Portland State University
PDXScholar

Dissertations and Theses | Dissertations and Theses

5-7-1993

Designing and Simulating a Multistage Sampling Rate Conversion System Using a Set of PC Programs

David Joseph Hagerty
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Part of the Electrical and Electronics Commons
Let us know how access to this document benefits you.

Recommended Citation
Hagerty, David Joseph, "Designing and Simulating a Multistage Sampling Rate Conversion System Using a Set of PC Programs" (1993). *Dissertations and Theses.* Paper 4697.
https://doi.org/10.15760/etd.6581

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

AN ABSTRACT OF THE THESIS OF David Joseph Hagerty for the Master of Science in Electrical Engineering presented May 7, 1993.

Title: Designing and Simulating a Multistage Sampling Rate Conversion System Using a Set of PC Programs

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[redacted]

Y.C. Jenq, Chair

[redacted]

Richard Tymerski

[redacted]

Gavin Bjork

The thesis covers a series of PC programs that we have written that will enable users to easily design FIR linear phase lowpass digital filters and multistage sampling rate conversion systems. The first program is a rewrite of the McClellan-Parks computer program with some slight modifications. The second program uses an algorithm proposed by Rabiner that determines the length of a lowpass digital filter. Rabiner used a formula proposed by Herrmann *et al.* to initially estimate the filter length in his algorithm. The formula, however, assumes unity gain. We present a modification to the formula so that the gain of the filter is normalized to accommodate filters that have a gain greater than one (as in the case of a lowpass filter used in an interpolator). We have also changed the input specifications from digital to analog. Thus, the user supplies the sampling rate, passband frequency,

stopband frequency, gain, and the respective maximum band errors. The program converts the specifications to digital. Then, the program iteratively estimates the filter length and interacts with the McClellan-Parks Program to determine the actual filter length that minimizes the maximum band errors. Once the actual length is known, the filter is designed and the filter coefficients may be saved to a file.

Another new finding that we present is the condition that determines when to add a lowpass filter to a multistage decimator in order to reduce the total number of filter taps required to implement the system. In a typical example, we achieved a 34% reduction in the total required number of filter taps.

The third program is a new program that optimizes the design of a multistage sampling rate conversion system based upon the sum of weighted computational rates and storage requirements. It determines the optimum number of stages and the corresponding upsampling and downsampling factors of each stage of the design. It also determines the length of the required lowpass digital filters using the second program.

Quantization of the filter coefficients may have a significant impact on the frequency response. Consequently, we have included a routine within our program that determines the effects of such quantization on the allowable error margins within the passband and stopband. Once the filter coefficients are calculated, they can be saved to files and used in an appropriate implementation. The only requirements of the user are the initial sampling rate, final sampling rate, passband frequency, stopband frequency, corresponding maximum errors for each band, and the weighting factors to determine the optimization factor.

We also present another new program that implements a sampling rate conversion from CD (44.1 kHz) to DAT (48 kHz) for digital audio. Using the third program to design the filter coefficients, the fourth program converts an input

sequence (either samples of a sine wave or a unit sample sequence) sampled at the lower rate to an output sequence sampled at the higher rate. The frequency response is then plotted and the output block may be saved to a file.

DESIGNING AND SIMULATING A MULTISTAGE SAMPLING RATE CONVERSION SYSTEM USING A SET OF PC PROGRAMS

by

DAVID JOSEPH HAGERTY

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

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University
1993

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of David Joseph Hagerty presented May 7, 1993.

[redacted]

Y.C. Jenq, Chair

[redacted]

Richard Tymerski

[redacted]

Gavin Bjork

APPROVED:

[redacted]

Rolf Schaumann, Chair, Department of Electrical Engineering

[redacted]

Roy W. Koch, Vice Provost for Graduate Studies and Research

TABLE OF CONTENTS

PAGE

LIST OF TABLES. v

LIST OF FIGURES. vi

INTRODUCTION. 1

DESIGN OF AN FIR LINEAR PHASE LOWPASS DIGITAL FILTER OF KNOWN LENGTH. 4

DESIGN OF AN FIR LINEAR PHASE LOWPASS DIGITAL FILTER OF UNKNOWN LENGTH. 9

REVIEW OF SOME FUNDAMENTAL DIGITAL SIGNAL PROCESSING CONCEPTS. 16

Sampling Rate Conversion. 16

Decimation by an Integer Factor M
Interpolation by an Integer Factor L
Conversion by a Rational Factor L/M

Implementation of a One-Stage Sampling Rate Conversion System Using FIR Structures. 22

An Efficient FIR Structure for Decimation by an Integer Factor M
An Efficient FIR Structure for Interpolation by an Integer Factor L
An Efficient FIR Structure for Conversion by a Rational Factor L/M

Implementation of a Multistage Sampling Rate Conversion System Using FIR Structures. 27

Design 1
Design 2
Design 3

Quantization Errors in the Direct Form FIR Linear Phase Digital Filters. 35

A-D Noise
Filter Coefficient Quantization Noise
Roundoff Noise
Conclusions

A NOVEL PC PROGRAM FOR DESIGNING MULTISTAGE SAMPLING RATE CONVERSION SYSTEMS. 45

A PC PROGRAM FOR IMPLEMENTING A CD-TO-DAT SAMPLING RATE CONVERSION. 53

Program Description. 53

Sampling a 10 kHz Sine Wave. 54

Sampling Rate Conversion
Windowing
Frequency Response
Quantization

Overall Frequency Response of the Three-Stage System. . . 60

Sampling Rate Conversion
Group Delay
Quantization

SUMMARY. 66

REFERENCES. 68

APPENDIX. 70

LIST OF TABLES

TABLE		PAGE
I	Comparison of Eqs. (10) and (13) for Estimating the Filter Length	13
II	Specifications for a Three-Stage Decimator	32
III	Specifications for a Four-Stage Decimator	35

LIST OF FIGURES

FIGURE		PAGE
1.	Lowpass Filter Design	4
2.	Input for McClellan-Parks Program	7
3.	Frequency Response of a 101-Point FIR Linear Phase Lowpass Digital Filter	8
4.	Rabiner's Algorithm	10
5.	Modified Algorithm	12
6.	Input for the Modifed Algorithm Program	14
7.	Frequency Response of the Lowpass Filter System	15
8.	Decimation by an Integer Factor M	16
9.	Interpolation by an Integer Factor L	18
10.	Two Ways of Interpreting Sampling Rate Conversion by L/M	21
11.	Generation of an Efficient Direct Form Structure for Realizing Decimation by an Integer Factor M	23
12.	An Efficient Direct Form Structure for Realizing Decimation by an Integer Factor M that Uses Symmetry	24
13.	Block Diagram for an Interpolator	24
14.	Efficient Structure for an Interpolator	25
15.	Block Diagram for Converting the Sampling Rate by L/M	26
16.	Block Diagram of a Program Structure for Converting the Sampling Rate by L/M	26

17. One-Stage Decimator. 28

18. Three-Stage Decimator. 28

19. Frequency Responses of Lowpass Filters for Three-Stage Decimator. 30

20. Four-Stage Decimator. 35

21. Input Data and Optimum Design Factors. 46

22. Three-Stage CD-to-DAT Sampling Rate Conversion. 47

23. Analysis Data for Optimum Design. 47

24. Effects of Filter Coefficient Quantization. 48

25. Lowpass Filter Design Specifications. 49

26. Spectra of the Digital Input Signal. 49

27. Frequency Responses of the Lowpass Filters for a Three-Stage CD-to-DAT Sampling Rate Conversion. 50

28. Lowpass Filter Coefficients are Saved to Files. 51

29. Actual Filter Lengths and Maximum Band Errors. 52

30. Effects of Filter Coefficient Quantization and Exit. 52

31. Input for the CD-to-DAT Sampling Rate Conversion Program. 53

32. Sampling Rate Conversion for a 10 kHz Signal with No Quantization. 55

33. Three-Stage Sampling Rate Conversion Algorithm. 56

34. Frequency Response for a 10 kHz Signal with No Quantization. 58

35. Frequency Response for a 10 kHz Signal with Quantization 59

36. Magnitude Response of the System with No Quantization. . 61

37. Phase Response of the System with No Quantization. 61

38. Group Delay of the System with No Quantization. 63

39. Magnitude Response of the System with Quantization. . . . 64

40. Phase Response of the System with Quantization. 64

41. Group Delay of the System with Quantization. 65

INTRODUCTION

Sampling rate conversion is an important digital signal processing concept that is used in a variety of applications. One such application is in the field of professional digital audio. There are three primary rates used, with a need to convert from one to another. The three include a broadcasting rate of 32 kHz, a consumer product rate (primarily CD) of 44.1 kHz, and a studio rate (primarily DAT) of 48 kHz [1].

Another application is in the area of digital communications, where a variety of coding formats may be used in different parts of a communications network [2]. One example is converting delta modulation (DM) format to pulse code modulation (PCM) format. Delta modulation is a simple technique that encodes the sign of each sample-to-sample difference of a highly over-sampled signal in a single bit. In contrast, pulse code modulation requires that each quantized sample be encoded in n bits. Intuitively, DM will operate at a higher sampling rate than PCM. Hence, there is a need for converting the sampling rates when changing formats.

Schafer and Rabiner [3] have shown that sampling rate changes are effected by interpolation and/or decimation and can be efficiently implemented by using finite impulse response (FIR) digital filters. Also, Crochiere and Rabiner [4] have shown that the computational rates and storage needed for conversion could be minimized by using more than one stage to change the sampling rate. Using these ideas, our paper presents new thoughts on some old algorithms and two new PC programs. One of the new programs allows users to easily design optimum

multistage sample rate conversion systems that are based upon minimizing a factor which is the weighted sum of the computational rate and the storage. The other program uses a set of filter coefficients that is output from the first program to simulate a CD-to-DAT sampling rate conversion.

This paper is divided into seven sections. In the section following the introduction, we discuss the design of an FIR linear phase lowpass digital filter with a predetermined number of filter taps, using the Parks-McClellan algorithm [5]. After a brief explanation of the algorithm, a Turbo Pascal version of the McClellan computer program [5] that incorporates the algorithm is used to solve an example problem.

In many cases, however, one does not know the required number of filter taps needed to meet the design specifications. Rabiner [6] proposed an algorithm for solving this problem. His algorithm initially estimates the filter length using Herrmann's formula and then uses the Parks-McClellan algorithm to see if the maximum band errors have been minimized. If they are not minimized, the estimated filter length is increased (or decreased) and the Parks-McClellan algorithm is called again. This process repeats until the maximum band errors have been minimized. Herrmann's formula and Rabiner's algorithm assume that the gain of the lowpass filter is one. However, a lowpass filter that is part of an interpolator system will require a gain that is greater than one. Consequently, we present a new modification to Herrmann's formula to normalize the gain.. We also propose a change to the input requirements for Rabiner's algorithm to include the gain of the lowpass filter, along with the sampling rate, passband frequency, and stopband frequency in analog terms rather than digital terms to more closely link the input to the designer's specifications. A flowchart of the modified algorithm and an example are presented. The example demonstrates that when the gain of the filter is greater

than five, the number of iterations required to determine the length of the filter is less than half of the number required using Rabiner's original algorithm.

In the fourth section, the primary digital signal processing concepts that are incorporated as part of the multistage design program are discussed. They include sampling rate conversion, FIR structures for sampling rate conversion, multistage design, and errors in the direct form FIR digital filters caused by quantization. We also introduce a new finding. This discovery is a condition that determines when to add a lowpass filter to a multistage decimator in order to reduce the total number of filter taps required to implement the system. In a typical example, we show a 34% reduction in the total required number of filter taps.

In the fifth section, the new multistage design program is presented. The program is demonstrated by designing a multistage system for CD-to-DAT conversion and calculating the lowpass filter coefficients for each stage. The program can optimize the design in approximately one second on a 486 machine. The user can modify the design or continue analyzing the original design by testing different word lengths to represent the filter coefficients. After this, the actual filter length and coefficients of each stage are determined and saved to ASCII files for further use. Hence, the total time spent designing and analyzing a sampling rate conversion system can be reduced from a day or so to minutes.

The other new program is presented in the sixth section. It tests the designed system by simulating a CD-to-DAT sampling rate conversion using the filter coefficients that were calculated in the previous program. The program generates a sampled sine wave or a unit sample, converts the sampling rate, and plots the frequency response of the converted output. All of this can be done in less than two minutes on a 486 machine.

The final section summarizes the results.

DESIGN OF AN FIR LINEAR PHASE LOWPASS DIGITAL FILTER OF KNOWN LENGTH

McClellan and Parks [7] have shown that the method of equiripple design based upon Tchebysheff approximation methods is a very effective way of designing lowpass as well as other types of FIR linear phase digital filters. The filters designed by this technique are optimal in the sense that the maximum weighted approximation errors in the frequency bands over the range of interest are minimized.

Suppose that we wish to design a type I FIR linear phase digital filter as in Fig. 1.

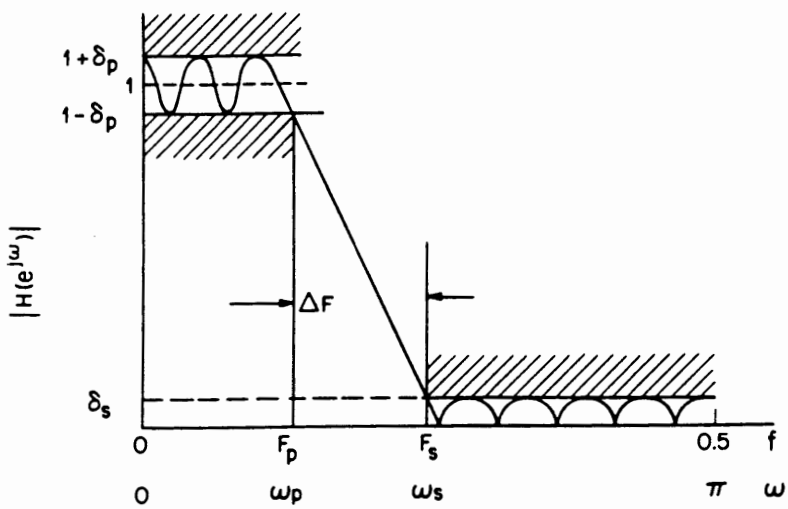


Figure 1. Lowpass filter design.

Recall that a type I FIR filter has a symmetric impulse response

$$h[n] = h[(N-1)-n], \qquad 0 \le n \le N-1 \tag{1}$$

where N is an odd integer. The frequency response is given by

the expression

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n]e^{-j\omega n}$$

$$= \sum_{n=0}^{\frac{N-1}{2}} h[n]e^{-j\omega n} + \sum_{n=\frac{N+1}{2}}^{N-1} h[n]e^{-j\omega n}$$

$$= \sum_{n=0}^{\frac{N-1}{2}} h[n]e^{-j\omega n} + \sum_{n=\frac{N-3}{2}}^{0} h[N-1-n]e^{-j\omega(N-1-n)}$$

$$= \sum_{n=0}^{\frac{N-1}{2}} h[n]e^{-j\omega n} + \sum_{n=0}^{\frac{N-3}{2}} h[n]e^{-j\omega(N-1-n)}$$

$$= e^{-j\omega(\frac{N-1}{2})}\{\sum_{n=0}^{\frac{N-1}{2}} h[n](e^{j\omega(\frac{N-1}{2}-n)} + e^{-j\omega(\frac{N-1}{2}-n)}) - h[\frac{N-1}{2}]\}$$

$$= e^{-j\omega(\frac{N-1}{2})}\{\sum_{n=0}^{\frac{N-1}{2}} 2h[n]\cos(\omega(\frac{N-1}{2}-n)) - h[\frac{N-1}{2}]\}$$

$$= e^{-j\omega(\frac{N-1}{2})}\{\sum_{k=0}^{\frac{N-1}{2}} 2h[\frac{N-1}{2}-k]\cos(\omega k) - h[\frac{N-1}{2}]\}$$

$$= e^{-j\omega(\frac{N-1}{2})}\sum_{k=0}^{\frac{N-1}{2}} a[k]\cos(\omega k) \tag{2}$$

where $a[0] = h[\frac{N-1}{2}]$ and $a[k] = 2h[\frac{N-1}{2}-k]$ for $k = 1,2,\ldots,\frac{N-1}{2}$. The terms $\cos(\omega k)$ can be expressed as the sum of powers of $\cos(\omega)$ in the form

$$\cos(\omega k) \quad = T_k(\cos\omega) \tag{3}$$

where $T_k(x)$ is the k th order Tchebysheff polynomial term, defined by $T_k(x) = \cos(k\cos^{-1}x)$. Thus,

$$H(e^{j\omega}) = e^{-j\omega(\frac{N-1}{2})}\sum_{k=0}^{\frac{N-1}{2}} a[k](\cos\omega)^k. \tag{4}$$

According to the alternation theorem used by Parks and McClellan [5], the

frequency response magnitude

$$\overline{H}(e^{j\omega}) = \sum_{k=0}^{\frac{N-1}{2}} a[k](\cos\omega)^k \tag{5}$$

will best approximate an ideal lowpass frequency response

$$H_d(e^{j\omega}) = \begin{cases} 1, & 0 \le \omega \le \omega_p \\ 0, & \omega_s \le \omega \le \pi \end{cases} \tag{6}$$

over the compact subset $F \subset [0,\pi]$, where $F = [0,\omega_p] \cup [\omega_s,\pi]$, provided that the maximum weighted error

$$\left\| E(e^{j\omega}) \right\| = \max_{\omega \in F} \left| E(e^{j\omega}) \right| \tag{7}$$

where

$$E(e^{j\omega}) = W(e^{j\omega})\left[H_d(e^{j\omega}) - \overline{H}(e^{j\omega})\right] \tag{8}$$

and

$$W(e^{j\omega}) = \begin{cases} \frac{1}{K} = \frac{\delta_s}{\delta_p}, & 0 \le \omega \le \omega_p \\ 1, & \omega_s \le \omega \le \pi \end{cases} \tag{9}$$

exhibits at least $\frac{N+1}{2}$ alternations on F. These alternations are a set of extremal frequencies such that $\omega_1 < \omega_2 < \ldots < \omega_{\frac{N-1}{2}} < \omega_{\frac{N+1}{2}}$ and $E(e^{j\omega_i}) = -E(e^{j\omega_{i+1}})$, for $i = 1,2,\ldots,\frac{N-1}{2}$. The weighting function will differ from Eq. (9) if other filter types are used instead of the type I filter.

McClellan *et al.* [5] wrote a FORTRAN program that determines the best approximation to an ideal lowpass filter (as well as other types of filters) using the equiripple design method that we just described. Since the $a[k]$'s are initially unknown, the program uses a Lagrange interpolation polynomial $A(e^{j\omega})$ of order $\frac{N+1}{2}$ that passes through the extremal frequencies to approximate Eq. (5). The output to the program includes the extremal frequencies and the FIR linear phase digital filter coefficients. We have translated the program into Turbo Pascal for use on a PC.

