Title	An efficient design or fractional-delay digital FIR filters using the Farrow structure
Author(s)	Pun, KSC; Wu, YC; Chan, SC; Ho, KL
Citation	The 11th IEEE Signal Processing Workshop on Statistical Signal Processing Proceedings, Singapore, 6-8 August 2001, p. 595-598
Issued Date	2001
URL	http://hdl.handle.net/10722/46311
Rights	©2001 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

AN EFFICIENT DESIGN OF FRACTIONAL-DELAY DIGITAL FIR FILTERS USING THE FARROW STRUCTURE

Carson K.S. Pun, Y.C. Wu, S.C. Chan, and K.L. Ho

Department of Electrical and Electronic Engineering,
The University of Hong Kong
Email: kspun@eee.hku.hk, ycwu@eee.hku.hk, scchan@eee.hku.hk, klho@eee.hku.hk

ABSTRACT

Fractional-delay digital filter (FD-DF), implemented using the Farrow structure, is very attractive in providing online tuning delay of digital signals. This paper proposes a new method for the design of such Farrow-based FD-DF using sum-of-powers-of-two (SOPOT) coefficients. Using the SOPOT coefficient representation, coefficient multiplication can be implemented with limited number of shifts and additions. Design examples show that the proposed method can greatly reduce the design time and complexity of the Farrow structure while providing comparable phase and amplitude responses.

I. INTRODUCTION

Fractional-delay digital filters (FD-DF) are very useful in delaying signals, which is required in many applications such as software radio, digital modems, arbitrary sampling rate conversion, time-delay estimation, etc. The Farrow structure [1] is particularly attractive because it can provide variable signal delay, making high-speed online tuning feasible. The basic principle of the Farrow structure is to approximate the impulse response of an ideal fractional-delay digital filter with delay μ by polynomial interpolation from the impulse responses of a limited set of fractional-delay digital filters with delays equally spaced within a range usually chosen to be $\mu = [-0.5, 0.5]$. To implement the Farrow structure, the signal will pass through these sub-filters and multiply with the appropriate powers of d to produce the output, Figure 1. The number of sub-filters required is equal to the order of the polynomial approximation used plus one. For precise control of the signal delay, the length of these sub-filters and the order of polynomial approximation will be considerable, requiring a large number of multipliers to implement this structure. As a result, higher power dissipation and larger area for VLSI implementation is expected.

In this paper, a novel algorithm for designing the Farrow structure with sum-of-powers-of-two (SOPOT) coefficients is proposed. SOPOT representation of filter coefficients is an attractive method for VLSI or hardware implementation of digital filters because multiplication of SOPOT coefficients can be implemented efficiently using hard-wired shifters and adders only (i.e. multiplier-less). More precisely, all the coefficients of the sub-filters in the Farrow structure are represented in SOPOT form and are implemented as additions and hardwired shifts. To further reduce the number of adders required in this structure, a transposed form of the sub-filters in the Farrow structure is employed which allows us to implement all the SOPOT multiplications with a single multiplier-block (MB) [3]. The application of MB to the efficient implementation of interpolated filters and filter banks were reported in [3]. Unfortunately, the design of such multiplier-less Farrow structure was missing. Using MB, the redundancy in the multiplication of the input with all the constant multipliers can be fully explored through the reuse of the immediate results generated. In principle, it is possible to remove all the redundancy found in the constant multipliers leading to a significant reduction in the number of adders required to implement the Farrow structure. The proposed design algorithm consists of two different steps: A FD-DF filter with real-valued coefficients is first designed. A flexible and efficient "random search" algorithm is then employed to search for the SOPOT coefficients while minimizing some criteria such as the number of

SOPOT terms used subjected to a given frequency specification. This random search algorithm is similar to the mutation of genetic algorithm (GA) and the random walk in stimulated annealing. The main difference here is that we have limited its search space to a small neighborhood of the real-valued solution. This greatly shortens the search time to a few minutes. Our experience also indicates that excellent SOPOT solutions can be obtained in a reasonably time even when the filters involved are IIR. This is very difficult to achieve by GA even with design time several orders of magnitude longer [9]. The latter is mainly due to high sensitivities of the poles. Another advantage of this algorithm is that it can also be used to minimize directly the hardware cost such as adder cells of the filters, taking into account round-off and overflow noise [9]. There are many methods for designing FD-DF with real-valued coefficients [1][4][8]. In this work, the prototype fractional delay filters for the Farrow structure are designed using complex Chebyshev approximation, which is readily available in MATLAB. They are then interpolated to obtain the sub-filter coefficients for generating the MB. Design examples show that more than half of the adders in the SOPOT coefficients can be reduced with slight or negligible degradation in frequency responses, representing significant saving in hardware resources and power dissipation.

