



Digital resampling by using polynomial interpolation. Farrow filter

Contents

[Introduction](#)

[Problem definition](#)

[Lagrange polynomial interpolation](#)

[Effective polynomial representation according to Horner structure](#)

[Using polynomial interpolation for digital signal resampling](#)

[Minimization of multipliers when calculating Lagrange polynomial interpolation](#)

[Conclusions](#)

[Reference](#)

Introduction

In digital communication systems it is necessary to provide symbol synchronizing of the transmitter and receiver. At the same time using independent clock generators leads to a difference of signal sampling frequencies of the transmitter and receiver.

In addition, in many cases there is a compensating task of the so-called fractional time delay by the value which is less than the sampling interval. The filter capable to compensate the arbitrary fractional time delay can be also used for synchronizing clock generators in view of their slow instability. For this purpose it is necessary to use an additional time error detector to calculate the fractional delay of signals.

Sometimes it is necessary to perform the sample rate conversion to the fractional number $\frac{P}{Q}$, where P and Q — are integers.

In this section we will consider one of the effective methods of solving the specified tasks on the basis of Lagrange polynomial interpolation [1]. Farrow filter realizing the method of Lagrange polynomial interpolation for digital resampling of signals will be also considered [2-5].

Problem definition

Let the input signal be $s(n)$, $n = \dots, 0, 1, 2$ which samples are taken with the sampling frequency F_s Hz. Then the sample n — corresponds to the timepoint $t_n = \frac{n}{F_s}$ sec.

It is necessary to recalculate digital values of the input signal $s(n)$ into digital values of the signal $y(k)$, $k = 0, 1, 2 \dots$ taken with the sampling frequency $F_y = \frac{P}{Q} \cdot F_s$, where P and Q are integers. Besides, the first sample $y(0)$ shall be delayed in time concerning the first sample of the input signal $s(0)$ for value $\Delta t = \frac{x_0}{F_s}$ sec., $-1 \leq x_0 < 0$.

Further in this section the variable n will be indexed with samples of the input signal $s(n)$ taken with the sampling frequency F_s , and the variable k will be indexed with the signal samples after resampling $y(k)$.

Thus, the k — sample $y(k)$ corresponds to the timepoint:

$$t_k = k \cdot \frac{Q}{P \cdot F_s} - \frac{x_0}{F_s}.$$

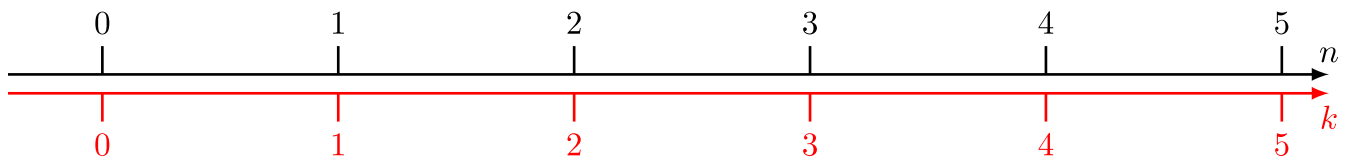
(1)

Having performed normalization concerning the sampling frequency $F_s = 1$ Hz, it is possible to write:

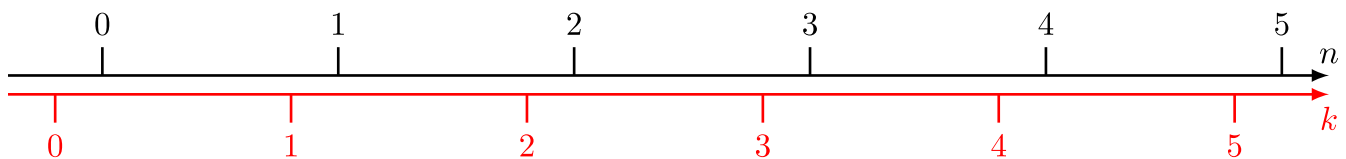
$$x_k = t_k \cdot F_s = k \cdot \frac{Q}{P} - x_0. \quad (2)$$

Temporal ratios of moments of capturing signal samples $s(n)$ and $y(k)$ at various values P , Q and x_0 , are shown in Figure 1.

