



HAL
open science

Comprehensive Survey of FIR-Based Sample Rate Conversion

Ali Zeineddine, Amor Nafkha, Stéphane Paquelet, Christophe Moy, Pierre Yves Jezequel

► **To cite this version:**

Ali Zeineddine, Amor Nafkha, Stéphane Paquelet, Christophe Moy, Pierre Yves Jezequel. Comprehensive Survey of FIR-Based Sample Rate Conversion. *Journal of Signal Processing Systems*, 2021, 93 (1), pp.113-125. 10.1007/s11265-020-01575-6 . hal-02912947

HAL Id: hal-02912947

<https://hal.science/hal-02912947v1>

Submitted on 7 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

REVIEW

Comprehensive Survey of FIR-based Sample Rate Conversion

Ali Zeineddine¹, Amor Nafkha^{2*}, Stéphane Paquelet³, Christophe Moy⁴ and Pierre Yves Jezequel¹

*Correspondence:

amor.nafkha@centralesupelec.fr

²IETR UMR CNRS 6164, SCEE,
CentraleSupélec, Avenue de la
Boulaie, 35510 Cesson Sévigné,
France

Full list of author information is
available at the end of the article

Abstract

Sample rate conversion (SRC) is ubiquitous and critical function of software defined radio and other signal processing systems (speech coding and synthesis, computer simulation of continuous-time systems, *etc.*). In this paper, we present a survey on linear phase finite impulse response (FIR) based sampling rate conversion. Many different FIR-based SRC solutions exist, such as classical FIR, polyphase, Farrow, cascaded-integrator-comb, and Newton structures. Each one of these solutions is presented differently in the literature, and SRC reference books introducing the subject are often missing hardware implementation aspects. The main objective of this paper is to provide a simple and comprehensive overview of main FIR-based SRC techniques from theoretical to hardware implementation aspects. The state of the art of FIR-based SRC filters is summed-up through a concise derivation of the different solutions from a common root: linear phase FIR filters. Each SRC solution is presented from both theoretical and practical implementation points of view. The paper provides a succinct tutorial that introduces SRC, and helps identifying and implementing the appropriate FIR-based SRC architecture for any given applications.

Keywords: Sample rate conversion; FIR filter; Polyphase filter; Farrow structure; Newton structure; CIC filter

Introduction

Flexibility and reconfiguration are essential requirements in modern transceivers. To implement these features at the front-end interface, sample rate conversion (SRC) is necessary [1]. However, each transceiver is designed for certain application requirements, that may range from high performance and low latency, down to low cost and low power consumption applications. In each case, the needed SRC implementation differs to comply with the system requirements. This does not apply to transceivers only, but also a wide range of signal processing applications are based on SRC, most notably in the audio processing domain. The importance of SRC made it a ubiquitous signal processing function used in many domains for over four decades [2]. Being a classical topic, the literature of SRC is very huge, has been addressed in a large number of reference books, and many solutions have been developed. However, extracting the needed information from the literature can be complicated as each solution is presented from its own point of view. And when referring to books about the subject, practical implementation aspects are often overlooked.

This article provides a concise tutorial presenting the different available SRC solutions, mainly for engineers in this field, looking for an efficient SRC implementation. This work focuses on finite impulse response (FIR) SRC filters due to their linear

phase, that is required in many applications which do not tolerate distortion of the signal's phase. Understanding these solutions facilitates the exploration of other SRC types, and also building new ones.

In the FIR-based SRC literature, five main filter structures can be identified: classical up-sampling/filtering/down-sampling (U-F-D) [2], polyphase [3], Farrow [4], cascaded-integrator-comb (CIC) [5], and Newton structures [6]. This tutorial emphasizes the unified representation shown in Figure 2. This vision derives the different SRC solutions from a common root, which helps in intuitively understanding the features and limitations of each solution, and its relation to the other structures. This tutorial also provides an up to date coverage of SRC solutions by including the modifications of the generalized Newton structure developed in the last four years [7, 8, 9, 10]. Last but not least, the practical implementation aspects are also presented for each SRC solution, providing thereby a guide to select and implement the best adapted SRC solutions for a given desired application.

This article is organized as follows. Section 1 presents the basic SRC concepts, and summarizes the unified vision approach. Then each of the following sections studies one of the main SRC solutions by deriving its structure through the introduced unified vision, and then by discussing its practical implementation aspects. Sections 2, 3, 4, and 5 present the U-F-D/Polyphase, Farrow, CIC, and Newton structures respectively. The generalization of the Newton structure is studied in Section 6. Then, Section 7 compares the different presented solutions, and shows how the proposed unified vision is used to study and select the appropriate sample rate conversion structures for a given application. Finally, Section 8 concludes this tutorial.

1 Linear Phase FIR SRC Filters

Let $x(t)$ be a limited-band continuous time signal with its highest frequency component designated as f_m . The sampling rate conversion (SRC) operation consists of changing the sampling frequency of the signal $x(t)$ without going through reconstruction and re-sampling. The definition of SRC operation and its relation to sampling and reconstruction is shown in Figure 1. Sampling $x(t)$ at a frequency $F_{in} = 1/T_{in}$ and $F_{out} = 1/T_{out}$ results in two series of samples $x(mT_{in})$ and $x(kT_{out})$, respectively. In order to correctly sample a signal, the Nyquist theorem states that F_{in} and F_{out} should be greater than or equal to $2 \times f_m$ [11]. Sampling rate conversion is then defined as the operation that takes the samples $x(mT_{in})$ as input, and outputs $x(kT_{out})$, with the change in the sampling frequency defined by the SRC factor R :

$$R = \frac{F_{out}}{F_{in}} = \frac{U}{D} \quad \text{such that} \quad U, D \in \mathbb{N}. \quad (1)$$

The SRC operation with $R > 1$ is called interpolation, while decimation is used to refer to the case of $R < 1$. The supposition that U and D are integers results from the fact that the SRC operation will be implemented in a digital system, that is quantified and can only handle finite precision. In the SRC jargon, the term “*coarse*” SRC is used to refer to a large sampling rate conversion factor that can be a large integer or simple rational (*e.g.* $R = 1/128$, $R = 64/10$), while the term

“*fine*” SRC is used to refer to a very small modification of the sampling factor and it concerns a fine-tuning factor close to unity (*e.g.* $R = 160/159$).

To implement an SRC operation of a factor $R = U/D$ using discrete time signal processing, the most straightforward solution is to increase the sampling frequency by U through inserting $U - 1$ zeros between input samples, followed by a decrease of factor D by keeping one sample from each D samples. Low-pass filtering is required between these two steps to find the values of the added zero samples (i.e. Figure 4) and potentially to remove any frequency components that may cause aliasing [2]. This work addresses the FIR based SRC methods due to their multiple advantages: stability, linear-phase, and implementation efficiency.

The frequency response of the SRC filter is either optimized to reject up-sampling images or to protect from aliasing. The filter is then called an anti-imaging (AI) or an anti-aliasing (AA) filter, respectively. In practice, the AI filter is used for interpolation, while the AA filter is used for decimation. However, nothing prevents the two types of filter to be used for either kind of SRC factors if the filtering response is acceptable. The AI and AA filters used for SRC are duals. Duality is the property where one operation performs the complementary of the other, such as modulation and de-modulation. The dual of an SRC system is found using the transposition theorem developed in [12], and also presented in chapter three of [2]. The unified vision in this article is developed for the AI filters version, and their corresponding AA duals can be found using the transposition theorem.

