# Fast Algorithm for Optimal Design of Block Digital Filters Based on Circulant Matrices

Ali Daher, El Houssaïn Baghious, Gilles Burel

*IEEE Signal Processing Letters, pp. 637-640, Vol. 15, 2008*

This is the accepted version generated by the authors - The published version is available at https://doi.org/10.1109/LSP.2008.2003988

# Fast Algorithm for Optimal Design of Block Digital Filters Based on Circulant Matrices

Ali Daher, El Houssain Baghious, and Gilles Burel, *Member, IEEE*

***Abstract*—Block digital filtering has been suggested to reduce the computational complexity and to increase the parallelism of computation in digital filtering systems. In this letter, a fast algorithm for optimal design of Block Digital Filters (BDFs) is developed. This algorithm, based on circulant, Toeplitz and shift cyclic matrices, does not only reduce the computational complexity of the design process but also decreases the memory requirements.**

***Index Terms*—Block digital filters, circulant matrix, Toeplitz matrix, time-varying systems, aliasing, overlap-save.**

## I. Introduction

In many signal processing applications, fast digital filtering is required. Transform-based block digital filtering is well known for two benefits : firstly, its ability to reduce the computational complexity, and secondly, the possibility to increase the parallelism of computation in digital filtering systems, which is so much required now with the availability of multiprocessor architectures. Many approaches to Block Digital Filters (BDFs) design exist. Some approaches compel the BDF to be time-invariant so that conventional filter synthesis techniques can be used [6]. The best known and most widely used approach is Overlap-save ([10, p. 558]). In some other approaches, no such constraint on the BDF is imposed so that the BDF can be time-variant [7]. For that, an optimal matrix-oriented approach was developed by G. Burel in [2]. Although the aliasing distortion is not null, the global distortion obtained with this optimal approach is lower than the global distortion obtained with overlap-save or other approaches.

On the other hand, many fast algorithms in the context of digital filtering have been obtained based on particular matrix structures [9], [11].

In this letter, we are explicitly establishing the relationship of the circulant, Toeplitz and cyclic shift matrices with the design of the optimal BDF. Within this process, a very fast algorithm is obtained. The proposed algorithm gives the same BDF as that obtained in [2] but with lower computational cost and less memory requirement.

This letter is organized as follows. In Section II, the principle of transform-based block digital filters is shown. In Section III, the existing algorithm for the optimal BDF design is presented. In Section IV, a description of our proposed algorithm is provided. In Section V, computational complexity and memory requirements are shown. Finally, a conclusion is drawn in Section VI.

The authors are with the Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance (Lab-STICC), Université de Bretagne Occidentale, BREST cedex 3, FRANCE.

