Turk J Elec Eng & Comp Sci, Vol.20, No.5, 2012, © TÜBİTAK
doi:10.3906/elk-1012-945

Noncausal forward/backward two-pass IIR digital filters in real time

Adnane MOUFFAK[1,*], **Mohamed Faouzi BELBACHIR**[2]
[1] *Department of Sciences and Techniques, Faculty of Sciences and Technology, University of Mascara, Mascara 29000-ALGERIA*
e-mail: amouffak@gmail.com
[2] *Laboratory: Signals, Systems and Data, University of Sciences and Technology of Oran Mohamed Boudiaf, Oran-ALGERIA*
e-mail: mf_belbachir@yahoo.fr

Received: 28.01.2011

Abstract

A novel method for implementing noncausal forward/backward 2-pass recursive digital filters in real time is presented. It is based on a segment-wise block processing scheme without overlapping. Factors that degrade the linearity of the overall system's transfer function are discussed. An analytical condition that corrects the system's linearity is elaborated upon using the state variable approach. A recursive algorithm is developed to compute an implementable condition for real-time filtering. A single first in, first out queue memory is introduced to ensure an organized and continuous data stream into the proposed system. This technique allows real-time, sample-by-sample filtering, and it yields reduced delay and data storage memory compared to previous works. Better performances in total harmonic distortion were also obtained. Experimental results are illustrated.

Key Words: *Noncausal filters, nonoverlapping approximation, real-time filtering, segment-wise processing*

1. Introduction

Noncausal filters are usually realized using a combination of 2-pass filtering and time reversal [1-7]. The first pass can be performed in a forward direction using a stable recursive digital filter, and the second pass can be performed in a backward direction using a noncausal subsystem implemented by 2 time reversal operations and a stable recursive digital filter, as shown in Figure 1. Another scheme can be used by performing the backward pass first and the forward pass second (Figure 2). Here, $H(z)$ is the transfer function of a causal and stable infinite impulse response (IIR) digital filter; $[x(n)]$ and $[y(n)]$ are the input and output sequences, where $[v(n)]$ and $[w(n)]$ are the outputs relative, respectively, to the first and second IIR filters; and S_1 and S_2 are the initial state vectors.

[*] Corresponding author: Department of Sciences and Techniques, Faculty of Sciences and Technology, University of Mascara, Mascara 29000-ALGERIA

These 2 schemes can realize stable noncausal transfer function $H(z)H(z^{-1})$ with arbitrary pole and zero locations inside and outside the unit circle. Unlike the causal IIR filter design problem, there is no conflict among stability, magnitude, and phase performances in designing noncausal IIR filters [2,4]. This class of filters has high performance in magnitude using, for example, an elliptic digital filter design and a linear phase [4]. These high performances are strongly required in some special cases of noise suppression filtering where the signal-to-noise ratio (SNR) is very low, such as in speech parameter estimation in a high-pass band [8-11]; for extracting weak auditory evoked potentials from spontaneous electroencephalogram signals [12]; or for filtering high-frequency noise from noisy electrocardiogram (ECG) signals [5,13]. The computational advantages of noncausal IIR filters over causal IIR or finite impulse response (FIR) filters are clearly indicated in [2,4,5]. However, an output of an IIR filter-based implementation of the above schemes is only achieved without errors when the input is of finite length [1,2]. Thus, this class of filters is often limited to problems for which offline filtering is permissible.

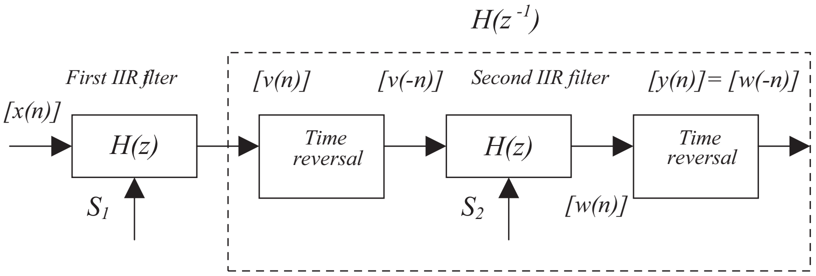

Figure 1. Noncausal 2-pass recursive digital filter scheme; the noncausal system is performed at the second pass.

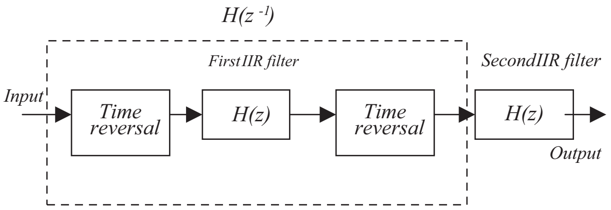

Figure 2. Noncausal 2-pass recursive digital filter scheme; the noncausal system is performed at the first pass.

Noncausal recursive filters can be extended to process real-time, infinite-length input sequences using block segment-wise processing techniques. A continuous infinite-length input is divided into finite-length sections and each section is filtered separately by the 2-pass scheme. The final output can be built using output sections yielded from finite-length section processing and overlap-based approximations such as overlap-and-save techniques [2,5], overlap-and-add techniques [3,4,6,7], or the nonoverlapping technique [1]. The final output is never achieved without errors; the magnitude of the systematic errors depends on section length and degrades the linearity of the overall system. The major problems related to this class of filters in real-time processing are:

-Unavoidable systematic errors occur from the noncausal subsystem, $H(z^{-1})$, and substantial data storage registers are required for acceptable systematic errors.

-The final output, $[y(n)]$, is substantially delayed because of time reversal operations and segment-wise block processing techniques for realizing the noncausal part, $H(z^{-1})$.

Czarnach [2] proposed a method using state vector representation allowing reduced systematic errors as compared to Kormylo and Jain's work [5]. Using a proper choice of the initial state vector of the second recursive filter, S_2, his technique can eliminate a part of the systematic errors yielded from the truncation of the first IIR filter's free response.

Powell and Chau [4] devised an efficient scheme for real-time 2-pass recursive digital filters with a linear phase, based on a well-known overlap-and-add algorithm for sectioned convolution. The noncausal portion of overall transfer function $H(z^{-1})$ is created at the first pass and causal portion $H(z)$ is created at the second pass, as shown in the general scheme in Figure 2. Section length M for overlap-add segment-wise block processing is chosen such that the impulse response of $H(z)$ decays to a negligible value after the Mth sample [4,14]. The major contributions of [4] were a reduction in the total delay between input and output with more efficiency than the previous method in [2].

Arias-de-Reyna and Acha [1] proposed an efficient scheme for real-time 2-pass recursive filters with a linear phase based on a segment-wise processing technique without overlapping. Their technique was based on interpreting the impulse response of the overall noncausal system as an autocorrelation. Their major contribution was a reduction in computational burden compared to the overlap-and-save method of Czarnach [2].

In this paper, a technique for realizing noncausal 2-pass recursive digital filters in real time is developed using a state variable approach. The proposed technique performs without overlapping and yields reduced delay and data storage as compared to Arias-de-Reyna and Acha's method [1]. Systematic errors are reduced compared to Powell and Chau's technique [4].