As an example, consider the design of a 101-point digital filter with a passband frequency of $\frac{\omega_p}{2\pi} = 0.42$, a stopband frequency of $\frac{\omega_s}{2\pi} = 0.46$,

$$H_d(e^{j\omega}) = \begin{cases} 1, & 0 \le \frac{\omega}{2\pi} \le 0.42 \\ 0, & 0.46 \le \frac{\omega}{2\pi} \le 0.5 \end{cases}$$

and

$$W(e^{j\omega}) = \begin{cases} \frac{10^{-6}}{10^{-2}} = 10^{-4}, & 0 \le \frac{\omega}{2\pi} \le 0.42 \\ 1, & 0.46 \le \frac{\omega}{2\pi} \le 0.5 \end{cases}$$

Figure 2 shows the format for data entry. The program may require several

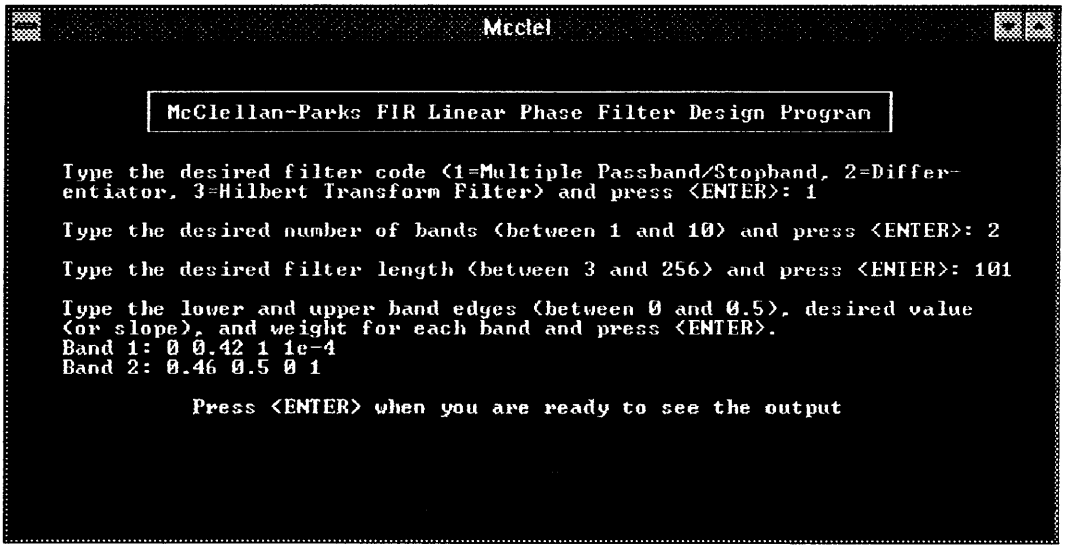
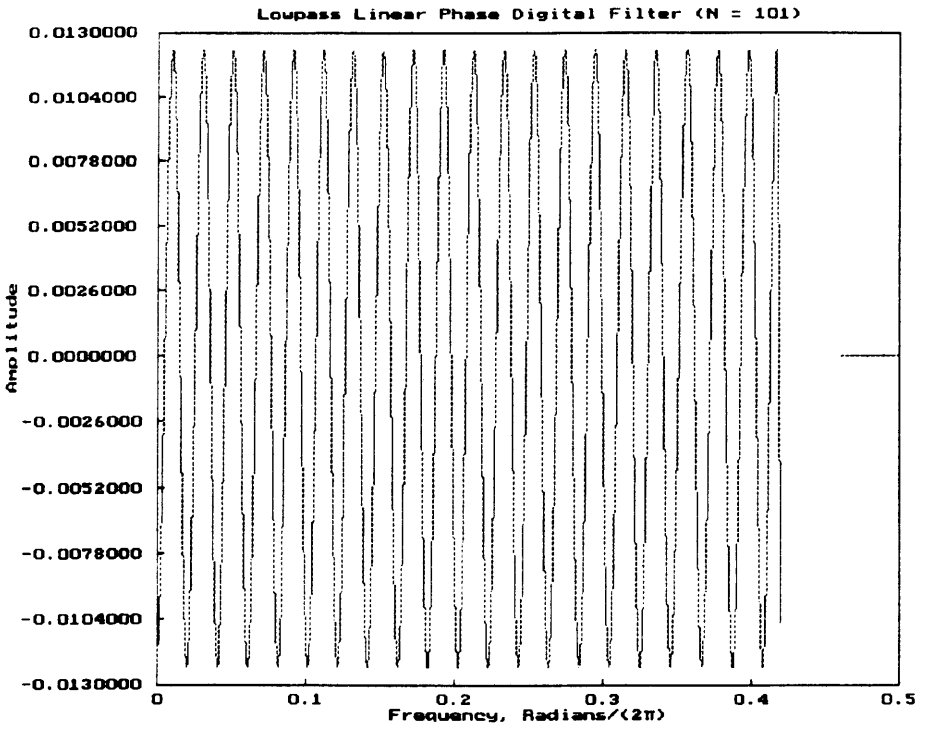


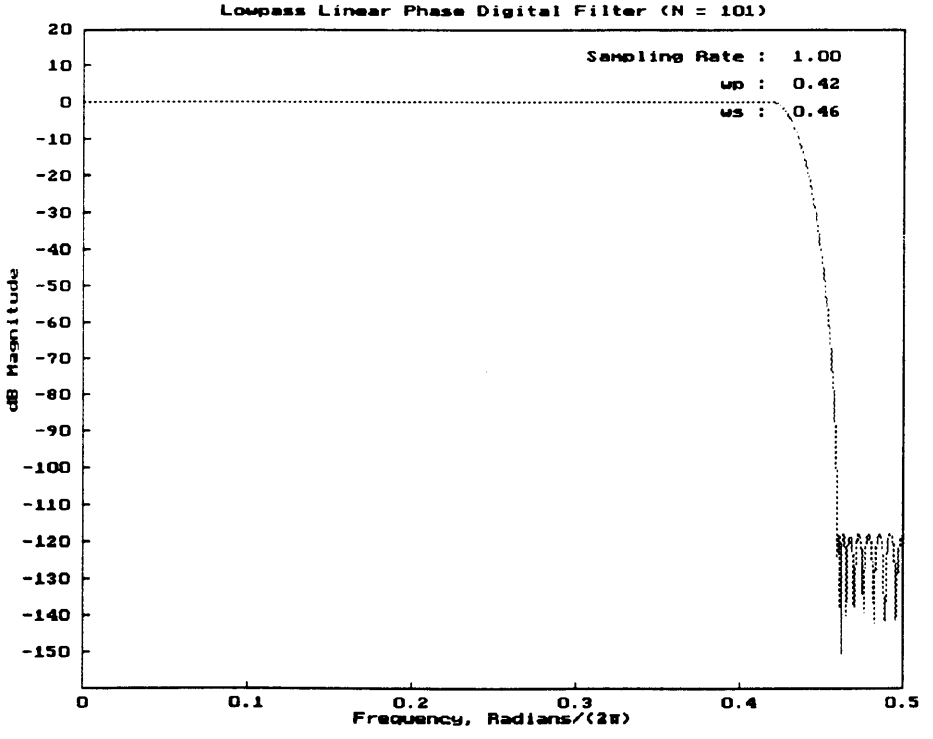
Figure 2. Input for McClellan-Parks program.

iterations to ensure that the maximum weighted error meets the passband and stopband requirements. Consequently, the program may take up a minute or so to determine the extremal frequencies and corresponding filter length.

The filter coefficients and extremal frequencies are not shown here but are part of the output. The frequency response is shown on the next page. Figure 3(a) shows the error in the passband while Fig. 3(b) shows the dB magnitude response. Note that there are $\frac{N+1}{2} = 51$ extremal frequencies.



(a)



(b)

Figure 3. Frequency response of a 101-point FIR linear phase lowpass digital filter.

DESIGN OF AN FIR LINEAR PHASE LOWPASS DIGITAL FILTER OF UNKNOWN LENGTH

The McClellan computer program requires that the filter length N, the passband frequency $\frac{\omega_p}{2\pi}$, the stopband frequency $\frac{\omega_s}{2\pi}$, and the ratio of passband-to-stopband error K are known. The program can then determine the polynomial with an order related to N that minimizes the maximum errors in the passband and stopband.

In many engineering problems, however, one of the parameters may not be known. A major breakthrough occurred when Herrmann *et al.* [8] discovered a mathematical relationship that estimates the filter length in terms of the other four filter parameters. Thus, it is possible to design a filter without knowing the length. Herrmann's formula is given by

$$\hat{N} = \frac{D_\infty(\delta_p, \delta_s)}{(\omega_p - \omega_s)/2\pi} - f(\delta_p, \delta_s)\frac{(\omega_p - \omega_s)}{2\pi} + 1 \tag{10}$$

where

$$D_\infty(\delta_p, \delta_s) = [a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3]\log_{10}\delta_s$$
$$+[a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6], \tag{11}$$

$$f(\delta_p, \delta_s) = b_1 + b_2(\log_{10}\delta_p - \log_{10}\delta_s), \tag{12}$$

and

$$a_1 = 5.309x10^{-3} \qquad a_4 = -2.660x10^{-3}$$
$$a_2 = 7.114x10^{-2} \qquad a_5 = -5.941x10^{-2}$$
$$a_3 = -4.761x10^{-1} \qquad a_6 = -4.278x10^{-1}$$
$$b_1 = 11.01217 \qquad b_2 = 0.51244.$$

Rabiner [6] has written an algorithm that uses Eq. (10) to estimate the filter length and then uses the Parks-McClellan algorithm to determine the actual filter length that minimizes the maximum allowable errors in the passband and stopband. Figure 4 illustrates the logic for the algorithm.The algorithm is described below.

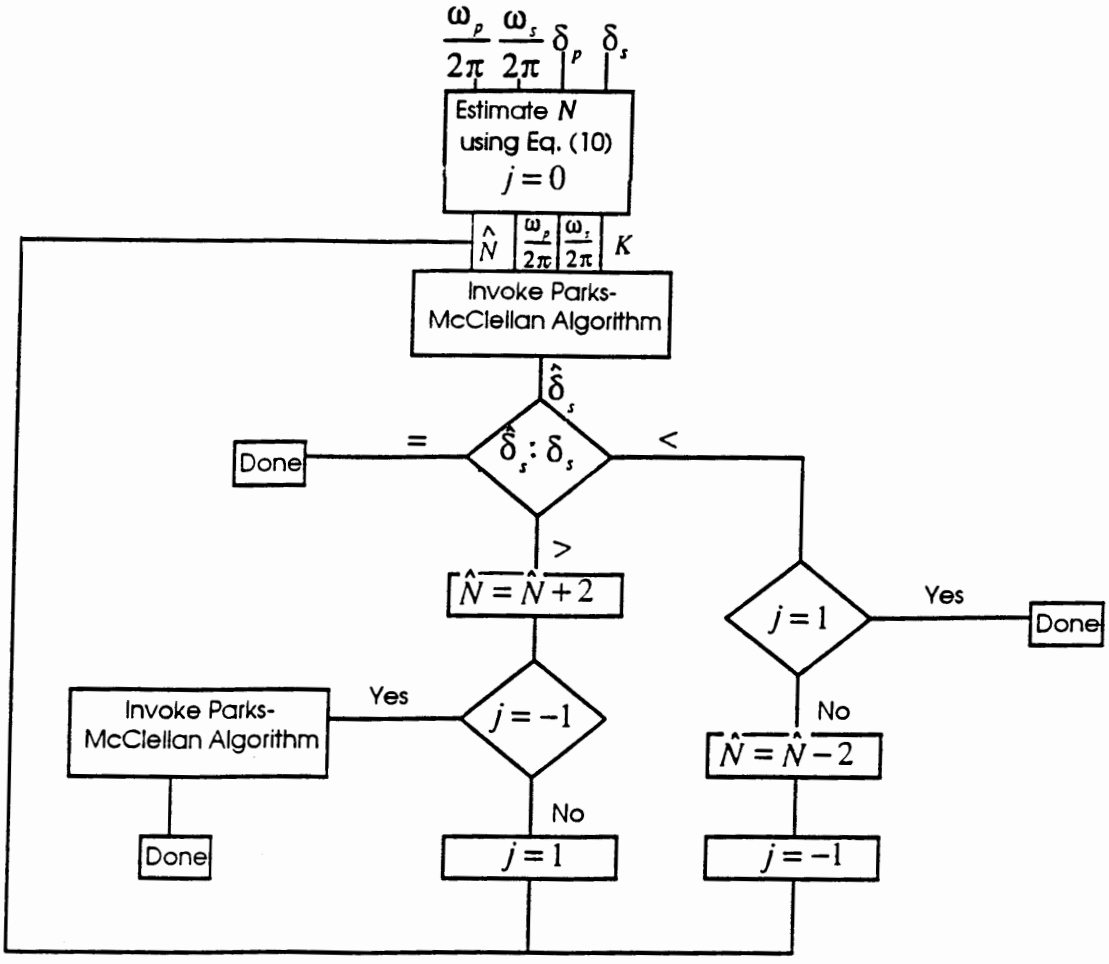


Figure 4. Rabiner's algorithm [6].

After estimating the filter length from Eq. (10), the direction parameter j is set to 0. The input parameters $\hat{N}, \frac{\omega_p}{2\pi}, \frac{\omega_s}{2\pi}$, and $K = \frac{\delta_p}{\delta_s}$ are used as input to the Parks-McClellan algorithm that returns the actual stopband error $\hat{\delta}_s$. This value is compared to δ_s. If they are equal (within some tolerance), the algorithm is done. If

$\hat{\delta}_s > \delta_s$, $\hat{N}$ is incremented by 2. This increases the order of the approximating polynomial by 1. If $j = -1$, the direction of estimation has changed and the minimum value of $\hat{N}$ that meets or exceeds the specifications has been reached. In that case, the Parks-McClellan algorithm must be invoked again and the algorithm is done. If $j \neq -1$, it is set to 1 and the Parks-McClellan algorithm is invoked for another iteration.

If $\hat{\delta}_s < \delta_s$ and $j = 1$, the current value of $\hat{N}$ is the minimum one that meets or exceeds the specifications and the algorithm is done. Otherwise, $\hat{N}$ is decreased by 2, j is set to 1, and the algorithm repeats. From empirical evidence, Rabiner found that for a large number of lowpass filters with unity gain in the passband, the filter length could be found after two or three iterations.

What about the lowpass filters used for interpolation? A gain of L is required of such filters. Intuitively, we can see that increasing the gain while leaving the other filter parameters unchanged will require a longer filter length. No allowance is made for this in Eq. (10), nor is one made in Rabiner's algorithm. Consequently, Rabiner's algorithm will require more iterations to arrive at a final solution.

We propose a modification to Eq. (10) to better address such problems. A simple approach is to normalize the frequency response in the passband. Thus, we can write the new relationship as

$$\hat{N} = \frac{D_\infty(\frac{\delta_p}{L}, \frac{\delta_s}{L})}{(f_p - f_s)/F} - f(\tfrac{\delta_p}{L}, \tfrac{\delta_s}{L})\frac{(f_p - f_s)}{F} + 1 \tag{13}$$

where

$$\begin{aligned} D_\infty(\tfrac{\delta_p}{L}, \tfrac{\delta_p}{L}) &= [a_1(\log_{10}\tfrac{\delta_p}{L})^2 + a_2\log_{10}\tfrac{\delta_p}{L} + a_3]\log_{10}\tfrac{\delta_s}{L} \\ &\quad + [a_4(\log_{10}\tfrac{\delta_p}{L})^2 + a_5\log_{10}\tfrac{\delta_p}{L} + a_6], \\ f(\tfrac{\delta_p}{L}, \tfrac{\delta_s}{L}) &= b_1 + b_2(\log_{10}\tfrac{\delta_p}{L} - \log_{10}\tfrac{\delta_s}{L}) \end{aligned} \tag{14}$$

$$= f(\delta_p, \delta_s), \tag{15}$$

and

$$a_1 = 5.309x10^{-3} \qquad a_4 = -2.660x10^{-3}$$

$$a_2 = 7.114x10^{-2} \qquad a_5 = -5.941x10^{-2}$$

$$a_3 = -4.761x10^{-1} \qquad a_6 = -4.278x10^{-1}$$

$$b_1 = 11.01217 \qquad b_2 = 0.51244.$$

The Rabiner algorithm can then be modified as in Fig. 5. In this case, original

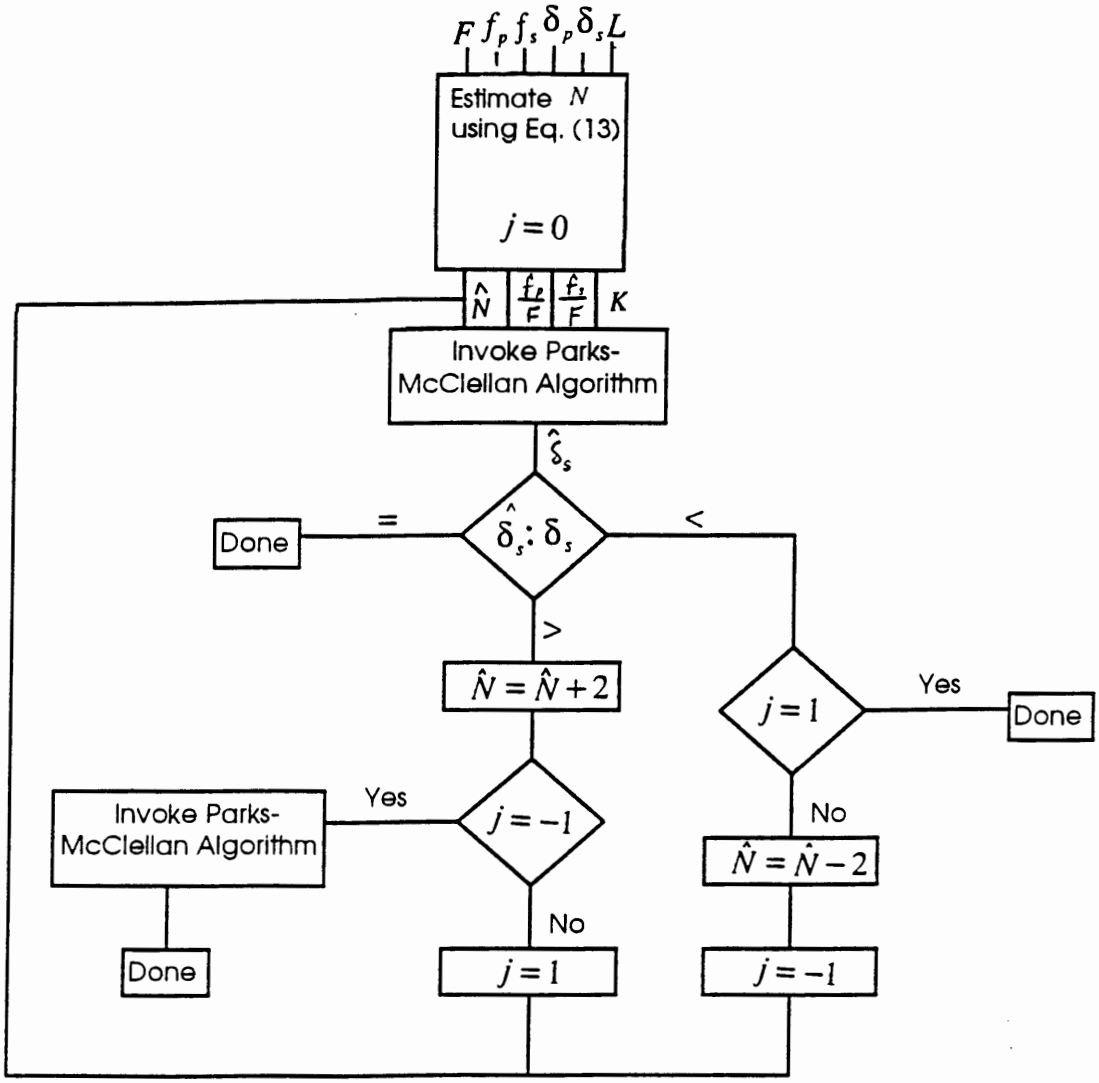


Figure 5. Modified algorithm .

specifications are stated in terms of frequency and are used to estimate the filter length. This feature links the program input more closely to the design problem.

The following example demonstrates the effectiveness of using Eq. (13) to estimate the filter length. Suppose that we want to design a lowpass filter which has the following parameters: $F = 48$ kHz, $f_p = 20$ kHz, $f_s = 22$ kHz, $\delta_p = 10^{-2}$, and $\delta_s = 10^{-6}$. We also want to vary the filter gain L. The test is to compare the number of iterations required to arrive at a filter length that meets the design specifications using Eqs. (10) and (13). The results are shown in Table I. It is

TABLE I

COMPARISON OF EQS. (10) AND (13) FOR ESTIMATING THE FILTER LENGTH

L	$\hat{N}$ - Eq. (10)	Iterations	N	$\hat{N}$ - Eq. (13)	Iterations	N
1	102	2	98	102	2	98
2	102	4	110	115	4	107
5	102	11	124	129	4	121
10	102	14	130	140	5	130

obvious that more iterations are required as L increases if Eq. (10) is used. A few more examples were tested, resulting in the following conclusions:

1. If $L = 1$ or 2, Eq. (13) was either comparable or superior to Eq. (10).
2. If $L > 2$, Eq. (13) was always superior to Eq. (10).
3. The estimations by both Eqs. (10) and (13) became less accurate as L increased.

Perhaps with more research, an estimate superior to Eq. (13) can be found by determining new constants.

We have written a Turbo Pascal PC program for the modified algorithm. The program will take from one second to two minutes to arrive at a satisfactory filter length, depending upon the other specified filter parameters. Figure 6 shows the input for the modified algortihm program.

