IPN Progress Report 42-168 February 15, 2007

Variable Sample Rate Conversion Techniques for the Advanced Receiver

A. Tkacenko[1]

One of the primary challenges in the development of the Advanced Receiver is the ability to accommodate a wide variety of possible data rates, motivated by the desire to support different missions for the Deep Space Network (DSN) under different adverse conditions. To conform to fixed architectures such as the analog-to-digital converter (ADC) used at the front end and tracking loops used subsequently, this requires that the sampling rate be varied entirely digitally after the ADC to accommodate the fixed parameters of the tracking loops. In this article, we present a series of methods to achieve variable sample rate conversion (SRC). Specifically, we focus on two sets of schemes to alter the sampling rate: coarse techniques to lower the bulk of the sampling rate near the desired amount while removing out-of-band artifacts due to noise and interference, and fine techniques used to accurately tailor the sampling rate to the exact desired value. Advantages and disadvantages of both sets of methods are investigated in terms of implementation complexities and performance metrics.

I. Introduction

In order to accommodate a wide variety of upcoming missions for the Deep Space Network (DSN) under a myriad of different adverse conditions, one of the primary goals of the Advanced Receiver replacement for the current Block V Receiver is the ability to support a vast range of data rates. For a typical conceivable scenario, it would be desirable to accommodate data rates varying continuously from 10 bits per second (bps) to over 100 Mbps.

In addition to being able to support such a wide range of data rates, the Advanced Receiver should also conform to certain fixed system architectures. These fixed architectures both preserve the modularity and minimize the computational complexity and required reconfigurability of the overall system. Examples of these fixed system architectures include the analog-to-digital converter (ADC) used at the front end of the receiver and the tracking loops (carrier synchronization, symbol timing, etc.) used subsequently. Typically, it is desirable to use a high-speed ADC with a fixed sampling rate and tracking loops with a fixed number of samples per symbol. For example, for the Deep Space Array-Based Network (DSAN), the sampling rate of the ADC is fixed at 1280 MHz [1] while the tracking loops typically will require 4 to

[1] Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

8 samples per symbol (samp/symb). In order to simultaneously accommodate these fixed systems, it is necessary to vary the sampling rate entirely digitally.

One example of a possible Advanced Receiver intermediate frequency (IF) system using digital variable sample rate conversion (SRC) is shown in Fig. 1. Here, the IF analog input, which consists of a data-modulated signal at data rate R (measured in symbols per second), is sampled at rate F_s (measured in samples per second), producing a real IF digital signal. This typically is then converted to a quadrature signal, yielding a complex baseband signal sampled at rate $F_s' = F_s/2$. To facilitate subsequent processing of the signal, the frequency band of interest typically is centered at zero frequency through the use of a numerically controlled oscillator (NCO). At this point, the sampling rate of the signal must be altered to accommodate the redundancy factor or number of samples per symbol as required in the tracking loops. For example, if the tracking loops require a redundancy factor of K (measured in samples per symbol), then the sampling rate coming into the tracking loops should be $F_s'' = KR$. Hence, the purpose of the variable SRC system is to alter the sampling rate from F_s' to F_s''.

As can be seen from Fig. 1, in a typical communications-type setting, we would like to keep the ADC sampling rate F_s and tracking loop redundancy factor K fixed. This will improve the overall modularity, computational complexity, and developmental costs of the entire system, by allowing us to leverage off of existing technology and communications blocks. In order to simultaneously accommodate these demands along with a variable data rate, the only blocks in Fig. 1 that require reconfigurability are the NCO and the SRC modules. Essentially, the NCO is used to center the signal of interest (SOI) to zero frequency, while the SRC module is used to "zoom in" to the desired SOI, removing any out-of-band artifacts in the process.

A more realistic scenario in which variable SRC could be used as a link between existing communications blocks for the DSN is shown in Fig. 2. For example, variable SRC could be used to process in the output obtained from the DSAN for a number of different applications, including telemetry and radio science, and even as an input to the existing Block V Receiver.

A. Outline

In Section II, we introduce the interpolation kernel signal model, of which the class of band-limited signals is a part. There, we discuss finite temporal support, polynomial-based kernels, and their relation to the band-limited sinc kernel. In Section III, we explore the SRC problem and mention how this can be done entirely digitally with sampled data. Methods for achieving variable SRC are proposed in Section IV. In particular, the emphasis is on two kinds of techniques: coarse schemes to adjust the bulk of the sample rate to the desired amount while removing out-of-band artifacts, and fine schemes to steer the sampling rate to its desired precise value. Implementation considerations are explored in Section V, and simulation examples of variable SRC in a communications-type environment are presented in Section VI. Finally, concluding remarks are made in Section VII.

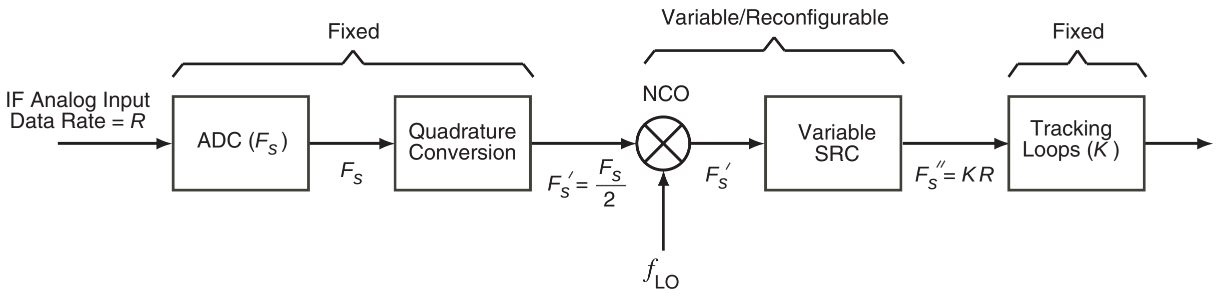


Fig. 1. Block diagram of a possible IF Advanced Receiver system employing purely digital variable SRC.

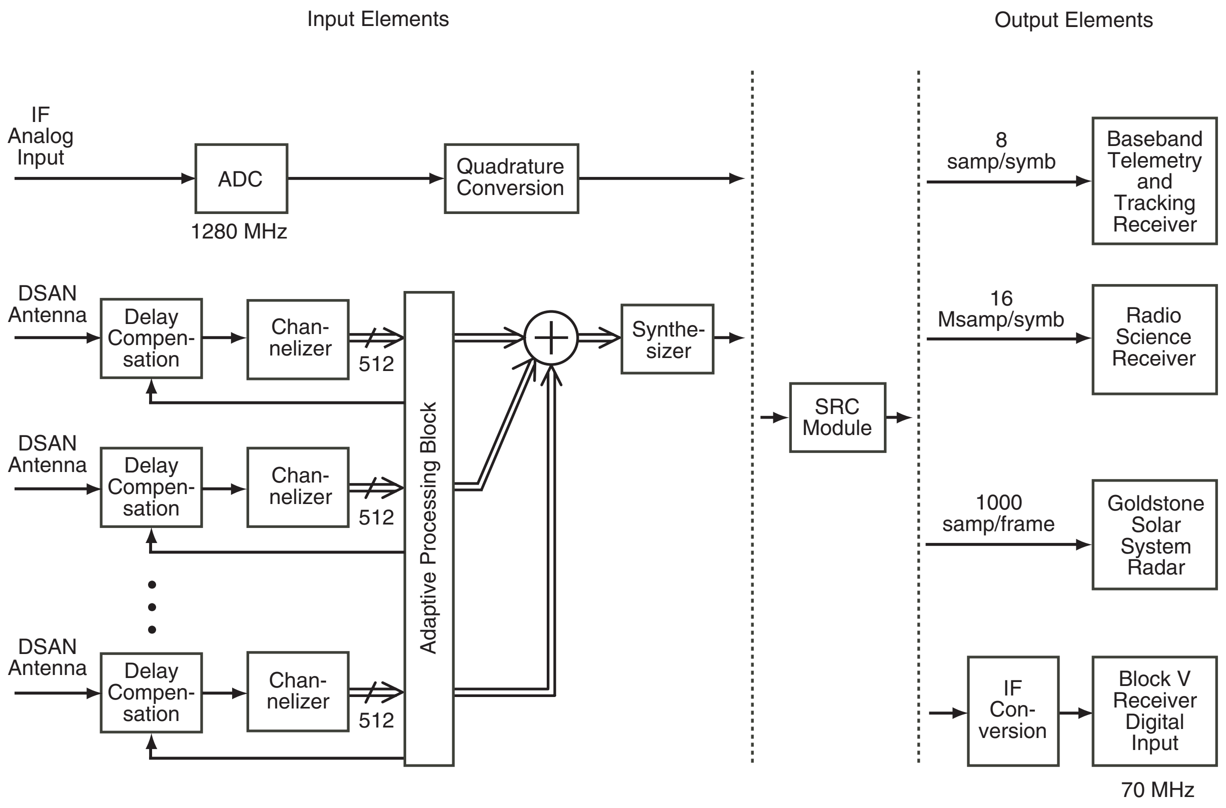


Fig. 2. Realistic scenario in which variable SRC could be used to link existing elements already in place for the DSN.

B. Notations

All notations used are as in [2]. In particular, continuous-time (analog) frequencies are presented using capital letters, whereas discrete-time (digital) frequencies will be presented using lowercase letters. For example, Ω and F will denote radian and normalized analog frequencies, respectively, whereas ω and f will denote the corresponding digital frequencies. Furthermore, parentheses will be used to denote continuous-time functions, whereas square brackets will be used to denote discrete-time functions. As an example, $x(t)$ would denote a continuous-time function for $t \in \mathbb{R}$, whereas $y[n]$ would denote a discrete-time function for $n \in \mathbb{Z}$.

II. Interpolation Kernel Signal Model

For the remainder of this article, we will assume that all of the analog signals that are to be sampled by an ADC are of the following form [3,4]:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k g(t - kT) \tag{1}$$

The signal model given by Eq. (1) will be referred to as the interpolation kernel signal model, where $g(t)$ denotes the interpolation kernel, T represents the kernel spacing interval, and $\{c_k\}$ denotes the set of basis coefficients corresponding to the signal $x(t)$.

Several interesting classes of signals are generated by the model given in Eq. (1) [4]. For example, from the Nyquist sampling theorem[2] [5], it is known that if $x(t)$ is band-limited to F_{BL}, meaning that $X(j2\pi F) = 0$ for $|F| \geq F_{BL}$, then $x(t)$ is of the following form:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right) \tag{2}$$

where the sinc function is defined as $\operatorname{sinc}(x) \triangleq \sin(\pi x)/\pi x$ and T_s is given by $T_s = 1/2F_{BL}$. Comparing Eq. (2) to Eq. (1), it can be seen that all band-limited signals can be expressed in terms of the interpolation kernel model, where we have

$$\begin{aligned} g(t) &= \operatorname{sinc}\left(\frac{t}{T_s}\right) \\ c_k &= x(kT_s) \\ T &= T_s \end{aligned} \tag{3}$$

It is for this reason that any signal band-limited to F_{BL} can be recovered from uniform sampling at a rate of $F_s = 1/T_s = 2F_{BL}$ (typically called the Nyquist rate [2,5]) as this will automatically yield the basis coefficients c_k's, as can be seen from Eq. (3).

After uniform sampling of the signal $x(t)$ from Eq. (1), we will have access to the sequence $x_d[n] \triangleq x(nT_s)$, where T_s denotes the sampling interval. For the remainder of this article, we will assume that the sampling interval and kernel spacing interval are one and the same, so that $T_s = T$. Furthermore, for the sake of simplicity, we will introduce a dilated interpolation kernel function $\widehat{g}(t)$, defined by

$$\widehat{g}(t) \triangleq g(tT) \Longleftrightarrow g(t) = \widehat{g}\left(\frac{t}{T}\right)$$

Then, using Eq. (1), we get,

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \widehat{g}\left(\frac{t}{T} - k\right) = \sum_{k=-\infty}^{\infty} c_k \widehat{g}\left(\frac{t}{T_s} - k\right) \tag{4}$$

Upon sampling $x(t)$ uniformly at $t = nT_s$, we obtain the discrete-time sequence $x_d[n] \triangleq x(nT_s)$, which is given by Eq. (4) to be the following:

$$x_d[n] = x(nT_s) = \sum_{k=-\infty}^{\infty} c_k \widehat{g}(n - k) = c_d[n] * \widehat{g}_d[n] \tag{5}$$

[2] This is sometimes referred to as the Nyquist–Whittaker–Shannon sampling theorem due to independent contributions from Nyquist (1928), Whittaker (1928), and Shannon (1949) [3,5].

Here $c_d[n] \triangleq c_n$ and $\widehat{g}_d[n] \triangleq \widehat{g}(n)$. In other words, from Eq. (5), the sampled sequence $x_d[n]$ is simply the discrete-time convolution [2] of the basis coefficients (the c_n's) with the sampled interpolation kernel sequence $\widehat{g}(n) = g(nT_s)$.

In order to reconstruct the original analog signal $x(t)$ from Eq. (1) or Eq. (4) at some prescribed value of time $t = t_0$, which will be an important problem later when we discuss the variable SRC problem, the only parameters needed are the basis coefficients c_k's, assuming the interpolation kernel $g(t)$ or $\widehat{g}(t)$ is a known function. From the obtained sampled sequence $x_d[n]$ given by Eq. (5), it can be seen that the sequence of basis coefficients $c_d[n]$ can be obtained as follows:

$$c_d[n] = \widehat{h}_d[n] * x_d[n]$$

where $\widehat{h}_d[n]$ is the convolutional inverse of $\widehat{g}_d[n]$ [2]. In the z-domain, this becomes

$$C_d(z) = \widehat{H}_d(z) X_d(z) \tag{6a}$$

where

$$\widehat{H}_d(z) = \frac{1}{\widehat{G}_d(z)} \tag{6b}$$

Here, $C_d(z)$, $X_d(z)$, $\widehat{H}_d(z)$, and $\widehat{G}_d(z)$ denote the z-transforms of $c_d[n]$, $x_d[n]$, $\widehat{h}_d[n]$, and $\widehat{g}_d[n]$, respectively. Hence, the basis coefficients can be obtained using the system shown in Fig. 3.

For the special case in which $x_d[n] = c_d[n]$, we say that the interpolation kernel $\widehat{g}(t)$ or $g(t)$ is interpolating [6,7]. Otherwise, the kernel is said to be non-interpolating. It can easily be shown that $\widehat{g}(t)$ or $g(t)$ is an interpolating kernel iff we have

$$\widehat{g}(n) = \delta[n] \Longleftrightarrow g(nT_s) = \delta[n] \quad \forall\, n \in \mathbb{Z} \tag{7}$$

where $\delta[n]$ is the Kronecker delta function [2] given by

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad \forall n \in \mathbb{Z}$$

Kernels that are interpolating as in Eq. (7) are said to satisfy the Nyquist criterion [5], and the functions $g(t)$ and $\widehat{g}(t)$ that satisfy Eq. (7) are said to be Nyquist(T_s) and Nyquist(1), respectively, based on the effective sampling interval used.