Section 2 presents an analytical investigation of the segment-wise processing effects when an infinite-length input is filtered. We study why the output cannot be achieved without systematic errors. We obtain a theoretical condition achieving the ideal output using a state variable approach. With this different point of view, we get the same expression as in [1]. An implementable condition calculated recursively is presented in Section 3. A proposed scheme based on segment-wise block processing without overlapping and a first in, first out (FIFO) queue memory are described in Section 4. Performances are evaluated, as well. Section 5 is concerned with some experimental results, and methodology and results are also illustrated. A conclusion is given in Section 6.

2. Analytical condition yielding a final output without errors

In this section, we investigate the analytical condition that eliminates the segment-wise processing effects. We emphasize factors of our study that make a real-time 2-pass filtered final output able to be exactly calculated. For this purpose, the basic scheme shown in Figure 1 is used.

The state variable description of the first and second recursive filters of the Nth-order identical subsystems of transfer function $H(z)$ are given by Eqs. (1) and (2), respectively.

$$\begin{cases} S_1(n+1) = AS_1(n) + Bx(n) \\ v(n) = CS_1(n) + Dx(n) \end{cases} \tag{1}$$

$$\begin{cases} S_2(n+1) = AS_2(n) + Bv(-n) \\ w(n) = CS_2(n) + Dv(-n) \end{cases} \tag{2}$$

Here, $S_1(n)$ and $S_2(n)$ are the state vectors, respectively, of the first and the second IIR filter. A, B, C, and D are the fixed system matrices.

An infinite-length input, $[x(n)]$, can be divided into M finite-length sections (Figure 3), such as:

$$[x(n)] = \sum_{k=0}^{+\infty} [x_k(n)] \quad 0 \leq k < +\infty, \tag{3}$$

and:

$$x_k(n) = \begin{cases} x(n) & \text{for } kM \leq n < (k+1)M \\ 0 & \text{outside} \end{cases} . \tag{4}$$

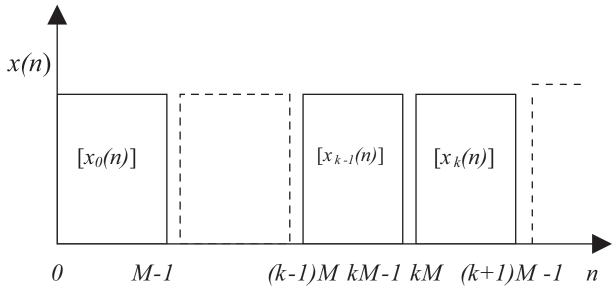

Figure 3. Segment-wise blocks for processing of infinite-length input [x(n)].

In order to allow for reduced data storage as compared to the previous work in [2,4], each section $[x_k(n)]$ is filtered by the 2-pass system until the Mth sample is reached. The first subsystem's initial state, S_1, was used, as in [1,2]. A theoretical condition for the second subsystem's initial state, S_2, which achieves the ideal output, is elaborated upon using a state variable approach.

When segment-wise processing is performed, the final output is obtained through the juxtaposition of the adjacent output sections resulting from each 2-pass IIR filter section. Since each output section is of infinite length, it is necessary to truncate it before processing the next section. According to this situation, the systematic error problem that prevents us from getting an exact output computation can be taken as a result of an unavoidable inherent truncation applied to the first-pass output section response. Once the truncation is carried out, it holds some input samples from exciting the second subsystem. The truncated response part cannot be recovered because it depends on upcoming future sections, in addition to the first subsystem's state vector [1,2].

Our recovering process consists of summarizing the second subsystem's excitation due to the truncated samples by modifying its initial state vector in order to get the same output section response without truncation. The ideal output response of the 2-pass system can be achieved without errors only if all samples, $v(n)$, are calculated until infinity. In this case, the second pass starts filtering with a zero initial state at $n = +\infty$, e.g., $S_2(-\infty) = 0$.

We assume that the first-pass filtering for each section $[x_k(n)]$ is stopped at the moment $n = (k+1)M-1$. The truncation effect is observed when the second pass is performed with a zero initial state vector. Thus, we are trying to recover this truncation effect without using a zero initial state vector, such that the second pass will be executed as if no truncation is done.

With Eq. (2), we get:

$$S_2\left[-(k+1)M+1\right] = A^n S_2\left[-(k+1)M+1-n\right] + \sum_{i=1}^{n} A^{i-1}.B.v\left[(k+1)M-1+i\right]. \tag{5}$$

If we consider that processing by the second filter block has been started with a zero initial state at infinity ($n = +\infty$), we get:

$$S_2\left[-(k+1)M+1\right] = A^{+\infty} S_2\left(-\infty\right) + \sum_{i=1}^{+\infty} A^{i-1}.B.v\left[(k+1)M-1+i\right]. \tag{6}$$

Matrix A describes a stable and causal recursive filter, such that all of its eigenvalues have a modulus smaller than one. This implies that $A^{+\infty}$ will converge to zero.

Thus:

$$S_2\left[-(k+1)M+1\right] = \sum_{i=1}^{+\infty} A^{i-1}.B.v\left[(k+1)M-1+i\right]. \tag{7}$$

Or:

$$S_2\left[-(k+1)M+1\right] = \sum_{i=0}^{+\infty} A^{i}.B.v\left[(k+1)M+i\right]. \tag{8}$$

The output sequence $v\left[(k+1)M+i\right]$ of the first subsystem can be determined by the state vector $S_1\left[(k+1)M\right]$. With Eq. (1), we get:

$$v\left[(k+1)M+i)\right] = C_A^i S_1\left[(k+1)\,M\right] + \sum_{n=0}^{i-1} C_A^{i-n-1} Bx\left[(k+1)\,M+n\right] + D_x\left[(k+1)\,M+n\right]. \tag{9}$$

With Eqs. (8) and (9), we get:

$$S_2\left[-(k+1)M+1\right] = \sum_{i=0}^{+\infty} A^i.B\left\{\begin{array}{l} C_A^i S_1\left[(k+1)\,M\right] + \\ \sum\limits_{n=0}^{i-1} C_A^{i-n-1} Bx\left[(k+1)\,M+n\right] + D_x\left[(k+1)\,M+n\right] \end{array}\right\}. \tag{10}$$

From Eq. (10), we get:

$$\begin{aligned} S_2\left[-(k+1)M+1\right] \quad &= \quad TS_1\left[(k+1)\,M\right] + \sum_{i=0}^{+\infty}\sum_{n=0}^{i-1} A^i BCA^{i-n-1} Bx\left[(k+1)\,M+n\right] \\ &\quad + \sum_{i=0}^{+\infty} A^i BDx\left[(k+1)\,M+i\right], \end{aligned} \tag{11}$$

where T is a fixed matrix depending only on the system, such as:

$$T = \sum_{i=0}^{+\infty} A^i BCA^i. \tag{12}$$

Notice that Eq. (12) can be computed as shown in [1,2], using the conventional eigenvalue diagonalizing matrices technique.

