POLYNOMIAL-BASED FILTERS IN BANDPASS INTERPOLATION AND SAMPLING RATE CONVERSION

Djordje Babic
Faculty of Information Technology, SlobomirP University
Bijeljina
BOSNIA AND HERZEGOVINA
Tel: +38755231; E-mail: djordje.babic@ptt.yu

ABSTRACT
If the SRC is performed between arbitrary sampling rates, then the SRC factor can be a ratio of two very large integers or even an irrational number. An efficient way to reduce the implementation complexity of a SRC system in those cases is to use polynomial-based interpolation filters with the impulse response $h_a(t)$ having the following properties. $h_a(t)$ is nonzero for an interval $0 \leq t < NT$ with $N$ being an even integer and is expressible in each subinterval of length $T$ by means of a polynomial of a low order. The length of polynomial segments $T$ can be equal to the input or output sampling interval, a fraction of the input or output sampling interval, or an integer multiple of the input or output sampling interval. The advantage of mimicking the above system lies in the fact that the actual implementations can be performed effectively by using the Farrow structure or its modifications.

So far, the polynomial-based filters have been used only for baseband interpolation and SRC. In the literature, it has been observed that in the passband applications the filter order of the polynomial-based filter increases. In this paper, we study application of the polynomial-based filters for the bandpass interpolation and SRC. It is shown that the polynomial-based filters, implemented using Farrow structure or its modification, can be effectively used also in the passband applications. We show through examples that the filter order is the same as for the corresponding baseband filter having same requirements.

1. INTRODUCTION

Discrete-time filter structures are used in many applications to interpolate new sample values at arbitrary points between the discrete-time input samples [1], [2]. Some applications are symbol synchronization in digital receivers, continuous time signal processing and arbitrary sampling rate conversion.

The interpolation filter can be implemented in an inexpensive way if its impulse response $h_a(t)$ can be expressed in a closed form by means of some mathematical function. The most straightforward way is to use polynomial-based interpolation filters. In these cases, an efficient overall implementation based on mimicking the system of Fig. 1 can be achieved by generating $h_a(t)$ to have the following properties [3], [4]. First, $h_a(t)$ is nonzero only in the interval $0 \leq t < NT$ with $N$ being an even integer. Second, in each subinterval $nT \leq t < (n+1)T$ for $n=0, 1, ..., N-2$ $h_a(t)$ is expressible as a polynomial of the given low order $M$. Third, $h_a(t)$ is symmetric around $t = NT/2$ to guarantee the phase linearity of the resulting overall system. The length of polynomial segments $T$ can be selected to be equal to the input or output sampling interval, a fraction of the input or output sampling interval, or an integer multiple of the input or output sampling interval. The advantage of mimicking the above system lies in the fact that the actual implementation can be efficiently performed by using the Farrow structure [5] or its modifications [3], [4], [6]–[9]. The main advantage of the Farrow structure lies in the fact that it consists of fixed finite-impulse response (FIR) filters and there is only one changeable parameter being the so-called fractional interval $\mu$. Besides this, the control of $\mu$ is easier during the operation than in the corresponding coefficient memory implementations [4], and the resolution of $\mu$ is limited only by the precision of arithmetic and not by the size of the memory. These characteristics of the Farrow structure make it a very attractive structure to be implemented using a VLSI circuit or a signal processor [4].

The structures derived from Farrow structure can be grouped as follows. First, the modified Farrow structure [3], [4] used for interpolation and the transposed modified Farrow structure [7], [8] used for decimation are considered. The advantage of the modified Farrow structure compared to the original one [3] is that the filter coefficients are symmetrical. By exploiting these symmetries, the number of multipliers required in implementing the fixed filters is halved compared to the original structure. In these two cases, the polynomial segments $T$ of $h_a(t)$ are equal to the input and output sampling rates in the interpolation and decimation cases, respectively. For the second class of structure, $T$ is a fraction of the sampling period and the structures under consideration are the systems consisting of a fixed linear-phase FIR interpolator in cascade with the modified Farrow structure and the transposed modified Farrow structure in cascade with a fixed linear-phase FIR decimator. Finally, the so-called prolonged Farrow structures are introduced for both interpolation and decimation purposes [11]. The generations of these structures differ from those of the structures in the first class in the fact that $T$ is a multiple of either the input or output sampling period.