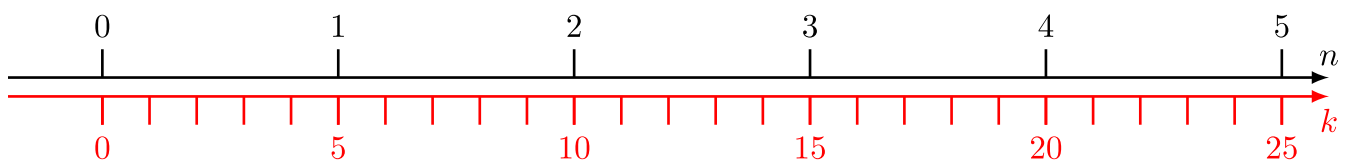
Special case 1: $P = 1, Q = 1, x_0 = 0, F_y = F_s$



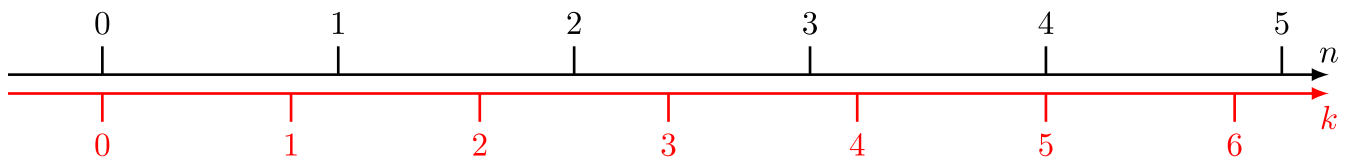
Special case 2: $P = 1, Q = 1, x_0 = 0.2, F_y = F_s$



Special case 3: $P = 5, Q = 1, x_0 = 0, F_y = 5 \cdot F_s$



Special case 4: $P = 5, Q = 4, x_0 = 0, F_y = \frac{5}{4} \cdot F_s$



Special case 5: $P = 5, Q = 4, x_0 = 0.2, F_y = \frac{5}{4} \cdot F_s$

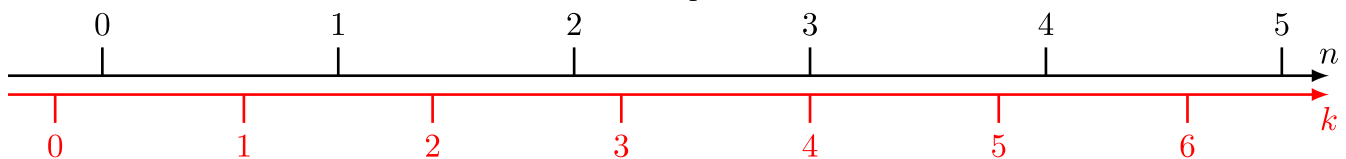


Рисунок 1. Figure 1. Temporal ratios of moments of capturing signal samples $s(k)$ and $y(n)$, at various values P , Q and x_0

Sampling moments of the input signal $s(n)$ are black, and — moments of capturing samples of the resampled signal $y(k)$ are red.

Special case 1. $P = Q = 1$, $x_0 = 0$. In this case the sampling frequency is $F_y = F_s$, and the signal $y(k)$ completely repeats $s(n)$.

Special case 2. $P = Q = 1$, $x_0 = 0.2$. In this case the sampling frequency is $F_y = F_s$, but the signal $y(k)$ has a fractional delay concerning the pickup signal $s(n)$.

Special case 3. $P > 1$, $Q = 1$, $x_0 = 0$. In this case the signal sampling frequency is $y(k)$, $F_y = P \cdot F_s$, that is P higher than the signal sampling frequency $s(n)$. Thus, we have the signal $s(n)$ digital interpolation.

