Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 701–706

ISBN 978-83-60810-51-4

Fractional Delay Filter Design for Sample Rate Conversion

Marek Blok
Faculty of Electronics, Telecommunications and Informatics
Gdańsk University of Technology
11/12 G. Narutowicza Street, 80-233 Gdańsk Wrzeszcz, Poland
Email: mblok@eti.pg.gda.pl

***Abstract*—With a large number of different standards of sample rates we often need to use sample rate conversion algorithms. If the resampling ratio is not expressed as the ratio of small integer numbers or is not a fixed value, the sample rate conversion algorithm based on fractional delay filters might be used since it allows for arbitrary resampling ratios. The performance of such algorithm depends solely on the method used to design fractional delay filters. In this paper we propose a novel classification of fractional delay filter design methods dividing them into three general categories: optimal fractional filter design, offset window method and polyphase decomposition. The proposed classification is based on differences in properties of the sample rate conversion algorithm based on fractional delay filters.**

I. Introduction

DIGITAL representation of analog signals has a lot of advantages but the problem arise when the sample rate with which signal was recored is different from the sample rate required for further processing. With a large number of sample rate standards available today such situation is quite common. A typical example is the CD (compact disc) to DAT (digital audio tape) conversion, when a signal sampled with the sample rate $F_{s,CD} = 44.1$ kSa/s needs to be converted into a signal with the sample rate $F_{s,DAT} = 48$ kSa/s [1].

The classic approach to the sample rate conversion (SRC) is presented in Fig. 1 [2]. First, the input sample rate F_{s1} is increased L-times by inserting $L-1$ zeros between each two input samples. Next, a lowpass filter is used to remove spectral images located at multiples of the input sample rate. This replaces zeros, which have been previously inserted, with values of the interpolated input signal. Finally, a sample rate is decreased M-times by leaving only every M-th sample, thus signal with the sample rate $F_{s2} = L/MF_{s1}$ is obtained. Interpolation and decimation factors used in this process can be computed using the following formulas

$$L = F_{s2}/\,\mathrm{gcd}(F_{s1}, F_{s2}) \quad (1)$$

$$M = F_{s1}/\,\mathrm{gcd}(F_{s1}, F_{s2}) \quad (2)$$

Such an approach to the SRC is relatively simple in implementation and interpretation but at the same time computationally inefficient. Nevertheless, computational efficiency can be readily improved with polyphase structures [3]. The more serious problem is that when L or M are about a hundred or larger, like in the aforementioned CD to DAT conversion with $L = 160$ and $M = 147$, design of the interpolation filter is problematic. The required transition band becomes very narrow and a very long impulse response of the interpolation filter is required. Therefore the main challenge in the SRC algorithm implementation is the interpolation filter design. A designer tries to obtain a shortest possible filter, which means lower computational costs and filter delay, fulfilling given specification described by passband ripples, stopband attenuation, and width and location of transition band. For very long impulse responses optimal solutions might not be reachable and less efficient filter design methods need to be used, like the window method. Moreover, when a ratio of input and output sample rates is an irrational number or when it varies in time, the factors L and M cannot be determined and the interpolation filter cannot be specified.

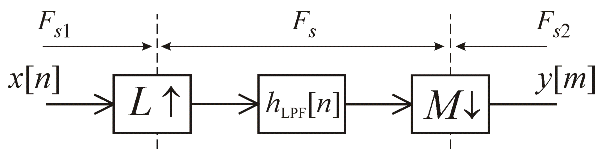

Fig. 1. Classic sample rate conversion algorithm.

Therefore, in practice, the classic sample rate conversion algorithm can be used only to resample a signal by factors which can be represented as a ratio of two relatively small integer numbers. For other resampling ratios a different approach to the resampling needs to be used.

Let us notice that the SRC algorithm actually has to compute values of signal samples in new time instants located between original samples (Fig. 2) [1], [4]–[6]. This means that we can treat each output sample of the SRC algorithm as an input sample delayed or advanced by a fraction of the input sampling period.

