An Efficient Filtering Structure for Spline Interpolation and Decimation

David Lamb, Luiz F. O. Chamon and Vítor Heloiz Nascimento

An efficient structure for spline-based fractional delay filtering for interpolation/decimation is introduced. Inspired by the Newton structures for Lagrange interpolation, it requires less than half the number of operations of a typical Farrow implementation. Moreover, it displays better frequency response characteristics than Lagrange-based filters. To obtain this structure, a matrix form of the Farrow transfer function is put forward and used to derive state-space transformations between the Lagrange-Farrow structure and its Newton counterpart. These transformations are then applied to the spline polynomial giving rise to the efficient Newton-like spline filtering method.

Introduction: Fractional delay (FD) filtering is a technique to evaluate a discrete-time signal at arbitrary—possibly non-integer multiple of the sampling rate—delays. FD filters are at the heart of many digital signal processing solutions such as asynchronous sample rate conversion (ASRC) [1], timing recovery in all-digital receivers for software-defined radio [2], and wave field synthesis [3]. A thorough review of FD filtering and applications can be found in [4].

Several structures have been presented in the literature to implement different polynomial FD filters. One of the most celebrated is the *Farrow structure* that can be used to efficiently implement any polynomial response [5]. Many improvements and modifications of this structure are available, most notably the *modified Farrow structure* that exploits coefficient symmetry to reduce the number of multipliers. Further optimizations are possible by constraining the response to a single class of polynomials. For instance, when considering only Lagrange polynomials, the *Newton structure* from [6, 7, 8] is by far the least computationally expensive. Nevertheless, limitations in the frequency response of Lagrange FD filters entail the use of higher order polynomials to meet requirements, leading designers to use polynomials with better characteristics such as splines. For this reason, the structure developed in this work aims to combine the performance of spline FD filters with the reduced complexity of the Newton structure.

Before proceeding, note that any interpolation structures can be used for decimation (and vice-versa) by means of network transposition [8, 9]. All structures described in this paper are therefore suitable for both interpolation and decimation. Thus, due to space constraints and without loss of generality, only interpolation is discussed in the sequel.

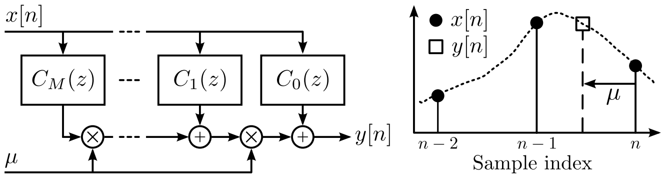
The Farrow structure: The Farrow structure (Fig. 1a) was introduced in [5] as a general implementation for arbitrary polynomial FD filters. Its transfer function can be written as

$$H(z,\mu) = \sum_{m=0}^{M} \left(\sum_{n=0}^{N} c_{mn}\, z^{-n} \right) \bar{\mu}^m, \tag{1}$$

where M is the polynomial order and N is the subfilter order—usually, $M = N$. The transfer function (1) is also parametrized by $\bar{\mu} \in [-1, 0)$, the intersample position, which controls the fractional delay of the filter as illustrated in Fig. 1b. In fact, one of the most important features of the Farrow structure is that it can implement variable delays. The $\{c_{mn}\}$ are coefficients of the filters $C_m(z) = \sum_{n=0}^{N} c_{mn} z^{-n}$ (see Fig. 1a) that uniquely define the polynomial being implemented. For clarity, they are typically collected in a matrix $\boldsymbol{C}$ that can be evaluated for classical polynomials such as Lagrange and Hermite using techniques from [10].

The modified Farrow structure reduces complexity using instead of $\bar{\mu}$ the transformed value $\mu = \bar{\mu} + 0.5 \in [-0.5, 0.5)$, taking advantage of the resulting symmetry in the $C_m(z)$ coefficients [11] (note the symmetry in the rows of (3) further ahead). However, the modified Farrow structure still requires $\mathcal{O}(M^2)$ multiplications for $M = N$.