This paper is organized as follows: the efficient Farrow structure with SOPOT coefficients and multiplier block is introduced in Section II. Its design algorithm will be described in Section III followed by several examples in Section IV. Finally, conclusions are drawn in Section V.

II. THE EFFICIENT FARROW STRUCTURE

As mentioned earlier, one problem with the implementation of variable fractional-delay digital filters is the dependence of the impulse responses of the FD-DF on the delay parameter μ. More precisely, the output of the FD-DF, $y[(m + \mu)T]$, is given by

$$y[(m + D + \mu)T] = \sum_{i=0}^{N-1} x[(m - i)T] \cdot h_\mu(i), \qquad (1)$$

where $x[mT]$ is the input signal sampled at a period T, $h_\mu(i)$ is the FD-DF with delay $D + \mu$ and D is an integer constant, and N is the length of $h_\mu(i)$. To avoid the implementation of a large number of filters with different delays, Farrow [1] proposed to approximate each impulse response $h_\mu(i)$ with the following P^{th} order polynomial in delay value μ such that the delay control is independent of the filter coefficients.

$$h_\mu(i) = \sum_{n=0}^{P} b_n(i) \mu^n. \qquad (2)$$

Substituting (2) into (1) gives

$$y[(m + D + \mu)T] = \sum_{n=0}^{P} \left[\sum_{i=0}^{N-1} x[(m - i)T] \cdot b_n(i) \right] \mu^n. \qquad (3)$$

Figure 1 shows the Farrow structure for implementing equation (3), where the input signal is passed through a number of sub-filters $b_n(i)$, $n = 0, ..., P$, and is multiplied by the appropriate powers of μ to produce the output. Though the Farrow structure is very useful in providing a continuous value of signal delay, it still requires large number of multiplications for implementation,

especially when P and N are large to provide very precise control of the frequency characteristics of the FD-DF. One method to avoid the expensive multipliers is to convert the filter coefficients in the following SOPOT representation

$$\hat{b}_n(i) = \sum_{k=1}^{L} b_{n,k}(i) \cdot 2^{a_k} ,$$ (4)

with $b_{n,k}(i) \in \{-1,1\}$ and $a_k \in \{-l,...,-1,0,1,...,l\}$, where l is a positive integer and its value determines the range of the coefficients. L is the number of terms used in the coefficient approximation and is usually limited to a small number. The coefficient multiplications can therefore be implemented as limited shifts and additions, resulting in a significant reduction in implementation complexity. Very often, there is also significant redundancy in these SOPOT coefficients, which appears as common sub-expressions among different SOPOT coefficients. Due to the z-operator, it is somewhat difficult to remove these sub-expressions without increasing much shift registers. Fortunately, thanks to the transposed form of the sub-filters, the Farrow structure can be rewritten as in Figure 2. In this new structure, the input is multiplied to a number of constant coefficients. Hence, the common sub-expressions within the SOPOT coefficients can be eliminated [5][6] using a single multiplier-block, which further reduces the complexity of the Farrow structure.

A number of methods were proposed for designing the Farrow structure-based FD-DF [1]. Given these real-valued coefficients of the Farrow structure, it remains to determine the SOPOT coefficients $\hat{b}_n(i)$ that satisfy a given specification with the minimum number of adders in the multiplier-block. Commonly used distortion measures are the least squares and the minimax criterion. Without loss of generality, we shall employ the minimax criteria in this paper. Let $H(e^{j\omega}, \mu)$ and $\hat{H}(e^{j\omega}, \mu)$ be the frequency responses of the real-valued Farrow structure and its SOPOT counterpart, then the design problem can be stated as follows:

Given a set of initial Farrow coefficients $b_n(i)$, the maximum number of terms L in each coefficient and the dynamic range l of the coefficients, determine the SOPOT coefficients $\hat{b}_n(i)$ such that the maximum value of phase response error δ_p is minimized subject to a given peak amplitude error $\delta_a < \varepsilon$, where

$$\delta_p = \max_{0<\omega<\pi, |\mu|<0.5} \left[\left| \frac{\arg\{H(e^{j\omega}, \mu)\} - \arg\{\hat{H}(e^{j\omega}, \mu)\}}{\omega} \right| \right] ,$$

subject to $\delta_a = \max_{0<\omega<\pi, |\mu|<0.5} \left| H(e^{j\omega}, \mu) - \hat{H}(e^{j\omega}, \mu) \right| < \varepsilon$. (5)

