First Quarterly Progress Report

NIH-NOI-DC-6-0002

Open Architecture Research Interface for Cochlear Implants

Philipos C. Loizou, Arthur Lobo, Nasser Kehtarnavaz, Venkat Peddigari and Hoi Lee

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

April 1, 2006 – June 30, 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). The PDA was chosen for several reasons: (1) it is equipped with a variety of input/output ports needed to interface externally with the recording and stimulating units, (2) it possesses powerful computing capability (some PDAs run as fast as 624 MHz) needed to implement complex speech processing algorithms, (3) it is flexible in as far as programming in either assembly language, C language or LabVIEW, (4) it is inexpensive, (5) has excellent wireless connectivity needed for assistive listening devices, and (6) has multimedia capability. Perhaps the primary benefit of using the PDA as the research platform for cochlear implants is that is easily adaptable to new and emerging technologies without the need to change or build new hardware.

2. Summary of activities for the quarter

In the present quarter, we focused on developing speech coding algorithms on the PDA using C and LabVIEW. The software development in C and LabVIEW was done in parallel. We also started working on the design of a single-channel stimulator, which will be eventually extended to a 24-channel stimulator with simultaneous channel stimulation capability.

2.1 Real-time LabVIEW implementation of a 16-channel noise-band vocoder

We plan to initially implement and debug various speech coding algorithms on the PC and then port and optimize them to run real-time on the PDA. With that in mind, we implemented a 16-channel noise-band vocoder on the PC using LabVIEW. We completed the real-time implementation of a 16-channel noise-band and sine-wave vocoders on the PC. We chose to implement the noise-band vocoders since most speech coding algorithms used in commercial implant devices are based on vocoders (Loizou, 2006). A graphical user interface (GUI) was developed that allows the user to change the number of channels, bandpass filter cutoff frequencies and envelope cutoff frequencies. Figure 1 shows a snapshot of the GUI. In this mode, the user speaks to a microphone, which is plugged to the mic input of the soundcard of the PC, and the synthesized output of the noise-band vocoder is played back to the speaker (or preferably headphones to avoid feedback) in real-time. In another mode, the LabVIEW program processes off-line a signal stored in a file (.wav format) and outputs the synthesized vocoded speech to the speaker. The developed LabVIEW program is a valuable research tool which can be used to assess the effect of speech processor parameter values such as the number of channels, filter bandwidths, frequency allocation, etc. on speech intelligibility. It will be posted

shortly on our website (www.utdallas.edu/~loizou/cimplants/PDA/) and made available to the research community. In the next quarter, we plan to port the PC LabVIEW implementation on the PDA.