The fractional delay (FD) between the current output sample $y[m]$ and the nearest input sample $x[n]$ can be computed using the following recursive formula [7], [8]

$$d[m] = d[m-1] - F_{s1}/F_{s2} + \Delta n[m] \in [-0.5, 0.5) \quad (3)$$

where the resampling ratio $F_{s1}/F_{s2} = M/L$ and $\Delta n[m]$ is a number of new samples required in the input buffer for

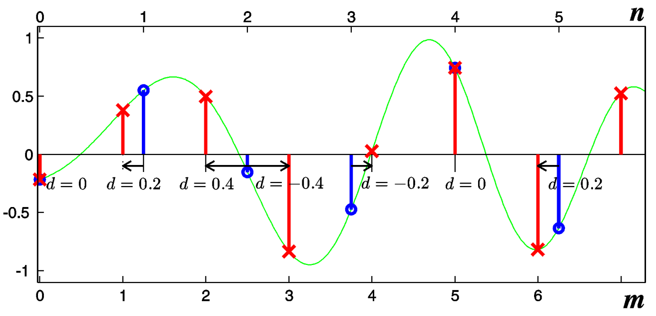

Fig. 2. Presentation of resampling process in the time domain. Black arrows show the delay between output samples and closest input samples. $L = 5$, $M = 4$.

computation of the next output sample

$$\Delta n[m] = \mathrm{round}(F_{s1}/F_{s2} - d[m-1]) \tag{4}$$

Using those two parameters we can formulate the resampling algorithm (Fig. 3):

1) start with $d[0] = 0$ and $\Delta n[0] = 0$,
2) wait for $\Delta n[m]$ new samples in the input buffer,
3) compute the output sample $y[m]$ delayed by $d[m]$,
4) calculate $\Delta n[m]$ and $d[m]$ for the next m and go back to step (2).

In the variable (or adjustable) fractional delay (VFD) approach to the SRC, computation of each output sample requires the FD filter with the impulse response approximately L-times shorter than the length of interpolation filter required in the classic approach. Moreover, contrary to the classic SRC, the resampling ratio doesn't need to be rational. To achieve this we need, however, to calculate a new set of filter coefficients for every output sample.

II. Fractional Delay Filter

The SRC based on FD filters has many advantages over the classic approach but its performance depends on a method used to design FD filters. The ideal frequency response of the FD filter with the total delay τ_d is defined by the following formula [9]

$$H_{\mathrm{id}}(f) = \exp(-j2\pi f \tau_d), \quad f \in [-0.5, 0.5) \tag{5}$$

which corresponds to the ideal impulse response

$$h_{\mathrm{id}}[n] = \mathrm{sinc}(n - \tau_d) \tag{6}$$

Since the ideal impulse response is infinite and non-causal, in practical applications, the frequency response (5) must be approximated with an finite order filter. In this paper we will consider the approximation with the use of FIR FD filter with the frequency response

$$H_N(f) = \sum_{n=0}^{N-1} h[n] \exp(-j2\pi f n) \tag{7}$$

where $h[n]$ is the impulse response of the length N.

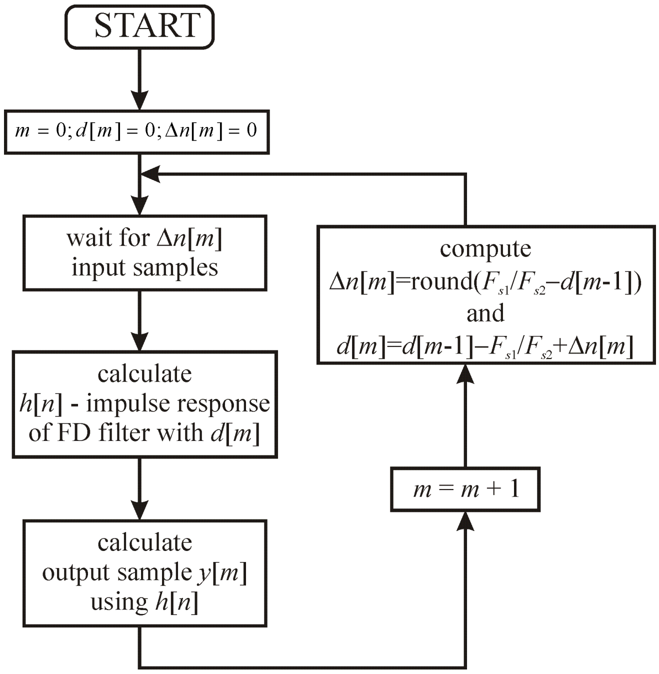

Fig. 3. Diagram of VFD based SRC algorithm.