The Newton structure: The Newton structure (Fig. 2) introduced in [6] and refined in [7, 8] is based on Newton's backward difference formula, an efficient algorithm for Lagrange polynomial interpolation. Of all optimized implementations of the Lagrange polynomial surveyed in [12], the Newton structure has the lowest complexity of only $\mathcal{O}(M)$ operations. However, it is restricted to the Lagrange polynomial, so that the only way to improve its frequency response is by increasing the order M, undermining the computational advantages and adding delay [8].



(a) (b)

Fig. 1. The Farrow structure: (a) block diagram; (b) intersample position (μ)

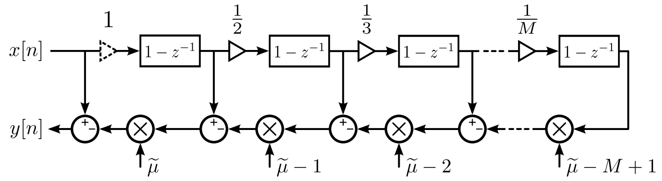


Fig. 2. The Newton structure for Lagrange interpolation

Farrow state-space transformations and the Newton structure: Since the Newton structure implements a Lagrange FD filter, it is clearly equivalent to a Farrow implementation of that same polynomial. However, the Newton structure has only been motivated so far as a direct implementation of Newton's backward difference formula [6, 7, 8]. To formalize the relation between these two structures, this section shows how the Newton structure can be derived directly from a Farrow-Lagrange filter. The motivation is that the same steps might lead to efficient structures when applied to other polynomials in the Farrow structure. For the sake of clarity, the following derivations are carried with $M = N = 3$, although they are valid for arbitrary values.

First, express the Farrow transfer function (1) in matrix form as

$$H_{\text{Farrow}}(z,\mu) = \boldsymbol{\mu}^T \boldsymbol{C} \boldsymbol{z}, \tag{2}$$

where $\boldsymbol{\mu} = [\begin{matrix} 1 & \mu & \mu^2 & \mu^3 \end{matrix}]^T$; $\boldsymbol{z} = [\begin{matrix} 1 & z^{-1} & z^{-2} & z^{-3} \end{matrix}]^T$; and $\boldsymbol{C}$ is chosen to implement a Lagrange polynomial [11]:

$$\boldsymbol{C}_{\text{Lagrange}} = \begin{bmatrix} -3 & 27 & 27 & -3 \\ 2 & -54 & 54 & -2 \\ 12 & -12 & -12 & 12 \\ -8 & 24 & -24 & 8 \end{bmatrix}. \tag{3}$$

Recall that $\mu \in [-0.5, 0.5)$, and let $\widetilde{\mu} = \mu - 1.5 = \bar{\mu} - 1$. Then, the Newton structure in Fig. 2 can be written in the same form as (2), yielding

$$H_{\text{Newton}}(z,\widetilde{\mu}) = \widetilde{\boldsymbol{\mu}}^T \widetilde{\boldsymbol{C}} \widetilde{\boldsymbol{z}}, \tag{4}$$

where $\widetilde{\boldsymbol{\mu}} = [\begin{matrix} 1 & \widetilde{\mu} & \widetilde{\mu}(\widetilde{\mu}-1) & \widetilde{\mu}(\widetilde{\mu}-1)(\widetilde{\mu}-2) \end{matrix}]^T$, $\widetilde{\mu} \in [-2, -1)$; $\widetilde{\boldsymbol{C}}$ is a diagonal matrix whose elements are $\{1, -1, 1/2, -1/6\}$; and $\widetilde{\boldsymbol{z}} = [\begin{matrix} 1 & 1 - z^{-1} & (1 - z^{-1})^2 & (1 - z^{-1})^3 \end{matrix}]^T$.

