Massive Parallelization of IIR Filters

Hans Georg BRACHTENDORF

Signal Processing Laboratory, University of Applied Sciences of Upper Austria, Softwarepark 11, 4232 Hagenberg, Austria

hans-georg.brachtendorf@fh-hagenberg.at

Submitted May 21, 2026 / Accepted September 22, 2026 / Online first November 20, 2026

Abstract. *Infinite-impulse-response (IIR) filters offer several advantages over finite-impulse-response (FIR) filters, as they have a lower filter order, fewer filter coefficients, and thus lower hardware requirements and lower power consumption. In addition, there are filter design techniques for approximately linear-phase IIR filters, which mitigate the most serious drawback of these filters. The subclass of wave digital filters is also characterized by low implementation complexity.*

However, due to their recursive structure, IIR filters are difficult to parallelize, and to date, there have been only a few publications addressing this bottleneck. They are either less suitable for massive parallelization or require an enormous amount of preprocessing effort.

In this paper, a novel technique is derived that is based on the Sherman-Morrison-Woodbury formula and the FFT and requires $O(n \log n)$ operations. The method requires only minimal preprocessing and is well-suited for massive parallelization. Adaptive switching of the filter coefficients is also easily achievable. A bit true MATLAB library has been developed.

Keywords

FFT based massive parallel IIR filtering, superfast $O(n \log n)$ algorithm, Sherman-Morrison-Woodbury formula, wave digital filters, bit true MATLAB library, GPU programming

1. Introduction

Infinite impulse response filters require generally a significantly lower filter order compared with finite impulse filters for fulfilling the design requirements. There exist also design techniques for IIR filters which guarantee nearly constant group delay, i.e., approximately linear phase, in the passband(s), relaxing the most severe drawbacks of these filters. This makes them attractive as alternative to FIR filters. Moreover their implementation as wave digital filters (WDF) [1–4] outperforms alternative techniques. Building blocks for WDFs are combinations of all-pass filters with low implementation costs. Nonetheless, the parallelization for high speed signal processing remains an issue, due to their recursive structure.

Several attempts have been made to parallelize IIR filters. An obvious way for parallelization is to break down the transfer function by partial fraction expansion, see, e.g., [5–7]. Partial fraction expansion leads to second order systems (SOS), their evaluation requires less clock cycles. However the degree of parallelism is small. The main problem of IIR filters, their feedback structure, which hinders a massive parallelization, remains unsolved.

One class of block IIR filtering is based on the Fast Fourier Transformation (FFT) [8–10]. Gnanasekaran et al. [10] showed that the method of Gold et al. [9] is a special case of the technique proposed by Burrus [8]. Essentially, a transformation to a block state space system is performed by preprocessing. This transformation requires solving linear systems of the same size as of the parallelization order. This block state space system is then solved with conventional techniques originally developed for parallel FIR filtering such as the well-known overlap-add and overlap-save methods. These techniques are suited for a massive parallization; the huge amount of preprocessing is still a disadvantage, especially when filters with different filter coefficients or adaptive filtering comes into account.

Parhi et al [11] developed a parallelization technique by introducing additional poles of the transfer function which are compensated by additonal zeros. Since these extra zeros lead to an augmented non-recursive (FIR) part of the transfer function, this method is only useful for a low degree of parallelization. A similar result was obtained by Brent et al [12] employing the companion matrix and a generalization of the geometric series. For an expansion in a geometric series see also [13]. As in [11], the parallelization is achieved by an additional non-recursive part. These methods [11–14] are less suited for a massive parallelization of IIR filters due to their augmented FIR parts.

This paper derives a novel technique employing the Sherman-Morrsion-Woodbury formula (SMW) in combination with a similarity transform by the FFT. Unlike previous FFT based techniques [8–10], no solutions of large linear systems of equations are required. Only a decomposition of the transfer function into low order subsystems (e.g., 1st or

DOI: 10.13164/re.2026.0579

2nd order transfer functions) is of advantage. This however is generally recommended since such systems are insensitive to rounding errors in quantized arithmetic [15]. Only a small amount of preprocessing is necessary, which can be done offline. The FFT of signals requires $O(n \log n)$ operations, the remaining matrix-vector multiplications require $O(n)$ flops.

Shortcomings of existing methods are discussed in Sec. 3. The SMW based parallelization has in common with existing block based techniques as the FFT is used for similarity transformation. Unlike block based techniques no zero padding is needed, reducing the computation amount from $O(2n \log 2n)$ to $O(n \log n)$, as discussed below.

The technique has been investigated in a bit true MATLAB library, using a canonical signed digit (CSD) representation [16].

