J. David Osés del Campo, Fernando Cruz–Roldán,
Manuel Blanco–Velasco, and Sergio L. Netto

# A Single Matrix Representation for General Digital Filter Structures

A matrix representation of a general single-input single-output (SISO) digital filter structure is addressed, proposing a single matrix that stores the complete description of the filter in a very compact and functional format. The proposed matrix contains the structural information corresponding to the block diagram (BD) connections and, at the same time, it can be seen as a valid computational algorithm to implement the filter in the time domain. With this matrix, the gap between the signal flow graph (SFG) and a bit-true implementation of the filter can be considerably reduced. Finally, simple methods to derive the matrix representation from/onto the block diagram and the state-space (SS) mapping are also described and illustrated with some examples.

## RELEVANCE

New digital filter structures are continuously proposed, presenting strategies to improve the behavior of the systems under, for instance, finite word length conditions [1]–[4]. In this context, and considering the increasing interest in mapping applications onto field-programmable gate arrays and other signal processors, it is very important to have a compact mathematical representation of the system that specifies the exact order in which the computations must be performed, allowing for the simulation of different alternatives and the selection of the best-suited realization for one's particular purposes. To reduce the time required to implement the filter, it is convenient to complement the information of classical BDs or SFGs with a mathematical description, in terms of matrices, which is closer to a valid ready-to-use computational algorithm.

Different representations can be found in the associated literature for describing, in matrix form, the equations corresponding to a general BD with $N$ nodes. In here, a new matricial representation is proposed with the following interesting characteristics: it is very compact; it preserves all information from the original filter structure; it can be readily transformed onto the BD, SFG, or the SS descriptions; and it can be easily mapped on a computable algorithm implementing the desired digital filter.

## PROBLEM STATEMENT

Among the currently known digital filter descriptions, one of the most compact and interesting [5], proposed by Crochiere and Oppenheim, has the general form

$$\mathbf{y}[n] = \mathbf{x}[n] + \mathbf{F}_c^T\mathbf{y}[n] + \mathbf{F}_d^T\mathbf{y}[n-1], \tag{1}$$

where

- $\mathbf{y}[n]$ is an $N \times 1$ column vector of the node signal values at instant $n$
- $\mathbf{x}[n]$ is an $N \times 1$ column vector of the input signal values at instant $n$
- $\mathbf{F}_c^T$ is an $N \times N$ matrix of coefficient branches
- $\mathbf{F}_d^T$ is an $N \times N$ matrix of coefficient-delay branches.

Alternative descriptions can be found in [6] and [7]. In model (1), the constant matrices $\mathbf{F}_c^T$ and $\mathbf{F}_d^T$ do not completely describe the system. To obtain a general model, completely characterized by constant matrices, the input vector $\mathbf{x}[n]$ can be written as $\mathbf{E}x[n]$, where for an $N$–node SISO network, $\mathbf{E}$ is a constant $N \times 1$ column matrix indicating the node where the input $x[n]$ is applied, and a possible scaling factor $k_x$. For simplicity, we shall employ the following notation for the general description based only on three constant matrices:

$$\mathbf{w}[n] = \mathbf{E}x[n] + \mathbf{F}\mathbf{w}[n] + \mathbf{G}\mathbf{w}[n-1], \tag{2}$$

where $\mathbf{F}$ and $\mathbf{G}$ are equivalent to $\mathbf{F}_c^T$ and $\mathbf{F}_d^T$, respectively, in (1). The vector $\mathbf{w}[n]$ contains the node signal values. By convention, and without loss of generality, we assume that the last element of $\mathbf{w}[n]$ corresponds to the filter output $y[n]$.

