This paper presents the real-time implementation of wavelet-based Advanced Combination Encoder on PDA platforms for cochlear implant studies. Three real-time implementations using the conventional FFT, our previous recursive DFT, and the wavelet transform are compared in terms of computational complexity, processing time and fixed-point accuracy. The results obtained show that this new real-time implementation on PDA platforms is computationally comparable to our previous real-time recursive DFT implementation while achieving a higher accuracy or a lower quantization error.

Index Terms—Cochlear implant signal processing, real-time signal processing, wavelet packet transform, PDA platforms, Advanced Combination Encoder.

1. INTRODUCTION

Cochlear implants are prosthetic devices which provide a sensation of sound to profoundly deaf people. More than 112,000 people around the world have been fitted with cochlear implants, as of 2006 [1]. Some of the components of a cochlear implant reside outside the body and some reside inside. The outside components consist of a microphone, a speech processor, and a transmitter. The inside components consist of a surgically implanted receiver and an electrode array in the cochlea to stimulate the auditory nerves. The speech captured by the microphone gets decomposed into a number of channels or frequency bands by the speech processor. Two commonly used signal processing strategies in commercial cochlear implants are the Continuous Interleaved Sampling (CIS) and Advanced Combination Encoder (ACE) [2], [6] strategies. The former strategy uses a bank of bandpass filters for signal decomposition while the latter uses the short-time Fourier transform [2].

A low-cost portable real-time solution has been developed by our research team for cochlear implant studies by using the PDA platform [3]. This platform allows researchers to easily and interactively examine signal processing algorithms. Noting that PDA platforms are limited in their processing and memory capabilities as compared to PC platforms, the challenge has been to implement the above signal processing strategies in real-time on these limited resource platforms. Since the real-time implementation of the FFT-based ACE running at high analysis rates presents a challenge on existing PDA platforms, we presented in [4] a recursive DFT-based ACE which can be run in real-time on PDAs at high analysis rates. In this paper, we present a real-time implementation of ACE on PDAs based on wavelet packet transform in place of FFT. This implementation has the advantage of higher accuracy as compared to the recursive DFT implementation.

Section 2 provides an overview of the traditional ACE signal processing strategy based on FFT and our previous work based on recursive DFT. Section 3 describes how wavelet packet transform is used in place of FFT. In section 4, the experimental set up along with a comparison of the three implementations (FFT, recursive DFT, and wavelet) in terms of computational complexity, frame processing time, and accuracy of analysis channels is presented. Finally, the conclusion is stated in section 5.

2. ADVANCED COMBINATION ENCODER (ACE) STRATEGY

The signal processor in a cochlear implant divides the input speech signal into a number of frequency bands and extracts the energy in each band. The number of bands varies from 12-22 depending on the implant. The signal decomposition in the ACE strategy [6] is normally implemented by using the short-time Fourier transform. Figure 1(a) shows the block diagram of the ACE signal processing, where the captured input speech signal is windowed. Often, a 128-point Hanning window is used. FFT is then computed on the windowed signal, and the power of all frequency bins falling in the frequency range of an analysis channel is summed up to produce the output of that channel. Among the analysis channel outputs, a set of maximum amplitude channels are selected for stimulation which are then compressed using a logarithmic compression map to fit within the electrical dynamic range. The process is repeated on subsequent speech frames obtained by shifting a window. The window is shifted such that the output analysis rate corresponds to the required stimulation rate or the maximum...
achievable stimulation rate such that the processing can be done in real-time.

As discussed in [4], an interactive implementation of ACE using FFT cannot be achieved in real-time on a PDA platform when the window is required to be shifted by one sample. As a solution, when the required stimulation rate is higher than the analysis frame rate, the analysis frames are repeated. As studies have shown, e.g. [2], due to the repetition of analysis frames, stimulation frames do not provide any extra information. In [4], we adopted a recursive computation of DFT to allow updating DFT per one sample window shift. In recursive DFT, the redundancy in the input speech frame is utilized when the window is shifted by one sample. Rather than computing the DFT based on all the samples in a frame, the DFT for the previous window is updated by carrying out only few computations. As a result, the processing time is reduced enabling the implementation to run in real-time on a PDA platform.