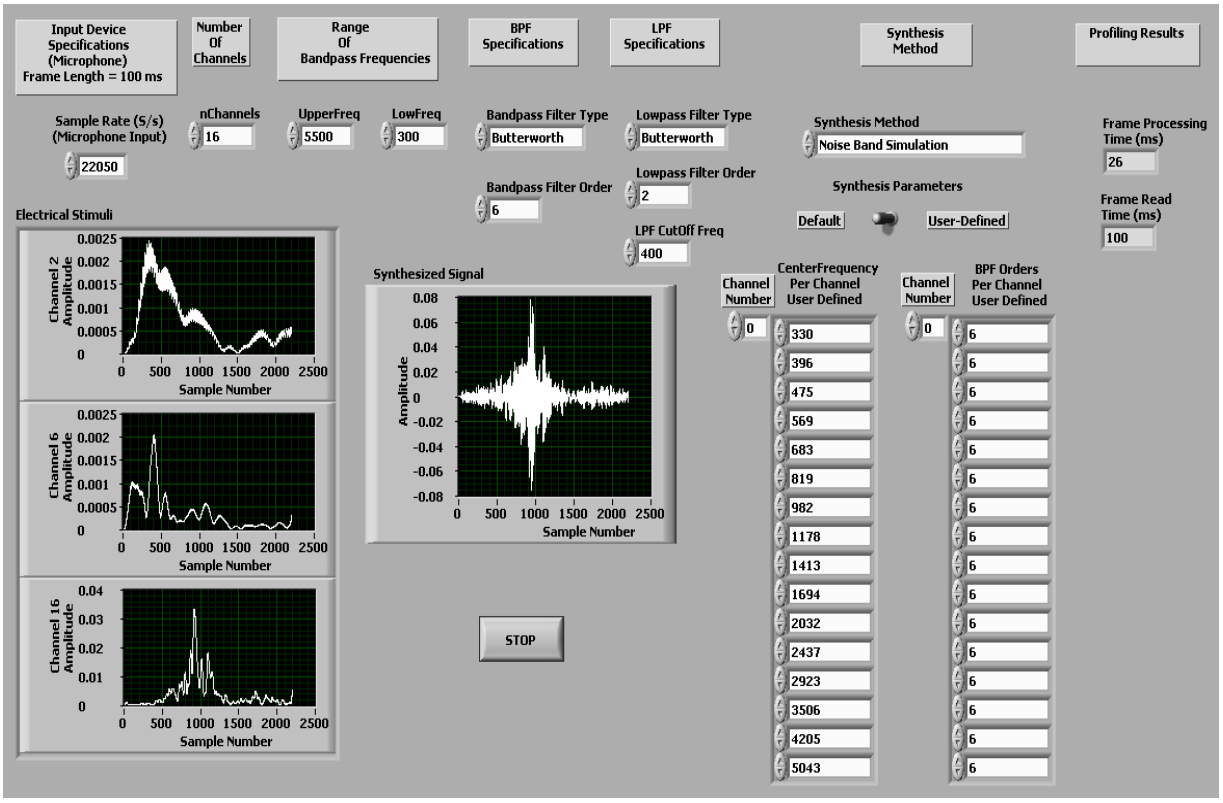


Figure 1. Snapshot of the GUI used for the LabVIEW implementation of the noise-band vocoder**.**

2.2. Filter implementation on the PDA using C and Intel's IPP

In parallel to the LabVIEW implementation of speech coding algorithms, we started implementing individual blocks of the CIS coding strategy using C language and Intel's Integrated Performance Primitives (IPP) routines. The IPP is a software library that provides a broad range of functionality for signal, image, speech, matrix math and audio processing applications (see www.intel.com/software/products/ipp/). The library functions are highly optimized for Intel's latest handheld processor architectures (PXA27x) and are therefore ideal for our application.

We started examining Intel's IPP routine for filtering and found out that the arithmetic precision used by this routine is not sufficient for our application. The filtering routine takes as input the speech signal (16-bits) along with 16-bit filter coefficients and produces the filtered output. The implementation is done in second-order biquad sections. For a sixth-order bandpass filter commonly used in commercial devices, three biquad sections are needed. Examination of the envelope output, obtained after bandpass filtering, full-wave rectification and low-pass filtering at 400 Hz, revealed numerical inaccuracies resulting from quantization noise. Figure 2 shows as an example the bandpass filtered output (top panel) for channel 10 (center frequency =2.1 kHz) and the

corresponding envelope output (bottom panel) obtained after processing the sentence "A boy fell from the window" taken from the HINT database. For comparison, Figure 3 shows the envelope output obtained using floating point arithmetic. It is clear that the Intel's IPP filtering routine performs well for high input amplitude values, but tends to floor the envelope when the input signal assumes low amplitude values. The flooring is caused by underflow problems often encountered with using limited precision and fixed-point arithmetic.

To remedy the above limitation with Intel's IPP filtering routine, we wrote our own filtering routine using ARM assembly language. Unlike the IPP routine, the new routine stores the intermediate results involved in filtering with higher precision. More specifically, we store the intermediate output of the product of two 16-bit numbers in 32-bit precision rather than truncating the product to 16 bits. In addition, we store the output from each biquad section in 32-bit precision. In contrast, the IPP routine truncates the output to 16 bits. Figure 4 shows the corresponding filtered waveform and envelope outputs obtained with our own filtering routine. The envelope output obtained with our own filtering routine is nearly identical to that obtained using floating-point arithmetic (Figure 3).