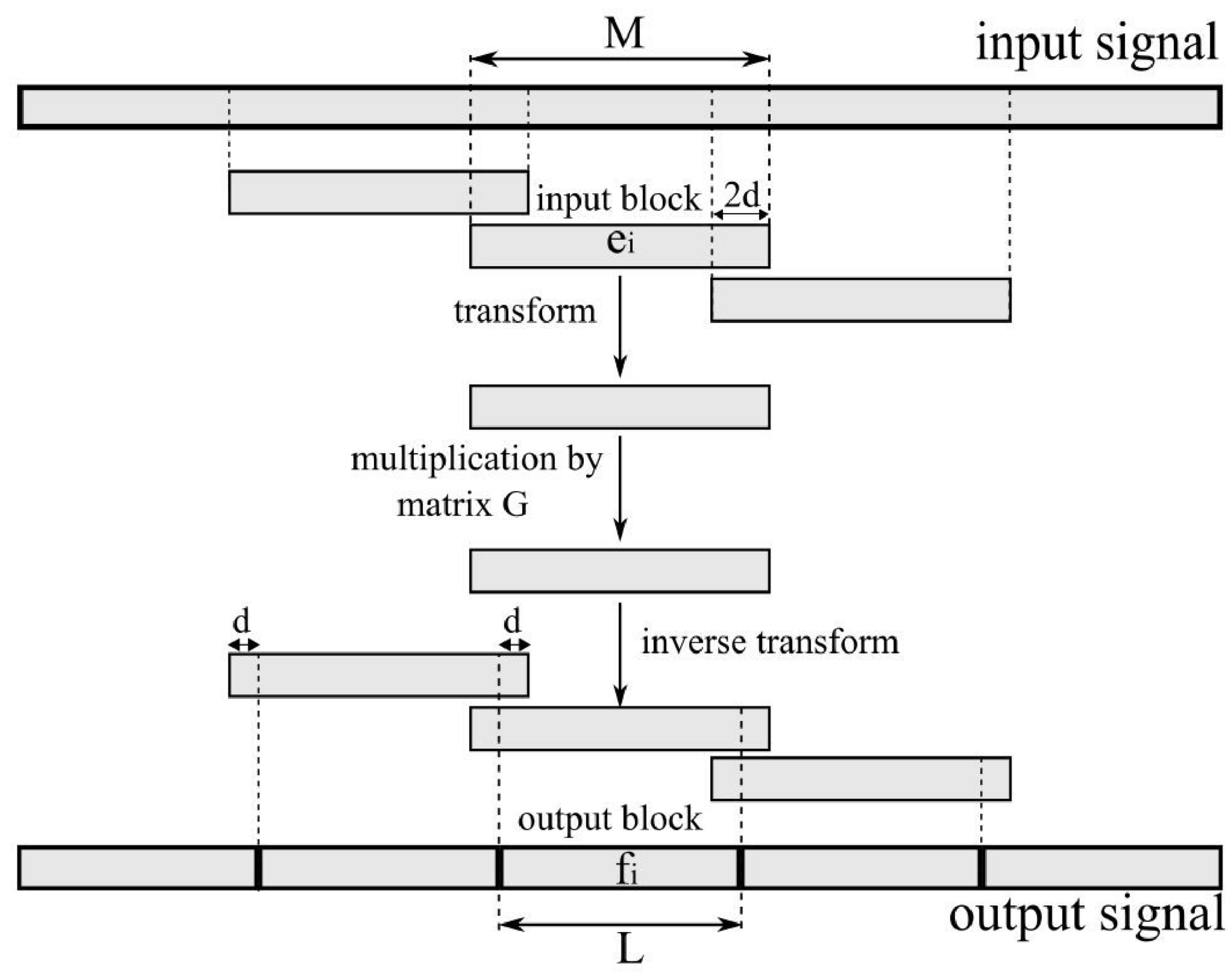


Fig. 1. Transform-based block digital filtering

## II. Transform-based Block digital filter

The principle of the transform-based block digital filtering is illustrated on Fig. 1. The input signal is divided into overlapping blocks of $M$ samples. Each block is then processed by transform filtering and provides $L$ samples of the output signal ($L \leq M$ and $2d = M - L$ is even to preserve symmetry).

In this letter, we consider that the input block size $M$ and the output block size $L$ are chosen based on hardware and software considerations. The transform used is the Discrete Fourier Transform (DFT).

Throughout the letter, we will note :

- $F_N$ : the $N \times N$ matrix corresponding to an $N-$point DFT and $F_N^{-1}$ as its inverse;
- $S$ : the $L \times M$ selection matrix, a binary matrix which selects $L$ values out of $M$ from the middle;
- $P_M$ : the $M \times M$ permutation matrix of the cyclic right shift operator;
- $A\left(:, j\right)$ : an $m \times 1$ vector composed by the elements of the $j^{th}$ column of an $m \times n$ matrix $A$;
- $A\left(i, :\right)$ : an $n \times 1$ vector composed by the elements of the $i^{th}$ row of an $m \times n$ matrix $A$;
- $\left\langle . \right\rangle_q$ : the modulo $q$ operation.

For an input block $e_i$, the output block $f_i$ is given by :

$$f_i = Ae_i \tag{1}$$

where the $L \times M$ matrix $A$ is decomposed as follows :

$$A = SF_M^{-1}GF_M \tag{2}$$

$G$ is a matrix of size $M \times M$. The elements of $G$ are chosen in order to obtain a frequency response close to the desired one. Often, and in our case, $G$ is considered as a diagonal matrix. $S$ is the $L \times M$ selection matrix given by :

$$S = [0_{L\times d}\, I_L\, 0_{L\times d}] \tag{3}$$

with $0_{L\times d}$ is the $L \times d$ zero matrix and $I_L$ is the $L \times L$ identity matrix.

The filtered signal is formed by the concatenation of the output blocks $f_i$.

