# PULSE-SHAPE FILTER DESIGN IN DIGITAL MODEMS EMPLOYING CIC FILTERS

Scott L. Talbot (University of Utah: Salt Lake City, Utah, USA; stalbot@eng.utah.edu);
Behrouz Farhang-Boroujeny (University of Utah: Salt Lake City, Utah, USA; farhang@ece.utah.edu).

## ABSTRACT

Cascaded integrator-comb (CIC) filters are frequently used in software radio modems as interpolation and decimation filters. However, CIC filters are known to exhibit passband droop and thus introduce inter-symbol interference (ISI). Many methods have been proposed in order to improve the passband characteristics of CIC filters. These methods usually increase hardware complexity in the modem. An alternative approach is to achieve the Nyquist (M) property and eliminate ISI by modifying the pulse-shaping filter (PSF) coefficients. Previously, PSF coefficients have been found through linear programming. However, linear programming is not necessarily the simplest or the best method in finding the filter coefficients. Recently a simple iterative least-squares algorithm has been proposed for Nyquist (M) filter design. We extend this method for use in a modem using CIC filters for interpolation and decimation. This method may also be used if additional filtering is required, or if other types of interpolation or decimation filters are used. When compared to the best published work, the proposed design method produces a PSF that when used in cascade with the CIC filters yields superior results in minimizing passband ripple, increasing stopband attenuation, and reducing ISI.

## 1. INTRODUCTION

One of the advantages in software defined radio modems is that they give a single platform the flexibility to handle multiple modulation methods, data rates, and signal processing techniques. Thus, it is important that the system have a high degree of programmability. It is also important to find signal processing techniques that are common to many or all of the waveforms since this provides simplicity in the modem.

Cascaded integrator-comb (CIC) filters were first introduced by Hogenauer [1] and are frequently used in software radio modems because they provide hardware simplicity and because of their flexibility in functioning as interpolation and decimation filter. Although CIC filters are efficient in the sense that they do not require multiplications, they do exhibit passband droop and introduce inter-symbol interference (ISI). Many techniques have been proposed in order to improve these negative characteristics of CIC filters [2]-[4]. The authors of [2] propose to sharpen the CIC frequency response by using multiple copies of a CIC filter. Their decimator structure then consists of the sharpened CIC filter and several half-band filters. In [3], the authors use an interpolated second-order polynomial filter, along with a single CIC filter, simplified half-band filters, and a programmable FIR filter. The authors of [4] propose to compensate for the CIC filter deficiencies and achieve Nyquist (M) characteristics in the cascade of the CIC filter and pulse-shaping filter (PSF) by simply modifying the PSF coefficients. The coefficients are chosen through the use of linear programming, but linear programming can be fairly complicated and may not necessarily yield the best results.

In this paper, we adopt a similar approach to that in [4] in that we wish to achieve the Nyquist (M) property with the cascade of the CIC filter and PSF by modifying the PSF coefficients. However, instead of using linear programming we use a novel Nyquist (M) filter design method to derive the filter coefficients [5]. We extend this method to solve for the PSF coefficients. Our approach produces filter coefficients that yield a system with less passband ripple, more stopband attenuation, and less ISI than the method proposed in [4]. This method may also be used to trade-off between filter length, Nyquist (M) characteristics, and stopband attenuation. Moreover, it can be used in combination with the methods of [2], [3] to achieve the Nyquist (M) property when a PSF is used in cascade with those decimation filters.

We begin by formulating the problem and introducing the Nyquist criterion in Section 2. We then discuss the design procedure for finding the Nyquist (M) filter in Section 3. Using the Nyquist (M) filter it is simple then to find the PSF coefficients for a system with a PSF/CIC filter cascade. This is detailed in Section 4. We then illustrate the design procedure through some design examples and compare these with those obtained in [4]. Finally we draw some conclusions in Section 6.