3. ACE VIA WAVELET PACKET TRANSFORM

As shown in Figure 1(b), in place of FFT, wavelet packet transform can be used to decompose the input speech signal into frequency bands. When using the wavelet packet transform across 6 stages, the signal can be divided into 64 bands as shown in Figure 2. A dyadic wavelet packet transform is applied at every stage to produce a binary wavelet packet tree. At the end of the 6th stage, the branches in the wavelet packet tree are arranged in ascending order according to the frequency bands.

Each branch of the wavelet decomposition can be compared to the first N/2 points of an N-point FFT. After arranging them in ascending order, the power in adjacent bands falling in the frequency range of a channel is summed up to produce the analysis channel output. Here the channel spacing is done according to the ACE strategy used in the commercial cochlear implant Nucleus [2]. Due to the downsampling at every stage, the analysis frame length gets reduced. In order to obtain a stimulation rate in the range of 250 pps (pulses per second) to 2400 pps per channel [2], [6], it is required to have 75% overlap between consecutive windows of the input speech signal. It is also possible to achieve these stimulation rates with no overlap by carrying out a linear interpolation procedure. In our implementation, the input sampling frequency is considered to be 22050 Hz and thus the analysis channel output rate after the 6-stage decomposition with no window overlap and no interpolation becomes 344.5 Hz. With 75% overlapped windows requiring no interpolation, the analysis frame rate becomes 1378 Hz. The analysis channel output is passed through a rectifier and a second order IIR lowpass filter to smooth the analysis channel output envelopes.

The implemented program has the option to use either Daubechies or Symmlet wavelets. The first type has the advantage of higher frequency resolution, and maximally flat response, and the second is almost symmetrical giving a linear phase response which is more desirable in speech processing [7]. The length of the filters used for decomposition can be varied among the following three options: 11th, 23rd, and 29th order. The higher the filter order, the lower is the aliasing effect, but at the cost of higher processing time.

4. REAL-TIME IMPLEMENTATION ON PDA PLATFORMS

The implementation was done using the LabVIEW graphical programming environment in a hybrid mode, where the core signal processing functions were done in C and built as Dynamic Link Library (DLL) within the interactive graphical environment of LabVIEW. Several optimizations were applied in order to have a solution running in real-time on the PDA platform. As the solution has to run on the fixed-point processor of a PDA, floating-point computations were converted into fixed-point. Two fixed-point implementations were done, one using 16-bit and the other using 32-bit fixed-point format. The optimization steps discussed in [5] were also taken. These steps consisted of memory management including initialization of buffers, and appropriately choosing the number of local functions and parameters passed to them.

In our implementation, the interactive Front Panel feature of LabVIEW allows for on-the-fly adjustment of the following parameters: most comfortable level, threshold level, number of analysis channels and number of maximum amplitude channels as depicted in Figure 3. In addition, one has the option of choosing the implementation word size between 16 bits and 32 bits, the wavelet type between Daubechies and Symmlet, and the filter order of decomposition filters. Users can view any one of these three outputs: incoming speech signal, analysis channel output before compression, and simulated electrode pulses.

One problem with the fixed-point implementation is the introduction of quantization error. In the recursive DFT approach, every time the DFT is updated, the quantization error gets increased steadily. To address this issue, the recursive DFT computation is reset every 50ms by computing FFT in order to keep the quantization error under 1 percent of the floating-point implementation [4]. When using the wavelet transform implementation, such increase of quantization error with time is avoided. Figure 4 shows the quantization error for the recursive DFT implementation versus the wavelet packet tree implementation for two analysis channels. Quantization error is measured in terms of the mean squared error in dB between the fixed-point and the floating-point analysis channel outputs. The quantization error when using the wavelet transform remains less than -100 dB which is much smaller than -6db when using the recursive DFT with reset.
Even though the number of filters increases in every stage due to the downsampling of the output at every stage, the number of computations gets considerably reduced. Table 1 provides a comparison of the computational complexity in terms of real multiplications and the data memory required for the 16-bit fixed-point implementation for the FFT, recursive DFT and wavelet packet implementations. It can be seen that for the wavelet packet decomposition with 29th order filters, the number of real multiplications is approximately one half of that of the recursive DFT, and the memory required for all the methods are less than 1K bytes. Table 2 shows the division of processing time in percentage among the required sub-processes in ACE. As expected, most of the processing time is consumed by the wavelet transform. Table 3 exhibits the time to process 11.6 ms speech frames on a 625MHz PDA with a sampling frequency of 22050 Hz and 22 analysis channels. The processing time includes the time required to produce the analysis channel output, linear interpolation in case of non-overlapping windows, lowpass filtering, selecting maximum amplitude channels and compressing the output using the logarithmic compression map. As can be seen from Table 3, the wavelet packet implementation is computationally comparable or more efficient than our previous recursive DFT implementation while achieving higher accuracy, i.e., smaller quantization error.

