A Recursive Wavelet-Based Strategy for Real-Time Cochlear Implant Speech Processing on PDA Platforms

Vanishree Gopalakrishna, Nasser Kehtarnavaz, *Senior Member, IEEE*, and Philipos C. Loizou*

***Abstract*—This paper presents a wavelet-based speech coding strategy for cochlear implants. In addition, it describes the real-time implementation of this strategy on a personal digital assistant (PDA) platform. Three wavelet packet decomposition tree structures are considered and their performance in terms of computational complexity, spectral leakage, fixed-point accuracy, and real-time processing are compared to other commonly used strategies in cochlear implants. A real-time mechanism is introduced for updating the wavelet coefficients recursively. It is shown that the proposed strategy achieves higher analysis rates than the existing strategies while being able to run in real time on a PDA platform. In addition, it is shown that this strategy leads to a lower amount of spectral leakage. The PDA implementation is made interactive to allow users to easily manipulate the parameters involved and study their effects.**

***Index Terms*—Cochlear implants, real-time implementation, recursive-wavelet computation, PDA platform.**

I. Introduction

COCHLEAR implants are surgically implanted prosthetic devices, which are used to provide a sensation of hearing in profoundly deaf people. Approximately 188 000 people around the world have been fitted with cochlear implants as of 2009 [1]. Cochlear implants have two components: a component residing outside the body consisting of a microphone, a speech processor, and a transmitter, and a component that is surgically implanted inside the body behind the ear consisting of a receiver and an electrode array [2]. The main function of the speech processor is to decompose the input signal into its frequency components, much like a healthy cochlea analyzes the input signal into its frequency components. Much of the success of cochlear implants can be attributed to the development of sophisticated signal-processing algorithms implemented in real-time on the speech processor. These algorithms were designed to some extent to mimic the function of a healthy cochlea [2].

A number of factors need to be considered when implementing in real-time signal-processing algorithms for cochlear implant applications. First, the incurred signal delay needs to be small. Second, the frequency decomposition, which can be done using either a filterbank or a fast Fourier transform (FFT), needs to yield adequate frequency resolution similar to that of the healthy ear. Fine spectral resolution would be highly desirable for transmission of fine spectral cues (“fine structure”) to the implant recipients. Fine resolution is also needed for the implementation of current steering or current focusing stimulation strategies [3]. Third, and perhaps most important, the complexity of the algorithm needs to be low, so as to minimize the power requirements. The latter criterion is important as it can prolong the battery life of the speech processor. Finally, the implementation needs to provide for good temporal resolution, which in turn will allow for the possibility of high-stimulation rates. Clearly, the aforementioned factors are interrelated. For instance, the use of filterbanks for spectral decomposition provides excellent temporal resolution, but limited (by the number of channels) frequency resolution. The use of FFT provides excellent frequency resolution, but limited (by the update frame rate) temporal resolution.

In this paper, we explore the use of wavelets for frequency decomposition and describe its real-time implementation on a PDA platform. A recursive-wavelet implementation is proposed that addresses the limitation of the FFT approach of reduced temporal resolution. The proposed implementation allows for high-stimulation rates and has low-computational complexity. The wavelet approach is shown to offer advantages over the previous methods based on the FFT and recursive discrete FT (DFT) [4].

This paper is organized as follows. An overview of common signal-processing strategies is provided in Section II. Section III describes three wavelet decomposition tree structures and Section IV presents a recursive mechanism for updating wavelet coefficients from frame to frame. Section V covers the implementation aspect and the optimization steps taken to reach a real-time working solution on a PDA platform. The results are presented in Section VI, where all the methods are compared in terms of computational complexity, spectral leakage, fixed-point accuracy, and real-time processing. Finally, the conclusions are given in Section VII.

II. Cochlear Implant Signal-Processing Strategies

A. Continuous Interleaved Sampling

Continuous interleaved sampling (CIS) [5] is a common signal-processing strategy in cochlear implants and is available

Manuscript received December 18, 2009; revised February 7, 2010 and March 24, 2010; accepted March 30, 2010. Date of publication April 15, 2010; date of current version July 14, 2010. This work was supported by the National Institutes of Health under Contract N01-DC-6-0002 from the National Institute on Deafness and Other Communication Disorders. *Asterisk indicates corresponding author.*

V. Gopalakrishna and N. Kehtarnavaz are with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA.

*P. C. Loizou is with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: loizou@utdallas.edu).

Digital Object Identifier 10.1109/TBME.2010.2047644

0018-9294/$26.00 © 2010 IEEE

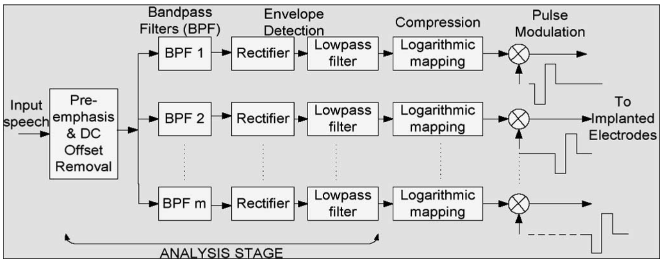


Fig. 1. Block diagram of the CIS strategy.

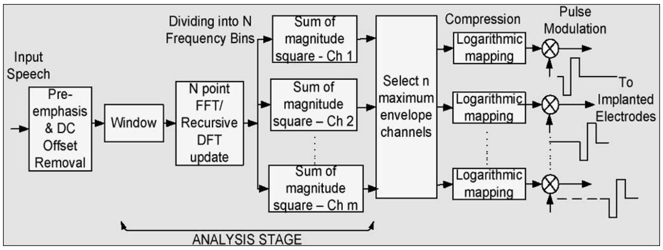


Fig. 2. Block diagram of the *n*-of-*m* strategy.

by all three implant devices (Med-El, Advanced Bionics Corporation and Cochlear Corporation). The number of analysis channels varies across the three-implant devices. A bank of bandpass filters is utilized to divide the preemphasized input speech signal into a number of frequency bands. Rectification and low-pass filtering with a cutoff frequency of 400 Hz follow the spectral decomposition to extract the envelope corresponding to each channel [5], [6]. The channel envelope is then compressed using a logarithmic compression map to ensure that the amplitudes fall within the electrical dynamic range. The compressed envelopes modulate biphasic pulses stimulating the implanted electrodes. Generally, 16–22 electrodes are used for stimulation, thus a bank of 16–22 bandpass filters are often used. The channels are often spaced logarithmically or semilogarithmically (linear spacing up to 1 kHz and logarithmic spacing thereafter) across the speech spectrum. Fig. 1 illustrates the block diagram of the CIS strategy.

B. Advanced Combination Encoder

Advanced combination encoder (ACE) is an *n*-of-*m* signal-processing strategy used in the Nucleus-24 implant device (Cochlear Corporation, Lane Cove, NSW). It uses the FFT to decompose the input speech signal into a number of frequency bands. Often the speech signal captured by a microphone is windowed using an 8-ms Hanning window with a sampling frequency of 16 kHz. In other words, a 128-point FFT is often utilized. The magnitude FFT spectrum is computed, and these magnitudes are summed over the frequency range of a channel to produce 22 envelopes [6]–[8]. Out of the 22 envelopes, 8–12 maximum envelopes are selected for stimulation. In the commercial Freedom processor, an 8-of-22 strategy is often implemented. The selected channel amplitudes are mapped or compressed to fit within the electrical dynamic range. The compressed amplitudes are then used to modulate the stimulating pulses.

