RICE UNIVERSITY

DIGITAL FILTERS WITH THINNED NUMERATORS

by

GLORIA FAYE BOUDREAUX

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

MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE:

A handwritten signature in cursive script, reading "Thomas W. Parks", is written over a horizontal line.

Dr. T. W. Parks, Professor of
Electrical Engineering
Chairman

A handwritten signature in cursive script, reading "Dan H. Johnson", is written over a horizontal line.

Dr. D. H. Johnson, Assistant Professor
of Electrical Engineering

A handwritten signature in cursive script, reading "C. S. Burrus", is written over a horizontal line.

Dr. C. S. Burrus, Professor of
Electrical Engineering

HOUSTON, TEXAS

MAY 1980

ABSTRACT

DIGITAL FILTERS WITH THINNED NUMERATORS

by

Gloria Faye Boudreaux

An algorithm is described for designing digital filters that require few multiplies to produce good frequency response. The process of reducing the number of multiplies needed to implement a digital filter is called thinning. The thinning algorithm uses Dynamic Programming techniques to optimally approximate a desired Finite Impulse Response (FIR) filter with another FIR filter that requires significantly fewer non-zero coefficients to produce similar frequency response characteristics. The effects of coefficient quantization and finite-precision computer arithmetic upon the thinned filter structure are also described.

Examples of thinned narrowband, broadband, lowpass, and bandpass filters are given. Several of these thinned filters require fewer than one-third the number of multiplies required for the corresponding unthinned filter while still retaining desirable frequency response characteristics.

To Anthea

ACKNOWLEDGEMENTS

I find it extremely difficult to express adequately my appreciation for several special people who have generously given of their time, insight, encouragement, and support. I would like to acknowledge my indebtedness to Dr. Joe Rosenbaum who introduced me to the area of Digital Signal Processing and to my thesis advisor, Dr. Tom Parks, who patiently provided ideas and insights instrumental in the evolution of this thesis. Conversations with Charles Loeffler and Howard Coleman also stimulated several of the ideas in this thesis. Special thanks go to Clay Latimer, Bud Wendt, Val and Charles Loeffler who helped with the typing and proofreading of this manuscript.

Emotional sustenance provided during distressful moments will always be fondly appreciated. I would like to thank my parents for their support and Dr. Don Johnson for his lively coaxing and advice. I am especially grateful to my dear friends; Anthea Coster for initially encouraging me to enter graduate school, Charles Loeffler for opportunely encouraging me to persevere, and Clay Latimer for repeatedly encouraging me to effect my dreams.

This work was supported in part by NSF Grant ENG-7809033, and in part by a Zonta International Amelia Earhart Fellowship.

TABLE OF CONTENTS

CHAPTER (I)	INTRODUCTION AND CHAPTER OUTLINE	1
CHAPTER (II)	BACKGROUND	
A)	Motivation for Thinning	5
B)	Thinning Problem Formulation	7
1)	Problem 1 - Optimum Piecewise-Exponential Approximation	8
2)	Problem 2 - Thinned Filter Tap Coefficients	9
CHAPTER (III)	LEAST-SQUARES THINNING	
A)	Dynamic Programming Solution to Problem 1	13
B)	Thinned Filter Coefficients	18
1)	Piecewise-Linear Segments	18
2)	Piecewise-Continuous Linear Segments	21
3)	Piecewise-Exponential Segments	22
4)	Symmetric Piecewise-Exponential Segments	29
CHAPTER (IV)	QUANTIZATION	
A)	Finite-Precision Arithmetic Errors	32
B)	Quantization Errors in Digital Filters	33
C)	Integer-Arithmetic Implementation of Thinned Digital Filters	
1)	Integer Filter Coefficients	38
2)	Minimum Word Length Requirements	43

CHAPTER (V)	EXAMPLES	
A)	Narrowband Lowpass Filters	51
B)	Wideband Lowpass Filters	53
C)	Narrowband Bandpass Filters	54
CHAPTER (VI)	CONCLUSIONS	75
REFERENCES		77

CHAPTER I - INTRODUCTION

The algorithm presented in this thesis designs Finite Impulse Response (FIR) filters that require few multiplications to optimally approximate a desired FIR filter. The resulting approximation filter is said to be "thinned", since several of the filter coefficients were optimally selected to be set equal to zero. The term thinning originates in antenna theory[2]. For instance, a thinned antenna array is an array of non-uniformly spaced elements derived from an equally spaced array by systematically removing or thinning out certain elements. The analogy to digital filtering arises from the fact that the transfer function of recursive and non-recursive digital filters can be represented as the ratio of polynomials

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N-1} a_i z^{-i}}{1 + \sum_{j=1}^{M-1} b_j z^{-j}} \quad (1.1)$$

Thus, in a digital filter with a thinned numerator, several of the coefficients $\{a_i\}$ of the numerator polynomial $N(z)$ are equal to zero. Thinning numerator coefficients reduces the number of arithmetic operations needed to implement a digital filter.

There are several reasons for wanting to reduce the number of arithmetic operations required to implement a filter. Digital filters are often implemented using digital computers. On many computers, multiplication is a relatively time consuming operation. Also, each

arithmetic operation computed in finite-precision computer arithmetic may produce a loss of precision via computer overflow, underflow, truncation, or rounding. Reducing arithmetic filter operations can therefore increase the speed of filter implementation and decrease the number of finite-precision arithmetic errors. These motivations are discussed further in Chapter 2.

The proposed thinning algorithm optimally approximates a model impulse response, $g(n)$, of length P with a similar thinned filter finite impulse response sequence, $h(n)$, that requires significantly fewer multiplications to implement. In particular, if the approximation sequence, $h(n)$, is required to be piecewise-exponential, then the number of non-zero thinned filter coefficients is dependent only on the number of segments used in the approximation. The thinning algorithm finds the optimal S -segment piecewise-exponential approximation sequence using a procedure called dynamic programming[1]. Chapter 2 delineates the thinning design algorithm into two sub-procedures. First, for a given pair of exponentials, the best least-squares S -segment piecewise-exponential approximation to a desired sequence, $g(n)$, of length P is found. Second, the optimal approximation sequence is reformulated in such a way that it is easily interpreted as the impulse response of a thinned FIR filter that requires a maximum of $2(S+2)$ multiplies to implement.

This thinned finite-length impulse response filter, $H(z)$, is realized as an FIR filter, $N(z)=H(z)D(z)$, cascaded with an IIR filter, $1/D(z)$. The second order IIR filter has poles corresponding to the exponentials used in the least-squares approximation. The segment boundaries of the optimum piecewise fit determine the locations of the non-zero coefficients of $N(z)$; the weights of the exponential bases in each of the S segments determine the values of the non-zero filter coefficients .

Chapter 3 explains how dynamic programming techniques can be used to solve the initial problem of finding the optimal S -segment piecewise-exponential approximation. Formulae for calculating the $2(S+1)$ non-zero coefficients of the thinned numerator polynomial $N(z)$ are given for the cases of real and complex basis exponentials. If the S -segment approximation is required to be piecewise-continuous or symmetric, then the number of thinned filter multiplies can be further reduced to $S+4$. Furthermore, symmetrical piecewise-continuous approximations require only $(4+S/2)$ multiplies.

The FIR-IIR cascade implementation of the thinned filter implicitly assumes that pole-zero cancellation will occur, resulting in a FIR filter $H(z)$. However, coefficient quantization could cause the poles and the zeros of the cascaded filters to deviate from their desired positions. These deviations could prevent pole-zero cancellation, thus producing an impulse response of theoretically infinite length. If the uncanceled poles lie on or outside the unit circle, the resulting IIR filter will be unstable. Chapter 4 analyzes the

effects of filter coefficient quantization and finite-precision computer arithmetic on the pole-zero cancellation assumption. Scaling strategies which minimize the chance of overflow when the filter is implemented in fixed-point arithmetic are discussed. Bounds on the minimum word lengths needed to insure pole-zero cancellation for the case of integer arithmetic are also given.

Several examples of thinned filters are given in Chapter 5. The design algorithm was used on high order narrowband and broadband, lowpass and bandpass filters. Several of the thinned filters require fewer than one-third the number of multiplies of the corresponding unthinned filter and yet still retain desirable frequency response characteristics. Chapter 6 contains a summary of the results and insights obtained and suggests possible avenues of future research.

CHAPTER II - BACKGROUND

A. MOTIVATION FOR THINNING

The transfer function of a digital filter $H(z)$ can be represented as a rational function of a numerator polynomial $N(z)$ and a denominator polynomial $D(z)$, where

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N-1} a_i z^{-i}}{1 + \sum_{j=1}^{M-1} b_j z^{-j}} \quad (2.1)$$

A digital filter with a thinned numerator has several of the coefficients $\{a_i\}$ of the numerator polynomial $N(z)$ equal to zero.

There are several reasons to reduce the number of arithmetic operations required to implement a filter. The most obvious reasons include an increase in computation speed and a decrease in the amount of hardware needed for filter realization. Frequently, a FIR digital filter is desired that is computationally fast yet still has a frequency response with a large attenuation in the stopband and a narrow transition region between the edges of the passband and the stopband. These design criteria are often conflicting for a given filter sampling rate. Recall that the frequency response of an FIR filter can be represented as a linear combination of a finite number of weighted complex exponentials of increasing angular frequency. The outermost pair of impulse response coefficients weight the highest frequency

cosine term in the linear combination. For many FIR filter design techniques [19], decreasing the width of the transition region while maintaining a given stopband attenuation necessitates increasing the order of the FIR filter. This increase reflects the need to include higher frequency cosine terms to produce a sharper transition region. Unfortunately, this increase in the order also increases the computational complexity of the filter. Thinning provides a method for simultaneously retaining the high filter order needed to produce sharp transition filters while reducing the computation time needed for filter calculations.

The calculations needed to implement a digital filter are computed in the finite-precision arithmetic of digital computers. Each arithmetic operation required for filter computation could result in a loss of accuracy through machine overflow, underflow, truncation, or rounding. The chances of fixed-point and integer overflow can be minimized through the use of scaling strategies [18]. When an L_1 scaling strategy is used, the dynamic range of the filter input, coefficients, and/or output is inversely proportional to the sum of the absolute values of the filter coefficients. Thinning filter coefficients could reduce the number of occasions for finite-precision arithmetic errors to occur, and could increase the dynamic range specifications of the filter.

B. THINNING PROBLEM FORMULATION

Various optimality criteria may be used to guide the selection of the non-zero filter coefficients to insure that the thinned digital filter differs minimally from a desired digital filter that had more equally spaced coefficients. Dynamic Programming techniques have already been employed to thin the uniformly spaced elements of array antennas [3;4;5].

The proposed thinning algorithm optimally approximates a desired impulse response, $g(n)$, of length P with an equal length piecewise-exponential sequence, $h(n)$, that can be generated using only $2(S+2)$ multiplies. Given a pair of exponential basis functions, z_1^n and z_2^n , the thinning algorithm uses dynamic programming techniques to find the best least-squares S -segment piecewise-exponential fit, $h(n)$, to the model impulse response sequence, $g(n)$. The design procedure exploits the fact that any linear combination, $Az_1^n + Bz_2^n$, of the user-specified exponential sequences can be generated by an appropriately chosen second order IIR filter,

$$[\alpha + \beta z^{-1}] F(z) \quad (2.2)$$

where

$$\alpha = A + B \quad (2.3)$$

$$\beta = -(Az_2 + Bz_1) \quad (2.4)$$

and

$$F(z) = \frac{1}{D(z)} = \frac{1}{(1-z_1 z^{-1})(1-z_2 z^{-1})} \quad (2.5)$$

Recall that the two poles of $F(z)$ produce the exponential impulse response

$$f(n) = \left\{ \begin{array}{ll} \frac{1}{z_1 - z_2} (z_1^{n+1} - z_2^{n+1}) u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{array} \right\} \quad (2.6)$$

where $u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise} \end{cases}$ is the unit step function.

