantee that the given set of specifications are met everywhere in the frequency domain. If the second condition can't be satisfied, that means one or some of the subfilter lengths should be increased. If the second condition is satisfied and the deviation detected by the sparse frequency grid points is much less than the given specification, one or some of the subfilters can be reduced.

The convergence criteria in Stage 2 are the same as usual, which were given at the beginning of this section.

6.6 Design Example

In this section, a design example is given to demonstrate the effectiveness of our new algorithm. The normalized passband and stopband edges of the filter are at 0.4π and 0.406π, respectively. The passband and stopband ripples are 0.01 and 0.001, respectively. The lengths of subfilters are $N_a = 81$, $N_{Ma} = 34$ and

$N_{Mc} = 38$. M is chosen to be 11. Tables 6.1 and 6.2 show the comparison between the fixed frequency grid point allocation scheme and dynamic frequency grid point allocation scheme. It should be pointed out that the grid number and the memory usage of the dynamic frequency grid point allocation scheme are averaged values. Both fixed and dynamic frequency grid point allocation schemes are programmed in Matlab 6.5, and are run on a Pentium IV PC of 2.5GHz. It can be seen from Tables 6.1 and 6.2 that the new dynamic frequency grid point scheme achieves significant savings in the memory usage and CPU time, while result in almost the same design result.

Allocation Scheme	Frequency Grid Point Number	Memory Usage (MB)	CPU Time (Hour)
Fixed Frequency Grid Point Allocation Scheme	4715	21.1	10
Dynamic Frequency Grid Point Allocation Scheme	1151	4.0	0.6
Saving (%)	75.6	81.0	94.0

Table 6.1: Comparison of design costs of fixed and dynamic frequency grid point allocation schemes

Allocation Scheme	Max Passband Deviation	Max Stopband Deviation
Fixed Frequency Grid Point Allocation Scheme	9.988×10^{-3}	9.992×10^{-4}
Dynamic Frequency Grid Point Allocation Scheme	9.994×10^{-3}	9.973×10^{-4}

Table 6.2: Comparison of design result of fixed and dynamic frequency grid point allocation schemes

6.7 Conclusion

In this chapter, ripple analysis is first done for FRM filters designed by nonlinear optimization techniques. A new dynamic frequency grid point allocation scheme is proposed for the efficient design of FRM filters utilizing nonlinear optimization techniques. Combining the proposed scheme with the SQP optimization technique, significant reductions in both memory usage and computing time have been achieved, compared with traditional fixed frequency grid point scheme. The design example verifies the effectiveness of the proposed scheme.

Chapter 7

Conclusion

In this thesis, FIR filter structures based on the parallel prefilter and the iterative design method are first reviewed. The iterative design method fixes the length of the even-length filter $H_e(z)$ to be 2, and only a suboptimal solution is obtained. To further improve the efficiency of the parallel prefilter and its equalizer, the weighted least square (WLS) technique is utilized. Design examples show the effectiveness of the WLS approach.

Given the fact that nonlinear optimization techniques can effectively reduce the filter length of a parallel prefilter and its equalizer, new equations are needed to estimate the filter lengths of a parallel prefilter and its equalizer, which are designed by a nonlinear optimization technique. To save computing time, the filter design task is formulated as a goal attainment problem. This goal attainment problem can be solved by a Matlab function, *fgoalattain*. By utilizing the

function *fgoalattain*, thousands of basic parallel filters are designed, and equations and a table are developed to estimate subfilter lengths in a basic parallel filter. Accuracy analysis shows that the accuracy of the developed equations and look-up table is satisfactory.

Frequency-response masking (FRM) is another efficient FIR filter structure. FRM is suitable for the design of a sharp filter with arbitrary passband bandwidth. In the past few years, nonlinear optimization techniques have been utilized to design FRM filters. These techniques extend the transition bandwidth of two masking filters, and reduce the filter lengths of two masking filters. With the introduction of nonlinear optimization techniques to FRM, the existing estimations of the optimum interpolation factor are no longer accurate. To develop a new equation to estimate the optimum interpolation factor of a FRM filter designed by a nonlinear optimization technique, new equations are first developed to estimate the length of the prototype filter and the total length of two masking filters. Based on the new developed equations, a new equation is derived for the optimum interpolation factor.

