

# A PDA Platform for Offline Processing and Streaming of Stimuli for Cochlear Implant Research

Hussnain Ali, *Student Member, IEEE*, Arthur P. Lobo, and Philipos C. Loizou\*

**Abstract**—A PDA-based research platform has been developed for implementing novel speech processing strategies and conducting psychophysical experiments with cochlear implant (CI) research that do not necessarily require real-time processing. The developed interface streams stimuli pulses to a CI unit in an offline mode from a Personal Computer via PDA platform using Windows Sockets (WINSOCK). Front-end of the application is run in MATLAB where stimuli pulses are created. Winsock establishes a TCP/IP connection with the PDA and starts the transmission of stimuli data. Server application installed on the PDA reads the stimulation data and forwards it to the SDIO board in packets where it is forwarded to the cochlear implant unit and pulses are then played in real-time. Versatility and flexibility are the key characteristics of the platform for easy implementation and testing of a wide range of applications and experiments without advanced programming skills.

## I. INTRODUCTION

COCHLEAR implants can be seen as a benchmark technology in neural prosthesis for their high success rate and widespread use to restore hearing to the profoundly deaf. The number of cochlear implant users has grown from 12000 in 1995 to nearly 200,000 today as estimated by National Institutes of Health [1]. This growth is attributed to ongoing research in this field both in industry and academia particularly in developing better speech processing algorithms to make auditory sensation as close as possible to natural hearing.

Easy development, assessment and real-time evaluation of speech processing algorithms require a flexible research platform to further invigorate the advancement in cochlear implants. While most implant manufacturers provide research speech processors which allow researchers to develop and test new signal processing algorithms, most research labs are unable to use them either due to limited technical resources or due to constrained framework of the interface provided by the manufacturer [2]. Limitations of these research platforms include flexibility, portability, wearability and long-term evaluation. One of the important factors which hinder their use for speech processing research is that a skilled programmer is required to implement the algorithms in high level or low-level language [3]. To address this, in [2] we proposed a PDA-based research

platform for cochlear implants research for both human and animal studies which overcomes these limitations and provides flexible software driven solution for both clinicians and researchers without requiring advanced programming skills or major hardware investment.

The PDA platform has since undergone numerous hardware and software updates and has been used to implement different algorithms. In [2], we demonstrated real-time implementation of a 16-channel noise-band vocoder algorithm in C and LabVIEW, which is a similar algorithm used in commercially available implants. In the same paper we also reported recording of EEG signals on the PDA acquired through a compact flash data acquisition card. In [4], Peddigari *et al.* presented the real-time implementation of cochlear implant signal processing system on the PDA platform in LabVIEW using dynamic link libraries. A recursive real-time DFT-based ACE implementation for high analysis rate was reported in [5]. Gopalakrishna *et al.* presented real-time implementation of wavelet-based Advanced Combination Encoder in [6] and a novel wavelet packet based speech coding strategy in [7] on PDA platform. Successful implementation of these algorithms on the PDA platform justifies its potential for implementing wide variety of algorithms with varying complexity for real-time applications.

Until now, all the coding and algorithm testing on the PDA platform had to be carried out in C or assembly which might be a cumbersome and time consuming process for a researcher trying to make small changes in the code. Also, it might not be suitable for clinicians having little knowledge of C or experiments that would not necessarily require real-time processing. To overcome these limitations, we propose a setup to use the PDA platform for offline experiments by interfacing the platform with a personal computer running MATLAB. The proposed setup allows researchers and clinicians to implement their algorithms in a familiar MATLAB environment and evaluate the performance of their algorithms with great ease without writing C code or learning the syntax and communication protocols. Hence, the complete PDA platform now supports both real-time and offline capabilities, a feature not available in any other CI research platform.

This paper is organized as follows. Hardware overview of the complete platform is presented in Section II. Section III describes the software architecture of the proposed system followed by Section IV which presents some of the applications and tools for behavioral and psychophysical testing.

Manuscript received March 24, 2011. 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.