This paper is organized as follows: in Sec. 2 the problem is stated. In Sec. 3 the shortcomings of existing methods are discussed. In Sec. 4 the novel parallelization technique based on the SMW formula is presented. Finally, Section 5 deals with a variant of the SMW based parallelization, which is also basis of the bit-true MATLAB realization.

1.1 Nomenclature

Let $\mathbf{F}$ be the coefficient matrix of the Discrete Fourier Transformation (DFT) and specifically $\mathbf{F}_n$ the DFT of dimension $n \times n$. We scale the DFT such that $\mathbf{F}$ is Unitarian, i.e. $\mathbf{F}^{-1} = \mathbf{F}^{\mathrm{H}}$, where $^{\mathrm{H}}$ is the Hermitian operator.

Moreover let $\mathbf{I}$ or specifically $\mathbf{I}_n$ be the identity matrix and $\mathbf{J}$, $\mathbf{J}_n$ the anti-diagonal identity matrix, respectively. An $n \times n$ matrix $\mathbf{P}$ represents a permutation matrix, specifically

$$\mathbf{P} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix} \tag{1}$$

and let $\mathbf{P}_m = \mathbf{P}^m$. Moreover, let

$$\mathbf{N} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix} \tag{2}$$

be a lower nilpotent matrix of dimension $n \times n$, i.e., $\mathbf{N}^n = 0$, the zero matrix.

Furthermore, let $\mathbf{e}_0 = (1, 0, \cdots, 0)^{\mathrm{T}}$ and $\mathbf{e}_{n-1} = (0, 0, \cdots, 1)^{\mathrm{T}}$ canonical column vectors of $\dim(\mathbf{e}) = n$ and $\mathbf{u}$, $\mathbf{v}$ arbitrary column vectors, respectively. Below, we use the identity

$$\mathbf{P} = \mathbf{N} + \mathbf{e}_0\, \mathbf{e}_{n-1}^{\mathrm{T}}. \tag{3}$$

2. Problem Formulation

We consider IIR filters of order M with transfer function

$$H(z) = \frac{\sum_{m=0}^{M} b_m\, z^{-m}}{\sum_{n=0}^{M} a_n\, z^{-n}} = \frac{B(z^{-1})}{A(z^{-1})}, \quad a_0 = 1 \tag{4}$$

where A, B are algebraic polynomials in z^{-1}.

It is known that a direct implementation of the filter either in the 1st or 2nd canonical form is very sensitive to a quantization of the a and b coefficients when M is large and therefore not suitable for an implemenation in digital hardware. Therefore, the transfer function is rewritten as a product

$$H(z) = \prod_{i=1}^{p} \tilde{H}_i(z) \tag{5}$$

of transfer functions $\tilde{H}_i(z)$ of first or second order filter sections, namely

$$\begin{aligned} \tilde{H}_i(z) &= \frac{\tilde{b}_{i0} + \tilde{b}_{i1}\, z^{-1}}{1 + \tilde{a}_{i1}\, z^{-1}} \\ \tilde{H}_i(z) &= \frac{\tilde{b}_{i0} + \tilde{b}_{i1}\, z^{-1} + \tilde{b}_{i2}\, z^{-2}}{1 + \tilde{a}_{i1}\, z^{-1} + \tilde{a}_{i2}\, z^{-2}} \end{aligned} \tag{6}$$

depending on the position of the poles and zeros. Poles and zeros on the real axis are realized as first order filters whereas conjugate complex ones are implemented as second order filters sections, respectively. The generic case is therefore the 2nd order block. In case of adaptive IIR filters, the control of stability from (6) is simple, which is another advantage of that form.

The product of the low degree polynomials from (5) are carried out in a series or cascade implementation. We note that in quantized arithmetic also the sequential arrangements must be taken into account. We refer to the vast literature on this topic. Below, we drop the subscript i for simplicity. After long division one obtains

$$\begin{aligned} \tilde{H}(z) &= b_0 + \frac{b_1\, z^{-1}}{1 + a_1\, z^{-1}}, & H(z) &= \frac{b_1\, z^{-1}}{1 + a_1\, z^{-1}} \\ \tilde{H}(z) &= b_0 + \frac{b_1\, z^{-1} + b_2\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}}, & H(z) &= \frac{b_1\, z^{-1} + b_2\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}} \end{aligned} \tag{7}$$

where $b_0 = \tilde{b}_0$, $b_1 = \tilde{b}_1 - \tilde{a}_1\, \tilde{b}_0$, $b_2 = \tilde{b}_2 - \tilde{a}_2\, \tilde{b}_0$ and $a_1 = \tilde{a}_1$, $a_2 = \tilde{a}_2$, respectively.