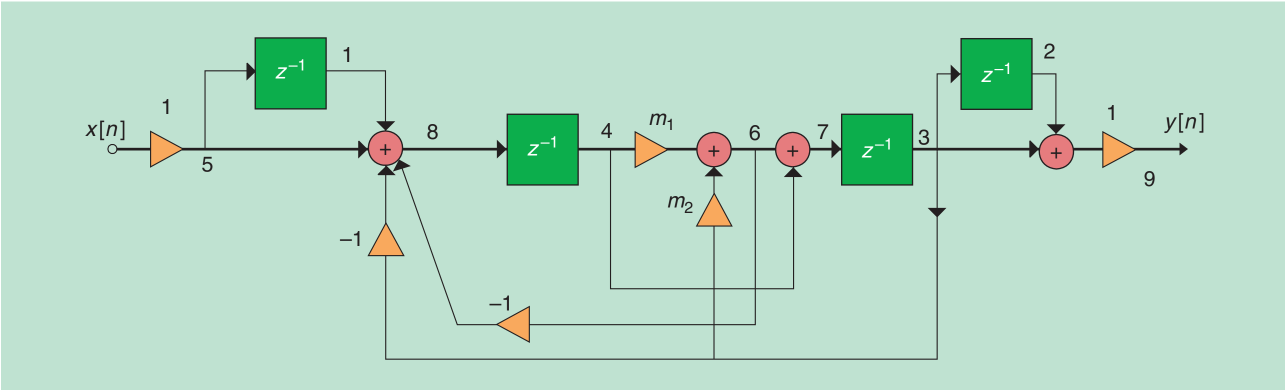
Let us consider as an example the filter of Figure 1 (which corresponds with Figure 2 of [8]), where the nodes in the BD have been previously labeled to obtain a valid and computable set of equations.

We can describe this structure by means of the following system of equations:

$$\mathcal{S}: \begin{cases} w_1[n] = w_5[n-1] \\ w_2[n] = w_3[n-1] \\ w_3[n] = w_7[n-1] \\ w_4[n] = w_8[n-1] \\ w_5[n] = x[n] \\ w_6[n] = w_3[n]\cdot m_2 + w_4[n]\cdot m_1 \\ w_7[n] = w_4[n] + w_6[n] \\ w_8[n] = w_1[n] - w_3[n] + w_5[n] - w_6[n] \\ y[n] = w_2[n] + w_3[n] \end{cases}. \tag{3}$$

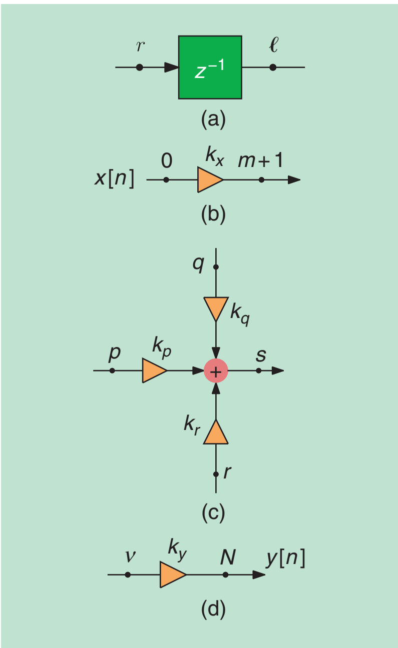
and the corresponding matrix representation, according to model (2), becomes (4) as seen in the box on the next page.

Digital Object Identifier 10.1109/MSP.2011.942837
Date of publication: 1 November 2011




**[FIG1]** **Modified CGIC lowpass filter from [8].**

$$
\begin{bmatrix} w_1[n] \\ w_2[n] \\ w_3[n] \\ w_4[n] \\ w_5[n] \\ w_6[n] \\ w_7[n] \\ w_8[n] \\ y[n] \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} x[n]
+
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & m_2 & m_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} w_1[n] \\ w_2[n] \\ w_3[n] \\ w_4[n] \\ w_5[n] \\ w_6[n] \\ w_7[n] \\ w_8[n] \\ y[n] \end{bmatrix}
+
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} w_1[n-1] \\ w_2[n-1] \\ w_3[n-1] \\ w_4[n-1] \\ w_5[n-1] \\ w_6[n-1] \\ w_7[n-1] \\ w_8[n-1] \\ y[n-1] \end{bmatrix}. \qquad (4)
$$




**[FIG2]** **Graphical representations of (a) delay element corresponding to each row of $\hat{\mathbf{G}}$. (b) Input branch of $\hat{\mathbf{E}}$. (c) Adder element corresponding to several rows of $\hat{\mathbf{F}}$ with the same first element. (d) Output branch corresponding to last row of $\hat{\mathbf{F}}$.**

Taking the $z$-transform of (2), we get

$$\mathbf{W}(z) = \mathbf{E}X(z) + \mathbf{F}\mathbf{W}(z) + \mathbf{G}\mathbf{W}(z)z^{-1}. \qquad (5)$$