Until now, all the reported designs utilize an odd-length FIR filter as the prototype filter in a FRM filter. In this thesis, problems that arise if an even-length FIR filter is utilized as the prototype filter in a FRM filter are first analyzed. In fact, nonlinear optimization techniques can overcome the difficulties faced by even-length prototype filters. A new nonlinear optimization design method is proposed to design a FRM filter. The new design method is based on the se-

quential quadratic programming (SQP) approach. By this new method, a FRM filter with an even-length prototype filter can have performance comparable with a FRM filter with an odd-length prototype filter. Two more filter structures are proposed, which are suitable for the synthesis of FRM filters with even-length prototype filters. Design examples show that these two filter structures lead to further savings in the number of multipliers and adders.

Finally, the distribution of ripple extrema of FRM filters designed by nonlinear optimization techniques is analyzed. A fixed frequency grid point allocation scheme is proposed based on the analysis of the distribution of ripple extrema. To save memory and CPU time, a dynamic frequency grid point allocation scheme is proposed. The new dynamic scheme uses sets of sparse frequency grid points to save memory and CPU time in the first stage. In the second stage, a set of highly dense frequency grid points is utilized to guarantee that the given specifications are satisfied everywhere in the frequency domain.

As far as the filter structures discussed in this thesis, they are all designed in the domain of real number. Further work can be carried out in the domain of integer. This is more attractive because integer solutions are easier to be implemented. At the same, the design methods discussed in this thesis should also be applicable to the design of other digital filter structures. New problems are quite possible to appear when these methods are applied to the design of other filter structure. How to solve these problems is the task for further research.

Bibliography

[1] H. Nyquist,"Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617-644, Apr. 1928.

[2] C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no.1, pp. 10-21, Jan, 1949.

[3] E. Y. Remez,"General communication methods of Chebyshev approximation," *Atomic Energy Translation* 4491, Kiev, U.S.S.R., pp. 1-85, 1957.

[4] G. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, 1963.

[5] H. D. Helms, "Nonrecursive digital filters: design methods for achieving specifications on frequency response," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 336-342, Sep. 1968.

[6] M. D. Canon, C. D. Cullum and E. Polak, *Theory of Optimal Control and Mathematical Programming*. New York: McGraw-Hill, 1970.

[7] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. 19, pp. 189-194, Mar. 1972.

[8] L. R. Rabiner and O. Herrmann,"On the design of optimum FIR low-pass filters with even impulse response duration," *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 329-336, Aug. 1973.

[9] O. Herrmann, L. R. Rabiner and D. S. K. Chan, "Practical design rules for optimum finite impulse response low-pass digital filters," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769-799, July/Aug. 1973.

[10] L. R. Rabiner, "Approximate design relationships for low-pass FIR digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 456-460, Oct. 1973.

[11] J. H. McClellan, T. W. Parks and L .R. Rabiner, "A computer program for designing optimum FIR linear phase filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 506-526, Dec. 1973.

[12] J. F. Kaiser, "Nonrecursive digital filter design using I_0-sinh window function," *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 20-23, Apr. 1974.

[13] F. W. Gembicki, "Vector optimization for control with performance and parameter sensitivity indices," Ph.D. Dissertation, Case Western Reserve University, Cleveland, Ohio, 1974.

[14] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, p. 631, Englewood Cliffs, NJ: Prentice Hall, 1975.

[15] J. F. Kaiser and R. W. Hamming, "Sharpening the response of a symmetric non-recursive filter by multiple use of the same filter," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 415-422, Oct. 1977.

[16] M. J. D. Powell, "A fast algorithm for nonlineary constrained optimization calculations," *Numerical Analysis*, G. A. Watson ed., Lecture Notes in Mathematics, Springer Verlag, vol. 630, pp.144-157, 1978.

[17] R. K. Brayton, S. W. Director, G. D. Hachtel and L. M. Vidigal, "A new algorithm for statistical circuit design based on quasi-newton methods and function splitting," *IEEE Trans. Circuit and Syst.*, vol. CAS-26, no. 9, pp. 784-794, Sep. 1979.

[18] J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[19] M. R. Bateman and B. Liu, "An approach to programmable CTD filters using coefficients 0, +1, and -1," *IEEE Trans. Circuits and Syst.*, vol. CAS-27, pp.451-456, Jun. 1980.