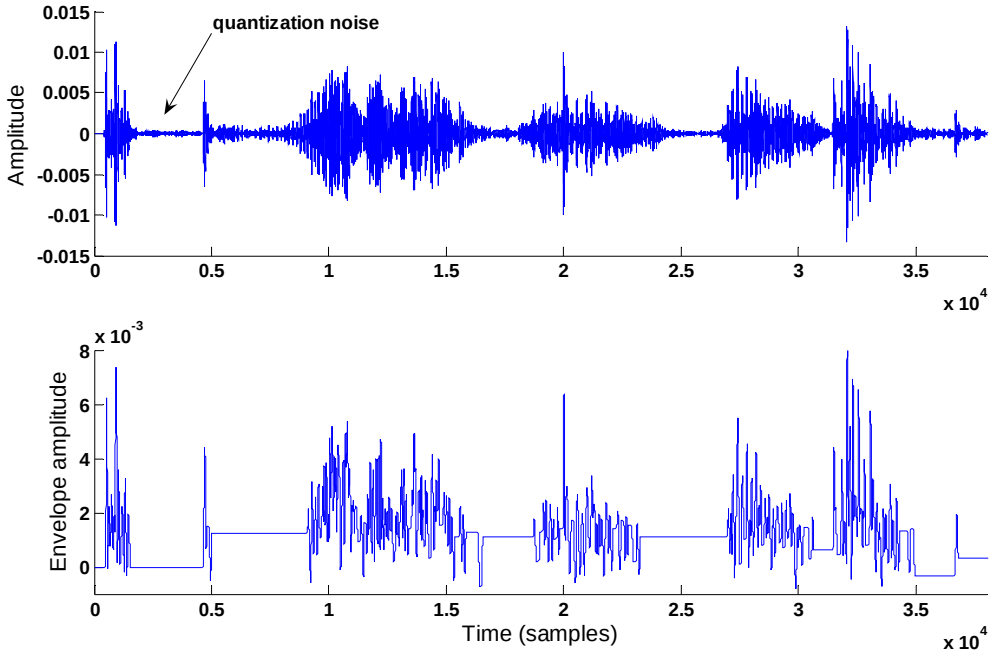


Figure 2. Top panel shows the bandpass filtered waveform (channel 10, center frequency=2.1 kHz) obtained by applying Intel's IPP filtering routine to a sentence. Bottom panel shows the corresponding envelope output obtained after full-wave rectification and low-pass filtering at 400 Hz. Sentence was taken from the HINT database and was sampled at 22 kHz.

