NONCAUSAL FILTERS: POSSIBLE IMPLEMENTATIONS AND THEIR COMPLEXITY

Davide Mattera[(*)] *Francesco Palmieri*[(**)] *Armando Fiore*[(*)]

* Università degli Studi di Napoli Federico II,
Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni,
Via Claudio, 21 I-80125 Napoli Italy
** Seconda Università degli Studi di Napoli,
Dipartimento di Ingegneria dell'Informazione,
Via Roma, 29 I-81031 Aversa Italy

ABSTRACT

Possible solutions for noncausal filter implementation are considered. Detailed calculation of the performance parameters associated with each possible solution is provided. Such calculation shows that the number and the position of the system poles affect the performance of each solution. This can have a profound impact over many linear-filtering applications where the designed filter is periodically adapted without constraining the poles to lie inside the unit circle and, at each adaptation step, the structure for its implementation has to be automatically chosen.

1. INTRODUCTION

In the class of linear time-invariant systems a very important position is held by those with Infinite Impulse Response (IIR) which present a limited number of poles and zeros. Such filters present limited computational complexity with respect to the classical Finite Impulse Response (FIR) filters when recursive implementation in time-domain is adopted [1]. Such an implementation is, however, very simple to be achieved only when all the system poles lie inside the unit circle, i.e., with reference to causal and stable filters.

Stable filters with poles lying also outside the unit circle are difficult to be approximatively implemented in practice also when processing delay is tolerable. For such a reason, the constraint that the system poles be inside the unit circle is often introduced in the filter-design procedure. This constraint, however, limits the potential performance of the obtained filter and renders more complex the design procedure; both these limitations may be significant in many real-word signal processing scenarios. When such a constraint is not imposed at the aim of simplifying the design procedure and avoiding potential performance loss, we are faced with the problem of realizing a filter with poles outside the unit circle. Since the constraint of stability of the obtained filter has to be necessarily imposed, a noncausal filter has to be realized; in practice, it means that a processing delay has to be tolerated.

In this paper we assume that a design procedure has already been performed and a linear filter with poles outside the unit circle has to be implemented. We consider in detail this often neglected problem by introducing all its possible solutions; we also provide quantitative calculation of all the parameters specifying the quality of each possible solution (i.e., the computational complexity and the processing power needed for real-time processing, the required memory and, obviously, the accepted delay) in dependence of the system characteristics.

2. IMPLEMENTATION OF A DESIRED FILTER

Let us assume that we need to implement a given digital filter. The desired filter may be known in terms of its transfer function in the Z-domain:

$$H_d(z) = \frac{\sum_{k=0}^{n_z} b_k z^{-k}}{1 + \sum_{k=1}^{n_p} a_k z^{-k}}, \tag{1}$$

and/or in terms of its impulse response $h_d(n)$. The set of zeros and poles of the system to be implemented are assumed to be determined by a design procedure which does not impose that poles and zeros are inside the unit circle. Therefore, we assume that the filter may be non-causal, but it is stable, i.e., the region of convergence (ROC) for $H_d(z)$ is an annular region including the unit circle. This means that in the transfer function, out of the n_p poles, those that possibly lay outside the unit circle, are to be associated with the anticausal part of the filter [1]. More specifically, suppose that the set of n_p poles are partitioned into two sets of n_s poles inside the unit circle and n_u outside ($n_p = n_s + n_u$); denote with n_z the number of zeros. For the considerations that will follow it is useful to assume that the stable noncausal impulse response can be considered negligible outside the range $\{-n_a, ..., n_b\}$. Such quantities mainly depend on the position of p_{ext} and p_{int}, where p_{ext} is the "dominant" external pole (i.e., the external pole closest to the unit circle) and p_{int} is the "dominant" internal pole (i.e., the internal pole closest to the unit circle): roughly, $n_a = K/\log|p_{ext}|$ and $n_b = K/\log(1/|p_{int}|)$ where K is a constant which depends on the quality of the implementation.

A designer faced with the problem of implementing such a filter has essentially three major choices:

1. Approximate a time-delayed version of the filter with a causal FIR filter (FIR approximation);
2. Approximate a time-delayed version of the filter with a causal and stable IIR filter (IIR approximation);
3. Represent the noncausal IIR filter by interconnecting smaller subfilters; each subfilter is required to be causal or anticausal (Decomposition).