All of the FIR based SRC filters can be related to at least one of the five main FIR based SRC structures: the U-F-D, polyphase, Farrow, CIC, and Newton structures. This article offers a unified derivation, starting from one common root, in order to derive these structures and to explain clearly their respective correspondences and differences as shown in Figure 2.

The linear phase FIR filter type is first considered, then more particular filter types are studied, and by exploiting the characteristics of the impulse response of these filters, the different implementation structures are derived. The linear phase FIR filters basic structure is the U-F-D or the polyphase structure. Then particular cases of linear phase FIR filters are considered, which are: polynomial based, B-spline, and Lagrange interpolation impulse responses. By exploiting the particular properties of each response, the Farrow, CIC, and Newton structure are found respectively, and reveal how to generalize the Newton structure in order to implement any polynomial based filter.

2 Fundamental SRC: U-F-D & Polyphase Structures

The U-F-D structure is a direct implementation of the theoretical SRC model presented in last section. It is theoretically capable of implementing any SRC operation using any linear phase FIR filtering response. However, it does not always create a practical efficient implementation [2, 13]. The polyphase structure improves the efficiency while keeping the flexibility of the U-F-D structure [3, 14, 15]. The derivation of these two structures are obtained as presented below.

2.1 Filter Structure Derivation

In the case of the U-F-D structure, the impulse response is directly implemented using an FIR filter structure as shown in the top of Figure 3. Since the input of the

filter is the output of the up-sampling of $x[m]$, then we know that for each U input samples to the filter, only one is non-zero. An example of $U = 4$ is shown in Figure 4, where for each output only a specific set of coefficients are used (squares, circles, triangles, or stars). Therefore, if a filtering operation of order $N - 1$ is required, the number of taps of the FIR filter has to be equal to $N \times U$. The transfer function of this filter is written as:

$$H(z) = \mathcal{Z}\{h[n]\} = \sum_{n=-\infty}^{+\infty} h[n]z^{-n} = \sum_{n=0}^{NU-1} h[n]z^{-n}, \quad (2)$$

with $h[n]$ being the impulse response coefficients. When implementing a rational SRC factor with $D > 1$, a number of the filter output samples is dropped by the following down-sampling operation, meaning that the filter is inefficient since it calculates unneeded samples. This is addressed by the polyphase structure. The classical polyphase structure is deduced from the U-F-D structure by rearranging the filter coefficients, through re-writing $H(z)$ in (2) as:

$$H(z) = \sum_{i=0}^{U-1} z^{-i} \sum_{j=0}^{N-1} h[i + jU]z^{-jU} = \sum_{i=0}^{U-1} z^{-i} H_i(z^U), \quad (3)$$

where the FIR filter is split into U filters $H_i(z)$ of order N in parallel, with each filter containing the coefficients used to find a certain output. The matrix representation of $H(z)$ offers a clear visual description of the filter structure:

$$H(z) = \begin{matrix} U \times 1 \\ \left[\begin{array}{cccc} \star_1 & \star_2 & \dots & \star_N \\ \blacktriangle_1 & \blacktriangle_2 & \dots & \blacktriangle_N \\ \vdots & \vdots & \ddots & \vdots \\ \blacksquare_1 & \blacksquare_2 & \dots & \blacksquare_N \end{array} \right] \end{matrix} \times \begin{matrix} \left[\begin{array}{c} 1 \\ z^{-1} \\ \vdots \\ z^{-N+1} \end{array} \right] \end{matrix}, \quad (4)$$

with each row i of the matrix H containing the coefficients of the sub-filter $H_i(z)$. The polyphase structure is then built using U sub-filters $H_i(z)$ of order N in parallel. For each output $y[k]$, the corresponding sub-filter $H_{i_k}(z)$ is selected by a multiplexer combined with a controller operating at the output sampling frequency F_{out} . This results in the polyphase structure shown in Figure 3, that operates efficiently using the sampling frequencies F_{in} and F_{out} only, without needing the higher UF_{in} sampling domain [16]. However for fine-tuned SRC factors, when the values of U and D are large, the structure still requires having U filters in parallel, which is problematic concerning the structure efficiency. This can be solved using polynomial based filters that are presented in the next section.

2.2 Practical Implementation

To implement the derived filter structures, a number of practical implementation aspects need to be addressed. In the case of the U-F-D and polyphase structure, these aspects are the following.

Hardware Implementation: For a given filter structure, many hardware implementation strategies exist, with each favoring a certain implementation constraint. Pipeline implementations favor high operation speed at the cost of added complexity [17], while a lower consumption at the cost of reduced speed can be achieved through reusing hardware elements [18]. The second design aspect concerns the quantization of the implementation, that defines the calculation precision. A wide range of quantization methods for both the filter parameters [19, 20, 21] and signals [22, 23, 24] exist in the literature.

SRC Controller: It is often supposed in the literature that for the SRC implementation, F_{in} and F_{out} are actually available digital clocks. However, practical systems often operate using only one digital clock higher than both sampling frequencies. The multi-rate operation is then made possible through an SRC controller that controls the samples flow. This is required because the number of inputs to the SRC filter is not equal to the number of outputs. In the case of the polyphase structure, for a given output instant k , two indexes are needed to control the filter state. The first being m_k that indicates the input samples used by the filters $H_i(z)$, and the second being i_k that specifies the branch of the polyphase filter that calculates the output $y[k]$. These two indexes are managed by the SRC controller in order to implement the desired SRC operation defined by U and D . The control algorithm that finds these indexes is shown in Algorithm 1, that describes the AI SRC controller operation.

Algorithm 1 Algorithm of the SRC Controller

Input: U, D
Output: m_k, i_k, k
1: $m_k = 0, i_k = 0, k = 0$
2: **while** True **do**
3: $i_k = i_k + D$
4: **while** $i_k \geq U$ **do**
5: $m_k ++$ {Get next input}
6: $i_k = i_k - U$
7: **end while**
8: $k ++$ {Find next output}
9: **end while**

Half-Band filters: When performing SRC of large factors, it is possible to improve the implementation efficiency by breaking the SRC operation into multiple stages as proposed in [13]. This is also more advantageous when it is possible to break the SRC factor into factors of two, that can be implemented efficiently using half-band filters. These filters are a particular case of the U-F-D structure, with a very efficient implementation that uses only 25% of the multipliers [2].

3 Efficient Arbitrary SRC: Farrow Structure

An efficient solution for finely-tuned SRC implementation is the structure developed by C. W. Farrow in [4]. This structure implements polynomial based filters, and is known in the literature under many names, most notably as the Farrow structure [25, 26]. The introduction of this structure enabled implementing high precision finely-tuned SRC far more efficiently than it was possible with the previously existing structures. A modification of the structure was developed in [27] that

exploits the symmetry of the impulse response in order to reduce the complexity. The transposed Farrow structure preferably used for SRC decimation was addressed on multiple occasions in [28, 29, 30, 31]. The transposed Farrow structures has many other applications beside SRC as discussed in [28, 32, 33].