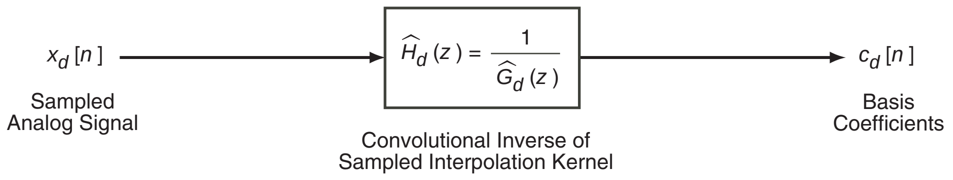

$x_d[n]$
Sampled Analog Signal
$\widehat{H}_d(z) = \frac{1}{\widehat{G}_d(z)}$
Convolutional Inverse of Sampled Interpolation Kernel
$c_d[n]$
Basis Coefficients

Fig. 3. Discrete-time system used to obtain the interpolation basis coefficients $c_k = c_d[k]$ from the sampled continuous-time signal $x(nT_s) = x_d[n]$.

While many important kernels such as the sinc kernel from Eq. (3) satisfy the Nyquist criterion, several notable others, including the cubic B-spline that we discuss in Section II.C, do not. In what follows, we will review the most commonly used interpolation kernels as well as their relation to the underlying sinc kernel that serves as the basis for the class of band-limited signals that is assumed here.

A. Sinc Interpolation Kernel

From Eq. (3), the dilated interpolation kernel $\widehat{g}(t)$ is given by

$$\widehat{g}(t) = \operatorname{sinc}(t) \tag{8}$$

The Fourier transform of $\widehat{g}(t)$, which we denote here by $\widehat{G}(j2\pi F)$, can be shown to be the following [2]:

$$\widehat{G}(j2\pi F) = \begin{cases} 1, & |F| < \dfrac{1}{2} \\ 0, & |F| \geq \dfrac{1}{2} \end{cases} \tag{9}$$

Plots of $\widehat{g}(t)$ from Eq. (8) and $\left|\widehat{G}(j2\pi F)\right|^2$ from Eq. (9) can be found in Figs. 4(a)[3] and 4(b),[4] respectively, along with several other interpolation kernels. As can be seen from Fig. 4(b), $\widehat{g}(t)$ is band-limited to $F = 1/2$. This implies that $\widehat{g}(t)$ is doubly infinite in duration [2,5] and smooth in the sense of being infinitely differentiable [3]. As a result, the task of variable SRC, which is explored in Section III, will be problematic as it will involve in general an infinite number of slowly decaying (proportional to $1/t$) computations over all input samples to compute a single output sample. Despite this issue, from Eq. (8) and Fig. 4(a), it can be seen that $\widehat{g}(t)$ is Nyquist(1), and so there is no need to use the system of Fig. 3 to compute the basis coefficients as the samples themselves are already the basis coefficients.

To overcome the problems resulting from variable SRC using the sinc interpolation kernel, we turn our focus to time-limited (i.e., finite temporal support) interpolation kernels. Not only are such kernels implementable in a practical setting, but they can often be implemented efficiently (see Section V) and with little performance loss compared to the true underlying sinc interpolation kernel (see Section VI).

B. Lagrange Piecewise Polynomial Kernels

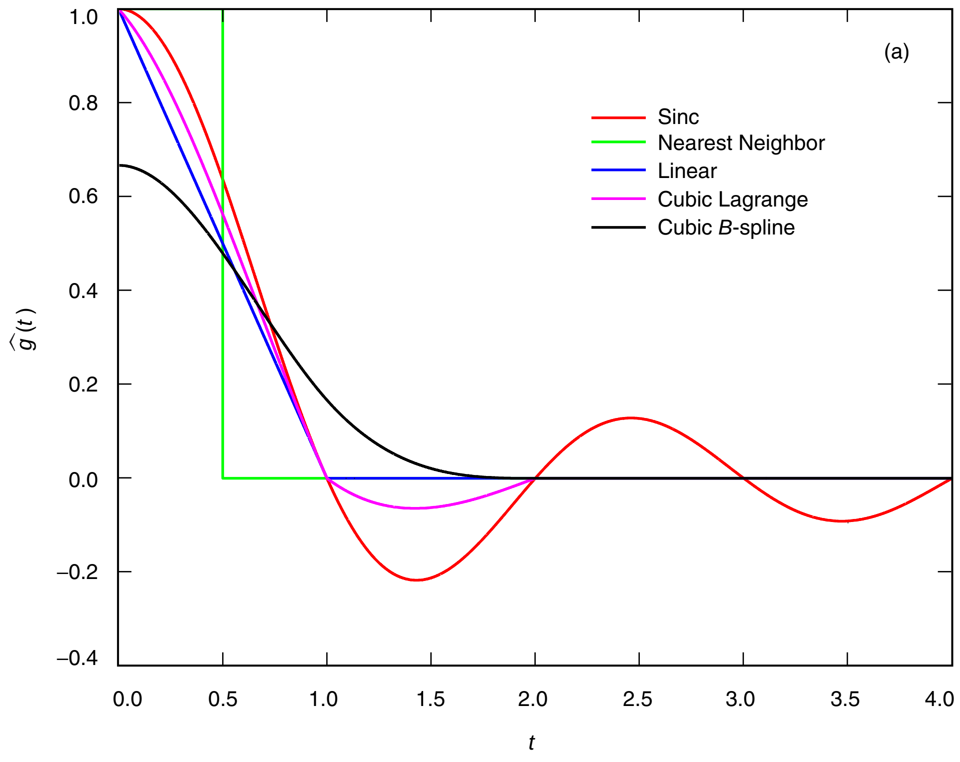
The class of Lagrange piecewise polynomial kernels is derived through application of the Lagrange interpolation formula [6,7]. Briefly, the Lagrange interpolation formula fits an $(M-1)$th-order polynomial to a set of M data points called knots. For example, if $\{t_k, x(t_k)\}_{k=0}^{M-1}$ denotes a set of data points or knots, where we assume $t_k \neq t_\ell$ for all $k \neq \ell$, the Lagrange interpolation formula produces the following $(M-1)$th-order polynomial:

$$y(t) = \sum_{k=0}^{M-1} x(t_k) p_k(t) \tag{10a}$$

where

[3] All of the interpolation kernels considered in the article are real and even, and so only the response for $t \geq 0$ is shown.

[4] Only the magnitude response plots are shown as all of the interpolation kernels here have zero phase response [2,5]. Furthermore, all of the kernels are real, and so the magnitude response plots are even and thus shown only over nonnegative frequency ranges.



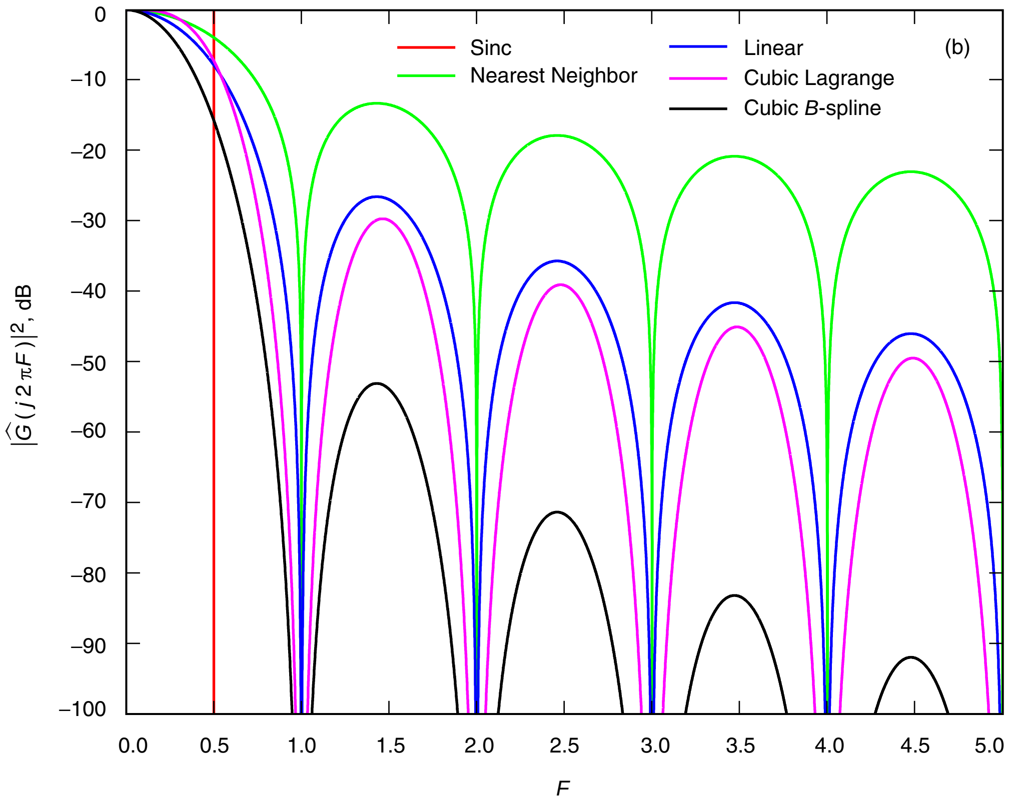


Fig. 4. Plots of several dilated interpolation kernels: (a) time-domain impulse response and (b) frequency-domain magnitude response.

$$p_k(t) \triangleq \prod_{\substack{\ell=0 \\ \ell \neq k}}^{M-1} \frac{t - t_\ell}{t_k - t_\ell} \tag{10b}$$

From Eq. (10b), it can be seen that we have $p_k(t_\ell) = \delta[k - \ell]$, and so $y(t_k) = x(t_k)$ for all t_k. In other words, the polynomial $y(t)$ matches the desired knot value $x(t_k)$ at the desired knot time instant t_k. Furthermore, an interesting thing to note is that the set of polynomials $\{p_k(t)\}$ in the expansion for $y(t)$ from Eq. (10a) depends on only the knot time instants $\{t_k\}$ and not on the knot values $\{x(t_k)\}$.

For the purposes of interpolation kernel design, the desired knot conditions are typically chosen to be the Nyquist(1) criterion constraints given in Eq. (7). In other words, the desired set of knots is usually the following set:

$$\{n, \delta[n]\}_{n \in \mathbb{Z}} \tag{11}$$

As the Lagrange interpolation formula is known to be numerically ill-conditioned as the number of knots increases in general [6,7], what typically is done is to fit a small-order polynomial to the closest knots. Specifically, an Nth-order Lagrange piecewise polynomial dilated interpolation kernel $\widehat{g}(t)$, at every point in time t, fits an Nth-order polynomial to the $(N+1)$ closest knots from the set given in Eq. (11). As the $(N+1)$ closest knots vary in steps according to the value of t, it follows that $\widehat{g}(t)$ in this case will indeed consist of piecewise polynomial segments. This method of construction provides a numerically stable approach to exactly satisfy all of the knot conditions given in Eq. (11).

1. Nearest Neighbor Interpolant. The most trivial example of a Lagrange piecewise polynomial interpolation kernel is the nearest neighbor interpolant, which fits a 0th-order polynomial (i.e., a constant) to the nearest knot from Eq. (11). This definition is somewhat ambiguous in the sense that if we are exactly in between two knot time values (which occurs when t is a half integer), then there is not one unique closest knot point. By convention, similar to the way in which the floor function is defined [2], the knot *to the left* is chosen whenever this ambiguity arises. This results in the following dilated interpolation kernel [3,6,7]:

$$\widehat{g}(t) = \begin{cases} 1, & -\dfrac{1}{2} \leq t < \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The Fourier transform of the nearest neighbor interpolant is by

$$\widehat{G}\left(j2\pi F\right) = \operatorname{sinc}\left(F\right) \tag{13}$$

Plots of $\widehat{g}(t)$ from Eq. (12) and $\left|\widehat{G}(j2\pi F)\right|^2$ from Eq. (13) can be found in Figs. 4(a) and 4(b), respectively. As can be seen, the nearest neighbor interpolant is a very poor approximation to the underlying sinc interpolant kernel corresponding to the true signal model. For the purposes of variable SRC, this interpolant performs satisfactorily only when lowering the sampling rate by a significant factor (see Section VI), since the given set of input samples is highly redundant in this case. Despite this, reconstructing a signal using the model of Eq. (4) and the nearest neighbor interpolant is trivial in that at any instant of time the output is dependent only upon one input sample and involves no computation.

2. Linear Interpolant. After the nearest neighbor interpolant, the next simplest example of a Lagrange piecewise polynomial interpolation kernel is the linear interpolant, which fits a 1st-order polynomial (i.e., a line) to the two nearest knots from Eq. (11). Unlike the nearest neighbor interpolant, there is no ambiguity with regard to the nearest knots at any given instant of time. The dilated interpolation kernel corresponding to the linear interpolant is given by the following expression [3,6,7]:

$$\widehat{g}(t) = \begin{cases} 1 - |t|\,, & |t| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Taking the Fourier transform of Eq. (14) yields the following:

$$\widehat{G}\,(j2\pi F) = \mathrm{sinc}^2(F) \tag{15}$$

Plots of $\widehat{g}(t)$ from Eq. (14) and $\left|\widehat{G}(j2\pi F)\right|^2$ from Eq. (15) can be found in Figs. 4(a) and 4(b), respectively. As can be seen, in both the time and frequency domains, the linear interpolant appears to be a much better approximation to the underlying sinc interpolation kernel than the nearest neighbor interpolant. The improvement over the nearest neighbor interpolant for the purposes of variable SRC is dramatic, as will be shown in Section VI. This improvement comes at the expense of a slight increase in computational complexity, as signal reconstruction using Eq. (4) in general requires two input samples and one time-varying multiplier to compute a single output sample. Here, the time-varying multiplier depends linearly on time.

3. Cubic Lagrange Interpolant. To avoid the ambiguity caused by the non-uniqueness of the nearest knot points from Eq. (11), which occurs iff the order of the piecewise polynomial interpolant is even, the next example of a Lagrange piecewise polynomial interpolation kernel typically used in practice is the cubic interpolant, which fits a 3rd-order polynomial (i.e., a cubic function) to the four nearest knots from Eq. (11). This leads to the following dilated interpolation kernel:

$$\widehat{g}(t) = \begin{cases} 1 - \frac{1}{2}|t| - |t|^2 + \frac{1}{2}|t|^3\,, & |t| < 1 \\ 1 - \frac{11}{6}|t| + |t|^2 - \frac{1}{6}|t|^3\,, & 1 \le |t| < 2 \\ 0, & |t| \ge 2 \end{cases} \tag{16}$$

After much algebraic manipulation, it can be shown that the Fourier transform of the cubic Lagrange interpolant is as follows:

$$\widehat{G}\,(j2\pi F) = \frac{2}{3}\left(\mathrm{sinc}^2(F) - \mathrm{sinc}^2(2F)\right) + \mathrm{sinc}^4(F) = \left[\frac{2}{3}\sin^2(\pi F) + \mathrm{sinc}^2(F)\right]\mathrm{sinc}^2(F) \tag{17}$$

Plots of $\widehat{g}(t)$ from Eq. (16) and $\left|\widehat{G}(j2\pi F)\right|^2$ from Eq. (17) can be found in Figs. 4(a) and 4(b), respectively. As can be seen, in both the time and frequency domains, the cubic Lagrange interpolant appears to be a closer fit to the underlying sinc interpolation kernel than the linear interpolant. Though the cubic Lagrange interpolant outperforms the linear one for the purposes of variable SRC, as we show in Section VI, the improvement is not as dramatic as that of the linear interpolant over the nearest neighbor one. This marginal improvement comes at the expense of a pronounced increase in computational complexity over the linear interpolant. To reconstruct a signal using the model of Eq. (4) requires in general four input samples and three time-varying multipliers to compute a single output sample. Furthermore, these time-varying multipliers have a cubic dependence on time, as opposed to the linearly varying multiplier used for linear interpolation.