## III. Previous work

Four methods for the optimal BDF design have been proposed by G. Burel in [2]. Each method has a domain of validity which depends on the kind of criterion and on the kind of transform. They provide the optimal BDF (matrix $G$) but they differ by computational complexity and memory requirements. The obtained BDF is periodically time-variant and there is an aliasing error but the global distortion is lower than the global distortion obtained with overlap-save or other approaches. These methods were derived from an approach based only on elementary matrix computation, hence it is easy to implement them with modern mathematical tools. The fourth method, which is the fastest, and which is applied in case of unweighed frequency (where no frequency is privileged) and unitary transform (DFT in our case), will be discussed. The criterion defined for the optimal BDF design is to minimize the quadratic error given by the following Frobenius norm :

$$e = \left\|SF_M^{-1}GF_M - A_d\right\|^2 \tag{4}$$

As given in [1], the elements $a_d(l,m)$ $(l : 0 \to L-1,\, m : 0 \to M-1)$ of the $L \times M$ matrix $A_d$ are extracted from the $K \times K$ matrix $H_d$ given by :

$$H_d = F_K^{-1}\overline{\overline{H}}_d F_K \tag{5}$$

$K$ is the desired frequency resolution ($K > M$, $K = bL$ with b an integer). $\overline{\overline{H}}_d$ is the $K \times K$ diagonal matrix with $\overline{\overline{H}}_d(k,k) = f(k)$ $(k = 0, ..., K-1)$ where $f(k)$ is the desired frequency response.

To reduce the complexity of calculation of the matrix $A_d$, G. Burel proposed in [2] to extract $a_d(l,m)$ from the elements $p_d(l,k)$ of an $L \times K$ matrix $P_d$ as below :

$$a_d(l,m) = p_d\left(l, \langle l+d-m\rangle_K\right) \tag{6}$$

The matrix $P_d$ is given by :

$$P_d = LF_L^{-1}\overline{\overline{P}}_d F_K^{-1} \tag{7}$$

where $\overline{\overline{P}}_d$ is the $L \times K$ matrix whose first row elements are the $f(k)$ and the other elements are null.

For the case of $G$ diagonal, its diagonal elements $g_n$ are given by :

$$g_n = \frac{(B(:,n))^H E(:,n)}{\|B(:,n)\|^2}\,; \quad n : 1 \to M \tag{8}$$

$(.)^H$ stands for the Hermitian transpose. $B$ and $E$ are the $L \times M$ matrices given below :

$$B = SF_M^{-1} \tag{9}$$

$$E = A_d F_M^{-1} \tag{10}$$

Evaluating the computational cost (the number of real multiplications) of this method, we have to :

- Use (7) to compute $P_d$ : this requires a 2D-FFT of matrix $\overline{\overline{P}}_d$. Its cost is approximately $LK\left(log_2(LK)\right)$ real multiplications. We remind that, with complex data, the required number of real multiplications for a $K$-point FFT is $K\left(log_2K-3\right)+4$ (see [8, p. 60]), that is, approximately, $Klog_2K$.
- Extract $A_d$ from $P_d$ according to (6) : this does not require multiplications.
- Compute $B$ given by (9) : it is only a selection of $L$ rows of the inverse transform matrix. It does not require multiplications.
- Compute $E$ given by (10) : it is an inverse transform of the rows of $A_d$. It requires $LMlog_2M$ real multiplications.
- Compute $g_n$ in (8) : it requires $2ML$ complex multiplications. As a complex multiplication is equivalent to three real multiplications (Golub's method), therefore computing $g_n$ requires $6ML$ real multiplications.

## IV. Proposed algorithm

In the previous algorithm, the calculation has been made without considering the particular structures of the different matrices and the tools that can reduce the computation complexity. To optimize the process, we propose a faster algorithm by using the properties of Toeplitz, circulant and shift cyclic matrices.

### A. Toeplitz, Circulant and cyclic shift matrices

An $M \times M$ matrix $T$ is known as Toeplitz matrix if it has constant values on each diagonal, that is, along the lines of entries parallel to the main diagonal :

$$t(i,j) = t_{j-i}\,; \quad i,j : 0 \to M-1 \tag{11}$$

The circulant matrix $C$ is a particular case of the Toeplitz structure with $t_{j-i} = t_{\langle j-i\rangle_M}$:

$$c(i,j) = c\left(\langle i+1\rangle_M, \langle j+1\rangle_M\right) = c_{\langle j-i\rangle_M}\,; \; i,j : 0 \to M-1 \tag{12}$$

The $M \times M$ circulant matrix $C$ is then formed from an $M$-vector by cyclically permuting the entries :

$$C = \begin{bmatrix} c_0 & c_1 & \cdots & \cdots & c_{M-1} \\ c_{M-1} & c_0 & c_1 & \cdots & c_{M-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ c_1 & \cdots & \cdots & c_{M-1} & c_0 \end{bmatrix} \tag{13}$$