The thinned filter in equation (2.2) requires only 4 multiplications to implement regardless of the length of the desired exponential sequence. Furthermore, piecewise-exponential sequences can be generated by cascading $F(z)$ with a FIR filter, $N(z) = H(z)D(z)$. The cascade realization results in the FIR filter, $H(z)$, with an impulse response, $h(n)$, that is a linear combination of weighted and shifted versions of $f(n)$. Schussler [6] has used a similar idea with an analog pulse-former and a tapped delay line.

The thinning algorithm can be divided into two sub-procedures. The two basic problems used to produce the optimal thinned filter, $H(z)$, are:

PROBLEM 1

Given the desired impulse response sequence $g(n)$, $n=0,1,\dots,P-1$, the two basis exponential sequences, z_1^n and z_2^n , and the order of the problem $S \ll P$, find the ordered set of breakpoints

$$\{\tau_0, \tau_1, \dots, \tau_S\} \in \{0, 1, \dots, P\},$$

and the set of weighting coefficients

$$\{A_0, A_1, \dots, A_{S-1}\} \text{ and } \{B_0, B_1, \dots, B_{S-1}\},$$

that yield the S-segment piecewise-exponential sequence

$$h(n) = \sum_{k=0}^{S-1} (A_k f(n) + B_k f(n-1)) [u(n-\tau_k) - u(n-\tau_{k+1})] \quad (2.7)$$

which minimizes the squared error

$$e^2 = \sum_{n=-\infty}^{\infty} [g(n) - h(n)]^2. \quad (2.8)$$

Recall,

$$f(n) = \begin{cases} \frac{1}{(z_1 - z_2)} [z_1^{n+1} - z_2^{n+1}] u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{cases}$$

is the impulse response of the IIR filter, $F(z)$, in equation (2.6).

PROBLEM 2

Given the ordered set of breakpoints

$$\{\tau_0, \tau_1, \dots, \tau_S\} \in \{0, 1, \dots, P\},$$

and the weighting coefficients

$$\{A_0, A_1, \dots, A_{S-1}\}, \{B_0, B_1, \dots, B_{S-1}\},$$

that describe the optimal piecewise-exponential sequence

$$h(n) = \sum_{k=0}^{S-1} (A_k f(n) + B_k f(n-1)) [u(n-\tau_k) - u(n-\tau_{k+1})],$$

find the thinned filter tap coefficients

$$\{\alpha_0, \alpha_1, \dots, \alpha_S\} \text{ and } \{\beta_0, \beta_1, \dots, \beta_S\}$$

such that

$$h(n) = \sum_{k=0}^S [\alpha_k f(n-\tau_k) + \beta_k f(n-1-\tau_k)] \quad (2.9)$$

where

$$f(n) = \begin{cases} \frac{1}{z_1 - z_2} [z_1^{n+1} - z_2^{n+1}] u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{cases} .$$

Note that $h(n)$ consists of weighted and shifted versions of $f(n)$, the impulse response of the IIR filter $F(z)$ in equation (2.6). Therefore, $h(n)$ is the impulse response of the "thinned" digital filter,

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{k=0}^S (\alpha_k + \beta_k z^{-1}) z^{-\tau_k}}{(1-z_1 z^{-1})(1-z_2 z^{-1})} \quad (2.10)$$

The poles of the filter correspond to the two user-specified exponentials. The locations of the non-zero numerator coefficients correspond to the optimum segment boundaries found in Problem 1. The values of the numerator coefficients are computed in Problem 2. The resulting thinned filter can be implemented with a maximum of $2(S+2)$ multiplies versus the P multiplies needed to implement the model FIR filter $G(z)$.

Equation (2.7) defines $h(n)$ to be a finite length sequence. This implies that $N(z)$ has zeros located at $z=z_1, z_2$ which cancel the poles of the IIR filter $F(z)$. The resulting FIR filter has the con-

figuration shown in Figure 2-1.

FIGURE 2-1. Filter $H(z)$ realized as an FIR filter cascaded with a second order recursive filter. The optimum coefficients $\{\alpha_k, \beta_k\}$ and locations $\{\tau_k\}$ of the non-uniformly spaced taps are generated using a dynamic programming algorithm.

In summary, the filter thinning algorithm uses two user-specified exponential sequences, z_1^n and z_2^n , to find the best least-squares S-segment piecewise-exponential fit to the desired impulse response sequence $g(n)$ of length P , where $S \ll P$. The algorithm then reformulates the optimal S-segment approximation into a form easily interpreted as the impulse response of a thinned filter, $H(z)$, which requires a maximum of $2(S+2)$ multiplications to implement. The thinned filter is realized as an FIR filter

$$N(z) = \sum_{k=0}^S (\alpha_k + \beta_k z^{-1}) z^{-k} \quad (2.11)$$

cascaded with a second order IIR filter

$$F(z) = \frac{1}{(1-z_1 z^{-1})(1-z_2 z^{-1})} \quad (2.12)$$

The locations of the non-zero coefficients of $N(z)$ correspond to the boundaries of the optimal S-segments. The values of the numerator coefficients are a function of the optimum weights $\{A_k, B_k\}$. The poles of $F(z)$ are chosen to correspond to the two user-specified basis exponentials. The formulae to compute the optimal tap delays $\{\tau_k\}$ and the non-zero numerator coefficients $\{\alpha_k, \beta_k\}$ are derived in Chapter 3.

CHAPTER III - LEAST-SQUARES THINNING

A. Dynamic Programming Solution to Problem 1

Recall that the first stage of the thinning algorithm finds the weights, $\{A_k, B_k\}, k=0,1,\dots,S-1$, and the segment boundaries $\{\tau_k\} \in \{0,1,\dots,P\}, k=0,1,\dots,S$, which produce the optimum piecewise-exponential approximation

$$h(n) = \sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)] [u(n-\tau_k) - u(n-\tau_{k+1})] \quad (3.1)$$

to the desired impulse response $g(n), n=0,1,\dots,P-1$. Here,

$$f(n) = \left\{ \begin{array}{ll} \frac{1}{z_1 - z_2} [z_1^{n+1} - z_2^{n+1}] u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{array} \right\} \quad (3.2)$$

One possible method for finding $h(n)$ is to consider all possible configurations of the $(S+1)$ breakpoints $\{\tau_k\}$ distributed among the $P+1$ points, $n=0,1,\dots,P$. Then for each configuration, the optimum weighting coefficients $\{A_k, B_k\}$ are computed along with the resulting approximation error. Finally, after an exhaustive search, the segment configuration is chosen which minimizes the squared-error

$$e^2 = \sum_{n=-\infty}^{\infty} [g(n) - h(n)]^2 \quad (3.3)$$

However, the total enumeration of $(P+1)$ tap locations taken $(S+1)$ at a time quickly grows into an intractable problem for even small values of P .

Dynamic Programming techniques [1;21] drastically reduce the number of tap configurations that must be examined to find the optimum breakpoint locations. This reduction is accomplished by converting an S-dimensional optimization problem into a sequence of S one-dimensional optimization problems. This time-saving approach is applicable to problems that satisfy the Principle of Optimality, which states: "an optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." [21;p.15]

Bellman [7] has shown that the problem of approximating a curve with line segments satisfies the Principle of Optimality. The formulation for linear-segment approximation can be extended to exponential-segment approximation. Let

$$d(A,B,i,j) = \sum_{n=i}^{j-1} [g(n) - (Af(n) + Bf(n-1))]^2, \quad i < j \quad (3.4)$$

be a measure of the distance or squared error between an exponential segment and the desired sequence $g(n)$ over the interval $n \in [i,j)$. The approximation algorithm seeks the $\{A_k\}$, $\{B_k\}$, and $\{\tau_k\}$ that minimize the S-dimensional error function

$$\begin{aligned} E(A_0, \dots, A_{S-1}; B_0, \dots, B_{S-1}; \tau_1, \dots, \tau_{S-1}) &= \sum_{n=0}^{P-1} [g(n) - h(n)]^2 \\ &= \sum_{n=0}^{P-1} \left[g(n) - \left[\sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)] [u(n-\tau_k) - u(n-\tau_{k+1})] \right] \right]^2 \end{aligned} \quad (3.5)$$

$$\begin{aligned}
&= \sum_{k=0}^{S-1} \left[\sum_{n=\tau_k}^{(\tau_{k+1})-1} [g(n) - (A_k f(n) + B_k f(n-1))] \right]^2 \\
&= \sum_{k=0}^{S-1} [d(A_k, B_k, \tau_k, \tau_{k+1})]
\end{aligned}$$

over the entire interval $n \in [0, P)$. Here, $\tau_0 = 0$, $\tau_S = P$, and $1 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_{S-1} \leq P-1$.

The dynamic programming approach breaks this S -dimensional problem into a sequence of S one-dimensional problems, by noting that for $2 \leq j \leq S$, the

$$\begin{aligned}
\{\text{best } j\text{-segment fit}\} &= \min \{ \{\text{best } (j-1)\text{-segment fit for } n \in [0, \tau_{j-1})\} \\
&\quad + \{\text{best } 1\text{-segment fit for } n \in [\tau_{j-1}, P)\} \} .
\end{aligned}$$

The algorithm begins by computing for all $0 \leq i < j \leq P$,

$$e_1(i, j) = \min_{\{A, B\}} [d(A, B, i, j)], \quad (3.6)$$

the minimum error produced by approximating the sequence $\{g(i), g(i+1), \dots, g(j-1)\}$ with one exponential segment.

Given the set of boundaries (i, j) , the optimum weights A and B , are relatively easy to obtain.

Define the weighted L_2 inner product

$$(x(n), y(n))_{i,j} = \sum_{n=i}^{j-1} [x(n)y^*(n)]w_{i,j}(n) \quad , \quad (3.7)$$

where the weighting function

$$w_{i,j}(n) = [u(n-i) - u(n-(j-1))] \quad , \quad (3.8)$$

and the weighted L_2 norm ,

$$||x(n)||_{i,j}^2 = (x(n), x(n))_{i,j} \quad (3.9)$$

The optimum $\tilde{A}$, $\tilde{B}$ can be found using a Gram Matrix [20] formulation:

$$\begin{bmatrix} \tilde{A} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} ||f(n)||_{i,j}^2 & (f(n), f(n-1))_{i,j} \\ (f(n-1), f(n))_{i,j} & ||f(n-1)||_{i,j}^2 \end{bmatrix}^{-1} \begin{bmatrix} (g(n), f(n))_{i,j} \\ (g(n), f(n-1))_{i,j} \end{bmatrix} \quad (3.10)$$

These values for $\tilde{A}$ and $\tilde{B}$ yield the

$$\min[d(A, B, i, j)] = ||g(n)||_{i,j}^2 - \tilde{A}(g(n), f(n))_{i,j} - \tilde{B}(g(n), f(n-1))_{i,j} \quad (3.11)$$

The algorithm then computes the best two-segment approximation

$$e_2(0, P) = \text{minimum} [d(A_0, B_0, 0, \tau_1) + d(A_1, B_1, \tau_1, P)] \quad (3.12)$$

$$\{A_0, A_1, B_0, B_1, \tau_1\}$$

$$\tau_1 \in \{2, 3, \dots, P-1\}$$

This function is also easily computed by finding the minima over A_k and B_k using equation (3.10), and then minimizing over τ_1 , by a discrete search over $\tau_1 = 2, \dots, P-1$.

