INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6” x 9” black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600
RICE UNIVERSITY

New Techniques for Digital Filter Design

by

Ivan W. Selesnick

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

C. Sidney Burrus, Chair
Maxfield & Oshman Professor of Electrical and Computer Engineering

Richard G. Baraniuk
Assistant Professor of Electrical and Computer Engineering

John E. Dennis
Noah Harding Professor of Computational and Applied Mathematics

Houston, Texas

May, 1996
New Techniques for Digital Filter Design

Ivan W. Selesnick

Abstract

Several new techniques for open problems in the frequency domain design of FIR and IIR digital filters, both iterative and analytic, are presented.

This thesis begins by putting forth the notion that explicitly specified transition bands have been introduced in the literature in part as an indirect approach for dealing with discontinuities in the desired frequency response. To overcome this, a rapidly converging algorithm is presented for the design of symmetric FIR filters according to a square error criterion that does not require specified transition bands. It does not exclude from the integral square error a region around the cut-off frequency, and yet, it overcomes Gibbs' phenomenon without resorting to windowing or 'smoothing out' the discontinuity of the ideal lowpass filter.

Also presented are algorithms for symmetric FIR filter design that modify the Parks-McClellan algorithm and a variation due to Vaidyanathan, to give a fairly complete set of design techniques for the design equiripple symmetric FIR filters.

Two types of filters, which are particularly well suited for both the approximation and implementation problems, are frequently overlooked because their design is substantially more difficult than the design of symmetric FIR and classical IIR filters. These two filter types are (1) non-symmetric FIR filters, and (2) IIR filters for which the numerator and denominator degrees need not be equal. For maximally-flat lowpass responses, analytic techniques for these two filter types are presented.

For non-symmetric FIR filter design, in which the magnitude and group delay are regarded separately (a classic problem in the design of both digital and analogue...
filters), it is shown that the delay can be reduced significantly while maintaining a very constant passband group delay, with no degradation in the magnitude response.

For IIR filters for which the numerator and denominator degrees are unequal, techniques for the design of generalized Butterworth and Chebyshev filters are presented.

This thesis also presents a technique to make more practical the rational Remez exchange algorithm. Lastly, a problem examined by Souto is revisited, and the use of Gröbner bases from computational algebraic geometry for this problem is described.
Acknowledgments

It is my greatest pleasure to thank my advisor Professor Sidney Burrus for his enthusiasm, confidence in me, patience, and continuing encouragement. It is his mentorship that has made this experience both enjoyable and rewarding. I also like to thank Professor Richard Baraniuk and Professor John Dennis for serving on my thesis committee with such geniality. I have also enjoyed discussions with Professor Hans Schüßler, who gave very constructive comments and encouragement. I feel privileged to receive such attention from him. Thanks to Professor Richard Baraniuk and Dr. Rob Nowak for bringing Gröbner bases to my attention with which I became so enthralled and upon which Chapters 6 and 10 depended.

I especially like to thank Markus Lang, with whom I had a most rewarding collaboration on several of the ideas in this thesis while he was at Rice. I like to thank my officemates while at Rice, especially Jan Odegard and Haitao Guo, for their help with the computer operating systems and for their friendship. Ramesh Gopinath’s example set a challengingly high standard for graduate research.

Thanks to Anthony Postert for bringing the Journal of Kitchen Supplies to my attention. Other friends have long given me appreciated company, Suzie Hardt, Chris Deaves; and more recently Haifa Nouaime. Thanks also to Marielba Rojas, the members of META-4, the Gouldie family, and Suzana Božić. My greatest thanks are to my parents for their unwavering support and love over the years, and to the rest of my family, here in the US and in England.

I must also acknowledge DARPA for providing me with three years of full financial support under an NDSEG fellowship. BNR and NSF also provided financial support which I gratefully acknowledge.
Contents

Abstract ii
Acknowledgments iv
List of Illustrations xi
List of Tables xv

1 Introduction 1
  1.1 Error Criteria and Filter Type 3
  1.2 Chapter Abstracts 5

2 Constrained Least Square Design of FIR Filters Without Specified Transition Bands 8
  2.1 Introduction 8
  2.2 Preliminaries 10
    2.2.1 Adams’ Error Criterion 13
    2.2.2 Approximation by Operator Norms 15
  2.3 A New Criterion for Lowpass Filter Design 17
  2.4 A New Algorithm for Lowpass Filter Design 19
    2.4.1 The Equality Constrained Minimization Problem 20
    2.4.2 The Exchange Iterations 24
  2.5 Interpretations and Extensions 31
    2.5.1 Chebyshev Solutions 32
    2.5.2 Trade-off Curves 34
    2.5.3 Specified Band Edges 35
3 Constrained Least Square Design of Multiband FIR Filters

3.1 Introduction ........................................... 47
3.2 Example ........................................... 47
3.3 New Algorithm ........................................... 50
   3.3.1 Equality Constraints ........................................... 51
   3.3.2 The Modified Exchange Iterations ........................................... 51
3.4 Example ........................................... 55
3.5 Conclusion ........................................... 57
3.6 A Matlab Program ........................................... 57

4 Complementing the Parks-McClellan Algorithm ........................................... 61
4.1 Introduction ........................................... 61
4.2 Equiripple Filter Design ........................................... 63
   4.2.1 The PM Program with an Affine Relation between $\delta_p$ and $\delta_s$ ........................................... 64
   4.2.2 A New Equiripple Lowpass Filter Design Algorithm: Specified $\delta_p$, $\delta_s$ and $\omega_c$ ........................................... 69
   4.2.3 Bandpass Filter Design ........................................... 73
4.3 Conclusion ........................................... 77

5 Flat Monotonic Passbands and Equiripple Stopbands ........................................... 79
5.1 Introduction ........................................... 79
5.2 The Design Algorithm ........................................... 81
  5.2.1 Specifying $\omega_s$ ........................................... 82
  5.2.2 Specifying $\delta_s$ ........................................... 84
  5.2.3 Passband Monotonicity ..................................... 85
5.3 Bandpass Filter Design ......................................... 86
5.4 Lowpass Differentiators ......................................... 88
5.5 Discussion and Conclusion ...................................... 90

6 Nonlinear-Phase Maximally-Flat Lowpass FIR Filters 93
  6.1 Introduction .................................................. 93
  6.2 Notation ...................................................... 95
  6.3 The Basic Problem Formulation ................................ 97
  6.4 Example ....................................................... 98
  6.5 Obtaining Solutions .......................................... 100
    6.5.1 Using Gröbner Bases ................................... 106
    6.5.2 The Number of Solutions ................................. 107
    6.5.3 Region I Filters ......................................... 114
  6.6 Calculating Filters in Region I ............................. 114
    6.6.1 Case 1 .................................................. 116
    6.6.2 Case 2 .................................................. 117
    6.6.3 Case 3 .................................................. 118
    6.6.4 Root Selection .......................................... 121
    6.6.5 Example ................................................ 121
  6.7 Specification of the DC Group Delay ....................... 125
    6.7.1 Case 1 ................................................ 126
    6.7.2 Case 2 ................................................ 126
    6.7.3 Selecting $L$ and $M$ .................................. 127
  6.8 Specification of the Half-Magnitude Frequency ............ 130
6.8.1 Case 1 ............................................. 131
6.8.2 Case 2 ............................................. 131
6.8.3 Example ............................................. 131

6.9 Specification of Both the DC Group Delay and Half-Magnitude

Frequency ............................................. 132
6.9.1 Case 1 ............................................. 135
6.9.2 Case 2 ............................................. 135
6.9.3 Case 3 ............................................. 135
6.9.4 Root Selection ............................................. 136
6.9.5 Selection of $K, L, M$ ............................................. 137
6.9.6 Example ............................................. 139
6.9.7 Approximate Specification Sectors ............................................. 139

6.10 A Longer Example ............................................. 140
6.11 Remark ............................................. 144
6.12 Conclusion ............................................. 144
6.13 All Real Solutions to the First the Example ............................................. 145

7 Generalized Digital Butterworth Filter Design 148

7.1 Introduction ............................................. 148
7.2 Notation ............................................. 149
7.3 Examples ............................................. 150
7.4 Discussion ............................................. 152
7.5 Derivations ............................................. 158
    7.5.1 Classical Digital Butterworth Filter ............................................. 158
    7.5.2 First Generalization ............................................. 159
    7.5.3 Second Generalization ............................................. 162
7.6 Further Remarks ............................................. 168
    7.6.1 Behavior for Odd $N$ ............................................. 168
7.6.2  A Note on Implementation .......................... 169
7.6.3  Butterworth Filters Having More Poles Than Zeros ........ 169
7.6.4  FIR Butterworth Filters ............................ 170
7.7  Conclusion ........................................ 170
7.8  Proofs ............................................. 172
   7.8.1  The First Generalization ......................... 173
   7.8.2  The Second Generalization ....................... 176
7.9  Connection to a Series of Gauss ....................... 178
7.10  Matlab Programs .................................. 179

8  Generalized Chebyshev II Filter Design 187
   8.1  Introduction ..................................... 187
   8.2  Zero-Shifting Algorithm ............................ 188
      8.2.1  Step-Size Reduction ............................ 196
      8.2.2  Remarks ...................................... 197
   8.3  Exchange Algorithm ................................. 198
      8.3.1  Interpolation Step ............................ 200
      8.3.2  Exchange Step ................................ 202
      8.3.3  Initialization ................................ 203
      8.3.4  Examples ..................................... 203
      8.3.5  Generalized Chebyshev I Filters .............. 205
   8.4  Conclusion ....................................... 207

9  A Modified Rational Remez Algorithm for Recursive
   Digital Filter Design 208
   9.1  Introduction ..................................... 208
   9.2  The Rational Remez Exchange Algorithm ............. 209
   9.3  Overcoming "Faulty" Reference Sets .................. 213
# Illustrations

2.1 The desired amplitude of an ideal lowpass filter. ................... 11
2.2 Flowgraph for constrained least square filter design. .............. 27
2.3 A best constrained $L_2$ filter. $\delta = 0.02$. ....................... 28
2.4 A best $L_2$ filter. ........................................... 29
2.5 An illustration of the convergence behavior. ......................... 30
2.6 A best constrained $L_2$ filter. $\delta = 0.004$. ...................... 33
2.7 Trade-off curve: integral square error — peak error size. .......... 35
2.8 A best constrained $L_2$ filter with a specified stopband edge. .... 38
2.9 Passband and stopband details. .................................. 42

3.1 The desired amplitude of an ideal bandpass filter. .................. 48
3.2 Previous algorithm applied to bandpass filter design. Even iterations. 49
3.3 Previous algorithm applied to bandpass filter design. Odd iterations. 49
3.4 Flowgraph for the exchange algorithm for the constrained least square design of multiband filters. ......................... 54
3.5 Modified algorithm applied to bandpass filter design. ............ 55
3.6 The first 4 (outer) iterations of the modified algorithm for the example. 56

4.1 An equiripple filter, $N = 21$. .................................. 62
4.2 Updating the reference set. ...................................... 66
4.3 Flowchart for “affine” PM exchange algorithm. ..................... 68
4.4 Convergence behaviour with negative $\delta$. ....................... 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 Updating the reference set for modified PM algorithm.</td>
<td>72</td>
</tr>
<tr>
<td>4.6 An optimal Chebyshev bandpass filter.</td>
<td>75</td>
</tr>
<tr>
<td>4.7 A bandpass filter produced by the new algorithm.</td>
<td>76</td>
</tr>
<tr>
<td>5.1 A filter having a flat monotonic passband and an equi-ripple stopband, $\omega_s = 0.6\pi$.</td>
<td>80</td>
</tr>
<tr>
<td>5.2 Filter structure for implementation and design of lowpass filter.</td>
<td>82</td>
</tr>
<tr>
<td>5.3 A filter having a flat monotonic passband and an equiripple stopband, $\delta_s = 0.02$.</td>
<td>85</td>
</tr>
<tr>
<td>5.4 A bandpass filter having a flat monotonic passband and equi-ripple stopbands.</td>
<td>86</td>
</tr>
<tr>
<td>5.5 Filter structure for implementation and design of a lowpass differentiator.</td>
<td>89</td>
</tr>
<tr>
<td>5.6 A differentiator having a flat monotonic passband and an equiripple stopband.</td>
<td>90</td>
</tr>
<tr>
<td>6.1 Flatness parameters $K$, $L$ and $M$.</td>
<td>97</td>
</tr>
<tr>
<td>6.2 A selection of nonlinear-phase maximally-flat filters.</td>
<td>101</td>
</tr>
<tr>
<td>6.3 Continuation of Figure 6.2.</td>
<td>102</td>
</tr>
<tr>
<td>6.4 The monotonic solution not shown in Figure 6.2.</td>
<td>103</td>
</tr>
<tr>
<td>6.5 A greater range of half-magnitude frequencies.</td>
<td>104</td>
</tr>
<tr>
<td>6.6 Locations in the $\omega_o$-$G(0)$ plane of each of the length 13 filters in region I for which $N = K + L + M + 1$.</td>
<td>112</td>
</tr>
<tr>
<td>6.7 Filters having a specified DC group delay.</td>
<td>129</td>
</tr>
<tr>
<td>6.8 The variation of the half-magnitude frequency with the specified DC group delay.</td>
<td>130</td>
</tr>
<tr>
<td>6.9 Filters having specified half-magnitude frequencies.</td>
<td>133</td>
</tr>
<tr>
<td>6.10 Specification sectors in the $\omega_o$-$G(0)$ plane for length 13 filters in region I.</td>
<td>138</td>
</tr>
</tbody>
</table>
6.11 Filters having a specified half-magnitude frequency and DC group delay, \( N = 13 \). ................................................................. 140

6.12 Approximation of the specification sectors shown in Figure 6.10. .... 141

6.13 Specification sectors in \( \omega_a-G(0) \) plane for length 23 filters in region I. 142

6.14 Filters having a specified half-magnitude frequency and DC group delay, \( N = 23 \). ................................................................. 143

6.15 All real solutions: \( K = 6, L = 3, M = 3, N = 13 \) ...................... 145

6.16 All real solutions: \( K = 6, L = 2, M = 4, N = 13 \) ...................... 146

6.17 All real solutions: \( K = 6, L = 1, M = 5, N = 13 \) ...................... 146

6.18 All real solutions: \( K = 6, L = 0, M = 6, N = 13 \) ...................... 147

7.1 A classical digital Butterworth filter. ............................................. 153

7.2 A digital Butterworth filter for which \( L > N \). .............................. 154

7.3 A digital Butterworth filter for which \( L > N \) and \( M > 0 \). .......... 155

7.4 A set of digital Butterworth filters for which \( L + M + N = 20 \). .... 156

7.5 A sequence of digital Butterworth filters. .................................... 165

8.1 A rational function \( F(x) \) for Chebyshev II filter design with an even number stopband zeros ..................................................... 191

8.2 An extra-ripple generalized Chebyshev II filter with an even number of stopband zeros ......................................................... 191

8.3 A rational function \( F(x) \) for Chebyshev II filter design with an odd number stopband zeros ..................................................... 192

8.4 An extra-ripple generalized Chebyshev II filter with an odd number of stopband zeros ......................................................... 192

8.5 Pre-convergence of zero-shifting algorithm .................................... 195

8.6 A set of extra-ripple Chebyshev II filters .................................... 199

8.7 Flowgraph for the exchange algorithm for Chebyshev II filters ....... 201
8.8 A generalized Chebyshev II filter with specified stopband edge \( \omega_s \). 204
8.9 A generalized Chebyshev II filter with a scaled extra ripple .... 205
8.10 A set of Chebyshev II filters having 16 poles and zeros (total) and the same stopband edge and stopband ripples size. ........ 206
8.11 A generalized Chebyshev I filter. ......................... 207

9.1 The filter for example 1 having 2 poles. ................. 218
9.2 The Chebyshev error as a function of the number of zeros for example 1. ........................................ 219
9.3 The filter for example 2 for which \( m = 8 \) and \( n = 3 \). .... 220
9.4 The filter for example 2 for which \( m = 7 \) and \( n = 2 \). .... 221

10.1 \( F_{4,2}(x) \). .............................................. 229
10.2 Filter response corresponding to \( F_{4,2}(x) \). .......... 229
10.3 \( F_{4,3}(x) \) .............................................. 230
10.4 Filter response corresponding to \( F_{4,3}(x) \). .......... 230
10.5 \( F_{3,2}(x) \) .............................................. 231
10.6 \( F_{2,2}(x) \) .............................................. 231

11.1 A constrained least square filter, representative of Chapter 2. ... 233
11.2 An equi-ripple filter, representative of Chapter 4. .......... 234
11.3 An FIR filter having a flat monotonic passband and an equi-ripple stopband, representative of Chapter 5. ................... 235
11.4 A maximally-flat nearly symmetric filter with reduced delay, representative of Chapter 6. ............................ 236
11.5 A generalized Butterworth filter, representative of Chapter 7. .. 237
11.6 A generalized Chebyshev II filter, representative of Chapter 8. ... 238
Tables

4.1 Exchange algorithms for equiripple filters. .......................... 78

6.1 Notation. ........................................................................ 96

6.2 The DC group delays and half-magnitude frequencies for the filters
shown in Figure 6.2. ......................................................... 102

6.3 The DC group delays and half-magnitude frequencies for the filters
shown in Figure 6.5. ......................................................... 104

6.4 A Gröbner basis. ............................................................. 107

6.5 Degree of reference polynomial $R_{K,L,M}(A)$. .................... 108

6.6 Number of real roots of reference polynomial $R_{K,L,M}(A)$. ..... 109

6.7 Number of real monotonic solutions, not counting time-reversals. ... 110

6.8 Regions I and II. ............................................................ 111

6.9 The DC group delay and the half-magnitude frequency $\omega_0$ for each
length 13 filter in region I for which $N = K + L + M + 1$. ......... 113

6.10 The DC group delays and half-magnitude frequencies for the filters
shown in Figure 6.7. ......................................................... 128

6.11 The DC group delays and half-magnitude frequencies for the filters
shown in Figure 6.9. ......................................................... 133

6.12 The flatness parameters for the filters shown in Figure 6.11. ..... 139

6.13 The flatness parameters for the filters shown in Figure 6.14. ..... 143

7.1 Notation. ........................................................................ 149
7.2 A specification table for generalized Butterworth filter design. .... 151
7.3 Transition region sharpness of the filters shown in Figure 7.4. ...... 157
7.4 Permissible ranges for c for the first generalization. ............. 161
7.5 Real roots of $J_N[(1 - x)^L] + cx^N - 4(1 - x)^L$. ............... 162
7.6 Permissible ranges for c for the second generalization. .......... 166
7.7 Real roots of a polynomial associated with the second generalization. 167
7.8 A summary of some formulas for digital Butterworth filter design. .. 171
7.9 Auxiliary polynomials. ........................................... 171

8.1 A specification table for generalized Chebyshev II filters ......... 198

11.1 Maximally-flat lowpass filter design: categorization of techniques. . 241
Chapter 1

Introduction

This dissertation presents several new techniques, both iterative and analytic, for the frequency domain design of digital filters. These techniques solve problems that have long remained open in the filter design literature. Both finite impulse response (FIR) and infinite impulse response (IIR) digital filters are considered. For FIR filters, techniques for the design of both linear-phase and nonlinear-phase filters are presented.

Throughout this dissertation, the main problem we consider is the approximation of the ideal lowpass frequency response (see Figure 2.1) by a realizable digital filter. Although this is the most basic problem in digital filter design, there are several problems that have remained open in the literature.

To describe the problems addressed in this dissertation, it is convenient to list two relevant questions:

1. Given a measure of approximation error, what types of filters are most suitable? (What filter structures and characteristics are most likely to yield good approximations with low complexity of implementation?)

2. Given a type of filter, for what measures of approximation error are convenient design techniques readily available?

In the process of designing a filter, it is expected that the type of measure of approximation error is suggested first, and that the type of filter which is most suitable

*At a few points in this dissertation we consider other ideal responses. Specifically, multiband bandpass filter responses in Chapters 3, 4 and 5, and differentiators in Chapter 5.*
follows from that. The remaining issue is the existence of techniques for the design of the specified type of filter according to the specified type of error measure. In practice, however, the types of filters used may depend as much on (2.) as on (1.). This dissertation provides new design techniques to help alleviate this.

It should be noted that the two most commonly used filter types (and approximation domains) are:

1. Symmetric FIR filters (with magnitude approximation).

2. IIR filters for which the numerator and denominator degrees are equal (with magnitude approximation).

For these two filter types, many expositions of standard design techniques can be found in textbooks and handbooks, and many computer programs implementing these techniques are readily available. However, as suggested in Chapter 2, for symmetric FIR filter design, the error criteria used in much of the literature often requires possibly unrealistic assumptions before optimality can be asserted. In Chapter 2, a new error criteria is formulated, and an efficient, robust algorithm is given for the design of symmetric FIR filters according to this criterion.

In addition, the two filter types enumerated above are very restrictive when the following two filter types, for which this thesis gives analytic design techniques, are considered in addition to the above two:

1. Non-symmetric FIR filters (with magnitude and group delay approximation).

2. IIR filters for which the numerator and denominator degrees need not be equal (with magnitude approximation).

These two types of filters generally more realistically reflect and are more accurately matched to the filter design problem. However, for the design of these two filter types, the approximation problem becomes substantially more difficult, and consequently no
iterative techniques have become standard or widely used — and except for special cases, no analytic solutions have been previously reported.

Although many papers address the design of such filters by numerical methods, few of the techniques described therein have become standard — and programs implementing those techniques have not become widely available. Chapters 6 and 7 give convenient analytic design techniques for these two filter types.

In this dissertation, we derive new design techniques for each of these four filter types. Iterative techniques that converge reliably and rapidly are given for the design of symmetric FIR filters. Analytic design techniques for non-symmetric FIR filters and IIR filters are given and can be implemented easily with a computer program. An iterative technique is also given for IIR filter design.

1.1 Error Criteria and Filter Type

Let us suppose first that exactly linear phase is desired, and that the delay is unimportant. Then the most appropriate type of filter is a symmetric FIR filter, as it achieves exactly linear phase. Because the approximation problem for such filters is more straight-forward than it is for other types of filters, convenient design techniques exist for several approximation criteria. Moreover, for such filters, one can develop convenient techniques for design criteria that have been more carefully formulated. Chapters 2 through 5 take advantage of this possibility and describe algorithms for the design of symmetric FIR filters according to several carefully formulated criteria.

Chapter 2 examines the use of specified transition bands and proposes and motivates a new method for approximating the discontinuity of the ideal lowpass response. Chapters 2 and 3 describe Remez-like exchange algorithms for the suggested criterion. Chapters 4 and 5 describe exchange algorithms that complement the Parks-McClellan (PM) algorithm, the most widely used technique for symmetric FIR filter design. The algorithm of Chapter 4, together with the PM algorithm and other previously reported algorithms, provide a flexible way in which to specify the various parameters
that characterize an equiripple filter.

Second, suppose that approximately linear phase is acceptable and that it is desired that the delay be low. Then it is beneficial to give up the symmetry requirement and to use the extra degrees of freedom so obtained to reduce the delay and improve the response magnitude. When the magnitude and group delay are regarded separately (a classic problem in the design of filters other than digital), the resulting approximation is substantially more difficult than it is in the symmetric case. A tractable analytic solution is most conceivable for maximally flat responses. Even so, for maximally-flat responses, analytic solutions have been previously given only for a special case. In Chapter 6, an analytic technique is given for a larger class of filters than previously. We find that the delay can be reduced significantly while maintaining the response magnitude and a relatively constant passband group delay.

Next, suppose that the phase is unimportant and that a good magnitude response is desired. Then a filter obtained by spectrally factoring a nonnegative real frequency response is most appropriate. For FIR filters, this procedure is straightforward – most design techniques for symmetric FIR filters (including the ones given in Chapters 2 through 5) can be adapted or used directly. However, compared to an FIR filter, an IIR filter having a low order denominator degree often provides a much better trade-off between approximation quality and implementation complexity.

For IIR filters having unequal numerator and denominator degrees, the corresponding approximation problem is again, substantially more difficult. However, for a maximally-flat response, an analytic design technique is possible and is presented in Chapter 7. For generalized Chebyshev IIR filters, an iterative numerical algorithm is presented in Chapter 8. A Remez-like exchange algorithm for the magnitude design of equiripple IIR filters is discussed in Chapter 9.
1.2 Chapter Abstracts

Chapter 2 puts forth the notion that explicitly specified transition bands have been introduced in the filter design literature in part as an indirect approach for dealing with discontinuities in the desired frequency response. We suggest that the use of explicitly specified transition bands is sometimes inappropriate because, to satisfy a meaningful optimality criterion; their use implicitly assumes a possibly unrealistic assumption on the class of input signals.

This chapter also presents an algorithm for the design of peak constrained lowpass FIR filters according to an integral square error criterion that does not require the use of specified transition bands. This rapidly converging, robust, simple multiple exchange algorithm uses Lagrange multipliers and the Kuhn-Tucker conditions on each iteration. The algorithm will design linear and minimum phase FIR filters and gives the best $L_2$ filter and a continuum of Chebyshev filters as special cases.

The algorithm is distinct from many other filter design methods because it does not exclude from the integral square error a region around the cut-off frequency, and yet, it overcomes Gibbs' phenomenon without resorting to windowing or "smoothing out" the discontinuity of the ideal lowpass filter.

Chapter 3 describes a modification to the algorithm of Chapter 2 that makes it converge for multiband filter design. The modified algorithm remains simple and converges rapidly in most cases.

Chapter 4 describes an exchange algorithm for the design of linear-phase FIR equiripple filters where the Chebyshev error in each band is specified. The algorithm is a hybrid of the algorithm of Hofstetter, Oppenheim and Siegel and the Parks-McClellan algorithm. This chapter also describes a modification of the Parks-McClellan algorithm where either the passband or the stopband ripple size is specified and the other is minimized.

Chapter 5 describes a modification of a technique proposed by Vaidyanathan for the
design of filters having flat passbands and equiripple stopbands. The modification ensures that the passband is monotonic and does so without the use of concavity constraints. Another modification described in this chapter adapts the method of Vaidyanathan to the design of lowpass differentiators having a specified degree of tangency at $\omega = 0$.

**Chapter 6** describes the design of nonlinear-phase maximally-flat lowpass FIR filters. By subjecting the response magnitude and the group delay (individually) to differing numbers of flatness constraints, a new family of filters is obtained. It is also shown how to achieve a specified half-magnitude frequency and DC group delay. It is found that by using these filters, the delay can be reduced significantly while maintaining relatively constant group delay, without significantly altering the response magnitude.

**Chapter 7** presents a formula-based method for the design of IIR filters having more zeros than (nontrivial) poles. The filters are designed so that their square magnitude frequency responses are maximally-flat at $\omega = 0$ and at $\omega = \pi$ and are thereby generalizations of classical digital Butterworth filters. A main result of this chapter is that, for a specified half-magnitude frequency and a specified number of zeros, there is only one valid way in which to split the zeros between $z = -1$ and the passband. Moreover, for a specified number of zeros and a specified half-magnitude frequency, the method directly determines the appropriate way to split the zeros between $z = -1$ and the passband. IIR filters having more zeros than poles are of interest, because often, to obtain a good trade-off between performance and the expense of implementation, just a few poles are best.

**Chapter 8** describes a program for the design of IIR digital filters having flat monotonic passbands and equiripple stopbands. The technique allows the user to specify the stopband ripple size and the stopband edge. The resulting filters are analogous to classical Chebyshev type II digital IIR filters, but the filters obtained with the algorithm described in this chapter have numerator degrees that are greater than the denominator degree.
Chapter 9 describes a type of Remez exchange algorithm for the design of stable recursive digital filters for which the Chebyshev norm of $H(\omega) - D(\omega)$ is minimized, where $H(\omega)$ and $D(\omega)$ are the realized and desired magnitude squared frequency responses. The number of poles and zeros (away from the origin) can be chosen arbitrarily and the zeros do not have to lie on the unit circle. Three difficulties in the use of the Remez algorithm and ways to overcome them are discussed. Aspects of near degeneracy are also discussed.

Chapter 10 discusses a technique for obtaining equiripple polynomials having a zero at $x = 0$ of specified multiplicity, an approximation problem discussed by Souto in his 1970 dissertation [104]. In this chapter, a differential equation in two polynomials is given and it is suggested that a Gröbner basis be used to obtain the sought coefficients. The resulting polynomials can be used to design analogue and digital IIR filters the properties of which are between those of the classical Butterworth and Chebyshev filters of types I and II.
Chapter 2

Constrained Least Square Design of FIR Filters Without Specified Transition Bands

2.1 Introduction

We consider the definition of optimality for digital filter design and suggest that a constrained least squared error criterion with no specified transition band is a useful complement to existing approximation criteria for filter design.

Consider, for example, a basic lowpass filtering scenario in which a signal of interest whose spectrum occupies the frequency range \((0, \omega_0)\) is embedded in an additive noise signal whose spectrum occupies the entire frequency range \((0, \pi)\). In this case, without further assumptions, no transition band naturally arises from the problem of removing the noise from the signal of interest: No part of the passband is more or less critical than any other part of the passband. Similarly, no part of the stopband is more or less critical than any other part of the stopband. In many practical cases there is no separation of the passband and stopband by a transition (or "don't care") band between them. Indeed, spectra of the desired and undesired signals often overlap.

An important exception to this is the design of filters used to select one out of two or more signals which have been designed to occupy well separated frequency bands. In this case, and other cases in which the signals to be filtered have no energy in a transition band, the use of transition bands in filter design is well motivated — the transition band constitutes a non-critical part of the frequency response. But even in these cases, there is in the "guard bands" usually some noise or other undesirable signals that one wants to remove. It is for this reason that transition region anomalies in
the frequency response are undesirable. Indeed, when large peaks occur in the "don't care" transition band of certain multiband Chebyshev filter designs [74], engineers decide they do care and alter the specifications or employ modified algorithms [103,107] to eliminate the peaks. In many cases, a transition band is introduced to reduce or remove the oscillations in the frequency response near the band edges caused by the Gibbs' phenomenon, not because transition bands naturally arises from the physics of the problem.

For the meaningful design of filters it is desirable to choose an error criterion that does not implicitly require unrealistic assumptions on the signals, such as the existence of a band separating desired signals and noise. Although these statements are informal, below we draw upon the results of Weisburn, Parks, and Shenoy [124] to give a mathematical justification for the inclusion of the transition region in the measure of approximation error.

This chapter (an earlier version of which is [98]) is organized as follows: In section 2.2 we establish notation and discuss ways in which Gibbs' phenomenon has previously been treated. We also discuss the constrained $L_2$ approach of Adams [1, 2] to filter design and the results of Weisburn, Parks, and Shenoy [124]. In section 2.3 we take into account the results of Weisburn et al. to modify the approach of Adams. The resulting design method adopts the view of Adams and is, at the same time, motivated by the results of Weisburn et al. The rapidly converging, robust, multiple exchange algorithm algorithm presented in section 2.4 produces peak constrained least square lowpass FIR filters. The result is a versatile design algorithm that will design linear and minimum phase FIR filters and which gives the best $L_2$ filter and a continuum of Chebyshev filters as special cases. Section 2.5 gives some interpretations and extensions of this method. A simple Matlab program that illustrates the algorithm for odd length lowpass filter design is given in the appendix of this chapter.
2.2 Preliminaries

The frequency response $H(\omega)$ of an FIR filter is given by the discrete-time Fourier transform of its impulse response $h(n)$:

$$H(\omega) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n}. \quad (2.1)$$

If $h(n) = h(N - 1 - n)$, then $H(\omega)$ has linear phase and can be written as

$$H(\omega) = A(\omega) e^{-jM\omega} \quad (2.2)$$

where $A(\omega)$ is the real-valued amplitude and $M = (N - 1)/2$ for length-$N$ filters [68, 93, 94]. For simplicity, symmetric odd length filters will be discussed in this chapter, in which case $A(\omega)$ can be written as

$$A(\omega) = \frac{1}{\sqrt{2}} a(0) + \sum_{n=1}^{M} a(n) \cos n\omega \quad (2.3)$$

where the impulse response coefficients $h(n)$ are related to the the cosine coefficients $a(n)$ by

$$h(n) = \begin{cases} 
\frac{1}{2} a(M - n) & \text{for } 0 \leq n \leq M - 1 \\
\frac{1}{\sqrt{2}} a(0) & \text{for } n = M \\
\frac{1}{2} a(n - M) & \text{for } M + 1 \leq n \leq N - 1 \\
0 & \text{otherwise}.
\end{cases} \quad (2.4)$$

Let $D(\omega)$ denote the desired amplitude. For example, see Figure 2.1 in which the desired amplitude of an ideal lowpass filter is shown. Approximating this discontinuous function by the cosine polynomial $A(\omega)$ given in (2.3) is the most basic filter design problem. Many of the various methods in the filter design literature can be distinguished by the ways in which they treat this discontinuity. Indeed, controlling the behavior of $A(\omega)$ in the region around the discontinuity has strongly influenced the development of filter design methods.

Two primary measures of approximation error are used in filter design. Let $E(\omega) = D(\omega) - A(\omega)$.
Figure 2.1: The desired amplitude of an ideal lowpass filter.

1. The weighted integral square error (or "L_2 error") is given by

\[
||E(\omega)||_2 = \left( \frac{1}{\pi} \int_0^\pi W(\omega)(A(\omega) - D(\omega))^2 \, d\omega \right)^{\frac{1}{2}}. \tag{2.5}
\]

2. The weighted Chebyshev error (or "L_\infty error") is given by

\[
||E(\omega)||_\infty = \max_{\omega \in [0, \pi]} |W(\omega)(A(\omega) - D(\omega))|. \tag{2.6}
\]

In both cases, \(W(\omega)\) is a nonnegative error weighting function. When \(W(\omega)\) is set to unity over \([0, \pi]\), the approximation measures above are called the unweighted (or uniformly weighted) integral square error and the unweighted (or uniformly weighted) Chebyshev error. The simplest method to design optimal FIR filters minimizes \(||A(\omega) - D(\omega)||_2\); we call the resulting filter the best \(L_2\) filter. As is well known, if the error weighting function is set to unity over \([0, \pi]\), then the best \(L_2\) filter is obtained by truncating the Fourier series of \(D(\omega)\) (the rectangular window method). Hence, for simple \(D(\omega)\), a closed form expression for the filter is easily found. But \(W(\omega) = 1\) is not generally used in practice, because best \(L_2\) filters with this error weighting possess large peak errors near the band edges. Moreover, the peak value of these "overshoots" does not diminish with increasing filter length. To overcome this
behavior, known as Gibbs' phenomenon, three main approaches have been employed: (i) the use of non-rectangular windows, (ii) the use of transition functions to continuously connect adjacent bands, and (iii) the use of zero-weighted transition bands placed between adjacent bands.

The use of these approaches has spawned a variety of filter design procedures having the following two desirable properties:

1. The procedure produces filters that do not suffer from Gibbs' phenomenon.

2. The procedure can be implemented using a computationally efficient numerical algorithm.

For example, variable order spline transition functions can be used with the integral square error approximation measure to obtain expressions for filters having good response behavior around the discontinuity [14]. Alternatively, when a zero-weighted transition band is used, the best weighted $L_2$ filter can be found by solving a system of linear equations [115]. In this case Gibbs' phenomenon is also eliminated: the peak error diminishes as the filter length increases. The use of zero-weighted transition bands also permits the meaningful use of the Chebyshev norm, an error measure for which the Parks-McClellan (PM) program produces optimal linear phase filters [68]. In fact, in order to design lowpass filters by minimizing the Chebyshev norm, either a transition function or a zero-weighted transition band must be specified:

For the discontinuous lowpass response given in Figure 2.1, the filter minimizing the unweighted Chebyshev error has a Chebyshev error of one half and is not unique.

Unfortunately, each of the three enumerated methods has its shortcoming:

1. **Windows:** Although the multiplication of the Fourier coefficients of $D(\omega)$ with a non-rectangular window is very simple, the method is generally considered sub-optimal because it is difficult to use it to minimize meaningful error measures (but see [28]) and because error weighting in the sense of (2.5) is not achieved.
2. **Transition Functions:** Although modifying the desired amplitude \( D(\omega) \) (so that it is no longer discontinuous) yields approximations that do not suffer from Gibbs’ phenomenon, the method does not directly approximate the original discontinuous desired lowpass amplitude function.

3. **Zero-Weighted Transition Bands:** By a zero-weighted transition band, we mean a region placed between two adjacent bands where \( W(\omega) \) is taken to be 0. Because \( A(\omega) - D(\omega) \) is not weighted there, zero-weighted transition bands are sometimes called "don't care" regions. The use of zero-weighted transition bands make the approximation problem easier. But if they are used, then unless the input signals have no energy in the transition band, the optimality of the best Chebyshev and \( L_2 \) filters in the operator norm sense [124] is problematic.

Therefore, although the use of these approaches makes easier the approximation of the discontinuous desired amplitude, they are all rather indirect methods for dealing with the discontinuity.

### 2.2.1 Adams’ Error Criterion

An early comparison of error criteria was made by Tufts and Francis [114]. More recently Adams [1, 2] described perhaps the most meaningful error criterion for filter design to date and suggested an iterative algorithm to design the corresponding best linear phase filters.

As Adams has noted, \( L_2 \) filter design is based on the assumption that the size of the peak errors can be ignored. Likewise, filter design according to the Chebyshev norm assumes the \( L_2 \) measure of approximation error is irrelevant. In practice, however, both of these criteria are important, a point Adams elaborates in [1]. Furthermore, Adams finds that the peak error of a best \( L_2 \) filter can be significantly reduced with only a slight increase in the \( L_2 \) error. Similarly, the \( L_2 \) error of an equiripple filter can be reduced with only a slight increase in the Chebyshev error. In Adams’ terminology, both equiripple filters and best \( L_2 \) filters are inefficient.
Consider lowpass filter design. To obtain filters having a better trade-off between these two criteria, Adams uses zero-weighted transition bands and proposes that $||A(\omega) - D(\omega)||_2$ be minimized subject to a constraint on the Chebyshev error. This is formulated as a quadratic program [1] as follows:

$$\min_{a \in \mathbb{R}^{N+1}} ||A(\omega) - D(\omega)||_2^2$$  \hspace{1cm} (2.7)

such that

$$L(\omega) \leq A(\omega) \leq U(\omega) \quad \text{for all } \omega \in [0, \omega_p] \cup [\omega_s, \pi].$$  \hspace{1cm} (2.8)

In (2.7), Adams weights the integral square error by the weight function:

$$W(\omega) = \begin{cases} 
W_p & \text{for all } \omega \in [0, \omega_p] \\
0 & \text{for all } \omega \in [\omega_p, \omega_s] \\
W_s & \text{for all } \omega \in [\omega_s, \pi].
\end{cases}$$  \hspace{1cm} (2.9)

In (2.8), the upper and lower bound functions $L(\omega)$ and $U(\omega)$ are given by

$$L(\omega) = \begin{cases} 
1 - \delta_p & \text{for all } \omega \in [0, \omega_p] \\
-\delta_s & \text{for all } \omega \in [\omega_s, \pi]
\end{cases}$$  \hspace{1cm} (2.10)

and by

$$U(\omega) = \begin{cases} 
1 + \delta_p & \text{for all } \omega \in [0, \omega_p] \\
\delta_s & \text{for all } \omega \in [\omega_s, \pi]
\end{cases}$$  \hspace{1cm} (2.11)

where $\delta_p$ and $\delta_s$ are the maximum allowed deviations from 1 and 0 in the passband and stopband.

This constrained $L_2$ approach allows the user to control the trade-off between the $L_2$ and Chebyshev errors, and produces best $L_2$ and Chebyshev filters as special cases. Of course, for a fixed filter length and a fixed $\delta_p$ and $\delta_s$ (each less than 0.5), it is not possible to obtain an arbitrarily narrow transition band. Therefore, if the band edges $\omega_p$ and $\omega_s$ are taken to be too close together, then the quadratic program (2.7,2.8) has no solution. Similarly, for a fixed $\omega_p$ and $\omega_s$, if $\delta_p$ and $\delta_s$ are taken too small,
then there is again no solution. In the terminology of quadratic programming [26], the feasible region is empty.

Although the algorithm in [1] tests for optimality upon termination by checking the non-negativity of Lagrange multipliers, during the iterations it does not enforce this non-negativity. For this reason, it may converge to a non-optimal filter. In [53,54] an algorithm for the design of nonlinear phase FIR filters according to the same error measure is proposed. It inspects the signs of Lagrange multipliers on each iteration so that, if the algorithm converges, then the filter to which it converges is guaranteed to be optimal. However, the algorithm in [54] is also not guaranteed to converge. But it was found that for lowpass filter design, whenever there exists a filter satisfying the constraints specified by the user, the algorithm converges in practice. To develop exchange algorithms for solving (2.7,2.8) that are guaranteed to converge to optimal filters, it is necessary to modify the algorithms of [1,53,54]. In [2], Adams et al. describe in detail appropriate modifications based on [27] that guarantee convergence and optimality. In [109], Sullivan and Adams extend the algorithm of [2] to the design of nonlinear phase FIR filters with constraints on the group delay.

2.2.2 Approximation by Operator Norms

Weisburn, Parks, and Shenoy [124] present a rigorous motivation for the use of the Chebyshev and $L_2$ error measures and discuss the use of zero-weighted transition bands (see also [101,102]). Using the theory of operator norms, they show that best Chebyshev filters minimize a worst case error signal energy, while best $L_2$ filters minimize a worst case pointwise error in the time domain. However, to show this optimality in the operator norm sense when a zero-weighted transition band is used, a hypothetical ideal prefilter is placed at the input, the frequency response of which is zero on the don't care region and 1 everywhere else [124].

Weisburn et al. begin by defining $E(\omega) = A(\omega) - D(\omega)$ as the frequency response of an error filter, $E$. They view the error filter as an operator, and use an operator
norm as a measure of approximation. The filter design problem then becomes one of finding \( A \) to minimize \( \| A - D \| \) where the norm is an operator norm. To define the norm of an operator, it is necessary to define a norm on the class of input signals, which we will denote \( \| \cdot \|_i \), and a norm on the class of output signals, \( \| \cdot \|_o \). Note that the norms used for the input and output signals do not have to be the same. The operator norm of \( E \) is then defined as

\[
\| E \| = \sup_{x \in U} \frac{\| Ex \|_o}{\| x \|_i}
\] (2.12)

where \( x \) represents the input signal of the error filter \( E \) and \( y = Ex \) represents the corresponding output signal. The supremum is over the space of input signals \( U \). This ratio indicates the amount by which the filter \( E \) "magnifies" or "attenuates" the input signal with respect to the chosen input and output norms.

It turns out that when the class of input signals is taken to be the space of all finite energy sequences and when the input and output signal norms are both taken to be the \( l_2 \) norm (\( \| x \|_i = \sqrt{\sum_n |x(n)|^2} \) and \( \| \cdot \|_i = \| \cdot \|_o \)), then the operator norm is equal to the unweighted Chebyshev norm, \( \| E(\omega) \|_\infty \), defined in (2.6). Therefore, the best Chebyshev filter minimizes a worst case output signal energy over a set of bounded energy input signals [124].

On the other hand, if the input signal space and norm is kept the same, but the output signal norm is taken to be \( \| y \|_o = \| y(n) \| \) for some fixed index \( n \), then the operator norm is equal to the unweighted integral square error, \( \| E(\omega) \|_2 \), defined in (2.5). Note that it is independent of the index \( n \). Consequently, the best \( L_2 \) filter minimizes a worst case pointwise error in the time domain over a set of bounded energy input signals [124].

Suppose \( D(\omega) \) is the usual discontinuous ideal lowpass frequency response. Further, suppose an error weighting function is used which is zero in a specified transition band. If the signals in the input class have no energy in the transition band, then the filters obtained by minimizing the weighted Chebyshev and \( L_2 \) norms are optimal in the operator norm sense [124]. However, if the input signals do have energy in the
specified transition bands, then the way in which the filters are optimal is unclear in general.

Suppose on the other hand, that a transition function is used to modify $D(\omega)$ so that it has a smooth transition between the passband and stopband. Then the best filters obtained according to any approximation measure do not correspond to the original desired discontinuous frequency response $D(\omega)$.

2.3 A New Criterion for Lowpass Filter Design

In light of the preceding discussion, we suggest that the use of explicitly specified transition bands in FIR filter design began in part with the desire to reduce peak errors near the band edges and that these "don't care" regions are, in some cases, used because they facilitate the approximation problem, not because they naturally arise from the filter problem under consideration.

Furthermore, because the minimization of the Chebyshev norm for lowpass filter design requires the use of a transition band, we propose for some problems that the $L_2$ norm be the primary optimization measure and that the peak errors be controlled by a constraint.* In the next section we present an algorithm for the design of peak constrained lowpass FIR filters according to an integral square error criterion that does not require the use of specified transition bands.

Let us define $E(\omega) = A(\omega) - D(\omega)$, where $D(\omega)$ is the ideal discontinuous lowpass amplitude. It will be convenient to make the meaning of the term "peak errors" more precise. By peak errors of a lowpass filter, we mean the values of $|E(\omega)|$ at the local minima and maxima of $A(\omega)$. With this definition, the peak errors do not include values of $E(\omega)$ at the edges of a transition band (see Figure 2.3).

Let $\omega_o$ be the cut-off frequency (the frequency at which the ideal lowpass amplitude is discontinuous). The design problem we propose uses the $L_2$ weight function and

*There are cases where the Chebyshev error should be truly minimized (e.g., narrow band interference at an unknown frequency), but simply reducing it or constraining it is generally sufficient.
lower and upper bound functions given by:

1. \( W(\omega) = 1 \) for all \( \omega \in [0, \pi] \)

2. \( U(\omega) = 1 + \delta_p, \; L(\omega) = 1 - \delta_p \), for all \( \omega \in [0, \omega_o] \)

3. \( U(\omega) = \delta_s, \; L(\omega) = -\delta_s \), for all \( \omega \in (\omega_o, \pi] \)

We propose the following problem formulation:

**New Criterion:** Minimize the unweighted integral square error (2.5) subject to the constraint that the local minima and maxima of \( A(\omega) \) lie within the specified lower and upper bound functions, \( L(\omega) \) and \( U(\omega) \).

The filters produced by the algorithm described below produces lowpass linear phase filters according to this criterion. They have frequency response amplitudes that are very similar in appearance to those obtained using the approach described by Adams. However, there are three main differences:

1. There is no region around the discontinuity that is excluded from the integral square error approximation measure. The algorithm minimizes the square error over the entire frequency range from 0 to \( \pi \) subject to the peak constraints.

2. The transition region is implicitly defined by the constrained \( L_2 \) minimization procedure. It is not dictated by the specification of transition band edges. Indeed, band edges are not specified.

3. There is no minimum achievable peak error size. That is, filters having arbitrarily small peak errors can be obtained. The problem of infeasibility encountered in the quadratic program formulation does not arise.

Because the approach taken here weights the \( L_2 \) error over the entire interval \( [0, \pi] \), there are no natural band edges to use in (2.8) of the quadratic program formulation. For this reason, the approach to filter design described here cannot be described by a
quadratic program. It should be noted that the new criterion also applies to multiband filter design, but for clarity, only the lowpass case is examined in this chapter. The extension to arbitrary responses is not developed here but it is expected that some extensions to non-piece-wise constant responses are possible.

