Efficient Algorithms and Design for Interpolation Filters in Digital Receiver

1 Xiaowei Niu, 2 Zhiming He, 3 Liwan Chen, 4 Qiang Chen, 5 Hui Xie

1, 2 School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu City, Sichuan 611731, China
1, 3, 4, 5 School of Electronic and Information Engineering, Chongqing Three Gorges University, 404000, Wan Zhou, China
Tel.: 15923892350
E-mail: nxw4525@126.com

Received: 11 March 2014 / Accepted: 30 April 2014 / Published: 31 May 2014

Abstract: Based on polynomial functions this paper introduces a generalized design method for interpolation filters. The polynomial-based interpolation filters can be implemented efficiently by using a modified Farrow structure with an arbitrary frequency response, the filters allow many pass-bands and stop-bands, and for each band the desired amplitude and weight can be set arbitrarily. The optimization coefficients of the interpolation filters in time domain are got by minimizing the weighted mean squared error function, then converting to solve the quadratic programming problem. The optimization coefficients in frequency domain are got by minimizing the maxima (MiniMax) of the weighted mean squared error function. The degree of polynomials and the length of interpolation filter can be selected arbitrarily. Numerical examples verified the proposed design method not only can reduce the hardware cost effectively but also guarantee an excellent performance.

Keywords: Polynomial-based filter, Modified Farrow structure, Interpolation, Basis function, Variable fractional delay.

1. Introduction

With increasing applications of digital receivers [10] in communication systems, the variable fractional delay (VFD) filters have received considerable attentions, and these filters have many attractive features since the Farrow structure [8, 18] was introduced, such as the accurate control of variable frequency characters, the simple real-time update of coefficient values, and regular implementation patterns etc. The polynomial-based technique is an important method to improve the signal to noise rate (SNR) [15, 21], which has been used to design VFD filters [12-14, 19] by implementing the filters in Farrow structure or its modifications. These kinds of filters allow to evaluate new sample values at arbitrary positions between existing samples of a discrete-time signal, and also have some other attractive features. Firstly, these kinds of filters have a piecewise polynomial impulse response. Secondly, they can be implemented efficiently by using the Farrow structure or its modifications. Thirdly, by weighting properly the output samples of these filters, it is easily to control the desired time instant for the interpolated output samples, which parallel FIR filters. Other VFD
techniques include the spline-based techniques [22] and the Lagrange-based techniques [1, 17].

The design methods of polynomial-based interpolation filters can be mainly divided into two different classes: the time domain method and the frequency domain method. The Lagrange-based interpolators [5, 6, 23, 24] and spline-based interpolators [2, 20] are the best known time domain methods. The advantages of these design methods are that the filters coefficients are easily available in the closed form.

However, these interpolation filters become poor when the frequency components close to half the sampling rate, because they neglect the frequency-domain information of the input signals. The second design method is to optimize the coefficients of the reconstructed impulse response in frequency domain [3, 4, 11, 16, 25], and the best known frequency domain method is the polynomial-based interpolation filters [8-10]. This design method enables one to obtain a better filtering characteristic than those obtained by the methods mentioned above. Such as, in [8] Farrow proposed a least-mean-square optimization of the polynomial-based fractional delay filters, but these methods not allow to optimize separately the coefficients of the interpolation filters in pass-band or stop-band. In [9] Harris et al. used the reordering of the polynomial coefficients to obtain polynomial expansions of the time series, and in [10] Hamila et al. used the hybrid analog/digital model to design the interpolation filters, and this method enables one to select arbitrarily the length of the impulse response, but it has a high computational complexity.

Motivated by the cited works above, in this paper we derive a new design method for the interpolation filters, this design method allows to piecewise optimize the pass bands and stop bands of these interpolation filters, the desired amplitude and weighted functions can be selected arbitrarily for each band, the length of the interpolation filter and the degree of the polynomials can be chosen independently. According to the design requirements, we introduce the interpolation functions \( \phi_a(t) = (t-1)u_t \). The optimization coefficients of the proposed filter can be performed either in the minimax method or in the least-mean-square method. For this proposed design method, we find that the first items of the optimization coefficients equal to zero, that is, \( c_0(t) = 0, i = 0,1,2,... \) which improve the hardware implementation ability effectively.

The rest of this paper is organized as follows. Polynomial-based interpolation filter and its impulse response are introduced in section 2. Section 3 presents the polynomial basis function design, and filter optimization in time domain and frequency domain, respectively. Section 4 introduces the Farrow structure of interpolation filters. In section 5, two simulation examples are provided to demonstrate effectiveness of the interpolation filter. Finally, conclusions are given in section 6.