If we assume that $\lambda = i - n - 1$ and we take into account that $0 \le i < +\infty$ and $0 \le n \le i-1$, we get:

$$\begin{aligned} S_2\left[-(k+1)M+1\right] &= TS_1\left[(k+1)\,M\right] + \sum_{n=0}^{+\infty} A^{n+1}\left[\sum_{\lambda=0}^{+\infty} A^{\lambda}BCA^{\lambda}\right] Bx\left[(k+1)\,M+n\right] \\ &\quad + \sum_{n=0}^{+\infty} A^{n}BDx\left[(k+1)\,M+n\right]. \end{aligned} \tag{13}$$

Finally, we get the initial state expression of the second subsystem:

$$S_2\left[-(k+1)M+1\right] = TS_1\left[(k+1)\,M\right] + \sum_{n=0}^{+\infty} A^{n}\left(ATB+BD\right)x\left[(k+1)\,M+n\right]. \tag{14}$$

We assume that:

$$\left\{\begin{array}{l} S_2^{(}0,k) = S_2\left[-(k+1)M+1\right] \\ S_1(0,k) = S_1(kM) \\ S_1^{(}0,k+1) = S_1\left[(k+1)\,M\right] \end{array}\right. . \tag{15}$$

Thus, we have:

$$S_2(0,k) = TS_1(0,k+1) + \sum_{n=0}^{+\infty} A^{n}\left(ATB+BD\right)x\left[(k+1)\,M+n\right]. \tag{16}$$

Eq. (16) presents the expression of the second subsystem's initial state achieving the ideal output without errors. We get the same equation found in [1] by using a state vector representation.

If we consider the processing of the finite-length input, where $x(n)$ is 0 for $n > (k+1)\,M$, we get from Eq. (16) the initial state vector of the second subsystem given by Czarnach in [2].

Note that the term $TS_1(0,k+1)$ corresponds to the previous input sample effects, whereas the sum depends on the future sample inputs until infinity. Therefore, Eq. (16) cannot be implemented in real time. In the next section, we will present a suitable approximation of this equation with a novel implemented scheme.

3. Recursive algorithm for an implementable condition of computation

In this section, we develop an implementable condition of the second IIR filter's initial state vector on the basis of the analytical investigations explained above. Unfortunately, a condition achieving an ideal output without errors seems theoretical and cannot be implemented because of the infinite number of upcoming future samples, $x(n)$, with $n > (k+1)\,M$.

As an implementable approximation, we can involve the upcoming samples from the next block as chosen in [1,2]. Therefore, we implement the following equation:

$$S_2(0,k) = TS_1(0,k+1) + \sum_{n=0}^{M-1} A^{n}\left(ATB+BD\right)x_{k+1}(n). \tag{17}$$

For this purpose, let:

$$\begin{cases} H = [A^n(ATB + BD)]_{0 \le n \le M-1} \\ H(n) = A^n(ATB + BD) \end{cases}, \tag{18}$$

and:

$$X_{k+1}^T = [x_{k+1}(n)]_{0 \le n \le M-1}. \tag{19}$$

From Eq. (17), we get the condensed form:

$$S_2(0,k) = TS_1(0,k+1) + HX_{k+1}. \tag{20}$$

The previous nonoverlapping scheme in [1] consists of 2 subsystems performing in parallel (Figure 4). The filtering subsystem is dedicated to executing the forward/backward processing section by section, and the auxiliary subsystem is used to reduce systematic errors by computing the initial state condition of the second pass, $S_2(0,k)$. However, a direct calculation of the term HX_{k+1} yields a substantial delay of M samples because this manner of computation needs to load all samples from the next segment, $[x_{k+1}(n)]$. Thus, with the previous nonoverlapping method in [1], the first IIR filter should stop and temporize processing after the Mth sample is reached from section $[x_k(n)]$. This temporization should be kept until the loading of all samples from the next segment, $[x_{k+1}(n)]$. Therefore, filtering is delayed by $2M$ samples and the first M samples should be discarded before calculating the term HX_{k+1}.

The scheme shown in Figure 4 leads to the same result, as the filtering subsystem cannot start processing until loading all $2M$ samples from the 2 adjacent sections, $[x_k(n)]$ and $[x_{k+1}(n)]$.

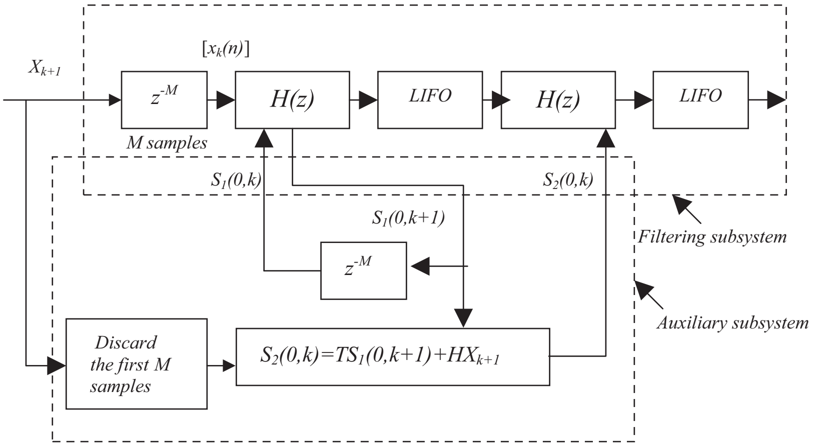

Figure 4. Previous nonoverlapping scheme for a noncausal 2-pass IIR filter.

In order to resolve the issue of the substantial delay imposed in Arias-de-Reyna and Acha's scheme [1], a key aspect of our idea is a recursive computation of the term HX_{k+1} in M steps progressively by loading one sample, $x_{k+1}(n)$, from the next segment, X_{k+1}, at each step n and simultaneously calculating vector $G_k(n)_{N\times 1}$, such that:

$$\begin{cases} G_k(n) = G_k(n-1) + H(n)x_{k+1}(n) \\ G_k(-1) = [0]_{N\times 1} \\ 0 \le n < M \end{cases}. \tag{21}$$

Notice that vector $G_k(n)_{N\times 1}$ is a recursive form of the term HX_{k+1}. Thus, it is fully computed when $n = M-1$, and we have:

$$G_k(M-1) = HX_{k+1}. \tag{22}$$

The second recursive filter's initial state vector can then be calculated by:

$$S_2(0,k) = G_k(M-1) + TS_1(0,k+1). \tag{23}$$

This dynamic process of loading upcoming sample $x_{k+1}(n)$ from the next segment X_{k+1} and progressively and recursively calculating vector $G_{_k}(n)$ at the same time can eliminate the delay of M samples imposed systematically in Arias-de-Reyna and Acha's method [1].

In order to ensure a continuous data stream into the first IIR filter and the auxiliary subsystem sample by sample without extra delay, a FIFO queue memory of M words in length is used to organize loaded samples $[x_{k+1}(n)]$ into a queue data structure, such that the first come is first served.

To understand how the FIFO queue memory runs [15], let us take a closer look at the basic concepts.

