The JM-Filter to Detect Specific Frequency in Monitored Signal

Marwan A. Jaber and Daniel Massicotte, *Senior Member, IEEE*

***Abstract*—The Discrete Fourier Transform (DFT) is a mathematical procedure that stands at the center of the processing inside a digital signal processor. It has been widely known and argued in relevant literature that the Fast Fourier Transform (FFT) is useless in detecting specific frequencies in a monitored signal of length *N* because most of the computed results are ignored. In this paper, we present an efficient FFT-based method to detect specific frequencies in a monitored signal, which will then be compared to the most frequently used method which is the recursive Goertzel algorithm that detects and analyses one selectable frequency component from a discrete signal. The proposed JM-Filter algorithm presents a reduction of iterations compared to the first and second order Goertzel algorithm by a factor of *r*, where *r* represents the radix of the JM-Filter. The obtained results are significant in terms of computational reduction and accuracy in fixed-point implementation. Gains of 15 dB and 19 dB in signal to quantization noise ratio (SQNR) were respectively observed for the proposed first and second order radix-8 JM-Filter in comparison to Goertzel algorithm.**

***Index Terms*—Discrete Fourier transform, Goertzel algorithm, one frequency detection, low complexity, fixed-point precision.**

I. Introduction

Digital signal processing (DSP) is an engineering field that continues to extend its theoretical foundations and practical implications in the modern world [1], [2]. From the fulfillment of day-to-day needs, such as personal communications, to sophisticated systems for biomedical and tactical applications, DSP has a strong and ever-increasing participation in the areas of work that are revolutionizing our society. DSP is the study that deals with the representation of signals as ordered sequences of numbers. DSP also deals with the fact on how to process those ordered sequences. There are many reasons for typical DSP such as estimating signal parameters, reduce as much as possible and eliminate unwanted interference, and transform a signal into a form that is more informative.

Signal monitoring is an expanding domain that deals with the detection of any abrupt changes in a specific known frequency, such as fault detection machine, or to scan a pre-selected set of frequencies [1], the recognition of the dual-tone multi-frequency (DTMF) signaling [3], [4], as in radio-frequency identification (RFID) tags [5], spectrum analyzer based on sliding transforms [6], [7], in wireless communication with the orthogonal frequency division multiplex (OFDM) [8], channelizer [9], and modulated filter banks [10] where low complexity, low power consumption and high throughput are required, in the detection of coding regions in large DNA sequences [11], [12] and [13], and other domains such as the detection of any abrupt changes in a specific known frequency, deployed in resonance frequency estimation to tune notch filter in grid converter [14], in phase lock loop (PLL) [15], [16], in induction motor fault detection by space vector angular fluctuation [17], in real-time (remote/in-situ) detection and identification of physical properties and trace chemical/bio substances in industrial and other environments and cracks detection of structural surface [18]. More recently, many other applications on power electronics to monitoring component and smart grid power quality [19]–[21].

Input/output pruning FFTs are efficient fast Fourier transform (FFT), where the efficiency can be increased by removing operations on consecutive input values which are zero [22], [23]. Furthermore, output pruning FFT is a method used to compute a discrete Fourier transform (DFT) where only a subset of the consecutive outputs is needed. Input/output pruning FFTs will be excluded from our performance comparison study since the desired k^{th} frequency is obtained by computing the first *k* outputs [22]–[26]. On the other hand, sparse FFTs are also efficient tools to compute the DFT where the input signal is *k*-sparse which means the input signal has *k* nonzero entries [27]–[29]. The most well-known technique to monitor and control a specific or single frequency is the Goertzel algorithm [30]. Goertzel algorithm is a powerful tool deployed in tone detection and frequency response measurements unlike the FFT, the pruning FFT and Sparse FFTs that are performed on blocks of samples. For instance, it has been proven that the Goertzel algorithm is more efficient than the FFT in DTMF detection [2]. In other words, we can state that the Goertzel Algorithm, an efficient method (in terms of multiplications) for computing output sequence for a given frequency bin in which *N* iterations is needed to compute each frequency of the input sequence of length *N*.

More recently, recursive discrete Fourier transform (RDFT) were proposed as recursive algorithms that can reduce the computational complexity in circuit implementation compared to

Manuscript received September 20, 2019; revised May 16, 2020 and December 2, 2020; accepted January 10, 2021. Date of publication January 25, 2021; date of current version March 3, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shogo Muramatsu. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by the Canadian Foundation for Innovation (CFI), and in part by the CMC Microsystems. Laboratoire des signaux et systèmes intégrés (www.uqtr.ca/lssi) and Chaire de recherche sur les signaux et l'intelligence des systèmes haute performance. *(Corresponding author: Daniel Massicotte.)*

The authors are with the Electrical and Computer Engineering Department, Université du Québec à Trois-Rivières, Trois-Rivières, QC G9A 5H7, Canada (e-mail: marwan.jaber@uqtr.ca; daniel.massicotte@uqtr.ca).

Digital Object Identifier 10.1109/TSP.2021.3053509

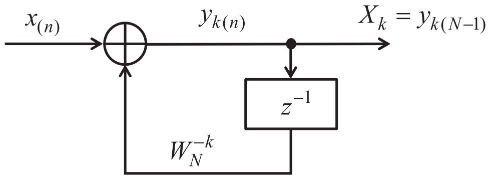

Fig. 1. The first order Goertzel algorithm.

Goertzel algorithm [31]–[35] which will be targeted for the computational complexity comparison.

Based on the recursive one frequency algorithm proposed in [36], this paper will propose a novel recursive algorithm JM-Filter[1] (Jaber-Massicotte Filter) that will reduce the number of iterations by a factor of *r* that will be deployed in the genomic DNA sequences in which the radix-2 JM-Filter detects the period-3 pattern in a coded Gene sequence [13].

This paper is organized as follows: Section II describes briefly the first and second order Goertzel algorithm. Section III details the proposed first and second order JM-Filter. Section IV details the reduced complexity of the proposed method. Section V draws on the performance results in terms of arithmetic complexity and fixed-point accuracy. Section VI elaborates the experimental results applied to genomic sequences and finally a conclusion is summarized in Section VII.

II. The First and Second Order Goertzel Algorithm

The derivation of the first order Goertzel algorithm, which is developed in [1] and [30] begins by noting that the DFT can be formulated in terms of a convolution, in fact the DFT of signal $x_{(k)}$ is defined as follows:

$$\begin{aligned} X_{(k)} &= \sum_{n=0}^{N-1} x_{(n)} W_N^{nk}, W_N^k = e^{-j\frac{2\pi k}{N}} \\ &= \sum_{n=0}^{N-1} x_{(n)} W_N^{-k(N-n)}, W_N^{-kN} = 1 \\ &= x_{(n)} * h_{(n)}\big|_{n=N}, \end{aligned} \quad (1)$$

where *N* is the length of the input data sequence $x_{(n)}$, $W_N = e^{-j2\pi/N}$, $j = \sqrt{-1}$, and $k = 0, 1, 2, \ldots, N-1$, * represents the convolution product of signal $x_{(n)}$ through a linear time invariant (LTI) filter with impulse response $h_{(n)} = W^{-nk} x_{(n)}$ and evaluating the result, $y_{k(n)}$, at $n = N$.