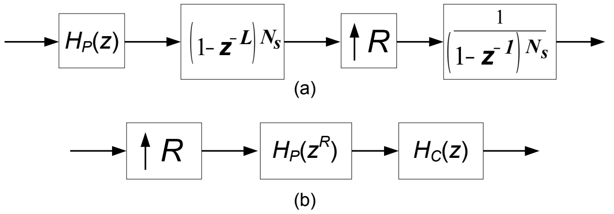


**Figure 1. (a) Digital modulator with CIC filter (b) Modified digital modulator with CIC filter.**

## 2. PROBLEM FORMULATION

We consider the pulse-shaping and interpolation portion of a digital modulator as shown in Fig. 1 (a). Here $H_p(z)$ is the PSF, and the three following blocks constitute a CIC interpolation filter with $N_s$ stages, sample rate conversion $R$, and parameter $L$ [1]. The parameter $L$ provides some limited control of zero placement in the CIC filter. The corresponding structure found in the demodulator consists of the reversal of these blocks and the replacement of the upsample-by-$R$ block by a downsample-by-$R$ block. Using the noble identity [6, pp. 119] and by letting

$$H_C(z)=\left[\frac{\left(1-z^{-RL}\right)}{\left(1-z^{-1}\right)}\right]^{N_s}=\left[\sum_{k=0}^{RL-1} z^{-k}\right]^{N_s}, \qquad (1)$$

we can modify the structure in Fig. 1 (a) to obtain the structure shown in Fig. 1 (b).

Let the cascade of $H_P(z^R)$, and $H_C(z)$ be defined as

$$H_{NQ}(z)=H_P\left(z^R\right)H_C(z). \qquad (2)$$

In the time domain, (2) is written as

$$h_{NQ}(n)=h_{PU}(n)*h_C(n), \qquad (3)$$

where * denotes convolution, and

$$h_{PU}(n)=\begin{cases} h_P(n/R), & \text{if } n \text{ is an integer multiple of } R \\ 0, & \text{otherwise,} \end{cases} \qquad (4)$$

i.e., the sequence $h_{PU}(n)$ is an upsampled version of $h_P(n)$. We wish to design a filter $H_{NQ}(z)$ of length $N_{NQ}+1$ such that $G(z)=H_{NQ}(z)H_{NQ}(z^{-1})$ satisfies the Nyquist criterion

$$g(Mn)=\begin{cases} 1, & n=0 \\ 0, & \text{otherwise} \end{cases} \qquad (5)$$

where $M$ is an integer called the over-sampling factor. It indicates the number of filter coefficients per symbol interval. The sequence $g(n)$ is the inverse $z$-transform of $G(z)$. Note that the Nyquist condition (5) is with reference to the high sampling rate of Fig. 1. A filter $G(z)$ that satisfies (5) is called Nyquist (M), [6], and $H_{NQ}(z)$ is referred to as a square-root Nyquist (M) filter.

**Table 1**
**Square-root Nyquist (M) Filter Design**

Inputs

$N_{NQ}$: filter length
$M$: oversampling factor
$\alpha$: rolloff factor
$\boldsymbol{\Gamma}$: diagonal weight matrix

Initialization

- Construct $\mathbf{S}'$ using (11), (12), and (19).
- Construct $\boldsymbol{\Phi}'$ using (14), and (15).
- Use Cholesky factorization to obtain $\mathbf{C}$ from $\boldsymbol{\Phi}' = \mathbf{C}^T\mathbf{C}$.
- Choose a target vector $\mathbf{d}$ and form the vector $\mathbf{u}$.
- Construct the initial vector $\mathbf{h}'_0$ from the samples of a square-root raised-cosine pulse-shape with rolloff factor $\alpha$.
- Let $i = 0$.

Iterations

- $\mathbf{B} = [\mathbf{I} \otimes \mathbf{h}'^T_i]\,\mathbf{S}'$
- $\mathbf{D} = [\mathbf{B}^T\ \mathbf{C}^T]^T$
- $\mathbf{h}' = (\mathbf{D}^T\boldsymbol{\Gamma}^2\mathbf{D})^{-1}\mathbf{D}^T\boldsymbol{\Gamma}^2\mathbf{u}$
- $\mathbf{h}'_{i+1} = (\mathbf{h}'_i + \mathbf{h}')/2$
- Increment $i$

Final step

- $\mathbf{h}' = \mathbf{h}'_i$
- Construct $\mathbf{h}$ from $\mathbf{h} = \mathbf{E}\mathbf{h}'$

We assume that the PSF coefficients are programmable. Thus for every different configuration of the CIC filter the PSF coefficients can be modified so that the cascade of these two filters $H_{NQ}(z)$ maintains the Nyquist ($M$) property. For each configuration of the CIC filter the filter coefficients $h_C(n)$ are known. When a particular $H_{NQ}(z)$ is found, the remaining task is to find the coefficients of the PSF, $h_P(n)$.