In vector-matrix form, the relation between the rows of $C$ can be described by :

$$C(n,:) = P_M C(n-1,:) \tag{14}$$

$$C(n,:) = (P_M)^{n-1} C(1,:) \tag{15}$$

$P_M$ is the $M \times M$ permutation matrix of the cyclic right shift operator given by :

$$P_M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{16}$$

$P_M$ is a particular circulant matrix with the attributes :

$$(P_M)^M = I_M \tag{17}$$

$$P_M P_M^t = I_M \tag{18}$$

The circulant matrices have many mathematical properties used in communications and information theory [4], [5]. The most interesting property is matrix diagonalisation. A circulant matrix $C$ can be decomposed as :

$$C = F_M^{-1} D F_M \tag{19}$$

with $D$ an $M \times M$ diagonal matrix, the diagonal of which is obtained by :

$$diag(D) = F_M C(:,1) \tag{20}$$

$$diag(D) = M F_M^{-1} C(1,:) \tag{21}$$

### B. Description of the algorithm

As described in [2], we have to minimize the error given in (4). Because $G$ is diagonal, by using (19), this is equivalent to minimize :

$$e = \|SC - A_d\|^2 \tag{22}$$

where $C$ is a circulant matrix.

The first step of our algorithm is to compute the matrix $A_d$ whose elements are extracted from the matrix $H_d$ as given in [1]. Because $\overline{\overline{H}}_d$ is a diagonal matrix, the matrix $H_d$ given by (5) is a circulant matrix. The elements of the first column of the matrix $H_d$ are derived from (20) as given in equations (23) and (24). We note that $A_d$, a block $L \times M$ of the $K \times K$ circulant matrix $H_d$, will have a rectangular Toeplitz structure $(A_d(i,j) = A_d(i+1,j+1)\,; \quad i: 0 \to L-2, j: 0 \to M-2)$.

$$H_d(:,1) = F_K^{-1} diag\left(\overline{\overline{H}}_d\right) \tag{23}$$

$$H_d(:,1) = F_K^{-1} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(K-1) \end{bmatrix} \tag{24}$$

The second step is to compute the matrix $C$ that minimizes the error $e$ given in (22). Noting that the Frobenius norm of a matrix is equal to the Frobenius norm of the vector formed by the concatenation of its rows, this gives :

$$e = \|vec(SC) - vec(A_d)\|^2 \tag{25}$$

$vec(U)$ is the vector formed by concatenating the rows of matrix $U$. Due to the structure of matrix $S$, $vec(SC)$ is then the vector obtained by concatenating rows ranging from $d+1$ to $M-d$ of matrix $C$. Using (15), (25) can be written as :

$$e = \left\| \begin{bmatrix} (P_M)^d \\ (P_M)^{d+1} \\ \vdots \\ (P_M)^{M-d-1} \end{bmatrix} C(1,:) - \begin{bmatrix} A_d(1,:) \\ A_d(2,:) \\ \vdots \\ A_d(L,:) \end{bmatrix} \right\|^2 \tag{26}$$

By using the pseudo-inverse and the $P_M$-matrix properties given in (17) and (18), the vector $C(1,:)$ minimizing (26) is given by :

$$C(1,:) = \frac{1}{L} \sum_{i=1}^{L} \left( \left(P_M^{-1}\right)^{i+d-1} A_d(i,:) \right) \tag{27}$$

Note that $P_M^{-1}$ is the permutation matrix of the cyclic left shift operator, then a matrix-vector product $\left(P_M^{-1}\right)^n X$ is simply a cyclic $n-$ elements left shift of the $M-$vector $X$ and does not require any operation of multiplication or addition.

The final step is to compute the diagonal elements of the matrix $G$. By referring to (21), they are obtained by :

$$diag(G) = M F_M^{-1} C(1,:) \tag{28}$$

Let us evaluate the computational cost of our algorithm. We have to :

- Compute the elements of the matrix $H_d$ according to (24) : this requires an inverse $K-$point FFT of the desired frequency response $f(k)$. Its cost is approximately $K(log_2 K)$ real multiplications.
- Extract the matrix $A_d$ from the matrix $H_d$ as described in [1] : this does not require any multiplication.
- Compute $C(1,:)$ as given in (27) : this does not require any multiplication.
- Compute $diag(G)$ as given in (28) : this requires an $M-$point FFT . Its cost is approximately $Mlog_2 M$ real multiplications.