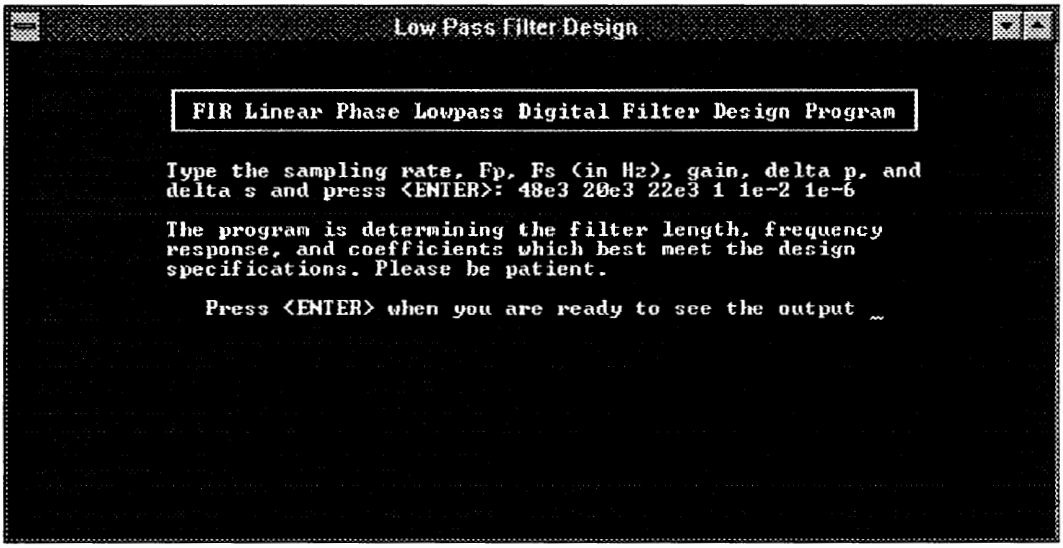
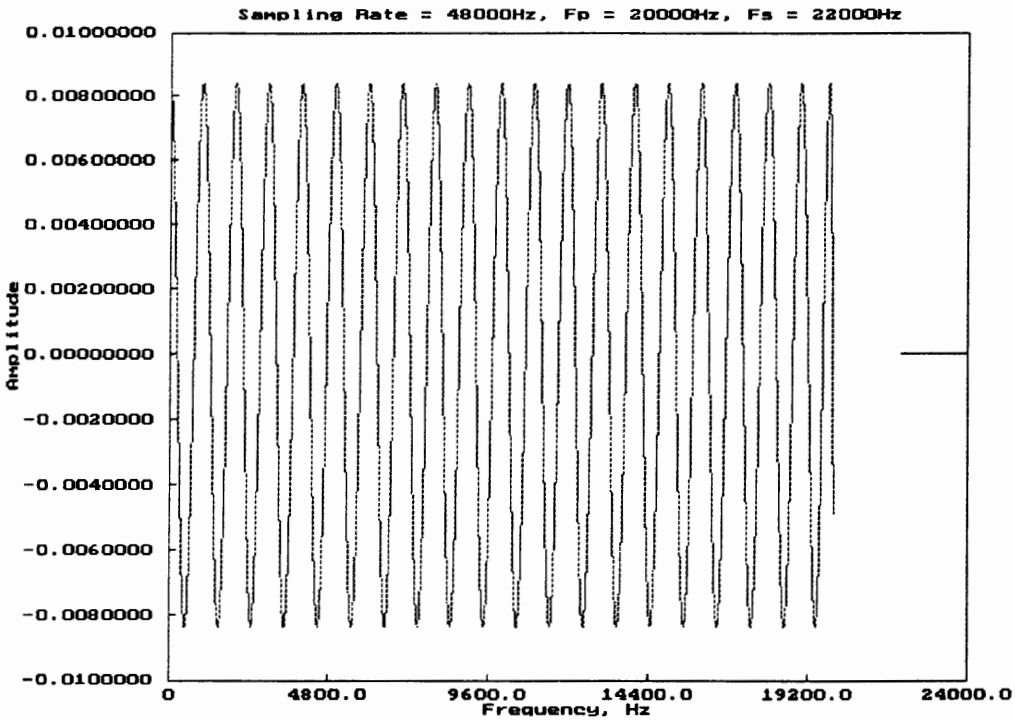
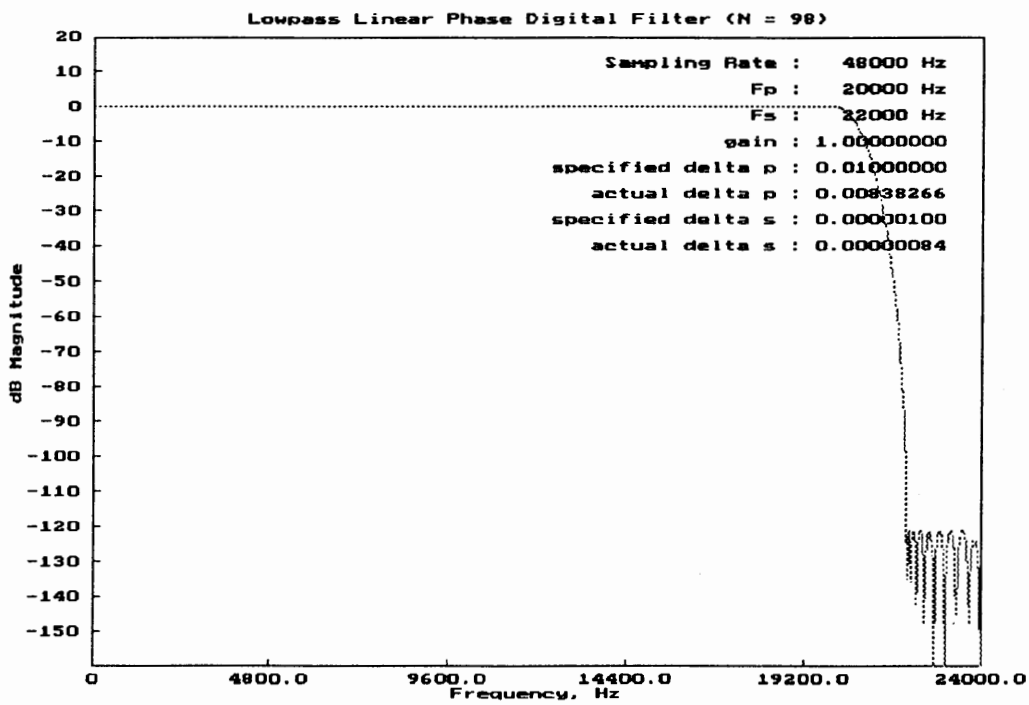


Figure 6. Input for the modified algorithm program.

On the following page, Fig. 7 shows the frequency response. The filter coefficients, the extremal frequencies, and some other band data are also available as part of the program, but are not shown here. The filter coefficients may be saved to a file at the conclusion of the program.



(a)



(b)

Figure 7. Frequency response of the lowpass filter system.

REVIEW OF SOME FUNDAMENTAL DIGITAL SIGNAL PROCESSING CONCEPTS

SAMPLING RATE CONVERSION

Decimation By An Integer Factor M

Suppose that we want to reduce the sampling rate of a digital signal by an integer factor M. If the initial sampling rate is F, the resulting sampling rate will be

$$F' = \frac{F}{M} \tag{16}$$

and the corresponding sampling period will be

$$T' = MT, \tag{17}$$

where $T = \frac{1}{F}$.

An easy way to understand decimation is to examine the time and frequency domain graphs. Figure 8 illustrates the overall process in a block diagram and the time and frequency responses in graphs as the signal is processed.

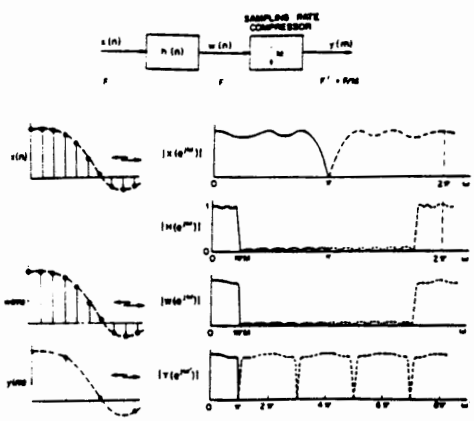


Figure 8. Decimation by an integer factor M [2].

In the frequency domain, the lowpass filter anticipates the sampling rate reduction by ensuring that the frequency components above $\frac{\pi}{M}$ are negligible. The frequency response of the lowpass digital filter approximates the ideal characteristic

$$H_d(e^{j\omega}) = \begin{cases} 1, & |\omega| < 2\pi(\frac{F'/2}{F}) = \frac{\pi}{M} \\ 0, & \frac{\pi}{M} \le |\omega| \le \pi \end{cases} \tag{19}$$

Suppose that the filter is a type I FIR linear phase system as in Eq. (1). The filtered output response to an arbitrary input sequence $x[n]$ will be

$$w[n] = \sum_{k=0}^{\frac{N-1}{2}} h[k]\{x[n-k] + x[n-(N-1-k)]\} - h\left[\frac{N-1}{2}\right]x\left[n - \frac{N-1}{2}\right]. \tag{20}$$

The sampling rate reduction is achieved by processing the filtered output through a sampling rate compressor. The compressor saves every M th sample and discards the rest. This process is called *downsampling* and can be represented mathematically by

$$y[m] = w[Mm]. \tag{21}$$

Substituting Eq. (20) into Eq. (21) results in

$$y[m] = \sum_{k=0}^{\frac{N-1}{2}} h[k]\{x[Mm-k] + x[Mm-(N-1-k)]\} - h\left[\frac{N-1}{2}\right]x\left[Mm - \frac{N-1}{2}\right]. \tag{22}$$

The process of lowpass filtering a signal and downsampling the result is called *decimation.*

Interpolation By An Integer Factor L

If we want to increase the sampling rate F of a digital signal by an integer factor L, the new sampling rate will be

$$F' = LF \tag{23}$$

and the corresponding sampling period will be

$$T' = \frac{T}{L}, \tag{24}$$

where $T = \frac{1}{F}$.

Again, we refer to the time and frequency domain graphs as an easy way of grasping the concept. Figure 9 illustrates the overall process in a block diagram and the time and frequency responses in graphs as the signal is processed.

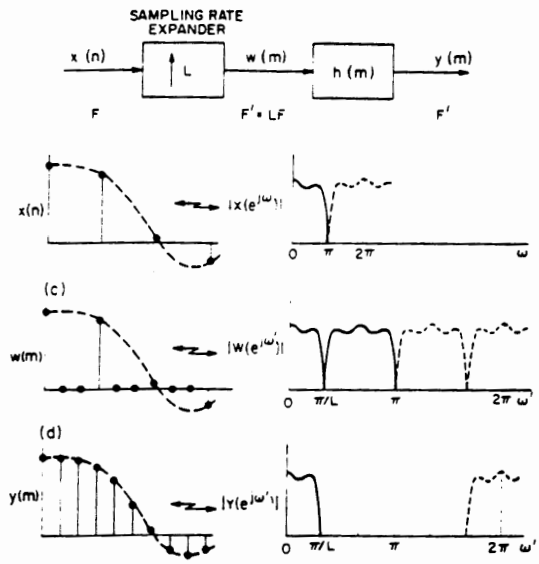


Figure 9. Interpolation by an integer factor L [2].

The sampling rate increase is the result of processing the signal through a sampling rate expander. The expander inserts $L-1$ zero-valued samples between each pair of samples of $x[n]$, resulting in the sequence

$$w[m] = \begin{cases} x[\frac{m}{L}], & m = 0, \pm L, \pm 2L, \ldots \\ 0, & \text{otherwise}. \end{cases} \tag{25}$$

This process is called *upsampling*.

In addition to having the baseband frequencies of interest (i.e., $-2\pi(\frac{f_c}{F})$ to $2\pi(\frac{f_c}{F})$, where f_c. is the cutoff frequency) in its spectrum, $w[m]$ also has images of the baseband in its spectrum. The images are centered at $\pm\frac{2\pi}{L}, \frac{4\pi}{L}, \ldots$. To eliminate these undesirable images, $w[m]$ must be filtered through a lowpass digital filter that approximates the ideal characteristic

$$H_d(e^{j\omega}) = \begin{cases} L, & |\omega'| < 2\pi(\frac{F/2}{F'}) = \frac{\pi}{L} \\ 0, & \frac{\pi}{L} \le |\omega'| \le \pi \end{cases} \tag{26}$$

where $\omega' = 2\pi(\frac{f}{F'}) = 2\pi(\frac{f}{LF})$. The passband gain must be L in order to compensate for the energy lost in filtering out the $L-1$ harmonic images. The process of upsampling followed by lowpass filtering is called *interpolation*.

Assume for the moment that the lowpass filter is an FIR system of infinite length. The filtered output will be

$$y[m] = \sum_{k=-\infty}^{\infty} h[m-k]w[k]. \tag{27}$$

Substituting Eq. (25) into Eq. (27) results in

$$y[m] = \sum_{k=-\infty}^{\infty} h[m-k]x[\frac{k}{L}], \tag{28}$$

where $x[\frac{k}{L}]$ is nonzero when $\frac{k}{L}$ is an integer. If we make the substitution

$$\frac{k}{L} = r, \tag{29}$$

we have

$$y[m] = \sum_{r=-\infty}^{\infty} h[m-rL]x[r]. \tag{30}$$

We now make another substitution

$$r = \left\lfloor \frac{m}{L} \right\rfloor - n. \tag{31}$$

where $\lfloor u \rfloor$ is the truncation of u to the next lowest integer. The result is

$$\begin{aligned} y[m] &= \sum_{n=-\infty}^{\infty} h[m - \lfloor \tfrac{m}{L} \rfloor L + nL]x[\lfloor \tfrac{m}{L} \rfloor - n] \\ &= \sum_{n=-\infty}^{\infty} h[m \oplus L + nL]x[\lfloor \tfrac{m}{L} \rfloor - n] \end{aligned} \tag{32}$$

where $m \oplus L$ denotes the value m modulo L, or the remainder of m divided by L. Next, we introduce the notation

$$g_{m\oplus L}[n] = h[m \oplus L + nL] \tag{33}$$

to highlight that $h[m \oplus L + nL]$ is periodic in m with period L. Since $g_{m \oplus L}[n]$ is the impulse response of the system at time m to an input at time $\left\lfloor \frac{m}{L} \right\rfloor - n$, a delay of the input will not necesarily result in the same delay of the output. Thus, the system is *time-varying*. Substituting Eq. (33) into Eq.(32) results in

$$y[m] = \sum_{n=-\infty}^{\infty} g_{m \oplus L}[n] x[\lfloor \tfrac{m}{L} \rfloor - n]. \tag{34}$$

Suppose that the filter is an FIR linear phase system having filter length N such that $N = QL$, where Q is an integer. If N is not an integer multiple of L, we can pad $h[k]$ with zeros so that the FIR filter length will be $N' = QL$, yet the sequence $h[k]$ will remain unchanged. In either case, Eq. (34) can be expressed as

$$y[m] = \sum_{n=0}^{Q-1} g_{m \oplus L}[n] x[\lfloor \tfrac{m}{L} \rfloor - n]. \tag{35}$$

When this form is employed, the number of computations is reduced by a factor of $\frac{1}{L}$ compared to the direct form for each output sample.

<u>Conversion By A Rational Factor *L/M*</u>

If the sampling rate of a digital signal is F and we convert it by a rational factor $\frac{L}{M}$, the resulting sample rate will be

$$F' = (\frac{L}{M})F \tag{36}$$

and the corresponding sampling period will be

$$T' = (\frac{M}{L})T, \tag{37}$$

where $T = \frac{1}{F}$. There are two ways of describing such a sampling rate conversion system. The first way is to consider the system as the cascade of an interpolation subsystem followed by a decimation subsystem. This is shown in Fig. 10(a). Note the order of the subsystems. If the order were reversed, there would be fewer samples used as a basis for interpolation. As a result, the output signal would be

less accurate. The second, more efficient way is to replace the two lowpass filters with one as shown in Fig. 10(b). This is possible since both filters are operating at the same sampling rate.

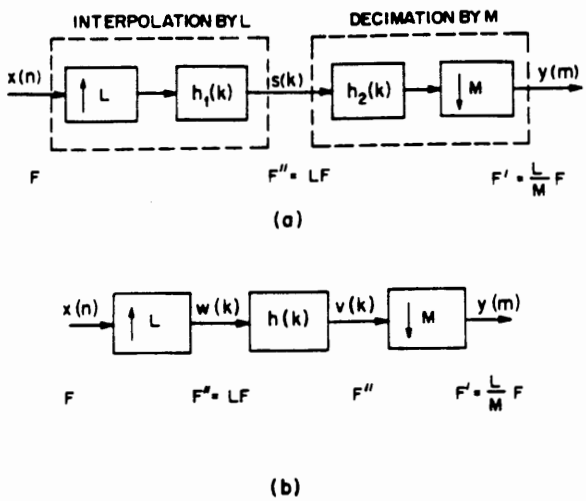


Figure 10. Two ways of interpreting sampling rate conversion by $\frac{L}{M}$ [2].

In order to fulfill the role of a lowpass digital filter for both interpolator and decimator, the filter should approximate the ideal characteristic

$$H_d(e^{j\omega''}) = \begin{cases} L, & |\omega''| < \min(\frac{\pi}{L}, \frac{\pi}{M}) \\ 0, & \min(\frac{\pi}{L}, \frac{\pi}{M}) \le \pi \end{cases} \tag{38}$$

where $\omega'' = 2\pi(\frac{f}{F''}) = 2\pi(\frac{1}{L})(\frac{f}{F})$.

Examining Fig. 10(b), we see that the time domain relationship for the interpolator subsystem is the same as Eq. (32). Thus, we have

$$v[k] = \sum_{n=-\infty}^{\infty} h[k \oplus L + nL] x[\lfloor \frac{k}{L} \rfloor - n]. \tag{39}$$

From Eq. (21), we have

$$y[m] = v[Mm]. \tag{40}$$

Substituting Eq. (39) into Eq. (40) results in

$$y[m] = \sum_{n=-\infty}^{\infty} h[Mm \oplus L + nL] x[\lfloor \tfrac{Mm}{L} \rfloor - n]. \tag{41}$$

We now use notation similar to that which we used in Eq. (32); namely,

$$g_{Mm \oplus L}[n] = h[Mm \oplus L + nL]. \tag{42}$$

Again, this is a convenient way of highlighting the periodicity. Also, this notation is makes it easy to translate the scrambled coefficients into a two dimensional sequential array in a computer program that implements a sampling rate conversion.

If the filter is an FIR linear phase system having filter length N and $N = QL$, where Q is an integer, Eq. (41) can be expressed as

$$y[m] = \sum_{n=0}^{Q-1} g_{Mm \oplus L}[n] x[\lfloor \tfrac{Mm}{L} \rfloor - n]. \tag{43}$$

When this form is employed, the number of computations is also reduced by a factor of $\frac{1}{L}$ compared to the direct form for each output sample.

IMPLEMENTATION OF A ONE-STAGE SAMPLING RATE CONVERSION SYSTEM USING FIR STRUCTURES

An Efficient FIR Structure For Decimation By An Integer Factor M

Consider the decimator model that was developed in the previous section and is shown again in Fig. 11(a). According to the model, the input sequence is first lowpass filtered at the high sampling rate F and then downsampled by the sampling rate compressor. The two-step process is described by Eqs. (20) and (21) and the equations are repeated here

$$w[n] = \sum_{k=0}^{N-1} h[k] x[n-k] \tag{44}$$

$$y[m] = w[Mm]. \tag{45}$$

A realization of these equations is shown in Fig. 11(b) and is called the *direct form structure*.

The decimator can be realized more efficiently, however, if we substitute Eq. (44) into Eq. (45). The result is Eq. (23), which is repeated for convenience as

$$y[m] = \sum_{k=0}^{N-1} h[k]x[Mm-k] \tag{46}$$

A realization of this equation is called the *transposed direct form structure* and it is shown in Fig. 11(c). The advantage of this realization is that the filter now operates at the low sampling rate $\frac{F}{M}$ and the computational rate is reduced.

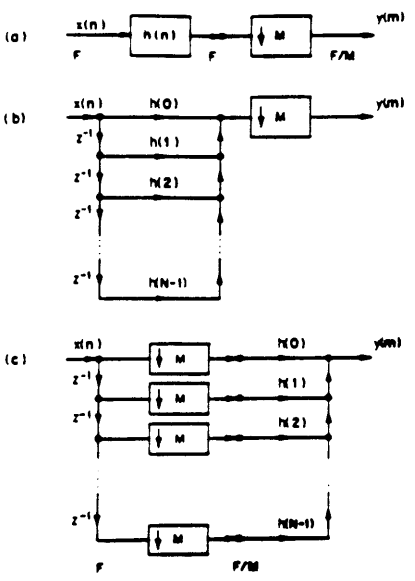


Figure 11. Generation of an efficient direct form structure for realizing decimation by an integer factor M [2].