Therefore, the response of the different nodes considered in $\mathbf{w}[n]$ to the input $x[n]$ can be expressed, in the $z$-domain, as $\mathbf{W}(z) = \mathbf{T}(z)\mathbf{X}(z)$, with the so-called transfer-function matrix given by

$$\mathbf{T}(z) = \begin{bmatrix} t_1(z) \\ t_2(z) \\ \vdots \\ t_{N-1}(z) \\ t_N(z) \end{bmatrix} = [\mathbf{I} - \mathbf{F} - z^{-1}\mathbf{G}]^{-1}\mathbf{E}, \qquad (6)$$

where $t_i(z)$ denotes the transfer function from the input to the $i$th system node. From the assumption that the last model node corresponds to the system output, the filter transfer function is given by

$$H(z) = t_N(z) = \frac{Y(z)}{X(z)}. \qquad (7)$$

## SOLUTION

Typically, matrices $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ in model (2) are quite sparse. Hence, they can be represented by more compact forms, thus avoiding unnecessary space for zero-valued elements. In addition, we can also take advantage of this sparsity to minimize the amount of computation needed to solve the associated equations.

For example, the `sparse` function of MATLAB generates the following storage organization: (`row, column`) `entry`. Using this notation, matrices $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ corresponding to the filter depicted in Figure 1 can be compactly expressed as shown in Table 1.

Looking at this sparse representation of matrices, it is clear that we can compress and summarize the complete filter model by combining the three constant matrices, with slight changes in the row orders and a zero introduced in $\mathbf{E}$, to form one single partitioned matrix, which we shall call $\mathbf{M}$, as follows:

$$\mathbf{M} = \begin{bmatrix} \hat{\mathbf{G}} \\ \hline \hat{\mathbf{E}} \\ \hline \hat{\mathbf{F}} \end{bmatrix}, \tag{8}$$

where we use the modified notation $\hat{\mathbf{G}}$ and $\hat{\mathbf{F}}$ to indicate that the rows of $\mathbf{G}$ and $\mathbf{F}$ have been sorted in increasing order of their first elements. In addition, $\hat{\mathbf{E}}$ indicates a change to zero of the second entry of $\mathbf{E}$. In short, we have that

- $\hat{\mathbf{G}}$ is an $m \times 3$ matrix, $m$ being the number of delays, of rows of the form $[\ell \; \mathrm{r} \; 1]$, denoting a delay block from node $r$ to node $\ell$, which corresponds to the operation $w_\ell[n] = w_r[n-1]$ and the circuit element depicted in Figure 2(a).
- $\hat{\mathbf{E}}$ is a $1 \times 3$ vector corresponding to the input branch, with the second entry always set to zero to allow us to identify the input node. By convention, and without loss of generality, we may consider that the input comes from node 0 to node $(m+1)$ with a scaling factor of $k_x$, such that $w_{m+1} = x[n] \cdot k_x$, as indicated in Figure 2(b).
- $\hat{\mathbf{F}}$ is a three-column matrix with the information of all adder-multiplier branches properly ordered. Repeated entries $s$ in the first column, with nodes $p$, $q$, and $r$ in the second column and respective gains $k_p$, $k_q$, and $k_r$ for different rows, correspond to the relationship $w_s[n] = w_p[n] \cdot k_p + w_q[n] \cdot k_q + w_r[n] \cdot k_r$, as illustrated in Figure 2(c). If an element in the first column of $\hat{\mathbf{F}}$ appears only once, no addition is performed and a single multiplication gives the value at the corresponding output node. Once again, by convention and without loss of generality, the last row corresponds to the filter output, whose graphical representation is depicted in Figure 2(d).