In [3], [4], [6]–[9] several design methods for polynomial-based interpolation filters have been developed in the time and in frequency domain. Using the frequency domain approach, it is possible to design the polynomial-based filter realized in the form of the Farrow structure with an arbitrary baseband (zero center frequency) frequency response. However, there are some applications where there is a need for bandpass (nonzero center frequency) interpolation. Some example applications are: bandpass sampling, bandpass sampling rate conversion, rational filter banks, etc. This paper shows that the polynomial-based filters can be effectively used for bandpass interpolation as well. It is also shown that for the given passband and stopband requirements the filter order does not increase compared to the baseband with the same requirements. The design methods presented in [3] and [4] and further studied in [10] can be effectively used for the design of bandpass polynomial interpolation filters with minor modifications. Furthermore, efficient narrow passband polynomial-based in-
Polynomial-Based Analogue Filters

This section gives necessary details about polynomial-based filters in general, their definition, and realization by using Farrow structure and its modifications. In order to derive efficient implementation form for the polynomial-based interpolation filter, we use sampling rate conversion as generic application for interpolation.

In the most general case, the interpolation (sampling rate conversion as well) can be regarded as a process of resampling an analogue signal according to the hybrid analogue/digital model shown in Fig. 1 [1]. In this system, the discrete-time input sequence \( x(n) \) with the sampling rate equal to \( F_\text{in} = 1/T_\text{in} \) is first converted into an analog continuous-time signal \( x_a(t) \) using the ideal digital-to-analog converter (DAC). The resulting signal is the following sum of the weighted and shifted impulses:

\[
x_a(t) = \sum_{k=-\infty}^{\infty} x(k) \delta(t - kT_\text{in}) ,
\]

where \( \delta(t) \) is the analogue Dirac delta function. This sequence is then filtered using an analogue filter with the impulse response \( h_a(t) \) to generate the continuous-time signal given by

\[
y_a(t) = \sum_{k=-\infty}^{\infty} x(k) h_a(t - kT_\text{in}) .
\]

Finally, the output \( y_a(t) \) is sampled at the time instants \( t = nT_\text{out} \) using the ideal analog-to-digital converter (ADC) to produce the following output discrete-time sequence \( y(n) \):

\[
y(n) = y_a(nT_\text{out}) = \sum_{k=-\infty}^{\infty} x(k) h_a(T_\text{out} - kT_\text{in}) .
\]

As has been originally suggested in [3], [4] when deriving the modified Farrow structure for interpolation, it is beneficial to construct \( h_a(t) \) as follows:

\[
h_a(t) = \sum_{n=0}^{N-1} c_n(n)f_n(T, t)
\]

where \( N \) is an even integer, the basis functions \( f_n(T, t) \) are given by

\[
f_n(T, t) = \begin{cases} 
(2T - nT) \left(\frac{2T - nT}{T}\right) - 1 
& \text{for } nT \leq t < (n + 1)T \\
0 & \text{otherwise},
\end{cases}
\]

and the \( c_n(n) \)'s are the adjustable parameters being related to each other as

\[
c_n(N - n - 1) = \begin{cases} 
c_n(n) & \text{for } m \text{ even} \\
c_n(n) & \text{for } m \text{ odd}
\end{cases}
\]

\( n = 0, 1, \ldots, N-1 \). As shown in Fig. 2, the resulting \( h_a(t) \) is characterized by the following properties:

1. \( h_a(t) \) is nonzero for \( 0 \leq t < NT \) and zero elsewhere.
2. In each subinterval \( nT \leq t < (n + 1)T \) for \( n = 0, 1, \ldots, N-1 \), \( h_a(t) \) is expressible as \( h_a(t) = \sum_{m=0}^{M} c_m(n)f_m(n, T, t) \), that is, \( h_a(t) \) is a piecewise-polynomial for \( 0 \leq t < NT \) and is of degree \( M \) in each subinterval.
3. \( h_a(t) \) is symmetric around \( t = NT/2 \), that is, \( h_a(NT-t) = h_a(t) \) for the time instants \( t = nT \) for \( n = 0, 1, \ldots, N/2-1 \) and \( n = N/2+1, N/2+2, \ldots, N \).