Further reduction of the computational rate can be realized if the filter coefficients are symmetric. If the filter length is even and the coefficients are symmetric, Eq. (46) can be written as

$$y[m] = \sum_{k=0}^{\frac{N}{2}-1} h[k]x[Mm-k] + \sum_{k=\frac{N}{2}}^{N-1} h[k]x[Mm-k]$$

$$= \sum_{k=0}^{\frac{N}{2}-1} h[k]x[Mm-k] + \sum_{k=0}^{\frac{N}{2}-1} h[N-1-k]x[Mm-(N-1-k)]$$

$$= \sum_{k=0}^{\frac{N}{2}-1} h[k]\{x[Mm-k] + x[Mm-(N-1-k)]\}. \qquad (47)$$

A realization of this form is called the *symmetric transposed direct form structure* and is shown in Fig. 12. This structure requires $\frac{N}{2}$ multiplications and $N-1$

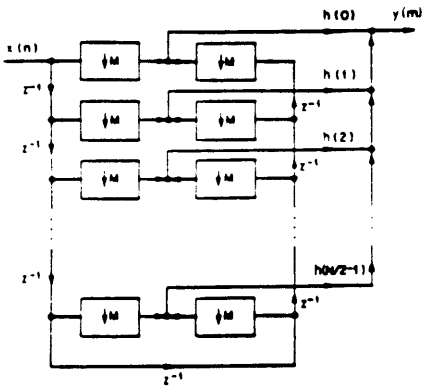


Figure 12. An efficient direct form structure for realizing decimation by an integer factor M that uses symmetry [2].

additions for every output sample. Since the compressor discards $M-1$ of every M input samples, the filter operates at the low sampling rate $\frac{F}{M}$. Consequently, only $\frac{F}{M}\left(\frac{N}{2}\right)$ multiplications and $\frac{F}{M}(N-1)$ additions are performed every second. If the filter length is odd, it can be shown that $\frac{F}{M}\left(\frac{N+1}{2}\right)$ multiplications and $\frac{F}{M}(N-1)$ additions are performed every second.

An Efficient FIR Structure For Interpolation By An Integer Factor L

A model for the interpolator is repeated in Fig. 13.

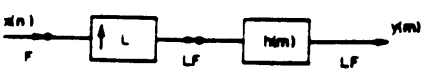


Figure 13. Block diagram for an interpolator [2].

As we have previously shown, the interpolator can be represented by the equation

$$y[m] = \sum_{n=0}^{Q-1} g_{m \oplus L}[n] x[\lfloor \tfrac{m}{L} \rfloor - n]. \tag{48}$$

The direct implementation of this equation is shown in Fig. 14. The sets of

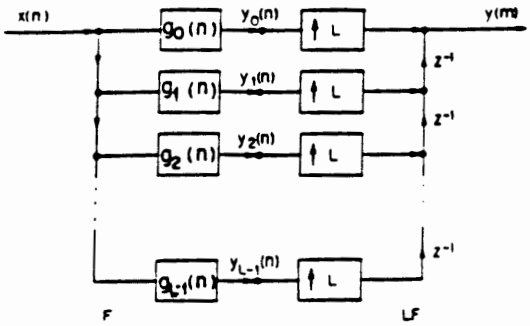


Figure 14. Efficient structure for an interpolator [2].

coefficients $g_0[n], g_1[n], \ldots, g_{L-1}[n]$ can be treated as L separate, linear, time-invariant filters that are decimated (by a factor L) versions of the impulse response $h[n]$ and operate at the low sampling rate F. When treated in this way, the sets of coefficients $g_0[n], g_1[n], \ldots, g_{L-1}[n]$ are referred to as *polyphase filters* [2]. The structure in Fig. 14 has L branches. Each branch requires Q multiplications and $Q-1$ additions, summing to a total of $LQ = N$ multiplications and $L(Q-1) = N - L$ additions for each input sample. Since each polyphase filter operates at the sampling rate F, FN multiplications and $F(N-L)$ additions are performed every second.

An Efficient FIR Structure For Conversion By A Rational Factor *L/M*

Efficient realizations of the decimator and the interpolator were obtained by commuting the filtering operations to occur at the low sampling rates. Such an operation is difficult for the model shown in Fig. 15 because the filter is located between the expander and compressor. Efficient structures do exist, however.

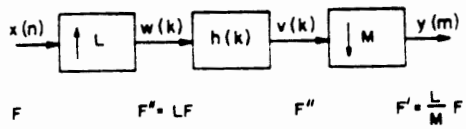


Figure 15. Block diagram for converting the sampling rate by $\frac{L}{M}$ [2].

Figure 16 shows how the equation can be implemented using a computer program

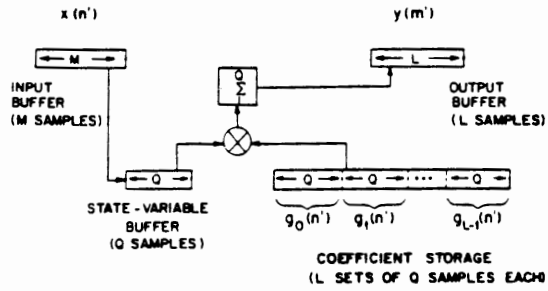


Figure 16. Block diagram of a program structure for converting the sampling rate by $\frac{L}{M}$ [2].

whereby a block of M input samples is converted into a block of L output samples. The program starts by initializing all buffers to zero. M input samples are then shifted into the input buffer and the first of those is shifted into the state variable buffer. Next, each of the Q samples of the state variable buffer is multiplied by one of the Q samples of the coefficient set $g_0[n]$ and the products are summed, resulting in $y[1]$. An additional decision must be made to calculate $y[m]$ when $m > 1$. If $\left\lfloor \frac{Mm}{L} \right\rfloor$ increases by one, another input sample is shifted into the state variable buffer. Otherwise, no change is made to the input buffer. In either case, an output sample is then calculated.

For each output sample, $Q = \frac{N}{L}$ multiplications and $Q - 1 = \frac{N}{L} - 1$ additions are required. Since each of the Q-tap FIR lowpass digital filters operates at the sampling rate $F\left(\frac{L}{M}\right)$, $F\left(\frac{N}{M}\right)$ multiplications and $F\left(\frac{N-L}{M}\right)$ are performed every second.

IMPLEMENTATION OF A MULTISTAGE SAMPLING RATE CONVERSION SYSTEM USING FIR STRUCTURES

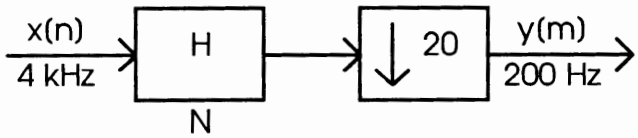
Multistage structures have proven to be very efficient when designing sampling rate conversion systems [2]. They are particularly useful in cases where the sampling rate factors L or M are much greater than one or they are approximately equal.

If a sampling rate conversion system has more than one stage, the ratio of transition band to sampling rate for each stage is greater than it is for a single stage. From Eq. (13), we can see that this ratio is inversely proportional to the estimated length of the lowpass filter. Consequently, the length of the lowpass filter for each stage will be less than the length of a lowpass filter for a single stage system. *In many cases, the sum of the lengths of the lowpass filters for all stages will be less than the length of the lowpass filter for a single stage.* This can significantly reduce the computational rates and required storage.

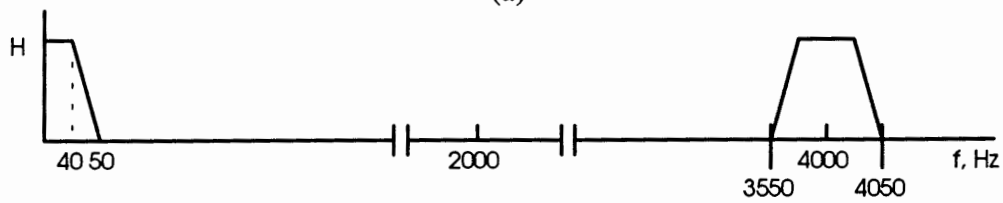
To demonstrate the effectiveness of the multistage design, consider the following example. Suppose that the input sampling rate is 4 kHz and the desired output sampling rate is 200 Hz. The decimation factor is $M = 4000 / 200 = 20$. Assume that the passband of the signal ranges from 0 to 40 Hz and the signal is attenuated for frequencies 50 Hz and above. Also, assume that the maximum allowable errors in the passband and stopband are 10^{-2} and 10^{-6}, respectively. We will now examine three different system designs.

Design 1

The first design is the single stage system as shown in Fig. 17(a). The frequency response of the lowpass filter is shown in Fig. 17(b). To estimate



(a)



(b)

Figure 17. One-stage decimator.

the lowpass filter length, we refer to Eq. (13).

$$\hat{N} = \frac{D_\infty(10^{-2},10^{-6})}{(50-40)/4000} - f(10^{-2},10^{-6})\frac{(50-40)}{4000} + 1$$

$$\cong 1,734 \text{ taps} \tag{49}$$

The required amount of storage is equal to the number of filter taps. The multiplication rate (assuming symmetry) is $(200)\left(\frac{1734}{2}\right) = 173,400$ mps (multiplications per second). The addition rate is $(200)(1734-1) = 346,600$ aps (additions per second).

Design 2

If we factor $M = 20$ into descending prime numbers [2], we have the structure shown in Fig. 18. In this case, there are three stages and each stage has a lowpass filter to prevent aliasing as the sampling rate is decreased. The passband

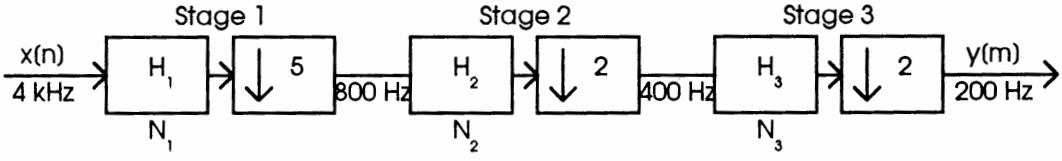


Figure 18. Three-stage decimator.

for each lowpass filter is the same; namely,

$$0 \le f \le f_p. \tag{50}$$

The stopband region varies depending upon the stage. If F_{i-1} is the incoming rate and F_i is the outgoing rate of the ith stage and we do not want aliasing in the final transition band, the stopband region can be described by the inequality

$$F_i - f_s \le f < \frac{F_{i-1}}{2}. \tag{51}$$

Thus, the transition bandwidth is

$$\Delta F_i = F_i - f_s - f_p. \tag{52}$$

For the last stage, however, the stopband region must be

$$f_s \le f < \frac{F_i}{2} \tag{53}$$

with the corresponding transition bandwidth

$$\Delta F_i = f_s - f_p. \tag{54}$$

The frequency responses of the filters are shown on the next page in Fig. 19.

For a multistage design, the overall passband magnitude response is the product of the magnitude responses of the stages [2]. If the lowpass filter of each stage i of an I-stage decimator satisfies the condition

$$1-\delta_p \le \left|H_i(e^{j2\pi f/F_i})\right| \le 1+\delta_p, \qquad f \in \text{passband}, \tag{55}$$

and if the responses align precisely in frequency in the passband (peak-to-peak in an equiripple design), we would have

$$(1-\delta_p)^I \le \left|H_0(e^{j2\pi f/F_0})\right| \le (1+\delta_p)^I, \qquad f \in \text{passband}. \tag{56}$$

Since δ_p is a small number, the above bound can be approximated as in the following inequality

$$1-I\delta_p \le \left|H_0(e^{j2\pi f/F_0})\right| \le 1+I\delta_p, \qquad f \in \text{passband}. \tag{57}$$

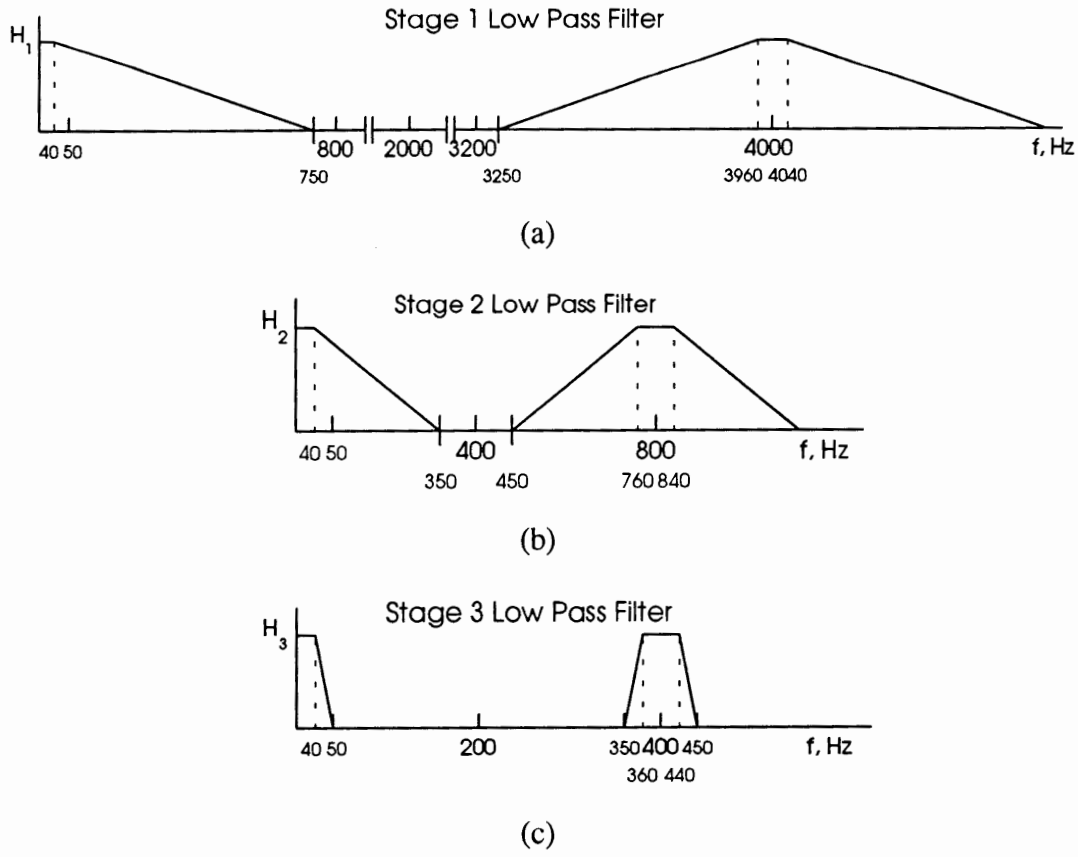


Figure 19. Frequency responses of lowpass filters for three-stage decimator.

Clearly, the passband ripple will increase with the number of stages. To avoid this increase, we define

$$\delta'_p = \frac{\delta_p}{I} \tag{58}$$

and replace δ_p by δ'_p in Eq. (55). Thus, we have

$$1-\delta'_p \le \left|H_i(e^{j2\pi f/F_i})\right| \le 1+\delta'_p, \qquad f \in \text{passband.} \tag{59}$$

This will guarantee the overall passband specification

$$1-\delta_p \le \left|H_0(e^{j2\pi f/F_0})\right| \le 1+\delta_p, \qquad f \in \text{passband.} \tag{60}$$

The modified error specification for each stage is $\delta'_p = \frac{10^{-2}}{3} = 3.33x10^{-3}$. A similar analysis for an interpolator lowpass filter with gain L is done in the Appendix.

Since the lowpass filter stopband ripple is less than one for each stage, any overlap of the stopbands will further reduce the overall stopband. Consequently, no adjustment is needed for the stopband specification

$$-\delta_s \le \left|H_0(e^{j2\pi f/F_0})\right| \le \delta_s, \qquad f \in \text{stopband}. \tag{61}$$

Estimates of the lowpass filter lengths can be determined from Eq. (13). We refer to Fig. 18 to determine the sampling rates for each of the lowpass filters. We can then use the formulas that we derived in the section on decimation and compute the computational rates and storage requirements. The complete specifications for the three-stage design are listed in TABLE II on the following page.

Design 3

If we examine TABLE II, we can see that stage 3 requires the longest lowpass filter length. In Design 3, we will show a way that can reduce the length by adding another stage consisting of one lowpass filter.

The filter length of the Ith stage of an I-stage sampling rate conversion system can be approximated by the following equation when the transition band is small compared to the sampling rate:

$$\hat{N}_I = \frac{D_\infty(\frac{\delta_p}{I}, \delta_s)}{(f_s - f_p)/F_{I-1}}. \tag{62}$$

Suppose we increase the number of stages to $I+1$ by adding a lowpass filter to the system. In that case, the filter length of the Ith stage is

$$\hat{N}'_I = \frac{D_\infty(\frac{\delta_p}{I+1}, \delta_s)}{(F_I - f_s - f_p)/F_{I-1}} \tag{63}$$

TABLE II

SPECIFICATIONS FOR A THREE-STAGE DECIMATOR

I	$\hat{N}_i$	M_i	F_{i-1}	F_i	f_{p_i}	f_{s_i}	δ_{p_i}	δ_{s_i}	mps	aps
1	25	5	4,000	800	40	750	$3.33x10^{-3}$	10^{-6}	10,400	19,200
2	12	2	800	400	40	350	$3.33x10^{-3}$	10^{-6}	2,400	4,400
3	191	2	400	200	40	50	$3.33x10^{-3}$	10^{-6}	19,200	38,000
Overall	228	20	4,000	200	40	50	10^{-2}	10^{-6}	32,000	61,600

Note: All frequencies are in Hz.

and the filter length of the $(I+1)$th stage is

$$\hat{N}'_{I+1} = \frac{D_\infty(\frac{\delta_p}{I+1}, \delta_s)}{(f_s - f_p)/F_I}. \tag{64}$$

If we add Eqs. (63) and (64), we have

$$\hat{N}'_I + \hat{N}'_{I+1} = \frac{D_\infty(\frac{\delta_p}{I+1}, \delta_s)}{(F_I - f_s - f_p)/F_{I-1}} + \frac{D_\infty(\frac{\delta_p}{I+1}, \delta_s)}{(f_s - f_p)/F_I}$$

$$= D_\infty(\frac{\delta_p}{I+1}, \delta_s) \cdot \left(\frac{F_{I-1}}{F_I - f_s - f_p} + \frac{F_I}{f_s - f_p} \right). \tag{65}$$

We want to determine the conditions that will ensure Eq. (65) is less than Eq. (63). Thus, we assume that

$$D_\infty(\frac{\delta_p}{I+1}, \delta_s) \cdot \left(\frac{F_{I-1}}{F_I - f_s - f_p} + \frac{F_I}{f_s - f_p} \right) < D_\infty(\frac{\delta_p}{I}, \delta_s) \cdot \left(\frac{F_{I-1}}{f_s - f_p} \right). \tag{66a}$$

From Eq. (14), we see that

$$D_\infty(\frac{\delta_p}{I}, \delta_s) \approx D_\infty(\frac{\delta_p}{I+1}, \delta_s). \tag{66b}$$

The inequality can be simplified as follows

$$\frac{F_{I-1}}{F_I - f_s - f_p} + \frac{F_I}{f_s - f_p} < \frac{F_{I-1}}{f_s - f_p}$$

$$\frac{f_s - f_p}{F_I - f_s - f_p} < \frac{F_{I-1} - F_I}{F_{I-1}} \tag{66c}$$

Since $F_{I-1} = D_I \cdot F_I$, we have the following inequality

$$\frac{f_s - f_p}{F_I - f_s - f_p} < \frac{D_I F_I - F_I}{D_I F_I}$$

$$\frac{f_s - f_p}{F_I - f_s - f_p} < \frac{D_I - 1}{D_I}$$

$$\frac{D_I}{D_I - 1} \cdot (f_s - f_p) < F_I - f_s - f_p$$

$$\left(\frac{2D_I - 1}{D_I - 1}\right) f_s - \left(\frac{1}{D_I - 1}\right) f_p < F_I. \tag{66d}$$

Thus, Eq. (66d) *is a sufficient condition for adding a lowpass filter to a multistage decimator to reduce the total number of filter taps required for the system*. The percent reduction in the required number of filter taps for the system is

$$\% \text{ Reduction} \cong \left(1 - \frac{\sum_{i=1}^{I-1} N_i' + D_\infty\left(\frac{\delta_p}{I+1}, \delta_s\right) \cdot \left(\frac{F_{I-1}}{F_I - f_s - f_p} + \frac{F_I}{f_s - f_p}\right)}{\sum_{i=1}^{I-1} N_i + D_\infty\left(\frac{\delta_p}{I}, \delta_s\right) \cdot \frac{F_{I-1}}{f_s - f_p}} \right) \cdot 100\% \tag{67}$$

As a direct consequence of the reduction in the total number of filter taps, the computational rates and storage requirements will also be reduced.