2. Polynomial-Based Interpolation Filter and its Impulse Response

In this paper, only the polynomial-based interpolation filters are proposed due to they can be implemented efficiently using the Farrow structure or its modifications. The design method for these kinds of filters is based on the interpolation functions to optimize the filter coefficients. Therefore, the selection of the basis functions and the optimal method for the filter coefficients are two key problems to design the polynomial-based interpolation filters.

If function \( f(x) \) in a neighborhood \( U(x_0) \) has \((n + 1)\)-order derivative, then for \( U(x_0) \), \( f(x) \) has \( n \)-order Taylor formula

\[
f(x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)(x-x_0)^i}{i!} = \sum_{i=0}^{n} a_i(x-x_0)^i, \quad (1)
\]

In mathematics, a set consists of certain giving functions from set \( X \) to \( Y \), then the set can be called a function space, extending the Taylor formula to a function space, one has

\[
f(x) = \sum_{i=0}^{n} a_i g_i(x), \quad g_i(x) = (x-x_0)^i, \quad (2)
\]

For the function space \( V \), if there are a function sequences \( \{v_n(x) \in V, n=0,1,2...\} \) to any function \( f(x) \in V \), \( f(x) = \sum_{n} c_n v_n(x) \) holds, then we can call the function sequences the basis sequences of function space \( V \).

Assume the function space \( V_m \) consists of all the effective interpolation functions. Construct the basis sequences \( \{v_n(x) \in V_m, n=0,1,2,\ldots\} \) of the function space \( V_m \), that is, \( \{v_n(x) \in V_m, n=0,1,2,\ldots\} \). The basis sequences can also be expressed as \( v_n(x) = v(x)^n, n=0,1,2\ldots \). Obviously, \( v(x) \in V_m \), considering the essence of interpolation method is that the original signals are double-sampled processing after low-pass filtering. Thus, any interpolation functions must have the similar characteristics of low-pass filtering functions. In the following part, based on the existing original samples we will reconstruct the approximating signal. To analyze the characteristics of interpolation functions, we use the basis functions with the unit interval \( \Delta_s \) to approximate original signal sectionally. For convenience to discuss, assume \( \Delta_s=1 \), then the power series of the basis functions can be written as follows

\[
\phi_a(t) = \begin{cases} \phi(t)^n, & 0 \leq t < 1, \\ 0, & n=0,1,2\ldots \end{cases}, \quad (3)
\]
If no other special instructions, assume the basis function \( \phi(t) \) is bounded in interval \([0,1]\). In Eq. (3), it is shown that the overall basis functions consist of the power series \( \phi(t)^n, n = 0,1,2, \ldots \). For convenience, in the following part we simply call the interpolation filter is based on the basis function \( \phi(t) \).

Assume the length of the interpolation filter as \( L \), then the impulse response of the interpolation filter can be reconstructed as follows

\[
h(t) = \sum_{i=-L/2}^{L/2-1} \sum_{n=0}^{N} c(n)\phi_n(t-i), -\frac{L}{2} \leq t < \frac{L}{2},
\]

where \( h(t) \) is the impulse response of the reconstructed filter, \( c_n(i) \) denote the filter coefficients, \( \phi_n(t-i) \) denote the basis functions of the interpolation filter, and \( N \) is the degree of the polynomials. Therefore, we can use the digital filter theory to design the reconstructed impulse response \( h(t) \), then piecewise approximate it, finally obtain the optimization coefficients as well as realize the design of polynomial-based interpolation filter. The impulse response \( h(t) \) can be expressed in a piecewise interval \([i, i+1]\), it is desirable

\[
h_i(t) = \sum_{n=0}^{N} c_n(i)\phi_n(t), 0 \leq t < 1,
\]

From Eq. (5), we know that the constructed impulse response in each unit interval can be piecewise optimized.