According to the same references, the filtering operation of first order Goertzel algorithm with the associated flow graph is depicted in Fig. 1, where we can write the recurrent equations as

$$y_{k(n)} = W_N^{-k} y_{k(n-1)} + x_{(n)}, \quad (2)$$

where $y_{k(-1)} = 0$.

The filter's output for the k^{th} frequency is

$$X_k = y_{k(N-1)}. \quad (3)$$

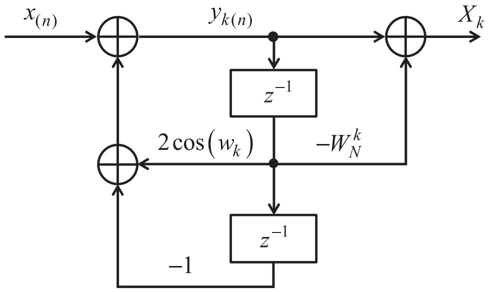

Fig. 2. The second order Goertzel algorithm.

The transfer function *H*(*z*) of the equivalent filter could be developed as

$$\begin{aligned} H(z) &= \frac{1}{1 - W_N^{-k} z^{-1}} = \frac{1 - W_N^{-k} z^{-1}}{1 - W_N^{-k} z^{-1}} \times \frac{1}{1 - W_N^{-k} z^{-1}} \\ &= \frac{1 - W_N^{-k} z^{-1}}{1 - 2\cos(2\pi k/n) z^{-1} + z^{-2}} \end{aligned} \quad (4)$$

where the second order Goertzel algorithm is obtained and the filtering operation with the associated flow graph is depicted in Fig. 2. The recurrent equations are

$$y_{k(n)} = 2\cos(2\pi k/N) y_{k(n-1)} - y_{k(n-2)} + x_{(n)}, \quad (5)$$

where $y_{k(-2)} = y_{k(-1)} = 0, w_k = 2\pi k/N$ and the k^{th} frequency output is expressed as:

$$X_k = y_{k(N-1)} - W_N^k y_{k(N-2)}, \quad (6)$$

In case of complex-valued input sequences, the computational complexity of the first order Goertzel algorithm based on real additions (⊕) and real multiplication (⊗) is

$$4N\text{real} \otimes \text{ and } 4N\text{real} \oplus \quad (7)$$

and according to Fig. 2, the computational cost of the second order Goertzel algorithm is

$$2N + 2\,\text{real} \otimes \text{ and } 4N - 2N\,\text{real} \oplus \quad (8)$$

which, compared to the DFT equation, gives a reduction of almost a factor of two in the number of real multiplications. This cost is halved again if the data is real-valued. Furthermore, the first order needs more resources than second order to execute the complex input data due to the complex value twiddle factors in the feedback that should multiply the previous data, (2). Beraldin and al. [37] showed an interesting overflow analysis in fixed-point implementations for the first and second order Goertzel algorithm. In fixed-point implementation, it was concluded that the first-order filter achieves better accuracy than the second-order filter. Thus, the first-order version is more interesting in terms of accuracy than the second-order version but lack in practicality in terms of implementation complexity.

III. The First and Second Order JM-Filter

The one iteration Decimation in Time (DIT) FFT algorithm expressed as [38]–[39]:

$$X_{(qV+v)} = \sum_{p=0}^{V-1} W_N^{[\![p(qV+v)]\!]_N} \sum_{m=0}^{r-1} x_{(mV+p)} W_N^{[\![mvV]\!]_N}, \quad (9)$$

[1] "Filter Configured to Detect Specific Frequencies of a Monitored Signal," US Patent Application No 62/677,587, 2018.

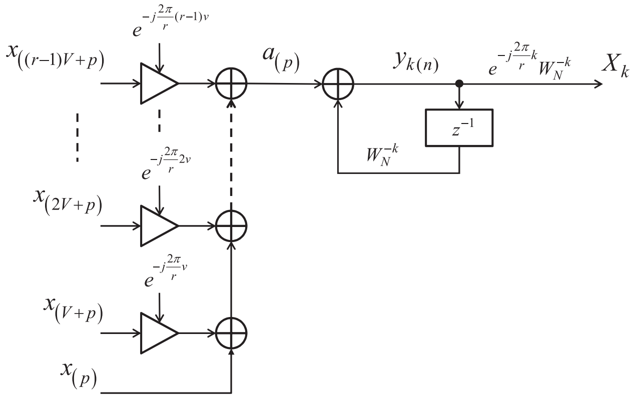

Fig. 3. The radix-*r* first order JM-Filter.

where $[\![x]\!]_N$ represents the operation x modulo N, $v = 0, 1, \ldots, V-1$, $q = 0, 1, \ldots, r-1$, and $V = N/r$.

To compute a specific frequency $X_{(k)}$ for a given k, the values of q and v must be known in advance. And so, by adopting the following notation [39]:

$$\begin{cases} 0 \le k < V \; q = 0 \text{ and } v = k \\ V \le k < 2V \; q = 1 \text{ and } v = k - V \\ \vdots \\ (r-1)V \le k < N \; q = (r-1) \text{ and } v = k - (r-1)V \end{cases} \tag{10}$$

By defining the second part of the (9) as follow:

$$a_{(p)} = \sum_{m=0}^{r-1} x_{(mV+p)} W_N^{[\![mvV]\!]_N} = \sum_{m=0}^{r-1} x_{(mV+p)} e^{-j\frac{2\pi\frac{N}{r}}{N}mv} = \sum_{m=0}^{r-1} x_{(mV+p)} e^{-j\frac{2\pi}{r}mv} \tag{11}$$

therefore, (9) can be expressed as:

$$X_k = \sum_{p=0}^{V-1} W_N^{pk} a_{(p)}, \tag{12}$$

where $k = qV + v$, and $W_N^k = e^{-j\frac{2\pi k}{N}} = e^{-j\frac{2\pi}{N}(qV+v)}$.

By examining (1) and (12) we can clearly notice that the filtering process of Goertzel algorithm is performed in the spatial domain $x_{(n)}$, meanwhile the proposed JM-Filter, the filtering process is accomplished in the radix-*r* partial DFT frequency domain defined by $a_{(p)}$ which will highlights the difference between the referenced models in Figs. 1 and 2 and the proposed models in figures 3 and 4. If the radix *r* is set to 2 in (12), we will obtain the radix-2 JM-Filter and so on.

As a result, the radix-*r* first order JM-Filter could be derived as:

$$y_{k(p)} = W_N^{-k} y_{k(p-1)} + a_{(p)}, \tag{13}$$

where $y_{k(-1)} = 0$ with $p = 0,1,\ldots,V-1$, and the k^{th} computed frequency is given by:

$$X_k = e^{-j\frac{2\pi}{r}k} W_N^{-k} y_{k(V-1)}. \tag{14}$$

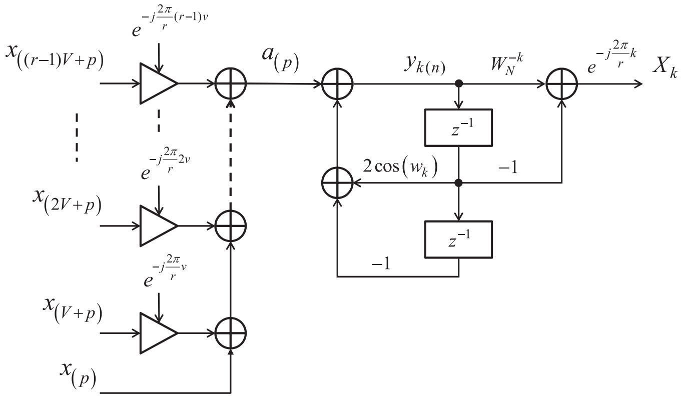

Fig. 4. The radix-*r* second order JM-Filter.

