International Journal of Electronics Communication and Computer Engineering
Volume 12, Issue 4, ISSN (Online): 2249–071X

Low Power Digital Filtering Using Adaptive Approximate Processing: IIR Filter Structures

Jeffrey Ludwig
University of California, Irvine, United States of America.
Corresponding author email id: jtludwig@uci.edu

Date of publication (dd/mm/yyyy): 25/07/2021

Abstract **– Techniques for reducing power consumption in digital circuits have become increasingly important because of the growing demand for portable multimedia devices. Digital filters, being ubiquitous in such devices, are a prime candidate for low power design. Algorithmic approaches to low power frequency selective digital filtering which is based on the concepts of adaptive approximate processing have been developed and formalized by introducing the class of approximate filtering algorithms in which the order of a digital filter is dynamically varied to provide time-varying stop band attenuation in proportion to the time-varying signal-to-noise ratio (SNR) of the input signal, while maintaining a fixed SNR at the filter output. Since power consumption in digital filter implementations is proportional to the order of the filter, dynamically varying the filter order is a strategy which may be used to conserve power. In this paper we introduce a class of approximate filter structures using IIR digital filter constituent elements. These filter structures are explored and shown to be an important element in the characterization of approximate filtering algorithms.**

Keywords **– Low Power Signal Processing, Adaptive Filtering, Approximate Signal Processing, Speech Signal Processing, Frequency-Selective Digital Filtering.**

I. INTRODUCTION

Living in the digital revolution of the information age, we are all accustomed to having the luxury of sophisticated communications and computation systems at our fingertips. It is a rarity to go through the day and not witness evidence of the explosive popularity of digital mobile devices such as iPhones and iWatches. The signal processing demands of such devices have increased dramatically in the last three decades as products continue to shrink in size and require increasing computational speed.

Adaptive filtering and energy efficiency has been a topic of great interest in recent years. For example, the interest in reducing the power consumption of digital filters used in edge computing and sensor networks is growing rapidly [1]. A fundamental tradeoff between power consumption and accuracy has been utilized to determine energy-optimum design parameters for deep in-memory architectures for efficient hardware realizations of machine learning algorithms [2]. Remarkably, in a recent paper the authors note that "optimization for power is one of the most important design objectives in modern digital signal processing (DSP) applications," and then demonstrate the efficacy of a hybrid energy-efficient methodology for digital finite duration impulse response (FIR) filters [3].

Approximate signal processing may be used for applications in which it is desirable to dynamically adjust the quality of signal processing results to the availability of resources, such as time, bandwidth, memory, and power. Recently, excellent research has been accomplished in the area of incremental refinement structures for approximate signal processing in the context of power-efficient approximate multiplication for applications in which exact computation is not necessary [4-8]. The design of an energy efficient digital IIR filter using approximate multiplication, with a similar objective to the IIR filters presented in this paper, is presented in [1].

Furthermore, approximate computing represents another recent innovation for minimizing power consumption via a variety of methods for reducing arithmetic activity [9-12].

Due to the nature of portability, ever-increasing processing demands are accompanied by definite constraints on power consumption since rechargeable batteries must be used. Consequently, the important task of designing low power, computationally powerful processors spurs great interest and activity in signal processing research, and most likely will continue to do so over the coming decades with the promise of artificial intelligence and incessant innovations in technology.

Digital filters represent a fundamental signal processing element which is found in all of the portable systems already mentioned and many others. Motivated by the growing demand for low power digital signal processing techniques for use in mobile devices, algorithmic approaches to low power frequency-selective digital filtering have been extensively developed [13-21]. It has been demonstrated that significant power savings may be achieved in digital filtering applications when the order of a digital filter is dynamically varied to provide time-varying stop band attenuation in proportion to the time-varying signal-to-noise ratio (SNR) of the input signal, while maintaining a fixed SNR at the filter output. In addition to providing the capability to dynamically conserve a limited resource such as battery power, this class of algorithms provides the foundation for the development of other algorithms which have the ability to intelligently respond to dynamic changes in the availability of other resources such as processor cycles in a shared environment.

Computational efficiency is of paramount importance for a broad class of signal processing algorithms designed to operate in an environment with resource limitations or other real-time constraints. A traditional approach to reducing computational complexity has been to find approximations to the signals involved in the processing prior to the application of a particular algorithm. Reducing the number of parameters or bits required to adequately represent a signal usually will reduce the amount of computation required to process the signal. For example, certain applications involving highly-correlated signals such as speech, sound, or images use various source coding methods to strip away redundancy from the signals before further processing or transmission. Examples of well established methods for signal approximation include: linear transform coding methods such as the discrete cosine, wavelet, wavelet packet, or Karhunen-Loeve transforms, subband coding methods, linear predictive coding methods, and vector quantization methods [22]. Given the amount of successful research and development that has been accomplished in approximating signals to enhance processing, it is sensible to consider the parallel problem of approximating the algorithms which are used to process these signals. Indeed, it is logical that the same cost vs. quality tradeoffs that are used when determining a signal approximation could be incorporated into the design of signal processing algorithms, in the spirit of maximizing the computational efficiency of complete systems with real-time resource constraints.

In this paper we pursue the goal of dynamically reducing computational cost while maintaining a desired level of output quality in the context of frequency-selective infinite impulse response (IIR) digital filtering. More specifically, our optimization criterion is to minimize average power consumption subject to the constraint that a desired SNR at the output of a frequency-selective IIR digital filter is maintained. This type of objective has been formally studied in the field of approximate processing in computer science [23]. Approximate processing is needed for applications in which it is desirable to dynamically adjust the quality of signal processing results to the availability of resources, such as time, bandwidth, memory, and power [16] [23]. An

early example of an approximate signal processing algorithm is the approximate discrete short-time Fourier transform [24]. More recently, excellent research has been accomplished in the area of incremental refinement structures for approximate signal processing in the context of sinusoidal detection using the fast Fourier transform and in the context of image decoding using the discrete cosine transform [16].

Adaptive filtering algorithms have traditionally been used to dynamically change the values of the filter coefficients based on an adaptation law, while maintaining a fixed filter order [25]. In contrast, in our adaptive approach to low power IIR filtering we show how to dynamically adjust the filter order.

This approach leads to filtering solutions in which the SNR of the filter output may be kept above a specified threshold while using as small a filter order as possible. Since power consumption is linearly proportional to the filter order, our approach achieves power reduction with respect to a fixed order filter whose output is similarly guaranteed to have the output SNR above the specified threshold. Maximum power reduction is achieved by dynamically minimizing the order of the digital filter. This paper centers on low power IIR digital filtering using adaptive approximate processing, or more concisely approximate IIR filtering.

II. LOW POWER DESIGN METHODOLOGIES

The designers of microprocessors have traditionally centered their efforts on increasing the processor clock rate, treating power dissipation as a design issue of secondary importance. An enormous demand for low power design in complementary metal-oxide silicon (CMOS) devices has exists for the following reasons:

- The demand for portable multimedia devices with high throughput and limited rechargeable battery weight and volume has sky-rocketed with the widespread use of cellular phones, laptop computers, and video conferencing systems.
- As the density and size of the chips and systems continues to increase, the design of adequate cooling systems is ever more challenging and important.
- Given that personal computers presently account for a significant percentage of total commercial electricity consumption, the demand for fixed workstations with low power consumption for the purpose of reducing electricity costs and the impact on climate issues will continue to thrive.

In summary, the issues of portability, heat dissipation, and the economics of commercial electricity consumption all serve as practical motivation for the design of low power computation and communication systems.