- Loading samples can be performed only from the bottom of the FIFO queue memory (write port).
- Reading samples can be done only from the top of the FIFO queue memory (read port).
- Reading tasks can be done only when the FIFO queue is full. In this case, loading of data depends on reading the data in a sequential process of reading and loading.
- When a sequential process of reading and loading is triggered, a sample is read from the top case of the FIFO queue memory, the top case content is cleared, and the case contents are shifted from the bottom to the top cases. Finally, a next sample is loaded into the bottom case. Therefore, the FIFO functioning can be assigned as a simple shift register with dual ports.
- When data are not completely loaded into the FIFO cases (the FIFO queue is empty or almost empty), reading is denied and the loading of data can be done independently from reading. Sequential loading from the bottom can thus be carried out without reading from the top. This situation can have effect only once before starting the filtering, when we load the first block, $[x_0(n)]$.

Therefore, according to the proposed FIFO queue memory, no reading and no filtering are permitted before sequentially loading the first segment, $[x_0(n)] = [x(0)\cdots x(M-1)]$, into the FIFO queue memory of M length. Once that operation is achieved, the sequential process of reading-loading is initiated, the filtering subsystem triggers the processing, and all upcoming samples from the next block are simultaneously loaded into the FIFO queue memory and used by the auxiliary subsystem for the recursive computation of S_2, as explained before.

Examples of FIFO queue memory contents before starting the filtering and during the sequential process of reading-loading when length $M = 3$ are illustrated respectively in Figures 5 and 6. In this example, the filtered section is [*x(0) x(1) x(2)*] and the upcoming samples for the auxiliary subsystem computation are [*x(3) x(4) x(6)*].

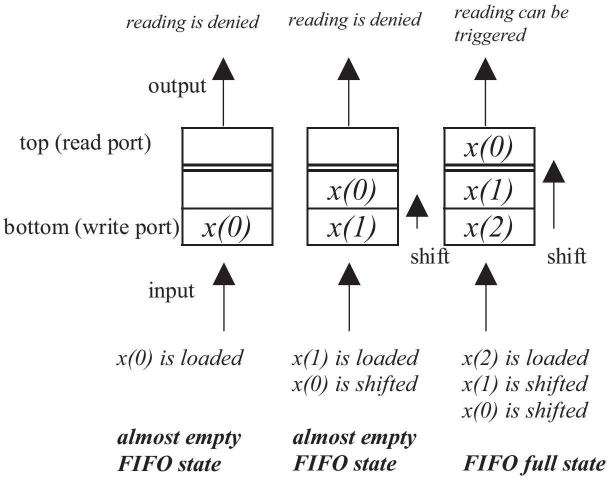

Figure 5. Example of FIFO queue memory contents before starting filtering ($M = 3$). Loading of data can be done independently from reading when the FIFO state is empty or almost empty.

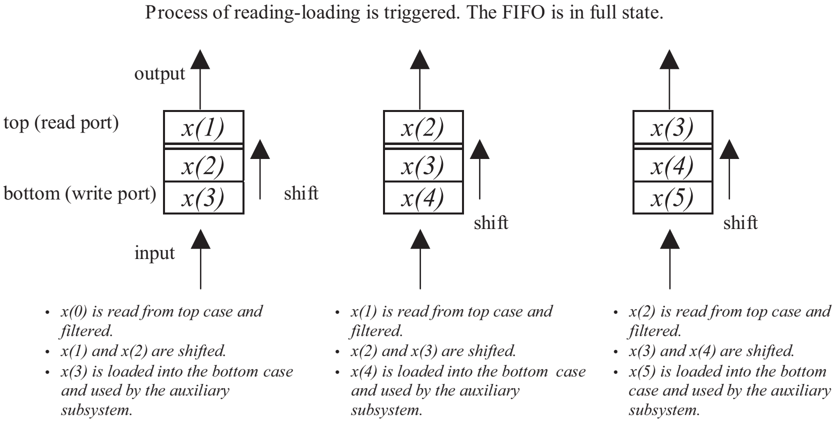
Process of reading-loading is triggered. The FIFO is in full state.
output
top (read port)
bottom (write port)
input
x(1)
x(2)
x(3)
shift
x(2)
x(3)
x(4)
shift
x(3)
x(4)
x(5)
shift
x(0) is read from top case and filtered.
x(1) and x(2) are shifted.
x(3) is loaded into the bottom case and used by the auxiliary subsystem.
x(1) is read from top case and filtered.
x(2) and x(3) are shifted.
x(4) is loaded into the bottom case and used by the auxiliary subsystem.
x(2) is read from top case and filtered.
x(3) and x(4) are shifted.
x(5) is loaded into the bottom case and used by the auxiliary subsystem.

Figure 6. Example of FIFO queue memory contents. The process of reading-loading is trigged and filtering is begun ($M = 3$).

In Section 4, we present a real-time scheme for noncausal 2-pass recursive filter implementation on the basis of the described FIFO queue memory and the recursive algorithm. Performances are evaluated and a comparative study between the proposed and previous works is illustrated.

4. Proposed method for real-time realization

4.1. Description of the proposed scheme

The proposed architecture allows sample-by-sample continuous data to the filtering 2-pass system, which can be divided into 2 subsystems (Figure 7) performing in parallel as follows.

-The filtering subsystem contains the following items:

- One delay of the M-length block caused by loading the first segment, $[x_0(n)] = [x(0) \cdots x(M-1)]$, before starting the filtering, when the FIFO queue is empty or almost empty.

- The first IIR filtering of M samples once the FIFO queue is full and the process of reading-loading is triggered.
- The first reverse operation of M samples using a last in, first out (LIFO) stack memory.
- The second IIR filtering.
- The second reverse operation of M samples using another LIFO stack memory.

-The auxiliary subsystem containing the$S_2(0,k)$ calculation, as follows:

- A recursive computation of HX_{k+1} by computing vector$G_k(n)$in M steps until obtaining$G_k(M-1)$.
- One matrix computation, $(N \times N) * (N \times 1)$, of $TS_1(0,k+1)$.
- A sum of vectors$G_k(M-1) + TS_1(0,k+1)$ for the $S_2(0,k)$calculation.

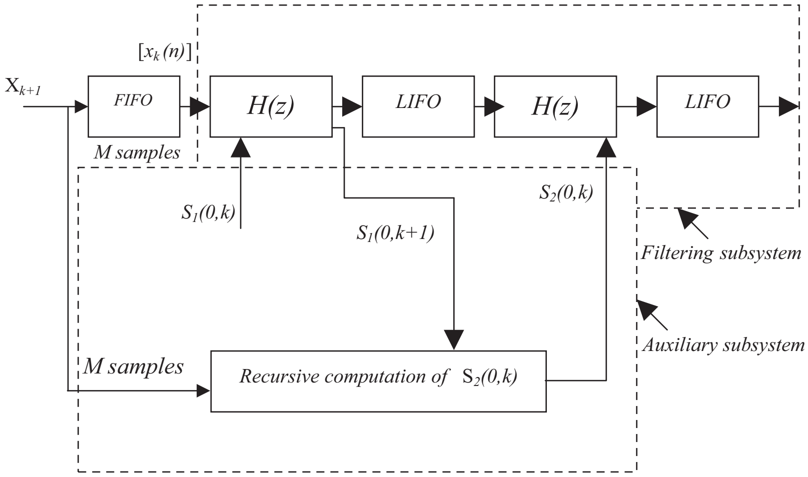

Figure 7. Our proposed scheme for a noncausal 2-pass IIR filter.