3.1 Filter Structure Derivation

It was shown in the last section that for each output $y[k]$, only one polyphase branch is selected by the multiplexer. Therefore it would be interesting that instead of having U filters in parallel, to have only one filter with its coefficients being calculated dynamically for each output using the index i_k . This can be made possible by defining the impulse response as being constructed using N polynomial pieces $P_j(\mu)$ with $j \in \{1, 2, \dots, N\}$. Each polynomial piece is of order $M - 1$ and is defined as:

$$P_j(\mu) = \sum_{l=1}^M c_l(j) \mu^{l-1}, \quad (5)$$

then P_j is used to calculate the filter's coefficient $h_{i_k}[j]$ corresponding to the output $y[k]$. It is then possible to rewrite the $H_{i_k}(z)$ filter expression in (3) as:

$$H_{i_k}(z) = \sum_{j=1}^N \left[\sum_{l=1}^M c_l(j) \mu(i_k)^{l-1} \right] z^{-j+1}, \quad (6)$$

where $\mu(i_k)$ is used to indicate the position of the coefficients of the filter i_k on the different polynomial pieces. To exploit the symmetry of the impulse response of a linear phase filter [34], the variable $\mu(i_k)$ used to develop the polynomials is defined as:

$$\mu(i_k) = \frac{i_k}{U} - \frac{1}{2} \in [-0.5; 0.5[. \quad (7)$$

Using this type of impulse response, the coefficients of each filter $H_{i_k}(z)$ in (3) are found as:

$$\begin{aligned} H_{i_k}(z) &= \underset{U \times 1}{\boldsymbol{\mu}_{i_k}} \times \underset{1 \times M}{\boldsymbol{P}} \times \underset{M \times N}{\boldsymbol{P}} \times \underset{N \times 1}{\boldsymbol{Z}(z)} \\ &= \begin{bmatrix} 1 \\ \mu_{i_k} \\ \vdots \\ \mu_{i_k}^{M-1} \end{bmatrix}^T \begin{bmatrix} c_1(1) & \dots & c_N(1) \\ c_1(2) & \dots & c_N(2) \\ \vdots & \ddots & \vdots \\ c_1(M) & \dots & c_N(M) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-N+1} \end{bmatrix}. \end{aligned} \quad (8)$$

Through a factorization by μ in (6), the Farrow structure definition is found as:

$$H_{i_k}(z) = \sum_{l=1}^M G_l(z) \mu(i_k)^{l-1} \quad \text{with} \quad G_l(z) = \sum_{j=1}^N c_l(j) z^{-j+1}. \quad (9)$$

The implementation of this structure is shown in Figure 5, which consists of M FIR filters $G_l(z)$, each of N taps. The outputs of the G_l filters are then evaluated

following the Horner's scheme for polynomial evaluation with $\mu(i_k)$. In this structure, the coefficients of the filters G_l are constant and symmetrical, allowing an efficient implementation by using only half of the multipliers.

3.2 Practical Implementation

The same practical implementation aspects presented in the last section apply the Farrow structure as well. Two other specific implementation aspects to the Farrow structure are: the polynomial coefficients and the fractional delay quantization.

Polynomial Coefficients: The Farrow structure has $M \times N$ degrees of freedom to define the filter response, i.e. the coefficients $c_l(j)$. When the linear-phase filter response is used, the degrees of freedom are divided by two, due to the symmetry of the response [34]. These coefficients can be found generally in two ways: using well known polynomial interpolation methods, or applying filter optimization techniques.

Three types of polynomial interpolation methods are widespread in the literature [35]. Due to their particular mathematical definition, these interpolation types can be implemented more efficiently using other structures than the Farrow structure as it will be developed later in this article. These are B-spline interpolation [36, 37, 38, 39], Lagrange interpolation [25, 40, 41], and Hermite interpolation [42, 43].

However, to correctly profit from the Farrow structure filtering performance, the polynomial coefficients should be defined using filter optimization methods. In the literature, different techniques are employed [4, 44, 45, 46]. The thesis of M. T. Hunter [47] develops in details the Farrow structure optimization problem, and provides a software implementation of the developed optimization algorithm.

Fractional Delay Quantization: Quantization of the fractional delay $\mu(i_k)$ affects two implementation aspects: the output sampling rate precision and the filtering performance. This work uses the approach developed in [47][48][49].

4 Low-cost Coarse SRC: CIC Filter

In 1981, E. B. Hogenauer developed the cascaded-integrator-comb (CIC) filters [5], illustrated in Figure 6, which are a class of SRC FIR filters with an implementation that does not require multipliers. These filters are mainly used to implement coarse SRC operation of a large integer SRC factor.

4.1 Filter Structure Derivation

The Farrow structure previously presented offers an SRC solution with a high degree of freedom permitting the implementation of different filter responses. Considering the most simple case of a polynomial filter, where the impulse response consists of only one polynomial piece ($N = 1$) of degree zero ($M = 1$), having a value of one. This corresponds to the Pi or rectangular function normalized to the input sampling period T_{in} noted as $\Pi(t/T_{in})$. Used for interpolation, the Pi function results in a zero-order-hold reconstruction. An interesting property of the Pi function is its frequency response $H_{\Pi}(f) = \text{sinc}(f/F_{in})$, with its zeros located at multiples of F_{in} , which is the optimal zero position for images suppression, since the up-sampling images are located on multiples of F_{in} [2]. The filtering performance of a single Pi

function may not be sufficient, however it can be improved by cascading N filters. This results in the impulse response of the B-spline interpolation of order $N - 1$:

$$\beta^N(t) = \underbrace{(\Pi * \Pi * \dots * \Pi)}_N(t). \quad (10)$$

The B-spline function $\beta^N(t)$ is piece-wise polynomial with N pieces of degree $N - 1$ each [36]. Therefore, this interpolation can be implemented using the Farrow structure as it was developed in [7][37]. However, by exploiting the particularity of the impulse response definition as a cascade of rectangular functions, a more efficient implementation structure is found. It was presented previously that NU filter taps are needed for an SRC filter of order $N - 1$. In the case of the rectangular function, we have U coefficients taken from the causal rectangular function, with all coefficients having a value of one, which can be expressed through the Z transform as:

$$H(z) = \sum_{n=0}^{U-1} z^{-n} = (1 - z^{-U}) \left(\frac{1}{1 - z^{-1}} \right) = C(z^U)I(z). \quad (11)$$

This is the non normalized moving average filter, which can be implemented in a recursive running-sum structure. It consists of two construction blocks: the integrator having the transfer function $I(z)$ and the comb with a delay z^{-U} having the transfer function $C(z^U)$ [50]. Then by cascading N rectangular function filters, the B-spline interpolation of order $N - 1$ is implemented. We know that the comb and the integrator are linear time-invariant systems and, therefore, their order of placement relative to one another in the structure can be exchanged. Then, benefiting from the second Noble identity, the comb blocks can be placed before the up-sampling operation optimizing the implementation by reducing the delay line of the comb blocks z^{-U} to only one delay element z^{-1} . This results in the structure presented by Hagenauer in [5], and shown in Figure 6, called the CIC interpolation filter, where a structure having N comb and N integrator blocks implements B-spline interpolation of order $N - 1$.

4.2 Practical Implementation

For CIC filters, an SRC controller is not needed since the filter implements a U-F-D structure. This section develops the particular implementation aspects to the CIC filters.

Output Re-scaling: The CIC filter has an inherent gain that can be expressed as $G_{CIC} = U^{N-1}$, which requires adding $I_G = \lceil N \log_2(U) \rceil$ bits to the processed signal in the case of fixed-point quantization. Certain implementations may require that the output of the CIC filter to be normalized to its input, requiring thereby an output re-scaling. This is done in two steps: coarse and fine gain adjustments. The coarse gain adjustment consists of normalizing the integer gain factor, by right shifting the quantization by $s = \lfloor \log_2(G_{CIC}) \rfloor$. Fine gain adjustment then finalizes the normalization by multiplying the samples with the factor $SF = 2^s / G_{CIC}$. The output of this multiplication is then rounded to the desired bit width. The reader

may refer to Hogenauer's work [5] for more details about the quantization in the intermediate stages.