The first column of $\mathbf{M}$ is an ordered list of natural numbers, where the numbers greater than $(m+1)$ can be repeated. The zero entry in the second column of $\mathbf{M}$ corresponds to the input signal and allows us to identify and separate, in a simple manner, the information corresponding to $\hat{\mathbf{G}}$, $\hat{\mathbf{E}}$, and $\hat{\mathbf{F}}$. The last entry, $N$, of the first column identifies the output node of the filter. In other words, the general matrix $\mathbf{M}$ is in the form

**[TABLE 1] MATLAB COMPACT REPRESENTATION OF SPARSE MATRICES E, F, AND G IN (4).**

| E | | F | | G | |
|---|---|---|---|---|---|
| (5,1) | 1 | (8,1) | 1 | (2,3) | 1 |
| | | (9,2) | 1 | (1,5) | 1 |
| | | (6,3) | $m_2$ | (3,7) | 1 |
| | | (8,3) | −1 | (4,8) | 1 |
| | | (9,3) | 1 | | |
| | | (6,4) | $m_1$ | | |
| | | (7,4) | 1 | | |
| | | (8,5) | 1 | | |
| | | (7,6) | 1 | | |
| | | (8,6) | −1 | | |

$$\mathbf{M} = \left[\begin{array}{ccc} 1 & \cdot & 1 \\ 2 & \cdot & 1 \\ 3 & \cdot & 1 \\ \vdots & \vdots & \vdots \\ m & \cdot & 1 \\ \hline m+1 & 0 & k_x \\ \hline m+2 & \cdot & \cdot \\ m+2 & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ m+s & \cdot & \cdot \\ m+s & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ N & \cdot & k_y \end{array}\right] \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\} \text{Delays} \\ \left.\begin{array}{c} \\ \end{array}\right\} \text{Input} \\ \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} \text{Adders and Multipliers} \\ \left.\begin{array}{c} \\ \end{array}\right\} \text{Output.} \end{array} \tag{9}$$

Using this newly proposed notation, all three matrices $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ (usually large and sparse) in model (2) are replaced by a single and compact matrix $\mathbf{M}$ containing the same information. In our particular example, for the filter depicted in Figure 1, we have

$$\hat{\mathbf{G}} = \begin{bmatrix} 1 & 5 & 1 \\ 2 & 3 & 1 \\ 3 & 7 & 1 \\ 4 & 8 & 1 \end{bmatrix}, \hat{\mathbf{E}} = [5 \quad 0 \quad 1],$$

$$\hat{\mathbf{F}} = \begin{bmatrix} 6 & 3 & m_2 \\ 6 & 4 & m_1 \\ 7 & 4 & 1 \\ 7 & 6 & 1 \\ 8 & 1 & 1 \\ 8 & 3 & -1 \\ 8 & 5 & 1 \\ 8 & 6 & -1 \\ 9 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix}. \tag{10}$$

### NODE ORDERING

Now we describe a simple procedure to obtain the matrix $\mathbf{M}$ from the block diagram, guaranteeing the computability of the resulting system of equations.

Different algorithms can be found for developing a correct node-ordering sequence, leading to a computable system of difference equations. In [5], a formal procedure based on the node precedence relations of the digital network is explained. Another approach, based on a three-step algorithm appears in [7]. Usually, the difference equations characterizing the filter can be put in a computable order by simple inspection, guaranteeing that the signal at a particular node does not depend on the signal of a node whose output is yet to be determined.

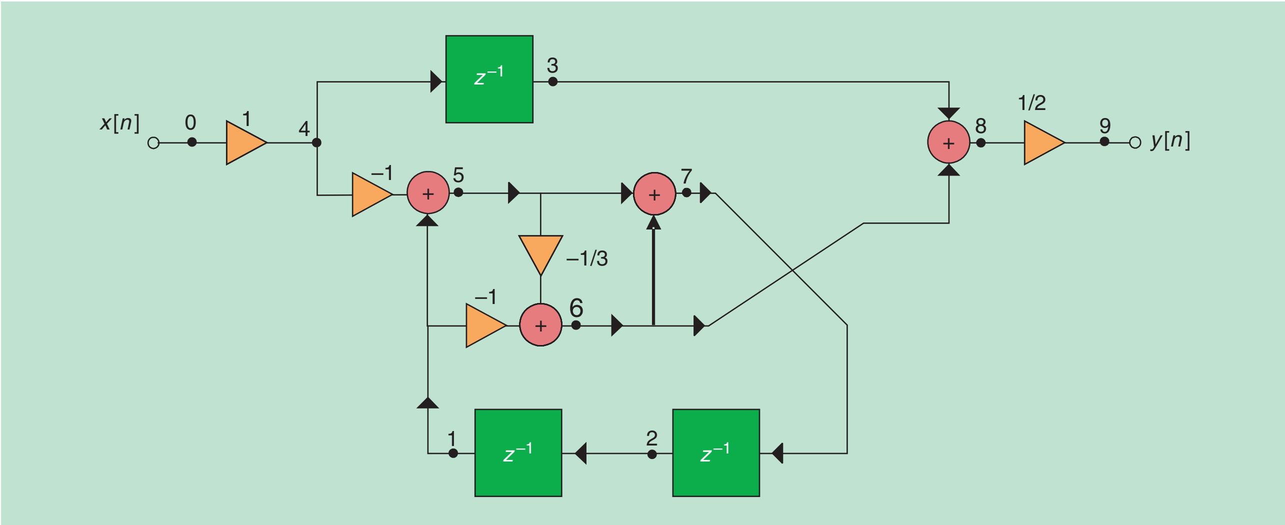
To organize the node-labeling process in a simple and efficient way to obtain matrix $\mathbf{M}$, we propose the following strategy for numbering the $N$ nodes of a general network containing $m$ unit delay blocks:

