Efficient Digit-Serial Rational Function Approximations and Digital Filtering Applications

Oskar Mencer, Martin Morf, Albert Liddicoat, Michael J. Flynn

Computer Systems Laboratory
Stanford, CA 94305, USA
email: {oskar,morf,lidicoat,flynn}@arithmetic.stanford.edu

Abstract

Continued Fractions (CFs) efficiently compute digit serial rational function approximations. Traditionally, CFs are used to compute homographic functions such as $y = \frac{ax+b}{cx+d}$, given continued fractions x, y, and integers a, b, c, d. Recent improvements in the implementation of CF algorithms open up their use to many digital filtering applications in both software and hardware. These improvements of CF algorithms include error control, more efficient number representation and the associated efficient conversions.

Continued fractions have been applied to digital filters in the frequency domain. These techniques include methods to compute optimal coefficients for rational transfer functions of digital filters and the realization of ladder forms for digital filtering.

We propose a time domain digital filtering technique that incorporates continued fraction arithmetic units. For the FIR filter example chosen, the proposed technique achieves a 45-50% reduction of the mean square error of the transfer function.

1. Introduction

The purpose of this paper is to show the connections between continued fractions(CFs) and digital filters. In addition, recent improvements in the implementation of CF algorithms open up their use to many digital filtering applications in software and hardware.

1.1. Rational Approximations and Continued Fractions

Signal processing and other data-intensive applications require fast approximation of specific elementary and non-elementary functions. In general, functions are approximated either with polynomials or rational functions (the ratio of two polynomials). More particularly, for many functions rational approximations converge faster than polynomial approximations.

Rational approximations offer efficient evaluation of elementary functions represented by the ratio of two polynomials.

$$f(x) \sim \frac{((a_n x + a_{n-1})x + \cdots a_1)x + a_0}{((b_m x + b_{m-1})x + \cdots b_1)x + b_0} = \qquad (1)$$

$$\frac{c_n x^n + c_{n-1}x^{n-1} \cdots c_1 x + c_0}{d_m x^m + d_{m-1}x^{m-1} \cdots d_1 x + d_0}$$

[7] evaluates rational approximations with a latency of $\max(m, n)$ multiply-add (MA) operations and a final division. If multiply and divide operations are regarded equal (dependent on implementation technology) continued fractions can provide more efficient structures for the evaluation of rational approximations. Every rational approximation can be transformed (see e.g. [1][4][10]) into a *finite continued fraction*: for $a_i, b_i \in \mathcal{R}$

$$\frac{A_n}{B_n} = a_0 + \frac{b_1}{a_1 + \frac{b_2}{a_2 + \cdots \frac{b_n}{a_n}}} = a_0 + \frac{b_1|}{|a_1} + \frac{b_2|}{|a_2} + \cdots + \frac{b_n}{a_n}$$

Every finite continued fraction can be transformed (see e.g. [1][4][10]) into a *simple continued fraction* form with partial quotients $b_i = 1$. The series notation $[a_0; a_1, \ldots a_n]$ denotes *simple continued fractions*.

The connection between rational approximations and simple CFs is expressed in the next equivalence.

Equivalence 1 *(Wall[5]) Given the ratio of two polynomials $\frac{f_0(x)}{f_1(x)}$,*

$$\begin{aligned} \frac{f_0(x)}{f_1(x)} &= \frac{a_{00}x^n + a_{01}x^{n-1} + \cdots + a_{0n}}{a_{11}x^{n-1} + a_{12}x^{n-2} + \cdots + a_{1n}} \equiv \\ &\equiv [r_1 x + s_1, r_2 x + s_2, \cdots, r_n x + s_n] \end{aligned}$$

with all $a_{ij} \neq 0$, and $r_i \neq 0$.

Equivalence 1 shows the main advantage of continued fractions. The powers of x in the polynomial representation are reduced to linear terms for each continued fraction digit.

Software packages such as MapleV[11] compute *minimax* coefficients a_{ij} automatically using Remez's[14] method. Many elementary functions have straightforward simple continued fraction expansions – the rational equivalent to Taylor series expansion around a point x_0.