Selecting the Filter Order: Three measures are used to select the appropriate CIC filter order for a given SRC operation: pass-band width, image attenuation level, and side lobe rejection levels. To simplify the design process, Hogenauer proposed in the original article tables that sums-up these measures for certain scenarios [5].

CIC Compensation Filter: The frequency response of B-spline interpolation is known to have an important attenuation in its pass-band. A solution to improve the frequency response was proposed in [51] that uses filter sharpening techniques resulting in the sharpened CIC filter structure, with better pass-band and improved filtering. It is also possible to improve the response by using compensation filters that aim at correcting the attenuation of the CIC filter [52, 53, 54, 55, 56].

Non-Recursive CIC Filter Structures: Another downside of the classical CIC filter is the recursive structure of the integrator blocks, that limits the maximum speed of the implementation. A non-recursive structure was proposed in [57] offering lower power consumption and higher operating speeds compared to the recursive CIC structure. Further improvements were achieved in [58] through a polyphase non-recursive CIC implementation.

5 Low-cost Arbitrary SRC: Newton Structure

The Newton interpolation structure presented in [6], and illustrated in Figure 7, offers a low-cost arbitrary SRC filter solution based on Lagrange interpolation. The structure modifies what was proposed in [59][60] to accommodate for a variable fractional delay.

5.1 Newton Structure Derivation

Lagrange interpolation is a widely used interpolation technique due its simplicity and the maximally flat pass-band of its frequency response. The Lagrange interpolation filter is piece-wise polynomial by definition and, therefore, can be implemented using the Farrow structure [25, 26]. In [61], the Lagrange interpolation filter response was developed using the B-spline response, and showed that the zeros of Lagrange interpolation are the same for both interpolations, hence sharing the role of image suppression. The frequency response of Lagrange interpolation focuses on approximating the ideal filter at $f = 0$ Hz the best way possible [62].

The Newton structure is a particular way of implementing the maximally flat FIR filter at $f = 0$ Hz. First, the delay factor μ in the case of the Farrow structure is transformed into the causal version of the delay expressed as $d = [(N - 1)/2] - \mu$. Then, the Z-domain transfer function of the ideal filter definition can be written as:

$$H^d(z) = z^{-d} \quad \text{with} \quad z = e^{j2\pi f T_{in}}. \quad (12)$$

The main mathematical problem of the above expression (12) is that in the Z-domain, the terms z are used to express delays by an integer number of samples.

However, in this case d is a real value and, therefore, evaluating $H^d(z)X(z)$ in this form will result in an ambiguous output expression. This was addressed in [63], where the transfer function $z^{-p/q}$ was approximated as a partial series of terms z^{-i} , with p and q being real numbers, and i being integer. This is done using the generalized binomial theorem, where it is possible to re-write $H^d(z)$ as:

$$\begin{aligned} H^d(z) &= [1 + (z^{-1} - 1)]^d \\ &= \sum_{k=0}^{+\infty} \frac{d(d-1) \dots (d-k+1)}{k!} (z^{-1} - 1)^k. \end{aligned} \quad (13)$$

This series converges for $|z^{-1} - 1| < 1$. Truncating this expression to the first N terms is known to result in the partial sum that satisfies the maximally flat criterion presented previously [60, 64, 65]. And since this criterion has a unique solution, then the partial sum of N terms corresponds to the Lagrange interpolation of order $N - 1$ [62][66]. The partial sum of (13) of order N corresponds to the Newton backward difference formula of order $N - 1$, that implements the Lagrange interpolation [59][60]. The filter structure is to be modified as it is shown by [6] in order to support variable fractional delays, resulting in the Newton interpolation structure, presented in Figure 7. This structure has a complexity of $O(N)$ compared to the Farrow structure implementing Lagrange interpolation of order N with a complexity of $O(N^2)$. It is important to note that a rigorous demonstration of the convergence of the Newton fractional-delay filter to the ideal fractional delay filter was recently published in [67].

5.2 Practical Implementation

The Newton structure does not have any different practical implementation aspects than those presented earlier in this paper. The same hardware implementation considerations and the same coefficients quantization guidelines presented for the U-F-D/polyphase structures apply for the Newton structure. The quantization of the fractional delay follows the same approach presented for the Farrow structure. Finally, the implementation of the SRC controller is also identical, with the only difference being the added circuit to calculate the value of the causal fractional delay d .

6 Improved Low-Cost Arbitrary SRC: Generalized Newton Structure

To extend the use of the Newton structure, D. Lamb *et al.* developed the transformation matrices that show the relation between the Farrow implementation of the Lagrange interpolation and the Newton interpolation structure [7]. Then they applied these transformations to B-spline interpolation of order 3 using the Farrow structure to get a modified Newton structure implementation. Later on, this transformation was used to modify the Newton structure in different ways to obtain improved low-cost arbitrary SRC solutions [8, 9, 10].

6.1 Farrow to Newton Transformation

The matrix representation of the Farrow structure transfer function presented in (8) uses a monomial basis vector $\boldsymbol{\mu}_k$, and a cumulative delay base vector $Z(z)$. The work

developed in [7] shows that it is possible to transform these base vectors to those of the Newton structure, allowing thereby to deduce the corresponding Newton structure implementation of any Farrow structure. The Newton structure is based on differentials between samples and their derivatives. In this case, the polynomial evaluation is done with the Newton polynomial base vector \mathbf{d}^T expressed as:

$$\mathbf{d}_{1 \times M}^T = [1, d, d(d-1), \dots, \prod_{i=0}^{M-2} (d-i)]. \quad (14)$$

For the delay vector, the Newton structure uses a differential delay base vector $\nabla Z(z)$ expressed as:

$$\nabla Z(z)_{N \times 1} = [1, (1-z^{-1}), \dots, (1-z^{-1})^{N-1}]. \quad (15)$$

Then, the matrix representation of the Newton structure transfer function is written as:

$$H_{i_k}(z) = \mathbf{d}_{1 \times M}^T \times \mathbf{Q}_{M \times N} \times \nabla Z(z)_{N \times 1}, \quad (16)$$

where the matrix \mathbf{Q} containing the coefficients describing the Newton structure implementation. By transforming the base vectors in (8) to those of the Newton structure in (16), the Newton structure coefficients matrix \mathbf{Q} corresponding to the matrix \mathbf{P} of the Farrow structure can be found as follows:

$$\begin{aligned} H(z, \mu) &= \boldsymbol{\mu}^T \mathbf{P} Z(z) \\ &= \boldsymbol{\mu}^T (T_d^T T_d^{-T}) \mathbf{P} (T_z^{-1} T_z) Z(z) \\ &= (T_d \boldsymbol{\mu})^T (T_d^{-T} \mathbf{P} T_z^{-1}) (T_z Z(z)) \\ &= \mathbf{d}^T \mathbf{Q} \nabla Z(z), \end{aligned} \quad (17)$$

where $T_d^{-T} = (T_d^T)^{-1}$ and the matrices T_d and T_z are the ones transforming the vectors $\boldsymbol{\mu}$ and $Z(z)$ to \mathbf{d} and $\nabla Z(z)$, respectively [7]. The analytical expressions of these matrices are developed in [8]. This transformation allows implementing any polynomial based response using a modified Newton structure as shown in Figure 8 for a random example. Concrete examples of modified Newton structures are provided in the following paragraphs.

