Third Quarterly Progress Report

NIH-NO1-DC-6-0002

Open Architecture Research Interface for Cochlear Implants

Arthur Lobo, Nasser Kehtarnavaz, Venkat Peddigari, Murat Torlak, Hoi Lee
Lakshmish Ramanna, Selami Ciftci, Phillip Gilley, Anu Sharma and Philipos C. Loizou

Department of Electrical Engineering
University of Texas-Dallas
2601 N. Floyd Rd
Richardson, TX 75080
Email: loizou@utdallas.edu

October 1, 2006 – December 31, 2006

1. Introduction

The main aim of this project is to develop a research interface platform which can be used by researchers interested in exploring new ideas to improve cochlear implant devices. This research platform includes a stimulator unit which can be used for electrical stimulation in animal studies, a recording unit for collecting evoked potentials from human subjects and a portable processor for implementing and evaluating novel speech processing algorithms after long-term use. The research platform chosen for this project is the personal digital assistant (PDA).

2. Summary of activities for the quarter

We continued with the C implementation of speech coding algorithms on the PDA. We also worked on recording EEG on the PDA using a different data acquisition card, and continued investigating a wireless transmission approach for transmitting signals from the PDA to the implant headpiece. Finally, we report on progress made on the design of a single-channel current source.

2.1 Real-time C implementation of a 16-channel noise-band vocoder on the PDA

The thrust of our efforts during this quarter was placed on achieving real-time implementation of a 16-channel noise-band vocoder on the PDA platform. Considering that most algorithms in commercial implant devices are based on vocoders (Loizou, 2006), we focused on the implementation of a 16-channel noise-band vocoder. The noise-band vocoder was implemented in C language using Microsoft Visual Studio 2005 on a 624-MHz iPAQ hx2790 PDA running Windows Mobile 5.0 (the LabVIEW implementation was reported in our last progress report and in Pediggari et al., 2007). The speech signal was acquired and played in 46.4 ms blocks in Full Duplex mode. The Microsoft Wave API was used for receiving and sending samples from/to the codec.

The C implementation was further optimized using the Intel's Integrated Performance Primitives (IPP) library. The signal processing IPP library in particular was used to implement the noise-band vocoder. The modular nature of the IPP routines allowed for a compact implementation of the noise-band vocoder. Only three lines of code are needed to estimate the envelope of the signal in a specific channel. An example usage of IPP functions to compute *N* samples of the envelope in channel 1 is shown below:

```
ippsIIR_BiQuadDirect_16s(ch1_in, signal, N, ch1_filter,
      BP_SECTIONS, DelayLineCH1);                    //Bandpass filter
ippsAbs_16s_I(signal,  N);                           // Rectification
ippsIIR_BiQuadDirect_16s_I(signal, N, lpf, LPF_ SECTIONS,
       DelayLineCH1_LPF);                            //Low pass filter
```

The above IPP filtering routine (ippsIIR_BiQuadDirect_16s) implements IIR filtering using bi-quad sections (3 sections were used in our implementation). We found that the

IPP 16-bit filtering function did not provide sufficiently high precision to match the floating-point output. For one, the intermediate precision between bi-quad sections was 16 bits, which was found to be inadequate. We therefore implemented the filtering in assembly language (ARM) using 32-bit precision for the intermediate samples, resulting in an output that matched the floating-point output very closely.

Time profiling was performed to assess the timing involved in the various signal processing functions operating on a 46.4 ms block. The timing distribution for a 46.4 ms block of data is shown in Table 1.

Function	Time (ms)	% processor load
Analysis filtering & envelope detection	6.0	48
Noise modulation	2.3	18
Synthesis filtering	4.3	34
Total	12.6	100

Table 1. Time profiling of the noise-band vocoder (16 channels) implementation on the PDA.

From the above Table we can conclude that the PDA implementation of the 16-channel noise band vocoder is on the order of 3.7 times faster than the real-time limit. Note that the synthesis filter stage is not present in an actual speech-processing strategy for cochlear implants. Consequently, the timing involved to implement say the CIS strategy will be considerably lower. For 16 channels, it was estimated to be approximately 7 times the real-time limit, after including the compression (i.e., acoustic to electrical mapping) stage.

2.2. Evoked Potential (EP) data acquisition on the PDA using LabVIEW

In Loizou et al. (2006), we reported on EEG data acquisition on the PDA using the C-Cubed compact flash card that plugged into the PDA. Software development was done in C. Our efforts during this past quarter focused on the development of real-time data acquisition routines in LabVIEW using a different data acquisition card. More specifically, we examined the use of the CF-6004 compact flash card manufactured by National Instruments (National Instruments, 2006). Our long-term goal is to compare the EEG/EP recordings acquired by the two cards and determine which card is more suitable for our application and produces cleaner recordings.