In this strategy, the window is shifted in such a way that the output rate after decomposition, referred to as analysis rate here, is made higher than or equal to the stimulation rate. Due to the real-time requirement on a PDA platform, which has limited processing and memory resources compared to a PC platform, one faces a limit on the maximum achievable analysis rate. To obtain high-stimulation rates, the analysis frames are often repeated. As stated in [6], this approach does not provide any extra information. In order to provide high-analysis rates, a faster implementation of ACE using recursive DFT was reported in our previous work [4], where the window was shifted by one sample at a time, and hence, the analysis rate was made equal to the input sampling rate. In the recursive-DFT approach, the DFT of a previous window is updated in a computationally efficient manner to compute the DFT of a speech segment based on the FFT values computed in past frames. As discussed in [4], since the implementation is done in fixed-point arithmetic for real-time operation, a resetting is done (and needed) by computing a full FFT every 50 ms to keep the growth of the quantization error within 1% of the floating-point implementation. The block diagram of the *n*-of-*m* ACE strategy is shown in Fig. 2.

III. Wavelet-Based *n*-of-*m* Strategy

In this paper, we present the implementation of the *n*-of-*m* strategy based on the wavelet transform. The wavelet approach can be used to decompose the input speech signal into a number of frequency bands [9]–[13], [19], [20]. Similar to the FFT-based *n*-of-*m* strategy, the number of maximum amplitude channel outputs $N_{\max}$, can be selected for stimulation and compressed using a logarithmic compression map. The block diagram of the wavelet-based *n*-of-*m* strategy is shown in Fig. 3.

A. Wavelet-Packet-Decomposition Tree Structures

Different tree structures can be used for the wavelet decomposition. The tree structure determines the bandwidth and cutoff frequency of each channel. Three different wavelet tree structures are considered in this paper. The wavelet tree structure shown in Fig. 4 appears in [14] and [15], where the frequency axis is divided according to a criticalband-like spacing [14], [16]. The frequency band extending from 0 to 0.015625 in normalized frequency is not used in the computation of the channel envelopes as it contains mostly dc information. In Fig. 4, the numbers next to the tree branches indicate the channel numbers [17].

The decomposed outputs from different branches occur at different frame rates. Hence, they need to be made of the same rate in order to have a constant analysis rate for all channels. This requires that the lower channels, which undergo several filtering stages and downsampling, are interpolated linearly to produce higher analysis rates [10]. In this study, we considered an analysis rate of $F_s/8$, where F_s denotes the input sampling frequency. For an input sampling frequency of 22 kHz, the analysis rate can be maintained at 2756 Hz. Hence, for a stimulation rate lower than 2756 p/s (pulses per second) per channel, the

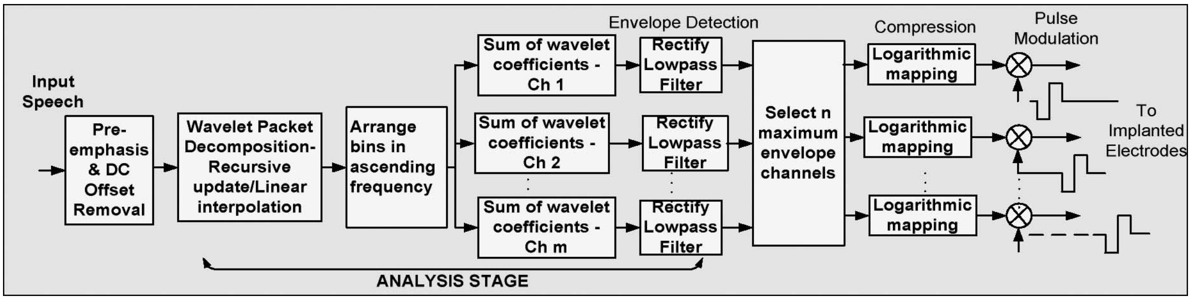

Input Speech
Pre-emphasis & DC Offset Removal
Wavelet Packet Decomposition-Recursive update/Linear interpolation
Arrange bins in ascending frequency
Envelope Detection
Sum of wavelet coefficients - Ch 1
Sum of wavelet coefficients - Ch 2
Sum of wavelet coefficients - Ch m
Rectify Lowpass Filter
Rectify Lowpass Filter
Rectify Lowpass Filter
Select n maximum envelope channels
Compression
Logarithmic mapping
Logarithmic mapping
Logarithmic mapping
Pulse Modulation
To Implanted Electrodes
ANALYSIS STAGE

Fig. 3. Block diagram of the *n*-of-*m* strategy using wavelet packet transform implementation.

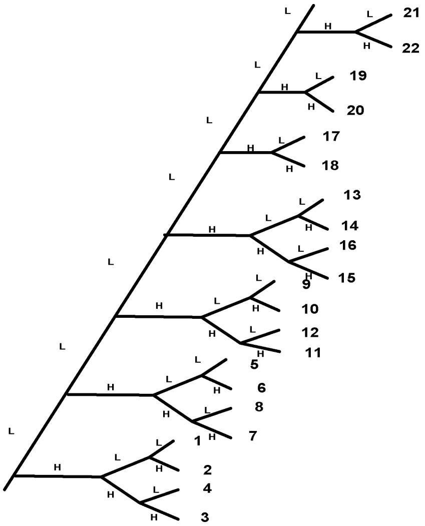


Fig. 4. Wavelet packet decomposition based on tree structure 1.

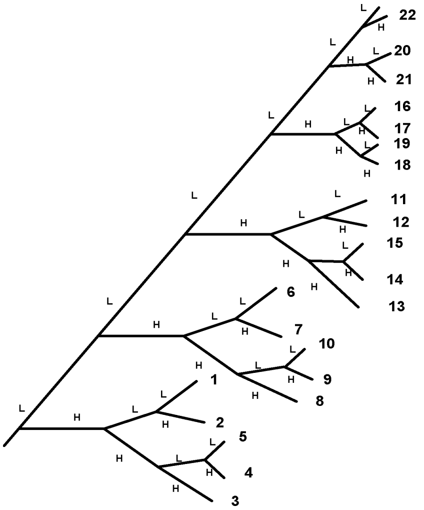


Fig. 5. Wavelet packet decomposition based on tree structure 2.

analysis channel frames are downsampled, while for stimulation rates higher than 2756 p/s, the output needs to be repeated or linearly interpolated.

In order to match the channel spacing of the decomposition tree to that used in the Nucleus family of implants [18], the aforementioned wavelet tree structure is modified to form the wavelet tree structure 2 shown in Fig. 5. The same approach can be applied for other cochlear implant devices. The corresponding channel numbers are shown by the branches [17] (note that low-channel numbers indicate high-frequency channels consistent with the numbering used in Nucleus devices). After the decomposition, the channel outputs are linearly interpolated to obtain the analysis rate of $F_s/8$.

The third wavelet tree structure considered is the binary tree shown in Fig. 6 [9]. In this structure, the branches are arranged according to the frequency bands in ascending order. The numbers next to the branches in Fig. 6 indicate the frequency bands [17]. In this configuration, each branch is treated as a frequency bin of the FFT. A channel output is computed by summing up all the frequency-band outputs falling within the frequency range of that channel. The number of channels can be varied and the frequency range used in the Nucleus device is considered here for grouping the frequency bins into different channels [18]. The produced analysis output at a rate of $F_s/64$ is thus made constant across all channels. With this wavelet-decomposition tree, the bandwidths of the channels become exactly the same as the frequency bands used in the Nucleus device. Clearly, the same approach can be applied to any other implant device with different frequency spacing and different number of electrodes. Table I gives the cutoff frequencies of the channels corresponding to the three tree structures for the input sampling rate of 16 kHz. Using the wavelet tree structure 3, different clinical maps other than the ones listed in the table can be obtained.