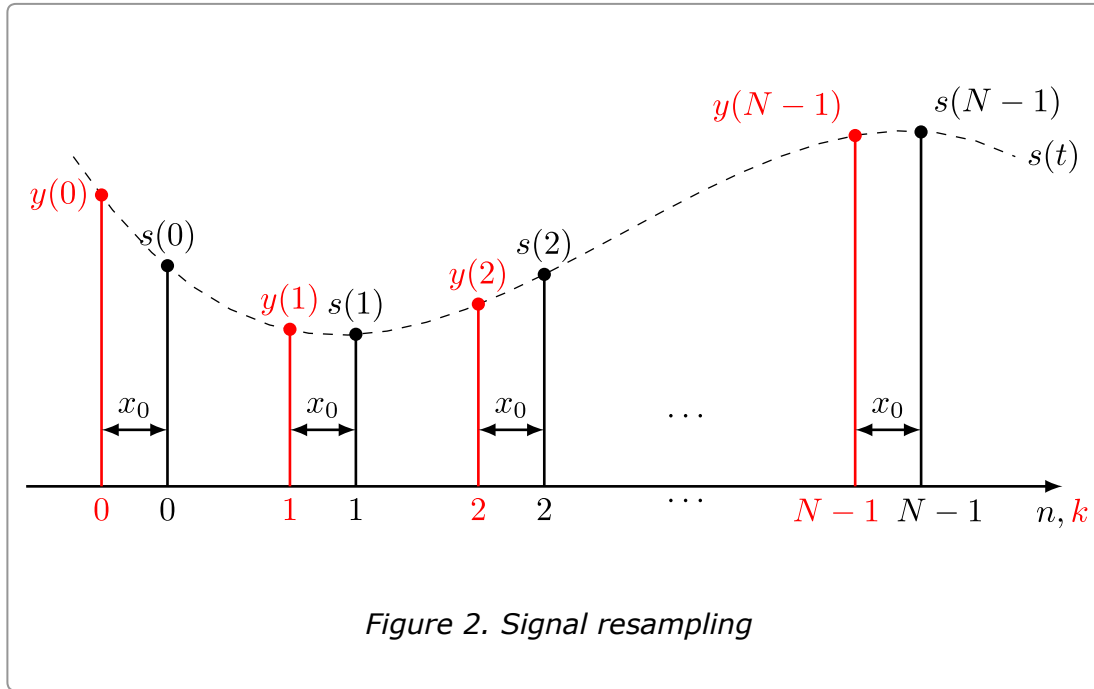


Figure 2. Signal resampling

Special case 4. $P > 1$, $Q > 1$, $P \neq Q$, $x_0 = 0$. In this case we have the fractional resampling of $s(n)$ signal. The signal sampling frequency $y(k)$ is equal to $F_y = \frac{P}{Q} \cdot F_s$.

Special case 5. $P > 1$, $Q > 1$, $P \neq Q$, $x_0 > 0$. In this case we have the fractional signal resampling $s(n)$.

plus adding the fractional delay. It is the most general case of the considered special ones. The signal sampling frequency $y(k)$ is equal to $F_y = \frac{P}{Q} \cdot F_s$.

To solve the resampling task it is necessary to carry out the continuous signal restoration $s(t)$ according to the available discrete signal $s(n)$, $n = 0, 1, 2, \dots$, and to calculate its discrete values for the timepoints $s(x_k)$ where x_k is calculated according to (2).

The resampling process for special case 2 (the fractional delay compensation) is shown in Figure 2.

As we use only indexes of signal samples, values x_k are also set not as absolute time values, but as normalized ones concerning the sampling frequency.

Lagrange polynomial interpolation

There is such a proof in mathematical analysis that the only polynomial of $N - 1$ degree passes through N points. For example, only one straight line can be drawn through two points, only one parabola can be drawn through three points. The only polynomial of $N - 1$ degree passing through N points is called Lagrange polynomial interpolation:

$$s(t) = \sum_{n=0}^{N-1} a_n \cdot t^n, \quad (3)$$

where a_n — are polynomial coefficients which are calculated on the basis of the discrete values $s(n)$, $n = 0 \dots N - 1$.

To calculate coefficients a_n we can compose the linear system (see Figure 2):

$$\begin{cases} a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + \dots + a_{N-1} \cdot 0^{N-1} = s(0), \\ a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + \dots + a_{N-1} \cdot 1^{N-1} = s(1), \\ a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + \dots + a_{N-1} \cdot 2^{N-1} = s(2), \\ \vdots \\ a_0 + a_1 \cdot (N-1) + a_2 \cdot (N-1)^2 + \dots + a_{N-1} \cdot (N-1)^{N-1} = s(N-1). \end{cases}$$

(4)

This system can be written down in the matrix form as:

$$\mathbf{M} \cdot \mathbf{a} = \mathbf{s},$$

(5)

where

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0^2 & \dots & 0^{N-1} \\ 1 & 1 & 1^2 & \dots & 1^{N-1} \\ 1 & 2 & 2^2 & \dots & 2^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & N-1 & (N-1)^2 & \dots & (N-1)^{N-1} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s(0) \\ s(1) \\ s(2) \\ \vdots \\ s(N-1) \end{bmatrix}.$$

(6)

The solution of the equation system can be as follows:

$$\mathbf{a} = \mathbf{M}^{-1} \cdot \mathbf{s},$$

(7)

where \mathbf{M}^{-1} — is the matrix inverse for the matrix \mathbf{M} .