The real-time data acquisition routines for acquiring EEG/EP were developed in LabVIEW using the LabVIEW PDA Module and the DAQmx utility supported by the LabVIEW environment. The NI data acquisition compact flash card CF-6004, shown in Figure 1, was used to record the EEG signals. This card possesses four analog channels and four digital I/O channels. It supports 14-bit resolution with 200 kHz sampling rate for a single channel, and 132 kHz aggregate sampling up to four analog channels. The digital

I/O channels can be configured as trigger pulses to initiate or terminate recording on the analog channels.

Figure 1. EEG data acquisition setup using the NI CF-6004 compact flash card.

Using the task configuration utility, we set the sampling rate for each channel to 1000 Hz for EEG/EP recordings. The number of samples acquired per channel at each epoch was specified by the user via a graphical-user-interface (see Fig. 2).

The data acquisition process consisted of playing back the stimuli repeatedly for the number of epochs specified by the user. The speech signal stimuli were padded with zeros during the inter stimuli interval and played back in blocks of a specified size (1024 samples). Simultaneously, the DAQ utility functions were used to acquire the EEG data from two analog channels, one for the EEG and one for the eye blink. In order to acquire the evoked potentials, one of the channels was used to generate a sync pulse which became high when the stimuli started playing or when each epoch started. This allowed for synchronization of the recorded evoked potentials with the presented stimuli.

The above task required resolving various implementation issues such as synchronization of the sync pulse with evoked potential recordings and buffer management to avoid overflows. The buffer overflow problem was resolved by setting higher sampling rates for each channel, allowing the pre-allocation of a large buffer size to avoid any buffer overflows. For instance, when the sampling rate was higher than 10,000 Hz, a buffer size of 100 KS/s was allocated automatically. This problem has not yet been fully resolved.

Currently, we are exploring ways to resolve the synchronization issue by generating the sync pulse at one of the digital I/O channels and feeding that as the input to the analog channel which is used for recording the sync pulse. Note that this synchronization issue was not encountered with the C-Cubed card. We plan to collect EEG and EP data with the NI card in the next quarter.

Figure 2. GUI used for real-time EEG data acquisition. Users can change the number of epochs run, select the analog channels, set the inter-stimuli interval and display the acquired channel outputs.

2.4 Wireless implementation

We worked on half-duplex speech transmission over wireless ad-hoc (peer-to-peer) network connection between two PDAs. A User Datagram Protocol (UDP) connection was used instead of a TCP connection over the peer-to-peer connection. UDP does not use acknowledgments and is preferred for real-time applications. In our current setup, one of the PDAs was used as a "client" and the other one as a "server". Since the client PDA emulates the implant headpiece, the client PDA used the built-in microphone to record audio (sampled at 22 kHz), and transmit the recorded audio signal wirelessly to the server PDA in wireless ad-hoc mode. The audio signal was played continuously without any disruption. In order to obtain a more robust connectivity, we kept the size of the transmitted data packets under Maximum Transmission Unit (MTU) size, which is 1500 bytes. The server PDA informs the user about the number and percentage of packet losses every 3 seconds by displaying this information on its screen.

Since the UDP connection does not guarantee quality of service (no retransmissions are requested), we have tested the performance of wireless connectivity in terms of packet losses. Specifically, we have conducted a preliminary experiment in two different scenarios. In the first scenario, the locations of the two PDAs were fixed in an open space with a PDA-to-PDA separation of 3 meters. A total of 10 trials were run in each scenario. The percentage of packets lost was recorded for each trial and is given in Table 2.

Trial No.	Percentage Packet Losses
1	**2.80**
2	**3.04**
3	**2.80**
4	**2.82**
5	**2.61**
6	**3.03**
7	**2.39**
8	**3.05**
9	**2.85**
10	**2.84**
Average	2.82

Table 2. Percentage of packets lost during a PDA-to-PDA wireless transmission.

In the second scenario, the two PDAs were placed on the user and the user was asked to walk around the lab. The average packet loss was approximately the same as shown in Table 2.