As shown in [1,2], initial state vector $S_1(0,k)$ is obtained from filtering the previous section, $[x_{k-1}(n)]$. The FIFO queue memory, explained above in Section 3, simultaneously allows a continuous data stream into the filtering and the auxiliary subsystems. Therefore, the real-time processing can be performed sample by sample. The 2 LIFO stack memories shown in Figure 7 can be implemented as described by Powell and Chau in [4].

4.2. Evaluation of the proposed scheme's performances

In this section, we compare the computational burdens, data storage capacity, and delay of the final output response yielded from the schemes in [1,2,4] and the proposed scheme. Without loss of generality and according to [1], we consider that the recursive filters of N order are implemented as a canonical structure. In this case, each IIR filtering process requires $(2N+1)$ multiplications and $2N$ additions per filtered sample. However, the computation of vector$G_k(n)$requires N multiplications and N additions per loaded sample. Therefore, processing section $[x_k(n)]$of M-length samples by the proposed noncausal 2-pass recursive filtering requires

$2(2N+1)M$ multiplications and $4NM$ additions. If we take into consideration the arithmetic operations related to$TS_1(0,k+1)$and the sum of vectors $G_k(M-1)+TS_1(0,k+1)$, the auxiliary subsystem requires $NM+N^2$ multiplications and $NM+N^2$ additions. Table 1 summarizes the computational burdens of our method and Table 2 illustrates computational efforts as compared between our technique and previous works [1,2,4]. It is clear that the proposed scheme yields approximately the same computational burden as compared to [1] with more efficiency than in [2]; our proposed method also yields the least delay as compared to the methods in [1,2,4] (Table 3).

Data storage capacity seems acceptable in our realization as compared to [1,2,4]. Our scheme requires 1 M-length FIFO queue memory and 2 M-length LIFO stack memories versus a memory of $2M$ in length and 2 M-length LIFO stack memories in [1,2], and 2 M-length LIFO stack memories in [4] (Table 3).

Table 1. Computational efforts of the proposed scheme for each M-length block of processing.

Elementary operations	Number of multiplications	Number of additions
Two filterings of M samples	*2(2N + 1)M*	*4NM*
$G_k(M-1)$ recursive calculation	*NM*	*NM*
One matrix computation, $TS_1(0,k+1)$ *(N × N) × (N × 1)*	N^2	*N(N – 1)*
One sum of 2 vectors: TS_1*(0,k + 1) +* G_k*(M – 1) (N × 1)*	0	*N*
Total	*5NM +* N^2+2M	*5NM +* N^2

Table 2. Computational efforts compared between several methods for each M-length block of processing.

Method	Number of multiplications	Number of additions
Arias-de-Reyna and Acha [1]	*5NM +* N^2+2M	*5NM +* N^2--N
Czarnach [2]	*6NM +* N^2+3M	*6NM +* N^2-N
Proposed	*5NM +* N^2+2M	*5NM +* N^2

Table 3. Delay of the final output that response processing yielded and data storage capacity in samples required by several schemes.

Method	Delay in samples	Data storage capacity in samples
Arias-de-Reyna and Acha [1]	$4M$	$4M$
Czarnach [2]	$4M$	$4M$
Powell and Chau [4]	$3M+1$	$2M$
Proposed	$3M$	$3M$

5. Experimental results

5.1. Methodology

In order to evaluate the proposed system performances as compared to the previous schemes in [1,2,4], MATLAB and C programming language models of these systems were created.

As is well known, these systems perform differently for input sequences with a finite length (less than M samples) than for inputs of infinite length (more than M samples). Implementations of the equivalent transfer function $H(z)H(z^{-1})$ can thus be tested in real time with infinite-length inputs using segment-wise block processing techniques via forward/backward schemes. The systematic errors can be evaluated in the time domain [1] using a criterion of a relative error, such that:

$$relative\,error = \frac{|y_{ideal}(n) - y(n)|}{|x_{\max}|}. \tag{24}$$

Here, ideal output $[y_{ideal}(n)]$ can be calculated by Czarnach's processing of the finite-length input [2], whereas, $y(n)$ is computed by the several segment-wise processing techniques given above and $x_{\max}$ is the maximum of input signal $[x(n)]$.

We can use the total harmonic distortion (THD) to measure systematic errors, known for the most part for degrading the linearity of these systems. This measurement is computed in the frequency domain [4] using a single frequency sinusoidal input of L length:

$$x(n) = W_r(n). \sin(2\pi.f_0.n). \tag{25}$$

Here, W_r is a rectangular window, and the frequency $f_0 = l/L$ should be inside the filter's passband. We choose an integer l for the sake of avoiding the Gibbs distortion caused by a finite-length input (multiplication by rectangular window W_r) [4]. However, we still get the same spectrum as that resulting from an infinite sinusoidal input. The THD can then be calculated using the following equation:

$$THD(dB) = 10\log_{10}\left(\frac{\sum\limits_{k=0}^{N_{FFT}-1}\left\|F_k - \tilde{F}_k\right\|_2}{\sum\limits_{k=0}^{N_{FFT}-1}\left\|\tilde{F}_k\right\|_2}\right). \tag{26}$$

This equation gives the ratio of the root mean square (RMS) error to output spectrum response F, and the ideal spectrum response $\tilde{F}$ over the RMS value of this ideal spectrum response $\tilde{F}$. Here, N_{FFT} is the number of fast Fourier transform points used to calculate the discrete Fourier transform.

5.2. Results

A noncausal filter having the specifications $F_p = 0.3, F_s = 0.325$, $\delta_p = 0.01\ dB$, and $\delta_s = 70dB$ [4] is used as an example to evaluate linearity in both the time and frequency domains, where $F_{p,s}$ is the normalized passband (stopband) frequency and $\delta_{p,s}$ is the passband (stopband) ripple.