Although we are still missing a general theory explaining the connection between transcendental functions and continued fractions([4], see Introduction by Peter Henrici), we can already take advantage of the known, isolated "gems" of continued fraction expansions such as:

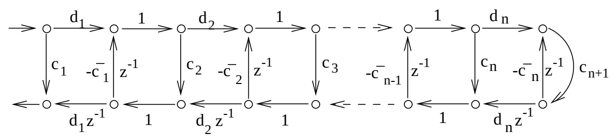

Figure 1: *The figure shows a ladder form for digital filtering, taken from [6], with complex c_i.*

$$\tan(x) = \left[0; \frac{1}{x}, -\frac{3}{x}, \frac{5}{x}, -\frac{7}{x}, \frac{9}{x}, -\frac{11}{x}, \ldots\right] \qquad (2)$$

A finite continued fraction with i partial quotients can always be transformed into a ratio $\frac{A_i}{B_i}$ with:

$$A_i = a_i A_{i-1} + b_i A_{i-2} \qquad (3)$$
$$B_i = a_i B_{i-1} + b_i B_{i-2} \qquad (4)$$

where $\frac{A_{i-1}}{B_{i-1}}$ corresponds to the value of the same continued fraction without the i^{th} partial quotient. Initial conditions are $A_0 = a_0$, $B_0 = 1$, $A_{-1} = 1$, and $B_{-1} = 0$. These equations show the link between ratios A/B and their continued fraction expansion. The equations serve as the basis for straightforward conversion between continued fractions and rational values. For a more detailed introduction to the theory of continued fractions see for example [8][9] or standard introductory texts such as [1][4][10].

1.2. CFs and Digital Filters in the Frequency Domain

Continued fractions can be applied to several tasks in digital filtering:

- computing coefficients[13] and poles[6] of the transfer function
- proof/test of stability of ladder forms[6]
- realization of ladder forms for digital filtering[12]

On the transfer function level, Remez's[14] method enables us to compute optimal (minimax) coefficients for any rational approximation – in our case the rational transfer function of a digital filter. A description of Remez's method for digital filter design can be found in [13].

The poles of the transfer function can be computed with the FG-algorithm as shown in [4]. [6] includes examples for finding poles of transfer functions in signal detection and speech analysis.

Stability of ladder forms with (reflection) coefficients $|c_k| < 1$ follows from the following continued fraction[6] derived from the transfer function of the ladder form shown in figure 1:

$$H(z) = c_1 + \cfrac{(1-|c_1|^2)z^{-1}}{\overline{c_1}z^{-1} + \cfrac{1}{c_2 + \cfrac{(1-|c_2|^2)z^{-1}}{\overline{c_2}z^{-1} + \cdots + \frac{1}{c_{n+1}}}}} \qquad (5)$$

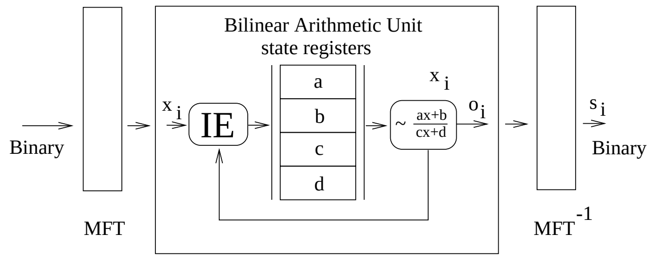

Figure 2: *The figure shows the state machine for the iteration equations (IE) of the SEPA algorithm.*

[12] goes from a rational approximation of the desired transfer function, to a continued fraction approximation of the transfer function, such as the form shown in Equivalence 1. Finally the paper shows how to build various ladder forms from the transfer functions in continued fraction form.

2. Continued Fraction Arithmetic and the M-log-Fraction Transform (MFT)

This section briefly summarizes a state-of-the-art continued fraction arithmetic algorithm developed in [9]. The main objective is to compute $f(x) = \frac{ax+b}{cx+d}$ with $[o_0, o_1, o_2 \ldots] = f([x_0, x_1, x_2, \ldots])$ where x_i's are the partial quotients of the simple input CF, and $o_i = f(x_i, \texttt{state})$ are the partial quotients of the output CF. The positional algebraic algorithm introduced in [15] requires one *state register* for each coefficient of $f(x)$.

- to *consume* an input quotient x_i apply $T'(x) = T(x_i + \frac{1}{x})$.
- to *produce* an output quotient o_i apply $T''(x) = \frac{1}{T(x) - o_i}$.