i) The first nodes $(1, 2, m)$ will be the output of the $m$ delay elements. When several delays are connected in series or the output of one delay block is connected to the input of another one, we shall enumerate the corresponding nodes in increasing order beginning at the outermost output. As examples, see nodes 2 and 1 in Figure 3, or nodes 3 and 2 in Figure 1, respectively.

ii) The input signal, $x[n]$, is connected from node zero to node $(m+1)$.

iii) We shall enumerate the rest of the nodes taking into account the fact that a new node can only be labeled if it depends on the signals previously determined. In general, these nodes will correspond to the output of the adders. The output of the filter, $y[n]$, will be the last node enumerated.

To illustrate the proposed process, we have labeled the nodes of the bireciprocal-lattice wave digital filter [9] presented in Figure 3, where $m = 3$. Initially, and by simple inspection, we can apply steps i) and ii) to number nodes $0, 1, \ldots, (m+1)$. As step iii) indicates, we continue enumerating the output of the adders, starting by the adders whose




[FIG3] **Example of node ordering result for wave digital filter [9].**

inputs have already been labeled, and, finally, the last node is assigned to the output.

Once the nodes have been properly labeled, matrix $\mathbf{M}$ can be determined, as described above, following the established node order. In this case, for the digital filter in Figure 3, we get

$$\mathbf{M} = \left[\begin{array}{ccc|c|ccccccccc} 1 & 2 & 3 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 & 9 \\ 2 & 7 & 4 & 0 & 4 & 1 & 5 & 1 & 5 & 6 & 3 & 6 & 8 \\ 1 & 1 & 1 & 1 & -1 & 1 & -\frac{1}{3} & -1 & 1 & 1 & 1 & 1 & \frac{1}{2} \end{array}\right]^{\mathrm{T}}. \tag{11}$$

Computability for the compact model can be checked by simple inspection of the submatrices $\hat{\mathbf{G}}$ and $\hat{\mathbf{F}}$ of matrix $\mathbf{M}$. Taking into account the fact that delay registers are updated with values calculated in the previous iteration (or initial conditions), the first entry of any row of matrix $\hat{\mathbf{G}}$ must be smaller than the second element of the same row. On the other hand, since a node can only be computed if all the required data are available, the first entry of any row of matrix $\hat{\mathbf{F}}$ must be greater than the second element of the same row.

Matrix $\mathbf{M}$ in (11) corresponds to the sparse model (12) shown in the box at the bottom of the page. In this sparse representation, computability requires matrix $\mathbf{F}$ to be a lower-triangular matrix.

### *REVERSE MODELING AND SET OF EQUATIONS*

Starting with the model matrix $\mathbf{M}$, one can readily obtain the BD or the SFG of the associated digital filter using the graphical representations seen in Figure 2. For that purpose, we must first identify the corresponding $\hat{\mathbf{G}}$, $\hat{\mathbf{E}}$, and $\hat{\mathbf{F}}$ submatrices by locating the null entry in the second column of $\mathbf{M}$. Hence, each row in $\hat{\mathbf{G}}$ represents a delay element, whereas all rows with identical entry in the first column of $\hat{\mathbf{F}}$ define a multiply-and-add element; finally, the input and output branches are respectively characterized by $\hat{\mathbf{E}}$ and the last row of $\hat{\mathbf{F}}$.