[20] R. Fletcher, "Practical methods of optimization," *Vol. 1, Unconstrained Optimization*, and *Vol. 2 Constrained Optimization*, John Wiley and Sons., 1980.

[21] P. E. Gill, W. Murray and M.H. Wright, *Practical optimization*, Academic Press, New York, pp. 155-204, 1981.

[22] J. W. Adams and A. N. Willson, "FIR digital filters with reduced computational complextiy: a novel design technique," *Proc. of Int. Symp. Circuits and Syst*, pp. 1067-1070, May 1983.

[23] J. W. Adams and A. N. Willson, "A new approach to FIR digital filters with fewer multipliers and reduced sensitivity," *IEEE Trans. Circuits and Syst.* vol. 30, no. 5, pp. 277-283, May 1983.

[24] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-31, pp. 583-591, Jun. 1983.

[25] Y. C. Lim, " Efficient special purpose linear programming for FIR filter design," *IEEE Trans. Acoustics, Speech, and Siganl Processing*, vol. ASSP-31, pp. 963-968, Aug. 1983.

[26] G. V. Reklaitis, A. Ravindran and K. M. Ragsdell, *Engineering Optimization: Methods and Applications*, A Wiley Interscience Publication, pp. 439-462, 1983.

[27] J. W. Adams, and A. N. Willson, "Some Efficient Digital Prefilter Stucture," *IEEE Trans. Circuits and Syst.* vol. CAS-31, no. 3 pp. 260-266, Mar. 1984.

[28] Y. Neuvo, Dong Cheng-Yu and S. K. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32, pp. 563-570, Jun. 1984.

[29] D. G. Luenberger, *Linear and Nonliear Programming*, 2nd Edition, Reading, MA: Addison-Wesley, pp. 30-76, 1984.

[30] G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, 1984.

[31] P. P. Vaidyanathan and G. Beitman, "On prefilters for digital filter design," *IEEE Trans. Circuits and Syst.*, vol. CAS-32, no. 5, pp. 494-499, May 1985.

[32] Y. C. Lim,"Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits and Syst.*, vol. 33, pp. 357-364, Apr. 1986.

[33] H. Babic, G. Rajan and S. K. Mitra, "Multiplierless FIR filter structure based on running sums and cyclotomic polynomials," *EURASIP*, pp. 159-162, 1986.

[34] T. Saramäki, "Design of FIR filters as a tapped cascaded interconnection of identical subfilters," *IEEE Trans. Circuits and Syst.*, vol. CAS-34, pp.1011-1029, Sep. 1987.

[35] H. Kikuchi, Y. Abe, H. Watanabe and T. Yanagisawa, "Efficient prefiltering for FIR digital filters," *Proc. Trans. Electron. Inform. Commun. Eng.* (*IECIE*), vol. E-70 , no. 10, pp. 918-927, Oct. 1987.

[36] T. Saramaäki, Y. Neuvo and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits and Syst.*, vol. 35, pp.70-88, Jan. 1988.

[37] H. Kikuchi, H. watanabe and T. Yanagisawa, "Interpolated FIR filters using cyclotomic polynomials," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3, pp. 2009-2012, Jun. 1988.

[38] T. Saramäki and A. T. Fam, "Subfilter approach for designing efficient FIR filters," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3, pp. 2903-2915, Jun. 1988.

[39] R. H. Yang, Bede Liu and Y. C. Lim, "A new structure of sharp transition FIR filters using frequency-response masking," *IEEE Trans. Circuits and Syst.*, vol. 35, pp. 955-966, Aug. 1988.

[40] H. Kikuchi, "Digital filter design with transfer function approximation based on circuit structure constraints," Ph.D. dissertation, Tokyo Institute of Technology, 1988.

[41] D. Pang, L. A. Ferrari and P. V. Sankar, "A unified approach to general IFIR filter design using B-spline functions," *Proc. Twenty-Third Asilomar Conf. Signals, Syst. and Computers*, vol. 1, pp. 228-232, Oct.-Nov. 1989.

[42] J. C. E. Cabezas and P. S. R. Diniz, "FIR filters using interpolated prefilters and equalizers," *IEEE Trans. Circuits and Syst.*, vol. 37, no. 1, pp. 17-23, Jan. 1990.