From transformations T' and T'' we obtain the following iteration equations:

$$\mathtt{a}_{\mathtt{i}+1} = \mathtt{c}_\mathtt{i}\mathtt{x}_\mathtt{i} + \mathtt{d}_\mathtt{i} \qquad \mathtt{b}_{\mathtt{i}+1} = \mathtt{c}_\mathtt{i} \qquad (6)$$
$$\mathtt{c}_{\mathtt{i}+1} = \mathtt{a}_\mathtt{i}\mathtt{x}_\mathtt{i} + \mathtt{b}_\mathtt{i} - \mathtt{o}_\mathtt{i}(\mathtt{c}_\mathtt{i}\mathtt{x}_\mathtt{i} + \mathtt{d}_\mathtt{i}) \qquad \mathtt{d}_{\mathtt{i}+1} = \mathtt{a}_\mathtt{i} - \mathtt{o}_\mathtt{i}\mathtt{c}_\mathtt{i}$$

The state-machine is shown in figure 2. Drawing the iteration equations in a signal-flow-graph (an unconventional way of looking at arithmetic algorithms) we obtain figure 3. The signal flow graph resembles a markov-chain, but more interestingly resembles the ladder form for stable digital filters shown above.

Details on the derivation of iteration equations for higher degree polynomials and further explanations can be found in [8][9]. For efficient hardware implementation of the above algorithm we use the M-log-Fraction Transform (MFT) to instantly convert between digits of binary numbers and the digits of the M-log-Fraction:

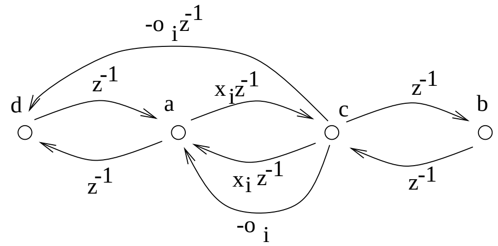

Figure 3: *The state registers a, b, c, d are the states of the shown signal-flow-graph. z^{-1} is a delay operator linking consecutive iterations.*

Corollary 2 *(from [9]) Signed-Digit Binary M-log-Fraction: A binary number B_R with n digits, and $s_i \in \{+1, -1\}$ is equivalent to a simple continued fraction with n partial quotients as follows:*

$$B_R = s_1 2^{-1} + s_2 2^{-2} + s_3 2^{-3} + s_4 2^{-4} + \ldots + s_n 2^{-n} \equiv$$
$$[0; s_1 2^1, -(s_1 2^{-1} + s_2 2^0), (s_2 2^0 + s_3 2^1), -(s_3 2^{-1} + s_4 2^0), \ldots$$
$$\pm(s_{n-1} 2^{-M_{n-1}} + s_n 2^{M_n})]$$

Applying the M-log-Fraction to the input and output CF in the positional algebraic algorithm yields rational arithmetic units based on 'shift and add' primitives.

3. CFs and Digital Filters in the Time Domain Applying the MFT to Digital Filtering

The previous section proposes a digit-serial continued fraction arithmetic unit computing the bilinear transformation $f(x) = \frac{ax+b}{cx+d}$. This arithmetic unit simplifies to a multiplication ($b = c = 0, d = 1$) as a special case of a bilinear transformation. One can use this bilinear arithmetic unit in digital filtering applications. The non-linear (bilinear) transformation allows us to model non-linearities and provides more degrees of freedom for adaptive filter applications. Each tap of a FIR filter contains a multiplier, $f(x) = a \cdot x$. We propose the *MFT filter* replacing the linear "tap" with a bilinear transformation, $f(x) = \frac{ax+b}{cx+d}$.

Digital filters can be constructed with the continued fraction arithmetic unit as shown in Figure 5. This structure simplifies to a finite impulse response (FIR) filter when $b_i = c_i = 0$ and $d_i = 1$. Starting with the optimal coefficients a_i of the FIR filter the MFT coefficients (a_i, b_i, c_i, and d_i) for the proposed filter can be adapted to improve or alter the characteristics of the filter. Note that the resulting filter is non-linear, but closely related to the initial FIR filter.

