



UNIVERSITY OF MANITOBA

**Design of Approximately Flat Group
Delay Selective IIR Filters Based on
Wave Digital Allpass Filters**

by

Jing Li

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba
Canada

© 2002



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-76787-6

THE UNIVERSITY OF MANITOBA
FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION

**Design of Approximately Flat Group Delay Selective IIR Filters
Based on Wave Digital Allpass Filters**

BY

Jing Li

**A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of
Manitoba in partial fulfillment of the requirement of the degree**

of

MASTER OF SCIENCE

Jing Li © 2002

Permission has been granted to the Library of the University of Manitoba to lend or sell copies of this thesis/practicum, to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film, and to University Microfilms Inc. to publish an abstract of this thesis/practicum.

This reproduction or copy of this thesis has been made available by authority of the copyright owner solely for the purpose of private study and research, and may only be reproduced and copied as permitted by copyright laws or with express written authorization from the copyright owner.

Abstract

The wave digital allpass filter is a computationally efficient signal processing building block. In this thesis, we review the properties of allpass filters and consider the applications in digital filtering. The digital filter structures under consideration are the conventional cascade-form realization and the parallel connection of two allpass filters (wave digital lattice filters).

Given the attenuation specifications, the filter parameters as well as the slope of the linear phase response are optimized in such a way that the group delay is approximated flat in the passband. The problem of allpass filter design for group delay equalization is solved by using the nonlinear least squares method. On the other hand, the problem for group delay approximation can be solved by using pole-zero placement and a modified Remez algorithm, respectively.

The proposed design scheme for group delay equalization consists of two basic steps. The first step involves obtaining a filter with the desired attenuation based on pole-zero placement. An important feature of the proposed method is the possibility of placing the poles and zeros in their optimum positions. The method is also modified to design IIR filters with approximately flat group delay in the passband. In the second step, this filter

is cascaded with an allpass compensation filter that is designed by the nonlinear least squares method. For IIR digital filters in the form of a parallel connection of two allpass subfilters, the modified Remez algorithm is applied. The rational approximation problem is transformed into a linear problem. The proposed algorithm converges after a few iterations, and the attenuation characteristic of the IIR digital filter is equiripple. Several examples are included illustrating the efficiency of the proposed methods.

Acknowledgments

I wish to express my sincere gratitude to my supervisor Prof. G. O. Martens for his unique guidance, continuous encouragement and suggestions throughout the years I have been working on these topics.

I would like to thank to the members of my committee, Prof. E. Shwedyk and Prof. H. C. Finlayson for their valuable suggestions.

I would also like to thank my friends and colleagues, especially Mr. Wong ThieuHoa, Dr. Burton G. Ayles, Dr. Jack A. Mathias, Ms. Zhang Yuhong, Mr. Saleh Sanussi Mneina, Mr. Huang Bin and Mr. James Millar.

I am deeply indebted to my parents for love, support, understanding and encouragement at all times.

Table of Contents

ABSTRACT	I
ACKNOWLEDGMENTS.....	III
LIST OF FIGURES.....	VII
LIST OF TABLES.....	XII
 CHAPTER 1 INTRODUCTION	 1
 CHAPTER 2 WAVE DIGITAL ALLPASS FILTER AND STRUCTURES	 6
2.1 ALLPASS FILTERS	6
2.2 LATTICE NETWORKS	10
2.3 WAVE DIGITAL ALLPASS STRUCTURES INVOLVING TWO-PORT ADAPTORS	13
2.3.1 <i>First-order allpass section</i>	13
2.3.2 <i>Second-order allpass section</i>	14
 CHAPTER 3 REALIZATION OF WAVE DIGITAL LATTICE FILTERS AS THE SUM OF WAVE DIGITAL ALLPASS FILTERS	 18
3.1 DIGITAL FILTERS MADE AS THE SUM OF TWO ALLPASS FUNCTIONS	18
3.2 WAVE DIGITAL LATTICE FILTERS	24

CHAPTER 4 APPROXIMATELY LINEAR PHASE USING POLE-ZERO

PLACEMENT	27
4.1 PRINCIPLES	27
4.2 CHARACTERISTIC FUNCTION AND ATTENUATION CALCULATION	29
4.3 THE MODIFIED POLE PLACEMENT AND ITS APPLICATION	30
4.3.1 <i>The modified pole-zero placement</i>	30
4.3.2 <i>Approximately linear phase approach</i>	35

CHAPTER 5 GROUP DELAY EQUALIZATION USING A CASCADED WAVE

DIGITAL ALLPASS FILTER	37
5.1 GROUP DELAY EQUALIZATION	37
5.2 GROUP DELAY EQUALIZATION USING NONLINEAR LEAST SQUARES METHOD	39
5.2.1 <i>Defining the group delay objective</i>	40
5.2.2 <i>Gauss-Newton method</i>	41
5.2.3 <i>Design algorithm</i>	44

CHAPTER 6 APPROXIMATELY LINEAR PHASE USING A SUM OF WAVE

DIGITAL ALLPASS FILTERS	46
6.1 THE PHASE RESPONSE OF IIR DIGITAL FILTERS USING THE SUM OF TWO ALLPASS SUBFILTERS	46
6.2 APPROXIMATELY LINEAR PHASE APPROACH	49
6.2.1 <i>Phase approximation problem</i>	49

6.2.2 <i>Design procedure</i>	50
CHAPTER 7 DESIGN EXAMPLES	54
7.1 EXAMPLES USING POLE-ZERO PLACEMENT	54
7.2 EXAMPLES USING GROUP DELAY EQUALIZATION	61
7.3 EXAMPLES USING THE SUM OF TWO WAVE DIGITAL ALLPASS FILTERS	81
7.4 SUMMARY	89
CHAPTER 8 CONCLUSIONS	92
REFERENCES	94
APPENDIX	101

List of Figures

Figure 2-1 Pole-Zero plot of the real coefficient allpass function of Eq. (2-2).....	7
Figure 2-2 The phase and group delay of the allpass function of Eq. (2-2).....	8
Figure 2-3 A lattice network with two-port terminations.....	10
Figure 2-4(a) wave-flow diagram of the lattice structure.....	12
Figure 2-4(b) simplified wave-flow diagram.....	13
Figure 2-5 A first-order allpass section.....	13
Figure 2-6 a first-order wave digital allpass filter structure.....	14
Figure 2-7 Second-order allpass sections.....	15
Figure 2-8 a second-order wave digital allpass filter structure.....	17
Figure 3-1 Block diagram of an 11-order lattice WDF using wave digital allpass filters.....	25
Figure 4-1 The critical frequencies in the passband and stopband.....	28
Figure 5-1 group delay equalization.....	38
Figure 5-2 The target group delay of an allpass filter design ($y=1$).....	39
Figure 5-3 the error norm vs. the scale factor y	44
Figure 6-1 IIR filter design using a parallel allpass filters.....	47
Figure 6-2 the phase of two allpass subfilters.....	48
Figure 7-1 the zeros of f , g and h in z -domain.....	55
Figure 7-2 The attenuation of the 7-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	56
Figure 7-3 The group delay of the 7-order approximately linear phase WDLF.....	56

Figure 7-4 The phase of the 7-order approximately linear phase WDLF.....	57
Fig. 7-5 The structure of the 7-order approximately linear phase WDLF using pole-zero placement.....	58
Figure 7-6 the zeros of f , g and h in z -domain.....	59
Figure 7-7 The attenuation of the 9-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	60
Figure 7-8 The group delay of the 9-order approximately linear phase WDLF.....	60
Figure 7-9 The phase of the 9-order approximately linear phase WDLF.....	61
Figure 7-10 The structure of the overall 14-order filter consists of 7-order WDLF cascaded 7-order allpass compensation filter.....	63
Figure 7-11 The attenuation of the 14-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	64
Figure 7-12 The group delay of the 14-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	64
Figure 7-13 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband.....	65
Figure 7-14 The zeros and poles of 7-order WDLF and 7-order allpass compensation filter.....	65
Figure 7-15 The zeros and poles of the overall filter.....	66
Figure 7-16 The group delay equalization using different allpass compensation filters..	66
Figure 7-17 The attenuation of the 17-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	68

Figure 7-18 The group delay of the 17-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	68
Figure 7-19 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband.....	69
Figure 7-20 The zeros and poles of 11-order WDLF and 6-order allpass compensation filter.....	69
Figure 7-21 The zeros and poles of the overall filter.....	70
Figure 7-22 The attenuation of the 9-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	71
Figure 7-23 The group delay of the 9-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	72
Figure 7-24 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband.....	72
Figure 7-25 The zeros and poles of 4-order WDLF and 5-order allpass compensation filter.....	73
Figure 7-26 The zeros and poles of the overall filter.....	73
Figure 7-27 The attenuation of the 13-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	75
Figure 7-28 The group delay of 13-order approximately linear phase WDLF (a) in the Whole region, (b) in the passband.....	75
Figure 7-29 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband.....	76

Figure 7-30 The zeros and poles of 6-order WDLF and 7-order allpass compensation filter.....	76
Figure 7-31 The zeros and poles of the overall filter.....	77
Figure 7-32 The gain in dB of the 11-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	78
Figure 7-33 The group delay of the 11-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	79
Figure 7-34 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband.....	79
Figure 7-35 The zeros and poles of 5-order WDLF and 6-order allpass compensation filter.....	80
Figure 7-36 The zeros and poles of the overall filter.....	80
Figure 7-37 7-order approximately linear phase WDLF structure with one branch pure delay.....	82
Figure 7-38 The zeros and poles of the WDLF filter.....	82
Figure 7-39 The attenuation of the 15-order approximately linear phase WDLF (a) in the whole region, (b) in the passband.....	83
Figure 7-40 The group delay in passband of the 15-order approximately linear phase WDLF.....	83
Figure 7-41 The phase in passband of the 15-order approximately linear phase WDLF..	84
Figure 7-42 The phase error of the 15-order approximately linear phase WDLF.....	84
Figure 7-43 The zeros and poles of the WDLF filter.....	87

Figure 7-44 The attenuation of the 25-order approximately linear phase WDLF filter (a) in the whole region, (b) in the passband.....	87
Figure 7-45 The group delay in passband of the 25-order approximately linear phase WDLF filter.....	88
Figure 7-46 The phase in passband of the 25-order approximately linear phase WDLF filter.....	88
Figure 7-47 The phase error of the 25-order approximately linear phase WDLF filter...	89
Figure 7-48 The group delay in the passband obtained by three approximate methods...	90
Figure 7-49 The phase in the passband obtained by three approximate methods.....	90

List of Tables

Table 1 Adaptor coefficients of the filter in example 1.....	55
Table 2 Adaptor coefficients of the filter in example 2.....	59
Table 3 Adaptor coefficients of the filter in example 3.....	62
Table 4 Adaptor coefficients of the filter in example 3.....	67
Table 5 Adaptor coefficients of the filter in example 4.....	71
Table 6 Adaptor coefficients of the filter in example 4.....	74
Table 7 Adaptor coefficients of the filter in example 5.....	78
Table 8 Adaptor coefficients of the filter in Example 6.....	81
Table 9 Adaptor coefficients of the filter in example 7.....	86

Chapter 1

Introduction

The design of digital filters to satisfy a combined magnitude and phase specification is required in many applications including communications, sonar and video signal processing. In the past, FIR filters were used because of the ease with which an exact linear phase could be obtained. However, the filter order required was often high and the resulting large delay was normally unacceptable. In addition, linear phase over the whole band is not usually required.

Traditionally, digital IIR filters have been synthesized by mapping a classical continuous-time filter to its digital equivalent by means of the bilinear transformation. There exist several low-noise and low-sensitivity structures for implementing the resulting filters such as wave digital and lossless discrete integrator (LDI) ladder filters as well as the parallel connection of two allpass filters (wave digital lattice filters) [1-3], [7-8], [28-30], [37-43]. However, these filters suffer from several drawbacks. IIR filters meet the given magnitude criteria with much lower order than FIR filters, but usually at the expense of a highly nonlinear phase response. In order to avoid this problem, many algorithms have been proposed for the simultaneous design of the magnitude and phase responses of IIR filters. The structures of the digital filter are very important for the efficiency of the filtering process. It is essential that filters can be realized using structures with low coefficient sensitivity.

One of the most efficient building blocks for implementing IIR filters are first- and second-order allpass filters [2-8], [12], [50-51]. Digital allpass filters possess unit

magnitude response and prescribed phase characteristics. Traditionally, allpass filters have been used as phase equalizers. Also conventional recursive digital filters can be realized as a parallel connection of two allpass filters. One of the best structures for implementing recursive digital filters is the wave digital lattice filter, which is related to certain analog prototype networks. The wave digital lattice filter consists of a parallel connection of two allpass subfilters. These allpass subfilters can be realized by using first- and second-order sections as basic building blocks. Allpass filters are also the basic building blocks of recursive half-band and N^{th} band filters, which have been found to be very efficient in sample-rate conversion applications.

One of the most difficult problems in digital filter synthesis is the simultaneous attenuation and phase approximation. For solving this problem, a systematic technique has been generated for designing a filter in such a manner that the maximum phase deviation from a linear phase has been minimized for the given attenuation specifications. The most straightforward approach to arrive at an IIR filter having simultaneously a selective attenuation and an approximately linear phase response in the passband is to create the filter in two steps. First, a filter with the desired attenuation is designed. Then, the phase response of this filter is made approximately linear in the passband by cascading it with an allpass phase or group delay equalizer [14-27]. In IIR digital filters in the form of a parallel connection of two allpass subfilters, the modified Remez algorithm is developed based on the use of the phase responses.

The approximation technique calculates stable filters from a given set of constraints on the attenuation and phase characteristics. Most of them need the numerical value of the

group delay, or an analytic expression of the phase. There are three approaches for the linear phase approximation problem in this thesis.

1. Flat approximation using pole-zero placement.
2. L_2 approximation using a nonlinear least square method.
3. L_∞ approximation using a modified Remez algorithm.

For the first approach, we have limited control over the approximation quality in the interval of interest. The pole-zero placement is based on the characteristic function [9-11]. Zeros are placed on the unit circle in the passband and poles are placed on the unit circle in the stopband. For flat group delay, we choose and move a quad of zeros not on the unit circle. There, an iterative procedure is required.

In the L_2 approach, generally, the approximation of group delay is a nonlinear problem. It is difficult to solve. Most approaches presented use (weighted) least squares to match an overall constant group delay. The constant group delay must be chosen carefully, since an inappropriate value easily results in an unstable filter [14-22]. In this thesis, we contribute a new approach to target allpass group delay based on nonlinear least squares. The new approach consists of two steps. First, the Gauss-Newton method is employed to obtain the group delay of the allpass filter. Second, the group delay error for the allpass filter is a quadratic function of the scale factor and the target group delay. There is one minimum value, which can be found by using the bisection method.

It also can be shown that for the L_∞ problem, the solution is uniquely characterized by the alternation theorem. As is well known, for an IIR filter with approximately linear phase, if one of the allpasses consists of pure delay elements, then the other one can be designed to obtain the desired selectivity. Several methods for the design of such systems

have been published [28-33], [37-40]. This thesis deals with the problem again in a more general way. We consider the case that the IIR filter consists of a parallel connection of two general wave digital allpass filters. It uses a modified Remez-type algorithm and deals with the phase of a general allpass filter. The solution of the phase approximation problem is convergent for the given examples.

This thesis is organized as follows:

Chapter 2 introduces the properties of allpass filters and the wave digital allpass filter structures under consideration in this thesis. It is shown how various basic analog circuit relationships and properties possess direct counterparts in digital filtering. A lattice network that operates between equal resistive terminations can be expressed as the difference of two fractions, each having the form of the reflectance of one of the lattice arm impedances with respect to the common termination. When the lattice is lossless, these reflectances reduce to allpass functions.

Chapter 3 describes how an IIR digital filter can be realized as a sum of allpass filters. Then, the wave digital lattice filter is considered in the direct parallel connection.

Chapter 4 illustrates approximately linear phase design using pole-zero placement for the characteristic function. In filter design (e.g. IIR digital filters), however, we need to find the optimum pole and zeros positions. We first present a zero-pole placement algorithm for digital filters with specifications in both the passband and stopband. Then, we apply this algorithm with one quad of zeros not on the unit circle in the passband to obtain a lowpass filter with approximately linear phase.

Chapter 5 shows an allpass equalization approximation approach using a wave digital lattice filter that is cascaded with an allpass as the realization structure. We introduce a

new nonlinear least square algorithm to design the group delay of the allpass and optimize the parameters of the allpass.

Chapter 6 shows the approximately linear phase using the sum of two wave digital allpass filters. In this chapter, a modified Remez –type algorithm is developed based on the use of the phase responses.

Chapter 7 shows several illustrative design examples using the three approximation methods, which are implemented using MATLAB.

Finally, conclusions and future directions for research are presented in Chapter 8.

Chapter 2

Wave Digital Allpass Filter and Structures

The allpass filter is a versatile building block which is quite useful in many signal processing applications [4, 7]. Conventional digital IIR filters can be realized as a parallel connection of two allpass filters. Well-known classes of such filters are the wave digital lattice filters, which are realized as a cascade of first- and second-order allpass sections. In this chapter, we review the properties and structures of allpass filters. Of particular interest are the wave digital allpass structures.

2.1 Allpass Filters

An N^{th} -order allpass filter is given by

$$A(z) = \frac{z^{-N} f(z^{-1})}{f(z)}, \quad (2-1)$$

where $f(z) = \sum_{k=0}^N \alpha_k z^{-k}$ and $\alpha_0 = 1$. Note that if $z = re^{j\omega}$ is a pole of a real coefficient

allpass transfer function, then it has a zero at $z = \frac{1}{r} e^{-j\omega}$. The numerator polynomial is

obtained from the denominator polynomial by reversing the order of the coefficients, and

vice-versa. Equation (2-1) implies that the poles and the zeros of the real coefficient

allpass function show mirror-image symmetry in the z -plane, as shown in Fig. 2-1 for the

following third-order allpass function:

$$A(z) = \frac{z^{-3} - 1.8856z^{-2} + 1.3298z^{-1} - 0.3185}{1 - 1.8856z^{-1} + 1.3298z^{-2} - 0.3185z^{-3}} \quad (2-2)$$

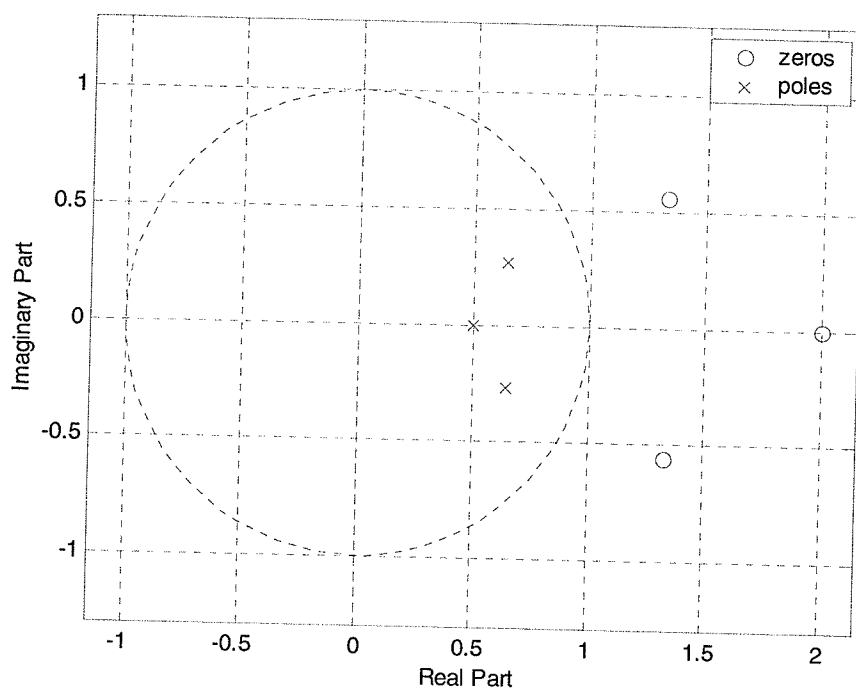


Fig. 2-1 Pole-Zero plot of the real coefficient allpass function of Eq. (2-2)

To show that the magnitude of $A(z)$ is indeed equal to one for all ω , it follows from Equation (2-1) that

$$A(z^{-1}) = \frac{z^N f(z)}{f(z^{-1})}. \quad (2-3)$$

Therefore,

$$A(z)A(z^{-1}) = \frac{z^{-N} f(z^{-1})}{f(z)} \frac{z^N f(z)}{f(z^{-1})} = 1. \quad (2-4)$$

Hence,

$$|A(e^{j\omega})|^2 = A(z)A(z^{-1}) \Big|_{z=e^{j\omega}} = 1. \quad (2-5)$$

It is easily verified that

$$A(e^{j0}) = 1 \quad (2-6)$$

$$A(e^{j\pi}) = (-1)^N = \begin{cases} 1 & N \text{ even} \\ -1 & N \text{ odd} \end{cases} \quad (2-7)$$

It is interesting to examine the behavior of the phase and group delay of allpass filters. Fig. 2-3 shows the phase and group delay of the stable third-order allpass transfer function of equation (2-2). From Fig. 2-2, the phase is a non-positive continuous function of ω in the range $\omega \in [0, \pi]$.

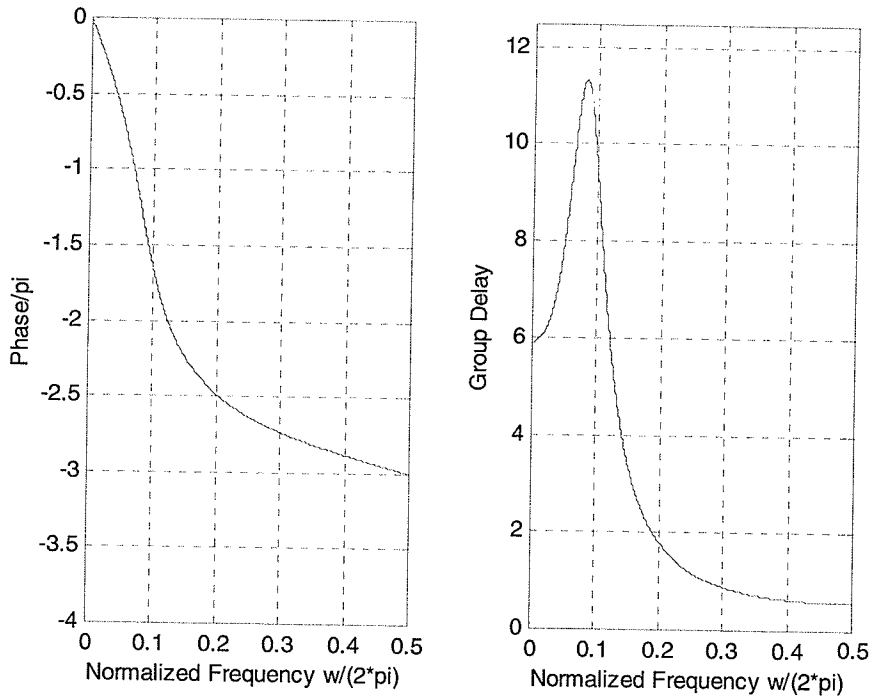


Fig. 2-2 The phase and group delay of the allpass function of Eq. (2-2)

From the above allpass function, we now state three very useful and important properties of a causal stable allpass function without proof [7, 49, 50].

First, setting $A(z) = Y(z)/U(z)$ reveals

$$|Y(e^{j\omega})|^2 = |U(e^{j\omega})|^2, \quad \text{for all } \omega \quad (2-8)$$

Integrating both sides from $\omega = -\pi$ to π and applying Parseval's relation, we obtain

$$\sum_{n=-\infty}^{\infty} |y(n)|^2 = \sum_{n=-\infty}^{\infty} |u(n)|^2. \quad (2-9)$$

It is convenient to interpret the two sides of equation (2-9) as the output energy and input energy of the digital filter, respectively. Thus, a causal stable allpass filter is a lossless structure.

Second, since a stable allpass function has its all poles inside the unit circle in the z -plane, all its zeros outside, and exhibits unit magnitude along the unit circle, it can be shown

$$|A(z)| < 1 \quad \text{for } |z| > 1, \quad (2-10a)$$

$$|A(z)| = 1 \quad \text{for } |z| = 1, \quad (2-10b)$$

$$|A(z)| > 1 \quad \text{for } |z| < 1. \quad (2-10c)$$

The last property of interest is with regard to the change in phase for a real stable allpass function over the frequency range $\omega \in [0, \pi]$. Let $\tau(\omega)$ denote the group delay function of the allpass filter $A(z)$, i.e.

$$\tau(\omega) = -\frac{d}{d\omega} [\arg A(e^{j\omega})], \quad (2-11)$$

Note that the unwrapped phase response of a stable allpass function $A(z)$ is a monotonically decreasing function of ω so that $\tau(\omega)$ is everywhere positive in range $\omega \in [0, \pi]$. Therefore, an N^{th} -order stable allpass transfer function satisfies the property

$$\int_0^\pi \tau(\omega) d\omega = N\pi \quad (2-12)$$

In other words, the change in the phase of an N^{th} -order stable allpass transfer function as ω goes from 0 to π is $N\pi$ radians.

The cascade connection of allpass filters is obviously an allpass filter. This is especially true if only first- and second-order sections are applied. The equation (2-1) can be expressed in factored form in terms of poles and zeros,

$$A(z) = z^{-N} \prod_{k=1}^{M_r} \frac{1 - \alpha_k z}{1 - \alpha_k z^{-1}} \prod_{k=1}^{M_c} \frac{(1 - \beta_k z)(1 - \beta_k^* z)}{(1 - \beta_k^* z^{-1})(1 - \beta_k z^{-1})} \quad (2-13)$$

where M_r is the number of real poles, and M_c is the number of complex conjugate pairs. For causal and stable systems we require that $-1 < \alpha_k < 1$ and $|\beta_k| < 1$.

2.2 Lattice Networks

A classical lattice two-port network is shown in Fig. 2-3 with lumped passive one-ports having impedances Z_1 and Z_2 that are positive real rational functions [1-3, 8].

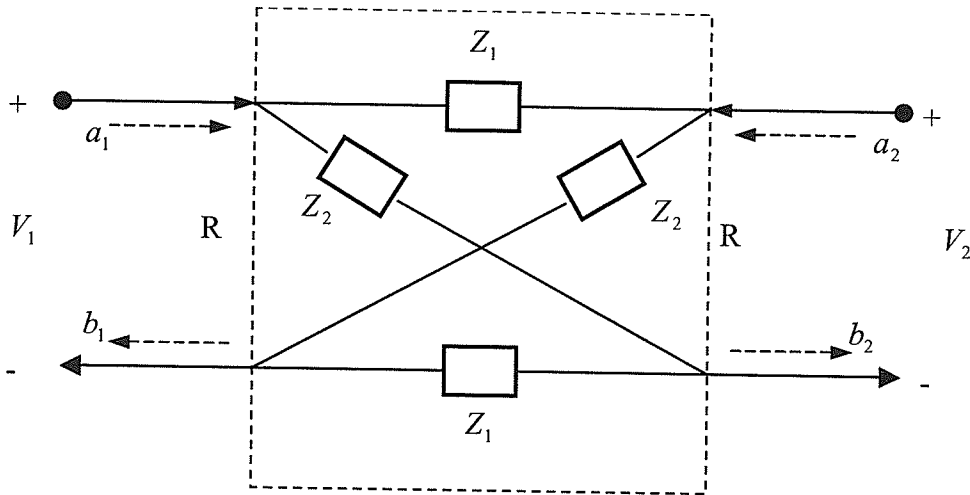


Fig. 2-3 A lattice network with two-port terminations

In some applications, it is useful to use two input signals simultaneously. The lattice structure can be described by the incident and reflected waves:

$$\begin{aligned} a_1 &= V_1 + RI_1 = V_{in1} & b_1 &= V_1 - RI_1 \\ a_2 &= V_2 + RI_2 = V_{in2} & b_2 &= V_2 - RI_2 \end{aligned}$$

where R is the port normalizing resistance. The lossless network is described by the scattering matrix, S :

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = S \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (2-14)$$

After the elimination of voltages and currents, we obtain

$$b_1 = \frac{[S_1(a_1 - a_2) + S_2(a_1 + a_2)]}{2} \quad (2-15)$$

$$b_2 = \frac{[S_1(a_2 - a_1) + S_2(a_1 + a_2)]}{2} \quad (2-16)$$

where

$$S_1 = \frac{Z_1 - R}{Z_1 + R} \quad (2-17)$$

and

$$S_2 = \frac{Z_2 - R}{Z_2 + R} \quad (2-18)$$

It is a symmetrical as well as a reciprocal two-port, and satisfies

$$S_{11} = S_{22} = (S_1 + S_2)/2 \quad (2-19)$$

$$S_{12} = S_{21} = (S_2 - S_1)/2 \quad (2-20)$$

Consequently, If Z_1 and Z_2 are reactances (lossless), the corresponding reflectances, S_1 and S_2 are allpass functions.

Fig. 2-3 illustrates the wave digital lattice filter described by equations (2-14) through (2-20). Note that the wave digital lattice filter consists of two allpass subfilters in parallel. These allpass subfilters have low sensitivity in the passband, but very high sensitivity in

the stopband. The normal transfer function of the filter, with a_1 as input and b_2 as output is

$$H = S_{21} = (S_2 - S_1)/2 \quad (2-21)$$

while the complementary transfer function is

$$H_c = S_{11} = (S_2 + S_1)/2 \quad (2-22)$$

In Fig. 2-4, S_1 and S_2 can be realized by using first- and second-order allpass sections as basic building blocks.

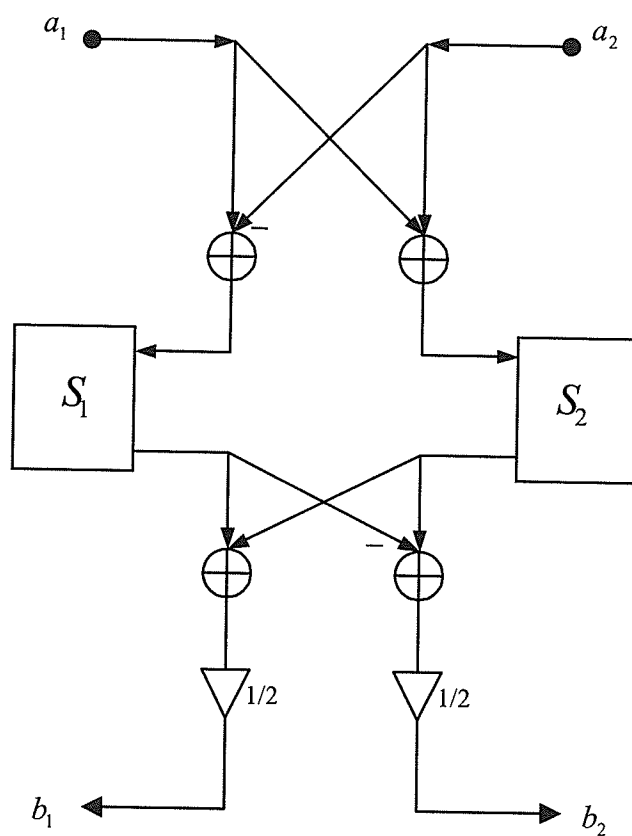


Fig. 2-4(a) wave-flow diagram of the lattice structure

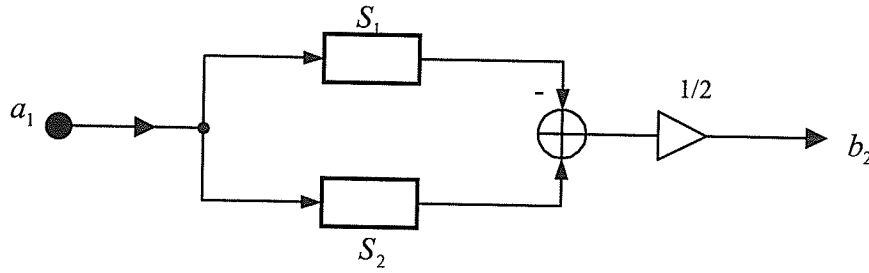


Figure 2-4(b) simplified wave-flow diagram

2.3 Wave Digital Allpass Structures Involving Two-port Adaptors

In this section, we consider first- and second-order allpass filter sections with a minimum number of multipliers and a minimum number of delays, shown as S_1 and S_2 in the wave digital lattice filter structure of Fig. 2-4. These allpass filters are composed of unit delays and two-port adaptors. The best structure can be chosen depending on the adaptor coefficient value γ .

2.3.1 First-order allpass section

A first-order allpass section is shown in Fig. 2-5. The multiplier coefficient γ is

$$\gamma = \frac{R_1 - R_2}{R_1 + R_2}, \quad (2-23)$$

where R_1 and R_2 are port resistances and $|\gamma| < 1$ for stability.

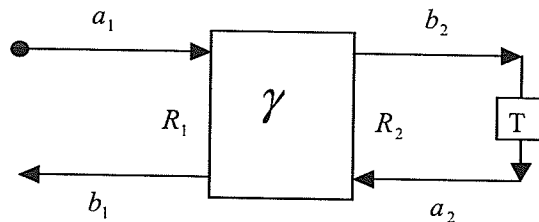


Figure 2-5 A first-order allpass section

The equations describing the two-port adaptor are

$$\begin{aligned} b_1 &= -\gamma a_1 + (1 + \gamma)a_2 \\ b_2 &= (1 - \gamma)a_1 + \gamma a_2 \end{aligned} \quad (2-24a,b)$$

Obviously from Fig. 2-5, we can obtain

$$a_2 = z^{-1}b_2 \quad (2-25)$$

Substituting (2-25) into (2-24b), we can obtain

$$b_2 = \frac{1 - \gamma}{1 - \gamma z^{-1}} a_1 \quad (2-26)$$

Substitute (2-25) and (2-26) into (2-24a), and then we can obtain the transfer function of a first-order allpass filter section

$$A_1(z) = \frac{b_1}{a_1} = \frac{-\gamma + z^{-1}}{1 - \gamma z^{-1}} \quad (2-27)$$

Fig. 2-6 shows a possible first-order wave digital allpass filter structure with only one multiplier and one delay.

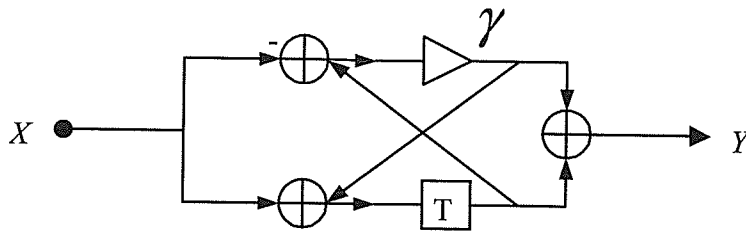


Fig. 2-6 a first-order wave digital allpass filter structure

2.3.2 Second-order allpass section

A second-order allpass section is shown in Fig. 2-7.