H. Ali, A. Lobo and P. Loizou are with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: loizou@utdallas.edu).

## II. HARDWARE OVERVIEW

The research platform as a whole comprises of:

- i. a portable processor (PDA) for implementing and evaluating novel speech processing algorithms after long-term use [2],
- ii. an interface board to connect the PDA with Freedom cochlear implant coil using secure digital input output (SDIO) port of the PDA [8], [9],
- iii. a bench-top stimulator designed for electrical stimulation and neurophysiologic studies with animals [10] and
- iv. a recording unit for collecting evoked potentials from human subjects [2].

Hardware required for offline processing and streaming of stimuli involves a PC running MATLAB, a PDA, SDIO interface board, and Freedom BTEs with coils. For testing purposes, an implant emulator is used. Fig. 1 shows the complete hardware setup.

### A. PDA

The PDA is connected to the PC running MATLAB via a docking station/cradle's USB connection using Active Sync (for Windows XP) or Windows Mobile Device Center (for Windows 7), which is a synchronization application for Windows Mobile based devices with the PC. PDA requirements include Intel (Marvell) XScale/PXA27x/PXA31x processors as cores with minimum 624MHz processing speed powered by Windows Mobile 5 or 6.5 operating system. In the presented system, HP iPAQ models hx2490 and hx2790 were used.

### B. SDIO Board

The SDIO board is a custom developed board used to communicate and interface the PDA with an actual cochlear implant. It plugs into the SD port of the PDA and has two output connectors which connect with the right and left ear (Behind the Ear units) BTEs via cochlear cables, thus allowing bilateral stimulation. It is equipped with a Xilinx Spartan 3 FPGA, Arasan SDIO interface controller, PROM, and power management circuitry to implement the communication interface between the PDA and the implant. The board is powered using a 5V battery source. For more details about the board, please refer to [8], [9], and [10].

### C. BTE and coil

Freedom coils with modified HS8 BTEs from Cochlear Ltd. [11] were used in the experiments.

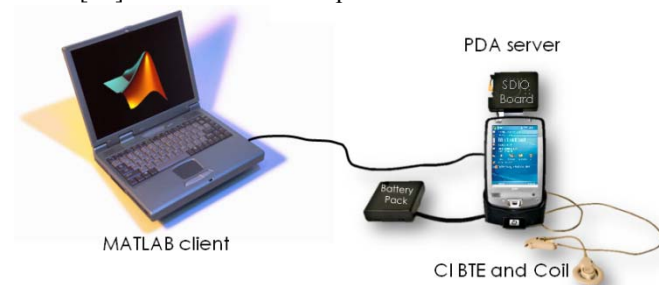


Fig. 1. Hardware modules for the MATLAB-PDA interface.

## III. SOFTWARE ARCHITECTURE

The software architecture is designed such that PDA acts a server which accepts the incoming connections while the PC acts as a client with MATLAB as a front-end as shown in Fig. 1. Overall design can be broken down into three main software components:

- i. Server running on the PDA,
- ii. MATLAB client (.mexw32 or .mexw64 dll) called from the MATLAB front-end, and
- iii. MATLAB front-end running on PC.

Server client interface is based on Winsock (Windows Sockets API) which is a technical specification that defines how Windows network software should access network services, especially TCP/IP. It defines a standard interface between a Windows TCP/IP client application (such as an FTP client or a Gopher client) and the underlying TCP/IP protocol stack [12]. Fig. 2 shows the transfer of parameters and amplitudes from MATLAB to the PDA and status returned from the PDA to MATLAB.

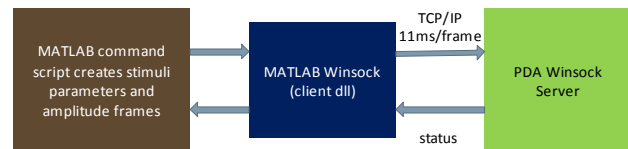


Fig. 2. MATLAB-PDA interface using Winsock.

### A. MATLAB front-end

The MATLAB front-end, as the name suggests, is the application layer of the system around which most researchers would work. It could either be a simple command script to create synthetic stimuli and stream them to the PDA by calling the client dll (dynamic link library - responsible for invoking client-server communication protocol) or it could be an elaborate GUI or application which implements speech coding algorithms and uses client dll as a backbone. A variety of applications can be created at the front-end suitable for different experiments using the same client dll. Some of these applications will be presented in the next section.