The use of this criterion for lowpass filter design is suitable for situations where the maximum peak error size is to be controlled and where there is no reason to assume that the input signals to be filtered have no energy in a transition band. If the peak errors are unimportant, then the sinc function obtained by minimizing an unweighted unconstrained $L_2$ is more suitable. When it is known that input signals do not have energy in a transition band (or very little), then the use of a specified transition band is well motivated. In this case, the transition band constitutes a non-critical part of the response. Examples of this are when well separated signals are to be filtered. For such applications, the PM algorithm, the algorithm of Adams', or the linear programming algorithms of [107] are better suited.

The approach we propose in this chapter and the algorithm we describe are intended to complement the existing methods by providing an approach with few assumptions for a basic lowpass filtering problem.

2.4 A New Algorithm for Lowpass Filter Design

The algorithm described below produces lowpass linear phase FIR filters according to the new criterion described in the previous section. It is a rapidly converging, robust, simple multiple exchange algorithm that uses Lagrange multipliers and the Kuhn-Tucker conditions on each iteration [26,108]. Although we have not proven its convergence, the algorithm converges in practice when used for lowpass filter design. A Matlab program that implements this algorithm is especially simple and is given in the appendix of this chapter. For multiband filter design, the algorithm must be modified to obtain robust convergence (chapter 3).

The algorithm will design linear and minimum phase lowpass FIR filters and gives
the best $L_2$ filter and a continuum of Chebyshev filters as special cases. The algorithm can be modified to allow different $L_2$ error weighting in different bands and to allow other types of constraints. We have designed lowpass filters of lengths over 3,000 and have used loose and tight constraints that differed in the passband and stopband by factors as much as $10^6$.

2.4.1 The Equality Constrained Minimization Problem

The amplitude $A(\omega)$ of the filter minimizing the $L_2$ error subject to peak constraints will touch the lower and upper bound functions at certain extremal frequencies of $A(\omega)$. (By extremal frequencies of $A(\omega)$ we mean local minima and maxima of $A(\omega)$). If these frequencies were known in advance, then the filter could be found by minimizing $\|E\|_2^2$ subject to equality constraints at these frequencies. The procedure below determines the appropriate set of frequencies by solving a sequence of equality constrained quadratic minimization problems. The solution to each minimization problem is found by solving a linear system of equations.

This iterative algorithm is based on those of [1,54]. The constraints are on the values of $A(\omega_i)$ for the frequency points $\omega_i$ in a constraint set. On each iteration, the constraint set is updated so that at convergence, the only frequency points at which equality constraints are imposed are those where $A(\omega)$ touches the constraint. The equality constrained problem is solved with Lagrange multipliers. The algorithm below associates an inequality constrained problem with each equality constrained one. According to the Kuhn-Tucker conditions, the solution to the equality constrained problem solves the corresponding inequality constrained problem if all the Lagrange multipliers are non-negative (where the signs of the multipliers are defined appropriately). If on some iteration a multiplier is negative, then the solution to the equality constrained problem does not solve the corresponding inequality constrained one. For this reason, before the constraint set is updated in the algorithm described below, constraints corresponding to negative multipliers (when they appear) are sequentially
dropped from the constraint set. In this way, an inequality constrained problem is
solved on each iteration, albeit over a possibly smaller constraint set. It turns out
that in the special case of lowpass filter design considered here, this simple iterative
technique converges in practice.

Let the constraint set \( S \) be a set of frequencies \( S = \{\omega_1, \ldots, \omega_r\} \) with \( \omega_i \in [0, \pi] \). Let \( S \) be partitioned into two sets, \( S_I \) and \( S_u \), where \( S_I \) is the set of frequencies where we wish to impose the equality constraint

\[
A(\omega) = L(\omega),
\]

while \( S_u \) is the set of frequencies where we wish to impose the equality constraint

\[
A(\omega) = U(\omega).
\]

Let us have \( S_I = \{\omega_1, \ldots, \omega_q\} \) and \( S_u = \{\omega_{q+1}, \ldots, \omega_r\} \). To minimize \( \|E(\omega)\|_2 \) subject
to these constraints we form the Lagrangian [26, 59, 108]:

\[
\mathcal{L} = \|E(\omega)\|_2^2 - \sum_{i=1}^q \mu_i [A(\omega_i) - L(\omega_i)] + \sum_{i=q+1}^r \mu_i [A(\omega_i) - U(\omega_i)].
\]

Necessary conditions for \( \|E(\omega)\|_2 \) to be minimized subject to the constraints above,
are obtained by setting the derivative of \( \mathcal{L} \) with respect to \( a_k \) and \( \mu_i \) to zero. This
yields the following equations:

\[
\frac{\partial \|E(\omega)\|_2^2}{\partial a_k} - \sum_{i=1}^q \mu_i \frac{\partial A(\omega_i)}{\partial a_k} + \sum_{i=q+1}^r \mu_i \frac{\partial A(\omega_i)}{\partial a_k} = 0
\]

for \( 0 \leq k \leq M \),

\[
A(\omega_i) = L(\omega_i)
\]

for \( 1 \leq i \leq q \), and

\[
A(\omega_i) = U(\omega_i)
\]
for \( q + 1 \leq i \leq r \). According to the Kuhn-Tucker conditions, when the Lagrange multipliers \( \mu_1, \ldots, \mu_r \) are all nonnegative, then the solution to (2.16,2.17,2.18) minimizes \(||E(\omega)||_2||z\) subject to the inequality constraints

\[
A(\omega_i) \geq L(\omega_i)
\]

for \( 1 \leq i \leq q \), and

\[
A(\omega_i) \leq U(\omega_i)
\]

for \( q + 1 \leq i \leq r \).

Recalling (2.3), and letting \( W(\omega) = 1 \), equations (2.16,2.17,2.18) can be written as

\[
a + G^T \mu = c
\]

(2.21)

and

\[
Ga = d
\]

(2.22)

where \( a \) is the length \( M + 1 \) vector of unknown filter parameters \( a = (a_0, \ldots, a_M)^T \), and \( \mu \) is the vector of Lagrange multipliers, one for each frequency in the constraint set: \( \mu = (\mu_1, \ldots, \mu_r)^T \). \( G \) is the \( r \) by \( M + 1 \) matrix of cosine terms that calculates the amplitude response \( A(\omega) \) of the filter \( a \) at the frequencies in the constraint set \( S \). The elements of \( G \) are given by

\[
G_{i,0} = \frac{-1}{\sqrt{2}}
\]

(2.23)

\[
G_{i,k} = -\cos k\omega_i
\]

(2.24)

for \( 1 \leq i \leq q, 1 \leq k \leq M \) and

\[
G_{i,0} = \frac{1}{\sqrt{2}}
\]

(2.25)

\[
G_{i,k} = \cos k\omega_i
\]

(2.26)
for \( q + 1 \leq i \leq r, 1 \leq k \leq M \). \( c \) is the vector of coefficients for the unconstrained optimal \( L_2 \) filter given by

\[
\begin{align*}
c_0 &= \frac{\sqrt{2}}{\pi} \int_0^\pi D(\omega) \, d\omega \\
c_k &= \frac{2}{\pi} \int_0^\pi D(\omega) \cos k\omega \, d\omega.
\end{align*}
\] (2.27) (2.28)

Notice that the elements of \( c \) do not depend upon the constraints imposed on \( A(\omega) \), in fact, they are Fourier coefficients. Indeed, if there are no constraints imposed upon \( A(\omega) \), then \( a(n) \) equals \( c(n) \), the best \( L_2 \) filter. The term \( d \) is the vector of values the amplitude response \( A(\omega) \) is made to interpolate and is given by

\[
d_i = -L(\omega_i)
\] (2.29)

for \( 1 \leq i \leq q \), and

\[
d_i = U(\omega_i).
\] (2.30)

for \( q + 1 \leq i \leq r \). It is necessary to introduce minus signs for the lower bound constraints so that at the solution to the inequality constrained minimization problem, all of the Lagrange multipliers will have the same sign. Equations (2.21,2.22) can be combined into one matrix equation:

\[
\begin{bmatrix}
I_{M+1} & G^t \\
G & 0
\end{bmatrix}
\begin{bmatrix}
a \\
\mu
\end{bmatrix}
= \begin{bmatrix}
c \\
d
\end{bmatrix}.
\] (2.31)

In (2.31), \( I_{M+1} \) is the \((M + 1)\) by \((M + 1)\) identity matrix. Similar expressions can be derived for the even length filter and the odd symmetric filters [68].

It is easy to verify that

\[
\begin{align*}
\mu &= (GG^t)^{-1}(Gc - d) \\
a &= c - G^t\mu
\end{align*}
\] (2.32) (2.33)

is the solution to (2.31). Therefore, if the number of constraints \( r \) is small compared to the number of filter coefficients \((M + 1)\), then the system (2.31) is computationally
very inexpensive to solve. It requires the solution to an \( r \times r \) system of linear equations.

This is attributed to the use of \( W(\omega) = 1 \) over \([0, \pi]\). It is interesting to note that on each iteration the cosine coefficients \( a \) are obtained by adding to the best \( L_2 \) (Fourier) coefficients \( c \), a correction term. This is in contrast to the window method, in which the best \( L_2 \) coefficients are multiplied by a window.

For non-uniform weighting functions, the identity matrix in (2.31) becomes a full symmetric matrix \([13,14]\) and equations (2.16,2.17,2.18) become

\[
\begin{bmatrix}
R & G^t \\
G & 0
\end{bmatrix}
\begin{bmatrix}
a \\
\mu
\end{bmatrix} =
\begin{bmatrix}
c \\
d
\end{bmatrix}
\tag{2.34}
\]

where the elements of the vector \( c \) are given by

\[
c_0 = \frac{\sqrt{2}}{\pi} \int_0^\pi W(\omega)D(\omega) \, d\omega
\tag{2.35}
\]

\[
c_k = \frac{2}{\pi} \int_0^\pi W(\omega)D(\omega) \cos k\omega \, d\omega
\tag{2.36}
\]

and the elements of the matrix \( R \) are given by

\[
R_{0,0} = \frac{1}{\pi} \int_0^\pi W(\omega) \, d\omega
\tag{2.37}
\]

\[
R_{0,k} = R_{k,0} = \frac{\sqrt{2}}{\pi} \int_0^\pi W(\omega) \cos k\omega \, d\omega
\tag{2.38}
\]

\[
R_{k,l} = R_{l,k} = \frac{2}{\pi} \int_0^\pi W(\omega) \cos k\omega \cos l\omega \, d\omega
\tag{2.39}
\]

and where \( G \) and \( d \) are the same as above. It is easy to verify that

\[
\mu = (GR^{-1}G^t)^{-1}(GR^{-1}c - d)
\tag{2.40}
\]

\[
a = R^{-1}(c - G^t\mu)
\tag{2.41}
\]

is the solution to (2.34).

### 2.4.2 The Exchange Iterations

The equality constrained optimization procedure described above is performed at each step of an iterative algorithm. At each iteration, the constraint set frequencies
are updated, in much the same way as are the reference set frequencies of the Remez algorithm.

The algorithm begins with an empty constraint set so that the first filter designed is the best unconstrained $L_2$ filter. Then constraints are iteratively imposed upon $A(\omega)$ at selected frequencies until the best constrained $L_2$ filter is obtained. The constraint set is updated (i) by locating the local maxima of $A(\omega)$ that exceed the upper constraint function $U(\omega)$ and (ii) by locating the local minima of $A(\omega)$ that fall below the lower constraint function $L(\omega)$. Note that the "induced" band edges of the passband and stopband are not extremal frequencies of $A(\omega)$. Unlike the program of Adams and the Parks-McClellan program, the band edges are not included in the constraint set (in the case of the PM program, the reference set). The mechanism that yields a sharp transition between the passband and stopband is the inclusion of the transition region in the integral square error.

The algorithm can be summarized in the following steps.

1. **Initialization**: Initialize the constraint set to the empty set: $S = \emptyset$.

2. **Minimization with Equality Constraints**: Calculate the Lagrange multipliers associated with the filter that minimizes $\|E(\omega)\|_2$ subject to the equality constraints $A(\omega_i) = L(\omega_i)$ for $\omega_i \in S_l$, and $A(\omega_i) = U(\omega_i)$ for $\omega_i \in S_u$. (Solve equation (2.32)).

3. **Kuhn-Tucker Conditions**: If there is a constraint set frequency $\omega_i$ for which the Lagrange multiplier $\mu_i$ is negative, then remove from the constraint set the frequency corresponding to the most negative multiplier and go back to step 2. Otherwise, calculate the new cosine coefficients using equation (2.33) and go on to step 4.

4. **Multiple Exchange of Constraint Set**: Set the constraint set $S$ equal to $S_l \cup S_u$ where $S_l$ is the set of frequency points $\omega_i$ in $[0, \pi]$ satisfying both $A'(\omega_i) =$
0 and \( A(\omega_i) \leq L(\omega_i) \), and where \( S_u \) is the set of frequency points \( \omega_i \) in \([0, \pi]\)
satisfying both \( A'(\omega_i) = 0 \) and \( A(\omega_i) \geq U(\omega_i) \).

5. **Check for Convergence:** If \( A(\omega) \geq L(\omega) - \epsilon \) for all frequency points in \( S_i \)
and if \( A(\omega) \leq U(\omega) + \epsilon \) for all frequency points in \( S_u \) then convergence has been
achieved. Otherwise, go back to step 2.

According to the Kuhn-Tucker conditions, because \( \mu \geq 0 \) is ensured for each set
of computed cosine coefficients \( a \), each filter minimizes the \( L_2 \) error subject to the
inequality constraints (2.19,2.20) over a set of frequencies. At convergence, the
constraint set frequencies are exactly those extrema of \( A(\omega) \) where \( A(\omega) \) touches the
lower and upper bound function. It should be noted that negative multipliers generally appear only during the early iterations of the algorithm. \( \epsilon \) in step 4 is a small
number (like \( 10^{-6} \)) indicating the numerical accuracy desired. In appendix 2.7, some
issues concerning the convergence of the algorithm and the optimality of the filters it
produces are discussed.

A flowgraph is shown in Figure 2.2. The Matlab program below implements
this algorithm with \( W(\omega) = 1 \). For the sake of space and clarity, it uses a grid of
frequency values. However, it is much preferable to refine the location of the extremal
frequencies by Newton’s method; otherwise a rather dense grid is sometimes required
for convergence. The use of Newton’s method is easily incorporated.

The computational complexity is \( O(M^3) \) per iteration, however, the computation
required for each iteration depends upon the size of the constraint set for that iteration. Some of the efficient computational techniques that have been used to improve
the implementation of the Parks-McClellan program can also be applied to the algorithm described here [3, 5, 6, 9, 24, 103]. These techniques are used to (1) increase
the speed of execution, (2) reduce memory requirements, and (3) improve numerical
accuracy of the result.

**Example 1:** We let \( D(\omega) \) be the usual ideal lowpass filter with a cut-off frequency
\( \omega_o = 0.3\pi \) shown in Figure 2.1. We use 31 cosine coefficients \( (M = 30 \) and the filter
Figure 2.2: Flowgraph for the exchange algorithm for the constrained least square design of lowpass filters.
length is 61), $U(\omega) = D(\omega) + \delta$, and $L(\omega) = D(\omega) - \delta$ with $\delta = 0.02$. We use the $L_2$ weighting function $W(\omega) = 1$. In 4 iterations, the above described algorithm converges to the frequency response amplitude shown in Figure 2.3. In the figure, the circular marks indicate the 14 constraint set frequency points upon convergence. Compared to the best unconstrained $L_2$ filter shown in Figure 2.4, the constrained filter in Figure 2.3 has a considerably smaller peak error near the band edge. This is achieved with a small increase in the transition width and the $L_2$ error. The $L_2$ error and the peak error associated with the best unconstrained $L_2$ filter are $\|E\|^2 = 0.003375$ and 0.09369 respectively. For the constrained $L_2$ filter, they are $\|E\|^2 = 0.003858$ and 0.02 respectively. The resulting “induced” band edges of the constrained $L_2$ filter are $\omega_{ip} = 0.2728\pi$ and $\omega_{is} = 0.3270\pi$.

![Figure 2.3](image)

Figure 2.3 : Best constrained $L_2$ filter ($N = 61$, $\omega_o = 0.3\pi$) with $\delta_p = \delta_s = 0.02$. $\|E\|^2 = 0.003858$. The “induced” band edges are $\omega_{ip} = 0.2728\pi$ and $\omega_{is} = 0.3270\pi$.

The convergence of the algorithm is illustrated in Figure 2.5. The amplitude $A_0(\omega)$ shown in Figure 2.5(a) is the best unconstrained $L_2$ filter (given by the sinc function). The first set of constraints formed are taken to be the extrema of $A_0(\omega)$ that violate the upper and lower bound constraints. When (2.32) is used with this constraint set to calculate $\mu$ it turns out that two of the Lagrange multipliers are negative. (The two constraints corresponding to these two negative Lagrange multipliers are
Figure 2.4: Best unconstrained $L_2$ filter ($N = 61$, $\omega_o = 0.3\pi$), given by the sinc function. $\|E\|^2 = 0.003375$, peak error = 0.09369.

the local maximum and the local minimum nearest $\omega = 0$.) In this example, after the constraint set frequency corresponding to the more negative of these two negative multipliers is removed from the constraint set and (2.32) is used again, there is still one negative multiplier. After the corresponding constraint is removed from the constraint set and (2.32) is used a third time to calculate $\mu$, it is found that all the multipliers are positive. Now (2.33) is used to compute the filter coefficients $a$, the new amplitude $A_1(\omega)$ is shown in Figure 2.5(b). The circular marks in this figure indicate the constraint set used to obtain $A_1(\omega)$. $A_1(\omega)$ interpolates $L(\omega)$ and $U(\omega)$ at the constraint points because (2.32,2.33) were derived from equations (2.16,2.17,2.18) and because equations (2.17,2.18) are interpolation equations. As above, the extrema of $A_1(\omega)$ that violate the upper and lower bound constraints are used to form a new constraint set. Equation (2.32) is used with this constraint set to compute a new set of multipliers. It is found that all the new multipliers are positive and so no constraints are removed from this constraints set. Equation (2.33) is then used and the amplitude $A_2(\omega)$ is shown in Figure 2.5(c). $A_3(\omega)$ is obtained similarly, also without the appearance of negative multipliers.

It should be noted that this example and the following examples were generated
(a) Iteration 0. Maximum constraint violation = 0.07369.

(b) Iteration 1. Maximum constraint violation = 0.006481.

(c) Iteration 2. Maximum constraint violation = 0.001308.

(d) Iteration 3. Maximum constraint violation = 0.000003541.

Figure 2.5: An illustration of the convergence behavior.
using a version of the program that uses a grid to approximately locate the extrema of $A(\omega)$ and Newton's method to refine these frequencies.

2.5 Interpretations and Extensions

There are several observations and interpretations of this algorithm that may be helpful in understanding it in relation to other approaches and in modifying it for other applications.

The constraint set at each step in the iteration contains the candidates for the final extremal frequencies that touch the constraint. Satisfying these constraints forces $A(\omega)$ to interpolate $L(\omega)$ and $U(\omega)$ at the frequencies in the constraint set. This is quite similar to the behavior of the Remez algorithm used in the Parks-McClellan program. The process of the algorithm has three major differences to the PM program:

1. The number of reference set frequencies in the PM program is fixed and does not change throughout the algorithm, while here, the number of constraint set frequencies does change and is generally smaller than the number used in the PM program. Also, the alternation of the signs of $E(\omega)$ that holds for a Parks-McClellan equiripple filter and is enforced at each step of the Remez algorithm does not necessarily hold here.

2. The size of the Chebyshev error changes (increases) on each step of the PM program, while here it is prescribed. Here, the $L_2$ error generally increases and the induced transition band generally widens.

3. In the PM program, the number of reference set frequencies is $M + 2$, this is one greater than the number of unknown cosine coefficients. In our algorithm, unless the constraints are tight, the number of frequencies is less than the number of unknowns, and those degrees of freedom not used to satisfy interpolation constraints are used to minimize the integral square error at each step.
Nevertheless, the overall behavior is similar to that of the PM program. As in the Remez algorithm used in the PM program, on each iteration (1) an optimization problem is solved over a finite set of frequencies and (2) the set of frequencies is updated. Indeed, the interpolation step of the Remez algorithm can be interpreted as an optimization problem: The filter found at each iteration of the Remez algorithm minimizes the Chebyshev error over the updated reference set [71].

Often it is desirable to include equality constraints on the value and derivatives of $A(\omega)$ at prescribed frequency points. An application is given, for example, in [65], where flatness at $\omega = 0$ is achieved by imposing appropriate equality constraints. The inclusion of these and other linear equality constraints on the cosine coefficients in the approach described in this chapter is straightforward. It requires only the use of extra Lagrange multipliers, the signs of which are unimportant. For example, magnitude squared design of minimum phase filters can be accomplished by taking the lower bound function $L(\omega)$ to be 0 in the stopband and by spectrally factoring the resulting nonnegative frequency response amplitude. A polynomial root finding algorithm and the computation of spectral factors is discussed in [55].

2.5.1 Chebyshev Solutions

Observe that if, for a fixed number of filter coefficients, the constraint on the weighted Chebyshev error $\|E(\omega)\|_\infty$ in [1,54] is chosen too small, (or equivalently, $\delta_p$ and $\delta_s$ in (2.7.2.8) are chosen too small) then no filter satisfies the constraint. In this case the algorithms of [1,54] can not converge. Although this problem can be avoided by computing the minimum value of $\|E(\omega)\|_\infty$ with the Parks-McClellan program, it is interesting to note that there is no minimum $\delta_p$ and $\delta_s$ below which the approach described in this chapter fails to converge. If $\delta_p$ and $\delta_s$ are taken to be small, then the transition region between the passband and stopband simply becomes wider. Therefore, for a fixed $\omega_o$, by decreasing $\delta_p$ and $\delta_s$, a continuum of Parks-McClellan equiripple filters are obtained.
Example 2: We use the same desired lowpass amplitude as in example 1, we keep \( M = 30 \), but we use \( \delta = 0.004 \). We again use the unity \( L_2 \) weighting function. The resulting frequency response shown in Figure 2.6 is obtained in 6 iterations and the size of the constraint set upon convergence is 30. Here, the peak error is significantly reduced with a corresponding increase in the transition width and \( L_2 \) error. The \( L_2 \) error associated with this filter is \( ||E||_2^2 = 0.004780 \). The resulting "induced" band edges of this constrained \( L_2 \) filter are \( \omega_{ip} = 0.2576\pi \) and \( \omega_{is} = 0.3421\pi \).

![Figure 2.6: Best constrained \( L_2 \) filter (\( N = 61, \omega_o = 0.3\pi \)) with \( \delta_p = \delta_s = 0.004 \). \( ||E||_2^2 = 0.004780 \). The "induced" band edges are \( \omega_{ip} = 0.2576\pi \) and \( \omega_{is} = 0.3421\pi \).](image)

Note that although this filter was not designed using the Parks-McClelland program, it corresponds to a best Chebyshev filter for appropriately chosen band edges because then the alternation theorem will be satisfied. For this example, this algorithm takes about the same number of iterations as does the Parks-McClelland program when the PM program is executed without a superior initialization of reference frequencies such as those given in [9,24,103]. This comparison is made to evaluate the convergence in the number of iterations, not to compare execution times.

It is interesting that the filter in example 2 is both (i) a best peak constrained \( L_2 \) filter, and (ii) a best Chebyshev filter for an appropriate transition band. There are other algorithms for the design of equiripple filters with specified peak errors.
[37, 40–42, 95, 96], but the peak constrained $L_2$ approach used here gives a way to design a subset of such filters which incorporates the $L_2$ error.

2.5.2 Trade-off Curves

In [1] Adams provides a curve illustrating the trade-off between the weighted $L_2$ error and the Chebyshev error for the filters produced by his approach. It is quite convincing that the filters on the endpoints of this curve do not provide the most desirable trade-off. The same is true here. By decreasing the peak error $\delta$ to 0, a curve illustrating the trade-off between the unweighted integral square error and $\delta$ can be obtained.

Example 3: Figure 2.7 shows the curve for length 61 filters designed using the approach of this chapter, where the cut-off frequency is $\omega_c = 0.3\pi$ and $\delta = \delta_p = \delta_s$. $\delta$ is varied to obtain the curve in the figure. The circle at the right end of the figure indicates the best unconstrained $L_2$ filter. As $\delta$ is decreased from that point, the $L_2$ error increases as illustrated. The circular mark at $\delta \approx 0.0086$ indicates the point where $\delta$ first becomes small enough to produce a Parks-McClellan equiripple filter. In this example, all points on the curve to the left of this point represent Parks-McClellan equiripple filters. Because there is no smallest value for $\delta$, the curve approaches a point on the $\delta = 0$ axis. We conjecture that the point on the $\delta = 0$ axis this curve approaches represents a maximally flat (digital Butterworth) filter [37].

We should note that even though all points on the curve for $\delta < 0.0086$ in Figure 2.7 represent equiripple filters, this is not true in general. For example, suppose $\delta_s$ is decreased towards zero and $\delta_p$ is kept constant. Then it is sometimes the case that the set of $\delta_s$, for which the points on the curve represent equiripple filters, is a union of disjoint intervals.
Figure 2.7: The trade-off curve for integral square error and peak error size: $\|E\|_2^2$ versus $\delta_s$. This curve illustrates the trade-off for filters of length 61 designed with $\omega_o = 0.3\pi$, and $\delta = \delta_p = \delta_s$.

2.5.3 Specified Band Edges

The approach taken in this chapter does not preclude the specification of a transition band edge in the design of a lowpass filter. Let $\omega_p < \omega_o$ be a specified passband edge. This subsection describes how to append the constraint

$$L(\omega_p) \leq A(\omega_p) \leq U(\omega_p)$$

(2.42)

to the formulation of section 2.3. By doing so, the frequency response amplitude $A(\omega)$ will be guaranteed to lie between the lower and upper bound functions $L(\omega)$ and $U(\omega)$ for all $\omega$ in the passband $[0, \omega_p]$. This is because the only way in which $A(\omega)$ can violate the lower and upper bound constraints in the passband is by violating one of them at $\omega_p$ or by violating one of them at a local extremal. Because the algorithm described in section 2.4.2 ensures that $L(\omega)$ and $U(\omega)$ are not violated at the local extremals of $A(\omega)$, it is sufficient to append the single constraint (2.42). The appropriate modification to the algorithm of section 2.4.2 is described below.

Appending the constraint (2.42) to the existing constraints requires modifying only the way in which the constraint set is updated. There are two issues that must...
be addressed.

First, notice that the constraint (2.42) may be satisfied by the filter produced by the basic algorithm of section 2.4.2 — the constraint may be met with no additional effort. This is the case when the induced passband edge \( \omega_p \) is closer to \( \omega_o \) than is the specified passband edge \( \omega_p \). When this is not the case, it is necessary to simply append \( \omega_p \) to the constraint set. To detect, during the iterative algorithm, exactly when it is necessary to include \( \omega_p \) in the constraint set, the following decision rule is used: Let \( \omega_a \) be the passband extremum of \( A(\omega) \) closest to \( \omega_o \). If \( \omega_a < \omega_p < \omega_o \) and \( A(\omega_p) < L(\omega_p) \), then include \( \omega_p \) in the constraint set, otherwise leave the constraint set unchanged.

The second issue that must be addressed is the occurrence of an over-constrained problem on some iteration. After appending \( \omega_p \) to the constraint set, the number of constraints may outnumber the number of cosine coefficients by one. When this occurs, equations (2.21,2.22) are in general over-determined and can not be solved. Consequently, a frequency must be removed from the constraint set before the algorithm can proceed. This occurs only when the constraints are relatively tight, in which case the algorithm described here reproduces a modified Parks-McClellan algorithm (see chapter 4). Likewise, the rule for deciding which frequency to remove is similar to the rule used in chapter 4 and is described below. Note that in this situation, the constraint set necessarily contains 0 and \( \pi \). Step 4 of the algorithm described in section 2.4.2 becomes:

4 Multiple Exchange of Constraint Set: Let \( S_l \) be the set of frequency points \( \omega_i \) in \([0, \pi]\) satisfying both \( A'(\omega_i) = 0 \) and \( A(\omega_i) \leq L(\omega_i) \). Let \( S_u \) be the set of frequency points \( \omega_i \) in \([0, \pi]\) satisfying both \( A'(\omega_i) = 0 \) and \( A(\omega_i) \geq U(\omega_i) \).

Let \( \omega_a \) be the passband extremum of \( A(\omega) \) closest to \( \omega_o \). If \( \omega_a < \omega_p < \omega_o \) and \( A(\omega_p) < L(\omega_p) \), then let \( S_l = S_l \cup \omega_p \).

If \( \omega = 0 \) is a local maximum of \( A(\omega) \), let \( E_0 = A(0) - U(0) \), otherwise set \( E_0 = L(0) - A(0) \). If \( \omega = \pi \) is a local maximum of \( A(\omega) \), let \( E_\pi = A(\pi) - U(\pi) \),
otherwise set $E_\pi = L(\pi) - A(\pi)$.

If $|S_l| + |S_u| = M + 2$ and $E_0 < E_\pi$, then remove 0 from $S_l$ or $S_u$, whichever contains 0.

If $|S_l| + |S_u| = M + 2$ and $E_0 \geq E_\pi$, then remove $\pi$ from $S_l$ or $S_u$, whichever contains $\pi$.

Set the constraint set $S$ equal to $S_l \cup S_u$.

A stopband edge $\omega_s$ can be specified instead of a passband edge in exactly the same way. If both the passband and the stopband are to be specified simultaneously, then the problem can be posed as a quadratic program as discussed above, and Adams’ approach should be used. When both band edges are specified, it is possible that no solution exists because the transition band can not be arbitrarily sharp, as mentioned above. Note that here a distinction is being made between the cut-off frequency $\omega_o$ and the band edges $\omega_p$ and $\omega_s$ ($\omega_p \leq \omega_o \leq \omega_s$).

**Example 4:** We use the same desired lowpass frequency response as in the previous two examples with $\delta = 0.020$ and $M = 30$, but we require that the passband edge be located at $0.285\pi$. We again use the $L_2$ weighting function $W(\omega) = 1$. In 7 iterations the algorithm converges to the frequency response shown in Figure 2.8. The size of the constraint set upon convergence is 28. The $L_2$ error associated with this filter is $||E||_2^2 = 0.006893$. The resulting “induced” stopband edge of this constrained $L_2$ filter is $\omega_{is} = 0.3376\pi$.

2.5.4 The Use of $L_2$ Weighting

Although the algorithm described in section 2.4.2 was introduced with a uniform $L_2$ weighting function, the algorithm can also be used with an $L_2$ weighting function that equals zero in a specified transition band, if so desired. As discussed above, when it is known that the signals in the input class have no (or little) energy in a transition region, the use of a zero-weighted (or lightly-weighted) transition band is well
motivated. The only modification required is the substitution of equation (2.34) for equation (2.31). The method for performing the multiple exchange of the constraint set remains unchanged. In this case, the $L_2$ weighting function possesses band edges, but the upper and lower bound functions are not enforced at these two frequencies. As in section 2.4.2, the upper and lower bound functions are used to constrain the frequency response amplitude only at its local extrema. This variation of the algorithm combines the approach of Adams' with the approach suggested in section 2.3 in which a uniform $L_2$ weighting is used. Like the uniformly $L_2$ weighted approach, this variation avoids the infeasibility problems associated with the quadratic program approach.

2.5.5 Remarks on Comparisons

As mentioned above, when the present algorithm yields an equiripple filter, it gives the same result as the PM algorithm when the PM algorithm is used with the appropriate specifications. Also, the frequency responses of the filters produced by the algorithm described in this chapter are similar to those obtained by the approach of Adams. It should be noted that although the responses are similar, they are not exactly the
same in general. This is because the quadratic program formulation Adams gives is not equivalent to the formulation given in section 2.3. The differences lie in the weighting of the error and the way in which the band edges are treated. Note that when the algorithm presented here and that of Adams each give an equiripple PM filter with the same lower and upper bound functions and the same transition band ("induced" in the case of the present algorithm) then certainly the two filters are identical.

The approaches of [37,40–42] should also be mentioned. While they employ implicitly defined transition bands and provide direct control of the peak errors, those algorithms (i) do not incorporate the $L_2$ error into the design procedure and (ii) provide only approximate control of the location of the cut-off frequency. (Although for lowpass filter design limitation (ii) can be overcome, see chapter 4 and [95]). In addition, it should also be noted that while the program of [107] is very flexible, it does not incorporate the $L_2$ error.

2.5.6 Multiband Filters

When used for the design of multiband filters, the simple algorithm we have described in this chapter for lowpass filter design does not, in general, converge. We have have found as in [2] that it is necessary to use a single point update procedure for some iterations to obtain robust convergence, see chapter 3. Thus, by maintaining the approach of using a sequence of equality constrained $L_2$ minimizations, best peak constrained $L_2$ multiband filters can be readily designed without the use of "don't care" regions. A program for the multiband case using the criterion described in this chapter is described in chapter 3. Adams et al. address algorithm issues concerning the design of multiband filters via the quadratic program formulation in detail in [2].
2.6 Conclusion

We have considered the design of optimal filters and have discussed the implicit assumptions associated with the use of explicitly specified transition bands in the frequency domain design of FIR filters. We have also put forth the notion that explicitly specified transition bands have been introduced in the filter design literature in part as an indirect approach for dealing with Gibbs' phenomenon occurring at the discontinuities in the desired frequency response. Moreover, the results of Weisburn, Parks, and Shenoy suggest that if a "don't care" region is used for filter design, then unless the input signals have no energy in the "don't care" region, the optimality of the best Chebyshev and $L_2$ filters in the operator norm sense is problematic. Because the minimization of the Chebyshev norm requires the use of a specified transition band, this suggests that the Chebyshev criterion is better suited as a constraint rather than the primary optimization criterion. This is also consistent with the motivation Adams gives to support the constrained $L_2$ approach described in [1].

This chapter (i) proposes that the unweighted (or weighted) integral square error be minimized such that the peak errors lie within the specified tolerances, and (ii) describes a simple multiple exchange algorithm for lowpass filter design according to this design formulation that is robust and efficient. Because the proposed approach does not ignore the $L_2$ error around the band edge, it does not implicitly assume that signals in the input class have no frequency content there. In addition, the constraints imposed upon the peak errors can be made arbitrarily small. For a fixed cut-off frequency, it also gives the best $L_2$ filter and a continuum of Chebyshev filters as special cases. With the weighting function $W(\omega) = 1$, the optimal filter coefficients are obtained by making a simple additive correction to the Fourier series coefficients.

The approach taken in this chapter is distinct from many other filter design methods because it does not exclude from the integral square error a region around the cut-off frequency, and yet, it overcomes Gibbs' phenomenon without resorting to windowing or "smoothing out" the discontinuity of the ideal lowpass filter. The al-
algorithm is also appealing because it can be implemented with an especially simple Matlab program.

Appendices

2.7 Optimality and Convergence

To discuss the optimization problem discussed above, define the feasible set $Q$:

$$Q = \{ a \in \mathbb{R}^{M+1} : L(\omega_i) \leq A(\omega_i) \leq U(\omega_i) \text{ for extremal frequencies } \omega_i \text{ of } A(\omega) \}. \quad (2.43)$$

The problem is to minimize $\|E\|_2^2$ over the set $Q$. Since $Q$ is closed and $\|A - D\|_2^2$ is a convex function of $a$, a minimizer exists. Usually uniqueness of a minimizer is established by ascertaining the convexity of the feasible set. However, the set $Q$ is not convex as is easily explained: Note that a maximally flat frequency response will always be feasible (it has extremal frequencies only at 0 and $\pi$, where $A(\omega)$ equals 1 and 0 respectively). If a filter is obtained by averaging a feasible equiripple (PM) filter and maximally flat (Herrmann) filter, then the frequency response amplitude of the filter will have local extrema around the cut-off frequency which will generally violate the upper and lower bound functions $L(\omega)$ and $U(\omega)$. Therefore $Q$ is not convex. Because $Q$ lacks convexity, it is necessary to use a different perspective to discuss optimality, as follows.

Although the minimization of $\|E\|_2^2$ over $Q$ is not a quadratic program, it is closely related to one. Suppose the $L_2$ weighting function is set to unity throughout the following discussion. If the algorithm converges, then the filter obtained by the algorithm is optimal in the sense that it is the solution to a meaningful quadratic program: Given a filter produced by the algorithm, define the two "induced" band edges as follows: Define the induced passband edge $\omega_p$ to be the highest frequency less than $\omega_o$ at which $A(\omega)$ equals $L(\omega)$. Similarly, define the induced stopband edge
\( \omega_{i2} \) to be the lowest frequency greater than \( \omega_o \) at which \( A(\omega) \) equals \( U(\omega) \). See Figure 2.9. Second, label the two extremals of \( A(\omega) \) that are adjacent to \( \omega_{ip} \) and \( \omega_{i2} \): Let \( \omega_a \) be the frequency at which \( A(\omega) \) achieves its first local maximum to the left of \( \omega_o \). Similarly, let \( \omega_b \) be the frequency at which \( A(\omega) \) achieves its first local minimum to the right of \( \omega_o \). With these definitions of \( \omega_a \), \( \omega_b \), \( \omega_{ip} \) and \( \omega_{i2} \), it can be said that, if \( \omega_p \) and \( \omega_s \) are two frequencies satisfying \( \omega_a \leq \omega_p \leq \omega_{ip} \), and \( \omega_{i2} \leq \omega_s \leq \omega_b \), then the filter obtained by the algorithm described above solves the quadratic program (2.7,2.8), where the weight function \( W(\omega) = 1 \) for all \( \omega \in [0, \pi] \) is used, not the weight function of expression (2.9).

![Graphs](image)

(a) passband detail  
(b) stopband detail

Figure 2.9 : Passband and stopband details of the amplitude shown in Figure 2.3. \( \omega_a = 0.2613\pi, \omega_{ip} = 0.2728\pi, \omega_{i2} = 0.3270\pi, \omega_b = 0.3386\pi. \)

The intervals \([\omega_a, \omega_{ip}]\) and \([\omega_{i2}, \omega_b]\) give insight into the properties of the solution. If \( \omega_s \in [\omega_{i2}, \omega_b] \) but \( \omega_p \) is taken to be a frequency less than \( \omega_a \), then the solution to the quadratic program (2.7,2.8) is a filter having peak errors in the passband that exceed \( \delta_p \). On the other hand, if \( \omega_p \) is taken to be a frequency greater than \( \omega_{ip} \), then the \( L_2 \) error of the resulting filter must be greater (because it is subject to additional
constraints). Similar statements are true for \( \omega_s \). Therefore, these values have the following two properties:

1. Filters obtained by solving the quadratic program (2.7,2.8, but with \( W(\omega) = 1 \forall \omega \)) with narrower transition widths \( (\omega_s - \omega_p < \omega_{is} - \omega_{ip}) \) necessarily have a greater \( L_2 \) error (assuming same the peak constraints).

2. Filters obtained by solving the quadratic program (2.7,2.8, but with \( W(\omega) = 1 \forall \omega \)) with wider transition widths \( (\omega_s - \omega_p > \omega_b - \omega_a) \) have peak errors around \( \omega_o \) that exceed \( \delta_p \) and/or \( \delta_s \).

Although we have not proven that the filters obtained by the algorithm are global minimizers of \( \|E\|^2 \) over \( Q \), the quadratic program analysis given here is highly suggestive.

Regarding the convergence of the algorithm, the non-convexity of \( Q \) has little bearing. Non-convexity of the feasible set is a problem for algorithms that proceed by updating one feasible solution to obtain another feasible solution by moving within the feasible set. The algorithm described above begins with the best unconstrained minimizer, and each filter produced during the course of the algorithm is infeasible. This sequence of filters approaches the feasible set. The algorithm terminates exactly when feasibility is achieved. The progress of this kind of algorithm is affected less by the lack of feasible set convexity.

The quadratic program analysis above also suggests that a unique minimizer exists (the QP has a unique minimizer), although we do not give a proof of this here. Note that because the algorithm always begins with the best unconstrained \( L_2 \) filter, there is no ambiguity about the initial filter used in the iterative procedure above. However, as mentioned above, we have not proven the convergence of the algorithm presented in this chapter, but have found it to converge reliably in practice for lowpass filter design. Indeed, even though it can be posed as a sequence of similar quadratic programs, the convergence of this algorithm is not supported by the theory of quadratic
programming. That it does in fact converge in practice is due to properties of the particular problem of lowpass filter design. When used for bandpass filter design, for example, this algorithm does not converge in general. For multiband filter design we have found as in [2] that it is necessary to modify the update procedure for some iterations to obtain robust convergence. It should be noted, however, that because the constrained $L_2$ approach of Adams can be formulated as a quadratic program, the existence of a unique optimal solution to the problem posed by him is assured, and the algorithm of [2] is guaranteed to converge to it.

2.8 Matlab Program

The Matlab program c12lp below implements the algorithm described in section 2.4.2. The program local.max in below computes the indexes of a vector corresponding to its local maxima by comparing each vector element with the two adjacent elements. This version of c12lp does not use Newton's method to refine the position of the extrema of $A(\omega)$, so a possibly dense grid is required in order to obtain convergence ($L \geq 2^{[\log_{10}(10m)]}$). A version using Newton's method (and other programs) can be obtained from the authors or electronically on the World Wide Web.

function h = c12lp(m,wo,up,lo,L)
% Constrained L2 Low Pass FIR filter design
% Author: Ivan Selesnick, Rice University, 1994
% See: Constrained Least Square Design of FIR
% Filters Without Specified Transition Bands
% by I.W.Selesnick, M.Lang, C.S.Burrus
% h : 2*m+1 filter coefficients
% m : degree of cosine polynomial
% wo : cut-off frequency in (0,pi)
% up : [upper bound in passband, stopband]
% lo : [lower bound in passband, stopband]
% L : grid size
% example
% up = [1.02, 0.02]; lo = [0.98, -0.02];
% h = c12lp(30,0.3*pi,up,lo,2^11);
\begin{verbatim}
x = sqrt(2);    \quad w = [0:L] * pi/L;
Z = zeros(2*L-1-2*m,1);   \quad q = round(wo*L/pi);
u = [up(1)*ones(q,1); up(2)*ones(L+1-q,1)];
l = [lo(1)*ones(q,1); lo(2)*ones(L+1-q,1)];
c = 2*[lo/r; [sin(w0*[1:m])./([1:m])]'/pi;
a = c;   \quad \text{% best L2 cosine coefficients}
mu = □;   \quad \text{% Lagrange multipliers}
SN = 1e-6;   \quad \text{% Small Number}
while 1
  \% ----- calculate A -------------------------------
  A = fft([a(1)*r; a(2:m+1); z(1-m:-1:2)]);   \quad A = real(A(1:L+1))/2;
  \% ----- find extremals -----------------------------
  kmax = local_max(A); \quad kmin = local_max(-A);
  kmax = kmax( A(kmax) > u(kmax) - 10*SN );
  kmin = kmin( A(kmin) < l(kmin) + 10*SN );
  \% ----- check stopping criterion -----------------
  Eup = E(kmax) - u(kmax); \quad Elo = l(kmin) - A(kmin);
  E = max([Eup; Elo; 0]); if E < SN, break, end
  \% ----- calculate new multipliers -----------------
  n1 = length(kmax); \quad n2 = length(kmin);
  O = [ones(n1,m+1); -ones(n2,m+1)];
  G = 0 .* cos(w([kmax;kmin])*[0:m]);
  G(:,1) = G(:,1)/r;
  d = [u(kmax); -l(kmin)];
  mu = (G*G') \ (G*c-d);
  \% ----- remove negative multiplier ---------------
  [min_mu,K] = min(mu);
  while min_mu < 0
    G(K,:) = □; d(K) = □;
    mu = (G*G') \ (G*c-d);
    [min_mu,K] = min(mu);
  end
  \% ----- determine new coefficients ---------------
  a = c-G*mu;
end
h = [a(m+1:-1:2); a(1)*r; a(2:m+1)]/2;
\end{verbatim}


```matlab
% finds location of local maxima
s = size(x); x = [x(:)]'; N = length(x);
b1 = x(1:N-1)<=x(2:N); b2 = x(1:N-1)>x(2:N);
k = find(b1(1:N-2)&b2(2:N-1))+1;
if x(1)>x(2), k = [k, 1]; end
if x(N)>x(N-1), k = [k, N]; end
k = sort(k); if s(2) == 1, k = k'; end
```
Chapter 3

Constrained Least Square Design of Multiband FIR Filters

3.1 Introduction

In the previous chapter we described a constrained least square approach to FIR filter design that does not use “don’t care” regions and described a very simple multiple-exchange algorithm for the design of lowpass (and highpass) linear phase FIR filters according to this approach. Unfortunately, when applied to the design of multiband filters (bandpass, bandstop, etc), the same algorithm does not converge reliably. In this chapter we describe a modification to that algorithm that makes it converge for multiband filter design. The modified algorithm remains simple and converges rapidly in most cases.

3.2 Example

As stated above, the algorithm of Chapter 2 may not converge when it is applied to bandpass filter design. In these cases, the failure of the algorithm to converge takes a specific form. Instead of converging to a single filter, the algorithm will end up cycling between two different filters, neither of which satisfy the specified peak gain constraints. The following example illustrates the way in which the algorithm of Chapter 2 may fail when it is used to design a length 63 bandpass filter.

Consider the design of a bandpass filter with cut-off frequencies at $\omega_1 = 0.2\pi$ and $\omega_2 = 0.4\pi$. The ideal frequency response amplitude $D(\omega)$ is shown in figure 3.1. Further, suppose that at the local maxima and minima of $A(\omega)$ in the passband,
\( A(\omega) \) is required to lie between 0.99 and 1.01. To simplify this illustrative example, the peak errors in the stopbands are not required to meet any ripple size constraints. When the algorithm of Chapter 2 is applied to the design of a length 63 bandpass filter with these constraints and cut-off frequencies, it fails to converge. After several iterations that algorithm will cycle between the two filters shown in Figures 3.2 and 3.3.

![Figure 3.1: The desired amplitude of an ideal bandpass filter.](image)

The algorithm of Chapter 2 employs a constraint set (a set of interpolation points) and proceeds as follows. On each iteration (1) the set of interpolation points is updated and (2) the least square error filter satisfying the interpolation constraints is found. The set of interpolation points is updated from one iteration to the next by setting it equal to the set of local minima and maxima where the new frequency response amplitude violates the lower and upper bound constraints.

In this example, this exchange algorithm of Chapter 2 fails to converge because on some iteration the new set of interpolation points is the same as a previous set. This is made clear in Figures 3.2 and 3.3 where it can be seen that the extremal points in one figure are the interpolation points in the other figure.
Figure 3.2: Previous algorithm applied to bandpass filter design. Even iterations.

Figure 3.3: Previous algorithm applied to bandpass filter design. Odd iterations.
3.3 New Algorithm

The modified algorithm remains simple, and although we have not proven its convergence, it converged for all examples with which it was tested, and for most of those examples, converged very rapidly. Like the algorithm of Chapter 2, the new algorithm is a multiple-exchange algorithm that uses Lagrange multipliers and the Kuhn-Tucker conditions on each iteration. It also gives the best $L_2$ filter and a continuum of Chebyshev filters as special cases. The algorithm is similar to that described in Chapter 2, however, it employs an additional inner-loop.