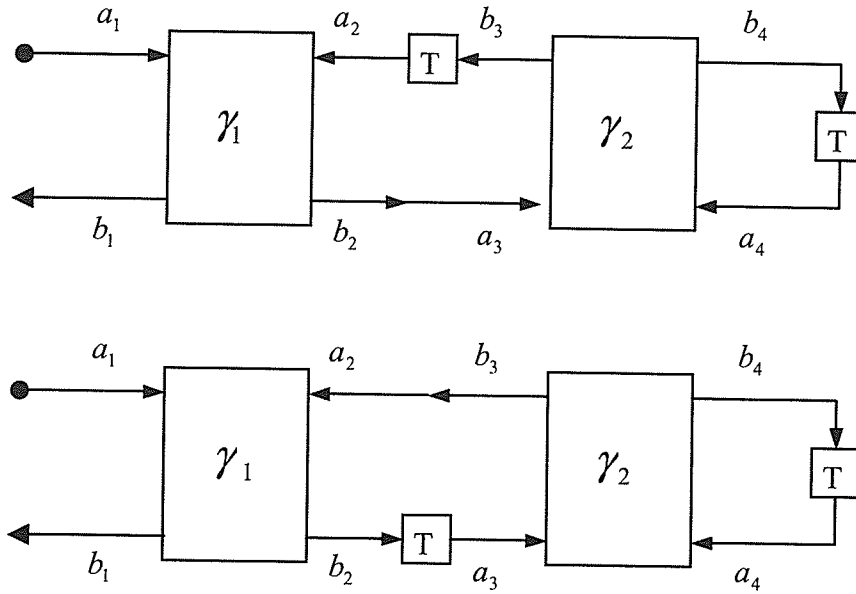


Fig. 2-7 Second-order allpass sections

From Fig. 2-7, we have the equations below describing two two-port adaptors:

$$\begin{aligned} b_1 &= -\gamma_1 a_1 + (1 + \gamma_1) a_2 \\ b_2 &= (1 - \gamma_1) a_1 + \gamma_1 a_2 \\ b_3 &= -\gamma_2 a_3 + (1 + \gamma_2) a_4 \\ b_4 &= (1 - \gamma_2) a_3 + \gamma_2 a_4 \end{aligned} \quad (2-28a, b, c, d)$$

In addition, we have

$$a_2 = z^{-1} b_3 \quad (2-29)$$

$$a_3 = b_2 \quad (2-30)$$

$$a_4 = z^{-1} b_4 \quad (2-31)$$

Substituting (2-31) into (2-28d)

$$b_4 = \frac{1 - \gamma_2}{1 - \gamma_2 z^{-1}} a_3 \quad (2-32)$$

Substituting (2-32) and (2-31) into (2-28c)

$$\begin{aligned} b_3 &= (1 + \gamma_2)z^{-1} \left(\frac{1 - \gamma_2}{1 - \gamma_2 z^{-1}} \right) a_3 - \gamma_2 a_3 \\ &= \left(\frac{(1 - \gamma_2^2)z^{-1}}{1 - \gamma_2 z^{-1}} - \gamma_2 \right) a_3 \\ &= \frac{-\gamma_2 + z^{-1}}{1 - \gamma_2 z^{-1}} a_3 \end{aligned} \quad (2-33)$$

Substituting (2-30) into (2-33)

$$b_3 = \frac{z^{-1} - \gamma_2}{1 - \gamma_2 z^{-1}} b_2 \quad (2-34)$$

Substituting (2-34) into (2-29)

$$a_2 = z^{-1} \left(\frac{z^{-1} - \gamma_2}{1 - \gamma_2 z^{-1}} \right) b_2 \quad (2-35)$$

Substituting (2-35) into (2-28b)

$$a_2 = \frac{(1 - \gamma_1)z^{-1} + \gamma_1 \gamma_2 - \gamma_2}{z + \gamma_1 \gamma_2 - \gamma_2 - \gamma_1 z^{-1}} a_1 \quad (2-36)$$

Substituting (2-36) into (2-28a), we obtain the transfer function of a second order allpass filter section:

$$A_2(z) = \frac{-\gamma_1 + \gamma_2(\gamma_1 - 1)z^{-1} + z^{-2}}{1 + \gamma_2(\gamma_1 - 1)z^{-1} - \gamma_1 z^{-2}} \quad (2-37)$$

Fig. 2-8 shows a possible second-order wave digital allpass filter structure with only two multipliers and delays.

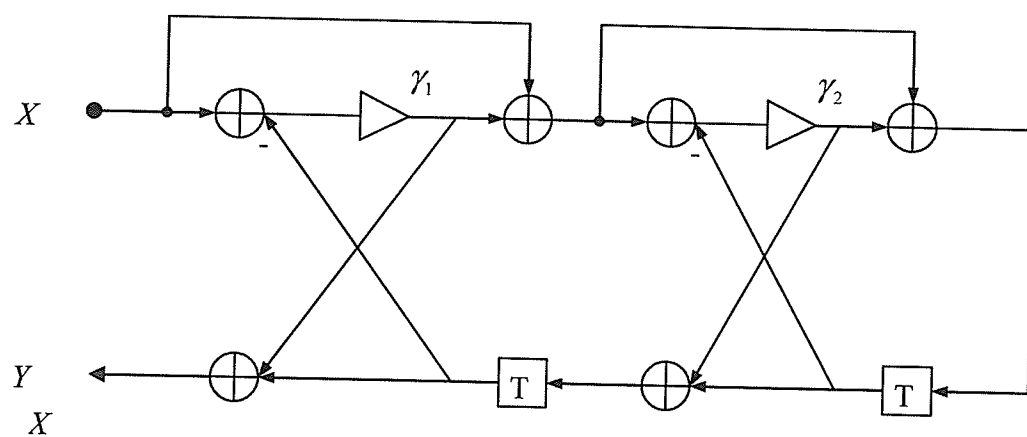


Fig. 2-8 a second-order wave digital allpass filter structure

Chapter 3

Realization of Wave Digital Lattice Filters as the Sum of Wave Digital Allpass filters

Recursive (IIR) Digital Filters can be realized as the sum of two allpass functions [5-8]. This filter structure can be viewed as a variant of the celebrated wave digital lattice filters [1-3] and constitutes a low-sensitivity structurally passive implementation.

Digital filters realized as the sum of two allpass functions possess a number of attractive features such as a minimum number of multipliers is required, low sensitivity with respect to coefficient quantization, simple signal scaling, low round-off noise and robustness with respect to limit cycles [5-8, 28-30].

3.1 Digital Filters Made as the Sum of Two Allpass Functions

Equation (2-1) can be written

$$A(z) = \frac{z^{-N} f(z^{-1})}{f(z)} = \frac{m(z) - n(z)}{m(z) + n(z)} \quad (3-1)$$

where $m(z)$ and $n(z)$ are the *Mirror Image (MI)* and *Anti-Mirror Image (AMI)* parts of $f(z)$, respectively, given by

$$m(z) = \frac{f(z) + z^{-N} f(z^{-1})}{2} \quad (3-2)$$

$$n(z) = \frac{f(z) - z^{-N} f(z^{-1})}{2} \quad (3-3)$$

The *MI* and *AMI* polynomials $m(z)$ and $n(z)$ have even and odd symmetry given by

$$m(z) = z^{-N} m(z^{-1}) \quad (3-4)$$

$$n(z) = -z^{-N} n(z^{-1}) \quad (3-5)$$

or equivalently

$$m_k = m_{N-k} \quad (3-6)$$

$$n_k = -n_{N-k} \quad (3-7)$$

where m_k and n_k are the coefficients of the polynomials $m(z)$ and $n(z)$, respectively.

Let $A_1(z)$ and $A_2(z)$ be two general allpass filters given by

$$A_1(z) = \frac{z^{-N_1} f_1(z^{-1})}{f_1(z)} = \frac{m_1(z) - n_1(z)}{m_1(z) + n_1(z)} \quad (3-8)$$

$$A_2(z) = \frac{z^{-N_2} f_2(z^{-1})}{f_2(z)} = \frac{m_2(z) - n_2(z)}{m_2(z) + n_2(z)} \quad (3-9)$$

where N_1 and N_2 are the order of the polynomials $f_1(z)$ and $f_2(z)$, respectively.

Consider the two digital filters $H_1(z)$ and $H_2(z)$ defined as the sum and difference of the two allpass functions in (3-8) and (3-9):

$$H_1(z) = \frac{A_1(z) + A_2(z)}{2} = \frac{\frac{m_1 - n_1}{m_1 + n_1} + \frac{m_2 - n_2}{m_2 + n_2}}{2} = \frac{m_1 m_2 - n_1 n_2}{(m_1 + n_1)(m_2 + n_2)} \quad (3-10)$$

$$H_2(z) = \frac{A_1(z) - A_2(z)}{2} = \frac{\frac{m_1 - n_1}{m_1 + n_1} - \frac{m_2 - n_2}{m_2 + n_2}}{2} = \frac{m_1 n_2 - n_1 m_2}{(m_1 + n_1)(m_2 + n_2)} \quad (3-11)$$

where m_1 , n_1 , m_2 and n_2 denote the MI and AMI parts $m_1(z)$, $n_1(z)$, $m_2(z)$ and $n_2(z)$ of the two polynomials $f_1(z)$ and $f_2(z)$, respectively. The order of the two filters $H_1(z)$ and $H_2(z)$ is $N = N_1 + N_2$.

From equation (2-4), we obtain

$$H_1(z)H_1(z^{-1}) + H_2(z)H_2(z^{-1}) = 1 \quad (3-12)$$

or, on the unit circle $z = e^{j\omega}$

$$|H_1(e^{j\omega})|^2 + |H_2(e^{j\omega})|^2 = 1. \quad (3-13)$$

Thus, the two filters $H_1(z)$ and $H_2(z)$ are said to form a power complimentary pair. If one filter has zero response at a certain frequency the other has unity response, and if one filter is a lowpass filter, the other is a highpass filter, etc. Since the complementary filter pair can be implemented with the cost of a single filter, they are very efficient in a tree-structured filter bank.

The characteristic function $K_1(z)$ related to $H_1(z)$ is defined by

$$K_1(z) = \frac{H_2(z)}{H_1(z)} = \frac{m_1n_2 - m_2n_1}{m_1m_2 - n_1n_2} \quad (3-14)$$

From (3-4) and (3-5), it is concluded that $K_1(z)$ is the quotient of an *AMI* polynomial by an *MI* polynomial. This is appropriate for a lowpass filter. $K_1(z)$ can be constructed by placing its zeros and poles in conjugate pairs on the unit circle, zeros in the passband and poles in the stopband. The quadratic factors containing these zeros and poles all have the *MI* form $z^2 + \alpha z + 1$. In addition, there must be a factor $(z-1)$ in the numerator conferring zero loss at $z = 1$ and making the numerator become *AMI*. The denominator will need a factor of $(z+1)$ to provide infinite loss at $z = -1$ and to make the numerator and denominator of $K_1(z)$ have the same degree.

When the zeros, poles and multiplying constant of $K_1(z)$ have been chosen to give an acceptable loss via $10\log_{10}[1+|K_1(e^{j\omega})|^2]dB$ one forms the sum of its numerator and denominator polynomials giving

$$n_1m_2 - m_1n_2 + m_1m_2 - n_1n_2 = (m_1 + n_1)(m_2 - n_2) \quad (3-15)$$

The zeros of this polynomial are found numerically. Those lying inside the unit circle belong to the factor $(m_1 + n_1)$, while those lying outside belong to $(m_2 - n_2)$. The two allpass functions are then defined once these two sets of zeros are known.

Next we will show that it is sufficient to realize the characteristic function when describing digital filters implemented as the sum or difference of two allpass functions. If $H_1(z)$ is the filter we are interested in, then equations (3-13) and (3-14) give the relation between $H_1(z)$ and $K_1(z)$ on the unit circle that is given below in terms of the filter loss

$$10\log \frac{1}{|H_1(e^{j\omega})|^2} = 10\log(1 + |K_1(e^{j\omega})|^2) \quad (3-16)$$

So far we have studied the properties of a filter corresponding to a given allpass realization. Now, we will consider the opposite problem to find an allpass realization when the overall filter is given.

Suppose an N^{th} -order recursive digital filter is given as

$$H_1(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1z^{-1} + \dots + p_Nz^{-N}}{1 + d_1z^{-1} + \dots + d_Nz^{-N}} \quad (3-17)$$

where the coefficients p_i and d_i are real. The zeros of $D(z)$ are strictly inside the unit circle and $P(z)$ is a *MI* polynomial, that is $P(z) = z^{-N}P(z^{-1})$ and $p_k = p_{N-k}$. Suppose further that there exists an N odd power complementary filter

$$H_2(z) = \frac{Q(z)}{D(z)} = \frac{q_0 + q_1 z^{-1} + \dots + q_N z^{-N}}{1 + d_1 z^{-1} + \dots + d_N z^{-N}} \quad (3-18)$$

where the coefficients q_i and d_i are real. $Q(z)$ is *AMI* polynomial, $Q(z) = -z^{-N}Q(z^{-1})$ and $q_k = -q_{N-k}$.

Now $H_1(z)$ and $H_2(z)$ above lead to the following relations:

$$P(z)P(z^{-1}) + Q(z)Q(z^{-1}) = D(z)D(z^{-1}) \quad (3-19)$$

$$P^2(z) - Q^2(z) = z^{-N}D(z)D(z^{-1}) \quad (3-20)$$

$$(P(z) + Q(z))(P(z) - Q(z)) = z^{-N}D(z)D(z^{-1}) \quad (3-21)$$

Add the two polynomials $P(z)$ and $Q(z)$ together and let z_1, \dots, z_r be the zeros of $P(z) + Q(z)$ inside the unit circle and z_{r+1}, \dots, z_N be the zeros of $P(z) + Q(z)$ outside.

Identify the zeros z_1, \dots, z_r inside the unit circle with the factor $m_2 + n_2$ and the zeros z_{r+1}, \dots, z_N outside the unit circle with the factor $m_1 - n_1$. This procedure uniquely defines the *MI* and *AMI* polynomials m_1 , n_1 , m_2 and n_2 .

From (3-4) and (3-5)

$$m(z^{-1}) - n(z^{-1}) = z^N(m(z) + n(z)) = z^N f(z) \quad (3-22)$$

so that the zeros z_k of $m + n$ correspond to zeros z_k^{-1} of $m - n$.

Then write

$$P(z) + Q(z) = (m_2 + n_2)(m_1 - n_1) = m_1 m_2 - n_1 n_2 + m_1 n_2 - n_1 m_2 \quad (3-23)$$

So, we can identify the corresponding *MI* and *AMI* polynomials as

$$P(z) = m_1 m_2 - n_1 n_2 \quad (3-24)$$

$$Q(z) = m_1 n_2 - n_1 m_2 \quad (3-25)$$

The polynomial $P(z) - Q(z)$ is given by

$$P(z) - Q(z) = (m_2 - n_2)(m_1 + n_1) = m_1 m_2 - n_1 n_2 - m_1 n_2 + n_1 m_2 \quad (3-26)$$

The factor $(P(z) + Q(z))(P(z) - Q(z))$ can thus be written as

$$(P(z) + Q(z))(P(z) - Q(z)) = (m_2 + n_2)(m_1 - n_1)(m_2 - n_2)(m_1 + n_1) \quad (3-27)$$

Therefore, we may identify $D(z)$ in (3-21) as

$$D(z) = (m_1 + n_1)(m_2 + n_2) \quad (3-28)$$

and

$$z^{-N} D(z^{-1}) = (m_1 - n_1)(m_2 - n_2) \quad (3-29)$$

Thus, according to (3-24), (3-25) and (3-28) it is concluded that the filters $H_1(z)$ and $H_2(z)$ given by (3-17) and (3-18) can be realized in the allpass form given by (3-10) and (3-11).

It now should be clear that the theorem given by A. N. Willson [8] can be summarized into the following:

Theorem: *The necessary and sufficient condition for a transfer function $H_1(z)$ ($H_2(z)$) to be realizable as the sum (or difference) of two allpass functions is that there exists a power complementary filter $H_2(z)$ ($H_1(z)$) such that $H_1(z)H_1(z^{-1}) + H_2(z)H_2(z^{-1}) = 1$ and that the characteristic function associated with the transfer function be a rational function formed as the quotient of an AMI (MI) polynomial by a MI (AMI) polynomial, both of the same degree.*

Certainly, not all digital filters given by (3-17) and (3-18) can be implemented as the sum (or difference) of two allpass functions. However, as examples, the classical odd-order

lowpass Butterworth, Chebyshev, Inverse Chebyshev, and Elliptic digital filters always satisfy the theorem above and can be implemented as in (3-10) and (3-11) [6, 8].

3.2 Wave Digital Lattice Filters

A wave digital lattice filter (WDLF) is a parallel two-branch structure where each branch consists of wave digital allpass filters [1-3]. As an example, in Fig. 3-1, an 11-order wave digital lattice filter is shown by using the cascade synthesis of elementary sections, S_1 and S_2 , which leads to the corresponding block diagram for the overall filter. It is interesting to note that digital filters of odd degree, obtained from the classical analog (Butterworth, Chebyshev, Inverse Chebyshev and Elliptic) prototypes, have explicit formulas given in [3].

In both lattice branches of the WDLF S_1 and S_2 are the reflectance of a reactance, i.e. allpass functions. Consequently, they may be written in the following form:

$$S_1 = \frac{-g_{1o}(z)}{g_1(z)} \quad (3-30)$$

and

$$S_2 = \frac{g_{2o}(z)}{g_2(z)} \quad (3-31)$$

where $g_{1o}(z) = z^{-N_1} g_1(z^{-1})$ and $g_{2o}(z) = z^{-N_2} g_2(z^{-1})$, respectively.

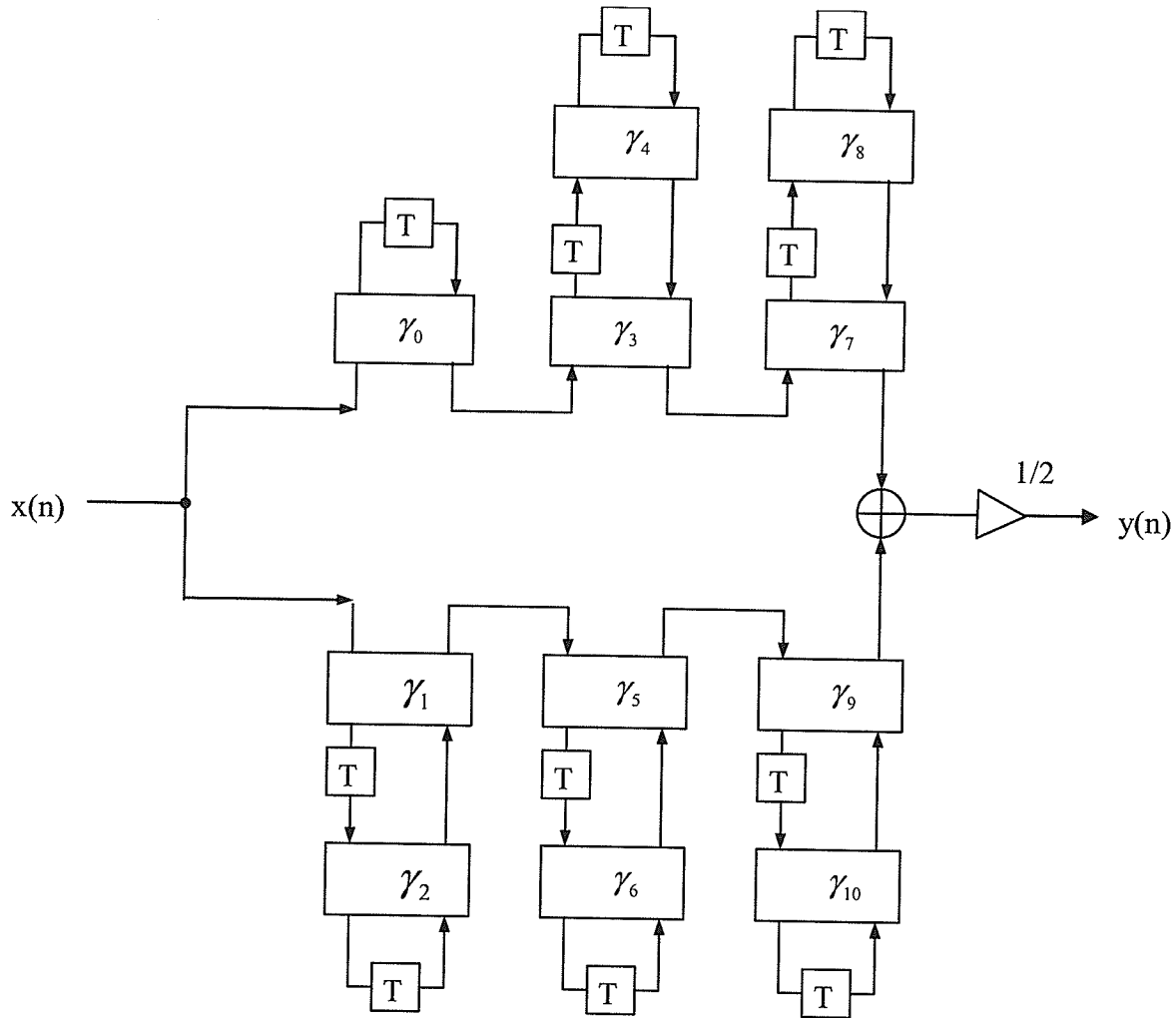


Fig 3-1 Block diagram of an 11-order lattice WDF using wave digital allpass filters

It is well known that the transfer functions of WDLF are given by

$$S_{11} = S_{22} = \frac{S_1 + S_2}{2} = \frac{h(z)}{g(z)} \quad (3-32)$$

$$S_{12} = S_{21} = \frac{S_2 - S_1}{2} = \frac{f(z)}{g(z)} \quad (3-33)$$

where f , g and h are the canonic polynomials[1-3].

From (3-30) to (3-33), we can see that

$$g(z) = g_1(z) \cdot g_2(z) \quad (3-34)$$

From (3-30) to (3-34), it is clear that

$$f(z) = \frac{1}{2} \{g_{1o}(z)g_2(z) + g_1(z)g_{2o}(z)\} \quad (3-35)$$

and

$$h(z) = \frac{1}{2} \{-g_{1o}(z)g_2(z) + g_1(z)g_{2o}(z)\} \quad (3-36)$$

Further, from (3-19)-(3-21) and (3-35)-(3-36), we obtain

$$f(z)f_o(z) + h(z)h_o(z) = g(z)g_o(z) \quad (3-37)$$

and

$$\begin{aligned} g(z)g_o(z) &= f^2(z) - h^2(z) \\ &= (f(z) - h(z))(f(z) + h(z)) \end{aligned} \quad (3-38)$$

where $f_o(z) = z^{-N}f(z^{-1})$, $h_o(z) = -z^{-N}h(z^{-1})$ and $g_o(z) = z^{-N}g(z^{-1})$.

Then, we have

$$g_1(z)g_{2o}(z) = f(z) - h(z) \quad (3-39)$$

or

$$g_{1o}(z)g_2(z) = f(z) + h(z) \quad (3-40)$$

The polynomial g_1 can be constructed from the zeros of $f - h$ inside the unit circle and the polynomial g_{2o} can be constructed from the zeros of $f - h$ outside the unit circle. It is the same for the polynomial g_2 which can be constructed from the zeros of $f + h$ inside the unit circle and the polynomial g_{1o} can be constructed from the zeros of $f + h$ outside the unit circle. Finally, the two allpass functions are obtained.

Chapter 4

Approximately Linear Phase Using Pole-Zero Placement

Chapter 3 shows the characteristic function can be constructed by placing its zeros and poles in conjugate pairs on the unit circle, zeros in the passband and poles in the stopband. In this chapter, we need to determine the pole and zero positions, and by an iteration process, let the poles and zeros gradually approach their optimum positions [9-13]. We review the principles of pole placement and modify it to design digital IIR filters which possess approximately linear phase.

4.1 Principles

A linear time-invariant filter can be described in two different ways. For example, in the z -domain, the transfer function for the filter can be realized as

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_i \alpha_i z^{-i}}{\sum_j \beta_j z^{-j}} = C_0 \frac{\prod_i (z - q_i)}{\prod_j (z - p_j)} \quad (4-1)$$

Here, the transfer function can be represented either with the coefficients of the polynomials (the α_i and β_j) or the zeros and poles of the polynomials (the q_i and p_j).

These representations will have different computational properties. For more accuracy, the pole-zero representation is implemented.

For lowpass filters, the passband and stopband attenuation with the frequency variance is illustrated in Fig. 4-1. Also shown is a typical approximating function, characterized by a series of arcs adjacent to the attenuation pole frequencies in the stopband and the reflection zero frequencies in the passband. In Fig. 4-1, A_p and A_s are the maximum attenuation in the passband and the minimum attenuation in the stopband, respectively. Also ω_c and ω_s are the passband edge frequency and the stopband edge frequency, respectively. We see that for every arc and reflection zero, there exists a minimum difference between the arc and the required attenuation. These minimum differences are influenced by the finite nonzero attenuation zeros and poles. If the positions of these zeros and poles vary, the minimum differences are changed. The task of the approximation is, for the given specifications, to place the zeros and poles, so that all the optimum differences are obtained, and thus find the transfer function.

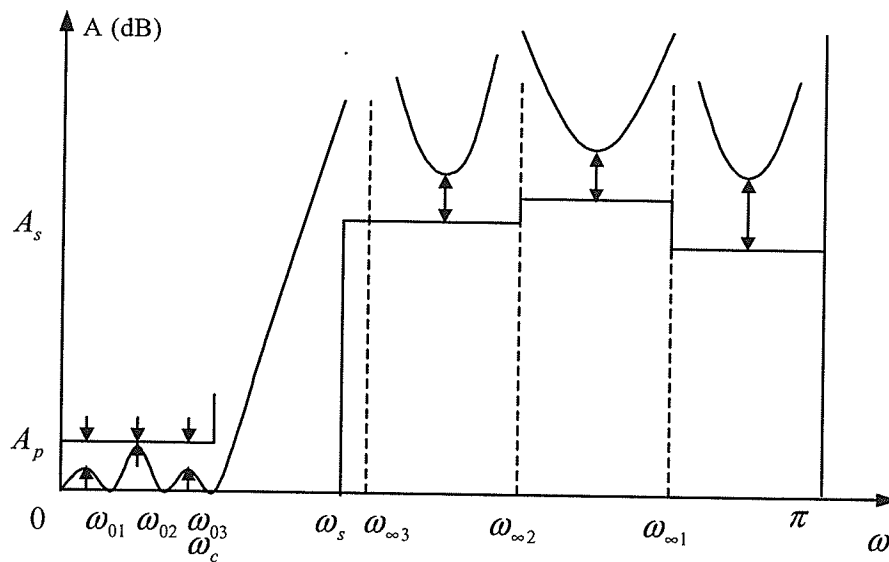


Fig. 4-1 The critical frequencies in the passband and stopband

The original pole placement algorithm was introduced by B. R. Smith and G. C. Temes [9].

4.2 Characteristic Function and Attenuation Calculation

In general, the attenuation $\alpha(z)$ is given in the z -domain by

$$\alpha(e^{j\omega}) = 10 \log_{10} \left[1 + |K(e^{j\omega})|^2 \right] \quad (4-2)$$

where $K(e^{j\omega})$ is the characteristic function,

$$K(e^{j\omega}) = C_0 \frac{\prod_{\mu=1}^m (z - e^{j\omega_{0\mu}})}{\prod_{\nu=1}^n (z - e^{j\omega_{\infty\nu}})} \quad (4-3)$$

where $z = e^{j\omega}$, C_0 is a real constant, m is the number of nonzero finite loss frequencies in the passband and n is the number of infinite loss frequencies in the stopband, and $m \leq n$. $\omega_{0\mu}$ ($\mu = 1, 2, \dots, m$) are the nonzero zero loss frequencies and $\omega_{\infty\nu}$ ($\nu = 1, 2, \dots, n$) are the nonzero infinite loss frequencies. Thus, $\alpha(e^{j\omega})$ is the loss characteristic as a function of ω .

In wave digital lattice filter design, we recall the transfer functions given by (3-32) and (3-33). The attenuation $a(e^{j\omega})$ of the WDLF is given by

$$\alpha(e^{j\omega}) = -10 \log_{10} |H(e^{j\omega})| \quad \text{dB} \quad (4-4)$$

where $H(e^{j\omega}) = S_{21}$.

The so-called characteristic function is defined by [3]

$$K(e^{j\omega}) = \frac{S_{11}}{S_{21}} = \frac{h}{f} \quad (4-5)$$

Thus, we can get from (4-2) and (4-5)

$$\alpha(e^{j\omega}) = 10 \log_{10} \left[1 + \left| \frac{h(e^{j\omega})}{f(e^{j\omega})} \right|^2 \right] \quad (4-6)$$

Here we see that the poles of the characteristic function are also the transmission zeros and the zeros of the characteristic function are the frequencies of maximum power transfer. Therefore, in the design of frequency selective networks the characteristic function poles are in the stopband and the characteristic function zeros are in the passband. In addition, $K(z)$ is the quotient of an *AMI* polynomial by an *MI* polynomial.

Then, h and f in polar form are defined as follows:

$$h(e^{j\omega}) = K_h (e^{j\omega} - 1) \prod_{i=1}^{N-1} (e^{j\omega} - r_{h_i} e^{j\omega_{h_i}}) \quad (4-7)$$

and

$$f(e^{j\omega}) = (e^{j\omega} + 1) \prod_{i=1}^{N-1} (e^{j\omega} - r_{f_i} e^{j\omega_{f_i}}) \quad (4-8)$$

where K_h is the constant factor.

4.3 The Modified Pole-Zero Placement and Its Application

The pole placement algorithm works with attenuation poles, i.e., transmission zeros, and iterates the position of the attenuation poles until an optimal filter is achieved. The optimality criterion is that the minimum distance between each arc of the attenuation and the attenuation constraint should be the same for all arcs.

4.3.1 The modified pole-zero placement

In this thesis, based on Unbehauen's method [11], we use the modified pole-zero placement algorithm to work on both zeros and poles of the characteristic function in the z -domain. To find these optimum zero and pole positions, we first need to determine the number of the loss zeros and poles. The initial zeros of the characteristic function are

placed in the passband and the initial poles of the characteristic function are placed in the stopband.

The attenuation is a function of the zeros and poles. If the zeros and poles are changed slightly, then the change in the attenuation function can be approximated as

$$\frac{\partial \alpha}{\partial x} = \frac{10}{\ln 10 \cdot (1 + |K(e^{j\omega})|^2)} \frac{\partial |K(e^{j\omega})|^2}{\partial x} \quad (4-9)$$

where x denotes a parameter.

Form equation (4-5) to (4-8), we obtain

$$|K(e^{j\omega})|^2 = \frac{|h(e^{j\omega})|^2}{|f(e^{j\omega})|^2} = K_h^2 \frac{(1 - \cos(\omega))}{(1 + \cos(\omega))} \frac{\prod_{i=1}^{N-1} (1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega - \omega_{h_i}))}{\prod_{i=1}^{N-1} (1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega - \omega_{f_i}))} \quad (4-10)$$

Taking the natural log of equation (4-10), we have

$$\begin{aligned} \ln |K(e^{j\omega})|^2 &= \ln K_h^2 + \ln \left(\frac{1 - \cos(\omega)}{1 + \cos(\omega)} \right) + \sum_{i=1}^{N-1} \ln (1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega - \omega_{h_i})) \\ &\quad - \sum_{i=1}^{N-1} \ln (1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega - \omega_{f_i})) \end{aligned} \quad (4-11)$$

Differentiating the left side of (4-11), we have

$$\frac{\partial \ln(|K(e^{j\omega})|^2)}{\partial x} = \frac{1}{|K(e^{j\omega})|^2} \frac{\partial |K(e^{j\omega})|^2}{\partial x} \quad (4-12)$$

Then we obtain

$$\frac{\partial |K(e^{j\omega})|^2}{\partial x} = \frac{|K(e^{j\omega})|^2 \partial \ln(|K(e^{j\omega})|^2)}{\partial x} \quad (4-13)$$

Substituting (4-13) into (4-9), we obtain

$$\frac{\partial \alpha}{\partial x} = \frac{10|K(e^{j\omega})|^2}{\ln 10 \cdot (1 + |K(e^{j\omega})|^2)} \frac{\partial \ln(|K(e^{j\omega})|^2)}{\partial x} \quad (4-14)$$

where the equations $\frac{\partial \ln(|K|^2)}{\partial x}$, for various choices of x , are

$$\frac{\partial \ln|K|^2}{\partial \omega_{h_i}} = -2r_{h_i} \left[\frac{\sin(\omega - \omega_{h_i})}{(1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega - \omega_{h_i}))} - \frac{\sin(\omega + \omega_{h_i})}{(1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega + \omega_{h_i}))} \right] \quad (4-15)$$

$$\frac{\partial \ln|K|^2}{\partial r_{h_i}} = 2 \left[\frac{r_{h_i} - \cos(\omega - \omega_{h_i})}{(1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega - \omega_{h_i}))} + \frac{r_{h_i} - \cos(\omega + \omega_{h_i})}{(1 + r_{h_i}^2 - 2r_{h_i} \cos(\omega + \omega_{h_i}))} \right] \quad (4-16)$$

$$\frac{\partial \ln|K|^2}{\partial \omega_{f_i}} = 2r_{f_i} \left[\frac{\sin(\omega - \omega_{f_i})}{(1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega - \omega_{f_i}))} - \frac{\sin(\omega + \omega_{f_i})}{(1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega + \omega_{f_i}))} \right] \quad (4-17)$$

$$\frac{\partial \ln|K|^2}{\partial r_{f_i}} = -2 \left[\frac{r_{f_i} - \cos(\omega - \omega_{f_i})}{(1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega - \omega_{f_i}))} + \frac{r_{f_i} - \cos(\omega + \omega_{f_i})}{(1 + r_{f_i}^2 - 2r_{f_i} \cos(\omega + \omega_{f_i}))} \right] \quad (4-18)$$

$$\frac{\partial \ln|K|^2}{\partial |K_h|} = \frac{2}{|K_h|} \quad (4-19)$$

where ω_{h_i} and ω_{f_i} are the angle of the zeros of h and f , respectively. Zeros and poles are considered in complex conjugate pairs in the above expressions, the factor $(z-1)$ in the numerator and the factor $(z+1)$ in denominator are excluded, and we take $\omega_{h_i}, \omega_{f_i} > 0$. r_{h_i} and r_{f_i} are the radii of the zeros of h and f , respectively. Since the characteristic function $K_1(z)$ can be constructed by placing its zeros and poles in conjugate pairs on the unit circle, r_{h_i} and r_{f_i} can be made equal to 1. The initial K_h is chosen to satisfy the specification at $\omega = \omega_c$.