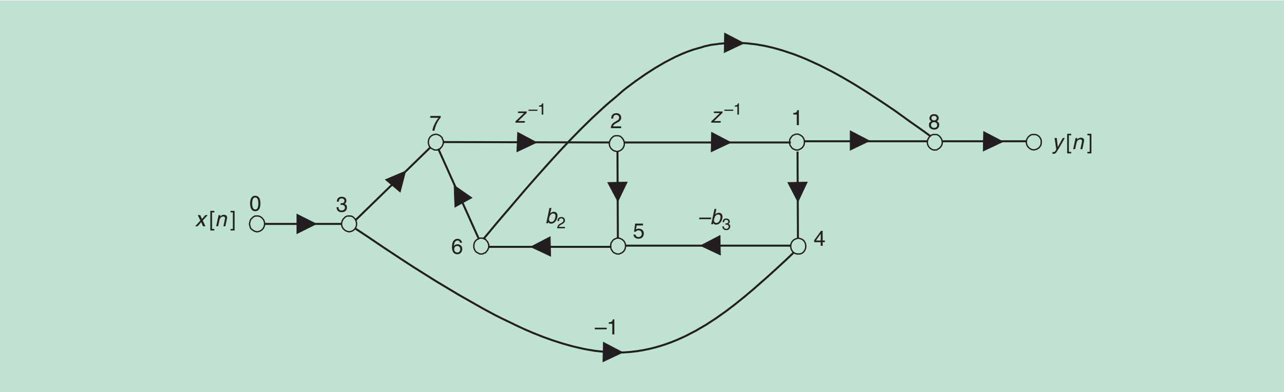
Following this strategy for matrix $\mathbf{M}$ determined by (10), one can readily obtain the digital filter represented in Figure 1. Alternatively, for the compact model

$$\mathbf{M} = \left[\begin{array}{cc|c|ccccccccc} 1 & 2 & 3 & 4 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 \\ 2 & 7 & 0 & 1 & 3 & 2 & 4 & 5 & 3 & 6 & 1 & 6 \\ 1 & 1 & 1 & 1 & 1 & 1 & -b_3 & b_2 & 1 & 1 & 1 & -1 \end{array}\right]^{\mathrm{T}}, \tag{13}$$

the SFG [10] shown in Figure 4 results, where the node numbers have been assigned as specified by the contents of $\mathbf{M}$.

In addition, the proposed model also allows us to obtain, in a simple and direct manner, the corresponding set of difference equations without the necessity of determining any graphical representation for the associated filter. In fact, following the same reasoning as above, the

$$\begin{bmatrix} w_1[n] \\ w_2[n] \\ w_3[n] \\ w_4[n] \\ w_5[n] \\ w_6[n] \\ w_7[n] \\ w_8[n] \\ y[n] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} x[n] + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & \frac{-1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} w_1[n] \\ w_2[n] \\ w_3[n] \\ w_4[n] \\ w_5[n] \\ w_6[n] \\ w_7[n] \\ w_8[n] \\ y[n] \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1[n-1] \\ w_2[n-1] \\ w_3[n-1] \\ w_4[n-1] \\ w_5[n-1] \\ w_6[n-1] \\ w_7[n-1] \\ w_8[n-1] \\ y[n-1] \end{bmatrix}. \tag{12}$$




**[FIG4]** **SFG corresponding to the compact model provided in (13).**

model matrix **M** in (13) can be easily mapped onto the set of equations

$$\mathcal{S}:\begin{cases} w_1[n]=w_2[n-1]\\ w_2[n]=w_7[n-1]\\ w_3[n]=x[n]\\ w_4[n]=w_1[n]+w_3[n]\\ w_5[n]=w_2[n]+w_4[n]\cdot(-b_3)\\ w_6[n]=w_5[n]\cdot b_2\\ w_7[n]=w_3[n]+w_6[n]\\ y[n]=w_1[n]-w_6[n]\end{cases}. \tag{14}$$

In the Appendix, a generalization of the classical `filter` function of MATLAB is presented. In this new function, the information of the filter structure is specified by the user incorporating the matrix M as the first parameter to be introduced. The main advantage of this generalization lies in the flexibility afforded when finite word-length effects are studied; these effects can be easily simulated using the quantize function. Taking into account that the different arithmetical operations to be performed are univocally specified in matrix M, the exact order in which the quantization errors appear, are propagated and finally accumulated in a certain register, can be analyzed. This framework facilitates the comparison of different structures.