A lot of effort in the design of MATLAB-to-PDA interface has been directed to make the MATLAB front-end as simple and user-friendly as possible while giving users the access to all possible feature space to experiment with. This is achieved by limiting all the overhead code and communication protocols to the PDA component or the server. The result is that a researcher only needs to prepare a vector of amplitudes for specified electrodes and provide stimuli parameters to run the system. This allows the researchers to focus more on experiments than on coding while saving a lot of time. The MATLAB front-end has following important functions:

- i. Load patient map
- ii. Load/Create stimulus data
- iii. Check stimulus parameters and amplitude levels for safe operation
- iv. Call *client* dll to stream data to the PDA.

Details of some of these functions are given below.

1) *Patient MAP*: First, the patient map is created either from a patient map file or an application [13] to load stimulation parameters specific to a patient. Patient map includes parameters like implant and electrode type, speech processing strategy, stimulation rate, pulse width, MCL and THR values to name a few. Complete map parameters are shown in Table I.

2) *Checking Parameters*: For the safe implant operation within the implant limits, a parameter checking routine is hardcoded in the main routine. Parameters which are checked for the safe operation are stimulation rate and pulse width for both left and right ear implants depending upon the implant type. For example, for the Nucleus N24 series implant, for the standard rate protocol (STD), the *total stimulation rate* which is the product of *stimulation rate per electrode* and *Nmaxima* for ACE (or *number of active channels* for CIS) should be less than 14,400 Hz. A higher stimulation rate up to 32 kHz is described as high-rate protocol (HR) for which the pulse-width is fixed at 12  $\mu$ s for 5 cycles per cell embedded protocol and 9.6  $\mu$ s for 4 cycles per cell embedded protocol.

Other than this, *rate-centric* and *pulse-width centric* parameter checking routines are also hardcoded. In a rate-centric routine, for example, if a user specifies stimulation rate and pulse width that are not realizable, the pulse width will be adjusted to fit the pulse rate. These routines are also hard-coded at the server end for further safety purposes.

3) *Stimulation Parameters*: The stimulation data is sent to the PDA server from the client dll in frames of 11.6 ms. Once the stimulation rate, pulse-width and number of active electrodes have been specified and verified for safe operation, following stimulation parameters are computed:

$$\text{Pulses per frame per channel} = \text{floor}(0.5 + (11.6 \times 10^{-3} \times \text{Rate}))$$

$$\text{Pulses per frame} = n_{\text{channels}} \times \text{Pulses per frame per channel}$$

where *nchannels* is the number of active electrodes and *Pulses per frame* is the number of stimulation pulses delivered to all active electrodes every frame interval.

4) *Stimulation Data*: Stimulation data essentially comprises of two data arrays containing magnitude and electrode information. Electrode information comprises of an array containing the stimulation sequence of the active electrodes while magnitude information corresponds to the amplitude levels of the stimuli for the respective electrodes. Stimulation data may be loaded from a pre-processed file or it may be created by implementing any speech coding strategy. Alternatively, a waveform of synthetic stimuli may be created for psychophysical experiments. The stimulation data, however, should correspond to the stimulation parameters already specified.

4) *Error Checking*: Before streaming, a final check on the stimuli amplitude levels is done to make sure that the amplitudes fall within the safe range specified by the patient's map. This is done by comparing the amplitude

TABLE I  
PATIENT MAP PARAMETERS

Parameter	Option
Implant/Electrode type	CI24RE(CS/CA) / CI24M / CI24R (CS/CA) / CI22M / ST
Left/Right ear Implant Strategy	ACE / CIS / SPEAK
Number of Implants	Unilateral / Bilateral
Electrode Configuration <sup>1</sup>	MP1 / MP2 / MP1+2
Number of Active electrodes or Nmaxima	(1 to 22) depending upon strategy
Left/Right Stimulation Rate	250Hz – 3500 Hz
Left/Right Pulse Width	9.6 $\mu$ s-400 $\mu$ s depending upon rate
Left/Right THR	0 – 255 clinical units
Left/Right MCL	0 – 255 clinical units

<sup>1</sup>Electrode configuration: monopolar mode MP1|MP2|MP1+2 [11]. For CI22M common ground mode CG(1), bipolar modes BP(2), BP+1(3), BP+2(4),....

levels of each electrode with the MCL and THR levels of that electrode. If any amplitude level is greater than the MCL value of that electrode, the amplitude is then saturated to the MCL level. On the other hand, if the amplitude levels are lower than the THR levels of the respective electrodes then they are floored to the THR level. The maximum amplitude level allowed varies across electrodes, cochlear implant patients and for different electrode arrays. Error checking routine is hardcoded both at client and server ends for safety reasons.