Because of the causality requirement, FD filters are usually characterized with a nonzero integer delay $D = \mathrm{round}(\tau_d)$, which for FIR filters is commonly selected close to the bulk delay $\tau_N = (N-1)/2$. With those two delays defined, we receive the following formula for the total delay

$$\tau_d = D + d = \tau_N + \varepsilon \tag{8}$$

where $d \in [-0.5, 0.5)$ is the fractional delay and ε is the net delay.

The performance of the FD filter is usually evaluated using the frequency domain error function [9]

$$E(f) = H_N(f) - H_{\mathrm{id}}(f) \tag{9}$$

but it is not sufficient to know just the errors of FD filters to assess the performance of the SRC algorithm based on FD filters. Relations between all FD filters used in the resampling are also important. These relations can be readily taken into account if we observe that the SRC algorithm based on FD filters (Fig. 1.3) is equivalent to the classic approach (Fig. 1) [7]. We only need to replace the interpolation filter with the overall filter, obtained by polyphase composition of FD filters used in resampling. This can be done only for rational resampling rates but the conclusions resulting from the overall filter can be readily adapted to arbitrary resampling ratios.

The composition of the impulse response of the overall filter is defined by the following formula [7]

$$h_o[m + nL] = h_{d[m]}[n]; \quad m = 0, 1, \ldots, L-1 \tag{10}$$

where $h_{d[m]}[n]$ is the impulse response of the FD filter with fractional delay $d[m]$. In order to obtain a proper overall filter, delays $d[m]$ need to be organized in the decreasing order

$$d[m-1] = d[m] + 1/L; \quad m = 1, \ldots, L-1 \tag{11}$$

Using the overall filter we can readily analyze distortions introduced by SRC algorithm based on FD filters since this filter should fulfill the same design requirements as the interpolation filter in the classic approach (Fig. 1).

It is worth noting that SRC algorithms with different decimation factors M, but with the same interpolation factor L, operate on the same set of fractional delays. Therefore, since the same set of FD filters is used, the overall filter also stays the same. Nonetheless, we must remember that when the output sample rate is smaller than the input sample rate ($M > L$) the cutoff frequency f_c of the interpolation filter should be lower

$$f_c = \min(0.5/L, 0.5/M) \tag{12}$$

which must be taken into account in the SRC algorithm design.

III. FD Filter Design for SRC

There are numerous approaches to FD filter design [9]. Typically, the FD filter design is discussed in the literature as a problem of approximation of the ideal FD filter. If the filter has to be used for the SRC such an approach is not always satisfactory. In this paper we propose to organize FD filter design methods into three general categories: the optimal FD filter design, the offset window method and the polyphase decomposition. As we will present further in this paper, SRC algorithms based FD filters belonging to each of those categories demonstrate different properties. In this paper only FIR FD filters are considered since the design and analysis of the SRC algorithm based on such filters is simpler than in case of IIR filters.

A. *Optimal FD filters*

In the optimal FD filter design an error dependent on given criteria based on complex approximation error (9) is minimized. The most commonly used criteria are maximal flatness of error frequency response (MF) and minimization of least square error (LS) or maximum magnitude of approximation error (minimax) in a given approximation band defined by its upper frequency f_a [9].

The design of optimal FD filters is quite complex since it involves solving matrix equation [9], in a case of recursive algorithms even several times [9], [10].