To describe the algorithm for multiband filter design, let $\omega_1, \ldots, \omega_K$ be the cut-off frequencies and $m_0, \ldots, m_K$ be the magnitudes of a $K + 1$ band filter whose desired frequency response amplitude is given by

$$D(\omega) = \begin{cases} 
m_0 & \text{for } 0 \leq \omega < \omega_1 \\
m_k & \text{for } \omega_k \leq \omega < \omega_{k+1} \\
m_K & \text{for } \omega_K \leq \omega < \pi. 
\end{cases} \tag{3.1}$$

For a bandpass filter, for example, we might have $\omega_1 = 0.2\pi$, $\omega_2 = 0.4\pi$, $m_0 = 0$, $m_1 = 1$, and $m_2 = 0$. In this case, $D(\omega)$ is shown in Figure 3.1.

Let the lower and upper bound functions $L(\omega)$ and $U(\omega)$ be specified by the user such that they are constant within each band* and satisfy the following:

1. $L(\omega) \leq D(\omega)$
2. $U(\omega) \geq D(\omega)$
3. $U(\omega) > L(\omega)$.

We propose that the integral square error be minimized such that the local minima and maxima of $A(\omega)$ lie within the lower and upper bound functions $L(\omega)$ and $U(\omega)$.

*It is not necessary that $L(\omega)$ and $U(\omega)$ be constant in each band, but they should be smooth. It is assumed that they are constant within each band for simplicity.
The amplitude $A(\omega)$ of the filter minimizing the $L_2$ error subject to these constraints will touch the lower and upper bound functions at certain extremal frequencies of $A(\omega)$. (By extremal frequencies of $A(\omega)$ we mean local minima and maxima of $A(\omega)$.) If these frequencies were known in advance, then the filter could be found by minimizing $\|E\|^2$ subject to equality constraints (interpolation requirements) at these frequencies. The iterative procedure below determines these frequencies by updating a set of candidate frequencies (a constraint set). Each filter in this iterative procedure is found by minimizing the $L_2$ error subject to interpolation requirements at these candidate frequencies. The sets of candidate frequencies will be called "constraint sets" because constraints will be imposed upon $A(\omega)$ over these frequencies.

On each iteration, the constraint set is updated so that at convergence, the only frequency points at which equality constraints are imposed are those where $A(\omega)$ touches the constraint. The equality constrained problem is solved with Lagrange multipliers.

3.3.1 Equality Constraints

The equality constraints are the same as those of section 2.4.1 and give rise to the same equations as those given in that section.

3.3.2 The Modified Exchange Iterations

The equality constrained optimization procedure described in section 2.4.1 is performed at each step of an iterative algorithm. The way in which the constraint set $S$ is updated is now described. This part of the algorithm differs from the algorithm of Chapter 2. To avoid the cycling that may occur when the algorithm of Chapter 2 is applied to multiband filter design, two constraint sets are used. The second constraint set, which we call $R$, is used to store the elements of the constraint set $S$ of the previous iteration of the algorithm.

After each iteration, the algorithm checks the values of $A(\omega)$ over the previous
constraint set frequencies. If $A(\omega)$ is within the lower and upper boundary functions $L(\omega)$ and $U(\omega)$ over these frequencies, then the algorithm proceeds exactly as does the algorithm of Chapter 2. However, if it is found that $A(\omega)$ violates the constraints at some frequency belonging to the previous constraint set, then, (i) that frequency where the violation is greatest is appended to the current constraint set $S$, and (ii) the same frequency is removed from the record of previous constraint set frequencies $R$.

The algorithm begins with an empty constraint set $S$ so that the first filter designed is the best unconstrained $L_2$ filter. Then constraints are iteratively imposed upon $A(\omega)$ at selected frequencies until the best constrained $L_2$ filter is obtained.

The algorithm can be summarized in the following steps. In this description, the set $R$ records the constraint frequencies of the previous iteration.

1. Initialization: Initialize both constraint sets to the empty set: $R = \emptyset$, $S = \emptyset$.

2. Minimization with Equality Constraints: Calculate the Lagrange multipliers associated with the filter that minimizes $\|E(\omega)\|_2$ subject to the equality constraints $A(\omega_i) = L(\omega_i)$ for $\omega_i \in S_i$ and $A(\omega_i) = U(\omega_i)$ for $\omega_i \in S_u$. (Solve equation (2.32)).

3. Kuhn-Tucker Conditions: If there is a constraint set frequency $\omega_i \in S$ for which the Lagrange multiplier $\mu_i$ is negative, then remove from the constraint set $S$ the frequency corresponding to the most negative multiplier and go back to step 2. Otherwise, calculate the new cosine coefficients using equation (2.33) and proceed to step 4.

4. Check for Violation over $R$. Calculate $a$ and $A(\omega)$. If $A(\omega_i) < L(\omega_i)$ or $A(\omega_i) > U(\omega_i)$ for some $\omega_i \in R$, then remove from $R$ the frequency corresponding to the greatest violation, append to $S$ the same frequency, and go back to step 2. Otherwise, proceed to step 5.
5. Multiple Exchange of Constraint Set: Set the constraint set $S$ equal to $S_l \cup S_u$ where $S_l$ is the set of frequency points $\omega_l$ in $[0, \pi]$ satisfying both $A'(\omega_l) = 0$ and $A(\omega_l) \leq L(\omega_l)$, and where $S_u$ is the set of frequency points $\omega_i$ in $[0, \pi]$ satisfying both $A'(\omega_i) = 0$ and $A(\omega_i) \geq U(\omega_i)$.

6. Check for Convergence: If $A(\omega) \geq L(\omega) - \varepsilon$ for all frequency points in $S_l$ and if $A(\omega) \leq U(\omega) + \varepsilon$ for all frequency points in $S_u$ then convergence has been achieved. Otherwise, go back to step 2.

According to the Kuhn-Tucker conditions, because $\mu \geq 0$ is ensured for each set of computed cosine coefficients $a$, each filter minimizes the $L_2$ error subject to the inequality constraints (2.19,2.20) over some set of frequencies. At convergence, the constraint set frequencies are exactly those extrema of $A(\omega)$ where $A(\omega)$ touches the lower and upper bound function. $\varepsilon$ in step 4 is a small number (like $10^{-6}$) indicating the numerical accuracy desired.

A flowgraph is shown in Figure 3.4. Comparing this flowgraph with the flowgraph given in Chapter 2 it can be seen that the two algorithms are very similar. The only modification of the earlier algorithm made is the test for constraint violation over the previous constraint set $R$. Note, however, that when a frequency in $R$ is appended to $S$, it is also removed from $R$. Therefore $R$ diminishes in size. This is important, because it ensures that eventually the inner loop will terminate. In some cases, many of the frequencies of $R$ are transferred to $S$, in which case the new constraint set is essentially replaced by the previous one. If the frequencies transferred from $R$ to $S$ were not deleted from $R$, then there are cases where the inner loop will not terminate.

Notice that the algorithm begins by initializing the sets $R$ and $S$ to the empty set. Therefore, on the first iteration of the algorithm, steps 2, 3, and 4 are trivial: the set of Lagrange multipliers in step 2 is the null set. The algorithm essentially begins by calculating the best unconstrained least square solution in step 5.

The Matlab program below implements this algorithm. For the sake of space and clarity, it uses a grid of frequency values. However, it is much preferable to refine the
Figure 3.4: Flowgraph for the exchange algorithm for the constrained least square design of multiband filters.

Notes:
- \( R, S \) are sets of frequencies.
- \( \mu \) is calculated using \( S \) as a constraint set.
- \( S \) is updated by a multiple exchange.
- \( \emptyset \) denotes the empty set.
- \( S/\omega \) denotes set difference.
location of the extremal frequencies by Newton's method; otherwise a rather dense grid is sometimes required for convergence. The use of Newton's method is easily incorporated.

3.4 Example

When the modified algorithm is applied to the design problem described in the example above, the filter illustrated in Figure 3.5 is obtained. The behavior of the algorithm is illustrated in Figure 3.6 which shows the filter response for first few (outer) iterations of the algorithm. The subfigures of Figure 3.6 show the frequency response amplitude at the point in the algorithm at which convergence is tested. The circular marks in the figure indicate the interpolation points in the set $S$ used to obtain the filter. Each amplitude shown in Figure 3.5 is calculated after several inner loop iterations. The algorithm converges after several more iterations.

![Figure 3.5: Modified algorithm applied to bandpass filter design.](image)

Notice that the middle local maxima on the third outer iteration shown in Figure 3.6 is flanked by two interpolation points. The simpler algorithm of Chapter 2 cannot achieve this behavior. It is the ability of the modified algorithm to arrive at this type of constraint set that allows it to solve the multiband case.
Figure 3.6: The first 4 (outer) iterations of the modified algorithm for the example.
3.5 Conclusion

This chapter shows that there exists a simple and effective multiple-exchange algorithm for the design of multiband linear phase FIR filters according to a constrained least square approach that does not use specified transition bands. The user supplies a lower and upper bound constraint that is satisfied by the local minima and maxima of the frequency response amplitude.

The algorithm of this chapter gives a design approach that is a hybrid of the Parks-McClellan algorithm [69] and the method of least squares. Accordingly, the algorithm of this chapter produces both least square filters and equiripple filters as special cases.

The approach outlined in this chapter does not exclude from the integral square measure of approximation error any region around the cut-off frequencies, and so it does not implicitly assume that input signals have no energy in those regions. At the same time, the algorithm does not preclude the use of "don't care" regions if their use is desired — appropriate \( L_2 \) weighting functions can be easily incorporated if desired.

Appendix

3.6 A Matlab Program

The Matlab program c12bp below implements the algorithm described in this chapter. This program designs bandpass filters, but can be used to design filters having more passbands and stopbands by making appropriate changes to the input arguments and to the initialization of \( u \), \( l \), and \( c \).

The program c12bp is very similar to the program c121p given in Chapter 2. The variables used are the same, and the structure of the two programs are very similar. The program c121p can be obtained from c12bp by basically deleting some lines of c12bp.

The program uses a grid to store the frequency response amplitude \( A \), the fre-
quency points w, and the lower and upper bound functions l and u. The variables kmax and kmin store the indices of the local maxima and minima of the amplitude, so that w(kmax) are local maxima, etc. kmax and kmin describe the set S in the algorithm description above. Similarly, the variables okmax and okmin represent the local extremal points of the amplitude on the previous iteration (the set R in the algorithm description above).

The variables lvo and uvo measure the constraint violation over the set okmin and okmax respectively: lvo measures the violation of the lower bound constraint, while uvo measures the violation of the upper bound constraint. When the variables lvo and uvo are negative, then the constraints are not violated over the points in okmin and okmax. Conversely, when the variable lvo is positive, then the lower bound constraint is violated by some point in kmin, and similarly for uvo.

function h = cl2bp(m,w1,w2,up,lo,L)
% h = cl2bp(m,w1,w2,up,lo,L)
% Constrained L2 band pass FIR filter design
% (Odd length only)
% Author: Ivan Selesnick, Rice University, 1995
% m : degree of cosine polynomial
% w1,w2 : fist and second band edges
% up : vector of upper bounds
% lo : vector of lower bounds
% L : grid size
% output:
% h : 2*m+1 filter coefficients
% example
% w1 = 0.3*pi;
% w2 = 0.6*pi;
% up = [0.02, 1.02, 0.02];
% lo = [-0.02, 0.98, -0.02];
% h = cl2bp(30,w1,w2,up,lo,2^11);

% ---- calculate Fourier coefficients and upper ----
% ---- and lower bound functions ---------------

q1 = round(L*w1/pi);
q2 = q2 = round(L*(w2-w1)/pi);
q3 = L + 1 - q1 - q2;
u = [u(1)*ones(q1,1); u(2)*ones(q2,1); u(3)*ones(q3,1)];
l = [l(1)*ones(q1,1); l(2)*ones(q2,1); l(3)*ones(q3,1)];
w = [0:L]*pi/L;
Z = zeros(2*L-1-2*m,1);
\( r = \sqrt{2};\)
c = [(w2-w1)*r; 2*[(sin(w2*[1:m])-sin(w1*[1:m]))./(1:m)].']/pi;
a = c;  % best L2 cosine coefficients
mu = [];  % Lagrange multipliers
SN = 1e-9;  % Small Number
okmax = []; uvo = 0;
okmin = []; lvo = 0;

while i
    if (uvo < -10*SN) || (lvo < -10*SN)
        % ----- include old extremal ---------------
        if uvo < lvo
            kmax = [kmax; okmax(k1)]; okmax(k1) = [];
        else
            kmin = [kmin; okmin(k2)]; okmin(k2) = [];
        end
    else
        % ----- calculate A ---------------
        A = fft([a(1)*r; a(2:m+1); Z; a(m+1:-1:2)]);
        A = real(A(1:L+1))/2;
        % ----- find extremals ---------------
        okmax = kmax;
        okmin = kmin;
        kmax = local_max(A); kmin = local_max(-A);
        kmax = kmax( A(kmax) > u(kmax)-SN );
        kmin = kmin( A(kmin) < l(kmin)+SN );
        % ----- check stopping criterion -------
        Eup = A(kmax)-u(kmax); Elo = l(kmin)-A(kmin);
        E = max([Eup; Elo; 0])
        if E < SN, break, end
    end

    % ----- calculate new multipliers -------
    n1 = length(kmax); n2 = length(kmin);
    D = [ones(n1,m+1); -ones(n2,m+1)];
    G = D.*cos(w([kmax;kmin])*[0:m]);
    G(:,1) = G(:,1)/r;
d = [u(kmax); -1(kmin)];
mu = (G*G')/(G*c-d);
% ----- remove negative multiplier ---------
[min_mu,K] = min(mu);
while min_mu < 0
    G(K,:) = []; d(K) = [];
    mu = (G*G')/(G*c-d);
    if K > n1
        kmin(K-n1) = []; n2 = n2 - 1;
    else
        kmax(K) = []; n1 = n1 - 1;
    end
    [min_mu,K] = min(mu);
end
% ----- determine new coefficients ----------
a = c-G'*mu;

% ----- calculate constraint violation ------
Aoxmax = a(1)/r + cos(w(okmax)*[1:m])*a(2:m+1);
Aokmin = a(1)/r + cos(w(okmin)*[1:m])*a(2:m+1);
[uvo,k1] = min(u(okmax) - Aokmax);
[lvo,k2] = min(Aokmin - l(okmin));
end

h = [a(m+1:-1:2); a(1)*r; a(2:m+1)]/2;
Chapter 4

Complementing the Parks-McClellan Algorithm

4.1 Introduction

The Chebyshev norm is widely used for the design of linear-phase FIR filters for two reasons. (i) For certain applications it is a meaningful error criterion as shown by Weisburn, Parks and Shenoy [124] and (ii) there is an excellent program, the Parks-McClellan (PM) program [63, 69, 78], that designs optimal filters according to this norm. Recall that in this approach to the design of digital filters, the passband and stopband edges are specified and the weighted Chebyshev error over both bands is minimized. In this chapter we revisit equiripple filter design and describe a complement to the PM program. In the program we propose, the weighted Chebyshev error is specified and the width of the transition region is variable.

For example, consider the usual ideal lowpass filter. Figure 4.1 shows a typical equiripple frequency response. The passband and stopband edges are denoted by $\omega_p$ and $\omega_s$. The Chebyshev errors in the passband and stopband are denoted by $\delta_p$ and $\delta_s$. The figure also shows the half-magnitude frequency, $\omega_0$. The filter in Figure 4.1 was designed to possess a specified half-magnitude frequency of $\omega_0 = 0.4\pi$ with specified Chebyshev errors of $\delta_p = \delta_s = 0.05$. The passband and stopband edges are "induced" by these specifications.

While the PM program allows the filter designer to specify $\omega_p$, $\omega_s$ and the ratio $\delta_p/\delta_s$, it does not allow the user to specify both $\delta_p$ and $\delta_s$ directly. Although the user can indirectly control $\delta_p$ and $\delta_s$ with the PM program by iteratively adjusting the band edges and the ratio $\delta_p/\delta_s$ [70, 75], the exchange algorithm below obtains the specified $\delta_p$ and $\delta_s$ directly.
Figure 4.1: $N = 21, \delta_p = \delta_s = 0.05, \omega_o = 0.4\pi, \omega_p = 0.3418\pi, \omega_s = 0.4580\pi$

The algorithm described below is similar to the algorithm of Hofstetter, Oppenheim and Siegel [41,42,76]. Although their algorithm predates the PM program and produces equiripple filters with specified $\delta_p$ and $\delta_s$, it is not widely used because it permits only limited control over the location of the band edges and produces only extra-ripple filters. Moreover, the location of the band edges can be controlled only indirectly in the Hofstetter algorithm: only the number of extremal frequencies in each band can be specified. Their algorithm is interesting, however, because extra-ripple filters have a minimum transition width property [76, p143] [42,78]. More specifically, for a fixed filter length and fixed $\delta_p$ and $\delta_s$, extra-ripple filters locally minimize the transition band width as a function of, say, the passband edge. The design formulation leading to these extra-ripple filters was originally described by Herrmann and Schüßler [37,40,93] and is also described in [76,78]. The algorithm of Hofstetter et al. can also be used for the design of maximum-ripple multiband filters having a specified ripple size in each band.

In [103] Shpak and Antoniou present an interesting modification to the PM program in which the Chebyshev errors in respective bands are not constrained by a ratio. In this way they are able to obtain extra-ripple filters without sacrificing the
ability to specify the band edges. This technique can be used to prevent transition region anomalies that sometimes occur in the design of optimal Chebyshev multiband filters [74]. By maintaining the ability to specify band edges in the design of extra-ripple filters, however, the Shpak-Antoniou algorithm provides less control of the weighted Chebyshev error. Clearly, for a fixed filter length, there is a tradeoff between the ability to specify the weighted Chebyshev error and the ability to specify the band edges. Another multiple exchange algorithm which addresses a similar problem is described in [62].

Another approach to the filter design problem uses linear programming [35, 107]. Steiglitz, Parks and Kaiser have presented a very general and flexible program [107] that meets Chebyshev and other constraints. In this chapter, we discuss multiple exchange Remez-type algorithms [83], as these are more efficient than linear programming methods.

The algorithms described below produce lowpass and bandpass linear-phase FIR filters having a specified Chebyshev error in each band and a single transition region frequency, such as the half-magnitude frequency, to control the location of the transition region. They are hybrids of the algorithm of Hofstetter, Oppenheim and Siegel [41, 42] and the PM algorithm. Like those algorithms, it employs a reference set of frequencies. On each iteration (i) an interpolation problem is solved and (ii) the reference set is updated. The efficient computational techniques used for the PM program [3, 5, 6, 9, 24, 103] can also be used for the algorithms below.

4.2 Equiripple Filter Design

The frequency response of a linear-phase FIR filter is given by the discrete-time Fourier transform of its impulse response and can be written as $H(\omega) = A(\omega)e^{-jM\omega}$ where $A(\omega)$ is a real valued periodic function of $\omega$ called the amplitude and $M = (N - 1)/2$ for length-$N$ filters [68, 94].

We first describe a version of the standard PM program for lowpass filter design
in which either $\delta_p$ or $\delta_s$ is specified and the other is minimized. We then describe the exchange algorithm for lowpass and bandpass filter design in which the Chebyshev error in each band is specified.

### 4.2.1 The PM Program with an Affine Relation between $\delta_p$ and $\delta_s$

The usual PM program can be modified so that it achieves a specified Chebyshev error in one band and minimizes the Chebyshev error in the other. This can be achieved by imposing an affine relationship between $\delta_p$ and $\delta_s$, where $\delta_p$ and $\delta_s$ denote the Chebyshev errors of the realized frequency response amplitude. Recall that by the standard PM algorithm for lowpass filter design, the user specifies a linear relationship between $\delta_p$ and $\delta_s$: the user specifies $N$, $\omega_p$, $\omega_s$ and $K$, and obtains a filter satisfying $\delta_p = K \delta_s$. However, the PM algorithm can be modified as described here so that the user specifies $N$, $\omega_p$, $\omega_s$, $K_p$, $K_s$, $\eta_p$ and $\eta_s$, and obtains a filter satisfying the following affine relationship between $\delta_p$ and $\delta_s$:

$$
\delta_p = K_p \delta + \eta_p 
$$

(4.1)

$$
\delta_s = K_s \delta + \eta_s. 
$$

(4.2)

(So that the problem is well posed, the parameters $K_p$, $K_s$, $\eta_p$ and $\eta_s$ supplied by the user must all be nonnegative and satisfy the inequalities: $K_p + \eta_p > 0$, $K_s + \eta_s > 0$ and $K_p + K_s > 0$.) The modified PM program minimizes $\delta$. When $\eta_p$ and $\eta_s$ are both taken to be 0 this becomes the usual linear relationship permitted by the PM program. However, if $K_s = \eta_s = 0$, then the stopband ripple size $\delta_s$ of the resulting equiripple filter has the specified value $\eta_s$ and the passband ripple size $\delta_p$ is minimized.

The first modification that needs to be made to the usual PM program is the interpolation step. Given a reference set of $M + 2$ frequencies that includes $\omega_p$ and $\omega_s$, let $\omega_1, \ldots, \omega_q = \omega_p$ denote those in the passband and let $\omega_q = \omega_{q+1}, \ldots, \omega_{M+2}$ denote those in the stopband, listed, in each case, in ascending order. The linear
system of equations to be solved on each iteration is given by (4.1), (4.2), and by

\[ A(\omega_i) = 1 + (-1)^{i+c} \delta_p \quad \text{for} \quad 1 \leq i \leq q \]
\[ A(\omega_i) = (-1)^{i+c} \delta_s \quad \text{for} \quad q + 1 \leq i \leq M + 2 \]  

(4.3)

where \( c \) is chosen to equal 0 or 1, whichever yields the equation \( A(\omega_p) = 1 - \delta_p \).

The system of equations given by (4.1, 4.2, 4.3) is linear in \( \delta, \delta_p, \delta_s \) and the filter coefficients. As in the PM algorithm, this system can be solved efficiently using interpolation formulas.

**Example 1:** Consider the design of a length-7 filter with \( \omega_p = 0.2\pi \) and \( \omega_s = 0.5\pi \).

Here \( M = 3 \) and \( A(\omega) \) can be written as \( A(\omega) = \sum_{k=0}^{3} a_k \cos kw \). Figure 4.2 shows a typical frequency response amplitude during the course of the algorithm. In this case, Equations (4.1, 4.2, 4.3) can be written as

\[
\begin{bmatrix}
1 & \cos \omega_1 & \cos 2\omega_1 & \cos 3\omega_1 & -1 & 0 & 0 \\
1 & \cos \omega_2 & \cos 2\omega_2 & \cos 3\omega_2 & 1 & 0 & 0 \\
1 & \cos \omega_3 & \cos 2\omega_3 & \cos 3\omega_3 & 0 & -1 & 0 \\
1 & \cos \omega_4 & \cos 2\omega_4 & \cos 3\omega_4 & 0 & 1 & 0 \\
1 & \cos \omega_5 & \cos 2\omega_5 & \cos 3\omega_5 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -K_p \\
0 & 0 & 0 & 0 & 0 & 1 & -K_s
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
\delta_p \\
\delta_s
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
a_1 \\
a_2 \\
a_3 \\
\delta_p \\
\delta_s
\end{bmatrix}
\]  

(4.4)

where \( \omega_1 = 0, \omega_p = \omega_2 = 0.2\pi, \omega_s = \omega_3 = 0.5\pi, \omega_4 = 0.75\pi, \omega_5 = \pi \) are indicated by circular marks in Figure 4.2. Comparing system (4.4) to the corresponding system for the usual PM algorithm, we note that there are two additional variables and two additional equations.

It is important to note that either \( \delta_p \) or \( \delta_s \) given by the solution to (4.1, 4.2, 4.3) may be negative. This generally occurs, if at all, during the early iterations of the algorithm. When it does occur, however, it is important to modify the interpolation step in a simple way to ensure convergence. If \( \delta_s \) given by Equations (4.1, 4.2, 4.3) is negative on some iteration, then it is necessary to repeat the interpolation step for
Figure 4.2 : Updating the reference set. \( N = 7, \omega_p = 0.2\pi, \omega_s = 0.5\pi \)

that iteration using different interpolation equations. The equations in this case are given by

\[
A(\omega_i) = 1 + (-1)^{i+c}\delta_p \quad \text{for} \ 1 \leq i \leq q \\
A(\omega_i) = 0 \quad \text{for} \ q + 1 \leq i \leq M + 2
\]

\( \delta_s = 0. \) \hfill (4.5)

In the length-7 example above, these equations can be written as

\[
\begin{bmatrix}
1 & \cos \omega_1 & \cos 2\omega_1 & \cos 3\omega_1 & -1 \\
1 & \cos \omega_2 & \cos 2\omega_2 & \cos 3\omega_2 & 1 \\
1 & \cos \omega_3 & \cos 2\omega_3 & \cos 3\omega_3 & 0 \\
1 & \cos \omega_4 & \cos 2\omega_4 & \cos 3\omega_4 & 0 \\
1 & \cos \omega_5 & \cos 2\omega_5 & \cos 3\omega_5 & 0 \\
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
\delta_p \\
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
1 \\
0 \\
0 \\
0 \\
\end{bmatrix}.
\]

(4.6)

Similarly, if \( \delta_p < 0 \) on some iteration, then the interpolation step for that iteration must be repeated with the equations

\[
A(\omega_i) = 1 \quad \text{for} \ 1 \leq i \leq q \\
A(\omega_i) = (-1)^{i+c}\delta_s \quad \text{for} \ q + 1 \leq i \leq M + 2
\]

\( \delta_p = 0. \) \hfill (4.7)
On the following iteration, after the reference set is updated, the Equations (4.1, 4.2, 4.3) are again used. We also note that $\delta_p$ and $\delta_s$ given by the solution to eqs (4.1, 4.2, 4.3) can not both be negative, because if they were, then $A(\omega)$ would have more extrema than is possible for a degree $M$ cosine polynomial.

The procedure to update the reference set from one iteration to the next is the multiple exchange of the PM algorithm: Let $S$ be the set obtained by appending $\omega_p$ and $\omega_s$ to the set of extrema of $A(\omega)$ in $[0, \pi]$. $S$ will have either $M + 2$ or $M + 3$ frequencies and will include both 0 and $\pi$. If $S$ has $M + 2$ frequencies, then take the new reference set to be $S$. If $S$ has $M + 3$ frequencies, then remove either 0 or $\pi$ from $S$ according to the following rule: If $\omega = 0$ is a local maximum of $A(\omega)$ then let $\alpha = 1$, otherwise set $\alpha = -1$. If $\omega = \pi$ is a local maximum of $A(\omega)$ then let $\beta = 1$, otherwise set $\beta = -1$. If

$$(A(0) - 1)\alpha - \delta_p < A(\pi)\beta - \delta_s$$  (4.8)

then remove 0 from $S$, otherwise remove $\pi$ from $S$, and take the new reference set to be the resulting set. The expressions on each side of the inequality indicate the amount by which the error exceeds its intended value. $\alpha$ and $\beta$ must be chosen appropriately because both the magnitude and the sign of this value is important: negative values appear in the design of filters possessing a scaled extra-ripple. The rule states that of 0 and $\pi$, the frequency to be retained in $S$ is the one at which the error exceeds its intended value the most. The reference set, in the example, is updated by updating only $\omega_s$. Its new value is indicated by the x mark in Figure 4.2.

A flowchart for this algorithm is shown if Figure 4.3. This modification of the PM algorithm is easily incorporated and permits (i) the specification of $\omega_p$ and $\omega_s$ and (ii) the affine constraint on $\delta_p$ and $\delta_s$. It is useful because by taking $K_s = \eta_p = 0$, or $K_p = \eta_s = 0$, the filter obtained by this algorithm achieves a specified Chebyshev error in one band and minimizes the Chebyshev error in the other. The algorithm works for other choices of $K_p, K_s, \eta_p, \eta_s$, but the meaning of arbitrary values for these parameters is unclear. The general affine constraint is a convenient way to solve the
Figure 4.3: Flowchart for the PM algorithm for lowpass filter design modified to include an affine constraint between $\delta_p$ and $\delta_s$.

Problem, without requiring special cases.

Example 2: This example illustrates the situation in which $\delta_p$ is negative on some iteration. The parameters are chosen to be $N = 19$, $\omega_p = 0.4\pi$, $\omega_s = 0.5\pi$, $K_p = 1$, $K_s = 0$, $\eta_p = 0$ and $\eta_s = 0.05$. In this case $M = 9$. When the reference set is initialized to be 4 equally spaced frequencies in the passband and 7 equally spaced frequencies in the stopband, the solution to eqs (4.1, 4.2, 4.3) gives $\delta_p = -1.9619$, $\delta_s = 0.05$ and the amplitude $A(\omega)$ shown in Figure 4.4(a). Because $\delta_p < 0$, the algorithm we describe requires that the iteration be repeated using (4.7) and the same reference set. The solution to (4.7) gives $\delta_p = 0$, $\delta_s = 0.0096$ and the amplitude shown in Figure 4.4(b). The reference set is then updated and on the next interpolation step,
\( \delta_p \) is positive, and remains positive for the duration of the algorithm. The amplitudes associated with this and the next iteration are shown in Figure 4.4(c) and 4.4(d). The algorithm converges in only a few more iterations.

4.2.2 A New Equiripple Lowpass Filter Design Algorithm: Specified \( \delta_p \), \( \delta_s \) and \( \omega_o \)

In order to exactly achieve specified values for \( \delta_p \) and \( \delta_s \) with the PM program, it is necessary to iteratively adjust the parameters \( \omega_p \), \( \omega_s \), and the ratio \( \delta_p/\delta_s \). We propose an algorithm for the design of equiripple lowpass filters that allows (i) the explicit specification of \( \delta_p \) and \( \delta_s \) and (ii) the specification of the half-magnitude frequency. The band edges \( \omega_p \) and \( \omega_s \) can not be explicitly specified — they are "induced" by the specified values of \( \delta_p \), \( \delta_s \) and \( \omega_o \). As above, on each iteration, the filter interpolating the appropriate values over the reference set is computed and the reference set is updated.

As in the PM program, extremal frequencies may migrate from one band to another during the course of the algorithm. Therefore the initialization of the reference set is not critical for convergence, but does affect the total number of iterations for convergence. The reference set here does not contain two band edges as in the PM program, instead, it contains the half-magnitude frequency, \( \omega_o \). Therefore, the reference set contains \( M + 1 \) frequencies, not \( M + 2 \) as in the PM program. The circular marks in Figure 4.1 indicate the reference set frequencies upon convergence for a length 21 filter. The resulting filter satisfies the alternation property for the correct choice of band edges, so it could have been designed using the PM program if the band edges had been known in advance.

The algorithm proceeds by computing the filter that alternately interpolates \( 1 + \delta_p \), \( 1 - \delta_p \) over the reference set frequencies in the passband, alternately interpolates \( \delta_s \), \( -\delta_s \) over the reference set frequencies in the stopband, and interpolates 0.5 at \( \omega_o \). As in the PM program, interpolation formulas can be used to find the filter efficiently.
Figure 4.4: \( N = 19, \omega_p = 0.4\pi, \omega_s = 0.5\pi \) (a) iteration 1 (b) iteration 1b (c) iteration 2 (d) iteration 3.
Note that, because \( \delta_p \) and \( \delta_s \) have been explicitly specified, \( \delta \) does not have to be computed at each iteration.

Suppose \( \omega_1, \ldots, \omega_{M+1} \), listed in increasing order, is a reference set of \( M + 1 \) frequencies in \([0, \pi]\) that includes \( \omega_o \). Let \( \omega_1, \ldots, \omega_{q-1} \) denote those in the passband (to the left of \( \omega_o \)) and let \( \omega_{q+1}, \ldots, \omega_{M+1} \) denote those in the stopband (to the right of \( \omega_o \)). The linear system of equations to be solved on each iteration is given by

\[
A(\omega_i) = 1 + (-1)^{i+c} \delta_p \quad \text{for } 1 \leq i \leq q - 1
\]
\[
A(\omega_o) = 0.5
\]
\[
A(\omega_i) = (-1)^{i+c+1} \delta_s \quad \text{for } q + 1 \leq i \leq M + 1
\]

where \( c \) is chosen to equal 0 or 1, whichever yields the equation \( A(\omega_{q-1}) = 1 + \delta_p \).

**Example 3:** Consider the design of a length 9 filter with \( \omega_o = 0.4\pi \), \( \delta_p = \delta_s = 0.1 \). Figure 4.5 shows a typical amplitude response before convergence is attained. In this case, the equations (4.9) can be written as

\[
\begin{bmatrix}
1 & \cos \omega_1 & \cos 2\omega_1 & \cos 3\omega_1 & \cos 4\omega_1 \\
1 & \cos \omega_2 & \cos 2\omega_2 & \cos 3\omega_2 & \cos 4\omega_2 \\
1 & \cos \omega_3 & \cos 2\omega_3 & \cos 3\omega_3 & \cos 4\omega_3 \\
1 & \cos \omega_4 & \cos 2\omega_4 & \cos 3\omega_4 & \cos 4\omega_4 \\
1 & \cos \omega_5 & \cos 2\omega_5 & \cos 3\omega_5 & \cos 4\omega_5 \\
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
a_4 \\
\end{bmatrix}
=
\begin{bmatrix}
1 - \delta_p \\
1 + \delta_p \\
0.5 \\
-\delta_s \\
\delta_s \\
\end{bmatrix}
\]

where \( \omega_1 = 0, \omega_2 = 0.1\pi, \omega_o = \omega_3 = 0.4\pi, \omega_4 = 0.7\pi, \omega_5 = \pi \) are indicated by circular marks in Figure 4.5.

The procedure to update the reference set from one iteration to the next is similar to the multiple exchange of the PM algorithm: Let \( S \) be the set obtained by appending \( \omega_o \) to the set of extrema of \( A(\omega) \) in \([0, \pi]\). \( S \) will have either \( M + 1 \) or \( M + 2 \) frequencies and will include both 0 and \( \pi \). If \( S \) has \( M + 1 \) frequencies, then take the new reference set to be \( S \). If \( S \) has \( M + 2 \) frequencies, then remove either 0 or \( \pi \) from \( S \) according to the following rules:

1. If \( A(\omega) \) has no extrema in the open interval \((0, \omega_o)\), then remove 0 from \( S \).
Figure 4.5: Updating the reference set for modified PM algorithm. \( N = 9, \omega_o = 0.4\pi, \delta_p = \delta_s = 0.1 \)

2. If \( A(\omega) \) has no extrema in the open interval \((\omega_o, \pi)\), then remove \( \pi \) from \( S \).

3. Otherwise, let \( \omega_2 \) be the extrema of \( A(\omega) \) in \((0, \omega_o)\) closest to 0, and let \( \omega_5 \) be the extrema of \( A(\omega) \) in \((\omega, \pi)\) closest to \( \pi \). If

\[
\delta_s |A(0) - A(\omega_2)| < \delta_p |A(\pi) - A(\omega_5)|
\]

(4.11)

then remove 0 from \( S \), otherwise remove \( \pi \) from \( S \).

Take the new reference set to be the resulting set \( S \). The reference set, in Figure 4.5 for example, is updated by updating \( \omega_2, \omega_4 \), and \( \omega_5 \). Their new locations are indicated by the x marks in Figure 4.5. It should be noted that case(1) and case(2) only occur when \( \omega_o \) is taken to be near 0 or \( \pi \) relative to the filter length (in these cases, the reference set upon convergence contains no frequencies in one of the bands).

Note that any transition region frequency can be fixed. Instead of the half-magnitude frequency, the half-power frequency, the passband edge, or stopband edge can be fixed by respectively imposing \( A(\omega_o) = 1/\sqrt{2}, A(\omega_o) = 1 - \delta_p, \) or \( A(\omega_o) = \delta_s \). It should also be noted that if the half-magnitude is taken to be too close to either 0 or \( \pi \) relative to the filter length, then there will exist no filter with the specified Chebyshev error in each band. Either the passband or stopband will be too narrow.
When the specified ripple sizes are achievable, this algorithm produces exactly the same lowpass filters as does the PM program, however, it allows one to specify a different set of parameters in the design process.

4.2.3 Bandpass Filter Design

The design of multiband filters achieving a specified Chebyshev error with specified half-magnitude frequencies requires more care than the design of lowpass filters with this approach. There are three reasons for this. (i) There is generally more than one equiripple filter satisfying these constraints. (ii) The procedure for updating the reference set of frequencies is less obvious because the optimal filter may have scaled extra ripples at frequencies other than 0 and π. (iii) The transition region of a multiband filter designed by the PM program may contain large undesirable peaks [74]. Despite these aspects of the multiband case, the algorithm below remains simple, robust, and rapid as long as the specified Chebyshev error is not taken too small relative to the filter length.

We consider the design of bandpass filters and denote the Chebyshev errors of the first stopband, the passband, and the second stopband by \( \delta_1 \), \( \delta_2 \), and \( \delta_3 \) respectively. The half-magnitude frequencies are denoted by \( \omega_a \) and \( \omega_b \). When the band edges are not explicitly specified, the non-uniqueness of the bandpass filter achieving a specified Chebyshev error with specified half-magnitude frequencies is easily ascertained. The specifications can be summarized by 5 values: \( \delta_1 \), \( \delta_2 \), \( \delta_3 \), \( \omega_a \) and \( \omega_b \). However, the PM algorithm requires 6 values for bandpass filters: 4 band edges and 2 ratios, \( \delta_2/\delta_1 \) and \( \delta_3/\delta_1 \). Therefore, in a complement to the PM program, we require an additional constraint.

We have chosen to require that the derivative of \( A(\omega) \) at the half-magnitude frequencies be equal in magnitude and opposite in sign. We chose this constraint because (i) in a sense, it weights the widths of the transition regions equally, because the width of the transition region is related to the slope of \( A(\omega) \) at the correspond-
ing half-magnitude frequency, and, (ii) it can be easily incorporated into a simple exchange algorithm.

The algorithm for producing equiripple bandpass filters is similar to the algorithm above for lowpass design. Excluding the two half-magnitude frequencies, the reference set for the bandpass case contains \( M - 2 \) frequencies, and is updated by locating the local extrema of the new frequency response amplitude \( A(\omega) \). The interpolation step consists of finding the filter that alternately interpolates \( 1 + \delta_i \), \( 1 - \delta_i \) over the reference set frequencies in band \( i \), interpolates 0.5 at \( \omega_a \) and at \( \omega_b \), and for which \( A'(\omega_a) = -A'(\omega_b) \). This filter can be found by solving a system of linear equations or by modifying the usual interpolation formulas.

It follows from the interpolation step that there will be at least \( M - 2 \) local extrema of \( A(\omega) \), however, there may be as many as \( M + 1 \). Because the reference set must contain \( M - 2 \) extremal frequencies, it will therefore be necessary to exclude 0, 1, 2 or 3 local minima and maxima when updating the reference set. The rule we use for updating the reference set is most easily stated by describing which local extrema are not included. Suppose \( \omega_1, \ldots, \omega_L \) are the local extrema of \( A(\omega) \) listed in order.

1. To exclude 1 local extremum \((L = M - 1)\), use the same update rule used for the lowpass case.

2. To exclude 2 local extrema \((L = M)\), find the index \( i \) that minimizes

\[
(E(\omega_i) - E(\omega_{i+1})) (-1)^{i+s}
\]

where \( s = 1 \) if \( \omega_i \) is a local maxima and \( s = 0 \) if \( \omega_i \) is a local minima. \( E(\omega) \) denotes the error function: \( E(\omega) = (A(\omega) - D(\omega))/\delta(\omega) \). If \( 1 < i < M - 2 \), then exclude \( \omega_i \) and \( \omega_{i+1} \) from the reference set. If \( i = 1 \) or \( i = L \), then exclude \( \omega_i \) and use the procedure above for excluding 1 local extremum.

3. To exclude 3 local extrema \((L = M + 1)\), use the procedure for excluding 1 extremum, followed by the procedure for excluding 2 extrema.
By following this simple procedure for updating the reference set the algorithm rapidly converges and, like the PM algorithm, is capable of producing equiripple filters with extra ripples at frequencies other than 0 and $\pi$.

Example 4: Figure 4.6 shows a bandpass frequency response with three scaled extra ripples. The reference set frequencies upon the convergence of the algorithm are indicated with circular marks. Again, although this filter was not obtained by the PM program, it could have been if the resulting band edges were known in advance. That is, the filter in Figure 4.6 is an optimal Chebyshev filter for the correct choice of band edges.

![Frequency Response](image)

Figure 4.6: An optimal Chebyshev bandpass filter. $N = 55$, $\delta_1 = \delta_2 = \delta_3 = 0.05$. The specified half-magnitude frequencies were $\omega_a = 0.1675\pi$, $\omega_b = 0.501\pi$. The "induced" band edges are $0.1480\pi$, $0.1870\pi$, $0.4815\pi$, and $0.5203\pi$.

For some specifications, the filters produced by this algorithm for bandpass filter design are not optimal Chebyshev filters for any choice of band edges. Specifically, this algorithm can produce filters possessing a pair of adjacent scaled extra ripples that straddle a half-magnitude frequency. The filter in Figure 4.7, for example, was obtained with this algorithm. Although it is not clear from the figure, the error at the local extremals neighboring the half-magnitude frequency $0.204\pi$ are below 0.05 in magnitude. Although the alternation property is satisfied on the extremal
reference set frequencies, it is not an optimal Chebyshev filter for any choice of band edges because the induced band edges can not be included in the reference set without destroying the alternation property. Nevertheless, this filter does achieve the specified Chebyshev error and has narrow transition bands of approximately equal width.

![Frequency Response Diagram]

Figure 4.7: A bandpass filter produced by the new algorithm. \( N = 55, \delta_1 = \delta_2 = \delta_3 = 0.05 \). The specified half-magnitude frequencies were \( \omega_a = 0.204\pi, \omega_b = 0.54\pi \). The “induced” band edges are \( 0.1844\pi, 0.2235\pi, 0.5205\pi, \) and \( 0.5595\pi \).

When the specified Chebyshev error is taken to be very small relative to the filter length this algorithm may occasionally produce filters with undesirable transition region behavior or may fail to converge. This is due to the necessarily wide transition regions associated with very small ripple sizes. When the transition regions are wide, the half-magnitude and derivative constraints become inappropriate since they no longer accurately reflect the behavior of the frequency response throughout the transition region. However, it should be noted that optimal Chebyshev multiband filters having very wide transition regions may also possess undesirable transition region behavior.

Indeed, the behavior of the frequency response of optimal Chebyshev multiband filters can be quite different than that of two-band filters. In [74] Rabiner, Kaiser and Schafer give three strategies for avoiding nonmonotonic transition region behavior:
(i) modify the stopband edge frequencies, (ii) modify the error weighting function, and (iii) design maximal ripple filters only. Shpak and Antoniou [103] address the occurrence of transition region ripples by employing extra \( \delta \) variables. By doing this they are able to obtain extra-ripple filters and can avoid some of the undesirable behaviors of multiband equiripple filters while maintaining specified band edges. The method described in this chapter, however, takes a different direction. Instead of introducing extra \( \delta \) variables, we give up the explicit control over the band edges, employ half-magnitude frequencies, and explicitly control the Chebyshev error in each band.

It should also be noted that all the exchange algorithms discussed in this chapter can be adopted for the design of minimum phase FIR filters. Grenet describes a simple modification of the PM program for constrained Chebyshev approximation that can be used to design linear-phase filters with nonnegative frequency response amplitudes [30]. If in each iteration of the exchange algorithm, the stopband interpolation condition \( A(\omega_i) = -\delta \) is replaced by \( A(\omega_i) = 0 \), then the resulting frequency response amplitude will be nonnegative. The FIR filter can then be spectrally factored to obtain a minimum phase filter. This technique is especially useful when the stopband ripple sizes of a multiband filter are unequal. For multiband filters for which the stopband ripple sizes are equal and for two-band filters, the classical technique of raising the amplitude and spectrally factoring the filter can be employed [39].

4.3 Conclusion

Optimal Chebyshev linear-phase FIR filters are usually found by fixing the filter length \( N \) and the band edges and by minimizing the weighted Chebyshev error. Another approach is to fix \( N \), \( \delta_p \), \( \delta_s \) and a single transition region frequency and to adjust the transition width. The same approach can be applied to the design of bandpass filters that achieve a specified Chebyshev error in each band and have transition regions of approximately equal width.
Table 4.1 classifies four approaches to the design of equiripple filters. The approaches under "Nonextra-ripple" produce filters that may or may not possess extra-ripples, depending on the specifications. The approaches under "Extra-ripple" are able to produce filters that are constrained to possess extra-ripples. (Recall, however, that the Shpak-Antoniou algorithm is a generalization of the PM algorithm and a variable number of extra ripples can be specified.) This table clarifies the relationship among previously reported exchange algorithms for equiripple linear-phase filter design and the way in which the algorithms described in sections 4.2.2 and 4.2.3 of this chapter relate to them.

It should be noted that if the spectrums of the signal and noise do not overlap and are confined to well defined bands with distinct edges, then the Parks-McClellan program is most suitable. If, however, the signal and noise are not separated and even overlap in frequency, then there are no separate passband and stopband edges. In this case it is suitable to choose a single cut-off frequency and to pass input signal energy below it and to reject all above it. The new method is well suited to this case because it gives an "induced" transition band and will meet the specified ripple sizes (see also Chapter 2 and [99]).

<table>
<thead>
<tr>
<th></th>
<th>Nonextra-ripple</th>
<th>Extra-ripple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Band edges $\omega_p, \omega_s$ specified</td>
<td>PM [63,69]</td>
<td>SA [103]</td>
</tr>
<tr>
<td>Weighted Chebyshev error $\delta_p, \delta_s$ specified</td>
<td>New</td>
<td>HS [37,40], HOS [41]</td>
</tr>
</tbody>
</table>
Chapter 5

Flat Monotonic Passbands and Equiripple Stopbands

5.1 Introduction

Linear phase FIR filters having very flat passbands and equiripple stopbands are important for several applications [122]. For example, consider the removal of high frequency noise from a low frequency signal by lowpass filtering: To reduce the distortion of the signal introduced by the filter, the use of a filter having a very flat passband is desirable. To maximize the stopband attenuation, the use of a filter having an equiripple stopband is desirable. Filters having very flat passbands are also useful in applications in which a filter appears in cascade with other filters, such as in a long-distance communication channel with "repeater stations" [32]. Furthermore, in [105] Steffen shows that such filters have good approximation properties. Briefly, filters that achieve a specified degree of flatness at $\omega = 0$ preserve the moments of an input signal to a specified degree, see also [92].