6.2 Modified Newton Structures:

In the most general case, the generalized Newton structure can implement any polynomial based filter response, however, the order of complexity will be comparable to that of the Farrow structure. Nevertheless in certain particular cases, the generalized Newton structure can offer much more efficient implementations than it is possible using the Farrow structure. This section presents some of these cases.

B-spline interpolation: The original work that proposed this transformation applied it to the case of B-spline interpolation of order 3 [7]. This resulted in the modified structure shown in Figure 9-(a), that modifies the classical Newton structure of order 3 using only three additional adders.

Hermite interpolation: A disadvantage of B-spline interpolation is the important attenuation in its pass-band, limiting its practicality for certain transceiver systems. Hermite interpolation is a more interesting interpolation method that keeps the flat pass-band of Lagrange interpolation while improving the side lobes rejection levels. The modification of the Newton structures to implement this type of interpolation was developed in [9], that resulted in filter structures having lower complexity than the classical Newton structures of the same order, while offering improved filtering performance. An example modified structure of order 3 is shown in Figure 9-(b).

Optimization methods: By using the Farrow to Newton transformation, the filter response relation to the generalized Newton structure coefficients is not explicit. However, this relation can be found using the analytical expressions of this transformation as developed in [8]. This results in the closed-form expression of the generalized Newton structure frequency response. It is then possible to use filter optimization methods to find customized Newton structures with the best filter response possible. An example is shown in Figure 9-(c) for a customized structure of order 3, where it is supposed that the modification introduces feedback lines from the last delay element only. Then by using filter optimization methods, the coefficients c_1 to c_8 that correspond to the optimal approximation of the desired filter response are found. A concrete example is developed in [8] that shows how it is possible to achieve similar filtering performance to the Farrow structure, using a modified Newton structure of a much lower complexity.

Reconfigurable structures: All the previously presented structures are modifications of the classical Newton structure. By linearly combining the transfer function of the classical structure with any other modified one, it is possible to find reconfigurable structures using only one variable parameter c . This is done by expressing the filter response as $H_R(z) = (1 - c)H_{Newton}(z) + cH_{Modified}(z)$. The structure implementing $H_R(z)$ is then capable of performing both interpolations (Lagrange for $c = 0$, and the second interpolation for $c = 1$), and also a range of interpolations that combine the characteristics of both interpolation methods by using a value of $c \in [0; 1]$. This design approach is developed in [10], where two examples are provided. The reconfigurable structure implementing both Lagrange and Hermite interpolations is shown in Figure 9-(d).

7 SRC Solutions Comparison

The developed presentation of the FIR based SRC filters, summed-up in Figure 2, offers an easy way to grasp the relations between the different structures, clarifying thereby how to select the most appropriate structure for a certain SRC operation, how to compare the different structures, and how they can be combined together to perform any SRC operation efficiently.

A simple chart to select the best adapted SRC solution for a given application is represented in Table 1. For coarse SRC, the choice has to be made between the U-F-D/polyphase solution and the CIC filter for best performance and low-cost applications respectively. While for fine SRC, the choice has to be made between the Farrow and Newton structures. With the generalization of the Newton structure,

it is possible to achieve low-cost implementation with filter response approaching what is possible with the Farrow structure as presented in the last section.

The CIC filter is the reference SRC solution with the lowest complexity. However, it is only suited for coarse SRC operations. For fine SRC, the Newton structure used to be the most efficient implementation before our proposition of the modified Newton structures. We developed in [17] the implementation of low-cost SRC filters on application specific integrated circuits (ASIC). The results, illustrated in Figure 10, show that a Newton structure of order 5 uses only 72% of the hardware resources of an equivalent implementation using the Farrow structure. However, the modified Newton structure for Hermite interpolation requires only half the resources while offering improved filtering performance. This demonstrates the advantage of the modified Newton structures that make possible the implementation of fine SRC with a complexity approaching that of the coarse SRC CIC filters of a similar order. This was not conceivable a decade ago before the propositions in [6, 7, 9].

For an efficient implementation, the SRC operation is implemented as a cascade of multiple SRC filters [1, 2, 13]. In order to partition an SRC operation into a multistage implementation, a simple guide is proposed below:

- 1 For SRC applications with a large changes in the sampling frequency, extract the first factor R_1 representing an interpolation or decimation operation by an integer factor larger than eight. Then either the CIC filters or the U-F-D/polyphase structures are used depending on the filtering requirements. If the factor R_1 can be broken into multiple factors of 2 or 1/2, then half-band filters offer improved filtering performance with a very low complexity.
- 2 After extracting the factor R_1 , identify if there exists a lower coarse rational or integer SRC factor R_2 . Usually when a CIC filter is used, this second stage combines further SRC with the CIC filter response compensation. The implementation choice of R_2 largely depends on the system, but is often implemented using the U-F-D / polyphase structures.
- 3 Finally, the fine adjustment of the output sampling frequency is represented by the factor $R_3 \approx 1$. Then depending on the application requirements, a Farrow or Newton structure is used for implementation, offering an average side lobe attenuation of 60-70 dB or 35-45 dB respectively for an order 5 implementation.

8 Conclusion

In this paper, a succinct presentation of the main FIR based SRC solutions was developed. This presentation is based on a deep analysis of the previously proposed solutions for FIR-based SRC filtering, which resulted first in the derivation of a unified vision of this issue, and second in the development of a design methodology guide for SRC designers. Both cases of coarse and fine SRC were considered, and two types of solutions for each case were given: best performance and low-cost solutions. Recently proposed fine SRC filters based on the Newton structure were also presented, that allow implementing fine SRC with a complexity approaching that of the more simple coarse SRC implementations. This presentation also covered the different practical implementation aspects in hardware, and provided a simple guide to help implementing any SRC operation using a multi-stage architecture.

Through this tutorial, any engineer in the field should be able to easily understand and use the basics of the presented SRC solutions.

Abbreviations

SRC: Sample rate conversion; FIR: Finite impulse response; CIC: Cascaded integrator-comb; U-F-D: Up-sampling/Filtering/ Down-sampling; AI: Anti-imaging; AA: Anti-aliasing;

Authors' contributions

AZ is the main author of the current paper. SP and AN came up with the original idea. As academic supervisors, CM and AN have proofread the paper several times and provided guidance throughout the whole preparation of the manuscript. SP, AZ, and PYJ analyzed and compared with an existing survey of sample rate converters. All of the authors participated, and they read and approved the final manuscript.

Funding

This work has been achieved within the Institute of Research and Technology IRT-B<>COM, dedicated to digital technologies. It has been funded by TéléDiffusion de France (TDF) and the French government through the National Research Agency (ANR) under the Investissements d'Avenir program, with reference number ANR-A0-AIRT-07.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analysed during the study.

Competing interests

The authors declare that they have no competing interests.

Author details

¹TDF, 155 bis Avenue Pierre Brosolette, 92120 Montrouge, France. ²IETR UMR CNRS 6164, SCEE, CentraleSupélec, Avenue de la Boulaie, 35510 Cesson Sévigné, France. ³IRT-BCOM, 1219 Avenue des Champs Blancs, 35510 Cesson Sévigné, France. ⁴Univ Rennes, CNRS, IETR - UMR 6164, Rue du Thabor, 35000 Rennes, France.