The calculation procedure of the matrix inverse is one of the most costly ones from the point of view of computing resources. In the general case the number of transactions $O(N) \sim N^3$, which is in proportion to the dimension cube of the square matrix is required. For example, to calculate coefficients of the cubic polynom for ($N = 4$) in the general case $O(4) \sim 4^3 = 64$ of multiplication is required that is implemented in practice in a difficult way. Further we will see how it is possible to reduce the number of multiplication operations to three, and at that, two of them will be multiplication by $\frac{1}{2}$ that is easily realized in integer arithmetic by one bit shifting on the right.

We will return to the matter below. Now we will consider effective representation of polynomials according to Horner structure.

Effective polynomial representation according to Horner structure

We will consider polynomial (3) in more detail:

$$s(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + \dots + a_{N-1} \cdot t^{N-1}.$$

(8)

We will put t outside the brackets and get:

$$s(t) = a_0 + t \cdot (a_1 + a_2 \cdot t + \dots + a_{N-1} \cdot t^{N-2}). \quad (9)$$

Again we will put t outside the brackets and get:

$$s(t) = a_0 + t \cdot (a_1 + t \cdot (a_2 + \dots + a_{N-1} \cdot t^{N-3})). \quad (10)$$

Thus, putting t outside the brackets possible number of times, we will get a set of the nested parentheses:

$$\begin{aligned} s(t) &= a_0 + t \cdot (a_1 + t \cdot (a_2 + \dots + t \cdot (a_{N-2} + a_{N-1} \cdot t))) = \\ &= t \cdot (t \cdot (\dots t \cdot (t \cdot a_{N-1} + a_{N-2}) \dots + a_2) + a_1) + a_0. \end{aligned} \quad (11)$$

For example, we will get a cubic polynomial for $N = 4$:

$$s(t) = t \cdot (t \cdot (t \cdot a_3 + a_2) + a_1) + a_0, \quad (12)$$

which can be realized in the form of the structure shown in Figure 3.

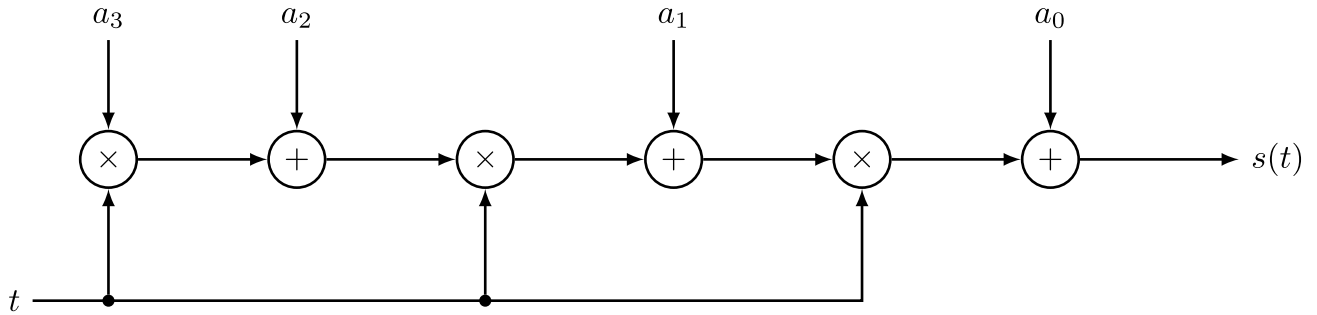


Figure 3. Calculating a cubic polynomial according to Horner structure.

This representation is called Horner structure and it is widely used for effective calculating values of polynomials [6]. Its main benefit is the fact that the multiplier and adder $N - 1$ are required to calculate the polynom value of $N - 1$ degree, in case of the known coefficients.

Using polynomial interpolation for digital signal resampling

Setting up polynomial interpolation is often carried out when using Lagrange orthogonal polynomial. However solving the system of linear equations (4) is a more general approach which can be applied not only to Lagrange interpolation, but also to other methods of polynomial interpolation (Hermite polynomial interpolation, spline interpolation and others).

We have already said that the calculation operation of the matrix inverse is a computationally complex challenge. Besides, the input data flow can be very long, and we can apply piecewise polynomial interpolation to processing. At the same time there is a question of choosing the polynomial order (the sample number of the input signal for interpolation).

The most often used method is piecewise and polynomial cubic interpolation in view of compromise between accuracy and computing costs. We will consider a method of piecewise and polynomial cubic interpolation in more detail.