The pattern emerging in the error function of the optimum 2-segment fit,

$$e_2(0,P) = \text{minimum}_{\tau_1 \in \{2,3,\dots,P-1\}} [e_1(0,\tau_1) + e_1(\tau_1,P)] \quad (3.13)$$

and the error function of the best three-segment fit,

$$e_3(0,P) = \text{minimum}_{\tau_2 \in \{4,5,\dots,P-1\}} [e_2(0,\tau_2) + e_1(\tau_2,P)] \quad (3.14)$$

yield a recurrence relation

$$e_j(0,P) = \text{minimum}_{\tau_{j-1} \in \{2j,2j+1,\dots,P-1\}} [e_{j-1}(0,\tau_{j-1}) + e_1(\tau_{j-1},P)], \quad 2 \leq j \leq S \quad (3.15)$$

for the optimum j-segment fit. Note that each successive iteration searches over fewer values of τ , since $(2j-1)$ points can be exactly fit with $(j-1)$ segments. That is,

$$e_{j-1}(0,r)=0, \quad 0 \leq r \leq 2j-1 \quad (3.16)$$

The algorithm stops at the S^{th} -iteration producing

$$e_S(0,P) = \text{minimum}_{\tau_{S-1} \in \{2(S-1),2S-1,\dots,P-1\}} [e_{S-1}(0,\tau_{S-1}) + e_1(\tau_{S-1},P)] \quad (3.17)$$

$$= \text{minimum}_{\{A_k\}, \{B_k\}, \{\tau_k\}} [E(A_0, \dots, A_{S-1}; B_0, \dots, B_{S-1}; \tau_1, \dots, \tau_{S-1})]$$

$$\{A_k\}, \{B_k\}, \{\tau_k\}$$

$$k=0,1,\dots,S-1$$

B. Thinned Filter Coefficients

1. Piecewise-Linear Segments

One of the simplest conceptual choices in designing a thinned filter is to make $f(n)$ a linear function of n , or a ramp. Piecewise linear approximations produce a good fit when the desired impulse response $g(n)$ is slowly varying. For example, this type of sequence occurs when $G(z)$ is a lowpass filter.

A second order IIR filter,

$$F(z) = \frac{1}{(1-z^{-1})^2} \quad \langle \text{----} \rangle \quad f(n) = (n+1)u(n), \quad (3.18)$$

with a double pole at $z=1$ can be used to generate piecewise-linear segments. Recall, dynamic programming was used to find the weights $\{A_k, B_k\}$ and the breakpoints $\{\tau_k\}$ that produced the optimum piecewise linear fit

$$\begin{aligned} h(n) &= \sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)][u(n-\tau_k) - u(n-\tau_{k+1})] \\ &= \sum_{k=0}^{S-1} [A_k (n+1)u(n) + B_k n u(n)][u(n-\tau_k) - u(n-\tau_{k+1})] \end{aligned} \quad (3.19)$$

The second stage of the thinning algorithm deals with computing the numerator coefficients $\{\alpha_k, \beta_k\}$ needed to realize the thinned filter impulse response:

$$h(n) = \sum_{k=0}^S [\alpha_k f(n-\tau_k) + \beta_k f(n-1-\tau_k)] \quad (3.20)$$

$$= \sum_{k=0}^S [\alpha_k(n+1-\tau_k)u(n-\tau_k) + \beta_k(n-\tau_k)u(n-\tau_k)].$$

The summations in equation (3.19) and (3.20) are equivalent when

$$\alpha_0 = A_0 \quad (3.21)$$

$$\beta_0 = B_0 \quad (3.22)$$

$$\alpha_k = (A_k - A_{k-1})(1+\tau_k) + (B_k - B_{k-1})\tau_k \quad (3.23)$$

$$\beta_k = -\tau_k(A_k - A_{k-1}) + (B_k - B_{k-1})(1-\tau_k), \quad k=1,2,\dots,S \quad (3.24)$$

where $A_S = B_S = 0$.

***** PROOF *****

Setting equation (3.19) and (3.20) equal to one another for all values of n yields the equation

$$0 = \sum_{k=0}^{S-1} [A_k(n+1)u(n) + B_k n u(n)] [u(n-\tau_k) - u(n-\tau_{k+1})] \quad (3.25)$$

$$- \sum_{r=0}^S [\alpha_r(n+1-\tau_r)u(n-\tau_r) + \beta_r(n-\tau_r)u(n-\tau_r)]$$

$$= \sum_{k=0}^{S-1} [((A_k + B_k)n + A_k)u(n-\tau_k) - ((A_k + B_k)n + A_k)u(n-\tau_{k+1})]$$

$$- \sum_{r=0}^S [(\alpha_r + \beta_r)n + (\alpha_r(1-\tau_r) - \beta_r\tau_r)]u(n-\tau_r)$$

$$\begin{aligned}
&= [(A_0 - \alpha_0) + (B_0 - \beta_0)]nu(n) + (A_0 - \alpha_0)u(n) \\
&\quad + \sum_{k=1}^{S-1} \left[(A_k - A_{k-1} - \alpha_k) + (B_k - B_{k-1} - \beta_k) \right] n + [(A_k - A_{k-1} - \alpha_k)(1 - \tau_k) + \beta_k \tau_k] u(n - \tau_k) \\
&\quad - [(A_{S-1} + \alpha_S) + (B_{S-1} + \beta_S)]nu(n-P) - [A_{S-1} + \alpha_S(1-P) - \beta_S P]u(n-P) .
\end{aligned}$$

For the values of $0 \leq n < \tau_1$, equation (3.25) can be satisfied by noting that the two linear equations

$$0 = (A_0 - \alpha_0) + (B_0 - \beta_0) \quad (3.26)$$

$$0 = A_0 - \alpha_0 \quad (3.27)$$

have the solution

$$\alpha_0 = A_0 \quad (3.28)$$

$$\beta_0 = B_0 \quad (3.29)$$

For $\tau_k \leq n < \tau_{k+1}$, $k=1, 2, \dots, S-1$, one obtains the equations

$$0 = (A_k - A_{k-1} - \alpha_k) + (B_k - B_{k-1} - \beta_k) \quad (3.30)$$

$$0 = (A_k - A_{k-1} - \alpha_k(1 - \tau_k)) + \beta_k \tau_k \quad (3.31)$$

which have the solution

$$\alpha_k = (A_k - A_{k-1})(1 + \tau_k) + (B_k - B_{k-1})\tau_k \quad (3.32)$$

$$\beta_k = -\tau_k(A_k - A_{k-1}) + (B_k - B_{k-1})(1 - \tau_k) . \quad (3.33)$$

Finally, for $n \geq S=P$, the sufficient equations

$$0 = (A_{S-1} + \alpha S) + (B_{S-1} + \beta S) \quad (3.34)$$

$$0 = (A_{S-1} + \alpha S(1-P)) - \beta S^P \quad (3.35)$$

are satisfied when

$$\alpha S = - [(P+1)A_{S-1} + PB_{S-1}] \quad (3.36)$$

$$\beta S = PA_{S-1} - (1-P)B_{S-1} \quad (3.37)$$

***** q.e.d. *****

Therefore, the thinned filter,

$$H(z) = \frac{\sum_{k=0}^S [\alpha_k + \beta_k z^{-1}] z^{-k}}{(1-z^{-1})^2}, \quad (3.38)$$

will yield a sequence identical to the optimum piecewise-linear approximation found by the dynamic programming algorithm. This filter configuration requires a maximum of $2(S+1)$ multiplications for the thinned numerator polynomial and one multiplication for the denominator polynomial $D(z)=1-2z^{-1}+1$. Examples of thinned lowpass filters designed using piecewise-linear segments are described in section 5.A .

2. Piecewise-Continuous Linear Segments

Bellman and Roth [8] have outlined a dynamic programming approach to the problem of finding the optimum piecewise-continuous linear fit to a given set of data points. In the original linear-

segment approximation, two coefficients were needed to describe each linear segment. Forcing the lines to intersect at a segment boundary eliminates one degree of freedom in the selection of the next segment. Consequently, the resulting thinned filter implementation requires only one non-zero tap coefficient per intermediate segment. Thus, the S-segment piecewise-continuous linear approximation sequence can be generated by a thinned filter that requires only $S+4$ multiplications. However, the dynamic program memory storage required for the piecewise-continuous algorithm is prohibitive when the desired sequence length, P , is greater than 20.

3. Piecewise-Exponential Segments

Piecewise-linear segments are useful for approximating sequences that exhibit local linear trends. Exponentials or exponentially weighted sinusoids may prove more expedient than linear segments in the approximation of rapidly varying sequences. The linear-segment formulation for tap coefficient weights is readily extended to exponential-segments. Purely exponential sequences can be generated by choosing real values for the poles, z_1 and z_2 . Exponential sequences modulated by a sinusoid can be generated by choosing $z_1 = z_2^*$.

Let

$$h(n) = \sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)][u(n-\tau_k) - u(n-\tau_{k+1})] \quad (3.39)$$

be the optimum S-segment piecewise-exponential approximation found

using dynamic programming techniques. Recall,

$$f(n) = \begin{cases} \frac{1}{z_1 - z_2} [z_1^{n+1} - z_2^{n+1}] u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{cases} \quad (3.40)$$

The thinned filter tap coefficients $\{\alpha_k, \beta_k\}$ needed to generate this piecewise-exponential sequence are:

$$\alpha_0 = A_0 \quad (3.41)$$

$$\beta_0 = B_0$$

and for $k=1, 2, \dots, S$,

$$\alpha_k = \begin{cases} \left[(A_k - A_{k-1})(\tau_k + 1) z_1^{\tau_k} + (B_k - B_{k-1}) \tau_k z_1^{\tau_k - 1} \right] & , z_1 = z_2 \\ \left[\frac{(A_k - A_{k-1})(z_1^{\tau_k + 1} - z_2^{\tau_k + 1}) + (B_k - B_{k-1})(z_1^{\tau_k} - z_2^{\tau_k})}{z_1 - z_2} \right] & , z_1 \neq z_2 \\ \left[\frac{[(A_k - A_{k-1}) \sin[(\tau_k + 1)\phi] + (B_k - B_{k-1}) \sin(\tau_k \phi)] |M|^{-1}}{\sin(\phi)} |M|^{\tau_k} \right] & , z_1 = z_2^* \\ & , z_1 = |M| e^{j\phi} \end{cases} \quad (3.42)$$

$$\beta_k = \begin{cases} \left[\frac{-(A_k - A_{k-1})\tau_k z_1^{\tau_k+1} + (B_k - B_{k-1})(1-\tau_k)z_1^{\tau_k-1}(1-\tau_k)}{z_1 - z_2} \right], & z_1 = z_2 \\ \left[\frac{-z_1 z_2 [(A_k - A_{k-1})(z_1^{\tau_k} - z_2^{\tau_k}) + (B_k - B_{k-1})(z_1^{\tau_k-1} - z_2^{\tau_k-1})]}{z_1 - z_2} \right], & z_1 \neq z_2 \\ \left[\frac{[|M|(A_k - A_{k-1})\sin(\tau_k \phi) + (B_k - B_{k-1})\sin[(\tau_k - 1)\phi]]|M|^{\tau_k}}{\sin(\phi)} \right], & z_1 = z_2^* \\ & z_1 = |M|e^{j\phi} \end{cases} \quad (3.43)$$

where $A_S = B_S = 0$.

***** PROOF *****

$$\begin{aligned} 0 &= \sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)] [u(n-\tau_k) - u(n-\tau_{k+1})] \\ &\quad - \sum_{r=0}^S [\alpha_r f(n-\tau_r) + \beta_r f(n-1-\tau_r)] \end{aligned} \quad (3.44)$$

ASSUME $z_1 = z_2$.