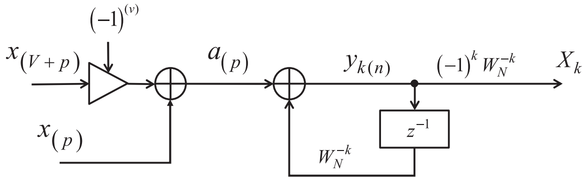

Fig. 5. The radix-2 first order JM-Filter.

Applying the recursive DFT formula as in [32] by using (13), the radix-*r* second-order JM-Filter will be

$$y_{k(p)} = 2\cos(2\pi k/N) y_{k(p-1)} - y_{k(p-2)} + a_{(p)}, \tag{15}$$

where $y_{k(-2)} = y_{k(-1)} = 0$, and from which the k^{th} computed frequency is

$$X_k = e^{-j\frac{2\pi}{r}k}\left(W_N^{-k} y_{k(V-1)} - y_{k(V-2)}\right). \tag{16}$$

The filtering operation for the first and second-order JM-Filter with the associated flow graphs are depicted in Figs. 3 and 4, respectively.

By examining Fig. 2 and Fig. 4, it is evident that the reduction in computation is mainly performed on the input sequence, where the length of the sequence $x_{(n)}$ in Fig. 2 is *N* meanwhile the length of the sequence $a_{(p)}$ in Figs 2 and 4 is reduced to *N*/*r*. The k^{th} computed frequency is obtained by mean of (14) and (16) where the complexity reduction of the sequence $a_{(p)}$ is elaborated in the next section for radices *r* = 2, 4 and 8.

IV. Complexity Reduction

By examining (12), further reductions in terms of complexity could be achieved for the radix-2 case, since

$$e^{-j\frac{2\pi}{r}mv} = e^{-j\pi mv} = (-1)^{mv}. \tag{17}$$

And so, based on (17), we can re-write (12) as

$$a_{(p)} = \sum_{m=0}^{1} x_{(Vm+p)}(-1)^{mv} = x_{(p)} + (-1)^v x_{(V+p)}, \tag{18}$$

from (13) the radix-2 JM-Filter first order filter would be

$$y_{k(p)} = W_N^{-k} y_{k(p-1)} + x_{(p)} + (-1)^v x_{(V+p)} \tag{19}$$

and the k^{th} computed frequency is given by the (14) (as shown in Fig. 5)

$$X_k = (-1)^k W_N^{-k} y_{k(V-1)}. \tag{20}$$

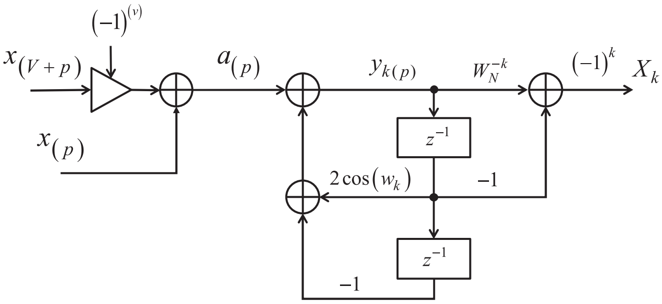

Fig. 6. The radix-2 second order JM-Filter.

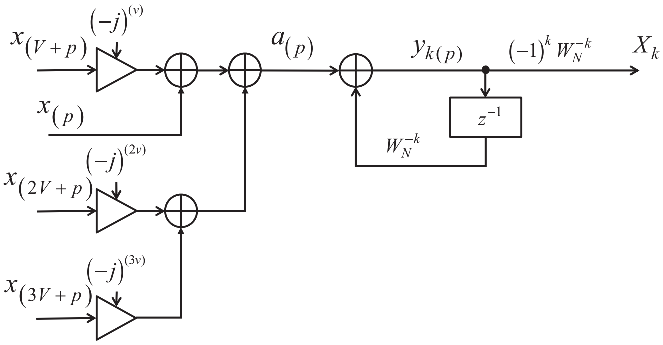

Fig. 7. The radix-4 first order JM-Filter.

Applying (17) and (18) on (15) and (16), the radix-2 second-order JM-Filter will be (Fig. 6)

$$y_{k(p)} = 2\cos(2\pi k/N) y_{k(p-1)} - y_{k(p-2)} + a_{(p)}, \tag{21}$$

where $y_{k(-2)} = y_{k(-1)} = 0$, and from which the k^{th} computed frequency is:

$$X_k = (-1)^k \left(W_N^{-k} y_{k(V-1)} \; - y_{k(V-2)} \right). \tag{22}$$

With the same reasoning as above, further reductions in terms of complexity for the radix-4 could be achieved

$$e^{-j\frac{2\pi}{4}mv} = e^{-j\frac{\pi}{2}mv} = (-j)^{mv}, \tag{23}$$

therefore, based on (23), we can re-write (13) as

$$\begin{aligned} a_{(p)} &= \sum_{m=0}^{3} x_{(Vm+p)} (-j)^{mv} \\ &= x_{(p)} + (-j)^v x_{(V+p)} + (-j)^{2v} x_{(2V+p)} + (-j)^{3v} x_{(3V+p)} \end{aligned} \tag{24}$$

the radix-4 first order JM-Filter (Fig. 7) would be

$$y_{k(p)} = W_N^{-k} y_{k(p-1)} + a_{(p)}, \tag{25}$$

and the k^{th} computed frequency is given by

$$X_k = (-j)^k W_N^{-k} y_{k(V-1)}. \tag{26}$$

The radix-4 second-order JM-Filter (Fig. 8) will be[2]

$$y_{k(p)} = 2\cos(2\pi k/N) y_{k(p-1)} - y_{k(p-2)} + a_{(p)}, \tag{27}$$

where $y_{k(-2)} = y_{k(-1)} = 0$, and from which the k^{th} computed frequency is

$$X_k = (-j)^k \left(W_N^{-k} y_{k(V-1)} \; - y_{k(V-2)} \right), \tag{28}$$

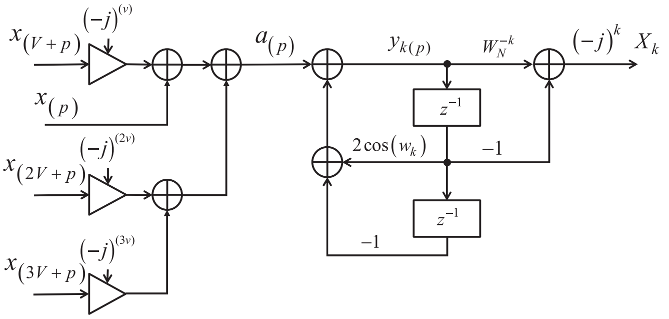

Fig. 8. The radix-4 second order JM-Filter.