References

1. Fa-Long, L., *Digital Front-End in Wireless Communications and Broadcasting: Circuits and Signal Processing*, Cambridge University Press, United Kingdom, 2011.
2. Crochiere R.E., Rabiner L.R., *Multirate Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1983.
3. M. Bellanger, G. Bonnerot and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 2, pp. 109-114, 1976.
4. C. W. Farrow, "A continuously variable digital delay element," *IEEE International Symposium on Circuits and Systems*, pp. 2641-2645, vol.3, Espoo, Finland, 1988.
5. E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155-162, 1981.
6. V. Lehtinen and M. Renfors, "Structures for interpolation, decimation, and nonuniform sampling based on Newton's interpolation formula," *International Conference on Sampling Theory and Applications*, 2009.
7. D. Lamb, L. F. O. Chamon and V. H. Nascimento, "Efficient filtering structure for spline interpolation and decimation," *Electronics Letters*, vol. 52, no. 1, pp. 39-41, 2016.
8. A. Zeineddine, S. Paquelet, A. Nafkha, C. Moy and P. Jezequel, "Generalization and Coefficients Optimization of the Newton Structure," *IEEE 25th International Conference on Telecommunications*, pp. 98-103, St. Malo, France, 2018.
9. A. Zeineddine, A. Nafkha, C. Moy, S. Paquelet and P. Jezequel, "Variable Fractional Delay Filter: A Novel Architecture Based on Hermite Interpolation," *IEEE 25th International Conference on Telecommunications*, pp. 93-97, St. Malo, France, 2018.
10. A. Zeineddine, S. Paquelet, M. Kanj, C. Moy, A. Nafkha and P. Y. Jezequel, "Reconfigurable Newton structure for sample rate conversion," *IEEE Global Conference on Signal and Information Processing*, pp. 271-275, Anaheim, CA, USA, 2018.
11. C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10-21, 1949.
12. T.A. C. M. Claasen, W.F.G. Mecklenbrauker, "On the transposition of linear time-varying discrete-time networks and its application to multirate digital systems," *Philips, J. Res.*, vol. 33, pp. 78-102, 1978.
13. R. Crochiere and L. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 5, pp. 444-456, 1975.
14. C. C. Hsiao, "Polyphase filter matrix for rational sampling rate conversions," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2173-2176, Dallas, TX, USA, 1987.
15. R. Bregovic, Y. J. Yu, T. Saramäki and Y. C. Lim, "Implementation of Linear-Phase FIR Filters for a Rational Sampling-Rate Conversion Utilizing the Coefficient Symmetry," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 548-561, 2011.
16. F. Harris, "Performance and design considerations of the Farrow filter when used for arbitrary resampling of sampled time series," *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, pp. 1745-1749, vol.2., Pacific Grove, CA, USA, 1997.

17. A. Zeineddine, S. Paquelet, A. Nafkha, P. Jezequel and C. Moy, "Efficient Arbitrary Sample Rate Conversion for Multi-Standard Digital Front-Ends," *IEEE 17th International New Circuits and Systems Conference*, pp. 1-4, Munich, Germany, 2019.
18. J.I. Choi, H.S. Jun, S.Y. Hwang, "Efficient hardware optimisation algorithm for fixed point digital signal processing ASIC design," *Electronics Letters*, vol. 32, no. 11, pp. 992-994, 1996.
19. R. Crochiere, "A new statistical approach to the coefficient word length problem for digital filters," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 3, pp. 190-196, 1975.
20. P. W. Wong, "Quantization and round off noises in fixed-point FIR digital filters," *IEEE Transactions on Signal Processing*, vol. 39, no. 7, pp. 1552-1563, 1991.
21. J. Qiao, P. Fu, S. Meng, "A Combined Optimization Method of Finite Wordlength FIR Filters," *First International Conference on Innovative Computing, Information and Control - Volume I*, pp. 103-106, Beijing, China, 2006.
22. L.B.Jackson, "On the interaction of round off noise and dynamic range in digital filters," *Bell System Technical Journal*, Vol. 49, no. 2, 159-184, 1970.
23. W. Sung, K. I. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," in *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 3087-3090, 1995.
24. D. Menard, R. Rocher, O. Sentieys, "Analytical Fixed-Point Accuracy Evaluation in Linear Time-Invariant Systems," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 10, pp. 3197-3208, 2008.
25. L. Erup, F. M. Gardner, R. A. Harris, "Interpolation in digital modems. II. Implementation and performance," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 998-1008, 1993.
26. T. I. Laakso, V. Valimäki, M. Karjalainen and U. K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30-60, 1996.
27. J. Vesma and T. Saramäki, "Interpolation filters with arbitrary frequency response for all-digital receivers," *IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World*, pp. 568-571 vol.2, Atlanta, GA, USA, 1996.
28. M. Henker, G. Fettweis, "Combined filter for sample rate conversion, matched filtering and symbol synchronization in software radio terminals," *Proceedings of the European Wireless*, pp. 61-66, 2000.
29. T. Hentschel, G. Fettweis, "Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals," *Frequenz*, Vol. 55, p. 185-188, 2001.
30. D. Babic, J. Vesma, T. Saramäki and M. Renfors, "Implementation of the transposed Farrow structure," *IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, Phoenix-Scottsdale, AZ, USA, 2002.
31. D. Babic and M. Renfors, "Power efficient structure for conversion between arbitrary sampling rates," *IEEE Signal Processing Letters*, vol. 12, no. 1, pp. 1-4, Jan. 2005.
32. D. Babic and M. Renfors, "Reconstruction of non-uniformly sampled signal using transposed Farrow structure," *IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, pp. III-221, Vancouver, BC, 2004.
33. W. Li, M. Tomisawa, "Transposed-Farrow-structure-based multirate filters for symbol timing synchronization in software defined radio (SDR)," *IEEE 60th Vehicular Technology Conference*, pp. 1668-1672, vol. 3, Los Angeles, CA, 2004.
34. S. Paquelet, V. Savaux, "On the symmetry of FIR filter with linear phase," *Digital Signal Processing*, vol. 81, pp. 57-60, 2018.
35. E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, pp. 319-342, vol. 90, no. 3, 2002.
36. M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp. 22-38, vol. 16, no. 6, 1999.
37. S. R. Dooley, R. W. Stewart and T. S. Durrani, "Fast on-line B-spline interpolation," *Electronics Letters*, pp. 1130-1131, vol. 35, no. 14, 1999.
38. G.V. Milovanovic, Z. Udovicic, "Calculation of coefficients of a cardinal b-spline," *Applied Mathematics Letters*, p. 1346-1350, vol. 23, no.11, 2010.
39. X. Huang, Y. J. Guo and J. A. Zhang, "Sample Rate Conversion Using B-Spline Interpolation for OFDM Based Software Defined Radios," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2113-2122, 2012.
40. V. Välimäki, "A new filter implementation strategy for Lagrange interpolation," *IEEE International Symposium on Circuits and Systems*, pp. 361-364, vol. 1, Seattle, WA, USA, 1995.
41. T. Deng, "Coefficient-Symmetries for Implementing Arbitrary-Order Lagrange-Type Variable Fractional-Delay Digital Filters," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4078-4090, 2007.
42. P. Soontornwong, S. Chivapreecha, C. Pradabpet, "A Cubic Hermite variable fractional delay filter," *IEEE International Symposium on Intelligent Signal Processing and Communications Systems*, Chiang Mai, Thailand, 2011.
43. C. Tseng, S. Lee, "Design of Fractional Delay Filter Using Hermite Interpolation Method," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 7, pp. 1458-1471, 2012.
44. J. Vesma, T. Saramäki, "Optimization and efficient implementation of FIR filters with adjustable fractional delay," *IEEE International Symposium on Circuits and Systems*, pp. 2256-2259, vol. 4, Hong Kong, 1997.
45. W. S. Lu, T. B. Deng, "An improved weighted least-squares design for variable fractional delay FIR filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 8, pp. 1035-1040, 1999.
46. H. Johansson, P. Lowenborg, "On the design of adjustable fractional delay FIR filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 4, pp. 164-169, 2003.
47. M. T. Hunter, *Design of polynomial-based filters for continuously variable sample rate conversion with applications in synthetic instrumentation and software defined radio*, PhD thesis, University of Central Florida Orlando, Florida, 2008.
48. F. Lopez, J. Vesma, T. Saramäki, M. Renfors, "The effects of quantizing the fractional interval in interpolation