III. DESIGN PROCEDURE

The design of Farrow structure is first designing the prototype filters with specific fractional delay, then through interpolating these prototype filters to acquire the subfilters of the Farrow structure. The Farrow structure prototype filters are designed using Complex Chebyshev Approximation which is readily available in Matlab as cremez. For example, if the interpolation order is 3, then there are 4 subfilters. That is we are required to design a batch of prototype filters all with equal length (say 10 or more) with frequency response of $H_i(e^{j\omega}) = e^{-j\Delta_i \omega}, i = 1,...10$ and $\Delta_i = -0.5 + (i-1)/9$. For the same impulse coefficients, these will then be interpolate using a third order polynomial using least-square. Repeat each prototype filters coefficients for this interpolation procedure and these final polynomial coefficients are the initial full-precision Farrow structure filter coefficients.

The optimization procedure consists of two stages. First, the SOPOT coefficients of the initial Farrow structure such that the performance measure in (5) is minimized using a random search technique. Then, the minimum number of adders needed in the multiplier block is determined. The generation of the multiplier-block from the SOPOT coefficients follows the algorithms proposed in [3]. Let b_I be the vector containing the initial values $b_n(i)$'s of the Farrow structure. The principle of the random search algorithm is to generate random candidate SOPOT coefficients in the neighborhood of b_I so as to search for the optimal discrete solution. More precisely, a new coefficient vector b_{NEW} is generated by adding to it a random vector to the original coefficient vector b_I as follows

$$b_{NEW} = \lfloor b_I + \alpha \cdot b_R \rfloor_{SOPOT} .$$ (6)

where α is a scale factor which control the size of neighborhood to be searched, b_R is a vector with its elements being random numbers in the range $[-1,1]$, and $\lfloor \cdot \rfloor_{SOPOT}$ is the rounding operation which convert its argument to the nearest SOPOT coefficients with maximum number of terms in each coefficient being L and dynamic range l. The performance measures δ_p and δ_a of the new coefficients are then calculated. The set that yields the minimum phase error with a given peak amplitude error ε is the optimum solution under the constraints of L and l. As this is a random search algorithm, the longer the searching time, the higher the chance of founding the optimal solution.

There are several advantages of this algorithm. First of all, with the computational power of nowadays personal computer (PC) the time for obtaining high quality solutions is manageable. In fact, for the problem considered here, the overall design time only takes less than 5 minutes to complete on a typical Pentium-400 PC using Matlab 5.3., including both the design of SOPOT coefficients and the multiplier-block design. Secondly, it is applicable to problem with general objective function probably with very complicated inequality constraints. Moreover, a set of possible solutions representing different tradeoffs between computational complexity and performance will be generated during the search. Therefore, it helps one to achieve an appropriate tradeoff for a given application. It is also possible to combine the two stages together to improve the performance but the computational time will be greatly increased.

IV. DESIGN EXAMPLES

Example 1

As a simple example, the famous cubic Lagrange interpolator [7], with coefficients shown in Table 1, is implemented using the proposed algorithm. The passband bandwidth under optimization is from 0 to 0.4π. The original peak ripple error and phase delay error are, respectively, 0.048769 and 0.008179. By multiplying all the coefficients by 6, they can be converted to simple integers. Using the concept of multiplier-block, the additions in implementing the multipliers 3 and 6 can be shared to reduce the total number of adders. The final Farrow structure implemented using the multiplier-block is shown in Figure 3. (*The ">>n" sign in Figure 3, means a hard-wired shift towards the LSB for n-bit. As for the "<<n" sign, it means a hard-wired shift towards the MSB for n-bit.*) It requires only 3 adders including the scaling (1/6) in SOPOT coefficients at the output.

Example 2

As another example, let's consider the coefficients provided by Farrow in [1]. The SOPOT coefficients obtained by the random search algorithm are shown in Table 2. The bandwidth under consideration for this filter is from 0 to 0.6π. The original peak ripple error and phase delay error are 0.006271 and 0.0032,

respectively; whereas for the SOPOT Farrow structure, the peak ripple error and phase delay error are 0.005371 and 0.0046, respectively. After common sub-expressions elimination, the multiplier-block requires only 13 adders compared favorably with 32 real multiplications in the original Farrow structure. These results show that the number of adders can be drastically reduce by using multiplier-block. The resultant Farrow structure filter has a much lower complexity than the real-valued Farrow structure but providing nearly the same phase delay and amplitude response.