(Note that this formulation is an extension of the piecewise-linear case where $z_1 = z_2 = 1$.)

$$\begin{aligned} 0 &= \sum_{k=0}^{S-1} [A_k (n+1) z_1^n u(n) + B_k n z_1^{n-1} u(n)] [u(n-\tau_k) - u(n-\tau_{k+1})] \\ &\quad - \sum_{r=0}^S [\alpha_r (n+1-\tau_r) z_1^{(n-\tau_r)} + \beta_r (n-\tau_r) z_1^{(n-1-\tau_r)}] u(n-\tau_r) \end{aligned} \quad (3.45)$$

$$\begin{aligned}
&= \sum_{k=0}^{S-1} [(A_k - \alpha_k z_1^{-\tau_k}) + (B_k - \beta_k z_1^{-\tau_k}) z_1^{-1}] n z_1^n u(n - \tau_k) \\
&\quad + (A_k - \alpha_k (1 - \tau_k) z_1^{-\tau_k} + \beta_k \tau_k z_1^{-(\tau_k+1)}) z_1^n u(n - \tau_k) \\
&- \sum_{q=1}^S [(A_{q-1} + B_{q-1} z_1^{-1}) n z_1^n + A_{q-1} z_1^n] u(n - \tau_q) \\
&- [(\alpha_S z_1^{-P} + \beta_S z_1^{-(P+1)}) n z_1^n + (\alpha_S (1-P) z_1^{-P} - \beta_S P z_1^{-(P+1)}) z_1^n] u(n-P) \\
&= [(A_0 - \alpha_0) + (B_0 - \beta_0) z_1^{-1}] n z_1^n u(n) + (A_0 - \alpha_0) z_1^n u(n) \\
&\quad + \sum_{k=0}^{S-1} [(A_k - A_{k-1} - \alpha_k z_1^{-\tau_k}) + (B_k - B_{k-1} - \beta_k z_1^{-\tau_k}) z_1^{-1}] n z_1^n u(n - \tau_k) \\
&\quad + (A_k - A_{k-1} - \alpha_k (1 - \tau_k) z_1^{-\tau_k} + \beta_k \tau_k z_1^{-(\tau_k+1)}) z_1^n u(n - \tau_k) \\
&- [(A_{S-1} + \alpha_S z_1^{-P}) + (B_{S-1} + \beta_S z_1^{-P}) z_1^{-1}] n z_1^n u(n-P) \\
&- (A_{S-1} + \alpha_S (1-P) z_1^{-P} - \beta_S P z_1^{-(P+1)}) z_1^n u(n-P)
\end{aligned}$$

For the values of $0 \leq n < \tau_1$, equation (3.44) can be satisfied by setting

$$\alpha_0 = A_0 \quad (3.46)$$

$$\beta_0 = B_0 \quad (3.47)$$

The sufficient conditions for $\tau_k \leq \tau_{k+1}, k=1, 2, \dots, S-1,$

$$0 = (A_k - A_{k-1} - \alpha_k z_1^{-\tau_k}) + (B_k - B_{k-1} - \beta_k z_1^{-\tau_k}) z_1^{-1} \quad (3.48)$$

$$0 = (A_k - A_{k-1} - \alpha_k (1 - \tau_k) z_1^{-\tau_k}) + \beta_k \tau_k z_1^{-(\tau_k+1)} \quad (3.49)$$

are satisfied when

$$\alpha_k = (A_k - A_{k-1})(1 + \tau_k) z_1^{\tau_k} + (B_k - B_{k-1}) \tau_k z_1^{\tau_k - 1} \quad (3.50)$$

$$\beta_k = - (A_k - A_{k-1}) \tau_k z_1^{\tau_k + 1} + (B_k - B_{k-1})(1 - \tau_k) z_1^{\tau_k} \quad (3.51)$$

Finally for $n \geq \tau_S$, the sufficient equations

$$0 = (A_{S-1} + \alpha_S z_1^{-P}) + (B_{S-1} - \beta_S z_1^{-P}) z_1^{-1} \quad (3.52)$$

$$0 = (A_{S-1} + \alpha_S (1 - P) z_1^{-P}) - \beta_S P z_1^{-(P+1)} \quad (3.53)$$

hold when

$$\alpha_S = - [A_{S-1}(1 + P) z_1^P + B_{S-1} P z_1^{P-1}] \quad (3.54)$$

$$\beta_S = A_{S-1} P z_1^{P+1} - B_{S-1} (1 - P) z_1^P \quad (3.55)$$

ASSUME $z_1 \neq z_2$. Then $f(n) = \frac{[z_1^{n+1} - z_2^{n+1}]}{z_1 - z_2}$.

$$\begin{aligned} 0 = & \sum_{k=0}^{S-1} [A_k f(n) + B_k f(n-1)] [u(n - \tau_k) - u(n - \tau_{k+1})] \\ & - \sum_{r=0}^S [\alpha_r f(n - \tau_r) + \beta_r f(n - 1 - \tau_r)] \end{aligned} \quad (3.56)$$

$$\begin{aligned}
&= \sum_{k=0}^{S-1} [(A_k - \alpha_k z_1^{-\tau_k}) z_1 + (B_k - \beta_k z_1^{-\tau_k})] z_1^n u(n - \tau_k) \\
&\quad - [(A_k - \alpha_k z_2^{-\tau_k}) z_2 + (B_k - \beta_k z_2^{-\tau_k})] z_2^n u(n - \tau_k) \\
&\quad - \sum_{q=1}^S [(A_{q-1} z_1 + B_{q-1}) z_1^n - (A_{q-1} z_2 + B_{q-1}) z_2^n] u(n - \tau_q) \\
&\quad - [(\alpha_S z_1^{(1-P)} + \beta_S z_1^{-P}) z_1^n - (\alpha_S z_2^{(1-P)} + \beta_S z_2^{-P}) z_2^n] u(n - P) \\
&= [[(A_0 - \alpha_0) z_1 + (B_0 - \beta_0)] z_1^n + [(A_0 - \alpha_0) z_2 + (B_0 - \beta_0)] z_2^n] u(n) \\
&\quad + \sum_{k=1}^{S-1} [[(A_k - A_{k-1} - \alpha_k z_1^{-\tau_k}) z_1 + (B_k - B_{k-1} - \beta_k z_1^{-\tau_k})] z_1^n u(n - \tau_k) \\
&\quad - [[(A_k - A_{k-1} - \alpha_k z_2^{-\tau_k}) z_2 + (B_k - B_{k-1} - \beta_k z_2^{-\tau_k})] z_2^n u(n - \tau_k)] \\
&\quad - [(A_{S-1} + \alpha_S z_1^{-P}) z_1 + (B_{S-1} + \beta_S z_1^{-P})] z_1^n u(n - P) \\
&\quad + [(A_{S-1} + \alpha_S z_2^{-P}) z_2 + (B_{S-1} + \beta_S z_2^{-P})] z_2^n u(n - P) \quad .
\end{aligned}$$

For $0 \leq n < \tau_1$, the sufficient equations

$$0 = (A_0 - \alpha_0) z_1 + (B_0 - \beta_0) \quad (3.57)$$

$$0 = (A_0 - \alpha_0) z_2 + (B_0 - \beta_0) \quad (3.58)$$

are satisfied when

$$\alpha_0 = A_0 \quad (3.59)$$

$$\beta_0 = B_0 \quad (3.60)$$

For $\tau_k \leq n < \tau_{k+1}, k=1,2,\dots,S-1$, the equations

$$0 = (A_k - A_{k-1} - \alpha_k z_1^{-\tau_k}) z_1 + (B_k - B_{k-1} - \beta_k z_1^{-\tau_k}) \quad (3.61)$$

$$0 = (A_k - A_{k-1} - \alpha_k z_2^{-\tau_k}) z_2 + (B_k - B_{k-1} - \beta_k z_2^{-\tau_k}) \quad (3.62)$$

must be satisfied.

The solution to this coupled linear system is

$$\alpha_k = \frac{1}{(z_1 - z_2)} [(A_k - A_{k-1}) [z_1^{\tau_k+1} - z_2^{\tau_k+1}] + (B_k - B_{k-1}) [z_1^{\tau_k} - z_2^{\tau_k}]] \quad (3.63)$$

$$\beta_k = \frac{-z_1 z_2}{(z_1 - z_2)} [(A_k - A_{k-1}) [z_1^{\tau_k} - z_2^{\tau_k}] + (B_k - B_{k-1}) [z_1^{\tau_k-1} - z_2^{\tau_k-1}]] \quad (3.64)$$

Finally, for $n \geq P$, the defining equations

$$0 = (A_{S-1} + \alpha_S z_1^{-P}) z_1 + (B_{S-1} + \beta_S z_1^{-P}) \quad (3.65)$$

$$0 = (A_{S-1} + \alpha_S z_2^{-P}) z_2 + (B_{S-1} + \beta_S z_2^{-P}) \quad (3.66)$$

hold when

$$\alpha_S = \frac{1}{z_1 - z_2} [-A_{S-1} [z_1^{P+1} - z_2^{P+1}] - B_{S-1} (z_1^P - z_2^P)] \quad (3.67)$$

$$\beta_S = \frac{z_1 z_2}{z_1 - z_2} [A_{S-1} [z_1^P - z_2^P] + B_{S-1} [z_1^{P-1} - z_2^{P-1}]] \quad (3.68)$$

If $z_1 = z_2^* = |M|e^{j\phi}$ and $\phi \neq 0, \pi$, then the impulse response produced by this complex conjugate pair of poles is

$$f(n) = \frac{1}{z_1 - z_2} [z_1^{\tau_{k+1}} - z_2^{\tau_{k+1}}] = \frac{|M|^n}{\sin(\phi)} \sin[(n+1)\phi] u(n) \quad (3.69)$$

If equation (3.69) is substituted into equations (3.63)-(3.64), then

$$\alpha_k = \frac{|M|^{\tau_k}}{\sin(\phi)} [(A_k - A_{k-1}) \sin[(\tau_k + 1)\phi] + [(B_k - B_{k-1}) \sin[\tau_k \phi]] |M|^{-1}] \quad (3.70)$$

$$\beta_k = \frac{|M|^{\tau_k}}{\sin(\phi)} [-|M| (A_k - A_{k-1}) \sin[\tau_k \phi] - (B_k - B_{k-1}) \sin[(\tau_k - 1)\phi]] \quad (3.71)$$

***** q.e.d. *****

4. Symmetric Piecewise-Exponential Segments

Linear phase FIR filters have symmetric impulse responses. The number of multiplications needed to implement these filters can be cut in half by taking advantage of the special symmetry of the impulse response. For example, let

$$G(z) = \sum_{n=0}^{P-1} g(n) z^{-n} \quad (3.72)$$

be the impulse response sequence of length P . Reformulating $G(z)$ produces a special realization of the original transform function

$$G(z) = \left\{ \begin{array}{ll} \sum_{n=0}^{\frac{P-1}{2}} g(n) [z^{-n} + z^{P-1-n}] & , P \text{ even} \\ \sum_{n=0}^{\frac{P-3}{2}} g(n) [z^{-n} + z^{P-1-n}] + g(\frac{P-1}{2}) z^{\frac{-(P-1)}{2}} & , P \text{ odd} \end{array} \right\} \quad (3.73)$$

that can be implemented using approximately $P/2$ multiplications.

A similar factor of two savings can be obtained in a thinned filter implementation if the approximation sequence, $h(n)$, is required to be symmetrical and the basis exponentials, z_1^n and z_2^n , are required to have unit modulus.