In our three-stage example,

$$\left(\frac{2D_I - 1}{D_I - 1}\right) f_s - \left(\frac{1}{D_I - 1}\right) f_p < F_I$$

$$\left(\frac{2\cdot 2-1}{2-1}\right)\cdot 50-\left(\frac{1}{2-1}\right)\cdot 40<200$$

$$110<200.$$

Therefore, adding a lowpass filter stage to the three-stage system will reduce the total number of filter taps. Referring to Eq. (67), we have

$$\%\ \text{Reduction} \cong \left(1-\frac{26+13+15+97}{25+12+191}\right)\cdot 100\%$$

$$\cong 34\%.$$

The designer should be aware that an additional lowpass filter may increase the group delay. In our example, the group delay increases approximately four samples from the three-stage design. See the Appendix for details. The four-stage design is shown below in Fig. 20.

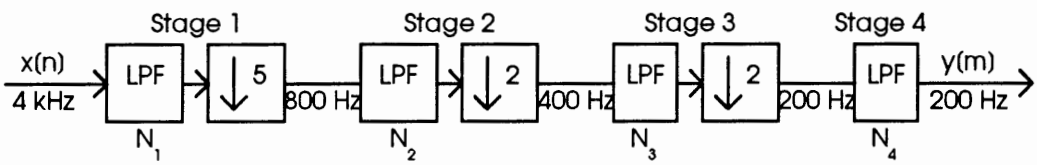


Figure 20. Four-stage decimator.

The complete specifications for the four-stage design are listed in Table III on the next page.

QUANTIZATION ERRORS IN THE DIRECT FORM FIR LINEAR PHASE DIGITAL FILTERS

Quantization is the process of transforming a sample of signal $x[n]$, whose value is one of a continuous set, into $\hat{x}[n]$, whose value is one of a discrete set and closest to the sample value. Quantization levels are usually spaced uniformly. In

TABLE III

SPECIFICATIONS FOR A FOUR-STAGE DECIMATOR

I	$\hat{N}_i$	M_i	F_{i-1}	F_i	f_{p_i}	f_{s_i}	δ_{p_i}	δ_{s_i}	mps	aps
1	26	5	4,000	800	40	750	$2.5x10^{-3}$	10^{-6}	10,400	20,000
2	13	2	800	400	40	350	$2.5x10^{-3}$	10^{-6}	2,800	4,800
3	15	2	400	200	40	150	$2.5x10^{-3}$	10^{-6}	1,600	2,800
4	97	1	200	200	40	50	$2.5x10^{-3}$	10^{-6}	9,800	19,200
Overall	151	20	4,000	200	40	50	10^{-2}	10^{-6}	24,600	46,800

Note: All frequencies are in Hz.

some cases, however, the spacing may be non-uniform. In addition, the sample values may be rounded or truncated to the nearest quantization level. For our discussion, we will assume that the quantization levels are uniformly spaced and that rounding is used to arrive at the nearest quantization level.

The difference between the quantized sample $\hat{x}[n]$ and the actual sample $x[n]$ is referred to as *quantization error* and is defined by the equation

$$e[n] = \hat{x}[n] - x[n]. \tag{68}$$

The quantization error range is

$$-\frac{Q}{2} < e[n] \leq \frac{Q}{2}, \tag{69}$$

where Q is the quantization step size that is defined by

$$Q = \frac{X_m}{2^B}. \tag{70}$$

X_m is the full-scale of the quantizer and B is the number of binary bits. Since $e[n]$is usually not known, a statistical model is useful in representing the effects of quantization. The underlying assumptions of this model are [9]

1. The error sequence $e[n]$ is a sample sequence of a stationary random process; i.e., the sequence properties do not vary with time.
2. The error sequence is uncorrelated with the sequence $x[n]$.
3. The random variables of the error process are uncorrelated; i.e., the error is a white-noise process.
4. The probability distribution of the error is uniform over the range of the quantization error.

There are three types of error which affect the performance of an FIR linear phase digital filter [10]. The first type occurs when the input samples to the filter are quantized. This is known as *A-D noise*. The second type occurs when the filter coefficients are quantized. Quantizing the results of arithmetic operations within the filter results is the third type of error, which is known as *roundoff noise*. Let us now examine each of these error types.

A-D Noise

Let the error process be defined by the RV (random variable) $\mathbf{e}_n$ with auniform distribution over $(-\frac{Q}{2},\frac{Q}{2})$. The mean of $\mathbf{e}_n$ is the expected value of $\mathbf{e}_n$ and can be determined as follows

$$\begin{aligned} m_{\mathbf{e}_n} &= \mathcal{E}\{\mathbf{e}_n\} \\ &= \int_{-\frac{Q}{2}}^{\frac{Q}{2}} e(\frac{1}{Q})de \\ &= \frac{e^2}{2Q}\Big|_{-\frac{Q}{2}}^{\frac{Q}{2}} \\ &= 0. \end{aligned} \tag{71}$$

The variance of $\mathbf{e}_n$ is

$$\sigma^2_{\mathbf{e}_n} = \mathcal{E}\{(\mathbf{e}_n - 0)^2\}$$

$$= \int_{-\frac{Q}{2}}^{\frac{Q}{2}} e^2 (\frac{1}{Q}) de$$

$$= \frac{e^3}{3Q}\Big|_{-\frac{Q}{2}}^{\frac{Q}{2}}$$

$$= \frac{Q^2}{12}. \tag{72}$$

If the input to an FIR N-tap filter is quantized, then the output noise due to the error is

$$y_{\mathrm{AD}}[n] = \sum_{k=0}^{N-1} h[k] e[n-k] \tag{73}$$

If $\mathbf{y}_{\mathbf{AD}}$ is the RV for the output, the mean of $\mathbf{y}_{\mathbf{AD}}$ is

$$\begin{aligned} m_{\mathbf{y}_{\mathbf{AD}}} &= \mathcal{E}\{\mathbf{y}_{\mathbf{AD}}\} \\ &= \mathcal{E}\{\sum_{k=0}^{N-1} h[k]\mathbf{e}_{\mathbf{n-k}}\} \\ &= \sum_{k=0}^{N-1} E\{h[k]\mathbf{e}_{\mathbf{n-k}}\} \\ &= \sum_{k=0}^{N-1} h[k] E\{\mathbf{e}_{\mathbf{n-k}}\} \\ &= 0. \end{aligned} \tag{74}$$

The variance of $\mathbf{y}_{\mathbf{AD}}$ is

$$\begin{aligned} \sigma^2_{\mathbf{y}_{\mathbf{AD}}} &= \mathcal{E}\{(\mathbf{y}_{\mathbf{AD}} - 0)^2\} \\ &= \mathcal{E}\{(\sum_{k=0}^{N-1} h[k]\mathbf{e}_{\mathbf{n-k}} - 0)^2\} \\ &= \mathcal{E}\{\sum_{k=0}^{N-1} (h[k]\mathbf{e}_{\mathbf{n-k}})^2 + 2\sum_{k=1}^{N-1} h[k]\mathbf{e}_{\mathbf{n-k}} (\sum_{j=0}^{k-1} h[j]\mathbf{e}_{\mathbf{n-j}})\} \\ &= \sum_{k=0}^{N-1} \mathcal{E}\{(h[k]\mathbf{e}_{\mathbf{n-k}})^2\} + 2\sum_{k=1}^{N-1} \mathcal{E}\{h[k]\mathbf{e}_{\mathbf{n-k}} (\sum_{j=0}^{k-1} h[j]\mathbf{e}_{\mathbf{n-j}})\} \\ &= \sum_{k=0}^{N-1} E\{(h[k]\mathbf{e}_{\mathbf{n-k}})^2\} + 2\sum_{k=1}^{N-1} \sum_{j=0}^{k-1} \mathcal{E}\{h[k]\mathbf{e}_{\mathbf{n-k}} h[j]\mathbf{e}_{\mathbf{n-j}}\} \end{aligned}$$

$$= \sum_{k=0}^{N-1} (h[k])^2 \mathcal{E}\{\mathbf{e}^2_{\mathbf{n-k}}\} + 2\sum_{k=1}^{N-1}\sum_{j=0}^{k-1} h[k]h[j]\mathcal{E}\{\mathbf{e}_{\mathbf{n-k}}\mathbf{e}_{\mathbf{n-j}}\}$$

$$= \sum_{k=0}^{N-1} (h[k])^2 \cdot \frac{Q^2}{12} + 2\sum_{k=1}^{N-1}\sum_{j=0}^{k-1} h[k]h[j]\cdot 0$$

$$= \frac{Q^2}{12}\sum_{k=0}^{N-1} (h[k])^2. \tag{75}$$

Using Parseval's equation [9], we can also state the variance in terms of the frequency as

$$\sigma^2_{y_{AD}} = \frac{Q^2}{12}\cdot\frac{1}{2\pi}\int_{-\pi}^{\pi} \left|H(e^{j\omega})\right|^2 d\omega. \tag{76}$$

In many applications, scaling is used as a preventative measure against overflow. Two of the most common methods are sum scaling and peak scaling. When sum scaling is used the following restriction is put on the filter coefficients

$$\sum_{k=0}^{N-1} (h[k])^2 \le 1. \tag{77}$$

When peak scaling is used, the frequency response is normalized so that

$$\max\left|H(e^{j\omega})\right| = 1. \tag{78}$$

Examining Eqs. (75) and (76), we see that

$$\sigma^2_{y_{AD}} \le \frac{Q^2}{12}. \tag{79}$$

<u>Filter Coefficient Quantization Noise</u>

Chan and Rabiner [10] reasoned that a statistical analysis of the filter coefficient quantization noise is appropriate even though the quantization of the coefficients for a given filter is done only once because of the unpredictability of the noise. The assumptions about the error process are the same as those previously mentioned; namely, the RV $\mathbf{e}_\mathbf{n}$ has a mean of 0 and a variance of $\frac{Q^2}{12}$.

Suppose that we have an N-tap FIR linear phase digital filter, where N is

odd. If we quantize the filter coefficients, the frequency response of the error due to quantization is

$$E(e^{j\omega}) = \sum_{k=0}^{\frac{N-3}{2}} 2e[k]\cos[(\frac{N-1}{2}-k)\omega] + e[\frac{N-1}{2}]. \tag{80}$$

The mean of the frequency response error is

$$\begin{aligned} m_{\mathbf{E}} &= \mathcal{E}\{\mathbf{E}\} \\ &= \mathcal{E}\{\sum_{k=0}^{\frac{N-3}{2}} 2\mathbf{e}_{\mathbf{k}} \cos[(\frac{N-1}{2}-k)\omega] + \mathbf{e}_{\frac{\mathbf{N-1}}{\mathbf{2}}}\} \\ &= \sum_{k=0}^{\frac{N-3}{2}} 2\mathcal{E}\{\mathbf{e}_{\mathbf{k}}\}\cos[(\frac{N-1}{2}-k)\omega] + \mathcal{E}\{\mathbf{e}_{\frac{\mathbf{N-1}}{\mathbf{2}}}\} \\ &= 0. \end{aligned} \tag{81}$$

The variance of the frequency response error is

$$\begin{aligned} \sigma_{\mathbf{E}}^2 &= \mathcal{E}\{(\mathbf{E}-0)^2\} \\ &= \mathcal{E}\{(\sum_{k=0}^{\frac{N-3}{2}} 2\mathbf{e}_{\mathbf{k}} \cos[(\frac{N-1}{2}-k)\omega] + \mathbf{e}_{\frac{\mathbf{N-1}}{\mathbf{2}}})^2\} \\ &= \mathcal{E}\{(\sum_{k=0}^{\frac{N-3}{2}} 2\mathbf{e}_{\mathbf{k}} \cos[(\frac{N-1}{2}-k)\omega])^2\} + \mathcal{E}\{\mathbf{e}_{\frac{\mathbf{N-1}}{\mathbf{2}}}{}^2\} \\ &\quad + \mathcal{E}\{2(\sum_{k=0}^{\frac{N-3}{2}} 2\mathbf{e}_{\mathbf{k}} \cos[(\frac{N-1}{2}-k)\omega])(\mathbf{e}_{\frac{\mathbf{N-1}}{\mathbf{2}}})\} \\ &= \mathcal{E}\{4\sum_{k=0}^{\frac{N-3}{2}} \mathbf{e}_{\mathbf{k}}^2 \cos^2[(\frac{N-1}{2}-k)\omega]\} + \frac{Q^2}{12} \\ &\quad + 0 \\ &= 4\sum_{k=0}^{\frac{N-3}{2}} \mathcal{E}\{\mathbf{e}_{\mathbf{k}}^2\}\cos^2[(\frac{N-1}{2}-k)\omega] + \frac{Q^2}{12} \end{aligned}$$

$$= 4\sum_{k=0}^{\frac{N-3}{2}} \frac{Q^2}{12}\cos^2[(\frac{N-1}{2}-k)\omega]+\frac{Q^2}{12}$$

$$= \frac{Q^2}{12}\cdot 4\sum_{k=0}^{\frac{N-3}{2}} \cos^2[(\frac{N-1}{2}-k)\omega]+\frac{Q^2}{12}$$

$$= \frac{Q^2}{12}\{4\sum_{k=0}^{\frac{N-3}{2}} \cos^2[(\frac{N-1}{2}-k)\omega]+1\} \tag{82}$$

Using the change of variables

$$n = \frac{N-1}{2}-k \tag{83}$$

in Eq. (82) results in

$$\sigma_{\mathrm{E}}^2 = \frac{Q^2}{12}\{4\sum_{n=1}^{\frac{N-1}{2}} \cos^2(n\omega)+1\}. \tag{84}$$

If the number of filter taps is even, we can use similar calculations to derive the mean and variance. The results are

$$m_{\mathrm{E}} = 0 \tag{85}$$

and

$$\sigma_{\mathrm{E}}^2 = \frac{Q^2}{3}\sum_{n=0}^{\frac{N-1}{2}} \cos^2[(\frac{N-1}{2}-n)\omega]. \tag{86}$$

We now define a weighting function

$$W_N(\omega) = \{\frac{1}{2N-1}\cdot[4\sum_{n=1}^{\frac{N-1}{2}} \cos^2(n\omega)+1]\}^{1/2} \tag{87}$$

and substitute Eq. (87) into Eq. (84). The result is

$$\sigma_{\mathrm{E}}^2 = \frac{Q^2}{12}(2N-1)W_N^2(\omega). \tag{88}$$

The standard deviation of the error is

$$\sigma_{\mathrm{E}} = \frac{Q}{2}\sqrt{\frac{2N-1}{3}}\cdot W_N(\omega). \tag{89}$$

To obtain a closed expression for $W_N(\omega)$, we rewrite Eq. (87) as

$$W_N(\omega) = \{\frac{1}{2N-1}\cdot[4\sum_{n=1}^{\frac{N-1}{2}}\frac{1}{2}(1+\cos(2n\omega))+1]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[(\mathrm{N}-1)+2\sum_{n=1}^{\frac{N-1}{2}}\cos(2n\omega)+1]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}+2\sum_{n=1}^{\frac{N-1}{2}}(\frac{e^{j2n\omega}+e^{-j2n\omega}}{2})]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}+\frac{e^{j(N+1)\omega}-e^{j2\omega}}{e^{j2\omega}-1}$$

$$+\frac{e^{-j(N+1)\omega}-e^{-j2\omega}}{e^{-j2\omega}-1}]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}+\frac{e^{j(N-1)\omega}-1-e^{j(N+1)\omega}+e^{j2\omega}}{1-e^{j2\omega}-e^{-j2\omega}+1}$$

$$+\frac{e^{-j(N-1)\omega}-1-e^{-j(N+1)\omega}+e^{-j2\omega}}{1-e^{j2\omega}-e^{-j2\omega}+1}]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}+\frac{2\cos(N-1)\omega-2}{2-2\cos 2\omega}$$

$$+\frac{-2\cos(N+1)\omega+2\cos 2\omega}{2-2\cos 2\omega}]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}-1+\frac{2\sin N\omega\sin\omega}{1-\cos 2\omega}]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}-1+\frac{2\sin N\omega\sin\omega}{2\sin^2\omega}]\}^{1/2}$$

$$= \{\frac{1}{2N-1}\cdot[\mathrm{N}-1+\frac{\sin N\omega}{\sin\omega}]\}^{1/2}. \tag{90}$$

From Eq. (90), we can see that $0 < W_N(\omega) \le 1$ for all N and that $W_N(\omega) = 1$ when $\omega = 0$ or π. Thus,

$$\sigma_{\mathbf{E}} \le \frac{Q}{2}\sqrt{\frac{2N-1}{3}}. \tag{91}$$

From Eq. (80), we have that **E** is a RV that is a linear combination of RVs from the error process in the time domain. Since the error process RVs are independent and 0 outside a finite interval $(-\frac{Q}{2},\frac{Q}{2})$, we can justify that **E** has an approximately normal distribution for large N, according to the central limit theorem [10].

We now return to the equiripple FIR linear phase digital filter design. Recall that design specifications consist of a set of disjoint frequency bands $\Omega_k \subset [0,\pi]$, where $k = 1,\ldots,P$ (some finite number), and a set of error bounds δ_k for a polynomial which is equiripple in the frequency bands and approximates the desired frequency response $H_{dk}(e^{j\omega})$. Thus, we have

$$\max_{\omega\in\Omega_k}\left|H_d(e^{j\omega}) - A(e^{j\omega})\right| = \delta_k, \quad \mathrm{k} = 1,\ldots,\mathrm{P}, \tag{92}$$

where $A(e^{j\omega})$ is the approximating polynomial. If we denote the quantized approximate frequency response by $A_Q(e^{j\omega})$, we have

$$\begin{aligned}\max_{\omega\in\Omega_k}\left|H_d(e^{j\omega}) - A_Q(e^{j\omega})\right| &= \max_{\omega\in\Omega_k}\left|H_d(e^{j\omega}) - A(e^{j\omega}) + A(e^{j\omega}) - A_Q(e^{j\omega})\right| \\ &\le \max_{\omega\in\Omega_k}\left|H_d(e^{j\omega}) - A(e^{j\omega})\right| + \max_{\omega\in\Omega_k}\left|A(e^{j\omega}) - A_Q(e^{j\omega})\right| \\ &\le \delta_k + \max_{\omega\in\Omega_k}\left|E(e^{j\omega})\right| \end{aligned} \tag{93}$$

Since the RV **E** has an approximately normal distribution, we have

$$\Pr\{\mathbf{E} \le 2\sigma_{\mathbf{E}}\} = 0.95. \tag{94}$$

Therefore, we can state with 95% confidence that

$$\begin{aligned}\max_{\omega\in\Omega_k}\left|H_d(e^{j\omega}) - A_Q(e^{j\omega})\right| &\le \delta_k + 2\sigma_E \\ &\le \delta_k + Q\sqrt{\frac{2N-1}{3}}. \end{aligned} \tag{95}$$

If we restrict the peak-to-peak amplitude of the input signal to be 1, the quantization step size will be $Q = 2^{-B}$. Substituting this value into Eq. (95) will allow us to determine the minimum number of bits required to ensure that filter coefficient quantization is negligible within a certain percentage range for a given frequency band. For example, if we want the filter coefficient quantization effects to be negligible within $\pm 1\%$, we have

$$(\delta_k + Q\sqrt{\frac{2N-1}{3}}) - \delta_k = 0.01\delta_k$$

$$2^{-B}\sqrt{\frac{2N-1}{3}} = 0.01\delta_k$$

$$2^{B} = \frac{100}{\delta_k}\sqrt{\frac{2N-1}{3}}$$

$$B = \log_{10}\left(\frac{100}{\delta_k}\sqrt{\frac{2N-1}{3}}\right)/\log_{10} 2$$

$$= \frac{2+\log_{10}\sqrt{\frac{2N-1}{3}} - \log_{10}\delta_k}{\log_{10} 2}. \tag{96}$$

Roundoff Noise

As in our discussion on A-D noise, the output will be affected by the error process $\mathbf{e_n}$, which has mean 0 and variance $\frac{Q^2}{12}$. If we denote the error at the output due to roundoff noise as the RV $\mathbf{y_{RO}}$, we have a mean

$$m_{\mathbf{y_{RO}}} = 0 \tag{97}$$

by using similar calculations as the ones we used in the A-D noise section. If we have a N-tap FIR linear phase filter, where N is odd, and the quantization takes place between the multiplications and the additions of the filter, there are $\frac{N+1}{2}$ white-noise sources at the output. In this case, the variance is

$$\sigma^2_{\mathbf{y_{RO}}} = \left(\frac{N+1}{2}\right)\frac{Q^2}{12}. \tag{98}$$

If the quantization takes place after the additions and a double-length accumulator is used, the variance will be

$$\sigma^2_{\mathbf{y_{RO}}} = \frac{Q^2}{12}. \tag{99}$$

If N is even, there will be $\frac{N}{2}$ white-noise sources at the output and the variance will change accordingly.