5) *Call to client*: Finally, client dll is invoked with stimulation parameters and data for each of the right and left implants as its input. Client routine is discussed below.

#### B. Client dll

The client dll initializes Winsock, creates a socket, connects to the server and transmits parameters and frames of stimuli created in the MATLAB front-end application. It does this in two steps to match the *receive* function on the server: first the number of 11 ms frames, *nframes*, and the number of pulses per frame are transmitted. Second, *nframes* frames are transmitted continuously with the time interval between frames set to 11ms. The dll is compiled from the C source using the MATLAB MEX compiler.

#### C. PDA Component - Server

The PDA component initializes Winsock, creates a socket, binds the socket, "listens" on the socket, accepts incoming connections, and performs blocking receives to receive the parameter and stimuli data from the client. The "receive" is performed within a thread in two steps. In the first step, information about the total number of 11 ms frames, denoted as *nframes* and the *number of pulses per frame* is received. The server then performs *nframes* receives, each time sending the data to the SDIO board verified by the error-checking routine. After *nframes* are sent and received, the server closes the socket and the connection.

The PDA server which runs continuously in a Windows thread will automatically initialize a new connection for the next incoming stimulus and wait for the client to transfer the corresponding next set of parameter and amplitude frames.

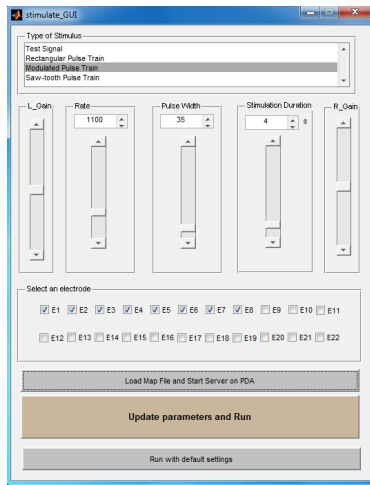


Fig. 3. Application designed in MATLAB to conduct psychophysical experiments

In this way, the transfer of amplitudes takes place on demand i.e. the transfer is made under the complete control of the user.

The PDA component is built as a Windows Mobile 5.0 executable using Visual Studio 2005/2008 Professional in C. The executable is deployed on the PDA and is run from the desktop remotely using the Windows Remote API (RAPI) application *prun*.

#### IV. TOOLS AND APPLICATIONS

Several applications were designed to address various behavioral and psychophysical experiments. Some of these applications are discussed below.

Fig. 3 shows a GUI for a simple psychophysical test application which streams different types of stimuli waveforms e.g., tones and chirps with different parameters to the specified electrodes. User has the ability to control each individual electrode and change the gain, stimulation rate, pulse width and number of active electrodes.

Fig. 4 shows GUI of another application which allows the user to convert an audio signal/file to a stimulus file and then stream it to the patient using the platform while providing complete control over stimulation parameters, speech processing strategy and patient map. Complete speech processing algorithm is implemented in MATLAB which acts as a flexible environment to implement algorithms with great ease.

The platform was also successfully integrated with Percept [14], which is a software developed by Sensimetrics Corporation to facilitate the design and assessment of sound processing strategies.

The PDA research platform was recently approved by FDA for clinical studies with cochlear implant users.

#### REFERENCES

[1] Ballon, Massie Santos, Sounding off. Inquirer. [Online] January 22, 2010. [Cited: Febraury 21, 2011], <http://business.inquirer.net/money/features/view/20100122-248873/Sounding-off>.