Next, we are required to find the arc minimum differences. Observing Fig. 4-1, for each arc, the arc minimum difference is in either the arc minimum or the step corner. To find the optimum zero and pole positions, the zero and pole frequencies must be modified, so that the minimum difference $\Delta\alpha_j$ is the same for each arc.

If there is more than one local maximum between two adjacent attenuation zeros or between an adjacent attenuation zero and a passband boundary, then only the one with the largest value is considered. The corresponding situation occurs when more than one local minimum is in the stopband between two adjacent attenuation poles or between an adjacent attenuation pole and a stopband boundary. It should be noted that the number of extrema available in each sub-region is always one greater than the number of approximation parameters in that interval. Thus there are a total of $m+n+k$ extrema to be considered if one has k sub-regions. For example, in the lowpass filter, there are two sub-regions. However, the attenuation α of (4-2) contains only $m+n+1$ parameters. For each sub-region, an additional free parameter M_k ($k=1,2,\dots,K$) is introduced.

In this manner, a total of $m+n+k$ linear equations are obtained for the $m+n+k+1$ variables ΔK_h , $\Delta\omega_{h_i}$, $\Delta\omega_{f_i}$ and M_k . One of the variables M_k is assigned a fixed value. During the iteration process, the sequence of the attenuation zeros and poles in the passband and stopband are confined to their corresponding regions. The change in the attenuation at n^{th} iteration can be approximated as

$$\begin{bmatrix} \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_{m+1} \\ \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial\alpha_1}{\partial K_h} & \frac{\partial\alpha_1}{\partial\omega_{h_1}} & \cdots & \frac{\partial\alpha_1}{\partial\omega_{h_m}} & \frac{\partial\alpha_1}{\partial\omega_{f_1}} & \cdots & \frac{\partial\alpha_1}{\partial\omega_{f_n}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial\alpha_{m+n+2}}{\partial K_h} & \frac{\partial\alpha_{m+n+2}}{\partial\omega_{h_1}} & \cdots & \frac{\partial\alpha_{m+n+2}}{\partial\omega_{h_m}} & \frac{\partial\alpha_{m+n+2}}{\partial\omega_{f_1}} & \cdots & \frac{\partial\alpha_{m+n+2}}{\partial\omega_{f_n}} & 1 \end{bmatrix} \begin{bmatrix} \Delta K_h \\ \Delta\omega_{h_1} \\ \vdots \\ \Delta\omega_{h_m} \\ \Delta\omega_{f_1} \\ \vdots \\ \Delta\omega_{f_n} \\ M_2 \end{bmatrix} \quad (4-20)$$

where $\Delta\alpha_j = \alpha(\omega_j) - \alpha_{spec}$. The left side is obtained through linearization of the attenuation at the current approximate value of the approximation parameter. The equation (4-20) contains the requirement that the approximation of the attenuation function at the point ω_j agrees with the specification A_{spec} , which are modified from (4-15) and (4-17) as follows:

$$\frac{\partial \alpha(\omega_j)}{\partial \omega_{h_i}} = \left(\frac{-\sin(\omega_j - \omega_{h_i})}{1 - \cos(\omega_j - \omega_{h_i})} + \frac{\sin(\omega_j + \omega_{h_i})}{1 - \cos(\omega_j + \omega_{h_i})} \right) \quad (4-21)$$

and

$$\frac{\partial \alpha(\omega_j)}{\partial \omega_{f_i}} = \left(\frac{\sin(\omega_j - \omega_{f_i})}{1 - \cos(\omega_j - \omega_{f_i})} - \frac{\sin(\omega_j + \omega_{f_i})}{1 - \cos(\omega_j + \omega_{f_i})} \right) \quad (4-22)$$

Then we can obtain the new angle of the zeros of h and f and the new K_h

$$K_h^{n+1} = K_h^{(n)} + \gamma \Delta K_h \quad (4-23)$$

$$\omega_{h_i}^{(n+1)} = \omega_{h_i}^{(n)} + \gamma \Delta\omega_{h_i} \quad (4-24)$$

$$\omega_{f_i}^{(n+1)} = \omega_{f_i}^{(n)} + \gamma \Delta\omega_{f_i} \quad (4-25)$$

where γ is a scale factor and $0 < \gamma < 1$. If the new zeros and poles are out of their related regions, the new zeros and poles will be modified. The iteration stops after the criterion is satisfied.

As the equations are nonlinear, the optimum zero and pole locations must be determined iteratively. During the iteration process, it is easy to ensure that zeros and poles do not leave the specified intervals. The iteration stops when the arc minimum difference is satisfied by

$$0 < \alpha - \alpha_{spec} < 0.0001 \quad (4-26)$$

Thus, the given attenuation in the passband and the stopband designs can be carried out in the following steps:

1. Choose the initial points ω_i in the passband for h and the stopband for f , and calculate the initial K_h .
2. Calculate the attenuation using (4-6), and scan the attenuation to find the local maximum point for the passband and the local minimum for the stopband.
3. Use (4-23) - (4-25) to obtain new zeros of f and h and make sure that they satisfy the bound.
4. If the equation (4-26) is not satisfied, then go back.

During the iteration process, it is easy to ensure that zeros and poles do not leave the specified intervals.

4.3.2 Approximately linear phase approach

To obtain approximately linear phase, we can design the group delay as flat as possible.

In practice, we are only interested in linear phase in the passband. As we know, in the design of IIR filters, the poles of the transfer function must be in the unit circle to ensure

stability. Most zeros are in practice positioned on the unit circle. But as mentioned in the previous chapter, in allpass filters, zeros are placed outside the unit circle. Thus, from characteristic function (4-6), the zeros in the passband may be initialized to any position about the unit circle such as one quad of zeros about the unit circle and the rest of zeros on the unit circle.

The main idea is to move the zeros and poles to an optimum position where the group delay is as flat as possible.

1. Initialization:

Choose initial points ω_{h_i} of h on the unit circle with one quad of points and initial points ω_{f_i} of f on the unit circle.

Use pole-zero placement to obtain the required attenuation.

2. Move one quad of zeros of h .

Check the group delay in the passband and the attenuation.

3. If the group delay satisfies $|\tau^{(k)} - \tau_{avg}| < \epsilon$, then approximately linear phase is obtained. If the attenuation is not satisfied, then move the remaining zeros of h and zeros of f until the attenuation specification is satisfied. Go back to step 2.

This approach takes considerable time until the group delay and attenuation specifications are satisfied.

Chapter 5

Group Delay Equalization Using a Cascaded Wave Digital Allpass Filter

The design of an allpass delay equalizer for IIR filters is considered. As we know, for example, an IIR lowpass filter does not have linear phase (or flat group delay). Rather, it typically has a peak in the transmission band. To produce a complete lowpass filter with an approximately linear phase across the entire passband, an allpass filter must be cascaded with the IIR lowpass filter. This allpass filter will increase the group delay at lower frequencies, but does not affect the magnitude response.

5.1 Group Delay Equalization

An important application of allpass filter design is the case of group delay equalization as shown in Fig. 5-1. We consider the equalization of an M^{th} -order IIR filter of the form

$$H_f(z) = \frac{N(z)}{D(z)} = \frac{\beta_0 + \beta_1 z^{-1} + \dots + \beta_M z^{-M}}{1 + \alpha_1 z^{-1} + \dots + \alpha_M z^{-M}} \quad (5-1)$$

with the phase response

$$\beta_f(\omega) = \arg\{H_f(e^{j\omega})\} \quad (5-2)$$

and the group delay

$$\tau_f = -\frac{d\beta_f(\omega)}{d\omega} \quad (5-3)$$

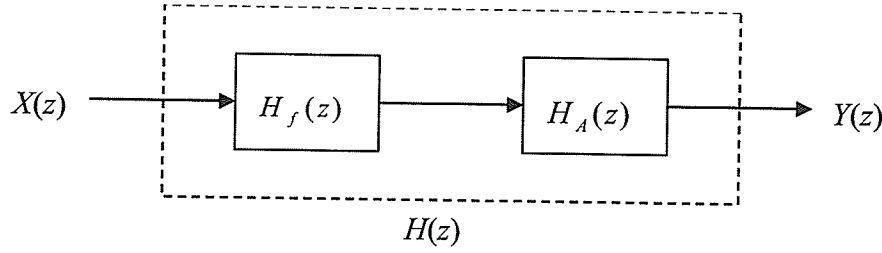


Fig. 5-1 group delay equalization

The equalizer is to be implemented such that the group delay of the allpass filter compensates the group delay of the IIR filter to produce an approximately flat overall group delay within a given frequency band, i.e.

$$\tau_f + \tau_A \approx \tau_0 \quad \omega \in [\omega_L, \omega_U] \quad (5-4)$$

where τ_0 is the nominal value of the desired group delay.

Hence, the allpass filter should approximate the following phase response

$$\tau_A = \tau_0 - \tau_f \quad \omega \in [\omega_L, \omega_U] \quad (5-5)$$

Here, we define the target group delay as

$$\tau_T = y * \max(\tau_f) - \tau_f \quad \omega \in [\omega_L, \omega_U] \quad (5-6)$$

where y is a scale variable. Then, an example of the target group delay of the allpass filter design is shown in Fig. 5-2.

Thus, the design problem is to minimize the group delay error

$$e = \min \|\tau_T - \tau_A\| \quad (5-7)$$

The group delay distortion increases rapidly near the edge of the passband, consequently the band of the group delay equalization has to be controlled precisely, because the performance of the equalization with given complexity depends highly on the group delay equalization band.

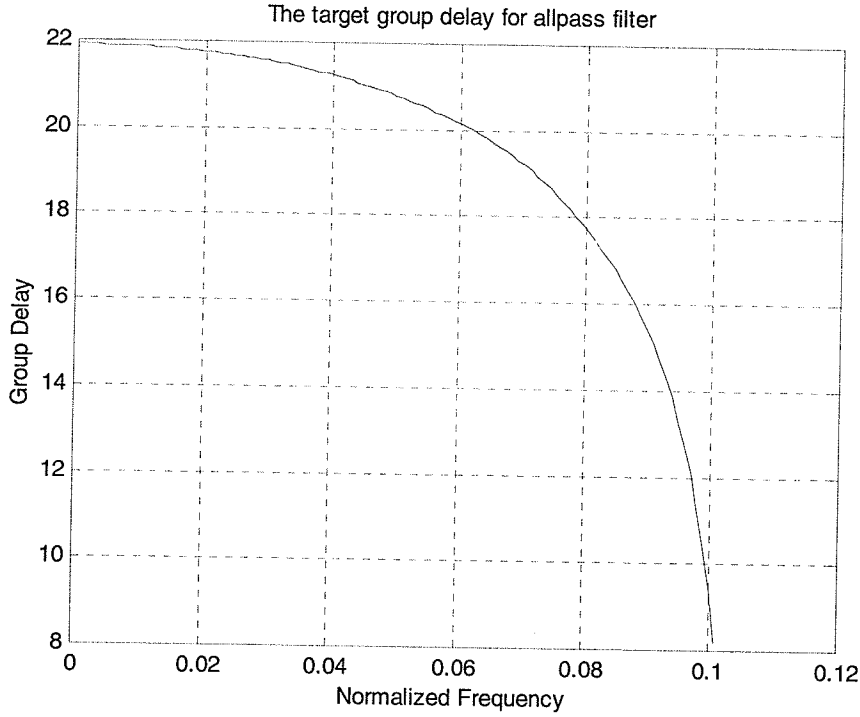


Fig. 5-2 The target group delay of an allpass filter design ($y=1$)

The second interesting application is the parallel allpass structure. The overall transfer functions are

$$H(z) = \frac{1}{2} (H_{A_1}(z) \pm H_{A_2}(z)) \quad (5-8)$$

The difference of the phase of allpass transfer functions $H_{A_1}(e^{j\omega})$ and $H_{A_2}(e^{j\omega})$ approximates a certain piecewise linear phase [28-33]. It will be considered in the next chapter.

5.2 Group Delay Equalization Using Nonlinear Least Squares Method

The development of the allpass compensation filter proceeds as follows:

1. Construct a target curve for the desired group delay response.
2. Choose an initial order for the allpass filter.

3. Optimize the allpass filter parameters.

The required order of the allpass filter is unknown. There is no specific order that must be used. It is common to use an order equal to that of the filter being compensated.

5.2.1 Defining the group delay objective

Before we can optimize the allpass filter, we will need to create a target curve of our desired group delay response. The optimizer will use this target curve as the objective for curve optimization. In practice, we wish to alter the curve in the passband to be an approximately flat line. The nature of the curve in the stopband region is unimportant. This area will not be part of the optimization.

The allpass transfer function defined by equation (2-1) can be rewritten in polar coordinates

$$H_A(z) = K \prod_{i=1}^N \frac{z - \frac{1}{r_i} e^{j\theta_i}}{z - r_i e^{j\theta_i}} \quad (5-9)$$

where $K = (-1)^n \prod_{i=1}^N r_i e^{j\theta_i}$, and $0 < r_i \leq 1$.

The equation for the phase response is

$$\beta_A(\omega) = -\sum_{i=1}^N \tan^{-1} \left(\frac{(1-r_i^2) \sin(\omega - \theta_i)}{(1+r_i^2) \cos(\omega - \theta_i) - 2r_i} \right) \quad (5-10)$$

If $K < 0$, π is added to $\beta_A(\omega)$.

The equation for the group delay is

$$\begin{aligned} \tau_A(\omega) &= -\frac{d\beta_A}{d\omega} \\ &= \sum_{i=1}^N \frac{1-r_i^2}{1+r_i^2 - 2r_i \cos(\omega - \theta_i)} \end{aligned} \quad (5-11)$$

The partial derivatives with respect to the parameters of equation (5-11) are

$$\frac{\partial \tau_A}{\partial r_i} = \frac{2((1+r_i^2)\cos(\omega-\theta_i)-2r_i)}{(1-2r_i\cos(\omega-\theta_i)+r_i^2)^2} + \frac{2((1+r_i^2)\cos(\omega+\theta_i)-2r_i)}{(1-2r_i\cos(\omega+\theta_i)+r_i^2)^2} \quad (5-12a)$$

$$\frac{\partial \tau_A}{\partial \theta_i} = \frac{2r_i(1-r_i^2)\sin(\omega-\theta_i)}{(1-2r_i\cos(\omega-\theta_i)+r_i^2)^2} - \frac{2r_i(1-r_i^2)\sin(\omega+\theta_i)}{(1-2r_i\cos(\omega+\theta_i)+r_i^2)^2} \quad (5-12b)$$

When $\theta_i = 0$ or $\theta_i = \pi$, (5-12) becomes

$$\frac{\partial \tau_A}{\partial r_i} = \frac{2((1+r_i^2)\cos(\omega-\theta_i)-2r_i)}{(1-2r_i\cos(\omega-\theta_i)+r_i^2)^2} \quad (5-13)$$

From equation (5-6), the target group delay in the passband then becomes

$$\tau_T = y \times \max(\tau_f) - \tau_{f_{passband}} \quad (5-14)$$

Then, the problem is to determine the minimum

$$E = \|\tau_T - \tau_A\|^2 \quad (5-15)$$

From (5-11), the problem is clearly a nonlinear data-fitting problem. Generally, the problem can be solved only by iteration.

5.2.2 Gauss-Newton method

We use radii r_i and angles θ_i as the polynomial parameters to represent the transfer function $H_A(e^{j\omega})$, as shown in (5-9). Define a vector x containing the design variables

$$x = [r_1, \dots, r_N, \theta_1, \dots, \theta_N]^T \quad (5-16)$$

In order to explicitly show the dependence of the group delay response $\tau_A(\omega)$ on the variables contained in x , we use the notation $\tau_A(\omega, x)$ whenever necessary. Likewise, we use the notation $E(\omega, x) = \tau_T(\omega) - \tau_A(\omega, x)$. In the Gauss-Newton method, the residuals of the least squares problem are linearized at every iteration step such that a sequence of

linear least squares problems is solved. In the group delay design problem, the residuals are $E(\omega_i, x)$, $i = 1, \dots, m$. At iteration step k , the current iteration is $x^{(k)}$. The first order Taylor series approximation of $\tau_A(\omega, x)$ about $x^{(k)}$ is given by

$$\tau_A(\omega, x) \approx \tau_A(\omega, x^{(k)}) + \nabla \tau_A(\omega, x^{(k)})(x - x^{(k)}) + h, \quad \|h\| = o(\|x - x^{(k)}\|) \quad (5-17)$$

where $\nabla \tau_A(x - x^{(k)})$ is the gradient vector of $\tau_A(\omega, x)$ evaluated at $x = x^{(k)}$. From equation (5-17), the linearization of $E(\omega_i, x)$ at iteration step k is

$$E(\omega, x) \approx \tau_T(\omega) - \tau_A(\omega, x^{(k)}) - \nabla \tau_A(\omega, x^{(k)})(x - x^{(k)}) \quad (5-18)$$

defining $s = x - x^{(k)}$ as a variable vector to be optimized. If $x^{(k)}$ is close to the minimum point of the nonlinear least squares problem, then the solution x of the linear least squares problem at iteration step k is

$$\min \|\tau_T(\omega) - \tau_A(\omega, x^{(k)}) - \nabla \tau_A(\omega, x^{(k)}) \cdot s\|^2 \quad (5-19)$$

Expanding the norm and ignoring the term that does not depend on s , equation (5-19) can be written as

$$\min (s^T \nabla \tau_A^{(k)T} \nabla \tau_A^{(k)} s - (\nabla \tau_A^{(k)} s)^T r^{(k)}) \quad (5-20)$$

where

$$r^{(k)} = \tau_T(\omega) - \tau_A(\omega, x^{(k)}) \quad (5-21)$$

and

$$\nabla \tau_A = \begin{bmatrix} \frac{\partial \tau_A(\omega_1)}{\partial r_1} & \dots & \frac{\partial \tau_A(\omega_1)}{\partial r_n} & \frac{\partial \tau_A(\omega_1)}{\partial \theta_1} & \dots & \frac{\partial \tau_A(\omega_1)}{\partial \theta_n} \\ \vdots & & & & & \vdots \\ \frac{\partial \tau_A(\omega_m)}{\partial r_1} & \dots & \frac{\partial \tau_A(\omega_m)}{\partial r_n} & \frac{\partial \tau_A(\omega_m)}{\partial \theta_1} & \dots & \frac{\partial \tau_A(\omega_m)}{\partial \theta_n} \end{bmatrix} \quad (5-22)$$

represents the Jacobian matrix.

Solving the minimization problem (5-20) is equivalent to computing the solution to the system of linear equations

$$\nabla \tau_A^{(k)} s = r^{(k)} \quad (5-23)$$

At every iteration step of the Gauss-Newton algorithm [54 and 55], the linear system (5-23) is solved for s , and the current update $s^{(k)}$ is set by s . The new iteration is computed by setting

$$x^{(k+1)} = x^{(k)} + \alpha s^{(k)} \quad \alpha > 0. \quad (5-24)$$

A line search algorithm can be used to optimize the step size α in every iteration step. However, the algorithm usually has a convergence even with a fixed step size. In this thesis, we use 2^{-i} as step size in every iteration.

Summarizing, the Gauss-Newton method works as follows:

1. Choose an initial guess $x^{(0)}$, set $k = 0$, $i = 0$. Compute $norm_0^{(k)} = \|\tau_T - \tau_A^{(k)}\|^2$.
2. Compute $r^{(k)}$ and $\nabla \tau_A^{(k)}$ from (5-21) and (5-22).
3. Solve $\nabla \tau_A^{(k)} s = r^{(k)}$ for s , and set $s^{(k)} = s$.
4. Set $x^{(k+1)} = x^{(k)} + 2^{-i} s^{(k)}$, if $norm_i^{(k)} > norm_0^{(k)}$, then $i = i + 1$.
5. Set $k = k + 1$, go to 2.

The iteration is stopped as soon as there is no significant change of $x^{(k)}$ from one iteration step to the next.

From equation (5-15), it is clear that the error is also a function of the scale factor y in quadratic form. To obtain the overall best group delay, we have to find the optimum scale factor where the error is a global minimum. We can construct an iterative algorithm for determining an optimal scale factor for minimum norm error. We can guess the initial

range $y \in [0.5, 2]$ and plot the error E versus scale factor y as in Fig. 5-3. Although there are several local minima, we can apply one dimensional search methods such as bisection, golden section search, Newton's method, or secant method to find the optimal value.

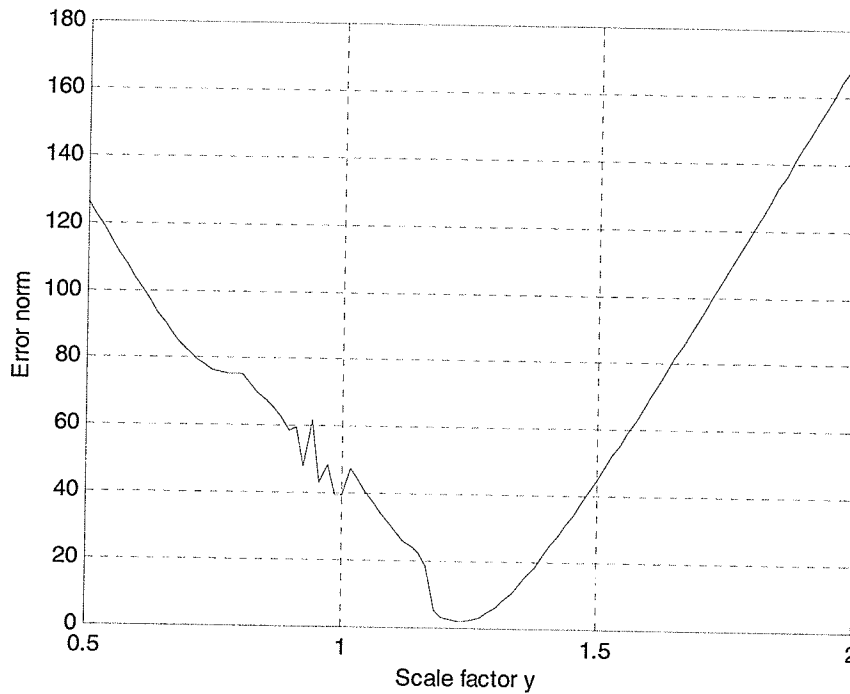


Figure 5-3 the error norm vs. the scale factor y

5.2.3 Design algorithm

The algorithm consists of two steps. The first step works in a way based on the *Gauss-Newton algorithm* to solve the nonlinear least-squares problem. The next uses the bisection algorithm to solve for the optimal scale factor with the minimum error norm.

The algorithm is thus performed in the following steps:

1. Initialization:

Set initial range for scale factor y : $y \in [0.5, 2]$. Choose the first, the mid, and the end scale factors to calculate τ_T , respectively.

Choose the first point of τ_T to calculate the initial coefficient vector a .

Select M points $\Omega = [\omega_1, \omega_2, \dots, \omega_M]$, $\omega_i \in [0, \omega_c]$ and calculate the Jacobian matrix $\nabla \tau_A$.

Compute the initial $a^{(0)}$, $\tau_A^{(0)}$ and $norm_0^{(0)} = \|\tau_T - \tau_A^{(0)}\|^2$

2. Solve equation (5-22) for the direction s .

Set $i = 0$, calculate $x^{(k+1)} = x^{(k)} + 2^{-i} s^{(k)}$ and $norm_i^{(k)} = \|\tau_T - \tau_A^{(k)}\|^2$

If $norm_i^{(k)} > norm_0^{(k)}$, set $i = i + 1$. Obtain $norm_i^{(k)}$ as small as possible for each step. Then set $k = k + 1$, obtain next $norm_i^{(k+1)}$.

3. Use the bisection method to find the optimal y with the minimum $norm$. If the optimal y cannot be found, then the range of y must be extended and the process goes back to step 1.

The method of finding each error with scale factor is iterative. The first and the last active set are selected, a solution is determined, and then, the mid, left and right set are selected.

This continues until the solution is optimal.

Chapter 6

Approximately Linear Phase Using a Sum of Wave Digital Allpass Filters

In Chapter 3, the wave digital lattice filter realized as the sum of two allpass filters is described. Like any other IIR filter, to meet given attenuation requirements but with the usual design principles a linear phase in the passband is not obtained. It will be shown that wave digital lattice filters can be designed to have an approximately linear phase in the passband.

6.1 The Phase Response of IIR Digital Filters Using the Sum of Two Allpass Subfilters

As we know, it is difficult for IIR (recursive) digital filter design to meet not only magnitude requirements, but also have an approximately linear phase in the passband. An IIR digital filter made from a parallel connection of two allpass subfilters can be viewed as a phase approximation problem. A block diagram of the system under investigation is given in Fig. 6-1, where $A_1(z)$ and $A_2(z)$ are transfer functions of real, stable allpasses having no common poles.

A well-known class of such filters is the lattice wave digital filter [2, 3], consisting of cascaded first- and second-order wave digital allpass filters, having some advantageous

properties such as low coefficient sensitivity in the passband and a low round-off noise level.

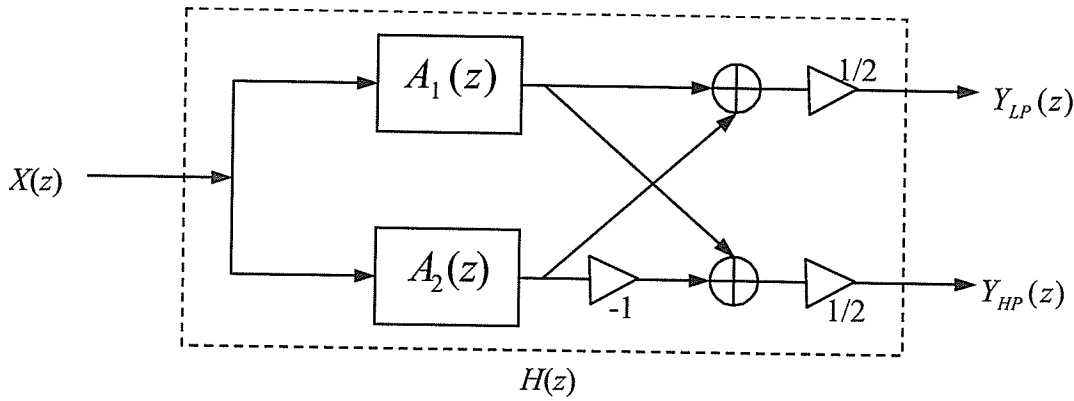


Fig. 6-1 IIR filter design using a parallel allpass filters

The overall frequency response can be written as

$$H(e^{j\omega}) = \frac{Y_{LP}}{X} = \frac{e^{j\beta_1(\omega)} + e^{j\beta_2(\omega)}}{2} \quad (6-1)$$

where $\beta_1(\omega)$ and $\beta_2(\omega)$ are the phase response of $A_1(z)$ and $A_2(z)$, respectively.

The overall frequency response can be written as

$$\begin{aligned} H(e^{j\omega}) &= \frac{e^{j\beta_1(\omega)} + e^{j\beta_2(\omega)}}{2} \\ &= \frac{e^{j(\beta_1(\omega) + \beta_2(\omega))/2}}{2} \left(e^{j(\beta_1(\omega) - \beta_2(\omega))/2} + e^{-j(\beta_1(\omega) - \beta_2(\omega))/2} \right) \\ &= e^{j(\beta_1(\omega) + \beta_2(\omega))/2} \cos\left(\frac{\beta_1(\omega) - \beta_2(\omega)}{2}\right) \end{aligned} \quad (6-2)$$

The magnitude function is consequently obtained by

$$|H(e^{j\omega})| = \left| \cos\left(\frac{\beta_1(\omega) - \beta_2(\omega)}{2}\right) \right| \quad (6-3)$$

The magnitude of the overall filter is thus limited by $|H(e^{j\omega})| \leq 1$.

The phase function is also given by

$$\beta(\omega) = \frac{\beta_1(\omega) + \beta_2(\omega)}{2} + \pi \cdot \text{sgn}\left(\cos\left(\frac{\beta_1 - \beta_2}{2}\right)\right) \quad (6-4)$$

$$\text{where } \text{sgn}(x) = \begin{cases} 1, & x \leq 0 \\ 0, & x > 0 \end{cases}$$

Thus, the design of the overall filter can be performed with the aid of the phase responses

$\beta_1(\omega)$ and $\beta_2(\omega)$, as shown in Fig. 6-2.

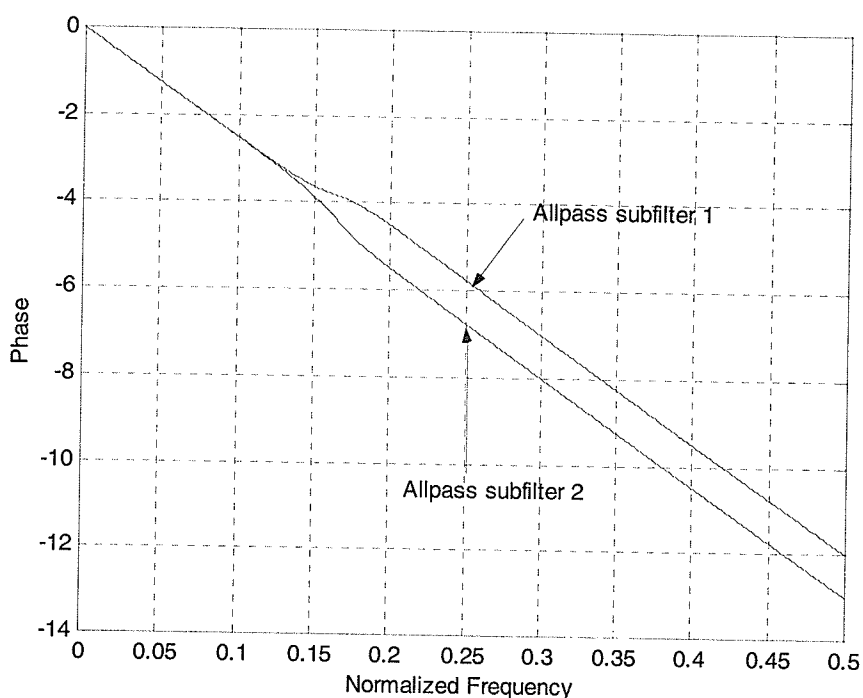


Fig. 6-2 the phase of two allpass subfilters

The main drawback of conventional filters realized in the form of equation (6-1) is that their phase response is nonlinear in the passband. There exist two methods to linearize the phase in the passband. The first method is to cascade $H(z)$ with an allpass filter as phase (or group delay) equalizer, which was considered in the previous chapter. The second method is to use the parallel allpass structure. As it is well known, a recursive digital filter with approximately linear phase can be designed, if one of the allpass

branches is a pure delay, while the other allpass is designed such that the desired selectivity is obtained. Several methods for the design of such systems have been published [28-33], [37-40].

In this thesis, we consider the second case in a more general sense that the IIR filter consists of a parallel connection of two general wave digital allpass filters. It uses a modified Remez-type algorithm and deals with $A_1(z)$ and $A_2(z)$ in such a way that their phase responses follow the same linear curve in the passband.

6.2 Approximately Linear Phase Approach

6.2.1 Phase approximation problem

We consider design of approximately linear phase (ALP) lowpass filters. Let ω_c and ω_s denote the passband and stopband edges respectively. The desired phase difference is

$$\beta_{pre}(\omega) = \beta_1(\omega) - \beta_2(\omega) = \begin{cases} 0 & \omega \in [0, \omega_c] \\ \pi & \omega \in [\omega_s, \pi] \end{cases} \quad (6-5)$$

Thus the two allpass transfer functions and $A_1(z)$ and $A_2(z)$ have to be found such that their phase difference $\beta_d(\omega)$ approximates $\beta_{pre}(\omega)$ according to the constraints determined by the tolerance scheme of one of the magnitude functions:

$$\begin{aligned} 1 - \delta_c(\omega) &\leq |H(e^{j\omega})| \leq 1, & \omega \in [0, \omega_c] \\ |H(e^{j\omega})| &\leq \delta_s(\omega) & \omega \in [\omega_s, \pi] \end{aligned} \quad (6-6a,b)$$

where δ_c and δ_s are the ripple factors in the passband and stopband, respectively.

The design yields two allpass subfilters, which describe general selective minimum phase systems as shown in Figure 6-2. The phase error function can be realized to minimize the phase error as

$$E(\omega) = W(\omega)[\beta_2(\omega) - \beta_1(\omega)] \quad (6-7)$$

where $W(\omega)$ is a weight function.

6.2.2 Design procedure

The phase approximation problem is a nonlinear problem, whereas the proposed techniques are based on the iterative adjustment of the phase and magnitude responses. The algorithm works in a way that is similar to the Remez multiple exchange algorithm for polynomial approximation problems. Using the results of the above, the *ALP* design algorithm for an odd N^{th} -order lowpass filter can be formulated.

First, the order of the allpass subfilters must be determined. It is known that their orders must differ by one. In addition, the order of $A_1(z)$ must be less than that of $A_2(z)$. The design procedure concentrates on $A_1(z)$ first. In the passband, $|H(e^{j\omega})| \approx 1$ and from (6-

3), we have $\beta_1(\omega) \approx \beta_2(\omega)$. For *ALP* it is also required that $\beta(\omega) \approx -\frac{N}{2}\omega$. Hence, in the

passband, both $\beta_1(\omega)$ and $\beta_2(\omega)$ should approximate to $-\frac{N}{2}\omega$. In the stopband,

$|H(e^{j\omega})| \approx 0$, we have $\beta_1(\omega) \approx \beta_2(\omega) + \pi$ from (6-3) and (6-5).

The magnitude of desired filter $|H(e^{j\omega})|$ can be obtained by using for e.g. Gazsi's method. Then the phase of A_1 can be determined

$$\beta_1(\omega) = -\frac{N}{2}\omega + \cos^{-1}\left(2 \cdot |H(e^{j\omega})|^2 - 1\right) \quad (6-8)$$

Thus, the A_1 subfilter coefficients can be obtained by the phase response [44].