In order to achieve a notable improvement in performance suitable for variable SRC, it is useful to consider interpolation kernels that do not satisfy the Nyquist criteria constraints of Eq. (7). This leads to interpolation kernels such as cubic B-splines, which are slightly more computationally complex than the cubic Lagrange interpolant, but which offer a dramatic improvement in performance and an excellent approximation to the underlying sinc interpolation kernel.

C. B-Spline Kernels

The class of B-spline dilated interpolation kernels consists of piecewise polynomial functions that are constructed through repeated convolution with the nearest neighbor interpolant from Eq. (12) [3,8,9]. In particular, the Nth-order B-spline, which we denote here by $\widehat{b}_N(t)$, is defined iteratively as follows [3,8]:

$$\widehat{b}_0(t) \triangleq \begin{cases} 1, & -\dfrac{1}{2} \le t < \dfrac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

$$\widehat{b}_N(t) \triangleq \widehat{b}_{N-1}(t) * \widehat{b}_0(t) = \int_{-\infty}^{\infty} \widehat{b}_{N-1}(\tau)\widehat{b}_0(t-\tau)\,d\tau \tag{19}$$

From Eqs. (18) and (19), it can be shown that $b_N(t)$ can be expressed in closed form as follows [3,4,8]:

$$\widehat{b}_N(t) = \sum_{k=0}^{N+1} \frac{(-1)^k}{N!} \binom{N+1}{k} \left[\left((t-k) + \frac{N+1}{2} \right)_+ \right]^N \tag{20}$$

Here, $(x)_+$ denotes the water-filling function defined as $(x)_+ \triangleq \max\{0, x\}$. Furthermore, from Eqs. (18) and (19), it can be easily shown that the Fourier transform of $\widehat{b}_N(t)$, which we denote here by $\widehat{B}_N(j2\pi F)$, is given by the following expression [3,4,8]:

$$\widehat{B}_N(j2\pi F) = \operatorname{sinc}^{N+1}(F) \tag{21}$$

It can be shown that the Nth-order B-spline $\widehat{b}_N(t)$ enjoys several interesting properties [3,4,8]. For example, it can be shown that $\widehat{b}_N(t)$ is of duration $(N+1)$ and is only nonzero for $-([N+1]/2) \le t < [N+1]/2$. More importantly, it can be shown that $\widehat{b}_N(t)$ is smooth in the sense that it is of regularity $(N-1)$, meaning that it is continuously differentiable $(N-1)$ times [7]. This high degree of smoothness results in B-splines offering a good approximation to the desired underlying sinc interpolant. In fact, it can be shown that B-splines enjoy the maximal order of approximation to the sinc interpolant for a given temporal support over the set of piecewise polynomial interpolation kernels [6,7].

D. Cubic B-Spline Interpolant

The most commonly used B-spline in practice is the cubic B-spline that corresponds to the case $N = 3$. It can be shown from Eq. (20) that the dilated interpolation kernel $\widehat{g}(t)$ has the following simplified form [6]:

$$\widehat{g}(t) = \begin{cases} \frac{2}{3} - \frac{1}{2}|t|^2(2-|t|), & |t| < 1 \\ \frac{1}{6}(2-|t|)^3, & 1 \le |t| < 2 \\ 0, & |t| \ge 2 \end{cases} \tag{22}$$

From Eq. (21), it can be shown that the Fourier transform of the cubic B-spline dilated interpolation kernel is as follows:

$$\widehat{G}(j2\pi F) = \mathrm{sinc}^4(F) \tag{23}$$

Plots of $\widehat{g}(t)$ from Eq. (22) and $\left|\widehat{G}(j2\pi F)\right|^2$ from Eq. (23) can be found in Figs. 4(a) and 4(b), respectively. As can be seen from Fig. 4(a), the cubic B-spline kernel does not satisfy the Nyquist(1) criterion of Eq. (7). Furthermore, it appears at first glance, from looking at the time-domain kernel shape as well as the passband region ($|F| < 1/2$) of Fig. 4(b), as though the spline is not a good approximation to the underlying sinc interpolant.

This, however, is misleading due to the fact that the cubic B-spline is not Nyquist(1). To appropriately compare the cubic B-spline to the sinc interpolant, we must reconstruct the interpolation signal model using Eq. (4) with the cubic B-spline chosen as the kernel, assuming the original samples were obtained by sampling the sinc interpolant. In order to accomplish this, we need to obtain the basis coefficients ($\{c_k\}$ or $c_d[n]$) using Fig. 3. Here, the input sampled sequence $x_d[n]$ corresponds to sampling the dilated sinc interpolant at the Nyquist rate, yielding $x_d[n] = \delta[n]$. To acquire the filter $\widehat{H}_d(z)$ used to obtain the basis coefficients sequence $c_d[n]$ from Fig. 3, we must compute $\widehat{g}_d[n] = \widehat{g}(n)$ using Eq. (22). From Eq. (22), we get

$$\widehat{g}_d[n] = \frac{1}{6}\delta[n+1] + \frac{2}{3}\delta[n] + \frac{1}{6}\delta[n-1] \tag{24}$$

Taking the z-transform of Eq. (24), we get

$$\widehat{G}_d(z) = \frac{1}{6}z + \frac{2}{3} + \frac{1}{6}z^{-1} = \frac{z^2+4z+1}{6z} \tag{25}$$

Hence, from Eqs. (6) and (25), it follows that $\widehat{H}_d(z)$ is given by

$$\widehat{H}_d(z) = \frac{1}{\widehat{G}_d(z)} = \frac{6z}{z^2+4z+1} \tag{26}$$

From Eq. (26), it can be shown that $\widehat{H}_d(z)$ has zeros at $z = 0, \infty$ and poles at $z = -2 \pm \sqrt{3}$. The pole $z = -2+\sqrt{3}$ is inside the unit circle ($|z| = 1$), whereas the pole $z = -2-\sqrt{3}$ is outside. In order to ensure a stable filter $\widehat{H}_d(z)$ (i.e., one whose region of convergence includes the unit circle [2]), it follows that $\widehat{h}_d[n]$ must contain a causal part due to the pole inside the unit circle and an anti-causal part corresponding to the pole outside [2]. Taking the inverse z-transform of Eq. (26) with this information about the region of convergence yields the following:

$$\widehat{h}_d[n] = \sqrt{3}\left[\left(-2-\sqrt{3}\right)^n u[-n-1] + \left(-2+\sqrt{3}\right)^n u[n]\right] \tag{27}$$

where $u[n]$ is the Heaviside unit step function defined as [2]

$$u[n] \triangleq \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

A plot of the impulse response $\widehat{h}_d[n]$ given in Eq. (27) is shown in Fig. 5. As can be seen, the impulse response is real and symmetric and decays very rapidly, due to the fact that the poles of $\widehat{H}_d(z)$ are sufficiently far away from the unit circle. This rapid decay will be useful later in terms of implementing the cubic B-spline in a practical setting, as is discussed in Section V.

To obtain the dilated sinc interpolation kernel approximant, which we denote here by $\widehat{g}_a(t)$, we use the basis coefficients sequence from Fig. 3, which is simply $\widehat{h}_d[n]$ from Eq. (27) in Eq. (4) with $T = T_s = 1$ here. This yields

$$\widehat{g}_a(t) = \sum_{n=-\infty}^{\infty} \widehat{h}_d[n]\widehat{g}(t-n) \tag{28}$$

where $\widehat{g}(t)$ is the cubic B-spline dilated interpolation kernel of Eq. (22). If $\widehat{G}_a(j2\pi F)$ denotes the Fourier transform of $\widehat{g}_a(t)$, then from Eqs. (28), (26), and (23) we have the following:

$$\widehat{G}_a(j2\pi F) = \widehat{H}_d\left(e^{j2\pi F}\right)\widehat{G}\left(j2\pi F\right) = \left(\frac{3}{2+\cos\left(2\pi F\right)}\right)\operatorname{sinc}^4(F) \tag{29}$$

Plots of the impulse responses of the ideal dilated sinc interpolation kernel from Eq. (8) and the cubic B-spline approximant from Eq. (28) are shown in Fig. 6(a). In addition, the corresponding magnitude

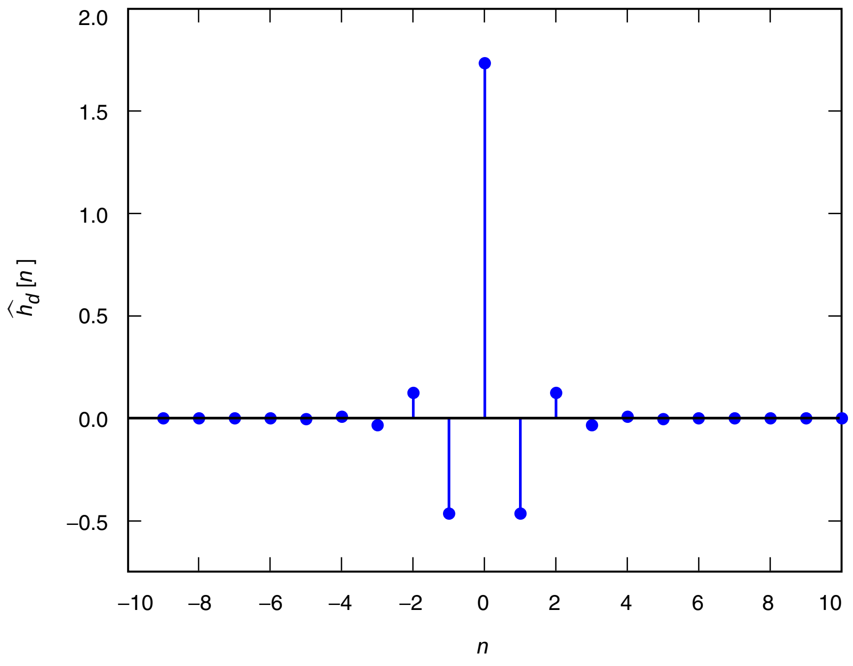


Fig. 5. Impulse response of the filter $\widehat{h}_d[n]$ given in Eq. (27).

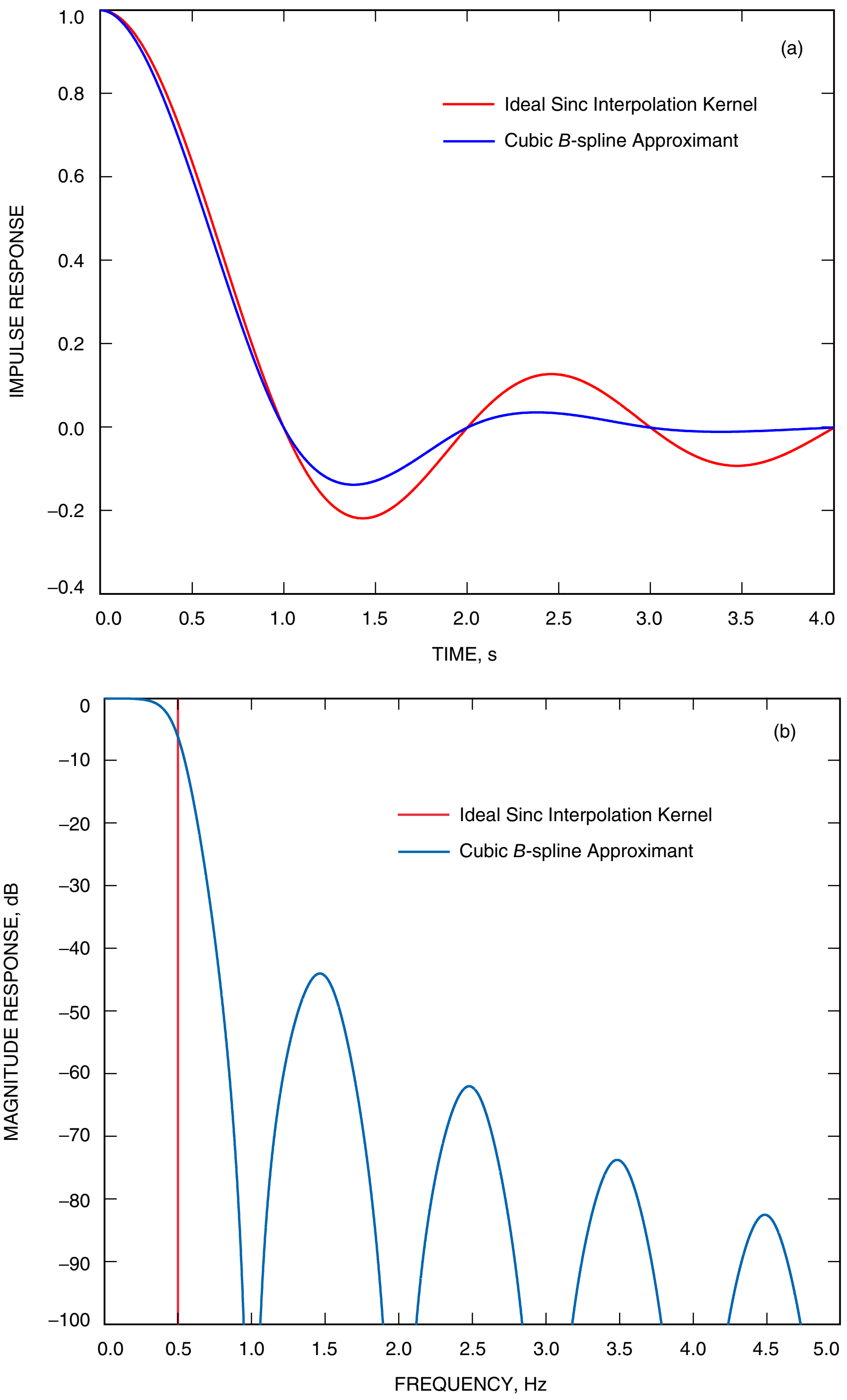


Fig. 6. Plots of the ideal dilated sinc interpolaton kernel and its *B*-spline approximant: (a) time-domain impulse response and (b) frequency-domain magnitude response.

responses from Eqs. (9) and (29) are plotted in Fig. 6(b). Comparing Fig. 4 to Fig. 6, it can be seen that the cubic B-spline effectively offers the closest approximation to the desired underlying sinc interpolant of any of the other kernels considered thus far. This closer fit to the sinc kernel leads to a dramatic improvement over the other interpolation kernels for variable SRC, as is shown in Section VI.