In order to make the digital interpolation filter has a linear phase response, the impulse response \( h(t) \) should be a symmetrical function, which means the functions \( h_i(t) \) and \( h_{i+1}(t) \) are symmetrical around \( Y \) axis as follows

\[
h_{i+1}(t) = \sum_{n=0}^{N} c_n(-i-1)\phi_n(t) = \sum_{n=0}^{N} c_n(i)\phi_n^*(t)
\]

where the functions \( \phi_n(t) \) and \( \phi_n^*(t) \) are symmetrical around \( t = 1/2 \), when exploiting the above symmetries, the number of optimization coefficients to be implemented can be reduced from \((N+1)L\) to \((N+1)L/2\), thus, the impulse response \( h(t) \) can also be reconstructed as follows

\[
h(t) = \sum_{i=-L/2}^{L/2-1} \sum_{n=0}^{N} c_n(i)\phi_n(t-i), -\frac{N}{2} \leq t < \frac{N}{2},
\]

where \( \phi_n(t-i) \) is got by \( \phi_n(t-i) \), and they are symmetrical around \( t = 0 \). The impulse response of the filter in frequency domain corresponding can be written as

\[
H(f) = \sum_{i=0}^{L/2-1} \sum_{n=0}^{N} c_n(i)\Phi_n(f),
\]

where \( H(f) \) and \( \Phi(f) \) are the Fourier transform of functions \( h(t) \) and \( \phi(t-i) \) respectively.

3. Design of Interpolation Functions and Filter Optimization

The fundamental idea for filter optimization is to approximate the existing original signal according to some time-domain or frequency-domain criteria, and the interpolation filter should have an efficient digital implementation structure. The design of interpolation functions can be carried out in the time domain and frequency domain, respectively. Thus, the optimization of filter coefficients \( c_n(i) \) can also be divided into two different classes, that is, the coefficients optimization in time domain and frequency domain, respectively.

3.1. Design of Interpolation Functions and Filter Optimization in Time Domain

In order to obtain the optimization coefficients of the reconstructed filter in time domain, we use the polynomial basis functions to piecewise approximate the desired low-pass filtering function. In the following we will derive the method to get the optimization coefficients of the reconstructed impulse response \( h(t) \), then to solve the most weighting coefficients.

Define the desired low-pass filtering function as \( g(t) \), and define function \( \varepsilon(h,g) \) as the error function between \( h(t) \) and \( g(t) \), that is, \( \varepsilon(h,g) \) is used to measure the approximating where degree between \( h(t) \) and \( g(t) \). The optimization of the filter coefficients can be got via minimizing the error function \( \varepsilon(h,g) \).

In order to reduce the error at interpolation points, the impulse response of the reconstructed filters should meet the following conditions

\[
h(t) = \begin{cases} 
1, & t = 0 \\
0, & t = \pm 1, \pm 2, \ldots, \pm L 
\end{cases}
\]

Equation (9) can be written in an alternative form as follows

\[
\sum_{n=0}^{N} c_n(0)\phi_n(0) = 1 \\
\sum_{i=0}^{L/2-1} \sum_{n=0}^{N} c_n(i)\phi_n(0) = 0, i = 1, 2, \ldots, L/2 - 1 \\
\sum_{i=0}^{L/2-1} \sum_{n=0}^{N} c_n(i)\phi_n(1) = \sum_{i=0}^{L/2-1} \sum_{n=0}^{N} c_n(i)\phi_n(L/2) = 0, i = 0, 1, 2, \ldots, L/2 - 1
\]
From Eq.(10), we can see $\varphi(0) \neq 0$, and $\phi(1)$, the endpoint values of the interpolation function, determine the feasible regions of $c_n(i)$.

Let

\[ C = [c_0(0), c_1(0), \ldots, c_n(0), c_n(1), \ldots, c_n(L/2 - 1), c_n(L/2 - 1), \ldots, c_n(L/2 - 1)]^T, \]

where $C_{i}^T = [c_0(i), c_1(i), \ldots, c_n(i)]^T$.

Then

\[ A_{eq} \cdot C = B_{eq}, \]

where

\[ A_{eq} = \begin{bmatrix} A_{eq_0}, 0, 0, \ldots, 0 & 0, 0, 0, \ldots, 0 & \ldots & 0, 0, \ldots, 0 \\
0, 0, \ldots, 0 & A_{eq_0}, 0, 0, \ldots, 0 & \ldots & 0, 0, \ldots, 0 \\
0, \ldots, 0, 0, 0, \ldots, 0 & 0, \ldots, 0, 0, \ldots, 0 & \ldots & 0, \ldots, 0, 0, \ldots, 0 \\
\end{bmatrix}, \]

\[ A_{eq_0} = [\varphi_0(0), \varphi_1(0), \ldots, \varphi_n(0)] \]

\[ A_{eq_1} = [\varphi_0(1), \varphi_1(1), \ldots, \varphi_n(1)] \]

\[ B_{eq} = [1, 0, 0, \ldots, 0]^T \]

In Eq.(13), $N$ denotes the degree of polynomials.