First, it is easy to show that the phase response $\theta_A(\omega)$ of $A(z)$ can be written as

$$\theta_A(\omega) = -M\omega + 2 \tan^{-1} \left(\frac{\sum_{k=1}^M \alpha_k \sin(k\omega)}{1 + \sum_{k=1}^M \alpha_k \cos(k\omega)} \right) \quad (6-9)$$

where M is the order and α_k are the coefficients of $A(z)$. From Fig. 6-2, it is clear when ω goes from 0 to π radians, the phase $\theta_A(\omega)$ decrease monotonically from 0 to $-M\pi$.

Thus, we obtain a set of equations

$$\frac{\sum_{k=1}^M \alpha_k \sin(k\omega_i)}{1 + \sum_{k=1}^M \alpha_k \cos(k\omega_i)} = \tan(\beta_i) \quad i = 1, 2, \dots, M \quad (6-10)$$

where $\beta_i = (\theta_A(\omega_i) + M\pi)/2$. After some manipulations, the equation (6-10) can be expressed as

$$\sum_{k=1}^M [\sin(k\omega_i) - \tan(\beta_i) \cos(k\omega_i)] \alpha_k = \tan(\beta_i) \quad (6-11)$$

which is a linear equation of filter coefficients α_k . Thus, it becomes the following in matrix form

$$Q\alpha = p \quad (6-12)$$

where the two vectors are

$$\begin{aligned} \alpha &= [\alpha_1, \alpha_2, \dots, \alpha_{2M}]^T \\ p &= [\tan(\beta_1), \tan(\beta_2), \dots, \tan(\beta_{2M})]^T \end{aligned} \quad (6-13a,b)$$

and the elements of the matrix Q are given by

$$q_{i,k} = \sin(k\omega_i) - \tan(\beta_i) \cos(k\omega_i) \quad i = 1, \dots, M \quad k = 1, \dots, M \quad (6-14)$$

By solving linear matrix equation (6-12), the desired solution is obtained by

$$\alpha = Q^{-1} p \quad (6-15)$$

The next stage is to design A_2 . In this case, both passband and stopband have to be considered. In the passband, $\beta_2(\omega)$ must also approximate $-\frac{N}{2}\omega$. In the stopband, it must approximate $\beta_1 - \pi$. The extremal $N_2 + 1$ points in the passband and stopband can then be distributed uniformly over the union of the two bands.

The complete algorithm can be summarized as follows:

1. Read the specification of the lowpass filter ω_c , ω_s , A_p , A_s , N and determine the

order of $A_1(z)$ and $A_2(z)$, $N_1 = \frac{N-1}{2}$ and $N_2 = \frac{N+1}{2}$, respectively.

2. Assume the order of the upper branch subfilter $A_1(z)$ as one less than that of $A_2(z)$.

Use Gazsi's method to obtain the desired magnitude of $A_1(z)$.

3. Determine the coefficients of A_1 using formulas from (6-11)-(6-15).
4. Select initial extremal points $\Omega = \{\omega_1, \omega_2, \dots, \omega_{N_2+1}\}$ on $[0, \omega_c] \cup [\omega_s, \pi]$.
5. Solve the system of $N_2 + 1$ equations

$$E(\omega_i) = W(\omega_i)[\beta_2(\omega_i) - \beta_1(\omega_i)] = (-1)^i \delta, \quad (6-16)$$

where $i = 1, \dots, N_2 + 1$ for δ and the filter coefficients α_i of $H_{A_2}(z)$.

6. Find the $N_2 + 1$ local extrema of $E(\omega)$ on $[0, \omega_c] \cup [\omega_s, \pi]$ with the condition that the maxima and minima alternate. Let the angles at which these extrema occur be $\Omega' = \{\omega'_1, \omega'_2, \dots, \omega'_{N_2+1}\}$.

7. If $|\Omega - \Omega'| < \varepsilon$, for $i = 1, \dots, N_2 + 1$ then go to next step. Otherwise, set $\Omega = \Omega'$ and recalculate.

8. Compute the coefficients of A_2 using formulas from (6-11)-(6-15).

In the fourth step of the algorithm, $N_2 + 1$ initial extrema points must be selected. A good set of initial points is to select the number of points in the passband as $\omega_c(N_2 + 1)/(\omega_c + \pi - \omega_s)$ rounded to the nearest integer [28]. The extreme points in the passband and stopband can then be distributed equidistantly in each respective band.

In step five, equation (6-16) is to be solved. The coefficients can be computed as functions of δ . For a given δ , we can find

$$\beta_2(\omega_i) = \frac{(-1)^i \delta}{W\omega_i} + \beta(\omega_i) \quad i = 1, 2, \dots, N_2 \quad (6-17)$$

and

$$f(\delta) = W(\omega_i)[\beta_2(\omega_i) - \beta_1(\omega_i)] - (-1)^i \delta \quad (6-18)$$

for $i = N_2 + 1$.

Next the bisection method is used to find the δ so that $f(\delta) = 0$. Finally, once the overall algorithm has converged, the coefficient of A_2 can be obtained.

Chapter 7

Design Examples

In this chapter, we design some approximately linear phase wave digital lattice filters using the algorithms described in Chapters 4 to 6. For each filter design, the attenuation response will be presented graphically, giving the resulting passband and stopband ripples. The group delay (and phase) response as well as the pole-zero plots will also be given. The coefficient values are tabulated for each design. The proposed design algorithms are implemented using MATLAB. Finally, the results for the different approximation methods are collected.

7.1 Examples using pole-zero placement

In this subsection, the modified pole-zero placement method described in Chapter 4 is employed to design two examples. At first, the modified pole-zero placement is used to obtain the desired filter to meet the specification. Then one quad of zeros of h is moved to adjust the group delay in the passband.

Example 1: Example 1 uses the specification from Example 1 in [37]. The specification considered is as follows: $\omega_c = 18^\circ$, $\omega_s = 36^\circ$, and $A_s = 40$. The maximum allowable peak-to-peak passband ripple is 0.1dB. In this example, the order of the filter meeting these requirements is 7. The filter having approximately linear phase is realized such that one quad of zeros of h is not on the unit circle.

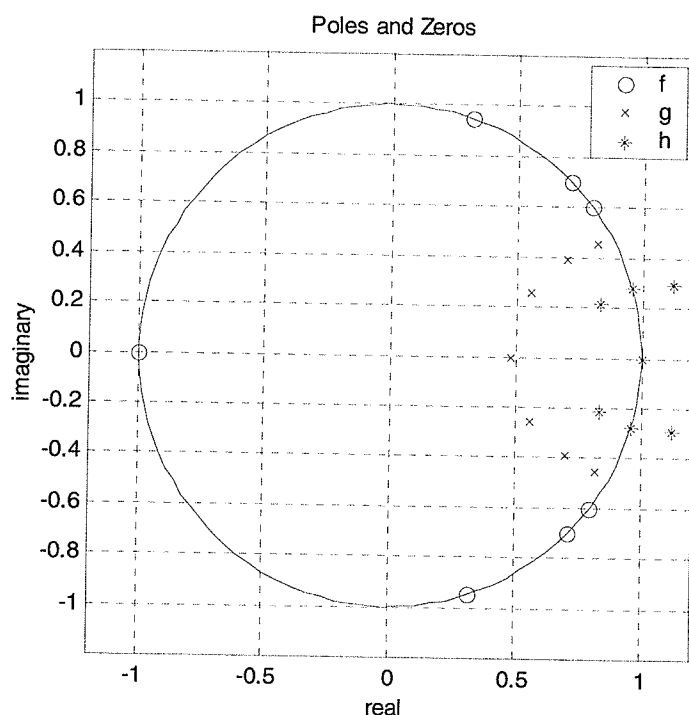
The adaptor coefficients of the filter are given in Table 1. Fig. 7-1 shows the zeros of f , g and h in the z -domain. The attenuation in the whole region and in the passband are shown

in Fig.7-2. The group delay and phase in the passband are shown in Figs.7-3 and 7-4, respectively. The solid curve in Figs 7-3 and 7-4 correspond to the proposed filter with one adjusted quad of zeros of h , and the dashed curve correspond to the filter without one adjusted quad of zeros of h . The structure of the 7-order WDLF is shown in Fig. 7-5.

Table 1 Adaptor coefficients of the filter in example 1

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.48402531087159	1	-0.64927016376335	0.85246699381758
A_2			2	-0.88533113277563	0.87343606560336
			3	-0.38340823013731	0.81219892309779

where $j = 1, \dots, M$, $i = 1, \dots, N$. M and N is the number of first- and second-order allpass filters, respectively.

Fig. 7-1 the zeros of f , g and h in z -domain

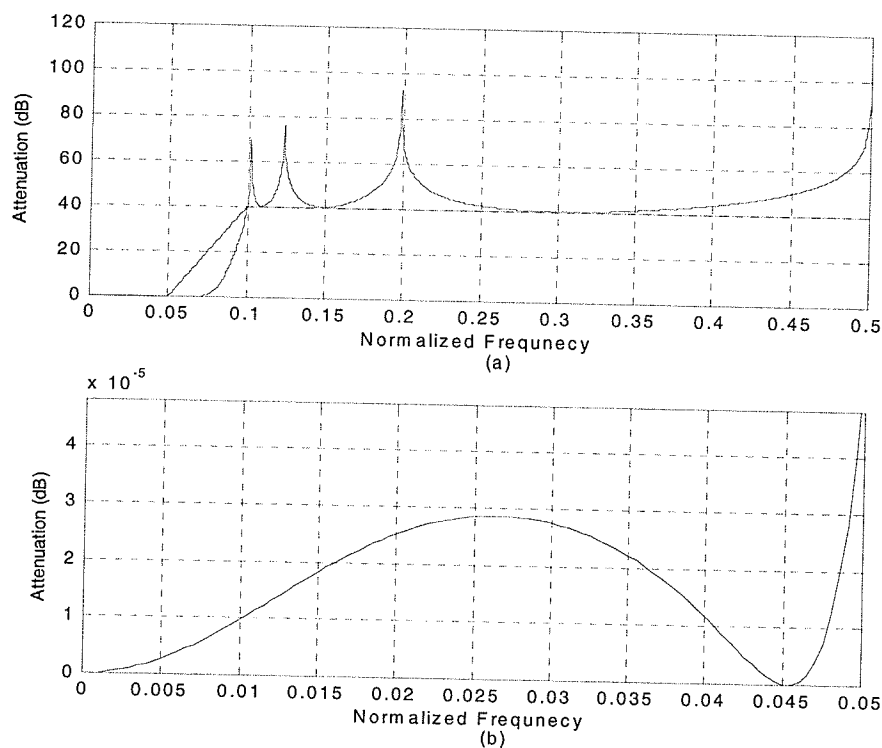


Fig. 7-2 The attenuation of the 7-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

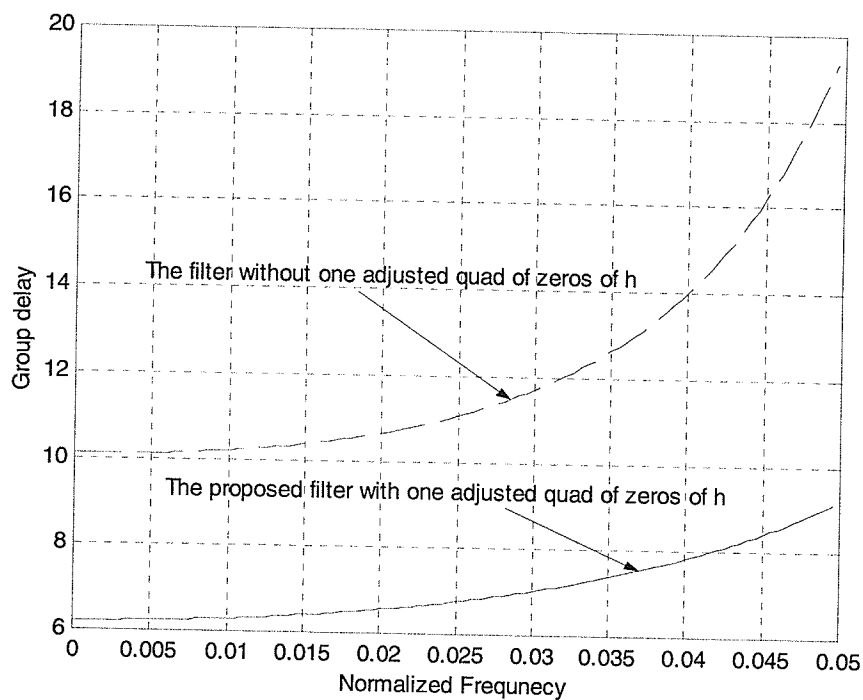


Fig. 7-3 The group delay of the 7-order approximately linear phase WDLF

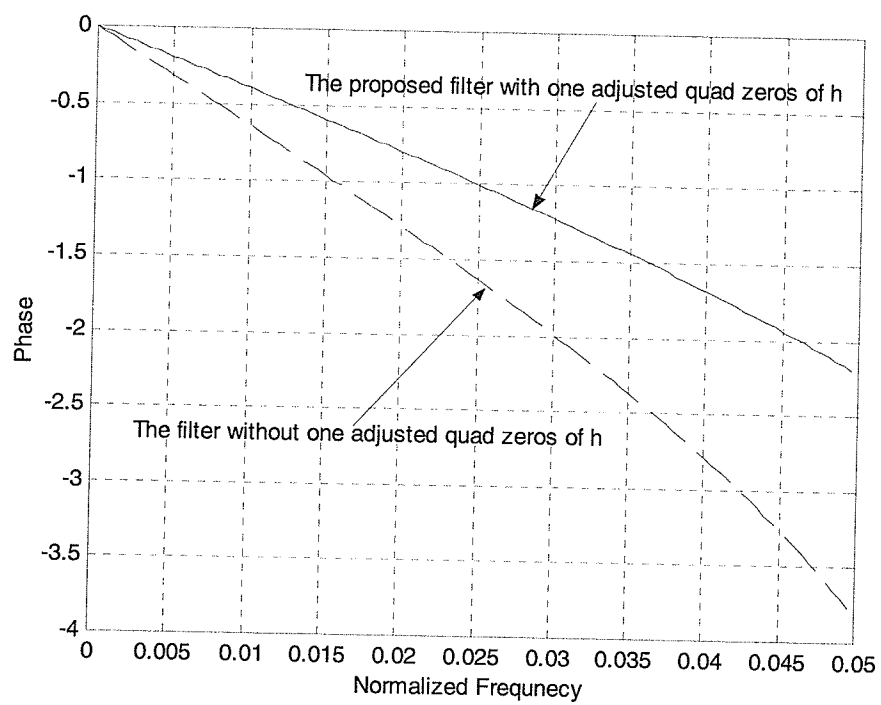


Fig. 7-4 The phase of the 7-order approximately linear phase WDLF

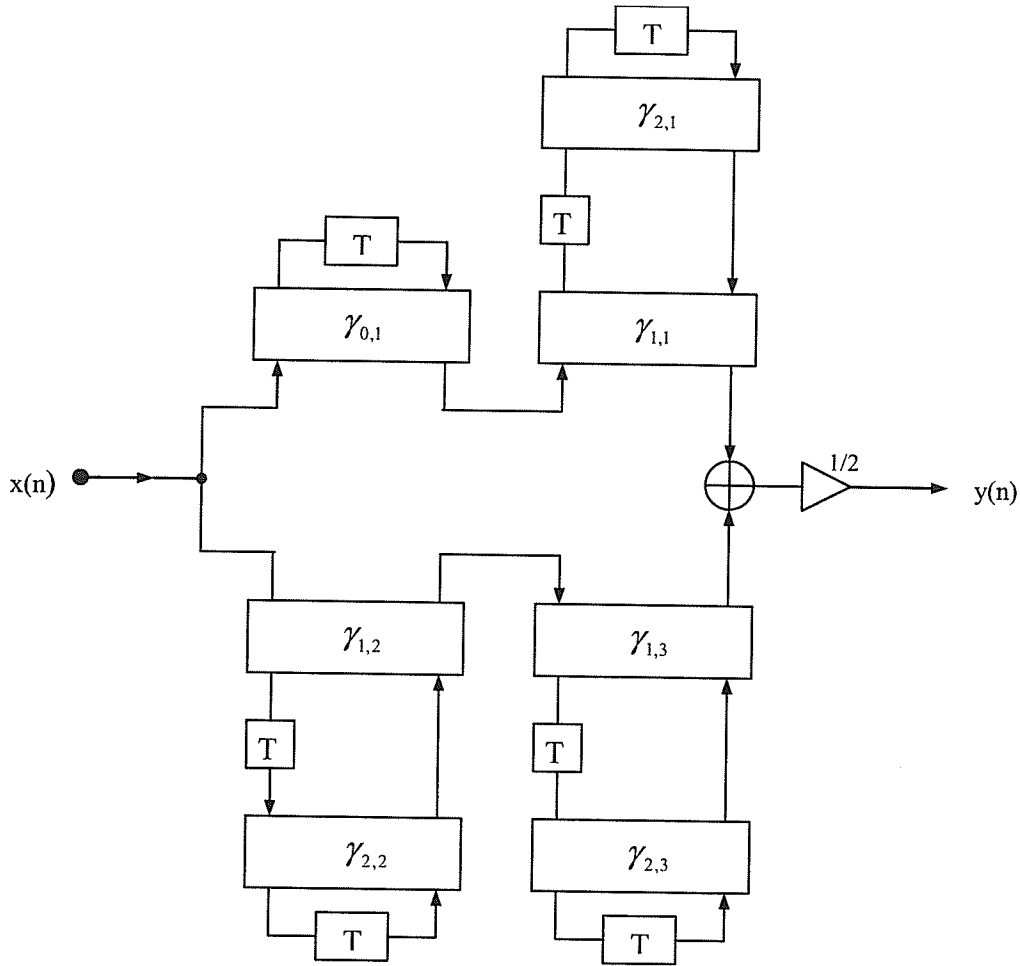


Fig. 7-5 The structure of the 7-order approximately linear phase WDLF using pole-zero placement

Example 2: The specification of Example 2 is given as the normalized passband edge at 0.1 and the normalized stopband edge at 0.2. The maximum allowable peak-to-peak passband ripple is 0.1 dB and the minimum stopband attenuation has three regions 50dB $\omega \in [0.2, 0.23]$, 60dB $\omega \in [0.23, 0.35]$ and 45dB $\omega \in [0.35, 0.5]$. In this example, we design a 9-order WDLF with one quad of zeros of h that is not on the unit circle.

The adaptor coefficients of the filter are given in Table 2. Fig. 7-6 shows the zeros of f , g and h in the z -domain, respectively. The attenuation in the whole region and in the passband are shown in Figs. 7-7. The group delay and phase in the passband are shown in Figs. 7-8 and 7-9, respectively. The solid curve in Figs. 7-8 and 7-9 correspond to the

proposed filter with one adjusted quad of zeros of h , and the dashed curve correspond to the filter without one adjusted quad of zeros of h .

Table 2 Adaptor coefficients of the filter in example 2

<div> <div> Coefficients </div> <div> Allpass </div> </div>	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.20015965295097	1	-0.32588561281678	0.45227971374174
			2	-0.84432486438242	0.51904823099232
A_2			3	-0.12396558413788	0.40737768338268
			4	-0.57249569209895	0.49381138827128

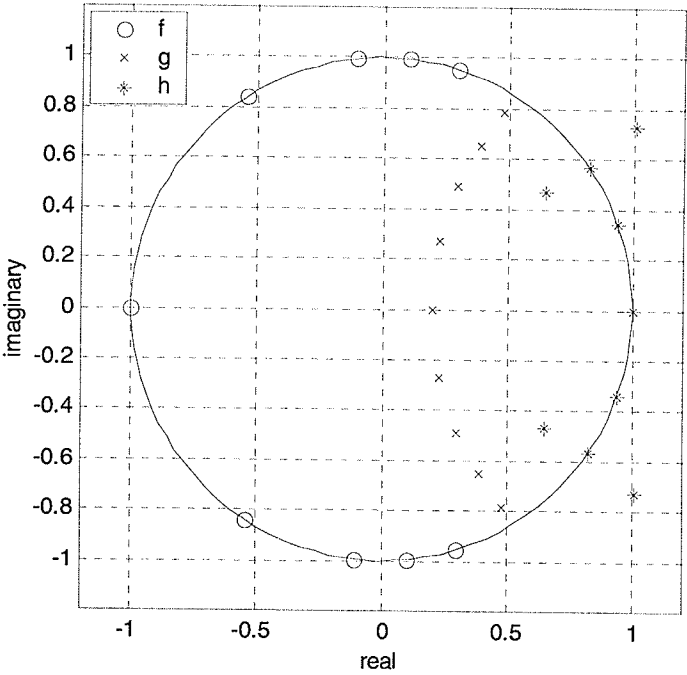


Fig. 7-6 the zeros of f , g and h in z -domain

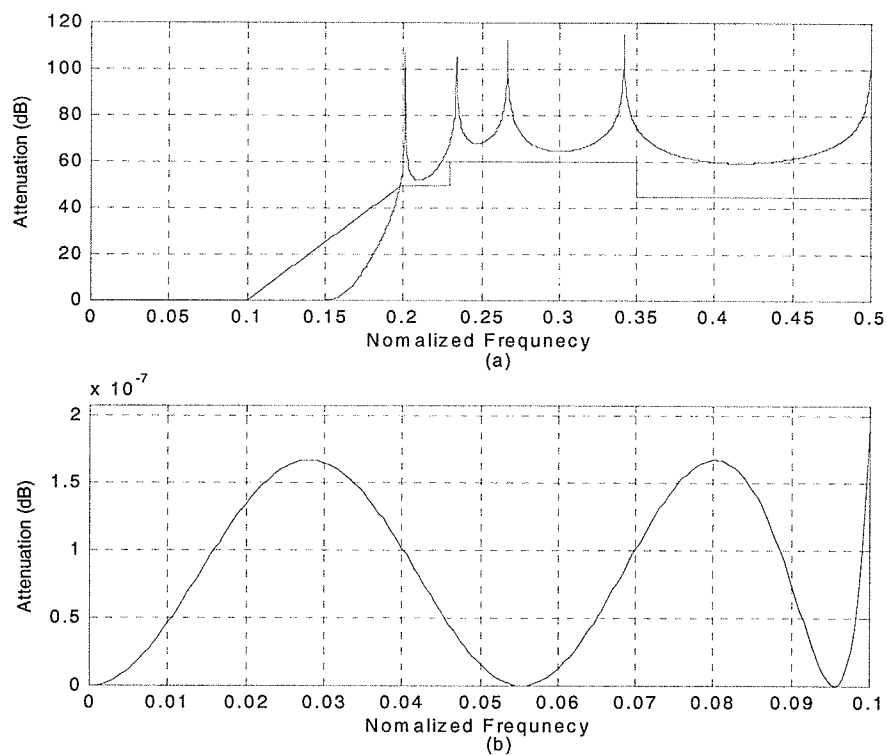


Fig. 7-7 The attenuation of the 9-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

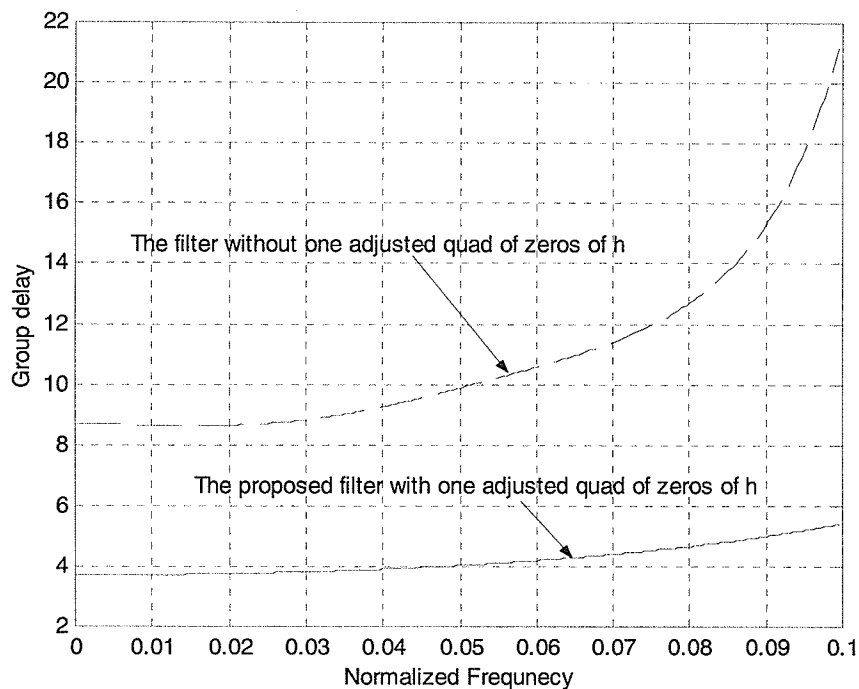


Fig. 7-8 The group delay of the 9-order approximately linear phase WDLF

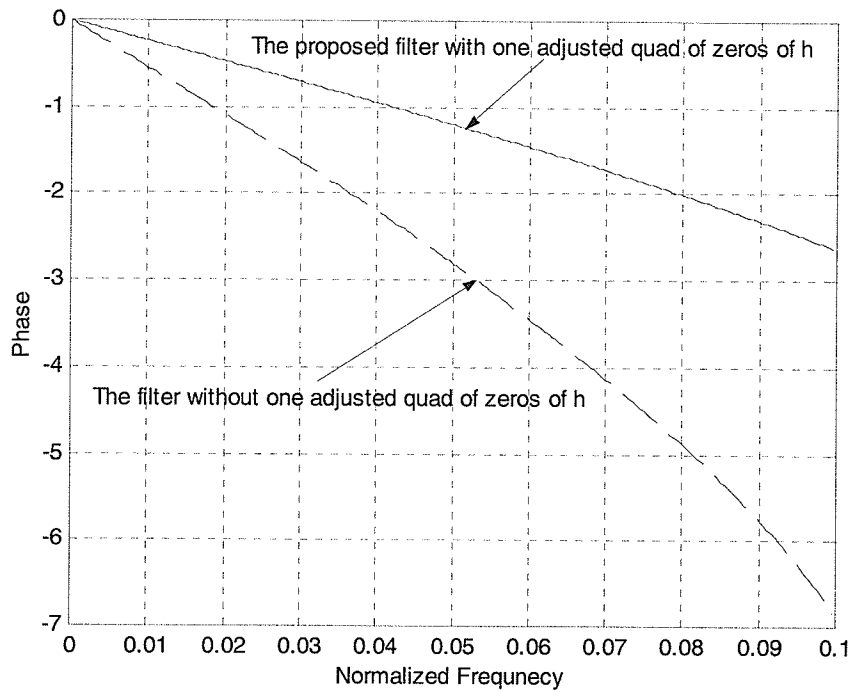


Fig. 7-9 The phase of the 9-order approximately linear phase WDLF

7.2 Examples using group delay equalization

In this section, we design an approximately linear phase WDLF cascaded with an allpass compensation filter as described in Chapter 5. At first, the modified pole-zero placement or Gazsi's method can be applied to design a filter to satisfy its attenuation specification. Then the nonlinear least square algorithm is used to obtain the allpass compensation filter. Finally, the equiripple flat group delay of the overall filter is obtained.

Example 3: The filter specification is given as the normalized passband edge at 0.1 and the normalized stopband edge at 0.15. The maximum allowable peak-to-peak passband ripple is 0.01dB and the minimum stopband attenuation is 60 dB.

In the design of the Cauer WDLF filter with approximately flat group delay in the passband, we first use the modified pole-zero placement to design the attenuation response with equiripple both in the passband and in the stopband meeting the specification. The order of the Cauer WDLF filter is 7. Then, the nonlinear least square algorithm is applied to design the 7-order allpass compensation filter. Finally, the overall 14-order filter is obtained with approximately flat group delay in the passband.

The adaptor coefficients of the filter are given in Table 3. The structure of the overall 14-order filter is shown in Fig. 10. The attenuation is plotted in Fig. 7-11. In Fig. 7-12, The group delay in the whole region and in the passband is plotted. The solid curve in Fig. 7-12 corresponds to the 7-order WDLF cascaded with a 7-order allpass filter, and the dashed curve corresponds to the 7-order WDLF. Fig. 7-13 illustrates the group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband. The zeros and poles of the WDLF and the allpass filter are shown in Fig. 7-14. The zeros and poles of the overall filter are shown in Fig. 7-15, respectively. Fig. 7-16 demonstrates the group delay equalization with different order allpass compensation filters.

Table 3 Adaptor coefficients of the filter in example 3

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.64623825173618	1	-0.70962719217858	0.81062384990040
A_2			2	-0.90516062926926	0.77597863135681
			3	-0.51178521137636	0.87313013420576
Allpass	2	0.83419795842662	4	-0.69978993594191	0.96860181863933
			5	-0.70934083903796	0.92276081363828
			6	-0.73621717981132	0.84790445895594

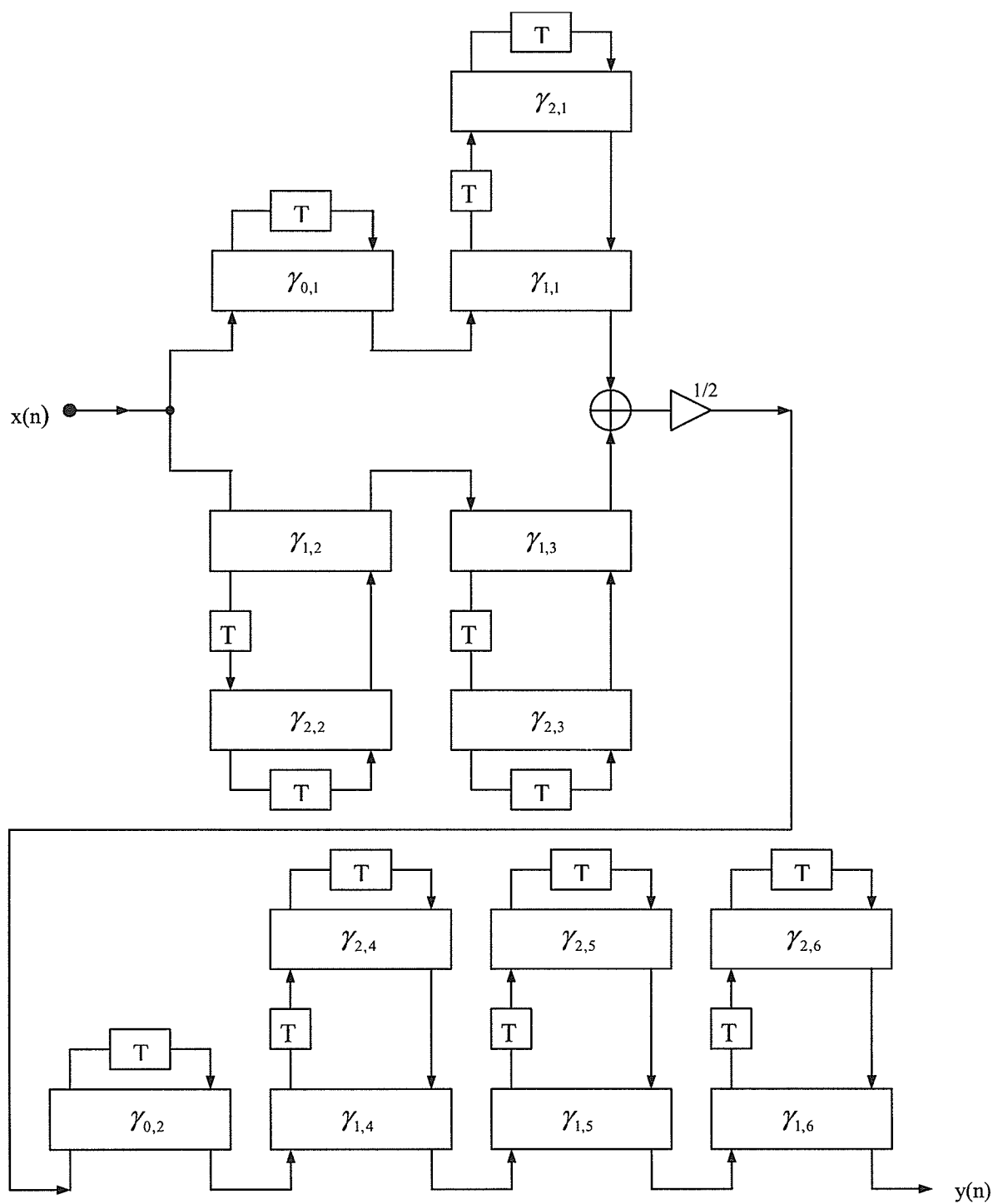


Fig. 7-10 The structure of the overall 14-order filter consists of 7-order WDLF cascaded 7-order allpass compensation filter

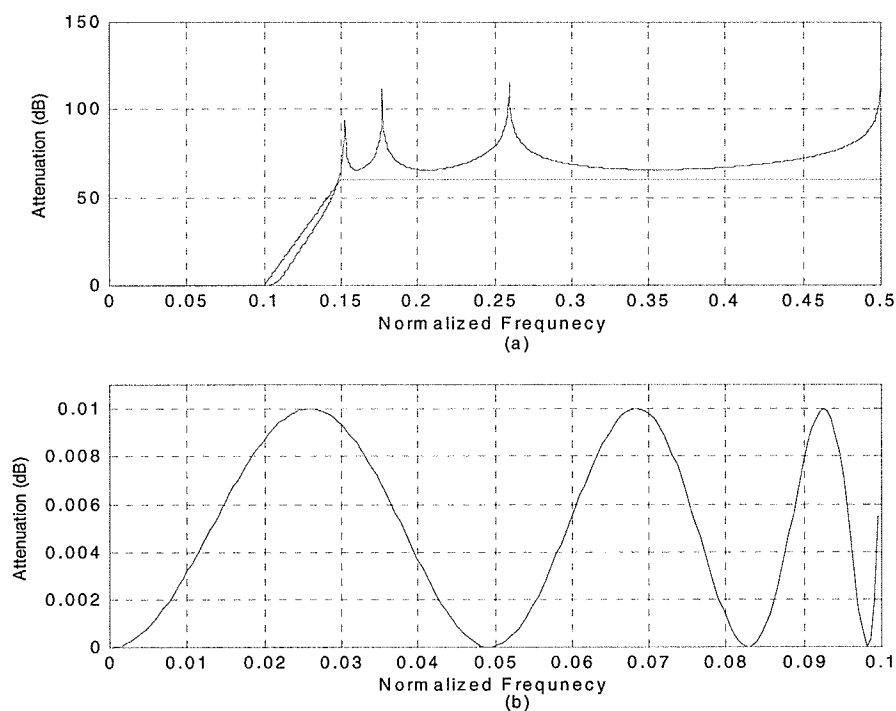


Fig. 7-11 The attenuation of the 14-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