It should be noted that this improvement comes at the cost of a slight increase in overhead compared to the interpolation kernels which satisfy the Nyquist(1) condition. In particular, the input samples (i.e., the sequence $x_d[n]$ from Eq. (5) and Fig. 3) must be prefiltered by the filter $\widehat{h}_d[n]$ to obtain the basis coefficients (i.e., the sequence $c_d[n]$ or the set $\{c_k\}$) to use for signal reconstruction in Eq. (4). Due to the decay of the response of $\widehat{h}_d[n]$ from Eq. (27), as seen from Fig. 5, it is sufficient in practice to implement this filter using a finite impulse response (FIR) truncated version of the infinite impulse response (IIR) given in Eq. (27). In addition to this slight increase in computational overhead, it turns out [6] that signal reconstruction using Eq. (4) for the cubic B-spline requires four input samples and three cubically dependent time-varying multipliers to compute a single output sample.

III. The Variable Sample Rate Conversion Problem

As mentioned in Section I, in order to maintain both a flexibility in data rate as well as the ability to use existing communications blocks for the Advanced Receiver, it becomes necessary to alter the sampling rate of the input sampled signal. Here, the analog signal sampled at the input is assumed to come from an interpolation kernel model such as Eq. (1) or Eq. (4). In particular, we assume that the interpolation kernel is the sinc function from either Eq. (3) or Eq. (8). For this case, if T_{in} denotes the input sampling interval, then the analog signal $x(t)$ input to the ADC is of the following form:

$$x(t) = \sum_{m=-\infty}^{\infty} x(mT_{\text{in}}) \operatorname{sinc}\left(\frac{t}{T_{\text{in}}} - m\right) = \sum_{m=-\infty}^{\infty} x_d[m] \operatorname{sinc}\left(\frac{t}{T_{\text{in}}} - m\right) \tag{30}$$

Here, $x_d[m] \triangleq x(mT_{\text{in}})$ denotes the discrete-time sampled signal obtained from the ADC.

To alter the sampling interval to some other output value, say T_{out}, we would like to generate the discrete-time sequence $y_d[n] \triangleq x\big((n+\epsilon)T_{\text{out}}\big)$, where ϵ is some fractional sampling offset parameter which satisfies $0 \le \epsilon < 1$. In order to obtain $y_d[n]$, from Eq. (30), we must calculate the following expression:

$$\begin{aligned} y_d[n] = x\big((n+\epsilon)T_{\text{out}}\big) &= \sum_{m=-\infty}^{\infty} x_d[m] \operatorname{sinc}\left(\left(\frac{T_{\text{out}}}{T_{\text{in}}}\right)(n+\epsilon) - m\right) \\ &= \sum_{m=-\infty}^{\infty} x_d[m] \operatorname{sinc}\left(\left(\frac{F_{\text{in}}}{F_{\text{out}}}\right)(n+\epsilon) - m\right) = \sum_{m=-\infty}^{\infty} x_d[m] \operatorname{sinc}\big(\rho(n+\epsilon) - m\big) \end{aligned} \tag{31}$$

Here, $F_{\text{in}} \triangleq 1/T_{\text{in}}$ and $F_{\text{out}} \triangleq 1/T_{\text{out}}$ denote, respectively, the input and output sampling rates, whereas $\rho \triangleq F_{\text{in}}/F_{\text{out}}$ denotes the sampling rate conversion factor.

It can be seen from Eq. (31) that in order to calculate even a single output sample of $y_d[n]$, we must compute the sum in Eq. (31) over the entire set of input samples $\{x_d[m]\}$. Furthermore, for the purposes of variable SRC in which the parameter ρ is subject to changing, the sinc terms in Eq. (31) must be changed according to the value of ρ. As this cannot be done practically, one alternative could be to truncate several of the terms in Eq. (31). However, as the sinc function decays inverse linearly (i.e., sinc(x) decays as $1/x$), a large number of terms must be kept in order to minimize degradation effects due to truncation.

Instead, what is typically done to achieve variable SRC is to model the continuous-time input as coming from an interpolation kernel signal model with a different kernel than the sinc function [10–12]. Specifically, the input analog signal $v(t)$ is modeled using Eq. (4) as follows:

$$v(t) = \sum_{k=-\infty}^{\infty} c_k \widehat{g}\left(\frac{t}{T_{\text{in}}} - k\right) = \sum_{m=-\infty}^{\infty} c_d[m]\, \widehat{g}\left(\frac{t}{T_{\text{in}}} - m\right) \tag{32}$$

Here, $c_d[m]$ is the sequence of basis coefficients obtained using Fig. 3 [i.e., we have $C_d(z) = \widehat{H}_d(z)X_d(z)$ as in Eq. (6)] and $\widehat{g}(t)$ is some dilated interpolation kernel. To alter the sampling interval to T_{out}, we compute $y_d[n] \triangleq v\big((n+\epsilon)T_{\text{out}}\big)$ similarly to what was done above. From Eq. (32), we get the following:

$$y_d[n] = \sum_{m=-\infty}^{\infty} c_d[m]\, \widehat{g}\big(\rho\,(n+\epsilon) - m\big) \tag{33}$$

Through sagacious choice of the interpolation kernel $\widehat{g}(t)$, we can achieve variable SRC using Eq. (33) both in a computationally feasible manner and without significant degradation compared to the underlying sinc model given in Eq. (31). For example, using the piecewise polynomial kernels discussed in Sections II.B and II.C, a single output sample $y_d[n]$ from Eq. (33) will require only a small finite number of basis coefficient samples $c_d[m]$. Furthermore, for piecewise polynomial kernels, the output $y_d[n]$ can be computed efficiently using the Farrow structure [11–13], as is discussed in Section V.B.

The formula in Eq. (33) is the fundamental variable SRC equation. It can be seen that the sinc SRC relation from Eq. (31) is just a special case of Eq. (33) with $c_d[m] = x_d[m]$ and $\widehat{g}(t) = \text{sinc}(t)$. At this point, it is insightful to analyze the spectral properties of the variable SRC output sequence from Eq. (33).

A. Spectral Properties of the Variable SRC Output

After much laborious algebraic manipulation, it can be shown that the discrete-time Fourier transform of $y_d[n]$ from Eq. (33), which we denote here by $Y_d(e^{j\omega})$, is as follows:

$$Y_d(e^{j\omega}) = \frac{1}{\rho}\sum_{k=-\infty}^{\infty} \widehat{G}\left(j\left(\frac{\omega - 2\pi k}{\rho}\right)\right) C_d\left(e^{j((\omega-2\pi k)/\rho)}\right) e^{j(\omega-2\pi k)\epsilon} \tag{34}$$

Here $\widehat{G}\,(j\Omega)$ denotes the continuous-time Fourier transform of $\widehat{g}(t)$ and $C_d(e^{j\omega})$ denotes the discrete-time Fourier transform of $c_d[n]$. Setting $\omega = 2\pi f$ in Eq. (34) to obtain a normalized frequency representation yields the following:

$$Y_d\big(e^{j2\pi f}\big) = \frac{1}{\rho}\sum_{k=-\infty}^{\infty} \widehat{G}\left(j2\pi\left(\frac{f-k}{\rho}\right)\right) C_d\left(e^{j2\pi((f-k)/\rho)}\right) e^{j2\pi(f-k)\epsilon} \tag{35}$$

From Eq. (35), it can be seen that the output spectrum $Y_d\big(e^{j2\pi f}\big)$ contains the scaled, linearly warped, fractionally delayed spectrum $(1/\rho)\widehat{G}\big(j(2\pi f/\rho)\big)C_d\big(e^{j(2\pi f/\rho)}\big)\, e^{j2\pi f\epsilon}$ plus alias terms shifted by all possible integer amounts in frequency.

An interesting observation can be made for the special case of the sinc SRC relation from Eq. (31). Suppose in this case that $x_d[n] = c_d[n]$ is band-limited to $f = f_{BL}$, meaning that $X_d\left(e^{j2\pi f}\right) = 0$ for $|f| \geq f_{BL}$ for some f_{BL} with $0 \leq f_{BL} < 1/2$ and $f \in [-1/2, 1/2)$. Furthermore, suppose that $\epsilon = 0$ and $f_{BL} < 1/2\rho$, where we assume $\rho \geq 1$ here. Then, in this case, from Eqs. (35) and (9), we have

$$Y_d\left(e^{j2\pi f}\right) = \frac{1}{\rho} X_d\left(e^{j(2\pi f/\rho)}\right), \quad f \in \left[-\frac{1}{2}, \frac{1}{2}\right) \tag{36}$$

and so $Y_d\left(e^{j2\pi f}\right)$ is simply a scaled and zoomed-in version of $X_d\left(e^{j2\pi f}\right)$, where the magnification factor of the zoom is the sampling rate conversion factor ρ. An illustration of the phenomenon exhibited in Eq. (36) is shown in Fig. 7.

Traditionally in digital signal processing, the process of zooming in and out of spectra is accomplished through the use of multirate systems such as decimation/interpolation[5] filter systems [5]. The zoom factor in these cases can be only a rational number and, in general, such zooming in/out can be accomplished only through fractional decimation/interpolation filter systems.

Despite this restriction, such decimation/interpolation systems are useful in a practical setting, where the desired signal of interest (SOI) present in the sampled sequence $x_d[n]$ is superimposed with noise and interference. In this case, these systems can simultaneously alter the sampling rate and remove out-of-band artifacts [5]. It should be noted that this is in contrast to the generic variable SRC scheme described by the formula in Eq. (33), which can alter only the sampling rate.

For large sampling-rate conversion factors, this naturally leads to the notion of two stages of SRC. In the first stage, the bulk of the rate is lowered by a generally fractional decimation filter system to remove out-of-band artifacts. Then, in the second stage, with the majority of these artifacts removed, the rate is lowered to its precise value using the variable SRC scheme of Eq. (33) with a suitably chosen interpolation kernel.

IV. Proposed Methods for Variable SRC

As mentioned above, the presence of out-of-band artifacts can lead to deleterious effects on SRC for the desired SOI. In order to mitigate the effects of these artifacts, we propose here to first use a fractional decimation filter system to simultaneously lower the sampling rate near the target rate while removing alias components due to the artifacts. Afterwards, we propose to use a fine interpolation scheme using the methods described in Sections II and III to precisely adjust the rate to its desired value, assuming that a sufficient amount of the out-of-band artifacts has been removed. This two-stage variable SRC scheme is illustrated in Fig. 8(a) along with a possible set of input and desired output spectra in Fig. 8(b).

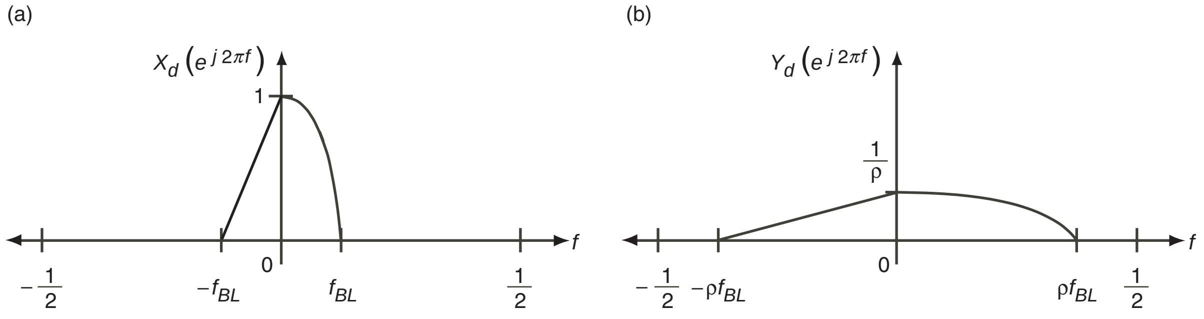


Fig. 7. Illustration of the frequency zooming property of variable SRC using the sinc interpolation kernel: (a) input spectrum and (b) output spectrum after SRC.

[5] The interpolation filter systems mentioned in [5] bear little relation to the interpolation schemes discussed in Section II.

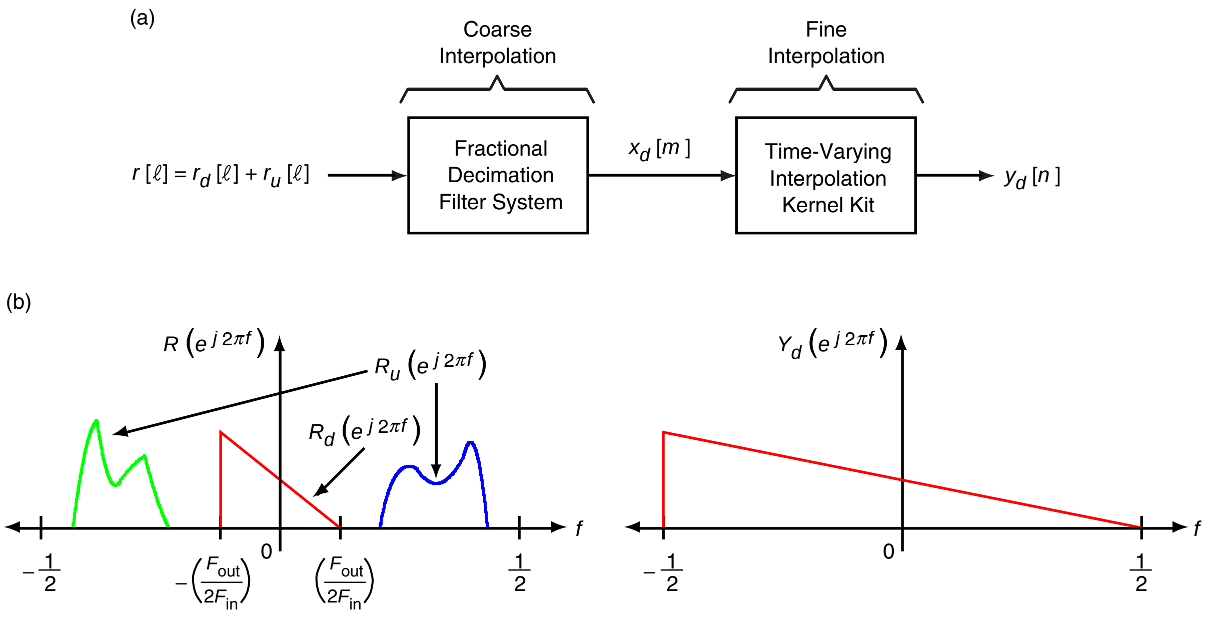


Fig. 8. Method for achieving variable SRC in the presence of out-of-band artifacts: (a) proposed two-stage variable SRC scheme and (b) a possible set of input and desired output spectra.