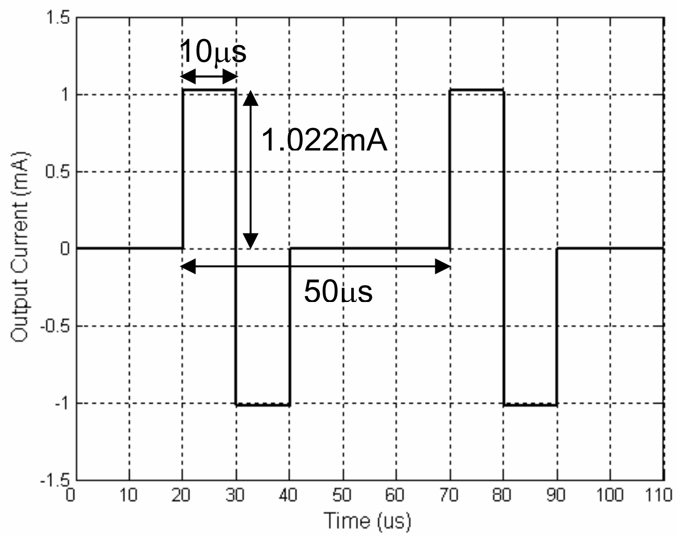
As can be seen from the experimental results, the packet losses are very small in both cases. While such packet loss rate might be acceptable for transmission of the speech signal, it might not be acceptable for our application. We will therefore investigate the reasons of these packet losses and propose solutions to reduce the packet loss under more severe situations. We will also include cochlear processing in the server PDA to complete the full wireless connectivity. If needed, we will also investigate different encoding schemes (e.g., ADPCM) to reduce the data rate.

2.5 Design of single-channel current source

A new integrated single-channel stimulator has been designed and implemented with AMS (Austriamicrosystems) 0.35μm CMOS process. The chip has been sent out for fabrication in order to evaluate the performance of the proposed current source. The single-channel stimulator will be evaluated first before designing the multi-channel stimulator.

The new single-channel stimulator contains a single current source and switches for sourcing and sinking current pulses. The current source is compact and has high output resistance larger than 100MΩ. It has a 9-bit amplitude resolution covering the current range from 2μA to about 1mA. Switches can accurately control the direction of the current sourcing and sinking from the electrode.

Fig. 3 show the post-layout simulation results of the new stimulator, which can provide wide range of current amplitude of biphasic current pulses while preserving charge balance. It supports stimulation rates up to 20,000 pulses/sec.



(a)

Fig. 3 Post-layout simulated charge-balanced biphasic current pulses.

3. Plans for next quarter

- We will pursue the implementation of the ACE strategy used by Cochlear Corporation in their Nucleus-24/Freedom device. We will profile the timing required by each signal-processing block.
- For the EEG/EP data acquisition on the PDA, we are planning the following activities:
  1) Make additional EEG recordings using different settings of the amplifier.
  2) We will focus on resolving the current synchronization issues with EEG data acquisition routines in LabVIEW. These synchronization issues were not encountered when using the C-Cubed card.
  3) Work on implementation of evoked potential (EP) data acquisition routines in C for the C-Cubed compact-flash card.
- For the wireless implementation, we will continue testing and assessing the robustness of wireless transmission between two PDAs using Wi-Fi technology.

- We will implement a laboratory-based current stimulator on a breadboard using discrete components. Additionally, we will be testing the fabricated integrated single-channel stimulator.

4. References

Loizou, P. (2006). "Speech processing in vocoder-centric cochlear implants," *Cochlear and Brainstem Implants* (ed. Moller, A.), Adv. Otorhinolaryngol. Basel, Karger, 64, 109–143.

Loizou, P., Lobo, A., Kehtarnavaz, N., Peddigari, V. and Lee, H. (2006). "Open architecture research interface for cochlear implants," *Second Quarterly Progress Report*, NIH-NOI-DC-6-0002.

National Instruments, (2006). *NI CF-6004 Compact Flash Data Acquisition for PDAs*, http://sine.ni.com/nips/cds/view/p/lang/en/nid/201664

Peddigari, V., Kehtarnavaz, N. and Loizou, P. (2007). "Real-time LabVIEW implementation of cochlear implant signal processing on PDA platforms," *Proc. IEEE Intern. Conf. Signal, Acoust. Speech Proc,* Honolulu, Hawaii (accepted).

5. Appendix

Lobo, A., Loizou, P., Kehtarnavaz, N., Torlak, M., Lee, H., Sharma, A. Gilley, P., Peddigari, V. and Ramanna, L. (2007). "A PDA-based research platform for cochlear implants," submitted to *3rd International EMBS Conference on Neuroengineering*.