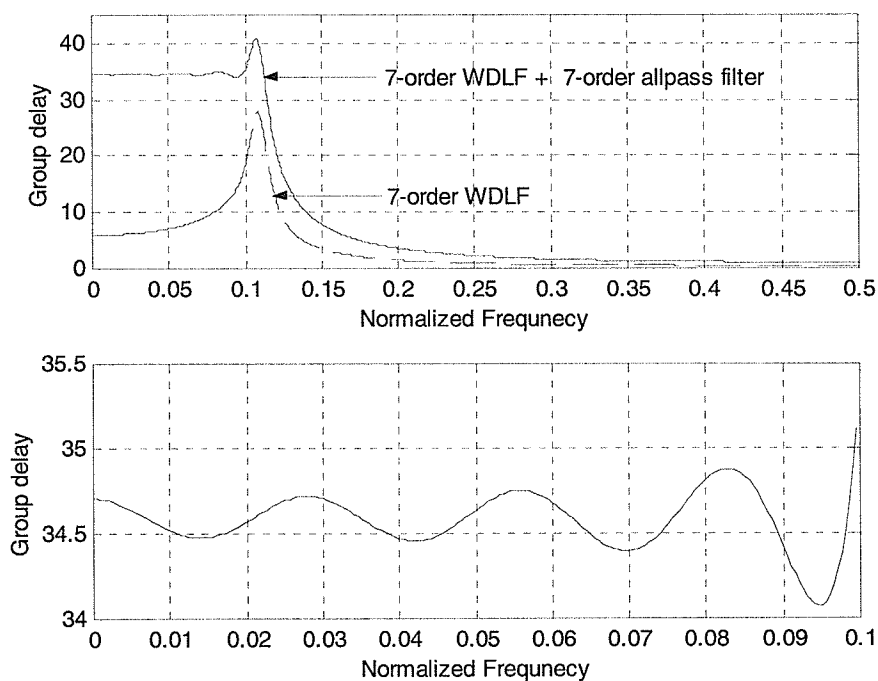


Fig. 7-12 The group delay of the 14-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

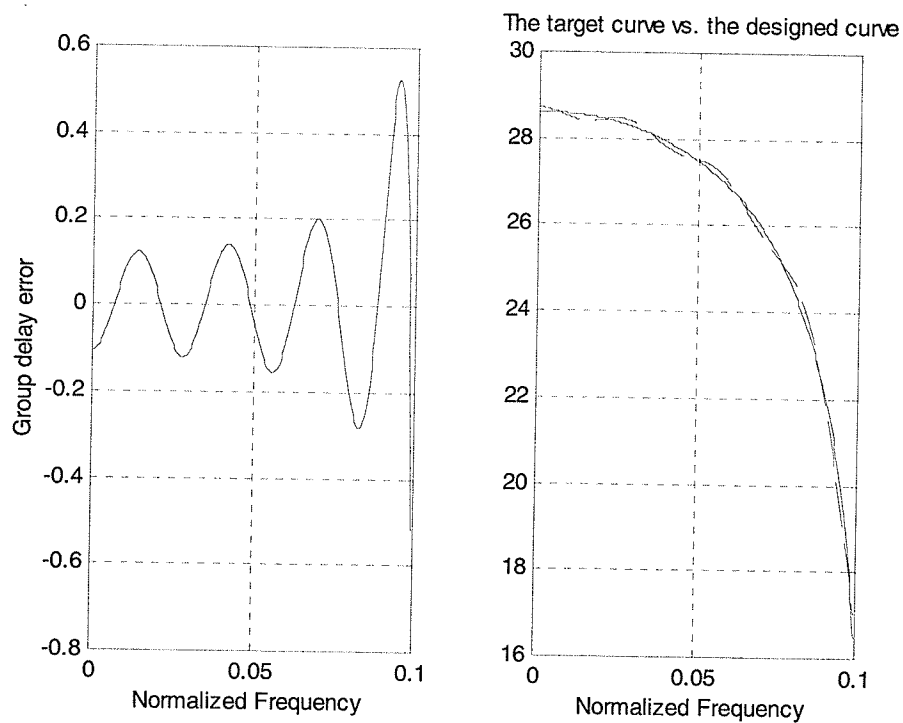


Fig. 7-13 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband

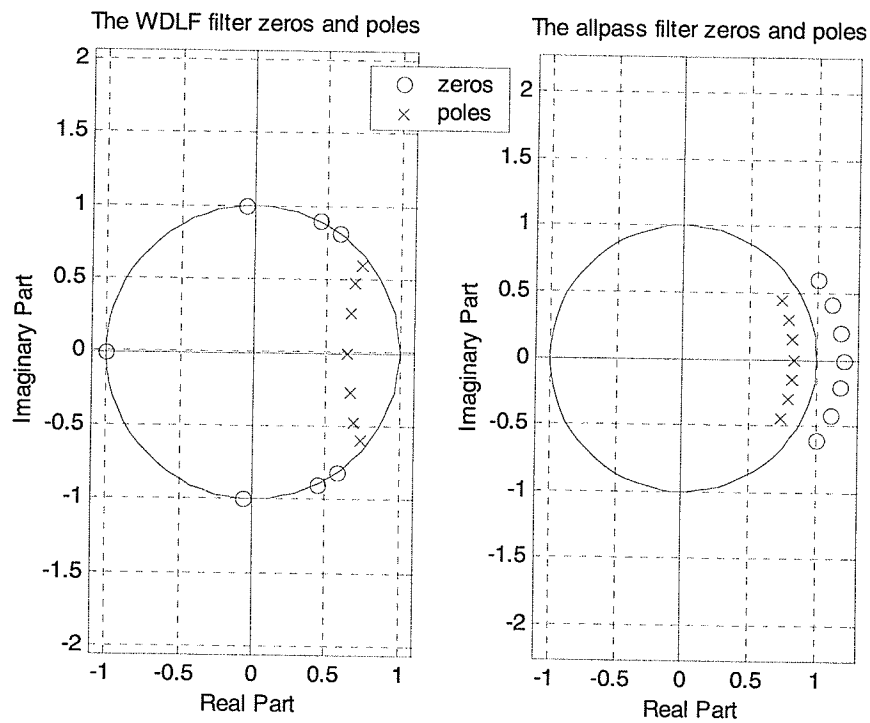


Fig. 7-14 The zeros and poles of 7-order WDLF and 7-order allpass compensation filter

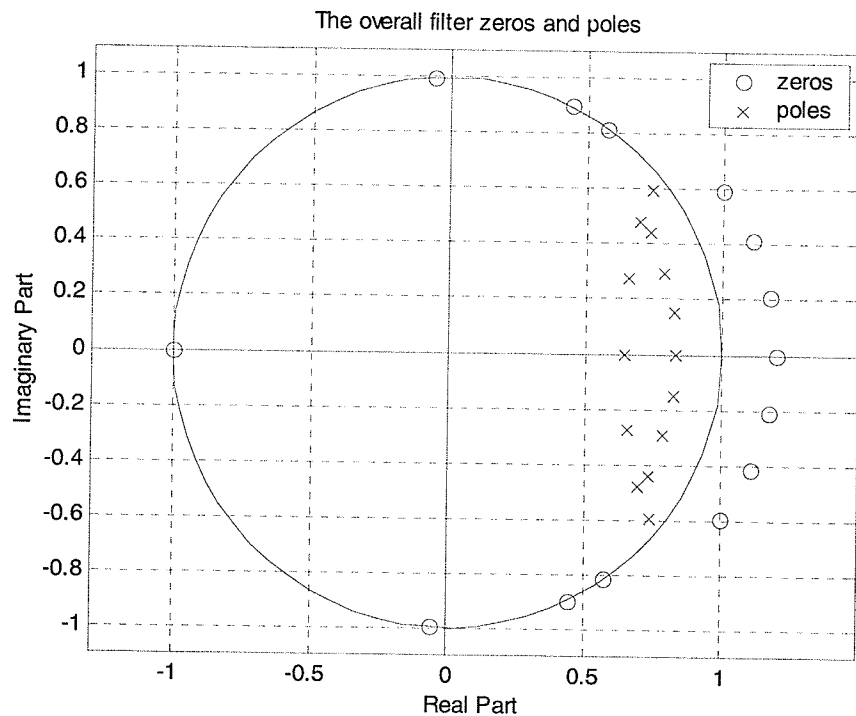


Fig. 7-15 The zeros and poles of the overall filter

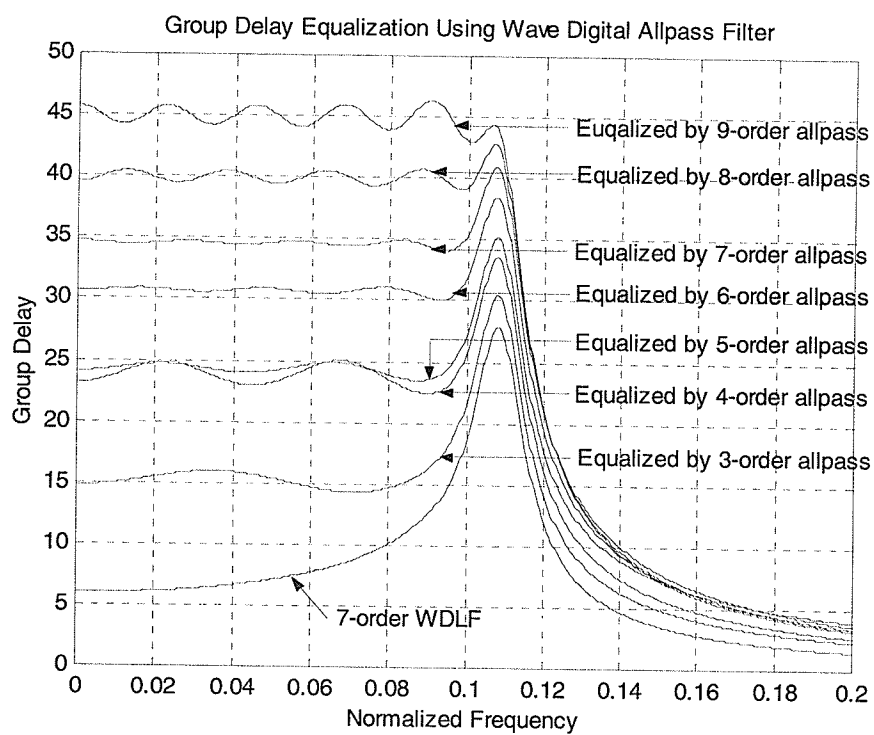


Fig. 7-16 The group delay equalization using different allpass compensation filters

Now, an approximately flat delay in the passband for an inverse Chebyshev WDLF is considered. We first use the modified pole-zero placement to design the attenuation response maximally flat in the passband and equiripple in the stopband meeting the specification. The order of the inverse Chebyshev WDLF filter is 11. Then a 6-order allpass compensation filter is cascaded to adjust the phase. The adaptor coefficients of the filter are given in Table 4. The attenuation is plotted in Fig. 7-17. In Fig. 7-18, the group delay in the whole region and in the passband is plotted. The solid curve in Fig. 7-18 corresponds to the 11-order WDLF cascaded 6-order allpass filter, and the dashed curve corresponds to the 11-order WDLF. Fig. 7-19 illustrates the group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband. The zeros and poles of the WDLF and the allpass filter are shown in Fig. 7-20. The zeros and poles of the 17-order overall filter are shown in Fig. 7-21.

Table 4 Adaptor coefficients of the filter in example 3

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.23124642352319	1	-0.27024420565435	0.57360359410895
			2	-0.67071392623179	0.70375127129424
A_2			3	-0.11481178965048	0.48330227823024
			4	-0.46638419255594	0.65303120219185
			5	-0.88302267492096	0.72743532853474
Allpass			6	-0.54756657172453	0.95174472061538
			7	-0.57741158027579	0.91824260936579
			8	-0.62621767991188	0.83771224492348

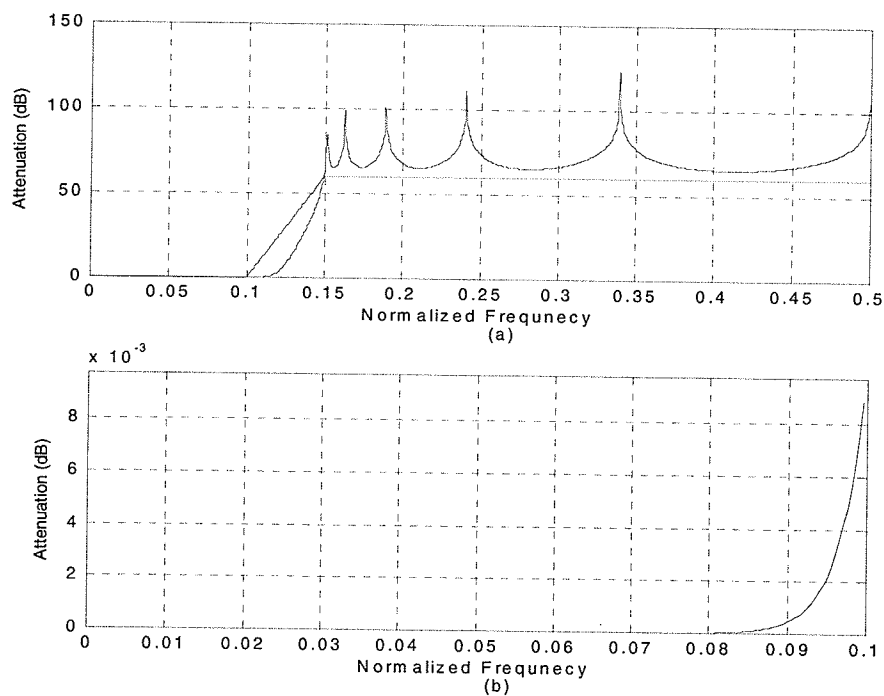


Fig. 7-17 The attenuation of the 17-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

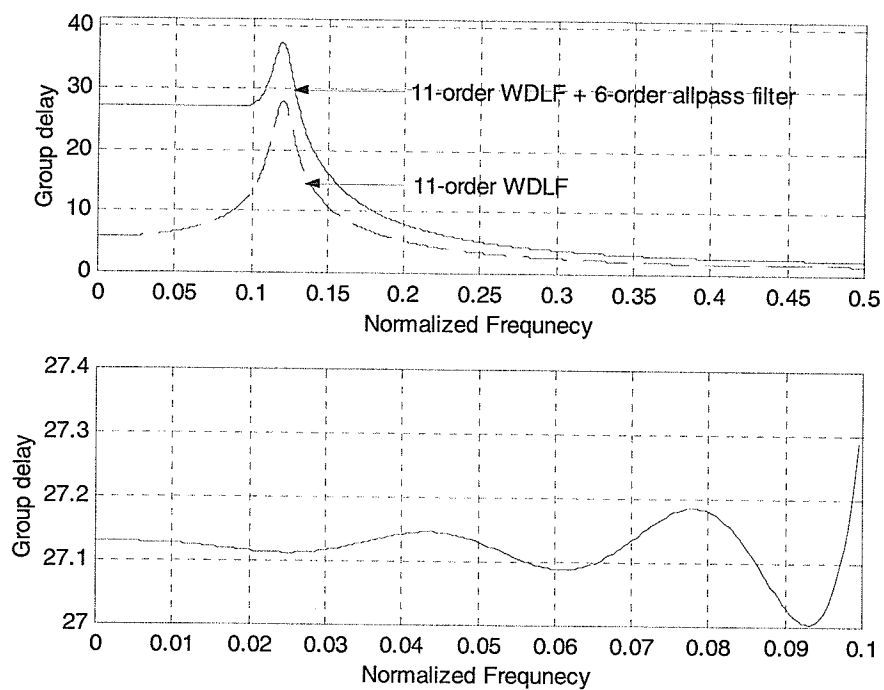


Fig. 7-18 The group delay of the 17-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

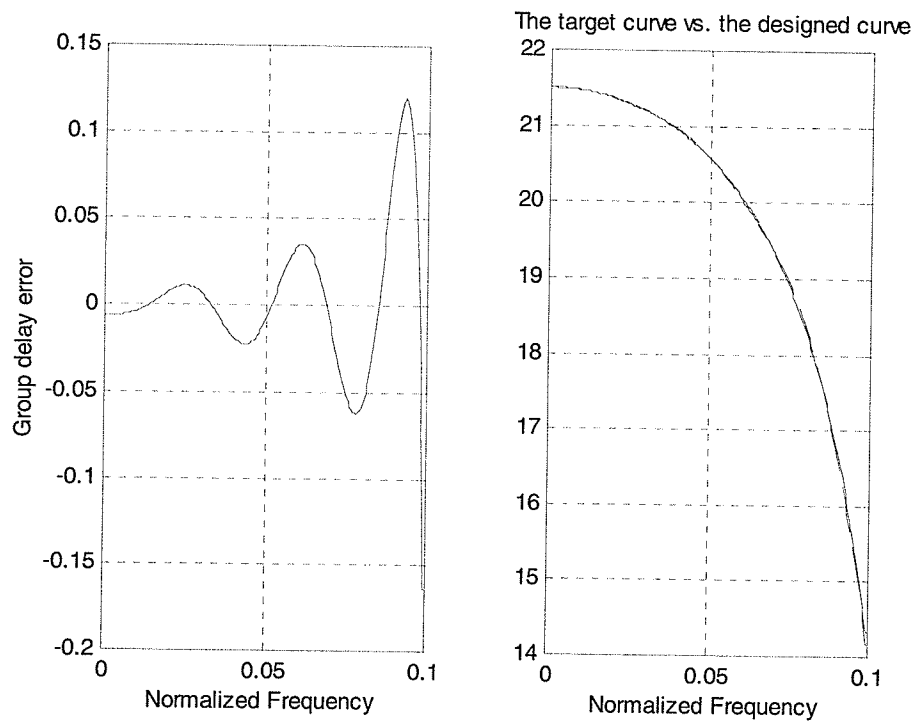


Fig. 7-19 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband

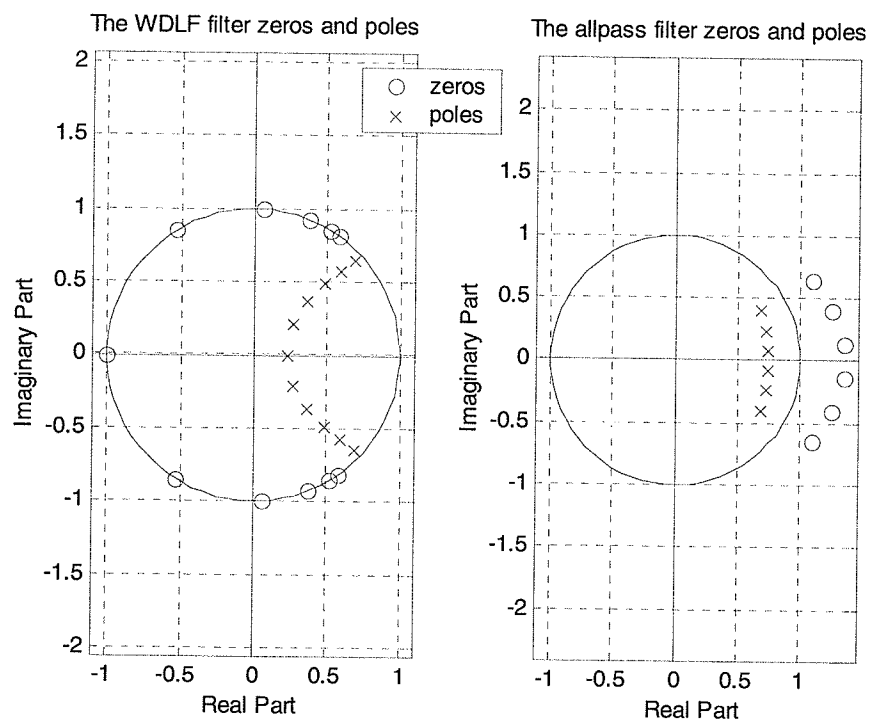


Fig. 7-20 The zeros and poles of 11-order WDLF and 6-order allpass compensation filter

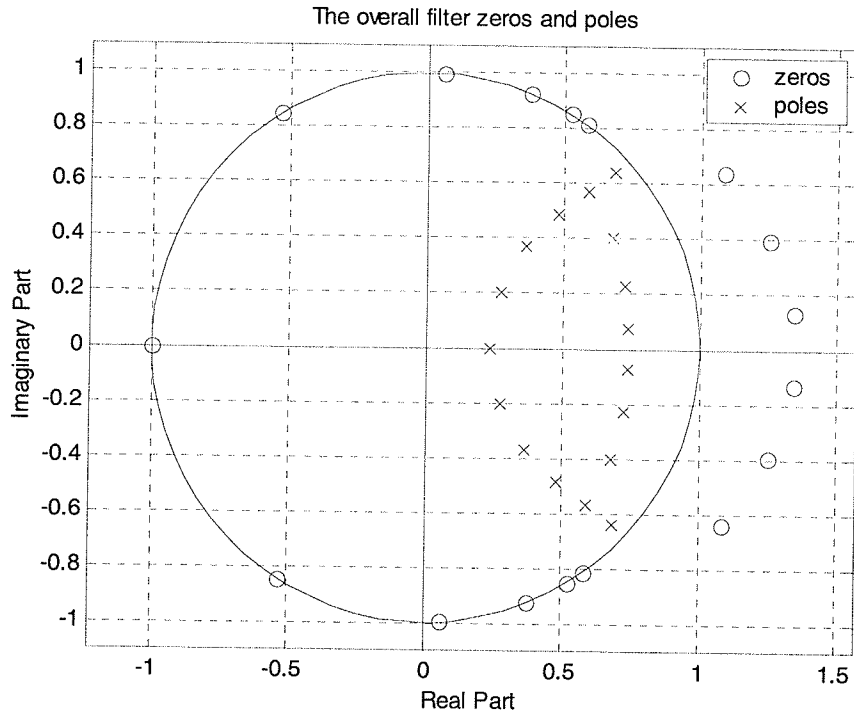


Fig. 7-21 The zeros and poles of the overall filter

Example 4: Example 4 is also chosen from Example 1 in [37]. The specification is the same as above. First, Gazsi's method [3] is used to design a 4-order Cauer WDLF satisfying the specification. Then, a 5-order allpass compensation is applied to equalize the designed filter's group delay.

The adaptor coefficients of the filter are given in Table 5. The attenuation is plotted in Fig.7-22. In Fig.7-23, the group delay in the whole region and in the passband is plotted. The solid curve in Fig. 7-23 corresponds to the 4-order Cauer WDLF cascaded 5-order allpass filter, and the dashed curve corresponds to the 4-order Cauer WDLF. Fig. 7-24 illustrates the group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband. The zeros and poles of the WDLF and the allpass filter are shown in Fig. 7-25. The zeros and poles of the 9-order overall filter are shown in Fig. 7-26.

Table 5 Adaptor coefficients of the filter in example 4

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1			1	-0.75866212828583	0.96331249546347
A_2	1	-0.89552297521926			
	2	0.69938551599336			
Allpass	3	0.77355386194320	2	-0.43869592969219	0.91935062562321
			3	-0.69502970630453	0.96128279555252

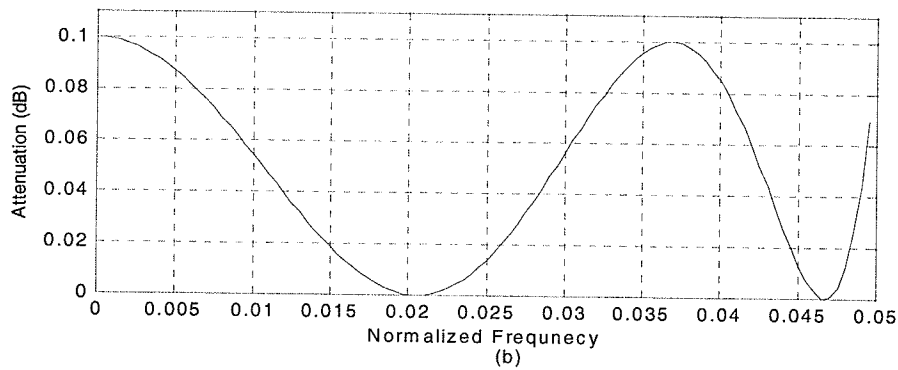
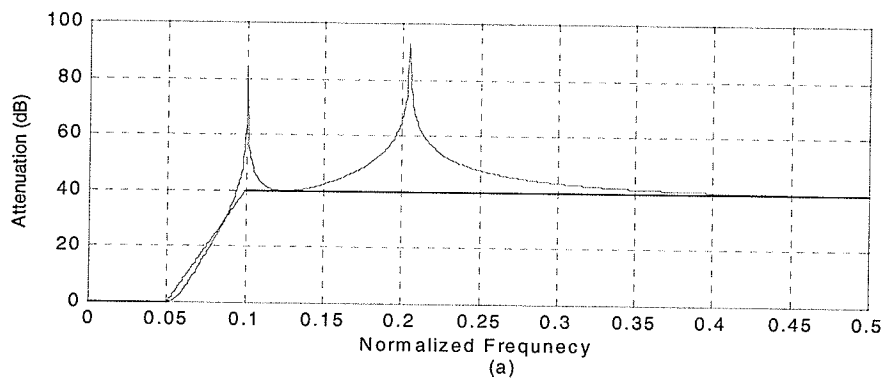


Fig. 7-22 The attenuation of the 9-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

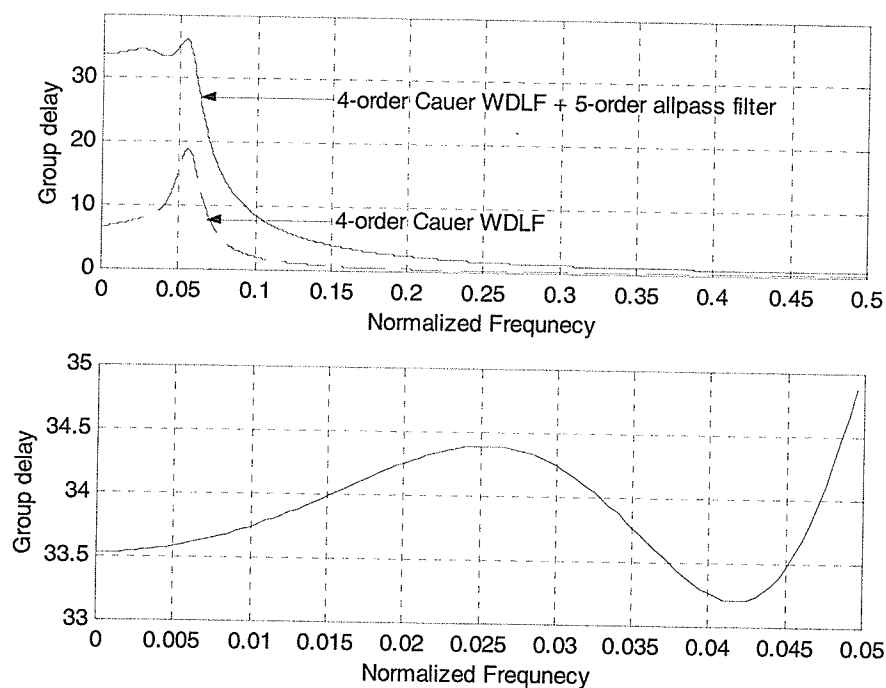


Fig. 7-23 The group delay of the 9-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

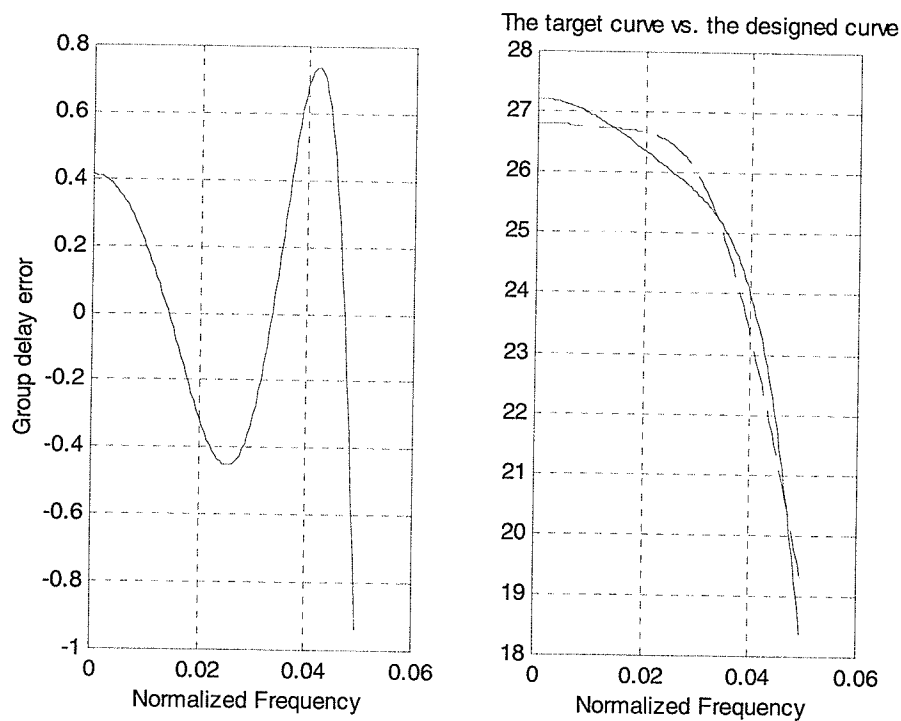


Fig. 7-24 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband

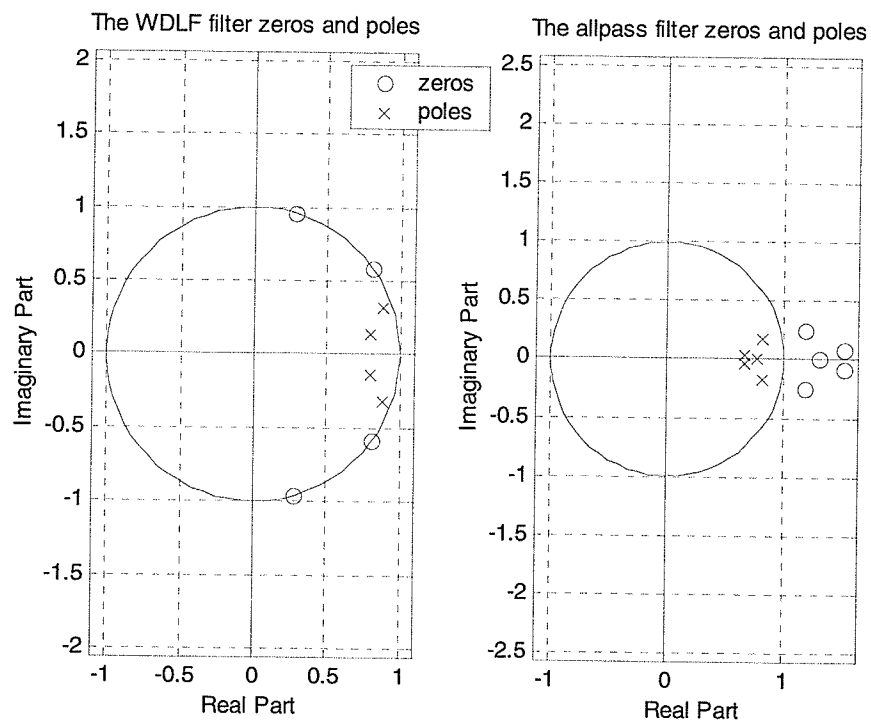


Fig. 7-25 The zeros and poles of 4-order WDLF and 5-order allpass compensation filter

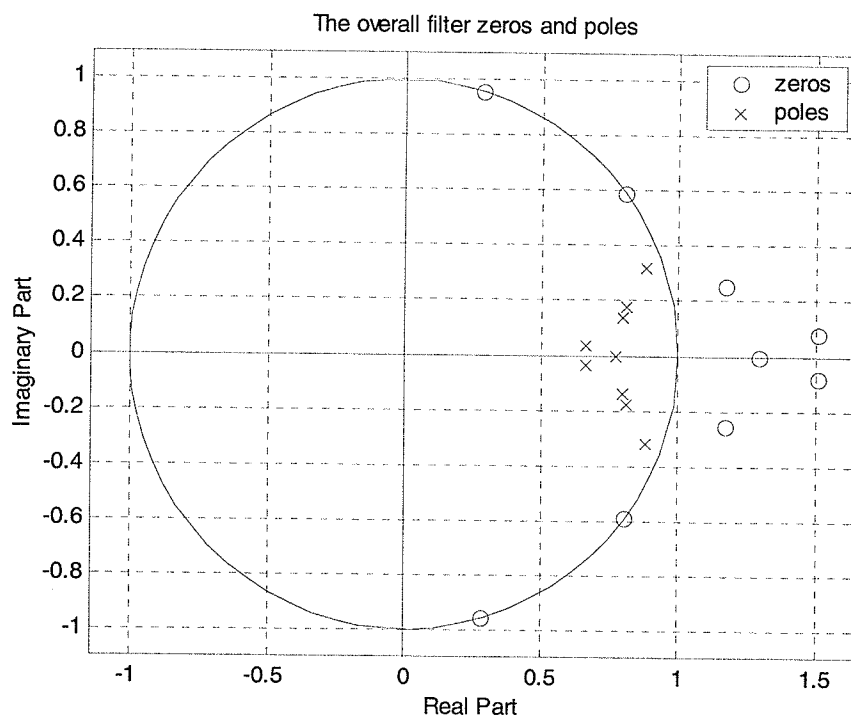


Fig. 7-26 The zeros and poles of the overall filter

Now, Gazsi's method [3] is used to design a 7-order inverse Chebyshev WDLF satisfying the specification. Then, 6-order allpass compensation is applied to equalize the designed filter's group delay.

The adaptor coefficients of the filter are given in Table 6. The attenuation is plotted in Fig.7-27. In Fig.7-28, the group delay in the whole region and in the passband is plotted. The solid curve in Fig. 7-28 corresponds to the 7-order inverse Chebyshev WDLF cascaded 6-order allpass filter, and the dashed curve corresponds to the 7-order inverse Chebyshev WDLF. Fig. 7-29 illustrates the group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband. The zeros and poles of the WDLF and the allpass compensation filter are shown in Fig. 7-30. The zeros and poles of the 13-order overall filter are shown in Fig. 7-31.