We use the mean square error (MSE) as the metric to compare the filter performance. Given a target transfer function $F(z)$ (for low pass filter) and the actually achieved transfer function $H(z)$, the MSE can be computed as follows:

$$F(z) = \begin{cases} 1 & z <= \texttt{cutoff} \\ 0 & z > \texttt{cutoff} \end{cases} \qquad (7)$$

$$MSE = \sum \frac{(|H(z)| - F(z))^2}{\texttt{\# of frequencies}} \qquad (8)$$

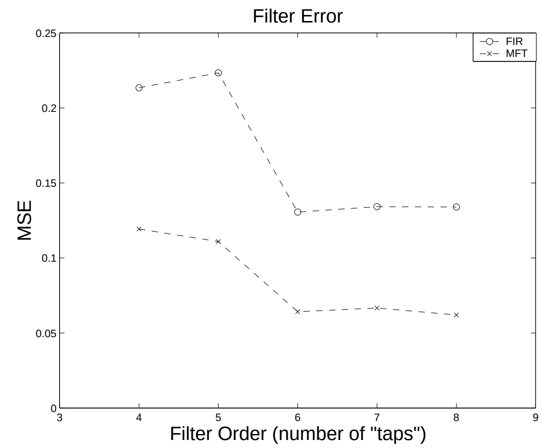

Figure 4: *The figure shows the mean square error (MSE) for optimized FIR filters and MFT filters for a varying number of taps (x-axis).*

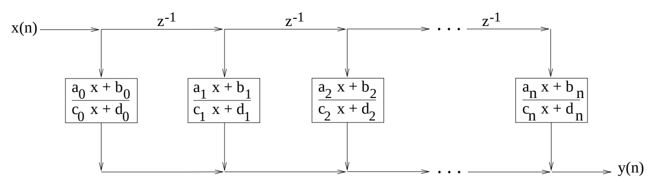

Figure 5: *The figure shows the MFT filter. Each tap of an FIR is replaced by a bilinear arithmetic unit.*

It is possible to improve the MSE of an FIR filter with the proposed MFT filter. We demonstrate a reduction of the MSE by 45% to 50% over that of the same order FIR filter designed using the Parks-McClellan optimal design algorithm (see for example [13]). Figure 6 shows the magnitude response of an optimized MFT filter and that of the standard FIR filter.

Figure 4 shows a comparison of the MSE for the proposed MFT filter and the standard FIR filter. The MSE of the 4^{th} order MFT filter is less than that of the 8^{th} order FIR filter.

4. Conclusions

Continued fractions model digital filters in the frequency domain (rational transfer functions). Continued fractions also enable efficient rational arithmetic units based on the MFT. Applying the basic $\frac{ax+b}{cx+d}$ unit to the time domain of FIR filters leads to MFT-based non-linear filters that are closely related to their respective FIR relative. These MFT-based filter enable us to model non-linear systems, while using the linear FIR filter as a starting point.

Surprisingly, even in our initial quite simplistic experiment the MFT-based filters reduces the mean square error(MSE) of a low-pass filter by $45 - 50\%$. In addition, the MFT-based filter maintains phase linearity in the pass band while improving the MSE.