The state space formulation of (4) with the vector of state spaces $\mathbf{Z} \in \mathbb{R}^M$ reads

$$\begin{aligned} \mathbf{Z}(z) &= z^{-1}\, \mathbf{A}\, \mathbf{Z}(z) + \mathbf{B}\, \mathbf{X}(z) \\ \mathbf{Y}(z) &= z^{-1}\, \mathbf{C}\, \mathbf{Z}(z) + \mathbf{D}\, \mathbf{X}(z) \end{aligned} \tag{8}$$

where z^{-1} represents the delay operator. In time domain we get

$$\begin{aligned} \mathbf{Z}[k+1] &= \mathbf{A}\, \mathbf{Z}[k] + \mathbf{B}\, \mathbf{x}[k+1] \\ \mathbf{y}[k+1] &= \mathbf{C}\, \mathbf{Z}[k] + \mathbf{D}\, \mathbf{x}[k+1]. \end{aligned} \tag{9}$$

Therein, the **ABCD** matrices read

$$\mathbf{A} = \begin{pmatrix} -a_1 & 1 & 0 & \cdots & \cdots & 0 \\ -a_2 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & 0 \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ -a_{M-1} & 0 & \cdots & \cdots & 0 & 1 \\ -a_M & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1 \\ \vdots \\ b_M \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}, \qquad \mathbf{D} = \begin{pmatrix} b_0 \end{pmatrix} \tag{10}$$

for the 2nd canonical form. A similar formulation holds true for the 1st canonical form.

We assume below that the signals are partitioned in blocks, i.e. $\mathbf{x}_\ell = (x_{\ell,i})_{i=0}^{n-1}$, $\ell = 1, 2, \ldots$ of length $n > M$ for the input signal and the same holds true for all other signals introduced next.

3. Discussion of Existing Techniques

3.1 Shortcomings of the Pole-Zero Cancellation Techniques

The methods in [11, 12, 14] have in common that the parallelization is achieved by a pole-zero cancellation, which is explained for a first order system

$$H(z) = \frac{1}{1 + a\, z^{-1}} = \frac{\sum_{\ell=0}^{n-1} a^\ell z^{-\ell}}{1 + a^n\, z^{-n}}. \tag{11}$$

For a large degree of parallelization n must be large and since $|a| < 1$, the term a^n is neglibible for large n. Moreover the filter becomes essentially a non-recursive (FIR) type. The filter order increases significantly from one to n. It is questionable why not to switch to a solely FIR filter instead.

3.2 Shortcomings of Parallel Realization IIR Filters

Rewriting (4) employing partial fraction expansion results into a sum of 1st or 2nd order systems $H_i(z)$

$$H(z) = \sum_{i=1}^{p} H_i(z) \tag{12}$$

and offers only a low degree of parallization, since M in (4) is moderate for IIR filters. The general problem of the dependencies caused by the recursive structure is unsolved. There exist numerous papers on implementation aspects, e.g., [5–7].

3.3 Shortcomings of Existing Block-Based Techniques

The methods proposed in [11–14] are block oriented and FFT based. They do essentially the following. Let the signals be dissected into blocks of length n and denoted as $\mathbf{x}_\ell$ etc. The filtering $\mathbf{Y}(z) = \mathbf{H}(z)\,\mathbf{X}(z)$ can be rewritten in block processing form

$$\sum_{l=0}^{L-1} \mathbf{A}_\ell\, \mathbf{y}_{k-\ell} = \sum_{l=0}^{L-1} \mathbf{B}_\ell\, \mathbf{x}_{k-\ell} \tag{13}$$

For sufficiently large n, L is two, i.e., the memory of the filter is one block. Moreover $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$ are banded matrices of a Toeplitz structure, which results directly from (4). Rewriting (13), one gets

$$\mathbf{y}_k = -\sum_{l=1}^{L-1} \mathbf{A}_0^{-1}\, \mathbf{A}_\ell\, \mathbf{y}_{k-\ell} + \sum_{l=0}^{L-1} \mathbf{A}_0^{-1}\, \mathbf{B}_\ell\, \mathbf{x}_{k-\ell}. \tag{14}$$