Based on Property 3, it is guaranteed that the resulting overall system has a linear phase that is a very attractive property in many applications. Furthermore, the generation of the above \( h_a(t) \) guarantees that in the frequency domain the zero-phase frequency response, when omitting the linear-phase term, is expressible as (see [3] and [4] for details)

\[
H_a(j2\pi f) = \sum_{n=0}^{N-1} \sum_{m=0}^{M} c_m(n)G_m(n, T, f),
\]

where \( G_m(n, T, t) \) is the Fourier transform of

\[
g_m(n, T, t) = (-1)^n f_n(n, T, t - NT/2) + f_n(N - 1 - n, T, t - NT/2).
\]

The above form is a direct consequence of the symmetry properties of the \( c_m(n) \)'s. Since the above approximating function is linear with respect to the unknowns, it enables one to optimize the overall filter to meet the given criteria in a manner similar to that used for synthesizing various types of linear-phase FIR filters [2]. In the above, \( T \), the length of polynomial segments, is not fixed. Therefore, \( T \) can be used to define different implementation structures discussed in Introduction. As will be seen in the following six sections, \( T \) can be chosen as \( T = \beta T_\text{in} \) or \( T = T_\text{out} \) where \( \beta \) is unity, an integer, one divided by an integer. The selection depends on whether decimation or interpolation is under consideration and on the structure performing the desired sampling rate conversion.

### 2.1 Modified Farrow structure

In this case, the desired \( h_a(t) \) is obtained from Eq. (4) by selecting \( T \) to be input sampling period in Fig. 1, that is, \( T = T_\text{in} \), yielding to [4]

\[
y(t) = \sum_{m=0}^{M} v_m(n, t)(2\mu_i - 1)^m,
\]

where
are the output samples of the $M+1$ fixed-linear-phase FIR filters transfer functions

$$C_m(z) = \sum_{k=0}^{N-1} c_m(k)z^{-k}$$

satisfying the symmetry properties given by Eq. (6). This results in the desired modified Farrow structure [3], [4] as shown in Fig. 3. When using the above-mentioned symmetries by using the direct-form structure exploiting the coefficient symmetries when implementing $M+1$ linear-phase FIR transfer functions $C_m(z)$ in Fig. 3, the resulting structure requires $(M+1)/2$ multipliers. Furthermore, by sharing the delays between the $C_m(z)$'s, the overall number of delays can be reduced to be $N-1$.

2.2. Modified transposed Farrow structure

When generating the transposed modified Farrow structure (Fig. 4), the desired $h_\mu(t)$ is obtained from Eq. (4) by selecting $T$ to be output sampling period in Fig. 1, that is, $T = T_{out}$, giving [8]

$$y(l) = \sum_{n=0}^{M-1} \sum_{m=0}^{N-1} c_m(n)v_m(n,l),$$

where

$$x(k)\text{ I&D} \rightarrow \sum_{m=0}^{N-1} c_m(n)\rightarrow y(l)$$

Fig. 4. Transposed modified Farrow structure.

$$v_m(n,l) = \sum_{k=0}^{N-1} x(n_l - k + N/2)c_m(k)$$

(10)

with

$$k_{low}(n,l) = \left( l + N/2 - n - 1 \right) T_{out} / T_a,$$

(14)

and

$$\mu_k = kT_{out} / T_a - \left[ kT_{out} / T_a \right].$$

(17)

2.3. Prolonged modified Farrow structures

In this case, the desired $h_\mu(t)$ is obtained from Eq. (4) by selecting $T$ to be multiple of the input sampling period in Fig. 1, that is, $T = JT_{in}$, yielding [10], [11]

$$y(l) = \sum_{n=0}^{M-1} \sum_{j=0}^{N-1} c_m(n)\left( n_j - j + N/2 \right) 2^{\mu_k(j) - 1} ,$$

(18)

where the $\mu_k(j)$’s for $j = 0, 1, \ldots, J-1$ are $J$ fractional intervals depending on the $\mu_k$ included in Eq. (18) as follows:

$$\mu_k(j) = (j + \mu_k) / J ,$$

(19)

The resulting implementation structure is given in Fig. 5.