### *STATE-SPACE MAPPING*

Now we focus our attention on the relationship between the proposed model and the SS representation [6], [7].

It must be mentioned that the compact matrix **M** retains all information regarding the original system architecture, allowing us to transform this model to/from any alternative representations such as the BD, SFG, set of difference equations, or the sparse matrix model, as discussed above. Meanwhile, the SS description is a distinct compact representation of a given system, including the input-output relationship as well as its internal operation, that does now allow a reverse mapping onto, for instance, the original SFG.

Despite this major difference, the proposed model can be readily mapped onto the SS description through the following algorithm:

i) Associate each of the $m$ delay output nodes, corresponding to the first-column entry in each row of $\hat{\mathbf{G}}$, to a system state $u_j[n+1]$, for $j = 1, 2, \ldots, m$.

ii) Identify the input $x[n]$ and output $y[n]$ nodes from $\hat{\mathbf{E}}$ and the last row of $\hat{\mathbf{F}}$, respectively.

**APPENDIX**

```
function [y,cf]=filt_M(M,x,ci)
% [y,cf]=filt_M(M,x,ci) filters data in vector x with the
% filter described by matrix M.
% INPUT PARAMETERS
% M → Matrix M.
% x → Input data.
% ci → Initial conditions of the delay blocks.
% OUTPUT PARAMETERS
% y → Output data.
% cf → Final state of the delay blocks.
L=max(M(:,1));
w=zeros(L,1);
m=find(M(:,2)==0)−1; % number of delay blocks
w([M(1:m,2)])=ci; % initial conditions
for n=1:length(x)
  for i=1:m
    w(i)=M(i,3)*w([M(i,2)]); % updating delay blocks
  end
  w(m+1)=M(m+1,3)*x(n); % input
  w(m+2:end)=0;
  for i=m+2:size(M,1) % rest of nodes
    r=w(M(i,2))*M(i,3); % multiply
    w(M(i,1))=w(M(i,1))+r; % addition
  end
  y(n)=w(end); % output (last node)
end
cf=w([M(1:m,2)]); % final conditions of delay blocks
```

iii) Solve partially the set $\mathcal{S}$ of difference of equations corresponding to matrix $\mathbf{M}$, writing each state $u_j[n+1]$ as a function of $u_j[n]$, $x[n]$ and $y[n]$. By doing so, one must eliminate dependence to any variable $w_i[n]$ not associated to a system state.

iv) Write the difference equations found in Step iii) in the form

$$\mathcal{U}: \begin{cases} \mathbf{u}[n+1] = \mathbf{A}\mathbf{u}[n] + \mathbf{B}x[n] \\ y[n] = \mathbf{C}\mathbf{u}[n] + \mathbf{D}x[n] \end{cases}, \tag{15}$$

where

- $\mathbf{u}[n] = [u_1[n]\ u_2[n]\ \ldots\ u_m[n]]^{\mathrm{T}}$ is the state vector at time $n$.
- $\mathbf{A}$ is the $m \times m$ system matrix, $\mathbf{B}$ is an $m \times 1$ column vector, $\mathbf{C}$ is a $1 \times m$ row vector, and $\mathbf{D}$ is a scalar.

Following this approach, we can readily obtain the SS description for each of the digital filters depicted in Figure 1:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & m_2 & 1+m_1 \\ 1 & 0 & -m_2-1 & -m_1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\mathbf{C}^{\mathrm{T}} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \mathbf{D} = 0; \tag{16}$$

Figure 3:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ -\frac{2}{3} \\ 1 \end{bmatrix},$$

$$\mathbf{C}^{\mathrm{T}} = \begin{bmatrix} -\frac{2}{3} \\ 0 \\ \frac{1}{2} \end{bmatrix}, \mathbf{D} = \frac{1}{6}; \tag{17}$$

and Figure 4:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -b_2 b_3 & b_2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1-b_2 b_3 \end{bmatrix},$$

$$\mathbf{C}^{\mathrm{T}} = \begin{bmatrix} 1 + b_2 b_3 \\ -b_2 \end{bmatrix}, \mathbf{D} = b_2 b_3. \tag{18}$$