Dynamic programming techniques can be used to find the optimal symmetric S -segment piecewise exponential approximation to a desired symmetrical sequence, $g(n)$. The initial portion of the symmetrical approximation sequence can be formulated using dynamic programming techniques to compute the optimum S_1 -segment piecewise-exponential fit to the first half of the desired symmetrical sequence $g(n), n=0,1,\dots,P_1$. Here,

$$S_1 = \begin{cases} (S+1)/2 & , S \text{ odd} \\ S/2 & , S \text{ even} \end{cases}$$

and

$$P_1 = \begin{cases} (P/2)-1 & , P \text{ even} \\ (P-3)/2 & , P \text{ odd} \end{cases}$$

The resulting S_1 -segment approximation sequence is then assumed to be symmetrical. For the special case of unit modulus poles, $z=z_1, z_2$, it can be shown that the thinned filter coefficients $\{\alpha_k\}$ and $\{\beta_k\}$, corresponding to the symmetrical approximation $h(n)$, also form symmetrical sequences. Consequently, the corresponding thinned filter can be implemented using approximately half of the $2(S+2)$ multiplications needed to implement a general thinned filter described in equation (2.10). In particular, a symmetrical thinned filter

implementation requires only $(S+3)$ multiplies for S odd, and $(S+4)$ multiplies for S even.

Thus, the thinned filter

$$H(z) = \frac{\sum_{k=0}^S [\alpha_k + \beta_k z^{-1}] z^{-\tau_k}}{(1-z_1 z^{-1})(1-z_2 z^{-1})} \quad (3.74)$$

will have an impulse response identical to the piecewise-exponential approximation found by the dynamic programming algorithm. It is interesting to note that it takes a maximum of $2(S+2)$ multipliers to implement this filter versus the P multiplications needed to realize the model filter, $G(z)$. Piecewise-exponential segments were used to produce several of the lowpass and bandpass thinned filters described in Chapter 5. The number of thinned filter multiplications can be cut in half if the piecewise approximation is required to be piecewise-continuous or symmetrical. Furthermore, a factor of 4 reduction can be obtained if the piecewise approximation is restricted to be piecewise-continuous and symmetrical.

CHAPTER IV - QUANTIZATION

A. FINITE-PRECISION ARITHMETIC ERRORS

The FIR-IIR cascade implementation of a thinned filter implicitly assumes that pole zero cancellation occurs, resulting in a finite-length impulse response filter, $H(z)$. This pole-zero cancellation would indeed occur if the digital filter were implemented using infinite-precision arithmetic. However, most digital filters are implemented on digital computers which are limited to finite-precision arithmetic. The lack of infinite-precision produces quantization errors in the representation of the filter coefficients and the arithmetic operations necessary to realize a filter.

The two most common forms of computer arithmetic are fixed-point, integer, and floating-point. These numerical notation schemes use a computer word of length b -bits to represent infinite-precision numbers. Finite-length registers constrain the computer encoded number to have a magnitude less than some arithmetic dependent number $K_{\max}$. For integer arithmetic $K_{\max} = 2^{b-1}$, for fixed-point arithmetic $K_{\max} = 1$, and $K_{\max} = 2^{(\text{maximum exponent})}$ for floating-point arithmetic. When multiplication or addition of two finite precision numbers results in a number greater than $K_{\max}$, a quantization error called overflow occurs. Note that multiplicative overflow can not occur in fixed-point arithmetic since the product of two fractions always results in a number less than one. Underflow errors occur when an arithmetic operation produces a small number whose magnitude is

greater than zero, yet smaller than $K_{\min}$, the smallest absolute value, other than zero, representable in a particular arithmetic. Multiplication of a b -bit encoded word by another b -bit word produces a product that could take $2b$ -bits to represent. Truncation errors occur when the least significant b -bits are discarded in the resulting product. Rounding errors occur when the least significant $(b+1)$ -bit is used to alter the value of the b most significant bits that are retained. Underflow, truncation, and rounding errors do not occur in integer arithmetic.

B. QUANTIZATION ERRORS IN DIGITAL FILTERS

Rounding and overflow are the quantization errors most commonly analyzed in the literature [12,16,18]. Rounding of the filter coefficients can significantly alter the frequency response of a digital filter [11]. The rounding errors also cause filter poles and zeros to deviate from their desired positions. In fact, quantization errors may move stable poles into unstable regions of the z^{-1} -plane. Kaiser [12] has formulated an estimate of the pole deviations as a function of pole positions for the case of an arbitrary order recursive lowpass filter. Rader and Gold [13] derived formulae for the pole deviations of second order recursive filters as a function of the magnitude, r , and the angle, θ , of the complex conjugate poles $z = re^{\pm j\theta}$. The resulting error bounds are extremely sensitive to poles near the real axis. Belal [14] has derived tighter bounds on quantization errors caused by poles with angles, θ/N , where N is an

integer. The special cases of $\theta=0, \pm\pi$ are included in his analysis.

Often filters are designed with poles exactly on the unit circle. In frequency-sampling filters [10,15], a comb filter is cascaded with a parallel bank of simple second-order resonators. Each resonator has a pair of complex conjugate poles. In the design stage, the complex conjugate poles of the resonator banks are chosen to coincide with the equally spaced unit circle zeros of the comb filter. Theoretically, pole-zero cancellation occurs in the cascade realization producing a stable, finite length, impulse response filter. The deviations of pole-zero locations resulting from quantization errors inhibit exact cancellation in practice. For this reason, the frequency-sampling filter is actually implemented with the unit radius poles and zeros moved slightly inside the unit circle. Decreasing the radii of the poles and zeros between $(1-2^{-12})$ and $(1-2^{-27})$ produces little change [10] in the behavior of the implemented frequency-sampling filter, while simultaneously providing some margin of safety from uncanceled pole positions occurring in unstable regions of the z^{-1} plane.

Thinned filters are analogous to frequency-sampling filters in that an all-zero filter is cascaded with a simple resonator whose poles are chosen to coincide with two of the zeros of the all-zero filter. Thus, many of the practical design procedures [15] of frequency-sampling filters may be used as a guide to the implementation of thinned filters.

Coefficient quantization is not the only source of roundoff errors in a digital filter. Each arithmetic operation necessary for filter implementation could result in roundoff error. Error analysis studies often treat roundoff errors as random variables with a word-length dependent probability density function centered at zero [11].

The output of a recursive digital filter,

$$Y(z) = \frac{\sum_{q=0}^M a_q z^{-q}}{1 + \sum_{k=1}^L b_k z^{-k}} = \frac{N(z)}{D(z)} \quad (4.1)$$

can be represented as a linear combination of the input samples, $x(n)$, and previous output samples, $y(n)$. That is,

$$y(n) = \sum_{q=0}^M a_q x(n-q) - \sum_{k=1}^L b_k y(n-k) \quad (4.2)$$

Assume that the quantized input has been normalized such that $x(n) \in (-1,1)$ and is stored in a B -bit word. Assume further that the filter coefficients $\{a_k, b_k\}$ are quantized to B' -bits. The product $[a_k x(n-k)]$ requires $(B+B')$ -bits to store. If this product is rounded to B bits, an error of magnitude $< 2^{-B}$ is introduced. This error, δ_n , is assumed to be a statistically independent, uniformly distributed, random variable with zero mean and variance

$$\text{var}(\delta_n) = \frac{2^{-2B}}{3} \quad (4.3)$$

If the input is properly scaled, and fixed-point arithmetic is used, then addition and subtraction will not produce any errors. Thus, the only filter errors are those generated by the $(M+L+1)$ multiplicative

quantization errors. Let

$$y'(n) = \sum_{q=0}^M a_q x(n-q) - \sum_{k=1}^L b_k y'(n-k) + \delta'_n \quad (4.4)$$

denote the actual computed filter output. δ'_n is the sum of the $(M+L+1)$ statistically independent random variables, δ_n . Thus, δ'_n is a statistically independent random variable with zero mean and variance

$$\text{var}(\delta'_n) = \frac{(M+L+1)2^{-2B}}{3} \quad (4.5)$$

Note that if a particular coefficient, a_k or b_k , is zero or one, then no roundoff is introduced in the product $[a_k x(n-k)]$ or $[b_k y'(n-k)]$, and the variance in equation (4.5) is reduced. For thinned filters $M=2(S+1)$ and $L=2$, and the total multiplication error δ'_n has a variance

$$\begin{array}{l} \text{var}(\delta'_n) \\ \text{thinned} \\ \text{filter} \end{array} = \frac{2(S+2)2^{-2B}}{3} \quad (4.6)$$

which is less than

$$\begin{array}{l} \text{var}(\delta'_n) \\ \text{model} \\ \text{filter} \end{array} = \frac{(P)2^{-2B}}{3}, \quad (4.7)$$

the multiplicative roundoff error variance generated by the model filter response, $g(n)$, of length P .

Another form of quantization error occurring in digital filters is overflow. Overflow errors can be inhibited by various scaling techniques [17]. These scaling strategies give some probabilistic guarantee that overflow will not occur by reducing the dynamic range of the filter. One of the most stringent scaling strategies uses an L_1 -norm formulation to guarantee that overflow will never occur. Let $r(z)$, defined in equation (4.1), be a desired digital filter. Let $1/D(z)$ have impulse response $d_n, n=0,1,\dots, \infty$. Then

$$y(n) = \sum_{q=0}^M a_q x(n-q) + \sum_{k=0}^{\infty} d_k x(n-k) \quad (4.8)$$

We want to guarantee that

$$|y(n)| < K_{\max}, \forall n \quad (4.9)$$

to avoid overflow errors. But

$$\begin{aligned} |y(n)| &= \left| \sum_{q=0}^M a_q x(n-q) + \sum_{k=0}^{\infty} d_k x(n-k) \right| \\ &\leq \left| \sum_{q=0}^M a_q x(n-q) \right| + \left| \sum_{k=0}^{\infty} d_k x(n-k) \right| \\ &\leq \sum_{q=0}^M |a_q| |x(n-q)| + \sum_{k=0}^{\infty} |d_k| |x(n-k)| \\ &\leq |x_{\max}| \left[\sum_{q=0}^M |a_q| + \sum_{k=0}^{\infty} |d_k| \right] \end{aligned} \quad (4.10)$$

where $|x_{\max}| \geq |x(n)|, \forall n$. Therefore

$$|x_{\max}| \leq \frac{K_{\max}}{\sum_{q=0}^M |a_q| + \sum_{k=0}^{\infty} |d_k|} \implies |y(n)| < K_{\max} \quad (4.11)$$

This scaling strategy formulation will be used to analyze an integer implementation of a thinned filter in a subsequent section.

C. INTEGER-ARITHMETIC IMPLEMENTATION OF THINNED DIGITAL FILTERS

Integer arithmetic operations can often be calculated faster than real arithmetic operations on a digital computer. However, designing digital filters with integer coefficients is an extremely difficult problem. One approach taken is to initially design a filter using real arithmetic. The resulting real coefficients are then somehow scaled and quantized to produce integer coefficients. The same strategy can be applied to produce thinned integer arithmetic digital filters. However, quantizing filter coefficients causes the poles and zeros to move from their original positions. Care must be taken to insure that the integer filter has zeros which cancel the poles of the second order recursive filter $1/D(z)$.

1. INTEGER FILTER COEFFICIENTS

Assume that $\{\alpha'_k\}, \{\beta'_k\}, k=0,1,\dots,S-1$ are integer coefficients obtained from quantizing the original thinned filter coefficients $\{\alpha_k\}, \{\beta_k\}, k=0,1,\dots,S-1$ in equations (3.41)-(3.43). Quantizing α_S and β_S will not necessarily produce a thinned numerator with zeros at the pole locations $z=z_1, z_2$. Recall that these two zeros were originally inserted into the unquantized numerator polynomial by forcing

the thinned filter impulse response, $h(n)$, to be zero for $n \geq P$. A similar strategy can be used to insert the desired zeroes into the numerator polynomial with quantized coefficients.