2.4. Prolonged transposed modified Farrow structure

When generating the prolonged transposed modified Farrow structure (Fig. 6), the desired $h_\mu(t)$ is obtained from Eq. (4) by selecting $T$ to be an integer multiple of the output sampling period in Fig. 1, that is, $T = JT_{out}$, giving [10], [11]

$$y_m(l) = \sum_{n=0}^{M-1} \sum_{m=0}^{N-1} c_m(n)v(m,n,l),$$

(20)

where

$$v(m,n,l) = \sum_{k=0}^{N-1} x(k)\left( 2^{\mu_k(j) - 1} \right)^m.$$
and 
\[ k_{up}(l,n,j) = \begin{cases} s(l,n,j) - 1 & \text{if } s(l,n,j) \text{ is an integer} \\ [s(l,n,j)] & \text{otherwise,} \end{cases} \] (23)

where 
\[ s(l,n,j) = (l - j)/J - n + N/2)T_{out}/T_{in}. \] (24)

### 2.5. Structures Consisting of a Cascade of a Linear-Phase FIR Interpolator and a Modification of the Farrow Structure

The proposed structure consisting of two basic building blocks is shown in Fig. 7. In the first block, the input sampling rate \( F_{in} \) of the input sequence \( x(n) \) is increased by an integer factor of \( L \) by using a linear-phase FIR filter transfer function of the following form:

\[ H_I(z) = \sum_{k=0}^{K_I-1} h_I(k)z^{-k}, \] (25)

where the impulse-response of \( H_I(z) \) is symmetric, that is,

\[ h_I(K_I - k) = h_I(k) \quad \text{for} \quad k = 0, 1, \ldots, K_I. \] (26)

It should be pointed out that in many cases, the computation complexity of the fixed linear-phase FIR interpolator can be significantly reduced by using a two-stage interpolator [1]. In this case, there exist two linear-phase FIR interpolator transfer functions \( H_I^{(1)}(z) \) and \( H_I^{(2)}(z) \) with the first one increasing the sampling rate by an integer factor \( L_1 \), followed by the second one increasing the resulting sampling rate further by an integer factor \( L_2 \). It is well known that this implementation is equivalent to that of Fig. 7 with \( L = L_1L_2 \) and

\[ H_I(z) = H_I^{(1)}(z^{L_1})H_I^{(2)}(z). \] (27)

After using a linear-phase fixed FIR interpolator, the sampling rate is increased by an integer factor \( L_1 \) followed by the second block, where the output sample rate is now \( LF_{in} \), instead of \( F_{in} \). Second, the desired output \( y(t) \) is obtained from Eq. (4) by selecting \( T \) to be \( T = T_{out}/L \).

The dual structure is shown in Fig. 8 consisting of two building blocks in cascade. The first block is the transposed Farrow structure that generates the output sequence, denoted by \( z(j) \), based on the overall input sequence \( x(n) \) such that the output sampling rate is an integer multiple of the desired output sampling rate \( F_{out} \), that is, the output sampling rate is \( LF_{in} \), with \( L \) being an integer. The output of the transposed Farrow structure \( z(j) \) is thus expressible as

\[ z(j) = \sum_{n=0}^{M-1} \sum_{j=0}^{N-1} c_{in}(n)v_{in}(n,j), \] (30)

where

\[ c_{in}(n,j) = k_{up}(n,j), \] (31)

and

\[ k_{up}(n,j) = \begin{cases} s(n,j) - 1 & \text{if } s(n,j) \text{ is an integer} \\ [s(n,j)] & \text{otherwise,} \end{cases} \] (32)

where

\[ s(n,j) = (j + N/2 - n)F_{out}/LT_{in}, \] (33)

and

\[ \mu_k = kL_{out}/T_{out} - kL_{in}/T_{out}. \] (35)

This block can be implemented using the transposed modified Farrow structure considered in Section 4 with the following differences. First, the output sample rate is now \( LF_{out} \), instead of \( F_{out} \). Second, the desired output \( h_d(t) \) is obtained from Eq. (12) by selecting \( T \) to be \( T = T_{out}/L \).