A. *Sources of Power Consumption*

Reducing both peak power and average power are important priorities in low power digital circuit design. Reducing the peak power levels is important mainly for reliability and proper circuit operation. The required battery weight and size in a portable system is proportional to the time-averaged power consumption. Methods which reduce the average power consumption offer the added benefit of reducing peak power consumption and thus improve reliability [26].

There are four components to the time-averaged power consumption in digital circuits using CMOS technology: switching power, short-circuit power, leakage power, and static power [26]. The total time-averaged

power consumption is the sum of these four individual components.

$$P_{\text{average}} = P_{\text{switching}} + P_{\text{short-circuit}} + P_{\text{leakage}} + P_{\text{static}}. \quad (1)$$

The switching power component dominates and typically accounts for more than 90% of the total average power consumption. This makes the switching power component the primary target for power reduction. In addition, the switching power is the most signal-dependent and algorithm-dependent component, making the switching power the primary focus for algorithmic-based approaches to low power design. We now give a brief summary of the four components of power consumption.

1. *Short-Circuit Power*

When there is a direct conducting path from the voltage supply to ground, the short-circuit power component is present. This component of power consumption is defined to be

$$P_{\text{short-circuit}} = I_{\text{short-circuit}} V_{dd}, \quad (2)$$

where $I_{\text{short-circuit}}$ is the short-circuit current and V_{dd} is the supply voltage. Through proper choice of transistor sizes, the short-circuit power can be kept below 10% of the total power consumption [26].

2. *Static Power*

Circuits that have a constant source of current between their power supplies are subject to power dissipation due to the resulting static currents. The static component of power consumption is defined to be

$$P_{\text{static}} = I_{\text{static}} V_{dd}, \quad (3)$$

where I_{static} is the static current and V_{dd} is the supply voltage. In static random-access memory (SRAM) amplifiers, pulsed circuits may be used to minimize static currents. However, algorithmic-based methods for reducing power consumption can have little or no effect on the static power component.

3. *Leakage Power*

The two types of leakage currents are reverse-bias diode leakage at the transistor drains and sub-threshold leakage through the channel of an “off” device. The leakage component of power consumption is defined to be

$$P_{leakage} = I_{leakage} V_{dd}, \quad (4)$$

where $I_{leakage}$ is the total leakage current and V_{dd} is the supply voltage. The magnitude of both components of the leakage current is set predominantly by the processing technology; thus, algorithmic-based methods for reducing power consumption will have little or no effect on the leakage power component.

4. *Switching Power*

The switching component of power for a CMOS gate with load capacitor C_L is given by

$$P_{\text{switching}} = \alpha C_L V_{dd}^2 f, \quad (5)$$

where a is the node transition activity factor, C_L is the physical load capacitance, V_{dd} is the supply voltage, and

f is the operating frequency. The two components of the node transition activity are transitions due to the static behavior of the circuit and transitions that occur due to the dynamic nature of the circuit. The node transition activity factor a is a function of the logic function being implemented, the logic style, the circuit topology, the input signal statistics, and the sequencing of operations. A system level approach which involves optimizing algorithms, architectures, logic design, circuit design, and physical design can be used to minimize the switched capacitance and thus, in turn, minimize the switching component of power.

B. *Summary of Previous Research*

In this section we give a brief overview of existing methods for reducing power consumption in CMOS devices, and highlight the context in which the contributions of this paper fit in with respect to other methods. To first order, the average switching power consumption $P_{\text{switching}}$ in (5) may be expanded as

$$P_{\text{switching}} = \sum N_i C_i C_L V_{dd}^2 f_s , \tag{6}$$

where C_i is the average capacitance switched per operation of type *i* corresponding to addition, multiplication, storage, or bus access, N_i is the number of operations of type *i* performed per output sample, V_{dd} is the operating supply voltage, and f_s is the sampling frequency.

Since the dominating switching component power consumption in CMOS devices is proportional to the square of the supply voltage V_{dd}, it is clear that supply voltage reduction will have a significant impact on the average switching power consumption. Indeed, reducing the supply voltage is the key to low power operation, even after taking into account the modifications to the system architecture which are required to maintain the computational throughput.

When the supply voltage is reduced by a factor *k*, the power consumption is reduced by a quadratic factor k^2. Unfortunately this power reduction comes at a price. When we reduce V_{dd} we encounter a corresponding decrea--se in throughput. An empirical model for the relationship between the supply voltage V_{dd} and circuit delay T_d is

$$T_d = \frac{k_d}{V_{dd}} , \tag{7}$$

where k_d is a constant determined experimentally and T_d is the circuit processing delay [26]. Thus, while reducing the supply voltage is an excellent way to reduce power consumption, there is an associated penalty to pay in decreased throughput. Typically this decrease in throughput is compensated for by introducing parallelism in the circuitry, which increases the required chip area. In this context we may trade a decrease in power consumption for an increase in chip area.

The supply voltage scaling approach to low power design achieves a reduction in average power consumption by scaling down the supply voltage at the expense of reducing throughput or increasing the required chip area. An alternative approach to low power design is to reduce the switching activity to the minimal level required to perform a given computation, since CMOS circuits do not dissipate power if they are not switching. For this purpose, we may formulate optimization problems for signal processing algorithms to minimize the circuit switching activity. Minimizing the number of multiplications and additions required to perform a given function

is one critical element in reducing the overall circuit switching activity. The framework we have developed for analyzing approximate filtering algorithms was developed for the purpose of reducing the average circuit switching activity via reducing the average number of operations required per output sample in a frequency-selective digital filter. Thus, our attention in this paper is focused on minimizing the switched capacitance in a CMOS circuit by dynamically minimizing the number of operations required to perform frequency-selective digital filtering, subject to output quality constraints.

Real-time digital filtering is an example of a class of applications in which there is no advantage in exceeding a bounded computation rate. For such applications, an architecture-driven voltage scaling approach has previously been developed in which parallel and pipelined architectures can be used to compensate for increased delays at reduced voltages [26]. This strategy can result in supply voltages in the 1 to 1.5 V range by using conventional CMOS technology. Power supply voltages can be further scaled using reduced threshold devices.

Once the power supply voltage is scaled to the lowest possible level, the design goal is to minimize the switched capacitance at all levels of the design abstraction. At the logic level, for example, modules can be simply shut down at a very low level based on signal values. Arithmetic structures such as ripple carry or carry select can also be optimized to reduce transition activity. Architectural techniques include optimizing the sequencing of operations to minimize transition activity, avoiding time-multiplexed architectures which destroy signal correlations, and using balanced paths to minimize glitching transitions. At the algorithmic level, the computational complexity or the data representation can be optimized for low power [26].

Another approach to reducing the switched capacitance and thus saving power is to lower N_i in (6). Efforts have been made to minimize N_i by intelligent choice of algorithm, given a particular signal processing task [27]. In digital filtering applications, the parameter N_i is approximately linearly proportional to the filter order. In the case of conventional filter design, the filter order in a particular application is typically fixed based on worst case signal statistics. This is inefficient if the worst case seldom occurs. More flexibility may be incorporated by using adaptive filtering algorithms, which are characterized by their ability to dynamically adjust the processing to the data by employing feedback mechanisms. In this paper, we illustrate how adaptive filtering concepts may be exploited to develop low power implementations for digital filtering by lowering N_i and thus reducing the switched capacitance and saving power.

Adaptive filtering algorithms have traditionally been used to dynamically change the values of the filter coefficients based on an adaptation law, while maintaining a fixed filter order [25]. In contrast, in our adaptive approach to low power filtering we show how to dynamically adjust the filter order. This approach leads to filtering solutions in which the SNR of the filter output may be kept above a specified threshold while using as small a filter order as possible. Since power consumption, according to (6), is linearly proportional to N_i, which in turn is linearly proportional to the filter order, our approach achieves power reduction with respect to a fixed order filter whose output is similarly guaranteed to have the output SNR above the specified threshold. Maximum power reduction is achieved by dynamically minimizing the order of the digital filter.