The general technique presented in this paper can be

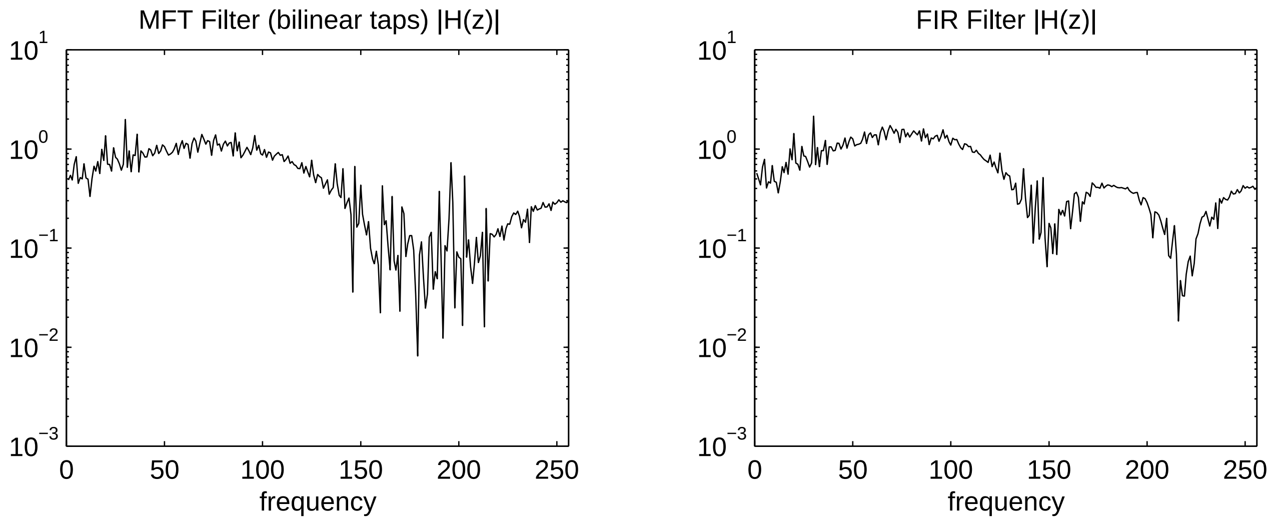

Figure 6: *The figure shows the amplitude response of an 8-top FIR filter and an 8-'tap' MFT filter.*

naturally extended to any linear filter. The next step is to find application domains that can make optimal use of the non-linearities modeled by the MFT-filter.

5. Acknowledgments

Thanks to Mitch Weintraub for valuable suggestions and comments. This research is partly supported by a grant from Compaq Systems Research Center.

6. References

[1] O. Perron, *Die Lehre von den Kettenbrüchen,* Band I,II , Teubner Verlag, Stuttgart, 1957.

[2] A. Hurwitz, *Über die Kettenbrüche, deren Teilnenner arithmetische Reihen bilden,* Vierteljahrsschrift d. naturforsch. Gesellschaft, Zürich, Jahrgang 41, 1896.

[3] R.W. Gosper, R. Schroeppel, M. Beeler, *HAKMEM,* Continued Fraction Arithmetic, MIT AI Memo 239, Feb. 1972.

[4] W.B. Jones, W.J. Thron, *Continued Fractions: Analytic Theory and Applications,* Encyclopedia of Mathematics and its Applications, Vol. 11, Addison-Wesley, Reading, Mass., 1980.

[5] H.S. Wall, *Analytic Theory of Continued Fractions,* Chelsea Publishing Company, Bronx, N.Y., 1948.

[6] W.B. Jones, A. Steinhardt, *Digital Filters and Continued Fractions,* Lect. Notes In Math, No. 932, p. 129, Springer-Verlag, Berlin, 1982.

[7] I. Koren, O. Zinaty, *Evaluating Elementary Functions in a Numerical Coprocessor Based on Rational Approximations,* IEEE Trans. on Computers, Vol. 39, No. 8, Aug. 1990.

[8] O. Mencer, M. Morf, M. J. Flynn, *Precision of Semi-Exact Redundant Continued Fraction Arithmetic for VLSI,* SPIE Advanced Signal Processing Algorithms, Architectures, and Implementations IX, Denver, July 1999.

[9] O. Mencer, M. J. Flynn, M.Morf, *The M-log-Fraction Transform (MFT) for Computer Arithmetic,* Technical Report, CSL-TR-99-784, Stanford, Sept. 1999. http://elib.stanford.edu/

[10] L. Lorentzen, H. Waadeland, *Continued Fractions with Applications,* Studies in Computational Mathematics 3, North-Holland, 1982.

[11] Waterloo Maple Inc., *Maple V,* http://www.maplesoft.com/.

[12] S.K. Mitra, R.J. Sherwood, *Canonic realizations of digital filters using the continued fraction expansion,* IEEE Trans. on Audio and Electroacoustics, Vol. AU-20, No. 3, p. 185-197, Aug. 1972.

[13] C. Britton Rorabaugh, *Digital Filter Designer's Handbook with C++ Algorithms, 2nd Edition ,* McGraw-Hill, 1997.

[14] E.Y. Remez, *General Computational Methods of Chebyshev Approximation,* Kiev, 1957. (see also L.A. Lyusternik, *Computing Elementary Functions,* Pergamon Press, 1965.)

[15] J.E. Vuillemin, *Exact Real Computer Arithmetic with Continued Fractions,* IEEE Trans. on Computers, Vol. 39, No. 8, Aug. 1990.