In the second block, the sampling rate is reduced by integer factor \( L \) in order to generate the desired output sequence \( y(t) \) with the sampling rate equal to \( F_{out} \). The desired sampling rate reduction can be performed using a single-stage or a two-stage linear-phase FIR decimator.

### 3. DESIGN OF THE BANDPASS FARROW STRUCTURE

The ultimate remaining problem in the model of Fig. 1 is to determine \( h_d(t) \) such that the following two conditions are met:
Condition 1: The components of the input sequence \(x(n)\) in the resulting pass-band region \(|f| \in \Omega_p\) of the output sequence \(y(l)\) are preserved and the other components are eliminated as well as possible in the decimation case \((R < 1)\) [in the interpolation case for \((R > 1)\)].

Condition 2: The overall system has an efficient implementation directly in the digital domain.

It can be easily verified (see, e.g., [1]) that first condition is ideally satisfied if the Fourier transforms of the input and output sequences \(x(n)\) and \(y(l)\) are related through

\[
Y_{\{\omega/2^jF_{out}\}}(\omega) = RX_{\{\omega/2^jF_{in}\}}(\omega) \quad \text{for } |\omega| \in \Omega_p.
\]

The corresponding Fourier transform of this filter is given by

\[
H_{\{\omega/2^j\}}(\omega) = \begin{cases} F_{\omega} & \text{for } |\omega| \in \Omega_p \\ 0 & \text{for } |\omega| \notin \Omega_p, \end{cases}
\]

where

\[
F_{\omega} = \begin{cases} F_{\text{out}} & \text{for } R < 1 \\ F_{\text{in}} & \text{for } R > 1. \end{cases}
\]

In order to generate a realizable overall system, the criteria for the Fourier transform \(h(\omega)\) are stated as follows:

\[
|1 - \delta_p| F_{\omega} \leq |H_{\{\omega/2^j\}}(\omega)| \leq |1 - \delta_p| F_{\omega} \quad \text{for } |\omega| \in \Omega_p
\]

where \(\Omega_p\) is the passband defined as \(\Omega_p = [f_{p1}, f_{p2}]\) and \(f_{p2}\) are the edges of the desired passband. While \(\Omega_p = [0, f_s/2]\) defines the stopband region. Above, \(F_{\omega}\) equals \(F_{\text{out}}\) in decimation case, and \(F_{\omega}\) in interpolation case. When satisfying these criteria the frequency components in the passband region \(\Omega_p\) are preserved according to the given tolerance.

The intuitive solution to the design problem is a filter modulation method. The bandpass filter is obtained, by designing the baseband filter and after that the impulse response is multiplied with \(\cos(2 \pi f_0 t)\) where \(f_0\) is a center frequency. However, this filter is not possible to implement by using Farrow structure or its modifications. Here, we propose the design method of [3]-[4] with some modifications. We show that the filter order does not increase compared to the corresponding baseband filter.

The minimax and the least-mean-square optimization methods introduced in [3]-[4] are probably the most convenient and the most flexible solutions for designing polynomial-based interpolation filters. The design procedure has been generalized in [10], and modified for optimization of prolonged and transposed polynomial-based filters.

The used design method has several design parameters. First of all, the design parameters include passband and stopband regions \(\Omega_p\) and \(\Omega_s\). The desired filter may have several passbands and stopbands as stated in [4]. Next, the minimum stopband attenuation \(A_s\) and maximum allowable passband ripple \(A_p\) are also included. These four design parameters are closely related to the system specifications, which also give the desired function as design parameter. Other design parameters are the number of polynomial segments \(N\) and the order of polynomial \(M\), which determine the number of multipliers in overall structure. Finally, some weighting function can be used to give different weight to passband and stopband [4], [10]. Though, the optimization criterion is stated for whole frequency axis up to infinity, in actual optimization we consider only finite frequency range. The envelope of the frequency response of basis functions is decreasing. Thus, after certain frequency point, the overall frequency response is guaranteed to be below required minimum stopband attenuation.