The matrices $\mathbf{A}_0^{-1}\,\mathbf{A}_\ell$ and $\mathbf{A}_0^{-1}\,\mathbf{B}_\ell$ are again Toeplitz matrices, one can interpret them as the linear algebra analogue to the convolution of the a and b coefficients of (4). Since $\mathbf{A}_0$ is always regular, the inverse matrix $\mathbf{A}_0^{-1}$ exists. We refer to the papers [11–14] for details. Since these matrices are also Toeplitz, the calculations in (14) can be performed in parallel by the overlap-add or overlap-save techniques employing the FFT. Nevertheless, three disadvantages must be reported: firstly, the calculations of $\mathbf{A}_0^{-1}\,\mathbf{A}_\ell$ and $\mathbf{A}_0^{-1}\,\mathbf{B}_\ell$ require a tremendous preprocessing step, making it less easy to switch from one filter coefficients to another. Secondly, employing fixed point arithmetric, the calculations are error prone. Thirdly, specifically $\mathbf{A}_0^{-1}\,\mathbf{A}_1$ is essentially a full lower triangular matrix. The fast calculation of (14) via the FFT requires therefore a zero padding to vector lengths of $2n$, i.e., an afford of $O(2n\,\log 2n)$ operations.

The techniques proposed below uses the fact that the matrices differ from circulant matrices only by a rank-1 or rank-2 update. Since no zero padding is required, the FFT can be performed in $O(n\,\log n)$ operations, compared with $O(2n\,\log 2n)$ of the block based technique above, since circulant matrices can be diagonalized via similarity transform by the FFT efficiently. The asymptotic gain is therefore a factor of 2. After a small amount of preprocessing, the calculation is of linear afford $O(n)$. The preprocessing consists mainly of reformulating (4) in SOS blocks (7). This however is recommended anyway, since it is well known that SOS are much less sensitive w.r.t. quantization of the filter coefficients, see, e.g., [16]. The coefficients of the SOS blocks appear directly as parameters in the algorithm, a switch to other filter coefficients is therefore much easier, since cumbersome matrix operations are not required. A disadvange is, that for any SOS a rank-2 update via the Sherman-Morrison-Woodbury formula (23) is needed, though this update requires linear computational afford $O(n)$.

Below, two variants are discussed, both employing the Sherman-Morrison-Woodbury formula and having the same computational afford. A third variant is obvious and only mentioned briefly.

4. Derivation of the Method

4.1 State Space Formulation

Since we partitioned the filter above in a cascade of 1st and 2nd order subsystems, we consider next only the case for $M = 2$. The case $M = 1$ follows immediately from the derivation as special case. Furthermore, let $\mathbf{u} = \mathbf{e}_0$ and $\mathbf{v} = \mathbf{e}_{n-1}$.

After partitioning the signals in blocks of length $n > M$, the state space formulation reads in time domain

$$\begin{aligned} \mathbf{Z}[\ell] &= \mathbf{A}_1\,\mathbf{Z}[\ell] + \mathbf{A}_2\,\mathbf{Z}[\ell-1] + \mathbf{B}\,x[\ell] \\ \mathbf{y}[\ell] &= \mathbf{C}_1\,\mathbf{Z}[\ell] + \mathbf{C}_2\,\mathbf{Z}[\ell-1] + \mathbf{D}\,x[\ell] \end{aligned} \tag{15}$$

where

$$\begin{aligned} &\mathbf{A}_1 = \begin{pmatrix} -a_1\,N & N \\ -a_2\,N & 0 \end{pmatrix},\ \mathbf{A}_2 = \begin{pmatrix} -a_1\,\mathbf{u}\mathbf{v}^{\mathrm{T}} & \mathbf{u}\mathbf{v}^{\mathrm{T}} \\ -a_2\,\mathbf{u}\mathbf{v}^{\mathrm{T}} & 0 \end{pmatrix},\ \mathbf{B} = \begin{pmatrix} b_1\,I_n \\ b_2\,I_n \end{pmatrix} \\ &\mathbf{C}_1 = \begin{pmatrix} N & 0 \end{pmatrix},\ \mathbf{C}_2 = \begin{pmatrix} \mathbf{u}\mathbf{v}^{\mathrm{T}} & 0 \end{pmatrix}, \qquad \mathbf{D} = \begin{pmatrix} b_0 \end{pmatrix} \end{aligned} \tag{16}$$