Table 6 Adaptor coefficients of the filter in example 4

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.43794658924586	1	-0.61389996692349	0.85002160404534
A_2			2	-0.33851820358150	0.78560302015257
			3	-0.87125171178117	0.87914260990579
Allpass			4	-0.49454465738833	0.94083581608900
			5	-0.56240178027900	0.94226848292028
			6	-0.63143305380558	0.91258651002009

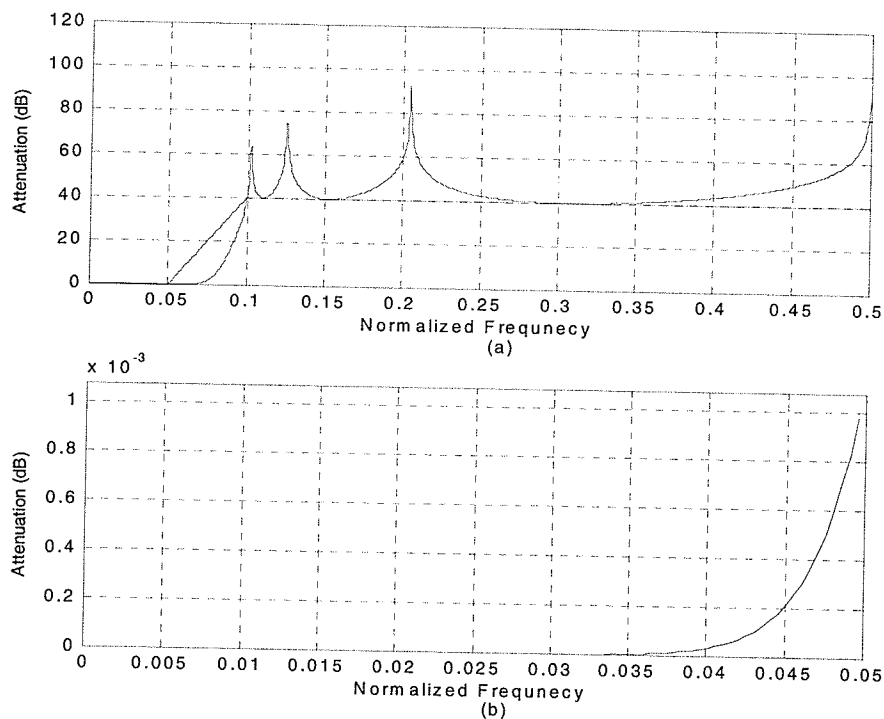


Fig. 7-27 The attenuation of the 13-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

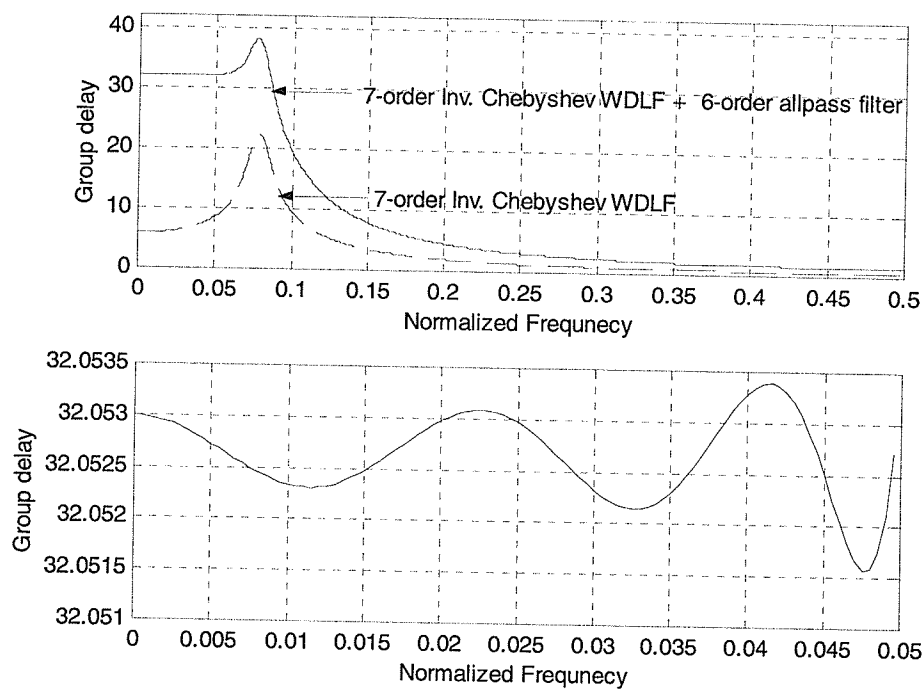


Fig. 7-28 The group delay of 13-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

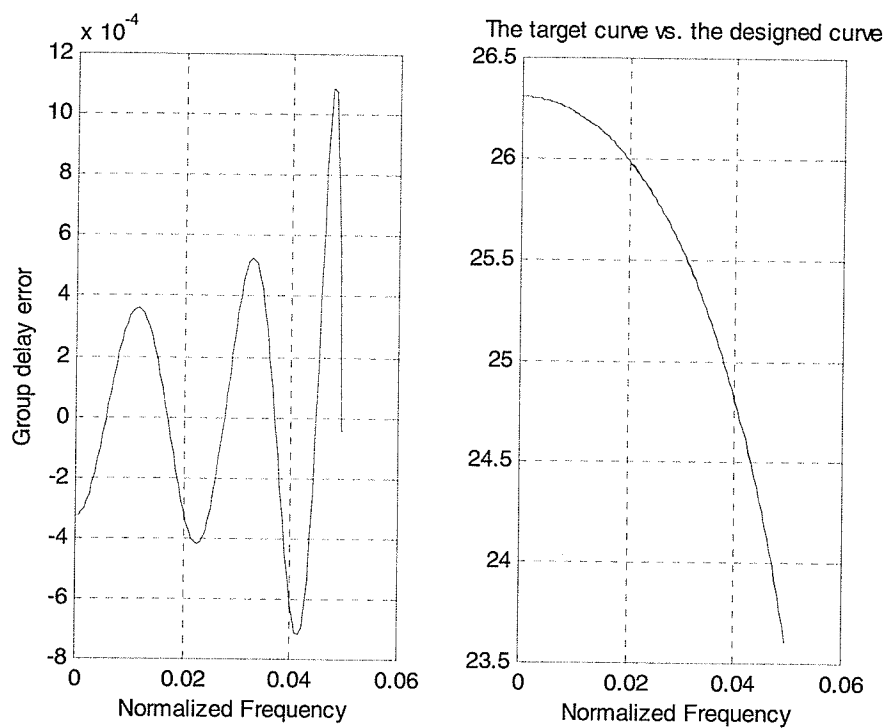


Fig. 7-29 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband

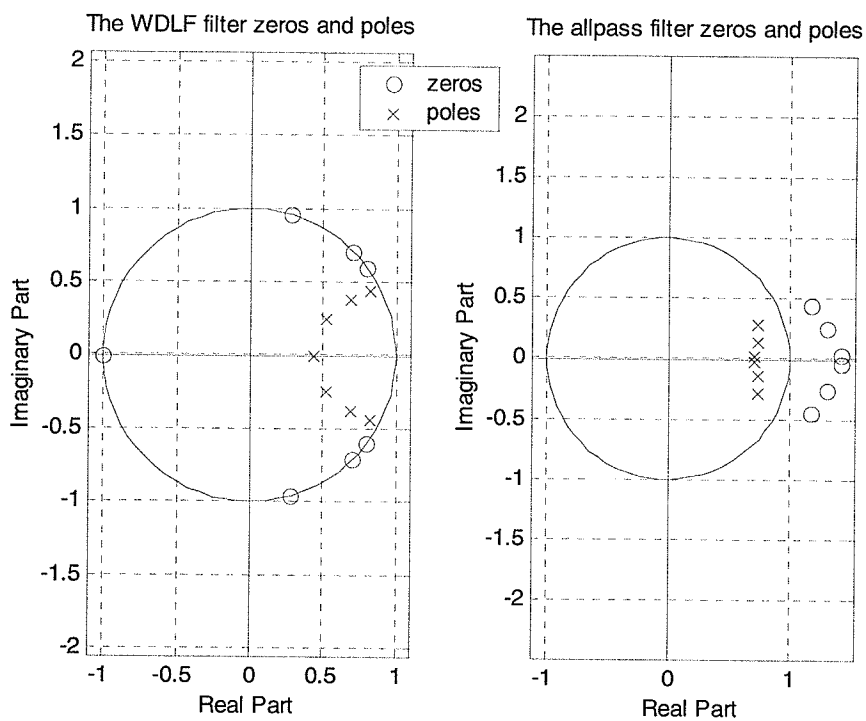


Fig. 7-30 The zeros and poles of 6-order WDLF and 7-order allpass compensation filter

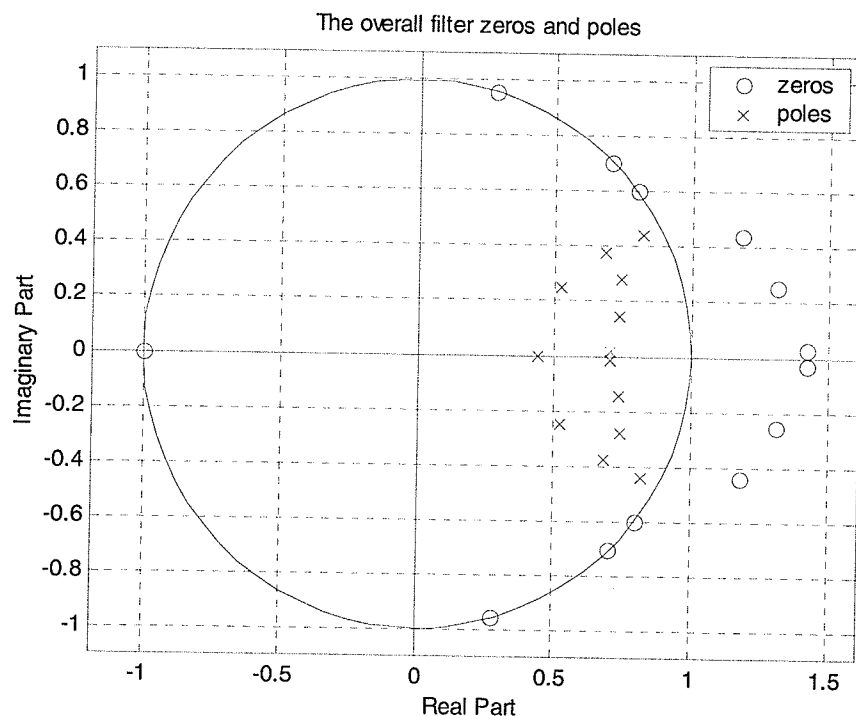


Fig. 7-31 The zeros and poles of the overall filter

Example 5: The example specification is taken as the channel filter in a mobile communications system, described by Ericsson Mobile Communications [53]. The example parameters are as follows: The filter order is $N=5$, the filter specification is given in Fig. 7-32, where the different levels correspond to a passband ripple of 0.3 and 0.9 dB and a stopband attenuation of 0.9, 20 and 49 dB.

We first use the modified pole-zero placement to design the filter equiripple both in the passband and in the stopband meeting the specification. Then, the 6-order allpass compensation is applied to equalize the designed filter's group delay. We apply the nonlinear least square algorithm to design the allpass compensation filter.

The adaptor coefficients of the filter are given in Table 7. The gain in dB (the negative of the attenuation) is plotted, for ease of comparison with [53], in Fig. 7-32. In Fig. 7-33, the group delay in the whole region and in the passband is plotted. The solid curve in Fig. 7-

33 corresponds to the 5-order WDLF in cascade with a 6-order allpass filter, and the dashed curve corresponds to the 5-order WDLF. Fig. 7-34 illustrates the group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband. The zeros and poles of the WDLF and allpass filter are shown in Fig. 7-35. The zeros and poles of the 11-order the overall filter are shown in Fig. 7-36.

Table 7 Adaptor coefficients of the filter in example 5

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.63454120673698	1	-0.88645063375293	0.58669689908906
A_2			2	-0.58398212193402	0.76228523705932
Allpass			3	-0.66732573660396	0.96948742329110
			4	-0.68258235903646	0.89356622518483
			5	-0.71838487087511	0.73180330642123

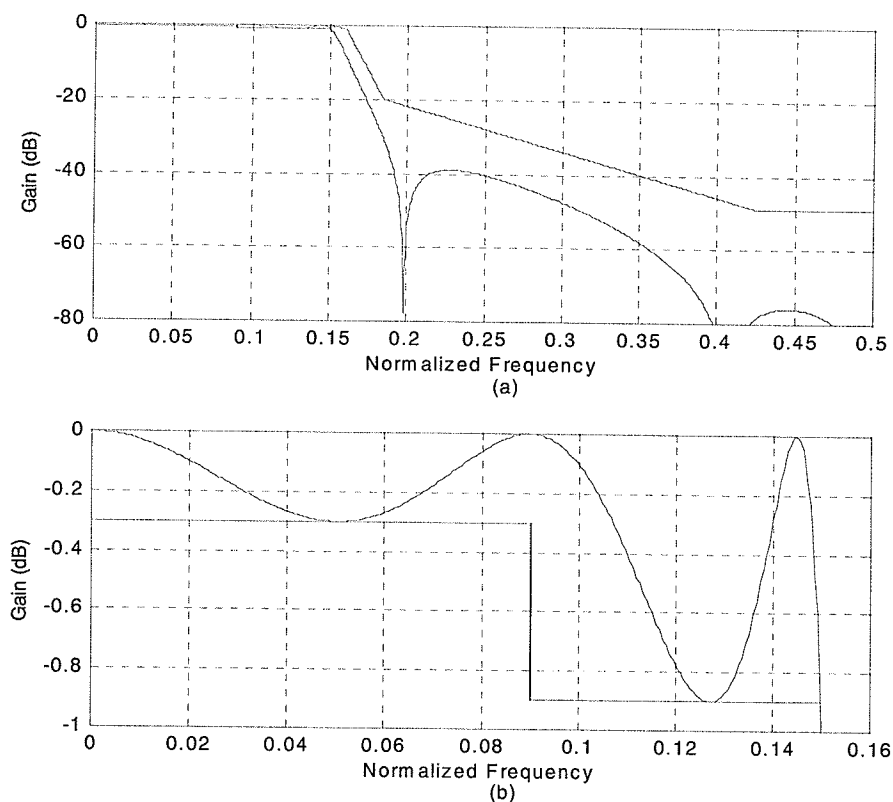


Fig. 7-32 The gain in dB of the 11-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

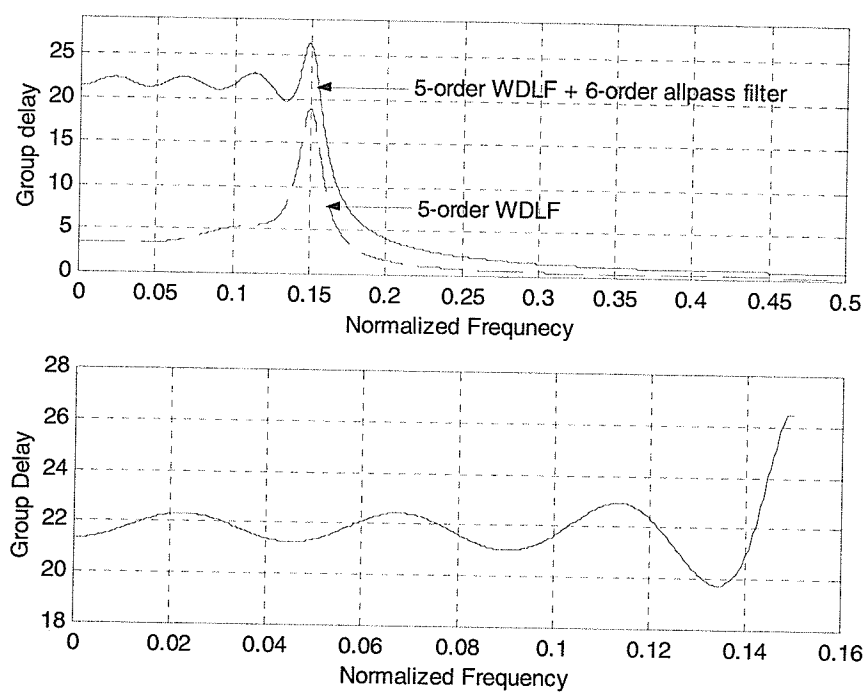


Fig. 7-33 The group delay of the 11-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

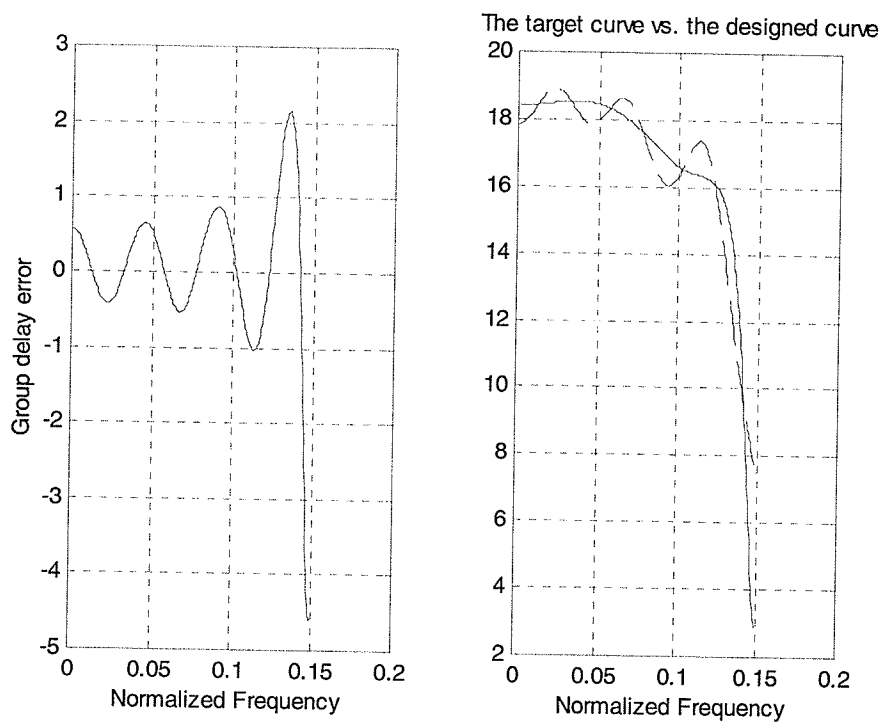


Fig. 7-34 The group delay error in the passband and the target curve vs. the designed allpass group delay curve in the passband

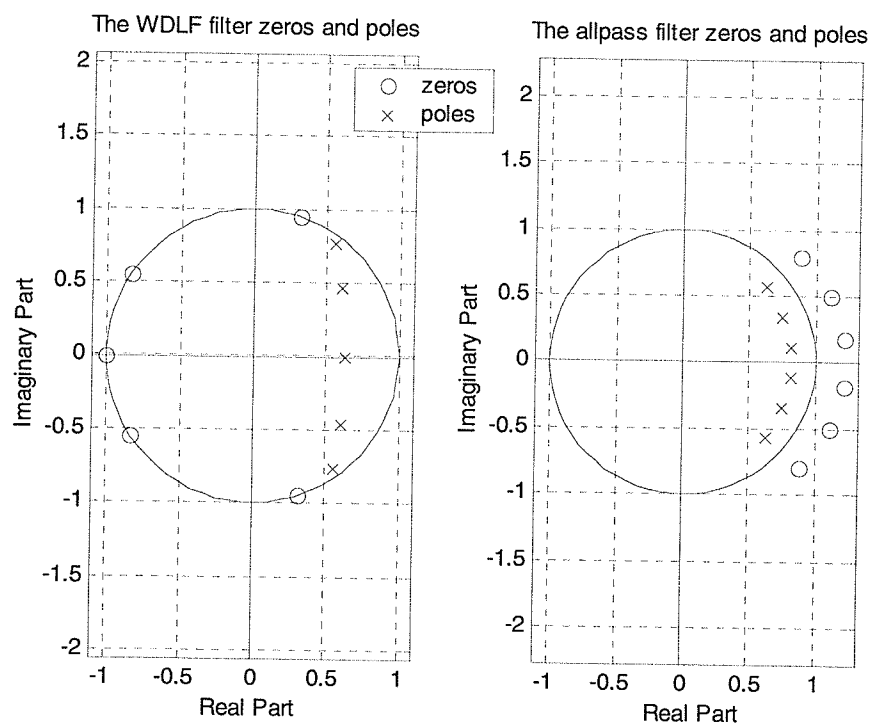


Fig. 7-35 The zeros and poles of 5-order WDLF and 6-order allpass compensation filter

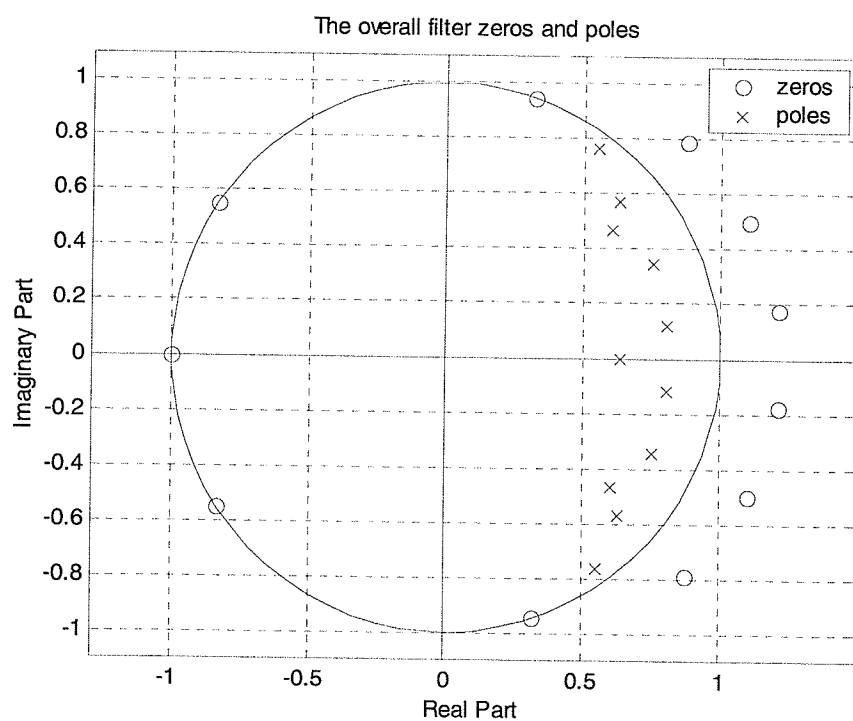


Fig. 7-36 The zeros and poles of the overall filter

7.3 Examples Using the Sum of Two Wave Digital Allpass Filters

In this section, an approximately linear phase wave digital lattice filter will be designed as a parallel connection of two wave digital allpass filters. The modified Remez algorithm, described in Chapter 6, is used.

Example 6: The example is the same as Example 1 in Section 7.1. In this example, the allpass A_1 is realized as a pure delay.

The adaptor coefficients of the allpass filter A_2 are given in Table 8. The structure of the 7-order WDLF is shown in Fig. 7-37. Fig. 7-38 illustrates the zeros of f and g in z -domain, respectively. The attenuation, group delay in the passband, phase in the passband and phase error are shown in Figs. 7-39 – 7-42.

Table 8 Adaptor coefficients of the filter in Example 6

Coefficients Allpass	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_2			1	-0.28579638386551	-0.75844304300228
			2	-0.29602937516233	-0.25695733934968
			3	-0.33982686066616	0.41572214727583
			4	-0.81523195376235	0.91547731026399

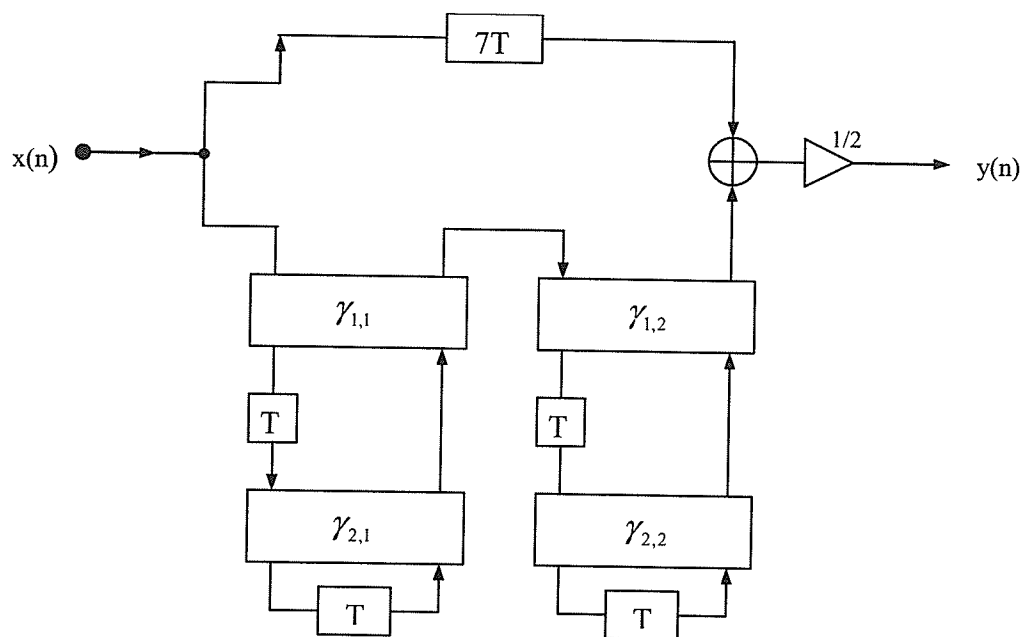


Fig. 7-37 7-order approximately linear phase WDLF structure with one branch pure delay

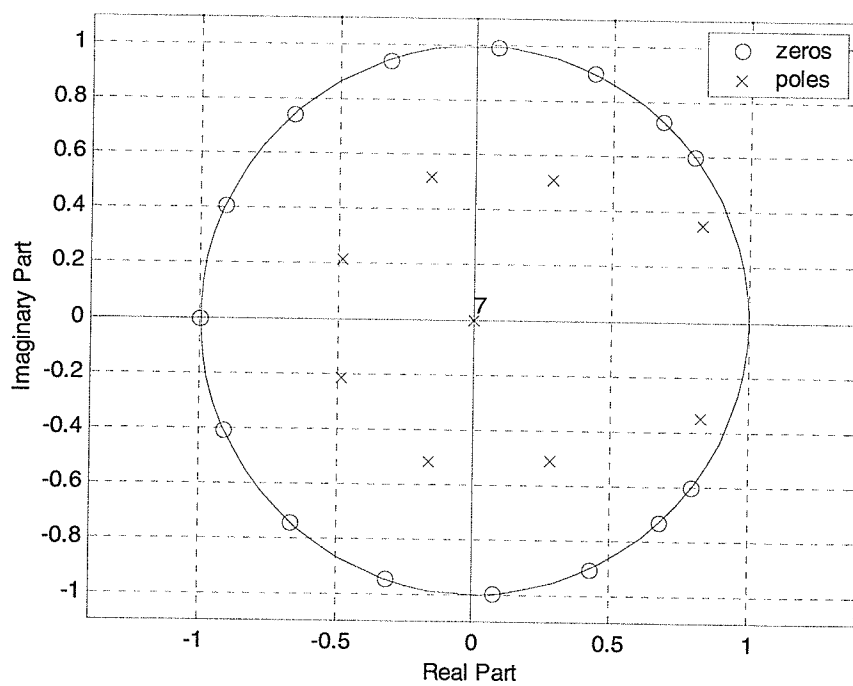


Fig. 7-38 The zeros and poles of the WDLF filter

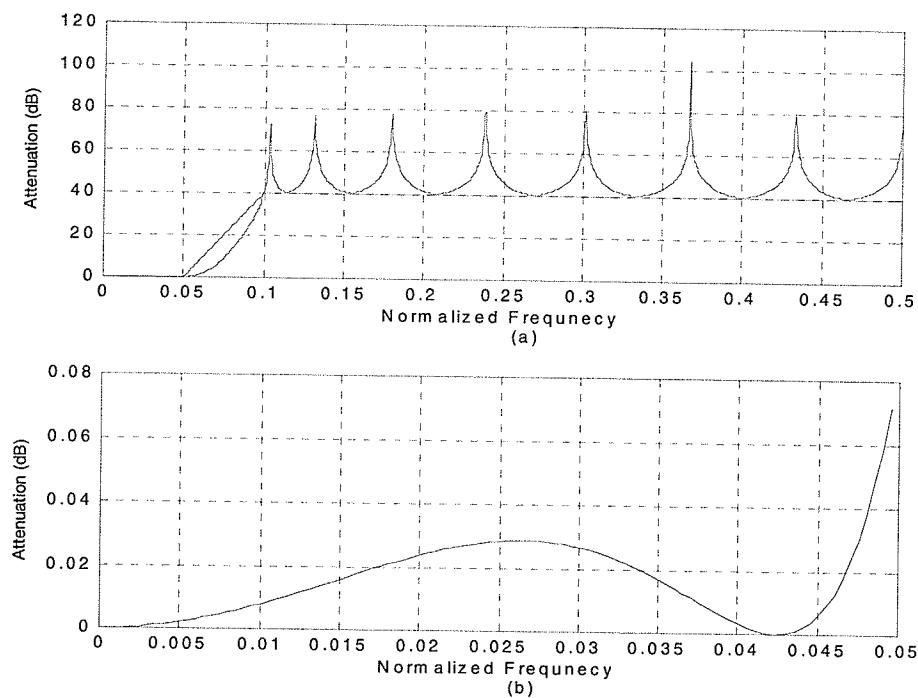


Fig. 7-39 The attenuation of the 15-order approximately linear phase WDLF
(a) in the whole region, (b) in the passband

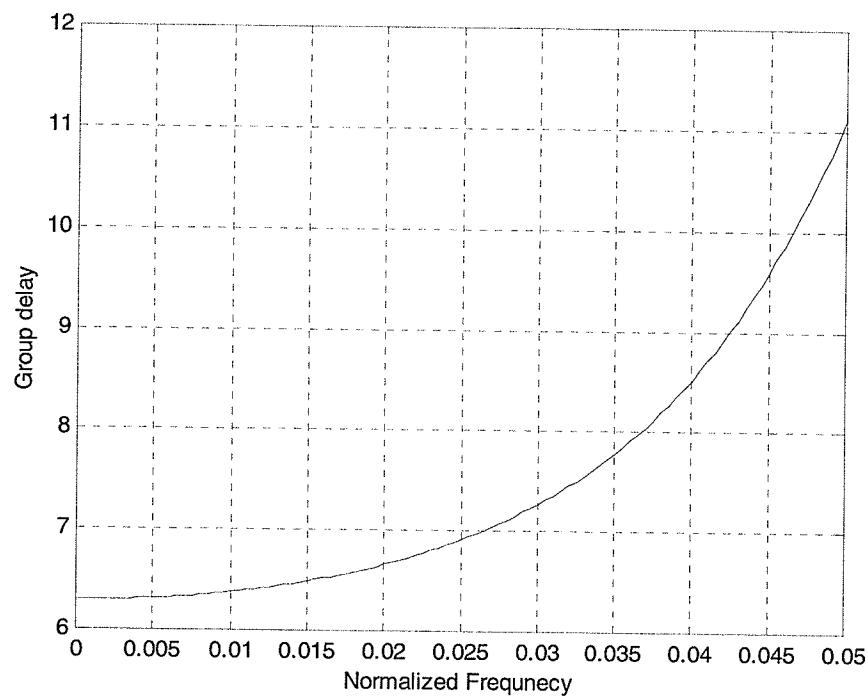


Fig. 7-40 The group delay in passband of the 15-order approximately linear phase WDLF

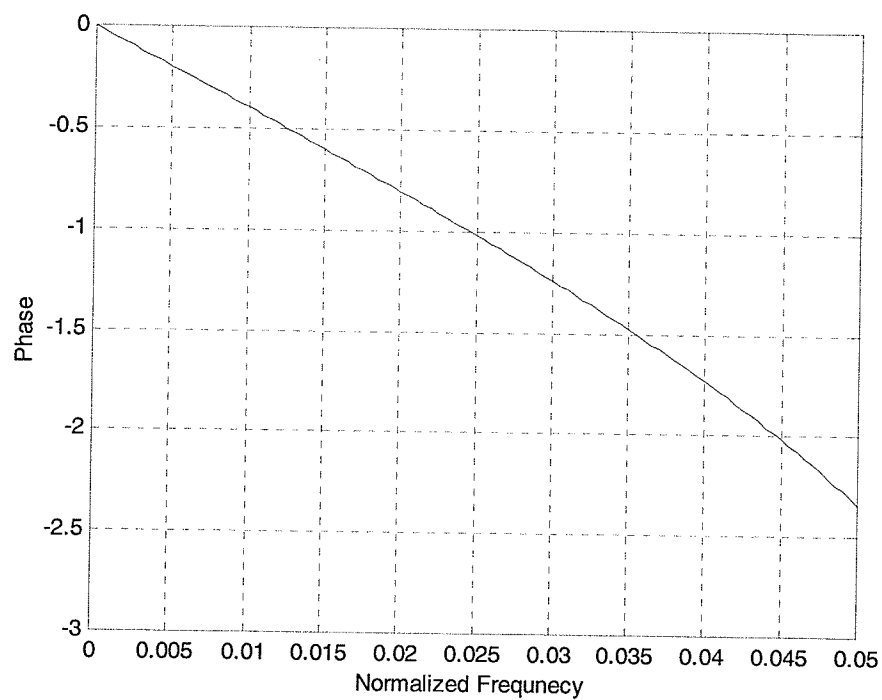


Fig. 7-41 The phase in passband of the 15-order approximately linear phase WDLF

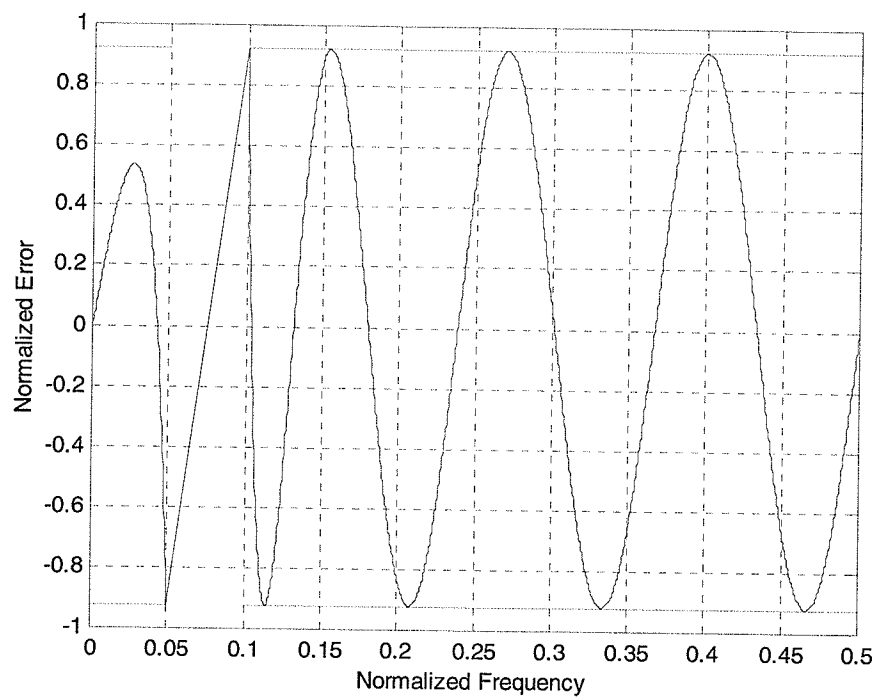


Fig. 7-42 The phase error of the 15-order approximately linear phase WDLF

Example 7: In this example, we design a lowpass wave digital lattice filter. The specification given is that the normalized passband edge is at 0.1 and the normalized stopband edge is at 0.2. The maximum allowable peak-to-peak passband ripple is 0.001dB and the minimum stopband attenuation is 40 dB. The order of the filter is 25.