When FIR approximation is adopted, the designer has different choices, such as:

1.a Determine the approximating causal FIR filter by truncating the delayed impulse response of the desired filter;

1.b Determine the approximating causal FIR filter by adopting a least-mean square criterion in the frequency domain.

When IIR approximation (i.e., option 2) is adopted, a criterion which minimizes the error in the frequency domain has to be adopted. When decomposition (i.e., option 3) is adopted, the anticausal subfilters, which may be obviously implemented by adopting FIR or IIR approximation, can also be realized backward in time (backward filtering) according to the recursive equation which defines each subfilter.

3. FIR SOLUTION

The most straightforward approach to the implementation of the desired filter is the option 1.a: truncate the impulse response to its non-negligible values and introduce a delay. The resulting causal FIR filter is $h_F(n) = h_d(n - n_a)$ for $n = 0, \ldots, n_a + n_b$, and zero otherwise. In such a case, as reported in Tab. I, the number of multiplications per output sample required by a direct-form implementation of the approximated FIR filter is $n_a + n_b + 1$ (these multiplications have to be performed in a single time-step), the required memory is $n_a + n_b$ signal samples and the processing delay (introduced to satisfy causality constraint) is n_a. The numbers n_a and n_b need to be sufficiently large so that all the significant values of the causal and anticausal parts of the impulse response $h_d(n)$ are included in the finite response of the filter.

When the FIR filter implementation is based on the Fast Fourier Transform (FFT), the computational complexity may be strongly reduced. In fact, in such a case, the number of complex multiplications per output sample of a causal FIR system with n tap is [1] $\dfrac{\log_2(2H)}{1 - \frac{n-1}{H}}$ where $H > n$ is the length of the input blocks on which the FFT is performed; this is also the maximum number of complex multiplications per output sample needed to satisfy the real-time processing constraint; the required memory is H and the delay is still H. Therefore, as reported in Tab. I, the FFT implementation of the FIR approximation requires $\dfrac{\log_2(2H)}{1 - \frac{n_a+n_b}{H}}$ complex multiplications per output sample; this is also the maximum number of complex multiplications per output sample needed to satisfy the real-time processing constraint; the required memory is H and the delay is $H + n_a$ ($H > n_a + n_b + 1$).

The efficiency of such an implementation depends clearly on the total length of the filter which is really determined by how close to the unit circle the system poles are located. The main advantage of such a solution is the inherent stability of the obtained FIR filter. The main disadvantage is that, when either $|p_{\rm int}|$, or $|p_{\rm ext}|$ is close to one, the values of n_a and n_b may be quite large. This may imply considerable computational complexity of this solution when compared with those considered in the following subsections.

4. OPTIMUM CAUSAL AND STABLE IIR FILTER

We can find an IIR causal and stable approximation of the delayed causal impulse response $h_I(n) = h_d(n - n_0)$, $n \geq 0$ ($h_I(n) = 0$, $n < 0$). The delay can be chosen as $n_0 = n_a$, or sufficiently large to allow the causal impulse response $h_I(n)$ to include most of the total energy of the desired response. In such a way, all the values of $h_d(n)$ for $n < -n_0$ are considered irrelevant, and, therefore, ignored. We have to note that our success in finding a good causal IIR approximation of the desired filter is dependent on the capability of an IIR filter to match an impulse response that may peak somewhere far from the sample $h(0)$ and that typically corresponds to a non-minimum phase system.

As far as finding the coefficients of a stable IIR filter, many techniques can be devised [1]. We point here to one quite popular that matches the desired impulse response $h_I(n)$ to a stable causal IIR filter in the frequency domain (on the unit circle). The cost function is:

$$\sum_{k=0}^{N-1} \left| H_d(e^{j\omega_k}) e^{-j\omega_k n_0} - \frac{\sum_{i=0}^{N_z} b_i e^{-j\omega_k i}}{\sum_{i=0}^{N_p} a_i e^{-j\omega_k i}} \right|^2, \qquad (2)$$

where $\{\omega_k\}$ ($k = 0, \ldots, N-1$) is a set of frequency samples and N_z and N_p are the number of zeros and poles in the approximating causal IIR filter, respectively.