Therefore, the optimization of filter coefficients can be expressed as

\[ \min_{c} \varepsilon(h, g), \quad \text{s.t. } A_{eq} \cdot C = B_{eq} \]

(14)

Because the filter coefficients $c_n(i)$ can be piecewise optimized in each unit interval, the optimization problem above can be divided into $L/2$ sub-parts as follows

\[ \min_{c} \varepsilon(h, g), \quad \text{s.t. } A_{eq} \cdot C = B_{eq} \]

(15)

where

\[ A_{eq} = \begin{bmatrix} \varphi_0(0), \varphi_1(0), \ldots, \varphi_n(0) \\
\varphi_0(1), \varphi_1(1), \ldots, \varphi_n(1) \\
\ldots \\
[1, 0]^T, i = 0 \\
[0, 0]^T, i = 1, 2, \ldots, L/2 - 1 \\
\end{bmatrix}, \]

\[ B_{eq} = \begin{bmatrix} 1 \end{bmatrix} \]

The optimization of the filter coefficients can be gotten by minimizing the least square error function, it is desirable

\[ \varepsilon(h, g) = \int_{0}^{1} (h(t) - g(t))^2 dt \]

(17)

According to Eqs. (7) and (17), the error function can be rewritten as follows

\[ \varepsilon(h, g) = \int_{0}^{1} \left( \sum_{n=0}^{N} c_n(i) \varphi_n(t) - g(t) \right)^2 dt \]

(18)

Let

\[ C_i = [c_0(i), c_1(i), \ldots, c_n(i)]^T, \]

\[ \Psi_n = \int_{0}^{1} \varphi_n(t) dt, \quad \Gamma_n = 2 \int_{0}^{1} \varphi_n(t) \cdot g_i(t) dt. \]

Neglect the constants, and further, the error function $\varepsilon(h, g)$ in Eq.(18) can be expressed as

\[ \varepsilon'(h, g_i) = C_i^T \cdot A \cdot C_i + B^T \cdot C_i, \]

(19)

where $\varepsilon'(h, g_i)$ denotes the function $\varepsilon(h, g)$ with neglecting the constants, and

\[ A = \begin{bmatrix} \Psi_0, \Psi_1, \ldots, \Psi_n \\
\Psi_0, \Psi_1, \ldots, \Psi_n \\
\ldots \\
\Psi_0, \Psi_1, \ldots, \Psi_n \end{bmatrix}, \]

\[ B = [-\Gamma_0, -\Gamma_1, \ldots, -\Gamma_N]^T. \]

Then the problem in Eq.(15) converts to the quadratic programming problem, which can be written as

\[ \min_{c_i} \quad C_i^T \cdot A \cdot C_i + B^T \cdot C_i, \quad \text{s.t. } A_{eq_i} \cdot C_i = B_{eq_i}, \]

(20)

Thus, given the corresponding parameters of interpolation filters, the optimization coefficients can be obtained, then the polynomial-based interpolation filter can be realized. Further we analyze the conditions of the quadratic programming problem.

When $\varphi(0) = 0$, the conditions $A_{eq} \cdot C_{eq} = B_{eq}$ is true only when $C_{eq} = 0$, and when $\varphi(1) = 0$, the conditions can be simplified as follows

\[ \begin{cases} [\varphi_0(0), \varphi_1(0), \ldots, \varphi_n(0)] \cdot C_{eq} = 1 \\
[\varphi_0(0), \varphi_1(0), \ldots, \varphi_n(0)] \cdot C_{eq} = 0, i \neq 0, \\
\varphi_0(1) \cdot c_0(i) = 0 \Rightarrow c_0(i) = 0 \end{cases}. \]

(21)
Hence, if a polynomial basis function $\varphi(t)$ meets $\varphi(1) = 0$ and $\varphi(0) \neq 0$, then $c_0(i) = 0$ holds, which not only reduces the dimensions of the original conditions but also improves the efficiency of finding an optimum solution, at the moment, the outputs of $v(0)$ are equal to zero, which can be removed directly, it reduces the hardware costs of the Farrow structure and made it parallel a $(L - 1)$-order FIR filter.