The adaptor coefficients of the filter are given in Table 9. Fig. 7-43 illustrates the zeros of f and g in the z -domain. The attenuation, group delay in the passband, phase in the passband and the phase error are shown in Figs.7-44 – 7-47.

Table 9 Adaptor coefficients of the filter in example 7

Coefficients Allpass					
	j	$\gamma_{0,j}$	i	$\gamma_{1,i}$	$\gamma_{2,i}$
A_1	1	0.00123020771562			
	2	-0.04210915086477			
	3	-0.05454887070244			
	4	-0.00862029504750			
	5	0.06607645579931			
	6	0.09310084575206			
	7	0.02260203760786			
	8	-0.10688174674955			
	9	-0.17986240805155			
	10	-0.06943850348418			
	11	0.26039309518054			
	12	1.00000000000000			
A_2	13	0.02360046847789			
	14	0.03803928497338			
	15	0.02446757455620			
	16	-0.02055915633919			
	17	-0.06363440907165			
	18	-0.06095624590675			
	19	-0.00372961445727			
	20	0.07503690533181			
	21	0.12727781642087			
	22	0.10664296008590			
	23	0.00719892221732			
	24	-0.43075638404067			
	25	1.00000000000000			

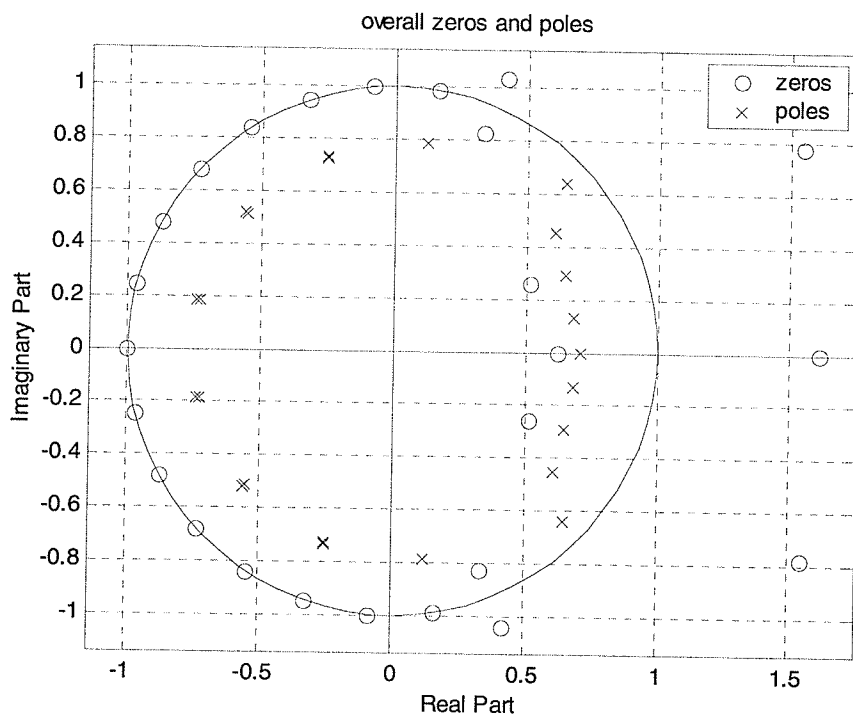


Fig. 7-43 The zeros and poles of the WDLF filter

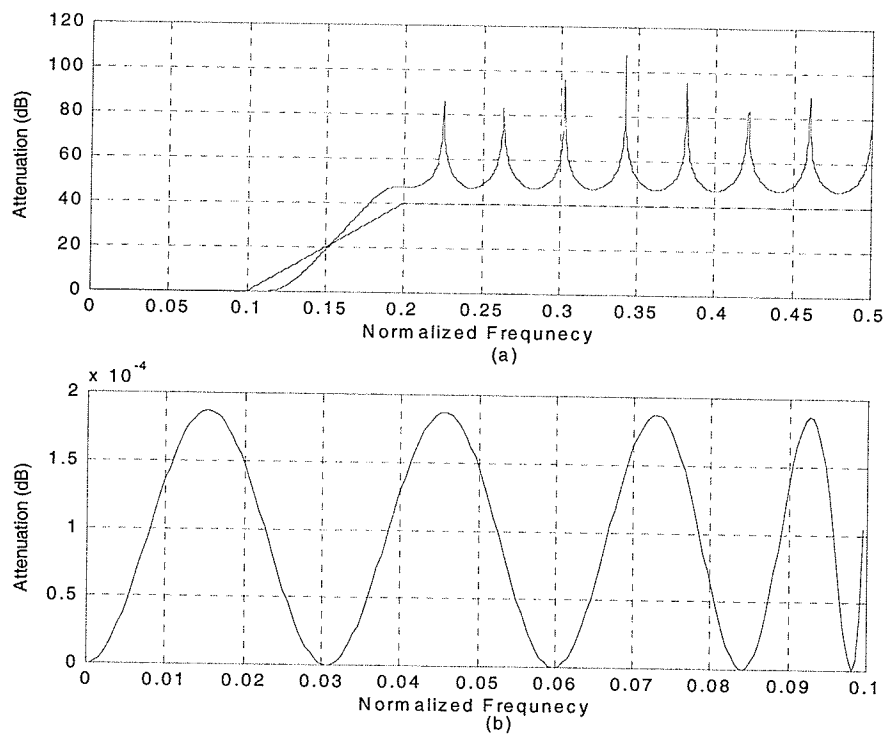


Fig. 7-44 The attenuation of the 25-order approximately linear phase WDLF filter
 (a) in the whole region, (b) in the passband

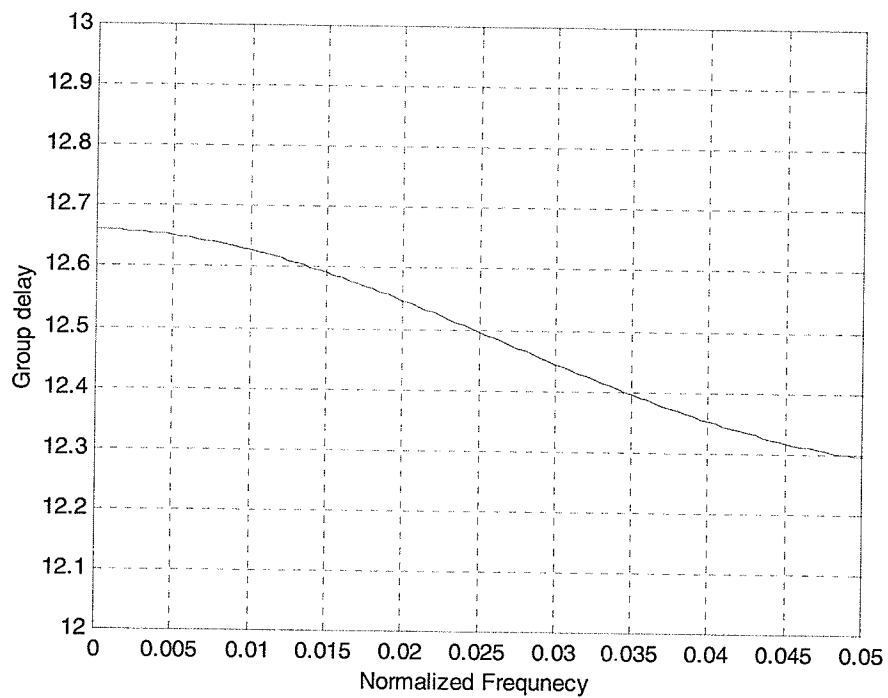


Fig. 7-45 The group delay in passband of the 25-order approximately linear phase WDLF filter

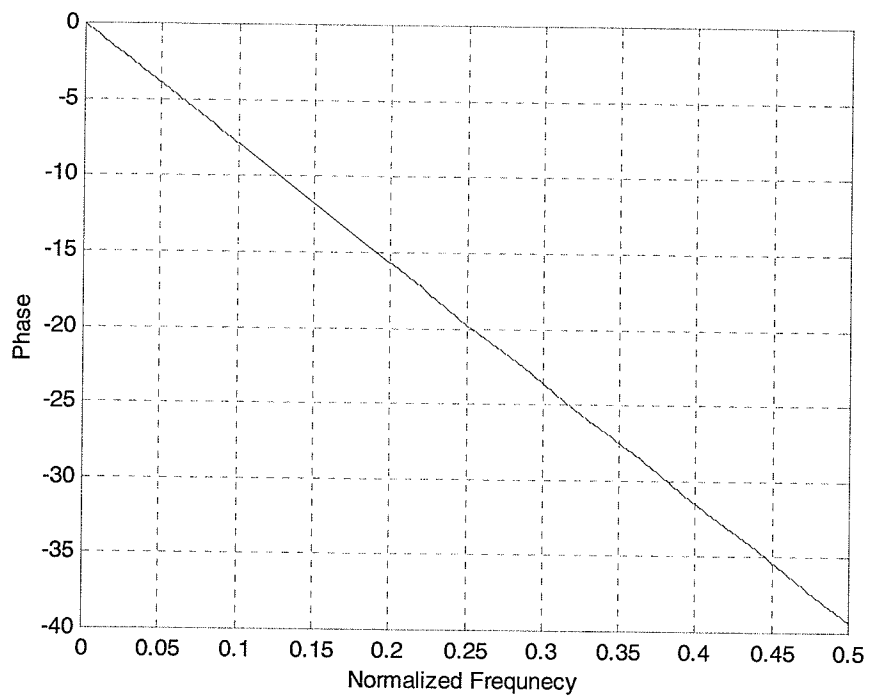


Fig. 7-46 The phase in passband of the 25-order approximately linear phase WDLF filter

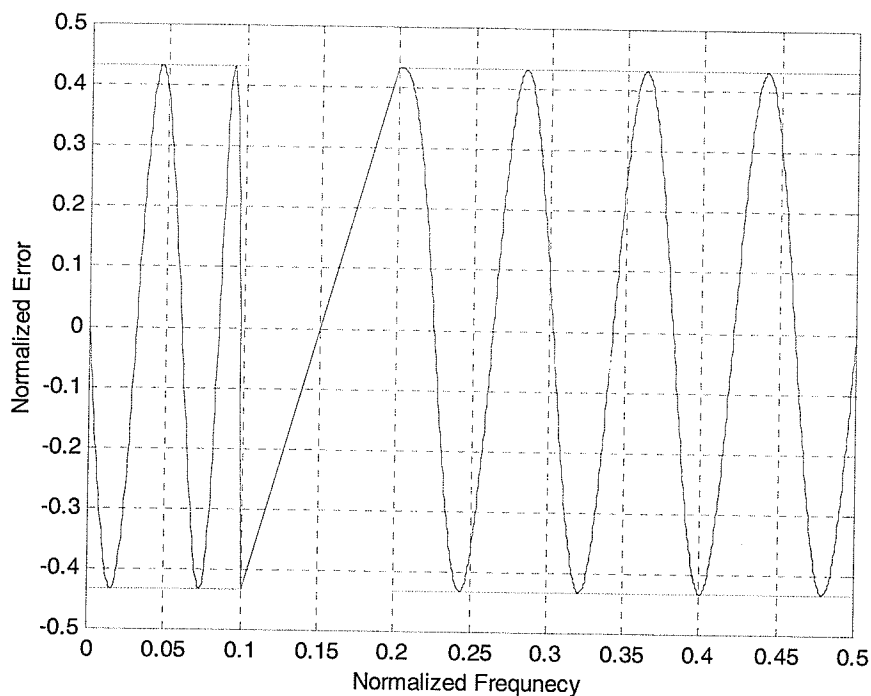


Fig. 7-47 The phase error of the 25-order approximately linear phase WDLF filter

7.4 Summary

In Fig. 7-48 and Fig. 7-49, the group delay and phase in the passband obtained by three approximate approaches are illustrated.

In Fig. 7-48, the group delay in the passband using an allpass compensation filter can be made higher (32 samples and 34 samples vs. 6 samples using pole-zero placement and 7 samples using the paralleling of two allpass filters). In addition, the group delay equalization can obtain a much flatter group delay in the passband, which means the phase in the passband is much more linear.

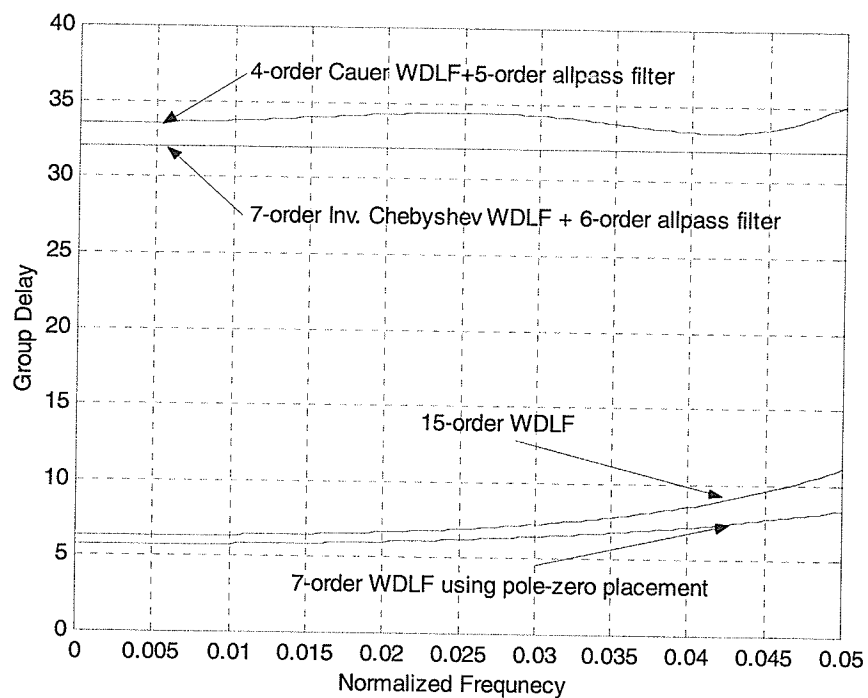


Fig. 7-48 The group delay in the passband obtained by three approximate methods

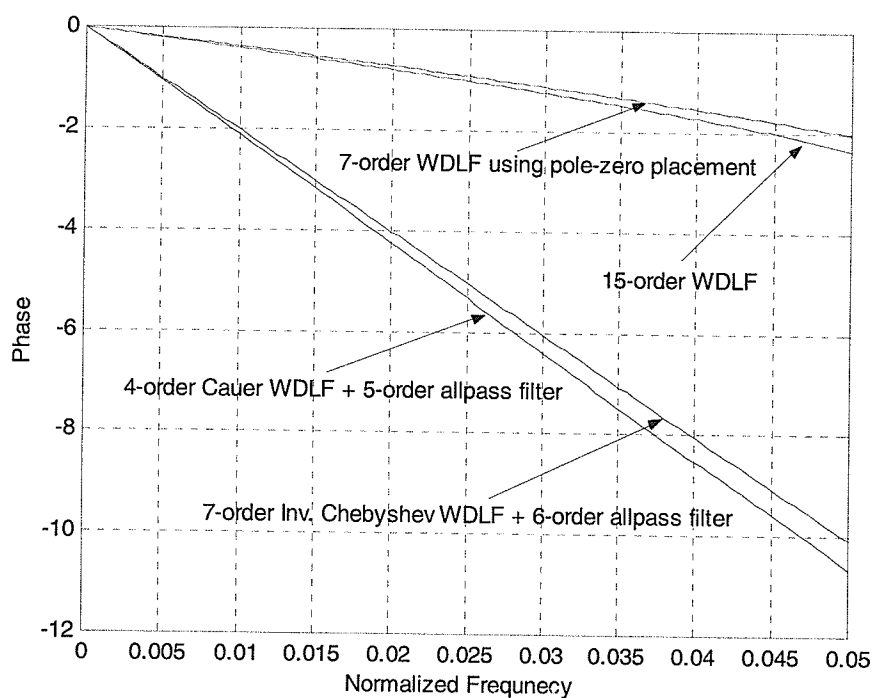


Fig. 7-49 The phase in the passband obtained by three approximate methods

In the pole-zero placement, phase correction is performed by moving a quad of zeros of h . It takes a lot of computing time to obtain an approximately linear phase. In addition, the pole-zero placement can be used to design filters based on various specifications.

In the case of the cascaded connection of one allpass compensation filter, the group delay equalization using nonlinear least squares can obtain better results. Group delay correction is performed where the additional group delay is used as a free parameter. The choice is between the compensator degree, group delay ripple and maximally flat group delay bandwidth. In addition, the poles and zeros of the allpass compensation filter are almost uniformly distributed in the passband region.

In the case of the parallel connection of two allpass filters, the approximate phases are obtained by increasing the filter order from 7 to 15. The modified Remez algorithm can make the unit phase error smaller, in both the passband and the stopband, and keep the group delay lower. The design procedure relies on the pre-construction of one of the allpass subfilters to have exact linear phase and the other allpass subfilter is constructed to satisfy the phase difference between the specifications and the exact linear phase. The hidden relationship between the attenuation and the phase has been employed for approximating the attenuation and phase simultaneously.

Chapter 8

Conclusions

This thesis describes three methods for the design of IIR digital filters with approximately flat group delay in the passband. These design methods can be applied to design filters realizable in a conventional cascade form or as a parallel connection of two allpass filters (wave digital lattice filters). The design problem is formulated as a complex approximation problem in the frequency domain. The first proposed method in pole-zero placement minimizes the group delay approximation error by the adjustment of the pole and zero positions. A quad of zeros corresponding to the passband are necessary to obtain a good group delay approximation. In addition, equiripple in the passband and in the stopband is considered. In the next proposed algorithm, the resulting constrained nonlinear least squares approximation problem is solved by a modified Gauss-Newton algorithm. Poles and zeros of the allpass compensation at angles corresponding to the passband are necessary to obtain a good group delay approximation. An important advantage of this design algorithm is the fact that no sophisticated initial solution is necessary to obtain a good final solution. Finally, the modified Remez algorithm is also realized for a digital filter consisting of two allpass subfilters in parallel. Convergence and optimality are observed in the calculated examples.

The methods developed in this thesis are very flexible and they can be modified by small changes for the different approximation tasks. For future research, the direction might be the following:

- The extension of the optimization of the different orders for numerator and denominator polynomials.
- Optimization of new structures of the IIR filter with maximally flat group delay.
- Optimization of the IIR filter banks for audio and image processing.
- Optimization of IIR filters for software radio.

REFERENCES

1. A. Fettweis, H. Levin, A. Sedlmeyer, "Wave digital lattice filters", *Int. j. cir. thero. appl.* vol. 2, pp.203-11, June, 1974.
2. A. Fettweis, "Wave digital filters: Theory and Practice", *Proc. IEEE*, vol. 74, pp. 270-327, Feb. 1986.
3. L. Gazsi, "Explicit formulas for lattice wave digital filters", *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 68-88, Jan. 1985.
4. S. K. Mitra and K. Hirano, "Digital all-pass networks", *IEEE Trans. Circuits Syst.*, vol. CSA-21, pp.688-700, Sep. 1974.
5. P. P. Vaidyanathan and S. K. Mitra, "Low passband sensitivity digital filters: A generalized viewpoint and synthesis procedures", *Proc. IEEE*, pp. 404-423, Apr. 1984
6. P. P. Vaidyanathan, S. K. Mitra, Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters", *IEEE Trans. Acoust., Speech & Signal Process.*, vol. ASSP-34, No.2, pp350-361, Apr. 1986.
7. P. A. Regalia, S. K. Mitra, P. P. Vaidyanathan, "The digital all-pass filter: a versatile signal processing building block", *Proc.of IEEE*, vol 76, No.1, pp. 19-36 Jan. 1988.
8. N. Willson and H. J. Orchard, "Insights into digital filters made as the sum of two allpass functions", *IEEE Trans. Circuits Syst. I*, vol. 42, No. 3, pp. 129-137, Mar. 1995.

9. B. R. Smith and G. C. Temes, "An iterative approximation procedure for automatic filter synthesis", *IEEE Trans.*, CT-2, pp. 107-112, 1965.
10. R. W. Daniels, "Approximation methods for electronic filter design with Applications to passive, active, and digital networks", *McGraw-Hill, New York*, 1974.
11. R. Unbehauen, "Netzwerk- und filter- synthesis", R.Oldenbourg Verlag Muenchen Wien, 1993.
12. J. G. Proakis and D. G. Manolakis, "Digital signal processing principles, algorithms, and applications", 2nd *Macmillan Publishing Comp.* 1992.
13. J. Teng, J. K. Fidler and L. Zheng, "Pole placement algorithm for equiripple bandstop filter design", *IEE Proc. Circuit Syst.*, Vol. 141, No. 4, pp. 325-327 Aug. 1994.
14. J. P. Thiran, "Recursive digital filters with maximally flat group delay", *IEEE Trans. On circuit theory*, Vol. CT-18, No. 6, pp. 659-664, Nov. 1971.
15. J. P. Thiran, "Equal-Ripple delay recursive digital filters", *IEEE Trans. On circuit theory*, Vol. CT-18, No. 6, pp. 664-669, Nov. 1971.
16. F. Leeb, "Lattice wave digital filters with simultaneous conditions on amplitude and phase", *Proc. IEEE Int. Conf. Acoust., Speech & Signal Proc.* pp. 1645-1648, May 1991.
17. I. Kunold, "Linear phase realization of wave digital lattice filters", *Proc. ISCAS* pp.1455-1458 1988.

18. T. Q. Nguyen, T. I. Laakso and R. D. Koilpillai, "Eigenfilter approach for the design of allpass filters approximating a given phase response", *IEEE Trans. on Signal Proc.* Vol. 42, No. 9 pp.2257-2263, Sep. 1994.
19. C. K. Chen and J. H Lee, "Design of digital all-pass filters using a weight least squares approach", *IEEE Trans. on Circuits and Systems II*, vol. 41, No. 5, pp. 346-351, May 1994.
20. M. Lang and T. I. Laakso, "Simple and robust method for the design of allpass filters using least-squares phase error criterion", *IEEE Trans. on Circuits and Systems II*, vol. 41, No. 1, pp. 40-48, Jan. 1994.
21. K. Matsuyama, M. Okuda and M. Ikehara, "Approximation of group delay response using weight least square method", *Proc. ISCAS* pp. 197-200, 1996.
22. M. Lang, "Allpass filter design and applications", *IEEE Trans on Signal Processing* vol. 46, No. 9, pp. 2505-2514, Sep. 1998.
23. J. L. Sullivan and J. W. Adams, "PCLS IIR digital filters with simultaneous frequency response magnitude and group delay specifications", *IEEE Trans on Signal Processing* vol. 46, No. 11, pp. 2853-2861, Nov. 1998.
24. J. Yli-Kaakinen and T. Saramaki, "Design of low-sensitivity and low-noise recursive digital filters using a cascade of low-order wave lattice filters", *IEEE Trans. Circuits Syst.II*, vol. 46, No.7, pp.906-914, July 1999.
25. J. Yli-Kaakinen and T. Saramaki, "An algorithm for the design of multiplierless approximately linear-phase lattice wave digital filters", *ISCAS 2000-IEEE v.II* pp.77-80, 2000.

26. C. Xiao, J. C. Olivier and P. Agathoklis, "Design of linear phase IIR filters via weighted least-squares approximation", *Proc. ISCAS 2001-IEEE* pp. 3817-3820 2001.
27. K. C. Haddad, Y. Yang, N. P. Galatsanos and H. Stark, "Allpass filter design using projection-based method under group delay constraints", *Proc. ISCAS 2001-IEEE* pp. 3833-3836 2001.
28. M. Renfors and T. Saramaki, "A class of approximately linear phase digital filters composed of allpass subfilters", *IEEE intern. Symp. on Circuits and Systems*, pp.678-681, 1986.
29. M. Renfors and T. Saramaki, "Recursive N th-band digital filters - Part I: Design and Properties; Part II: Design of Multistage Decimators and Interpolators", *IEEE Trans. Circuits Syst.*, vol. CSA-34, pp.24-51, Jan. 1987.
30. M. Rnefors and E. Zigouris, "Signal processor implementation of digital all pass filters", *IEEE Trans. Acoust., Speech & Signal Process.*, vol. ASSP-36, No.5, pp714-729, May 1988.
31. M. Ikkehara, M. Funaishi and H. Kuroda, "Design of complex all-pass networks using Remez algorithm", *IEEE Trans. on Circuits and Systems II*, vol.39, No. 8, pp.549-555, Aug. 1992.
32. M. Ikkehara, M. Funaishi and H. Kuroda, "Design of IIR digital filters using all-pass networks", *IEEE Trans. on Circuits and Systems II*, vol. 41 No. 3, pp. 231-235, Mar. 1994.

33. S. Lawson, "Direct approach to design of PCAS filters with combined gain and phase specification", *IEE Proc.-Vis. Image Signal Process.*, vol 141, No. 3, pp. 161-167, June 1994.
34. G. M. Raz and B. D. Van Veen, "Blind equalization and identification of nonlinear and IIR systems — A least squares approach", *IEEE Trans on Signal Processing* vol. 48 No. 1 pp. 192-200, Jan. 2000.
35. T. Henk, "The generation of arbitrary-phase polynomials by recurrence formulae", *Int. j. cir. thero. appl.* vol. 9, pp.461-478, 1981.
36. G. O. Matens, Private communications.
37. H. Johansson and L. Wanhammar, "Design of linear-phase lattice wave digital filters", http://www.es.isy.liu.se/publications/papers_and_reports/index.html. 1996.
38. H. Johansson and L. Wanhammar, " High-Speed recursive filter structures composed of identical All-Pass subfilters for interpolation, decimation, and QMF banks with perfect magnitude reconstruction", *IEEE Trans. Circuits Syst.II*, vol. 46, No. 1, pp. 16-28, Jan. 1999.
39. H. Johansson and L. Wanhammar, "Wave digital filter structures for High Speed Narrow-Band and Wide-Band filtering", *IEEE Trans. Circuits Syst.II*, vol. 46, No. 6, pp. 726-741, June 1999.
40. H. Johansson and L. Wanhammar, "Filter structures composed of all-pass and FIR filters for interpolation and decimation by a factor of two", *IEEE Trans. Circuits Syst.II*, vol. 46, No. 7, pp. 896-905, July. 1999.

41. H. Johansson and T. Sanamaki, "A class of complementary IIR filters", *Proc. IEEE ISCAS*, v. III pp. 299-302, 1999.
42. M. A. Zahhad, M. Yaseen, T. Henk, "Arbitrary amplitude and linear phase approximations for non-prototype ladder and lattice wave digital filters", *Int. j. cir. thero. appl.* vol. 24, pp.605-620, 1996.
43. J. A. Nossek and H. D. Schwarz, "Wave digital filters with applications in communication systems", *Proc. IEEE ISCAS*, pp. 845-848, May 1983.
44. S. C. Pei and C. C. Tseng, "IIR multiple notch filter design based on allpass filter", *IEEE Trans. Circuits Syst.II*, vol. 44, No. 2, pp. 133-136, Feb. 1997.
45. P. P. Vaidyanathan, P.A. Regalia, S. K. Mitra, "Design of doubly complementary IIR digital filters using a single complex allpass filter, with multirate applications", *IEEE Trans. Circuits Syst.*, vol. CAS-34, No. 4, pp. 378-389, Apr. 1987.
46. A. Fettweis, T. Leickel, M. Bolle, U. Sauvagerd, "Realization of Filterbanks by Means of Wave Digital Filters", *Proc. IEEE ISCAS*, New Orleans, LA, May. 1990.
47. T. Leickel and A. Fettweis, "Efficient digital-signal-processor realization of multirate filter banks using wave digital filters", *Proc. IEEE ISCAS*, San Diego, CA, May 1992.
48. K. Lu, M. S. Anderson and S. Summerfield, "Approximately Linear-Phase Design of Allpass-Based QMF Banks", *IEE Colloquium Digest*, 1996/238, 16th Saraga Colloquium on Digital and Analogue Filters, pp. 11/1-11/6, 1996.

49. K. Lu, M. S. Anderson and S. Summerfield, "Design of Approximately Linear-Phase Allpass-Based QMF Banks", *Proc. 1st Int. Symp. Digital Signal Processing*, London, pp. 56-61, 1996.
50. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ Prentice-Hall, 1993.
51. S. K. Mitra, "Digital signal processing a computer-based approach", *McGraw-Hill*, New York, 1998.
52. B.W. Suter, "Multirate and wavelet signal processing", San Diego, CA, *Academic Press*, 1998.
53. P. Persson, S. Nordebo, and I. Claesson, "A multimode mean field annealing technique to design recursive digital filters", *IEEE Trans. Circuits Syst.*, vol. 48, No. 12, pp.1151-1154, Dec. 2001.
54. V. Pereyra, "Iterative methods for solving nonlinear least squares problems", *SIAM J. Numer. Anal.* vol. 4, No. 1, pp.27-36, 1967.
55. P. E. Gill and W. Murray, "Algorithms for the solution of the nonlinear least-square problem", *SIAM J. Numer. Anal.* vol. 15, No. 5, pp.977-992, Oct. 1978.
56. M. R. Jarmasz, "A simplified synthesis of lossless two-port wave digital and analog filters," *Ph.d. Thesis*, U. of Manitoba, Winnipeg, MB. 1990.
57. J. Wang, "Design of even-order complex wave digital filters", *M. Sc. Thesis*, U. of Manitoba, Winnipeg, MB. 1996.
58. Y. Zhang, "Design and realization of FIR and birciprocal wave digital filters", *M. Sc. Thesis*, U. of Manitoba, Winnipeg, MB. 2000.