In Fig. 8(a), the input sampled signal $r[\ell]$ consists of a desired SOI component $r_d[\ell]$ superimposed with an undesired component $r_u[\ell]$ that may contain out-of-band artifacts as in the example of Fig. 8(b). The purpose of the fractional decimation filter system is to remove the undesired component while coarsely lowering the sampling rate to the desired amount. Once the undesired component is suitably suppressed, the sampling rate is finely adjusted to its desired value using a fit to an interpolation kernel model as in Eq. (33).

In this section, we present several methods for carrying out the coarse interpolation stage from Fig. 8(a). There, the pros and cons of each method are addressed with regard to the fine interpolation step carried out subsequently as well as to the variable SRC problem as a whole. For all cases, it is assumed that the desired SOI is centered at baseband (zero frequency) as in the example shown in Fig. 8(b).

A. Single-Stage Adjustable Filter

As will be shown in Section VI, the interpolation schemes of Section II provide a closer fit to the underlying sinc interpolant at the output sample rate in general when the sampling rate conversion factor $\rho = F_{\text{in}}/F_{\text{out}}$ is large. Intuitively, this means that, if the input sample rate is very large compared to the desired output rate, the samples will appear to be very redundant in the absence of out-of-band artifacts. As a result of this redundancy, it will be easier to match the underlying sinc interpolant at the lower rate samples in this case.

Referring to Fig. 8(b), one way to preserve the inherent redundancy of the desired SOI embedded in the fullband process $r[\ell]$ while removing the undesired out-of-band artifacts is to simply filter out the undesired component. This is illustrated in Fig. 9(a), with the desired characteristics of the filter $H(z)$ shown in Fig. 9(b).

With reference to Fig. 8(b), the cutoff frequency f_c of the lowpass filter $H(z)$ from Fig. 9 should be greater than $F_{\text{out}}/2F_{\text{in}}$ but small enough to eliminate the residual out-of-band component $r_u[\ell]$. As ideal bandpass filters are unrealizable [2,5], in practice $H(z)$ would be implemented using a linear phase FIR filter approximation, such as an equiripple design obtained using the McClellan–Parks algorithm [2,5].

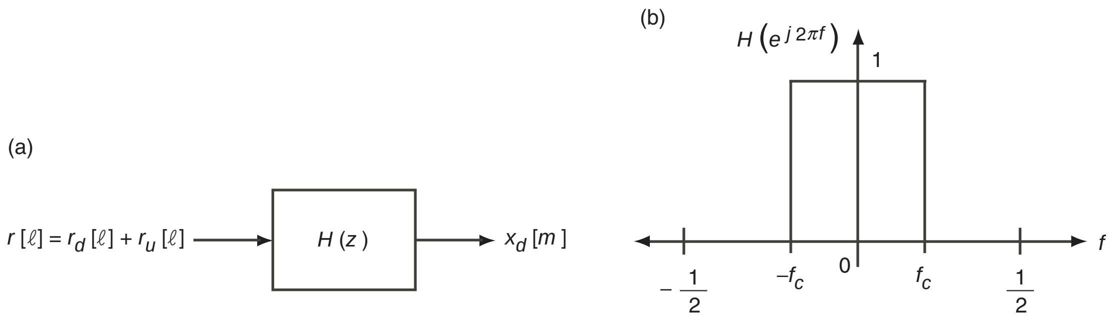


Fig. 9. Single-stage adjustable filter system to remove out-of-band artifacts: (a) block diagram and (b) desired frequency response characteristics of the filter $H(z)$.

The single-stage filter scheme of Fig. 9 is advantageous because it is conceptually a simple method to process the input signal prior to the fine interpolation step since only one stage of filtering is required. Furthermore, as the inherent redundancy of the SOI is preserved in this case, this facilitates the task of the fine interpolator. As a result of this redundancy, lower-order and lower-complexity fine interpolation schemes such as the nearest neighbor or linear interpolants from Section II may yield sufficiently satisfactory SRC performance.

Despite these advantages, from a practical point of view, the single-stage filter scheme is not well-suited for preprocessing prior to fine interpolation. Since the filtering must be done at the high input sample rate F_{in}, this scheme is computationally intensive, as a large number of filter taps will be required to implement $H(z)$ in general. A more important problem is the fact that the spectral characteristics of the filter itself must be changed for different output sample rates F_{out}. Calculating the filter taps necessary to implement a filter with a different cutoff frequency is a very computationally intensive process that cannot be done practically using field programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs), such as those expected to be used for the Advanced Receiver. As a result, the single-stage adjustable filter scheme of Fig. 9 is not well-suited for the problem of variable SRC in the presence of out-of-band artifacts.

B. Single-Stage Adjustable Fractional Decimation Filter

Traditionally, in digital signal processing, to achieve SRC by a rational factor, a fractional decimation filter system is used [5]. This scheme is illustrated in Fig. 10(a) along with the desired characteristics of the filter $H(z)$ shown in Fig. 10(b). If the input sample rate is F_{in}, then the output sample rate is $(L/M)\,F_{\text{in}}$ [5], which can be made as close to the target sample rate of F_{out} as desired through proper choice of L and M.

The single-stage fractional decimation filter scheme of Fig. 10 has several advantages. As the output sample rate $(L/M)\,F_{\text{in}}$ can, in theory, be designed arbitrarily close to the desired output rate of F_{out}, this can make the subsequent fine interpolation stage from Fig. 8(a) unnecessary. Furthermore, through the use of a polyphase decomposition [5] of the filter $H(z)$, all effective filtering can be accomplished at the lower output sampling rate of $(L/M)\,F_{\text{in}}$.

Unfortunately, there are also several disadvantages of using the single-stage fractional decimation filter system for variable SRC. Similarly to the situation with the single-stage adjustable filter scheme described in Section IV.A, both the rate parameters L and M as well as the spectral characteristics of the filter $H(z)$ need to be reconfigured for different output sampling rates F_{out}. As calculating filter taps tailored for different output sampling rates is not feasible using FPGAs or ASICs, this fractional decimation filter scheme cannot be used in practice. Furthermore, as the value of the decimation factor M increases, the filter design problem becomes more ill-conditioned [2,5], since the transition bandwidth becomes smaller

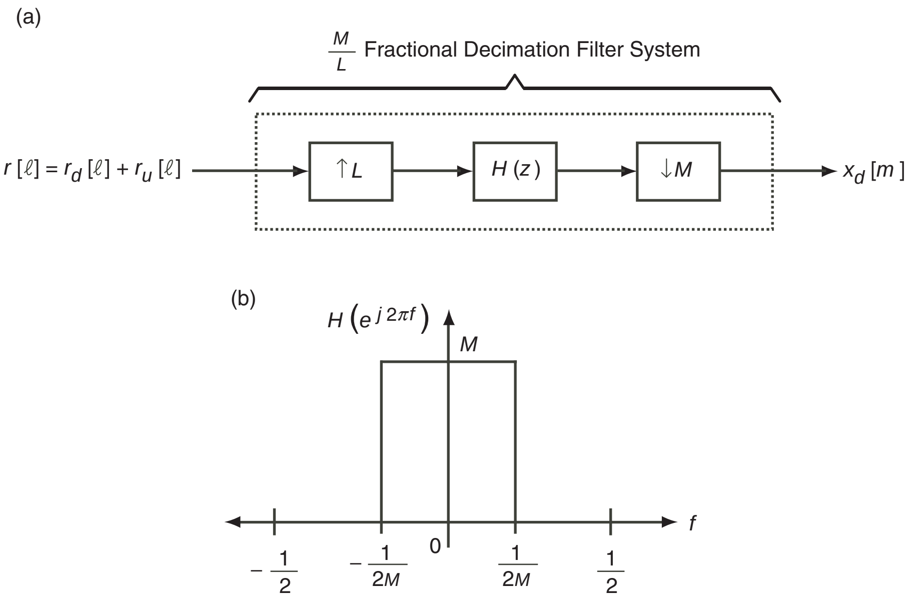


Fig. 10. Single-stage adjustable fractional decimation filter system to remove out-of-band artifacts: (a) block diagram and (b) desired frequency response characteristics of the filter $H(z)$.

and smaller, which leads to an increased number of filter taps required to perform the filtering. As the number of taps required varies inverse linearly with the transition bandwidth [2,5], the number of taps required grows to the point where the filter design problem becomes numerically ill-conditioned.

C. Multi-Stage Fixed Fractional Decimation Filter

To overcome the practicality issues of having to reconfigure filter coefficients to accommodate a variety of output sample rates, one approach that can be taken is to use a *fixed* filter architecture an *adjustable* number of times. One example of this approach is shown in Fig. 11. In this setting, a fixed M/L fractional decimation system, implemented as in Fig. 10, is used an adjustable K number of times.

The multi-stage fractional decimation filter system described in Fig. 11 enjoys many advantageous properties. As opposed to the adjustable designs considered in Sections IV.A and IV.B, this multi-stage system does not require filter reconfigurability, as all filter taps used can be precomputed and hard wired into the Advanced Receiver. The only parameter that needs to be adjusted is the number of stages K to be used to lower the sampling rate. Furthermore, as the rate parameters L and M are fixed, these can be easily chosen to yield a numerically well-conditioned filter design. In addition, the filtering at each stage can be carried out efficiently at a rate of L/M times the input rate by exploiting the polyphase decomposition of the fixed filter.

One of the disadvantages of the multi-stage fractional decimation filter system is that the sample rate at the output is confined to the discrete logarithmically uniform values $F_{\text{out}} = (L/M)^K F_{\text{in}}$. For

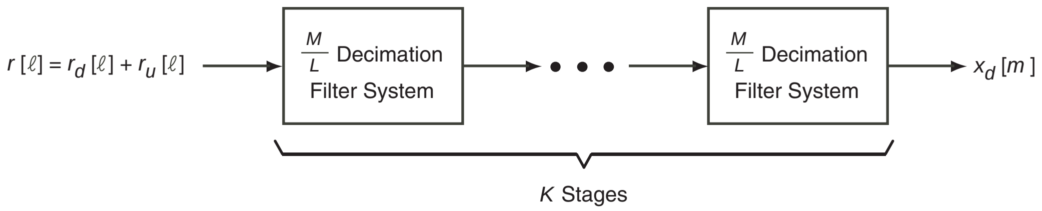

$r[\ell] = r_d[\ell] + r_u[\ell]$
$\frac{M}{L}$ Decimation Filter System
$\frac{M}{L}$ Decimation Filter System
$x_d[m]$
K Stages

Fig. 11. Multi-stage fixed fractional decimation filter system with an adjustable number of stages.

the purposes of variable SRC, this will result in a residual amount of out-of-band artifacts aliasing into the desired SOI at the fine interpolation stage of Fig. 8(a). Through conservative choice of L and M, however, this effect can be minimized, but at the expense of a greater system latency when the number of stages K becomes large. In addition to this aliasing issue, most of the inherent redundancy of the SOI will be removed at the output of the system of Fig. 11, making the subsequent fine interpolation stage of Fig. 8(a) more problematic. This will necessitate the use of higher-order, higher-complexity fine interpolation schemes such as the cubic Lagrange or cubic B-spline interpolants of Section II to yield sufficient reconstruction for SRC.

Notwithstanding its disadvantages, the multi-stage fixed fractional decimation filter scheme of Fig. 11 is well-suited for the purposes of coarse interpolation for variable SRC. Its overall lack of reconfigurability and efficient implementation make it a fitting system to use for coarse interpolation.

D. Multi-Stage Fixed Cascaded Expanded Filter

One of the disadvantages of the multi-stage fractional decimation filter system of Section IV.C is that it removes the inherent redundancy of the SOI, making the subsequent fine interpolation stage from Fig. 8(a) more prone to errors. In order to preserve this redundancy, one approach that can be taken is to not lower the rate of the input signal, as in the method of Section IV.C, and instead to perform all filtering at the high input sample rate. This approach is similar to the single-stage filter scheme presented in Section IV.A, with the exception that here the filter structure will remain fixed, as desired in a practical implementation.

In order to obtain a multi-stage filtering scheme as in Section IV.C without lowering the sample rate at each stage, we take the expander factor L from Fig. 11 to be $L = 1$ and apply the noble identities [5] to move each filter $H(z)$ to the left of each decimator M. Finally, to keep the sample rate high, we remove the effective decimation factor M^K obtained after applying the noble identities. This results in the structure shown in Fig. 12.

As in Section IV.B, the filter $H(z)$ is designed to have the spectral characteristics as shown in Fig. 10(b) for a *fixed* choice of M. This ensures that the cascaded response $H_{\text{casc}}(z)$ given by

$$H_{\text{casc}}(z) = \prod_{k=0}^{K-1} H\left(z^{M^k}\right)$$

is lowpass with cutoff frequency $f_c = 1/2M^K$. (As the decimators have been removed from the system, $H(z)$ should have a passband gain of unity (i.e., 1 instead of M) to prevent numerical instability issues from arising.) It should be noted here that the expanded filter response $H\left(z^{M^k}\right)$ is implemented in the same way as $H(z)$, with the exception that each delay element z^{-1} is replaced by the expanded delay element z^{-M^k}.

The multi-stage cascaded expanded filter system from Fig. 12 exhibits several advantages. Like the multi-stage scheme of Section IV.C, the filter structure is fixed for a prescribed M and so all filter

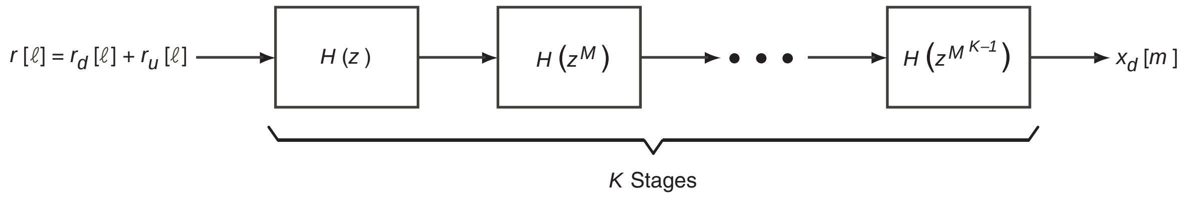


Fig. 12. Multi-stage fixed cascaded expanded filter with an adjustable number of stages.

coefficients can be precomputed and stored in the Advanced Receiver. As before, the only parameter that needs to be adjusted is the number of stages K to be used. Furthermore, as all filtering is performed at the high input sample rate, all of the inherent redundancy of the SOI is preserved, making the subsequent fine interpolation stage of Fig. 8(a) less problematic.