Conclusions

The A-D noise and the roundoff noise both can reduced to $\frac{Q^2}{12}$. The filter coefficient quantization noise, however, cannot be reduced to that level as is shown in Eq. (88). Consequently, the most significant quantization noise for the direct form FIR filter is the noise that results from filter coefficient quantization.

If we use the transposed direct form structure or the polyphase form structure, the A-D noise and roundoff noise will remain unchanged because they are constants for a given value of Q [2]. Also, they will not be affected by an expander or a compressor. The filter coefficient quantization noise will be unchanged because the frequency response of the system is the same as the one resulting from the direct form structure.

A NOVEL PC PROGRAM FOR DESIGNING MULTISTAGE SAMPLING RATE CONVERSION SYSTEMS

Utilizing the previously discussed digital signal processing concepts, we have written a PC program that will determine an optimum multistage design for a given sampling rate conversion. The program consists of an input section, an analysis section, and an output section. Required data include the following: initial sampling rate, final sampling rate, passband frequency, stopband frequency, maximum passband error, maximum stopband error, and two weighting factors for optimizing the design.

We will explain the program by way of an example. Suppose that we are in the field of professional audio and we want to design a sampling rate conversion system for changing the CD sampling rate of 44.1 kHz to the DAT sampling rate of 48 kHz. Typical specifications are a passband frequency of 20 kHz, a stopband frequency of 22 kHz, a maximum error in the passband of 10^{-2}, and a maximum error of 10^{-6} in the stopband.

Once the data has been entered, the program reduces the ratio of the initial sampling rate to the final sampling rate until no common factors exist in the numerator and denominator. The numerator and denominator are then factored to products of prime numbers and arranged so that the numerator factors are increasing, since they represent interpolator stages, while the denominator factors are decreasing, since they represent decimator stages [2]. This determines the maximum number of stages for the system.

The optimizing factor is determined by multiplying the first weighting factor by the total additions per second and the second weighting factor by the total storage and summing them. The total additions per second and storage are determined by the filter lengths and operating frequencies of each stage. The filter lengths are limited to 400 by the program.

A new design is created by combining adjacent stages. A new optimizing factor is then calculated and compared to the original optimizing factor. If the new optimizing factor is less than the old, the new design is the optimum design. Otherwise, the old design is still the optimum design. This process of combining stages and comparing optimizing factors continues until further reduction would only increase the optimizing factor. This process lasts less than one second on a 33 MHz 486-based PC and is shown in Fig. 21.

Figure 21. Input data and optimum design factors.

A block diagram of the design is shown on the next page in Fig. 22.

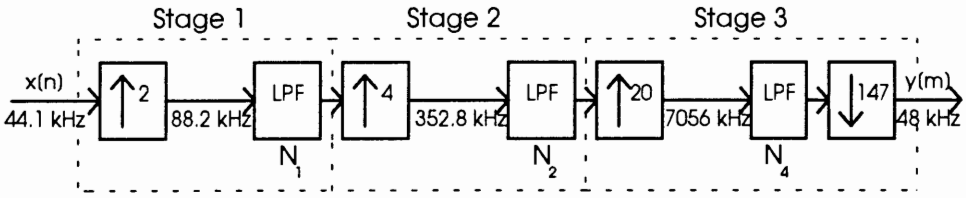


Figure 22. Three-stage CD-to-DAT sampling rate conversion system.

Once the minimum design is determined, extensive data from the design is available to the user as shown in Fig. 23. The user can change the design

Multi Stage Filter Design

NUMBER OF COMPUTATIONS FOR TRANSPOSED FORM

Sample Rate Conversion by L: MPS = In S.R. * N. APS = In S.R. * (N-L).
Sample Rate Conversion by 1/M: MPS = Out S.R. * N/2 if N is even or (N+1)/2 if N is odd. APS = Out S.R. * (N-1).
Sample Rate Conversion by L/M: MPS = In S.R. * 1/M * N.
APS = In S.R. * 1/M * (N-L).

Stage	est. N	(Hz) in S.R.	L	M	(Hz) out S.R.	Mult/sec	Add/sec	Storage
1	286	44100	2	1	88200	12612600	12524400	286
2	52	88200	4	1	352800	4586400	4233600	52
3	179	352800	20	147	48000	429600	381600	179
Total						17628600	17139600	517

If you want to examine the filter coefficient quantization effects, type Y. If you want to try another design, type N. Press <ENTER> after you make your choice. y

Type the number of bits to quantize the filter coefficients and press <ENTER>: 24

Figure 23. Analysis data for optimum design.

if the optimum one is not satisfactory and a new table will be displayed. If the user is satisfied with the initial design, he or she can examine the effects on the maximum frequency band errors caused by filter coefficient quantization. This is shown on the next page in Fig. 24.

Multi Stage Filter Design

REQUIRED NUMBER OF BITS TO COMPENSATE FOR FILTER COEFFICIENT QUANTIZATION

Original δp = 1.0E-0002, original δs = 1.0E-0006.
Maximum δp after quantization to be within 1% of the original δp with a 95% confidence interval for each stage. Note, the overall δp will be less than listed if the filter lengths are different for each stage and the ripples do not align. Also, the overall δs will be much less than listed where the stopbands overlap since δs for each stage is less than 1.

Stage	est. N	gain	delta p	delta s	(incl. sign bit) coeff. bits	24 bit delta p	24 bit delta s
1	286	2	0.000078125	0.000001000	26	0.000079783	0.000002658
2	52	4	0.000078125	0.000001000	24	0.000078853	0.000001728
3	179	20	0.000078125	0.000001000	25	0.000079424	0.000002299
Overall less than:			0.010000000	0.000001000		0.010172290	0.000002658

If you want to change the number of bits, type Y. Otherwise, type N. Press <ENTER> after you make your choice. n

If you want to examine the filter design specifications, type Y. If you want to change the original specifications, type N. Press <ENTER> after you make your choice. y

Figure 24. Effects of filter coefficient quantization.

The required number of bits ensures that the quantization effects are negligible. The user can experiment with different numbers of bits to see the effects on the maximum band errors. Note that the overall passband error is maximized when the equiripples in the passband from each stage align perfectly. This can only occur if all of the filters are the same, which is not possible. Thus, the ripples will beat against one another and the overall passband error will less than the maximum. The overall stopband error will be less than the lowest frequency stopband because all of the stopband errors are less than one.

If the user finds the filter coefficient quantization effects unacceptable, he or she may return to the beginning of the program and modify the specifications or the design. Otherwise, the user may examine the lowpass filter design specifications as shown in Fig. 25. Note that the stopband frequency of the first stage is identical to the overall stopband frequency.

Multi Stage Filter Design

DESIRED LOWPASS FILTER SPECIFICATIONS

Stage	Sampling Rate	Fp	Fs	gain	delta p	delta s
1	88200	20000	22000	2	0.000078125	0.000001000
2	352800	20000	66200	4	0.000078125	0.000001000
3	7056000	20000	330800	20	0.000078125	0.000001000

Press <ENTER> when you are ready to calculate and save the filter coefficients. ...

Figure 25. Lowpass filter design specifications.

Let us examine the frequency domain to see how the stopbands are determined. The passband of the input signal ranges from 0 to 20 kHz, while the stopband frequency is 22 kHz. The spectra of the signal are harmonic images of the baseband (-22.05 k Hz to 22.05 kHz) centered about integer multiples of the incoming sampling rate (44.1 kHz) and are shown in Fig. 26. The frequency

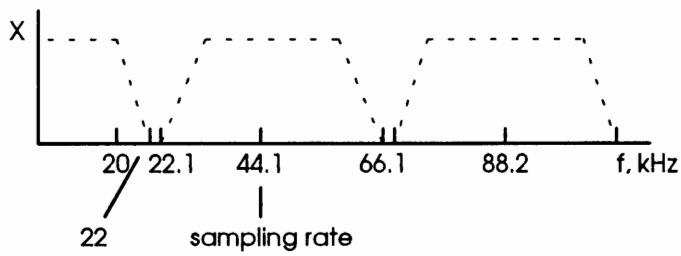


Figure 26. Spectra of the digital input signal.

responses of the lowpass filter stages are shown on the following page. Since our program restricts the input stopband frequency to be less than $\min\left(\frac{F_{in}}{2}, \frac{F_{out}}{2}\right)$,

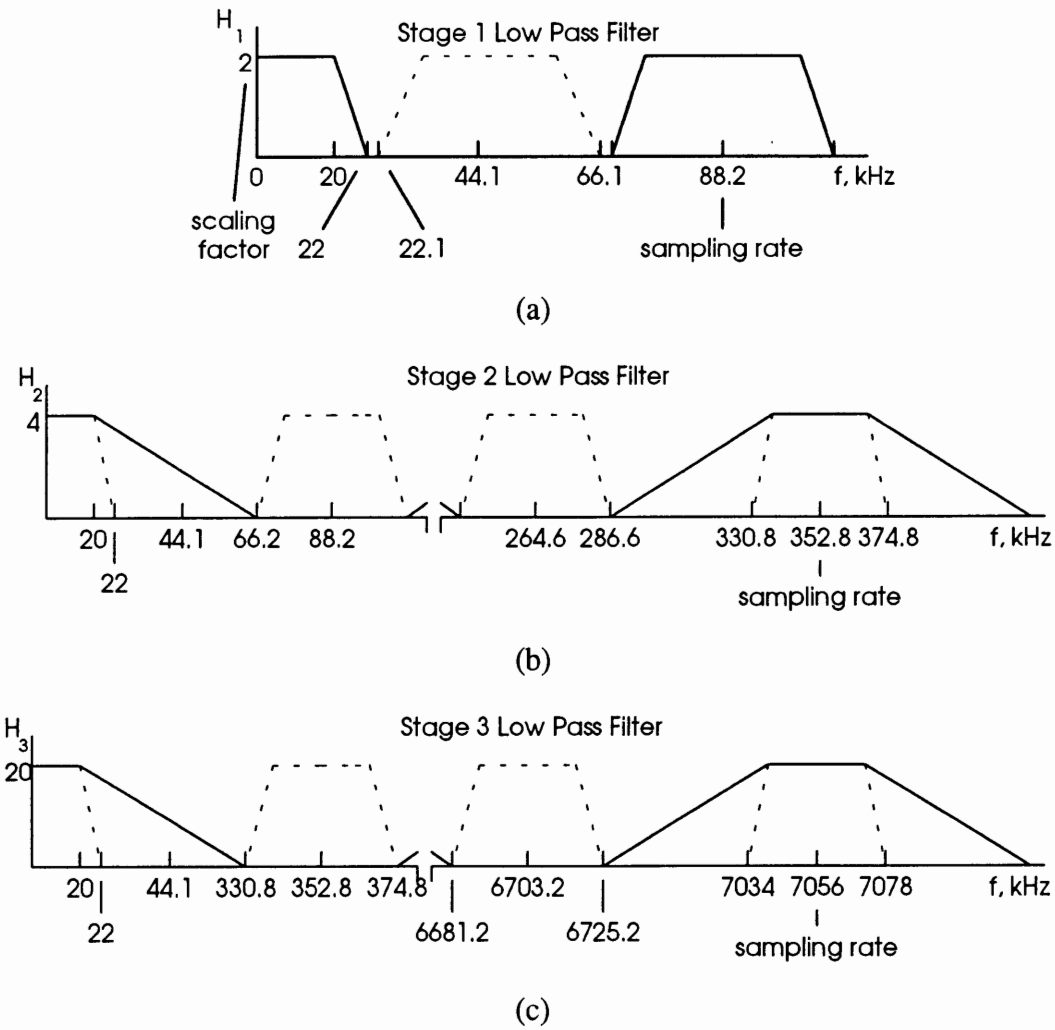


Figure 27. Frequency responses of the lowpass filters for a three-stage CD-to-DAT sampling rate conversion.

there is no danger of aliasing after the sampling rate is downsampled. Thus, it is unnecessary to factor M into multiple stages. The only decision to be made is whether to add a lowpass filter stage or not. In our example, the system stopband frequency requirement is met in the first stage. Consequently, no additional lowpass filter is needed. This approach is not always the optimum one when converting by a rational factor $\frac{L}{M}$, however. See the appendix for the more general case.

The next screen allows the user to save the filter coefficients to separate files. The actual length of each lowpass filter is determined using the algorithm that we proposed in Fig. 5. After each determination, the program calculates the filter coefficients and saves them to a file. Up until now, the program has responded instantaneously to the screen dialog. Calculation of the filter coefficients can take from a few seconds to several minutes, depending upon the number of stages and the actual filter lengths. The program estimates the time it takes when it is running on a 33 MHz 486-based PC.

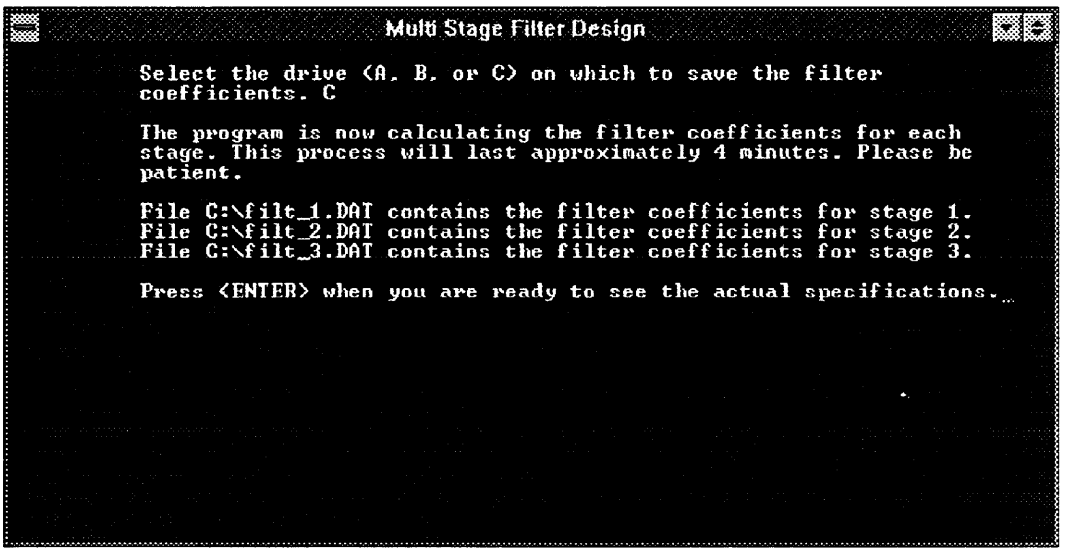


Figure 28. Lowpass filter coefficients are saved to files.

The next screen displays the actual filter lengths and maximum errors, along with other quantization data. The user can again analyze the effects of filter coefficient quantization for different numbers of bits.

Multi Stage Filter Design

ACTUAL LOWPASS FILTER SPECIFICATIONS

(Required number of bits includes the sign bit)

Stage	N	Sample Rate	Fp	Fs	gain	delta p	delta s	bits
1	288	88200	20000	22000	2	0.000071842	0.000000920	26
2	54	352800	20000	66200	4	0.000076998	0.000000986	24
3	169	7056000	20000	330800	20	0.000061982	0.000000793	26

Actual is less than:	0.009323270	0.000000920
Required is less than:	0.010000000	0.000001000

Type the number of bits to quantize the actual filter coefficients and press <ENTER>: 24

Figure 29. Actual filter lengths and maximum band errors.

The final screen allows the user to continue analyzing the quantization effects or exit the program.

Multi Stage Filter Design

24 bit delta p	24 bit delta s
0.000073505	0.000002583
0.000077739	0.000001727
0.000063244	0.000002055
0.009496078	0.000002583
0.010000000	0.000001000

If you want to change the number of bits, type Y. Otherwise, type N. Press <ENTER> after you make your choice. n

Press <ENTER> when you are ready to exit the program.

Figure 30. Effects of filter coefficient quantization and exit.

A PC PROGRAM FOR IMPLEMENTING A CD-TO-DAT SAMPLING RATE CONVERSION

PROGRAM DESCRIPTION

In the last section, we designed a system for converting a CD sampling rate to a DAT sampling rate. How well does it satisfy our design specifications? To answer this question, we have written another PC program that implements the conversion using the filter coefficients that are the output of the multistage design program. We will step through the program and explain the output as we go.

The input screen is shown below in Fig. 31. The user can choose from one

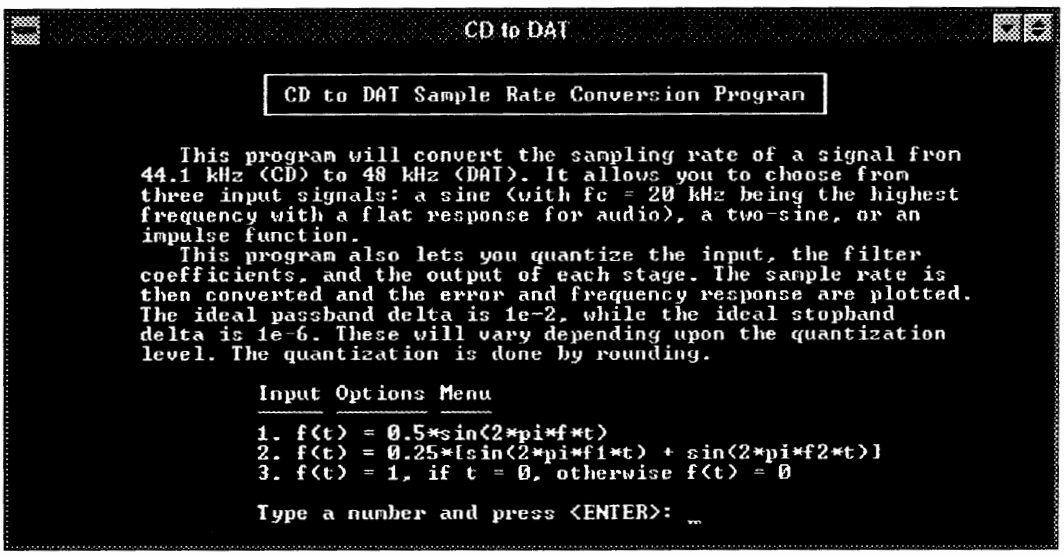


Figure 31. Input for the CD-to-DAT sampling rate conversion program.

of three input signals to test the conversion. If option 1 is selected, the user must enter a frequency from the passband. If option 2 is selected, the user must enter two frequencies from the passband. Option 3 is a unit sample sequence.

The user next decides whether to quantize the input and the filter coefficients and output of each stage. Once the user makes a selection, the program generates the appropriate signal and samples it at the CD sampling rate of 44.1 kHz. After 11 blocks of 147 samples ($\frac{L}{M} = \frac{160}{147}$) are collected, the program converts the sequence to 11 blocks of 160 samples, with a resulting sampling rate of 48 kHz. The frequency response is then plotted on the screen. Finally, the user may save the sampled output sequence to a file. The total program takes approximately 15 seconds on a 486-based PC with a math coprocessor.

We will now demonstrate the program through two examples.

SAMPLING A 10 KHZ SINE WAVE

Sampling Rate Conversion

Suppose that we select option 1 as our input, where the frequency of the signal is 10 kHz and we choose not to quantize. The program samples the signal at 44.1 kHz and reads the filter coefficients that were saved to files earlier. The sampling rate conversion follows as shown in Fig. 32 on the next page. The first stage of the conversion interpolates every input sample by a factor of 2. The second stage then interpolates every output sample from the first stage by a factor of 4. The process is repeated $M = 147$ times, so that there are $L_1 \cdot L_2 \cdot M = 2 \cdot 4 \cdot 147 = 1{,}176$ samples at the beginning of the third stage. The third and final stage uses the polyphase implementation to interpolate every 147th sample by a factor of 20, resulting in 160 samples. A flowchart of the algorithm is shown on the following page in Fig. 33.

```
CD to DAT
1. f(t) = 0.5*sin(2*pi*f*t)
2. f(t) = 0.25*[sin(2*pi*f1*t) + sin(2*pi*f2*t)]
3. f(t) = 1, if t = 0, otherwise f(t) = 0

Type a number and press <ENTER>: 1

Type the frequency of the sine wave and press <ENTER>: 10e3

Do you want to quantize? Enter Y or N and press <ENTER>: n

The program will now convert the sample rate. Please be patient.

The input has been sampled and the coefficient files have been read.

Block 1 of 11 blocks of input data has been converted.
Block 2 of 11 blocks of input data has been converted.
Block 3 of 11 blocks of input data has been converted.
Block 4 of 11 blocks of input data has been converted.
Block 5 of 11 blocks of input data has been converted.
Block 6 of 11 blocks of input data has been converted.
Block 7 of 11 blocks of input data has been converted.
Block 8 of 11 blocks of input data has been converted.
Block 9 of 11 blocks of input data has been converted.
Block 10 of 11 blocks of input data has been converted.
Block 11 of 11 blocks of input data has been converted.
```

Figure 32. Sampling rate conversion for a 10 kHz signal with no quantization.