and $\mathbf{Z}[\ell] = (\mathbf{Z}_1[\ell],\, \mathbf{Z}_2[\ell])^{\mathrm{T}} \in \mathbb{R}^{2n}$. Here the nilpotent matrix $\mathbf{N} \in \mathbb{R}^{n\times n}$ (2) represents a delay operator (z^{-1}) within a signal block in matrix form and represents the recursive structure. Furthermore the matrix $\mathbf{E} = \mathbf{u}\mathbf{v}^{\mathrm{T}}$ selects the last element of the 1st row, i.e., $\mathbf{E} = (e_{ij}) \in \mathbb{R}^{n\times n}$ where $e_{1n} = 1$ and zero otherwise. Moreover, $\mathbf{A}_1,\, \mathbf{A}_2 \in \mathbb{R}^{2n\times 2n}$, $\mathbf{C}_1,\, \mathbf{C}_2 \in \mathbb{R}^{n\times 2n}$, $\mathbf{B} \in \mathbb{R}^{2n\times n}$ and $D \in \mathbb{R}$. The sparse matrix $\mathbf{A}_2$ represents the coupling between the blocks indexed ℓ and $\ell - 1$ and is of rank two. From the blockwise state space equation one has to solve for $\mathbf{Z}[\ell]$

$$(\mathbf{I}_{2n} - \mathbf{A}_1)\,\mathbf{Z}[\ell] = \mathbf{A}_2\,\mathbf{Z}[\ell-1] + \mathbf{B}\,x[\ell] = \tilde{b} \tag{17}$$

where $\tilde{b} = \mathbf{A}_2\,\mathbf{Z}[\ell-1] + \mathbf{B}\,x[\ell]$ is the right-hand-side of the equation system. Due to the nilpotent matrix $\mathbf{N}$, the standard way for solving the state space equation requires a serial calculation.[1] To circumvent this serial computation, we rewrite the equation by a circulant extension with a rank-2 update, i.e.

$$\left(\mathbf{I}_{2n} + \begin{pmatrix} a_1\,\mathbf{P} & -\mathbf{P} \\ a_2\,\mathbf{P} & 0 \end{pmatrix}\right)\,\mathbf{Z}[\ell] - \begin{pmatrix} a_1\,\mathbf{u}\mathbf{v}^{\mathrm{T}} & -\mathbf{u}\mathbf{v}^{\mathrm{T}} \\ a_2\,\mathbf{u}\mathbf{v}^{\mathrm{T}} & 0 \end{pmatrix}\,\mathbf{Z}[\ell] = \tilde{b} \tag{18}$$

using (3). The 1st matrix on the left-hand-side is a regular block circulant matrix which can be diagonalized by a similarity transformation employing the DFT. The second matrix is a sparse and singular rank-2 matrix.

Since $\mathbf{u}$, $\mathbf{v}$ are canonical vectors, $\hat{\mathbf{u}} = \mathbf{F}\,\mathbf{u}$ and $\hat{\mathbf{v}} = \mathbf{F}\,\mathbf{v}$ correspond to the 1st or last columns of the DFT coefficient matrix and $\mathbf{F}\,\mathbf{u}\,\mathbf{v}^{\mathrm{T}}\,\mathbf{F}^{\mathrm{H}} = \hat{\mathbf{u}}\,\hat{\mathbf{v}}^{\mathrm{H}}$. Hence $\hat{\mathbf{u}}$ is unity. The transformation

$$\mathbf{F}\,\mathbf{P}\,\mathbf{F}^{\mathrm{H}} = \mathrm{diag}\{(w^i)_{i=0}^{n-1}\} = \hat{\mathbf{P}} \tag{19}$$

where $w = \exp(-\mathrm{j}2\pi/n)$, yields a diagonal matrix $\hat{\mathbf{P}}$ since $\mathbf{P}$ is circulant. After a similarity transformation of the system of equations one obtains therefore

$$\left(\mathbf{I}_{2n} + \begin{pmatrix} a_1\,\hat{\mathbf{P}} & -\hat{\mathbf{P}} \\ a_2\,\hat{\mathbf{P}} & 0 \end{pmatrix}\right)\,\hat{\mathbf{Z}}[\ell] - \begin{pmatrix} a_1\,\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} & -\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} \\ a_2\,\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} & 0 \end{pmatrix}\,\hat{\mathbf{Z}}[\ell] = \hat{\tilde{b}} \tag{20}$$

where $\hat{\tilde{b}} = \mathbf{F}\,\tilde{b}$. The 1st matrix on the left-hand-side is a block diagonal matrix with 2×2 blocks. The 2nd matrix remains a sparse rank-2 matrix. Considering (23) and setting

$$\mathbf{A} = \left(\mathbf{I}_{2n} + \begin{pmatrix} a_1\,\hat{\mathbf{P}} & -\hat{\mathbf{P}} \\ a_2\,\hat{\mathbf{P}} & 0 \end{pmatrix}\right), \quad \mathbf{U}\,\mathbf{V}^{\mathrm{H}} = -\begin{pmatrix} a_1\,\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} & -\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} \\ a_2\,\hat{\mathbf{u}}\hat{\mathbf{v}}^{\mathrm{H}} & 0 \end{pmatrix} \tag{21}$$