Figure 5.1 shows the frequency response amplitude of a length 33 filter having these characteristics. It equals 1 at $\omega = 0$ and has 21 derivatives equal to 0 at $\omega = 0$. The stopband edge, $\omega_s$, and the Chebyshev error in the stopband, $\delta_s$, are shown in the figure. The parameters used in this chapter are:

- $N$: filter length
- $L$: degree of flatness at $\omega = 0$
- $\omega_s$: stopband edge
- $\delta_s$: stopband Chebyshev error.
Figure 5.1: A filter having a flat monotonic passband and an equiripple stopband. $N = 33$, $L = 22$, $\omega_s = 0.6\pi$, $\delta_s = 0.0175$

The design of linear phase FIR filters having very flat passbands and equiripple stopbands has been studied by several authors. Darlington [21] described some transformation principles for filters of this type. Kaiser, Steiglitz and Parks have used linear programming methods [48, 106, 107]. While linear programming is a very general and flexible design method for filter design, it is more computationally intensive and is no more immune to numerical difficulties than are exchange algorithms. In [122] Vaidyanathan presents a method based upon the Remez exchange algorithm. The method he describes employs the Parks-McClellan algorithm [68] and a special filter structure. This structure, which has also been used by Schüßler and Steffen [91, 92, 105], enforces a specified degree of flatness at $\omega = 0$ and results in a design algorithm with good numerical properties and a filter implementation with good sensitivity properties. However, the filters obtained by the method described in [122] do not necessarily have monotonic passbands, which is sometimes a requirement.

This chapter describes a modification to the filter design method of [122]. It produces odd-length linear phase filters with equiripple stopbands and monotonic passbands having a specified degree of flatness at $\omega = 0$. Although passband monotonicity can be ensured by linear programming methods (by the use of derivative
constraints), it is preferable to use the modification of [122] described below, because of its (1) computational efficiency, (2) good numerical properties during the design, and (3) its low sensitivity filter structure. This chapter also discusses bandpass filter design and adapts the techniques of [122] to the design of lowpass differentiators.

5.2 The Design Algorithm

The structure described in [122], shown in Figure 5.2, achieves an \( L \) degree of flatness at \( \omega = 0 \). The filter length \( N \) must be odd, otherwise the delay component is a fractional delay which we wish to avoid. The filter transfer function is given by

\[
H(z) = z^{-(N-1)/2} + H_1(z)H_2(z).
\]  

(5.1)

where \( H_1(z) \) is given by

\[
H_1(z) = \left(1 - \frac{z^{-1}}{2}\right)^L.
\]  

(5.2)

Taking \( H_2 \) to be a highpass filter whose impulse response is symmetric and of length \( N - L \), \( H_2(e^{j\omega}) \) can be written [68] as \( H_2(e^{j\omega}) = e^{-j\frac{N-L-1}{2}\omega}A_2(\omega) \) where \( A_2(\omega) \) is the frequency response amplitude, a real valued function of \( \omega \). When \( L \) is chosen to be even, \( \left(\frac{1-e^{-j\omega}}{2}\right)^L \) can be written as

\[
\left(\frac{1-e^{-j\omega}}{2}\right)^L = \left(e^{-j\omega/2}\right)^L(-1)^{L/2}\left(\frac{\sin\omega}{2}\right)^L
\]

where we have used the identity \( \frac{1-e^{-j\omega}}{2} = je^{-j\omega/2}\sin\frac{\omega}{2} \). Therefore \( H(e^{j\omega}) \) can be written as \( H(e^{j\omega}) = e^{-j\frac{N-1}{2}\omega}A(\omega) \) where the frequency response amplitude can be written as

\[
A(\omega) = 1 + A_2(\omega)(-1)^{L/2}\left(\frac{\sin\omega}{2}\right)^L.
\]  

(5.3)

It should be noted that the use of \( L \), here and below, includes all the derivatives that are made to match the desired response, and so includes the 0\textsuperscript{th} derivative.
Let $M = (N - L - 1)/2$ and denote the filter coefficients of $H_2$ by $h_2(0), \ldots, h_2(N - L - 1)$. $A_2(\omega)$ can be written as

$$A_2(\omega) = h_2(M) + 2 \sum_{n=0}^{M-1} h_2(n) \cos(\omega(M - n)).$$  \hspace{1cm} (5.4)

Two approaches to the problem formulation for which exchange algorithms can be used are the following:

1. Specify $N, L, \omega_s$; minimize $\delta_s$.

2. Specify $N, L, \delta_s$; minimize the passband width, (minimize $\omega_s$).

The first of these two options is the traditional approach in which the bands of interest are well defined and the Chebyshev norm of the error function over those bands is minimized. The second version is a variation of this approach in which the Chebyshev error in the stopband is specified but the band edge, however, is not fixed. In this case, no band edge is actually used during the course of the design procedure. The band edge that results is the one that corresponds to the specified Chebyshev error $\delta_s$ and the specified degree of flatness $L$.

5.2.1 Specifying $\omega_s$.

The first of these two approaches is solved by applying the Remez exchange algorithm [68] over just the stopband. In other words, the modification made to the method
of [122] is to simply weight the passband error by 0. The use of the Remez algorithm in this way will yield the coefficients of $H_2$ that minimize

$$\left\| 1 + A_2(\omega)(-1)^{L/2} \left( \sin \frac{\omega}{2} \right)^L \right\|_\infty$$

(5.5)

over the stopband. On each iteration, a reference set of stopband frequencies is updated and the filter $H_2$ is found such that $A(\omega)$ alternately interpolates $\delta_+ \text{ and } -\delta_+$ over the reference set frequencies. The size of the reference set is $q = \frac{N-L+3}{2}$. Let $\omega_1, \ldots, \omega_q$ denote the reference set frequencies ordered in increasing order. The equation

$$A(\omega_i) = \delta_+ (-1)^i$$

(5.6)

that appears in the course of the Remez algorithm becomes

$$A_2(\omega_i) = \frac{\delta_+ (-1)^i - 1}{(-1)^{L/2} \left( \sin \frac{\omega_i}{2} \right)^L}$$

(5.7)

which is linear in the coefficients of $H_2$ and $\delta_+$. Solving these equations for $1 \leq i \leq q$ gives the coefficients of $H_2$ and $\delta_+$. These can be found efficiently by using the interpolation formulas as in the Park-McClellan algorithm. The reference set is updated as in the Parks-McClellan algorithm and a new filter $H_2$ is found, and so on, until convergence is obtained. Quadratic convergence to the unique optimal solution is guaranteed by the appropriate use of the Remez algorithm.

It should be noted that any implementation of the Remez algorithm which allows the user to give an arbitrary weighting function and an arbitrary desired magnitude response can be used: Setting

$$W(\omega) = \begin{cases} 0 & \text{for } \omega < \omega_s \\ (-1)^{L/2} \left( \sin \frac{\omega}{2} \right)^L & \text{for } \omega \geq \omega_s \end{cases}$$

(5.8)

and

$$D(\omega) = \frac{-(1)^{L/2}}{\left( \sin \frac{\omega}{2} \right)^L}$$

(5.9)
equation (5.5) becomes

\[ \| (A_2(\omega) - D(\omega))W(\omega) \|_\infty \] (5.10)

which is the appropriate formulation for use with the Remez algorithm.

5.2.2 Specifying \( \delta_s \)

To specify \( \delta_s \) and leave the stopband edge variable, we use an approach similar to that of [96]. Like the Remez algorithm, this approach employs a set of stopband reference frequencies. On each iteration (1) an interpolation problem is solved and (2) the reference set is updated. The reference set here, however, does not contain the stopband edge (indeed, it is not specified). Therefore the reference set contains \( N - L + 1 \) stopband frequencies.

Given a set of reference frequencies, the filter that alternately interpolates \(-\delta_s\) and \(\delta_s\) is found. The interpolation is such that the filter interpolates \(-\delta_s\) at the first reference set frequency. Note that because \(\delta_s\) is specified by the user, it does not have to be found as in the Remez algorithm. Also note that a filter can be found that satisfies this interpolation requirements because the number of reference set frequencies equals the number of filter parameters. At each iteration, the local extremal frequencies of \(A(\omega)\) in \((0, \pi]\) are found and are taken to be the reference set frequencies for the next iteration.

Example 1: In Figure 5.3, the circular marks indicate the reference set frequencies upon convergence when this approach is used. Notice that the stopband edge is not included among these circular marks and its location is controlled only indirectly. However, with this approach, the user can directly specify the \(\delta_s\) parameter.

This approach produces the same set of filters as does the use of the Remez algorithm described above, however, it gives a different way to specify the filter parameters in the design process. Note that this approach is also similar to the approach of Hofstetter, Oppenheim and Siegel to the design of extra-ripple filters [41, 42]. The
Figure 5.3: A filter having a flat monotonic passband and an equiripple stopband. \( N = 33, L = 22, \delta_s = 0.02 \)

similarity lies in (1) the ability to specify \( \delta \) and the use of this specified value during the interpolation process and (2) the variability of the band edge. The approach described above is also like the algorithm of Hofstetter et al in that, while we have no proof of convergence, in practice the algorithm duplicates the rapid, robust convergence of the Remez algorithm.

5.2.3 Passband Monotonicity

The passband can be shown to be monotonic by the following reasoning. Recall that when no degree of flatness is imposed upon \( A(\omega) \) the maximum number of frequencies in \([0, \pi]\) at which the derivative of \( A(\omega) \) equals zero is \((N + 1)/2\) \([76]\). Note also that additional degrees of flatness imposed at \( \omega = 0 \) reduces the number of frequencies at which \( A'(\omega) \) can equal zero. Because the filters produced by the methods described above have the property that \( A'(\omega) \) equals zero at \((N + 1 - L)/2\) frequencies in the stopband, it appears that there can be no passband frequencies (other than \( \omega = 0 \)) at which \( A'(\omega) \) equals zero. Therefore, the passband will be monotonic.
5.3 Bandpass Filter Design

This method can also be applied to the design of bandpass filters having very flat passbands. In this case, we specify a passband frequency, $\omega_p$, where we wish to impose flatness constraints. The deviations from 0 in the first and second stopbands are denoted by $\delta_1$ and $\delta_2$ respectively.

Example 2: Fig 5.4 shows the frequency response amplitude of a length 55 bandpass filter.

The appropriate filter structure has the transfer function $H(z) = z^{-(N-1)/2} + H_1(z)H_2(z)$ with

$$H_1(z) = \left(\frac{1 - 2(\cos \omega_p)z^{-1} + z^{-2}}{4}\right)^{L/2}$$ (5.11)

where $L$ is even, $N$ is odd, and $H_2$ is a filter whose impulse response is symmetric and of length $N - L$. The overall frequency response $H(e^{j\omega})$ can then be written as $H(e^{j\omega}) = e^{-j\frac{N-1}{2}\omega}A(\omega)$ where the frequency response amplitude $A(\omega)$ is given by

$$A(\omega) = 1 + (-1)^{L/2} \left(\frac{\cos \omega_p - \cos \omega}{2}\right)^{L/2} A_2(\omega)$$ (5.12)

and where $A_2(\omega)$ is given by (5.4).
In keeping with the previous discussion, we desire that the passband be monotonic on both sides of $\omega_p$. To ensure this behavior in the exchange algorithms described below, it is necessary that $L$ be a multiple of 4. When 4 divides $L$, the zeros of $H_1(z)$ have even multiplicity, making $A_1(\omega)$ a nonnegative function. Then $A(\omega)$ is concave over the passband with appropriately chosen $H_2$.

As above, there are two approaches for which simple exchange algorithms are well suited:

1. Specify $N$, $L$, $\omega_p$, stopband edges, $K = \delta_2/\delta_1$; minimize $\delta_1$.

2. Specify $N$, $L$, $\omega_p$, $\delta_1$, $\delta_2$; minimize passband width.

First we describe approach (1), which uses the Remez algorithm with a zero-weighted passband. Because our approach places all the reference set frequencies in the stopbands, and because the Remez algorithm requires that the error function alternate sign over the reference frequencies, the reference set must contain exactly one stopband edge at each iteration. For example, see Figure 5.4 in which the circular marks indicate the reference frequencies upon convergence. In this figure, the first stopband edge is included in the final reference set, but the second is not. Note that bandpass filters designed such that the passband is concave and flat at $\omega_p$ have passbands that are generally quite symmetric around $\omega_p$. For this reason, we suggest that the stopband edges are taken to be $\omega_a = \omega_p - \omega_i$ and $\omega_b = \omega_p + \omega_i$ for some $\omega_i$.

The reference set is updated by the following procedure. First compute the set of all local extremal frequencies of $A(\omega)$ in $[0, \pi]$. Calling this set $R$, remove $\omega_p$ from $R$. $R$ will then contain either $\frac{N-L+1}{2}$ or $\frac{N-L+3}{2}$ frequencies. If $R$ contains $\frac{N-L+3}{2}$ frequencies, then remove either 0 or $\pi$ as follows: if $|A(\pi)| < K|A(0)|$ then remove $\pi$, otherwise remove 0. Next, append either $\omega_a$ or $\omega_b$ to $R$: if $|A(\omega_b)| < K|A(\omega_a)|$ then append $\omega_a$, otherwise append $\omega_b$. $R$ is the new reference set and has size $\frac{N-L+3}{2}$.

On each iteration, the filter $H_2$ is found such that $A(\omega)$ interpolates $\delta_1(-1)^i$ over the reference set frequencies in the first stopband and $K\delta_1(-1)^i$ in the second stop-
band. The resulting interpolation equations are linear in the coefficients of \( H_2 \) and \( \delta_1 \). Convergence to a filter with a concave passband is quadratic.

A similar algorithm is used for approach (2) in which \( \delta_1 \) and \( \delta_2 \) are specified and the stopband edges are left variable. The reference set is updated in the same manner, except no stopband edge is appended to \( R \). Let \( \omega_1, \ldots, \omega_q \) denote the reference set frequencies ordered in increasing order. On each iteration, the filter \( H_2 \) is found such that

\[
A(\omega_i) = (-1)^{i+c}\delta_1 \quad \text{for } \omega_i < \omega_p \tag{5.13}
\]
\[
A(\omega_i) = (-1)^{i+c+1}\delta_2 \quad \text{for } \omega_i > \omega_p \tag{5.14}
\]

where \( c \) equals 0 or 1, whichever gives \( A(\omega) = -\delta_1 \) at the highest reference frequency less than \( \omega_p \). If the filter in Figure 5.4 were designed by specifying \( \delta_1 \) and \( \delta_2 \), the reference set upon convergence would exclude the first stopband edge \( \omega_s = \omega_p - \omega_1 \).

### 5.4 Lowpass Differentiators

Lowpass digital differentiators can also be designed using the approach described above. However, the parameterization for differentiators having a specified degree of tangency at \( \omega = 0 \) is more complicated, and the simple structure used above must be modified. Let \( C(\omega) = 1 - \cos \omega \) and let \( L \) and \( N \) both be even. The frequency response amplitude of a length \( N \) differentiator with \( L \) degrees of flatness can be expressed as

\[
A(\omega) = \left( \sin \frac{\omega}{2} \right) \left[ 2 + d_1C(\omega) + d_2C(\omega) + \cdots + d_{L/2-1}C^{L/2-1}(\omega) + A_2(\omega)C^{L/2}(\omega) \right] \tag{5.15}
\]

where \( A_2(\omega) \) is an arbitrary cosine polynomial of degree \( (N-L-1)/2 \) and the first few \( d_i \) are as follows: \( d_1 = 1/6, \ d_2 = 3/80, \ d_3 = 5/448, \ d_4 = 35/9216, \ d_5 = 63/45056, \ d_6 = 231/425984 \). The general formula appears to be given by

\[
d_k = \frac{1 \cdot 3 \cdot 5 \cdots (2k-1)}{k! \cdot (2k+1) \cdot 2^{2k-1}}. \tag{5.16}
\]
Figure 5.5: Filter structure for implementation and design of a length $N$ lowpass differentiator. $C(z) = -(1 - z^{-1})^2/2$, $S(z) = (z^{-1} - 1)/2$, $N$ even.

For the amplitude given in (5.15), $A(0) = 0$, $A'(0) = 1$, and $A^{(k)}(0) = 0$ for $k = 2, \ldots, L$.

Because the method described above for lowpass filter design uses a reference set of \textit{stopband} frequencies only, exactly the same procedure can be used here. Accordingly, it is possible to either (1) specify the stopband edge $\omega_s$ and leave $\delta_s$ variable or (2) specify $\delta_s$ and leave $\omega_s$ variable. As above, the interpolation equations are linear in the coefficients of $A_2(\omega)$ and $\delta_s$.

Figure 5.5 shows the filter structure of an even-length differentiator for which $L = 6$. In the figure, $H(z)$ is a linear-phase transfer function of order $N - L - 1$. A maximally-flat differentiator can be obtained by setting $H(z) = 0$, but see also [51,92]. The structure for odd-length differentiators is similar.

For odd length differentiators, the amplitude response can be written as

$$A(\omega) = (\sin \omega) \left[ 1 + d_1 C(\omega) + d_2 C^2(\omega) + \cdots + d_{L/2 - 1} C^{L/2 - 1}(\omega) + A_2(\omega) C^{L/2}(\omega) \right]$$

(5.17)

where $A_2(\omega)$ is an arbitrary cosine polynomial and the first few $d_i$ are as follows:

$d_1 = 1/3$, $d_2 = 2/15$, $d_3 = 2/35$, $d_4 = 8/315$, $d_5 = 8/693$. The general formula
Figure 5.6: $N = 58$, $L = 34$, $\delta_s = 0.01$

appears to be

$$d_k = \frac{k!}{1 \cdot 3 \cdot 5 \cdots (2k + 1)}.$$  

For the amplitude given in (5.17), $A(0) = 0$, $A'(0) = 1$, and $A^{(k)}(0) = 0$ for $k = 2, \ldots, L$.

**Example 3:** Figure 5.6 shows the frequency response amplitude of a length 58 digital differentiator for which $L = 34$ designed by this approach.

### 5.5 Discussion and Conclusion

Even though the use of structures improves the numerical properties of the design procedure, there are limits on the filter length due to the difficulty associated with $(\sin \omega/2)^L$ in (5.7) when $L$ is very large. The reader is referred to [122] for discussions on IFIR filter design and implementation considerations, including coefficient sensitivity and roundoff noise.

It should be noted that passband monotonicity is achieved by this method without any explicit constraints on the concavity of the frequency response amplitude. It is obtained simply by locating all reference set frequencies in the stopband, so that the passband is shaped by the constraints embedded in the filter structure. Consequently,
in the design of lowpass filters by the algorithm described above, the location of the passband edge arises as the result of the stopband specification, which is given in terms of either $\omega_s$ or $\delta_s$, but not both. Similar tradeoffs between the ability to directly achieve specified parameters are discussed for equiripple lowpass and bandpass filters in chapter 4 and in [96].

Note that when $L$ is taken to be 2 for the lowpass case, the filter can be expressed analytically using Chebyshev polynomials [76]. More interestingly, if $L$ is taken to be 4 for the bandpass case, then the subset of maximal ripple bandpass filters can be found using analytic methods involving Zolotarev polynomials as described by Chen and Parks in [17]. Analytic solutions for higher values of $L$ in each case do not appear to be known.

We also wish to note that filters minimizing an integral square error having a specified degree of flatness at $\omega = 0$ are discussed in [73,91,92].

The lowpass filters designed by the method described above are analogous to the classical type II Chebyshev (or inverse Chebyshev) IIR filters [68]. Filters analogous to the classical type I Chebyshev IIR filters can be designed by the same principles. Linear phase FIR filters that are analogous to the classical Butterworth and Elliptic IIR filters are the maximally flat FIR filters of [37] and the equiripple FIR filters obtained by the Parks-McClellan algorithm. Thus, FIR analogues to each of the four classical IIR filter types can be designed without the use of general linear programming methods and without the need to explicitly impose linear constraints. The advantages of this is that (1) linear programming methods can be computationally intensive and (2) the use of linear constraints on derivatives become ill-conditioned for modest filter lengths. It is interesting to note (i) that maximally flat filters can be designed by employing simple filter structures [120], (ii) that equiripple filters can be designed by employing exchange algorithms, and (iii) that the filters described in this chapter can be designed by combining the use of a simple structure and by employing exchange algorithms. This is satisfying because the characteristics of the filters designed in this
chapter combine the characteristics of maximally flat filters and equiripple filters.

Also, recall that the four classical IIR filter types all have an equal number of poles and zeros. It is possible to design IIR filter with an unequal number of poles and zeros by combining the techniques described above with the rational Remez exchange algorithm discussed in Chapters 8 and 9 and [97].
Chapter 6

Nonlinear-Phase Maximally-Flat Lowpass FIR Filters

6.1 Introduction

A fair amount of attention has been focused on the design of non-symmetric digital FIR filters. Laakso et al. discuss in [52] the design of non-symmetric FIR filters for fractional delay applications. Other recent work has given attention to the design of FIR filters that approximate, in the Chebyshev norm, a given response magnitude and group delay [15, 109] or a given response magnitude and phase [12, 49, 57, 72]. Baher [7] gives an analytic technique for obtaining non-symmetric FIR filters having a maximally-flat behavior (see also [84, 90]). In [7], the degrees of flatness of the response magnitude and group delay at $\omega = 0$ can differ by no more than one. In this chapter, an analytic technique is given for the same problem Baher addresses in [7], but the degree of flatness of the response magnitude and group delay need not be approximately equal.

The motivation for the design of such filters has all-ready been well expressed.

...the question arises as to whether it is possible to drop phase linearity outside the passband and utilise the available degrees of freedom to shape the amplitude response. (From [7].)

...the exact linear phase and minimum phase solutions provide the extreme solutions: namely, the exact linear phase design unnecessarily constrains the linear phase response in the full frequency domain, while the minimum phase design drops the phase approximation altogether... The goal is to
design FIR filters whose properties are between those of exact linear and minimum phase filters. (From [57].)

Like the above cited papers, this chapter considers the problem of giving up exactly linear phase for approximately linear phase in exchange for a smaller delay and improvement in the response magnitude. The type of approximation considered in this chapter is Taylor. That is, derivatives of the realized response magnitude and group delay are made to vanish at $\omega_o = 0$ and $\omega_o = \pi$.

The approach taken in this chapter is appropriate when:

1. Exactly linear phase is not required.

2. Some degree of phase linearity is desired.

3. A maximally-flat frequency response is desired.

The family of new FIR filters described below permits the degrees of freedom to be divided between the frequency response magnitude and the group delay in a way that is achieved by neither linear-phase nor square magnitude design methods. This family of filters is conveniently described by forming a right triangle (see Table 6.5) that extends infinitely along its hypotenuse and one of its legs. The set of linear-phase maximally-flat filters forms the hypotenuse, while the set of maximally-flat filters designed in the square magnitude domain forms the other leg. The filters introduced below lie in the interior of the triangle. They make available trade-offs between the magnitude approximation and the phase linearity provided by neither linear-phase nor square magnitude maximally-flat filters.

The non-symmetric lowpass FIR filters discussed below are obtained by subjecting the frequency response magnitude and the group delay (individually) to differing numbers of flatness constraints. The equations that arise are nonlinear and have multiple solutions — some of which are extraneous (having unacceptable behavior in the interval $\omega \in (0, \pi)$ or being complex valued). Because multiple solutions exist and
because of the lack of guaranteed convergence, Newton's method and other numerical multi-variable equation solvers may be of limited utility. Since the equations form a system of multivariate polynomial equations, the use of a Gröbner basis is possible [19]. With a Gröbner basis, the solutions to the equations can be found by computing the roots of a set of univariate polynomials. Several software packages are available for the computation of Gröbner basis (a topic of computational algebraic geometry), for example [31].

Unfortunately, the computation of a Gröbner basis can be intensely computationally burdensome. However, this chapter identifies a class of filters which can be obtained without the computational load required for a Gröbner basis calculation. The computation needed to obtained filters in this class is (i) linear system solving and (ii) polynomial root finding. For filters in this class, it is shown how to achieve a specified half-magnitude frequency and DC group delay (not necessarily integer). It is also found that by using filters in this class, the delay can be reduced while maintaining relatively constant group delay, without significantly altering the response magnitude.

6.2 Notation

The transfer function of a length $N$ FIR filter is denoted by $H(z) = \sum_{n=0}^{N-1} h(n)z^{-1}$. The real and imaginary parts of the frequency response are denoted by $R(\omega) = \Re\{H(e^{j\omega})\} = \sum_{n=0}^{N-1} h(n)\cos n\omega$ and $I(\omega) = \Im\{H(e^{j\omega})\} = -\sum_{n=0}^{N-1} h(n)\sin n\omega$. The frequency response square magnitude is then given by $F(\omega) = R^2(\omega) + I^2(\omega)$ and the group delay by $G(\omega) = (I(\omega)R'(\omega) - R(\omega)I'(\omega))/F(\omega)$.

The number of zeros of $H(z)$ at $z = -1$ is denoted by $K$. The zeros at $z = -1$ produce a flat behavior in the frequency response magnitude at $\omega = \pi$. Note that because $F(\omega)$ and $G(\omega)$ are even functions of $\omega$, for odd $l$, $F^{(l)}(0)$ and $G^{(l)}(0)$ equal
Table 6.1 : Notation.

<table>
<thead>
<tr>
<th>Functions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(\omega)$</td>
<td>magnitude squared</td>
</tr>
<tr>
<td>$G(\omega)$</td>
<td>group delay</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>filter length</td>
</tr>
<tr>
<td>$K$</td>
<td>number of zeros at $z = -1$</td>
</tr>
<tr>
<td>$A$</td>
<td>group delay at $\omega = 0$</td>
</tr>
<tr>
<td>$\omega_o$</td>
<td>half-magnitude frequency</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flatness</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K + L + M$</td>
<td>total degrees of flatness</td>
</tr>
<tr>
<td>$K$</td>
<td>degree of magnitude flatness at $\omega = \pi$</td>
</tr>
<tr>
<td>$L$</td>
<td>degree of group delay flatness at $\omega = 0$</td>
</tr>
<tr>
<td>$M$</td>
<td>degree of magnitude flatness at $\omega = 0$</td>
</tr>
</tbody>
</table>

zero. The degree of flatness of the response magnitude at $\omega = 0$ is denoted by $M$:

$$F^{(2i)}(0) = 0 \quad i = 1, \ldots, M. \quad (6.1)$$

The degree of flatness of the group delay at $\omega = 0$ is denoted by $L$:

$$G^{(2i)}(0) = 0 \quad i = 1, \ldots, L. \quad (6.2)$$

A filter satisfying these conditions will be said to have flatness parameters $(K, L, M)$. The DC constraint,

$$F(0) = 1 \quad (6.3)$$

is also needed.

For the filters described in the first part of this chapter, $N = K + L + M + 1$. The only zeros lying on the unit circle are those $K$ zeros at $z = -1$. The number of zeros not lying at $z = -1$ is then $L + M$. Hence, all the degrees of freedom available are used to attain degrees of flatness of the frequency response at $\omega = 0$ and $\omega = \pi$. Later on, specifications on the group delay at $\omega = 0$ and the half-magnitude frequency will be
Figure 6.1: Flatness parameters $K$, $L$ and $M$.

achieved by giving up one or two degrees of flatness in which case $N = K + L + M + 2$
or $N = K + L + M + 3$.

The moments of $H(z)$ will be useful. They are denoted and defined as:

\[ m(k) = \sum_{n=0}^{N-1} n^k h(n). \] (6.4)

The half-magnitude frequency is that frequency at which the response magnitude equals one half. Like the 3 dB point, it indicates the location of the cut-off point.

6.3 The Basic Problem Formulation

To obtain maximally-flat FIR filters having different degrees of magnitude and group
delay flatness, the following mathematical problem formulation is suggested. Given
the flatness parameters $K$, $L$, $M$, (with $K > 0$, $M \geq 0$, $L \leq M$), find $N$ real filter coefficients $h(0), \ldots, h(N - 1)$ such that:


2. $F(0) = 1$. 
3. $H(z)$ has a root at $z = -1$ of order $K$.

4. $F^{(2i)}(0) = 0$ for $i = 1, \ldots, M$.

5. $G^{(2i)}(0) = 0$ for $i = 1, \ldots, L$.

Note that no constraints are made on the group delay at $\omega = \pi$ because the phase is unimportant in the stopband (also, the phase of 0 is undefined).

6.4 Example

To clarify the problem formulation above, an example is given before the design technique is described.

Example 1: Figure 6.2 shows the frequency responses of four different FIR filters of length 13. Their pole-zero plots are shown in Figure 6.2. Each of these filters has 6 zeros at $z = -1$ ($K = 6$) and 6 zeros contributing to the flatness of the passband at $z = 1$ ($L + M = 6$). The four filters shown were obtained using the four values $L = 0, 1, 2, 3$. Several remarks are in order.

The filter having linear phase corresponds to the values $L = 3, M = 3$. This symmetric filter is given by the formulas for maximally flat filters Herrmann presents in [38]. Clearly all the derivatives of the group delay of this filter are zero. It is labeled here by $L = 3, M = 3$ because those are the values for which the problem formulation above produces this filter.

The filter corresponding to the values $L = 0, M = 6$ can also be obtained using Herrmann’s formulas by spectrally factoring a length 25 maximally-flat symmetric filter. The group delay and impulse response shown in Figure 6.2 corresponding to $L = 0, M = 6$ are those of the minimum phase spectral factor. Not counting time-reversals, there are four distinct real-valued impulse responses for the $L = 0, M = 6$ case, which are obtained by flipping zeros about the unit circle. Counting time-reversals and complex-valued impulse responses, there are $2^6$ solutions.
The other two filters shown in Figure 6.2 ($L = 2, M = 4$ and $L = 1, M = 5$) can not be obtained using the formulas of Herrmann. They are new and provide a compromise solution.

Just as there are multiple solutions for the $L = 0$ case, there are multiple solutions in the cases for which $L > 0$. For $L = 1, M = 5$, there are two distinct real solutions (not counting time-reversal) that possess monotonic response magnitudes. The monotonic $L = 1, M = 5$ real solution not shown in figure 6.2 is shown in Figure 6.4. In addition to the filter shown in Figure 6.4, there is a real solution that is not monotonic. It is shown in Figure 6.17.

For $L = 2, M = 4$, there are two real solutions (not counting time-reversals) possessing non-monotonic response magnitudes (see Figure 6.16). For $L = 3, M = 3$ there are three real solutions (not counting time-reversal) possessing non-monotonic response magnitudes (see Figure 6.15). It should be noted, however, that it is true that in Figure 6.15 one of the three non-monotonic solutions is quite acceptable and looks almost as if it is monotonic.

Note that, while the four solutions for $L = 0, M = 6$ all have exactly the same magnitude, the two real solutions possessing monotonic response magnitudes for $L = 1, M = 5$ have differing magnitudes. (They are, however, very similar, as can be observed in Figure 6.17 in which they almost coincide to form a heavier line).

Our attention to passband monotonicity is motivated by the following. For the problem being addressed here, it is advantageous to know in advance that a frequency response magnitude is monotonic because this ensures a good behavior in $(0, \pi)$. If a solution is not monotonic, then the behavior in $(0, \pi)$ might be unacceptable due the potentially large errors, as evinced by Figures 6.15 through 6.17.

Observe that for the filters shown in Figure 6.3, the way in which the passband zeros are split between the interior of the unit circle and its exterior is given given by the values of $L$ and $M$. It is satisfying that for each $K, L$, and $M$, there is exactly one real solution possessing a monotonic magnitude and having this property. (Note
that, for \( K = 6, L = 1, M = 5 \), of the two real monotonic solutions shown in Figure 6.2 and 6.4, just the one shown in Figure 6.2 has this property.) It is interesting to note that the location of these zeros in this regard was not part of the way in which the problem was formulated.

It may be observed that the half-magnitude frequencies of the four filters in Figure 6.2 are unequal. (The response magnitudes obtained for the case \( L = 3, M = 3 \) has the smallest half-magnitude frequency; in the figure, they increase with increasing \( M \).) That the half-magnitude frequencies are unequal is to be expected, because the half-magnitude frequency was not included in the problem formulation. In the problem formulation of Section 6.3 both the half-magnitude frequency and the DC group delay can be only indirectly controlled by specifying \( K, L, \) and \( M \). Later on, achieving a specified half-magnitude frequency by giving up a degree of flatness will be described. Also, the specification of the group delay at \( \omega = 0 \) will be described.

The solutions shown in Figure 6.2 have approximately equal half-magnitude frequencies because they have the same number of zeros at \( z = -1 \). A set of solutions, the half-magnitudes of which are located over a greater interval, can be obtained by varying \( K \). Figure 6.5 shows the frequency response of seven filters of length 13. The seven filters are obtained by varying \( K \) from 2 to 8.

The same filter length is used throughout most of the examples in this chapter so that comparisons between them can be more easily made.

### 6.5 Obtaining Solutions

In terms of the moments of the filter, the square magnitude derivatives at \( \omega = 0 \) are given by:

\[
F^{(2n)}(0) = \binom{2n}{n} m^2(n) + 2(-1)^n \sum_{i=0}^{n-1} \binom{2n}{i} (-1)^i m(i) m(2n - i)
\]  

(6.5)

where the \( i^{th} \) moment \( m(i) \) is defined in (6.4). The general expressions for the derivatives of the group delay at \( \omega = 0 \) become very cumbersome; however, they can be
Figure 6.2: A selection of nonlinear-phase maximally-flat filters. $K = 6$, $L + M = 6$, $N = 13$. The four filters shown are solutions to the problem formulation of Section 6.3.
Table 6.2: The DC group delays and half-magnitude frequencies for the filters shown in Figure 6.2. \( L + M \) is the number of zeros contributing to the passband.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( K )</th>
<th>( L )</th>
<th>( M )</th>
<th>( \omega_0/\pi )</th>
<th>( G(0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>0.5502</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.5613</td>
<td>4.3201</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>0.5731</td>
<td>2.8290</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>0.5862</td>
<td>1.2934</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.3: \( K = 6, \ L + M = 6, \ N = 13 \). Continuation of Figure 6.2. Pole-zero plots of the filters shown in Figure 6.2. The number of zeros at \( z = -1 \) is 6 for each pole-zero plot.
Figure 6.4: The monotonic solution for $K = 6$, $L = 1$, $M = 5$, $N = 13$, not shown in Figures 6.2 and 6.3.
Figure 6.5: A greater range of half-magnitude frequencies. $N = 13 = K + L + M + 1$. See Table 6.3. Note that in Table 6.3, the filters are listed in order of decreasing $\omega_0$.

Table 6.3: The DC group delays and half-magnitude frequencies for the filters shown in Figure 6.5.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$L$</th>
<th>$M$</th>
<th>$\omega_0/\pi$</th>
<th>$G(0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>0.7915</td>
<td>4.771131</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0.7197</td>
<td>5.294824</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>0.6642</td>
<td>4.567345</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0.6070</td>
<td>5.1654</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0.5613</td>
<td>4.3201</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>0.5073</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3</td>
<td>0.4656</td>
<td>3.9683</td>
<td></td>
</tr>
</tbody>
</table>
simplified when the magnitude equations (6.5) are satisfied. When \( F(0) = 1 \) and \( F^{(2i)}(0) = 0 \) for \( i = 1, \ldots, n \), the group delay derivatives at \( \omega = 0 \) are given by:

\[
G^{(2n)}(0) = (-1)^n \sum_{i=0}^{n} \frac{2n + 1 - 2i}{2n + 1} \binom{2n + 1}{i} (-1)^i m(i) m(2n + 1 - i).
\]

(6.6)

Notice that the expression for \( F^{(2n)}(0) \) and \( G^{(2n)}(0) \) are nonlinear in \( m(i) \) and hence nonlinear in \( h(i) \).

From (6.5), the first few derivatives of the square magnitude at \( \omega = 0 \) are:

\[
F(0) = m_0^2
\]

(6.7)

\[
F^{(2)}(0) = 2m_1^2 - 2m_0m_2
\]

(6.8)

\[
F^{(4)}(0) = 6m_2^2 + 2m_0m_4 - 8m_1m_3
\]

(6.9)

\[
F^{(6)}(0) = 20m_3^2 - 2m_0m_6 + 12m_1m_5 - 30m_2m_4.
\]

(6.10)

From (6.6), the first few derivatives of the group delay at \( \omega = 0 \) are:

\[
G(0) = m_0m_1
\]

(6.11)

\[
G^{(2)}(0) = -m_0m_3 + m_1m_2
\]

(6.12)

\[
G^{(4)}(0) = m_0m_5 - 3m_1m_4 + 2m_2m_3
\]

(6.13)

\[
G^{(6)}(0) = -m_0m_7 + 5m_1m_6 - 9m_2m_5 + 5m_3m_4.
\]

(6.14)

Note that \( F^{(0)}(0) = m_0^2 \). Hence the equation \( F^{(0)}(0) = 1 \) requires that \( m_0 \) equal 1 or \(-1\). The solutions for which \( m_0 \) equals \(-1\) are obtained by negating the solutions for which \( m_0 \) equals 1. Therefore, throughout this chapter, \( m_0 = 1 \) is used. (Expression (6.11) becomes \( G(0) = m_1 \).) Also, throughout this chapter, the variable \( A \) is used to represent the DC group delay, the group delay at \( \omega = 0 \) \( (A = G(0) = m_1) \).

Because the filter coefficients can be determined from sufficiently many of its moments, having enough moments permits one to obtain a solution. Since \( m_0 = 1 \), \( m_1 = A \), from \( F^{(2)}(0) = 0 \) one gets \( m_2 = A^2 \), then using \( G^{(2)}(0) = 0 \), one gets \( m_3 = A^3 \), and so on. However, when \( L < M - 2 \), this process breaks down and prevents
one from finding sufficiently many moments to compute a solution. If \( L < M \), then it follows from (6.5) and (6.6) that \( m(i) = A^i \) for \( i = 0, \ldots, 2L + 2 \). If \( L = M \), then it follows from (6.5) and (6.6) that \( m(i) = A^i \) for \( i = 0, \ldots, 2L + 1 \).

### 6.5.1 Using Gröbner Bases

One method of computing all solutions is provided by the Gröbner basis of the equation set. Given a system of multivariate polynomials, a Gröbner basis (GB) is a new set of multi-variable polynomial equations, having the same set of solutions [19]. When the lexicographic ordering of monomials is used, and there is a finite number of solutions, the "last" equation of the GB will be a polynomial in a single variable — so its roots can be computed. These roots can be substituted into the remaining equations, etc — like back substitution in Gaussian elimination for linear equations. Using the purely lexical ordering of monomials with \( A < h(12) < h(11) < \cdots < h(0) \), the Gröbner basis for the case \( K = 6, L = 2, M = 4 \) was found using "Singular" [31] and is given in Table 6.4.

The first observation to be made is that there are 8 solutions for \( A \): the 8 roots of the first polynomial in Table 6.4. Also note that each of the filter coefficients appears linearly in the remaining equations, hence, there are exactly 8 solutions to the problem formulated above. These 8 solutions can be found by computing the roots of the first polynomial in Table 6.4 and solving the remaining equations using these values. For the problem considered here, the general structure of the Gröbner basis is the same regardless of the actual values of \( K, L, \) and \( M \): The first polynomial of the Gröbner basis is a univariate polynomial in \( A \), the next \( N \) polynomials are linear in the filter coefficients \( h(n) \). For a given \( K, L, M \), the polynomial in \( A \) that heads the Gröbner basis will be called the reference polynomial below and denoted by \( R_{L,K,M}(A) \).

It should be noted that the time-reversal of each solution \( h \) to the original problem must also be a solution, therefore, the roots of the reference polynomial must occur in pairs symmetrically around \( (N - 1)/2 \). (If \( A_0 \) is a root, then \( N - 1 - A_0 \) is also a
Table 6.4: The Gröbner basis for the case $K = 6$, $L = 2$, $M = 4$, $N = 13$.

<table>
<thead>
<tr>
<th>Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2A^6 - 96A^7 + 1928A^8 - 21024A^9 + 134858A^{10} - 513264A^{11} + 1102872A^{12} - 1162656A + 405405$</td>
</tr>
<tr>
<td>$92160h(12) - 2A^6 + 66A^5 - 845A^4 + 5280A^3 - 16523A^2 + 23364A - 10395$</td>
</tr>
<tr>
<td>$7680h(11) - A^6 + 25A^5 - 230A^4 + 950A^3 - 1689A + 945$</td>
</tr>
<tr>
<td>$15360h(10) + 2A^6 - 68A^5 + 885A^4 - 5560A^3 + 17323A^2 - 24132A + 10395$</td>
</tr>
<tr>
<td>$1536h(9) + A^6 - 27A^5 + 262A^4 - 1122A^3 + 2041A - 1155$</td>
</tr>
<tr>
<td>$6144h(8) - 2A^6 + 70A^5 - 933A^4 + 5936A^3 - 18475A^2 + 25284A - 10395$</td>
</tr>
<tr>
<td>$768h(7) - A^6 + 29A^5 - 302A^4 + 1366A^3 - 2577A + 1485$</td>
</tr>
<tr>
<td>$4608h(6) + 2A^6 - 72A^5 + 989A^4 - 6456A^3 + 20267A^2 - 27204A + 10395$</td>
</tr>
<tr>
<td>$768h(5) + A^6 - 31A^5 + 350A^4 - 1730A^3 + 3489A - 2079$</td>
</tr>
<tr>
<td>$6144h(4) - 2A^6 + 74A^5 - 1053A^4 + 7168A^3 - 23371A^2 + 31044A - 10395$</td>
</tr>
<tr>
<td>$1536h(3) - A^6 + 33A^5 - 406A^4 + 2262A^3 - 5333A + 3465$</td>
</tr>
<tr>
<td>$15360h(2) + 2A^6 - 76A^5 + 1125A^4 - 8120A^3 + 28843A^2 - 42564A + 10395$</td>
</tr>
<tr>
<td>$7680h(1) + A^6 - 35A^5 + 470A^4 - 3010A^3 + 9129A - 10395$</td>
</tr>
<tr>
<td>$92160h(0) - 2A^6 + 78A^5 - 1205A^4 + 9360A^3 - 38123A^2 + 75972A - 56475$</td>
</tr>
</tbody>
</table>

In the example of Table 6.4, it is found that of the 8 roots of the reference polynomial, 6 are real. Each of the 6 consequent solutions appear in pairs with its time-reversal. Not counting time-reversals, there are 3 solutions to the problem. One of these solutions is monotonic and is shown in Figure 6.2. The other two solutions are not monotonic and are shown in Figure 6.16.

### 6.5.2 The Number of Solutions

It is informative to form a table of the degrees of the reference polynomial for different $K$, $L$, $M$. It turns out that the degree of the reference polynomial is independent of $K$, the number of zeros at $z = -1$. The degree of the reference polynomial as a function of the remaining two parameters $L$ and $M$ is given in Table 6.5 for the first few $L$ and $M$. Notice that
Table 6.5: Degree of reference polynomial $\mathcal{R}_{K,L,M}(A)$.

<table>
<thead>
<tr>
<th>$L$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>16</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>26</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>48</td>
<td>24</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

1. Along the diagonal $L = M$, the degree increases linearly with $L$ and $M$: \[\text{degree} = 2L + 1 = 2M + 1.\]

2. Along the first column $L = 0$, the degree increases exponentially with $M$: \[\text{degree} = 2^M.\]

This suggests that the complexity required for computing the solutions when $L = M$ and when $L = 0$ must differ. (Of course, the filters for $L = 0$, can be obtained by spectral factorization — we are referring to computing solutions by the method discussed here — the goal here is to compute solutions between the diagonal and the column $L = 0$). The different solutions obtained for $L = 0$ and a fixed $M$ can be obtained from a single such solution by flipping zeros reciprocally about the unit circle. It is to be expected that the number of solutions for $L = 0$ increases exponentially with $M$, for the number of possibilities when $M$ zeros lie off the unit circle is $2^M$. The number of real solutions is less than this because most of the filters obtained by flipping zeros are complex. The requirement that the solution be real means that the zeros off the real-line must occur in complex conjugate pairs.

What happens for $0 < L < M$? How does the degree of the reference polynomial as a function of $L$ and $M$ change from exponential to linear?
Table 6.6: Number of real roots of reference polynomial $R_{K,L,M}(A)$.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>?</td>
<td>?</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td>17</td>
</tr>
</tbody>
</table>

Because the number of real solutions equals the number of real roots of the reference polynomial, this number is also of interest. Again, this number is independent of $K$. In Table 6.6, the number of real roots is shown for the first few values of $L$ and $M$. Most relevant is the number of real monotonic solutions. This number is shown as a function of $L$ and $M$ in Table 6.7. It becomes clear from examining this table that the triangle can be split into two distinct regions:

Let region I be defined by:

$$M \geq L \geq \frac{M}{2} - 1 \quad \text{for even } M$$

(6.15)

and

$$M \geq L \geq \frac{M - 1}{2} \quad \text{for odd } M.$$  

(6.16)

Let region II be defined by:

$$0 \leq L < \frac{M}{2} - 1 \quad \text{for even } M$$

(6.17)

and

$$0 \leq L < \frac{M - 1}{2} \quad \text{for odd } M.$$  

(6.18)
Table 6.7: Number of real monotonic solutions, not counting time-reversals.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>4?</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4?</td>
<td>2?</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Region I includes the diagonal \( L = M \), to which linear-phase solutions correspond. Region II includes the column \( L = 0 \), to which filters designed in the magnitude squared domain correspond. Using the "floor" notation, region I consists of pairs \( L, M \) for which \( \left\lfloor \frac{M-1}{2} \right\rfloor \leq L \leq M \). Region II consists of pairs \( L, M \) for which \( 0 \leq L \leq \left\lfloor \frac{M-1}{2} \right\rfloor - 1 \). See Table 6.8.

From Table 6.7, it can be seen that for each \( L \) and \( M \) in region I, there is one real solution having a monotonic response magnitude, while for each \( L \) and \( M \) in region II, there is more than one. We conjecture, that in region II there are \( 2^{\frac{M-1}{2}} \) solutions having a monotonic response magnitude for odd \( M \), and \( 2^\frac{M-1}{2} \) for even \( M \).

An additional observation is worth noting. For a given \( L \) and \( M \), how many of the \( L + M \) "passband" zeros lie inside the unit circle — and how many lie outside? It turns out that in region I, the real monotonic solution (the only ones in region I in which we are interested) has \( L \) zeros outside the unit circle and \( M \) zeros outside, as mentioned above. (If the time-reversal is used, then these numbers are flipped.) This distribution of the zeros inside and outside the unit circle is consistent with the properties of linear-phase filters.
Although for each \( L \) and \( M \) in region II, there is more than one real solution having a monotonic response magnitude, only one of these solutions has a monotonic group delay. We conjecture that of the real monotonic solutions in region II, only the solution having the monotonic group delay has the property that the zeros split between the exterior and interior of the unit circle according to \( L \) and \( M \).

Computing the filters for \( K, L, M \) in region II quickly becomes intractable using Gröbner basis because the sizes of the polynomials grows exponentially with \( M \). Also, the coefficients of the polynomials of the Gröbner basis for region II problems become extremely large (we computed Gröbner bases for which polynomial coefficients were hundreds of digits long!). (The coefficients are integers because the original set of polynomials have integer or rational coefficients and because “Singular” converts rational coefficient polynomials into integer coefficient polynomials by multiplication by the appropriate integer.) We were able to compute very few filters in region II using Gröbner basis - the complexity of the problem grows explosively.
In the problem formulation above, $\omega_0$ and $G(0)$ are determined by $K$, $L$, and $M$. But it is useful to know what values of $\omega_0$ and $G(0)$ can be achieved, and what the relationships among these parameters are. To this end, consider the following example in which filters of length 13 are examined. For each $(K, L, M)$ with the property that (i) $K + L + M + 1 = 13 = N$ and (ii) $(L, M)$ is in region I, Table 6.9 tabulates the single real solution having a monotonic response magnitude. Table 6.9 gives $\omega_0$ and $G(0)$. It is informative, and useful later on, to take $(\omega_0, G(0))$ as a pair of coordinates, and to plot these points in a plane. Figure 6.6 shows where these points lie for the filters listed in Table 6.9.