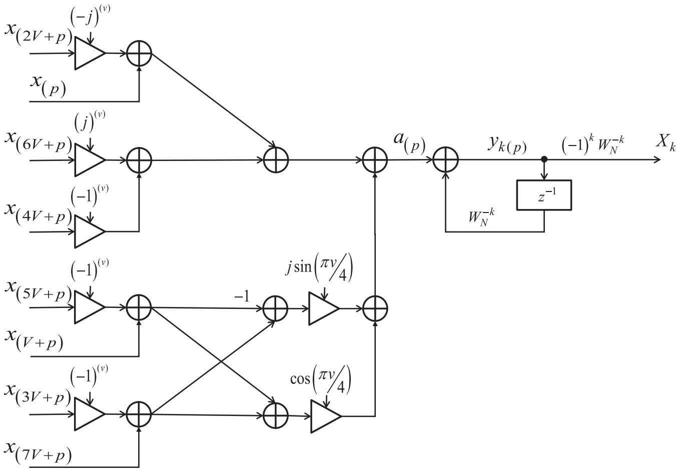

Fig. 9. The radix-8 first order JM-Filter.

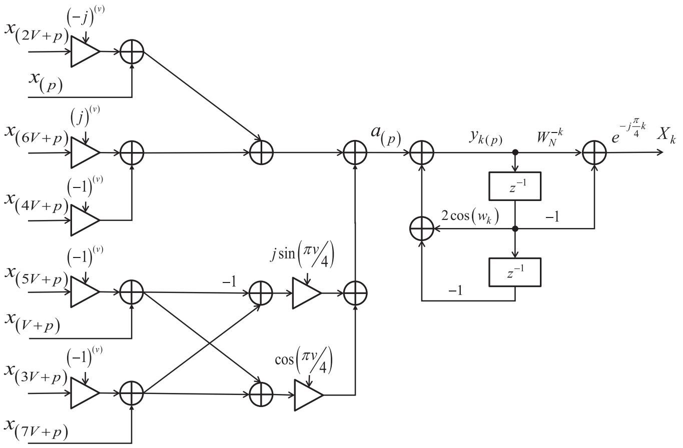

Fig. 10. The radix-8 second order JM-Filter.

with $k = 0,1,\ldots, N-1$ to compute all frequencies.

By examining (13) further reductions in terms of complexity could be achieved for the radix-8 case, therefore, we can re-write (13) for the radix-8 first order JM-Filter (Fig. 9) as:

$$y_{k(p)} = W_N^{-k} y_{k(p-1)} + a_{(p)}, \tag{29}$$

with

$$\begin{aligned} a_{(p)} &= \sum_{m=0}^{7} x_{(Vm+p)} e^{-j\frac{\pi}{4}mv} \\ &= x_{(p)} + (-j)^v x_{(2V+p)} + (-1)^v x_{(4V+p)} + (j)^v x_{(6V+p)} \\ &\quad + e^{-j\frac{\pi}{4}v} \left(x_{(V+p)} + (-1)^v x_{(5V+p)} \right) \end{aligned}$$

[2] The equations (13), (25), (29) and (15), (21), (27), (32) are repeated to ensure the complete uniformity of each version of the algorithm in which $a_{(p)}$ is computed according to the radices 2, 4 and 8.

TABLE I

COMPUTATIONAL COMPLEXITY IN TERMS OF REAL ARITHMETIC OPERATIONS FOR THE FIRST AND SECOND ORDER GOERTZEL ALGORITHM AND THE CITED RDFT COMPARED TO THE PROPOSED FIRST AND SECOND ORDER JM-FILTER (RADICES-2, 4 AND 8) FOR A COMPLEX VALUED SEQUENCE SIGNAL OF LENGTH N

Methods	**First Order**		**Second Order**	
	⊗	⊕	⊗	⊕
Goertzel [30]	$4N^2$	$4N^2$	$2N(N+3)$	$4N(N+2)$
[32]	–	–	$2N(N+1)$	$4N(N+1)$
[33]	–	–	$2N(N+3)$	$4N(N+2)$
[34]	–	–	$N^2/2-8$	$N^2+9.5N-22$
[31]	–	–	$(N+1)(N-2)$	$N(2N+7)-2$
[35]	–	–	$2N(N+1)$	$4N \times (N+1)$
JM-Filter radix-2	$2N^2$	$3N^2$	$N(N+2)$	$N(3N-2)$
JM-Filter radix-4	N^2	$5N^2/2$	$N(N/2+2)$	$N(5N/2-2)$
JM-Filter radix-8*	N^2	$13N^2/8$	$N(3N/4+2)$	$N(13N/8-2)$
JM-Filter radix-r	$N\left(\frac{4N}{r}+N_{a_{(p)}}^{MULT}\right)$	$N\left(\frac{4N}{r}+N_{a_{(p)}}^{ADD}\right)$	$N\left(\frac{2N}{r}+N_{a_{(p)}}^{MULT}+2\right)$	$N\left(\frac{4N}{r}+N_{a_{(p)}}^{ADD}-2\right)$

*Including the trivial multiplication $\pm\sqrt{2}/2$ and $\pm j\,\sqrt{2}/2$ to compute $a(p)$.

$$+ e^{j\frac{\pi}{4}v}\left(x_{(7V+p)} + (-1)^v x_{(3V+p)}\right) \tag{30}$$

and the k^{th} computed frequency is given by

$$X_k = e^{\left(-j\frac{\pi}{4}k\right)} W_N^{-k} y_{k(V-1)}. \tag{31}$$

The radix-8 second-order JM-Filter (Fig. 10) will be

$$y_{k(p)} = 2\cos(2\pi k/N) y_{k(p-1)} - y_{k(p-2)} + a_{(p)}, \tag{32}$$

where $y_{k(-2)} = y_{k(-1)} = 0$, and the k^{th} computed frequency is

$$X_k = e^{\left(-j\frac{\pi}{4}k\right)}\left(W_N^{-k} y_{k(V-1)} - y_{k(V-2)}\right), \tag{33}$$

with $k = 0, 1, \ldots, N-1$.

V. PERFORMANCE RESULTS – COMPLEXITY AND ACCURACY

Goertzel and JM-Filter are dedicated to compute an arbitrary specific frequency (a single frequency) and not a subset of consecutive K frequencies as usually performed by input/output pruning FFT methods [22]–[24]. According to [22] and [23] the only frequency that could be detected or monitored is the first one $X_{(0)}$ ($k = 0$) and that is why input/output pruning FFTs will be excluded from our performance comparison study due to the increasing complexity associated with the computation of the desired k^{th} frequency that is obtained by computing the first k outputs.

The performance evaluation results are based on real additions (⊕) and real multiplication (⊗) for the execution of the Goertzel and the proposed JM-Filter algorithm for different radix-r. In term of accuracy, the algorithms are executed in fixed-point that is defined by the Signal to Quantization Noise Ratio (SQNR).

A. Complexity – Number of Real Arithmetic Operations

As detailed in Section II, the computational complexity of the first order Goertzel algorithm in the case of complex-valued input sequences is $4N$ real ⊗ and $4N$ real ⊕, and from Fig. 2, the computational cost of the second order Goertzel algorithm is thus [30]

$2\text{N} + 2$ real ⊗ and $4\text{N} - 2$ real ⊕, which gives a reduction of almost a factor of two in the number of real multiplications and if the data is real-valued, this cost is almost halved again.