[43] Y. C. Lim, "Design of discrete-coefficient-value linear phase FIR filter s with optimum normalized peak ripple magnitude," *IEEE Trans. Circuits and Syst.*, vol.37, pp.1480-1486, Dec. 1990.

[44] P. S. R. Diniz and J. C. E. Cabezas, "Design of FIR equaliser by sharpeing identical subfilters," *IEE Proc. Circuits, Devices and Syst.*, vol. 138 , no. 3, pp. 413 - 417, Jun. 1991.

[45] D. Pang, L. A. Ferrari and P. V. Sankar, "A unified approach to IFIR filter design using B-spline functions," *IEEE Trans. Signal Processing*, vol. 39, pp. 2115-2118, Sep. 1991.

[46] R. H. Yang and Y. C. Lim, "Grid density for the design of one- and two-dimension FIR filters," *Inst. Elec. Eng. Electro. Lett.*, vol. 27, no. 22, pp.2053-2055, Oct. 1991.

[47] Y. C. Lim, J. H. Lee, C. K. Chen and R. H. Yang, "A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design," *IEEE Trans. Signal Processing*, vol. 40, pp.551-558, Mar. 1992.

[48] Y. C. Lim and Y. Lian, "The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique," *IEEE Trans. Circuits and Syst. II*, vol. 40, pp. 88-95, Feb. 1993.

[49] J. K. Hinderling, T. Tueth, K. Easton, D. Eagleson, D. Kindred, R. Kerr and J. Levin, "CDMA Mobile Station Modem ASIC," *IEEE J. of Solid-State Circuits*, vol.28, no. 3, pp.253-260, Mar. 1993.

[50] Y. Lian and Y. C. Lim, "New prefilter structure for designing FIR filters," *IEE Electron. Letters*, vol. 29, pp. 1034-1036, May 1993.

[51] R. J. Hartnett and G. F. Boundreaux-Bartels, "On the use of cyclotomic polynomial prefilters for efficient FIR filter design," *IEEE Trans. Signal Processing*, vol. 41, no. 5, pp. 1766-1779, May 1993.

[52] C. K. Chen and J. H. Lee, "Design of sharp FIR filters with prescribed group delay," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 1, pp. 92-95, May 1993.

[53] Y. Lian and Y. C. Lim, "Reducing the complexity of FIR filters by using parallel structures," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 1, pp. 100-103, May, 1993.

[54] Y. C. Lim and Y. Lian, "Frequency-response masking approach for digital design: complexity reduction via masking filter factorization," *IEEE Trans. Circuits and Syst. II*, vol. 41, pp. 518-525, Aug. 1994.

[55] Y. Lian and Y. C. Lim, "Reducing the complexity of frequency-response masking filters using half band filters," *Signal Processing*, vol. 42, no. 3, pp. 227-230, Mar. 1995.

[56] Y. Lian, "The optimum design of half-band filter using multi-stage frequency-response masking technique," *Signal Processing*, vol. 44, no. 7, pp. 369-372, Jul. 1995.

[57] Y. Lian, "The design of computationally efficient finite impulse response digital filters," Ph.D. Thesis, National University of Singapore, 1995.

[58] T. Saramaki, "Design of computionally efficient FIR filters using periodic subfilters as building blocks" *The Circuits and Filters Handbook*, edited by W.-K. Chen, CRC Press, Inc., pp. 2578-2601, 1995.

[59] C. K. Chen and J. H. Lee, "Design of sharp-cutoff FIR digital filters with prescribed constant group delay," *IEEE Trans. Circuits and Syst. II*, vol 43, pp. 1-13, Jan. 1996.

[60] W. J. Oh and Y. H. Lee, "Design of efficient FIR filters with cyclotomic polynomial prefilters using mixed integer linear programming," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp.1287 - 1290, May 1996.

[61] W. J. Oh and Y. H. Lee, "Design of efficient FIR filters with cyclotomic polynomial prefilters using mixed integer linear programming," *IEEE Signal Processing Letters*, vol. 3, pp. 239 - 241, Aug. 1996.