Figure 6.6: Locations in the $\omega_0$-$G(0)$ plane of each of the length 13 filters in region I for which $N = K + L + M + 1$. (They are tabulated in Table 6.9). The horizontal axis indicates the half-magnitude frequency. The three integers by each point give $K$, $L$ and $M$ respectively.
Table 6.9: The DC group delay and the half-magnitude frequency $\omega_c$ for each length 13 filter in region I for which $N = K + L + M + 1$.

<table>
<thead>
<tr>
<th>$K$</th>
<th>$L$</th>
<th>$M$</th>
<th>$\omega_c/\pi$</th>
<th>$G(0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>6</td>
<td>0.8813</td>
<td>5.417864</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>7</td>
<td>0.8852</td>
<td>4.337345</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>8</td>
<td>0.8894</td>
<td>3.276630</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0.7857</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>0.7915</td>
<td>4.771131</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>7</td>
<td>0.7973</td>
<td>3.637600</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0.7197</td>
<td>5.294824</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>0.7261</td>
<td>4.066220</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0.6561</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>0.6642</td>
<td>4.567345</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>0.6727</td>
<td>3.283241</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0.6070</td>
<td>5.165387</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>0.6159</td>
<td>3.760695</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>0.5502</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0.5613</td>
<td>4.320112</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>0.5073</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>4</td>
<td>0.5200</td>
<td>3.345999</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0.4498</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3</td>
<td>0.4656</td>
<td>3.968287</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>0.4089</td>
<td>4.751821</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0.3439</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>2</td>
<td>0.3698</td>
<td>3.342554</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0.3023</td>
<td>4.267949</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0.2143</td>
<td>6</td>
</tr>
</tbody>
</table>
6.5.3 Region I Filters

The filters associated with region I can be designed without using Gröbner basis. For $L$ and $M$ in region I, the equations, although nonlinear, can be solved by more direct linear algebraic methods, described in the next section. In region I, all the variables in the problem formulation, except $A$, the group delay at $\omega = 0$, are linearly related and can be eliminated, yielding a polynomial in $A$. This approach is described in the next section. By using the following derivation, the filters associated with region I can be designed without the explosive computational growth needed for Gröbner basis design of region II filters.

6.6 Calculating Filters in Region I

Recall that the degree of $H(z)$ is $K + L + M$. First, the $K$ zeros at $z = -1$ can be included in $H(z)$ explicitly by letting

$$H(z) = (z + 1)^K P(z). \quad (6.19)$$

$P(z)$ contains the $L + M$ zeros of $H(z)$ that contribute to the passband flatness ("$P$" is used here for "passband"). The degree of $P(z)$ is $L + M$. Let $h$ denote the vector of size $N$ of coefficients of $H(z)$:

$$h = \begin{bmatrix}
  h(0) \\
  h(1) \\
  h(2) \\
  \vdots \\
  h(N-1)
\end{bmatrix} \quad (6.20)$$
and let $p$ denote the vector of size $L + M + 1$ of coefficients of $P(z)$:

$$
\mathbf{p} = \begin{bmatrix}
    p(0) \\
p(1) \\
p(2) \\
    \vdots \\
p(L + M)
\end{bmatrix}.
$$

(6.21)

The two vectors $h$ and $p$ can be related by a convolution matrix $T$ of size $N$ by $(L + M + 1)$:

$$
\mathbf{h} = T \mathbf{p}.
$$

(6.22)

Let $m$ denote the vector of the first $(L + M + 1)$ moments:

$$
\mathbf{m} = \begin{bmatrix}
    m(0) \\
m(1) \\
m(2) \\
    \vdots \\
m(L + M)
\end{bmatrix}.
$$

(6.23)

Then two vectors $m$ and $h$ can by related by a type of Vandermonde matrix $Q$ of size $(L + M + 1)$ by $N$:

$$
\mathbf{m} = Q \mathbf{h}.
$$

(6.24)

Combining (6.22) and (6.24), one gets $\mathbf{m} = QT \mathbf{p}$. A solution can be written in terms of either the moments $\mathbf{m}$ or the filter coefficients $\mathbf{h}$. It turns out that solving the problem is facilitated by using the moments rather than the coefficients of $P(z)$, because Equations (6.5) and (6.6) are more conveniently written in terms of the moments than the filter coefficients.

One also gets the two relationships, $\mathbf{p} = (QT)^{-1} \mathbf{m}$ and $\mathbf{h} = T(QT)^{-1} \mathbf{m}$.

It is necessary to consider three cases:
1. $L = M$ (the linear-phase diagonal), or $L = M - 1$.

2. $L = M - 2$.

3. The remaining filters in region I.

**6.6.1 Case 1**

Clearly, the monotonic linear-phase filters obtained when $L = M$ are more conveniently obtained via the formulas of Herrmann. The case $L = M$ is included here (i) for completeness and (ii) to verify that solutions to the problem formulation above (when $L = M$) are in fact the same as the filters of Herrmann.

The following discussion is valid for $L = M$ or $L = M - 1$. Inspecting the equations $F^{(2i)}(0) = 0$ and $G^{(2i)}(0) = 0$ for $i = 0, \ldots, L$, it follows that

$$m(i) = A^i \quad \text{for} \quad i = 0, 1, 2, \ldots, L + M + 1. \quad (6.25)$$

Therefore $h$ can be found in terms of $A$ via the equation $h = T(QT)^{-1}m$.

To obtain the values $A$ for which the flatness constraints are satisfied, consider the moment $m(L + M + 1)$. The moment $m(L + M + 1)$ is given by

$$m(L + M + 1) = \sum_{n=0}^{N-1} n^{L+M+1} h(n) \quad (6.26)$$

$$= v^t h \quad (6.27)$$

$$= v^t T(QT)^{-1} m \quad (6.28)$$

where $v$ is clear. Since $m(L + M + 1)$ also equals $A^{L+M+1}$ from (6.25), the reference polynomial for $A$ is therefore

$$A^{L+M+1} = v^t T(QT)^{-1} m. \quad (6.29)$$

The filter coefficients are obtained via $h = T(QT)^{-1}m$.

For example, when $K = 6$, $L = 3$, $M = 3$, then

$$v^t T(QT)^{-1} = \begin{bmatrix} 31185 & -\frac{173619}{2} & 74403 & -\frac{58603}{2} & 6195 & -\frac{1421}{2} & 42 \end{bmatrix} \quad (6.30)$$
and the reference polynomial is

\[ 2A^7 - 84A^6 + 1421A^5 - 12390A^4 + 59003A^3 - 148806A^2 + 173619A - 62370. \] (6.31)

As expected, \((N - 1)/2 = 6\) is a root of this polynomial. The solution obtained using this root is the only solution having a monotonic response magnitude — and is the same as that filter obtained using the formulas of Herrmann. For \(L = M - 1\), the filters obtained here are not symmetric.

### 6.6.2 Case 2

Let \(L = M - 2\). This case is similar to case 1. It follows from Equations (6.5) and (6.6) that

\[ m(i) = A^i \quad \text{for} \quad i = 0, 1, 2, \ldots, L + M. \] (6.32)

As above, \(h\) can be found in terms of \(A\) via \(h = TQT^{-1}m\).

The equation \(F^{(2M)}(0) = 0\) here simplifies to

\[ (2M - 1)A^{2M} - 2MAm(2M - 1) + m(2M) = 0. \] (6.33)

To obtain the values \(A\) for which the flatness constraints are satisfied, consider the two moments \(m(2M - 1)\) and \(m(2M)\). They can be written in terms of \(m\).

\[ m(i) = \sum_{n=0}^{N-1} n^i h(n) \] (6.34)

\[ = v_i^T h \] (6.35)

\[ = v_i^T TQT^{-1} m \] (6.36)

for \(i\) equal to \(2M - 1\) and \(2M\). The reference polynomial for \(A\) can now be found by substituting these two moment expressions in Equation (6.33).
6.6.3 Case 3

In the following, suppose \( L, M \) lie in region I and \( L < M - 2 \).

The approach in the following, is to write the moments in terms of \( A \) and to substitute these moment expressions into the equation \( F^{(2M)}(0) = 0 \) to obtain the reference polynomial in \( A \).

Inspecting the equations \( F^{(2i)}(0) = 0 \) for \( i = 0, \ldots, M \) and \( G^{(2i)}(0) = 0 \) for \( i = 0, \ldots, L \), it follows that

\[
m(i) = A^i \quad \text{for} \quad i = 0, 1, 2, \ldots, 2L + 2
\]  \quad (6.37)

when \( M > L \). It will be useful to partition the vector \( m \) into two sub-vectors \( m_a \) and \( m_b \) of lengths \( 2L + 3 \) and \( M - L - 2 \) respectively:

\[
m = \begin{bmatrix} m_a \\ m_b \end{bmatrix}
\]  \quad (6.38)

so that

\[
m_a = \begin{bmatrix} 1 \\ A \\ A^2 \\ \vdots \\ A^{2L+2} \end{bmatrix}
\]  \quad (6.39)

and \( m_b \) represents the remaining moments whose dependence on \( A \) is to be determined:

\[
m_b = \begin{bmatrix} m(2L + 3) \\ m(2L + 4) \\ \vdots \\ m(L + M) \end{bmatrix}
\]  \quad (6.40)

The next step is to obtain expressions for the moments \( m(i) \) for \( i \) from \( L + M + 1 \) to \( 2M - 2 \) in terms of the lower order moments. Expressions for these moments
are needed because they appear in the equations $F^{(2i)}(0) = 0$ for $i = 0, \ldots, M - 1$. Expressions for these moments are obtained by using the definition (6.4). Let $m_c$ be the vector of length $(M - L - 2)$ of these moments:

$$m_c = \begin{bmatrix} m(L + M + 1) \\ m(L + M + 2) \\ \vdots \\ m(2M - 2) \end{bmatrix}.$$  

(6.41)

Then

$$m_c = Vh$$  

(6.42)

$$= VT(QT)^{-1}m$$  

(6.43)

$$= Um$$  

(6.44)

where $V$ is a type of Vandermonde matrix of size $(M - L - 2)$ by $N$:

$$V_{i,j} = (j - 1)^{L+M+i}$$  

(6.45)

for $1 \leq i \leq M - L - 2$ and $1 \leq j \leq N$, and where $U$ is defined as $VT(QT)^{-1}$. It will be useful to partition $U$ into two sub-matrices: $U_1$ of size $(M - L - 2)$ by $(2L + 3)$, and $U_2$ of size $(M - L - 2)$ by $(M - L - 2)$ so that:

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}.$$  

(6.46)

Then

$$m_c = U_1m_a + U_2m_b.$$  

(6.47)

Finally, looking closely at the equations $F^{(2i)}(0) = 0$ for $i = L+2, \ldots, M - 1$, one finds that these equations can be written as

$$Y \begin{bmatrix} m_1 \\ m_c \end{bmatrix} = r$$  

(6.48)
where $Y$ is the matrix of size $(M - L - 2)$ by $2(M - L - 2)$ having the following form

\[
Y = \begin{bmatrix}
-(2L+4)A & (2L+4) \\
-(2L+6)A^3 & (2L+6)A^2 & -(2L+6)A & (2L+6) \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
-(2M-2)A^{c-2} & \ldots & \ldots & \ldots & \ldots & (2M-2)
\end{bmatrix}
\]

(with $c = 2(M - L) - 5$) and where $\mathbf{r}$ is the vector of length $M - L - 2$

\[
\mathbf{r} = \begin{bmatrix}
\tau_0 \\
\tau_1 \\
\vdots \\
\tau_{M - L - 3}
\end{bmatrix}
\]

where

\[
r_k = A^{2L+4+2k} \sum_{i=0}^{i+2k} (-1)^i \binom{2L + 2k + 4}{i}
\]

\[
= -A^{2L+4+2k} \binom{2L + 2k + 3}{2k + 1}.
\]

The sum of binomial coefficients was simplified using a standard identity [29].

Define $Y_1$ and $Y_2$ to be two sub-matrices of $Y$, each of size $(M - L - 2)$ by $(M - L - 2)$ so that $Y = [Y_1 \ Y_2]$. Using the equations above, the sought moment vector $\mathbf{m}_s$ can be found as follows:

\[
Y_1 \mathbf{m}_s + Y_2 \mathbf{m}_s = \mathbf{r}
\]

\[
Y_1 \mathbf{m}_s + Y_2(U_1 \mathbf{m}_s + U_2 \mathbf{m}_s) = \mathbf{r}
\]

\[
(Y_1 + Y_2U_2) \mathbf{m}_s = \mathbf{r} - Y_2U_1 \mathbf{m}_s
\]

\[
\mathbf{m}_s = (Y_1 + Y_2U_2)^{-1}(\mathbf{r} - Y_2U_1 \mathbf{m}_s).
\]

Note that the elements of the matrix $(Y_1 + Y_2U_2)$ are polynomials in $A$ (see the example below). A symbolic programming language, such as Maple, makes the computation of $\mathbf{m}_s$ via (6.56) straight-forward, however.
To determine \( A \), the reference polynomial in \( A \) is obtained by substituting the expressions obtained for the filter moments into the equation \( F^{(2M)}(0) = 0 \). \( A \) is taken to be one of the roots of the resulting univariate polynomial, and with this value of \( A \), all the moments can be found using the expressions above. Finally, the filter coefficients \( h(n) \) can be obtained by solving the linear system of equations \( h = T(QT)^{-1}m \).

6.6.4 Root Selection

As mentioned above, the roots of the reference polynomial are located symmetrically around \((N - 1)/2\) (because the time-reversal of a solution must also be a solution). (But when \( M = L \), the reference polynomial has odd degree and the root at \((N - 1)/2\) is simple.) Which real pair of roots produces the single monotonic response magnitude? It was found that in general the pair of roots lying closest to \((N - 1)/2\) is the sought pair.

Since it is known that the reference polynomial is symmetric about \((N - 1)/2\), before computing its roots, it is worthwhile translating it to obtain an even polynomial in \( A \) and reducing its degree by letting \( A^2 = b \). The roots of the new polynomial can be mapped back, to obtain the roots of the reference polynomial.

6.6.5 Example

Example 2: To clarify the expressions described above, this example is given for \( K = 4, L = 2, M = 6 \) \((N = 13)\). For these values, \( m(i) = A^i \) for \( 0 \leq i \leq 6 \) and the
The matrices described above are:

\[
\mathbf{m}_a = \begin{bmatrix}
1 \\
A \\
A^2 \\
A^3 \\
A^4 \\
A^5 \\
A^6
\end{bmatrix}
\]  \hspace{1cm} (6.57)

\[
\mathbf{m}_b = \begin{bmatrix}
m(7) \\
m(8)
\end{bmatrix}
\]  \hspace{1cm} (5.58)

\[
\mathbf{T} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\
4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\
1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\
0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\
0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\
0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\
0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\
0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (6.59)
\[ Q = \begin{bmatrix}
1 & 1 & 1 & 1 & \cdots & 1 & 1 \\
0 & 1 & 2 & 3 & 4 & \cdots & 11 & 12 \\
0 & 1 & 4 & 9 & 16 & \cdots & 121 & 144 \\
0 & 1 & 8 & 27 & 64 & \cdots & 1331 & 1728 \\
0 & 1 & 16 & 81 & 256 & \cdots & 14641 & 20736 \\
0 & 1 & 32 & 243 & 1024 & \cdots & 161051 & 248832 \\
0 & 1 & 64 & 729 & 4096 & \cdots & 1771561 & 2985984 \\
0 & 1 & 128 & 2187 & 16384 & \cdots & 19487171 & 35831808 \\
0 & 1 & 256 & 6561 & 65536 & \cdots & 214358881 & 429981696 \\
\end{bmatrix} \] (6.60)

\[ V = \begin{bmatrix}
0 & 1 & 512 & 19683 & 262144 & \cdots & 2357947691 & 5159780352 \\
0 & 1 & 1024 & 59049 & 1048576 & \cdots & 25937424601 & 61917364224 \\
\end{bmatrix} \] (6.61)

\[ Y = \begin{bmatrix}
-8 A & 1 & 0 & 0 \\
-120 A^5 & 45 A^2 & -10 A & 1 \\
\end{bmatrix} \] (6.62)

\[ r = \begin{bmatrix}
-7 A^8 \\
-84 A^{10} \\
\end{bmatrix} \] (6.63)

\[ U_1 = \begin{bmatrix}
155925 & -\frac{2584215}{2} & 2037618 & -1399985 & 521136 & -114744 & 15372 \\
1824325 & -7241240 & 274792353 & -75495420 & 27293680 & -5765760 & 723912 \\
\end{bmatrix} \] (6.64)

\[ U_2 = \begin{bmatrix}
-1230 & 54 \\
-51480 & 1695 \\
\end{bmatrix} \] (6.65)

\[(Y_1 + Y_2 U_2)^{-1} = \begin{bmatrix}
\frac{3 A^2 - 36 A + 113}{16 A^3 - 288 A^2 + 1724 A - 3432} & \frac{1}{240 A^3 - 4320 A^2 + 25860 A - 51480} \\
-\frac{2 A^3 - 205 A + 858}{4 A^3 - 72 A^2 + 431 A - 858} & \frac{2 A}{60 A^3 - 1080 A^2 + 6465 A - 12870} \\
\end{bmatrix} \] (6.66)
\[
\mathbf{m}_3 = \begin{bmatrix} m(7) \\ m(8) \end{bmatrix} = (Y_1 + Y_2 U_2)^{-1}(r - Y_2 U_1 \mathbf{m}_2)
\]  
(6.67)

This gives:

\[
m(7) = (154 A^{10} - 2520 A^9 + 7910 A^8 + 102480 A^7 - 1247568 A^6 + \\
7318080 A^5 - 27529020 A^4 + 63914400 A^3 - 83541501 A^2 + \\
49533660 A - 6081075) / (160 A^3 - 2880 A^2 + 17240 A - 34320)
\]  
(6.68)

and

\[
m(8) = (14 A^{11} - 7175 A^{10} + 132510 A^9 - 1247568 A^8 + 7318080 A^7 - \\
27529020 A^6 + 63914400 A^5 - 83541501 A^4 + 49533660 A^3 - \\
6081075 A) / (20 A^3 - 360 A^2 + 2155 A - 4290).
\]  
(6.69)

Substituting these expressions for \(m(i)\) into the equation \(F^{(12)}(0) = 0\), one obtains the reference polynomial \(\mathcal{R}_{4,2,6}(A)\):

\[
4 A^{12} - 288 A^{11} + 9260 A^{10} - 175440 A^9 + 2175162 A^8 - \\
18519264 A^7 + 110406800 A^6 - 460618560 A^5 + 1317973984 A^4 - \\
2469896448 A^3 + 2765955690 A^2 - 1514878200 A + 212837625 = 0.
\]  
(6.70)

By making the change of variable, \(A = \sqrt{d} + 6\), the following lower order polynomial is obtained.

\[
4 d^6 - 244 d^5 + 4362 d^4 - 23632 d^3 + 28384 d^2 - 33174 d - 46575 = 0.
\]  
(6.71)

The roots of the lower order polynomial are:

\[
\{0.86977247 \pm 1.66295317, -0.70219452, 7.3807806, 18.989484, 33.592385\}.
\]  
(6.72)

Each of the real positive roots of the reduced polynomial are mapped to a pair of real roots of the reference polynomial located symmetrically around \((N - 1)/2 = 6\).
Hence, the real roots of the reference polynomial are:

\[ \{0.20410624, 1.6423074, 3.2832408, 8.7167592, 10.357693, 11.795894\} \]. \hfill (6.73)

The pair of roots giving rise to the single monotonic response magnitude is that pair lying closest to 6, namely, \( A = 3.2832408 \) and \( A = 8.7167592 \). The solution obtained using one of these two values is the time-reversal of the solution obtained using the other. In practice, it would generally make sense to employ the solution having smaller \( G(0) \).

### 6.7 Specification of the DC Group Delay

It is necessary to give up one degree of flatness for the ability to specify the value of the group delay at \( \omega = 0 \). In this case, the filter length \( N \) is related to the degrees of flatness by \( N = K + L + M + 2 \).

**Problem formulation:** Given \( A \) (the desired group delay at \( \omega = 0 \)), the integer \( K \), and the sum \( L + M \), (with \( K > 0, L + M > 0 \)), find \( L, M \), and \( N \) real filter coefficients \( h(0), \ldots, h(N - 1) \) such that:

1. \( N = K + L + M + 2 \).
2. \( F(0) = 1 \).
3. \( G(0) = A \).
4. \( H(z) \) has a root at \( z = -1 \) of order \( K \).
5. \( F^{(2i)}(0) = 0 \) for \( i = 1, \ldots, M \).
6. \( G^{(2i)}(0) = 0 \) for \( i = 1, \ldots, L \).
7. \( F(\omega) \) is monotonic in \((0, \pi)\).

The design of filters in region I with a specified \( G(0) \) is done relatively easily. The equations are roughly the same as those in Section 6.6, with slightly different
matrix dimensions. The main difference is that no reference polynomial for $A$ is needed. The design of filters in region II with a specified $G(0)$ will not be addressed here (computing them is too difficult with Gröbner basis and we don’t know another way to obtain them — more mathematical structure needs to be found before that problem becomes tractable). It should be noted that for this problem formulation there may not exist a solution having an acceptable response magnitude in $(0, \pi)$: The existence of an acceptable solution depends on the specified DC group delay. For a specified DC group delay, the flatness parameters must be chosen appropriately — this is discussed in detail below.

Taking $N = K + L + M + 2$, the degree of $H(z)$ is $K + L + M + 1$ and, as above, the $K$ zeros at $z = -1$ can be included in $H(z)$ by letting $H(z) = (z + 1)^K P(z)$ where the degree of $P(z)$ is $L + M + 1$. Let $h = [h(0), \ldots, h(N-1)]^T$, $p = [p(0), \ldots, p(L+M+1)]^T$. Let $T$ be the convolution matrix of size $N$ by $(L + M + 2)$ so that $h = Tp$. Let $m$ denote the vector of the first $L + M + 2$ moments, $m = [m(0), \ldots, m(L+M+1)]^T$. Then $m = Qh$ where $Q$ is the appropriate type of Vandermonde matrix of size $(L + M + 2)$ by $N$. As in the previous section, one has $p = (QT)^{-1}m$ and $h = T(QT)^{-1}m$.

It is necessary to consider two cases:

1. $L = M$ or $L = M - 1$.

2. The remaining filters in region I.

6.7.1 Case 1

Suppose $A = G(0)$ is given and $L = M$ or $L = M - 1$. As above, $m(i) = A^i$ for $i = 0, 1, 2, \ldots, L + M + 1$, therefore, $h$ can be obtained directly from $h = T(QT)^{-1}m$.

6.7.2 Case 2

Proceed as in Section 6.6.3, but use $m_b = [m(2L + 3), \ldots, m(L + M + 1)]^T$ and $m_c = [m(L + M + 2), \ldots, m(2M)]^T$. The modified matrix $V$ is of size $(M - L - 1)$
by \( N \) and has entries \( V_{i,j} = (j - 1)^{L+M+1+i} \) for \( 1 \leq i \leq M - L - 1 \) and \( 1 \leq j \leq N \). \( U, U_1, U_2, Y, Y_1 \) and \( Y_2 \) are all given by the same expressions as in Section 6.6.3 but their new dimensions are as follows: \( U \) is of size \((M - L - 1) \) by \((L + M + 2)\). \( U_1 \) is of size \((M - L - 1) \) by \((2L + 3)\). \( U_2 \) is of size \((M - L - 1) \) by \((M - L - 1)\). \( Y \) is of size \((M - L - 1) \) by \(2(M - L - 1)\). \( Y_1 \) and \( Y_2 \) are of size \((M - L - 1) \) by \((M - L - 1)\). \( r \) is of length \((M - L - 1)\).

### 6.7.3 Selecting \( L \) and \( M \)

Given \( G(0), K, \) and \( L+M, \) obtaining a filter of length \( N = K + L + M + 2 \) possessing a monotonic response magnitude requires selecting \( L \) and \( M \) appropriately. For an incorrect choice of \( L \) and \( M \), the resulting response magnitude may not be monotonic, and may therefore have an unacceptable error between 0 and \( \pi \).

**Example 3:** To illustrate how to select \( L \) and \( M \), consider again the filters shown in Figure 6.2, the DC group delays of which are tabulated in Table 6.2. Note that (i) for the case \( K = 6, L = 2, M = 4 \), the filter in Figure 6.2 has \( G(0) = 4.32 \), and (ii) for the case \( K = 6, L = 1, M = 5 \), the filter in Figure 6.2 has \( G(0) = 2.83 \). This suggests that a filter of the same length as those in Figure 6.2, having \( G(0) \in [2.83, 4.32] \) and a monotonic response magnitude, can be obtained using \( K = 6, L = 1, M = 4 \). Then the total degree of flatness is one less than those of Figure 6.2 (11 instead of 12), but any DC group delay in the interval \([2.83, 4.32]\) can be achieved. Figure 6.7 shows a \( K = 6, L = 1, M = 4, N = 13 \) filter having a DC group delay of 3.2.

For another example based on Figure 6.2, a filter having \( G(0) \in [4.32, 6] \) can be obtain by using \( K = 6, L = 2, M = 3 \). Figure 6.7 shows such a filter having a DC group delay of 4.8.

As \( A \), the specified DC group delay, is varied within the range \([4.32, 6]\), the half-magnitude frequency also varies. Note that \( \omega_o \) is not part of the problem formulation above, so is determined by \( K, L, M, \) and \( A \). However, it is informative to plot the way in which \( \omega_o \) varies as \( A \) varies over \([4.32, 6]\), as is done in Figure 6.8. The filter
Table 6.10: The DC group delays and half-magnitude frequencies for the filters shown in Figure 6.7.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( K )</th>
<th>( L )</th>
<th>( M )</th>
<th>( \omega_0/\pi )</th>
<th>( G(0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>0.5235</td>
<td>3.2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.5124</td>
<td>4.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

in Figure 6.7 with \( G(0) = 4.8 \) is indicated by a circular mark in Figure 6.8.

Note that the endpoints of the curve in Figure 6.8 are the two points in Figure 6.6 indicated by \((6, 3, 3)\) and \((6, 2, 4)\). In general, when \( L < M \), the endpoints of this curve are indicated by \((K, L + 1, M)\) and \((K, L, M + 1)\). When \( L = M \), however, the upper endpoint of the curve is at \( G(0) = (N - 1)/2 \).

From this discussion, it should be clear that, given \( G(0) \) and \( K \), the selection of \( L \) and \( M \) should be such that a curve like that of Figure 6.8 contains a point \((\omega_o, A)\) for which \( A \) is the specified DC group delay. That can be ascertained by using \((K, L + 1, M)\) and \((K, L, M + 1)\) in the problem formulation of Section 6.3 — and examining the DC group delay of the real monotonic solution in each case. (Note in addition, that this DC group delay is given by a root of a reference polynomial — so it can be found without computing the filter coefficients.)

The choice of \( K \) (the number of zeros at \( z = -1 \)) should be made to approximately locate the transition region. Also, to obtain filters having \( G(0) > (N - 1)/2 \), the time-reversal of solutions can be used.

It is interesting to observe the behavior of the pole-zero plot as the specified DC group delay is varied: When \( K \) is held constant and \( A \) is varied continuously towards \((N - 1)/2\), zeros lying inside the unit circle sequentially travel along the real axis through \( z = -1 \) to its exterior.
Figure 6.7: $K = 6$, $L + M = 6$, $N = 13$. Specified DC group delay: $G(0) = 4.8$ and $G(0) = 3.2$. For both filters, the number of zeros at $z = -1$ is 6.
Figure 6.8: $K = 6, L = 2, M = 3, N = 13$. The curve describes the way in which the half-magnitude frequency varies as the specified DC group delay varies.

6.8 Specification of the Half-Magnitude Frequency

As in Section 6.7, it is necessary to give up one degree of flatness for the ability to achieve a specified half-magnitude frequency. Again, only filters in region I are considered here.

**Problem formulation:** Given $\omega_0$ (the desired half-magnitude frequency), find $K, L, M,$ and $N$ real filter coefficients $h(0), \ldots, h(N-1)$ such that:

1. $N = K + L + M + 2$.
2. $F(0) = 1$.
3. $F(\omega_0) = \frac{1}{4}$.
4. $H(z)$ has a root at $z = -1$ of order $K$.
5. $F^{(2i)}(0) = 0$ for $i = 1, \ldots, M$.
6. $G^{(2i)}(0) = 0$ for $i = 1, \ldots, L$.
7. $F(\omega)$ is monotonic in $(0, \pi)$.
Recall that $F(\omega)$ is the square of the response magnitude, so a quarter is used in item 3 above. The paragraph preceding the enumeration of the two cases in Section 6.7 also applies here. As above, it is necessary to consider two cases:

1. $L = M$ or $L = M - 1$.

2. The remaining filters in region I.

In both cases, it is necessary to determine the correct value for $G(0)$.

6.8.1 Case 1

As in Section 6.7, $m(i) = A^i$ for $i = 0, 1, 2, \ldots, L + M + 1$ and $h = T(QT)^{-1}m$. However, here, $A$ is to be determined via the equation $F(\omega_0) = \frac{1}{4}$, which is a polynomial equation in $A$. Therefore, $A$ can be found by computing a root of this polynomial, and $h$ is found via $h = T(QT)^{-1}m$.

6.8.2 Case 2

The comments of Section 6.7.2 apply here. The coefficients $h$ are found in terms of the undetermined DC group delay, $A$, and substituted into the equation $F(\omega_0) = \frac{1}{4}$ to obtain a polynomial in $A$.

6.8.3 Example

Example 4: To illustrate how to solve this problem, consider the filters shown in Figure 6.5, the half-magnitude frequencies of which are tabulated in Table 6.3. The following discussion follows that of Section 6.7.3. Note that (i) for the case $K = 5$, $L = 3$, $M = 4$, the filter shown in Figure 6.5 has $\omega_0 = 0.607\pi$, and (ii) for the case $K = 6$, $L = 2$, $M = 4$, the filter shown in Figure 6.5 has $\omega_0 = 0.561\pi$. This suggests that a filter of the same length as those of figure 6.5, having $\omega_0 \in [0.561\pi, 0.607\pi]$, can be obtained using $K = 5$, $L = 2$, $M = 4$. Then the total degree of flatness is one
less than those of Figure 6.5 (11 instead of 12), but any half-magnitude frequency in
the interval \([0.561\pi, 0.607\pi]\) can be achieved.

For example, when \(\omega_o = 0.585\pi\) is specified, and \(K = 5, \ L = 2, \ M = 4\), then
the polynomial equation given by \(F(\omega_o) = \frac{1}{4}\) has four real roots. Consequently, there
are two real solutions (not counting time-reversals). It turns out that the response
magnitudes of these two solutions are both monotonic, and in fact, almost identical —
they coincide in Figure 6.9. Their group delay functions, however, are quite different.
See Figure 6.9 and Table 6.11.

For another example based on Figure 6.5, a filter having \(\omega_o \in [0.507\pi, 0.561\pi]\) can
be obtained by using \(K = 6, \ L = 2, \ M = 3\). Figure 6.9 shows two such filters each
having \(\omega_o = 0.535\pi\). (Their response magnitudes are almost identical and coincide in
Figure 6.9, however).

As \(\omega_o\), the specified half-magnitude frequency, is varied between \(0.507\pi\) and \(0.561\pi\),
the DC group delay also varies. Note that \(G(0)\) is not part of the problem formulation
above, so is determined by \(K, L, M, \) and \(\omega_o\). But the way in which \(G(0)\) and \(\omega_o\) vary
together for this example was illustrated in Figure 6.8. The existence of two solutions
for which \(\omega = 0.535\pi\) is explained by the fact that the curve shown in Figure 6.8 has
two points at which \(\omega_o\) equals \(0.535\pi\). Those two points represent the two filters in
Figure 6.9 for which \(\omega_o = 0.535\pi\).

6.9 Specification of Both the DC Group Delay and Half-
Magnitude Frequency

This section, which describes how one achieves both a specified half-magnitude fre-
quency and a specified DC group delay simultaneously, makes the approach taken in
this chapter more flexible and useful. (However, it should be noted that the existence
of acceptable solutions does depend on the \(\omega_o\) and \(G(0)\) specified.)

Achieving a specified \(\omega_o\) and \(G(0)\) is made possible by giving up two degrees of
flatness. In this case, the filter length \(N\) is related to the flatness parameters by
Figure 6.9: $N = 13 = K + L + M + 2$. Specified half-magnitude frequency: $\omega_o = 0.535\pi$ and $\omega_o = 0.585\pi$. See Table 6.11. Each apparent response magnitude is actually two response magnitudes coinciding to form one line.

Table 6.11: The DC group delays and half-magnitude frequencies for the filters shown in Figure 6.9.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$L$</th>
<th>$M$</th>
<th>$\omega_o/\pi$</th>
<th>$G(0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>0.585</td>
<td>4.7773</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.9643</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
<td>0.535</td>
<td></td>
<td>5.5562</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.5470</td>
</tr>
</tbody>
</table>
\[ N = K + L + M + 3. \] Again, only filters in region I are considered here.

**Problem formulation:** Given the filter length \( N \), \( A \) (the desired group delay at \( \omega = 0 \)), and \( \omega_0 \) (the desired half-magnitude frequency), find \( K \), \( L \), \( M \) and the \( N \) real filter coefficients \( h(0), \ldots, h(N - 1) \) such that:

1. \( N = K + L + M + 3. \)
2. \( F(0) = 1. \)
3. \( F(\omega_0) = \frac{1}{4}. \)
4. \( G(0) = A. \)
5. \( H(z) \) has a root at \( z = -1 \) of order \( K. \)
6. \( F^{(2i)}(0) = 0 \) for \( i = 1, \ldots, M. \)
7. \( G^{(2i)}(0) = 0 \) for \( i = 1, \ldots, L. \)
8. \( F(\omega) \) is monotonic in \( (0, \pi). \)

Taking \( N = K + L + M + 3, \) the degree of \( H(z) \) is \( K + L + M + 2 \) and, as above, the \( K \) zeros at \( z = -1 \) can be included in \( H(z) \) by letting \( H(z) = (z + 1)^K P(z) \) where the degree of \( P(z) \) is \( L + M + 2. \) Let \( h = [h(0), \ldots, h(N-1)]^t, \) \( p = [p(0), \ldots, p(L+M+2)]^t. \) Let \( T \) be the convolution matrix of size \( N \) by \( (L + M + 3) \) so that \( h = Tp. \) Let \( m \) denote the vector of the first \( L+M+3 \) moments, \( m = [m(0), \ldots, m(L+M+2)]^t. \) Then \( m = Qh \) where \( Q \) is the appropriate type of Vandermonde matrix of size \( (L + M + 3) \) by \( N. \) As above, one has \( p = (QT)^{-1}m \) and \( h = T(QT)^{-1}m. \)

It is necessary to consider three cases:

1. \( L = M \) or \( L = M - 1. \)
2. \( L = M - 2. \)
3. The remaining filters in region I.
6.9.1 Case 1

Suppose \( A = G(0) \) and \( \omega_o \) are given and \( L = M \) or \( L = M - 1 \). As above, \( m(i) = A^i \) for \( i = 0, \ldots, L + M + 1 \). Let \( \alpha \) denote the moment \( m(L + M + 2) \). Then \( \mathbf{m} = [1, A, A^2, \ldots, A^{L+M+1}, \alpha]^t \). One obtains \( \mathbf{h} \) as a linear function of \( \alpha \) via \( \mathbf{h} = T(QT)^{-1}\mathbf{m} \). The equation \( F(\omega_o) = \frac{1}{4} \) yields a quadratic polynomial equation for \( \alpha \), so \( \alpha \) can be found via the quadratic formula.

6.9.2 Case 2

Suppose \( A = G(0) \) and \( \omega_o \) are given and \( L = M - 2 \). As in Section 6.6.2, \( m(i) = A^i \) for \( i = 0, \ldots, 2L + 1 \). Let \( \alpha \) denote the moment \( m(2L + 3) \). In this case \( \mathbf{m} = [1, A, A^2, \ldots, A^{2L+2}, \alpha, m(2M)]^t \). Note that (6.33) applies here, so one obtains

\[
m(2M) = 2MA\alpha - (2M - 1)A^{2M}.
\]

Again, one obtains \( \mathbf{h} \) as a linear function of \( \alpha \) via \( \mathbf{h} = T(QT)^{-1}\mathbf{m} \), and the equation \( F(\omega_o) = \frac{1}{4} \) yields a quadratic polynomial equation for \( \alpha \).

6.9.3 Case 3

Suppose \( A = G(0) \) and \( \omega_o \) are given and \( L, M \) are in region I with \( L < M - 2 \). The following discussion is similar to that of 6.6.3. Partition \( \mathbf{m} \) so that

\[
\mathbf{m} = \begin{bmatrix} m_a \\ \alpha \\ m_b \end{bmatrix}
\]

where \( m_a = [1, \ldots, A^{2L+2}]^t \), \( \alpha \) denotes the moment \( m(2L + 3) \), and \( m_b = [m(2L + 4), \ldots, m(L + M + 2)]^t \). Also, define \( \mathbf{m_c} = [m(L + M + 3), \ldots, m(2M)]^t \). Then \( \mathbf{m_c} = U\mathbf{m} \), as in Section 6.6.3, where \( U = VT(QT)^{-1} \) and \( V \) is a type of Vandermonde matrix of size \((M - L - 2)\) by \( N \): \( V_{i,j} = (j - 1)^{L+M+2+i} \) for \( 1 \leq i \leq M - L - 2 \) and \( 1 \leq j \leq N \). Partition \( U \) into 3 sub-matrices so that \( U = [U_1 \quad U_2] \) where \( U_1 \) is of
size \((M - L - 2)\) by \((2L + 3)\), where \(s\) is a column vector, and \(U_2\) is of size \((M - L - 2)\) by \((M - L - 1)\). Then

\[
m_c = U_1 m_a + U_2 m_b + \alpha s.
\] (6.76)

As is done in Section 6.6.3, the equations \(F^{(2)}(0) = 0\) for \(i = L + 2, \ldots, M\) can be written as

\[
Y \begin{bmatrix} \alpha \\ m_b \\ m_c \end{bmatrix} = r
\] (6.77)

where \(Y\) has the form given in (6.49) but is of size \((M - L - 1)\) by \(2(M - L - 1)\), and \(r\) has the form given in (6.50) but is of length \((M - L - 1)\). Partition \(Y\) into 3 sub-matrices so that \(Y = [g \ Y_1 \ Y_2]\) where \(g\) is a column vector, \(Y_1\) is a square matrix, and \(Y_2\) is a matrix of size \((M - L - 1)\) by \((M - L - 2)\). With these expressions, one obtains \(m_b\) as follows:

\[
\alpha g + Y_1 m_b + Y_2 m_c = r
\] (6.78)

\[
\alpha g + Y_1 m_b + Y_2(U_1 m_a + U_2 m_b + \alpha s) = r
\] (6.79)

\[
(Y_1 + Y_2 U_2) m_b = r - Y_2 U_1 m_a - \alpha (g + Y_2 s)
\] (6.80)

\[
m_b = (Y_1 + Y_2 U_2)^{-1}(r - Y_2 U_1 m_a - \alpha (g + Y_2 s)).
\] (6.81)

This gives \(m\) as a linear function of \(\alpha\), and in turn, gives \(h\) as a linear function of \(\alpha\). The equation \(F(\omega) = \frac{1}{4}\) yields a quadratic polynomial equation for \(\alpha\).

6.9.4 Root Selection

The polynomial for \(\alpha\) above is second degree in each case. It was found that usually one of the roots gives a solution with an unacceptable response. Always one of the roots will give a solution with a monotonic response, however. Although a scheme for assigning the appropriate root to \(\alpha\) is not given here, the appropriate root can be easily determined by inspecting the two solutions to which they give rise.
6.9.5 Selection of $K$, $L$, $M$

Given a specified $\omega_o$ and $G(0)$, an important part of the use of these equations is the selection of $K$, $L$, and $M$. If the correct $K$, $L$, $M$ are not chosen, then the solutions given by the equations above may be unacceptable: They may be complex-valued and/or their frequency response may contain unacceptably large values.

To understand the relationship among $K$, $L$, $M$, $\omega_o$, and $G(0)$, it is useful to construct a figure by plotting the way in which $\omega_o$ and $G(0)$ vary together, as was done in Figure 6.8. By plotting such curves for each pair, $(K, L+1, M)$ and $(K, L, M+1)$, in region I, it is found that a partitioning of a portion of the plane is obtained, as illustrated in Figure 6.10 for $N = 13$. Given a specified $\omega_o$ and $G(0)$, (in one of the sectors), this figure indicates exactly how $K$, $L$, and $M$ should be selected so that a real monotonic solution is obtained that possesses the specified $\omega_o$ and $G(0)$ (as will be explained below). Several remarks are in order.

Notice that the nodes where the curves meet are those points shown in 6.6. These are points where the maximal degree of flatness is attained ($N = K + L + M + 1$). Note also, that each two nodes in Figure 6.10 are also neighbors in the triangle shown in Table 6.5. For example, consider the point in Figure 6.10 indicated by $(7,2,3)$. Its two neighbors having greater $G(0)$, $(8,2,2)$ and $(6,3,3)$, lie above and to the right of $(7,2,3)$ in Table 6.5. (Note that the $K$ coordinate is not used in Table 6.5.) The two neighbors of $(7,2,3)$ having smaller $G(0)$, $(8,1,3)$ and $(6,2,4)$, lie to left of and below $(7,2,3)$ in Table 6.5.

Recall from Sections 6.7 and 6.8, that the curves in Figure 6.10 are points where $N = K + L + M + 2$: One degree of flatness is given up in order to achieve a specified $G(0)$ or a specified $\omega_o$ (but not both). In that case, it was found that if the endpoints of a curve segment have flatness parameters $(K, L + 1, M)$ and $(K, L, M + 1)$, then the points along the curve segment have flatness parameters $(K, L, M)$.

Similarly, the flatness parameters for a point inside one of the sectors of Figure 6.10 is given by the minimum of the flatness parameters of the nodes at the vertices
of the sector. Consider the four sided sectors in Figure 6.10, note that the flatness parameters of the four nodes at its vertices can be written as \((K + L, L + 1, M), (K, L + 1, M), (K + 1, L, M + 1),\) and \((K + 2, L, M)\). It follows that the interior points have flatness parameters \((K, L, M)\).

Given a specified half-magnitude frequency \(\omega_o\) and a specified DC group delay \(G(0)\), the appropriate flatness parameters \(K, L,\) and \(M\) are selected by (i) locating the point \((\omega_o, G(0))\) in a sector in a figure such as Figure 6.10, and (ii) taking \(K, L,\) and \(M\) to be the minimum of the flatness parameters of the 3 or 4 encompassing nodes.

![Figure 6.10](image)

Figure 6.10: Specification sectors in the \(\omega_o-G(0)\) plane for length 13 filters in region I. The nodes are points at which \(N = K + L + M + 1.\) (They are tabulated in Table 6.9). The three integers by each point give \(K, L\) and \(M\) respectively.
Table 6.12: The flatness parameters for the filters shown in Figure 6.11.

<table>
<thead>
<tr>
<th>N</th>
<th>$\omega_0/\pi$</th>
<th>G(0)</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0.636</td>
<td>3.5</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.5</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

6.9.6 Example

**Example 5:** In this example a set of filters of length 13 is designed. Each filter achieves a specified half-magnitude frequency of $0.636\pi$. The specified DC group delays are varied from 3.5 to 6 in steps of one half. Notice that all the response magnitudes are very similar to one another. The filter responses are shown in Figure 6.11 and the flatness parameters used are tabulated in Table 6.12. Note that the group delay in the stopband behaves erratically, but recall that the phase in the stopband is of no importance.

A conclusion that can be drawn from this example is that the group delay can be reduced, while retaining relatively linear phase, without significantly altering the response magnitude.

6.9.7 Approximate Specification Sectors

As seen above, the selection of $K$, $L$, and $M$ ensuring monotonicity requires a figure such as Figure 6.10 in which the specification sectors are shown. However, drawing these curves (although straight-forward) requires more computation than is desirable.
for a convenient filter design program. Satisfactory specification sectors are obtained by connecting the nodes by linear line segments, as illustrated in figure 6.12. The resulting diamond shapes are adequate approximations of the actual sectors and are easily obtained. The only points in the \( \omega_o - G(0) \) plane at which the approximate sectors provide different flatness parameters than do the exact sectors are near the boundaries of the sectors. For these \( \omega_o - G(0) \) coordinates, "erroneously" selecting the flatness parameters of an adjacent sector leads only to slight non-monotonicity.

6.10 A Longer Example

Example 6: To avoid giving the reader the impression that the approach taken in this chapter is applicable only to filters of length 13, a filter of length 23 is considered here. The \( \omega_o \) of the previous example (\( \omega_o = 0.636\pi \)) is used. The DC group delay is varied from 7 to 11 in steps of one half. The specification sectors are shown in Figure 6.13. The filter responses are shown in Figure 6.14 and the flatness parameters used are tabulated in Table 6.13.

Notice in Figure 6.14 that the group delay can be reduced by just over a third.
Figure 6.12: Approximation of the specification sectors shown in Figure 6.10.
(4/11) while maintaining a very flat group delay over the passband. In addition, the magnitudes of the filters with smaller delay have sharper transition regions (albeit, negligibly).

Figure 6.13: Specification sectors in $\omega_0$-$G(0)$ plane for length 23 filters in region I.
Table 6.13: The flatness parameters for the filters shown in Figure 6.14.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \omega_0/\pi )</th>
<th>( G(0) )</th>
<th>( K )</th>
<th>( L )</th>
<th>( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>0.636</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.5</td>
<td>7</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.5</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.5</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.5</td>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 6.14: \( N = 23 = K + L + M + 3 \). Specified half-magnitude frequency and DC group delay. See Table 6.13. The response magnitudes are essentially the same, but the filters having smaller delay have very slightly steeper transition region responses.
6.11 Remark

Even though the problem formulations described above require the solutions to non-linear equations, a straight-forward solution method was found to be possible. The following items facilitated the procedure.

1. Using moments $m(i)$ to obtain convenient expressions for $F^{(2i)}(0)$ and $G^{(2i)}(0)$.

2. Identifying region I as a class of filters for which the problem formulations are more easily solved.

3. Eliminating linearly dependent variables to obtain a polynomial in $G(0)$.

4. Obtaining a tiling of part of the $\omega_c - G(0)$ plane by restricting attention to real monotonic solutions in region I.

6.12 Conclusion

That the problem of simultaneous magnitude and group delay approximation is a difficult one is made evident by the sophisticated iterative algorithms that are required [15,57,109]. However, as is shown in this chapter, if a maximally-flat approximation is employed, then, for a class of FIR filters, there exists a straight-forward design technique. This design technique

1. requires only the solution to linear systems and the roots of polynomials,

2. permits the specification of the half-magnitude frequency and the DC group delay, and

3. produces filters having reduced delay (over symmetric filters) while maintaining a very flat group delay in the passband, with essentially identical magnitude.