III. Approximate Filtering

A brief summary of approximate filtering is now given. The basic idea is to begin filtering a given input sign-

-al with a frequency-selective digital filter of some nominal order, as shown in Fig. 1. This filter has well-defined passband and stopband regions in frequency. After L output samples have been produced, we use the most recent block of L input and output samples to form an easily computable low power estimate of the current input signal-to-noise ratio (SNR), defined as the ratio of the input signal power in the passband of the filter to the input signal power in the stopband of the filter. In Fig. 1 the decision module D uses the signal power estimates Px and Py to form an estimate of the temporally local input SNR. This estimate of the input SNR is then used to update the filter order to be the minimum value which guarantees that the output SNR, defined as the ratio of the output signal power in the passband of the filter to the output signal power in the stopband of the filter, will be greater than or equal to a pre-specified minimum tolerable output SNR. This filter order is then used to produce another block of L output samples, and the filter order update process is repeated.

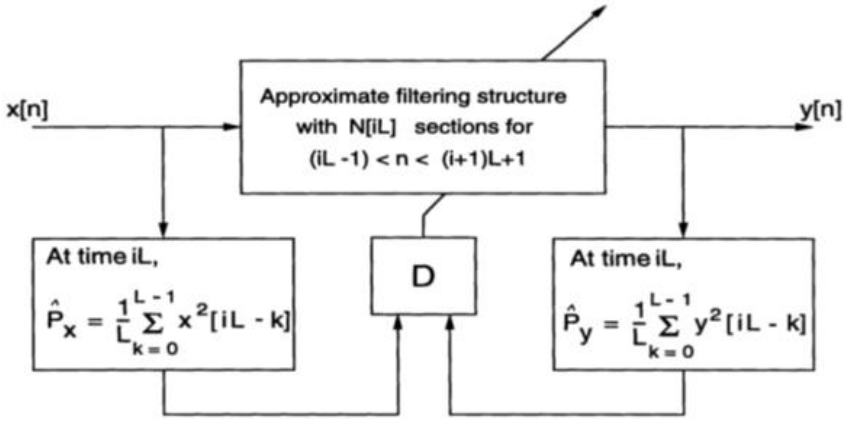

Fig. 1. An overview of approximate filtering. The adaptation strategy for updating the filter order after each new set of *L* output samples is defined by the decision module *D*.

IV. Approximate Filter Structures

In approximate filtering algorithms, the order of a frequency-selective digital filter is varied in a way defined by a control strategy and an approximate filter structure. A collection of frequency-selective digital filters, one for each filter order *N* in a given range $N_{\min} \le N \le N_{\max}$, constitutes an approximate filter structure *H*. Each filter structure *H* must possess the property that its progressively higher order filters have progressively increased average attenuation in the stopband region (s) while maintaining close to unity gain in the passband region (s). The passband *PB*, stopband *SB*, and transition band *TB* regions for all filters in the approximate filter structure *H* are identical. The passband and stopband regions must be explicitly specified in the definition of an approximate filter structure, and by default the transition band is defined to span the remaining portions of the spectrum $\omega \in [-\pi, \pi]$ which are not included in the passband or stopband regions.

The primary advantage of an IIR filter structure is that it can provide significantly better stopband attenuation and less delay than an FIR filter structure having the same number of coefficients. This is a consequence of the output feedback which generates an infinite impulse response with only a finite number of parameters [28]. FIR filter structures are desirable for their guaranteed stability even in the presence of coefficient quantization and for the possibility of an exact linear phase characteristic. However, it should be noted that some commercially available DSP chips can implement certain FIR filters more computationally efficiently than standard IIR filters because the chip architecture has been optimized for a particular FIR filter. In addition, there exist nonlinear

phase FIR filters which can provide significantly better stopband attenuation than the linear phase FIR filters. Therefore, the statement that IIR filters are always more computationally efficient than FIR filters should not be made without careful consideration of the variables at hand.

Two classes of IIR approximate filter structures, truncation and replacement filter structures, are introduced. A replacement filter structure is characterized by the relationship between the coefficients of filters of different orders being completely unconstrained; the coefficients of each individual filter may be selected or replaced independently. In a truncation filter structure this is not allowed. In a truncation filter structure with IIR constituent elements, the set of pole/zero pairs defining each lower order filter is similarly constrained to be a subset of the pole/zero pairs defining the filter with maximum order $N_{\max}$. Thus, the lower order constituent elements in a truncation filter structure are truncated versions of the higher order constituent elements. In a replacement filter structure, the relationship between the coefficients defining filters of different orders are not necessarily related in any way; the pole/zero pairs defining each individual filter may be replaced independently. Given this, we expect the replacement filter structures with unconstrained filter specifications to perform better than the truncation filter structures with constraints. This expectation will be confirmed in our analysis and simulations.

It is clear that truncation filter structures may be described with fewer independent filter coefficients than replacement filter structures. Associated with this property is the fact that approximate filtering using a truncation filter structure requires less memory, chip area, and bus accesses than approximate filtering using a replacement filter structure. In summary, while replacement filter structures have the advantage of offering better filtering performance, truncation filter structures are more power efficient.

We now carefully examine approximate filter structures with IIR constituent filter elements. A conceptual diagram of an IIR replacement filter structure is given in Fig. 2 As an example, consider that at time $n = n_0$, the order of the filter structure may be decreased by one from $N_{\max}$ to $(N_{\max}$ -1) by simply setting the coefficient pair $(aN_{\max}, aN_{\max})$ in Fig. 2 to zero. More generally at time $n = n_0$, the order of the approximate filter may be set to any order $N_0 \le N_{\max}$ by simply setting the coefficient pairs $(aN_{\max}, aN_{\max}) \ldots (aN_0 + 1, aN_0 + 1)$ to zero. It is important to note that the data stream in the middle of the replacement filter structure shown in Fig. 2 continues to shift through all of the $N_{\max}$ vertical delay elements, regardless of which coefficients have been set to zero. In addition, we note that all coefficients are allowed to change their values at each instant in time, according to the elements in the filter structure *H*.

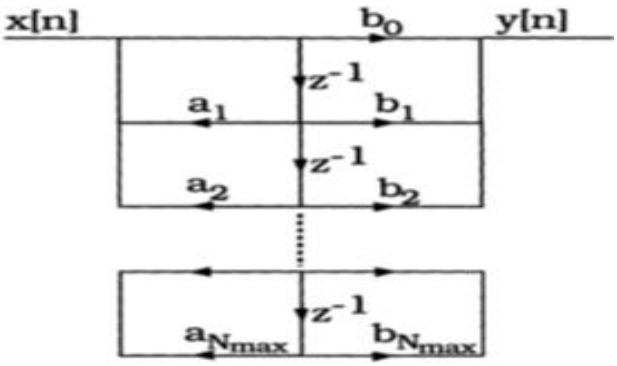

Fig. 2. Conceptual diagram of the IIR replacement filter structure.

Conceptual diagrams of the IIR truncation filter structure are given in Fig. 3. In Fig. 3(a) is a conceptual diagram of the signal flow graph, while in Fig. 3(b) we show a clocked shift register hardware block diagram. As an example, consider that at time $n = n_0$, the order of the filter structure may be decreased by two from $2M_0$ to $(2M_0 - 2)$ by simply truncating the last second-order section and taking the output to be yM_0 -1 [n]. In general, at time $n = n_0$, the order of the truncation filter structure may be set to any even order $2M \le 2M_0$ by truncating the last $(M_0 - M)$ second-order sections from the cascade structure shown in Fig. 3 and taking the output to be yM [n]. If we desire to increase the order of the IIR truncation filter structure, second-order sections may be added to the truncation filter structure at any time.