B. Types of Wavelets

Two widely used wavelet basis of Daubechies and Symmlet are considered in our implementation. Each family has its own advantages; the former provides a maximally flat response, while the latter is more symmetrical [19]. Three filter lengths of 12, 24, and 30 are included in our implementation. The shorter the response, the higher the aliasing and the lower the processing time.

IV. Recursive-Wavelet Implementation

Due to the need for downsampling at every stage of the wavelet packet tree (WPT), the output rate after the sixth stage becomes 1/64th of the input sampling rate. Generally, the stimulation rate is maintained between 250 to 2400 p/s per

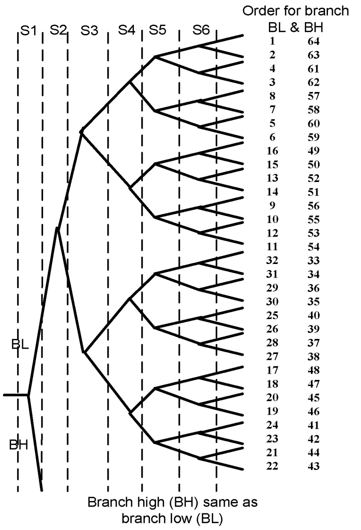


Fig. 6. Wavelet packet decomposition based on tree structure 3.

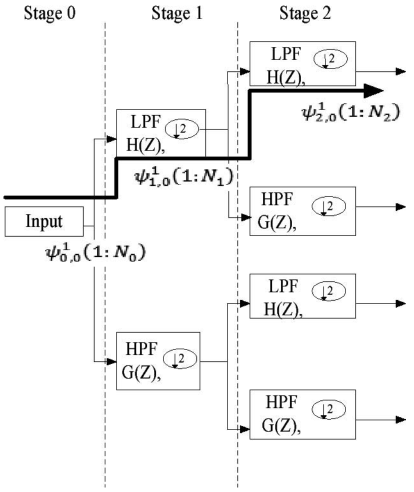


Fig. 7. Two-stage wavelet decomposition with the highlighted branch describing the overlap update.

TABLE I
LOWER AND UPPER CUTOFF FREQUENCIES OF THE CHANNELS CORRESPONDING TO THE THREE WAVELET TREE STRUCTURES (INPUT SAMPLING RATE IS 16 KHZ)

Channel No./ cutoff frequencies Hz	Wavelet tree structure 1	Wavelet tree structure 2	Wavelet tree structure 3 (as in Nucleus)
22	62.5-93.75	125-250	125-250
21	93.75-125	250-375	250-375
20	125-187.5	375-500	375-500
19	187.5-250	500-625	500-625
18	250-375	625-750	625-750
17	375-500	750-875	750-875
16	500-625	875-1000	875-1000
15	625-750	1000-1125	1000-1125
14	750-875	1125-1250	1125-1250
13	875-1000	1250-1500	1250-1500
12	1000-1250	1500-1750	1500-1750
11	1250-1500	1750-2000	1750-2000
10	1500-1750	2000-2250	2000-2250
9	1750-2000	2250-2500	2250-2625
8	2000-2500	2500-3000	2625-3000
7	2500-3000	3000-3500	3000-3500
6	3000-3500	3500-4000	3500-4000
5	3500-4000	4000-4500	4000-4625
4	4000-5000	4500-5000	4625-5250
3	5000-6000	5000-6000	5250-6000
2	6000-7000	6000-7000	6000-7000
1	7000-8000	7000-8000	7000-8000

channel [6], [7] depending on the patient's MAP. Hence, in order to keep the stimulation rate within this range, the decomposed signal has to be either repeated or interpolated, so as to have the analysis rate at least as equal as the stimulation rate. When the stimulus frame is repeated, however, no extra information is provided. To avoid this, the input window is shifted here by one sample at a time and the wavelet coefficients are updated similar to a recursive-DFT approach [4].

When computing the wavelet coefficients within a short-duration window, with the window being shifted by one sample at a time, allows the analysis rate to be equal to the input sampling rate. However, the recomputation of the wavelet coefficients, each time the window is shifted, creates a bottleneck for the real-time operation of the wavelet decomposition on a PDA platform, which has a relatively limited processing power and memory. This real-time challenge is met here by devising a computationally efficient recursive mechanism to compute the new wavelet coefficients based on the previously computed wavelet coefficients.

In what follows, we present the recursive algorithm for the path shown in Fig. 7. The same recursive algorithm is applicable to any other path. The block diagram shown in Fig. 8 provides the updating procedure for two-stage decomposition with one-sample shift in the window position.

The pseudocode for recursively updating wavelet coefficients is given below.

Let $\psi_{j,n}^{p}(1:N_j)$ be a window of N_j samples at *j*th stage and branch number *p* at time instant *n*. At the *j*th stage, there are a total of 2^j branches.

```
Update the input window (shift by one sample)
    for stage_number $s = 1$ to $L_{\mathrm{stage}}$
        for branch_number $p = 1$ to $2^s$
            update circular buffer pointers, such that
                $\psi_{s,n}^{p}(1:N_s-1) = \psi_{s,n-2^s}^{p}(2:N_s)$
            update last sample of window $\psi_{s,n}^{p}(N_s)$
        end
end
```

The following equations give the updates for the $(j+1)$th stage at branches $2p$ and $2p+1$:

$$\psi_{j+1,n}^{2p}(1:N_{j+1}-1) = \psi_{j+1,n-2^{j+1}}^{2p}(2:N_{j+1}) \tag{1}$$

$$\psi_{j+1,n}^{2p+1}(1:N_{j+1}-1) = \psi_{j+1,n-2^{j+1}}^{2p+1}(2:N_{j+1}) \tag{2}$$

$$\psi_{j+1,n}^{2p}(N_{j+1}) = \sum_{k=0}^{l=1} \psi_{j,n}^{p}(N_j-k)h(k) \tag{3}$$

$$\psi_{j+1,n}^{2p+1}(N_{j+1}) = \sum_{k=0}^{l=1} \psi_{j,n}^{p}(N_j-k)g(k). \tag{4}$$

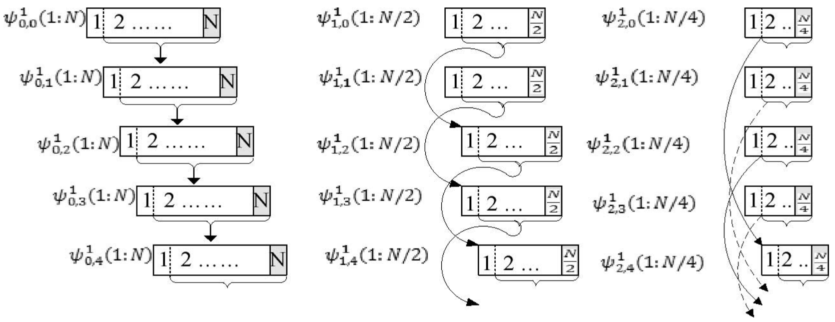


Fig. 8. Wavelet coefficients update for one sample window shift.