## 3. NYQUIST (M) FILTER DESIGN

The first part of the task is to design the Nyquist (M) filter $H_{NQ}(z)$. For this we use the design method presented in [5]. This method uses an iterative weighted least-squares algorithm that attempts to minimize the magnitude of the stopband response of $H_{NQ}(z)$. This particular cost function can be formulated as

$$\xi_s=\int_{f_o}^{1-f_o}\left|H_{NQ}\left(e^{j2\pi f}\right)\right|^2 df. \qquad (6)$$

where $f_o$ is the stopband edge. In [5], the stopband edge is defined as

$$f_o = \frac{1+\alpha}{2M}, \tag{7}$$

following the notation of the square-root raised cosine filter in [7], where $\alpha$ is the rolloff factor. The algorithm is summarized in Table 1. The algorithm takes as its inputs $N_{NQ}$ (one less than the length of $H_{NQ}(z)$ ), the oversampling factor $M$, the rolloff factor $\alpha$, and the weight matrix $\mathbf{\Gamma}$.

This algorithm designs a linear-phase low-pass filter, so the filter coefficients are symmetric. Let us define the column vector

$$\mathbf{h}'_{NQ} = \left[h_{NQ}(0) \quad h_{NQ}(1) \quad \cdots \quad h_{NQ}((N_{NQ}-1)/2)\right]^{\mathrm{T}}$$

when $N_{NQ}$ is odd, and

$$\mathbf{h}'_{NQ} = \left[h_{NQ}(0) \quad h_{NQ}(1) \quad \cdots \quad h_{NQ}(N_{NQ}/2)\right]^{\mathrm{T}}$$

when $N_{NQ}$ is even, and T denotes transposition. The full-length vector $\mathbf{h}_{NQ}$ is thus

$$\mathbf{h}_{NQ} = \mathbf{E}\mathbf{h}'_{NQ} \tag{8}$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{I} \\ \mathbf{J} \end{bmatrix}. \tag{9}$$

$\mathbf{I}$ is the identity matrix, and $\mathbf{J}$, for $N_{NQ}$ odd, is the antidiagonal matrix with the antidiagonal elements of 1 and, for $N_{NQ}$ even, is obtained by removing the first row the antidiagonal matrix.

The conditions imposed in (5) are shown in [5] to be equivalent to

$$\mathbf{h}'^{\mathrm{T}}_{NQ}\mathbf{S}'_n\mathbf{h}'_{NQ} = \begin{cases} 1, & n = 0 \\ 0, & n = mM, m \neq 0 \end{cases} \tag{10}$$

for $n = 0, 1, \ldots, N_{NQ}$ where

$$\mathbf{S}'_n = \mathbf{E}^T\mathbf{S}_n\mathbf{E} \tag{11}$$

are constant matrices with the elements of $\mathbf{S}_n$ given by

$$[\mathbf{S}_n]_{k,l} = \begin{cases} 1, & \text{when } k-l = n \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The cost function (6) is shown to be equivalent to

$$\xi_s = \mathbf{h}'^{\mathrm{T}}_{NQ}\mathbf{\Phi}'\mathbf{h}'_{NQ} \tag{13}$$

where

$$\mathbf{\Phi}' = \mathbf{E}^{\mathrm{T}}\mathbf{\Phi}\mathbf{E} \tag{14}$$

and the elements of $\mathbf{\Phi}$ are given by

$$\phi_{kl} = \begin{cases} 1-2f_o, & k = l \\ -2f_o\operatorname{sinc}(2f_o(k-l)), & k \neq l. \end{cases} \tag{15}$$

The constraints in (10) can be relaxed in order to trade-off between other design considerations. Other design goals besides the Nyquist (M) criterion include minimizing the magnitude of the stopband of the filter, providing increased immunity to timing jitter, and reducing the peak-to-average power ratio of the modulated signal [5]. In this paper, we only consider minimizing the stopband response, and achieving the Nyquist (M) property.