where

$$\mathbf{U} = \begin{pmatrix} \hat{\mathbf{u}} & 0 \\ 0 & \hat{\mathbf{u}} \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} -a_1\,\hat{\mathbf{v}} & -a_2\,\hat{\mathbf{v}} \\ \hat{\mathbf{v}} & 0 \end{pmatrix} \tag{22}$$

the system of equations can be solved with linear effort (i.e., $O(n)$) employing the Sherman-Morrison-Woodbury formula (23) as shown in the appendix. The terms $\mathbf{A}^{-1}\,\mathbf{U}$, $\mathbf{V}^{\mathrm{H}}\,\mathbf{A}^{-1}$ and the matrix $\mu \in \mathbb{R}^{2\times 2}$ of (23) can be preprocessed, both are of rank 2. Therefore the rank-2 update can be calculated in $O(n)$ operations. The two filter parameters a_1, a_2 show up directly in the matrices (21) , their replacement in case of filter adaptivity is therefore simple.

The main task is therefore the similarity transformation by the FFT for the input and output signals requiring $O(n \log n)$ operations, which is needed for implementing the nominator polynomial (FIR) anyway via overlap-add or -save techniques.

The same procedure may be applied to the 1st canonical form, which is contained as a special case with $\mu \in \mathbb{R}$ in the SMW formula. This case is therefore omitted here for brevity. One remark is necessary: the 2nd order blocks may also realized as a cascade of 1st order blocks with conjugate complex coefficients $a_1,\, a_1^* \in \mathbb{C}$, which is the third variant mentioned in the introduction. In the next section the second variant is presented.

5. Implementation Aspects

The nominator polynomial in (4), $B(z^{-1})$, can be efficiently computed in parallel employing either the FFT based well known overlap-add or overlap-save techniques which is not considered here in more detail. The denominator polynom $A(z^{-1})$ or autoregressive (AR) part of the filter is again rewritten as a cascade of 1st and 2nd order blocks as of (5). We consider therefore only the AR part of a 2nd order filter, namely

$$H(z) = \frac{1}{1 + a_1\,z^{-1} + a_2\,z^{-2}} \tag{23}$$

and partition the signals again in blocks of length $n > M$.

[1] Alternatively, a preprocessing $(\mathbf{I}_{2n} - \mathbf{A}_1)^{-1}\,\mathbf{A}_2$ and $(\mathbf{I}_{2n} - \mathbf{A}_1)^{-1}\,\mathbf{B}$ may be performed.

Below we employ the $n \times n$ matrices

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ a_1 & 1 & 0 & & \vdots \\ a_2 & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & 1 & 0 \\ & & a_2 & a_1 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 & 0 & & a_2 & a_1 \\ & 0 & 0 & & a_2 \\ & & \ddots & \ddots & \\ & & & 0 & 0 \\ & & & & 0 \end{pmatrix} \tag{24}$$

where $\mathbf{A}$ is of full rank and $\mathbf{B}$ of rank two. The matrices can be rewritten in compact form as

$$\begin{aligned} \mathbf{A} &= \mathbf{I}_n + a_1\,\mathbf{N} + a_2\,\mathbf{N}^2, \\ \mathbf{B} &= a_1\,\mathbf{e}_0\,\mathbf{e}_{n-1}^{\mathrm{T}} + a_2\,\mathbf{e}_0\,\mathbf{e}_{n-2}^{\mathrm{T}} + a_2\,\mathbf{e}_1\,\mathbf{e}_{n-1}^{\mathrm{T}} \end{aligned} \tag{25}$$

using (2). The system of equations for block filtering reads

$$\mathbf{A}\,\mathbf{y}_\ell + \mathbf{B}\,\mathbf{y}_{\ell-1} = \mathbf{x}_\ell. \tag{26}$$

From (3) one obtains after a short calculation

$$\mathbf{A} = \mathbf{I}_n + a_1\,(\mathbf{P} - \mathbf{e}_0\,\mathbf{e}_{n-1}^{\mathrm{T}}) + a_2\,(\mathbf{P}^2 - \mathbf{e}_1\,\mathbf{e}_{n-1}^{\mathrm{T}} - \mathbf{e}_0\,\mathbf{e}_{n-2}^{\mathrm{T}}). \tag{27}$$

Let $\mathbf{A}_1 = \mathbf{I}_n + a_1\,\mathbf{P} + a_2\,\mathbf{P}^2$ and note that $\mathbf{B} = \mathbf{A}_1 - \mathbf{A}$ is of rank 2. Rewriting (25) yields

$$\mathbf{A}_1\,\mathbf{y}_\ell - \mathbf{B}\,\mathbf{y}_\ell = \mathbf{x}_\ell - \mathbf{B}\,\mathbf{y}_{\ell-1} = \tilde{b}. \tag{28}$$