In general, for the radix-r case, the computational complexity of the first and second order radix-r JM-Filter are respectively

$$4N/r + N_{a_{(p)}}^{MULT}\text{real} \otimes \text{ and } 4N/r + N_{a_{(p)}}^{ADD}\text{real} \oplus,$$

$$2N/r + 2 + N_{a_{(p)}}^{MULT}\text{real} \otimes \text{ and } 4N/r - -2 + N_{a_{(p)}}^{ADD}\text{real} \oplus,$$

where $N_{a_{(p)}}^{ADD}$and $N_{a_{(p)}}^{MULT}$ are the total number of the required operations required to compute $a_{(p)}$. As a result, and according to Figs. 5 and 6, the computational complexity of the first and second order radix-2 JM-Filter, including $N_{a_{(p)}}^{MULT}$ and $N_{a_{(p)}}^{ADD}$, is respectively

$$2N\,\text{real} \otimes \text{ and } 3N\,\text{real} \oplus,$$

$$N + 2\,\text{real} \otimes \text{ and } 3N - 2\,\text{real} \oplus,$$

which give us a reduction in the multiplications' computational cost by a factor of 2, where $N_{a_{(p)}}^{MULT} = 0$ and we need $3N$ real additions compared to $4N$ real additions needed by Goertzel algorithm as shown in Table I.

According to Figs. 7 and 8, the computational complexity of the first and second order radix-4 JM-Filter is respectively

$$N\,\text{real} \otimes \text{ and } 5N/2\,\text{real} \oplus,$$

$$N/2 + 2\,\text{real} \otimes \text{ and } 5N/2 - 2\,\text{real} \oplus,$$

which give us a reduction in the multiplications' computational cost by a factor of 4, where $N_{a_{(p)}}^{MULT} = 0$, and we need 5N/2 real additions compared to $4N-2$ real additions required for Goertzel algorithm.

Based on Figs. 9 and 10, the computational complexity of the first and second order radix-8 JM-Filter is respectively:

$$N\,\text{real} \otimes \text{ and } 13N/8\,\text{real} \oplus,$$

$$3N/4 + 2\,\text{real} \otimes \text{ and } 13N/8 - 2\,\text{real} \oplus,$$

where $N_{a_{(p)}}^{MULT} = \frac{N}{2}$and where $N_{a_{(p)}}^{ADD} = \frac{9N}{8}$.

The Table I summarize the complexity operations for the proposed JM-Filter radices-2, 4, 8 and r, Goertzel algorithm and the previously published RDFT algorithms [23]–[27].

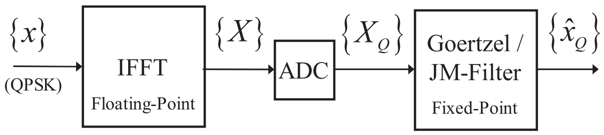

Fig. 11. Fixed-point simulation with QPSK signals.

B. Accuracy – SQNR Evaluations

Goertzel algorithm is the most powerful algorithm used in the detection of specific frequencies in monitored signal's applications. Their implementation is very attractive in fixed point due to the reduction in cost compared to the floating-point implementation. In digital signal processing, signal-to-quantization noise ratio (SQNR) is a measure of signal strength relative to the background noise.

The model illustrated in Fig. 11 represents the simplified transmitter-receiver commonly used in orthogonal frequency-division multiplexing (OFDM) communication systems with a quadrature phase-shift keying (QPSK) modulation. The first block represents the inverse FFT (IFFT) at the transmitter, the intermediate block is the analog to digital converter (ADC) with finite Q-bit word length that is connected to the FFT block in order to reconstruct the input fixed-point signal. Several fixed-point simulations were conducted to compute the SQNR in term of the input/output data (called variable data) bit's word-length (bit width) and the twiddle factor (called coefficient) [40]. The SQNRs' simulations, are calculated based on this equation

$$\mathrm{SQNR} = 10\log_{10}\left(\frac{\|\{x\}\|}{\|\{x\} - \{\hat{x}_Q\}\|}\right) dB, \tag{34}$$

where $\|x\|$ define the norm-L$_2$ function of the signal x, x and x_Q represent the signal x in floating and fixed-point respectively. The norm is calculated on the complex valued signal sequence of length N measured in decibels (dB). The higher the ratio, the less obtrusive the background noise is. Two major concerns in the computation of the Goertzel algorithm are the speed and high SQNR. The fixed-point implementation generates noise sources due to the bit representation in hardware implementation that propagate through the system that can modifies the overall system accuracy.

According to [41], the first order Goertzel algorithm performs better than the second order in fixed-point implementation due to the presence of two recursions in the second order, in which both methods have the same number of iterations N. To increase the SQNR for real-valued input sequences, [42] proposed to apply a scaling factor O(1/N) on the input data for the first order and a scaling factor O(1/N^2) for the second order. With regard a complex-valued input sequence, [41] proposed a scaling factor $\pi/4N$ to the input sequence $x_{(n)}$ for the first order filter. Applying a scaling factor on the recursive filter will assure its stability by avoiding overflow. Therefore, the scaling factor has an important role in the accuracy and stability of the filter.

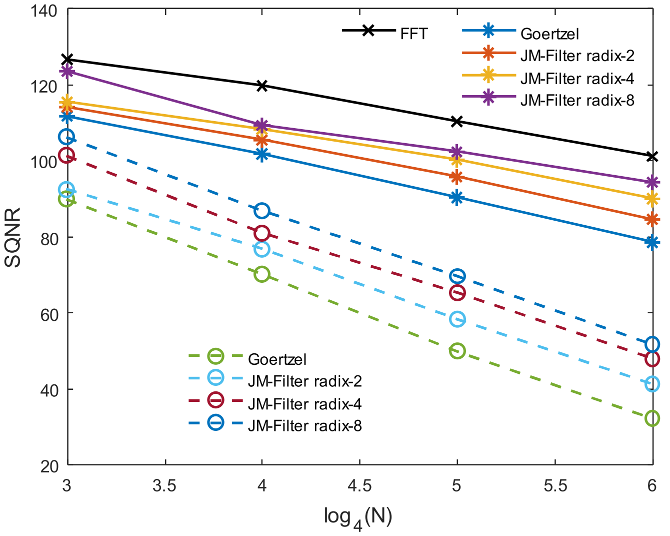

Fig. 12. Accuracy comparison between the first order (solid line) and second order (dashed line) Goertzel algorithm versus the proposed JM-Filter for the radices 2, 4 and 8 where the FFT is added as reference. All N output frequencies are computed for all methods.

In our performance comparison between the proposed JM-Filter versus Goertzel algorithm, the scaling factor is maximized for all possible scenarios to avoid overflow and to maximize the accuracy (SQNR). The analysis is performed for different sequence length which was applied on the first/second order Goertzel and the JM-Filter for the radices-2, 4 and 8. In our fixed-point model, the integer part of the input sequence word-length $\{X_k\}$ is fixed to 2-bit which will include the sign bit. The rest of the bit word-length is given to the fractional part. On the other hand, the integer part of the coefficients is fixed to 1-bit which is the sign bit and the rest is dedicated for the fractional part.

Fig. 12 presents the fixed-point evaluation results in which the variable data and coefficients word-length are fixed to 24-bit. The sizes of the simulated input sequence are 64, 256, 1024 and 4096 (mostly used in the OFDM systems). Table II shows the SQNR and scaling factor. The scaling factors was adjusted to maximize the accuracy. Table III shows the complexity evaluation for Goertzel and JM-Filter methods based on real value operations to compute one specific output frequency considering N complex-valued inputs.