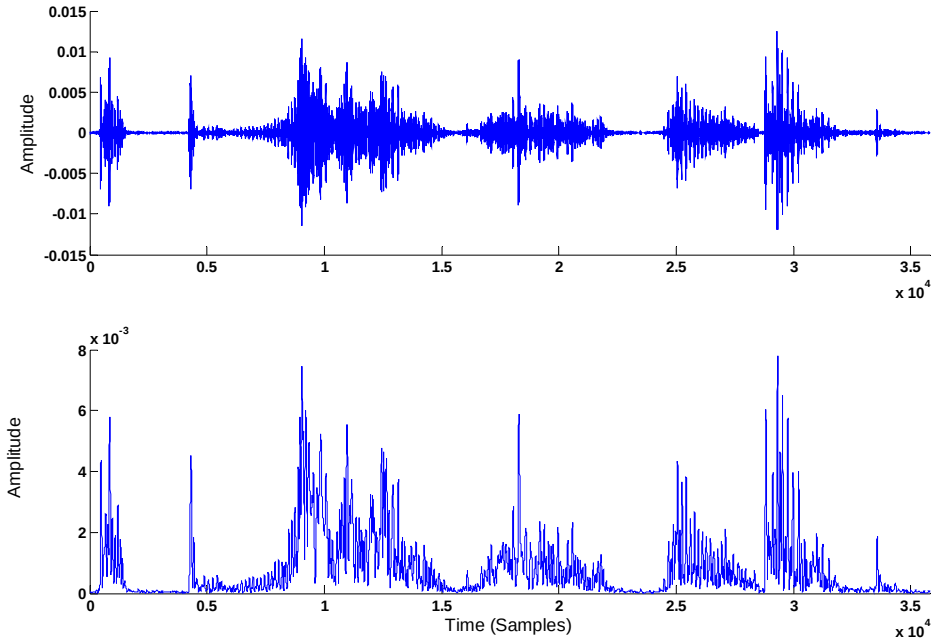


Figure 3. Bandpass filtered waveform (top panel) and envelope output (bottom panel) obtained using floating-point precision.

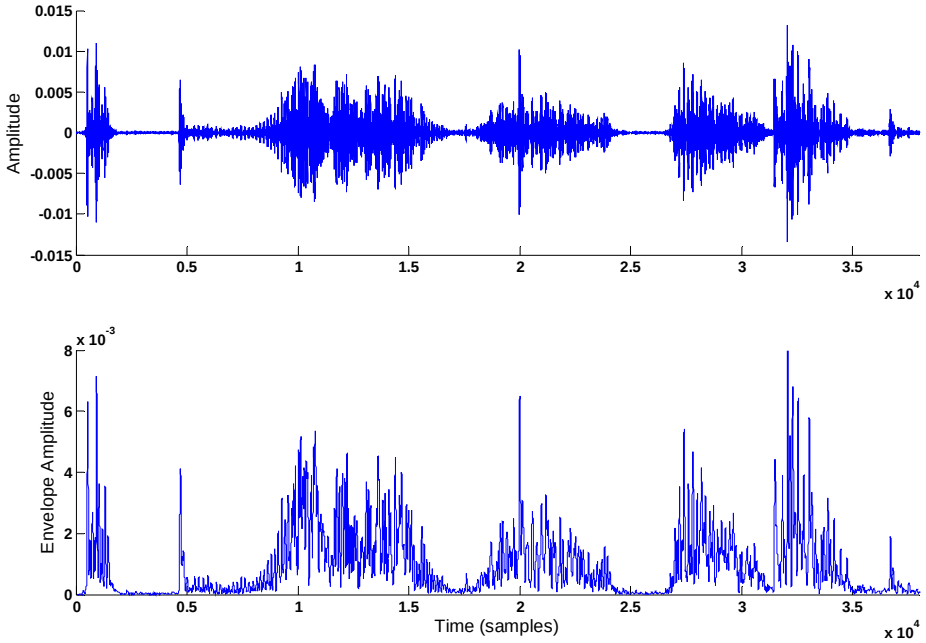


Figure 4. Bandpass filtered waveform (top panel) and envelope output (bottom panel) obtained using our own