The algorithm allows the constraints of (10) to be relaxed as in

$$\mathbf{h}'^{\mathrm{T}}_{NQ}\mathbf{S}'_n\mathbf{h}'_{NQ} \approx d_n \quad n = 0,1,\ldots,N_{NQ} \tag{16}$$

where the $d_n$ are a set of desired/target values. Combining the set of equations in (16) we get

$$\mathbf{B}\mathbf{h}'_{NQ} \approx \mathbf{d} \tag{17}$$

where

$$\mathbf{B} = \left[\mathbf{I}_{N_{NQ}+1} \otimes \mathbf{h}'^{\mathrm{T}}_{NQ}\right] \mathbf{S}', \tag{18}$$

$$\mathbf{S}' = \left[\mathbf{S}'^{\mathrm{T}}_0 \ \mathbf{S}'^{\mathrm{T}}_1 \cdots \mathbf{S}'^{\mathrm{T}}_{N_{NQ}}\right]^{\mathrm{T}}, \tag{19}$$

and

$$\mathbf{d} = \left[d_0 \ d_1 \cdots \ d_{N_{NQ}}\right]^{\mathrm{T}}.$$

$\mathbf{I}_{N_{NQ}+1}$ is the identity matrix of size $(N_{NQ}+1)$x$(N_{NQ}+1)$, and $\otimes$ denotes the Kronecker product.

The author of [5] applies the Cholesky factorization to expand $\mathbf{\Phi}'$ as, $\mathbf{\Phi}' = \mathbf{C}^{\mathrm{T}}\mathbf{C}$ where $\mathbf{C}$ is an upper triangular matrix, and uses this to rearrange (13) as

$$\xi_s = \left(\mathbf{C}\mathbf{h}'_{NQ}\right)^{\mathrm{T}}\mathbf{C}\mathbf{h}'_{NQ} = \left\|\mathbf{C}\mathbf{h}'_{NQ}\right\|^2. \tag{20}$$

Here, $\|\cdot\|$ denotes the 2-norm of a vector. In order to minimize the cost function $\xi_s$ one must attempt to minimize the length of the vector $\mathbf{C}\mathbf{h}'_{NQ}$. Thus the additional design goals

$$\mathbf{C}\mathbf{h}'_{NQ} \approx \mathbf{0} \tag{21}$$

are added, where $\mathbf{0}$ is a column vector with zero elements.

Combining (17) and (21), one obtains

$$\mathbf{D}\mathbf{h}'_{NQ} \approx \mathbf{u} \tag{22}$$

where $\mathbf{D} = \left[\mathbf{B}^T \mathbf{C}^T\right]^T$ and $\mathbf{u} = \left[\mathbf{d}^T \mathbf{0}^T\right]^T$. The approximation (22) is an over-determined system of soft equations for which we seek a solution for the unknown vector $\mathbf{h}'_{NQ}$.

To solve (22), [5] defines the error vector

$$\mathbf{v} = \mathbf{\Gamma}\left(\mathbf{D}\mathbf{h}'_{NQ} - \mathbf{u}\right), \tag{23}$$

where $\mathbf{\Gamma}$ is a diagonal matrix whose diagonal elements are a set of weights to be given to the elements of the difference $\mathbf{D}\mathbf{h}'_{NQ} - \mathbf{u}$. Larger weights are assigned to the elements where minimization should be emphasized. A weight of zero is assigned to the elements that are treated as *don't care* elements. In this way, the different design goals mentioned above can be achieved. In [5], the author chooses the solution as the vector $\mathbf{h}'_{NQ}$ that minimizes the norm of the vector $\mathbf{v}$.

Since we attempt to find a Nyquist (M) filter and the square-root raised-cosine pulse-shape is a Nyquist pulse shape that closely fits our desired design goals, we (like [5]) initialize the algorithm with the square-root raised-cosine coefficients as shown in Table 1. In this way, we find our desired filter $H_{NQ}(z)$.