Example 3

Our last example will be on a Farrow structure with higher polynomial order. The prototype filters are designed using complex Chebyshev approximation and interpolated by a 5th order polynomial. The bandwidth under consideration is from 0 to 0.75π. The SOPOT coefficients are shown in Table 3. After common-expression elimination, the sub-filters require only 18 adders to achieve a peak ripple error of 0.026376 and maximum phase delay error of 0.0059. The frequency responses of the proposed structure and its real-values counterpart are shown in Figure 4. It can be seen that they are very close to each other. The performance and arithmetic complexity of the various implementations are summarized in Table 4.

	$b_3(\cdot)$	$b_2(\cdot)$	$b_1(\cdot)$	$b_0(\cdot)$
0	1/6	0	-1/6	0
1	-½	½	1	0
2	½	-1	-½	1
3	-1/6	½	-1/3	0

Table 1. Coefficients of the Lagrange-based FD-DF.

	$b_3(\cdot)$	$b_2(\cdot)$	$b_1(\cdot)$	$b_0(\cdot)$
0	$-2^{-6}+2^{-8}$	$2^{-4}-2^{-7}$	0	$-2^{-6}+2^{-10}$
1	$2^{-4}+2^{-6}-2^{-8}$	$-2^{-2}+2^{-5}+2^{-8}$	-2^{-6}	$2^{-4}-2^{-7}$
2	$-2^{-1}+2^{-3}-2^{-5}$	$2^{-1}+2^{-3}+2^{-7}$	$2^{-3}-2^{-5}+2^{-8}$	$-2^{-3}-2^{-5}-2^{-9}$
3	$2^{-0}-2^{-3}+2^{-5}$	$-2^{-1}+2^{-5}+2^{-8}$	$-2^{-0}-2^{-2}+2^{-5}$	$2^{-1}+2^{-3}-2^{-7}$
4	$-2^{-0}+2^{-3}-2^{-5}$	$-2^{-1}+2^{-5}+2^{-8}$	$2^{-0}+2^{-2}-2^{-5}$	$2^{-1}+2^{-3}-2^{-7}$
5	$2^{-1}-2^{-3}+2^{-5}$	$2^{-1}+2^{-3}+2^{-7}$	$-2^{-3}+2^{-5}-2^{-8}$	$-2^{-3}-2^{-5}-2^{-9}$
6	$-2^{-4}-2^{-6}+2^{-8}$	$-2^{-2}+2^{-5}+2^{-8}$	2^{-6}	$2^{-4}-2^{-7}$
7	$2^{-6}-2^{-8}$	$2^{-4}-2^{-7}$	0	$-2^{-6}+2^{-10}$

Table 2. SOPOT coefficients for the proposed Farrow structure.

	$b_5(\cdot)$	$b_4(\cdot)$	$b_3(\cdot)$	$b_2(\cdot)$	$b_1(\cdot)$	$b_0(\cdot)$
1	$2^{-5}-2^{-7}$	$-2^{-3}+2^{-6}-2^{-10}$	$-2^{-5}-2^{-7}$	$2^{-3}+2^{-5}+2^{-7}$	2^{-7}	$-2^{-5}-2^{-9}$
2	$-2^{-4}-2^{-6}$	$2^{-2}-2^{-5}-2^{-7}$	$2^{-3}+2^{-6}+2^{-10}$	$-2^{-1}+2^{-3}+2^{-6}$	$-2^{-5}+2^{-9}$	$2^{-4}+2^{-6}-2^{-10}$
3	$2^{-2}-2^{-4}-2^{-8}$	$-2^{-2}-2^{-5}-2^{-7}$	$-2^{-1}-2^{-6}-2^{-8}$	$2^{-0}-2^{-2}+2^{-5}$	$2^{-3}-2^{-7}$	$-2^{-2}+2^{-4}+2^{-7}$
4	$-2^{-2}-2^{-5}-2^{-7}$	$2^{-3}+2^{-5}-2^{-8}$	$2^{-0}+2^{-4}+2^{-8}$	$-2^{-1}-2^{-5}-2^{-7}$	$-2^{-0}-2^{-2}$	$2^{-1}+2^{-3}-2^{-9}$
5	$2^{-2}+2^{-5}+2^{-7}$	$2^{-3}+2^{-5}-2^{-8}$	$-2^{-0}-2^{-4}-2^{-8}$	$-2^{-1}-2^{-5}-2^{-7}$	$2^{-0}+2^{-2}$	$2^{-1}+2^{-3}-2^{-9}$
6	$-2^{-2}+2^{-4}+2^{-8}$	$-2^{-2}-2^{-5}-2^{-7}$	$2^{-1}+2^{-6}+2^{-8}$	$2^{-0}-2^{-2}+2^{-5}$	$-2^{-3}+2^{-7}$	$-2^{-2}+2^{-4}+2^{-7}$
7	$2^{-4}+2^{-6}$	$2^{-2}-2^{-5}-2^{-7}$	$-2^{-3}-2^{-6}-2^{-10}$	$-2^{-1}+2^{-3}+2^{-6}$	$2^{-5}-2^{-9}$	$2^{-4}+2^{-6}-2^{-10}$
8	$-2^{-5}+2^{-7}$	$-2^{-3}+2^{-6}-2^{-10}$	$2^{-5}+2^{-7}$	$2^{-3}+2^{-5}+2^{-7}$	-2^{-7}	$-2^{-5}-2^{-9}$