Optimal FD filters might seem to be the best choice for the SRC since they offer the best possible approximation of the ideal FD filter for particular filter length and approximation band. Nevertheless, each filter is optimized separately, which means that relations between all the filters used in the resampling process are neglected. In the result magnitude response of the overall filter obtained from optimal FD filters (Fig. 4) exhibits large lobes in the stopband, at the frequencies corresponding to the components of the input signal located above f_a and images of those components (Fig. 5) [7].

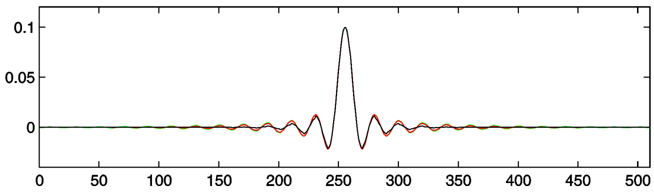

Fig. 4. Examples of impulse responses of overall filters. For FD filters of the length $N = 51$, black - maximally flat, green - minimax with $f_a = 0.45$ and red - LS with f_a = 0.475. Interpolation factor $L = 10$. Frequency normalized by the input sample rate.

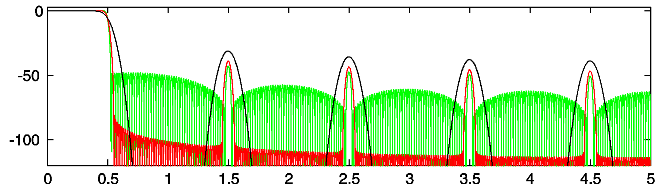

Fig. 5. Magnitude responses of overall filters from Fig. 4

Thus, the SRC algorithm implemented using optimal filters performs correctly only for signals with band limited to the approximation band of FD filters used in the resampling. For fullband signals the additional filter preceding the actual resampling, limiting the band of the input signal, is required. Such a prefilter increases computational cost of the resampling but with multistage implementation [1], [11] which improves computational efficiency of the resampling, the prefiltering can be readily incorporated into the interpolation filter of the first resampling stage.

The additional advantage of the optimal FD filter approach is that when such a filter is implemented using the extracted window concept (Fig. 6) [12]–[14], parameters of the resampling algorithm can be readily modified during runtime since only a symmetric window $w_{ref}[n]$ and coefficients of a low order polynomial used for computation of gain correction $\alpha(d)$ need to be replaced. This makes such an approach well suited for SRC algorithms prototyping, when we need to verify which filter type or approximation band width should be selected in the final implementation or when a versatile application, which leaves the decision on selection of the filter type and its specification to the user, is needed.

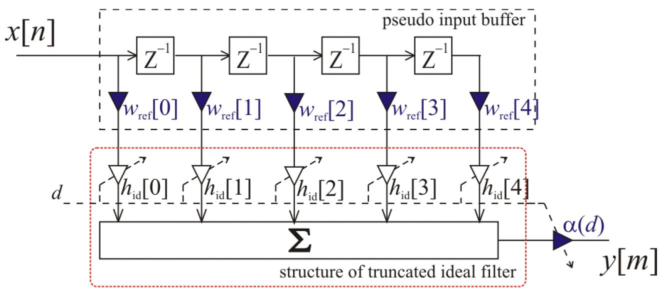

Fig. 6. Optimal FD filter structure based on the extracted window method

B. FD filter design using offset window

The second approach to the FD filter design is based on the window method with window which is offset accordingly to the fractional delay of the designed filter [15]–[18]. The filter design formula is simple

$$h_d[n] = w_d[n]h_{\text{id}}[n] \tag{13}$$

where

$$w_d[n] = w((n-d)T_{s1}) \tag{14}$$

is the offset window which is a sequence of samples of prototype continuous time window $w(t)$ sampled with sampling period T_{s1} in instants delayed by d.

Let us notice that the impulse response of the overall filter composed of truncated impulse responses of the ideal FD filter (6) is the truncated ideal response of the $1/L$-th band interpolation filter

$$h_{\text{LPF,id}}[n] = \text{sinc}(2f_c n) \tag{15}$$

where cutoff frequency $f_c = 0.5/L$ and the gain of the filter is equal to L.