filtering routine which stored intermediate results in 32-bit precision.

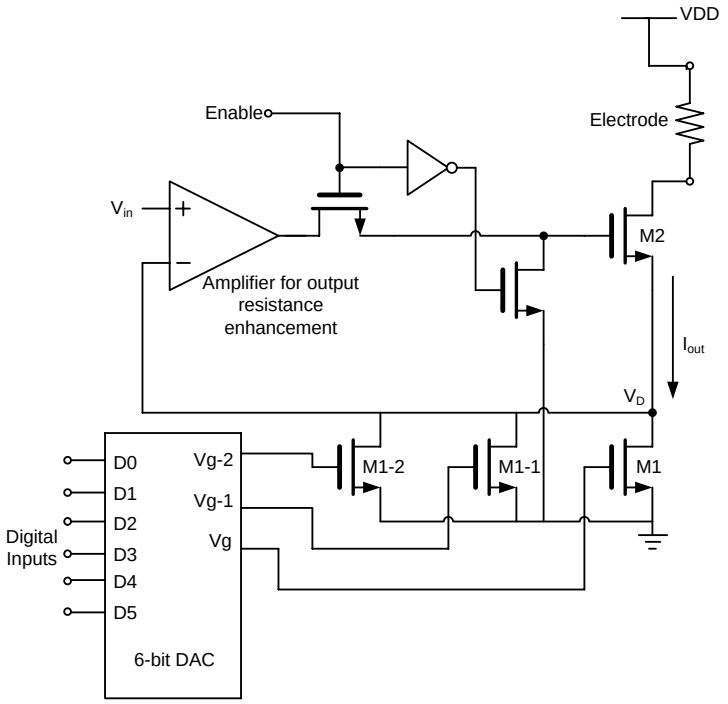
Other problems observed with the design of bandpass filters include the non-flat frequency response of the low-frequency (<1000 Hz) bandpass filters. We will work on tackling those problems in the next quarter.