The cubic polynomial can be constructed with four points according to (7). For any $1 \leq x_k < N - 2$ it is possible to get two samples more to the right and more to the left of x_k

as it is shown in Figure 4a. Then the value $y(k)$ can be calculated on the basis of four input samples $s(n)$, $s(n-1)$, $s(n-2)$, $s(n-3)$.

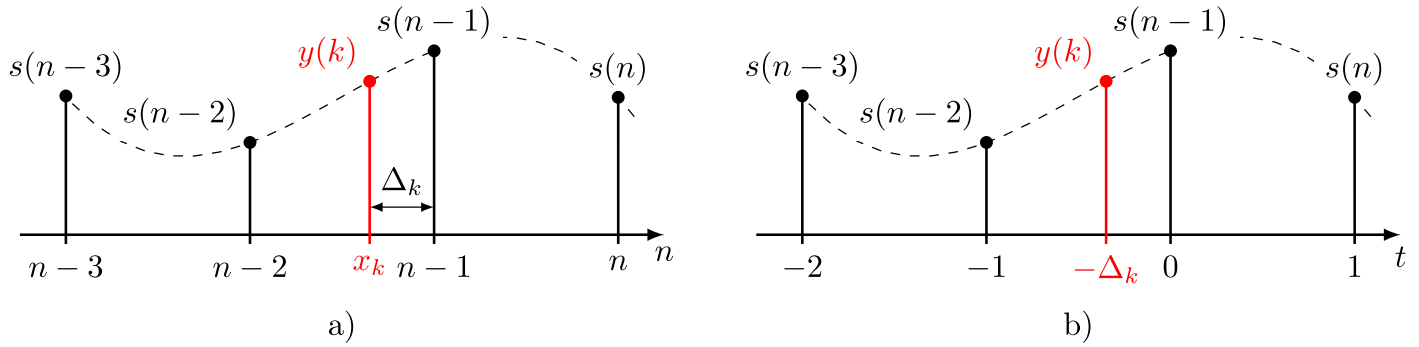


Figure 4. Piecewise and cubic interpolation

Let us pay attention to the fact that the matrix \mathbf{M} depends only on indexes of input signal samples. It means that to escape recalculating \mathbf{M} and the matrix inverse \mathbf{M}^{-1} for each new k , we can hold fix X-direction indexes. Let it be $x_k = n - 1 - \Delta_k$ (see Figure 4a), then instead of x_k we will use the value $0 \leq \Delta_k < 1$ as it is shown in Figure 4b. In this case we can calculate the matrix inverse \mathbf{M}^{-1} once and use it for any x_k .

The system of equations to calculate polynomial coefficients with constant indexes according to Figure 4b is as follows:

$$\begin{cases} a_0 + a_1 \cdot (-2) + a_2 \cdot (-2)^2 + a_3 \cdot (-2)^3 = s(n-3), \\ a_0 + a_1 \cdot (-1) + a_2 \cdot (-1)^2 + a_3 \cdot (-1)^3 = s(n-2), \\ a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + a_3 \cdot 0^3 = s(n-1), \\ a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3 = s(n). \end{cases} \quad (13)$$

Then we can present the matrix \mathbf{M} and get the inverse one to it \mathbf{M}^{-1} :

$$\mathbf{M} = \begin{bmatrix} 1 & -2 & 4 & -8 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{M}^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{6} & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & -1 & \frac{1}{2} \\ -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \end{bmatrix}. \quad (14)$$

We do not give the calculation procedure of the matrix inverse. You can independently be convinced that $\mathbf{M} \cdot \mathbf{M}^{-1} = \mathbf{I}$, where \mathbf{I} — is the identity matrix.

Polynomial coefficients according to (7) can be got as result of the matrix inverse multiplication \mathbf{M}^{-1} by the vector of the input signal samples \mathbf{s} :

$$\mathbf{a} = \mathbf{M}^{-1} \cdot \mathbf{s} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{6} & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & -1 & \frac{1}{2} \\ -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \end{bmatrix} \cdot \begin{bmatrix} s(n-3) \\ s(n-2) \\ s(n-1) \\ s(n) \end{bmatrix}. \quad (15)$$

We will perform matrix multiplication (15) and get expressions for polynomial coefficients:

$$\begin{cases} a_0 = s(n-1), \\ a_1 = \frac{1}{6} \cdot s(n-3) - s(n-2) + \frac{1}{2} \cdot s(n-1) + \frac{1}{3} \cdot s(n), \\ a_2 = \frac{1}{2} \cdot s(n-2) - s(n-1) + \frac{1}{2} \cdot s(n), \\ a_3 = -\frac{1}{6} \cdot s(n-3) + \frac{1}{2} \cdot s(n-2) - \frac{1}{2} \cdot s(n-1) + \frac{1}{6} \cdot s(n). \end{cases} \quad (16)$$

Owing to expression (16) it is possible to notice that polynomial coefficients depend on held values of the input signal $s(n)$. We can interpret (16) for calculating coefficients as differential equations of the FIR-filter. Then the separate FIR-filter will carry out calculating each coefficient and the resampling filter structure can be presented as it is shown in Figure 5.