## 4. PULSE-SHAPE FILTER DESIGN

Now that we have designed a Nyquist (M) filter, it is simple to obtain the desired PSF coefficients from (3). Suppose that we desire the length of the sequence $h_P(n)$ to be of length $N$+1. We then define

$$\mathbf{h}_P = [h_P(0)\ h_P(1)\ \ldots\ h_P(N)]^T.$$

The sequence $h_C(n)$ is of length $N_C+1 = N_s(RL-1)+1$, and we define

$$\mathbf{h}_C = [h_C(0)\ h_C(1)\ \ldots\ h_C(N_C)]^T.$$

We thus have a length $N_{NQ}+1 = N_s(RL-1)+RN+1$ sequence

$$\mathbf{h}_{NQ} = [h_{NQ}(0)\ h_{NQ}(1)\ \ldots\ h_{NQ}(N_{NQ})]^T.$$

It is important that we design a Nyquist (M) filter of length $N_{NQ}$ defined above. We can then rewrite (3) as

$$\mathbf{h}_{NQ} = \mathbf{H}_C\,\mathbf{h}_P \tag{24}$$

$$\mathbf{h}_{NQ} = \begin{bmatrix} h_C(0) & 0 & \cdots & 0 \\ h_C(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ h_C(R-1) & 0 & \cdots & 0 \\ h_C(R) & h_C(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_C(N_C) & h_C(N_C-R) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_C(N_C) \end{bmatrix} \begin{bmatrix} h_P(0) \\ h_P(1) \\ \vdots \\ \vdots \\ h_P(N) \end{bmatrix},$$

thus, $\mathbf{H}_C$ is a $(N_{NQ}+1)$x$(N+1)$ matrix.

When a solution has been found for $\mathbf{h}_{NQ}$ using the algorithm in Table 1, we can use (24) to find a solution for $\mathbf{h}_P$. Equation (24) is an overdetermined system of linear equations. Note that $\mathbf{H}_C$ has full column rank, thus, there exists a unique least-squares solution $\mathbf{h}_P$ for the system of equations in (24) [8, pp. 222-223]. There are many ways of computing the least-squares solution [8], but one straightforward method is to calculate

$$\mathbf{h}_P = \mathbf{H}_C^+\mathbf{h}_{NQ}, \tag{25}$$

where $\mathbf{H}_C^+$ is the Moore-Penrose pseudo-inverse defined as

$$\mathbf{H}_C^+ = \left(\mathbf{H}_C^T\mathbf{H}_C\right)^{-1}\mathbf{H}_C^T. \tag{26}$$

Once the filter coefficients $\mathbf{h}_P(n)$ have been calculated they are quantized to the desired bit-precision.

Please note that although we have presented the problem using CIC filters as interpolation and decimation filters, the problem may be reformulated using different interpolation and decimation filters and/or additional filters. The overdetermined system of equations will involve a matrix that is different than $\mathbf{H}_C$ but the design method will remain the same.

## 5. DESIGN EXAMPLE

In this section, we present design results and compare them with those found in [4] using the same system setup. For the system we have a 71-tap PSF with $M$=12, and 12-bit coefficients. The CIC filter consists of $N_s$ = 4 stages, has sample-rate conversion factor $R$ = 3, and parameter L = 1. We first found a suitable $H_{NQ}(z)$ using the method in Table 1 with $N_{NQ} = N_s(RL-1)+RN+1 = 219$, $\alpha = 0.25$. Recall that $\mathbf{\Gamma}$ is a weight matrix corresponding to how the different conditions given in $\mathbf{u}$ are weighted. The vector $\mathbf{u}$ is split into two sections: $\mathbf{d}$ represents the conditions that relate to the Nyquist (M) property, and the remaining part corresponds to the stopband response minimization. The weights in $\mathbf{\Gamma}$ that correspond to the stopband minimization are all given a value of 1. The weights in $\mathbf{\Gamma}$ corresponding to the elements $d_n$, where $n = mM$ are given the value of 2. All remaining elements in $\mathbf{\Gamma}$ are given a value of 0.