- filters," *Proceedings of the Nordic Signal Processing Symposium*, Sweden, 2000.
49. F. Lopez, J. Vesma, M. Renfors, "Defining the word length of the fractional interval in interpolation filters," *11th European Signal Processing Conference*, Toulouse, France, 2002.
 - 50.
 51. A. Y. Kwentus, Z. Jiang, A. N. Willson, "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 457-467, 1997.
 - 52.
 53. G. Molnar, M. Vucic, "Closed-Form Design of CIC Compensators Based on Maximally Flat Error Criterion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 12, pp. 926-930, 2011.
 54. A. Fernandez-Vazquez, G. J. Dolecek, "Maximally Flat CIC Compensation Filter: Design and Multiplierless Implementation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 2, pp. 113-117, 2012.
 55. G. J. Dolecek, S. K. Mitra, "Simple method for compensation of CIC decimation filter," *Electronics Letters*, vol. 44, no. 19, pp. 1162-1163, 2008.
 56. G. J. Dolecek, "Simple wideband CIC compensator," *Electronics Letters*, vol. 45, no. 24, pp. 1270-1272, 2009.
 57. Y. Gao, L. Jia, J. Isoaho, H. Tenhunen, "A Comparison Design of Comb Decimators for Sigma-Delta Analog-to-Digital Converters," *Analog Integrated Circuits and Signal Processing*, vol. 22, p. 51-60, 2000.
 58. T. K. Shahana, R. K. James, B. R. Jose, K. P. Jacob, S. Sasi, "Polyphase Implementation of Non-recursive Comb Decimators for Sigma-Delta A/D Converters," *IEEE Conference on Electron Devices and Solid-State Circuits*, pp. 825-828, Tainan, Taiwan, 2007.
 59. P. Depalle, S. Tassart, "Fractional delay lines using Lagrange interpolators," *International Computer Music Conference*, pp.341-343, Hongkong, 1996.
 60. S. Tassart, P. Depalle, "Analytical approximations of fractional delays: Lagrange interpolators and allpass filters," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.1, pp. 455-458, Munich, 1997.
 61. A. Franck, K. Brandenburg, "A Closed-Form Description for the Continuous Frequency Response of Lagrange Interpolators," *IEEE Signal Processing Letters*, vol. 16, no. 7, pp. 612-615, 2009.
 62. E. Hermanowicz, "Explicit formulas for weighting coefficients of maximally flat tunable FIR delayers," *Electronics Letters*, vol. 28, no. 20, pp. 1936-1937, 1992.
 63. V. Ramachandran, C. S. Gargour, M. Ahmadi, "Generation of digital transfer functions of the FIR type approximating $z^{\sup -p/q}$," *IEEE International Symposium on Circuits and Systems*, vol.4, pp. 3260-3262, New Orleans, LA, USA, 1990.
 64. S. C. Pei, P. H. Wang, "Closed-form design of maximally flat FIR Hilbert transformers, differentiators, and fractional delayers by power series expansion," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 4, pp. 389-398, 2001.
 65. S. Samadi, M. O. Ahmad, M. N. S. Swamy, "Results on maximally flat fractional-delay systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 11, pp. 2271-2286, 2004.
 66. P. J. Kootsookos, R. C. Williamson, "FIR approximation of fractional sample delay systems," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 269-271, 1996.
 67. S. Paquelet, A. Zeineddine, A. Nafkha, P. Jezequel, C. Moy, "Convergence of the Newton Structure Transfer Function to the Ideal Fractional Delay Filter," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1354-1358, 2019.

Tables

Table 1 Choosing the best adapted SRC solution

	Best Performance	Low-Cost
Coarse SRC	U-F-D / Polyphase	CIC filter
Fine SRC	Farrow structure	Newton structure