In the same way as the overall filter is created, the overall window can be composed of windows used to design each FD filter. For the window offsetting method, the overall window is simply the L times interpolated window of the same type as the protopype window used to design FD filters. Thus, using FD filters designed with offset window we actually design overall filter using window method while designing only a fraction of the whole filter with each FD filter. Therefore, although the performance of FD filter designed using offset window method is worse than the performance of optimal filters, the overall filter performs better in the stopband (Fig. 7). The magnitude response of the overall window does not exhibit large lobes in stopband which are typical to the overall window extracted from overall filter of the SRC algorithm based on optimal FD filters (Fig. 8). In consequence the overall interpolation filter also does not exhibit large lobes (Fig. 9) [18] in stopband, which means that the SRC algorithm based on FDs filter designed with this method can be used in the resampling of fullband signals without need for a prefilter.

Additionally, using this approach we can readily manipulate the location of the transition band of the interpolation filter. The impulse response of the fullband FD filter (6) needs only

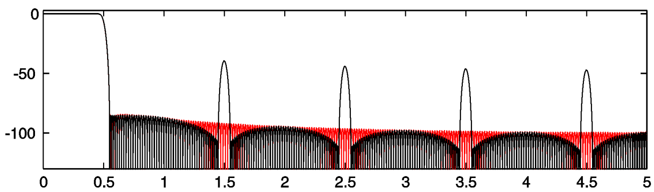

Fig. 7. Magnitude responses of overall filters for optimal FD minimax filters - black, and filters designed using offset window method with window extracted from optimal FD minimax - red. $f_a = 0.45$, $N = 51$ and $L = 10$.

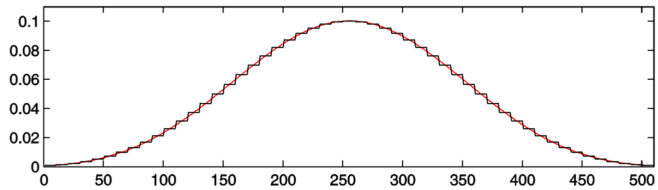

Fig. 8. Overall windows corresponding to filters presented in Fig.7.

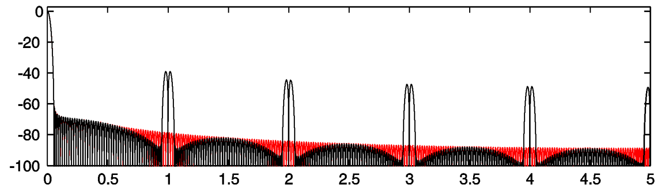

Fig. 9. Magnitude responses of windows for Fig.8.

be replaced with ideal impulse response of the bandlimited FD filter

$$h_{\text{id}}[n] = \text{sinc}(Lf_c(n-\tau_d)) \tag{16}$$

where f_c is the assumed cutoff frequency of the ideal overall filter (15).

The concept of the design based on offset window is simple when a prototype window $w(t)$ is given as a continuous function of time, like in case of raised cosine windows [19]. The problem with raised cosine windows is, that even with optimized coefficients, filter designed using such a window is worse than the optimal filter by few dB. On the other hand, offsetting other types of windows is more problematic. For example, the Kaiser window [20] is defined as a continuous function of time variable but coefficients recalculation for different delays is numerically too demanding for most real time applications of the SRC. If the window is not defined in the time domain as a continuous function, like the Chebyshev window [20] which is defined in the frequency domain, then in addition to coefficients calculation formula being too complex, the window offsetting procedure is not straightforward.

Offsetting procedures for such windows are based on the fact that window offsetting can be interpreted as delaying a discrete symmetric prototype window by a fraction of the sampling period. Thus, when a prototype window is not a continuous function, it might be resampled using a delay operator implemented either in the frequency domain [21], [22] or in the time domain, e.g. using short FD filter [18].