Once the Nyquist (M) filter $\mathbf{H}_{NQ}(z)$ is found, we use (25) to find the least-squares solution $\mathbf{h}_P$. The response of this filter (i.e., the composite of $H_C(z)$ and $H_P(z)$) compared with the Wasserman filter is given in Fig. 2. Only a portion of the frequency response has been shown for the purpose of clarity. The integration of the magnitude of the filter response $H_{NQ}(z)$ over the stopband, i.e., (6), is shown in Table 2. We can see that the proposed method has better stopband attenuation than the Wasserman filter.

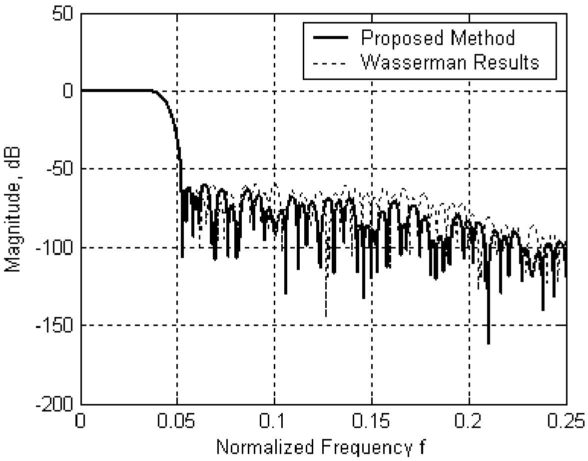


**Figure 2. Filter response comparison between the proposed method and the Wasserman method.**

**Table 2: Filter Performance**

| | $\xi_s = \int_{f_o}^{1-f_o} \left\| H_P\left(e^{j2\pi f}\right) H_C\left(e^{j2\pi f}\right)\right\|^2 df$ |
|---|---|
| Proposed | 3.2426x10$^{-8}$ |
| Wasserman | 7.6376x10$^{-8}$ |
| | $\sum_{\text{zero crossings}} \left[h_P(n) * h_C(n)\right] * \left[h_P(n) * h_C(n)\right]$ |
| Proposed | 2.4549x10$^{-4}$ |
| Wasserman | 3.8309x10$^{-3}$ |

Table 2 also shows the amount of ISI that the different methods exhibit. We can see that the proposed method once again outperforms the Wasserman method. A comparison of the passband responses of the two filters is shown in Fig. 3. From this we can see that the passband response of the current method has less ripple. The filter coefficients are shown in Table 3.

## 6. CONCLUSION

In this paper, we have shown a straight-forward method for designing the PSF when used in cascade with CIC filters. This approach is simpler than linear programming and may use less hardware than the methods of [2], and [3]. The system approach is similar to that taken by [4], but the design method is different. Our approach is to use an iterative weighted least-squares algorithm to design a Nyquist (M) filter, and then to find the least-squares solution to an overdetermined system of linear equations to find the PSF coefficients. Our approach yielded a filter with less passband ripple, better stopband attenuation, and less ISI than the Wasserman filter which relies on complicated linear programming.