To obtain (4) from (2), suffices to find two transformations $\boldsymbol{T}_{\boldsymbol{\mu}}$ and $\boldsymbol{T}_{\boldsymbol{z}}$ such that $\widetilde{\boldsymbol{\mu}} = \boldsymbol{T}_{\boldsymbol{\mu}} \boldsymbol{\mu}$, $\widetilde{\boldsymbol{z}} = \boldsymbol{T}_{\boldsymbol{z}} \boldsymbol{z}$, and $\widetilde{\boldsymbol{C}} = \boldsymbol{T}_{\boldsymbol{\mu}}^{-T} \boldsymbol{C} \boldsymbol{T}_{\boldsymbol{z}}^{-1}$, with $\boldsymbol{A}^{-T} = (\boldsymbol{A}^T)^{-1}$, for invertible $\boldsymbol{A}$. Given these transformations, we would have

$$\begin{aligned} H_{\text{Farrow}}(z,\mu) &= \boldsymbol{\mu}^T \boldsymbol{C} \boldsymbol{z} = \boldsymbol{\mu}^T (\boldsymbol{T}_{\boldsymbol{\mu}}^T \boldsymbol{T}_{\boldsymbol{\mu}}^{-T}) \boldsymbol{C} (\boldsymbol{T}_{\boldsymbol{z}}^{-1} \boldsymbol{T}_{\boldsymbol{z}}) \boldsymbol{z} \\ &= (\boldsymbol{T}_{\boldsymbol{\mu}} \boldsymbol{\mu})^T (\boldsymbol{T}_{\boldsymbol{\mu}}^{-T} \boldsymbol{C} \boldsymbol{T}_{\boldsymbol{z}}^{-1}) (\boldsymbol{T}_{\boldsymbol{z}} \boldsymbol{z}) = \widetilde{\boldsymbol{\mu}}^T \widetilde{\boldsymbol{C}} \widetilde{\boldsymbol{z}} = H_{\text{Newton}}(z,\widetilde{\mu}). \end{aligned}$$

These transformations can be derived in three steps: (i) find $\boldsymbol{T}_{\boldsymbol{\mu}}$; (ii) find $\boldsymbol{T}_{\boldsymbol{z}}$; (iii) check that $\boldsymbol{T}_{\boldsymbol{\mu}}$ and $\boldsymbol{T}_{\boldsymbol{z}}$ indeed transform $\boldsymbol{C}$ into $\widetilde{\boldsymbol{C}}$.

(i) The fractional delay transformation is derived in two parts. First, the fractional interval range is made identical among structures. Changes in the range of the intersample position are common and have been used, for instance, as a means to reduce complexity in the derivation of modified Farrow structures [11, 13]. Since $\mu \in [-0.5, 0.5)$ and $\widetilde{\mu} \in [-2, -1)$, $\boldsymbol{T}'_{\boldsymbol{\mu}}$ is obtained from the relation $\widetilde{\mu} = \mu - 1.5$ as

$$\begin{bmatrix} 1 \\ \widetilde{\mu} \\ \widetilde{\mu}^2 \\ \widetilde{\mu}^3 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 8 & 0 & 0 & 0 \\ -12 & 8 & 0 & 0 \\ 18 & -24 & 8 & 0 \\ -27 & 54 & -36 & 8 \end{bmatrix} \boldsymbol{\mu}. \tag{5}$$

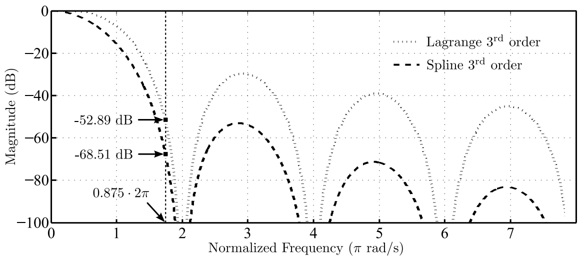


Fig. 3. Frequency response of 3rd order Lagrange and spline filters

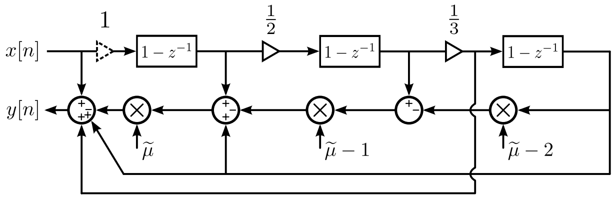


Fig. 4. Novel Newton-like spline interpolation structure

Notice that the n-th row of $\boldsymbol{T}'_{\boldsymbol{\mu}}$ collects the coefficients of the polynomial $(\mu - 1.5)^{n-1}$.