5. CONCLUSION
In this paper, the real-time implementation of the ACE[6] signal processing strategy in cochlear implants is accomplished on PDA platforms. Such platforms have limited resources which make the real-time implementation challenging. This is the first time such a wavelet-based implementation on PDA platforms is reported. This real-time implementation avoids the growing quantization error problem of our previous recursive DFT implementation [4].

6. ACKNOWLEDGMENTS
This work has been funded by the NIH contract N01-DC-6-0002 from NIDCD. Interested readers may contact Dr. Philip Loizou (contract PI) at loizou@utdallas.edu for a copy of the LabVIEW PDA code.

7. REFERENCES

<table>
<thead>
<tr>
<th>No. of real multiplications &amp; memory for 1024 sample speech frame, 16-bit fixed-point (F : frame length, L : filter response length) (mul: multiplications mem: memory)</th>
<th>FFT (N = 128-point)</th>
<th>Recursive DFT</th>
<th>Wavelet Packet (Daubechies/Symmlet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>≈2N</td>
<td>≈3N</td>
<td>L * (no of stages) + ∑k=1^no.of.stages 2^k</td>
<td></td>
</tr>
<tr>
<td>≈2M (mul)</td>
<td>≈400K (mul)</td>
<td>≈73K (mul)</td>
<td></td>
</tr>
<tr>
<td>≈512B (mem)</td>
<td>≈768B (mem)</td>
<td>≈678B (mem) (11th order)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>≈846B (mem) (23rd order)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>≈184K (mem) (29th order)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Comparison of computational complexity between FFT, recursive DFT and wavelet packet implementation of ACE.

<table>
<thead>
<tr>
<th>Subprocess</th>
<th>Wavelet decomposition with overlap</th>
<th>Arrange ascending order, combine, rectify</th>
<th>Lowpass filter</th>
<th>Select n maximum amplitudes</th>
<th>Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of time</td>
<td>80</td>
<td>3</td>
<td>1.5</td>
<td>12</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Table 2: Percentage processing time for different ACE sub-processes using wavelet packet transform.
Specifications
22 channels, Frame Length 256 samples = 11.6 ms, FFT: 128-point, Wavelet Packet 29th order decomposition filter

<table>
<thead>
<tr>
<th></th>
<th>FFT</th>
<th>Recursive DFT</th>
<th>Wavelet Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Using C (Non-interactive) 16-bit</td>
<td>Using LabVIEW + C (Interactive)</td>
<td>Using LabVIEW + C (Interactive)</td>
</tr>
<tr>
<td>16-bit</td>
<td>2ms</td>
<td>2.5ms</td>
<td>2.35ms</td>
</tr>
<tr>
<td>32-bit</td>
<td>&lt;1ms</td>
<td>3.2ms</td>
<td>1.3ms</td>
</tr>
</tbody>
</table>

Table 3: Processing times for the three ACE implementations for speech frames of length 11.6 ms on a 625MHz clock PDA.

Fig. 1(a) Block diagram of ACE using FFT.

Fig. 1(b) Block diagram of ACE using wavelet packet transform.

Fig. 2 Wavelet packet binary tree structure and branch orders.

Fig. 3. Interactive LabVIEW Front Panel of ACE implementation using wavelet packet transform.

Fig 4. MSE, in dB, between floating-point and fixed-point analysis channel outputs for recursive DFT (dashed lines) and wavelet transform (solid lines) implementations.