Our proposed method reveals substantial gains compared to the first/second order Goertzel algorithm. The improvement in accuracy results can be explained by the following points: (i) with the increasing radix of the JM-Filter, the total amount of iteration will decrease that is associated with a reduction in the arithmetical operations, and (ii) by decreasing the total amount of iteration and to avoid overflow, the scaling factor is an asset in improving the accuracy of our proposed method.

The fixed-point results of the radix-2 FFT have been added as reference. We have not applied a scaling factor at each stage of the FFT process as it was proposed in [40], [41]. We only applied the adjustment of scaling factor at the input sequence in the same manner for Goertzel and the proposed filters. The loss of accuracy comes from the recursive operations and the presence of multiplication operations in the feedback of filters.

For an input sequence of length 4096, the first order radix-8 JM-Filter manifests a gain of 15.7 dB compared to the first order Goertzel algorithm, meanwhile the second order

TABLE II
ACCURACY (SQNR) AND SCALING FACTOR MAXIMIZING THE SQNR FOR REFERENCE AND PROPOSED METHODS

METHODS		**Sequence length (*N*)**									
		64		256		288		1024		4096	
		Scaling Factor	SQNR (dB)	Scaling Factor	SQNR (dB)	Scaling Factor	SQNR (dB)	Scaling Factor	SQNR (dB)	Scaling Factor	SQNR (dB)
FFT		2	126.6	2	119.7	–	–	1	110.2	1	104.0
First Order											
Goertzel		1/2	111.7	1/2	101.7	2	100.9	1/2	90.2	1/2	78.4
Proposed JM-Filter	Radix-2	1/2	114.0	1/2	105.4	2	105.2	1/2	95.7	1/2	84.4
	Radix-4	1/2	115.3	1/2	108.1	2	108.0	1/2	100.1	1/2	89.9
	Radix-8*	1	123.4	1/2	109.2	–	–	1/2	102.3	1/2	94.1
Second Order											
Goertzel		1/40	89.7	1/192	70.1	144	71.0	1/1024	49.8	1/3584	32.2
Proposed JM-Filter	Radix-2	1/36	92.4	1/106	76.7	162	72.5	1/426	58.1	1/1536	41.0
	Radix-4	1/16	101.2	1/70	80.9	64	81.2	1/227	65.1	1/904	47.7
	Radix-8*	1/8	106.1	1/33	86.8	–	–	1/128	69.5	1/512	51.5

TABLE III
COMPLEXITY EVALUATION FOR GOERTZEL AND PROPOSED METHODS BASED ON REAL VALUE OPERATIONS TO DETECT ONE OUTPUT FREQUENCY

METHODS		**SEQUENCE LENGTH (*N*)**											
		64			256			288			1 024		
		Number of iter.	$\otimes$	$\oplus$	Number of iter.	$\otimes$	$\oplus$	Number of iter.	$\otimes$	$\oplus$	Number of iter.	$\otimes$	$\oplus$
FFT Radix-2		6	768	1 152	8	4 096	6 144	10	20 480	30 720	10	20 480	30 720
First Order													
Goertzel		64	256	256	256	1 024	1 024	288	1 152	1 152	1 024	4 096	4 096
Proposed JM-Filter	Radix-2	32	128	192	128	128	768	144	576	864	512	2 048	3 072
	Radix-4	16	64	160	64	256	640	72	288	720	256	1 024	2 560
	Radix-8*	8	64	104	32	256	416	36	288	468	128	1 024	1 664
Second Order													
Goertzel		64	131	258	256	515	1 026	288	579	1 154	1024	2 051	4 098
Proposed JM-Filter	Radix-2	32	66	190	128	258	766	144	290	862	512	1 026	3 070
	Radix-4	16	34	158	64	130	638	72	146	718	256	514	2 558
	Radix-8*	8	50	102	32	194	414	36	218	466	128	770	1 662

*Including the trivial multiplication $\pm\sqrt{2}/2$ and $\pm j\sqrt{2}/2$ to compute $a(p)$.

radix-8 JM-Filter manifests a gain of 19.3 dB in comparison to the second order Goertzel algorithm. Significant and interesting gain is observed for the second order, but the first order stay the best avenue in fixed-point.

VI. EXPERIMENTAL RESULTS IN GENOMIC SIGNAL PROCESSING

DNA processing or Genomic signal processing is an expanding domain in which the four nucleotides A, C, G, and T need to be converted into numerical value that should be processed. The coding region of a gene exhibits a period-3 pattern, that is translated into large peaks in the spectral domain that will occur at $k = N/3$ of the DFT coefficients [43], [44]. In this subsection we will be deploying the radix-2 second order JM-Filter for the period-3 pattern detection as shown in Fig. 13 in which the FFT, Goertzel and the radix-2 JM-Filter have exactly given the same prediction of the period-3 behavior [13].

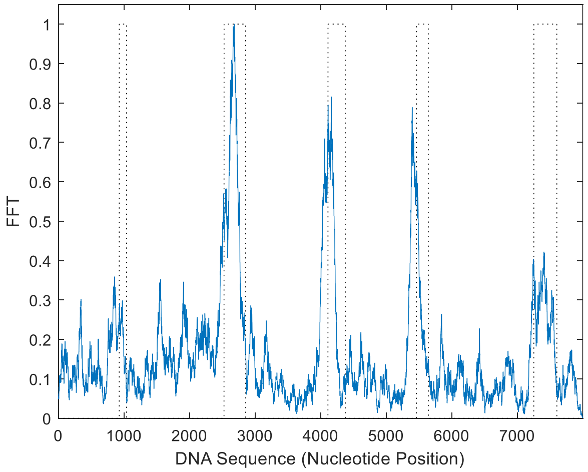

Fig. 13. Detection of period-3 behavior by deploying three methods: the FFT, Goertzel and the proposed JM-Filter where all three methods have manifested a matched result. The dashed line describes the ideal gene positions.

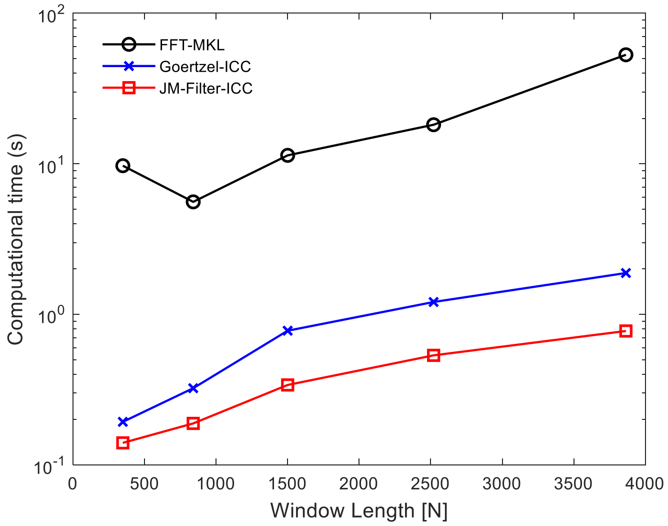

Fig. 14. Computational time for proposed and Goertzel methods, both for second order algorithm, and the FFT for different window sizes *N*.