Transfer function $H(z)$ is determined using a well-known conventional elliptic design [4]. We get a seventh-order elliptic filter defined by coefficient vectors A and B such that:

$$H(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4} + a_5z^{-5} + a_6z^{-6} + a_7z^{-7}}{1 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}b_5z^{-5} + b_6z^{-6} + b_7z^{-7}}, \tag{27}$$

and:

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 0.1796 \\ 0.8249 \\ 1.9352 \\ 2.8650 \\ 2.8650 \\ 1.9352 \\ 0.8249 \\ 0.1796 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.8601 \\ 3.0921 \\ 2.6427 \\ 1.9480 \\ 0.7836 \\ 0.2577 \\ 0.0253 \end{bmatrix}. \tag{28}$$

We consider an infinite length input of $L = 4100$ samples filtered by the 2-pass systems shown above such that the segment-wise processing is performed with a section length of 205 samples. We use the same input signal as in [1], which consists of the first odd harmonics of a square wave with a fundamental period of 160 samples, as follows:

$$x(n) = \sum_{j=1}^{4} \frac{1}{2j-1} \sin[0.0125\pi(2j-1)n]. \tag{29}$$

All 4 harmonics are inside the passband. The linearity of this system can be observed in the time domain when we obtain the same shapes from both the input and output without considering transients and delays. Figures 8-11 illustrate, respectively, the input, the proposed scheme's output response, and relative errors yielded from the proposed and the Powell and Chau realizations. Sequences are presented without considering transients and total delays. Results obtained from [1,2] are not given here, because we obtain the same outputs and relative errors with the proposed method.

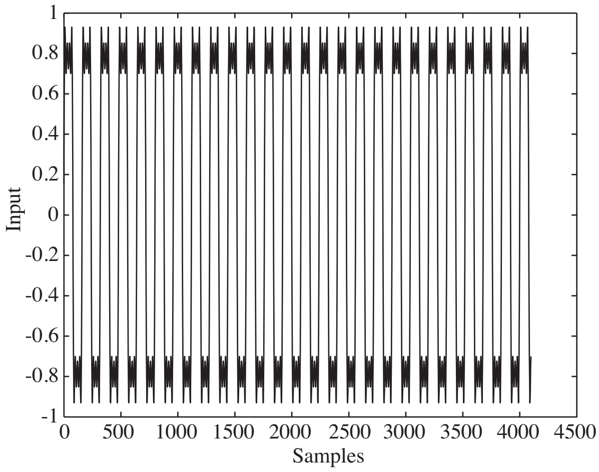

Figure 8. Input signal of 4100 samples with a fundamental period of 160 samples.

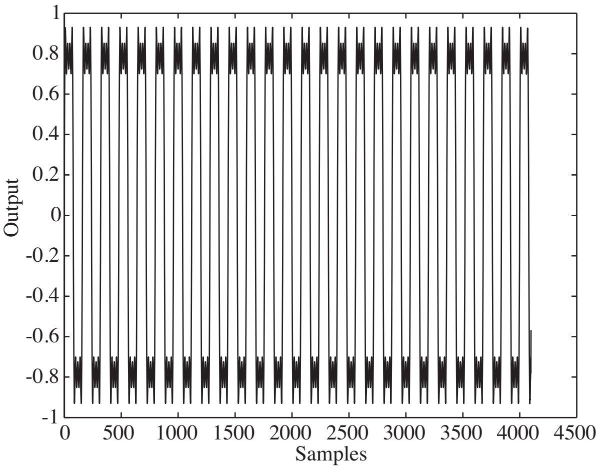

Figure 9. Output response yielded from the proposed 2-pass recursive filter scheme with section length $M = 205$.

In the second experiment, an ECG signal with additive synthetic Gaussian noise is filtered by a Parks-McClellan optimal FIR filter and the noncausal 2-pass recursive filters with the different schemes shown above. The low-pass filter has the following specifications: F_p = *49.5 Hz*, F_s = *54 Hz*, δ_p = *0.001 dB*, and δ_s = *140 dB* as in [13], and sampling frequency = *250 Hz.* With the Parks-McClellan optimal FIR filter, we get the order of 366, whereas we get the order of 13 using an elliptic design with the same specifications. In order to show the graphs clearly, we use a zoom for one ECG. Results are illustrated in Figures 12-14. Notice that the filtered ECG signals from the proposed realization and from Arias-de-Reyna and Acha's scheme [1] and Czarnach's scheme [2] are identical if we do not take into consideration transients and delay caused by real-time processing; therefore, we only present the filtered ECG signal from the proposed scheme. The filtered ECG signal from Powell and Chau's realization [4] is not presented here because the difference is not visible; however, relative error illustrations (Figures 15 and 16) clearly show the gain in systematic error reduction with our method (maximum relative errors of about 5.5 × 10^{-3}) as compared to that in [4] (about 14 × 10^{-3}).

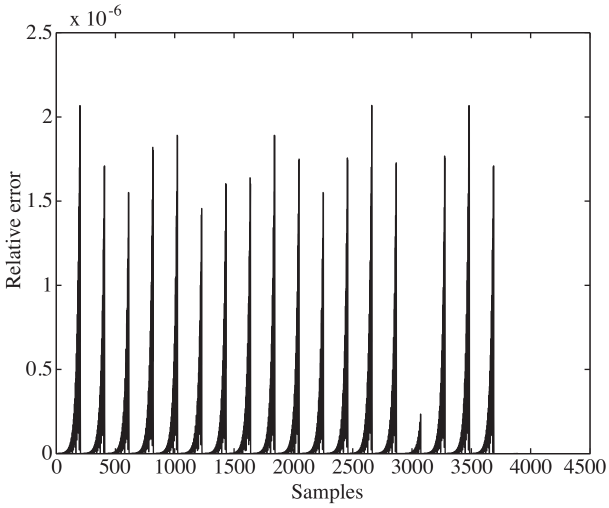

Figure 10. Relative errors yielded from the proposed 2-pass recursive filter scheme ($M = 205$).

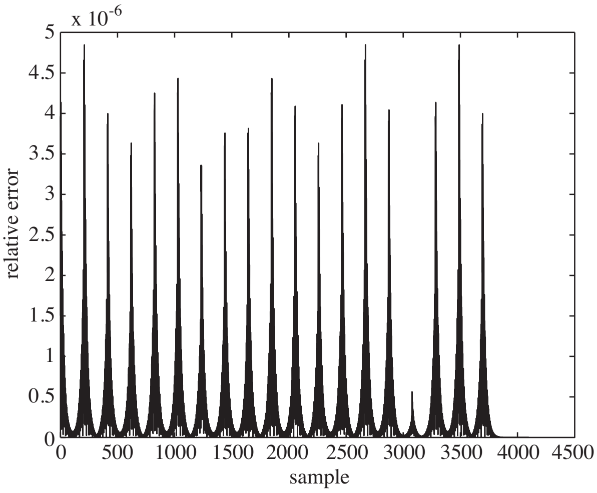

Figure 11. Relative errors yielded from Powell and Chau's 2-pass recursive filter realization (M = 205).

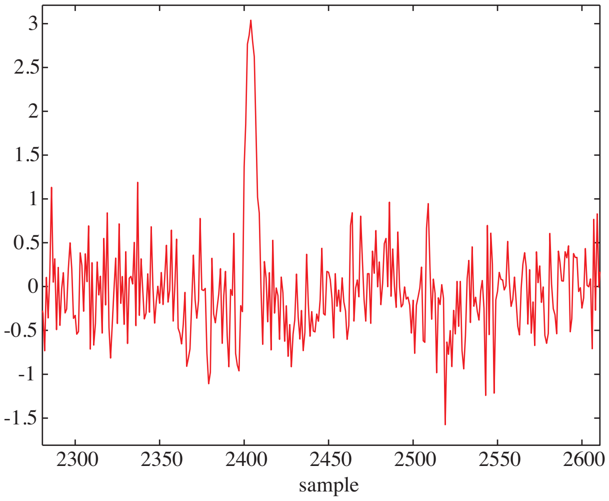

Figure 12. Noisy ECG signal with a total length of $L = 5000$ samples.