Each FIR-filter calculates one polynomial coefficient, then they will be transferred to Horner structure to calculate a polynomial at the current value of the fractional delay Δ_k . The structure given in Figure 5 bears the name of the Farrow filter in honor of the scientist who used the similar filter to change of the signal delay for the first time [2-5].

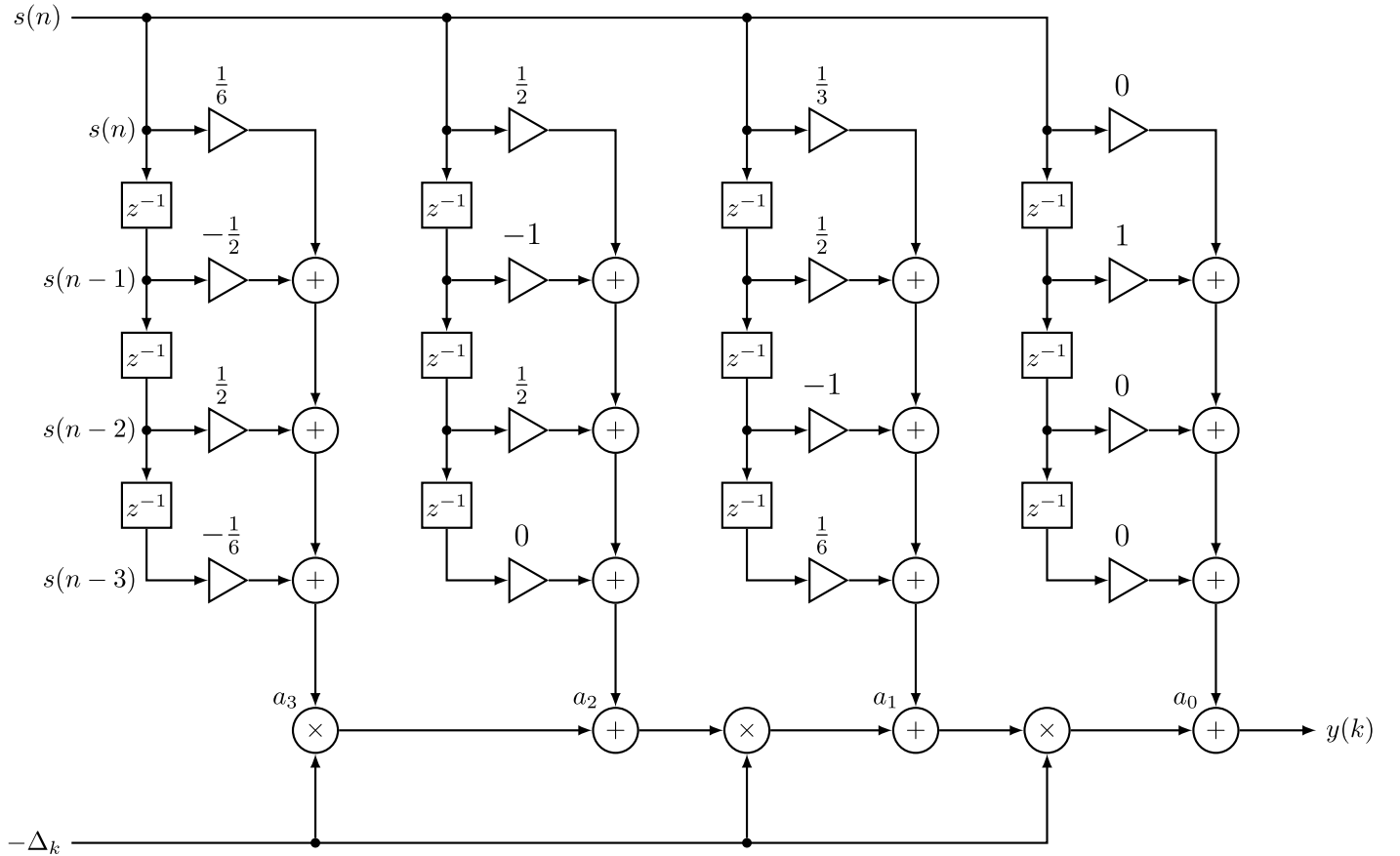


Figure 5. Structure of Farrow filter for digital resampling by using Lagrange polynomial interpolation

It is important to mark that the Farrow filter output $y(k)$ is delayed corresponds to the input signal $s(n)$ by one sample in units of input signal samples (in case of the sampling frequency F_s).