Note that for all $j = 0, 1, \ldots, L_{\text{stage}} - 1$, $p = 0, 1, \ldots, 2^j - 1$, $n = 0, 1, \ldots$, and l is the decomposition filter length with filter response h and g, as shown in Fig. 7. For an input speech window of N samples, the number of samples at the jth stage output is $N_j = N/2^j$. It is easy to see that when the window is shifted by one sample, the corresponding windows in all the stages also get shifted by one sample. The time instance at which the window is used to perform the updating depends on the stage. For example, in stage 1, the window preceding the previous window is used to perform the updating. In general, at any given stage, the window used to perform the updating is the window, which occurred 2 to power of (stage number-1) instances before.

In our implementation, 128 sample windows are used to compute the wavelet-decomposition coefficients. For a six-stage decomposition, an input of 128 samples yields an output of 2 samples for every window, which is averaged to give a single output.

Though this update algorithm allows us to run the wavelet strategy in real time on a PC platform, it does not run in real time on a PDA platform for an input sampling rate of 22 kHz or higher. As a result, we have considered a shift of four samples per window in our PDA implementation, as shown in Fig. 9. This way, the processing burden associated with the window updating in every stage is reduced compared to that of the one-sample shift case. Furthermore, the number of windows to be saved in memory is reduced compared to the one-sample shift case. For four-sample shifting windows, which correspond to 97% overlap, the resulting analysis rate is made $F_s/4$. Thus, for a 22-kHz input sampling rate, the analysis rate becomes 5512.5 Hz, which is sufficiently high and much higher than what is used commercially (about 900 p/s per channel).

In general, for a 2^{sh} samples shift with $sh = 1, 2, \ldots$, the update procedure is given by

$$\begin{aligned}\psi^{2p}_{j+1,n}(1 : N_{j+1} - 1) &= \psi^{2p}_{j+1,n-2^{j+1-sh}}(2 : N_{j+1}),\\ \psi^{2p+1}_{j+1,n}(1 : N_{j+1} - 1) &= \psi^{2p+1}_{j+1,n-2^{j+1-sh}}(2 : N_{j+1})\end{aligned} \tag{5}$$

$$\psi^{2p}_{j+1,n}(N_{j+1}) = \sum_{k=0}^{l-1} \psi^{p}_{j,n}(N_j - k)h(k) \tag{6}$$

$$\psi^{2p+1}_{j+1,n}(N_{j+1}) = \sum_{k=0}^{l-1} \psi^{p}_{j,n}(N_j - k)g(k) \tag{7}$$

$$\forall\, j = sh + 1, \ldots L_{\text{stage}} - 1, \qquad p = 0, 1, \ldots, 2^j - 1, \quad n = 0, 1, \ldots \text{ and}$$

$$\psi^{2p}_{j+1,n}(1 : N_{j+1} - 2^{sh-(j+1)}) = \psi^{2p}_{j+1,n-1}(2^{sh-(j+1)} : N_{j+1}) \tag{8}$$

$$\psi^{2p+1}_{j+1,n}(1 : N_{j+1} - 2^{sh-(j+1)}) = \psi^{2p+1}_{j+1,n-1}(2^{sh-(j+1)} : N_{j+1}) \tag{9}$$

$$\psi^{2p}_{j+1,n}(N_{j+1} - \delta) = \sum_{k=0}^{l-1} \psi^{p}_{j,n}(N_j - 2\delta - k)h(k) \tag{10}$$

$$\psi^{2p+1}_{j+1,n}(N_{j+1} - \delta) = \sum_{k=0}^{l-1} \psi^{p}_{j,n}(N_j - 2\delta - k)g(k) \tag{11}$$

$\forall\, j = 0, 1 \ldots, sh$ and $\delta = 0, 1, \ldots 2^{sh-(j+1)} - 1$. For a shift of 2^{sh} samples, the memory required at any stage can be computed as follows:

Number of filters in stage $j = 2^j$

number of windows to be kept in memory for a filter at any branch of stage

$$j = \begin{cases} 1, & j \le sh \\ 2^{j-sh}, & j > sh, \end{cases}$$

number of samples to be buffered in every window

$$= \begin{cases} L - 1 + 2^{sh-j}, & j \le sh \\ L, & j > sh \end{cases}$$

yielding a total of

$$\text{TM} \approx \sum_{j=0}^{sh} \left(2^j(l - 1 + 2^{sh-j})\right) + \sum_{j=sh+1}^{L_{\text{stage}}-1} \left(2^j l 2^{j-sh}\right) \tag{12}$$

where TM indicates the total memory required. Table II provides the computational complexity and memory requirements for a shift of one and four samples with filter lengths of 12, 24, 30, and a six-stage decomposition.

The analysis frame rate can be increased either by overlapping input windows or by linear interpolation of the analysis channel outputs. Fig. 10 provides a comparison of the accuracy of the wavelet coefficients obtained by using 97% overlapping windows and upsampling (by a factor 16) in order to make the

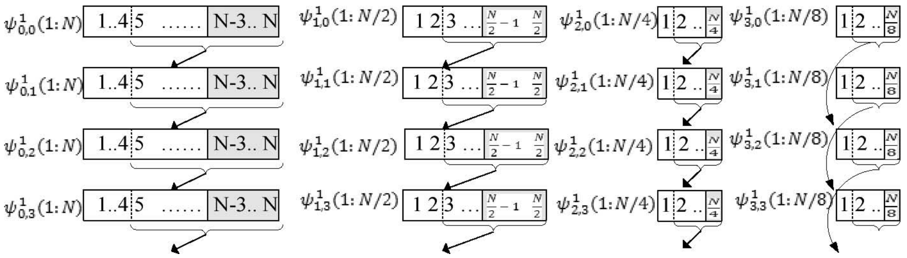


Fig. 9. Wavelet coefficients update for four samples window shift.

TABLE II
COMPUTATIONAL COMPLEXITY AND MEMORY REQUIREMENTS FOR THE WAVELET IMPLEMENTATION WITH AND WITHOUT UPDATING

Strategy used (*m*: Frame length) (*l*:Filter length)	Coefficient update		No coefficient update	
	Shift window by 1 sample	Shift window by 4 samples	Shift window by 1 sample	Shift window by 4 samples
No. of multiplications	(2+4+8+16+32+64) *m***l*= 126 *m***l*	(2*2+4+8+16+32+64) *m***l*/4=32 *m***l*	768 *m***l*	768**l** *m*/4 = 192 *m***l*
Memory requirement (mem)	(1+4+16+64+256+1024) * *l*+ 128 = 1365 *l* + 128	(1+2+4+16+64+256) **l* + 128 = 343 *l* + 128	64*128+(*l* -1) * (63+2*31+4*15+ 8*7+16*3+ 32*1)= 321 (*l* -1) + 8192	64*128 + (*l*-1)* (63+2*31 +4*15+8*7 +16*3 +32*1) = 321 (*l* -1) + 8192
l=30, *m*=256 samples(11.6 ms), fixed-point 16-bit	≈967K multiplies, ≈82KB mem	≈245K multiplies, ≈21KB mem	≈5.9M multiplies, ≈35KB mem	≈1.4M multiplies, ≈35KB mem

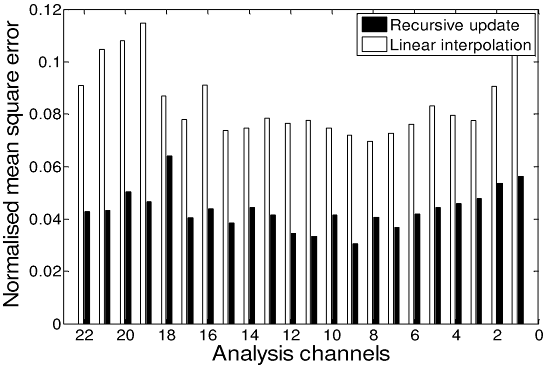


Fig. 10. NMSE of analysis channel outputs obtained from wavelet-based *n*-of-*m* strategy using linear interpolation and wavelet-based *n*-of-*m* strategy using 97% overlap window recursive update.

analysis rate equal to that of the overlapping method. In our implementation, the wavelet tree structure 3 and the Symmlet basis function were used with the filter response of length 30. Both the Symmlet and Daubechies wavelet basis functions produced similar outputs.