Therefore, given the integer coefficients $\{\alpha'_0, \alpha'_1, \dots, \alpha'_{S-1} ; \beta'_0, \beta'_1, \dots, \beta'_{S-1}\}$, α'_S and β'_S are found such that

$$0 = h'(n) = \sum_{k=0}^S [\alpha'_k f(n-\tau_k) + \beta'_k f(n-\tau_k-1)] \quad , \quad n \geq P \quad (4.12)$$

$$\text{Here, } f(n) = \begin{cases} \frac{1}{z_1 - z_2} [z_1^{n+1} - z_2^{n+1}] u(n) & , z_1 \neq z_2 \\ (n+1) z_1^n u(n) & , z_1 = z_2 \end{cases} ,$$

Equation (4.12) is satisfied when

$$\alpha'_S = \begin{cases} \left[- \sum_{j=0}^{S-1} [P(\alpha'_j z_1 + \beta'_j) + (\alpha'_j (1-\tau_j) z_1 - \beta'_j \tau_j)] z_1^{(P-1-\tau_j)} \right] , z_1 = z_2 & (4.13) \\ \left[\frac{- \sum_{j=0}^{S-1} [(\alpha'_j z_1 + \beta'_j) z_1^{(P-\tau_j)} - (\alpha'_j z_2 + \beta'_j) z_2^{(P-\tau_j)}]}{z_1 - z_2} \right] , z_1 \neq z_2 \\ \left[\frac{\sum_{j=0}^{S-1} (\alpha'_j |M| \sin[(P+1-\tau_j)\phi] + \beta'_j \sin[(P-\tau_j)\phi]) |M|^{(P-1-\tau_j)}}{\sin(\phi)} \right] , z_1 = z_2^* \\ \left[\frac{\sum_{j=0}^{S-1} (\alpha'_j |M| \sin[(P+1-\tau_j)\phi] + \beta'_j \sin[(P-\tau_j)\phi]) |M|^{(P-1-\tau_j)}}{\sin(\phi)} \right] , z_1 = |M| e^{j\phi} \end{cases}$$

$$\beta^-_S = \begin{cases} \left[\sum_{j=0}^{S-1} [(P-1)(\alpha^-_j z_1 + \beta^-_j) + (\alpha^-_j (1-\tau_j) z_1 - \beta^-_j \tau_j)] z_1^{(P-\tau_j)} \right], & z_1 = z_2 \quad (4.14) \\ \left[\frac{\sum_{j=0}^{S-1} (\alpha^-_j z_1 + \beta^-_j) z_2 z_1^{P-\tau_j} - (\alpha^-_j z_2 + \beta^-_j) z_1 z_2^{P-\tau_j}}{z_1 - z_2} \right], & z_1 \neq z_2 \\ \left[\frac{-\sum_{j=0}^{S-1} (\alpha^-_j |M| \sin[(P-\tau_j)\phi] + \beta^-_j \sin[(P-1-\tau_j)\phi]) |M|^{P-\tau_j}}{\sin(\phi)} \right], & z_1 = z_2 = |M| e^{j\phi} \end{cases}$$

***** PROOF *****

For $z_1 = z_2$, $f(n) = (n+1)z_1^n u(n)$.

$$0 = h^-(n) = \sum_{j=0}^S [\alpha^-_j (n+1-\tau_j) z_1^{n-\tau_j} + \beta^-_j (n-\tau_j) z_1^{(n-1-\tau_j)}] u(n-\tau_j), n \geq P \quad (4.15)$$

$$\begin{aligned} &= n z_1^n \sum_{j=0}^S [(\alpha^-_j z_1^{-\tau_j} + \beta^-_j z_1^{-(1+\tau_j)}) u(n-\tau_j)] \\ &\quad + z_1^n \sum_{k=0}^S [(\alpha^-_k (1-\tau_k) z_1^{-\tau_k} - \beta^-_k \tau_k z_1^{-(1+\tau_k)}) u(n-\tau_k)] . \end{aligned}$$

Thus, equation (4.15) is satisfied when

$$0 = \sum_{j=0}^S [(\alpha^-_j z_1^{-\tau_j} + \beta^-_j z_1^{-(1+\tau_j)}) u(n-\tau_j)], \quad n \geq P \quad (4.16)$$

$$0 = \sum_{k=0}^S [(\alpha^-_k (1-\tau_k) z_1^{-\tau_k} - \beta^-_k \tau_k z_1^{-(1+\tau_k)}) u(n-\tau_k)], \quad n \geq P \quad (4.17)$$

For $n \geq P$, equations (4.16) and (4.17) simplify to

$$\alpha'_S z_1^{-P} + \beta'_S z_1^{-(P+1)} = - \sum_{j=0}^{S-1} (\alpha'_j z_1^{-\tau_j} + \beta'_j z_1^{-(1+\tau_j)}) \quad (4.18)$$

$$\alpha'_S (1-P) z_1^{-P} - \beta'_S P z_1^{-(P+1)} = - \sum_{k=0}^{S-1} [\alpha'_k (1-\tau_k) z_1^{-\tau_k} - \beta'_k \tau_k z_1^{-(1+\tau_k)}] \quad (4.19)$$

However, equations (4.18) and (4.19) are merely two linear equations in two unknowns α'_S and β'_S . The solution

$$\alpha'_S = -z_1^P \sum_{j=0}^{S-1} [P(\alpha'_j + \beta'_j z_1^{-1}) + (\alpha'_j (1-\tau_j) - \beta'_j \tau_j z_1^{-1})] z_1^{-\tau_j} \quad (4.20)$$

$$\beta'_S = z_1^{P+1} \sum_{j=0}^{S-1} [(P-1)(\alpha'_j + \beta'_j z_1^{-1}) + (\alpha'_j (1-\tau_j) - \beta'_j \tau_j z_1^{-1})] z_1^{-\tau_j} \quad (4.21)$$

can be easily obtained.

For $z_1 \neq z_2$, $f(n) = \frac{1}{z_1 - z_2} [z_1^{n+1} - z_2^{n+1}] u(n)$. Once again, setting equation (4.12) equal to zero yields the formulation

$$\begin{aligned} 0 = h'(n) &= \sum_{j=0}^S [\alpha'_j f(n-\tau_j) + \beta'_j f(n-1-\tau_j)] \\ &= \sum_{j=0}^S [(\alpha'_j z_1^{1-\tau_j} + \beta'_j z_1^{-\tau_j}) z_1^n - (\alpha'_j z_2^{1-\tau_j} + \beta'_j z_2^{-\tau_j}) z_2^n] u(n-\tau_j) \end{aligned} \quad (4.22)$$

For $n \geq P$, equation (4.22) will be satisfied if

$$\alpha'_S z_1^{1-P} + \beta'_S z_1^{-P} = - \sum_{j=0}^{S-1} (\alpha'_j z_1 + \beta'_j) z_1^{-\tau_j} \quad (4.23)$$

$$\alpha'_S z_2^{1-P} + \beta'_S z_2^{-P} = - \sum_{k=0}^{S-1} (\alpha'_k z_2 + \beta'_k) z_2^{-\tau_k} \quad (4.24)$$

The values α'_S and β'_S needed to satisfy equations (4.23)-(4.24) are

$$\alpha'_S = \frac{1}{z_1 - z_2} \sum_{j=0}^{S-1} [(-\alpha'_j z_1 + \beta'_j) z_1^{P-\tau_j} + (\alpha'_j z_2 + \beta'_j) z_2^{P-\tau_j}] \quad (4.25)$$

$$\beta'_S = \frac{1}{z_1 - z_2} \left[\sum_{j=0}^{S-1} (\alpha'_j z_1 + \beta'_j) z_2 z_1^{P-\tau_j} - (\alpha'_j z_2 + \beta'_j) z_1 z_2^{P-\tau_j} \right]. \quad (4.26)$$

For $z_1 = z_2^* = |M|e^{j\phi}$, equations (4.13)-(4.14) simplify to

$$\alpha'_S = \frac{-1}{\sin(\phi)} \sum_{j=0}^{S-1} (\alpha'_j |M| \sin[(P+1-\tau_j)\phi] + \beta'_j \sin[(P-\tau_j)\phi]) |M|^{P-1-\tau_j}. \quad (4.27)$$

$$\beta'_S = \frac{1}{\sin(\phi)} \left[\sum_{j=0}^{S-1} (\alpha'_j |M| \sin[(P-\tau_j)\phi] + \beta'_j \sin[(P-1-\tau_j)\phi]) |M|^{P-\tau_j} \right] \quad (4.28)$$

***** q.e.d. *****

Equations (4.13) and (4.14) will not necessarily produce integer values for α'_S and β'_S for arbitrary poles, z_1 and z_2 . However, if the quantized poles are restricted to be a repeated real or a purely imaginary pole pair with integer modulus, then α'_S and β'_S are purely integer. In these restricted cases, equations (4.13)-(4.14) simplify to

$$\alpha'_S = \begin{cases} - \sum_{j=0}^{S-1} [P(\alpha'_j z_1 + \beta'_j) - (\alpha'_j (1-\tau_j) z_1 - \beta'_j \tau_j)] z_1^{(P-1-\tau_j)}, & z_1 = z_2 \\ \sum_{j=0}^{S-1} [\alpha'_j |M| \Psi(P+1-\tau_j) + \beta'_j \Psi(P-\tau_j)] |M|^{(P-1-\tau_j)}, & z_1 = |M|e^{j\phi} \end{cases} \quad (4.29)$$

$$\beta'_S = \begin{cases} \sum_{j=0}^{S-1} [(P-1)(\alpha'_j z_1 + \beta'_j) + (\alpha'_j (1-\tau_j) z_1 - \beta'_j \tau_j)] z_1^{(P-\tau_j)}, & z_1 = z_2 \\ - \sum_{j=0}^{S-1} [\alpha'_j |M| \Psi(P-\tau_j) + \beta'_j \Psi(P-1-\tau_j)] |M|^{(P-\tau_j)}, & z_1 = z_2^* = |M|e^{j\phi} \end{cases} \quad (4.30)$$

Here,

$$\psi(n) = \sin[n(\pi/2)] = \left\{ \begin{array}{ll} 0, & \text{mod}_4(n)=0,2 \\ 1, & \text{mod}_4(n)=1 \\ -1, & \text{mod}_4(n)=3 \end{array} \right\} . \quad (4.31)$$

Thus, for the restricted case given in equations (4.29)-(4.30), a thinned filter with integer coefficients will have zeros at $z=z_1$ and $z=z_2$. Rounding errors do not occur in integer arithmetic. Only the occurrence of overflow produces errors in integer coefficient filter implementation. However, since the thinned filter structure is realized as $N(z)=H(z)D(z)$ cascaded with $F(z)=1/D(z)$, a scaling strategy can be devised which prevents the possibility of overflow.

2. MINIMUM WORD LENGTH REQUIREMENTS

Assume the filter configuration shown in Figure 4-1. Let $H'(z)$ be the integer coefficient thinned filter with impulse response, $h'(n)$ defined in equation (4.12). Let $F'(z)=1/D'(z)$ be the IIR

filter with restricted poles leading to equations (4.29)-(4.30).

FIGURE 4-1. A cascade implementation of the thinned integer coefficient FIR filter $N'(z)=H'(z)D'(z)$ and the IIR filter

$$F'(z) = \frac{1}{D'(z)} = \frac{1}{1-b_1z^{-1}-b_2z^{-2}}.$$

Let $X(z)$, $Y_1(z)$, $Y_2(z)$, and $Y_3(z)$ denote the Z-transforms of $x(k)$, $y_1(k)$, $y_2(k)$, and $y_3(k)$, respectively. Ideally,

$$\begin{aligned} Y_3(z) &= X(z)H'(z)D'(z)\left(\frac{1}{D'(z)}\right) \\ &= X(z)H'(z) \\ &= Y_1(z) \end{aligned} \tag{4.32}$$

This theoretical result will hold only if overflow does not occur in the filter implementation. If z_1 and z_2 are unstable poles, one would theoretically need infinite length computer words to implement the IIR section $1/D'(z)$. The following analysis takes advantage

of the fact that $N'(z)$ contains $D'(z)$, and calculates a reasonable finite bound on the minimum word-length needed to implement this filter without overflow.