Despite these advantages, though, there are several disadvantages of using the multi-stage cascaded expanded filter system. One disadvantage is that the bandwidth of the output is confined to the discrete logarithmically uniform values $1/M^K$. This is in fact even more stringent than the sample rate output values achievable using the multi-stage decimation filter approach from Section IV.C. For this reason, in a practical application, M should be chosen to be small (for example, $M = 2$) in order to allow for more variety in output bandwidth. Another disadvantage is that implementing an expanded response $H\big(z^{M^k}\big)$ requires more care in hardware than implementing a stage of fractional decimation from the scheme of Section IV.C. This is due to the fact that each delay element z^{-1} from $H(z)$ must be implemented as z^{-M^k} to obtain $H\big(z^{M^k}\big)$. Furthermore, the filter latency is in general larger than the multi-stage fractional decimation scheme of Section IV.C.

In spite of these disadvantages, the multi-stage cascaded expanded filter scheme of Fig. 12 is well-suited for removing out-of-band artifacts for variable SRC. The fact that it does not require filter reconfiguration and preserves the redundancy of the SOI makes it quite a valuable system to use prior to fine interpolation.

V. Implementation Issues

There are several important practical considerations to take into account when implementing the proposed two-stage variable SRC system described in Section IV and Fig. 8(a). In this section, we focus individually on some of the unique implementational aspects pertaining to the coarse and fine interpolation stages.

A. Coarse Interpolation Implementation Issues

All of the methods proposed for coarse interpolation require the implementation of one or more lowpass filters. As mentioned in Section IV, since the ideal lowpass characteristics desired are not realizable, these filters typically are implemented as linear phase FIR filters in practice [2,5]. The linear phase property, which can be practically implemented only with FIR filters [2,5], ensures that the phase spectral characteristics of the input signal will not incur any distortion, as desired. In addition to this, the FIR property ensures that the resulting filter will not suffer from numerical instability problems such as limit cycles [2,5], which can occur with IIR filters.

Typically, the filters are designed to satisfy the equiripple property [2,5], which, by the alternation theorem, can be shown to be equivalent to minimizing the maximum ripple size in the individual bands of interest (i.e., the passband and stopband for lowpass filters). The coefficients of such filters can be obtained using the McClellan–Parks program [2,5], which in turn uses the Remez exchange algorithm from optimization theory.

As with other lowpass filter design methods, the order or number of filter tap multipliers required for equiripple filters to achieve a given set of ripple sizes is inversely proportional [2,5] to the transition bandwidth of the filter, which is the difference between the stopband and passband frequencies. In other words, if f_P and f_S denote, respectively, the passband and stopband frequencies of the lowpass filter, and N_f denotes the required order, then we have [2,5]

$$N_f \propto \frac{1}{f_S - f_P} = \frac{1}{\Delta f}$$

where $\Delta f \triangleq f_S - f_P$ denotes the transition bandwidth of the lowpass filter to be designed.

As far as the variable SRC problem is concerned, this means that small bandwidth filters will require a larger filter order or number of multipliers. Since coefficient multiplications account for the majority of the processing time required to implement a filter, it is insightful here to investigate the number of such multipliers needed to carry out the coarse interpolation methods proposed in Section IV.

To properly compare the multiplication computational complexity of the coarse interpolation schemes of Section IV, an appropriate metric is the number of multiplications required per input sampling interval, which we denote here by ν. This figure of merit is appropriate in that filtering at a lower rate is more advantageous than filtering at the high input sample rate, from a computational overhead point of view. For example, suppose that N_f multipliers are required to implement the filter $H(z)$ for both the single-stage adjustable filter system of Section IV.A as well as the single-stage adjustable fractional decimation filter method of Section IV.B. Then, assuming the fractional decimation filter (FDF) system has been implemented efficiently using the polyphase decomposition [5] of $H(z)$, we have the following:

$$\nu_{\mathrm{F}} = \frac{N_f}{T_s} \qquad \text{(single-stage filter)}$$

$$\nu_{\mathrm{FDF}} = \frac{\left(\dfrac{L}{M}\right) N_f}{T_s} = \left(\frac{L}{M}\right) \nu_{\mathrm{F}} \qquad \text{(single-stage fractional decimation filter)}$$

Clearly we have $\nu_{\mathrm{FDF}} < \nu_{\mathrm{F}}$ as $L < M$ here, since we are assuming to be lowering the sample rate in this setting. This is intuitively consistent in the sense that the filtering of the single-stage filter must be accomplished at the high input sample rate, whereas the filtering of the single-stage fractional decimation filter can be done at the lowered sample rate (in this case L/M of the input sample rate).

Intuitively, it is also clear that ν should depend upon the amount by which the output sample rate differs from the input sample rate. A plot of the multiplication computational complexity ν as a function of the sampling rate conversion ratio[6] $\mathcal{R} \triangleq F_{\mathrm{out}}/F_{\mathrm{in}}$ is shown in Fig. 13. For the sake of fair comparison, in all of the plots shown in Fig. 13, the filter for each method at every SRC ratio $\mathcal{R}$ was designed to have the same passband/stopband ripple size and a transition bandwidth commensurate with the cutoff frequency f_c of the filter.

Several important observations can be made from Fig. 13. First of all, with the exception of the single-stage adjustable filter, whose bandwidth varies linearly with the SRC ratio $\mathcal{R}$, all of the multiplication complexity factors ν are piecewise steps as a function of $\mathcal{R}$. This is because all other coarse interpolation schemes are carried out over a discrete number of stages (i.e., the multi-stage schemes of Sections IV.C and IV.D) or over a prescribed resolution granularity (i.e., the single-stage fractional decimation filter of Section IV.B). Hence, for any one of these individual schemes, for a given value of the SRC ratio $\mathcal{R}$ there is a region of ratios in the neighborhood of $\mathcal{R}$ over which the same filter architecture applies, and so the same multiplication complexity results.

In addition to this, another observation that can be made from Fig. 13 is that the single-stage fractional decimation filter scheme of Section IV.B is the only coarse interpolation method whose complexity increases with the SRC ratio $\mathcal{R}$. Intuitively, the reason for this is that this scheme is the only one that can be tailored to completely perform the required variable SRC task. To see this, recall from above that the multiplication complexity ν is a function of the resolution granularity, which is simply the decimation parameter M from Fig. 10. Using the polyphase identity [5], this parameter can be restricted to be a

[6] The sampling rate conversion ratio $\mathcal{R}$ is simply the reciprocal of the sampling rate conversion factor ρ from Section III. As we assume $1 \leq \rho < \infty$ here, it follows that $0 < \mathcal{R} \leq 1$.

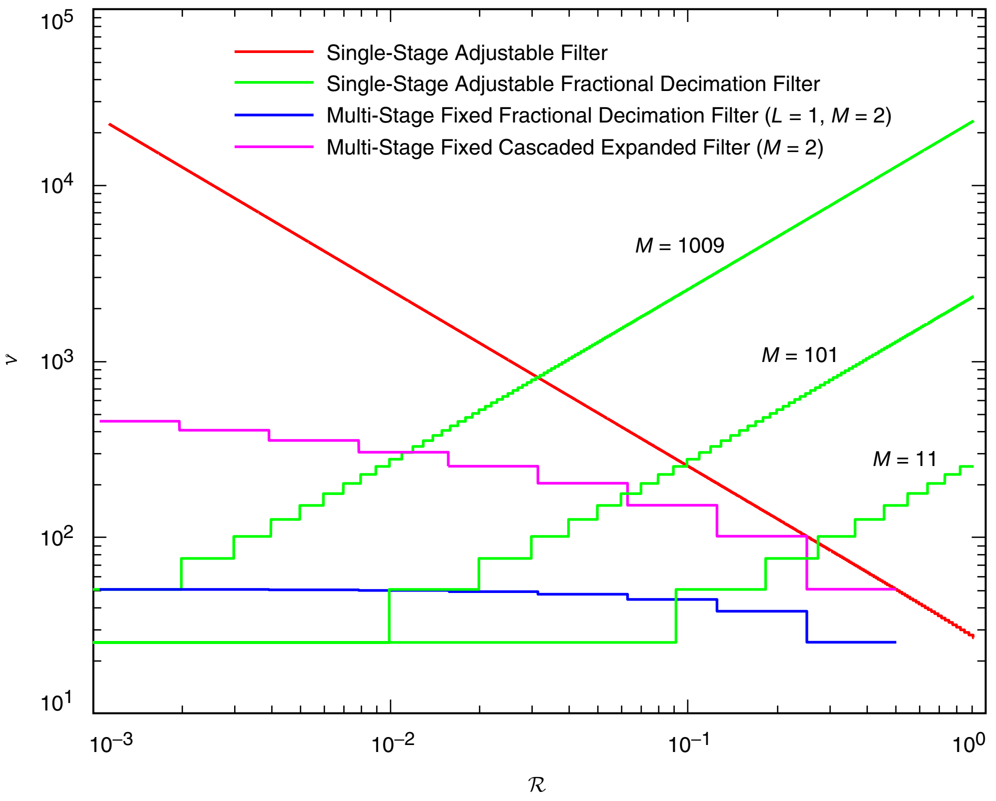


Fig. 13. Multiplication computational complexity (i.e., multiplications per input sample rate required) as a function of the sampling rate conversion ratio for the coarse interpolation schemes.

prime number without loss of generality. By using an adjustable value of the expansion parameter L from Fig. 10, we generate a set of exact possible SRC ratios of the form m/M for $1 \leq m \leq M - 1$. Clearly as M increases, the number of possible exact SRC ratios increases and the filter scheme is said to be less granular. Intuitively, as the granularity decreases, the need for the subsequent fine interpolation step for variable SRC from Fig. 8(a) decreases as well. This, however, comes at the expense of a much larger complexity, as evidenced in Fig. 13. For example, from Fig. 13, in order to achieve an output SRC ratio resolution accurate to at least one, two, or three decimal places, we need $M = 11$, 101, or 1009, respectively. This order-of-magnitude increase in output SRC ratio resolution comes at the expense of an order-of-magnitude increase in multiplication complexity, as seen in Fig. 13.

A final observation that can be made regarding Fig. 13 is that the adjustable coarse interpolation methods of Sections IV.A and IV.B are in general much greater in computational complexity than the fixed methods of Sections IV.C and IV.D. This brings to light the impracticality of the adjustable methods for use in the Advanced Receiver. In addition to requiring the computation of different filter coefficients for different values of the SRC ratio $\mathcal{R}$, these methods also require more memory and resource allocation to implement these filters. Though the fixed multi-stage coarse interpolation methods are not guaranteed to completely remove out-of-band artifacts, they can perform satisfactorily (see Section VI) and with low computational complexity.

B. Fine Interpolation Implementation Issues

All of the coarse interpolation schemes presented in Section IV require the design and implementation of linear time invariant (LTI) [2,5] filters. In contrast to this, the fine interpolation stage of Fig. 8(a) requires the implementation of the linear time-varying (LTV) [2,5] filtering operation given in Eq. (33).

Implementing such filters is difficult in general in that all filter taps vary with time. However, for the special case in which the interpolation kernel is a piecewise polynomial function, the LTV filtering operation from Eq. (33) can be efficiently realized using a combination of LTI filters and a small set of time-varying multipliers. The resulting architecture is known as the Farrow structure [11–13] and is the focus of this section.

Recall from Eq. (33) that we wish to compute the following expression:

$$y_d\,[n] = \sum_{m=-\infty}^{\infty} c_d\,[m]\,\widehat{g}\big(\rho\,(n+\epsilon) - m\big)$$

To obtain a computationally efficient rendition of the above expression, we decompose the time argument $\rho\,(n+\epsilon)$ into integer and fractional parts. In particular, we decompose $\rho\,(n+\epsilon)$ as

$$\rho\,(n+\epsilon) = q + r \tag{37}$$

where we have

$$\begin{aligned} q &= \big\lfloor \rho\,(n+\epsilon) \big\rfloor \\ r &= \big[\rho\,(n+\epsilon)\big]\,\mathrm{mod}1 \end{aligned} \tag{38}$$

Substituting Eq. (37) into Eq. (33) yields

$$y_d[n] = \sum_{m=-\infty}^{\infty} c_d[m]\widehat{g}\big((q-m)+r\big) = \sum_{k=-\infty}^{\infty} c_d[q-k]\widehat{g}(k+r) \tag{39}$$

From this point onward, we assume that $\widehat{g}\,(t)$ is an Nth-order piecewise polynomial function over the integers. It should be noted that all of the dilated interpolation kernels presented in Section II satisfy this condition, with the exception of the sinc and nearest neighbor interpolants. As $\widehat{g}\,(t)$ is an Nth-order polynomial over any consecutive set of integers, it follows that we have

$$\widehat{g}\,(k+r) = C_0[k]\,(k+r)^0 + C_1[k]\,(k+r)^1 + \cdots + C_N[k]\,(k+r)^N, \quad \forall\; k \in \mathbb{Z}, \quad 0 \le r < 1 \tag{40}$$

where $\{C_\ell[k]\}_{\ell=0}^{N}$ denotes the set of polynomial coefficients of $\widehat{g}(t)$ over the interval $t \in [k, k+1)$. An important observation here is that the polynomial coefficients depend only on the integer index k and not on the fractional parameter r. Expanding the binomial terms in Eq. (40) using the binomial theorem yields the following equivalent form for $\widehat{g}\,(k+r)$:

$$\widehat{g}\,(k+r) = \sum_{\ell=0}^{N} a_\ell[k]\,r^\ell \tag{41a}$$

where

$$a_\ell[k] = \sum_{m=\ell}^{N} \binom{m}{\ell} C_m[k]\, k^{m-\ell} \tag{41b}$$

Substituting Eq. (41) into Eq. (39) yields the following:

$$y_d[n] = \sum_{k=-\infty}^{\infty} c_d[q-k] \left(\sum_{\ell=0}^{N} a_\ell[k]\, r^\ell \right) = \sum_{\ell=0}^{N} \left(\sum_{k=-\infty}^{\infty} a_\ell[k]\, c_d[q-k] \right) r^\ell = \sum_{\ell=0}^{N} \big(a_\ell[q] * c_d[q]\big) r^\ell \tag{42}$$

From Eq. (42), it can be seen that, for piecewise polynomial interpolation kernels, the original LTV filter described by Eq. (33) can be implemented by filtering the input signal $c_d[q]$ through $(N+1)$ LTI filters $\{a_\ell[q]\}_{\ell=0}^{N}$ and then combining these outputs using time-varying multipliers of the form r^ℓ for $0 \le \ell \le N$.

If $A_\ell(z)$ denotes the z-transform of $a_\ell[q]$ for all ℓ, then $y_d[n]$ from Eq. (42) can be efficiently computed using the system of Fig. 14, known as the Farrow structure [11–13]. It should be noted here that the input must be buffered to produce samples according to the time index q from Eq. (38).