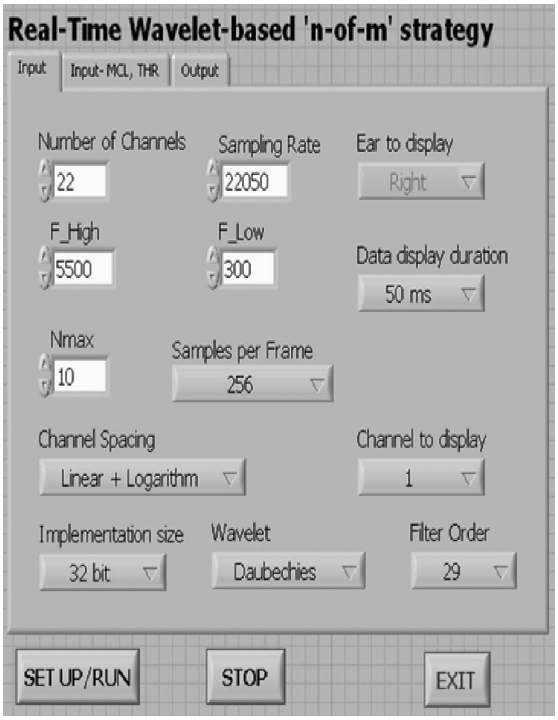
Accuracy is measured in terms of the normalized mse (nmse) between the analysis channel outputs obtained using the earlier two methods and the outputs obtained using FFT. It can be seen that with overlapping windows the analysis channel output gets closer to the one achieved by the FFT approach. The mse is averaged over 30 IEEE sentences [21]. The normalized mse error is computed as follows:

$$\text{mse} = \left(\frac{1}{N * \text{channels}}\right) * \sum_{n=1}^{N} \sum_{\text{ch}=1}^{\text{channels}} \left(\frac{(\varphi_{\text{FFT}}(n,\text{ch}) - \varphi_{\text{WPT}}(n,\text{ch}))^2}{\max(\varphi_{\text{FFT}}(n,\text{ch}),\varepsilon)^2}\right) \quad (13)$$

where N is the total number of samples, φ_{FFT} is the analysis channel output obtained using the FFT, φ_{WPT} is the analysis channel output obtained using the WPT structure 3 with either overlapping windows and recursive update or with nonoverlapping windows and linear interpolation, and $\varepsilon = 0.1\%$ of $\max(\varphi_{\text{FFT}}(n,\text{ch})) \quad \forall n = 1,\ldots,N$.

V. PDA IMPLEMENTATION

The software implementation was done using hybrid LabVIEW programming. Graphical programming in LabVIEW was used in order to achieve user interactivity and real-time display. The signal-processing functions were programmed in C and inserted as dynamic-link libraries (DLL) within LabVIEW. Several optimizations steps were done to achieve real-time throughput. As discussed in [4] and [22], these steps consisted of using fixed-point arithmetic, applying efficient memory management, and using built-in functions. PDA platforms often use fixed-point processors. Thus, to avoid the computational burden associated with performing floating-point operations on fixed-point



Real-Time Wavelet-based 'n-of-m' strategy

Number of Channels 22
Sampling Rate 22050
Ear to display Right
F_High 5500
F_Low 300
Data display duration 50 ms
Nmax 10
Samples per Frame 256
Channel Spacing Linear + Logarithm
Channel to display 1
Implementation size 32 bit
Wavelet Daubechies
Filter Order 29

Fig. 11. Front panel of the interactive wavelet program running in real time on a PDA platform.

processors, all the operations were converted into fixed point using either 16-bit or 32-bit word size. All the filters used to compute the wavelet coefficients were finite-impulse response (FIR) filters, which avoided the quantization error growth problem encountered in the recursive-DFT approach.

Static memory allocation was done in place of dynamic memory allocation. Global variables were used instead of passing the parameters to every function. Several circular buffers of powers of 2 length were used for the updates. Since the output at every stage had to be saved in memory and used for updating, the number of windows to save in memory differs from stage to stage. The use of circular buffers at every stage for every branch helped to efficiently update the pointers.

The developed interactive front panel of the LabVIEW program or the graphical user interface is shown in Fig. 11. The PDA is connected to a Freedom cochlear implant emulator coil using a secure digital input output (SDIO) interface board, which we previously reported in [23] and [24]. Researchers can change the number of analysis channels, most comfortable levels, threshold levels, number of maximum amplitude channels, filter spacing, wavelet family, and filter length. They can also choose to observe the captured speech signal and analysis channel outputs. Since displaying a large buffer on the front panel is time-consuming, the buffer is downsampled and the panel is refreshed once every 500 ms. The length of the buffer to be displayed can be chosen out of two options of 50 or 100 ms. When using the program in the binaural mode, researchers can choose either the right implant, the left implant or both.

VI. Results and Discussion

In this section, we compare the performance of the proposed recursive wavelet-based *n*-of-*m* strategy, and the recursive DFT-based *n*-of-*m* and CIS strategies, in terms of computational complexity, spectral leakage, fixed-point accuracy, and real-time processing.

A. Computational Complexity

The three wavelet tree structures differ from each other in the way the channel spacing is implemented. The number of computations required for the first wavelet tree structure is the smallest among the three. Table III provides the computational complexity of the three WPT structures. In Table III, the number of real multiplications required to decompose the signal using the CIS, FFT-based *n*-of-*m*, recursive DFT-based *n*-of-*m*, and recursive wavelet-based *n*-of-*m* strategies for the three tree structures using the three different decomposition filter lengths are listed. For the CIS strategy, sixth-order bandpass filters and second-order lowpass filters are considered. The number of analysis channels is taken to be 22 for all the strategies to ensure a fair comparison. As seen from Table III, the use of the recursive wavelet-based *n*-of-*m* strategy requires fewer computations than the recursive DFT-based *n*-of-*m* strategy.

B. Spectral Leakage

The main function of the cochlear implant signal processor is to divide the input speech signal into a number of frequency bands and to extract the signal strength (envelope) in each band, so that the corresponding electrode can be stimulated accordingly. A good measure of performance is spectral leakage. This measure indicates how much or what percentage of the energy in one band is leaked into adjacent bands [20], [25].