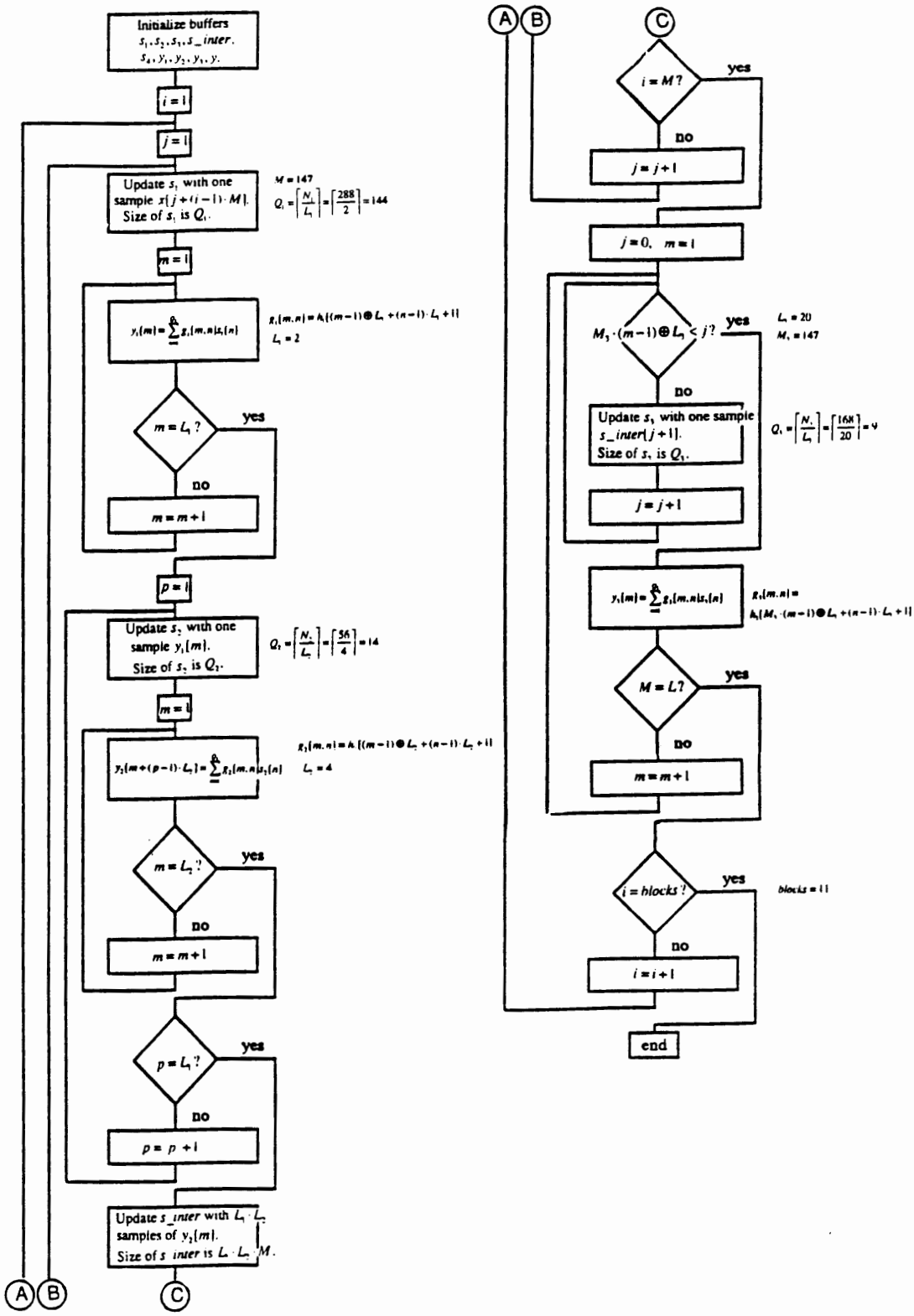


Figure 33. Three-stage sampling rate conversion algorithm.

Windowing

Transparent to the user, the program eliminates the first two blocks of the output sequence because of the group delay associated with the system (more about that later). The remaining samples of the output sequence are multiplied by a 4-term Blackman-Harris window sequence [11] to smooth the boundaries of the sequence and thus reduce spectral leakage in the frequency domain.

Frequency Response

A practical tool for analyzing the frequency content of a finite-duration sequence is the *discrete Fourier transform* (DFT), which is given by

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n] e^{-j(\frac{2\pi}{N})kn}, & 0 \le k \le N-1 \\ 0, & \text{otherwise.} \end{cases} \tag{100}$$

The DFT represents samples of the discrete time Fourier transform (DTFT), which is continuous and periodic in ω with period 2π.

A requirement of the DFT is that $x[n]$ and $X[k]$ have the same length, N. If the length of $x[n]$ is less than the length of $X[k]$, we can always pad $x[n]$ with zeroes. If the length of $x[n]$ is greater than the length of $X[k]$, we can use other techniques to make them equal [9].

If we compute the DFT using Eq. (100), N complex multiplications and $N-1$ complex additions are required to compute each $X[k]$. The total computations required for the DFT sequence of length N is N^2 complex multiplications and $N \cdot (N-1)$ additions. If we use symmetry and the periodicity of $e^{-j(\frac{2\pi}{N})kn}$, we can decompose Eq. (100) into successively smaller DFT computations [9]. This technique is known as the *decimation-in-time fast Fourier transform (FFT) algorithm* and results in significantly fewer computations than the direct method. In the case where N is an integer power of 2, $N \log_2 N$ complex

multiplications and $(N-1)\cdot\log_2 N$ complex additions are required to compute the DFT sequence.

The program uses a 2,048-point FFT to calculate the frequency response. Since the windowed sequence has 1,440 points, it is padded with zeroes so that the 2,048-point FFT can be performed. The frequency response is shown below in Fig. 34, where we see that the frequency content in the stopband is less than the required -120 dB.

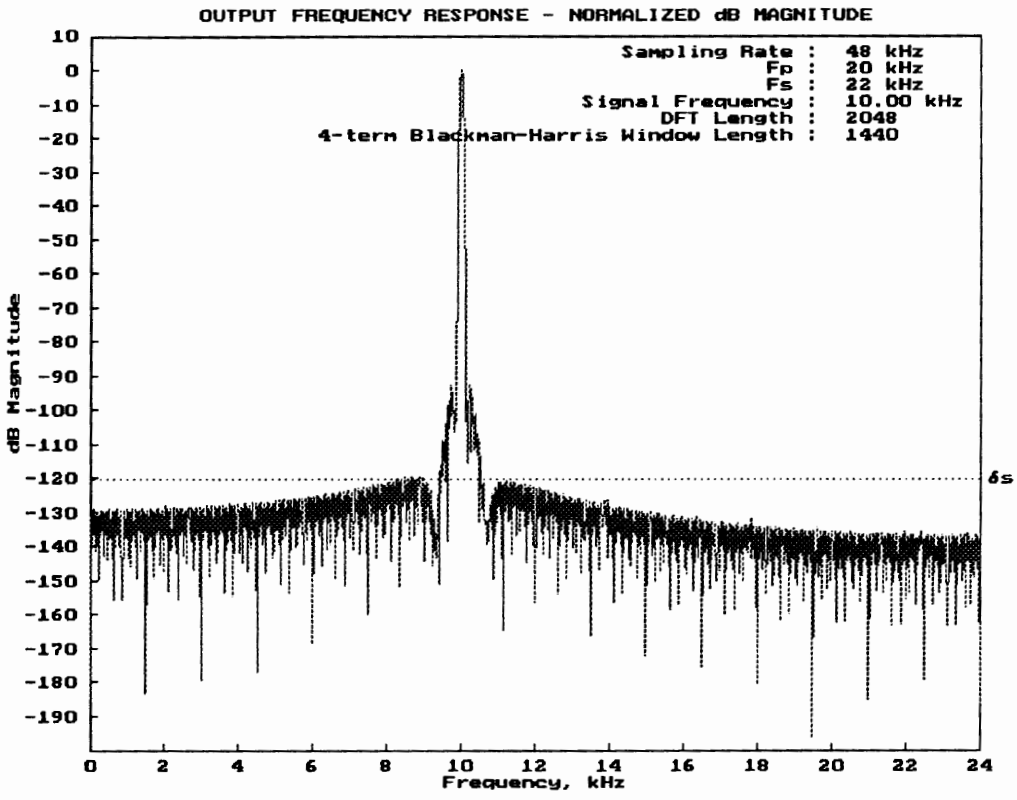


Figure 34. Frequency response for a 10 kHz signal with no quantization.

Quantization

If we had chosen to quantize, we would have had the option to quantize the input and the filter coefficients and output of each stage using from 8 to 32 bits (including the sign bit). Suppose that we quantize the input using 16 bits, the filter

coefficients using 24 bits, and the output using 24 bits. The frequency response is shown below in Fig. 35. Again, we see that the frequency content in the stopband is less than the required -120 dB.

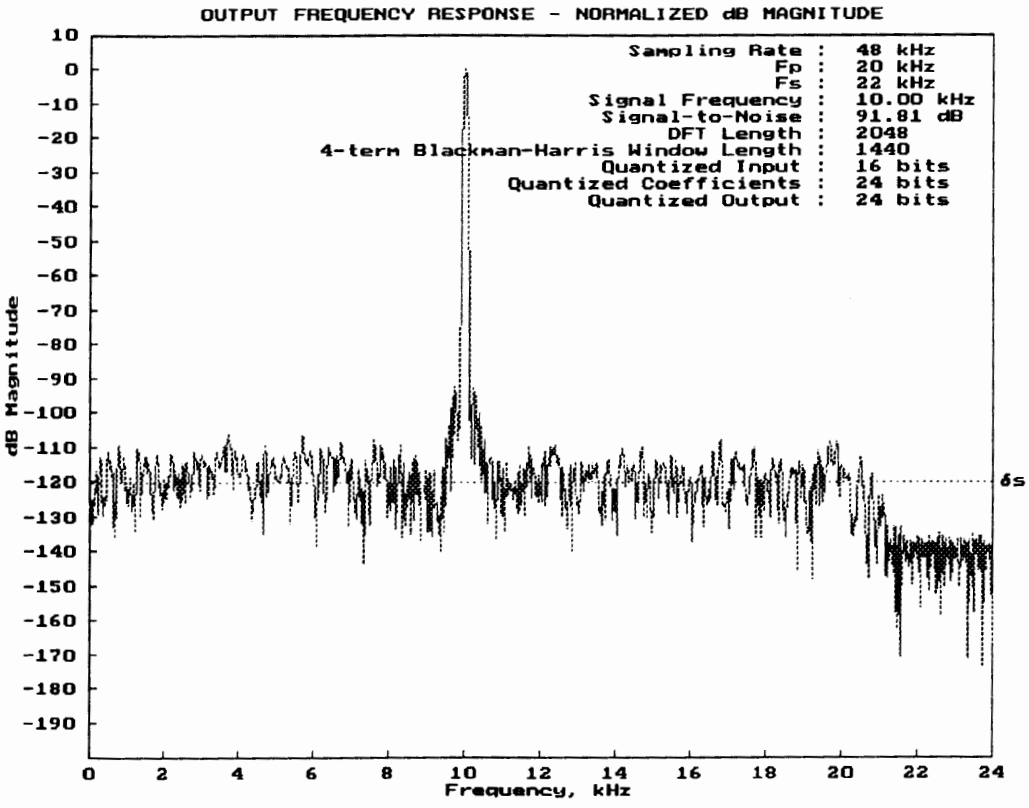


Figure 35. Frequency response for a 10 kHz signal with quantization.

Audio systems typically require a minimum signal-to-noise ratio (SNR) of 90 dB. One way to estimate the SNR is the following procedure. The total output power is given by the expression

$$\mathcal{E}\{\mathbf{y}_{\text{Total}}{}^2\} = \sum_{n=0}^{N-1} \left|y[n]\right|^2 , \tag{101}$$

where **y** is the RV for expressing the output of the sampling rate conversion system. According to Parseval's Equation for the DFT [9], the total output power can also be expressed as

$$\mathcal{E}\{\mathbf{y}_{\mathbf{Total}}{}^2\} = \frac{1}{N}\sum_{k=0}^{N-1}\left|Y[k]\right|^2, \tag{102}$$

The program estimates the total noise power by summing the total power except for a small number of values centered about and including the tone frequency (10 kHz), averaging that number, and multiplying the average by the total number of points (1,024) in the spectrum. The result can be used to estimate

$$\mathrm{SNR} = 10\log_{10}\left[\frac{\mathcal{E}\{\mathbf{y}_{\mathbf{Total}}{}^2\} - \mathcal{E}\{\mathbf{y}_{\mathbf{Noise}}{}^2\}}{\mathcal{E}\{\mathbf{y}_{\mathbf{Noise}}{}^2\}}\right]. \tag{103}$$

We see in Fig. 35 that the SNR is more than the required 90 dB.

OVERALL FREQUENCY RESPONSE OF THE THREE-STAGE SYSTEM

Sampling Rate Conversion

If we select option 3 as our input, we will obtain the overall frequency response of the three-stage system. The frequency response magnitude and phase are shown in Figs. 36 - 38.

Group Delay

Let us return our attention to the concept of group delay, which is defined by the equation [9]

$$\mathrm{grd}[H(e^{j\omega})] = -\frac{d}{d\omega}\{\arg[H(e^{j\omega})]\}. \tag{104}$$

For a type I FIR linear phase digital filter defined by

$$H(e^{j\omega}) = e^{-j\omega(\frac{N-1}{2})}\sum_{k=0}^{\frac{N-1}{2}} a[k]\cos(\omega k), \tag{105}$$

it is easy to see that the group delay is $\frac{N-1}{2}$ samples. A type II filter also has the same group delay. Since the multistage design program uses either a type I or a type II FIR linear phase digital filter, the lowpass filter of each stage will have a

group delay of $\frac{N-1}{2}$ samples associated with it.

At stage i of a multistage system, upsampling by L_i will increase the group delay of any lowpass filter preceding it by a factor of L_i. In a similar manner, downsampling by M_i will decrease the group delay of any lowpass filter preceding it by a factor of $\frac{1}{M_i}$. The group delays of each stage will be additive and the overall group delay of the system is

$$\begin{aligned}\tau_g &= \frac{1}{M}\left[\tau_{g_3} + L_3(\tau_{g_2} + L_2\tau_{g_1})\right] \\ &= \frac{1}{M}\left[\frac{N_3-1}{2} + L_3\left(\frac{N_2-1}{2} + L_2\left(\frac{N_1-1}{2}\right)\right)\right] \\ &= \frac{1}{147}\cdot\left[\frac{169-1}{2} + 20\cdot\left(\frac{54-1}{2} + 4\cdot\left(\frac{288-1}{2}\right)\right)\right] \\ &\cong 82 \text{ samples.} \qquad (106)\end{aligned}$$

The group delay vs frequency *for the passband* is plotted in Fig. 38. The values were calculated using Eq. (104) and they are the same as the theoretical value calculated in Eq. (106).

Quantization

If we quantize the input, filter coefficients, and output as we did before, we have the response as shown in Figs. 39 - 41.

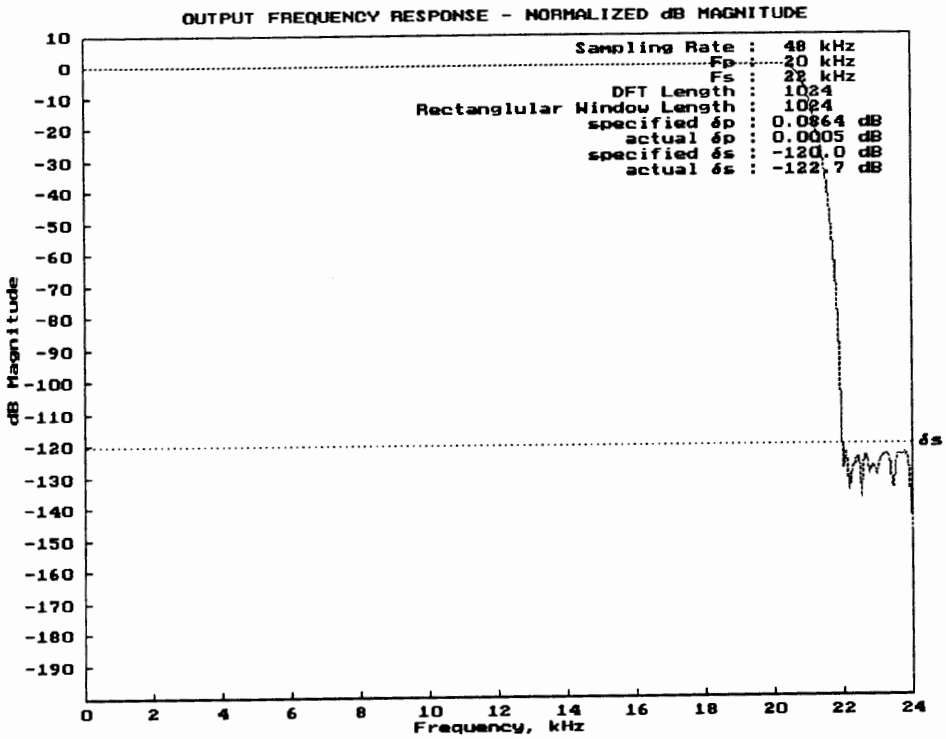


Figure 36. Magnitude response of the system with no quantization.

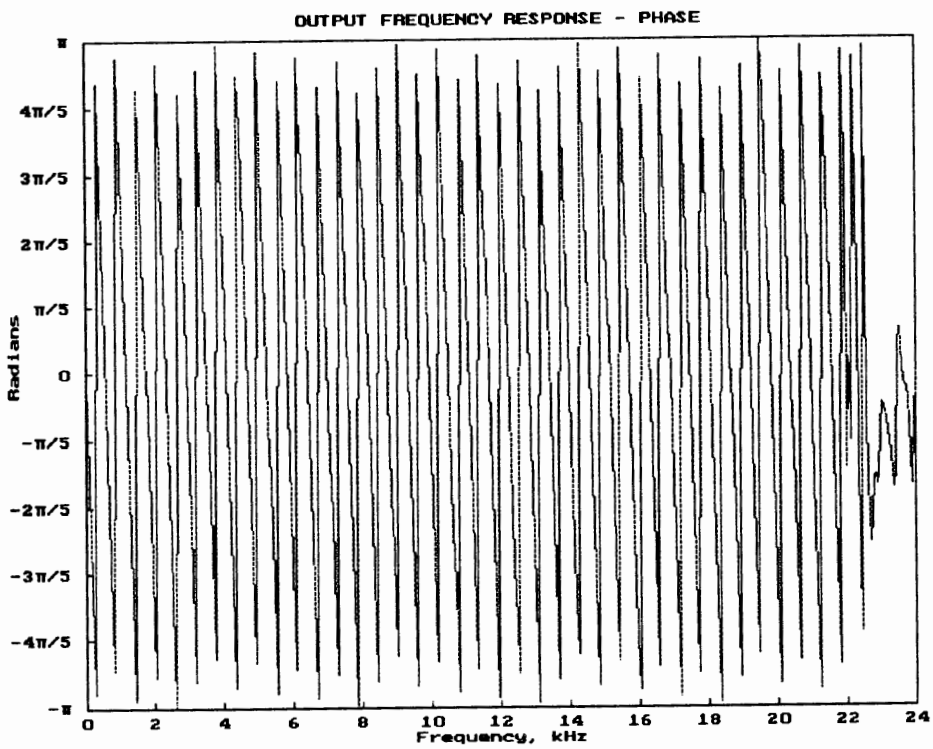


Figure 37. Phase response of the system with no quantization.