This problem is usually solved by a classical optimization method under the constraint of stability of the overall filter, i.e. $|p_j| < 1$ $j = 1, \ldots, N_p$ with p_j zeros of $\sum_{\ell=0}^{N_p} a_\ell z^{-\ell}$. The matlab routine ***invfreqz*** implementing this method utilizes as optimization algorithm a damped Gauss-Newton method [3], initialized with the values $\{a_i, b_i\}$ which minimize the following unconstrained cost function:

$$\sum_{k=0}^{N-1} \left| H_d(e^{j\omega_k}) e^{-j\omega_k n_0} \sum_{i=0}^{N_p} a_i e^{-j\omega_k i} - \sum_{i=0}^{N_z} b_i e^{-j\omega_k i} \right|^2$$

The minimum of such a quadratic cost function can be easily determined by solving a linear system.

Unfortunately, as already pointed out, many free parameters need to be fixed for the algorithm to be applied, such as the number of zeros and the number of poles of the approximating filter, the number N of the numerical frequencies utilized in the cost function and the already mentioned delay n_0 introduced to improve the quality of the approximation. The optimum choice of the delay n_0 under this approach has been considered in [2] where an algorithm for optimum selection of n_0 is proposed. Moreover, to ran the algorithm we need to choose also the number of iterations, constraint tolerance, etc. Furthermore, the implemented filter may not provide a sufficiently good approximation of the desired one unless a large number of poles and zeros is used. Note that the algorithm for IIR approximation considered here can also be utilized for FIR approximation (option 1.b) by specializing to zero the number of poles of the approximating structure ($N_p = 0$).

5. DECOMPOSITION-BASED IMPLEMENTATIONS

When the overall noncausal filter is decomposed in a interconnected structure involving only causal and anticausal elements, two kinds of problems need to be solved:

a) how to decompose the structure of the overall filter utilizing only causal and anticausal elements;

b) how to implement the anticausal elements of the structure.

These issues will be addressed in the following subsections.

5.1. Decomposing the overall filter

There are three main basic ways for interconnecting different systems: parallel, series and feedback. All these tools can be utilized for obtaining the required decomposition and many special structures may be conceived. However, when feedback is introduced, stability of the overall system does not follow from stability of each subsystem. For such a reason, we do not consider feedback in the interconnecting structures. We therefore have three fundamental structures utilizing only causal and anticausal elements:

1 A cascade of two elements where the first filter contains all the n_s poles that lie inside the unit circle while to the second element of the structure are assigned all the n_u poles that lie outside the unit circle. The zeros of the overall filter may be partitioned in all the possible ways among the two elements of the cascade.

2 This second structure can be obtained from the first one by inverting the order of the two subsystems in the structure 1.

3 A parallel of two filters: the first one contains the poles that lie inside the unit circle and the second one contains those outside the unit circle.

When a cascade decomposition is adopted, there are 2^{n_z} different ways of splitting the zeros between the two subsystems; since different partitions may produce different lengths of the impulse responses of the two elements of the cascade (in, particular, that of the anticausal system), proper allocation of zeros represents a significant issue. We propose, here, to consider the pole p_{ext} and minimize the amplitude of the component of the impulse response associated with this pole. Let us note, however, that the optimum partition can be determined when n_z is sufficiently small by evaluating the length of the two impulse responses in correspondence of each of the 2^{n_z} possible alternatives.

5.2. Implementing the anticausal filters

Once decomposed in causal and anticausal subfilters, the implementation of the causal component can be realized by utilizing a classical recursive structure [1]. The problem of implementing the anticausal subfilter admits two possible approach:

1 This problem can be seen as a special case of the general problem of implementing a noncausal filter; therefore, the previously considered solutions (FIR and IIR approximations) can also be adopted for implementing the anticausal component.

2 A stable and anticausal filter can be implemented by recursively filtering backward in time the difference equation that defines it, provided that a sufficiently large number of input samples is stored. In such a way, processing delay and system memory can be traded-off with the required computational complexity.