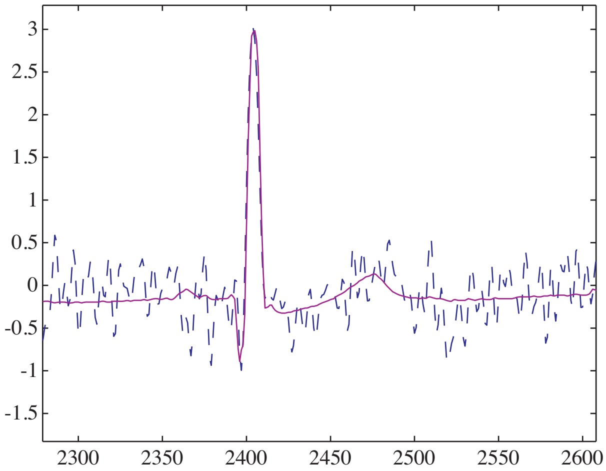

Figure 13. Filtered ECG signal (dashed line) with Parks-McClellan optimal FIR filter, original ECG signal (solid line), section length $M = 250$ samples, total length $L = 5000$ samples, FIR filter with order of 336.

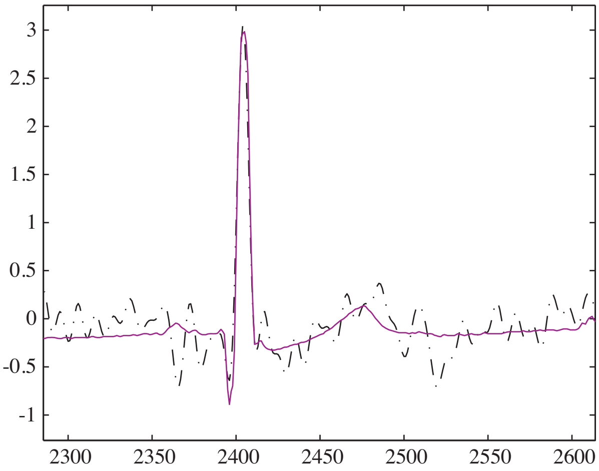

Figure 14. Filtered ECG signal (dashed line) with proposed noncausal 2-pass recursive filters, original ECG signal (solid line), section length $M = 250$ samples, total length $L = 5000$ samples, elliptic recursive filter with order of 13.

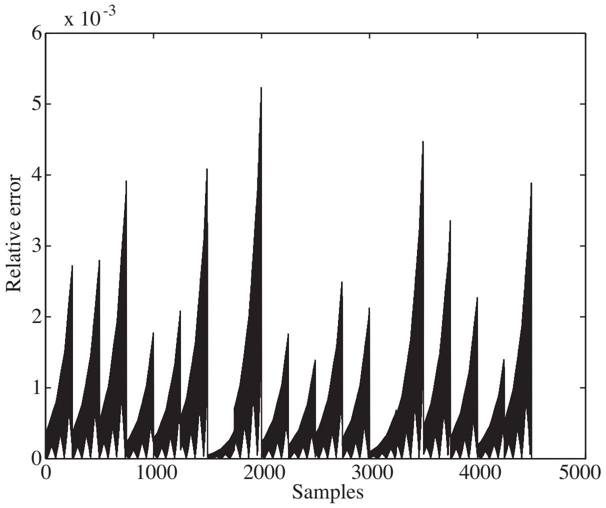

Figure 15. Relative errors yielded from filtered noisy ECG signal using the proposed 2-pass recursive filter scheme, section length $M = 250$, total length signal $L = 5000$. The maximum relative error is about 5.5×10^{-3}.

5.3. Discussion of results

We observe from the first experiment that both the input and output sequences obtained from our proposed realization seem the same and have identical shapes (Figures 8 and 9) because all 4 harmonics of the input are inside the filter's passband; therefore, linearity is observed and reached in the time domain.

The relative error is much smaller in the proposed realization, with a maximum of about 2.2×10^{-6} compared to that in [4] (about 4.9×10^{-6}) (Figures 10 and 11) .

Notice that in the general scheme shown in Figure 1, the causal part is realized without errors, which are reported and generated from the noncausal part. For this reason, from Figure 10, relative errors seem to be zero at the right side of each filtered section, whereas they appear to the left side.

In Figure 11, relative errors are symmetric compared to the axis separating 2 adjacent filtered sections because with the general scheme of Figure 2, relative errors are generated from both the causal and noncausal parts. Realizing the causal part first systematically eliminates a considerable part of the relative errors.

In the frequency domain, our method leads to –130.065 dB; however, Powell and Chau's method reaches about –120.969 dB. Linearity is thus improved in our proposed realization.

With the real noisy ECG signal experiments, it is clear from Figures 12-14 that the noncausal 2-pass IIR filters lead to the best filtering compared to the Parks-McClellan optimal FIR filter. Even with the same filter specifications and a high FIR filter order, the results are better in real-time processing with the noncausal 2-pass IIR filters class. When we evaluate the mean of the absolute error between the filtered noisy ECG signal compared to both types of filter and the original ECG signal, we get a mean of 0.0376 for our filtering and a mean of 0.0517 for the Parks-McClellan optimal FIR processing.

To determine the recommended section length M that should be used in segment-wise processing to minimize the effect of truncation caused by preventing future upcoming samples from exciting the 2-pass system, we repeat the first experiment described above with the same recursive filter specifications and the single frequency sinusoidal input of Eq. (25) with $f_0 = l/L$ and $l = 39$, as in [4]. We use several different section lengths. We then calculate the THD yielded from each experiment. Table 4 presents the results from our proposed scheme and those of Powell and Chau. Figure 17 illustrates the relationship between THD and section length M in the samples.

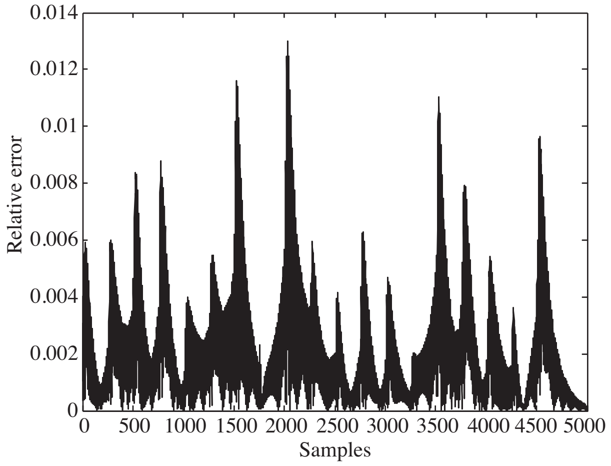

Figure 16. Relative errors yielded from filtered noisy ECG signal using Powell and Chau's 2-pass recursive filter scheme, section length $M = 250$, total length signal $L = 5000$. The maximum relative error is about 14×10^{-3}.