The assembly and C language software development was done in a Microsoft eMbedded Visual C++ SP4 environment using the Intel Assembler for XScale Microarchitectures (version 2.2) and the Intel C++ Compiler (version 2.0) respectively. A full duplex audio demonstration was developed for a 624 MHz Dell Axim X30 PDA running Windows Mobile 2003. The data acquisition routines were written using Microsoft Windows CE waveform audio functions. The sampling rate was 22 kHz with a buffering delay of 46 ms. The code was also tested on a 624 MHz Dell Axim X51v PDA running Windows Mobile 5.0 and a 520 MHz Bsquare PXA270 processor based development board running Windows CE 5.0.

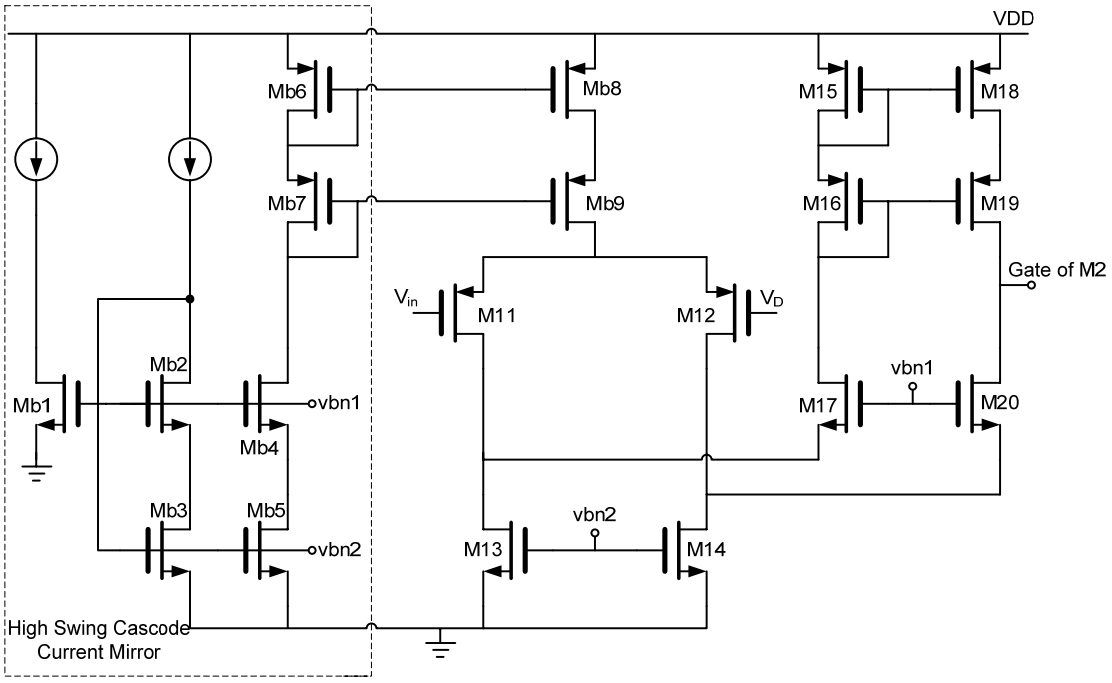
2.3 Design of single-channel stimulator

Different current-controlled stimulations using different types of current sources have been studied. The voltage-controlled resistor (VCR) based current source (Ghovanloo and Najafi, 2005) is utilized for our stimulator. The reason for choosing the VCR current source is for its capability to achieve high accuracy of the stimulus current value along with high voltage compliance. Fig. 5(a) shows the circuit diagram of the single-channel stimulator system. The output current I_{out} of the stimulator is equal to $V_D/(R_{M1}+R_{M1-1}+R_{M1-2})$, as the transistors M1, M1-1 and M1-2 are all operating in deep triode region while keeping $V_D \approx V_{in} << (V_g-V_{th})$, where V_{th} is the threshold voltage of M1. It should be noted that the gain of the amplifier used in the stimulator system should be high in order to achieve constant stimulus current irrespective of the value of the (electrode) impedance. However, it is difficult to maintain high dc gain for an amplifier with low V_{in}. As a result, the folded-cascade amplifier with high-swing cascade current mirror shown in Fig. 5(b) is proposed to address this issue. With the use of high-swing cascade current mirror, the gain of the amplifier is improved by over 20dB compared to the conventional cascade current mirror used by Ghovanloo and Najafi (2005). The folded-cascade amplifier achieves over 95dB gain with V_{in}=30mV. Therefore, the accuracy of the stimulus current I_{out} generated from stimulator can be significantly improved. Moreover, the transistors M1-1 and M1-2 are employed in the stimulator circuit to improve the linearity of the stimulus current under different output voltages of the DAC. With a 6-bit DAC, 64 voltage levels of V_g are generated, which will provide 64 different amplitudes of stimulus current to the electrode. It is important to ensure that every voltage step increment of V_g can give rise to equal step increments of I_{out} to preserve linearity.

HSPICE simulations were run (see Figure 6) to examine the accuracy of the stimulus current (y-axis) for different drain voltages of M2 modeling different electrode impedances (x-axis) and different V_g (1.4V, 2.4V and 3.4V). Simulation results in Fig. 6 show that the 6-bit stimulus current is achieved due to the use of a 6-bit voltage mode current-steering DAC. In addition, the output stimulator achieves 95% voltage compliance (from 0.18V to 3.6V) and accommodates changes in electrode impedances, while maintaining constant stimulus current within 0.2% of the desired value irrespective of the change of drain voltages of M2 from 0.18V to 3.6V. Moreover, with equal increase of V_g, there is almost an equal step increment of the stimulus current. It is thus verified that the linearity of the stimulator is attained by using transistors M1-1 and M1-2. Table 1 provides the summary of the simulation results.



(a)



(b)

Figure 5:(a) Circuit diagram of the single-channel stimulator using a 6-bit DAC and a VCR current source (Ghovanloo and Najafi, 2005) and (b) the amplifier used in the stimulator system with high-swing cascade current mirror.