Let $h'(j)$ and $n'(j)$ be the inverse Z-transform of $H'(z)$ and $N'(z)$ respectively. It is not unreasonable to assume that one has some a priori bound

$$(1) \quad |x(j)| < 2^{B_x-1}, \forall j \quad (4.33)$$

$$(2) \quad |h'(j)| < 2^{B_h-1}$$

$$(3) \quad |n'(j)| < 2^{B_n-1}$$

$$(4) \quad |b'_i| < 2^{B_D-1}, i=1,2$$

which can be used to establish L_1 bounds on

$$(5) \quad |y_1(j)| < 2^{B_{y_1}-1}, \forall j \quad (4.34)$$

$$(6) \quad |y_2(j)| < 2^{B_{y_2}-1}.$$

Suppose it is known that

$$|y_3(j)| < 2^{B_{y_3}-1}, \forall j \quad (4.35)$$

Using the recursive filter formulation of $F'(z)$, one can obtain the identity

$$y_3(k) = y_2(k) - b'_1 y_3(k-1) - b'_2 y_3(k-2) \quad (4.36)$$

Taking the absolute value of both sides of this equation yields:

$$\begin{aligned}
|y_3(k)| &= |y_2(k) - b_1 y_3(k-1) - b_2 y_3(k-2)| \\
&\leq |y_2(k)| + |b_1| |y_3(k-1)| + |b_2| |y_3(k-2)| \\
&< 2^{(B_D-1)} [2^{(B_{y_2}-1)} + (2)2^{(B_{y_3}-1)}] \\
&\leq 2^{(B_{y_2}+B_D-2)} + 2^{(B_{y_3}+B_D-1)} \\
&\leq 2^{\max[B_{y_2}-1, B_{y_3}]+B_D} \\
&= 2^{B-1}
\end{aligned} \tag{4.37}$$

where $B = \max[B_{y_2}-1, B_{y_3}]+B_D+1$.

Therefore, the filter $1/D^*(z)$ can be implemented in B -bit integer arithmetic without overflow, if the bounds, B_{y_2} and B_{y_3} , are known. One of these bounds, B_{y_2} , is relatively straight forward to calculate.

$$\begin{aligned}
|y_2(k)| &= \left| \sum_{i=0}^{P+1} n^*(i)x(k-i) \right| \leq \sum_{i=0}^{P+1} |n^*(i)| |x(k-i)| \\
&< 2^{(B_x-1)} \sum_{i=0}^{P+1} |n^*(i)| \\
&\leq 2^{B_{y_2}-1}
\end{aligned} \tag{4.38}$$

The minimum value for B_{y_2} is therefore given by the expression:

$$2^{B_{y_2}-1} \leq 2^{B_x} \sum_{k=0}^{P+1} |n^*(i)| < 2^{B_{y_2}} \tag{4.39}$$

If these minimum word lengths, B_{y_2} and B_x , are used to implement the FIR and IIR sections respectively, then no overflow will occur. Furthermore, if no overflow errors are made then

$$y_1(k) = y_3(k), \quad \forall k. \quad (4.40)$$

But

$$\begin{aligned} |y_1(k)| &= \left| \sum_{i=0}^{P-1} h'(i)x(k-i) \right| \leq \sum_{i=0}^{P-1} |h'(i)| |x(k-i)| \\ &< 2^{(B_x-1)} \sum_{i=0}^{P-1} |h'(i)| \\ &\leq 2^{B_{y_1}-1}. \end{aligned} \quad (4.41)$$

Therefore, establishing an upper bound for $|y_1(k)|$ automatically defines a value for the minimum word length needed to correctly implement the filter $H'(z)$. In particular, B_{y_1} must satisfy the following inequalities :

$$2^{B_{y_1}-1} \leq 2^{B_x} \sum_{i=0}^{P-1} |h'(i)| < 2^{B_{y_1}} \quad (4.42)$$

Since $y_1(k) = y_3(k)$, then $|y_1(k)| < 2^{B_{y_1}-1}$ implies that

$$|y_3(k)| < 2^{B_{y_1}-1}.$$

Therefore, an upper bound has been established on the filter output, $y_3(k)$, by taking advantage of the special structure of the thinned filter and the special properties of integer arithmetic. If the FIR section $N'(z)$ is implemented in B_{y_2} -bit arithmetic, where

$$B_{y_2} = B_x + \Psi\left(\sum_{i=0}^{P-1} |h'(i)|\right) \quad (4.43)$$

and the IIR section $F'(z)$ is implemented in

$$B = \max[B_{y_2}-1, B_{y_1}] + B_D + 1 \quad (4.44)$$

bit arithmetic, then no overflow will occur in the thinned filter implementation. Here,

$$\Psi(n) = m \quad (4.45)$$

where m is an integer such that $2^{m-1} < n \leq 2^m$.

For general $H'(z)$, a worst-case scaling strategy can be found. Recall,

$$\begin{aligned} |y_1(k)| &< 2^{(B_x-1)} \left(\sum_{k=0}^{P-1} |h'(i)| \right) < 2^{B_x-1} [2^{B_h-1} P] \\ &\leq 2^{(B_x+B_h+\Psi(P)-2)}. \end{aligned} \quad (4.46)$$

Therefore, $y(k)$ can be generated using

$$B_{y_1} = [B_x + B_h - 1 + \Psi(P)] \quad (4.47)$$

bit arithmetic. Furthermore,

$$|y_2(n)| < 2^{(B_x-1)} \sum_{i=0}^{P+1} |n'(i)| \quad (4.48)$$

But there are only $2(S+1)$ non-zero coefficients, $n'(i)$, in the thinned filter implementation. Therefore,

$$\begin{aligned}
 |y_2(n)| &< 2^{(B_x-1)} (2^{(B_n-1)})^{2(S+1)} \\
 &< 2^{(B_x+B_n+2(S+1))-1}
 \end{aligned} \tag{4.49}$$

which requires a maximum of

$$[B_x + B_n + 2(S+1)] \tag{4.50}$$

-bits to implement. For example, assume

$$\begin{aligned}
 B_x &= 12 \text{ bit integer input } x(k) \\
 B_h &= 10 \text{ bit representation of } h'(k) \\
 B_n &= 8 \text{ bit integer coefficients } \alpha'_k \text{ and } \beta'_k \\
 B_D &= 2 \text{ bit quantized coefficients of } D'(z) .
 \end{aligned}$$

Several of the example thinned filters in chapter 5 use $P=65$ and $S=5$. Plugging in these filter specifications into equations (4.47), (4.30), and (4.44), produces the worst-case minimum word lengths

$$\begin{aligned}
 B_{y_1} &= 12+10-1+ 2(65) = 28 \text{ bits} \\
 B_{y_2} &= 12+8 + 2(6) = 23 \text{ bits} \\
 B &= \max[23-1, 28]+2+1 = 31 \text{ bits}
 \end{aligned}$$

necessary to implement the quantized integer coefficient thinned filter without overflow errors.

For $P=101$ and $S=10$,

$$B_{y_1} = 21 + \psi(101) = 28 \text{ bits}$$

$$B_{y_2} = 20 + \psi(10) = 24 \text{ bits, and}$$

$$B = \max[24-1, 28] + 2 + 1 = 31 \text{ bits .}$$

Therefore, thinned filters with these specifications could be exactly implemented in 32-bit arithmetic. Even the most stringent worst-case analysis produces reasonable minimum word length criteria for implementing integer arithmetic thinned filters without overflow.

CHAPTER V - EXAMPLES

A. NARROWBAND LOWPASS FILTERS

The thinned filter algorithm is used to approximate a narrowband lowpass filter designed using the Remez Exchange Algorithm [9]. In the first example, the model impulse response $g(n)$ is of length $P=101$. The passband edge corresponds to the normalized frequency, $F_p=0.0$; the stopband edge is $F_s=.01$. The dynamic programming thinning algorithm is used to find the best $S=5$ segment piecewise-linear approximation. The impulse responses of the model filter and the thinned filter are shown in Figure 5-1. Note that the large impulse response coefficients $g(0)$ and $g(P-1)$ are well approximated by the thinned filter impulse response. These particular coefficients weight the fastest varying cosine term used to produce the frequency responses plotted in Figure 5-2. Both the model and the thinned filter have a very sharp cutoff in the transition region between F_p and F_s . The thinned filter stopband deviation is only slightly worse than that of the model filter. Furthermore, the thinned filter requires only 13 multiplies to implement. The original filter requires $P=101$ multiplications to implement in the direct form; 51 multiplications are required to implement the model filter in a method that takes advantage of the symmetry of $g(n)$. This corresponds to a 1 to 4 ratio in the number of multiplications necessary for implementing the thinned filter versus the model filter.

The second example is also a lowpass filter designed with a wider transition region to allow greater stopband attenuation. The length $P=65$ model impulse response is generated by the Remez-Exchange algorithm using a passband edge of $F_p=.001$ and a stopband edge $F_s=.025$ as design criteria. The model and the $S=5$ segment piecewise-linear approximation are shown in Figure 5-3. The corresponding frequency responses are depicted in Figure 5-4. Once again, the approximate filter produces a sharp-cutoff region and loses only 3 db of attenuation in the stopband. The ratio of thinned filter multiplications to model filter multiplications is 13 to 33 or slightly less than $1/3$.

The thinned filter coefficient taps $\{\alpha_k\}$ and $\{\beta_k\}$ are displayed in Figure 5-5 and 5-6 respectively. These coefficients produce a thinned numerator polynomial $N(z)$ whose 67 roots are plotted in Figure 5-7. Figure 5-8 displays the 65 zeros of the original model filter. Note that the thinned numerator plot has a double root at $z=1$ to cancel the poles $z_1=z_2=1$ used in the IIR section $F(z)$. Figure 5-9 shows the results of moving the design poles inside the unit circle to $z_1=z_2=.99$. This permits the pole deviations produced by quantization errors to lie within the stable regions of the z -plane. The frequency response of this stable pole implementation is shown in Figure 5-9.

Another quantization analysis example is shown in Figures 5-10 and 5-11. The thinned coefficient taps $\{\alpha_k\}$ and $\{\beta_k\}$ are scaled to have a maximum absolute value of 127. The fractional parts of the scaled coefficients are then rounded to produce purely integer coefficients that could be stored in an 8-bit integer word. Since the poles $z_1=z_2=1$ were repeated real poles on the unit circle, equations (4.29) and (4.30) could be invoked to calculate the integer coefficients, α'_S and β'_S , necessary to insert a double zero in the thinned numerator. The integer taps produces the impulse response and frequency response indicated in Figures 5-10 and 5-11 respectively. Even though this filter has unstable poles, the minimum word length criteria given in equations (4.39) and (4.44) can be used to guarantee pole-zero cancellation and consequently a stable, finite length, impulse response filter.

B. WIDEBAND LOWPASS FILTER

The thinning algorithm also works well on wideband filters. Figure 5-12 shows the length $P=101$ impulse response of a lowpass filter designed with $F_p=.25$ and $F_s=.26$. The pole locations, $z_1=z_2^*=e^{j\pi/2}$ are chosen to coincide with the passband edge. These poles are used to find the best 10-segment piecewise-sinusoidal approximation, also shown in Figure 5-12. Both impulse responses produce sharp-cutoff frequency responses illustrated in Figure 5-13. The stopband attenuation of the thinned filter exceeds the minimum stopband attenuation of the equal-ripple filter by less than 2 dB.