Table 3. SOPOT coefficients for the proposed Farrow structure.

	Ex 1.	Ex 2.	Ex 3.
Real-valued (Multipliers)	4	32	48
SOPOT (Adders)	N/A	48	78
Multiplier-block (Adders)	3	13	18
% of adders reduction	25%	72.92%	76.92%
Design Time used on Pentium-400 (Minutes)	N/A	4	6

Table 4. Comparison between various implementation schemes.

V. CONCLUSION

A new method for the design Farrow-based FD-DF using sum-of-powers-of-two (SOPOT) coefficients is proposed. This method has the advantage of fast design time with good frequency response of the Farrow structure and able to reduce the no. of terms of SOPOT coefficient in order to reduce hardware complexity. Design examples show the robustness of this method for designing Farrow structure filters with different specifications.

REFERENCES

[1] C. W. Farrow, "A continuously variable digital delay element", in Proc. of ISCAS 1988, pp.2641-2645.
[2] A. G. Dempster and N. P. Murphy, "Efficient interpolators and filter banks using multiplier blocks," IEEE Trans. Signal Processing, vol. 48. pp. 257-261, Jan. 2000
[3] A.G. Dempster and M.D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. CAS. II, vol. 42, no.9, pp. 569-577, Sep. 1995.
[4] T. I. Laakso, V. Valimaki, M. Karjalainen and U. K. Laine, "Splitting the unit delay, tools for fractional delay filter design," IEEE Signal Processing Magazine, pp.30-60, Jan. 1996.