To measure the spectral leakage, we conducted a test in which a synthetic signal (e.g., tone) of known frequency was used to serve as the input signal. This signal was decomposed into 22 bands using the wavelet-packet-decomposition algorithm and the signal strength was measured in each band. The ratio of the power distributed in bands other than the band, where the signal exists to the total power was defined to be the leakage factor (LF). This factor provides a measure of how much energy is leaked into other bands. Hence, the lower the LF, the better the performance in terms of providing good frequency specificity. The LF for channel "ch" was calculated as per the following equation:

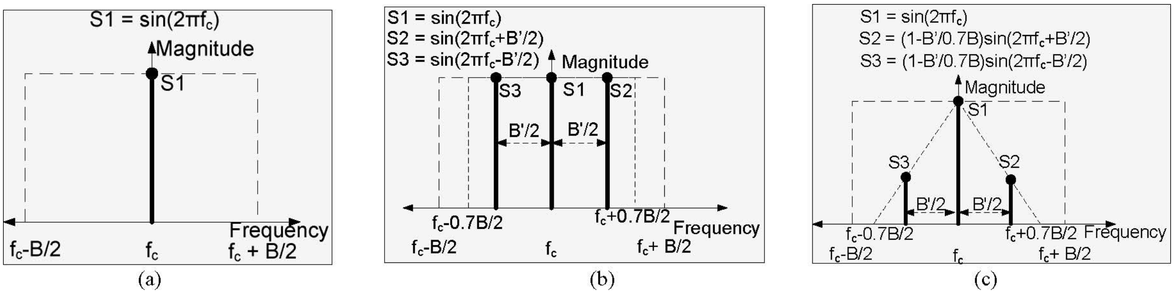
$$\mathrm{LF} = 100 \times \left(\frac{\sum_{n=1}^{N} \sum_{k \neq \mathrm{ch}}^{\mathrm{channels}} \left(\varphi(n,k)^2 \right)}{\sum_{n=1}^{N} \sum_{k=1}^{\mathrm{channels}} \varphi(n,k)^2} \right) \tag{14}$$

where $\varphi(n,k)$ is the analysis channel output of channel *k* at time instant *n* and *N* is the total number of samples in the test input.

Three different sets of inputs were used to measure spectral leakage. The first one was a sinusoidal signal with the frequency coinciding with the center frequency of the band, as indicated in Fig. 12(a). The second was the sum of three sinusoidal signals, with one signal having a frequency equal to the center frequency of the band. The other two sinusoids were randomly

TABLE III
COMPUTATIONAL COMPLEXITY OF DIFFERENT STRATEGIES

Approximate number of real multiplications for m=256 sample frames, 128-point FFT, 22 channels, and l = 12, 24 & 30 length wavelet packet filter	Filter-bank	FFT (N-point)	Recursive DFT update	Wavelet Packet (Symmlet/Daubechies)								
				Wavelet packet tree 1			Wavelet packet tree 2			Wavelet packet tree 3		
	channels* (6th order BPF)	$2N$ ($\log_2 N$) + $(N/2)*2$	$(N/2)*4$ + $(N/2)*2$	$m*l$ (2+1+0.5+0.25) + $3m*l/32$ + $3m*l/64$ + $3m*l/128$ +$5m/8$*channels			$2.125m*l$ (1+0.5+0.25)+ $2m*l/8$+1.25 $m*l/16$ + 0.25 $m*l/32$ +$5m/8$*channels			$m*l$ *(6) + $m/8$*channels*5		
	112K	519K	100K	15K	28K	33K	16K	29K	35K	22K	40K	50K



(a) (b) (c)

Fig. 12. (a) Input 1 = $S1$. (b) Input 2 = $S1 + S2 + S3$. (c) Input 3 = $S1 + S2 + S3$.

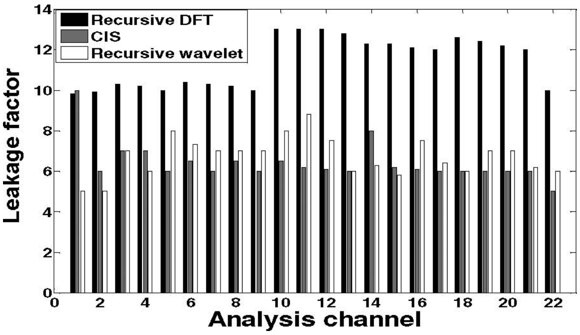


Fig. 13. Comparison of LF for recursive DFT, CIS, and recursive-wavelet (tree structure 3) implementations.

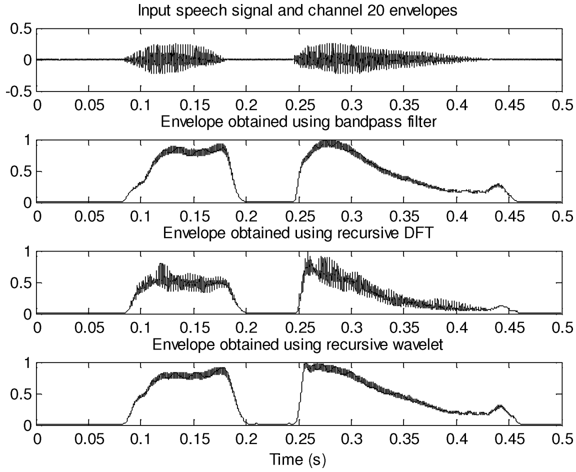


Fig. 14. Channel 20 envelope obtained using the bandpass filter, recursive-DFT, and recursive-wavelet strategies for the syllable "asa."

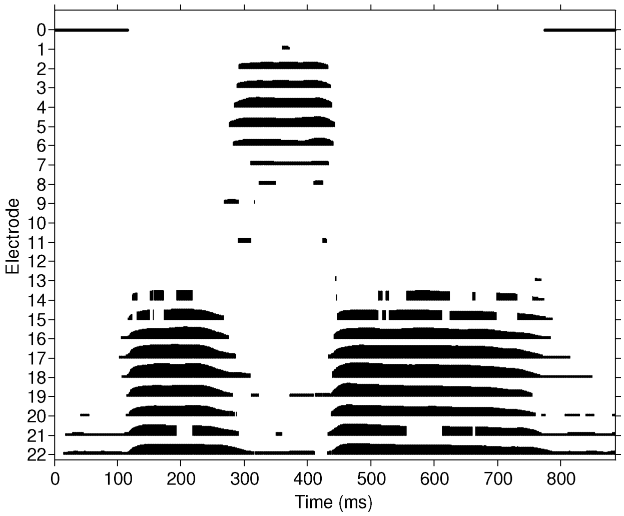


Fig. 15. Electrodogram of syllable "asa" using the recursive wavelet based n-of-m strategy (channel 1 represents most basal channel and channel 22 represents most apical).

distributed within 70% of the bandwidth of that channel, as shown in Fig. 12(b). The amplitudes of all the three signals were considered to be of the same strength. The third set was similar to the second set, but their amplitudes were varied inversely proportional to the distance from the center frequency, as shown in Fig. 12(c). Fifty different realizations were considered for the second and third sets. This was done for all 22 bands.