## CONCLUSIONS

A new compact model based on a single matrix $\mathbf{M}$ that describes the input-output relationship, as well as all internal information, of a general digital filter has been presented, and its application widely illustrated. This matrix $\mathbf{M}$ complements the block diagram representation, preventing possible errors during the signal-flow graph analysis and, at the same time, saving all internal data in a very compact manner. Moreover, matrix $\mathbf{M}$ can be interpreted as a simple alternative for the computable set of difference equations that describes the digital-filter evolution in time, shortening considerably the time required to implement the system. Finally, a simple node-ordering sequence oriented to facilitate the construction of matrix $\mathbf{M}$ has also been proposed and relationship to the SS representation was explicited, emphasizing the positive aspects of the proposed model.

## ACKNOWLEDGMENTS

This work was partially supported by the Spanish Ministry of Science and Innovation through project TEC2009-08133, the Ministry of Education through project PHB2008-0017, and by the Brazilian Research Agency CAPES through project 205/09.

## AUTHORS

*J. David Osés del Campo* (doses@ics.upm.es) (doses@ics.upm.es) is an associate professor at the Universidad Politécnica de Madrid, Spain.

*Fernando Cruz-Roldán* (fernando.cruz@uah.es) is a professor at the Universidad de Alcalá (UAH), Spain.

*Manuel Blanco-Velasco* (manuel.blanco@uah.es) is an associate professor at UAH, Spain.

*Sergio L. Netto* (sergioln@lps.ufrj.br) is an associate professor at Federal University of Rio de Janeiro, Brazil.

## REFERENCES

[1] K. Uesaka and M. Kawamata, "Evolutionary synthesis of digital filter structures using genetic programming," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 12, pp. 977–983, Dec. 2003.

[2] T.-B. Deng and Y. Nakagawa, "SVD-based design and new structures for variable fractional-delay digital filters," *IEEE Trans. Signal Processing*, vol. 52, no. 9, pp. 2513–2527, Sept. 2004.

[3] R. Lehto, T. Saramäki, and O. Vainio, "Synthesis of narrowband linear-phase FIR filters with a piecewise-polynomial impulse response," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 10, pp. 2262–2276, Oct. 2007.

[4] G. Li, J. Chu, and J. Wu, "A matrix factorization-based structure for digital filters," *IEEE Trans. Signal Processing*, vol. 55, no. 10, pp. 5108–5112, Oct. 2007.

[5] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, no. 4, pp. 581–595, Apr. 1975.

[6] S. K. Mitra, *Digital Signal Processing. A Computer Based Approach*, 3rd ed. New York: McGraw-Hill, 2006.

[7] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing. System Analysis and Design, 2nd ed.* Cambridge, U.K.: Cambridge Univ. Press, 2010.

[8] G. Paul and A. Pal, "High speed power efficient CGIC digital filters for VLSI applications," in *Proc. 2006 Annu. IEEE India Conf.*, Sept. 15–17, 2006, pp. 1–4.

[9] L. Gazsi, "Explicit formulas for lattice wave digital filters," *IEEE Trans. Circuits Syst.*, vol. 32, no. 1, pp. 68–88, Jan. 1985.

[10] S. Mitra and K. Hirano, "Digital all-pass networks," *IEEE Trans. Circuits Syst.*, vol. 21, no. 5, pp. 688–700, Sept. 1974.

SP

from the GUEST EDITORS (continued from page 17)

evaluates their discusses existing metrics, including perceptually based ones, computed either on 3-D data or on 2-D projections, and evaluates their correlation performance with existing subjective studies.

Study Group 12 (SG12) of the Telecommunication Standardization Section of the International Telecommunication Union (ITU-T) has been involved for many years in standardizing methods for multimedia quality assessment, both subjective and objective. "Multimedia Quality Assessment Standards in ITU-T SG12," by Coverdale et al., gives an overview of existing and emerging SG12 standards, with a special focus on models that predict quality on the basis of parameters and bit stream information available during network planning and monitoring phases.

We hope that this special issue has reached its objective of providing researchers and professionals in the field of multimedia signal processing with timely articles addressing not only the latest advances in the evaluation and assessment of multimedia quality, but also trends and challenges, which in turn provides a solid basis for further progress in this exciting and dynamic field. Enjoy reading! SP