Appendix

MATLAB program

```
% part 1
% using pole-zero placement
%
clear all;

format long
global j
j=sqrt(-1);
tol=1e-6;
%
% ***** Specification *****
%
Ap=0.1; As=40; fc=0.05; fs=0.1; F=1; n=7;
tola=1e-6; tol=1e-3;
wc=2*pi*fc/F;
ws=2*pi*fs/F;
delta_c=1-10^(-Ap/20);
delta_s=10^(-As/20);
epison=1e-3;
w_init=0;
w_final=pi;
NoOfPts=1000;
App=Ap-tola;
Ass=As+tol;

% Step1
% select initial extremal N points
% initial h and f in z-domain
% select n points for h in passband and n points for f in stopband
[w_h_pts,w_f_pts,h_init,f_init]=SelectInitPtsHF(n,wc,ws);
Kh=find_Kh_from_fz_hz(f_init,h_init,Ap,n,wc); % find Kh
hk=[Kh;h_init(2:length(h_init))]; % obtain h with new Kh
% calculate g from f and h
[g1z,g2z,gz]=Obtain_g1_g2_from_f_h(f_init,hk,n);
% polar
fz_Polar=rectAngularToPolar(f_init);
hz_Polar=rectAngularToPolar(hk);
gz_Polar=rectAngularToPolar(gz);
%
K=f_init(1)/gz(1);
arrayofzeros=f_init(2:length(f_init));
arrayofpoles=gz(2:length(gz));
%
fz_init=f_init;
gz_init=gz;
hz_init=hk;

[Atten,Mag,Phase,Delay,w]=FreqResponses_z(K,arrayofzeros,arrayofpoles,w_init,w_final,NoOfPts);
[Atten_wc,Mag_wc,Phase_wc,Delay_wc]=FreqResponses_z_w(K,arrayofzeros,arrayofpoles,wc);
[Atten_ws,Mag_ws,Phase_ws,Delay_ws]=FreqResponses_z_w(K,arrayofzeros,arrayofpoles,ws);
% Atten.
atten_c=Atten(w<=wc);
```

```

atten_s=Atten(w>=ws);
% find local min. points in the stopband
[Att_max_c,wh_max_c,Att_min_c,wh_min_c]=Find_LocalMin_Err_atten_PB(atten_c,w,wc
,length(wh_pts),arrayofzeros,arrayofpoles,K);
% find local min. points in the stopband
[Att_min_s,wf_min_s,Att_max_s,wf_max_s]=Find_LocalMin_Err_atten_SB(atten_s,w,ws
,length(wf_pts),arrayofzeros,arrayofpoles,K);
% w
wmax_c=[wh_max_c,wc];
wmin_s=[ws,wf_min_s];
wmin_c=wh_min_c;
wmax_s=wf_max_s;
% delta_atten.
delta_atten_c=[App-Att_max_c,App-Atten_wc]/App; % passband
delta_atten_s=[Atten_ws-Ass,Att_min_s-Ass]/Ass; % stopband
% modified differential Attenuation - right side matrix
delta_atten=[delta_atten_c,delta_atten_s]';

% Matrix A
A=MatrixofDevAtten(Kh,wh_min_c,wf_max_s,wmax_c,wmin_s,f_init,gz,hk,wc,ws);
%
delta_w=A\delta_atten; % calculate delta_w
delta_w(length(delta_w),1)=1;
Delta_w=delta_w';
%
% find new parameters from old parameters and diff.
[Kh_new,wh_new,wf_new,hz,fz]=ObtainNew_h_f_Kh(Kh,w_h_pts,w_f_pts,Delta_w,wc,ws,
delta_atten); [g1z,g2z,gz]=Obtain_g1_g2_from_f_h(fz,hz,n);
fz_Polar=rectAngularToPolar(fz);
hz_Polar=rectAngularToPolar(hz);
gz_Polar=rectAngularToPolar(gz);
K=fz(1)/gz(1);
arrayofzeros=fz(2:length(fz));
arrayofpoles=gz(2:length(gz));
h_roots=hz(2:length(hz));

[Atten,Mag,Phase,Delay,w]=FreqResponses_z(K,arrayofzeros,arrayofpoles,w_init,w_
final,NoOfPts);
[Atten_wc,Mag_wc,Phase_wc,Delay_wc]=FreqResponses_z_w(K,arrayofzeros,arrayofpol
es,wc);
[Atten_ws,Mag_ws,Phase_ws,Delay_ws]=FreqResponses_z_w(K,arrayofzeros,arrayofpol
es,ws);
%
% Atten.
atten_c=Atten(w<=wc);
atten_s=Atten(w>=ws);
% find local min. points in the stopband
[Att_max_c,wh_max_c,Att_min_c,wh_min_c]=Find_LocalMin_Err_atten_PB(atten_c,w,wc
,length(wh_new),arrayofzeros,arrayofpoles,K);
% find local min. points in the stopband
[Att_min_s,wf_min_s,Att_max_s,wf_max_s]=Find_LocalMin_Err_atten_SB(atten_s,w,ws
,length(wf_new),arrayofzeros,arrayofpoles,K); % w
wmax_c=[wh_max_c,wc]; % extremal points in passband
wmin_s=[ws,wf_min_s]; % extremal points in stopband
wmin_c=wh_min_c; % roots in passband
wmax_s=wf_max_s; % roots in stopband
% delta_atten.
delta_atten_c=[App-Att_max_c,App-Atten_wc]/App; % passband
delta_atten_s=[Atten_ws-Ass,Att_min_s-Ass]/Ass; % stopband
delta_atten=[delta_atten_c,delta_atten_s]'; % modified differential
attenuation - right side matrix
%
count=1;

```

```

while max(delta_atten_c)-min(delta_atten_c)>0.001 | max(delta_atten_s)-
min(delta_atten_s)>0.001
    Kh_old=Kh_new;
    wh_old=wh_new;
    wf_old=wf_new;
    hz_old=hz;
    fz_old=fz;
    gz_old=gz;
    % Matrix A

A=MatrixofDevAtten(Kh_old,wh_min_c,wf_max_s,wmax_c,wmin_s,fz_old,gz_old,hz_old,
wc,ws);
    delta_w=A\delta_atten; % delta_w
    delta_w(length(delta_w),1)=1;
    Delta_w=delta_w';
    %

[Kh_new,wh_new,wf_new,hz,fz]=ObtainNew_h_f_Kh(Kh_old,wh_old,wf_old,Delta_w,wc,w
s,delta_atten);
    [g1z,g2z,gz]=Obtain_g1_g2_from_f_h(fz,hz,n);
    fz_Polar=rectAngularToPolar(fz);
    hz_Polar=rectAngularToPolar(hz);
    gz_Polar=rectAngularToPolar(gz);
    count=count+1;
    K=fz(1)/gz(1);
    arrayofzeros=fz(2:length(fz));
    arrayofpoles=gz(2:length(gz));
    h_roots=hz(2:length(hz));
    %

[Atten,Mag,Phase,Delay,w]=FreqResponses_z(K,arrayofzeros,arrayofpoles,w_init,w_
final,NoOfPts);

[Atten_wc,Mag_wc,Phase_wc,Delay_wc]=FreqResponses_z_w(K,arrayofzeros,arrayofpol
es,wc);

[Atten_ws,Mag_ws,Phase_ws,Delay_ws]=FreqResponses_z_w(K,arrayofzeros,arrayofpol
es,ws);

    % Atten.
    atten_c=Atten(w<=wc);
    atten_s=Atten(w>=ws);

[Att_max_c,wh_max_c,Att_min_c,wh_min_c]=Find_LocalMin_Err_atten_PB(atten_c,w,wc
,length(wh_new),arrayofzeros,arrayofpoles,K); % find local min. points in the
stopband

[Att_min_s,wf_min_s,Att_max_s,wf_max_s]=Find_LocalMin_Err_atten_SB(atten_s,w,ws
,length(wf_new),arrayofzeros,arrayofpoles,K); % find local min. points in the
stopband
    % w
    wmax_c=[wh_max_c,wc]; % extremal points in passband
    wmin_s=[ws,wf_min_s]; % extremal points in stopband
    wmin_c=wh_min_c; % roots in passband
    wmax_s=wf_max_s; % roots in stopband
    % delta_atten.
    delta_atten_c=[App-Att_max_c,App-Atten_wc]/App; % passband
    delta_atten_s=[Atten_ws-Ass,Att_min_s-Ass]/Ass; % stopband
    delta_atten=[delta_atten_c,delta_atten_s]'; % modified differential
Attenuation - right side matrix

end % end while loop

```

```

B=1;
if B
    figure(60)
    plot_f_g_h(fz,gz,hz),grid
    figure(70)
    plot(w(w<=wc)/pi/2,Ap,'r',wt/pi/2,At,'r',w(w>=ws)/pi/2,As,'r'),hold on
    plot(w/pi/2,Atten,'k'),grid,xlabel('Normalized
Frequency'),ylabel('Attenucation(dB)')
    hold off
    figure(75)
    A1=(0:0.001:Ap);
    plot(w(w<=wc)/pi/2,Ap,'r',wc/pi/2,A1,'r'),hold on
    plot(w/pi/2,Atten,'k'),axis([0 wc/pi/2 0 1.1*Ap]),xlabel('Normalized
Frequency'),ylabel('Attenucation(dB)')
    hold off
    figure(80)
    plot(w/pi/2,Phase),grid,ylabel('Phase')
    xlabel('Normalized Frequency'),
    figure(90)
    plot(w/pi/2,Delay,'k'),grid,
    xlabel('Normalized Frequency'),ylabel('Group Delay')
end % End if loop

% Part2
% cascaded an allpass filter to obtain flat group delay in passband

[Atten,Mag,Phase,Delay,w]=plot_Z_domain(fz,gz,hz,w_init,w_final,NoOfPts,Ap,As,w
c,ws,count,B);
% cascaded an allpass filter
% adjust the phase (or group delay)
[r1,r2,angle2,scale,peak]=NonlinearLSAllpassDelay(na,w_init,w_final,NoOfPts,wc,
ws,fz,gz);

[a_den,a_num,m]=ConstructAllpass(r1,r2,angle2);
Delay_Ap=AllpassDelay_z(a_den,w);
Delay_oa=Delay+Delay_Ap;
figure(100)
plot(w/pi/2,Delay_oa,'k'),grid
ylabel('Group Delay/Sample'),xlabel('Normalized Frequency')
title('Group Delay Equalizaiton')
fz_o=polyMul_prodRep(fz,a_num);
gz_o=polyMul_prodRep(gz,a_den);

count=20; B=1;
[Atten_o,Mag_o,Phase_o,Delay_o,w]=plot_Z_domain(fz_o,gz_o,[],w_init,w_final,NoO
fPts,Ap,As,wc,ws,count,B);
figure(30)
plot(w(w<=wc)/pi/2,Delay_o(w<=wc),'k'),grid,
ylabel('Group Delay'),xlabel('Normalized Frequency')
figure(35)
subplot(1,2,1)
plot(w(w<=wc)/pi/2,scale*peak-Delay_o(w<=wc),'k'),grid
%title('The error of group delay in passband'),
xlabel('Normalized Frequency'),ylabel('Group delay error')
subplot(1,2,2)
target=scale*peak-Delay(w<=wc);
plot(w(w<=wc)/pi/2,target,'--',w(w<=wc)/pi/2,Delay_Ap(w<=wc),'k'),grid
title('The target curve vs. the designed curve')
xlabel('Normalized Frequency')
figure(40)
subplot(1,2,1)
zplane(fz(2:length(fz)),gz(2:length(gz))),grid
title('The WDLF filter zeros and poles')

```

```

subplot(1,2,2)
zplane(a_num(2:length(a_num)),a_den(2:length(a_den))),grid
title('The allpass filter zeros and poles')
legend('zeros','poles')

part 3
% To obtain overall ALP lowpass based on two allpass filters
% using modified Remez algorithm
% alp0

clear all;
addpath('/MATLABR11/work/FindZeros')
addpath('/MATLABR11/work/Special Math Fcts/Special Math Fcts')
addpath('/MATLABR11/work/WDLF')
addpath('/MATLABR11/work/IIR')
format long
%
% ***** Specification *****
%
Ap=0.001;
As=40;
fc=10;
fs=20;
F=100;
wc=2*pi*fc/F;
ws=2*pi*fs/F;
delta_c=1-10^(-Ap/20);
delta_s=10^(-As/20);
N=25;           % Set the order of overall filter
N1=(N+1)/2;     % the order of allpass subfilter1
N0=N-N1;        % the order of allpass subfilter0
epison=1e-3;
% estimate the overall lowpass filter based on 4 types IIR filter [Gazsi-86]
% Filter Type:
%      Butterworth:      1
%      Chebyshev:        2
%      Cauer:            3
%      Inverse Chebyshev: 4
%Type=input('What kind filter type will be chosed: Type=');
Type=4;
% Estimation of n
if Type==4
    Type1=2;
else
    Type1=Type;
end
phi_c=tan(wc/2);
phi_s=tan(ws/2);
n=Gazsi_Formula(Type1, Ap, As, phi_c, phi_s);

switch Type
case{1} % Butterworth Type: 1
    [sigma, f, g, h]=ButterworthFct(n, Ap, As, phi_c, phi_s);
    disp('Butterworth Filter Design')
case{2} % Chebyshev: 2
    disp('Chebyshev Filter Design')
    [sigma,f,g,h]=Chebyshev_f_g_h(n, Ap, As, phi_c, phi_s);
case{3} % Cauer Type: 3
    if Bireciprocal
        n=Birecip_n;
    end
    [sigma,f,g,h]=CauerFilterFct(n, Ap, As, phi_c, phi_s); % required by
    Birecip_g_from_f_phsi to obtain f

```

```

%[f,g,h]=f_g_h_CauerDesign(Ap, As, phi_c, phi_s);
if Bireciprocal
    [g,h]=Birecip_g_from_f_phsi(n,f); % n must be odd
    [f,g,h]=g_mononic(f,g,h); % makes g monic and rescales f and h accordingly
end
disp('Cauer Filter Design')
case{4} %Inverse Chebyshev Type: 4
    disp('Inverse Chebyshev Filter Design')
    [sigma, f, g, h]=InvCheb_f_g_h1(n, Ap, As, phi_c, phi_s);
end % end switch
%
% *****
%
Wc=1/(2*acos(1-delta_c)); % weight in passband; Wc
%Wc=1;
Ws=1/(2*asin(delta_s)); % weight in stopband; Ws
%Ws=5;
x=0; % count
delta_max=5; % max. delta
m=50;
delta_step=delta_max/m;
delta=0.5; % initial delta
%
% Step1
% select initial extremal N points
%
[beta0,beta1,w_all]=Beta_DenFromMag(N,f,g,wc,ws);
den0=AllpassDenFromPhaseRep(beta0,w_all,N0); % obtain the den. of allpass0
based on assumed ALP
Stable0=CheckStable(den0,N0);

[w_c,w_s,Np,Ns]=SelectInitPts(N1,wc,ws); % select initial N extremal points;
wi
wn1=ws; % let N+1 point: ws
wi=[w_c,w_s];
Beta0=AllpassFreqRep_SumRep_Z_1(den0,wi,N0);
%
% Step2
%
[f_delta,d]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,delta,Beta0,den0,Np);
[f_delta_neg,d_neg]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,-
delta,Beta0,den0,Np);
delta=delta+delta_step;

while sign(f_delta)==sign(f_delta_neg)

[f_delta,d]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,delta,Beta0,den0,Np);
[f_delta_neg,d_neg]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,-
delta,Beta0,den0,Np);
delta=delta+delta_step;
end
fd=[f_delta_neg,f_delta];
dt=[d_neg,d];
[delta0,f0,den1,Stable1]=find_delta_f_bisection_a(N,wc,ws,Wc,Ws,w_c,w_s,wn1,Np,
fd,dt,Beta0,den0); % find delta @ f(delta)=0

B=0;
if B
    figure(10)
    plot(dt,fd,delta0,f0,'or'),title('delta vs. f(delta)'), grid
end

```

```

[Err,wT]=Error_w_Beta(N,wc,ws,Wc,Ws,x,w_c,w_s,wn1,delta0,den0,Beta0); %
calculate the weight phase error of allpass subfilter1
e_max=max(Err);
e_min=min(Err);
d=max(abs(e_max),abs(e_min));
x=x+1;

B=0;
if B
    figure(20)
    plot(wT/2/pi,Err,wT/2/pi,delta0,'r',wT/2/pi,-
delta0,'r',wT/2/pi,d,'g',wT/2/pi,-d,'g'),title('Error at f(delta)=0'),grid
end

[w,err]=SelectExtremalPts_a(N1,Err,wT,wc,ws);
[w_c,w_s,Np]=PtsInPassandStopBand(N1,w,err,wc,ws);
wp=[w_c,w_s];
Diff=Diffwi_wp(wi,wp);
Beta0=AllpassFreqRep_SumRep_Z_1(den0,wp,N0);
%
%Step3
%
while max(abs(Err))-abs(delta0)>1e-3
    wi=wp;
    % go back Step2
    %
    delta=delta0;

[f_delta,d]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,delta,Beta0,den0,Np);
[f_delta_neg,d_neg]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,-
delta,Beta0,den0,Np);
    delta=delta+delta_step;

    while sign(f_delta)==sign(f_delta_neg)

[f_delta,d]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,delta,Beta0,den0,Np);
    [f_delta_neg,d_neg]=FindFdeltafrom2Allpass(N,wc,ws,Wc,Ws,wi,wn1,-
delta,Beta0,den0,Np);
    delta=delta+delta_step;
end
fd=[f_delta_neg,f_delta];
dt=[d_neg,d];

[delta0,f0,den1,Stable1]=find_delta_f_bisection_a(N,wc,ws,Wc,Ws,w_c,w_s,wn1,Np,
fd,dt,Beta0,den0); % find delta @ f(delta)=0

B=0;
if B
    figure(30)
    plot(dt,fd,delta0,f0,'or'),title('delta vs. f(delta)'), grid
end

[Err,wT]=Error_w_Beta(N,wc,ws,Wc,Ws,x,w_c,w_s,wn1,delta0,den0,Beta0); %
calculate the weight phase error of allpass subfilter1
e_max=max(Err);
e_min=min(Err);
d=max(abs(e_max),abs(e_min));

B=0;
if B
    figure(70)
    plot(wT/2/pi,Err,wT/2/pi,delta0,'r',wT/2/pi,-
delta0,'r',wT/2/pi,d,'g',wT/2/pi,-d,'g'),title('Error at f(delta)=0'),grid

```

```

end

[w,err]=SelectExtremalPts_a(Nl,Err,wT,wc,ws);
[w_c,w_s,Np]=PtsInPassandStopBand(Nl,w,err,wc,ws);
wp=[w_c,w_s];
xc=w_c/pi/2;
xs=w_s/pi/2;
Nl=length(wp);
Np=length(w_c);           % points in passband
Ns=Nl-Np;
Diff=Diffwi_wp(wi,wp)
x=x+1;
Beta0=AllpassFreqRep_SumRep_Z_1(den0,wp,N0);
end
figure(170)
plot(wT/2/pi,Err,'k',wT/2/pi,delta0,'k',wT/2/pi,-
delta0,'k',wT/2/pi,d,'k',wT/2/pi,-d,'k'),grid
xlabel('Normalized Frequncy'),ylabel('Normalized Error')

H_num=NumOfTrprodRep_Z_ALP(denl,den0); % obtain overall lowpass filter num
H_den=DenOfTrprodRep_Z_ALP(denl,den0); % obtain overall lowpass filter den
K=H_num(1)/H_den(1);
zeros=H_num(2:length(H_num));
poles=H_den(2:length(H_den));

B=1;
if B
    figure(180)
    hs=[];
    zplane(zeros,poles),grid
    legend('zeros','poles')
    title('overall zeros and poles')
    Phi0=AllpassFreqRep_SumRep_Z_1(den0,wT,N0);
    Phi1=AllpassFreqRep_SumRep_Z_1(denl,wT,N1);
    figure(185)
    plot(wT/pi/2,Phi0/pi,wT/pi/2,Phi1/pi,'r',wT/pi/2,-N/2*wT/pi,'-g'),axis([0
0.05 -2 0]),grid
    title('The Phase of Allpass subfilter0(Phi0) and Allpass subfilter1(Phi1)')
    xlabel('Frequency (Hz)')
end

w_init=0;
w_final=pi;
NoOfPts=1000;
[Atten,Mag,Phase,Delay,w]=FreqResponses_z(K,zeros,poles,w_init,w_final,NoOfPts)
;

figure(121)
plot(w/pi/2,-20*log10(Mag/2)),axis([0 0.05 0 Ap]),grid,xlabel('Normalized
Frequency'),ylabel('Attenuation (dB)'),title('Ripple in passband')
figure(122)
wt=w(w>wc & w<ws);
At=linspace(Ap,As,length(wt));
subplot(2,1,1)
plot(w(w<=wc)/pi/2,Ap,'-k',wt/pi/2,At,'-k',w(w>=ws)/pi/2,As,'-k'),
hold on
plot(w/pi/2,-20*log10(Mag/2)),grid,xlabel('Normalized
Frequency'),ylabel('Attenuation (dB)')
axis([0 0.5 0 120])
hold off
subplot(2,1,2)
plot(w(w<=wc)/pi/2,-20*log10(Mag(w<=wc)/2)),grid,xlabel('Normalized
Frequency'),ylabel('Attenuation (dB)')

```

```
figure(123)
plot(wT/pi/2,(Phi0+Phi1)/2),grid,xlabel('Normalized Frequnecy'),ylabel('Phase')
figure(124) %axis([0 0.05 min(Phase) max(Phase)]),
plot(w/pi/2,Delay),grid,xlabel('Normalized Frequnecy'),ylabel('Group
Delay/Sample'),axis([0 0.05 12 13])

B=1;
if B
    den_psi=z_To_psi_domainForPoly(length(den1),den1);
    [gammas,gammas1,gammas2]=AllpassToAdaptorCoef(den_psi)
    den_psi0=z_To_psi_domainForPoly(length(den0),den0);
    [gammas_0,gammas1_0,gammas2_0]=AllpassToAdaptorCoef(den_psi0)
end
```

Some useful functions

```
function A=MatrixofDevAtten(Kh,wh,wf,wmax_c,wmin_s,f,g,h,wc,ws)
% A=MatrixofDevAtten(Kh,wh,wf,wmax_c,wmin_s)
% input:
%     Kh: constant of h
%     wh: roots of h
%     wf: roots of f
%     wmax_c: extremal points in passband
%     wmin_s: extremal points in stopband
%     f:
%     g:
%     h:
%     wc:
%     ws:
% output:
%     A: the Matrix
%

Mk=0; % lowpass/highpass
wi=[wmax_c,wmin_s];
ni=length(wi);
nh=length(wh);
nf=length(wf);

B=1;
if B
for u=1:ni
    A_Kh(u,1)=2/Kh;
end

for u=1:ni
    for v=1:nh
        A_h(u,v)=(-sin(wi(u)-wh(v))/(1-cos(wi(u)-wh(v)))+sin(wi(u)+wh(v))/(1-
cos(wi(u)+wh(v))));
    end
end

for u=1:ni
    for v=1:nf
        A_f(u,v)=(-sin(wi(u)-wf(v))/(1-cos(wi(u)-wf(v)))+sin(wi(u)+wf(v))/(1-
cos(wi(u)+wf(v)))); % notice
    end
end % end for loop
end % end if loop

for u=1:ni
```

```

    if wi(u)<=wc
        A_M(u,1)=0;
    else
        A_M(u,1)=1;
    end
end

A=[A_Kh,A_h,A_f,A_M];

Function[Emax_pb,wmax_pb,Emin_pb,wmin_pb]=Find_LocalMin_Err_atten_PB(atten_p,w,
wc,n,zeros,poles,K)
%[Emax_pb,wmax_pb,Emin_pb,wmin_pb]=Find_LocalMin_Err_atten_PB(atten_p,w,wc,n,ze
ros,poles,K)
% Update: add zeros & poles & K
% calculate w and Err for no min. or max atten in passband.

v=1;
u=1;
w_c=w(w<=wc);
Pt1=1;
Pt_n=length(w_c);
Emin_p=[];
Emax_p=[];
wmin_p=[];
wmax_p=[];
[wMIN,eMIN,wMAX,eMAX]=ExtremaInInterval(w_c,atten_p,w_c(Pt1),w_c(Pt_n));
if isempty(eMIN) | isempty(eMAX)
    disp('Local extremal Atten. in passband can not be found');
    Emax_pb=[];
    wmax_pb=[];
    Emin_pb=[];
    wmin_pb=[];
    break;
end
if isnumeric(eMIN)
    for h=1:length(eMIN)
        wmin_p(u)=wMIN(h);
        Emin_p(u)=eMIN(h);
        u=u+1;
    end
end
if isnumeric(eMAX)
    for h=1:length(eMAX)
        wmax_p(v)=wMAX(h);
        Emax_p(v)=eMAX(h);
        v=v+1;
    end
end
%
if length(Emin_p)<n
    dn=n-length(Emin_p);
    w_cmin=linspace(wc/10,wc*0.9,dn);

[Atten_temp,Mag_temp,Phase_temp,Delay_temp]=FreqResponses_z_w(K,zeros,poles,w_c
min);
    Emin_p=sort([Emin_p,Atten_temp]);
    wmin_p=sort([wmin_p,w_cmin]);
end
if length(Emax_p)<n
    dn=n-length(Emax_p);
    w_cmax=linspace(wc/10,wc*0.9,dn);

```

```

[Atten_temp, Mag_temp, Phase_temp, Delay_temp]=FreqResponses_z_w(K, zeros, poles, w_c
max);
    Emax_p=sort([Emax_p, Atten_temp]);
    wmax_p=sort([wmax_p, w_cmax]);
end
%
B=1;
if B % for gdp1,2,3
if length(Emin_p)>n
    for u=1:length(Emin_p)
        for v=u:length(Emin_p)
            E_temp=Emin_p(u);
            w_temp=wmin_p(u);
            if E_temp>Emin_p(v)
                e1=Emin_p(v);
                w1=wmin_p(v);
                Emin_p(u)=e1;
                wmin_p(u)=w1;
                Emin_p(v)=E_temp;
                wmin_p(v)=w_temp;
            end
        end
    end
    for k=1:n
        Emin_pb(k)=Emin_p(k);
        wmin_pb(k)=wmin_p(k);
    end
else
    Emin_pb=Emin_p;
    wmin_pb=wmin_p;
end
if length(Emax_p)>n
    for u=1:length(Emax_p)
        for v=u:length(Emax_p)
            E_temp=Emax_p(u);
            w_temp=wmax_p(u);
            if E_temp>Emax_p(v)
                e1=Emax_p(v);
                w1=wmax_p(v);
                Emax_p(u)=e1;
                wmax_p(u)=w1;
                Emax_p(v)=E_temp;
                wmax_p(v)=w_temp;
            end
        end
    end
    for k=1:n
        Emax_pb(k)=Emax_p(length(Emax_p)-k+1);
        wmax_pb(k)=wmax_p(length(Emax_p)-k+1);
    end
else
    Emax_pb=Emax_p;
    wmax_pb=wmax_p;
end
end % end if B loop

```

```

function[r1_new, r2_new, angle2_new, scale_f, peak]=NonlinearLSallpassDelay(na, w_in
it, w_final, NoOfPts, wc, ws, fz, gz)
% using bisection method to find min(norm) with optimum scale factor.
%

```

```

Tol=1e-8;
Pts=100;    % number of sample points in the passband
w=linspace(w_init,w_final-Tol,NoOfPts);
Delay=obtainDelayCurve(fz,gz,w);    % group delay of H(z)
peak=max(Delay);    % the peak of delay
wx=wc*1.01;
w_x=linspace(w_init,wx-Tol,Pts);    % sample points
Delay_x=obtainDelayCurve(fz,gz,w_x);
%
B=1;
if B
scale1=0.5;    % initial scale factor [0.5:2]
scale2=2;

target1=scale1*peak-Delay_x;    % target curve
Pt1=target1(1);    % the first point of target curve
[r1_init1,r2_init1,angle2_init1]=InitialAllpassInPolar(wx,na,Pt1);
[norm1,r1_1,r2_1,angle2_1]=NonLeastSq4Allpass(r1_init1,r2_init1,angle2_init1,w_x,target1);

target2=scale2*peak-Delay_x;
Pt1=target2(1);
[r1_init2,r2_init2,angle2_init2]=InitialAllpassInPolar(wx,na,Pt1);
[norm2,r1_2,r2_2,angle2_2]=NonLeastSq4Allpass(r1_init2,r2_init2,angle2_init2,w_x,target2);
[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,scale2,peak,Delay_x,na,w_x,wx);
norm=[norm1,norm_l,norm_m,norm_r,norm2];
norm_min=min(norm);
%norm_max=max(norm);
norm_pre=10;
diff_norm=abs(norm_pre-norm_min) % set initial diff_norm

while diff_norm>0.001 %0.0001
    norm_pre=norm_min;
    while ((norm_m==norm_min)) && (norm_l<norm1) & (norm_r<norm2))
        scale_f=scale_m;
        scale1=scale_l;
        target1=scale1*peak-Delay_x;
        Pt1=target1(1);
        [r1_init1,r2_init1,angle2_init1]=InitialAllpassInPolar(wx,na,Pt1);

[norm1,r1_1,r2_1,angle2_1]=NonLeastSq4Allpass(r1_init1,r2_init1,angle2_init1,w_x,target1);

        scale2=scale_r;
        target2=scale2*peak-Delay_x;
        Pt1=target2(1);
        [r1_init2,r2_init2,angle2_init2]=InitialAllpassInPolar(wx,na,Pt1);

[norm2,r1_2,r2_2,angle2_2]=NonLeastSq4Allpass(r1_init2,r2_init2,angle2_init2,w_x,target2);

[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,scale2,peak,Delay_x,na,w_x,wx);
        norm=[norm1,norm_l,norm_m,norm_r,norm2];
        norm_min=min(norm);
        %norm_max=max(norm);
        %diff_norm=norm_max-norm_min
        diff_norm=abs(norm_pre-norm_min)
    end
    while (norm_l==norm_min)

```

```

    scale_f=scale_l;
    scale2=scale_m;
    target2=scale2*peak-Delay_x;
    Pt1=target2(1);
    [r1_init2,r2_init2,angle2_init2]=InitialAllpassInPolar(wx,na,Pt1);

[norm2,r1_2,r2_2,angle2_2]=NonLeastSq4Allpass(r1_init2,r2_init2,angle2_init2,w_
x,target2);

[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,scale2,peak,D
elay_x,na,w_x,w_x);
    norm=[norm_l,norm_l,norm_m,norm_r,norm2];
    norm_min=min(norm);
    %norm_max=max(norm);
    %diff_norm=norm_max-norm_min
    diff_norm=abs(norm_pre-norm_min)
end
while (norm_r==norm_min)
    scale_f=scale_r;
    scale1=scale_m;
    target1=scale1*peak-Delay_x;
    Pt1=target1(1);
    [r1_init1,r2_init1,angle2_init1]=InitialAllpassInPolar(wx,na,Pt1);

[norm1,r1_1,r2_1,angle2_1]=NonLeastSq4Allpass(r1_init1,r2_init1,angle2_init1,w_
x,target1);

[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,scale2,peak,D
elay_x,na,w_x,w_x);
    norm=[norm_l,norm_l,norm_m,norm_r,norm2];
    norm_min=min(norm);
    %norm_max=max(norm);
    %diff_norm=norm_max-norm_min
    diff_norm=abs(norm_pre-norm_min)
end
while (norm_l==norm_min)
    scale_f=scale_l;
    scale2=scale_l;
    target2=scale2*peak-Delay_x;
    Pt1=target2(1);
    [r1_init2,r2_init2,angle2_init2]=InitialAllpassInPolar(wx,na,Pt1);

[norm2,r1_2,r2_2,angle2_2]=NonLeastSq4Allpass(r1_init2,r2_init2,angle2_init2,w_
x,target2);

    scale1=scale1*0.9;
    target1=scale1*peak-Delay_x;
    Pt1=target1(1);
    [r1_init1,r2_init1,angle2_init1]=InitialAllpassInPolar(wx,na,Pt1);

[norm1,r1_1,r2_1,angle2_1]=NonLeastSq4Allpass(r1_init1,r2_init1,angle2_init1,w_
x,target1);

[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,scale2,peak,D
elay_x,na,w_x,w_x);
    norm=[norm_l,norm_l,norm_m,norm_r,norm2];
    norm_min=min(norm);
    %norm_max=max(norm);
    %diff_norm=norm_max-norm_min
    diff_norm=abs(norm_pre-norm_min)
end
while (norm2==norm_min)
    scale_f=scale2;

```

```

    scale1=scale2;
    target1=scale1*peak-Delay_x;
    Pt1=target1(1);
    [r1_init1,r2_init1,angle2_init1]=InitialAllpassInPolar(wx,na,Pt1);

[norm1,r1_1,r2_1,angle2_1]=NonLeastSq4Allpass(r1_init1,r2_init1,angle2_init1,w_
x,target1);

    scale2=scale2*1.1;
    target2=scale2*peak-Delay_x;
    Pt1=target2(1);
    [r1_init2,r2_init2,angle2_init2]=InitialAllpassInPolar(wx,na,Pt1);

[norm2,r1_2,r2_2,angle2_2]=NonLeastSq4Allpass(r1_init2,r2_init2,angle2_init2,w_
x,target2);

[norm_l,scale_l,norm_m,scale_m,norm_r,scale_r]=ObtainNormLeftMidRight(scale1,sc
ale2,peak,Delay_x,na,w_x,wx);
    norm=[norm1,norm_l,norm_m,norm_r,norm2];
    norm_min=min(norm);
    %norm_max=max(norm);
    %diff_norm=norm_max-norm_min
    diff_norm=abs(norm_pre-norm_min)
end
end
%
end % end if
% Final results
scale_f
target_f=scale_f*peak-Delay_x;
Pt1=target_f(1);
[r1_init,r2_init,angle2_init]=InitialAllpassInPolar(wx,na,Pt1);
[norm_new,r1_new,r2_new,angle2_new]=NonLeastSq4Allpass(r1_init,r2_init,angle2_i
nit,w_x,target_f);

function[norm,r1_new,r2_new,angle2_new]=NonLeastSq4Allpass(r1,r2,angle2,w,targe
t)
%

Tol=1e-8;
n1=length(r1);
n2=length(r2);
max_k=10;
maxCount=15;
minCount=5;
count=0;
s=1;
norm=1;

while ((norm>Tol) | (count<minCount)) & (count<maxCount)
    count=count+1;
    [a_den,a_num,n]=ConstructAllpass(r1,r2,angle2);
    delay_a=AllpassDelay_z(a_den,w);
    %
    B=0;
    if B
        figure(10)
        plot(w/pi/2,delay_a,w/pi/2,target),grid,
        title('Initial Group Delay of Allpass Filter')
        ylabel('Group Delay/Sample'),xlabel('Normalized Frequency')
    end
    %
    b=(target-delay_a).';

```

```

norm=sqrt(sum((abs(b)).^2));
A=JacobianMatrix4Allpass_ra(r1,r2,angle2,n,w);
s=A\b;
s=s.';
s=CleanComplexNo(s,Tol);

k=0; % count
count1=0;
norm_r=2*norm;
while ((norm_r>=norm) | (sum(abs(r1_new)>1)>=1) | (sum(abs(r2_new)>1)>=1))
& (count1<100)
    count1=count1+1;
    if ~isempty(r1)
        r1_new=r1+(2.^-k)*s(1:n1);
    else
        r1_new=[];
    end
    if ~isempty(r2)
        r2_new=r2+(2.^-k)*s((n1+1):(n1+n2));
        angle2_new=angle2+(2.^-k)*s((n1+n2+1):(n1+n2+n2));
        angle2_new=mod(angle2_new,2*pi);
    else
        r2_new=[];
        angle2_new=[];
    end
    [a_den,a_num,n]=ConstructAllpass(r1_new,r2_new,angle2_new);
    delay_a=AllpassDelay_z(a_den,w);
    %
    B=0;
    if B % test allpass
        figure(11)
        plot(w/pi/2,delay_a,w/pi/2,target),grid,title('Group Delay of Allpass
Filter')
        ylabel('Group Delay/Sample'),xlabel('Normalized Frequency')
    end
    k=k+1;
    b=(target-delay_a).';
    norm_r=sqrt(sum((abs(b)).^2));
end
norm=norm_r;
r1=r1_new;
r2=r2_new;
angle2=angle2_new;
end

function A=JacobianMatrix4Allpass_ra(r1,r2,angle2,n,w)
% A=JacobianMatrix4Allpass_ra(r1,r2,angle2,n,w)
% Input:
%   r1: the radius of 1st order allpass section
%   r2: the radius of 2nd order allpass section
%   angle2: the angle of 2nd order allpass section
%   n: the order of allpass filter
%   w: the frequency.
% Output:
%   A: Jacobian Matrix
%
if rem(n,2)==0 % even order with 2nd allpass sections, only.
    for k=1:length(w) % obtain least squares matrix (Jacobian Matrix)
        for i=1:length(r2)
            A(k,i)=Dif_DelayRad(r2(i),angle2(i),w(k));
            A(k,(i+length(r2)))=Dif_DelayAng(r2(i),angle2(i),w(k));
        end
    end
end

```

```

        end
    end
else
    for k=1:length(w) % obtain least squares matrix (Jacobian Matrix)
        A(k,1)=Dif_DelayRad_1st(r1,w(k));
        for i=1:length(r2)
            A(k,i+1)=Dif_DelayRad(r2(i),angle2(i),w(k));
            A(k,(i+1+length(r2)))=Dif_DelayAng(r2(i),angle2(i),w(k));
        end
    end
end
end
end

```

```

function [g1z,g2z,gz]=Obtain_g1_g2_from_f_h(fz,hz,n)
% [g1z,g2z,gz]=Obtain_g1_g2_from_f_h(fz,hz,n)
% odd order

tol=1e-12;
g1go2=polyAdd_prodRep(fz,hz); % g1go2=f+h
golg2=polySub_prodRep(fz,hz); % golg2=f-h
Kh=hz(1);

zerosArray1=g1go2(2:length(g1go2));
g1_zeros=zerosArray1(abs(zerosArray1)<1);
g1z=[1;g1_zeros];
zerosArray2=golg2(2:length(golg2));
g2_zeros=zerosArray2(abs(zerosArray2)<1);
g2z=[1;g2_zeros];

g_zeros=[g1_zeros;g2_zeros];
zerosArray3=g_zeros(abs(g_zeros)>tol);
g_temp=[1;zerosArray3];
Kg=sqrt(abs((1-Kh.^2)/polyVal_prodRep(0,g_temp))); % z domain
gz=[Kg;g_zeros];

```