The floating-point implementation was used to compute the LF. After averaging the LF over all the realizations, the LFs for the recursive DFT and FFT implementations were found

TABLE IV
NMSE [SEE (15)] BETWEEN FLOATING-POINT AND FIXED-POINT IMPLEMENTATIONS OF THE RECURSIVE-WAVELET STRATEGY

Method	Wavelet packet tree 1		Wavelet packet tree 2		Wavelet packet tree 3	
Normalized MSE (%)	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
	0.67	≈0.05	0.71	0.05	0.9	0.07

TABLE V
REAL-TIME PROCESSING TIME FOR 11.6 MS INPUT SPEECH FRAMES AND 22 ANALYSIS CHANNELS ON A PDA PLATFORM

Strategy	n-of-m using recursive DFT	n-of-m using wavelet packet decomposition tree structure with filter length of 30 (non-overlapping windows)			n-of-m using recursive wavelet update for wavelet packet decomposition tree structure 3 with filter length equal to:		
Processing time in ms		tree 1	tree 2	tree 3	12	24	30
	2	≈ 0.5	≈0.5	≈1	5	8	10

to be nearly the same. Also, it is worth pointing out that both Daubechies and Symmlet wavelets produced more or less the same leakage. Fig. 13 shows the results expressed in terms of the LF [see (14)]. As can be seen from Fig. 13, the recursive wavelet-based *n*-of-*m* strategy yielded a lower LF than that obtained with the recursive DFT-based strategy. The LF was found to be comparable to CIS.

C. Tracking Temporal Features

Fig. 14 illustrates a sample envelope output of channel 20 for syllable "asa" spoken by a male speaker using the wavelet-based *n*-of-*m* strategy. The extracted envelopes using the band-pass filter and recursive-DFT strategies are superimposed for comparison. As can be seen, the wavelet-based *n*-of-*m* strategy performs on par with the other strategies as far as being able to track temporal envelope features. Fig. 15 shows the electrodogram for the syllable "asa." The decomposition was done by the recursive-wavelet method using the wavelet tree structure 3. In the *n*-of-*m* strategy, eight maximum amplitude analysis channels were selected out of 22.

D. Fixed-Point Accuracy

To assess the accuracy of the fixed-point implementation, we computed the averaged nmse between the envelope amplitude values computed using the fixed-point and floating-point implementations

$$\mathrm{nmse}(\%) = \frac{100}{N * \mathrm{channels}} \times \sum_{n=0}^{N} \sum_{\mathrm{ch}=1}^{\mathrm{channels}} \left(\frac{(\varphi_{\mathrm{FLTP}}(n,\mathrm{ch}) - \varphi_{\mathrm{FIXP}}(n,\mathrm{ch}))^2}{\max(\varphi_{\mathrm{FLTP}}(n,\mathrm{ch}), \varepsilon)^2} \right) \quad (15)$$

where φ_{FLTP} and φ_{FIXP} are the analysis channel output obtained using the floating-point and fixed-point implementations, respectively, *N* is the total number of samples in the sentence, and $\varepsilon = 0.1\%$ of $\max(\varphi_{\mathrm{FLTP}}(n,\mathrm{ch})) \quad \forall n = 1, \ldots, N$.

Table IV shows the comparison of the nmse [see (15)] between the floating-point, and 32-bit and 16-bit fixed-point analysis channel outputs of the three-wavelet tree structures for the case of four-sample shift per window. As can be seen from Table IV, the mse remains below 1%. The mse was obtained by averaging over all the 22 channels for 30 IEEE sentences.

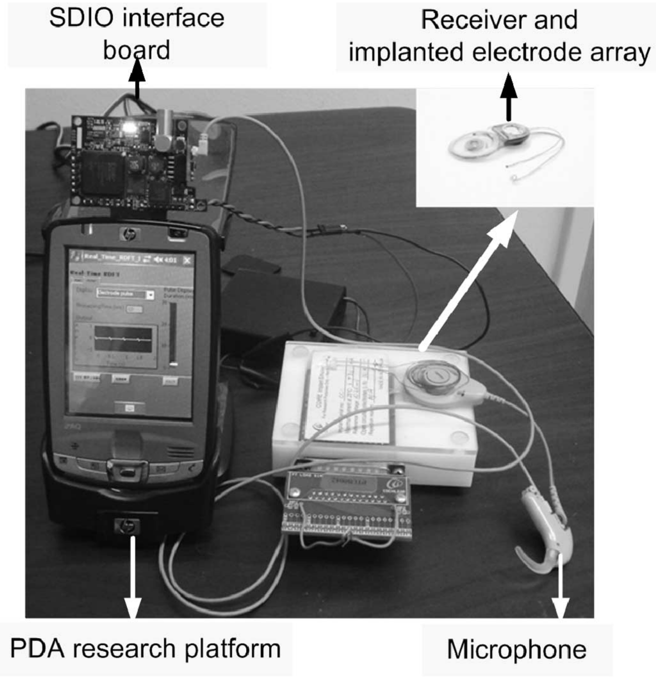


Fig. 16. PDA-based research platform designed for cochlear implant studies.

E. Real-Time Processing

Table V shows the actual time required to process speech frames of 11.6 ms duration (22 kHz sampling rate) on a typical PDA with a clock frequency of 624 MHz (see Fig. 16 for test setup). The PC version of the program took less than 1 ms at a clock frequency of 3.2 GHz. The timings reported in columns 1 through 4 correspond to the recursive DFT-based ACE, recursive wavelet-based *n*-of-*m* using the WPT structures 1, 2, and 3 with nonoverlapping windows and with the filter length of 30 on a PDA platform. The last 3 columns show the timings obtained using the third tree structure with 97% overlapping windows and filter lengths of 12, 24, and 30. As can be seen, real-time implementation of the recursive-wavelet approach was accomplished with all filter lengths. It is worth pointing out that timings shown in Table V include the decomposition, rectification, low-pass filtering, selection of $N_{\max}$ amplitudes, and logarithmic compression for the 16-bit fixed-point implementation with 22 analysis channels.

VII. Conclusion

In this paper, a recursive wavelet-based *n*-of-*m* strategy was proposed as an alternative to the existing ACE strategy that is being used in commercial implants (e.g., Freedom device from Cochlear Corporation). The real-time implementation of this strategy on a PDA platform was also demonstrated. It was shown that the proposed strategy achieves lower computational complexity as well as lower spectral leakage compared to the recursive DFT-based implementation. Due to these advantages, it is expected that future generations of cochlear implants will deploy the proposed recursive wavelet-based strategy.

Acknowledgment

Interested readers may contact Dr. P. Loizou (contract PI) at loizou@utdallas.edu for a copy of the LabVIEW PDA code.

References

[1] National Institute on Deafness and Other Communication Disorders, "Cochlear Implants," National Institutes of Health. (2009, Aug.). publication no. 09-4798 [Online]. Available: http://www.nidcd.nih.gov/health/hearing/coch.asp

[2] P. Loizou, "Mimicking the human ear," *IEEE Signal Process. Mag.*, vol. 15, no. 5, pp. 101–130, Sep. 1998.

[3] B. H. Bonham and L. M. Litvak, "Current focusing and steering: Modeling, physiology, and psychophysics," *Hear. Res.*, vol. 242, no. 1–2, pp. 141–153, 2008.

[4] V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou, "Real-time PDA-based recursive Fourier transform implementation for cochlear implant applications," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 2009, pp. 1333–1336.

[5] B. S. Wilson, C. C. Finley, D. T. Lawson, R. D. Wolford, D. K. Eddington, and W. M. Rabinowitz, "Better speech recognition with cochlear implants," *Nature*, vol. 352, pp. 236–238, 1991.

[6] P. Loizou, "Speech processing in vocoder-centric cochlear implants," *Cochlear and Brainstem Implants., Adv. Otorhinolaryngol*, vol. 64, pp. 109–143, 2006.

[7] A. Vandali, L. Whitford, K. Plant, and G. Clark, "Speech perception as a function of electrical stimulation rate using the Nucleus 24 cochlear implant system," *Ear Hear*, vol. 21, no. 6, pp. 608–624, 2000.