According to the discussion mentioned above, the expression of interpolation function can be written as follows

$$\varphi(t) = \sum_{n=0}^{\infty} c_n t^n, \quad (23)$$

The interpolation function needs to satisfy the following conditions

$$\begin{cases} 
\varphi(0) \neq 0 \\
\varphi(1) = 0
\end{cases}, \quad (24)$$

that is,

$$\begin{align*}
\sum_{n=0}^{N} c_n(0) \varphi_n(0) &= 1 \\
\sum_{n=0}^{N} c_n(i) \varphi_n(0) &= 0, i = 1, 2, ..., L/2 - 1 \quad (25) \\
\sum_{n=0}^{N} c_n(i) \varphi_n(1) &= 0, i = 0, 1, 2, ..., L/2 - 1
\end{align*}$$

Then the coefficients of the interpolation function can be derived by substituting the Eq.(25) into the expression of the polynomial function, as given by Eq.(23)

$$\begin{align*}
&c_0 \neq 0 \\
&\sum_{n=-\infty}^{\infty} c_n = 0
\end{align*} \quad (26)$$

In Eq.(25), we can see, different kinds of filters can be constructed by these filter coefficients. However, with the increasing of polynomial orders, the interpolation filter not only becomes very complex for the implementation of Farrow structure but also degradation in the efficiency of signal processing. A low order $N$ is an ideal choice for interpolation function. Hence, we choose the linear function as the basis of the proposed filter as follows

$$\phi(t) = kt + b, \quad k, b \in R, k \neq 0, \quad (27)$$

According the conditions in equations (24) and (25), it is desirable

$$\varphi(t) = k(t - 1), k \in R, k \neq 0. \quad (28)$$

From Eq.(28) we can see, for all linear basis functions, only the functions $k(t - 1)$ satisfy the conditions that we mentioned above, that is, only these functions satisfy $c_k(i) = 0$. Future we can simplify the basis function when take $k = 1$, then we get the proposed interpolation function as follows

$$\varphi(t) = t - 1. \quad (29)$$

### 3.2. Filter Optimization in Frequency Domain

The polynomial-based interpolation filter in time domain generally is not a very practical approach to the application of signal processing. Because the frequency band of the signal is usually known, but the time domain characters of signals is unknown. The proposed method mentioned above can also convert the filter optimization problem from time domain to frequency domain.

It is desired to design the reconstruction filter which can be implemented by using the Farrow structure or its modifications, and not be controlled by the length of the filter and the degree of the polynomials. Thus, the goal is to optimize the Farrow structure coefficients, and the reconstructed impulse response $H(f)$ in frequency domain. Based on this idea and the demands above, given $N$, $L$, and a compact subset $X$ as well as a desired function $G(f)$ that is continuous for $f \in X$ and a weight function $w(f)$ that is positive for $f \in X$, find $(M + 1)N/2$ filter coefficients $c_n(i)$ can be implemented as follows

$$\min C \in \mathbb{C}^N \quad s.t. \quad Aef.C = Bef^* \quad (30)$$

where $H$ and $G$ denote the Fourier transform of functions $h(t)$ and $g(t)$ respectively, and

$$C = [c_0, c_1, ..., c_{N-1}, 0, 0, ..., 0]^T,$$

$$Aef = \begin{bmatrix}
0, 0, ..., 0, Aef_0, ..., 0 \\
0, 0, ..., 0, Aef_1, ..., 0 \\
\vdots \\
0, 0, ..., 0, Aef_{N-1}, Aef_0, ..., Aef_{L-1}
\end{bmatrix}, \quad (31)$$

$$Bef = [\Phi_x(0), \Phi_x(0), ..., \Phi_x(0)]$$

where $\Phi(f)$ is the Fourier transform of polynomial basis function $\varphi(t)$. Because the reconstructed
impulse response $h(t)$ is a symmetrical function around $t = 0$, so the frequency response $H(f)$ is real, the error function $\varepsilon(H, G)$ in frequency domain can be defined as

$$\varepsilon(H, G) = \max_{f \in X} \left[ |W(f)|H(f) - D(f)|^2 \right] df,$$

where the frequency band $X$ consists of the specific pass-bands and stop-bands. $G(f)$ is the frequency response of the desired interpolation function. $W(f)$ is the weighting function in accordance with the requirements of target interpolation filter in the specific frequency band, that is, the greater the weight, the smaller the peak error. In order to improve the attenuation of the filter in stop-band, we can preset a small pass-band weight.

Therefore, the problem in Eq.(30) can be written as follows

$$\min_{C} \varepsilon(H, G) = \max_{f \in X} \left[ |W(f)|H(f) - D(f)|^2 \right] df,$$

subject to $Aef_i \cdot C = Bef_i$, (33)

Actually, in practical application, it is often unnecessarily to get the frequency response of the target interpolation function on the overall frequency bands, and only need to satisfy the characteristics in some specific frequency bands.