The matrix $\mathbf{A}_1$ is a circulant matrix and can be diagonalized by the FFT, namely

$$\hat{\mathbf{A}}_1 = \mathbf{F}\,\mathbf{A}_1\,\mathbf{F}^{\mathrm{H}} = \mathbf{I}_n + a_1\,\hat{\mathbf{P}} + a_2\,\hat{\mathbf{P}}^2 \tag{29}$$

where the diagonal matrix $\hat{\mathbf{P}}$ is given by (19). Moreover $\mathbf{B} = \mathbf{U}\,\mathbf{V}^{\mathrm{H}}$ can be decomposed by $n \times 2$ matrices $\mathbf{U}$, $\mathbf{V}$, i.e.

$$\mathbf{U} = \begin{pmatrix} \mathbf{e}_0, & \mathbf{e}_1 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} a_1\,\mathbf{e}_{n-1} + a_2\,\mathbf{e}_{n-2}, & a_2\,\mathbf{e}_{n-1} \end{pmatrix}. \tag{30}$$

The matrix $\hat{\mathbf{B}} = \mathbf{F}\,\mathbf{B}\,\mathbf{F}^{\mathrm{H}} = \hat{\mathbf{U}}\,\hat{\mathbf{V}}^{\mathrm{H}}$ is a rank-2 matrix, where $\hat{\mathbf{U}} = \mathbf{F}\,\mathbf{U}$, $\hat{\mathbf{V}} = \mathbf{F}\,\mathbf{V}$. After similarity transformation (28) is rewritten by

$$\hat{\mathbf{A}}_1\,\hat{\mathbf{y}}_\ell - \hat{\mathbf{B}}\,\hat{\mathbf{y}}_\ell = \hat{\mathbf{x}}_\ell - \hat{\mathbf{B}}\,\hat{\mathbf{y}}_{\ell-1} = \hat{b}. \tag{31}$$

This system of equations can be solved in parallel in an effort of $O(n)$ employing the Sherman-Morrison-Woodbury formula (23), since $\hat{\mathbf{A}}_1$ is a diagonal matrix and $\hat{\mathbf{B}}$ of rank 2.

6. Conclusion

This paper presents a technique for the massive parallelization of IIR filters employing a similarity transform realized by the FFT in conjunction with the Sherman-Morrison-Woodbury formula, requiring $O(n \log n)$ operations. Unlike existing FFT based techniques, only a small amount of preprocessing is required. This enables an easy update of the filter coeffcients. Since no zero padding is required, as for the overlap-save and overlap-add techniques, this algorithm outperforms existing FFT based methods. Asymptotically one gains a factor of two. The technique has been investigated in a bit true MATLAB library, using a canonical signed digit representation. Due to its simplicity, the method is well-suited for massive parallelization such as for GPUs.

It is well-known, that the FIR or non-recursive part can be parallelized also by the Discrete Cosine Transformation (DCT), saving some calculations. In future work, the use of the DCT for the recursive part employing the Sherman-Morrison-Woodbury formula shall be investigated.

Acknowledgments

This project AMOR ATCZ203 has been co-financed by the European Union using financial means of the European Regional Development Fund (INTERREG) for sustainable cross boarder cooperation. Further information on INTERREG Austria-Czech Republic is available at https://www.at-cz.eu/at.

Data Availability

The datasets generated during and/or analyzed during the current study and MATLAB source files are available from the corresponding author on request.

References

[1] BRACHTENDORF, H. G., DALPIAZ, C., STEIGER, M. Multiplier-less broadband and linear phase digital Hilbert transformers. *Circuits, Systems, and Signal Processing*, 2024, vol. 43, no. 10, p. 6475–6486. DOI: 10.1007/s00034-024-02682-6

[2] ZEINTL, C., BRACHTENDORF, H. G. Linear phase design of lattice wave digital filters. In *International Conference Radioelektronika (RADIOELEKTRONIKA)*. Prague (Czech Republic), 2018, p. 1–5. DOI: 10.1109/RADIOELEK.2018.8376376

[3] JOHANSSON, H., WANHAMMAR, L. Wave digital filter structures for high-speed narrow-band and wide-band filtering. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1999, vol. 46, no. 6, p. 726–741. DOI: 10.1109/82.769781

[4] YLI-KAAKINEN, J., SARAMAKI, T. Design of very low-sensitivity and low-noise recursive filters using a cascade of low order lattice wave digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1999, vol. 46, no. 7, p. 906–914. DOI: 10.1109/82.775386