Second, the vector on the left-hand side of (5) must become $\widetilde{\boldsymbol{\mu}}$, where each element is a polynomial of $\widetilde{\mu}$. Once again, the transformation is based on the coefficients of these polynomials as in

$$\boldsymbol{T}''_{\boldsymbol{\mu}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 2 & -3 & 1 \end{bmatrix}. \tag{6}$$

Finally, $\widetilde{\boldsymbol{\mu}} = \boldsymbol{T}''_{\boldsymbol{\mu}} \boldsymbol{T}'_{\boldsymbol{\mu}} \boldsymbol{\mu}$, so that the intersample position transformation is chosen as $\boldsymbol{T}_{\boldsymbol{\mu}} = \boldsymbol{T}''_{\boldsymbol{\mu}} \boldsymbol{T}'_{\boldsymbol{\mu}}$.

(ii) The filter basis z^{-1} of the Farrow structure must be changed into the differentiator basis $1 - z^{-1}$ used by the Newton structure. This can be done using the matrix

$$\boldsymbol{T}_{\boldsymbol{z}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -3 & 3 & -1 \end{bmatrix}, \tag{7}$$

whose n-th row represents the coefficients of $(1 - z^{-1})^{n-1}$.

(iii) The matrices $\boldsymbol{T}_{\boldsymbol{\mu}}$ and $\boldsymbol{T}_{\boldsymbol{z}}$ derived in items (i) and (ii) are designed to perform the transformations $\boldsymbol{\mu} \to \widetilde{\boldsymbol{\mu}}$ and $\boldsymbol{z} \to \widetilde{\boldsymbol{z}}$, respectively. It is straightforward to see by direct evaluation that they also fulfill $\widetilde{\boldsymbol{C}} = \boldsymbol{T}_{\boldsymbol{\mu}}^{-T} \boldsymbol{C} \boldsymbol{T}_{\boldsymbol{z}}^{-1}$. By interpreting these transformations as changes in the bases of the interpolation operator $\boldsymbol{C}$, the low complexity of the Newton structure is explained by the fact that it uses bases in which the coefficient matrix is diagonal, reducing the number of operations required to evaluate the weighted inner product in (2).

Novel structure for spline interpolation: As mentioned before, the main disadvantage of the Newton structure is that it can only implement the Lagrange polynomial, which has poor frequency response. It is well known that splines have better properties for signal processing applications and converge to the ideal interpolator as their order goes to infinity [14]. Indeed, Fig. 3 compares the frequency response of 3rd order Lagrange and spline interpolators. It shows the latter displays an extra 16dB of attenuation at the $0.875 \cdot 2\pi$ normalized band edge, which corresponds to the worst-case image attenuation when interpolating a signal oversampled by 4. Spline polynomials can naturally be implemented using the Farrow structure by deriving $\boldsymbol{C}$ in (2) similar to [15]:

$$\boldsymbol{C}_{\text{spline}} = \begin{bmatrix} 1 & 23 & 23 & 1 \\ -6 & -30 & 30 & 6 \\ 12 & -12 & -12 & 12 \\ -8 & -4 & -24 & 8 \end{bmatrix}. \tag{8}$$

The proposed structure is derived by applying the same transformations from the previous section to $\boldsymbol{C}_{\text{spline}}$, yielding a Newton-like structure for spline interpolation. Explicitly,

$$\boldsymbol{T}_{\boldsymbol{\mu}}^{-T} \boldsymbol{C}_{\text{spline}} \boldsymbol{T}_{\boldsymbol{z}}^{-1} = \begin{bmatrix} 1 & 0 & \frac{1}{6} & \frac{1}{6} \\ 0 & -1 & 0 & \frac{1}{6} \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{1}{6} \end{bmatrix} = \boldsymbol{C}_{\text{LCN}}. \tag{9}$$

Notice that $\boldsymbol{C}_{\text{LCN}}$ is quasi-diagonal and that its coefficients have trivial hardware implementations. The full structure is depicted in Fig. 4 and its computational complexity is compared in Table 1 to that of the modified Farrow (suitable for Lagrange and spline) and the Newton structure (Lagrange only). Only three additional adders are necessary to turn a 3rd order Lagrange-only Newton structure into a spline interpolation structure that is largely simpler than its Farrow counterpart.