Thus, define the error function in a specific frequency band as $\varepsilon_i(H, G)$. Further, in a piecewise frequency band $f_i \in X$, let

$$\varepsilon_i(H, G) = \max_{f \in X} \left[ |W(f)|H(f_i) - D(f_i)|^2 \right] df_i,$$

(34)

Then the problem mentioned above can be expressed as follows

$$\min_{C} \varepsilon_i(H, G),$$

subject to $Aef_i \cdot C_i = Bef_i$, (35)

where

$$Aef_i = \begin{bmatrix} \Phi_0(0), \Phi_0(0), \ldots, \Phi_0(0) \\ \Phi_0(j2\pi f_i), \Phi_0(j2\pi f_i), \ldots, \Phi_0(j2\pi f_i) \end{bmatrix},$$

$$Bef_i = \begin{bmatrix} \{2\pi \delta(2\pi f_i), 0\}^T, i = 0 \\ \{0, 0\}^T, i = 1, 2, \ldots, L/2 - 1 \end{bmatrix},$$

(36)

According the Eqs. (8) and (34), the error function in the frequency domain can be written as follows

$$\varepsilon_i(H, G) = \max_{f \in X} \left[ W(f_i) \cdot H_i - G_i(f_i)|^2 \right] df_i,$$

$$= \max_{f \in X} \left[ \left( \sum_{n=0}^{N} c_n(i) \Phi_n(f_i) \right)^2 \right] df_i,$$

(37)

where $\Phi(n)$ is the symmetrical function around $t = 0$, so the frequency response $H(f)$ is real, the error function $\varepsilon_i(H, G)$ in frequency domain can be defined as

$$\varepsilon_i(H, G) = \max_{f \in X} \left[ |W(f)|H(f) - D(f)|^2 \right] df_i,$$

(32)

Neglect the constants, and further, the error function $\varepsilon_i(H, G)$ can be expressed

$$\varepsilon_i'(H, G) = C_i^T \cdot A \cdot C_i + B^T \cdot C_i,$$

(39)

where $\varepsilon_i'(H, G)$ denotes the function $\varepsilon(H, G)$ with neglecting the constants, and

$$A = \begin{bmatrix} \Theta_0, \Theta_1, \ldots, \Theta_N \\ \Theta_{N+1}, \Theta_{N+2}, \ldots, \Theta_{2N+1} \\ \ldots \\ \Theta_{N^2}, \Theta_{N^2+1}, \ldots, \Theta_{N^2+N^2+N+1} \end{bmatrix},$$

$$B = \begin{bmatrix} -\Lambda_0, -\Lambda_1, \ldots, -\Lambda_N \end{bmatrix}^T,$$

(40)

Hence, the problem in Eq.(35) converts to the quadratic programming problem, which can be written as

$$\min \max_{f \in X} \left[ C_i^T \cdot A \cdot C_i + B^T \cdot C_i \right],$$

subject to $Aef_i \cdot C_i = Bef_i$, (41)

Thus, the filter optimization coefficients can be got by solving the quadratic programming problem. Given the length of filter, the degree of polynomials, pass bands and stop bands, and the weighting function, etc, the filter optimization coefficients in frequency domain can be obtained by utilizing the proposed method. In section 5, some examples will be given to verify the performance of the proposed algorithm.
We can also preset the weight values properly in piecewise frequency band \( f_j \) according to the performance requirements of signal processing. Because the polynomial-based interpolation filter can be implemented efficiently by using the Farrow structure or its modifications, in the following the Farrow structure will be introduced.

4. The Farrow Structure of Interpolation Filters

If the impulse response \( h(t) \) is a piecewise polynomial, then the Farrow structure or its modifications can be implemented. The Farrow structure has some features to make it attractive in signal processing:

- The number of FIR sub-filters is \( N+1 \), and the length of these sub-filters is \( L \).
- Filter coefficients are determined directly by the polynomial coefficients of the impulse response.
- The main advantage of the Farrow structure is that all the filter coefficients are fixed.
- The resolution of the fractional interval is limited only by the precision of the arithmetic, not by the size of the coefficient memory.
- These characters of the Farrow structure make it a very attractive structure to be implemented using a VLSI circuit or a signal processor, etc.

This filter structure consists of parallel FIR filters with fixed coefficient values. The desired time instant for the interpolated output samples can be easily controlled by properly weighting the output samples of these FIR filters. In this paper the proposed interpolation filters have a piecewise polynomial impulse response, so they can be implemented efficiently by using the Farrow structure or its modifications. When the interpolation function is \( \varphi(t) = t - 1 \), by substituting it into Eq.(6), and the filter coefficients have the feature as follows

\[
\begin{cases}
  c_n(i) = c_n(i-1), \text{n is even} \\
  c_n(i) = -c_n(i-1), \text{n is odd}
\end{cases}
\]  

When exploiting the above symmetries, the number of the coefficients to be implemented can be reduced from \( (N+1)L \) to \( (N+1)L/2 \).