5.2.1. Decomposition plus FIR or IIR approximation

When we utilize an approximating FIR filter for implementing only the anticausal elements of the chosen structure, we can overcome one of the shortcomings of FIR approximation, i.e., the fact that an internal pole close to the unit circle leads to a large computational complexity. In particular, n_b taps (due to the causal part of the overall filter) present in the pure FIR approximation are replaced from $n_s + z_s$ elements required by an IIR implementation of the causal component of the overall system where z_s denotes the number of zeros assigned to the causal element of the structure. The length of FIR filter approximating the anticausal element of the structure is therefore roughly equal to the length of the anticausal filter; it can be roughly estimated with n_a. Therefore, as reported in Tab. I, when the direct-form implementation of the FIR filter is considered, the number of multiplications per output sample is $n_a + n_s + z_s + 2$ (these multiplications have to be performed in a single time-step), the required memory is $n_a + n_s + z_s$ signal samples and the processing delay (introduced to satisfy the constraint of causality) is still n_a.

When the FIR filter implementation is based on the Fast Fourier Transform (FFT), the computational complexity may be strongly reduced. As reported in Tab. I, the FFT implementation with a decomposition and an FIR approximation of the anticausal component requires $\frac{\log_2(2H)}{1-\frac{n_a}{H}} + n_s + z_s$ complex multiplications per output sample ($H > n_a$); this is also the maximum number of complex multiplications per output sample needed to satisfy the real-time processing constraint; the required memory is $H + n_s + z_s$ and the processing delay is $H + n_a$.

5.2.2. Backward filtering

The anticausal nature of the subfilter can be exploited to implement it in a recursive fashion. In fact, knowledge of its poles and zeros is equivalent to knowledge of a discrete-time difference equation that describes the behavior of the system:

$$y(n) = \sum_{k=0}^{z_u} d_k x(n-k) - \sum_{k=1}^{n_u} c_k y(n-k) \quad (3)$$

Such an equation can be re-written in $n_u + 1$ different ways:

$$y(n-i) = \sum_{k=0}^{z_u} \frac{d_k}{c_i} x(n-k) - \sum_{k=0}^{i-1} \frac{c_k}{c_i} y(n-k) - \sum_{k=i+1}^{n_u} \frac{c_k}{c_i} y(n-k) \quad (4)$$

with $i = 0, \ldots, n_u$ and $c_0 \triangleq 1$. By choosing $i = 0$ in (4), we obtain relation (3). When $1 \leq i \leq n_u - 1$, the relation (4) cannot be used to implement any system because the output at time k depends on the output samples at time instants $k-1$ and $k+1$. Finally, since all system poles lie outside the unit circle, the system realized according to the recursive relation (4) with $i = n_u$ represents the desired anticausal and stable system provided that $z_u \leq n_u$.

Such an anticausal system can be realized provided that the input signal is recorded and is utilized backward in time according to its anticausal nature. Such a processing has to be realized in a block-by-block fashion by utilizing, in a time-reversed structure, the classical methods for implementing IIR filters [1].

A particular characteristic of backward filtering is that the output at time k depends on the future outputs. Since the first output sample of the block to be calculated is the last one in temporal order, we are unable to correctly calculate it. For such a reason, we need to set the initial conditions in backward-filtering to dummy values and to discard the incorrectly obtained output samples until the transient behavior due to the incorrect initial conditions extinguishes; we may estimate with n_a the length of this transient behavior. Alternatively, we can also estimate the unknown initial conditions of the backward filter (or exactly determine them by using FIR approximation).

A block of length $B > n_a$ (containing the input samples in the time-instants $(k, k+B-1)$) is taken in input and a block of length $B - n_a$ (containing the overall output samples in the time-instants $(k, k+B-n_a-1)$) is provided in output; then, the next block contains the input samples in the interval $(k+B-n_a, k+2B-n_a-1)$ and the output block contains the values in the time interval $(k+B-n_a, k+2B-2n_a-1)$, and so on.

The processing delay changes in the different positions of the block. The maximum value is the delay of the first element of the block; it includes the length B of each block and the discrete-time m_p needed to process it. Note that, to satisfy real-time-processing constraint, m_p must be smaller than the discrete-time $B - n_a$ needed to collect the remaining part of the next block. The memory needed for implementing such a structure is roughly an input and output block of B samples and the $n_s + n_u + n_z$ values in the two recursive structures. We have also determined the number of complex multiplications per each output sample and the minimum number of complex multiplications per each discrete-time instant required by the processor and we have reported them in Tab. I for two possible implementations of the anticausal filters, the direct-form and the parallel-form implementation.