[5] ZELMAT, M., LAMINI, E.-S., TAGZOUT, S., et al. Precision analysis for an optimal parallel IIR filter's implementation. *Circuits, Systems, and Signal Processing*, 2022, vol. 41, p. 1–35. DOI: 10.1007/s00034-022-01988-7

[6] HOSEMANN, M., FETTWEIS, G. On enhancing SIMD-controlled DSPs for performing recursive filtering. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 2006, vol. 44, no. 1–2, p. 313–322. DOI: 10.1007/s11265-006-9714-4

[7] KUTIL, R. Parallelization of IIR filters using SIMD extensions. In *International Workshop on Systems, Signal Processing and Their Applications (IWSSIP)*. Bratislava (Slovakia), 2008, p. 65–68. DOI: 10.1109/IWSSIP.2008.4604368

[8] BURRUS, C. Block realization of digital filters. *IEEE Transactions on Audio and Electroacoustics*, 1972, vol. 20, no. 4, p. 230–235. DOI: 10.1109/TAU.1972.1162387

[9] GOLD, B., JORDAN, K. A note on digital filter synthesis. *Proceedings of the IEEE*, 1968, vol. 56, no. 10, p. 1717–1718. DOI: 10.1109/PROC.1968.6709

[10] GNANASEKARAN, R., MITRA, S. A note on block implementation of IIR digital filters. *Proceedings of the IEEE*, 1977, vol. 65, no. 7, p. 1063–1064. DOI: 10.1109/PROC.1977.10614

[11] PARHI, K., MESSERSCHMITT, D. Pipeline interleaving and parallelism in recursive digital filters. I. Pipelining using scattered look-ahead and decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1989, vol. 37, no. 7, p. 1099–1117. DOI: 10.1109/29.32286

[12] BRENT, R., BING, Z. A stabilized parallel algorithm for direct-form recursive filters. *IEEE Transactions on Computers*, 1991, vol. 40, no. 3, p. 333–336. DOI: 10.1109/12.76411

[13] MOYER, A. An efficient parallel algorithm for digital IIR filters. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Philadelphia (USA), 1976, p. 525–528. DOI: 10.1109/ICASSP.1976.1170109

[14] VETTERLI, M. Running FIR and IIR filtering using multirate filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1988, vol. 36, no. 5, p. 730–738. DOI: 10.1109/29.1582

[15] BAUER, P. H., LECLERC, L.-J. A computer-aided test for the absence of limit cycles in fixed-point digital filters. *IEEE Transactions on Signal Processing*, 1991, vol. 39, no. 11, p. 2400–2410. DOI: 10.1109/78.97995

[16] HEWLITT, R., SWARTZLANTLER, E. Canonical signed digit representation for FIR digital filters. In *IEEE Workshop on Signal Processing Systems (SiPS)*. Lafayette (USA), 2000, p. 416–426. DOI: 10.1109/SIPS.2000.886740

About the Author . . .

Hans Georg BRACHTENDORF graduated in Electrical Engineering from RWTH Aachen, Germany in 1989 and obtained the Ph.D. degree from the University of Bremen at the Institute for Electromagnetic Theory and Microelectronics in 1994, also in Electrical Engineering, respectively. From 1994–2001 he was an Assistant Professor (C1) also at the University of Bremen and obtained the Venia Legendi (Habilitation) from the same university in 2001. In 1997–1998 he was affiliated with the Wireless Laboratory of Bell Laboratories/Lucent Technologies in Murray Hill/New Jersey. In 2001 he joined the Fraunhofer Institute for Integrated Circuits in Erlangen, Germany. Since 2005 he is full professor at the University of Applied Science of Upper Austria for System Design and Simulation, Communications and Signal Processing.

Appendix A: Sherman-Morrison-Woodbury Formula

Let $\mathbf{B} = \mathbf{A} + \mathbf{U}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{A}$ is an invertible $n \times n$ matrix, $\mathbf{U}$, $\mathbf{V}$ are $n \times k$ matrices. Then

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\,\mathbf{U}\,\mu\,\mathbf{V}^{\mathrm{H}}\,\mathbf{A}^{-1}, \quad \mu = (\mathbf{I}_k + \mathbf{V}^{\mathrm{H}}\,\mathbf{A}^{-1}\,\mathbf{U})^{-1} \tag{A1}$$

where the superscript $^{\mathrm{H}}$ represents the Hermitian operator. If the $k \times k$ matrix μ is regular, the inverse $\mathbf{B}^{-1}$ exists.