One measure of the performance of an IIR approximate filter structure is the signal-to-noise ratio (SNR) improvement factor given in (8). For the *k*th-order filter in an approximate filter structure, the SNR improvement factor is:

$$\mathrm{SNRI}[k] \quad \frac{^{\Delta}SB}{\int_{SB} |H_N(\omega)|^2 d\omega} \tag{8}$$

For each fixed filter order in the range $N_{\min} \le N \le N_{\max}$, we would like to maximize the SNR improvement factor in (8) for optimal approximate filtering performance. From (8)

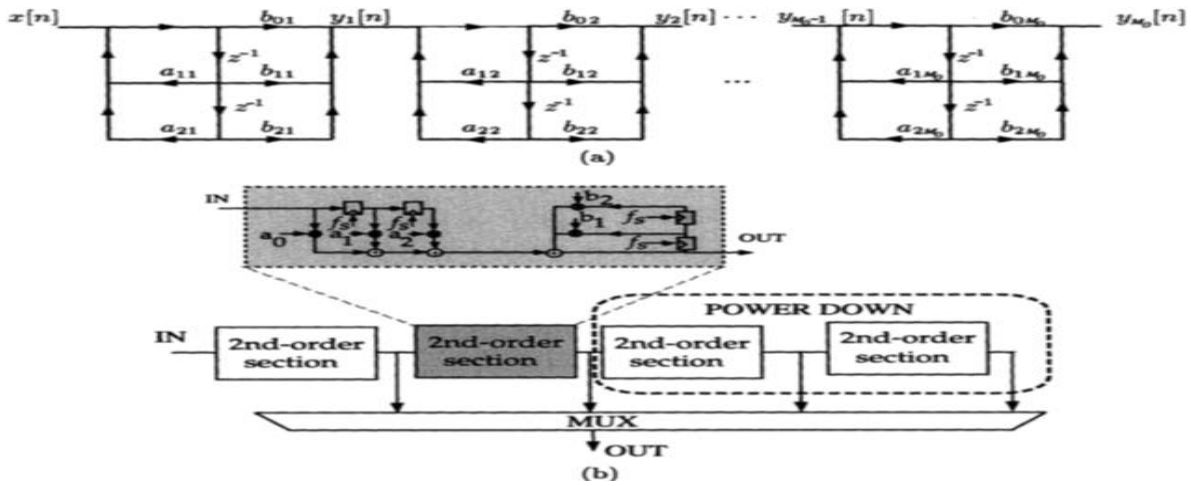

Fig. 3. Conceptual diagrams of the IIR truncation filter structure: (a) the signal flow graph, and (b) the clocked shift register block diagram.

we see that for filter order k this is equivalent to determining the *k*th-order IIR filter with minimum stopband power.

We also use the output power noise- to-signal ratio (OPSNR), defined in [13], as an additional performance metric to encapsulate the state transition error (STE), also defined in [13]. By intelligently choosing an appropriate approximate filter structure, it is possible to reduce the effect of the STE in approximate filtering.

Since finding optimal IIR filter structures designed according to these two performance metrics (SNRI and OPNSR) involves an unsolvable constrained nonlinear optimization problem, we do not pursue direct IIR filter structure design. Instead we evaluate the performance of IIR approximate filter structures using four classical IIR digital filter constituent elements, namely Butterworth, Chebyshev, inverse Chebyshev, and elliptic digital filters.

A. *Replacement Filter Structures*

A replacement filter structure H_R is defined by a set of $(N_{max} - N_{min} + 1)$ digital filters, one for each filter order N in a given range $N_{min} \le N \le N_{max}$. We denote this set by

$$H_R = \{H_{Nmax}(\omega), H_{Nmax}\text{-}1(\omega), .., H_{Nmin}(\omega)\} \quad (9)$$

We define the filter structure H_R by defining its constituent filter elements. These filters must be all IIR and should possess similar spectral characteristics for obvious practical reasons. IIR approximate filter structures have IIR constituent filter elements and the replacement quality. Because IIR approximate filter structures have the replacement quality, the coefficients of the filter of a particular order are unrelated to the coefficients of filters of different orders within the structure.

In Fig. 4 we have plotted the frequency responses magnitudes for the Butterworth IIR replacement filter structure for $N_{min} = 2$ and $N_{max} = 10$. In Fig. 5, Fig. 6, and Fig. 7 we show similar plots for the Chebyshev, inverse Chebyshev, and elliptic replacement filter structures. All the filters have been normalized such that the maximum ripple in the passband is equal to 0.01.

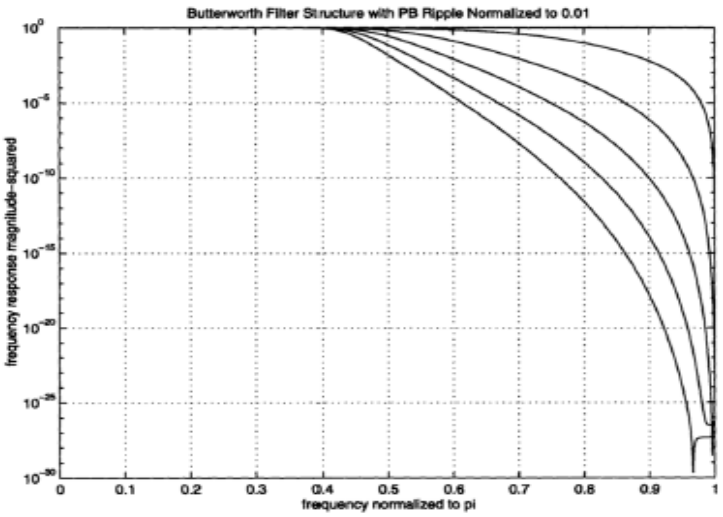

Fig. 4. Frequency response magnitude-squared plots for the Butterworth IIR replacement filter structure.

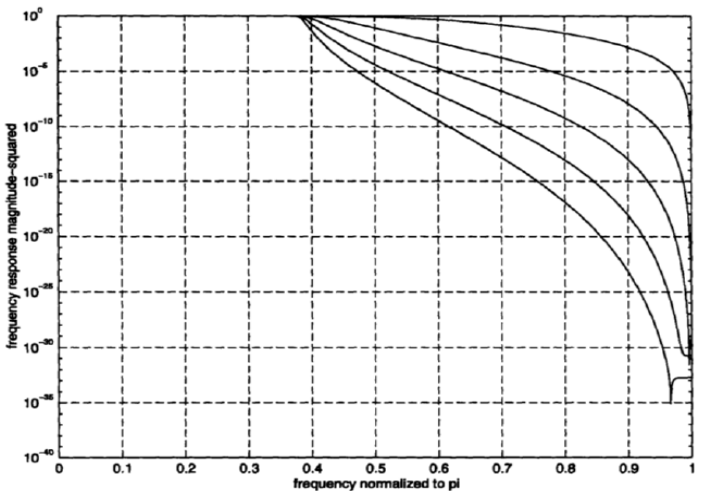

Fig. 5. Frequency response magnitude-squared plots for the Chebyshev IIR replacement filter structure.

The performance of an IIR approximate filter structure is measured by the SNR improvement factor. From inspection of the frequency response magnitude-squared plots, we observe that the elliptic replacement filter responses have visually the lowest stopband power, and thus we would expect the elliptic replacement filter structure to have the best performance profile. Indeed, this is confirmed in Fig. 8, where we show a comparison of the performance profiles for the four IIR replacement filter structures considered thus far.