The Farrow structure of Fig. 14 is advantageous in many respects. For all of the finite-duration piecewise polynomial kernels of interest from Section II, the set of LTI subfilters $\{A_\ell(z)\}$ is FIR and typically small in cardinality. For example, for the cubic B-spline of Section II.D, using Eq. (22), we have $N = 3$ and the subfilters are as follows [14]:

$$A_0(z) = \frac{1}{6}\left(z + 4 + z^{-1}\right)$$

$$A_1(z) = \frac{1}{2}\left(z - z^{-1}\right)$$

$$A_2(z) = \frac{1}{2}\left(z - 2 + z^{-1}\right)$$

$$A_3(z) = \frac{1}{6}\left(z^2 - 3z + 3 - z^{-1}\right)$$

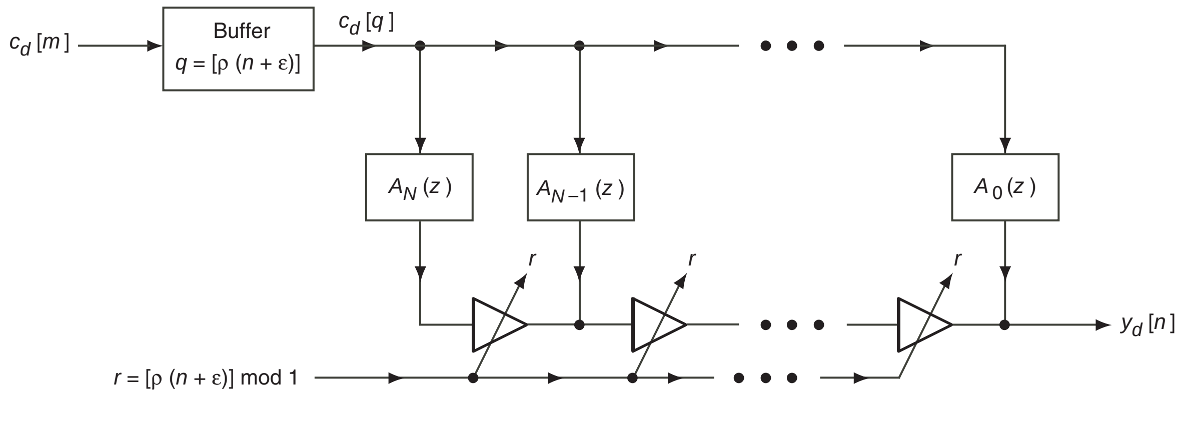


Fig. 14. Computationally efficient Farrow structure for fine interpolation with piecewise polynomial dilated interpolation kernels.

Clearly these filters are FIR and small in order. In addition, these subfilters can be further implemented efficiently by exploiting the real linear phase properties that $A_0(z)$ and $A_2(z)$ are symmetric, whereas $A_1(z)$ and $A_3(z)$ are antisymmetric [2,5].

Another advantage of the Farrow structure of Fig. 14 is that there is only one tunable or time-varying parameter, namely the fractional number r from Eq. (38). As this parameter is bound by $0 \leq r < 1$, this suggests that the Farrow structure can be easily migrated to a fixed-point design, in which the parameter r is confined to being a fixed-point fractional number.

VI. Simulation Examples of Variable SRC

At this point, we are ready to proceed with analyzing simulation results for variable SRC using the system of Fig. 8. First we will focus on the performance of the fine interpolation methods using the interpolation kernels from Section II. In particular, we compare the interpolation results from these kernels with the underlying sinc interpolant in the absence of any out-of-band aliasing artifacts. This analysis for the variable SRC problem appears to be absent from the literature and is included here for the sake of completeness.

Afterwards, we focus on a more practical communications-type problem that incorporates both the coarse and fine interpolation stages from Fig. 8(a) to achieve variable SRC in the presence of noise. There, we show the deleterious effects of aliasing in SRC and also show how the coarse interpolation stage can be used to mitigate these effects. In addition, it is shown that several of the subsequent fine interpolation methods yield bit-error rates (BERs) close to that obtained from the matched-filter bound [15].

A. Fine Interpolation Simulation Results

Recall that, for the fine interpolation stage of the proposed variable SRC system of Fig. 8, the output samples $y_d[n]$ are obtained using the fundamental SRC formula from Eq. (33), which is repeated below for the sake of convenience:

$$y_d[n] = \sum_{m=-\infty}^{\infty} c_d[m]\, \widehat{g}\big(\rho\,(n+\epsilon) - m\big)$$

In a practical setting, $\widehat{g}(t)$ is chosen to be one of the finite support dilated interpolation kernels from Section II. However, the analog input signal at the ADC is assumed to come from a sinc interpolation kernel model. If we wish to perform SRC on these samples, then the correct sequence of output samples $\widetilde{y}_d[n]$ is given by Eq. (31) to be

$$\widetilde{y}_d[n] = \sum_{m=-\infty}^{\infty} x_d[m]\, \mathrm{sinc}\big(\rho\,(n+\epsilon) - m\big) \tag{43}$$

It is worthwhile to compare how close the practically computed sequence $y_d[n]$ from Eq. (33) comes to the correct sequence $\widetilde{y}_d[n]$ from Eq. (43), for a given interpolation kernel. For the sake of comparison here, we opt to use the normalized mean-squared error (MSE) as a figure of merit. If we denote the normalized MSE by ξ, then we have the following:

$$\xi \triangleq \frac{\displaystyle\sum_n \big|y_d[n] - \widetilde{y}_d[n]\big|^2}{\displaystyle\sum_n \big|\widetilde{y}_d[n]\big|^2} \tag{44}$$

Suppose that the input sequence $x_d[n]$ obtained at the ADC is a lowpass Gaussian process band-limited to f_{BL}, similar to the scenario shown in Fig. 7(a). In this setting, $x_d[n]$ is obtained by filtering a white-noise Gaussian process by a lowpass filter, such as one of those designed in Section V.A. Furthermore, suppose that the SRC ratio is $\mathcal{R}$ and that $f_{BL} = (1/2)\mathcal{R}$. This corresponds to critical resampling in that the rate has been lowered to the point at which no redundancy remains, as can be seen in Fig. 7 with $f_{BL} = 1/2\rho = (1/2)\mathcal{R}$.

If $\overline{\xi}$ denotes the normalized MSE from Eq. (44) averaged over several ensemble realizations of the input $x_d[n]$, then a decibel plot of $\overline{\xi}$, averaged over 2048 runs, as a function of $\mathcal{R}$ is shown in Fig. 15 for the finite support interpolation kernels considered in Section II. From Fig. 15, several observations can be made. First of all, the higher-order cubic interpolants from Section II greatly outperform the lower-order nearest neighbor and linear ones. This significant increase in performance helps justify the additional computational overhead required for the higher-order methods. In addition to this observation, from Fig. 15 it can also be seen that, for all of the interpolants considered, the general trend is that the normalized MSE increases as the SRC ratio increases towards unity. This is consistent with the intuition that it is more difficult to match the underlying sinc interpolant when the input samples become less redundant.

Another unusual observation that can be made from Fig. 15 is that, while the cubic B-spline offered the best performance among the interpolation kernels considered, the behavior of the MSE was not monotonic with the SRC ratio. In particular, it can be seen that the cubic B-spline yielded its smallest MSE for $\mathcal{R} \approx 1/4$. Though there is no known theoretical reason for this to be the case, one possible heuristic explanation for this phenomenon is that for $\mathcal{R} = 1/4$ the time duration of the cubic B-spline interpolant exactly matches the size of the side-lobe duration of the underlying sinc interpolant. Regardless of this inconsistency in behavior, the cubic B-spline by far outperformed the other interpolation kernels and is perhaps best-suited for use as a fine interpolation scheme for variable SRC.

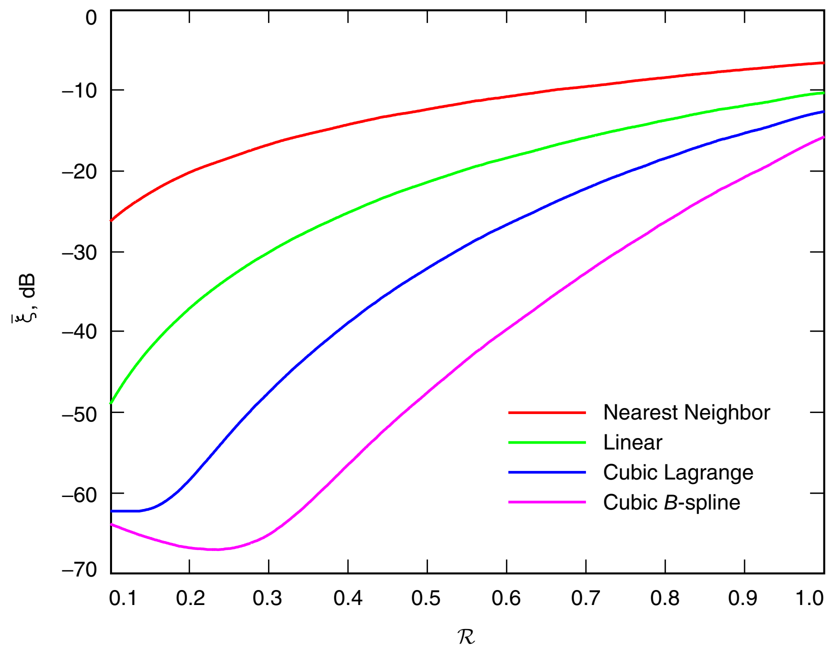


Fig. 15. Decibel plot of the normalized MSE $\overline{\xi}$ ensemble averaged over 2048 runs as a function of the SRC ratio $\mathcal{R}$.

B. Digital Communications-Type Application Simulation Results

One conceivable application of the proposed variable SRC scheme from Fig. 8 is in a digital communications setting in which the equivalent sampled baseband communications signal is sampled at a higher rate than the matched-filter system sample rate. An example of a possible scenario of this for the additive white Gaussian noise (AWGN) channel [15] is shown in Fig. 16.

Here, $d[n]$ denotes the data signal which consists of symbols from a data constellation such as binary phase-shift keying (BPSK) or quadrature phase-shift keying (QPSK) [15]. The digital filters $P_{TX}(z)$ and $P_{RX}(z)$ denote, respectively, the transmitter and receiver pulse-shaping filters. If the underlying analog pulse-shaping filter frequency response is $P(j2\pi F)$ and the symbol interval is T_{sym}, then it can be shown that the frequency responses of the digital pulse-shaping filters are as follows [5]:

$$\begin{aligned} P_{TX}\left(e^{j2\pi f}\right) &= \frac{1}{\left(\frac{T_{\text{sym}}}{L_{TX}}\right)} \sum_{k=-\infty}^{\infty} P\left(j2\pi\left(\frac{f-k}{\left(\frac{T_{\text{sym}}}{L_{TX}}\right)}\right)\right) \\ P_{RX}\left(e^{j2\pi f}\right) &= \frac{1}{\left(\frac{T_{\text{sym}}}{L_{RX}}\right)} \sum_{k=-\infty}^{\infty} P^*\left(j2\pi\left(\frac{f-k}{\left(\frac{T_{\text{sym}}}{L_{RX}}\right)}\right)\right) \end{aligned} \tag{45}$$

In the time domain, it can be shown that Eq. (45) is equivalent to saying that the impulse responses of the digital transmitter and receiver filters are obtained by sampling the underlying analog pulse-shaping filter impulse response and its matched-filter [15] impulse response, respectively, L_{TX} and L_{RX} times more finely than the symbol rate T_{sym} [5].

For the simulation results presented here, we take the underlying analog pulse-shaping filter to be a square-root raised-cosine (SRRC) filter whose frequency response is as follows [15]:

$$P(j2\pi F) = \begin{cases} \sqrt{T_{\text{sym}}}, & |F| < \frac{1-\alpha}{2T_{\text{sym}}} \\ \sqrt{T_{\text{sym}}}\cos\left[\frac{\pi T_{\text{sym}}}{2\alpha}\left(|F| - \left(\frac{1-\alpha}{2T_{\text{sym}}}\right)\right)\right], & \frac{1-\alpha}{2T_{\text{sym}}} \le |F| < \frac{1+\alpha}{2T_{\text{sym}}} \\ 0, & |F| \ge \frac{1+\alpha}{2T_{\text{sym}}} \end{cases}$$

Here, the parameter α denotes the roll-off factor [15], which satisfies $0 \le \alpha \le 1$.

The threshold detector from Fig. 16 is used to make decisions on the received, matched-filtered output in an attempt to recreate the data stream $d[n]$ at the transmitter information source. Typically, the thresholds are chosen according to the maximum-likelihood principle [15] based on the input constellation and noise present in the model.

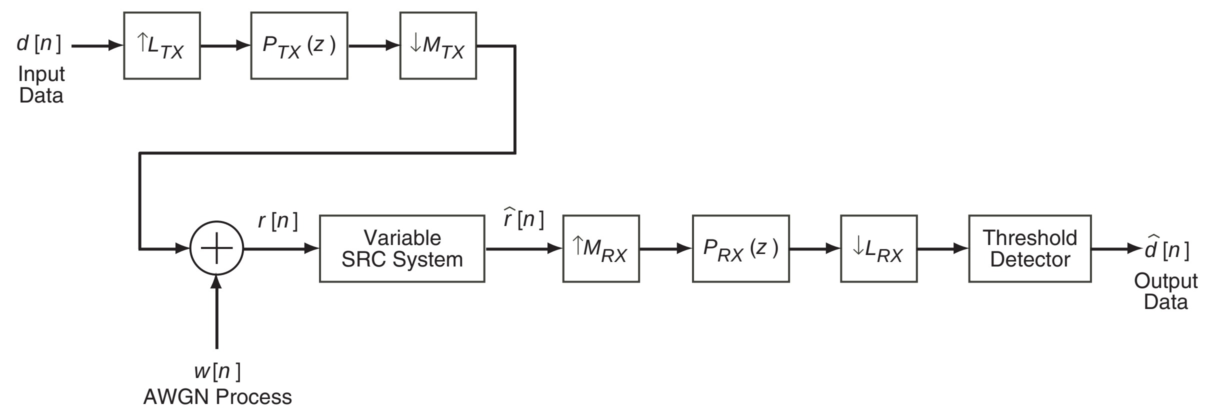


Fig. 16. Baseband model of an AWGN channel that requires SRC.

For the SRC simulations here, we have opted to use the following baseband model parameters:

- QPSK data constellation with Gray code mapping [15]
- SRRC pulse-shaping filter with a roll-off factor of $\alpha = 0.35$
- $L_{TX} = 163$, $M_{TX} = 10$ ($\Longrightarrow$ input sample rate of 16.3 samp/symb)
- $L_{RX} = 4$, $M_{RX} = 1$ ($\Longrightarrow$ target sample rate of 4 samp/symb)

From this, it can be seen that the sampling rate conversion factor is $\rho = 16.3/4 = 4.075$, meaning that the sampling rate must be lowered by a factor of 4.075.

In order to reflect a practical scenario for the variable SRC system of Fig. 16, we considered the following methods for variable SRC:

- The system of Fig. 8(a) but with no coarse interpolation stage used prior to fine interpolation
- The multi-stage fixed fractional decimation filter system of Section IV.C used for coarse interpolation ($L = 1$, $M = 2$)
- The multi-stage fixed cascaded expanded filter system of Section IV.D used for coarse interpolation ($M = 2$)

To ascertain the performance of the above variable SRC methods, we opted to calculate the BER obtained from numerous Monte Carlo simulations, which we denote here by P_b. The BERs obtained were compared against the ideal BER corresponding to the matched-filter bound given below [15]:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{46}$$

Here, $Q(x)$ is the one-dimensional Marcum Q-function [15], and E_b and N_0 denote, respectively, the energy per bit of the input data stream and the noise power spectral density [15]. As is usually the case in a digital communications environment [15], all BERs here were calculated as a function of the signal-to-noise ratio (SNR) parameter E_b/N_0.