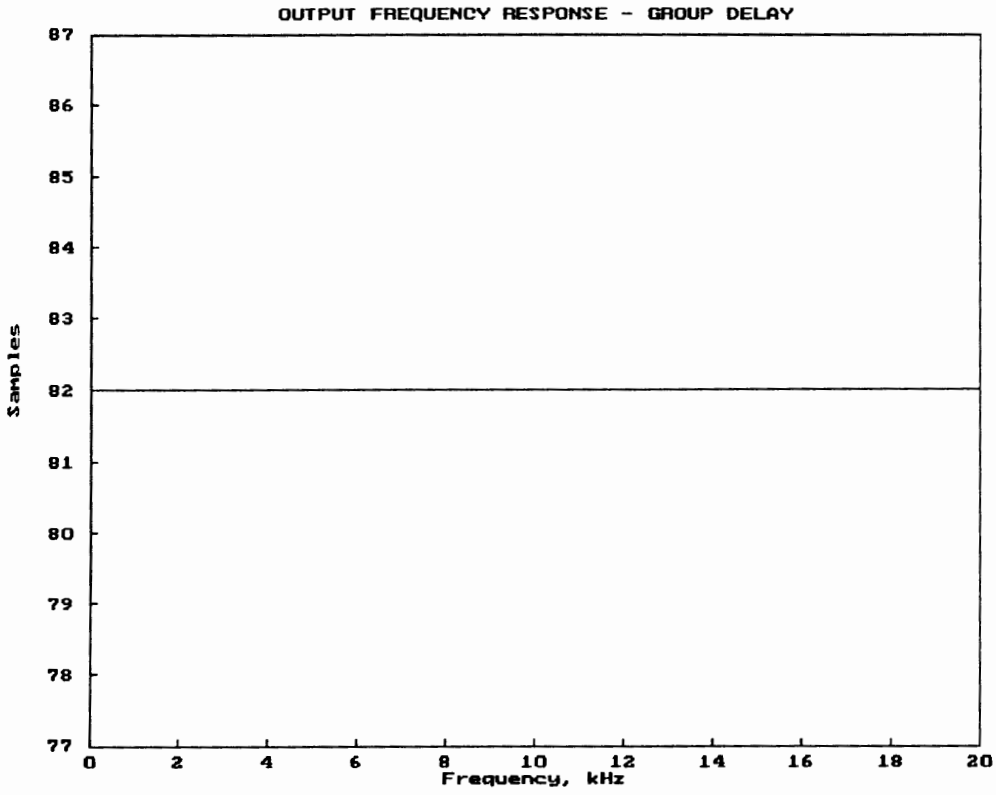


Figure 38. Group delay of the system with no quantization.

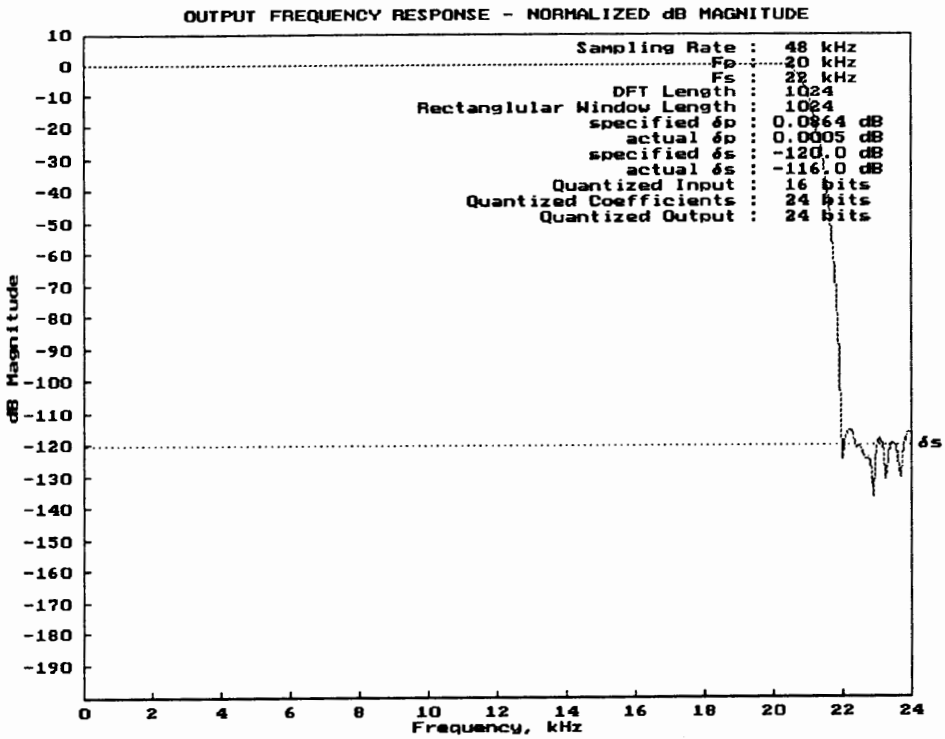


Figure 39. Magnitude response of the system with quantization.

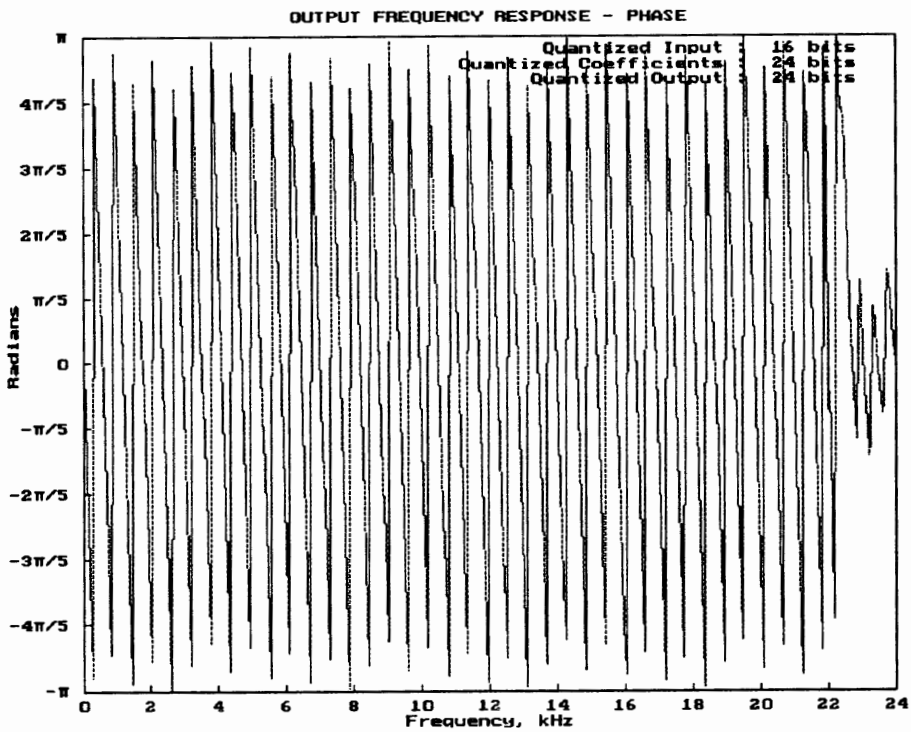


Figure 40. Phase response of the system with quantization.

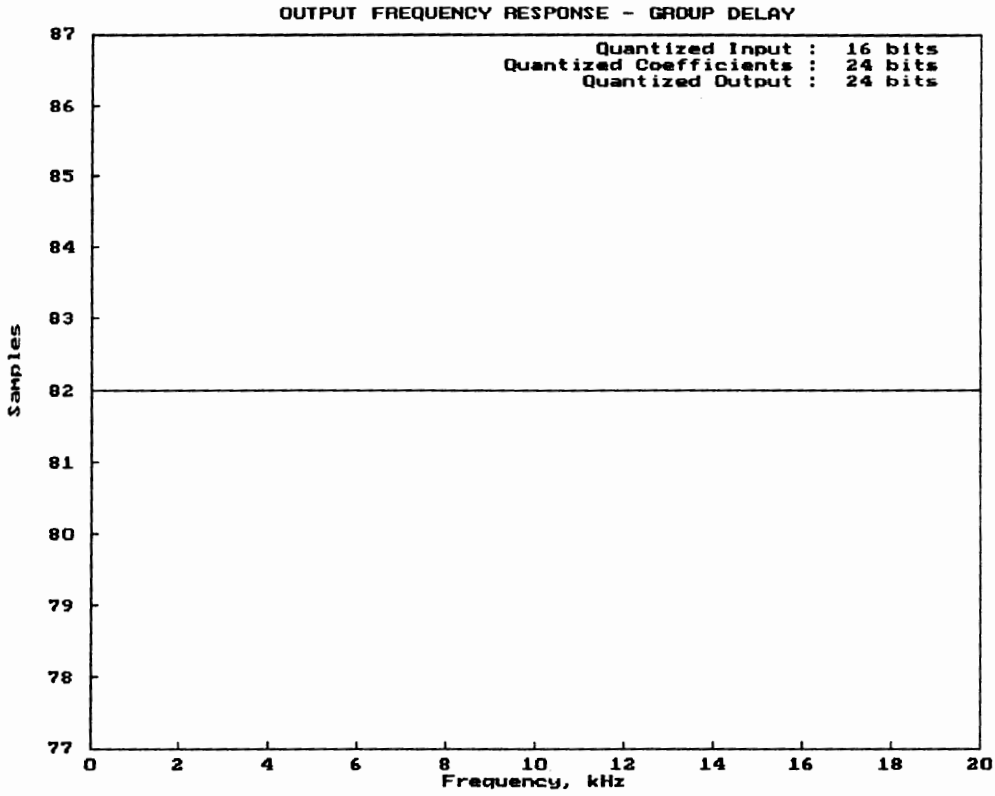


Figure 41. Group delay of the system with quantization.

SUMMARY

We covered several important topics in this paper. The first topic was the evolution of the design of FIR linear phase lowpass digital filters. We started by discussing the Parks-McClellan algorithm, which requires the filter parameters N, ω_p, ω_s, and $K = \frac{\delta_s}{\delta_p}$ as input. Next, we discussed a relationship discovered by Herrmann *et al.* that allows one to estimate the filter length if the other four filter parameters are known. This led Rabiner to propose an algorithm that uses Herrmann's filter length estimation equation as a starting point to determine the actual filter length (or any one of the other filter parameters) by interacting with the Parks-McClellan algorithm in an iterative manner. We proposed a modification to Herrmann's equation to give a better estimate in the case of an interpolation filter. Also, we restated the equation in terms of analog frequency to more closely link it to a design problem. We modified Rabiner's algorithm and wrote a PC program that will quickly design a filter given F, f_p, f_s, L, δ_p, and δ_s.

We reviewed some digital signal processing concepts regarding interpolation, decimation, and multistage design, and discovered the condition that determines when to add a lowpass filter to a multistage decimator in order to reduce the total number of filter taps required to implement the system. The group delay, however, may increase.

We presented the outline of a PC program for designing multistage sampling rate conversion systems. We demonstrated the program by designing a three-stage CD-to-DAT sampling rate conversion system. The program greatly

simplifies the design of multistage sampling rate conversion systems and can calculate the filter coefficients for each stage very quickly.

To demonstrate the effectiveness of the multistage design program, we wrote an implementation program for the CD-to-DAT sampling rate conversion. Included was the algorithm used for the conversion, along with the output from several examples.

REFERENCES

[1] R. Lagadec, D. Pelloni, and D. Weiss, "A 2-channel, 16-bit Sampling Frequency Converter for Professional Digital Audio," *Proc. of the 1982 IEEE Int. Conf. on Acoust. Speech Signal Process.*, pp. 93-96, Paris, May 1982.

[2] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[3] R.W. Schafer and L.R. Rabiner, "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, vol. 61, pp. 642-702, June 1973.

[4] R.E. Crochiere and L.R. Rabiner, "Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow-band Filtering," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-23, no. 5, pp. 449-456, Oct. 1975.

[5] J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," *IEEE Trans Audio Electroacoust.*, vol. AU-21, pp. 506-526, Dec. 1973.

[6] L.R. Rabiner, "Approximate Design Relationship for Lowpass FIR Digital Filters," *IEEE Trans Audio Electroacoust.*, vol. AU-21, pp. 456-460, Oct. 1973.

[7] J. H. McClellan and T.W. Parks, "A Unified Approach to the Design of Optimum FIR Linear Phase Digital Filters," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 697-701, Nov. 1973.

[8] O. Herrmann, L.R. Rabiner, and D.S.K. Chan, "Practical Design Rules for Optimum Finite Impulse Response Lowpass Digital Filters," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 769-799, Jul.-Aug. 1973.

[9] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[10] D.S.K. Chan and L.R. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," *IEEE Trans Audio Electroacoust.*, vol. AU-21, pp. 354-366, Aug. 1973.

[11] F.J. Harris, " On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51-83, Jan. 1978.

APPENDIX

On page 31, we addressed the need to modify the passband tolerance requirement of each stage in a multistage decimator system design. Otherwise, the passband ripple would increase with the number of stages. We will now address that issue.

Suppose that the lowpass filter of each stage i of an I-stage interpolator satisfies the condition

$$L_i - \delta_p \le \left|H_i(e^{j2\pi f/F_i})\right| \le L_i + \delta_p, \qquad f \in \text{passband}, \tag{107}$$

where

$$L = \prod_{i=1}^{I} L_i . \tag{108}$$

If the responses align precisely in frequency in the passband (peak-to-peak in an equiripple design), we would have

$$\prod_{i=1}^{I}(L_i - \delta_p) \le \left|H_0(e^{j2\pi f/F_0})\right| \le \prod_{i=1}^{I}(L_i + \delta_p), \quad f \in \text{passband}. \tag{109}$$

Since δ_p is a small number, the above bounds can be approximated by the following inequality

$$L - \sum_{i=1}^{I}\frac{L}{L_i}\delta_p \le \left|H_0(e^{j2\pi f/F_0})\right| \le L + \sum_{i=1}^{I}\frac{L}{L_i}\delta_p, \quad f \in \text{passband}. \tag{110}$$

Clearly, the passband ripple will increase with the number of stages. To avoid this increase, we define

$$\delta_p' = \frac{\delta_p}{\sum_{i=1}^{I}\frac{L}{L_i}} \tag{111}$$

and replace δ_p by δ'_p in Eq. (107). Thus, the lowpass filter of each stage satisfies

$$L_i - \delta'_p \le \left|H_i(e^{j2\pi f/F_i})\right| \le L_i + \delta'_p, \qquad f \in \text{passband}. \tag{112}$$

On page 34, we mentioned that adding a lowpass filter stage to a decimator system may increase the group delay of the system. In the three-stage design, the group delay is given by

$$\tau_g = \frac{1}{M}\left[\tau_{g1} + M_1\tau_{g2} + M_1M_2\tau_{g3}\right], \tag{113}$$

where

$$M = M_1M_2M_3. \tag{114}$$

In the four-stage system, the group delay is given by

$$\tau'_g = \tau_{g4} + \frac{1}{M}\left[\tau_{g1} + M_1\tau_{g2} + M_1M_2\tau_{g3}\right]. \tag{115}$$

Clearly, the group delay of the lowpass filter will not be reduced by the decimation factor and may increase the overall group delay. More work needs to be done to optimize the decimation design by weighting the total additions per second and storage and the group delay in deciding whether to add a lowpass filter or not.

On page 50, we stated that a $\frac{L}{M}$ sampling rate conversion system is optimized by addressing the system stopband frequency either in the first or the last stage of an I-stage system, where I is greater than one. We now derive the condition for determining which alternative should be used.

If we address the system stopband frequency requirement in the first stage of an I-stage system, we have

$$\hat{N}_1 = \frac{D_\infty\left(\frac{\delta_p}{I}, \delta_s\right)}{\left(f_s - f_p\right)/F_1}, \tag{116}$$

where F_0 is the initial sampling rate and F_I is the final sampling rate. Also, we have

$$\hat{N}_I = \frac{D_\infty\left(\frac{\delta_p}{I}, \delta_s\right)}{\left(F_{I-2} - f_s - f_p\right)/F_{I-1}}. \tag{117}$$

If we address the system stopband frequency requirement in the last stage of an $I+1$-stage system, we have

$$\hat{N}_1' = \frac{D_\infty\left(\frac{\delta_p}{I+1}, \delta_s\right)}{\left(F_0 - f_s - f_p\right)/F_1}, \tag{118}$$

$$\hat{N}_I' = \frac{D_\infty\left(\frac{\delta_p}{I+1}, \delta_s\right)}{\left(F_{I-2} - f_s - f_p\right)/F_{I-1}}, \tag{119}$$

and

$$\hat{N}_{I+1}' = \frac{D_\infty\left(\frac{\delta_p}{I+1}, \delta_s\right)}{\left(f_s - f_p\right)/F_I}. \tag{120}$$

Since

$$D_\infty\left(\frac{\delta_p}{I}, \delta_s\right) \approx D_\infty\left(\frac{\delta_p}{I+1}, \delta_s\right), \tag{121}$$

the sum of the lowpass filter lengths from stage 2 through I are approximately equal. *To optimize based upon storage requirements*, if the the filter length calculated from Eq. (116) is less than sum of the filter lengths calculated from Eqs. (118) and (120), we should meet the overall stopband frequency requirement in the first stage, rather than adding a lowpass filter as a final stage. Mathematically, we can write

$$\frac{D_\infty\left(\frac{\delta_p}{I},\delta_s\right)}{\left(f_s-f_p\right)/F_1} < \frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(F_0-f_s-f_p\right)/F_1} + \frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(f_s-f_p\right)/F_I}$$

$$\frac{F_1}{\left(f_s-f_p\right)} < \frac{F_1}{\left(F_0-f_s-f_p\right)} + \frac{F_I}{\left(f_s-f_p\right)}$$

$$\frac{F_1}{\left(f_s-f_p\right)} - \frac{F_I}{\left(f_s-f_p\right)} < \frac{F_1}{\left(F_0-f_s-f_p\right)}$$

$$\frac{F_1-F_I}{F_1} < \frac{f_s-f_p}{F_0-f_s-f_p}. \tag{122}$$

Since $F_1 = L_1 F_0$ and $F_I = \left(\frac{L}{M}\right)F_0$, we have

$$1-\frac{L}{L_1 M} < \frac{f_s-f_p}{F_0-f_s-f_p}. \tag{123}$$

To optimize based upon additions per second, assume that $N_i = Q_i L_i$, where Q_i is an integer, and the lowpass filter has an even number of taps. Thus, we have

$$F_0\left(\frac{D_\infty\left(\frac{\delta_p}{I},\delta_s\right)}{\left(f_s-f_p\right)/F_1} - L_1\right) < F_0\left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(F_0-f_s-f_p\right)/F_1} - L_1\right) + \frac{F_I}{2}\left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(f_s-f_p\right)/F_I}\right)$$

$$F_0\left(\frac{D_\infty\left(\frac{\delta_p}{I},\delta_s\right)}{\left(f_s-f_p\right)/F_1} - L_1\right) < F_0\left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(F_0-f_s-f_p\right)/F_1} - L_1\right) + \frac{LF_0}{2M}\left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(f_s-f_p\right)/F_I}\right)$$

$$\left(\frac{D_\infty\left(\frac{\delta_p}{I},\delta_s\right)}{\left(f_s-f_p\right)/F_1} - L_1\right) < \left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(F_0-f_s-f_p\right)/F_1} - L_1\right) + \frac{L}{2M}\left(\frac{D_\infty\left(\frac{\delta_p}{I+1},\delta_s\right)}{\left(f_s-f_p\right)/F_I}\right). \tag{124}$$

We can derive a similar inequality if L_i is not an integer multiple of N_i or the lowpass filter has an odd number of taps. We can also derive a weighted inequality by combining Eqs. (123) and (124).

Another factor that should be considered when deciding whether to attenuate the system stopband frequencies in the first stage or add a lowpass filter as a final stage to attenuate the frequencies is group delay. If the attenuation is done in the first stage, we have

$$\tau_g = \frac{1}{M}\left[\tau_{gI} + L_I \tau_{gI-1} + \ldots + \frac{L}{L_1 L_2}\tau_{g2} + \frac{L}{L_1}\tau_{g1}\right]. \tag{125}$$

If the attenuation is done by adding a lowpass filter as a final stage, we have

$$\tau'_g = \tau_{gI+1} + \frac{1}{M}\left[\tau'_{gI} + L_I \tau'_{gI-1} + \ldots + \frac{L}{L_1 L_2}\tau'_{g2} + \frac{L}{L_1}\tau'_{g1}\right]. \tag{126}$$

If $\tau_g < \tau'_g$, we may want to optimize by attenuating the system stopband frequencies in the first stage.

An overall optimization factor would be a linear combination of the addition rate, storage, and group delay.