If we want to decrease computational costs of window offsetting even more, after computing several offset windows for different delays, the discrete window prototype can be approximated with piecewise polynomial. Precise window offsetting can be then performed in time domain with separate polynomials approximating window segments corresponding to each sample of impulse response of the designed FD. Usually second or third order polynomials offer a sufficient performance [23].

C. Polyphase subfilters

The last category of FD filter design methods suited for the needs of the SRC is directly related to the classic approach. Let us notice that a polyphase decomposition of $1/L$-th band optimal interpolation filter [3], designed for example using the Parks-McClellan algorithm, into L subfilters

$$h_{d[m]}[n] = h_{LPF}[m + nL]; \quad m = 0, 1, \ldots, L-1 \quad (17)$$

gives us L fullband FD filters, each with different delay $d[m]$. With those filters stored in memory we can implement the SRC algorithm with interpolation factor L. The problem is that for large L we need to design and store very long impulse response of the interpolation filter, in some cases even longer than several thousands samples. Such optimal filter design might not be possible due to accumulation of numerical errors during design. Additionally, a particular interpolation filter can only be used for the resampling with the given factor L. On the other hand, for a given length of the overall filter we gain a possibility to improve the attenuation in stopband of the overall filter in the exchange for increased ripples in passband (Fig. 10). This is a significant advantage. For example with stopband attenuation equal to 86 dB (Fig. 10) with two previous approaches to the FD filter design, when no prefilter is used, passband ripples of the overall filter are about 10^{-3} dB. In most application we don't need such a high precision in passband. Relaxing the specification in passband and allowing ripples equal to 0.02 dB, improves stopband attenuation by 30 dB (Fig. 10) for the same filter length.

This approach might seem inappropriate for incommensurate or variable resampling ratios, since we get only L FD filters, but we actually don't need to directly design the interpolation filter for the required ratio. We only need to design the prototype interpolation filter for some low integer interpolation factor, for example $L = 10$ (Fig. 10). Subsequently, using the Farrow structure [11], [24]–[26] (Fig. 11) we can approximate the FD filter and obtain the impulse response for any required fractional delay.

The idea behind the Farrow structure is that the overall filter impulse response is approximated with a low order piecewise polynomial with each segment

$$h[n] = \sum_{m=0}^{p} c_m[n] d^m \quad (18)$$

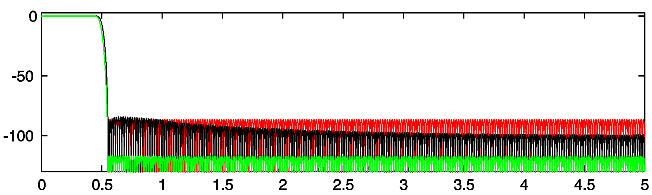

Fig. 10. Magnitude responses of overall filters combined from FD filters designed using offset window (from Fig. 7) - black and minimax interpolation filters with the same transition band and impulse response length designed with similar passband ripple level like previous one - red and with larger passband ripple level - green.

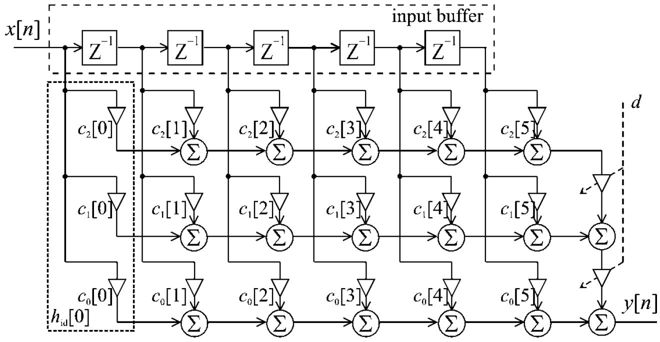

Fig. 11. Farrow structure of order $p = 2$ approximating the FD filter of the length $N = 6$. Thick dashed box indicates structure coefficients $c_m[0]$ of the polynomial approximating the first sample of the impulse response $h_{\mathrm{id}}[0]$.