## V. Results

To approximate the computational complexity of the algorithms, we have considered that the computational cost is the required number of real multiplications. Then, the cost of the existing algorithm developed in [2] is about

TABLE I
COMPUTATIONAL COMPLEXITY

| $(M, L, K)$ | **algorithm 1** | **algorithm 2** |
|---|---|---|
| $(32, 24, 96)$ | $3.42 \times 10^4$ | $7.92 \times 10^2$ |
| $(256, 200, 1024)$ | $4.33 \times 10^6$ | $1.23 \times 10^4$ |
| $(2048, 1024, 8192)$ | $2.29 \times 10^8$ | $1.29 \times 10^5$ |

TABLE II
MEMORY REQUIREMENTS

| $(M, L, K)$ | **algorithm 1** | **algorithm 2** |
|---|---|---|
| $(32, 24, 96)$ | 12 KB | 1.2 KB |
| $(256, 200, 1024)$ | 819 KB | 11.8 KB |
| $(2048, 1024, 8192)$ | 33 MB | 90.1 KB |

$LK(log_2 KL) + LMlog_2 M + 6ML$ while it is only about $K(log_2 K) + M(log_2 M)$ for our proposed algorithm. In addition, due to the structure of circulant and Toeplitz matrices, our algorithm requires less memory. An $M \times M$ circulant matrix can be represented by its first column; therefore, we require memory to store only $M$ elements. An $L \times M$ matrix with Toeplitz structure can be represented by its first column and its first row; therefore, we require memory to store only $M + L - 1$ elements.

In order to illustrate that, we give in Table I and Table II the computational complexity (number of real multiplications) and the memory requirements (in Kilobyte or Megabyte) for different typical values of $M$, $L$ and $K$. The existing algorithm is noted as algorithm 1 and our proposed algorithm is noted as algorithm 2.

The proposed algorithm provides a faster method to design the optimal BDF. We obtain the same BDF developed in [2] but we can see from Table I and Table II that, for example, for $M = 2048$, $L = 1024$ and $K = 8192$, the proposed algorithm requires approximately $1775$ times fewer multiplications and $366$ times less space memory than the existing algorithm. There are many practical applications in which such large values of block sizes, $M$ and $L$, and frequency resolution $K$ are required.

## VI. CONCLUSION

Design of fast computationally efficient algorithms has been a major focus of research activity in digital signal processing. Transform-based block digital filters (BDFs) are well known for their ability to reduce the computational complexity and to increase the parallelism of computation in digital filters. In this letter, we have shown an existing matrix-oriented approach to design an optimal BDF. To improve that, we have proposed a new faster algorithm to design the same BDF by using the properties of circulant matrices. The proposed algorithm gives the same BDF coefficients, and therefore, the same global distortion, as the existing algorithm but it has a lower computational cost and less memory is required. This allows a simple implementation of the optimal BDF design in many filtering applications.

## REFERENCES

[1] G. Burel, “A matrix-oriented approach for analysis and optimisation of block digital filters,” in *WSEAS Int. Conf. on Signal, Speech and Image Processing (ICOSSIP 2002)*, September 2002.

[2] ——, “Optimal design of transform-based block digital filters using a quadratic criterion,” *IEEE Trans. Signal Processing*, vol. 52, pp. 1964–1974, July 2004.

[3] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Second ed. Baltimore, MD: The John Hopkins Univ. Press, 1989.

[4] R. M. Gray, *Toeplitz and Circulant Matrices: A review*. Now Publishers Inc., 2006.

[5] D. Kalman and J. E. White, “Polynomial equations and circulant matrices,” *The American Mathematical Monthly*, vol. 108, pp. 821–840, November 2001.

[6] I. S. Lin and S. K. Mitra, “Overlapped block digital filtering,” *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 586–596, August 1996.

[7] C. M. Loeffler and C. S. Burrus, “Optimal design of periodically time-varying and multirate digital filters,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 991–997, October 1984.

[8] H. S. Malvar, *Signal Processing With Lapped Transforms*. Reading, MA: Artech House, 1992.

[9] Z. Mou and P. Duhamel, “Short-length fir filters and their use in fast nonrecursive filtering,” *IEEE Trans. Signal processing*, vol. 39, pp. 1322–1332, June 1991.

[10] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[11] M. Teixeira and D. Rodriguez, “A class of fast cyclic convolution algorithms based on block pseudocirculants,” *IEEE Signal Processing Letters*, vol. 2, pp. 92–94, May 1995.