To assess performance we also consider the STE performance metric which measures the corruptive effect of instanta

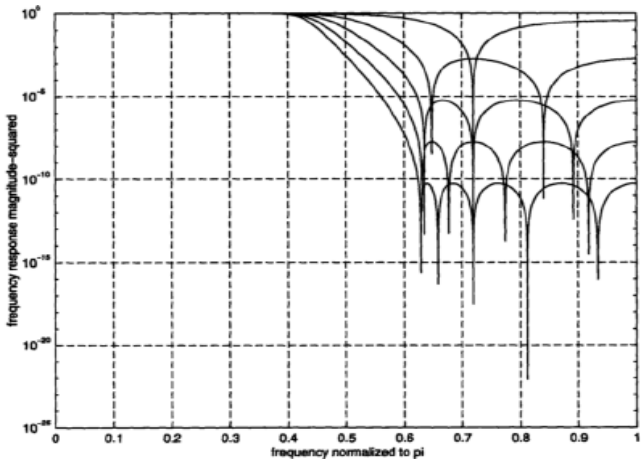

Fig. 6. Frequency response magnitude-squared plots for the inverse Chebyshev IIR replacement filter structure.

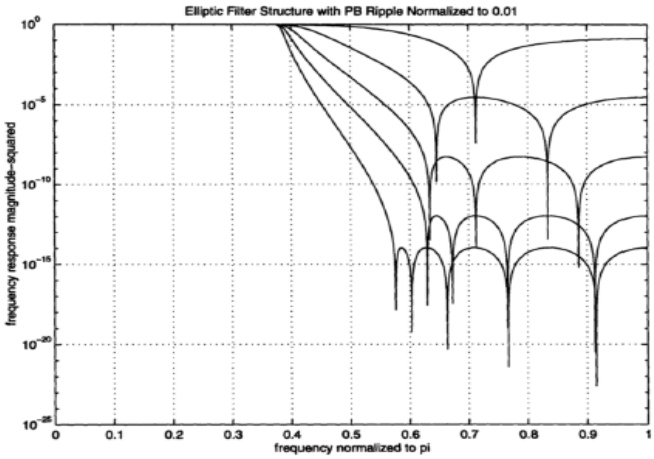

Fig. 7. Frequency response magnitude-squared plots for the IIR elliptic replacement filter structure.

neously switching from the maximum filter order to each of the other lower order filters in the approximate filter structure. The STE performance metric for a filter of order N is defined as the output power noise-to-signal ratio, OPNSR [N], as defined in [13]. The output power noise-to-signal ratio OPNSR [N_k] here is defined as that arising from instantaneously switching from the $N_{\max}$ - order filter to the N_k - order filter. In Fig. 9 we show a comparison of the STE performance metric for the same set of four IIR replacement filter structures. In this case the Chebyshev IIR replacement filter structure is the best in terms of STE performance.

In closing, we make a few notes on the relative number of operations required to implement different IIR filter types. In general an order-N IIR filter requires $2N$ additions and $2N$ multiplies per output sample to implement in direct form.

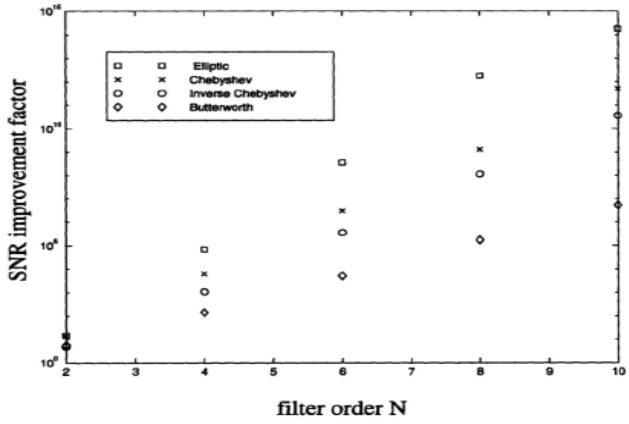

Fig. 8. Comparison of the performance profiles for the Butterworth, Chebyshev, inverse Chebyshev, and elliptic IIR replacement filter structures.

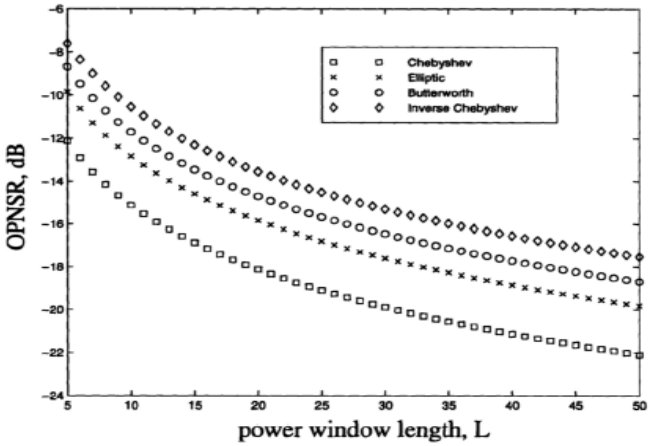

Fig. 9. Comparison of the OPNSR for the Butterworth, Chebyshev, inverse Chebyshev, and elliptic IIR replacement filter structures.

However only $2N$ additions and $(N + 1)$ multiplies are needed to implement an Nth-order Butterworth filter in direct form, due to the fact that all its zeros are at $z = -1$. This is true for Chebyshev and inverse Chebyshev IIR filters as well, which are both all-pole filters in the analog domain and thus transform to having all their zeros at $z = -1$ in the discretetime domain via the bilinear transformation.

In [29] it is shown that an elliptic filter can be implemented as the sum of two all pass filters. With this special decomposition only N multiplies per output sample are required. Even without this allpass decomposition, an elliptic filter can be implemented with $1{:}5N$ multiplies due to the symmetry of its numerator polynomial coefficients. Thus, there is no great advantage to the Butterworth Chebyshev, and inverse Chebyshev filters having all their zeros at z = -1. Elliptic filters can be implemented equally as efficiently with a better SNR improvement factor performance profile. Because of the equiripple nature of both the passband and the stopband, the elliptic filter requires a much smaller order than that of a Butterworth or Chebyshev filter meeting the same specifications. Thus, the elliptic replacement filter structure is, in general, a good choice for IIR approximate filtering.

B. *Truncation Filter Structures*

A truncation filter structure H_T is defined by a set of $(N_{\max} - N_{\min} + 1)$ digital filters, one for each filter order N in a given range $N_{\min} \le N \le N_{\max}$. We denote this set by

$$H_T = \{H_{N\max}(\omega),\ H_{N\max-1}(\omega), ..., H_{N\min}(\omega)\} \quad (10)$$

Again we define the filter structure H_T by defining its constituent filter elements. These filters must be all IIR and should possess similar spectral characteristics for obvious practical reasons. To define an IIR truncation filter structure, we must first define the IIR constituent filter with maximum order, $H_{N\max}$. Then the rest of the filters of lower orders are defined by a pruning sequence pole/zero pairs. In this type of approximate filter structure the filter of maximum order $H_{N\max}(\omega)$ may be a digital Butterworth filter, a Chebyshev filter, an inverse Chebyshev filter, or an elliptic filter. In defining an IIR truncation filter structure there is freedom to choose the pruning sequence to meet desired performance specifications. As a final note, we mention that in all our analyses and simulations we normalize each constituent filter element to have a unity DC gain.

The SNR improvement factor represents the factor by which the first set of *N* sections of a truncation filtering structure improves upon the input SNR. Fig. 10 shows the SNR improvement factor as a function of *N* for the case of truncations of a Butterworth filter with a half-power frequency of $\pi/2$ implemented as a cascade of ten second-order sections. The stopband in this case was defined to be $\omega \in [5\pi/8, \pi]$.