It is expected that the filters discussed in this chapter may find some of the same applications as the symmetric maximally-flat filters, for example [47,121], and possibly some of the applications given in [52].
Appendix

6.13 All Real Solutions to the First the Example

For filters of length 13 with $N = 13 = K + L + M + 1$, figures 6.15 through 6.18 show all the real solutions to the problem formulation stated in Section 6.3, both the monotonic and the non-monotonic. It can be seen that many of the responses that are non-monotonic take on huge values and are therefore unacceptable. The only exception is the response in Figure 6.15 that almost appears to be monotonic, but in fact, is not. That response shows that for some specifications, a non-monotonic response is acceptable.

![Frequency response and Group delay graphs]

Figure 6.15: All real solutions: $K = 6$, $L = 3$, $M = 3$, $N = 13$
Figure 6.16: All real solutions: $K = 6$, $L = 2$, $M = 4$, $N = 13$

Figure 6.17: All real solutions: $K = 6$, $L = 1$, $M = 5$, $N = 13$
Figure 6.18: All real solutions: $K = 6, L = 0, M = 6, N = 13$
Chapter 7

Generalized Digital Butterworth Filter Design

7.1 Introduction

Probably the best known and most commonly used method for the design of IIR digital filters is the transformation of the classical analog filters (the Butterworth, Chebyshev I and II, and Elliptic filters) [68]. One advantage of this technique is the existence of formulas for these filters. Unfortunately, all such IIR filters have an equal number of poles and zeros. It is desirable to be able to design filters having more zeros than poles (away from the origin), for implementation purposes. This chapter presents a method for the design of maximally-flat lowpass IIR filters having more zeros than poles and which possess a specified half-magnitude frequency. It is worth noting that not all the zeros are restricted to lie on the unit circle. The method consists of the use of a formula, a transformation of variables, and a spectral factorization. Note that no phase approximation is done; the approximation is in the magnitude squared - as are the classical IIR filter types.

Another main result of this chapter is that for a specified number of zeros and a specified half-magnitude frequency, there is only one valid way to divide the number of zeros between \( z = -1 \) and the passband. This is in contrast to the classical digital Butterworth filter, for which all the zeros lie at \( z = -1 \), regardless of the position of the half-magnitude frequency in \((0, \pi)\). The formulas given below provide a direct way to determine the number of zeros that must lie at \( z = -1 \) and the number of zeros that must contribute to the passband.

Given a half-magnitude frequency \( \omega_o \), the filters produced by the formulas described below are optimal in the sense that the maximum number of derivatives at
Table 7.1: Notation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L + M$</td>
<td>total number of zeros</td>
</tr>
<tr>
<td>$L$</td>
<td>number of zeros at $z = -1$</td>
</tr>
<tr>
<td>$M$</td>
<td>number of zeros contributing to the passband</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of poles</td>
</tr>
<tr>
<td>$\omega_o$</td>
<td>half-magnitude frequency</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flatness</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L + M + N$</td>
<td>total degrees of flatness</td>
</tr>
<tr>
<td>$M + N$</td>
<td>degree of flatness at $\omega = 0$</td>
</tr>
<tr>
<td>$L$</td>
<td>degree of flatness at $\omega = \pi$</td>
</tr>
</tbody>
</table>

$\omega = 0$ and $\omega = \pi$ are set to zero, under the constraint that the filter possesses the half-magnitude frequency $\omega_o$ and a monotonic response magnitude. The digital IIR filters obtained by transforming the classical Butterworth filters, and the FIR filters obtained by the use of Herrmann’s formulas [38] are both special cases of the filters produced by the formulas given in this chapter.

7.2 Notation

Let $B(z)/A(z)$ denote the transfer function of a digital filter. Its frequency response magnitude $M(\omega)$ is given by $|B(e^{j\omega})/A(e^{j\omega})|$. Throughout this chapter, the degree of $B(z)$ will be denoted by $L + M$, where $L$ is the number of zeros at $z = -1$ and $M$ is the number of zeros that contribute to the passband. No filter in this chapter has zeros on the unit circle other than at $z = -1$. The degree of $A(z)$ will be denoted by $N$.

The zeros at $z = -1$ produce a flat behavior in the frequency response magnitude at $\omega = \pi$, while the remaining zeros, together with the poles, are used to produce a flat behavior at $\omega = 0$. The meaning of the parameters is shown in Table 7.1. The half-magnitude frequency is that frequency at which the magnitude equals one half.
7.3 Examples

The classical digital Butterworth filters (defined by $L = N$ and $M = 0$) are special cases of the filters discussed in this chapter. The first generalization of the classical digital Butterworth filter described below permits $L$ to be greater than $N$: $L > N$ with $M = 0$. It turns out that when $L > N$, the restriction that $M$ equal zero limits the range of achievable half-magnitude frequencies, as will be elaborated below. This motivates the second generalization. In addition to permitting $L$ to be greater than $N$, the second generalization of the classical digital Butterworth filter described below permits $M$ to be greater than zero: $L \geq N$ and $M > 0$.

Example 1: Figure 7.1 shows the frequency response, pole-zero plot, and group delay for a classical digital Butterworth filter of order 4 ($L = 4$, $M = 0$, $N = 4$). It has a half-magnitude frequency of $0.4585\pi$.

Figure 7.2 shows the frequency response, pole-zero plot, and group delay for an IIR filter with $L = 6$, $M = 0$, $N = 4$. It was designed to have the same half-magnitude frequency as the previous example ($\omega_0 = 0.4585\pi$).

Figure 7.3 shows the frequency response, pole-zero plot, and group delay for an IIR filter with $L = 16$, $M = 7$, $N = 4$. It was designed to have a half-magnitude frequency of $0.4585\pi$.

As mentioned above, for a specified half-magnitude frequency $\omega_0$ and a specified number of zeros ($L + M$), there is only one correct way to split the zeros between $z = -1$ and the passband. To illustrate this property, it is helpful to construct a table that indicates the appropriate values for $L$, $M$ and $N$. When $N = 4$ and $L + M$ equals 4, $\ldots$, 10, table 7.2 gives the appropriate choice for $L$ and $M$ to achieve a desired half-magnitude frequency. As can be seen from the table, the intervals cover the interval (0,1) and do not overlap. This will be true in general, as long as the number of poles is at least one. Notice that in the case of the classical Butterworth filter ($L + M = N$), $L$ equals $N$ and $M$ equals zero, regardless of the specified half-magnitude frequency. As will be explained below, these intervals can be unambiguously computed by inspecting
Table 7.2: A specification table for generalized Butterworth filter design. For the choice $L$, $M$, and $N$ shown in the table, the interval of permissible half-magnitude frequencies $\omega_0$ is given by $\omega_{\min}$ and $\omega_{\max}$. $L + M$ is the numerator degree (number of zeros) and $N$ is the denominator degree (number of poles).

<table>
<thead>
<tr>
<th>$L + M$</th>
<th>$L$</th>
<th>$M$</th>
<th>$N$</th>
<th>$\omega_{\min}/\pi$</th>
<th>$\omega_{\max}/\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.5349</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.4620</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>0.5349</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.5299</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>0.4620</td>
<td>0.6446</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>0.6017</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.4140</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>0.4140</td>
<td>0.5299</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>0.5299</td>
<td>0.6446</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>0.6446</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.3788</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>0.3788</td>
<td>0.4807</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>0.4807</td>
<td>0.5754</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0.5754</td>
<td>0.6756</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0.6756</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.3515</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>0.3515</td>
<td>0.4435</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>0.4435</td>
<td>0.5266</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0.5266</td>
<td>0.6093</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>0.6093</td>
<td>0.6996</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>0.6996</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.3294</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>4</td>
<td>0.3294</td>
<td>0.4141</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>0.4141</td>
<td>0.4891</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>0.4891</td>
<td>0.5615</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>0.5615</td>
<td>0.6359</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>0.6359</td>
<td>0.7188</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>0.7188</td>
<td>1</td>
</tr>
</tbody>
</table>
the roots of an appropriate set of polynomials.

**Example 2:** To illustrate the different trade-offs that can be achieved with the generalized Butterworth filters described in this chapter, it is useful to examine a set of filters all of which have the same half-magnitude frequency and the same total number of poles and zeros \((L + M + N)\). For example, when \(L + M + N\) is fixed at 20 and the half-magnitude \(\omega_0\) is fixed at 0.6\(\pi\), the filters shown in Figure 7.4 are obtained. The number of poles of the filters in this figure vary from 0 to 10 in steps of 2. It is also interesting to compare the slope of the magnitude \(M(\omega)\) at the half-magnitude frequency — this indicates the sharpness of the transition band. This information is summarized in Table 7.3. The third graph of the figure shows the negative reciprocal of the slope of \(M(\omega)\) at \(\omega_0\). It turns out that when an odd number of poles between 1 and 9 is used, the resulting filter is very similar to the filter having one fewer pole. Notice from the table and the figure, that for this example, as the number of poles and zeros become more equal, the slope of the magnitude at \(\omega_0\) becomes more negative and that the transition region becomes sharper. However, it is interesting to note that the improvement in magnitude is greatest when the number of poles is increased from 0 to 2. Also, notice the behavior of the group delay as the number of poles and zeros are varied. When the cost of implementing a filter with many poles is taken into consideration, the filters with 2 or 4 poles appear to attain a good trade-off between performance and implementation cost.

### 7.4 Discussion

Several authors have addressed the design and the advantages of IIR filters with an unequal number of (nontrivial) poles and zeros.

While [89,116,117] give formulas for IIR filters with Chebyshev stopbands having more zeros than poles, these methods require that all the zeros lie on the unit circle.

In [61] Martinez and Parks describe an exchange algorithm for the Chebyshev design of IIR filters in which all the zeros lie on the unit circle. Other variations on
Figure 7.1: A classical digital Butterworth filter. $L = 4, M = 0, N = 4$. 
Figure 7.2: A digital Butterworth filter for which $L > N$. $L = 6$, $M = 0$, $N = 4$. 
Figure 7.3: A digital Butterworth filter for which \( L > N \) and \( M > 0 \). \( L = 16, \ M = 7, \ N = 4 \).
Figure 7.4: A set of digital Butterworth filters for which $L+M+N = 20$. Shown in the figures are the filters having half-magnitude frequency $\omega_0 = 0.6\pi$, and $N = 0 : 2 : 10$. $N = 10$ corresponds to the filter having the steepest transition and the most peaked group delay.
Table 7.3: Transition region sharpness of the filters shown in Figure 7.4. For the half-magnitude frequency \( \omega_0 = 0.6\pi \) and \( L + M + N = 20 \), the table shows the correct values of \( L \) and \( M \), and the derivative of the magnitude at \( \omega_0 \), for a fixed \( L + M \) and \( N \).

<table>
<thead>
<tr>
<th>( L )</th>
<th>( M )</th>
<th>( N )</th>
<th>( M'(\omega_0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>12</td>
<td>0</td>
<td>-1.4366</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>2</td>
<td>-2.5410</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>-3.1869</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>-3.6882</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>8</td>
<td>-3.8012</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>10</td>
<td>-3.9430</td>
</tr>
</tbody>
</table>

This is \([43, 58, 87, 100]\). In \([43]\) Jackson improves the Martinez/Parks algorithm and notes that the use of just 2 poles “is often the most attractive compromise between computational complexity and other performance measures of interest.”

In \([88]\) Saramäki discusses the trade-offs between numerator and denominator order and describes an iterative algorithm in which zeros are not constrained to lie on the unit circle for the design of filters having Chebyshev stopbands. In \([87, 88]\), Saramäki finds that the classical Elliptic and Chebyshev filter types are seldom the best choice. However, to our knowledge, no formulas have been presented for the design of IIR filters in which zeros are not constrained to lie on the unit circle.

The FIR maximally-flat filter \([38, 46, 80, 86, 111, 120]\) is a special case of the filters described in this chapter. It is also likely that the generalized Butterworth filters described in this chapter can be used with some of the filter design techniques that employ FIR maximally-flat filters, for example \([121]\).

The ability to design the non-classical IIR filters described in this chapter and the papers cited above, allows the user to trade-off between properties of FIR filters and properties of the four classical IIR filters. Note that one reason FIR filters
are sometimes preferred over IIR filters is the relative ease with which they can be implemented. However, by using a low-order recursive section, a potentially better trade-off can be obtained.

7.5 Derivations

The approach described below provides formulas for two nonnegative polynomials \( P(x) \) and \( Q(x) \). Then, by (i) using a suitable transformation \( x = \frac{1}{2}(1 - \cos \omega) \) as in [38]) and (ii) taking a spectral factor, a stable IIR filter \( B(z)/A(z) \) is obtained having a magnitude squared frequency response \( |M(\omega)|^2 \) given by

\[
|M(\omega)|^2 = \frac{P\left(\frac{1}{2} - \frac{1}{2} \cos \omega\right)}{Q\left(\frac{1}{2} - \frac{1}{2} \cos \omega\right)}.
\]

Accordingly, \( P(x)/Q(x) \) is designed to approximate a lowpass response over \( x \in [0,1] \). This results in a formula-based method for the design of generalized digital Butterworth filters. No iterations are required for finding \( P(x) \) and \( Q(x) \).

We begin by deriving the classical digital Butterworth filter. This establishes notation and makes clear the way in which the generalization uses the same ideas in its derivation.

7.5.1 Classical Digital Butterworth Filter

Let the degree of \( P(x) \) be \( L \) and the degree of \( Q(x) \) be \( L \), and define the rational function \( F(x) = P(x)/Q(x) \). To find \( P(x) \) and \( Q(x) \) so that \( F(x) \) possesses the lowpass behavior shown in figure 7.1, we will require that \( F(x) \) have \( L \) degrees of flatness at \( x = 1 \) and that \( F(x) - 1 \) have \( L \) degrees of flatness at \( x = 0 \).

In order to obtain \( L \) degrees of flatness at \( x = 1 \), \( F(x) \) must have the following form:

\[
F(x) = \frac{P(x)}{Q(x)} = \frac{(1 - x)^L}{Q(x)}.
\]
The $L$ degrees of flatness is obtained by specifying a root of order $L$ at $x = 1$. In order that $F(x) - 1$ have an $L$ degree of flatness at $x = 0$, $F(x)$ must satisfy

$$F(x) - 1 = \frac{P(x) - Q(x)}{Q(x)} = -\frac{cx^L}{Q(x)}$$

(7.2)

where $c$ is an appropriately chosen constant. Solving equations (7.1) and (7.2) for $Q(x)$ gives

$$Q(x) = (1 - x)^L + cx^L$$

(7.3)

and

$$F(x) = \frac{(1 - x)^L}{(1 - x)^L + cx^L}.$$  

(7.4)

Note that $|M(\pi/2)|^2 = F(1/2) = \frac{1}{1+c}$. Clearly, $c$ should be chosen so that this value lies between 0 and 1. Therefore, $c$ should be chosen to be greater than zero. Notice that when $L$ is odd and $c$ is chosen to be 1, the degree of $Q(x)$ is decreased by 1 because the leading terms of $Q(x)$ cancel — in this case the number of nontrivial poles becomes $L - 1$.

To choose $c$ to achieve a specified half-magnitude frequency is straightforward. Let $\omega_o$ denote the specified half-magnitude frequency (the frequency at which $|M(\omega_o)|$ equals $\frac{1}{2}$). The equation $|M(\omega_o)| = \frac{1}{2}$ becomes $F(x_o) = \frac{1}{4}$ where $x_o = \frac{1}{2}(1 - \cos \omega_o)$. Solving this equation for $c$, one obtains

$$c = 3\frac{(1 - x_o)^L}{x_o^L}.$$  

(7.5)

For the classical digital Butterworth filter, $\omega_o$ can be chosen to be any value in $(0, \pi)$.

### 7.5.2 First Generalization

The first generalization of the classical digital Butterworth filter has more zeros than poles and, as in the classical case, all the zeros lie at $x = 1$ ($z = -1$).
Let $L$ denote the number of zeros at $x = 1$ and let $N$ denote the number of poles with $L \geq N$. Then, as above,

$$F(x) = \frac{P(x)}{Q(x)} = \frac{(1 - x)^L}{Q(x)} \quad (7.6)$$

where $Q(x)$ has degree $N$. But

$$F(x) - 1 = \frac{P(x) - Q(x)}{Q(x)} = -\frac{x^N U(x)}{Q(x)} \quad (7.7)$$

where $U(x)$ is a polynomial of degree at most $L - N$. (The degree of $x^N U(x)$ cannot exceed the degree of $P(x) - Q(x)$). Solving equations (7.6) and (7.7) for $Q(x)$ gives

$$Q(x) = (1 - x)^L + x^N U(x). \quad (7.8)$$

Since $Q(x)$ has degree $N$ and since $N$ is no greater than $L$, $Q(x)$ must equal the polynomial obtained by taking only the first $N + 1$ coefficients of $(1 - x)^L + x^N U(x)$. Notice that $U(x)$ can always be chosen so that the remaining coefficients of this polynomial are zero. Using the identity, $(1 - x)^L = \sum_{i=0}^{L} \binom{L}{i} (-x)^i$, $Q(x)$ can be written as

$$Q(x) = \sum_{i=0}^{N} \binom{L}{i} (-x)^i + c x^N. \quad (7.9)$$

Introducing the notation $\mathcal{T}_N$ for polynomial truncation (discarding all terms beyond the $N^{th}$ power), $Q(x)$ can be written as

$$Q(x) = \mathcal{T}_N \{(1 - x)^L\} + c x^N \quad (7.10)$$

and $F(x)$ can be written as

$$F(x) = \frac{(1 - x)^L}{\mathcal{T}_N \{(1 - x)^L\} + c x^N}. \quad (7.11)$$

The constant term of $U(x)$, $c$, becomes the free parameter that, as in the classical case, must be chosen to lie within an appropriate range. The allowable ranges for $c$ are given in table 7.4. When $c$ is chosen to lie in the ranges shown in the table, then $0 < F(x) < 1$ for $x \in (0,1)$. 

Table 7.4: Permissible ranges for $c$ for the first generalization.

<table>
<thead>
<tr>
<th>$N$ even</th>
<th>$c \geq 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$ odd</td>
<td>$c \geq \frac{L-1}{N}$</td>
</tr>
</tbody>
</table>

It turns out that when $N$ is odd, the degree of the denominator can be reduced by 1 if $c$ is chosen to be $\frac{L}{N}$.

To choose $c$ to achieve a specified half-magnitude frequency is straightforward. Let $\omega_o$ denote the specified half-magnitude frequency and let $x_o = \frac{1}{2}(1 - \cos \omega_o)$. Solving the equation $F(x_o) = \frac{1}{4}$ for $c$ yields

$$c = \frac{4(1 - x_o)^L - \mathcal{T}_N\{1 - x\}^L(x_o)}{x_o^N}.$$  \hspace{1cm} (7.12)

The value this expression gives for $c$ may not lie in the appropriate range shown in Table 7.4. If this is the case, then the specified half-magnitude frequency is too high for the current choice of $L$ and $N$. It should be noted that, although the passband can be made arbitrarily narrow, it cannot be made arbitrarily wide for a fixed $L$ and $N$ ($L > N$).

The greatest half-magnitude frequency achievable for a fixed $L$ and $N$ can be found by setting $c$ equal to the appropriate value shown in Table 7.4 and solving (7.12) for $x_o$. It is seen that $x_o$ is a root of an $L$ degree polynomial:

$$\mathcal{T}_N\{(1 - x)^L\} + cx^N - 4(1 - x)^L = 0.$$ \hspace{1cm} (7.13)

Consequently, $x_o$ can be found by using a polynomial root finder. All roots of this polynomial which are not real and in the interval (0,1) can be discarded. As shown in the appendix, it turns out that there will be only one real root in (0,1). We also found that the real roots have the properties listed in Table 7.5. The maximum achievable half-magnitude frequency is given by $\omega_o = \arccos(1 - 2x_o)$. 
Table 7.5: The number and locations of the real roots of $J_N\{(1-x)^L\} + cx^N - 4(1-x)^L$ for $L > N > 0$.

<table>
<thead>
<tr>
<th></th>
<th>$L$ even</th>
<th>$L$ odd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$ even</td>
<td>2 real roots: $x_1 \in (0, 1), x_2 &gt; 1$</td>
<td>1 real roots: $x_1 \in (0, 1)$</td>
</tr>
<tr>
<td>$c = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$ odd</td>
<td>2 real roots: $x_1 \in (0, 1), x_2 = 1$</td>
<td>3 real roots: $x_1 \in (0, 1), x_2 = 1, x_3 &gt; 1$</td>
</tr>
<tr>
<td>$c = \binom{L-1}{N}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 3: For $L = 6$ and $N = 4$, the boundary value for $c$ from table 7.4 is 0, so the polynomial equation (7.13) becomes:

$$J_4\{(1-x)^6\} - 4(1-x)^6 = 0. \quad (7.14)$$

Its roots are: 3.9476, 0.3798 ± 1.1659j, 0.4262 ± 0.3245j, 0.4404. Therefore, for this choice of $L$ and $N$, the interval of $x_0$ is $(0, 0.4404]$ — the interval of $\omega_0$ is $(0, 0.4620\pi]$.

To obtain filters having wider passbands with the same number of zeros and (nontrivial) poles, it is necessary to move at least one zero from $x = 1 \ (x = -1)$ to the passband. Maximally flat filters having passband zeros, the second generalization of the classical digital Butterworth filter, are discussed in the next section. (Passband zeros are so named here, because they contribute to the flatness of the frequency response at $\omega = 0$).

7.5.3 Second Generalization

The second generalization of the classical digital Butterworth filter uses additional zeros lying off the unit circle. These zeros are used here to obtain a higher degree of flatness at $\omega = 0$. Such a filter is shown in Figure 7.3. The filters described below possess a degree of flatness of $M + N + 1$ at $\omega = 0$, and a degree of flatness of $L$ at $\omega = \pi$. 
Let the degree of \( P(x) \) be \( L + M \), and let the degree of \( Q(x) \) be \( N \), with \( L > N \) and \( M > 0 \). The \( M \) zeros off the unit circle are used to increase the degree of flatness at \( x = 0 \). This leads to the following pair of equations:

\[
F(x) = \frac{P(x)}{Q(x)} = \frac{(1 - x)^L S(x)}{Q(x)}
\]  
\[ (7.15) \]

\[
F(x) - 1 = \frac{P(x) - Q(x)}{Q(x)} = -\frac{x^{M+N+1} U(x)}{Q(x)}
\]  
\[ (7.16) \]

where \( S(x) \) is a polynomial of degree at most \( M \) and where \( U(x) \) is a polynomial of degree at most \( L - N - 1 \). Solving equations (7.15) and (7.16) for \( Q(x) \) gives

\[
Q(x) = (1 - x)^L S(x) + x^{M+N+1} U(x).
\]  
\[ (7.17) \]

Following the same reasoning as above, since \( Q(x) \) is a polynomial of degree \( N \), \( Q(x) \) must be the polynomial obtained by taking only the first \( N + 1 \) coefficients of \( (1 - x)^L S(x) - x^{M+N+1} U(x) \). However, the coefficients of \( S(x) \) and \( U(x) \) must be determined so that the remaining higher power coefficients of this polynomial \( Q(x) \) are zero. Since \( U(x) \) can be chosen so that the last \( L - N \) coefficients of \( Q(x) \) are zero, it remains to choose the coefficients of \( S(x) \) such that the coefficients of \( Q(x) \) of powers \( N + 1 \) thru \( M + N \) are zero. Let \( S(x) = s_0 + s_1 x + \cdots + s_M x^M \). Since \( S(x) \) can be scaled by a constant without changing the function \( F(x) \), there are \( M \) degrees of freedom. Note that there are \( M \) coefficients of \( Q(x) \) that must be set equal to zero. This gives rise to a system of \( M \) linear equations. It is expected that these equations become ill conditioned, because they effectively involve the specification of many derivatives of a polynomial at a single point. A closed form solution for \( S(x) \) was found to be given by the following:

\[
S(x) = \sum_{k=0}^{M} \binom{M + N - k}{N} \binom{L - N + k - 1}{k} x^k
\]  
\[ (7.18) \]

where \( \binom{n}{k} \) is a binomial coefficient. Using the polynomial \( S(x) \) given by this expression, \( P(x) \) is given by \( (1 - x)^L S(x) \). \( Q(x) \) can be found by simply taking the first
$N + 1$ coefficients of $P(x)$. Using the polynomial truncation notation, one has

$$Q(x) = \mathcal{T}_N[(1 - x)^L S(x)] .$$

(7.19)

It turns out that $(1 - x)^L S(x)$ can be written as

$$(1 - x)^L S(x) = \sum_{k=0}^{M+L} \binom{M + N - k}{M} \binom{M + L}{k} (-x)^k .$$

(7.20)

To evaluate $\binom{n}{k}$ for negative values of $n$ we use the following convention [85]:

$$\binom{n + k - 1}{k} = (-1)^k \binom{-n}{k}$$

(7.21)

for $k \geq 0$. In addition, note that $\binom{n}{k} = 0$ for $n \geq 0$, $k < 0$; and that $\binom{n}{k} = 0$ for $n \geq 0$, $k > n$.

In contrast to the classical digital Butterworth filter and its first generalization above, there is no extra degree of freedom in equations (7.18) and (7.19). The exact location of the half-magnitude frequency is dictated by the parameters $L$, $M$ and $N$. Its location can be only approximately positioned. The filters given by (7.18) share this property with the maximally-flat FIR filters given in [38,93].

**Example 4:** Fixing $L + M$ at 22 and $N = 4$, the frequency response magnitudes of the filters obtained using (7.18) for $L = 5, \ldots, 21$, $M = 22 - L$ are shown in Figure 7.5. In order to obtain filters having transition bands that lie between those shown in Figure 7.5, a degree of flatness at either $x = 0$ or $x = 1$ can be given up and the extra degree of freedom can be used to more accurately position the location of the transition band.

When a degree of flatness is given up, the degree of flatness at $x = 1$ becomes $M + N$, rather than $M + N + 1$ as in (7.16). In this case, the numerator $P(x)$ and denominator $Q(x)$ of $F(x)$ are given by the following:

$$P(x) = (1 - x)^L (R(x) + cT(x))$$

(7.22)

$$Q(x) = \mathcal{T}_N\{ P(x) \}$$

(7.23)
Figure 7.5: A sequence of digital Butterworth filters. \( L = 5, \ldots, 21, M = 22 - L, N = 4 \). The widest band filter corresponds to \( L = 5 \).

where

\[
R(x) = \sum_{k=0}^{M-1} \binom{M+N-k-1}{N} \binom{L-N+k-1}{k} x^k. \tag{7.24}
\]

and

\[
T(x) = x \sum_{k=0}^{M-1} \binom{M+N-k-2}{N-1} \binom{L-N+k}{k} x^k. \tag{7.25}
\]

It turns out that \((1 - x)^L R(x)\) can be written as

\[
(1 - x)^L R(x) = \sum_{k=0}^{M+L} \binom{M+N-k-1}{M-1} \binom{M+L-1}{k} (-x)^k. \tag{7.26}
\]

and that \((1 - x)^L T(x)\) can be written as

\[
(1 - x)^L T(x) = -\sum_{k=0}^{M+L} \binom{M+N-k-1}{M-1} \binom{M+L-1}{k-1} (-x)^k. \tag{7.27}
\]

The free parameter here is the constant \( c \). If so desired, it can be chosen to more accurately position the location of the transition band (at the expense of a degree of flatness at \( x = 0 \) or \( x = 1 \)). The variable \( c \) must be chosen to lie in the ranges shown in table 7.6. When \( N \) is even, the positive endpoint of this interval is that
Table 7.6: Permissible ranges for $c$ for the second generalization.

<table>
<thead>
<tr>
<th>$N$ even</th>
<th>$-1 \leq c \leq \frac{L - N}{M + N}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$ odd</td>
<td>$\frac{L - N}{N} \leq c$</td>
</tr>
</tbody>
</table>

Point beyond which $F(x)$ is no longer monotonic - and the negative endpoint of this interval is that point beyond which $F(x)$ is no longer nonnegative.

To choose $c$ to achieve a specified half-magnitude frequency, note that $F(x)$ can be written as

$$F(x) = \frac{(1 - x)^L R(x) + c(1 - x)^L T(x)}{\mathcal{J}_N((1 - x)^L R(x)) + c \mathcal{J}_N((1 - x)^L T(x))}.$$  

(7.28)

As above, let $\omega_o$ denote the specified half-magnitude frequency and let $x_o = \frac{1}{2}(1 - \cos \omega_o)$. Solving the equation $F(x_o) = \frac{1}{4}$ for $c$ yields

$$c = \frac{4(1 - x_o)^L R(x_o) - \mathcal{J}_N((1 - x)^L R(x))(x_o)}{\mathcal{J}_N((1 - x)^L T(x))(x_o) - 4(1 - x_o)^L T(x_o)}.$$  

(7.29)

The value this expression gives for $c$ may not lie in the appropriate range given by table 7.6. If this is the case, then the specified half-magnitude frequency is either too high or too low for the current choice of $L$, $M$ and $N$ — it is necessary to alter the distribution of zeros between $x = 1$ ($x = -1$) and the passband.

For fixed $L$, $M$, and $N$, the minimum and maximum permissible values of the half-magnitude frequency $\omega_o$ can be computed by (i) setting $c$ to the values in table 7.6, (ii) solving (7.29) for $x$ and (iii) using $\omega = \arccos (1 - 2x)$. When $c$ is finite, it is seen that $x$ is a root of the $L + M$ degree polynomial:

$$\mathcal{J}_N((1 - x)^L (R(x) + cT(x))) - 4(1 - x)^L (R(x) + cT(x)) = 0.$$  

(7.30)

Note that when $N$ is odd, $c$ can be chosen to be arbitrarily large. Letting $c$ approach
Table 7.7: The number and locations of the real roots of the polynomials used to compute the minimum and maximum permissible values of the half-magnitude frequency \( \omega_o \) for a fixed \( L, M, \) and \( N, \) with \( L > N > 1, M > 1. \) 'e' denotes 'even', 'o' denotes 'odd'.

<table>
<thead>
<tr>
<th>L</th>
<th>M</th>
<th>N</th>
<th>Num. and loc. of real roots ( (\omega_{\text{min}}) )</th>
<th>Num. and loc. of real roots ( (\omega_{\text{max}}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>e</td>
<td>e</td>
<td>2: ( x_1 \in (0,1), x_2 &lt; 0 )</td>
<td>2: ( x_1 \in (0,1), x_2 &gt; 1 )</td>
</tr>
<tr>
<td>o</td>
<td>e</td>
<td>e</td>
<td>1: ( x_1 \in (0,1) )</td>
<td>3: ( x_1 \in (0,1), x_2 &lt; 0, x_3 &gt; 1 )</td>
</tr>
<tr>
<td>e</td>
<td>o</td>
<td>e</td>
<td>3: ( x_1 \in (0,1), x_2 &lt; 0, x_3 &gt; 1 )</td>
<td>1: ( x_1 \in (0,1) )</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>e</td>
<td>2: ( x_1 \in (0,1), x_2 &gt; 1 )</td>
<td>2: ( x_1 \in (0,1), x_2 &lt; 0 )</td>
</tr>
<tr>
<td>e</td>
<td>e</td>
<td>o</td>
<td>4: ( x_1 \in (0,1), x_2 = 0, x_3 &lt; 0, x_4 &gt; 1 )</td>
<td>2: ( x_1 \in (0,1), x_2 = 1 )</td>
</tr>
<tr>
<td>o</td>
<td>e</td>
<td>o</td>
<td>3: ( x_1 \in (0,1), x_2 = 0, x_3 &gt; 1 )</td>
<td>3: ( x_1 \in (0,1), x_2 = 1, x_3 &lt; 0 )</td>
</tr>
<tr>
<td>e</td>
<td>o</td>
<td>o</td>
<td>3: ( x_1 \in (0,1), x_2 = 0, x_3 &lt; 0 )</td>
<td>3: ( x_1 \in (0,1), x_2 = 1, x_3 &gt; 1 )</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>o</td>
<td>2: ( x_1 \in (0,1), x_2 = 0 )</td>
<td>4: ( x_1 \in (0,1), x_2 = 1, x_3 &lt; 0, x_4 &gt; 1 )</td>
</tr>
</tbody>
</table>

Infinity, the expression for \( F(x) \) in (7.28) approaches

\[
F(x) = \frac{(1 - x)^L T(x)}{\mathcal{J}_N((1 - x)^L T(x))}
\]  

(7.31)

in which case the appropriate polynomial equation becomes:

\[
\mathcal{J}_N((1 - x)^L T(x)) - 4(1 - x)^L T(x) = 0.
\]  

(7.32)

Therefore, for both even and odd \( N, \) the range of achievable half-magnitude frequencies can be found by computing the roots of the appropriate pair of polynomials. In all the examples we examined, each polynomial has exactly one real root in the interval \((0,1).\) It is this root that is used to compute the interval permissible values of \( \omega_o \) - because \( \omega_o \) must lie in \((0, \pi).\) The number and the location of the real roots of the polynomials used to compute this minimum and maximum are given in table 7.7.

It turns out that for a specified half-magnitude frequency and for a specified number of total zeros \( (L+M) \) and (nontrivial) poles \( (N) \), there is exactly one choice of \( L \) and \( M \) for which the specified half-magnitude is achievable. The appropriate
\( L \) and \( M \) can be found systematically by finding the roots of the appropriate set of polynomials described in the preceding paragraphs. Also, because it is known that the root sought for each case is the real one in the interval \((0,1)\), a general polynomial root finder is unnecessary — a more efficient program can be written that computes only the desired root.

### 7.6 Further Remarks

By using the formulas above, a program can be written that requires from the user the three parameters: the number of poles \((N)\), the total number of zeros \((L + M)\), and the half-magnitude frequency \((\omega_o)\). By tabulating a table such as table 7.2, the appropriate way to split the number of zeros between \(z = -1\) and the passband \((L \) and \( M)\) can be determined. The corresponding formula can then be used to compute \(F(z)\). After a transformation and spectral factorization (which can be carried out by transforming the roots), the filter coefficients are obtained. A Matlab program is reproduced and discussed in the appendix to this chapter (section 7.10) and is available on the Internet. Table 7.8 gives a summary of the filter design formulas. Table 7.9 gives auxiliary polynomials.

Note that, if desired, a frequency other than the half-magnitude frequency can be specified. To specify a frequency \(\omega_o\) for which \(M(\omega) = M_o\) is possible for any \(M_o\), \(0 < M_o < 1\). The resulting design formulas differ only in that they contain slightly different constants.

#### 7.6.1 Behavior for Odd \(N\)

Note that when \(N\) is odd, one of the poles must lie on the real line. When there are zeros that lie off the unit circle, in the passband \((M > 0)\), it is expected that the pole lying on the real line does little to contribute to the performance of the frequency response. This is indeed true. In some situations, a pole and a zero will lie on the unit circle and, depending on the specified half-magnitude frequency, almost cancel.
For this reason, it is expected that generalized digital Butterworth filters having and odd number of poles with passband zeros will be of little interest — they have been presented in this chapter for completeness.

7.6.2 A Note on Implementation

Note that the numerator can be implemented as a cascade of two FIR sections, one of which has all its zeros at \( z = -1 \). It can be beneficial to implement these sections separately, because the FIR section with all its zeros at \( z = -1 \) can be implemented without the use of multiplications. Since the number of zeros at \( z = -1 \) depends on the specified half-magnitude frequency, the number of multiplications is dependent on this parameter. A lowpass filter with a relatively narrow passband will have more zeros at \( z = -1 \) than will a lowpass filter with a relatively wide passband — and it will thus appear to require fewer multiplications. Note that in the design of odd length linear phase FIR filters, this can be overcome by implementing a wideband lowpass filter as a sum of narrowband filter and a pure delay. It is unclear if an analogous technique exists for the minimum phase IIR filters described in this chapter.

7.6.3 Butterworth Filters Having More Poles Than Zeros

Although the focus of this chapter has been on digital Butterworth filters having more zeros than poles, it should be noted that the design of Butterworth filters having more poles than zeros can also be easily carried out. For such filters, all the zeros will always lie at \( z = -1 \), and the formula for \( F(x) \) is very similar to that of (7.4). If \( L \) is the number of (nontrivial) zeros and \( N \) is the number of poles \( (N > L) \), \( F(x) \) is given by

\[
F(x) = \frac{(1 - x)^L}{(1 - x)^L + cx^N}.
\]  

(7.33)

To choose \( c \) to achieve a desired half-magnitude frequency \( \omega_o \), the following formula can be used:

\[
c = 3 \frac{(1 - x_o)^L}{x_o^N}
\]  

(7.34)
where $x_o = \frac{1}{2}(1 - \cos \omega_o)$. Like the classical case, $\omega_o$ can be chosen to be any value in $(0, \pi)$.

### 7.6.4 FIR Butterworth Filters

The design of FIR digital Butterworth filters having a specified half-magnitude frequency can also be carried out using the formulas given above. But in the FIR case, the formulas of Herrmann [38] can be used, even for half-magnitude frequencies that lie between those of the maximally-flat filters described by Herrmann: Recall that the formulas described by Herrmann produce a discrete set of filters, and consequently only a discrete set of half-magnitude frequencies can be obtained for a fixed filter length. However, as described in [44, 111], by appropriately averaging two neighboring Herrmann filters, maximally-flat filters between the two "adjacent" filters can be obtained. Therefore, Herrmann's formulas can be used to obtain maximally-flat FIR filters having specified half-magnitude frequencies exactly. As above, the use of this technique means giving up a single degree of flatness at either $\omega = 0$ or $\omega = \pi$ in order to achieve the specified half-magnitude frequency exactly. In the FIR case, the filters obtained from the formulas in this chapter are the same as the filters that can be obtained using Herrmann's formulas.

One difference between the FIR and IIR cases is that in the FIR case there is a maximum and minimum achievable passband width. In other words, the half-magnitude frequency can not be made arbitrarily close to 0 or $\pi$. This is one situation in which the use of a single pole could be useful — by using a single pole, the specified half-magnitude frequency can be chosen to be any value in $(0, \pi)$.

### 7.7 Conclusion

The design of generalized classical digital Butterworth filters can be carried out without the need to solve ill conditioned equations. By using appropriate formulas and a transformation, and by taking a spectral factor, maximally flat IIR filters having
Table 7.8 : The expression for \( F(x) \) gives the magnitude squared function in the \( x \) domain in terms of a constant \( c \). When \( c \) is chosen according to the expression given in the table, \( F(x_o) \) equals 1/4.

<table>
<thead>
<tr>
<th>( F(x) )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{(1 - x)^L}{(1 - x)^L + cx^L} )</td>
<td>( \frac{3(1 - x_o)^L}{x_o^L} )</td>
</tr>
<tr>
<td>( \frac{(1 - x)^L}{\mathcal{J}_N((1 - x)^L) + cx^N} )</td>
<td>( \frac{4(1 - x_o)^L - \mathcal{J}_N((1 - x)^L)(x_o)}{x_o^N} )</td>
</tr>
<tr>
<td>( \frac{(1 - x)^L(R(x) + cT(x))}{\mathcal{J}_N((1 - x)^L(R(x) + cT(x)))} )</td>
<td>( \frac{4(1 - x_o)^L R(x_o) - \mathcal{J}_N((1 - x)^L R(x))(x_o)}{\mathcal{J}_N((1 - x)^L T(x))(x_o) - 4(1 - x_o)^L T(x_o)} )</td>
</tr>
<tr>
<td>( \frac{(1 - x)^L S(x)}{\mathcal{J}_N((1 - x)^L S(x))} )</td>
<td>---</td>
</tr>
</tbody>
</table>

Table 7.9 : Auxiliary polynomials.

\[
S(x) = \sum_{k=0}^{M} \binom{M + N - k}{N} \binom{L - N + k - 1}{k} x^k
\]

\[
R(x) = \sum_{k=0}^{M-1} \binom{M + N - k - 1}{N} \binom{L - N + k - 1}{k} x^k
\]

\[
T(x) = x \sum_{k=0}^{M-1} \binom{M + N - k - 2}{N - 1} \binom{L - N + k}{k} x^k
\]
more zeros than (nontrivial) poles can be easily designed — and without the restriction that all zeros lie on the unit circle. In addition, for a fixed number of zeros and a fixed number of (nontrivial) poles, the formulas above give a direct way of finding the correct way to split the number of zeros between $z = -1$ and the passband.

The maximally-flat FIR filters described by Herrmann [38] and the classical Butterworth filter are special cases of the filters given by the formulas described in this chapter.

Appendices

7.8 Proofs

In this section the statements regarding the first generalization are proven. It will be useful to note a few properties. The first is a recursive relation on the truncation of the polynomial $(1 - x)^{L}$:

$$
\mathcal{J}_{N}[(1 - x)^{L}] = (1 - x)\mathcal{J}_{N-1}[(1 - x)^{L-1}] + (-1)^{N} \binom{L-1}{N} x^{N}.
$$

(7.35)

The next property concerns the derivative of a truncated polynomial: For any polynomial $G(x)$,

$$
\frac{\partial \mathcal{J}_{N}[G(x)]}{\partial x} = \mathcal{J}_{N-1} \left\{ \frac{\partial G(x)}{\partial x} \right\}.
$$

(7.36)

A third property will also be very useful:

Identity: For $L \geq N \geq 0$

$$
\mathcal{J}_{N}[(1 - x)^{L}] = (1 - x)^{L} - (-x)^{N+1} \sum_{i=0}^{L-N-1} \binom{L - i - 1}{N} (1 - x)^{i}.
$$

(7.37)

Proof: Rewrite (7.37) as

$$
(-1)^{N} \left[ \mathcal{J}_{N}[(1 - x)^{L}] - (1 - x)^{L} \right] = x^{N+1} \sum_{i=0}^{L-N-1} \binom{L - i - 1}{N} (1 - x)^{i}.
$$

(7.38)
Using the binomial theorem and collecting like terms, the left hand side of (7.38) can be written as

\[ \text{LHS} = x^{N+1} \sum_{k=0}^{L-N-1} \binom{L}{N+k+1} (-x)^k. \]

The right hand side of (7.38) can be rewritten in the following way:

\[ \text{RHS} = x^{N+1} \sum_{i=0}^{L-N-1} \binom{L-i-1}{N} (1-x)^i, \]

\[ = x^{N+1} \sum_{i=0}^{L-N-1} \binom{L-i-1}{N} \sum_{k=0}^{i} \binom{i}{k} (-x)^k. \]

\[ = x^{N+1} \sum_{k=0}^{L-N-1} \sum_{i=k}^{L-N-1} \binom{L-i-1}{N} \binom{i}{k} (-x)^k. \]

(7.39) (7.40) (7.41)

To show that the left and right hand sides are equal, it remains to show that corresponding coefficients of like powers of \( x \) are equal: For \( 0 \leq k \leq L - N - 1 \) and \( L > N \), we need

\[ \binom{L}{N+k+1} = \sum_{i=k}^{L-N-1} \binom{L-i-1}{N} \binom{i}{k}, \]

(7.42)

which follows from a well-known binomial identity [113].

\[ \square \]

7.8.1 The First Generalization

We begin by showing that if \( c \) is chosen according to table 7.4 and \( F(x) \) is given by equation (7.11) for \( L > N \), then \( 0 < F(x) < 1 \) for \( x \in (0,1) \). Since the numerator of \( F(x) \) is positive over \((0,1)\), it will be necessary to show that the denominator \( Q(x) \), given by

\[ Q(x) = T_N \{ (1 - x)^L \} + cx^N, \]

(7.43)

is also positive over \((0,1)\). To show that \( F(x) < 1 \) over \((0,1)\), note that \( F(x) < 1 \) is equivalent to \( F(x) - 1 < 0 \) or, using (7.8), \( U(x) > 0 \) over \((0,1)\). Note that \( U(x) \) is chosen to cancel the coefficients of \((1 - x)^L\) for powers \( N + 1 \) thru \( L \) so that \( Q(x) \) has
degree \( N \). That is, \( U(x) \) is given by \( c + [\mathcal{J}_N\{(1-x)^L\} - (1-x)^L]/(x^N). \) To summarize this paragraph, we will show that \( 0 < F(x) < 1 \) for \( x \in (0,1) \) by showing that \( Q(x) \) and \( U(x) \) are both positive over \((0,1)\). We will consider the two cases, \( N \) even and \( N \) odd, separately.

\( Q(x) > 0, N \) even:

To show that \( Q(x) \) is positive over \((0,1)\) when \( c > 0 \) and \( N \) is even, note that (7.37) gives

\[
\mathcal{J}_N\{(1-x)^L\} = (1-x)^L + x^{N+1} \sum_{i=0}^{L-N-1} \binom{L-i-1}{N}(1-x)^i \tag{7.44}
\]

for even \( N \). Since \( x^k(1-x)^i \) is positive over \((0,1)\), and since each of the binomial coefficients in the sum is positive, this shows that \( \mathcal{J}_N\{(1-x)^L\} \) is positive over \((0,1)\) for even \( N \). Therefore, when \( c > 0 \) and \( N \) is even, we have \( Q(x) > 0 \) over \((0,1)\).

\( U(x) > 0, N \) even:

To show that \( U(x) \) is positive over \((0,1)\) when \( c > 0 \) and \( N \) is even, write \( U(x) \) as

\[
U(x) = c + [\mathcal{J}_N\{(1-x)^L\} - (1-x)^L]/(x^N). \tag{7.45}
\]

We can use (7.44) to write

\[
U(x) = c + x \sum_{i=0}^{L-N-1} \binom{L-i-1}{N}(1-x)^i \tag{7.46}
\]

for even \( N \). Again, since the rightmost term is a positively weighted sum of polynomials positive over \((0,1)\), we have the result \( U(x) > 0 \) over \((0,1)\) when \( c > 0 \) and \( N \) is even.

\( Q(x) > 0, N \) odd:

To show that \( Q(x) \) is positive over \((0,1)\) when \( c > \binom{L-1}{N} \) and \( N \) is odd, note that (7.35) gives

\[
\mathcal{J}_N\{(1-x)^L\} + \binom{L-1}{N}x^N = (1-x)\mathcal{J}_{N-1}\{(1-x)^{L-1}\} \tag{7.47}
\]
for odd $N$. Using (7.44) it follows that

$$J_N[(1-x)^L] + \left(\frac{L-1}{N}\right)x^N =$$

$$\left(1-x\right)\left(1-x)^{L-1} + x^N \sum_{i=0}^{L-N-1} \left(\frac{L-i-2}{N-1}\right)(1-x)^i\right)$$

(7.49)

for odd $N$. Since the right hand side is a product of two polynomials positive over $(0,1)$, we have the result $Q(x) > 0$ over $(0,1)$ when $c > \left(\frac{L-1}{N}\right)$ and $N$ is odd.

$U(x) > 0$, $N$ odd:

To show that $U(x)$ is positive over $(0,1)$ when $c > \left(\frac{L-1}{N}\right)$ and $N$ is odd, write $U(x)$ as

$$U(x) = c + [J_N[(1-x)^L] - (1-x)^L]/(x^N).$$

(7.50)

We can use (7.48) to write

$$U(x) = c - \left(\frac{L-1}{N}\right) + (1-x) \sum_{i=0}^{L-N-1} \left(\frac{L-i-2}{N-1}\right)(1-x)^i$$

(7.51)

for odd $N$. Again, since the rightmost term is positive over $(0,1)$ we have the result $U(x) > 0$ over $(0,1)$ when $c > \left(\frac{L-1}{N}\right)$ and $N$ is odd.