According to the equation

\[
h_j(\mu_k) = \sum_{n=0}^{N} c_n(i)\varphi^n(\mu_k) = \sum_{n=0}^{N} c'_n(i)\mu_k^n,
\]

When \( \varphi(\mu_k) = a\mu_k + b \), and apply the binomial theorem

\[
c'_n(i) = \sum_{m=0}^{N} c_m(i)c_m^n \cdot a^n \cdot b^{m-n},
\]

Then we can get that \( \varphi'(t) = -t \), which has a Farrow structure, and further, the Farrow structure can be simplified

\[
c'_0(i) = \sum_{n=0}^{N} c_n(i) \cdot (-1)^n = \left\{ \begin{array}{ll} 1, & i = 0 \\ 0, & \text{others} \end{array} \right.
\]

According Eqs.(43) and (45), the Farrow structure of the \((t-1)\)-based interpolation filter can be got, which is shown in Fig. 1.

Fig. 1. Farrow structure for the \((t-1)\)-based interpolation filter.
In Fig. 1, the length of interpolation filter is \(2L\), the degree of polynomials is \(N\). It is shown that the proposed interpolation filter consists of \(NL\) adders, \(NL + N + 1\) multipliers and \(N(L - 1)\) delays, and in practical application, the delays can be shared \([11]\), that is, only \(L - 1\) delays are needed. In the following part, some examples will be given to compare the performance of the proposed interpolation filter with the Lagrange-based interpolator.

5. Numerical Examples

This section provides two design examples to illustrate the flexibility of the mentioned MiniMax method, the least-mean-square synthesis method, and the performances of the proposed interpolation filters. In example 1, we will discuss the approximating performance of the proposed interpolation filter. In example 2, we will compare the error rate of the proposed interpolation filter over those obtained using the Lagrange design method, and the simulation signals are three kinds of linear modulation signals, i.e. MSK, 32QAM, 256QAM.

### 5.1. Example 1

Take the interpolation function \(\varphi(t) = t - 1\), and utilize the proposed optimal method to approximate the raised-cosine filters in time domain, it is shown in Fig 2, and the optimization coefficients of the filter in time domain are shown in Table 1. In Fig. 2, it has shown that the proposed interpolation filters have a good approximating performance to the original low-pass filters in time domain. The parameter \(\beta\) denotes the roll off factor of raised-cosine filter. When exploiting the symmetries in Eq. (6), the number of the optimization coefficients to be implemented can be halved, thus, Table 1 only shows halves of the optimization coefficients. The length of the interpolation filter is \(L = 6\), the degree of the polynomials is \(N = 3\).

![Graph](image)

**Table 1.** Optimization coefficients of the proposed filter versus \(\beta = 0.5\).

<table>
<thead>
<tr>
<th>(i)</th>
<th>(c_0(t))</th>
<th>(c_1(t))</th>
<th>(c_2(t))</th>
<th>(c_3(t))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-0.7472</td>
<td>1.5581</td>
<td>1.3054</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.1356</td>
<td>0.5536</td>
<td>0.6892</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.0033</td>
<td>0.1516</td>
<td>0.1483</td>
</tr>
</tbody>
</table>

### 5.2. Example 2

For some linear modulation transmitted signals, such as PAM, PSK, QAM, the received signals of the receivers can be expressed as follows

\[
x(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT - \varepsilon(t)T)e^{j\theta(t)} + n(t)
\]

where \(a_n\) denotes the sending complex data, \(T\) is symbol period, \(\varepsilon(t)\) denotes the timing error function, \(\theta(t)\) is the carrier phase difference function, \(g(t)\) is impulse response function of the system, and \(n(t)\) denotes the zero mean Gaussian white noise. The functions \(\varepsilon(t)\) and \(\theta(t)\) usually change slowly, so we assume they are constants in a short time. The simulation below assumes that the response of system is a raised-cosine function with the roll off factor \(\alpha = 0.5\), preset the timing error \(\varepsilon = 0.5\), and carrier phase difference \(\theta = 0\).