[62] R. H. Yang and Y. C. Lim, "A dynamic frequency grid allocation scheme for the efficient design of equiripple FIR filters," *IEEE Trans. Signal Processing*, vol. 44, no. 9, pp. 2335-2339, Sep. 1996.

[63] G. L. Do and K. Feher, "Efficient filter design for IS-95 CDMA systems," *IEEE Trans. Consumer Electronics*, vol. 42, no. 4, pp.1011-1020, Nov. 1996.

[64] K. Ichige, K. Ueda, M. Iwaki and R. Ishii, "A new estimation formula for minimum filter length of optimum FIR low-pass digital filters," *Proc. of*

Int. Conf. Information, Communications and Signal Processing, vol. 3, pp. 1303-1307, Sep. 1997.

[65] Y. Lian and Y. C. Lim, "Structure for narrow and moderate transition band FIR filter design," *IEE Electron. Letters*, vol. 34. pp. 49-51, Jan. 1998.

[66] H. K. Garg, *Digital Signal Processing Algorithms: Number Theory, Convolution, Fast Fourier Transforms, and Applications*, Boca Raton, Fla. : CRC Press, pp. 225-400, 1998.

[67] K. Ichige, M. Iwaki and R. Ishii, "A new estimation formula for minimum filter length of optimum FIR digital filters," *Proc. of Fourth Int. Conf. Signal Processing*, vol. 1, pp. 89-92, Oct. 1998.

[68] T. Coleman, M. A. Branch, A. Grace, *Optimization toolbox for use with Matlab, user's guide*, Jan. 1999.

[69] H. J. Oh, S. Kim, G. Choi and Y. H. Lee, "On the use of interpolated second-order polynomials for efficient filter design in programmable down-conversion," *IEEE Journal On Selected Areas in Commun.*, vol 17, pp.551-560, Apr. 1999.

[70] T. Saramäki and Y. C. Lim, "Use of the Remez algorithm for designing FIR filters utilizing the frequency-response masking approach," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3 pp.449-455, Jun. 1999.

[71] M. Kim, D. Kim, J. Chung and M. Lim, "1:N interpolation FIR filter design for SSB/BPSK-DS/CDMA," *Proc. Joint Conf. Comm. and Info.*, Korea, May 2000.

[72] H. J. Oh and Y. H. Lee,"Design of discrete coefficient FIR and IIR digital filters with prefilter-equalizer structure using linear programming," *IEEE Trans. Circuits and Syst. II*, vol 47, pp.562-565, Jun. 2000.

[73] K. Ichige, M. Iwaki and R. Ishii, "Accurate estimation of minimum filter length for optimum FIR digital filters," *IEEE Trans. Circuits and Syst. II*, vol. 47, pp. 1008-1016, Oct. 2000.

[74] M. Kim, D. Kim, J. Chung and M. Lim, "Pulse-shaping filter design for SSB/BPSK-DS/CDMA using look-up table," *Proc. IEEE Workshop Signal Processing Syst.*, pp. 407-415, Oct. 2000.

[75] T. Saramäki and H. Johansson, "Optimization of FIR filters using the frequency-response masking approach," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 2, pp. 177-180, May 2001.

[76] Y. Lian and M. H. Poh, "FPGA implementation of IS-95 CDMA baseband filter," *Proc. of 4th Int. Conf. ASIC*, Oct. 2001.

[77] Y. Lian, L. Zhang and C. C. Ko, "An improved frequency response masking approach for designing sharp FIR filters," *Signal Processing*, vol. 81, pp. 2573-2581, Dec. 2001.

[78] P. Venkataraman, *Applied optimization with MATLAB programming*, pp.214 217, 2002.

[79] Y. J. Yu and Y. C. Lim, "FRM based FIR filter design - the WLS approach," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3, pp. 221-224, May 2002.

[80] T. Saramäki and J. Yli-Kaakinen, "Optimization of frequency-response-masking based FIR filters with reduced complexity," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3, pp. 225-228, May 2002.

[81] C. Z. Yang and Y. Lian, "A modified structure for the design of sharp FIR filters using frequency response masking technique," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol. 3, pp. 237-240, May 2002.

[82] D. Li, Y. C. Lim, Y. Lian and J. Song, "A polynomial-time algorithm for designing FIR filters with power-of-two coefficients," *IEEE Trans. Signal Processing*, vol. 50, pp. 1935-1941, Aug. 2002.