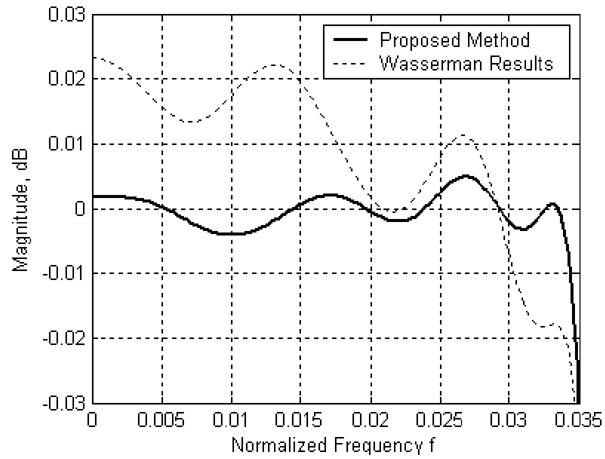


**Figure 3. Passband response comparison between the proposed method and the Wasserman method.**

**Table 3: $H_P(z)$ Filter Coefficients**

| Filter Tap | | | | Value |
|---|---|---|---|---|
| *h*(0) | = | *h*(70) | = | -0.0004882812500 |
| *h*(1) | = | *h*(69) | = | 0.0000000000000 |
| *h*(2) | = | *h*(68) | = | 0.0004882812500 |
| *h*(3) | = | *h*(67) | = | 0.0009765625000 |
| *h*(4) | = | *h*(66) | = | 0.0007324218750 |
| *h*(5) | = | *h*(65) | = | -0.0004882812500 |
| *h*(6) | = | *h*(64) | = | -0.0019531250000 |
| *h*(7) | = | *h*(63) | = | -0.0024414062500 |
| *h*(8) | = | *h*(62) | = | -0.0012207031250 |
| *h*(9) | = | *h*(61) | = | 0.0019531250000 |
| *h*(10) | = | *h*(60) | = | 0.0046386718750 |
| *h*(11) | = | *h*(59) | = | 0.0048828125000 |
| *h*(12) | = | *h*(58) | = | 0.0009765625000 |
| *h*(13) | = | *h*(57) | = | -0.0048828125000 |
| *h*(14) | = | *h*(56) | = | -0.0092773437500 |
| *h*(15) | = | *h*(55) | = | -0.0080566406250 |
| *h*(16) | = | *h*(54) | = | 0.0002441406250 |
| *h*(17) | = | *h*(53) | = | 0.0109863281250 |
| *h*(18) | = | *h*(52) | = | 0.0168457031250 |
| *h*(19) | = | *h*(51) | = | 0.0117187500000 |
| *h*(20) | = | *h*(50) | = | -0.0039062500000 |
| *h*(21) | = | *h*(49) | = | -0.0214843750000 |
| *h*(22) | = | *h*(48) | = | -0.0283203125000 |
| *h*(23) | = | *h*(47) | = | -0.0158691406250 |
| *h*(24) | = | *h*(46) | = | 0.0126953125000 |
| *h*(25) | = | *h*(45) | = | 0.0407714843750 |
| *h*(26) | = | *h*(44) | = | 0.0473632812500 |
| *h*(27) | = | *h*(43) | = | 0.0200195312500 |
| *h*(28) | = | *h*(42) | = | -0.0327148437500 |
| *h*(29) | = | *h*(41) | = | -0.0817871093750 |
| *h*(30) | = | *h*(40) | = | -0.0888671875000 |
| *h*(31) | = | *h*(39) | = | -0.0280761718750 |
| *h*(32) | = | *h*(38) | = | 0.0981445312500 |
| *h*(33) | = | *h*(37) | = | 0.2526855468750 |
| *h*(34) | = | *h*(36) | = | 0.3796386718750 |
| | | *h*(35) | = | 0.4287109375000 |

 

## 7. REFERENCES


[1] E.B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 29, pp. 155-162, Apr. 1981.

[2] A.Y. Kwentus, Z. Jiang, and A.N. Willson Jr., "Application of Filter Sharpening to Cascaded Integrator-comb Decimation Filters," *IEEE Trans. Signal Processing,* vol. 45, pp. 457-467, Feb. 1997.

[3] H.J. Oh, S. Kim, G. Choi, and Y. H. Lee, "On the Use of Interpolated Second-order Polynomials for Efficient Filter Design in Programmable Downconversion," *IEEE J. Select. Areas Commun.,* vol. 17, pp. 551-560, Apr. 1999.

[4] L. Wasserman, and A.N. Willson, "A Variable-rate Filtering System for Digital Communications," in *Proc. ICASSP,* 1999, pp. 1497-1500.

[5] B. Farhang-Boroujeny, "A Universal Square-root Nyquist (M) Filter Design for Digital Communication Systems," Submitted to *IEEE Trans. Signal Processing.* Also found in these proceedings.

[6] P.P. Vaidyanathan, *Multirate Systems and Filter Banks,* Prentice Hall, Upper Saddle River, NJ, 1993.

[7] J.G. Proakis, *Digital Communications*. McGraw-Hill, New York, 3rd Edition, 1995.

[8] G.H. Golub, and C.F. Loan, *Matrix Computations,* 2nd ed., The Johns Hopkins University Press, Baltimore, Maryland, 1989.