Given these parameters, a simulation is carried out to analyze three kinds of modulation signals mentioned above, that is, MSK, QAM32, and QAM256, respectively. The received signals are sampled by four times of the symbol rate. When the timing error is obtained, then use the \((t - 1)\)-based interpolation filter and Lagrange-based interpolator to filter the sampling signals, respectively. Fig. 3, Fig. 4 and Fig. 5 show the constellation charts of timing synchronization of the output signals.
In Fig. 3, Fig. 4 and Fig. 5, it is shown that the constellation charts of the output signals by the proposed interpolation filter is much smaller than by the Lagrange-based interpolator, which means that the proposed interpolation filter has a much lower error rate than the Lagrange-based one. Thus, the proposed interpolation filter has a better filtering performance than the Lagrange-based interpolator.

Assume the filter length is $2L$, the polynomials degree is $N$, then comparing with the Lagrange-based interpolation filter, this proposed one saves $(N+2)L-2$ adders and $(N+2)L-2$ multipliers due to $c_{ij}(i)=0, i=0,1,2,...,N$ and the symmetries in Eq.(6). In most practical applications, the reconstruction pulse response $h(t)$ can get a good approximating to the original signal when $N$, the degree of polynomial, is small [7]. Hence, in this case the $(t-1)\text{-based}$ interpolation filter reduces the hardware costs greatly. Table 2 can explain it clearly.

So, comparing with Lagrange-based interpolation filter the proposed one not only reduces the hard costs greatly but also has a better filtering performance.

Table 2. Hardware costs for the interpolation filters.

<table>
<thead>
<tr>
<th></th>
<th>$L$</th>
<th>$N$</th>
<th>Multipliers ($a^1$)</th>
<th>Adders ($b^1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>1</td>
<td>22 (72.4%)</td>
<td>20 (74.7%)</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>1</td>
<td>42 (73.9%)</td>
<td>40 (74.8%)</td>
</tr>
<tr>
<td>1</td>
<td>60</td>
<td>2</td>
<td>63 (65.4%)</td>
<td>60 (66.5%)</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>2</td>
<td>123 (66.0%)</td>
<td>120 (66.6%)</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>3</td>
<td>154 (61.8%)</td>
<td>150 (24.6%)</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>3</td>
<td>304 (62.1%)</td>
<td>300 (62.5%)</td>
</tr>
</tbody>
</table>

$a$ and $b$ denote the percentage of saving multipliers and adders, respectively.

![MSK signal by polynomial Lagrange-based interpolation filter](image1)

![MSK signal by polynomial (t-1)-based interpolation filter](image2)

(a) Lagrange-based interpolation filter        (b) (t-1)-based interpolation filter

**Fig. 3.** Constellation charts of output signal (MSK) by the filters.

![QAM32 signal by polynomial Lagrange-based interpolation filter](image3)

![QAM32 signal by polynomial (t-1)-based interpolation filter](image4)

(a) Lagrange-based interpolation filter        (b) (t-1)-based interpolation filter

**Fig. 4.** Constellation charts of output signal (QAM32) by the filters.
6. Conclusions

The main contribution of this paper was a proposed general design for the polynomial-based interpolation filter. The minimax method or least-mean-square method to optimize the filter coefficients in time domain and frequency domain, respectively. The length of the interpolation filter, the degree of the polynomials, the pass-bands and stop-bands, the desired response, and weighting function were used to optimize the filter coefficients.

We also found the association between the proposed interpolation functions and the Farrow structure, further it will extend the ability to meet different signal processing environments. We have analyzed the decomposition expression of the reconstructed impulse response, realized the filter Farrow structure by using the proposed polynomial function. We found that the first items of the optimization coefficients of the proposed interpolation filter equal to zero. Actually, in most practical applications the reconstructed impulse response can get a good approximating performance to the desired one by using a low approximation order \( N \) (generally \( N \leq 3 \)).

In this respect, the proposed interpolation filter saves the hardware costs greatly. If the length of filter is \( 2L \), the degree of polynomials is \( N \), then compares with the Lagrange-based interpolator, the proposed one saves \( (N+2)L - 1 \) adders and \( (N+2)L - 1 \) multipliers. Examples indicated that the proposed interpolation filter not only has a good filtering performance but also reduces the implementation complexity of the Farrow structure.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (10876006, 61001049), to a certain degree and it also benefited by the Fundamental Research Funds for the Central Universities (ZYGX2011J013) and Opening Topic Fund for Key Laboratory of Computer Architecture (CARCH201103). This work was also supported by Technology Project Foundation of Chongqing Education Committee (KJ131110), and Key Laboratory of signal and information processing Chongqing Three Gorges University.

References