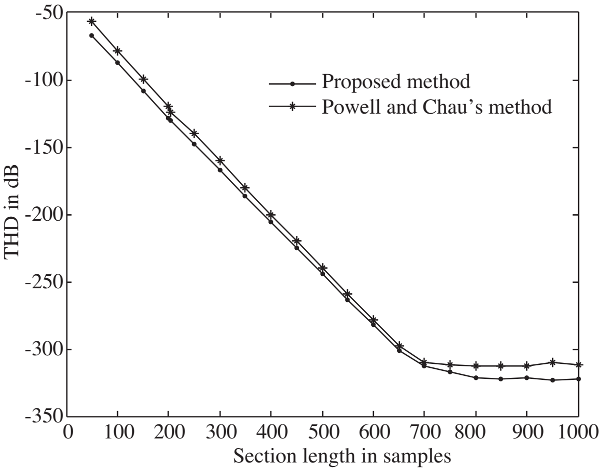

Figure 17. THD in dB versus section length in samples.

Table 4. THD in dB versus section length in samples.

Section length M in samples	THD from our proposed method	THD from Powell and Chau's method
50	–67.14	–56.84
100	–87.51	–78.79
150	–108.20	–99.60
200	–128.12	–119.99
205	–130.13	–123.96
250	–147.79	–140.16
300	–167.28	–160.21
350	–186.64	–180.18
400	–205.91	–200.08
450	–225.10	–219.91
500	–244.23	–239.63
550	–263.31	–259.20
600	–282.34	–278.55
650	–300.99	–297.40
700	–312.83	–310.52
750	–317.21	–311.91
800	–321.07	–312.87
850	–322.09	–312.30
900	–321.40	–312.77
950	–323.13	–309.99
1000	–322.26	–311.76

It seems that at about $M = 800$ samples, the effect of truncated upcoming samples will be negligible and the THD will reach the lowest levels in our proposed scheme (about –321 dB), which cannot be improved even if we use a greater section length. This lowest level of THD will be less (about –312 dB) in Powell and Chau's realization. This means that even when as large of a section length M is used as possible in order to get impulse responses of recursive filters decaying to very low level [14], the proposed method yields the best results in these extreme conditions as compared to those of Powell and Chau's method. Notice that the experiments are realized with a floating point representation in double precision (about 64 bits). Thus, with $N = 64$ bits of quantification, we get an SNR (dB) of about $6N + 1.76 = 385.76\,dB$.

6. Conclusion

The origin of systematic errors was investigated in the case of real-time noncausal recursive filters via forward/backward realization. An analytical condition was elaborated upon using a state variable representation approach. On the basis of this study, an implementable condition was proposed with a recursive technique of calculation. It was associated with a novel scheme based on a FIFO queue memory, allowing sample-by-sample processing and reduced delay and data storage memory as compared to the previous nonoverlapping method in [1]. An evaluation of the proposed scheme performances was done and a comparison with previous works was illustrated. Experimental results were given in order to show our contribution. It seems that our proposed realization leads to the lowest delay and THD.

Via experimental results, we have shown that the forward/backward realization with the causal part performed first allows systematically reduced relative errors compared to those of Powell and Chau's method, which performs the forward pass the second time.

Finally, an empirical section length M (about 800 samples) was recommended to get the best relative errors and THD. Results demonstrated that even with extreme conditions in which the section length leads approximately to the ideal response, our proposed method seems better compared to that of Powell and Chau.

Investigations related to the backward/forward scheme will be expanded in the future in order to improve this realization. The class of noncausal 2-pass IIR filters will be tested and approved in the near future with real applications in speech parameter estimation concerning multiband dysperiodicity analyses of disordered connected speech.

Acknowledgments

The authors wish to thank Prof Francis Grenez from Université Libre de Bruxelles for his helpful critiques and suggestions.

References

[1] E. Arias-de-Reyna, J.I. Acha, "A new method for designing efficient linear phase recursive filters", Digital Signal Processing, Vol. 14, pp. 1-17, 2004.

[2] R. Czarnach, "Recursive processing by noncausal digital filters", IEEE Transactions on Acoustics, Speech, & Signal Processing, Vol. 30, pp. 363-370, 1982.

[3] B. Djokie, M. Popovie, M. Lutovac, "A new improvement to the Powell and Chau linear phase IIR filters", IEEE Transactions on Signal Processing, Vol. 46, pp. 1685-1688, 1998.

[4] S.R. Powell, P.M. Chau, "A technique for realizing linear phase IIR filters", IEEE Transactions on Signal Processing, Vol. 39, pp. 2425-2435, 1991.

[5] J. Kormylo, V.K. Jain, "Two-pass recursive digital filter with zero phase shift", IEEE Transactions on Acoustics, Speech, & Signal Processing, Vol. 22, pp. 384-387, 1974.

[6] A. Kurosu, S. Miyase, S. Tomiyama, T. Takebe, "A technique to truncate IIR filter impulse response and its application to real-time implementation of linear phase IIR filters", IEEE Transactions on Signal Processing, Vol. 51, pp. 1284-1292, 2003.

[7] A.N. Willson, H.J. Orchard, "An improvement to the Powell and Chau linear phase IIR filters", IEEE Transactions on Signal Processing, Vol. 42, pp. 2842-2848, 1994.

[8] H. Farsi, "Speech pre-processing for pitch and pitch-cylce evolutions smoothing", Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 14, pp 225-240, 2006.

[9] A. Alpan, Y. Maryn, A. Kacha, F. Grenez, J. Schoentgen, "Multi-band dysperiodicity analyses of disordered connected speech", Speech Communication, Vol. 53, pp. 131- 141, 2011.

[10] A. Alpan, J. Schoentgen, F. Grenez, "Automatic multi-category classification based on connected speech of disordered voices", 9th International Conference on Advances in Quantitative Laryngology, Voice, and Speech Research, 2010.

[11] C. Mertens, F. Grenez, L. Crevier-Buchman, J. Schoentgen, "Reliable tracking based on speech sample salience of vocal cycle length perturbations", Proceedings of 11th Annual Conference of the International Speech Communication Association, pp. 2566-2569, 2010.

[12] S. Aydın, "Comparison of basic linear filters in extracting auditory evoked potentials", Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 16, pp 111-123, 2008.

[13] T. Raita-aho, T. Saramaki, O. Vainio, "A digital filter for ECG signal processing", IEEE Transactions on Instrumentation and Measurement, Vol. 43, pp. 644-649, 1994.

[14] T. Laakso, V. Valimaki, "Energy-based effective length of the impulse response of a recursive filter", IEEE Transactions on Instrumentation and Measurement, Vol. 48, pp. 7-17, 1999.

[15] W.R. Davis, E.C. Oh, A.M. Sule, P.D. Franzon, "Application exploration for 3-D integrated circuits: TCAM, FIFO, and FFT case studies", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 17, pp. 496-506, 2009.