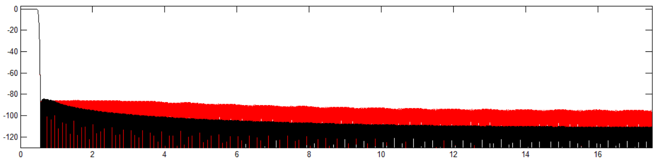

Fig. 12. Magnitude responses of overall filters obtained using the Farrow structure for interpolation factor $L = 35$. Farrow structure coefficients computed based on two first filters presented in Fig. 10. The length of the impulse response of FD filters $N = 51$.

approximating separate sample of the FD filter impulse response.

Thus using the Farrow structure we can keep the advantages of the SRC algorithm specification flexibility resulting from of the direct design of the interpolation filter and at the same time implement any arbitrary resampling ratio (Fig. 12). That way we can freely select cutoff frequency and adjust passband ripple level while improving the stopband attenuation. However, for each different specification we need to replace all coefficients of the Farrow structure.

It is worth noting that although the Farrow structure can be used to implement the VFD filter belonging to any category described in this paper, the structure is the most beneficial when used with polyphase filters obtained from the lowpass prototype of the interpolation filter.

IV. CONCLUSIONS

In this paper properties of the SRC algorithm based on FD filters have been presented. The dissimilarities of different methods of the FD filter design have been analyzed using the overall filter or the overall window. Based on the observed properties the classification of FD filter design methods into three categories have been proposed. To the first category belong the optimal FD filter design methods which offer the smallest possible approximation error for a given filter length and approximation band width. For this category the overall filter of the SRC algorithm exhibits large lobes in the stopband. This means that either the input signal must be bandlimited or a prefilter needs to be used before resampling.

Since FD filters belonging to this group are closely related to the symmetric window design method, such an adjustable FD filter can be implemented using extracted window method. This implementation allows for simple change window type or width of the approximation band which might be useful in some applications.

If the properties of the overall filter in the stopband have to be improved, the offset window method should be used to design FD filters instead of the optimal filters. This results in elimination of large lobes in stopband and additionally the cutoff frequency of the overall filter can be readily changed. A design of FD filters using the offset window method when compared with design of optimal filters with symmetric extracted window is more numerically complex but the SRC algorithm based on the offset window approach does not require additional prefilter since large lobes in stopband of the overall filter are suppressed.

The last category includes FD filters designed by means of polyphase decomposition of the interpolation filter. This approach offers the best performance since we can directly optimize the interpolation filter. In this approach using polyphase decomposition we obtain only few FD filters from the whole family but the Farrow structure can be used to obtain FD filters with any required fractional delay. This makes an implementation of the arbitrary resampling ratio possible using FD filters from this group. Unlike with previous categories, polyphase filters and Farrow structure coefficients need to be redesigned each time we want to change specification of the SRC algorithm. This is the price we have to pay for the flexibility of the overall filter specification.

ACKNOWLEDGMENT

This work was supported by the Polish Ministry of Science and Higher Education under the research project financed from the state budget designated for science in the years 2010-2012.

REFERENCES

[1] K. Rajamani, L. Yhean-Sen, and C. W. Farrow, "An efficient algorithm for sample rate conversion from CD to DAT," *IEEE Signal Process. Lett.*, vol. 7, no. 10, pp. 288–290, 2000.

[2] S. Mitra and J. Kaiser, *Handbook for Digital Signal Processing*. John Wiley & Sons, 1993.

[3] F. Harris, *Multirate Signal Processing for Communication Systems*. Prentice Hall, 2004.

[4] E. Hermanowicz, M. Rojewski, and M. Blok, "A sample rate converter based on a fractional delay filter bank," in *Proc. ICSPAT 2000*, Dallas, Tx, USA, Oct. 16–19, 2000.

[5] G. Evangelista, "Design of digital systems for arbitrary sampling rate conversion," *Signal Process.*, vol. 83, no. 2, pp. 377–387, Feb. 2003.

[6] A. Tarczynski, W. Kozinski, and G. D. Cain, "Sampling rate conversion using fractional-sample delay," in *Proc. IEEE ICASSP'94*, Adelaide, Australia, May 1994, pp. 285–288.