Figures

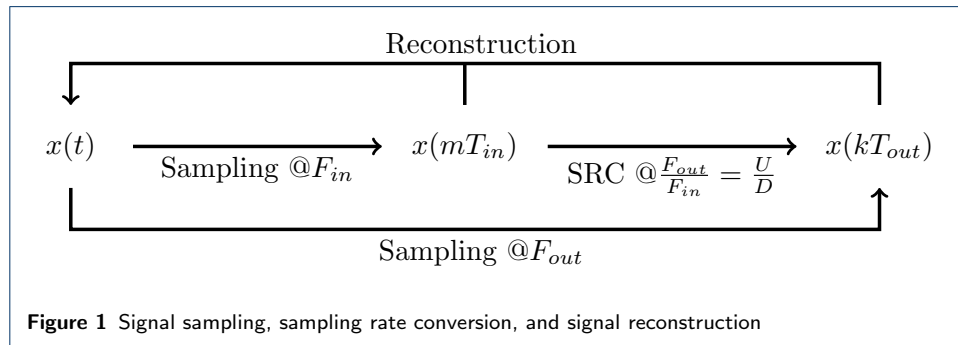


Figure 1 Signal sampling, sampling rate conversion, and signal reconstruction

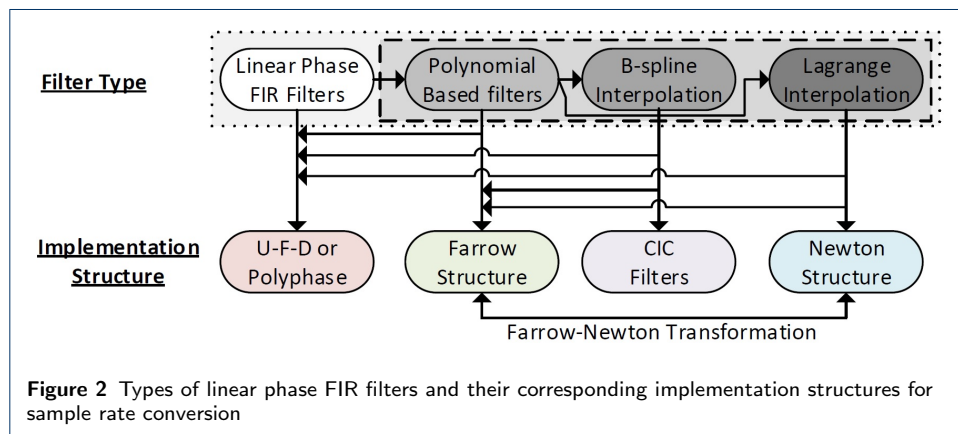
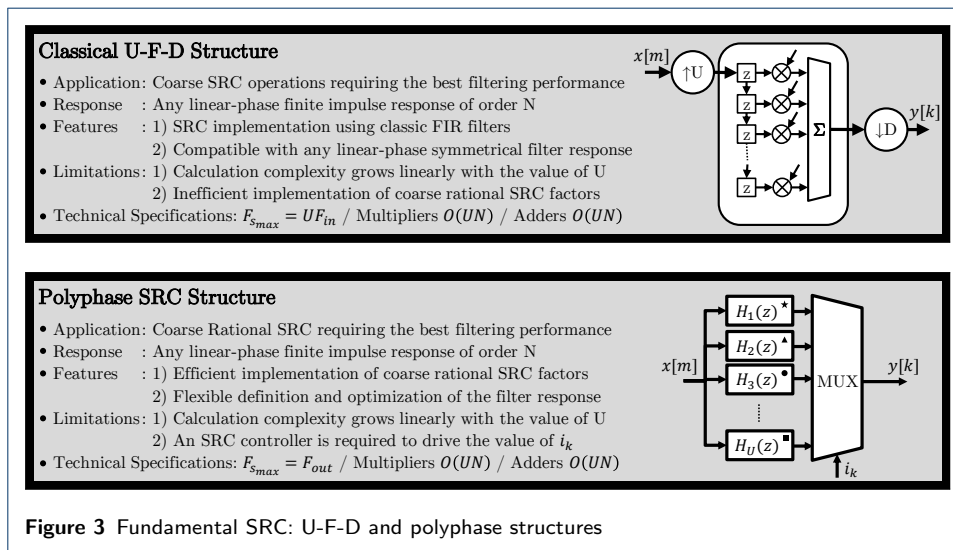
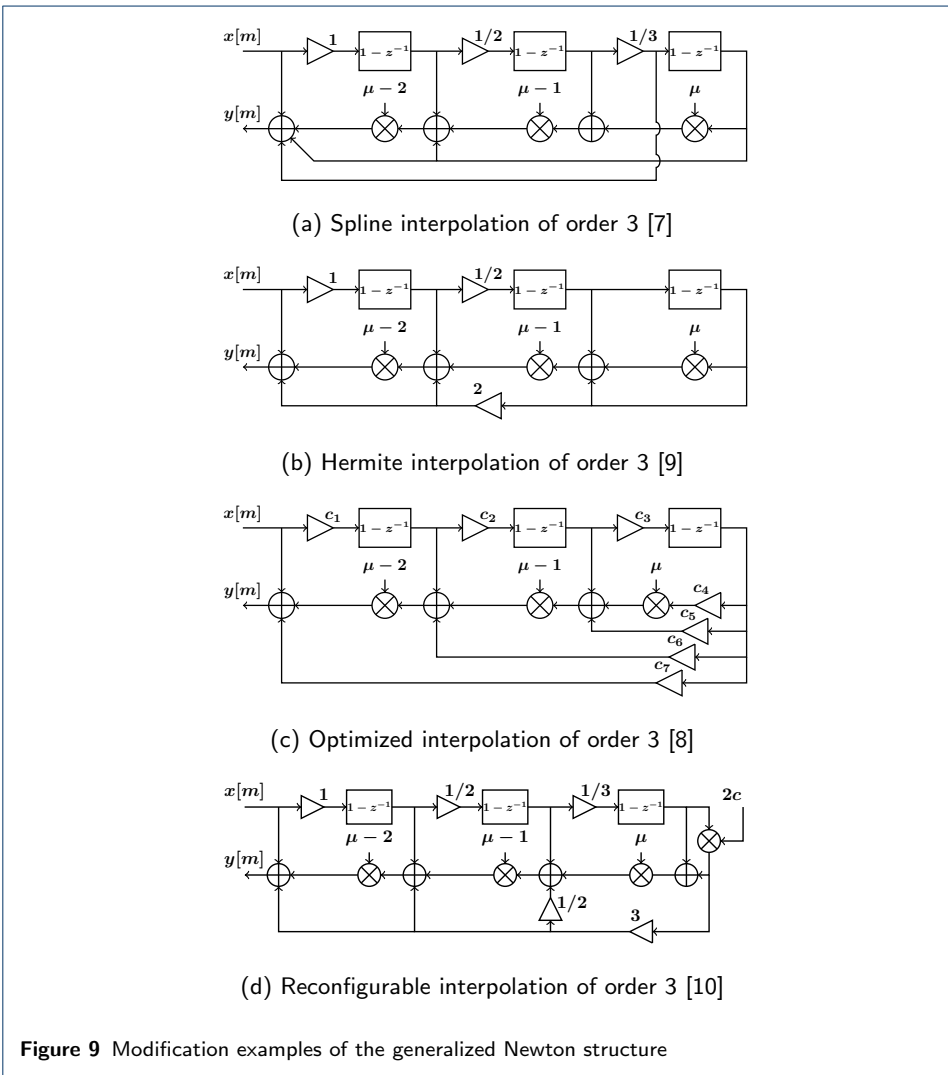
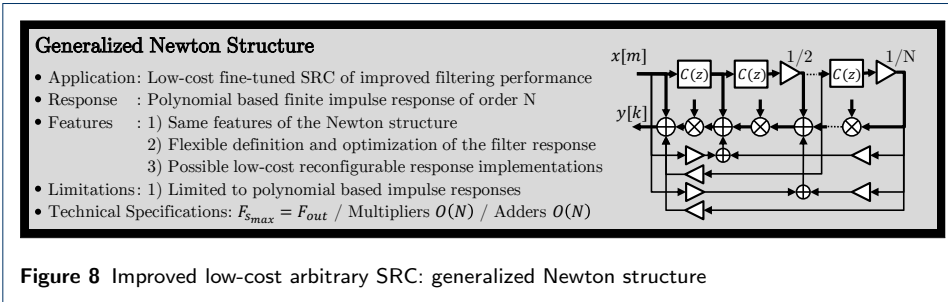
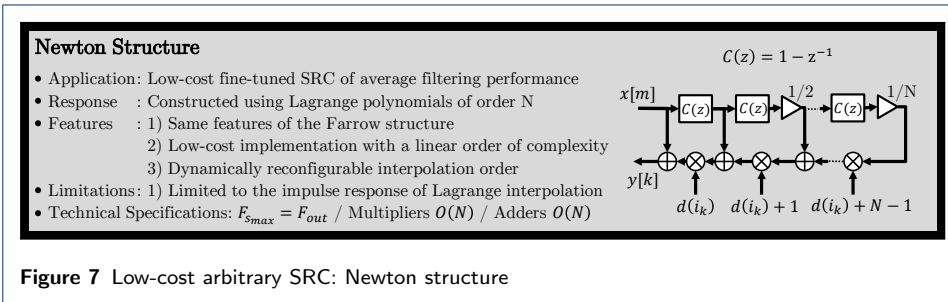


Figure 2 Types of linear phase FIR filters and their corresponding implementation structures for sample rate conversion





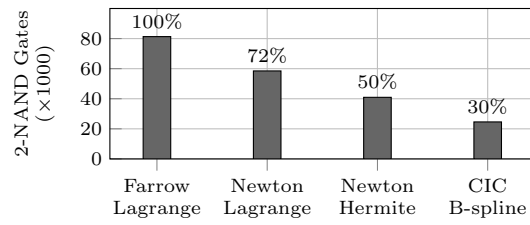


Figure 10 Complexity on ASIC hardware of order 5 implementations