A plot of the BERs computed using the variable SRC scheme of Fig. 8(a) in which no coarse interpolation was performed is shown in Fig. 17. As can be seen, all of the fine interpolation methods yielded poor performance compared to the ideal matched-filter bound. The reason for this phenomenon is that out-of-band noise present in the received signal $r[n]$ from Fig. 16 aliased into the data portion of the spectrum

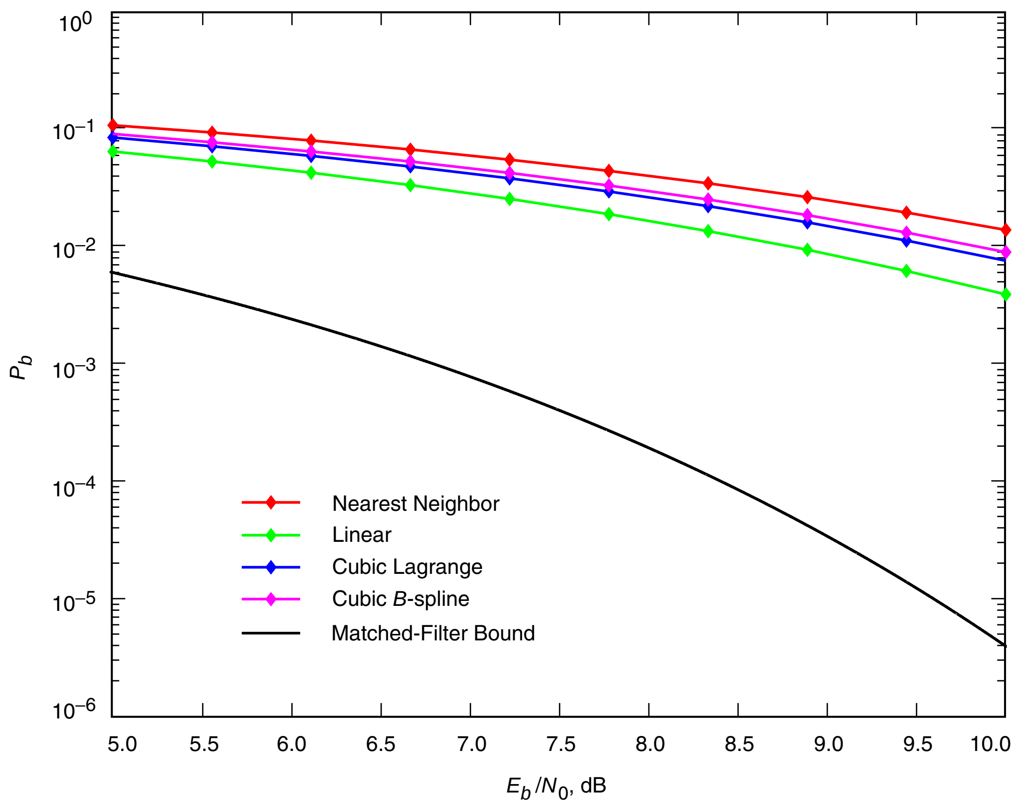


Fig. 17. BER P_b as a function of the SNR parameter E_b/N_0 using the proposed variable SRC system with no coarse interpolation.

of the SRC output signal $\widehat{r}[n]$. As the matched-filter system can mitigate only the effects due to in-band noise and not those due to aliased out-of-band noise, this resulted in a severely degraded performance of the overall system. Essentially, the fine interpolation systems attempted to fit the received signal to smooth functions, which is not appropriate here due to the high-frequency out-of-band noise present.

In Fig. 18, we have plotted the BERs obtained using the multi-stage fixed fractional decimation filter system of Section IV.C for coarse interpolation. Here, we chose $L = 1$ and $M = 2$, which implies that we need to implement $K = \lfloor \log_2(\rho) \rfloor = 2$ stages. As can be seen, the presence of coarse interpolation in this case led to a dramatic improvement in BER. Furthermore, the performance improved as the order of the fine interpolant increased. It can be observed that the BER corresponding to the cubic B-spline interpolant, which was the best among all fine interpolation methods considered, is very close to the ideal BER offered by the matched-filter bound. Even though there is a residual amount of out-of-band noise that aliases into the data component of the SRC output, this effect is minor, as evidenced by the BER results of Fig. 18.

As a final test of the proposed variable SRC schemes presented, we considered using the multi-stage fixed cascaded expanded filter system of Section IV.D for the coarse interpolation stage. Here, we chose $M = 2$, which, as with the decimation filter system considered above, requires $K = \lfloor \log_2(\rho) \rfloor = 2$ stages. A plot of the BERs observed using this coarse interpolation approach is shown in Fig. 19. Of all of the coarse interpolation schemes considered, it can be seen that the cascaded extended filter system performed the best in terms of BER. This is due to the extra redundancy of the desired SOI present at the output of the coarse interpolation stage. Though difficult to see from Fig. 19, the higher-order interpolation kernels outperformed the lower-order ones. The BER of the cubic B-spline interpolant, which was the best of the

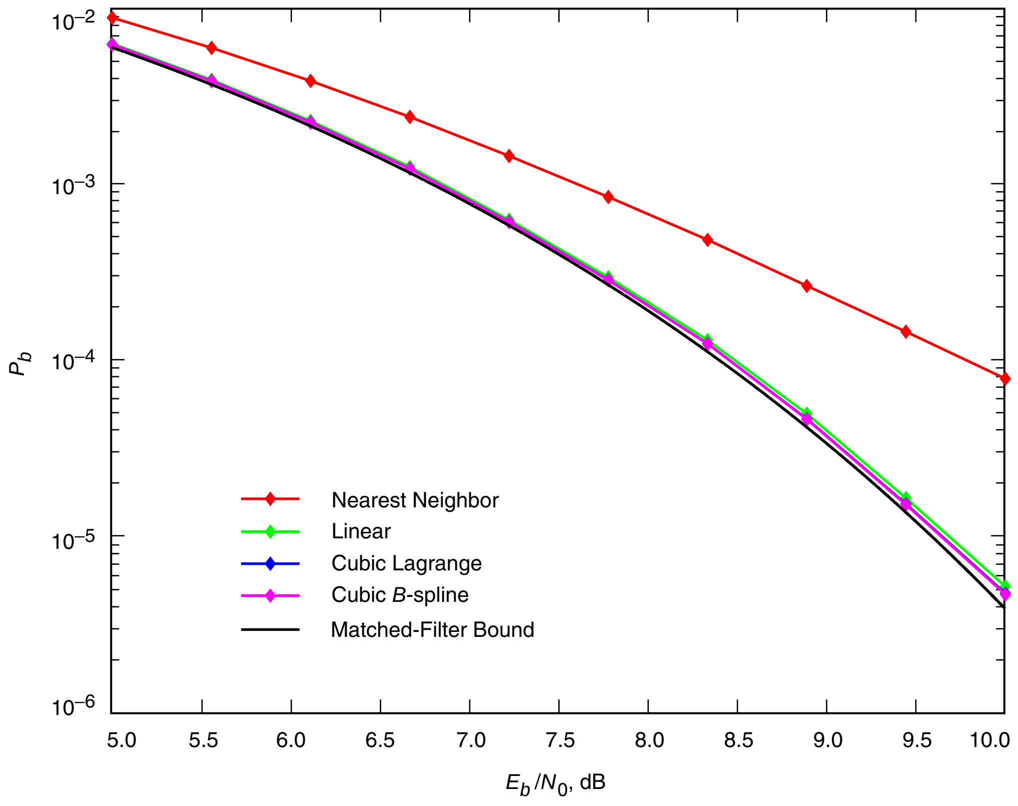


Fig. 18. BER P_b as a function of the SNR parameter E_b/N_0 using the proposed variable SRC system with the multi-stage fixed fractional decimation coarse interpolation scheme.

methods considered here, is indiscernible from the matched-filter bound BER. As with the decimation filter system from above, even though there is a residual amount of out-of-band noise that aliases into the data portion of the SRC output, this effect is very minor, as is apparent from Fig. 19.

VII. Concluding Remarks

In this exposition, we presented a variety of approaches to carry out variable SRC, ranging from interpolation schemes to accommodate general conversion ratios to methods to simultaneously convert the sample rate while filtering out aliasing artifacts. To achieve variable SRC in an environment such as that expected for the Advanced Receiver, we proposed a two-stage approach: a coarse interpolation stage to lower the bulk of the rate and/or remove out-of-band aliasing components, and a fine interpolation stage to precisely adjust the sample rate to its desired target value. Simulation results presented showed the strengths and weaknesses of the various coarse/fine interpolation schemes discussed.

Several interesting open problems concerning variable SRC still remain. At this time, it is not clear how to adjust the sampling rate when the target rate is itself unknown. For example, if one of the transmit module links to the Advanced Receiver is in an adverse environment and must change its data rate, it is not clear how to alter the sampling rate autonomously at the receiver.

Another interesting open problem is finding a way to couple the coarse and fine interpolation stages from the proposed variable SRC scheme into one composite operation. Though the proposed two-stage approach was shown to perform well in a practical application, it is still somewhat ad hoc in the sense that

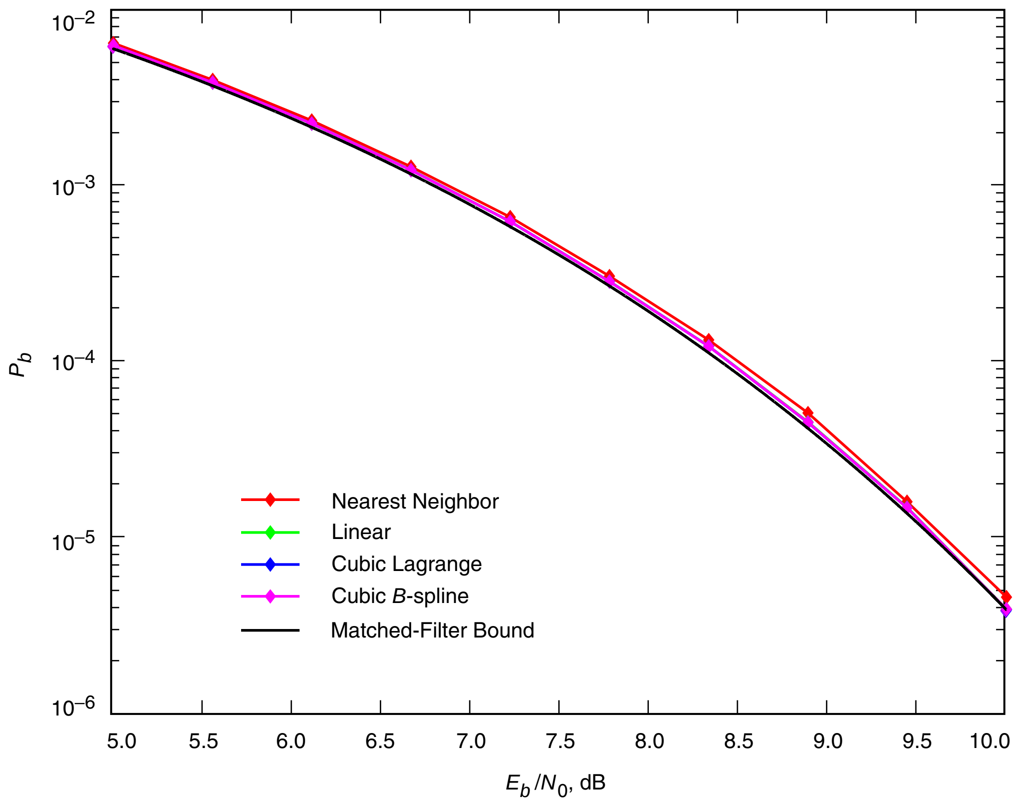


Fig. 19. BER P_b as a function of the SNR parameter E_b/N_0 using the proposed variable SRC system with the multi-stage fixed cascaded expanded filter coarse interpolation scheme.

the primary goal of the coarse interpolation stage is to remove aliasing artifacts that cannot be removed using the subsequent fine interpolation stage. It may be worthwhile to focus on fine interpolation schemes that also can account for the effects of out-of-band spectral components. This will involve the investigation of new classes of interpolation kernels that may be useful for future variable SRC work for the Advanced Receiver.

References

[1] R. Navarro and J. Bunton, “Signal Processing in the Deep Space Array Network,” *The Interplanetary Network Progress Report*, vol. 42-157, Jet Propulsion Laboratory, Pasadena, California, pp. 1–17, May 15, 2004. http://ipnpr/progress_report/42-157/157N.pdf

[2] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd. ed., Upper Saddle River, New Jersey: Prentice Hall PTR, 1999.

[3] M. Unser, A. Aldroubi, and M. Eden, “Polynomial Spline Signal Approximations: Filter Design and Asymptopic Equivalence with Shannon’s Sampling Theorem,” *IEEE Trans. Inform. Theory*, vol. 38, no. 1, pp. 95–103, January 1992.

[4] P. P. Vaidyanathan, “Generalizations of the Sampling Theorem: Seven Decades after Nyquist,” *IEEE Trans. Circuits Syst. I*, vol. 48, no. 9, pp. 1094–1109, September 2001.

[5] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, New Jersey: Prentice Hall PTR, 1993.

[6] P. Thévenaz, T. Blu, and M. Unser, "Interpolation Revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, July 2000.

[7] T. Blu, P. Thévenaz, and M. Unser, "Complete Parameterization of Piecewise-Polynomial Interpolation Kernels," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1297–1309, November 2003.

[8] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part I—Theory," *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.

[9] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part II—Efficient Design and Applications," *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 834–848, February 1993.

[10] T. A. Ramstad, "Digital Methods for Conversion Between Arbitrary Sampling Frequencies," *IEEE Trans. Acoust., Speech, signal Processing*, vol. ASSP-32, no. 3, pp. 577–591, June 1984.

[11] F. M. Gardner, "Interpolation in Digital Modems—Part I: Fundamentals," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 501–507, March 1993.

[12] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in Digital Modems—Part II: Implementation and Performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998–1008, June 1993.

[13] C. W. Farrow, "A Continuously Variable Digital Delay Element," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS 1988)*, Espoo, Finland, pp. 2641–2645, June 6–9, 1988.

[14] A. Haftbaradaran and K. Martin, "An interpolation Filter Based on Spline Functions for Non-Synchronized Timing Recovery," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, vol. IV, Vancouver, Canada, pp. 309–312, May 23–26, 2004.

[15] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communications Techniques: Signal Design and Detection*, Upper Saddle River, New Jersey: Prentice Hall PTR, 1994.