To specify a truncation IIR approximate filter structure, we begin by selecting the IIR filter of maximum order $N_{\max}$. The poles and zeros for the lower order filters are then defined, and the problem is to choose the best ordered sequence of poles and zeros to prune away from the given IIR filter with order $N_{\max}$ in order to obtain each of the lower order filters. The combination of the filter of order $N_{\max}$ with the order pole/zero pruning sequence defines the truncation IIR approximate filter structure H_T. To give some insight into the nature of a truncation IIR approximate filter structure specification, we present the following example.

Let us consider the case of a Butterworth filter of order $2M_0$. A cascade structure for this filter consists of a serial connection of M_0 second-order Direct-Form II sections, as was previously shown in Fig. 3. Each section corresponds to a pair of conjugate poles of the Butterworth filter and two zeros (both located at *z* = -1). Denoting the frequency response of the order $-2M_0$ Butterworth filter by $H_{M0}(\omega)$, we may write

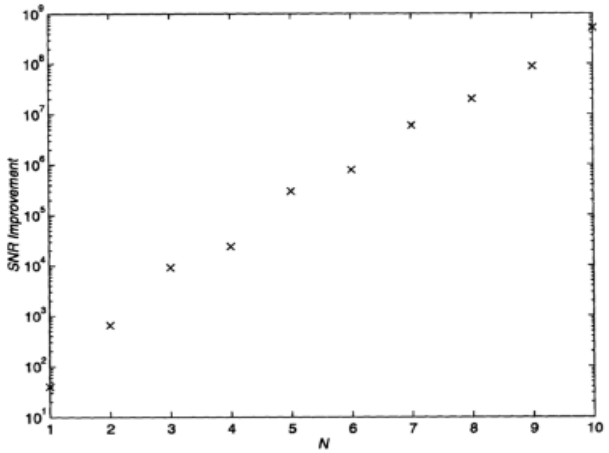

Fig. 10. Performance profile for truncations of a 20th-order Butterworth filter with half-power frequency $\pi/2$.

$$H_{Mo}(\omega) = G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega) \quad (11)$$

where $G_i(\omega)$ denotes the frequency response of the *i*th second-order section in the cascade structure of Fig. 3. It can be furthermore assured that $G_i(0)=1$. If only the first *N* sections $(N \le M_0)$ of the cascade structure in Fig. 3 are used, the resulting order-2*N* truncated Butterworth filter has the frequency response $H_N(\omega)$, given by

$$H_N(\omega) = \prod_{k=1}^{N} G_k(\omega) \tag{12}$$

We are free to assign the Butterworth pole pairs to each of the second-order sections $G_k(\omega)$. It is desirable to make this assignment assure that as the number of second-order sections is increased, the average attenuation in the stopband of the filter also increases, while keeping the passband gain of each of the filters in H_T close to unity. One strategy for making such a pole-pair assignment is as follows: the ordered set of second-order sections $G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega)$ is chosen from the $M_0!$ possible ordered sets to be which minimizes the objective function.

$$J_T(G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega)) = \max_{1 \le k \le M} |\,|H_k(\omega)|^2 - 1 \quad \omega \in PB, \tag{13}$$

Where *PB* denotes the spectral region of support of the passband of $H_k(\omega)$. In other words, given the order-$2M_0$ filter $H_{M0}(\omega)$, the problem is to determine the sequence of filters $G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega)$ such that $J_T(G_1(\omega)G_2(\omega)...G_{M0}(\omega))$ is minimized. The second-order sections $G_k(\omega)$ define the pole/zero truncation (pruning) sequence since $G_1(\omega)$ is truncated first to obtain the (M_0-1)- section filter, $G_2(\omega)$ is truncated second to obtain the (M_0-2)- section filter, and so on. Thus each ordered truncation sequence $G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega)$ defines a corresponding truncation filter structure.

$$H_T = H_N(\omega), H_{N-1}(\omega),...H_1(\omega), \tag{14}$$

Which can be used in the approximate filtering algorithm. We define the globally optimal truncation filter structure H_T to be the particular truncation filter structure which minimizes $J_T(G_1(\omega)G_2(\omega)G_3(\omega)...G_{M0}(\omega))$. That is,

$$H_T^* = \arg\min_{1 \le k \le (M_0!)^2} J_T(H_T^k). \tag{15}$$

As indicated in (15), in order to find H_T^* we must exhaustively search over $(M_0!)^2$ distinct filter structures H_T^k and evaluate $J_T(H_T^k)$ for each one. As defined earlier, H_T^* is the truncation filter structure which results in the minimum value of $J_T(H_T^k)$ over the range $1 \le k \le (M_0!)^2$. Since digital Butterworth filters have all their zeros at *z* = -1, the ordering of the zeros in this truncation filter structure does not matter. All second-order section pole pairs are accompanied by two zeros at *z* = -1. For general IIR filters in which the zero locations are not all the same, the optimization must be done over all possible pole/zero pair combinations, resulting in $(M_0!)^2$ distinct filter structures to search over. There are $(M_0!)^2$ possibilities since for each of the $M_0!$ distinct pole pair orderings there exist $M_0!$ distinct possible zero pair orderings. In the case of the Butterworth filter structure, since the ordering of the zero pairs does not matter (all of the zeros are at *z* = -1), only $M_0!$ distinct filter structures exist.

To illustrate, consider the application of this strategy to a Butterworth filter with $M_0 = 3$ and a half-power frequency of $\pi/2$. As explained earlier, there are $M_0! = 3\times2\times1 = 6$ distinct filter structures $H_T^1 \ldots H_T^6$ to consider in determining the optimal truncation filter structure, H_T^*, for which $J_T(H_T^*)$ is minimum. For this case we have

$$H_N(\omega) = \prod_{k=1}^{3} G_k(\omega) \tag{16}$$

with

$$H_T^1 = \{H_1^6(\omega),\ H_4^1(\omega),\ H_2^1(\omega)\} \tag{17}$$

$$= \{G_1(\omega)G_2(\omega)G_3(\omega),\ G_1(\omega)G_2(\omega), G_1(\omega)\}, \tag{18}$$

$$H_T^2 = \{H_6^2(\omega),\ H_4^2(\omega),\ H_2^2(\omega)\} \tag{19}$$

$$= \{G_1(\omega)G_3(\omega)G_2(\omega),\ G_1(\omega)G_3(\omega)G_1(\omega)\}, \tag{20}$$

$$H_T^3 = \{H_6^3(\omega),\ H_4^3(\omega),\ H_2^3(\omega)\} \tag{21}$$

$$= \{G_2(\omega)G_1(\omega)G_3(\omega),\ G_2(\omega)G_1(\omega), G_2(\omega)\}, \tag{22}$$

$$H_T^4 = \{H_6^4(\omega),\ H_4^4(\omega),\ H_2^4(\omega)\} \tag{23}$$

$$= \{G_2(\omega)G_3(\omega)G_1(\omega),\ G_2(\omega)G_3(\omega), G_2(\omega)\}, \tag{24}$$

$$H_T^5 = \{H_6^5(\omega),\ H_4^5(\omega),\ H_2^5(\omega)\} \tag{25}$$

$$= \{G_3(\omega)G_1(\omega)G_2(\omega),\ G_3(\omega)G_1(\omega), G_3(\omega)\}, \tag{26}$$

and

$$H_T^6 = \{H_6^6(\omega),\ H_4^6(\omega),\ H_2^6(\omega)\} \tag{27}$$

$$= \{G_3(\omega)G_2(\omega)G_1(\omega),\ G_3(\omega)G_2(\omega), G_3(\omega)\}, \tag{28}$$

Overlays of the frequency responses of the three filters in each of the above truncation filter structures H_T^k above are shown in Fig. VII. It should be observed that as the number of sections (*N*) is increased, the average attenuation the stopband also increases. On the other hand, as can be seen in Fig. VII, the filter gain remains close to unity in most of the passband.