The actual optimization problem can be defined in minimax or least-mean-square sense, subject to some time-domain conditions [3], [4]. In most DSP applications, only frequency-domain conditions are given. In this case, it is possible to exploit all degrees of freedom given by the polynomial-based filter to satisfy requirements. However, in some application there are also some time-domain conditions. For example, in some image processing application it is desired to have continuous approximating function, or it may be desired to have continuous first derivative. These constrained optimization problems can be converted into corresponding unconstrained ones by modifying properly the approximating function and the desired function [3]. This is usually achieved by finding relations between coefficients of the polynomial-based filter.

Minimax Optimization Problem: Given \(N, M\), and a compact subset \(X \subset \{0, \infty\}\) as well as a desired function \(D(f)\) being continuous for \(f \in X\) and a weight function \(W(f)\) being positive for \(f \in X\), find the \((M+1)/2\) unknown coefficients \(c_n(n)\) to minimize

\[
\delta_{\omega} = \max_{f \in X} W(f) |H_{\omega}(f) - D(f)|
\]

subject to the given time-domain conditions of \(h(\omega)\). Here, \(H_{\omega}(f)\) is the frequency response and \(D(f)\) is desired function according to the specifications.

Least-Mean-Square Optimization Problem: Given the same parameters and functions as for the above problem, find the \((M+1)/2\) unknown coefficients \(c_n(n)\) to minimize

\[
\delta_{\omega} = \int_{X} [W(f)|H_{\omega}(f) - D(f)|] ^{2} df
\]

subject to the given time-domain conditions of \(h(\omega)\).

### 4. Design Examples

This section illustrates design of several bandpass filters, by means of an example, for the decimation case. Because of the duality between the decimators and interpolators, the results of this comparison are also valid for the interpolation structures.

It is desired to design the bandpass filter with the following requirements. The passband edges are \(f_{p1} = 0.2F_{\text{out}}\) and \(f_{p2} = 0.3F_{\text{out}}\). The stopband edges are located at \(f_{s1} = 0.1F_{\text{out}}\) and \(f_{s2} = 0.4F_{\text{out}}\), respectively, whereas the minimum stopband attenuation is 60 dB and the maximum allowable magnitude deviation from unity in the passband is 0.1. For simplicity, we concentrate in the sequel on designing filters in such a manner that the passband average is scaled to be unity.

When using the minimax optimization criterion proposed in [3], [4], the transposed modified Farrow structure of Section 2.2 meets the given criteria by \(N = 24\) and \(M = 3\). For this structure, the overall number of multipliers is \((M+1)/2 + M = 51\). The magnitude and impulse responses for the filter are shown in Fig. 9 and 10.

If we design corresponding baseband filter, with \(f_{p} = 0.15F_{\text{out}}\)
and $F_{out}$, the same requirements are met with transposed Farrow structure with $N = 24$ and $M = 3$. The filter length and polynomial order are the same. The magnitude and impulse responses for the filter are shown in Fig. 11 and 12.

By using the prolonged transposed Farrow structure of Section 2.4, it is possible to design the narrowband bandpass filter. The requirements are as follows. The passband edges are $f_{p1} = 0.1 F_{out}$ and $f_{p2} = 0.15 F_{out}$. The stopband edges are located at $f_{s1} = 0.05 F_{out}$ and $f_{s2} = 0.2 F_{out}$, respectively, whereas the minimum stopband attenuation is 60 dB and the maximum allowable magnitude deviation from unity in the passband is 0.1. The magnitude and impulse responses for the filter are shown in Fig. 11 and 12.

5. CONCLUSIONS

This paper shows that the polynomial-based filters implemented by using Farrow structure can be effectively used for design and implementation of bandpass filter. These bandpass filters may be used in rational filter banks, bandpass interpolation and sampling rate conversion. It has been shown, that the same design methods as for baseband filters can be used. For the bandpass filters designed here, the filter order is the same as for the corresponding baseband filters with the same requirements.

6. REFERENCES