[8] W. Nogueira, A. Buchner, T. Lenarz, and B. Edler, "A psychoacoustic "NofM"-type speech coding strategy for cochlear implants," *EURASIP J. Appl. Signal Process.*, vol. 18, pp. 3044–3059, 2005.

[9] V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou, "Real-time implementation of wavelet-based advanced combination encoder on PDA platforms for cochlear implant studies," presented at the IEEE Int. Conf. Acoust., Speech Signal Process., Dallas, TX, 2010.

[10] A. Paglialonga, G. Tognola, G. Baselli, M. Parazzini, P. Ravazzani, and F. Grandori, "Speech processing for cochlear implants with discrete wavelet transform: Feasibility study and performance evaluation," in *Proc. IEEE EMBS Conf.*, 2006, pp. 3763–3766.

[11] A. Derbel, M. Ghorbel, M. Samet, and A. Hamida, "Implementation of strategy based on auditory model based wavelet transform speech processing on DSP dedicated to cochlear prosthesis," in *Proc. 4th Int. Symp. Comput. Intell. Intell. Inf.*, 2009, pp. 143–148.

[12] G. Tian, Y. Shuli, and Y. Datian, "Application of wavelet in speech processing of cochlear implant," in *Proc. IEEE EMBS Conf.*, 2005, pp. 5339–5342.

[13] K. Nie, N. Lan, and S. Gao, "Wavelet-based feature extraction of speech signal for cochlear implants," in *Proc. IEEE EMBS Conf.*, 1999, vol. 1, p. 654.

[14] D. Sinha and H. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3463–3479, Dec. 1993.

[15] S. Krimi, K. Ouni, and N. Ellouze, "Realization of a psychoacoustic model for MPEG1 using gammachirp wavelet transform," presented at the 13th Eur. Signal Process. Conf. (EUSIPCO 2005), Antalya, Turkey.

[16] X. Han and K. Nie, "Implementation of spectral maxima sound processing for cochlear implants by using Bark scale frequency band partition," in *Proc. IEEE EMBS Conf.*, 2001, vol. 2, pp. 2097–2101.

[17] X. Zeng, W. Zhao, and J. Sheng, "Corresponding relationships between nodes of decomposition tree of wavelet packet and frequency bands of signal subspace," *Acta Seismologica Sinica*, vol. 21, no. 1, pp. 91–97, 2008.

[18] *Software User Manual, N95246F Issue 1,* Nucleus MATLAB Toolbox 2.11, Cochlear Corporation, Lane Cove, NSW, 2002.

[19] W. Nogueira, A. Giese, and B. Buchner, "Wavelet packet filterbank for speech processing strategies in cochlear implants," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 2006, vol. 5, pp. 121–124.

[20] A. Derbel, F. Kalel, B. Hamida, and M. Samet, "Wavelet filtering based on Mellin transform dedicated to cochlear prostheses," in *Proc. IEEE EMBS*, 2007, pp. 1900–1903.

[21] "IEEE Recommended Practice for Speech Quality Measurements," *IEEE Trans. Audio Electroacoust.*, vol. 17, no. 3, pp. 225–246, Sep. 1969.

[22] V. Peddigari, N. Kehtarnavaz, and P. Loizou, "Real-time LabVIEW implementation of cochlear implant signal processing on PDA platforms," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 2007, vol. 2, pp. 356–360.

[23] A. Lobo, H. Lee, S. Guo, N. Kehtarnavaz, V. Peddigari, M. Torlak, D. Kim, L. Ramanna, S. Ciftci, P. Gilley, A. Sharma, and P. Loizou. (2007). Interface board development for Freedom cochlear implant, single-channel stimulator for animal studies and post processing of CAEPs, Fifth quarterly progress report NIHN01DC60002. [Online]. Available: http://utd.edu/∼loizou/cimplants/

[24] D. Kim, A. Lobo, R. Ramachandran, N. Gunupudi, V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou. (2008). Hardware testing of revised SDIO board (v.3), integration of LabVIEW code with SDIO board, Eleventh quarterly progress report NIHN01DC60002. [Online]. Available: http://utd.edu/∼loizou/cimplants/

[25] I. Cheikhrouhou, R. B. Atitallah, K. Ouni, B. Hamida, N. Mamoudi, and N. Ellouze, "Speech analysis using wavelet transforms dedicated to cochlear prosthesis stimulation strategy," in *Proc. First Int. Symp. Control, Commun. Signal Process.*, 2004, pp. 639–642.

[26] A. Lobo, P. Loizou, N. Kehtarnavaz, M. Torlak, H. Lee, A. Sharma, P. Gilley, V. Peddigari, and L. Ramanna, "A PDA-based research platform for cochlear implants," in *Proc. IEEE EMBS Conf. Neuroeng.*, 2007, pp. 28–31.

Vanishree Gopalakrishna received the B.E. degree in electronics and communication from Visvesvaraya Technological University, Karnataka, India, in 2005, the M.S. degree in electrical engineering from the University of Texas at Dallas (UTD), Richardson, TX, in 2008, where she is currently working toward the Ph.D. degree.

She is also a Research Assistant in the Department of Electrical Engineering, UTD. Her current research interests include real-time speech processing for cochlear implants on PDA platform and pattern recognition.

Nasser Kehtarnavaz (S'83–M'86–SM'92) received the Ph.D. degree in electrical and computer engineering from Rice University, Houston, TX, in 1987.

He is currently a Professor in the Department of Electrical Engineering and the Director of the Signal and Image Processing Laboratory, University of Texas at Dallas, Richardson, TX. He is also Co-Editor-in-Chief of the *Journal of Real-Time Image Processing*. He has authored or coauthored more than 200 papers and seven books covering real-time processing aspects of signal and image processing. His research interests include image processing, real-time signal and image processing, biomedical image analysis, and pattern recognition.

Dr. Kehtarnavaz is currently the Chair of the Dallas Chapter of the IEEE Signal Processing Society and a Distinguished Lecturer of IEEE Consumer Electronics Society.

Philipos C. Loizou received the B.S., M.S., and Ph.D. degrees from Arizona State University (ASU), Tempe, AZ, in 1989, 1991, and 1995, respectively, all in electrical engineering.

From 1995 to 1996, he was a Postdoctoral Fellow in the Department of Speech and Hearing Science, ASU, where he was involved in research related to cochlear implants. From 1996 to 1999, he was an Assistant Professor at the University of Arkansas at Little Rock, Little Rock, AR. He is currently a Professor and holds the Cecil and Ida Green Chair in the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX. He is the author of the book *Speech Enhancement: Theory and Practice* (Boca Raton, FL: CRC, 2007) and the coauthor of the textbooks *An Interactive Approach to Signals and Systems Laboratory* (Austin, TX: National Instruments, 2008), and *Advances in Modern Blind Signal Separation Algorithms: Theory and Applications* (San Raphael, CA: Morgan and Claypool Publishers, 2010). His research interests include the areas of signal processing, speech processing, and cochlear implants.

Dr. Loizou is a Fellow of the Acoustical Society of America. He is currently an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING and *International Journal of Audiology*. He was an Associate Editor of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, from 1999 to 2002 and of the IEEE SIGNAL PROCESSING LETTERS, from 2006 to 2008. From 2007 to 2009, he was a member of the Speech Technical Committee of the IEEE Signal Processing Society.