Table 1. Numerical values for $J_T(H_T^1) \ldots J_T(H_T^6)$ for the butterworth optimal truncation filter structure. Note that $J_T^*(H_T^*) = J_T(H_T^2)$.

Filter Structure	**Value of $J_T(H_T^k)$**
H_T^1	0.5767
$H_T^* = H_T^2$	0.4926
H_T^3	0.5767
H_T^4	0.9553

Filter Structure	Value of J_T (H_T^k)
H_T^5	1.3438
H_T^6	1.3438

After evaluating $J_T(H_T^k)$ over the range $1 \le k \le 6$, we determined empirically that $J_T(H_T^2)$ is the minimum, and thus for this example $J_T^*(H_T^*) = J_T(H_T^2)$. All six of the values of $J_T(H_T^k)$ are tabulated in Table 1, for reference.

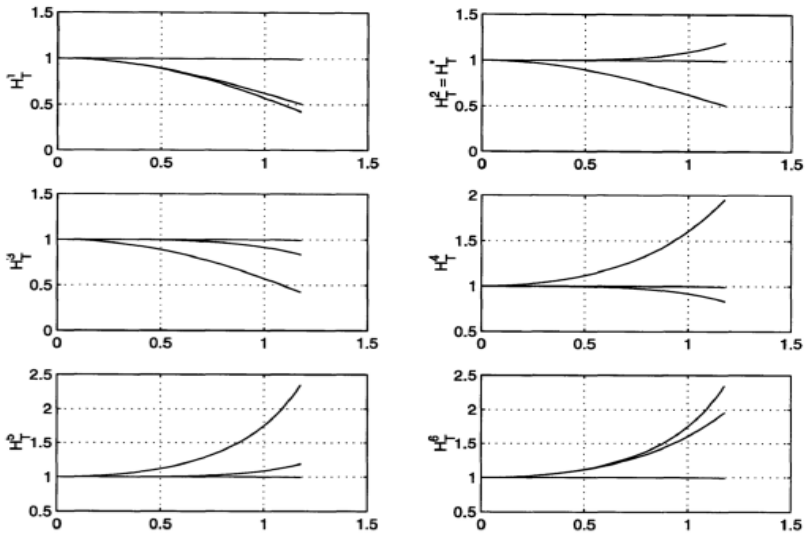

Fig. 11. Magnitude-squared frequency responses for truncations of a 6thorder Butterworth filter with 1, 2, and 3 second-order sections, for each of the possible distinct truncation filter structures. The optimal truncation filter structure is $H_T^* = H_T^2$.

V. Experimental Design and Results of the Low-Power Adaptive IIR Digital Filter

In this section we present computer simulations which show that significant power savings may be achieved when the order of an IIR digital filter is dynamically varied to provide time varying stopband attenuation in proportion to the time-varying SNR of the input signal, while maintaining a fixed level of output quality. We highlight experiments involving speech signals to demonstrate the practical viability of the low-power adaptive IIR digital filter presented in this paper. As we shall see, significant reduction in power consumption over a fixed order IIR digital filter is achieved in simulations involving the demultiplexing frequency-division multiplexed (FDM) speech signals.

We illustrate the potential of the low-power adaptive IIR digital filter to reduce power consumption in speech processing. We use a Butterworth truncation filter structure with 10 second-order sections. This approximate filter structure and the adaptation control strategy described in Section 3 was applied to two speech signals which had been frequency division multiplexed. The power window length was chosen to be L = 100 and the minimum tolerable output SNR was set to 1,000. The IIR filters in the Butterworth truncation filter structure each had a half-power frequency of $\pi/2$. The stopband was defined to be between $5\pi/8$ and π, while the passband was defined to be between 0 and $3\pi/8$. One speech signal was spectrally centered in the passband

region of the lowpass filter and the other was modulated into the stopband region of the lowpass filter. The sampling rate for each of the speech signal was 16,000 Hz. Fig. 12 shows the speech signal in the passband, the speech signal in the stopband, and the evolution of the number of filter sections used by the approximate filtering technique. Since we are using cascades of second order subsections, the power consumption is directly proportional to the number of active subsections, which can be enabled or disabled with low overhead. Since we are calculating the model order over a window length L = 100, less than 1% overhead is needed for these calculations. Examination of the figure shows that as would be expected, the number of filter sections is large when the input SNR is small. Furthermore, the number of filter sections is small when the input SNR is high. If we compare the power consumption for in the lowpower adaptive filter to the power consumption of a fixed IIR filter order need to handle the worst-case signal statistics, we see that a power reduction of approximately 63% is achieved in this simulation. This can be best understand with a visual inspection of Fig. 12, where we can see the time-varying number of filter sections as a function of sample number ranges between 2 and 10, while the number of sections in a fixed-order IIR filter that would ensure we are able to maintain the minimum tolerable output SNR of 1,000 at all times is 8.

For comparison of the of the power savings achieved by the adaptive digital filter presented in this paper to other results recently reported in the literature, we present the findings of three other experiments. First, a recent state-of-the-art

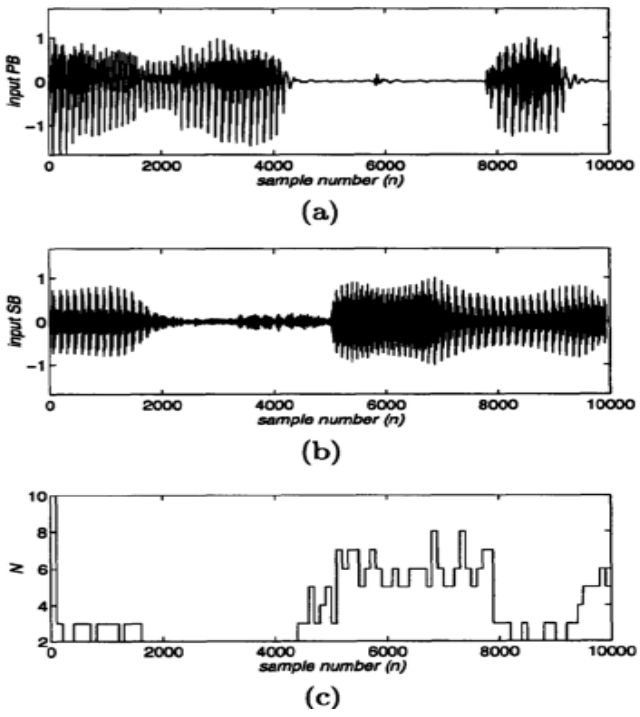

Fig. 12. Demultiplexing of FDM speech using low power frequency selective filtering. (a) passband speech, (b) stopband speech, and (c) number of filter sections as a function of sample number.

digital FIR filter using data-driven clock gating and multibit flip-flops combined achieved 22% to 25% power reduction compared to that using a conventional design [3]. Secondly, in another very recent study, a novel design for an energy efficient IIR digital filter achieved nearly 63% reduction in energy with a negligible deviation of the frequency response from the standard implementation [1]. Finally, a low power reconfigurable FIR digital filter based on dual mode operation achieved power savings up to 37.97% in simulations using

speech signal processing, similar to the simulations in this section [30]. These results demonstrate how low-power digital signal processing continues to be an area of focused interest and innovation.

VI. CONCLUSIONS

The main contribution of this paper is the development of a framework for the design and implementation of IIR approximate filters using signal-dependent algorithms which meet fixed performance specifications while dynamically minimizing power consumption. We have defined two types of approximate filter structures and investigated their relative performance. The replacement structures were shown to perform the best in terms of maximizing the SNR improvement factor, while truncation filter structures were shown to have the advantage of less storage requirements which is equivalent to less power consumption in CMOS devices. IIR elliptic filters were shown to be excellent choices for constituent filter elements in approximate filter structures. Thus, the decision to use a truncation or replacement filter structure depends on the application as well as the associated power and performance specifications.