[5] M. Potkonjak, M. B. Srivastava and A. P. Chandrakasan, "Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common subexpression elimination," IEEE Trans. Computer-aided Design, vol. 15, pp.151-165, Feb. 1996.
[6] R. Pasko, P. Schaumont, V. Derudder and D. Durackova, "Optimization method for broadband modem FIR filter design using common subexpression elimination," in Proc. of Symposium on System Synthesis 1997, pp.100-106.
[7] L. Erup, F. Gardner, and R. A. Harris, "Interpolation in digital modems — part II: implementation and performance," IEEE Trans. Commun., vol. 41, pp. 908-1008, Jun. 1993.
[8] J. Vesma and T. Saramaki, "Design and properties of polynomial-based fractional delay filters," in Proc. ISCAS 2000, pp.104-107.
[9] Carson K.S. Pun, S C. Chan and K. L. Ho., "Efficient design of a class of multiplier-less perfect reconstruction two-channel filter banks and wavelets with prescribed output accuracy," to appear in 11th IEEE Workshop on Statistical Signal Processing 2001.
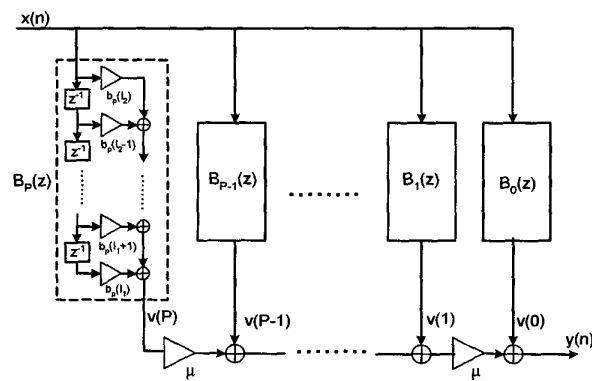
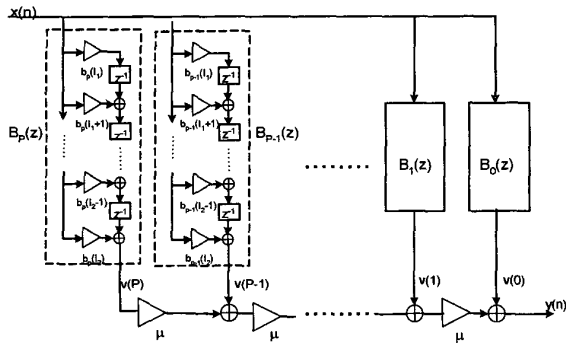Figure 1. Original Farrow structure.
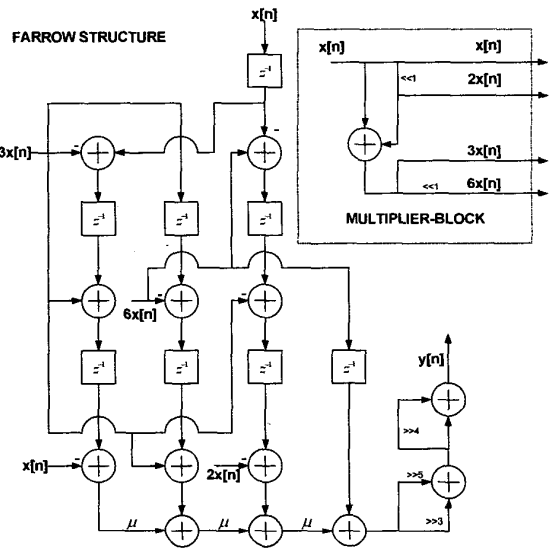
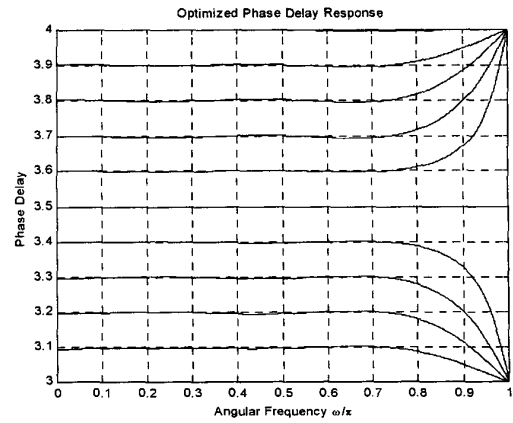Figure 2. Proposed implementation of the Farrow structure.
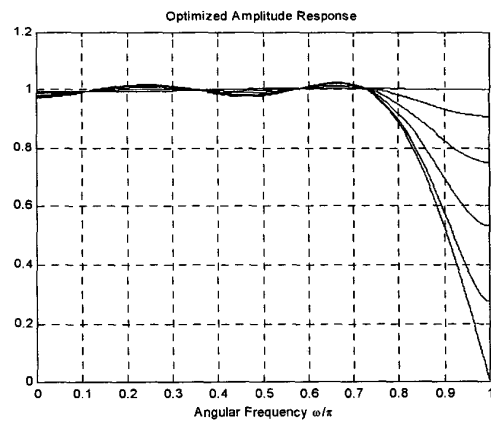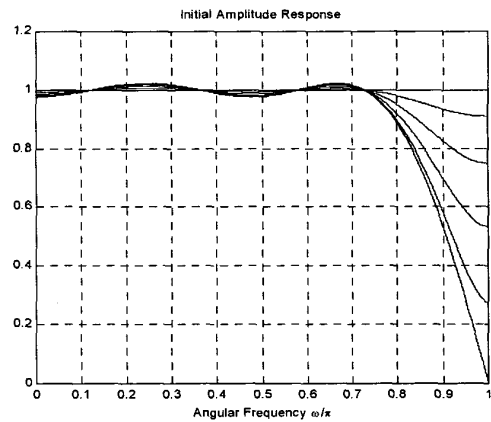

Figure 3. Farrow structure of Lagrange interpolator in Example 1.

Figure 4. Frequency and phase delay responses in Example 3.