Using fixed X-direction values allows to get rid of the matrix inversion on a real time basis. However Farrow filter also demands a significant amount of multipliers by $\frac{1}{6}, \frac{1}{3}$ and $\frac{1}{2}$. At the same time in case of realizing in integer arithmetic multiplications by $\frac{1}{2}$ can be considered trivial because they are realized as one place shifting on the right.

In the following paragraph we modify Farrow filter to reduce the number of multipliers when calculating polynomial coefficients.

Minimization of multipliers when calculating Lagrange polynomial interpolation

Upon realizing signal resampling filters we have to aim at minimizing the number of multipliers when calculating polynomial coefficients.

Let's look attentively at (16). The coefficient a_3 can be presented in the form of:

$$a_3 = \frac{1}{6} \cdot (s(n) - s(n-3)) + \frac{1}{2} \cdot (s(n-2) - s(n-1)). \quad (17)$$

In (16) it is also possible to notice that

$$a_1 + a_3 = -\frac{1}{2} \cdot s(n-2) + \frac{1}{2} \cdot s(n), \quad (18)$$

from which it is possible to calculate the coefficient a_1 as:

$$a_1 = \frac{1}{2} \cdot (s(n) - s(n-2)) - a_3. \quad (19)$$

To calculate the coefficient a_2 it is possible to use the following conclusion from (16):

$$a_1 + a_2 + a_3 = s(n) - s(n-1), \quad (20)$$

then:

$$a_2 = s(n) - s(n-1) - a_1 - a_3. \quad (21)$$

Thus, we can rewrite (16) in the optimized form only with three multipliers (and at that, two of them are equal to $\frac{1}{2}$):

$$\begin{cases} a_0 = s(n-1), \\ a_3 = \frac{1}{6} \cdot (s(n) - s(n-3)) + \frac{1}{2} \cdot (s(n-2) - s(n-1)), \\ a_1 = \frac{1}{2} \cdot (s(n) - s(n-2)) - a_3, \\ a_2 = s(n) - s(n-1) - a_1 - a_3. \end{cases} \quad (22)$$

The optimized digital resampling filter structure corresponding to (22) is given in Figure 6.