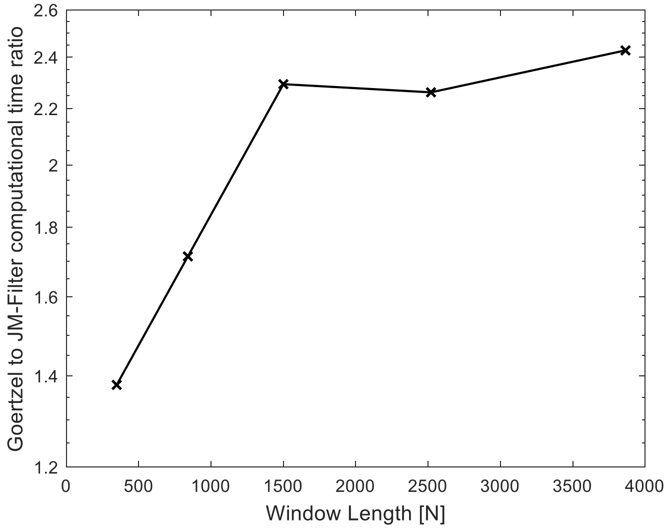

Fig. 15. Computational time ratio between our proposed method and Goertzel for different window sizes *N*.

To evaluate the complexity reduction of the proposed JM-Filter, the computational time for the period-3 pattern was performed on the genomic sequence NC_003281 (chromosome III of *C.elegans*) of length $M = 13\ 783\ 801$ [45] in which the applied window sizes *N* are 348, 840, 1500, 2520 and 3864. The detection of the exons' period 3 behavior is accomplished by sliding the window on the sequence of length *M*. Fig. 14 shows the computational time and the speedup is shown in Fig. 15 in which different window sizes were applied. The second order Goertzel and JM-Filter are coded in C language and compiled on the Intel C 64' compiler for applications running on Intel 64 (ICC) version 19.0.4.243 build 20190416. For a comparative study, the Intel' optimized MKL FFT function is also added. The technical specification of the used platform is: two Intel processors Xeon CPU E5-2620 v3 with 12 cores (total of 24 cores) at 2.40 GHz and 258 GB of RAM memory. Fig. 15 reveals a speedup of 2.35 for a different window size more than 1500 that is due to the complexity reduction by factor of 2. The gain is greater than of 2 is associated to the reduction of memory access' time caused by reducing the amount of iterations to *N*/2.

VII. Conclusion

This paper has presented a more efficient algorithm for computing a specific frequency compared to the well-known Goertzel algorithm, in which we have proven a reduction in the arithmetic computational load by a factor of *r*, where *r* is the radix of the introduced JM-Filter, and a significant gain in accuracy in term of SQNR. The accuracy gain is principally because the recursive equation of our proposed algorithm compared to Goertzel has been reduced from *N* to *N/r*.

The future work will be exploring the overflow analysis for our proposed JM-Filter for radices-2, 4 and 8 and the implementation on a pipelined specific-integrated circuit and on FPGA structure.

Acknowledgment

The authors would also like to thank Philippe Massicotte and Loïc Bachelot for their help in the Xeon CPU setups.

References

[1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, pp. 542–544, 2007.

[2] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, 3rd ed., Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.

[3] C. J. Chen, "Modified Goertzel algorithm in DTMF detection using the TMS320C80," Tech. Rep., Texas Instruments SPRA006, Jun. 1996.

[4] D. Schwingshackl, T. Mayerdorfer, and D. StrauBnigg, "Universal tone detection based on the Goertzel algorithm," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, 2006, pp. 410–413.

[5] B. Corbett and M. Woodman, "Radio frequency identification tags," U.S. Patent 7338497, Jun. 17, 2008.

[6] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 74–80, Mar. 2003.

[7] C. Park, "Fast, accurate, and guaranteed stable sliding discrete Fourier transform [sp tips&tricks]," *IEEE Signal Process. Mag.*, vol. 32, no. 4, pp. 145–156, Jul. 2015.

[8] M. Ding, Z. Shen, and B. L. Evans, "An achievable performance upper bound for discrete multitone equalization," in *Proc. IEEE Global Telecommun. Conf.*, 2004, pp. 2297–2301.

[9] B. H. Tietche, O. Romain, and B. Denby, "Sparse channelizer for FPGA-based simultaneous multichannel DRM30 receiver," *IEEE Trans. Consum. Electron.*, vol. 61, no. 2, pp. 151–159, May 2015.

[10] J. Alhava and M. Renfors, "Recursive algorithms for modulated and extended lapped transforms," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 61, no. 1, pp. 191–201, Jan. 2014.

[11] A. R. Fuentes *et al.*, "Detection of coding regions in large DNA sequences using the short time Fourier transform with reduced computational load," in *Proc. Prog. Pattern Recognit., Image Anal. Appl.*, 2006, pp. 902–906.

[12] H. T. Bui, "Pipelined FPGA design of the Goertzel algorithm for exon prediction," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2012, pp. 572–575.

[13] D. Massicotte, M. A. Jaber, M. Massicotte, and P. Massicotte, "Low complexity frequency monitoring filter for fast exon prediction sequence analysis," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Ottawa, ON, Canada, 2019, pp. 15.

[14] R. Peña-Alzola, M. Liserre, F. Blaabjerg, M. Ordonez, and T. Kerekes, "A Self-commissioning notch filter for active damping in a three-phase LCL -filter-based grid-tie converter," *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6754–6761, Dec. 2014.

[15] I. Carugati, P. Donato, S. Maestri, D. Carrica, and M. Benedetti, "Frequency adaptive PLL for polluted single-phase grids," *IEEE Trans. Power Electron.*, vol. 27, no. 5, pp. 2396–2404, May 2012.

[16] K. M. Singh, "Simultaneous estimation of moving-vibration parameters by sliding Goertzel algorithm in PLL technique," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 334–343, Feb. 2019.

[17] D. Kostic-Perovic, M. Arkan, and P. Unsworth, "Induction motor fault detection by space vector angular fluctuation," in *Proc. Conf. Rec. IEEE Ind. Appl. Conf. 35th IAS Annu. Meeting World Conf. Ind. Appl. Elect. Energy*, Rome, Italy, 2000, pp. 388–394.

[18] Z. Zhou, K. Zhang, J. Zhou, G. Sun, and J. Wang, "Application of laser ultrasonic technique for non-contact detection of structural surface-breaking cracks," *Opt. Laser Technol.*, vol. 73, pp. 173–178, Oct. 2015.

[19] X. Song, Z. Wang, S. Li, and J. Hu, "Sensorless speed estimation of an inverter-fed induction motor using the supply-side current," *IEEE Trans. Energy Convers.*, vol. 34, no. 3, pp. 1432–1441, Sep. 2019.

[20] J. Yu, Y. Xu, Y. Li, and Q. Liu, "An inductive hybrid UPQC for power quality management in premium-power-supply-required applications," *IEEE Access*, vol. 8, pp. 113342–113354, 2020.

[21] P. Sundararajan *et al.*, "Condition monitoring of DC-Link capacitors using Goertzel algorithm for failure precursor parameter and temperature estimation," *IEEE Trans. Power Electron.*, vol. 35, no. 6, pp. 6386–6396, Jun. 2020.

[22] M. Medina-Melendrez, M. Arias-Estrada, and A. Castro, "Input and/or output pruning of composite length FFTs using a DIF-DIT transform decomposition," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4124–4128, Oct. 2009.