Table 1: Computational complexity

	Modified Farrow		Newton		Proposed	
	Add	Mult	Add	Mult	Add	Mult
Lagrange	11	11	6	4	–	–
Spline	11	11	–	–	9	4

Conclusion: A novel structure for spline interpolation/decimation was proposed. First, the Newton structure was derived using a series of transformations of the Farrow-Lagrange structure. These transformations were applied to the spline coefficient matrix yielding a novel Newton-like structure for spline interpolation. More general results using this matrix formulation as well as direct optimization of the coefficients in the new structure will be addressed in future works.

Acknowledgment: Dr. Nascimento's work is supported in part by FAPESP (2014/04256-2) and CNPq (306268/2014-0) grants.

D. Lamb, L. F. O. Chamon and V. H. Nascimento (*Electronic Systems Department, University of São Paulo, Brazil*)

E-mail: dlamb@usp.br, chamon@usp.br, vitor@lps.usp.br

References

1 J. Kleider and C. Steenhoek, "A new filter implementation strategy for Lagrange interpolation," in *IEEE Military Commun. Conf.*, 2014, pp. 724–730.

2 X. Huang, Y. Guo, and J. A. Zhang, "Sample rate conversion using B-spline interpolation for OFDM based software defined radios," *IEEE Trans. on Commun.*, vol. 60[8], pp. 2113–2122, 2012.

3 A. Franck, A. Graefe, T. Korn, and M. Strauss, "Reproduction of moving sound sources by wave field synthesis: An analysis of artifacts," in *Int. Conf. Audio Eng. Soc.*, 2007.

4 T. Laakso, V. Valimaki, M. Karjalainen, and U. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Process. Mag.*, vol. 13[1], pp. 30–60, 1996.

5 C. Farrow, "A continuously variable digital delay element," in *IEEE Int. Symp. on Circuits and Syst.*, 1988, pp. 2641–2645.

6 S. Tassart and P. Depalle, "Fractional delays using Lagrange interpolators," in *Int. Comput. Music Conf.*, 1996.

7 C. Candan, "An efficient filtering structure for Lagrange interpolation," *IEEE Signal Proc. Lett.*, vol. 14[1], pp. 17–19, 2007.

8 V. Lehtinen and M. Renfors, "Structures for interpolation, decimation, and nonuniform sampling based on Newton's interpolation formula," in *Int. Conf. on Sampling Theory and Applicat.*, 2009.

9 T. Hentschel and G. Fettweis, "Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals," *Frequenz*, vol. 55[5/6], pp. 185–188, 2001.

10 D. Wise and R. Bristow-Johnson, "Performance of low-order polynomial interpolators in the presence of oversampled input," *J. Audio Eng. Soc.*, 1999.

11 V. Valimaki, "A new filter implementation strategy for Lagrange interpolation," in *IEEE Int. Symp. on Circuits and Syst.*, 1995, pp. 361–364.

12 A. Franck, "Efficient algorithms and structures for fractional delay filtering based on Lagrange interpolation," *J. Audio Eng. Soc.*, vol. 56[12], pp. 1036–1056, 2009.

13 M. Hunter and W. Mikhael, "A novel Farrow structure with reduced complexity," in *IEEE Int. Midwest Symp. on Circuits and Syst.*, 2009, pp. 581–585.

14 M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16[6], pp. 22–38, 1999.

15 S. Dooley, R. Stewart, and T. Durrani, "Fast on-line B-spline interpolation," *IET Electron. Lett.*, vol. 35[14], pp. 1130–1131, 1999.