We will next prove that the polynomial of (7.13) has exactly one real root in $(0,1)$ for $L > N$ when (i) $N$ is even and $c$ is chosen to equal $0$ or (ii) $N$ is odd and $c$ is chosen to equal $\left(\frac{L-1}{N}\right)$.

Exactly one real root in $(0,1)$, $N$ even:

Let us define $G_e(x)$ as $G_e(x) = J_N[(1-x)^L] - 4(1-x)^L$. Take $N$ to be even for the following discussion. To show that $G_e(x)$ has exactly one real root in $(0,1)$, first note that $G_e(0) = -3$ and $G_e(1) = \left(\frac{L-1}{N}\right)$. Therefore, $G_e(x)$ must have at least one root in $(0,1)$. Next, use (7.36) to write the derivative of $G_e(x)$ as

$$G'_e(x) = 4L(1-x)^{L-1} - L J_{N-1}[(1-x)^{L-1}].$$

(7.52)

Using (7.37), $G'_e(x)$ can be written as

$$G'_e(x) = 3L(1-x)^{L-1} + L x^N \sum_{i=0}^{L-N-1} \left(\frac{L-i-2}{N}\right)(1-x)^i$$

(7.53)
for even $N$. Therefore, $G_e^*(x)$ is positive over $(0,1)$ — and thus $G_e(x)$ can have no more than one root in $(0,1)$. Therefore, $G_e(x)$ has exactly one root in $(0,1)$ for even $N$.

Exactly one real root in $(0,1)$, $N$ odd:

Let us define $G_o(x)$ as $G_o(x) = \mathcal{T}_N\{(1-x)^L\} + \binom{L-1}{N}x^N - 4(1-x)^L$. Use (7.35) to write $G_o(x)$ as

$$G_o(x) = (1-x)\mathcal{T}_{N-1}\{(1-x)^{L-1}\} - 4(1-x)^L$$  \hspace{1cm} (7.54)

for odd $N$. It will be convenient to show that $G_o(x)$ has exactly one root in $(0,1)$ for odd $N$ by showing that $G_o(x)/(1-x)$ has exactly one root in $(0,1)$ for odd $N$. Note that

$$\frac{G_o(x)}{1-x} = \mathcal{T}_{N-1}\{(1-x)^{L-1}\} - 4(1-x)^{L-1}.$$  \hspace{1cm} (7.55)

Compare (7.55) with $G_e(x)$ above. Because $G_e(x)$ has exactly one root in $(0,1)$ for even $N$, $G_o(x)$ has exactly one root in $(0,1)$ for odd $N$.

7.8.2 The Second Generalization

The proofs for the second generalization are similar to those for the first generalization — and will not be given. However, some of the relevant identities are listed below, and it is shown that $0 < F(x)$ when $N$ is even and $c$ is in the interval stated in table 7.6.

The following two identities show that when $N$ is even, the truncated polynomials $\mathcal{T}_N\{(1-x)^L S(x)\}$ and $\mathcal{T}_N\{(1-x)^L R(x)\}$ are positive over $(0,1)$.

Identity: For $L \geq N \geq 0$, $M \geq 0$,

$$\mathcal{T}_N\{(1-x)^L S(x)\} =$$

$$(1-x)^L S(x) + (-1)^N x^{M+N+1} \sum_{i=0}^{L-N-1} \binom{M+i}{M} \binom{L-i-1}{N} (1-x)^i$$  \hspace{1cm} (7.56)

where $S(x)$ is given in (7.18).
Identity: For \( L \geq N > 0, M > 0 \),

\[
\mathcal{T}_N\{(1 - x)^L R(x)\} =
\]

\[
(1 - x)^L R(x) + (-1)^N x^{M + N} \sum_{i=0}^{L-N} \binom{M + i - 1}{M - 1} \binom{L - i - 1}{N}(1 - x)^i
\]

(7.57)

where \( R(x) \) is given in (7.24).

The following identity shows that when \( N \) is odd, the polynomial \( \mathcal{T}_N\{(1 - x)^L T(x)\} - (1 - x)^L T(x) \) is positive over (0,1).

Identity: For \( L \geq N > 0, M > 0 \),

\[
\mathcal{T}_N\{(1 - x)^L T(x)\} =
\]

\[
(1 - x)^L T(x) - (-1)^N x^{M + N} \sum_{i=0}^{L-N} \binom{M + i - 1}{M - 1} \binom{L - i - 1}{N - 1}(1 - x)^i
\]

(7.58)

where \( T(x) \) is given in (7.25).

Note that \( R(x) \) and \( T(x) \) have positive coefficients, thus they are positive for all positive \( x \). In the formulas above, the term \( R(x) + cT(x) \) appears. The following two identities can be used in conjunction with the previous two identities to support the required bound on \( c \) stated above in Table 7.6.

Identity: For \( L \geq N > 0, M > 0 \),

\[
R(x) - T(x) = (1 - x) \sum_{i=0}^{M-1} \binom{M + N - i - 1}{N} \binom{L - N + i}{i} x^i.
\]

(7.59)

Therefore, \( R(x) - T(x) \) is positive for all \( x \) in (0,1). Consequently, \( R(x) + cT(x) \) is positive for all \( c > -1 \) (since \( R(x) + cT(x) = R(x) - T(x) + (1 + c)T(x) \)).

Identity: For \( L \geq N > 0, M > 0 \),

\[
\mathcal{T}_N\{(1 - x)^L (R(x) + \frac{M}{M+N} T(x))\} =
\]

\[
(1 - x)^L (R(x) + \frac{M}{M+N} T(x)) +
\]

\[
(-1)^N \frac{M}{M+N} x^{M+N+1} \sum_{i=0}^{L-N-1} \binom{M + i}{M} \binom{L - i - 1}{N}(1 - x)^i.
\]

(7.60)
Using this identity, it can be shown that when \( N \) is even and \(-1 < c < \frac{N}{N+K} \), the function \( \mathcal{I}_N \{ (1-x)^L (R(x) + cT(x)) \} \) is positive over \((0,1)\). For even \( N \):

\[
\mathcal{I}_N \{ (1-x)^L (R(x) + cT(x)) \} = \mathcal{I}_N \{ (1-x)^L (R(x) + \frac{N}{N+K} T(x)) \} + (c - \frac{N}{N+K}) \mathcal{I}_N \{ (1-x)^L T(x) \} = (1-x)^L (R(x) + \frac{N}{N+K} T(x)) + Y_1(x) + (c - \frac{N}{N+K})((1-x)^L T(x) + Y_2(x)) = (1-x)^L (R(x) + cT(x)) + Y_1(x) + (\frac{N}{N+K} - c) Y_2(x)
\]

where \( Y_1(x) \) is the right hand side of (7.60) and \( Y_2(x) \) is the right most term of (7.58). Since \( Y_1(x) \) and \( Y_2(x) \) are both positive over \((0,1)\), and since \( R(x) + cT(x) \) is positive over \((0,1)\) for \( c > -1 \), the expression above shows that \( \mathcal{I}_N \{ (1-x)^L (R(x) + cT(x)) \} \) is positive over \((0,1)\) for even \( N \) and \(-1 < c < \frac{N}{N+K} \). It follows that \( 0 < F(x) \) for even \( N \) and \(-1 < c < \frac{N}{N+K} \).

### 7.9 Connection to a Series of Gauss

The polynomials \( R(x), T(x), \) and \( S(x) \) given above, are special cases of the Gauss hypergeometric series [66], \( F(a, b; c; z) \), given by

\[
F(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!}
\]

(7.62)

where the pochhammer symbol \((a)_k\) denotes \((a) \cdot (a+1) \cdot (a+2) \cdots (a+k-1)\). Note that when \( a \) or \( b \) is a negative integer, the \( F(a, b; c; z) \) becomes a polynomial. The polynomials \( R(x), T(x), \) and \( S(x) \) can be written as

\[
S(x) = \frac{(M+1)_N}{N!} F(-M, L - N; -M - N; x)
\]

(7.63)

\[
R(x) = \frac{(M)_N}{N!} F(1 - M, L - N; 1 - M - N; x)
\]

(7.64)

\[
T(x) = \frac{(M-1)_N N!}{(N-1)!} F(2 - M, L - N - 1; 2 - M - N; x).
\]

(7.65)

There are many recurrence formulas for this function. Using these recurrence formulas, one can write recursion formulas for \( R(x), S(x), \) and \( T(x) \). For example: if
\( S_0(x) = 1, \quad S_1(x) = (L - N)/(M + N)x + 1, \) then \( S_{M-1}(x) \) computed by the recursive formula

\[
s_k(x) = \frac{1}{M + N - k} \left( ((k + L - N)x + M + N - 2k)S_{k-1}(x) + k(1 - x)S_{k-2}(x) \right)
\]

(7.66)

equals \( S(x) \) given above.

These relationships may facilitate the computation of the roots of the polynomials, as suggested in [67, 112]. There is also a differential equation the solution of which is the Gauss hypergeometric series, see [66].

### 7.10 Matlab Programs

Three Matlab programs for the design of digital IIR Butterworth filters are reproduced below. The first program, `spec_table.m`, constructs a specification table like Table 7.2. The second program, `general_butter.m`, returns the filter numerator and denominator coefficients. The third program, `choose.m`, provides binomial coefficients (at the time of this writing Matlab does not include a binomial coefficient function).

The program `spec_table.m` proceeds by constructing one row of the specification table at a time. The variable \( F_0 \) represents the "half-magnitude" squared (one quarter). If it is desired that a frequency other than the half-magnitude frequency be specified, then \( F_0 \) can be modified — no other modifications of the programs are needed. For example, to specify the half-power point, \( F_0 \) should be changed to one half. The vector \( Y \) represents the coefficients of the polynomials given in Expressions 7.13 and 7.30, depending on the case. The coefficients are listed in order of decreasing powers in \( Y \) — that is the Matlab convention.

The commands \( r = \text{roots}(Y) \) and \( r = r(\text{imag}(r)==0) \) give the real roots of \( Y(x) \). To locate the single real root of \( Y(x) \) in \([0, 1]\) we simply find the real root lying closest to one half. The command \([\text{temp}, i] = \text{min(abs(r-0.5))}\) gives as \( i \)
the index of r lying closest to one half. The command \( \text{w} \text{ow} \text{.} \text{max} = \text{a} \cos(1-2\pi r(i)) \) carries out the inverse of the transformation \( x = \frac{1}{2}(1 - \cos \omega) \). The variables GR and GT represent the polynomials \((1 - x)^L R(x)\) and \((1 - x)^L T(x)\) in Equations 7.26 and 7.27. The variables AGR and AGT represent the polynomials \( \mathcal{T}_N\{(1 - x)^L R(x)\} \) and \( \mathcal{T}_N\{(1 - x)^L T(x)\} \).

In spec_table.m, the single real root of \( Y(x) \) in \([0, 1]\) is found by computing all the roots and by then selecting the root sought (as described above). It should be noted, however, that because it is known in advance that \( Y(x) \) has exactly one root in \([0, 1]\), a more efficient root finding algorithm can be used to compute this root. The approach taken in spec_table.m was motivated by the wish to use built-in functions as far as possible.

The program general_butter.m proceeds by locating the row of the specification table in which the specified half-magnitude frequency appears. The appropriate values for \( L, M, \) and \( N \) are found on that row. The polynomials \( P(x) \) and \( Q(x) \), represented by \( P \) and \( Q \) are determined by using the appropriate expression from the text in a straightforward way. The variable \( R \) and \( T \) represent the polynomials \( R(x) \) and \( T(x) \) in Expressions 7.24 and 7.25. The z-domain transfer function is obtained as in [35] by mapping the roots of \( P(x) \) and \( Q(x) \) via the transformation \( x = \frac{1}{2}(1 - \cos \omega) \): Write \( \cos \omega \) as \( \frac{1}{2}(e^{i\omega} + e^{-i\omega}) \), and in turn, as \( \frac{1}{2}(z + \frac{1}{z}) \). Solving \( z = \frac{1}{2}(1 - (\frac{1}{2}(z + \frac{1}{z})) \) for \( z \) leads to the quadratic equation

\[
z^2 + z(4x - 2) + 1 = 0
\] (7.67)

whose roots are \( z = 1 - 2x \pm \sqrt{(1 - 2x) - 1} \). To obtain the minimum phase solution, we seek those transformed roots that lie inside the unit circle. They are found in general_butter.m by sorting the transformed roots by absolute value.

\begin{verbatim}
function table = spec_table(Z,P)
% table = spec_table(Z,P);
% Constructs a specification table for
% generalized digital Butterworth filter design.
\end{verbatim}
% Z : total number of zeros
% P : total number of poles
% Example:
%    Z = 10; P = 2;
%    table = spec_table(Z,P);
% % required subprogram : choose.m
if Z <= P
    table = [Z 0 P 0 1];
    break
end
Fo = (1/2)^2;  % value of mag squared at half-mag freq.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% begin with all zeros at z = -1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = Z;
M = 0;
N = P;
if N > 0
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % construct polynomial for checking boundary %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    i = N:-1:0;
    k = L:-1:N+1;
    Y = [choose(L,k).*(-1).^k, (1-Fo)*choose(L,i).*(-1).^i];
    if rem(N,2) == 1
        Y(L+1-N) = Y(L+1-N) - choose(L-1,N)*Fo;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % extract appropriate root %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if rem(N,2) == 1
        % since it is known that in this case Y(x) has a root at
        % x=1, (see table 7.5 in text) we choose to remove it directly.
        [Y,r] = deconv(Y,[1 -1]);
    end
    r = roots(Y);
    r = r(imag(r)==0);
    [temp,i] = min(abs(r-0.5));
    wo_max = acos(1-2*r(i));
    table = [Z, 0, P, 0, wo_max/pi];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% increment number of passband %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% zeros by one in a loop %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for M = 1:Z-P-1
    L = Z-M;
    k = M+L:-1:0;
GR = choose(M+N-k-1,M-1).*choose(M+L-1,k).*(-1)."k;
GT = choose(M+N-k-1,M-1).*choose(M+L-1,k-1).*(-1)."(k-1);
AGR = GR(M+L+1-N:M+L+1);
AGT = GT(M+L+1-N:M+L+1);
%%%% --- wo_max ------------------------------- % %%
if rem(N,2) == 0
    c = (L-N)/(M+N);
else
    c = (L-N)/N;
end
Y = [zeros(1,M+L-N), Fo*AGR+c*Fo*AGT]-GR-c*GT;
%%%%%%% extract appropriate root % %%
if rem(N,2) == 1
    % since it is known that in this case Y(x) has a root at
    % x=1, (see table 7.7 in text) we choose to remove it directly.
    [Y,r] = deconv(Y,[1 -1]);
end
r = roots(Y);
if r(imag(r))==0;
    [temp,i] = min(abs(r-0.5));
    wo_max = acos(1-2*r(i));
%%%% --- wo_min ------------------------------- % %%
if N > 0
    wo_min = table(M,5)*pi;
elseif N > 1
    wo_min = table(M-1,5)*pi;
else
    % for FIR filters, wo_min on the first row of the spec
    % table is not 0, so we need to compute it.
    Y = [zeros(1,M+L-N), Fo*(AGR-AGT)]-GR+GT;
    r = roots(Y);
    if r(imag(r))==0
        [temp,i] = min(abs(r-0.5));
        wo_min = acos(1-2*r(i));
    end
    table = [table; [L,M,N, wo_min/pi, wo_max/pi]];
end
if N > 0
    table = [table; [P,Z-P,N, table(Z-P,5), 1]];
end

function [b,a,b1,b2] = general_butter(Z,P,wo)
% [b,a,b1,b2] = general_butter(Z,P,wo)
Design of digital Butterworth filters with unequal numerator and denominator degrees.

input
Z : total number of zeros
P : total number of (nontrivial) poles
wo : half-magnitude frequency in (0,pi)
output
b/a : IIR filter
b : length Z+1 vector of polynomial coefficients
a : length P+1 vector of polynomial coefficients
b = conv(b1,b2); b1 contains all zeros at z=-1,
b2 contains all other zeros.
Example
Z = 10; P = 2; wo = 0.6*pi;
[b,a,b1,b2] = general_butter(Z,P,wo);

Ivan Selesnick, Rice University, September 1995.

SM = 1e-7;
table = spec_table(Z,P);
k = max(find((table(:,4) < wo/pi+SM) & (table(:,5) > wo/pi-SM)));
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
disp(' Table: ')
disp(' ')
disp(' L   M   N   wo_min/pi wo_max/pi')
disp(' ')
disp(table)

L+M : total number of zeros
L : number of zeros at z=-1
M : number of zeros contributing to passband flatness
N : total number of poles

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
error ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if length(k) == 0
    disp(' The half-magnitude frequency, wo, must lie in ')
    disp(' the appropriate interval. See the table. ')
    break
end

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
calculate x-domain function ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
L = table(k,1);
M = table(k,2);
N = table(k,3);
xo = (1-cos(wo))/2;
Fo = (1/2)^2;

if N == 0

% FIR
k = M-1:-1:0;
R = [0, choose(M-k-1,0).*choose(L+k-1,k)];
T = [choose(M-k-2,-1).*choose(L+k,k), 0];
c = (Fo/(1-xo)*L - polyval(R,xo))/polyval(T,xo);
S = R + c*T;
Q = 1;
elseif M == 0

% all zeros at z = -1
k = min([L,N]):-1:0;
Q = choose(L,k).*(-1).^k;
if L < N
    Q = [zeros(1,N-L), Q];
end

end

c = ((1/Fo)*(1-xo)*L - polyval(Q,xo))/(xo^N);
Q(1) = Q(1) + c;
S = 1;
else % M > 0

% some zeros contribute to passband
k = M-1:-1:0;
R = [0, choose(M+N-k-1,N).*choose(L-N+k-1,k)];
T = [choose(M+N-k-2,N-1).*choose(L-N+k,k), 0];
k = N:-1:0;
AGR = choose(M+N-k-1,N-1).*choose(M+L-1,k).*(-1).^k;
AGT = choose(M+N-k-1,N-1).*choose(M+L-1,k-1).*(-1).^k;
c = (((1-xo)*L)*polyval(R,xo) - Fo*polyval(AGR,xo)) / ...
    (Fo*polyval(AGT,xo) - ((1-xo)*L)*polyval(T,xo));
S = R + c*T;
Q = AGR + c*AGT;
end

tmp = 1 - 2*roots(S);
br1 = tmp+sqrt(tmp.^2-1);
br2 = tmp-sqrt(tmp.^2-1);
\begin{verbatim}

br = sort([br1; br2]+eps*sqrt(-1));  \% sort by absolute value
br = br(1:M);  \% take roots INSIDE unit circle
b2 = real(poly(br));

tmp = 1 - 2*roots(q);
ar1 = tmp+sqrt(tmp.^2-1);
ar2 = tmp-sqrt(tmp.^2-1);
ar = sort([ar1; ar2]+eps*sqrt(-1));  \% sort by absolute value
ar = ar(1:N);  \% take roots INSIDE unit circle
a = real(poly(ar));

\% construct b1  \%---------------------------------------------------------
b1 = choose(L,0:L);

\% normalize  \%---------------------------------------------------------
b2 = b2*sum(a)/(sum(b1)*sum(b2));
b = conv(b1,b2);

function a = choose(n,k)
\% a = choose(n,k)
\% BINOMIAL COEFFICIENTS
\% allowable inputs:
\% n : integer, k : integer
\% n : integer vector, k : integer
\% n : integer, k : integer vector
\% n : integer vector, k : integer vector (of equal dimension)

nv = n;
kv = k;
if (length(nv) == 1) & (length(kv) > 1)
    nv = nv * ones(size(kv));
elseif (length(nv) > 1) & (length(kv) == 1)
    kv = kv * ones(size(nv));
end

a = nv;
for i = 1:length(nv)
    n = nv(i);
k = kv(i);
    if n >= 0
        if k >= 0
            if n >= k
                c = prod(1:n)/(prod(1:k)*prod(1:n-k));
            end
        end
    end
end

\end{verbatim}
else
    c = 0;
end
else
    c = 0;
end
else
    if k >= 0
        c = (-1)^k * prod(1:k-1)/(prod(1:k)*prod(1:n-1));
    else
        if n >= k
            c = (-1)^(n-k)*prod(1:k-1)/(prod(1:n-k)*prod(1:n-1));
        else
            c = 0;
        end
    end
end
a(i) = c;
end
Chapter 8

Generalized Chebyshev II Filter Design

8.1 Introduction

In this chapter, we describe a program for the design of IIR digital filters having flat monotonic passbands and equiripple stopbands. The approximation is performed on the magnitude squared of the frequency response. The resulting filters are analogous to the classical Chebyshev II digital IIR filters [68], but the filters obtained with the algorithm described in this chapter have numerator degrees that are greater than the denominator degree. That is, they have more nontrivial zeros than poles. The digital IIR filters obtained by transforming the classical Chebyshev II filters have an equal number of zeros and poles. Also, generalized Chebyshev I filters can be obtained from the Chebyshev II filters.

The program described in this chapter employs two distinct algorithms: a “zero-shifting” algorithm and a rational Remez-like exchange algorithm. The use of both algorithms together leads to a robust program. The algorithms complement each other well. Each program, used by itself, has short-comings that prevent it from being practical. Used together, they give a practical solution to the problem addressed in this chapter.

In this chapter we only consider filters whose denominator degrees are even. The problems associated with odd degree denominators are discussed in Chapters 7 and 9.

The problem addressed in this paper has also been addressed by Saramäki in [88]. The major difference between [88] and this chapter is the iterative numerical algorithm that is suggested. Consequently, for additional motivation and discussion
of some of the properties of the filters, we refer the reader to [88]. Another difference between [88] and this chapter lies in the way in which the passband edge is treated and in the monotonicity of the passband. In [88], both passband and stopband edges are specified, and the passband may not be monotonic; in this chapter, only the stopband edge is to be specified, and passband monotonicity is maintained. Also, [88] addresses the design of bandpass filters, which are not covered here (although it is expected that the numerical algorithm described in this chapter can be extended to the bandpass case).

Unbehauen has described a formula-based method for Chebyshev II filters having more poles than zeros for which all the zeros are constrained to lie in the stopband on the unit circle [117]. However, for some specifications, the sought filter will have zeros that contribute to the flat behavior of the passband at \( \omega = 0 \). It appears that closed form solutions for such filters cannot be obtained, and that an iterative numerical algorithm, like the one described in this chapter, is necessary.

We should note that other closed form expressions can be used for the design of IIR filters having equiripple stopbands [33,34,118]. However, as elegant and useful as these design techniques are, they require that the poles be known before hand. The problem these papers address is: given a denominator (a set of poles), find a numerator (set of zeros) that yield an equiripple stopband behavior. One property of these techniques is that all the zeros will lie on the unit circle. The solutions to the problem addressed in this chapter do not necessarily have all the zeros on the unit circle. Also, the approach taken in this chapter designs the numerator and the denominator of the transfer function together.

### 8.2 Zero-Shifting Algorithm

A "zero-shifting" algorithm was described posthumously by Maehly in [60] and is used in [117,119]. It is useful for the design of Chebyshev II filters with a specified stopband ripple size \( \delta \) and with an "extra-ripple". Indeed, as in the Parks-McClellan
algorithm, an extra ripple may appear — either scaled or of maximal size. In the extra-ripple case, all the stopband zeros can be said to contribute to a ripple. (The passband zeros and poles contribute to the flatness of the passband at \( \omega = 0 \).) As in the FIR case, as the location of the transition band is varied, the number of ripples in the stopband varies. With a fixed stopband ripple size, for only a finite number of cut-off frequencies does the filter attain an extra ripple of maximal size. In the FIR case, such extra-ripple filters can be most conveniently obtained with the algorithm of Hofstetter et al. [41], which is discussed in Chapter 4. The zero-shifting algorithm, as used in this chapter, will be used to obtain only the “extra-ripple” filters. As such, it is analogous to the algorithm of Hofstetter et al. in the filters it produces — although it works entirely differently.

As in the classical case, and the cases described in other chapters, the magnitude squared function is obtained and spectrally factored. The transformation \( \tau = \frac{1}{2}(1 - \cos \omega) \) is used. In this case, a rational function that is nonnegative over \([0,1]\) is to be obtained. It will be transformed and spectrally factored (this can be done by simply transforming the poles and zeros of the rational function) to obtain the filter coefficients.

For the zero-shifting algorithm design of extra-ripple generalized Chebyshev II filters, the form of the transfer function is in part known. Let us denote the number of zeros in the stopband by \( L \), the number of zeros contributing to the passband by \( M \), and the number of poles by \( N \). Then the problem becomes one of finding a rational function \( F(x) \)

\[
F(x) = \frac{P(x)S(x)}{Q(x)}
\]  
(8.1)

where

1. degree \( P(x) \) is \( M \)
2. degree \( S(x) \) is \( L \)
3. degree \( Q(x) \) is \( N \).
The polynomial \( S(x) \) is to have all its zeros in the stopband. The polynomials \( P(x) \) and \( Q(x) \) are used to obtain the highest degree of flatness at \( x = 0 \) subject to the constraints on \( S(x) \).

To clarify this discussion, we have the following problem formulation. Given \( L, M, N \), and \( \delta \), find polynomials \( P(x), S(x) \) and \( Q(x) \) of respective degrees \( L, M, \) and \( N \), such that

1. \( F(0) = 1 \)
2. \( F^{(i)}(0) = 0 \), for \( i = 1, \ldots, M + N + 1 \).
3. \( F(x) \geq 0 \) for all \( x \in [0, 1] \).
4. \( F(x) \) has \( \left\lfloor \frac{L}{2} \right\rfloor \) local maxima in \((0, 1]\) of value \( \delta^2 \).

For example, see Figure 8.1 which gives the solution for \( L = 6, M = 4, N = 2 \), and \( \delta = 0.05 \). The Chebyshev II filter obtained by transforming the poles and zeros of the function \( F(x) \) shown in Figure 8.1 is shown in Figure 8.2. Another example is shown in Figures 8.3 and 8.4.

If \( S(x) \) is fixed, then \( P(x) \) and \( Q(x) \) can be found by solving a system of linear equations. Note that

\[
F(x) - 1 = \frac{P(x)S(x)}{Q(x)} - 1 \tag{8.2}
\]

\[
= \frac{P(x)S(x) - Q(x)}{Q(x)} \tag{8.3}
\]

\[
= -x^{M+N+1}U(x) \tag{8.4}
\]

where \( U(x) \) is some polynomial of degree \( L - N - 1 \). The last equality simply asserts the existence of the polynomial \( U(x) \). The last equality comes from imposing the flatness constraint at \( x = 0 \). The expression incorporates the higher order root at \( x = 0 \) of \( F(x) - 1 \). This equation can be rewritten as

\[
Q(x) = P(x)S(x) + x^{M+N+1}U(x). \tag{8.5}
\]
Figure 8.1: A rational function $F(x)$ for Chebyshev II filter design. The graph on the right is a magnification of the stopband behavior of $F(x)$. $L = 6$, $M = 4$, $N = 2$, $\delta = 0.05$. The size of the ripples of $F(x)$ is $\delta^2$.

Figure 8.2: An extra-ripple generalized Chebyshev II filter. $L = 6$, $M = 4$, $N = 2$, $\delta = 0.05$. The stopband edge is $\omega_s = 0.4955\pi$. 
Figure 8.3: A rational function $F(x)$ for Chebyshev II filter design. The graph on the right is a magnification of the stopband behavior of $F(x)$. $L = 5$, $M = 5$, $N = 2$, $\delta = 0.05$.

Figure 8.4: An extra-ripple generalized Chebyshev II filter. $L = 5$, $M = 5$, $N = 2$, $\delta = 0.05$. The stopband edge is $\omega_s = 0.5957\pi$. 
Since the degree of $Q(x)$ is $N$, we have $Q(x) = T_N\{P(x)S(x)\}$ where $T_N$ denotes polynomial truncation as in Chapter 7. From this equation, it follows that $P(x)$ must be found such that

$$\text{coeff}(P(x)S(x), k) = 0$$

for $N + 1 \leq k \leq M + N$. Here $\text{coeff}(T(x), k)$ denotes the coefficient of $x^k$ of $T(x)$. This gives $M$ equations for $P(x)$. Without loss of generality, $P(x)$ can be made monic (another linear equation). These $M + 1$ equations will be denoted by

$$T_{sp} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where $p$ is a vector formed from the coefficients of $P(x)$. The last equation is the monotonic constraint. We will denote the right hand side by $v$ below. In this case, given $S(x)$, the linear equations above uniquely determine the polynomial $P(x)$ that maximizes the flatness at $x = 0$. The denominator $Q(x)$ is then found via $Q(x) = T_N\{P(x)S(x)\}$. The problem remains to obtain $S(x)$ such that the function $F(x)$ exhibits an equiripple behavior in its stopband in $[0, 1]$. The following iterative zero-shifting technique describes how to update $S(x)$ appropriately, so that equiripple behavior is obtained.

The zero-shifting technique is so-called, as it proceeds by updating the zeros of $S(x)$. Because we seek a function $F(x)$ that is nonnegative in $x \in [0, 1]$, its zeros in $x \in (0, 1)$ must be double zeros. The zero (if any) at $x = 1$ will be a simple (first order) root. The occurrence of a root at $x = 1$ depends on the parity of $L$, the degree of $S(x)$. When $L$ is even, the stopband behavior of $F(x)$ will be as shown in Figure 8.1. When $L$ is odd, the stopband behavior of $F(x)$ will be as shown in Figure 8.3. Consequently, when $L$ is even, $S(x)$ can be written as

$$S(x) = (x - x_1)^2(x - x_2)^2 \cdots (x - x_{L/2})^2,$$

while if $L$ is odd, $S(x)$ can be written as

$$S(x) = (x - x_1)^2(x - x_2)^2 \cdots (x - x_{(L-1)/2})^2(x - 1).$$
With this expression, the free parameters are the locations of the zeros of \( S(x) \). Note that in both cases, even and odd \( L \), the number of free parameters is the same as the number of local maxima of \( F(x) \) in the stopband, which can be written as \([\frac{L}{2}]\).

It should be noted that when \( S(x) \) is an arbitrary polynomial, then the polynomial \( Q(x) \) that is obtained via \( Q(x) = \mathcal{J}_N\{P(x)S(x)\} \) may have its roots in the interval of approximation \([0, 1]\). This is unacceptable, because the roots of \( Q(x) \) are poles of \( F(x) \). However, when the zeros of \( S(x) \) are double zeros, as they are here, this occurrence of poles in the interval of approximation does not arise. This can be shown by contradiction: if they were to arise, then the function \( F(x) \) must have more zeros than it can possibly have, given its degree.

The idea of the zero-shifting algorithm, is to shift the zeros of \( S(x) \) so that the local maxima of \( F(x) \) in the stopband become equal in size. As in [60], the natural logarithm of the function is used.

\[
\ln F(x) = \ln S(x) + \ln P(x) - \ln \mathcal{J}_N\{P(x)S(x)\}. \quad (8.10)
\]

Given a set of stopband zeros \( x_1, \ldots, x_{[L/2]} \), and the function \( F(x) \) obtained using these stopband zeros, let the local maxima of \( F(x) \) in the stopband be denoted by \( r_1, \ldots, r_{[L/2]} \). For example, consider Figure 8.5. The two double zeros near 0.64 and 0.82 are \( x_1 \) and \( x_2 \). The two local maxima are to be labeled \( r_1 \) and \( r_2 \). (\( r_1 \) lies between \( x_1 \) and \( x_2 \), and \( r_2 \) lies between \( x_2 \) and 1.) The local maxima \( r_i \) can be found numerically.

The zero-shifting algorithm proceeds by updating the stopband zeros by solving the following system of equations.

\[
\ln F(r) + \frac{\partial \ln F(r)}{\partial x} \cdot \Delta x = \ln \delta^2 \quad (8.11)
\]

where \( x = (x_1, \ldots, x_{[L/2]})^t \) and \( r = (r_1, \ldots, r_{[L/2]})^t \).

The matrix of derivatives is most conveniently split into three parts:

\[
\frac{\partial \ln F(r_i)}{\partial x_j} = J_{i,j}^S + J_{i,j}^P - J_{i,j}^O, \quad (8.12)
\]
Figure 8.5: A typical stopband behavior of $F(x)$ during the course of the zero-shifting algorithm before convergence is attained. $L = 5$.

These terms are given by:

$$J_{i,j}^S = \frac{1}{S(r_i)} \frac{\partial S(r_i)}{\partial x_j}$$ \hspace{1cm} (8.13)

$$J_{i,j}^P = \frac{1}{P(r_i)} \frac{\partial P(r_i)}{\partial x_j}$$ \hspace{1cm} (8.14)

$$J_{i,j}^Q = \frac{1}{Q(r_i)} \frac{\partial Q(r_i)}{\partial x_j}$$ \hspace{1cm} (8.15)

The term $J_{i,j}^S$ can be simplified:

$$J_{i,j}^S = \frac{2}{x_j - r_i}$$ \hspace{1cm} (8.16)

To find $J^P$ requires implicit differentiation. Differentiate the equation

$$T_S p + T_S \frac{\partial p}{\partial x_j} = 0$$ \hspace{1cm} (8.18)
\[
\frac{\partial p}{\partial x_j} = -[T_s]^{-1} \left[ \frac{\partial T_s}{\partial x_j} \right] p \tag{8.19}
\]

It remains to find \( J^Q \).

\[
\frac{\partial Q(r_i)}{\partial x_j} = \frac{\partial}{\partial x_j} J_n \{ P(x)S(x) \}(r_i) = J_n \left\{ \frac{\partial P(x)S(x)}{\partial x_j} \right\}(r_i) \tag{8.21}
\]

\[
= J_n \left\{ P(x) \frac{\partial S(x)}{\partial x_j} + S(x) \frac{\partial P(x)}{\partial x_j} \right\}(r_i) \tag{8.22}
\]

Solving the system 8.12,

\[
\Delta x = \left[ \frac{\partial \ln F(r)}{\partial x} \right]^{-1} \left[ \ln F(r)/\delta^2 \right]. \tag{8.23}
\]

By updating \( x \) by \( x = x + \Delta x \), convergence to the (extra-ripple) equiripple solution is obtained.

8.2.1 Step-Size Reduction

Because the step size might be too large at the beginning of the algorithm, it is necessary to decrease the step size appropriately on certain iterations. A systematic robust and simple way of doing this is now described.

To ensure that the step size taken is not too long, we employ the following criterion that is most easily described by an example. Suppose the degree of \( S(x) \) is 5, so that Figure 8.5 represents the typical stopband behavior of \( F(x) \) before convergence is attained.

We suggest that \( x_1 + \Delta x_1 < r_1 \) and that \( x_2 + \Delta x_2 \) lies in the interval \((r_1, r_2)\). In general, we wish that for \( 1 \leq j \leq \lfloor \frac{L}{2} \rfloor \) lie within the interval \((r_{j-1}, r_j)\) (if the degree of \( S(x) \) is even, then the term \( r_{L/2} \) is to be replaced by 1.) If this condition is not met, then we suggest successively halving \( \Delta x \) until it is met. This technique ensures that the iterative process remains on-track. This technique can be easily checked as
follows. Form a vector \( g \),

\[
g = [x_1 + \Delta x_1, r_1, x_2 + \Delta x_2, r_2, \ldots].
\]  

(8.24)

If this vector is in ascending order, then the condition we impose on \( \Delta x \) is satisfied. Otherwise, the condition we seek is violated, and we set \( \Delta x = \frac{1}{2} \Delta x \).

To calculate the stopband edge of a filter obtained by executing the zero-shifting algorithm, write the equation \( F(x_s) = \delta^2 \) as

\[
P(x_s)S(x_s) - \delta^2 Q(x_s) = 0.
\]  

(8.25)

The stopband edge \( x_s \) is a root of this polynomial, so can be found using a polynomial root finder. \( x_s \) is identified as the smallest real root in \([0,1]\). The stopband edge \( \omega_s \) is given by \( \omega_s = \arccos(1 - 2x_s) \).

The convergence of the zero-shifting appears to be quadratic and robust.

### 8.2.2 Remarks

Note that the stopband edge was not part of the design procedure that uses the zero-shifting algorithm — and so it can not be explicitly specified. It is implied and induced by the chosen degrees of \( P(x) \), \( S(x) \), and \( Q(x) \), and by the chosen stopband ripples size \( \delta \). By increasing the degree of \( S(x) \) by one and by decreasing the degree of \( P(x) \) by one, the total number of zeros is unchanged, but the stopband edge is different (giving a wider passband).

For a fixed ripple size \( \delta \), if the total number of zeros \( (L + M) \) is fixed, and the number of poles \( (N) \) if fixed, and \( L \) is varied, then a table can be obtained that gives the stopband edge of each of the filters. Table 8.1 is such a table for \( L + M = 10 \) and \( N = 2 \). The responses of the filters tabulated in the table are plotted together in Figure 8.6. As can be seen in the figure, the stopband edges can not be precisely positioned with these filters — only a few stopband edges are available.

We have found it difficult to modify the zero-shifting algorithm to obtain filters having specified stopband edges that lie between the stopband edges of two “adjacent”
Table 8.1: A specification table for generalized Chebyshev II filters for which the total number of zeros \((L + M)\) is 10 and the total number of poles \((N)\) is 2. \(\delta = 0.05\). \(\omega_s\) denotes the stopband edge. Recall that for these filters, the number of zeros on the unit circle in the z-plane is equal to \(L\), and the number of zeros contributing to the passband is \(M\). The degree of flatness at \(\omega = 0\) is \(M + N + 1\). The filter responses are shown in Figure 8.6.

<table>
<thead>
<tr>
<th>(L)</th>
<th>(M)</th>
<th>(N)</th>
<th>(\omega_s / \pi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>2</td>
<td>0.1459</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>0.2319</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0.3164</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0.4034</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0.4955</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0.5957</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>2</td>
<td>0.7075</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>0.8332</td>
</tr>
</tbody>
</table>

extra-ripple filters. However, given these two “adjacent” extra-ripple filters obtained with the zero-shifting algorithm, a rational Remez-like exchange algorithm can be used. To exercise precise control of the stopband edge, the rational Remez-like exchange algorithm of the next section is suggested.

8.3 Exchange Algorithm

In this section we describe an exchange algorithm that allows the user to specify the stopband edge, \(\omega_s\), exactly (in addition to the stopband ripple size). Consider the filters shown in Figures 8.2 and 8.4. They both have ten zeros and two poles. Their stopband edges are \(0.4955\pi\) and \(0.5957\pi\) respectively. To design a Chebyshev II filter having a stopband edge \(\omega_s\) that lies in the interval \([0.4955\pi, 0.5957\pi]\), say at \(0.55\pi\), exactly, the filter shown in Figure 8.4 provides an excellent initialization.
Figure 8.6: A set of extra-ripple Chebyshev II filters. $L + M = 10$, $N = 2$, $\delta = 0.05$. Their stopband edges are tabulated in Table 8.1.

for an exchange algorithm. (To be precise, its extremal points provide an excellent initialization of the reference set of an exchange algorithm). In general, a table like Table 8.1 can be used to determine the appropriate initialization. Because the zero-shifting algorithm was used to construct the filters of Table 8.1 and because their stopband frequencies are easily found, this approach is straightforward.

Several aspects of a rational Remez exchange algorithm are discussed in Chapter 9 and in [97]. The methods described there-in make it more practical. One problem that is not addressed in Chapter 9 and in [97] is the determination of an appropriate initialization of the reference set. Unlike the polynomial case which converges for all initializations, the rational Remez algorithm does not always converge from arbitrary initializations. The problem that arises is the occurrence of poles in the interval of approximation. In Chapter 9, a technique for overcoming this problem is described, if this problem occurs during the course of the algorithm (and not on the first iteration). However, if poles occur in the interval of approximation on the first iteration, then the technique of Chapter 9 can not be used.

For the design of Chebyshev II filters, however, a suitable initialization is provided
by the extra ripple filter obtained using the zero-shifting algorithm described above. The initialization issues will be further discussed after a rational Remez-like algorithm is described.

Like the exchange algorithms for polynomial approximation, the algorithm for the rational case proceeds by iteratively (i) solving an interpolation problem over an specified set of reference points, and (ii) updating the set of reference points. A flowgraph of the exchange algorithm is shown in Figure 8.7. It is similar to the flowgraph for an exchange algorithm for polynomial approximation; the major difference is the branch taken when a root of $\hat{Q}$ occurs in $[0, 1]$. In the polynomial case, there is no denominator — so this situation does not arise and does not need special attention. For this reason, the set of secondary reference points, $s_i$, is not needed in the polynomial exchange algorithms. The points $s_i$ are the reference points from a previous iteration.

### 8.3.1 Interpolation Step

Above, the rational function $F(x)$ was written as $F(x) = \frac{P(x)S(x)}{Q(x)}$, where $P(x)$ and $S(x)$ are related via a system of linear equations as described above. In terms of the coefficients of $P(x)$ and $S(x)$, however, this parameterization is nonlinear — so it is not a suitable form to use for the interpolation problem. For the interpolation problem, the more convenient form to use is given by

$$F(x) = 1 + \frac{x^{M+N}U(x)}{Q(x)} \tag{8.26}$$

where the degree of $U(x)$ is $L - N$. Then the interpolation equations $F(x_i) = y_i$ are linear in the coefficients of $U(x)$ and $Q(x)$. The size of the reference set is $L + 1$. The values $y_i$ are taken to be

$$y = (y_1, y_2, \ldots, y_{L+1})^t = (\delta, 0, \delta, 0, \ldots)^t. \tag{8.27}$$

The interpolation equation

$$F(x_i) = y_i \tag{8.28}$$
Figure 8.7: Flowgraph for the exchange algorithm for Chebyshev II filters.
becomes

\[ x_i^{M+N}U(x_i) - Q(x_i)(y_i - 1) = 0 \]  \hspace{1cm} (8.29)

for \( i = 1, \ldots, L + 1 \). In matrix notation, this becomes

\[
\begin{bmatrix}
\text{diag}(x^{M+N}) \cdot V_1 & -\text{diag}(y - 1) \cdot V_2
\end{bmatrix}
\begin{bmatrix}
u \\
q
\end{bmatrix} = 0
\]  \hspace{1cm} (8.30)

where \( V_1 \) and \( V_2 \) are appropriate Vandermonde-like matrices. (They are not Vandermonde exactly only because they are not square). Here \( u \) and \( q \) are vectors formed from the coefficients of \( U(x) \) and \( Q(x) \).

### 8.3.2 Exchange Step

Assuming that \( F(x) \) is viable (by viable, we mean that no poles occur in \([0,1]\)), the way in which the reference set is updated is the same as that of the Remez algorithm. Update the reference points \( r_i \) by letting \( r_2, \ldots, r_{L+1} \) be the stopband extremals of this function \( F(x) \) listed in ascending order. Let \( r_1 \) be equal to \( z_s \), the specified stopband frequency after transformation \((x_s = \frac{1}{2}(1 - \cos \omega_s))\).

However, as discussed in Chapter 9, \( F(x) \), the solution to the interpolation problem may possess poles in \([0,1]\), (may not be viable). If \( F(x) \) is not viable (\( \hat{F}(x) \) in the flowgraph), then the reference set used to obtain \( F(x) \) is modified as shown in the flowgraph in Figure 8.7. In the flowgraph, the function \( F(x) \) is only updated when it is found to be viable. The function \( \hat{F}(x) \) in the flowgraph \((\hat{F}(x) = 1 + \frac{x^{M+N}U(x)}{\hat{Q}(x)})\) is the temporary function for which “viability” is checked.

The loop begun when roots of \( \hat{Q} \) occur in \([0,1]\) must eventually terminate because with each traversal of the loop, \( r_i \) approaches \( s_i \) — and because the solution to the interpolation problem over \( s_i \) had no poles in \([0,1]\), eventually, as \( r_i \) is updated, \( \hat{F} \) will become viable. In practice, this takes only a few iterations.
8.3.3 Initialization

The initialization consists of initializing the reference points \( r_i \) and \( s_i \). The reference set should be initialized by using the local extremal points of the filter obtained with the zero-shifting algorithm. The set \( \{s_i\}_{i=1}^{t+1} \) must be a set of points such that when the interpolation problem is solved over \( s_i \), a viable function \( F(x) \) is obtained. The reference points \( r_i \) are the “update” of the points \( s_i \). It is suggested that \( s_1 \) be taken to be less than \( r_1 \) (\( r_1 = x_s \)). This is so that, if poles occur in \([0,1]\) on the first iteration, then the modification of \( r_i \) described in the flowgraph does not alter the order of the reference points \( r_i \).

8.3.4 Examples

For this example, suppose the specifications are as follows: Find the Chebyshev II filter with ten zeros and two poles (away from the origin) having a stopband edge of \( \omega_s = 0.55\pi \) and a stopband ripple of \( \delta = 0.05 \). Consider Figures 8.2 and 8.4 and Table 8.1. From the table, it can be seen that the desired stopband edge of \( 0.55\pi \) lies between the stopband edges of the filters shown in Figures 8.2 (for which \( L = 6 \) and \( M = 4 \)) and 8.4 (for which \( L = 5 \) and \( M = 5 \)). Therefore, the exchange algorithm described above should be executed with \( L = 5 \) and \( M = 5 \). The resulting filter is shown in Figure 8.8.

As the stopband edge \( \omega_s \) is varied from \( 0.5957\pi \) to \( 0.4955 \), the number of stopband extremal points increases for the filters tabulated in Table 8.1. Therefore, for some stopband edges specifications in this interval, there must exist a Chebyshev II filter having an extra ripple on non-maximal size (a scaled extra ripple, as called in [76]). Figure 8.9 illustrates a scaled extra-ripple filter. It was obtained by specifying the stopband edge \( \omega_s = 0.503\pi \). As the specified stopband edge is increased from this point, the two zeros on the unit circle closest to \( z = -1 \) move along the unit circle and meet at \( z = -1 \) and then part — one remains at \( z = -1 \) while the other zero travels along the real line towards the passband. As the stopband edge is further increased,
the zero at \( z = -1 \) will depart that point and travel along the real line, towards the passband. This is most easily demonstrated using a graphical interface program that allows the user to observe the movement of poles and zeros as the stopband edges is varied under the user's control.

For the following example, we design a set of Chebyshev II filters for which each filter has \( L + M + N = 16 \). By varying the number of poles and zeros, keeping the total number of poles and zeros equal to 16, the trade-off between numerator and denominator degrees can be examined. The resulting filters are illustrated in Figure 8.10. The frequency response magnitude having the broadest transition is that of the FIR filter — the response having the sharpest is that of the classical IIR Chebyshev II filter having 8 poles and 8 zeros. It can be seen that (i) trading a few zeros for poles gives an improved response (sharper transition) and that (ii) the filter having an equal number of poles and zeros (the classical Chebyshev II filter) does not provide that much improvement. Note that the classical Chebyshev II filter is probably the most sensitive to the implementation issues of IIR filters because it has the most poles. Therefore, a good trade-off might be provided by the filter having two or four
Figure 8.9: A generalized Chebyshev II filter with a scaled extra ripple. $L = 5$, $M = 5$, $N = 2$, $\delta = 0.05$, $\omega_s = 0.503\pi$.

poles.