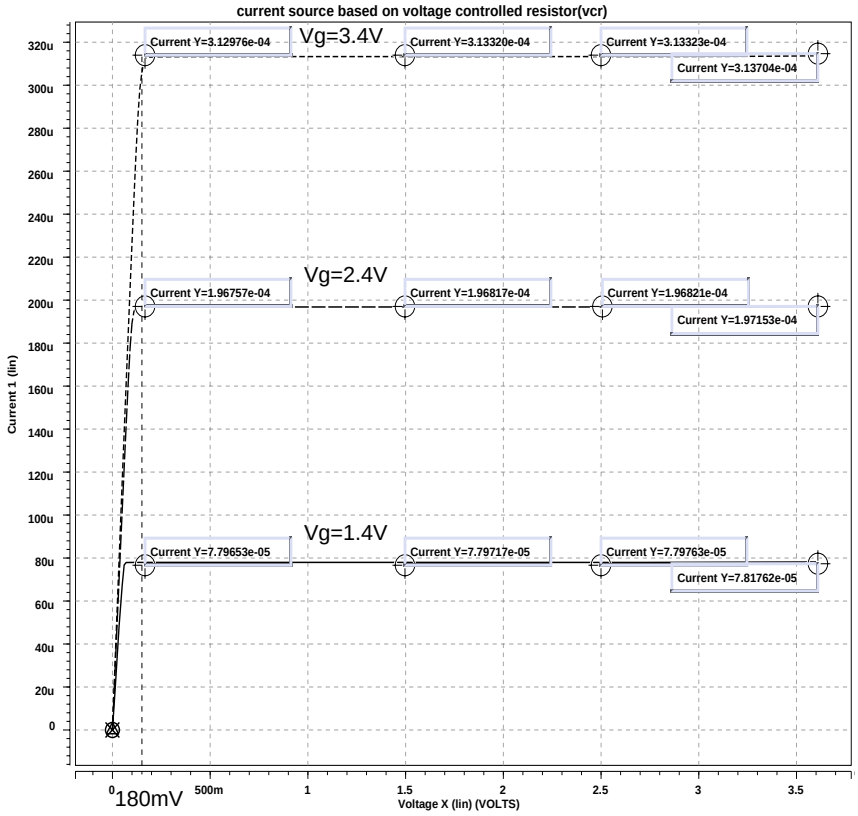


Figure 6 Simulation results of the stimulator system based on the VCR current source design.

One of the major challenges involved in the design of VCR current sources is the fact that the amplitude resolution of the stimulus current is limited by the number of bits of V_g, which in turn depends on the input supply voltage of the DAC. If the amplitude resolution increases, i.e., the step increment of V_g decreases, it is very difficult to maintain the linearity of the stimulus current even when the number of additional transistors added in parallel to M1 increases. It is very time consuming to fine tune the linearity of the stimulus current when the resolution goes up to 8 bits. In addition, due to the requirement of bipolar simultaneous stimulation with different amplitudes, both sinking and sourcing capabilities of the current source are necessary. The number of DACs required for both sinking and sourcing capabilities in the VCR current source increases, and hence the power consumption of the multi-channel stimulator increases significantly with the number of channels. The design of a new current source is underway to tackle these issues and will be pursued in the coming quarter.

Table 1:
Simulation results of a single-channel current-controlled stimulator with VCR current source

Intended Stimulation Mode	Bipolar
Stimulation Amplitude Range (μA)	5-313
Amplitude Quantization (bits)	6
Voltage Compliance	3.42V (0.18V – 3.6V)
Maximum Current Consumption (μA)	32
Supply Voltage	3.6V
Technology	AMS 0.35μm CMOS

2.4 Other activities

- Presentation of our plans and status of the present contract to a neuro-engineering conference (*NeuroEngineering Now*) held at UTD.
- Visit of Drs. Jay Rubinstein, Don Eddington, Mario Svirsky and Charlie Finley to our lab.
- In depth discussion of our plans about the design of the stimulator hardware with Dr. Charlie Finley. Dr. Finley provided valuable feedback about the design of the stimulator hardware. His advice will also be sought later on regarding techniques to remove electrical artifacts from evoked potential recordings.

3. Plans for next quarter

- Port and optimize the LabVIEW code (currently running real-time on the PC) for the PDA.
- Continue analyzing the performance (in terms of arithmetic precision) of existing signal processing IPP routines related to cochlear implants. Comparisons will be made between a MATLAB floating-point implementation of a 16-channel CIS strategy running on the PC and a fixed-point implementation running on the PDA equipped with Intel's PXA27x processor.
- Work on the real-time implementation of a 16-channel noise-band vocoder on the PDA. Such an implementation would be useful in studies with normal-hearing listeners investigating learning effects following changes (e.g., frequency map) to the processing strategy.
- Extend the current stimulator design to accommodate for a higher compliance voltage and better current amplitude resolution.
- Work on interfacing the PDA with Neuroscan hardware to acquire evoked potential data.

4. References

Ghovanloo, M. and Najafi, N. (2005). “A compact large voltage-compliance high output-impedance programmable current source for implantable microstimulators,” *IEEE Transactions on Biomedical Engineering*, 52(1), 97-105.

5. Appendix

N. Kehtarnavaz, V. Peddigari, N. Kim, and P. Loizou (2006). Hybrid programming for DSP lab courses: A cochlear implant system design project,” submitted to *Proc. IEEE International Conference on Signal, Acoustics and Speech Processing.*

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