[7] M. Blok, "Collective filter evaluation of an FSD filter-based resampling algorithm," in *OSEE 2002*, TechOnLine, Bedford, Massachusetts, USA, Jan. 15, 2002. [Online]. Available: www.eetimes.com/design/signal-processing-dsp/4017905

[8] P. Delegacz, "Incommensurate sampling rate conversion," M. Eng. thesis, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland, 2011, in Polish.

[9] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay — tools for fractional delay filter design," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, 1996.

[10] M. Blok, "Optimal fractional sample delay filter with variable delay," in *OSEE 2002*, TechOnLine, Bedford, Massachusetts, USA, Mar. 18, 2002. [Online]. Available: http://www.eetimes.com/design/analog-design/4018005

[11] E. Hermanowicz, "On designing a wideband fractional delay filter using the Farrow approach," in *Proc. EUSIPCO'2004*, Austria, Sep. 6–10, 2004, pp. 961–964.

[12] ——, "A nearly optimal variable fractional delay filter with extracted Chebyshev window," in *Proc. IEEE ICECS'98*, vol. 2, Lisboa, Portugal, Sep. 7–10, 1998, pp. 401–404.

[13] M. Blok, "On practical aspects of optimal FSD filter design using extracted window method," in *Proc. ECCTD'2011*, Linkoping, Sweden, Aug. 29–31, 2011, pp. 330–333.

[14] ——, "Versatile structure for variable fractional delay filter based on extracted window method," in *Proc. PWT'2011*, Poznań, Poland, Dec. 9, 2011, pp. 64–67.

[15] G. D. Cain, A. Yardim, and P. Henry, "Offset windowing for FIR fractional-sample delay," in *Proc. IEEE ICASSP'95*, May 1995, pp. 1276–1279.

[16] ——, "Optimal two-term offset windowing for fractional delay," *Electron. Lett.*, vol. 32, no. 6, pp. 526–527, Mar. 1996.

[17] A. Yardim, G. D. Cain, and A. Lavergne, "Performance of fractional-delay filters using optimal offset windows," in *Proc. IEEE ICASSP'97*, vol. 3, Apr. 21–24, 1997, pp. 2233–2236.

[18] M. Blok, "Fractional delay filter design with extracted window offsetting," in *Proc. MixDes'2012*, Warsaw, Poland, May 24–26, 2012.

[19] H. Albrecht, "A family of cosine-sum windows for high-resolution measurements," in *Proc. ICASSP'01*, vol. 5, 2001, pp. 3081–3084.

[20] S. K. Mitra and J. F. Kaiser, Eds., *Handbook of Digital Signal Processing*. Wiley & Sons, 1993, ch. 4, Finite Impulse Response Filter Design by T. Saramäki.

[21] T. I. Laakso, T. Saramäki, and G. D. Cain, "Asymmetric Dolph-Chebyshev, Saramäki, and transitional windows for fractional delay FIR filter design," in *Proc. MWSCAS'95*, Aug. 1995, pp. 580 – 583.

[22] S.-C. Pei and Y. Lai, "Closed form variable fractional time delay using FFT," *IEEE Signal Process. Lett.*, vol. 19, no. 5, pp. 299–302, May 2012.

[23] M. Blok, "Sample rate conversion with fractional delay filters designed using windows offset by means of polynomial interpolation," *Telecommunications Review and Telecommunications News (Przeglad Telekomunikacyjny i Wiadomosci Telekomunikacyjne)*, vol. 4, 2012, in Polish.

[24] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE ISCAS'88*, Espoo, Finland, 1988, pp. 2641–2645.

[25] M. Blok, "Farrow structure implementation of fractional delay filter optimal in Chebyshev sense," in *Proc. SPIE*, vol. 6159, Wilga, Poland, May 30 – Jun. 2, 2005, p. 61594K.

[26] F. J. Harris, "Performance and design of Farrow filter used for arbitrary resampling," in *Proc. DSP'97*, vol. 2, Santorini, Greece, 1997, pp. 595–599.