[83] T. Saramäki and Y. C. Lim, "Use of the Remez algorithm fordesigning FRM based FIR filters," *Circuit, Systems, Signal Processing*, vol. 22, no. 2, pp. 77-97, Mar. 2003.

[84] L. C. R. de Barcellos, S. L. Netto and P. S. R. Diniz, "Optimization of FRM filters using the WLS-Chebyshev Approach," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 99-113, Mar. 2003.

[85] Y. Lian and C. Z. Yang, "Complexity reduction by decoupling the masking filters from the bandedge shaping filter in the FRM technique," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 115-135, Mar. 2003.

[86] Y. Lian, "Complexity Reduction for Frequency-Response Masking Based FIR Filters via Prefilter-Equalizer Technique," *Circuit, Systems and Signal Processing*, vol. 22, no. 2, pp.137-155, Mar. 2003.

[87] H. Johansson and T. Saramäki, "Two-channel FIR filter banks utilizing the FRM approach," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 157-192, Mar. 2003.

[88] M. B. Furtado Jr., P. S. R. Diniz and S. L. Netto, "Optimized prototype filter based on the FRM approach for cosine-modulated filter banks," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 193-210, Mar. 2003.

[89] Y. C. Lim, Y. J. Yu, H. Q. Zheng and S. W. Foo, "FPGA implementation of digital filters synthesized using FRM technique," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 211-218, Mar. 2003.

[90] O. Gustafsson, H. Johansson and L. Wanhammar, "Single filter frequency masking high-speed recursive digital filters," *Circuit, Systems, and Signal Processing*, vol. 22, no. 2, pp. 219-238, Mar. 2003.

[91] W. S. Lu and T. Hinamoto, "Optimal design of frequency-response-masking filters using semidefinite programming," *IEEE Trans. Circuits and Syst. I*, vol. 50, pp. 557-568, Apr. 2003.

[92] W. S. Lu and T. Hinamoto, "Optimal design of FIR frequency-response-masking filters using second-order cone programming," *Proc. of IEEE Int. Symp. Circuits and Syst.*, vol.3, pp.878-881, May 2003.

[93] T. Saramäki, J. Yli-Kaakinen and H. Johansson, "Optimization of frequency-response masking based FIR filters," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 563-590 Oct. 2003.

[94] W. R. Lee, V. Rehbock and K. L. Teo, "Frequency-response masking based FIR filter design with power-of-two coefficients and suboptimum PWR," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 591-600, Oct. 2003.

[95] O. Gustafsson, H. Johansson and L. Wanhammar, "Single filter frequency-response masking FIR filter," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 601-630, Oct. 2003.

[96] S. L. Netto, L. C. R. de Barcellos and P. S. R. Diniz, "Efficient design of narrowband cosine-modulated filter banks using a two-stage frequency-response masking approach," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 631-642, Oct. 2003.

[97] Y. Lian, "A modified frequency response masking structure for high-speed FPGA implementation of programmable sharp FIR filters," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 643-654, Oct. 2003.

[98] S. W. Foo and W. T. Lee, "Application of fast filter bank for transcription of polyphonic signals," *Circuits, Systems, and Computers*, vol. 12, no. 5, pp. 655-674, Oct. 2003.

[99] W. S. Lu and T. Hinamoto, "Optimal design of IIR frequency-response-masking filters using second-order cone programming," *IEEE Trans. Circuits and Syst. I*, vol 50, pp. 1401-1412, Nov. 2003.

[100] J. H. Yu and Y. Lian, "Frequency-response masking based filters with the even-length bandedge shaping filter," *Proc. of IEEE Int. Symp. Circuits Syst.*, vol. 5, pp. 536-539, May 2004.

[101] W. R. Lee, V. Rehbock and K. L. Teo, "A weighted least-square-based approach to FIR filter design using the frequency-response masking technique," *IEEE Signal Processing Letters*, vol. 11, pp. 593-596, Jul. 2004.

[102] Y. C. Lim and R. Yang, "On the Synthesis of Very Sharp Decimators and Interpolators Using the Frequency-Response Masking Technique," *IEEE Trans. Signal Processing*, vol. 53. no. 4. pp. 1387-1397, Apr. 2005.