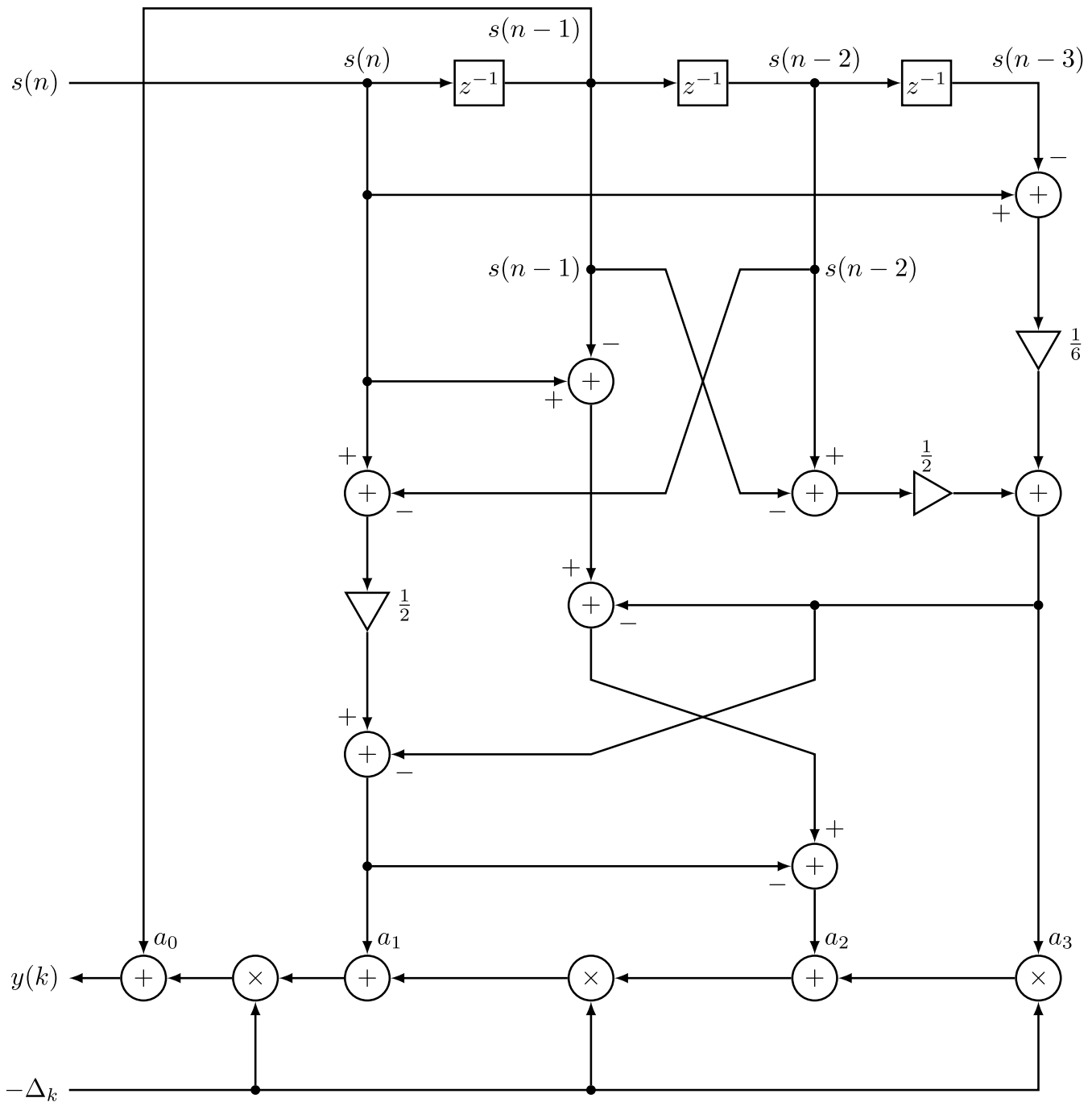


Figure 6. Optimized filter function chart of digital resampling

Thus, we have got the algorithm which calculates coefficients of Lagrange polynomial interpolation using only one multiplier by $\frac{1}{6}$ and two trivial multipliers by $\frac{1}{2}$. In the following sections we will review examples of using the filter of digital resampling on the basis of Lagrange polynomial interpolation.

Conclusions

In this section we have considered structures of digital resampling filters on the basis of Lagrange polynomial interpolation.

Horner structure to calculate polynomial values was considered. The number of multipliers and adders of Horner structure is equal to the polynomial order.

The Farrow filter structure is got, and its optimization is performed. As a result coefficients of Lagrange polynomial interpolation can be calculated when using only three multipliers. One multiplication by $\frac{1}{6}$ and two trivial multiplication by $\frac{1}{2}$ are required.

In the [following section](#) we will consider characteristics of the digital resampling filter and examples of its use to compensate the fractional time delay, interpolation of signals and fractional signal sampling frequency change.

You can provide your any feedbacks, questions and wishes in the message board or guestbook.

Reference

[1] *Kahaner D., Moler C., Nash S. Numerical Methods and Software.* Prentice Hall, 1988.

[2] *Farrow C.W. A Continuously Variable Digital Delay Element.* Circuits and Systems, IEEE International Symposium. 1988, p. 2641–2645. vol. 3

[3] *Gardner Floyd M. Interpolation in Digital Modems-Part I: Fundamentals:* IEEE Transactions on Communications, Vol. 41, No. 3, March 1993, P. 501-507.

[4] *Erup L., Gardner Floyd M., Harris Robert A. Interpolation in Digital Modems-Part II: Implementation and Performance:* IEEE Transactions on Communications, Vol. 41, No. 6, June 1993, p.998-1008.

[5] *Franck A. Efficient Algorithms for Arbitrary Sample Rate Conversion with Application to Wave Field Synthesis.* PhD thesis. Universitätsverlag Ilmenau, Ilmenau, 2012. [\[PDF\]](#)

[6] *McConnell J. Analysis of Algorithms: An Active Learning Approach.* Jones and Bartlett Publishers, 2001.

Select spelling error with your mouse and press Ctrl + Enter
 Powered by Orphus

Copyright © 2015-2017 Sergey Bakhurin



dsplib.org is licensed under a [Creative Commons Attribution-NoDerivatives 4.0 International License](#).