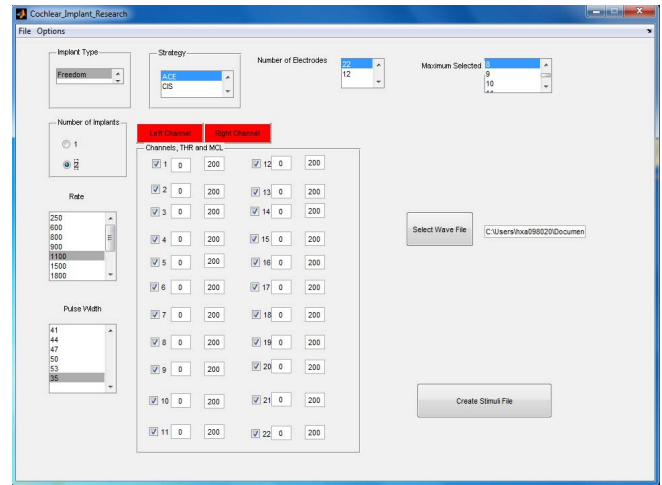


Fig. 4. Process and stream application for converting an audio signal to stimuli pulses using a set of parameters (provided by the user) and streaming them to the CI using the PDA platform.

[2] Lobo, A.P., Loizou, P.C., Kehtarnavaz, N., Torlak, M., Hoi Lee, Sharma, A., Gilley, P., Peddigari, V., Ramanna, L., "A PDA-based Research Platform for Cochlear Implants," *3rd Int. IEEE/EMBS Conf on Neural Engineering 2007. CNE '07*, pp.28-31, 2-5 May 2007.

[3] Ahmad, T.J., Ali, H., Ajaz, M. A., Khan S.A., "A DSK-based simplified speech processing module for Cochlear Implant research," *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, pp.369-372, 19-24 April 2009

[4] Peddigari, V., Kehtarnavaz, N., Loizou, P., "Real-Time Labview Implementation of Cochlear Implant Signal Processing on PDA Platforms," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol.2, pp.II-357-II-360, 15-20 April 2007.

[5] Gopalakrishna, V., Kehtarnavaz, N., Loizou, P., "Real-time PDA-based recursive Fourier transform Implementation for cochlear implant applications", *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp.1333-1336, 19-24 Apr '09

[6] Gopalakrishna, V., Kehtarnavaz, N., Loizou, P., "Real-time implementation of wavelet-based Advanced Combination Encoder on PDA platforms for cochlear implant studies," *IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2010*, pp.1670-1673, 14-19 March 2010.

[7] Gopalakrishna, V., Kehtarnavaz, N., Loizou, P.C., "A Recursive Wavelet-Based Strategy for Real-Time Cochlear Implant Speech Processing on PDA Platforms," *IEEE Transactions on Biomedical Engineering*, vol.57, no.8, pp.2053-2063, Aug. 2010.

[8] A. Lobo, H. Lee, S. Guo, *et al.* (2007). Interface board development for Freedom cochlear implant, single channel stimulator for animal studies and post processing of CAEPs, Fifth quarterly progress report NIH N01DC60002. [Online]. Available: <http://utd.edu/~loizou/cimplants/>

[9] D. Kim, A. Lobo, N. Kehtarnavaz, and P. Loizou. (2008). Hardware testing of revised SDIO board (v.3), integration of LabVIEW code with SDIO board, 11<sup>th</sup> quarterly progress report NIH N01DC60002. [Online]. Available: <http://utd.edu/~loizou/cimplants/>

[10] Kim, D., Gopalakrishna, V., Song Guo, *et al.*, "On the design of a flexible stimulator for animal studies in auditory prostheses," *2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies. ISABEL 2009*, pp.1-5, 24-27 Nov. '09.

[11] J. F. Patrick, P. A. Busby, and P. J. Gibson, "The development of the Nucleus Freedom Cochlear implant system," *Trends Amplif.*, vol. 10, pp. 175-200, 2006.

[12] Makofsky, S., *Pocket PC Network Programming*, Addison-Wesley Professional, 2003, ch. 1.

[13] N. Gunupudi, A. Lobo, D. Kim, N. Kehtarnavaz, and P. Loizou. (2008). "Open Architecture Research Interface for Cochlear Implants", Twelfth quarterly progress report NIH N01DC60002. [Online]. Available: <http://utd.edu/~loizou/cimplants/>

[14] Goldsworthy, Ray, Percept. [Online]. Available: <http://www.sens.com/percept/>