The thinned filter implementation requires less than half the number of multiplications needed to realize the original lowpass filter.

C. NARROWBAND BANDPASS FILTER

The final example is that of a length $P=101$ narrowband bandpass filter with center frequency $F_c=.25$ and bandwidth $=.02$. The poles $z_1=z_2^*=e^{j\pi/2}$ are chosen to correspond to the center frequency of the model filter. $S=10$ segments were used in this approximation example. The impulse responses of the original bandpass filter and the 10-segment piecewise-sinusoidal approximation are shown in Figure 5-14. Figure 5-15 demonstrates the remarkable agreement between the frequency responses of the two filters. The minimum stopband attenuation level deviates from the original equal-ripple attenuation by less than 1.5 dB. Again, the multiplications needed are reduced by a factor of 2. Quantizing the numerator coefficients to 8-bits causes some of the $\{\alpha'_k\}$ and $\{\beta'_k\}$ coefficients to equal zero in this example. In fact, only eleven 8-bit integer coefficients are necessary to generate the integer coefficient thinned filter frequency response shown in Figure 5-16. The original bandpass filter has its zeros plotted in Figure 5-17; the thinned integer coefficient filter has two extra zeros, shown in Figure 5-18, which are located at the design pole locations of $z_1=z_2^*=e^{j\pi/2}$. The model bandpass filter is also approximated with an exponentially weighted sinusoid corresponding to the poles $z_1=z_2^*=.99e^{j\pi/2}$. The resulting thinned filter has stable poles if sufficient word-length registers are used. The

piecewise-exponential approximation is shown in Figure 5-19. The corresponding frequency responses are shown in Figure 5-20. Again, close agreement exists between the model filter and the thinned filter with poles moved slightly inside the unit circle.

FIGURE 5-1. — Impulse response of a narrowband lowpass filter designed using $F_p=0.0$ and $F_s=0.01$ band edges in the Remez Exchange Algorithm.
 5-Segment Piecewise-linear approximation sequence

FIGURE 5-2. — Frequency response of a narrowband lowpass filter designed using $F_p=0.0$ and $F_s=0.01$ band edges in the Remez Exchange Algorithm.

— Frequency response of an $S=5$ segment piecewise-linear approximation sequence.

FIGURE 5-3. Narrowband Lowpass Filter

— Impulse response of a length $P=65$ equal ripple lowpass FIR filter designed using the Remez Exchange algorithm with $F_p=0.001$ and $F_s=0.025$.

..... $S=5$ segment piecewise-linear approximation sequence

FIGURE 5-4. **Narrowband Lowpass Filter**

— Frequency response of a length $P=65$ equal ripple lowpass FIR filter designed using the Remez Exchange algorithm with $F_p=0.001$ and $F_s=0.025$.

... Frequency response of an $S=5$ segment piecewise-linear approximation sequence.

FIGURE 5-5. Numerator tap coefficients $\{a_k\}$ of the optimum 5-segment piecewise-linear approximation filter shown in Figure 5-4.

FIGURE 5-6. Numerator tap coefficients $\{s_k\}$ of the optimum 5-segment piecewise-linear approximation filter shown in Figure 5-4

FIGURE 5-7. Zeros of the thinned narrowband lowpass filter shown in Figure 5-4.

FIGURE 5-8. Zeros of the narrowband lowpass equal ripple filter shown in Figure 5-4.

FIGURE 5-9. Narrowband Lowpass Filter

— Frequency response of a length $P=65$ equal ripple lowpass FIR filter designed using the Remez Exchange algorithm with $F_p=0.001$ and $F_s=0.025$.

— Frequency response of an $S=5$ segment piecewise-exponential approximation sequence generated with $f(n)=(n+1)(.99)^n u(n)$.

FIGURE 5-10.

Narrowband Lowpass Filter

—— Impulse response of a length $P=65$ equal ripple lowpass FIR filter designed with $F_p=0.001$ and $F_s=0.025$.

..... $S=5$ segment piecewise-linear approximation impulse response sequence of a thinned filter with integer coefficients quantized to 8-bits.

FIGURE 5-11. — Frequency response of a length $P=65$ equal ripple lowpass FIR filter designed using the Remez Exchange algorithm with $F_p=.001$ and $F_s=0.025$.
 — Frequency response of the 5-segment thinned filter with integer coefficients quantized to 8-bits.

FIGURE 5-12. Broadband Lowpass Filter

- impulse response of a length $P=101$ equal ripple broadband lowpass FIR filter with $F_p=.25$ and $F_s=.26$.
- $S=10$ segment piecewise-sinusoidal approximation sequence

FIGURE 5-13 Broadband Lowpass Filter

- Frequency response of a length $P=101$ equal ripple broadband lowpass FIR filter designed with $F_p=.25$ and $F_s=.26$.
- - - Frequency response of the $S=10$ segment thinned filter.

FIGURE 5-14. Narrowband Bandpass Filter

— impulse response of a length $P=101$ equal ripple narrowband bandpass FIR filter.

— $S=10$ segment piecewise-sinusoidal approximation sequence

FIGURE 5-15. — Frequency response of a length $P=101$ equal ripple narrowband bandpass FIR filter designed with center frequency, $F_c=.25$ and total bandwidth $=0.02$.
 - - - Frequency response of the $S=10$ segment approximation.

FIGURE 5-16. — Frequency response of an equal ripple bandpass filter designed with $P=101$, $F_c=.25$, and bandwidth $=0.02$
 — Frequency response of the $S=10$ segment approximation generated using 8-bit coefficients in the thinned filter.

FIGURE 5-17. Zeros of the equal ripple bandpass filter shown
in Figure 5-15

FIGURE 5-18. Zeros of the 8-bit integer coefficient narrowband
bandpass thinned filter described in Figure 5-16.

FIGURE 5-19.

Narrowband Bandpass Filter

— impulse response of an equal ripple narrowband bandpass FIR filter of length $P=101$.

— 10 segment approximation sequence designed with complex conjugate poles at $z = \pm .99j$.

FIGURE 5-20. — Frequency response of the equal ripple bandpass filter described in Figure 5-15.

— Frequency response of the 10 segment approximation sequence designed with conjugate poles at $z = \pm .99j$.

CHAPTER VI - CONCLUSIONS

An algorithm has been presented which finds the best S segment piecewise-exponential approximation to a desired impulse response of length P . Realizing the finite length approximation as the impulse response of an IIR filter cascaded with a thinned FIR filter reduced the number of multiplies necessary for filtering to a maximum of $2(S+2)$ per output sample. When the approximation is restricted to be piecewise-continuous, or symmetric, only $(S+4)$ multiplies were needed. A symmetrical piecewise-continuous approximation requires only $(S/2 + 3)$ multiplications. When the tap coefficients $\{\alpha_k, a_k\}$ were quantized to integer coefficients, some became zero, further reducing the required number of multiplies. Minimum word-length requirements prohibiting overflow in integer arithmetic implementation were derived. Examples of the approximation procedure for lowpass and bandpass filters were generated with little difference in the frequency response of the desired and thinned filters. The thinned filter implementation required from $1/2$ to $1/4$ the number of multiplications needed to implement the original model filter. Several examples were generated with the design poles moved inside the unit circle in case quantization errors prevented exact pole-zero cancellation. The results obtained were also in close agreement with the frequency response of the original model filter.

The algorithm seemed to work best on filters with sharp cut-offs. The question of ideal pole locations has not been solved. Good results were obtained when the poles were placed at locations corresponding to the center frequency of narrowband filters and the passband edge of lowpass filters. Computer algorithms, other than dynamic programming, exist which can be used to find the optimum S-segment piecewise-exponential approximation to a given sequence. These algorithms may prove to be more efficient to implement. In particular, an expedient algorithm is needed to solve the case of piecewise-continuous segment approximation.

REFERENCES

- (1) Bellman, R., Dynamic Programming, Princeton University Press
Princeton, N.J., (1957)
- (2) Skolnik, M.I. and Sherman, J.W., "Planar Arrays with Unequal-
ly Spaced Elements," The Radio and Electronic Engineer,
pp.173-184 (Sept. 1964)
- (3) Skolnik, M.I., Nemhauser, G., and Sherman, J.W., "Dynamic Pro-
gramming Applied to Unequally Spaced Arrays," IEEE Transac-
tions on Antennas and Propagation, pp.35-43 (Jan. 1964)
- (4) Pruzan, P.M. and Munch-Anderson, B., "On the Application of
Dynamic Programming-Type Algorithms to Antenna Design,"
SIAM Journal on Applied Mathematics, Vol. 15(5), pp.1113-
1129 (Sept. 1967).
- (5) Arora, R.K. and Krishnamacharyulu, N.C.V., "Synthesis of
Unequally Spaced Arrays Using Dynamic Programming,"
IEEE Transactions on Antennas and Propagation, pp.593-
595 (Sept. 1968).
- (6) Schussler, H.W., "Design of a Link-Type Matched Filter,"
Research Report EE 557, pp.68-83, Cornell University
(Feb. 1963).
- (7) Bellman, R., "On the Approximation of Curves by Line Segments
Using Dynamic Programming," Communications of the ACM, p.284
(June 1961).
- (8) Bellman, R. and Roth, R., "Curve Fitting By Segmented Straight
Lines," American Statistical Association Journal, pp.1079-
1084 (Sept. 1969).
- (9) McClellan, J.H., Parks, T.W., and Rabiner, L.R., "A Computer
Program for Designing Optimum FIR Linear Phase Digital Fil-
ters," IEEE Transactions on Audio Electroacoustics, Vol. AU-21,
pp.506-526 (Dec. 1973).

- (10) Gold,B. and Rader,C.M., Digital Processing of Signals, McGraw-Hill, Inc.,(1969).
- (11) Peled,A., and Liu, B., Digital Signal Processing, John Wiley & Sons, Inc.,(1976).
- (12) Kaiser,J.F., "Some Practical Considerations in the Realization of Linear Digital Filters," Proceedings 3rd Annual Allerton Conference on Circuit and System Theory, pp.621-633 (1965).
- (13) Rader,C.M., and Gold,B., "Effects of Parameter Quantization on the Poles of a Digital Filter," Proceedings IEEE, Vol. 55 (5), pp.688-689 (May 1967).
- (14) Belal, A.A., "On the Quantization Error Bounds in Second-Order Digital Filters with Complex Conjugate Poles," IEEE Transactions on Circuit and Systems, CAS-24 (1), p.45 (Jan. 1977) .
- (15) Weinstein,C.J. "Quantization Effects in Digital Filters," Technical Report 468, Lincoln Lab, M.I.T., Lexington, Mass.,(Nov 1969).
- (16) Claasen,T.A.C.M., Mecklenbrauker,W.F.G., and Peek,J.B.H., "Effects of Quantization and Overflow in Recursive Digital Filters," IEEE Transactions ASSP, Vol ASSP-24 (6),pp.517-529 (Dec. 1976).
- (17) Claasen,T.A.C.M., and Kristiansson,L.O.G., "Necessary and Sufficient Conditions for the Absence of Overflow Phenomena in a Second-Order Recursive Digital Filter,"IEEE Transactions ASSP, Vol. ASSP-23 (6), pp.509-515 (Dec. 1975) .
- (18) Jackson,L.B., "On the Interaction of Round-Off Noise and Dynamic Range in Digital Filters," Bell System Technical Journal, Vol. 49 (2), pp.159-184 (Feb. 1970).
- (19) Oppenheim,A.V., and Schafer,R.W., Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1975).
- (20) Davis,P.J., Interpolation & Approximation, Dover Publications Inc., New York (1975) .

- (21) Bellman, R.E., and Dreyfus, S.E., Applied Dynamic Programming, Princeton University Press, Princeton, N.J. (1962).