8.3.5 Generalized Chebyshev I Filters

For the last example, we show that generalized Chebyshev I filters (with more zeros than poles) can be obtained by simply transforming a Chebyshev II filter. By taking the rational function $F(x)$ used above and forming the rational function $G(x) = 1 - F(x)$, a nonnegative rational function is obtained whose numerator and denominator degrees are the same as those of $F(x)$. (Note however, that if the denominator of $F(x)$ were to have a higher degree than its numerator, then the degree of the numerator of $1 - F(x)$ will in general be equal to the degree of its denominator). The function $G(x)$ can be transformed to obtain a Chebyshev I filter. Note that $G(x)$ is given by $\frac{e^{xM+NU(x)}}{Q(x)}$.

To relate the passband ripple size of a Chebyshev I filter to the stopband ripple size of a Chebyshev II filter, label them $\delta_p$ and $\delta_s$ respectively. Let us normalize the Chebyshev I filter so that its maximum value is 1 (as is traditional for IIR filters). To obtain a Chebyshev I filter having a passband ripple size of $\delta_p$, design a Chebyshev
Figure 8.10: A set of generalized Chebyshev II filters having 16 poles and zeros (total) and the same stopband edge ($\omega_s = 0.6\pi$) and stopband ripple size ($\delta = 0.05$). The FIR filter has the broadest transition, while the classical Chebyshev II filter has the sharpest transition.
Figure 8.11: A generalized Chebyshev I filter. $L + M = 12$, $N = 2$, $\delta_p = 0.05$, $\omega_s = 0.35\pi$. The zero at $z = 1$ is of order 8.

II filter having a stopband ripple of $\delta_s = \sqrt{1 - (1 - \delta_p)^2}$.

Let the number of zeros be 12 and the number of poles be 2, let the passband ripple size $\delta_p$ be 0.05, and the passband edge be $\omega_p$ is 0.35$\pi$. The resulting Chebyshev I filter is illustrated in 8.11.

8.4 Conclusion

This chapter has shown how to use two algorithms together, the “zero-shifting” algorithm and a Remez-like exchange algorithm, to give a robust, convenient technique for obtaining generalized Chebyshev II (and Chebyshev I) IIR digital filters. They are generalizations in that they permit more zeros than poles (away from the origin).
Chapter 9

A Modified Rational Remez Algorithm for Recursive Digital Filter Design

9.1 Introduction

The algorithm described in this chapter minimizes the Chebyshev norm of $H(\omega) - D(\omega)$ where $H(\omega)$ and $D(\omega)$ are the realized and desired magnitude squared frequency responses respectively. The approach constrains $H(\omega)$ to be nonnegative, for then it can be spectrally factored to obtain a stable filter whose magnitude squared frequency response approximates $D(\omega)$. To obtain the nonnegative approximations the rational Remez exchange algorithm described by Powell [71] (also see [123]) is modified.

It appears that the rational Remez exchange algorithm is used infrequently for the design of IIR digital filters for three reasons. First is the need to solve a set of nonlinear equations at the interpolation step of each iteration. However, these equations can be converted into a generalized eigenvalue problem, a technique we haven’t seen in the filter design literature since [110].

Second is the necessity that the magnitude squared approximation be nonnegative. The relevant constrained approximation problem can, however, be solved almost as easily as the unconstrained problem. In fact in [30] it was shown how to incorporate upper and lower bound constraints in the Parks-McClellan FIR filter design program [70]. Below, similar modifications to the rational algorithm are described.

Possibly the most important reason the rational Remez exchange algorithm has not been more widely used is that it is not guaranteed to converge. It fails to converge when all the solutions to the nonlinear equations associated with the interpolation step have denominators that are not strictly positive. A method is suggested below
for overcoming this situation by perturbing the reference set appropriately. It was observed that it is sometimes necessary to change the reference only slightly to make the rational Remez converge successfully.

An earlier version of this chapter is [97]. Some related papers require all the zeros of the filter to lie on the unit circle [36, 43, 61, 87] or require a special form for the frequency response [64, 100]. Other optimization procedures have been suggested in [16, 22, 58]. The differential correction algorithm used in [23, 77] is a robust algorithm but is computationally intensive since it requires the solution to a sequence of linear programming problems and does not take advantage of the alternation property (see below). [4] gives an interesting approach to the interpolation step, but suggests that the desired response in the stopband be made a small positive number, which is not necessary here.

Throughout this chapter, by "poles and zeros", we mean those that lie away from the origin.

### 9.2 The Rational Remez Exchange Algorithm

The Remez exchange algorithm for Chebyshev approximation by rational functions is based on the alternation property and an interpolation step, as is the polynomial Remez algorithm. We use the notation,

$$H(\omega) = \frac{b_0 + b_1 \cos(\omega) + \ldots + b_m \cos(m\omega)}{1 + a_1 \cos(\omega) + \ldots + a_n \cos(n\omega)} = \frac{B(\omega)}{A(\omega)}$$

(9.1)

for the realized magnitude squared frequency response. $\mathcal{R}_{m,n}$ denotes the subset of such functions for which $A(\omega)$ has no zeros in $[0, \pi]$. Let $S \subset [0, \pi]$ be a union of intervals and let $D(\omega)$ be the desired non-negative function. By the best rational Chebyshev approximation from $\mathcal{R}_{m,n}$ to $F(\omega)$ over $S$ we mean the function $H(\omega)$ in $\mathcal{R}_{m,n}$ that minimizes

$$||H(\omega) - F(\omega)|| = \max_{\omega \in S} |H(\omega) - F(\omega)|.$$
The error function $H(\omega) - D(\omega)$ is denoted $E(\omega)$. Recall that the best Chebyshev approximations by polynomials ($n = 0$) are uniquely characterized by an alternation property. However, in the rational case, this condition is only sufficient [71]:

**Theorem 1.** Let $(\omega_1, \ldots, \omega_{m+n+2})$ be a sequence of points of $S$ in ascending order (a reference set), and let $D(\omega)$ be a continuous function on $S$. If $H(\omega)$ is in $\overline{R}_{m,n}$ and if the equations

$$H(\omega_i) + (-1)^i \delta = D(\omega_i)$$

(9.2)

for $i = 1, \ldots, m+n+2$ hold for $|\delta| = ||H(\omega) - D(\omega)||$, then $H(\omega)$ is the best Chebyshev approximation to $D(\omega)$ from the set of rational functions $\overline{R}_{m,n}$.

However, the size of the reference set of the best approximation may be less than $m+n+2$, in which case, the best approximation is degenerate [11,123].

The progression of the rational Remez algorithm relies on the following key fact. If (i) the set $S$ over which the approximation is performed consists of exactly $m+n+2$ points and (ii) the best approximation does indeed have $m+n+2$ extremal frequencies, then the best approximation over $S$ can be found by solving (9.2). This is an interpolation problem and its solution is explained below.

The rational Remez algorithm follows the same strategy as the polynomial Remez algorithm:

1. **Initialization:** Select a reference set of $m+n+2$ points.

2. **Interpolation:** Calculate the best approximation to $D(\omega)$ over this reference set. (Solve the system (9.2)).

3. **Update:** Update the reference set exactly as in the polynomial Remez algorithm. Go back to step 2.

**Interpolation Step:** Although the system in (9.2) is nonlinear in the filter coefficients, it can be written as a generalized eigenvalue problem [71]: rewrite (9.2)
as

\[ B(\omega_i) + (-1)^i \delta A(\omega_i) = D(\omega_i) A(\omega_i). \]  \hfill (9.3)

$|\delta|$ is called the leveled reference error. In matrix notation,

\[ M_1 b + \delta D_1 M_2 a = D_2 M_2 a \]  \hfill (9.4)

where

\[ b = (b_0, ..., b_m)^t \quad a = (1, a_1, ..., a_n)^t \]

\[ (M_1 b)_i = B(\omega_i) \quad (M_2 a)_i = A(\omega_i) \]

\[ (D_1)_{i,i} = (-1)^i \quad (D_2)_{i,i} = D(\omega_i) \]

where $D_1$ and $D_2$ are diagonal matrices and $M_1$ and $M_2$ are matrices of cosine terms. Because $M_1$ has full rank $m + 1$ there is a matrix $Q$ of size $n + 1$ by $m + n + 2$ with full rank such that $QM_1 = 0$. Applying $Q$ to (9.4) one eliminates $b$ and obtains the equation for $\delta$ and $a$

\[ \delta Q D_1 M_2 a = Q D_2 M_2 a. \]  \hfill (9.5)

Once $\delta$ and $a$ are found, $b$ is found by solving a linear system (see (9.4)). Equation (9.5) is a generalized eigenvalue problem (it is of the form $Ax = \lambda Bx$). Since there are $n + 1$ generalized eigenvalues $\delta$, one must choose an appropriate one. This is straightforward because there will be at most one generalized eigenvalue for which the corresponding denominator $A(\omega)$ is positive over the current reference set [71,81]. If there is no such value, then the best approximation from $\bar{R}_{m,n}$ to $D(\omega)$ over the reference set is degenerate. However, even if there is a generalized eigenvalue that gives rise to a denominator positive over the reference set, it may become negative elsewhere on $S$. In either case, the Remez algorithm fails and one must use some corrective measure or an alternative approximation method (see below). The two reasons the algorithm may fail are:
1. The best approximation from $\bar{R}_{m,n}$ to $D(\omega)$ over $S$ is degenerate.

2. Sensitivity to the initial reference set.

It is interesting to note that degeneracy of the best approximation over the set $S$ is very rare: for a given function, all intervals on which it has degenerate best approximations form a set of measure zero [82]. In this chapter it is assumed that the best approximation is non-degenerate. Near degenerate best approximations are, however, not uncommon. Furthermore, it is the nearly degenerate best approximations that are more computationally difficult to find, for they are sensitive to the initial reference set. Unless the usual reference set update procedure is modified, failure of the the rational Remez algorithm for these near degenerate cases is imminent.

If $E^*_{m,n}$ denotes the Chebyshev error of the best approximation from $\bar{R}_{m,n}$ and if the best approximation from $\bar{R}_{m,n}$ is nearly degenerate, then $E^*_{m-1,n-1}$, the Chebyshev error of the best approximation from $\bar{R}_{m-1,n-1}$, is usually only slightly higher than $E^*_{m,n}$. That is, by reducing the number of poles and zeros both by one, a nearly equivalent approximation can be obtained. Therefore, it is advantageous to reduce the order in this way. For by doing so, the computation required for implementing the filter is reduced while the increase in the Chebyshev error is small. (See example 2 below.)

**Updating the Reference Set:** As in the polynomial Remez algorithm, a new reference set is found such that: (i) The current error function, $E(\omega)$, alternates sign on the new reference set. (ii) $|E(\omega)| \geq |\delta|$ for each point, $\omega$, of the new reference set. (iii) $|E(\omega)| > |\delta|$ for at least one point, $\omega$, of the new reference set.

Note that the number of points in the stopband of the initial reference set should not exceed the number of zeros by more than one. (Because, of course, there can not be more stopband zeros than there are zeros.)

**Convergence:** As in the polynomial case, it can be shown that $|\delta|$ increases from one iteration to the next as long as $A(\omega)$ has no zeros in $[0, \pi]$. Moreover, on each iteration, $|\delta|$ gives a lower bound for the Chebyshev error of the best approximation.
An upper bound for $E^*$ is given simply by the maximum of $|E(\omega)|$. These upper and lower bounds for $E^*$ give a meaningful stopping criteria.

It should be noted that the inclusion of a weighting function is straightforward.

### 9.3 Overcoming "Faulty" Reference Sets

When no solution to the generalized eigenvalue problem of the interpolation step gives rise to a positive denominator $A(\omega)$, it is suggested that the reference set be perturbed in a systematic manner.

First, suppose that the reference set on some iteration gives rise to a positive $A(\omega)$ in (9.3). As noted above, it may be the case that the new reference set obtained by updating the current reference set with a multiple (or single) point exchange scheme may fail to give rise to a positive $A(\omega)$.

One way of overcoming this failure is given by the differential correction algorithm [8,20,23,56]. It is possible to combine the Remez and differential correction algorithm as is done in [45], but because the differential correction algorithm is itself an iterative procedure, another method is preferred.

The single point exchange scheme for updating the reference set is typically carried out by first finding the point, call it $\omega_{\text{new}}$, at which $|E(\omega)|$ attains its maximum value and second, by replacing a point in the reference set by $\omega_{\text{new}}$. The appropriate point to replace, call it $\omega_r$, is uniquely determined by the conditions listed above for updating the reference set.

If the reference set obtained by the single point exchange scheme fails to provide a positive denominator, instead of replacing $\omega_r$ by $\omega_{\text{new}}$, our approach replaces $\omega_r$ by $(\omega_r + \omega_{\text{new}})/2$. If $\omega_r$ and $\omega_{\text{new}}$ are located on opposite ends of $[0, \pi]$, as occasionally occurs, then subtracting $\pi$ is necessary. If the resulting reference set again fails to provide a positive denominator or if $|E(\omega_r + \omega_{\text{new}})/2)| < |\delta|$, then our approach replaces $\omega_r$ by $\frac{3}{4}\omega_r + \frac{1}{4}\omega_{\text{new}}$. For as long as the new reference fails to provide a positive denominator and an increase in $|\delta|$ our approach replaces $\omega_r$ by $(1 - \frac{1}{2\epsilon})\omega_r + \frac{1}{2\epsilon}\omega_{\text{new}}$. 
That is, our approach employs successively smaller perturbations to the reference set.

If no viable reference set is found, then, with respect to the grid density, the new reference point \( (1 - \frac{1}{2k})\omega_r + \frac{1}{2k}\omega_{\text{new}} \) will eventually equal \( \omega_r \). In this case, our approach uses another value for \( \omega_{\text{new}} \). Namely, \( \omega_{\text{new}} \) is taken to be the point at which \( |E(\omega)| \) attains its second greatest local maximum. With this new value of \( \omega_{\text{new}} \), our approach carries out the single point exchange again, and subsequently replaces \( \omega_r \) by \( (1 - \frac{1}{2k})\omega_r + \frac{1}{2k}\omega_{\text{new}} \) for \( k = 1, 2, 3, \ldots \) until a viable reference set is found. Again, if none is found, \( \omega_{\text{new}} \) is taken to be the point at which \( |E(\omega)| \) attains its third greatest local maximum, and so on.

By testing this sequence of candidate reference set updates, our approach usually finds one that yields a positive denominator and an increase in \(|\delta|\). Continuing in this manner usually (but not always) results in successful convergence to the best approximation.

Sometimes however, no perturbation of the reference set by a grid point results in a viable reference set. In this case, either the best approximation is actually degenerate, or more likely, more than a perturbation is needed to obtain a reference set from which the Remez algorithm can be made to converge. In our experience, this can be overcome by moving a reference point from one end of the interval \([0, \pi]\) and inserting it between the two reference points on the other side of the interval.

These observations were collected primarily from experiences with the design of low pass filters, but it is our expectation that the same phenomena are found in general and that the same corrective measures will prove useful. The preceding discussion also assumes that a viable initial reference set has been found. Usually it is not difficult to find an initial reference set for which a solution \( A(\omega) \) to (9.3) is positive over \([0, \pi]\), although we have not arrived at a robust method for doing so.
9.4 Constrained Rational Remez Algorithm

It is necessary to find a nonnegative function approximating $D(\omega)$, a problem addressed in [30] for FIR filter design, in which optimality is maintained. Here, appropriate modifications are made to the rational Remez algorithm.

Constraints on the maximum and minimum values of $H(\omega)$ are imposed; call these constraints $u(\omega)$ and $l(\omega)$ for "upper" and "lower". The interpolation step now computes the rational function interpolating $D(\omega_i) + (-1)^i \delta$, $u(\omega_i)$, or $l(\omega_i)$ at $\omega_i$ depending on the error function. The lower constraint is violated at $\omega_i$ if

$$D(\omega_i) - |\delta| \text{sgn}(D(\omega_i) - H(\omega_i)) < l(\omega)$$

while the upper constraint is violated at $\omega_i$ if

$$D(\omega_i) + |\delta| \text{sgn}(H(\omega_i) - D(\omega_i)) > u(\omega).$$

The resulting equations are as above, (9.4), but

$$(D_1)_{i,i} = \begin{cases} 0 & \text{if } (9.6) \text{ or } (9.7) \text{ at } \omega_i \\ (-1)^i & \text{else} \end{cases}$$

$$(D_2)_{i,i} = \begin{cases} u(\omega_i) & \text{if } (9.7) \text{ at } \omega_i \\ l(\omega_i) & \text{if } (9.6) \text{ at } \omega_i \\ D(\omega_i) & \text{else} \end{cases}$$

As above there is a matrix $Q$ such that $QM_1 = 0$, and by applying $Q$ to (9.4) we obtain again a generalized eigenvalue problem. Except for the differences in $D_1$ and $D_2$, the interpolation step is the same.

Updating the reference set from one iteration to the next in the constrained Remez algorithm requires some more care than it does in the unconstrained version. While the unconstrained version uses $|H(\omega_i)| - |\delta|$ to select new reference points (this value should be positive), the unconstrained version should use this value at points where the constraint is not violated and one of the values, $l(\omega_i) - H(\omega_i)$ or $H(\omega_i) - u(\omega_i)$, where the upper or lower bound constraint is violated.
9.5 Examples

Example 1: The 7 minimum phase filters with a total number of poles and zeros equal to 12 and having an even number of poles (were designed for an ideal low pass filter with a pass band edge at 1309\(\pi\)/2048 and a stop band edge at 1422\(\pi\)/2048 (so that the band edge is at 2\(\pi\)/3). The total number of grid points used was 2049 for the interval \([0, \pi]\) including the end points and the zero weighted transition band. The filter having 2 poles is illustrated in Figure 9.1. The Chebyshev error as a function of the number of zeros is plotted in Figure 9.2. As can be seen, the use of two poles significantly reduces the Chebyshev error of the best approximation. For this example, when the number of zeros is greater than 6, the optimal filter possesses zeros lying off the unit circle.

Example 2: A filter is designed with 8 zeros and 3 poles whose magnitude squared frequency response is nearly degenerate. The ideal frequency response is a low pass filter with a pass band edge at 1426\(\pi\)/2048 and a stop band edge at 1475\(\pi\)/2048. The total number of grid points used was 2049. This is an example in which updating the reference set from iteration to iteration requires small perturbations, for the usual exchange methods lead to failure.

The Chebyshev error for the resulting filter, shown in Figure 9.3, is \(E_{g,3}^* = 0.040120\). A pole and a zero almost cancel as is typical for nearly degenerate approximations. Here, the zero is at \(z = -1\) and the pole is just inside the unit circle on the real line. Here, it makes sense for practical considerations to decrease the number of poles and zeros by one each. The resulting lower order filter, shown in Figure 9.4, is no longer nearly degenerate and the Chebyshev error is only slightly greater at \(E_{r,2}^* = 0.040581\). Notice that the lower order filter has a scaled “extra” ripple.

In general, as the best approximation for a fixed number of poles and zeros becomes more degenerate, the size of the extra ripple in the best approximation of lower order rises to the Chebyshev error. When the best approximation is in fact degenerate, then there is exact pole-zero cancelation and the best approximation of lower order
is identical.

If the degree of the approximating function is reduced by reducing only the number of poles by one, then one obtains $E_{8,2}^* = 0.040483$. If the number of zeros is reduced by one, then one gets $E_{7,3}^* = 0.040487$. As expected, $E_{8,2}^*$ and $E_{7,3}^*$ lie between $E_{8,3}^*$ and $E_{7,2}^*$, suggesting that, since $E_{8,3}^* \approx E_{7,2}^*$, the best trade-off between complexity and quality of approximation is given by the filter with 7 zeros and 2 poles. That is, when an approximation is nearly degenerate, reducing both the number of zeros and poles by one generally makes sense.

### 9.5.1 Use an Even Number of Poles!

It should be noted that these near degeneracy problems generally arise when some zeros contribute to the passband and the number of poles is odd. In this case, one of the poles must lie on the real line. When zeros lie inside the unit circle, the real pole does little to contribute to the response and is often located near $z = -1$. Therefore, it is advisable to use an even number of poles in such cases.

### 9.6 Summary

This correspondence describes a Remez algorithm for the magnitude squared design of IIR digital filters in the frequency domain. The number of poles and zeros (away from the origin) can be chosen arbitrarily and the zeros do not have to lie on the unit circle. The generalized eigenvalue problem was used to solve the relevant nonlinear equations of the interpolation step. Nonnegativity constraints were imposed so that spectral factorization can be employed. Reference set degeneracy is overcome by adjusting the reference set using a systematically obtained sequence of successively smaller perturbations. In addition, the algorithm appears to be numerically stable. One requirement of the algorithm is an initial reference set for which a solution $A(\omega)$ to (9.3) is positive over $[0, \pi]$.

One example illustrated the way in which the Chebyshev error of the optimal
filter behaves as a function of the total number of zeros when the number of poles and zeros is kept constant. A second example examined a nearly degenerate best approximation and aspects of near degeneracy were discussed.

![Pole-Zero Plot](image1)

**Figure 9.1**: The filter for example 1 having 2 poles.
Figure 9.2: The Chebyshev error as a function of the number of zeros for example 1.
Figure 9.3: The filter for example 2 for which $m = 8$ and $n = 3$. 
Figure 9.4: The filter for example 2 for which $m = 7$ and $n = 2$. 
Chapter 10

Monic Polynomials Approximating Zero

10.1 Introduction

In this chapter, we revisit an approximation problem addressed by F. S. Souto in his 1970 dissertation "A Mixed Flat and Equal-ripple Criterion for Filter Design" [104].

The $n^{th}$ degree monic polynomials $x^n$ and $\frac{1}{2^n} T_n(x)$ (the monic Chebyshev polynomial of degree $n$) both approximate 0 in the interval $[0,1]$. From the set of monic $n^{th}$ degree polynomials, the polynomial $x^n$ is an optimal approximation to 0 according to a Taylor series (derivative) criterion. The Chebyshev polynomial is an optimal approximation to 0 according to the the $L_\infty$ or Chebyshev criterion. Note that $x^n$ is very flat at $x = 0$ and that $T_n(x)$ possesses ripples of equal size.

Souto was interested in polynomials that mix these two approximation criteria. Specifically, he sought a monic polynomial with a specified number of roots at $x = 0$ and a specified number of equal-sized ripples over $[0,1]$. Such polynomials are optimal approximations to zero over $[0,1]$ in the Chebyshev sense subject to the constraint on the number of roots at $x = 0$. See Figure 10.1.

Both $x^n$ and $T_n(x)$ are used in filter design, and it is of interest to expand the available methods for filter design by using these mixed-type polynomials, as Souto did in [104]. One reason for considering such polynomials in filter design is the flexibility they provide. The number of derivative constraints directly influences the Chebyshev norm of the monic polynomial of fixed degree that best approximates 0 over $[0,1]$ in the Chebyshev sense. For a fixed degree, the minimum achievable Chebyshev norm varies with the number of derivative constraints at $x = 0$.

Mixed flat-Chebyshev polynomials can be produced easily with a Remez-type al-
algorithm. Here, however, we are interested in the existence of more direct methods. One reason for this interest is that direct methods may be applicable to the problem of approximation by rational functions. Although, the Remez algorithm can be applied to approximation by rational functions, several convergence problems arise, so motivating other techniques.

Properties of mixed flat-Chebyshev polynomials have been studied by Borwein and Erdélyi in [10], in which these polynomials are called incomplete Chebyshev polynomials (for some of their coefficients are zero). However, we have not seen the approach for constructing these polynomials, given in this chapter, elsewhere.

### 10.2 Chebyshev Polynomials

In this section the Chebyshev polynomials are derived in a way that can be (roughly) generalized to the desired polynomials.

Consider the polynomials $T_n(x) - 1$ and $T_n(x) + 1$ with $n$ even. The roots of these polynomials in the interval $(0,1)$ are double roots, and the roots of $T_n(x) - 1$ at $x = 1$ and $x = -1$ are simple. Note that they have no other roots and that $T_n(x) - 1$ and $T_n(x) + 1$ share no roots. Because the derivative of $T_n(x)$ is zero exactly at the roots of $T_n(x) + 1$ and $T_n(x) - 1$ in $(0,1)$, $T_n'(x)$ shares its roots (which are simple) with $(T_n(x) - 1)(T_n(x) + 1)$. By counting roots and degrees, and by making leading coefficients equal, the well known relationship

$$n^2(T_n^2(x) - 1) = (T_n'(x))^2(x^2 - 1)$$  \hspace{1cm} (10.1)

is obtained [50]. The monic polynomial $M(x) = \frac{1}{2^{n-1}}T_n(x)$ satisfies $n^2(M^2(x) - \frac{1}{2^{n-1}}) = (M'(x))^2(x^2 - 1)$.

While most presentations of the Chebyshev polynomials solve this differential equation to obtain $\arccos(T_n) = n \arccos(x)$ at this point, here we take another approach. By substituting a generic polynomial with unknown coefficients into (10.1) a set of nonlinear equations for the coefficients of $T_n$ is obtained. For example, if
$n = 4$, then the equations obtained by substituting $M(x) = x^n + \sum_{i=0}^{n-1} c_i x^i$ into the differential equation above require that the following eight expressions be 0:

\begin{align*}
-8c_3 & \quad (10.2) \\
-16 - 16c_2 - 7c_3^2 & \quad (10.3) \\
-20c_2c_3 - 24c_1 - 24c_3 & \quad (10.4) \\
-26c_1c_3 - 32c_0 - 16c_2 - 9c_3^2 - 12c_2^2 & \quad (10.5) \\
-12c_2c_3 - 8c_1 - 28c_1c_2 - 32c_0c_3 & \quad (10.6) \\
-32c_0c_2 - 6c_1c_3 - 4c_2^2 - 15c_3^2 & \quad (10.7) \\
-4c_1c_2 - 32c_0c_1 & \quad (10.8) \\
\frac{1}{4} - c_1^2 - 16c_0^2 & \quad (10.9)
\end{align*}

Although these are, at first glance, nonlinear equations in the coefficients $c_i$, notice that they can be solved simply by making a sequence of substitutions. The first equation in this list requires that $c_3$ is zero (this is as expected since $T_n(x)$ is an even polynomial for even $n$). Having ascertained the value of $c_3$, the second equation in this list gives $c_2 = -1$. When the equations are used in this order, the coefficients of $T_4(x)$ are easily obtained. The same is true for all $T_n(x)$.

### 10.3 Mixed-type Polynomials

To obtain polynomials approximating 0 by a mixed flat-Chebyshev criterion, the following form will be used:

$$F_{k,n}(x) = x^k(A(x))^2 \quad (10.10)$$

where $A(x)$ is a polynomial of degree $n$. The degree of $F_{k,n}(x)$ is $2n + k$. This form incorporates the flatness requirement at $x = 0$. A polynomial $A(x)$ is sought, so that $F(x)$ has equal-sized ripples over $[0,1]$.

In order to carry out a procedure to find $A(x)$ that is similar to the procedure used above to obtain the Chebyshev polynomials, it is first necessary to find an
equation analogous to (10.1). The polynomial $F_{k,n}(x)$ we seek has $n$ double roots in the interval $(0,1)$, equal-sized ripples of size $L$ in $(0,1)$, and takes on the value $L$ at $x = 1$. See Figure 10.1. As above, we consider the roots of the polynomials $F(x)$, $F(x) - L$, and the roots of the derivative of $F(x)$. It follows that for the polynomial $F(x)$ we seek, $F(x)(F(x) - L)$ shares a number of roots with $F'(x)$. As above, the roots of $F(x)(F(x) - L)$ in $(0,1)$ are double roots and the roots of $F'(x)$ are simple. This suggests that a specific relationship similar to (10.1) must exist between $F(x)(F(x) - L)$ and $(F'(x))^2(x - 1)$, however, the degrees do not match up. It is only possible to assert that they must share a number of roots. Using (10.10) leads to a relationship between $x^k A(x)^2 - 1$ and $(2xA'(x) + kA(x))(x - 1)$. By matching degrees and leading coefficients, this leads to the requirement, that for the $n^{th}$ degree polynomial $A(x)$ we seek, there must exist a positive number $L$ and a monic polynomial $U(x)$ of degree $k - 1$ such that

\[
(2n + k)^2(x^k A(x)^2 - L) = (2xA'(x) + kA(x))^2(x - 1)U(x).
\]  

(10.11)

Unfortunately, this system of equations is not as simple as (10.1). After substituting generic monic polynomials with unknown coefficients into (10.11) and equating like powers of $x$, the resulting equations are not essentially linear as are those of (10.1). For example, the equations associated with $F_{4,2}(x)$ require that each of the following eight expressions be zero (where $A(x) = a_0 + a_1 x + x^2$ and $U(x) = u_0 + u_1 x + u_2 x^2 + x^3$):

\[
-64 u_2 + 32 a_1 + 64
\]  

(10.12)

\[
-96 u_2 a_1 + 28 a_1^2 - 64 u_1 + 64 u_2 + 96 a_1 + 64 a_0
\]  

(10.13)

\[
-36 u_2 a_1^2 - 96 u_1 a_1 + 96 u_2 a_1 - 64 u_2 a_0 + 36 a_1^2 + 80 a_1 a_0 - 64 u_0 + 64 u_1 + 64 a_0
\]  

(10.14)

\[
-36 u_1 a_1^2 + 36 u_2 a_1^2 - 48 u_2 a_1 a_0 - 96 u_0 a_1 + 96 u_1 a_1 - 64 u_1 a_0 + 64 u_2 a_0 + 48 a_1 a_0 + 48 a_1^2 + 64 u_0
\]  

(10.15)

\[
-36 u_0 a_1^2 + 36 u_1 a_1^2 - 48 u_0 a_1 a_0 + 48 u_2 a_1 a_0 - 16 u_2 a_0^2 + 96 u_0 a_1 - 64 u_0 a_0 + 64 u_1 a_0 + 16 a_1^2
\]  

(10.16)

\[
36 u_0 a_1^2 - 48 u_0 a_1 a_0 + 48 u_1 a_1 a_0 - 16 u_1 a_0^2 + 16 u_2 a_0^2 + 64 u_0 a_0
\]  

(10.17)
16 \left(3u_0a_1 - u_0a_0 + u_1a_0\right) a_0 \quad (10.18)

16u_0a_0^2 - 64L \quad (10.19)

Note that $U(x)$ does appear linearly in (10.11), and so the coefficients of $U(x)$ can be found in terms of the coefficients of $A(x)$ by linear algebra. However, after $U(x)$ is eliminated from the equations, the resulting equations are not inviting. Again, using $F_{4,1}(x)$ for example, $u_2 = \frac{a_1}{2} + 1$, $u_1 = -\frac{5a_1}{16} + \frac{a_1}{2} + 1 + a_0$ and $u_0 = \frac{3a_1}{16} - \frac{5a_1}{16} + \frac{a_1}{2} - \frac{3a_1a_0}{4} + a_0 + 1$. The equations (10.12-10.19) above, after eliminating $U(x)$ become:

\[-\frac{27a_1^4}{4} + 12a_1^3 + 32a_1^2a_0 - 20a_1^2 - 48a_1a_0 - 16a_2^2 + 32a_1 + 64a_0 + 64 \quad (10.20)\]

\[-\frac{a_1}{4}\left(27a_1^4 - 72a_1^3 - 120a_1^2a_0 + 120a_1^2 + 288a_1a_0 + 32a_3 - 192a_1 - 384a_0 - 384\right) \quad (10.21)\]

\[\frac{27a_1^4}{4} - 9a_1^3a_0 - \frac{45a_1^4}{4} - 15a_1^3a_0 + 41a_1^3a_0 + 18a_1^2 + 16a_1a_0 - 48a_1a_0^2 - 16a_3^2 + 36a_3^2 + \]

\[32a_1a_0 + 64a_3^2 + 64a_0 \quad (10.22)\]

\[3(3a_1^3 - a_1^2a_0 - 5a_1^2 - 12a_1a_0 + 4a_3 + 8a_1 + 16a_0 + 16)a_1a_0 \quad (10.23)\]

\[3a_1^3a_0^2 - 5a_1^2a_0^2 - 12a_1a_0 + 8a_1a_0 + 16a_3^2 + 16a_3^2 - 64L \quad (10.24)\]

Using the purely lexical ordering of monomials with $a_1 < a_0$, the Gröbner basis [19] of the first two of these equations are given by the following polynomials:

\[31050a_0^6 - 427491a_1^7 + 3623544a_0^6 - 24731232a_1^5 - 23761024a_0^5 + 63315456a_1^3 + 40826880a_0^3 + 62595072a_2^5 - 234078208a_1 - 25038028a_0 - 250380288 \quad (10.25)\]

\[-2808a_0^6 + 56241a_1^7 - 548712a_0^6 + 4023840a_1^5 - 10057216a_0^5 - 46133760a_1^3 + 6377472a_0^3 + 62595072a_1a_0 + 18120704a_1 \quad (10.26)\]

and

\[27a_0^6 - 432a_0^5 + 4032a_0^4 - 29184a_0^3 + 35840a_0^2 + 57344a_0^1 - 196608a_0^0 - 262144a_1^2 - 262144a_1 \quad (10.27)\]

Notice that (10.27) is a polynomial in a single variable. Therefore, the coefficient $a_1$ of $A(x)$ can be found by computing the appropriate root of (10.27). The coefficient $a_0$ of $A(x)$ can then be computed by (10.26). The roots of (10.27) are
\{0, \ -1.55857, \ -0.50503 \pm 0.82455i, \ 1.91389 \pm 9.02319i, \ 2.19563 \pm 1.65705i, \ 10.34958\} \quad (10.28)

Discarding the roots off the real line, the appropriate root can be chosen by graphing the resulting polynomial \( F(x) \) over \((0,1)\) for each real root. It turns out that the appropriate root to use for the value of \( a_1 \) is \(-1.55857\). The resulting value for \( a_0 \) from (10.26) is \(0.577508\). The polynomial we seek is thus

\[
F_{4,2}(x) = x^4 \left( 0.577508 - 1.55857x + x^2 \right)^2.
\]

The size of the ripple, \( L \), is given by (10.19), which evaluates to \(3.5849E - 04\). The polynomial \( U(x) \) corresponding to this solution is \(0.0042998 + 0.0391110x + 0.2207130x^2 + x^3\). The frequency response magnitude squared of an analogue filter constructed using \( F_{4,2}(x) \) is illustrated in Figure 10.2.

In principal, the multi-variable system of polynomial equations for the coefficients of \( A(x) \) and \( U(x) \) given by (10.11), could be solved using the Gröbner basis for the computation of \( F_{k,n}(x) \) for higher \( k, n \), however, the software currently being used (Maple) can handle only disappointingly small degree \((n \leq 4)\) problems.

The fact that the Gröbner basis gives a solution to this problem is very interesting. Roughly, given a system of multi-variable polynomials, the Gröbner basis is a new system of multi-variable polynomials having the same solution set. The Gröbner basis is similar to the row-reduced echelon form in linear algebra in that the last polynomial in the basis is a polynomial in a single variable. When the roots of this polynomial are known, they can be substituted into the other polynomials of the basis, etc. If there are a finite number of solutions to the original system of equations, this use of the Gröbner basis make it possible to compute them by computing the roots of a set of polynomials.

Other examples appear in the figures.
10.4 Remarks

The Chebyshev polynomials have many special properties. Are any of these properties also satisfied by the mixed-type polynomials $F_{k,n}(x)$? For example the Chebyshev polynomials obey a second order linear differential equation as well as the first order nonlinear differential equation (10.1). There also exist recurrence relations for the Chebyshev polynomials. Most important for us, however, are the locations of the roots of the polynomials. The roots of the Chebyshev polynomials are well known to be samples of the cosine function, but, there is no similarly simple expression for the roots of $F_{n,k}$. Erdélyi, however, has found estimates of the locations of the zeros of these polynomials [25].

10.5 Conclusion

This chapter has described a novel technique using Gröbner bases for constructing monic polynomials of the form $x^k P(x)$ that best approximate 0 in the Chebyshev sense. To our knowledge, of the techniques for constructing such polynomials, this is the first that does not employ an iterative numerical algorithm (for example, Remez-like algorithms and the differential correction algorithm). The problem of constructing such polynomials is a classical one in approximation theory. In practice, the use of Gröbner bases as described in this chapter is suitable only for relatively small degree problems. However, using an appropriate change of variables and using a different ordering of monomials, may make the use of Gröbner bases more practical for greater degrees.
A mixed flat-Chebyshev polynomial; $k = 4$, $n = 2$.

Figure 10.1: $F_{4,2}(x) = x^4 (0.57750 - 1.55857x + x^2)^2$, $L = 3.5849E - 04$.

Figure 10.2: Filter response corresponding to $F_{4,2}(x)$. 

Figure 10.3: \( F_{4,3}(x) = x^4(-0.309460 + 1.464776678x - 2.15139068x^2 + x^3)^2, \) \( L = 1.5413E-05. \)

Figure 10.4: Filter response corresponding to \( F_{4,3}(x) \).
Figure 10.5: $F_{3,2}(x) = x^3 (0.5148440674 - 1.488704097x + x^2)^2$, $L = 6.8330E - 04$

Figure 10.6: $F_{2,2}(x) = x^2 (42 - 24 \cdot 3^{1/2} + (9 - 6 \cdot 3^{1/2})x + x^2)^2$, $L = (52 - 30 \cdot 3^{1/2})^2$
Chapter 11

Conclusion

In this dissertation, we have described several new techniques for the design of digital filters. We have introduced, in Chapter 2, a new criterion for the constrained least square design of symmetric FIR filters. This formulation overcomes several fundamental problems in the way in which the discontinuity of the ideal lowpass filter response is traditionally treated. For this formulation, simple and robust programs are given in Chapters 2 and 3. Chapters 4 and 5 give algorithms that complement the Parks-McClellan algorithm, the most widely known digital filter design technique. In Chapters 6 through 9 we give new design techniques for two types of filters for which the approximation problem is substantially more difficult: (i) IIR filters with an unequal numerator degree and denominator degree and (ii) non-symmetric FIR filters with magnitude and group delay approximation. The design techniques of Chapters 6 and 7 are analytic. In Chapters 6 and 10, Gröbner bases from computational algebraic geometry, are used to obtain interesting new results for difficult problems in approximation theory relevant to filter design.

11.1 Summary

To summarize some of the types of digital filters for which design techniques are described in this thesis, a set of representative filters are shown on the following pages. Figures 11.1 through 11.6 show a representative lowpass filter from the chapters, illustrating the different types of responses discussed in this thesis. For each of the filters in Figures 11.1 through 11.6, the sum of the numerator degree and denominator degree is 22. In addition, they are all designed to have a half-magnitude frequency
at $0.6\pi$ ($M(0.6\pi) = \frac{1}{2}$). In the figures, the deviation from 1 in the stopband is constrained to be at most, or specified to be, 0.015; the deviation from 0 in the stopband, 0.03.

![Frequency response graph](image)

![Pole-zero plot graph](image)

![Group delay graph](image)

![Impulse response graph](image)

Figure 11.1: A constrained least square filter, representative of Chapter 2.

To achieve the specified half-magnitude frequency $0.6\pi$ for the filters shown in the following figures, some of the design techniques described in the chapters needed slight modification.

1. The constrained least square filters of Chapter 2 were not designed to meet a half-magnitude frequency specification. However, as noted in Chapter 2, linear
Figure 11.2: An equi-ripple filter, representative of Chapter 4.
Figure 11.3: An FIR filter having a flat monotonic passband and an equi-ripple stopband, representative of Chapter 5.
Figure 11.4: A maximally-flat nearly symmetric filter with reduced delay, representative of Chapter 6.
Figure 11.5: A generalized Butterworth filter, representative of Chapter 7.
Figure 11.6: A generalized Chebyshev II filter, representative of Chapter 8.
constraints can be easily included. The filter shown in Figure 11.1 was obtained by including the half-magnitude constraint (a linear constraint) among the constraints used on each iteration. The weighting function used in the design of the filter shown in Figure 11.1 was taken to equal 1 in \([0, 0.6\pi)\) and 2 in \((0.6\pi, 1]\).

2. The filters discussed in Chapter 5 were not designed to meet both a specified edge frequency and a specified stopband ripple size simultaneously. However, by reducing the number of passband flatness constraints by one, this can be achieved. The appropriate number of flatness constraints at \(\omega = 0\) can be determined by constructing a specification table as in done in Chapter 8.

3. To achieve a specified half-magnitude frequency in Chebyshev II filter design as in Figure 11.6, use \(y = (\frac{1}{2}, 0, \delta, 0, \delta, \ldots)^t\) instead of \(y = (\delta, 0, \delta, 0, \ldots)\), (see Equation 8.27).

11.2 Open Problems

We feel that several very interesting problems (related to this dissertation) remain open. One of these problems regards exchange algorithms for rational functions. A second problem regards analytic design techniques for maximally-flat filter design with magnitude and group delay approximation.

11.2.1 Rational Exchange Algorithms

The problem to which we are referring is the development of an exchange algorithm for constructing rational functions with an alternation property. By an exchange algorithm, we mean an algorithm that iteratively (i) solves an interpolation problem over a set of reference points and (ii) updates the reference points by a set of simple exchange rules. The algorithm of Remez [71, 83] and of Hofstetter et al. [41] are examples. It should be noted that other algorithms, like the differential correction algorithm [8] and the zero-shifting algorithm [60], while useful, are not exchange al-
algorithms as we mean here. The differential correction algorithm (an iterative linear programming method) is computationally burdensome and does not use the alternation property. The zero-shifting algorithm uses the alternation property, but the way in which the reference points (in this case, the zeros) are updated is not an exchange rule as in the algorithms of Remez and Hofstetter et al.

The difficulty at the center of exchange algorithms for rational functions is the occurrence of poles in the interval of interest. In Chapters 8 and 9 we gave methods (exchange rules) for overcoming this situation when it arises during the course of the algorithm (but not on the first iteration). The methods we gave in Chapters 8 and 9 employ the reference points from an earlier iteration. In Chapter 8 we found that a suitable initialization can be obtained by using the zero-shifting algorithm (that is not an exchange algorithm). In Chapter 9 we did not address the problem of initialization.

A set of exchange rules, applicable to the occurrence of poles in the interval of interest, that does not depend on reference points from an earlier iteration will eliminate the initialization problem. For in this case, a viable initialization is unnecessary (by viable, we mean that no poles occur in the interval of interest). To our knowledge, so far no such set of exchange rules is known. A set of such exchange rules, for the Remez algorithm, may not exist — recall that the problems with degeneracy (see Chapter 9) can make this a more difficult problem. However, an exchange algorithm like that of Hofstetter et al. (which yields a "maximum ripple" function), is more likely. Such an algorithm would be extremely interesting and would be very helpful in developing an understanding of Chebyshev approximation by rational functions.

11.2.2 Magnitude and Group Delay Approximation for Maximally-Flat Filters

A second open problem of great interest is the design of maximally-flat digital filters with magnitude and group delay approximation. While Chapter 6 gives the solution
to this problem for a family of FIR filters ("region I" filters, see Chapter 6, Table 6.8), the problem, as it is treated in Chapter 6 remains open for "region II" FIR filters and for IIR filters. (While Gröbner bases can, in principle, be used to solve these problems, in practice the computational requirements for a Gröbner basis solution preclude it.) We should remark that by magnitude and group delay approximation of (lowpass) maximally-flat filters, we here mean, design via a set of flatness constraints on the magnitude at $\omega = 0$ and at $\omega = \pi$ and on the group delay at $\omega = 0$, as in Chapter 6.

Table 11.1 summarizes techniques for maximally-flat lowpass filter design. Note that stable IIR filters (with rational transfer functions) can not have exactly linear phase (hence "None" appears in that category). Also, note that a linear phase FIR maximally-flat filter can be spectrally factored to obtain a magnitude only design — so Herrmann’s expressions are applicable to that case (hence the arrow symbol in that category). Table 11.1 clarifies the relationships among the design problems and their solutions.

An additional problem, not treated in this dissertation is the bandpass case. Expressions, like those given in Chapter 7 are yet to be derived for maximally-flat IIR bandpass filters having unequal numerator and denominator degrees.

Table 11.1 : Maximally-flat lowpass filter design: categorization of techniques.

<table>
<thead>
<tr>
<th></th>
<th>FIR</th>
<th>IIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear phase</td>
<td>Herrmann [38]</td>
<td>None</td>
</tr>
<tr>
<td>Magnitude approximation only</td>
<td>$\uparrow$</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>Magnitude and group delay approximation</td>
<td>Chapter 6</td>
<td>Open problem</td>
</tr>
</tbody>
</table>
11.3 Problem Formulation and Software

Throughout this dissertation, the formulation of the relevant approximation problem has been a key factor in developing the techniques here-in. For example, the constrained least square approach of Chapters 2 and 3 are well posed problems for arbitrarily tight constraints on the peak errors — this is due to the way in which the approximation around the discontinuity is achieved. In Chapter 6, a passband monotonicity condition yielded a well posed problem with a unique solution, even with a continuous variation of the specified DC group delay and half-magnitude frequency. That monotonicity condition also made it possible to distinguish two distinct regions in the set of solutions (see Table 6.8) — a distinction upon which the analytic technique given in that chapter depended. The design of the generalized digital Butterworth and Chebyshev filters in Chapters 7 and 8 was also made possible by stipulating a passband monotonicity condition and by constructing appropriate specification tables.

The way in which the problem is formulated affects the way in which the user interacts with the program that implements the design technique. It is important that this be done well for the technique to be useful in practice. Generally this means that the way in which the user states specifications to the program be simple yet flexible. Throughout this dissertation, we have also attempted to keep the techniques computationally efficient. This is another criteria for a technique to be useful in practice. Mathworks intends to include, in the Matlab 5 Signal Processing Toolbox, the constrained least square techniques of Chapters 2 and 3 and the generalized Butterworth design technique of Chapter 7, suggesting that those techniques do satisfy such criteria.
Bibliography


[41] E. Hofstetter, A. Oppenheim, and J. Siegel. A new technique for the design of 
nonrecursive digital filters. In *Proc. of the Fifth Annual Princeton Conference 
on Information Sciences and Systems*, pages 64–72, October 1971. Also in [79].

[42] E. Hofstetter, A. Oppenheim, and J. Siegel. On optimum nonrecursive digi-
tal filters. In *Proc. of the Ninth Allerton Conference on Circuits and System 

[43] L. B. Jackson. An improved Martinez/Parks algorithm for IIR design with 
unequal numbers of poles and zeros. *IEEE Trans. on Signal Processing*, 

[44] B. C. Jinaga and S. C. Dutta Roy. Coefficients of maximally flat low and 
high pass nonrecursive digital filters with specified cutoff frequency. *Signal 

differential correction algorithm for rational approximation. *Mathematics of 

filters with maximally-flat pass and stop bands. In IEEE ASSP Soc. Digital 

nonrecursive filter by multiple use of the same filter. *IEEE Trans. on Acoust., 


[97] I. W. Selesnick, M. Lang, and C. S. Burrus. Magnitude squared design of recursive filters with the Chebyshev norm using a constrained rational Remez