REFERENCES

[1] R. Pilipovi'c, V. Risojevi'c, and P. Buli'c (2021). "On the Design of an Energy Efficient Digital IIR A-Weighting Filter using Approximate Multiplication." Sensors, 21, 732. https://doi.org/10.3390/s21030732
[2] M. Kang, S. K. Gonugondla and N. R. Shanbhag (2020, December)." Deep In-Memory Architectures in SRAM: An analog approach to approximate computing." Proceedings of the IEEE, 108(12), pp. 2251-2275. https://doi.org/10.1109/JPROC.2020.3034117
[3] P. Agathoklis, L. Touil, A. Hamdi, I. Gassoumi, and A. Mtibaa (2020)." Design of low-power structural FIR filter using data-driven clock gating and multibit flip-flops." Journal of Electrical and Computer Engineering. https://doi.org/10.1155/2020/8108591
[4] M.S. Kim, A.A.D.B. Garcia, L.T. Oliveira, R. Hermida, and N. Bagherzadeh (2018). "Efficient mitchell's approximate log multipliers for convolutional Neural Networks." IEEE Trans. Comput., 68, pp. 660-675.
[5] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi (2018)." Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications." IEEE Trans. Circuits Systems I Regular Paper, 65, pp. 2856-2868.
[6] R. Pilipovi´c and P. Buli´c (2020). "On the design of logarithmic multiplier using radix-4 booth encoding." IEEE Access, 8, pp. 64578-64590.
[7] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi (2018). "Approximate hybrid high radix encoding for energy-efficient inexact multipliers." IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 26, pp. 421-430.
[8] W. Liu, T. Cao, P. Yin, Y. Zhu, C. Wang, E. E. Swartzlander, and F. Lombardi (2019). "Design and analysis of approximate redundant binary multipliers." IEEE Trans. Comput., 68, pp.804-819.
[9] A. Agrawal, J. Choi, K. Gopalakrishnan, S. Gupta, R. Nair, J. Oh, D.A. Prener, S. Shukla, V. Srinivasan, and Z. Sura (2016, October)." Approximate computing: Challenges and opportunities." Proceedings of the 2016 IEEE International Conference on Rebooting Computing (ICRC), San Diego, CA, USA, 17-19 October, pp. 1-8.
[10] S. Mittal. (2016). "A survey of techniques for approximate computing." ACM Comput. Surv. (CSUR), 48 (62).
[11] N. E. Jerger and J. S. Miguel (2018). "Approximate Computing." IEEE Micro, 38, pp. 8-10.
[12] L. Eeckhout (2018). "Approximate Computing, Intelligent Computing." IEEE Micro 38, pp. 6-7.
[13] J.T. Ludwig. (1997, September) "Low power digital filtering using adaptive approximate processing." Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT RLE. September 2, 1997.
[14] J.M. Winograd, J.T. Ludwig, S.H. Nawab, A. Chandrakasan, and A.V. Oppenheim (1996, November). "Flexible systems for digital signal processing." AAAI Fall Symposium on Flexible Computation in Intelligent Systems: Results, Issues, and Opportunities, Cambridge, MA.
[15] J.M. Winograd, J.T. Ludwig, S.H. Nawab, A. Chandrakasan, and A.V. Oppenheim (1995, July)." Approximate processing and incremental refinement concepts." Proceedings 2nd ARPA Rapid Prototyping of Application-Specific Signal Processors (RASSP) Conference, pp. 257-261, Washington, D.C.
[16] J.M. Winograd (1997)." Incremental refinement structures for approximate signal processing." Ph.D. thesis, Boston University, February, 1997.
[17] S.H. Nawab and J.M. Winograd (1995). "Approximate signal processing using incremental refinement and deadline-based algorithms." Proceedings of the International Conference on Speech, Acoustics, and Signal Processing, pp. 2857-2860, Detroit, MI, April, 1995.
[18] S.H. Nawab, A.V. Oppenheim, A. Chandrakasan, J.T. Ludwig, J.M. Winograd (1997). "Approximate signal processing." Journal on VLSI Signal Processing, 15(1- 2), January-February, 1997.
[19] J.T. Ludwig, S. H. Nawab, and A. P. Chandrakasan (1996, March)." Low-power digital filtering using approximate processing." IEEE Journal on Solid State Circuits, 31(3) pp. 395-400.
[20] J.T. Ludwig, S.H. Nawab, and A.P. Chandrakasan (1996, August)." Convergence results on adaptive approximate filtering." Advanced Signal Processing Algorithms (F.T. Luk, ed.), Proceedings of SPIE, Denver, CO.
[21] J.T. Ludwig, S.H. Nawab, and A. P. Chandrakasan (1995, May). "Lowpower filtering using approximate processing for DSP applications." Proceedings of the Custom Integrated Circuits Conference (CICC), pp. 185-188, Santa Clara, CA.
[22] N.S. Jayant and P. Noll (1984) "Digital Coding of Waveforms: Principles and Applications to Speech and Video." Prentice Hall, Englewood Cliffs, NJ.
[23] V.R. Lesser, J. Pavlin, and E. Durfee (1988). "Approximate processing in real-time problem solving." AI Magazine, pp. 49-61.
[24] E. Dorkan (1993). "Approximate processing and knowledge-based reprocessing of non-stationary signals." Ph.D. thesis, Boston University, September, 1993.
[25] S. Haykin (1994). "Adaptive filtering theory." Prentice Hall, Englewood Cliffs, NJ.

[26] A.P. Chandrakasan and R.W. Broderson (1995). "Low power digital CMOS Design." Kluwer Academic Publishers, Norwell, MA.
[27] B.M. Gordon and T.A. Meng (1994, April). "Low power subband video decoder architecture." Proceedings of the International Conference on Speech, Acoustics, and Signal Processing, Sidney, Australia.
[28] J.J. Shynk (1989, April). "Adaptive IIR filters." IEEE ASSP Magazine, 6, pp. 4-21.
[29] P.P. Vaidyanathan (1993). "Multirate Systems and Filter Banks." Prentice Hall, Englewood Cliffs, NJ.
[30] S. Padmapriya and V. Lakshmi Prabha (2015). "Design of an efficient dual mode reconfigurable FIR filter architecture in speech signal processing." Microprocessors and Microsystems, 39(7), pp. 521-528.

AUTHOR'S PROFILE

Jeffrey Ludwig, was born in Peoria, IL on November 23, 1968. He has a S.B. in aeronautics and astronautics (1991) and an S.M (1993) and a Ph.D. (1997) in electrical engineering and computer Science, all from the Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A. He served as Director of Jump Labs, the research division of Jump Trading, a Chicago-based high frequency proprietary trading firm. Before that he was a portfolio manager at SAC Capital Management, New York where he managed quantitative futures strategies spanning equities, fixed income, commodities, and volatility, and also the Director of the Algorithmic Trading Group at Barclays Capital in New York and London. Prior to that, he was a Senior Vice President and Portfolio Manager for Pacific Investment Management company (Pimco) for 5 5 years, where he served as Head of Equity Derivatives. Pimco recruited him from Credit Suisse First Boston in New York where he was a successful proprietary equity arbitrage trader. He has 20 years of investment experience. Professor Ludwig is an Assistant Professor of Teaching in the Department of Mathematics at University of California, Irvine, U.S.A. since 2018. He serves as the faculty advisor for the math finance concentration and specializes in innovative teaching and collaborative research with industry in mathematical finance and quantitative trading.