The calculation for direct-form implementation follows from the following observations: the causal filter calculates $B - n_a$ output samples per block and requires $n_s + z_s + 1$ multiplications per sample; the anticausal element determine B output samples per block and requires $n_u + z_u + 1$ multiplications per sample; the overall outputs per block is $B - n_a$; structure 2, differently from the alternative ones, cannot start working before the input block is completely collected. Note also that, in the parallel-form implementation, the transient behaviors of the subfilters associated with the external poles have different lengths n_i. This affects the complexity expecially when the transient is relevant (excessive delays are not acceptable).

6. CONCLUSIONS

We have considered the problem of noncausal filter implementation; a detailed calculation of the performance parameters of each solution has been provided. Such a calculation shows that significant advances can be achieved by recursive time-backward filtering when there are external poles very close to the unit circle. Not always, however, this is the best choice. For such a reason, the quantitative study about noncausal filter implementation, reported in Tab. I, can have a profound impact on a very large variety of applications where linear filtering constitutes an important tool. In the literature, in fact, the advantages of unconstrained filter design (i.e., complexity of the design procedure and potential performance improvements) are often neglected and, also when considered, filter design is strictly associated with a particular filter implementation which, as shown in the paper, cannot be the best one for different positions of the poles. This is very relevant in adaptive scenarios (where, at each adaptation step, a different linear system has to be automatically implemented) and it affects the overall advantage of utilizing unconstrained filter design.

7. REFERENCES

[1] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing*, Macmillan publishing company, 1988.

[2] R. Vuerinckx, Y. Rolain, J. Schoukens, and R. Pintelon, "Design of stable IIR filters in the complex domain by automatic delay selection," *IEEE Trans. on Signal Processing*, vol. 44, pp. 2339 – 2344, 1996.

[3] J.E.Jr. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

	Computational Complexity (# of mult. per each output)	Maximum filter delay	Required memory	Real-time constraint (max # of mult. per time step)
FIR Approx. + direct-form	$n_a + n_b + 1$	n_a	$n_a + n_b$	$n_a + n_b + 1$
FIR Approx. + FFT	$\frac{\log_2(2H)}{1 - \frac{n_a+n_b}{H}}; H > n_a + n_b + 1$	$n_a + H$	$2H$	$\frac{\log_2(2H)}{1 - \frac{n_a+n_b}{H}}$
IIR Approx.	$n_a + n_s + z_s + 2$	n_a	$n_a + n_s + z_s$	$n_a + n_s + z_s + 2$
Decom. + FIR (or IIR) in direct-form	$n_a + n_s + z_s + 2$	n_a	$n_a + n_s + z_s$	$n_a + n_s + z_s + 2$
Decom. + FIR with FFT	$\frac{\log_2(2H)}{1 - \frac{n_a}{H}} + n_s + z_s$	$n_a + H$	$H + n_s + z_s$	$\frac{\log_2(2H)}{1 - \frac{n_a}{H}} + n_s + z_s$
Decomposition + backward filtering in direct-form	$\frac{B}{B-n_a}(n_u+z_u+1)+n_s+z_s+1$	$B + m_p$ $m_p \leq B - n_a$ $m_p > 0$	$2B + n_p + n_z$	$\max\left(n_s+z_s+1, \frac{B}{m_p}(n_u+z_u+1)\right)$ Str. 1
				$\frac{B(n_u+z_u+1)+(B-n_a)(n_s+z_s+1)}{m_p}$ Str. 2
				$\max\left(n_s+z_s+1, \frac{B}{m_p}(n_u+z_u+1)\right)$ Str. 3
Decomposition + backward filtering in parallel form	$n_s+z_s+1+2n_u + 2\frac{\sum_{i=1}^{n_u} n_i}{B-n_a}$	$B + m_p$ $m_p \leq B - n_a$ $m_p > 0$	$2B + n_p + n_z$	$\max(n_s + z_s + 1, \frac{2n_u(B-n_a)+2\sum_{i=1}^{n_u} n_i}{m_p})$ 1
				$\frac{(B-n_a)(n_s+z_s+1)+2n_u(B-n_a)+2\sum_{i=1}^{n_u} n_i}{m_p}$ 2
				$\max(n_s + z_s + 1, \frac{2n_u(B-n_a)+2\sum_{i=1}^{n_u} n_i}{m_p})$ 3

Tab. I: Performance parameters of each possible solution.