[23] D. Castro-Palazuelos, M. Medina-Melendrez, D. Torres-Roman, and Y. V. Shkvarko, "Unified commutation-pruning technique for efficient computation of composite DFTs," in *Proc. EURASIP J. Adv. Signal Process.*, 2015, pp. 1–22.

[24] H. Krishnamoorthi, A. Spanias, and V. Berisha, "A frequency/detector pruning approach for loudness estimation," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 997–1000, Nov. 2009.

[25] D. Qin, J. Ren, and Y. Xu, "An efficient pruning algorithm for IFFT/FFT based on NC-OFDM in 5G," in *Proc. 2018 2nd Int. Conf. Inventive Commun. Comput. Technol.*, 2018, pp. 432–435.

[26] T. Ayhan, W. Dehaene, and M. Verhelst, "A 128∶2048/1536 point FFT hardware implementation with output pruning," in *Proc. Eur. Signal Process. Conf.*, 2014, pp. 266–270.

[27] S. Liu *et al.*, "Sparse discrete fractional Fourier transform and its applications," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6582–6595, Dec. 2014.

[28] D. Wei and Y. Li, "Generalized sampling expansion with multiple sampling rates for lowpass and bandpass signals in the fractional Fourier domain," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4861–4974, Sep. 2016.

[29] D. Wei and Y. Li, "Novel tridiagonal commuting matrices for types I, IV, V, VIII DCT and DST matrices," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 483–487, Feb. 2014.

[30] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," in *Proc. Amer. Math. Monthly*, 1958, pp. 34–35.

[31] S. Chin Lai *et al.*, "Low computational complexity, low power and low area design for the implementation of recursive DFT and IDFT algorithms," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 56, no. 12, pp. 921–925, Dec. 2009.

[32] L. D. Van and C. C. Yang, "High-speed area-efficient recursive DFT/IDFT architectures," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, pp. 357–360, May 2004.

[33] L. D. Van, C. T. Lin, and Y. C. Yu, "VLSI architecture for the low computation cycle and power-efficient recursive DFT/IDFT design," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E90-A, no. 8, pp. 1644–1652, Aug. 2007.

[34] P. K. Meher *et al.*, "Novel recursive solution for area-time efficient systolization of discrete Fourier transform," in *Proc. 2007 Int. Symp. Signals, Circuits Syst.*, 2007, pp. 1–4.

[35] S.-S. Lai, "Low-computation-cycle, power efficient and reconfigurable design of recursive DFT for portable digital radio mondial receiver," *IEEE Trans. Circuits Syst.—II: Exp. Briefs*, vol. 57, no. 8, pp. 647–651, Aug. 2010.

[36] M. Jaber and D. Massicotte, "The Radix-r one stage FFT kernel computation," in *Proc. Int. Conf. Acoustic, Speech, Signal Process.*, 2008, pp. 3585–3588.

[37] J. A. Beraldin and W. Steenaart, "Overflow analysis of a fixed-point implementation of the Goertzel algorithm," *IEEE Trans. Circuits Syst.*, vol. 3.6, no. 2, pp. 322–324, Feb. 1989.

[38] M. Jaber, "Low complexity and high performance of the Fourier Transform," Ph.D. Dissertation, Université du Québec à Trois-Rivières, 2012.

[39] M. A. Jaber and D. Massicotte, "Fast method to detect specific frequencies in monitored signal," in *Proc. Int. Symp. Commun., Control Signal Process.*, Limassol, Mar. 2010, pp. 1–5.

[40] W. Chang and T. Nguyen, "On the fixed-point accuracy analysis of FFT algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4673–4682, Oct. 2008.

[41] M. Medina-Melendrez, M. Arias-Estrada, and A. Castro, "Overflow analysis in the fixed-point implementation of the first-order Goertzel algorithm for complex-valued input sequences," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, 2009, pp. 620–623.

[42] M. Medina-Melendrez, M. Arias-Estrada, and A. Castro, "Using a scaling factor in O(1/N) for the fixed-point implementation of the second-order Goertzel filter," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 3218–3221.

[43] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A dynamic scaling FFT processor for DVB-T applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 2005–2013, 2004.

[44] A. Dimitris, "Genomic signal processing," *IEEE Signal Process. Mag.*, vol. 18, no. 4, pp. 8–20, 2001.

[45] The National Center for Biotechnology Information (NCBI), GenBank database, Caenorhabditis elegans chromosome III, NC_003281. [Online]. Available: https://www.ncbi.nlm.nih.gov/nuccore/NC_003281

Marwan A. Jaber received the B.Sc.A. and M.Sc.A. degrees in electronic engineering from Northrop University, Inglewood, CA, USA and the Ph.D. degree in electrical engineering from the Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada. He is currently the Founder and the Chief Technology Officer of Carbon's and Neutron's technology, Texas, USA. He has more than twelve years of progressive experience in digital signal processing, algorithms architectures and explorations, digital filtering, audio and speech processing, low-power and real-time systems, communication systems, image processing, parallel processing and massively parallel processing environment, blind equalization, time-frequency analysis, array signal processing, parallel computing, and project management. His detailed work includes experience with methods of signal processing including Fourier analysis (FFT, IFFT, DFT and Z-Transform), advanced digital and analog filtering, linear de-convolution, and buried signal extraction. His software work includes embedded-processor fast Fourier transform and interpolation software for the FMCW radar, Airborne CFAR radar, and offline FFT processing software for test signals downloaded from the FMCW radar and Airborne CFAR radar. Detailed work experience includes methods of signal processing in vibration analysis for machines fault detection. He is currently involved in a research project that will touch the lossless data compression and encryption.

Daniel Massicotte (Senior Member, IEEE) received the B.Sc.A. and M.Sc.A. degrees in electrical engineering and industrial electronics from the Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada, in 1987 and 1990, respectively, and the Ph.D. degree in electrical engineering from the École Polytechnique de Montréal, Montréal, QC, Canada, in 1995. In 1994, he joined the Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, where he is currently a Full Professor. He is the Founder of the Laboratory of Signal and Systems Integration. Since 2001, he has been the Founding President and the Chief Technology Officer with Axiocom Inc. From 2011 to 2018, he was the Head of the Industrial Electronic Research Group and from 2014 to 2020, the Head of the Electrical and Computer Engineering Department. He was the Head of the Research Chair in signals and intelligence of high performance systems. He has authored or coauthored more than 200 technical papers in international conferences and journals, as well as of several inventions. His research interests include advanced VLSI implementation, digital signal processing for wireless communications, measurement, and biomedical and control problems for linear or nonlinear dynamic complex systems. He was the recipient of the Douglas R. Colton Medal for Research Excellence Award by the Canadian Microelectronics Corporation, the PMC-Sierra High Speed Networking and Communication Award, and the Second place at the Complex Multimedia/Telecom IP Design Contest from Europractice. He has proposed several methods based on modern signal and biosignal processing such as machine or deep learning, transform domain, and metaheuristics. He was also the General Chair of the IEEE NEWCAS 2014 and a Guest Editor of the Springer *Analog Integrated Circuits and Signal Processing* for the special issues of NEWCAS 2013. He is also a Member of the Ordre des Ingénieurs du Québec, Groupe de Recherche en Électronique Industrielle, and Microsystems Strategic Alliance of Quebec.