

A Physical Layer Implementation on Reconfigurable Underwater Acoustic Modem

Mehmet Aydinlik, A. Turan Ozdemir, Milica Stajanovic
 MIT Sea Grant College Program, Cambridge, MA, 02139, USA,
 aydinlik@mit.edu, aturan@mit.edu, millitsa@mit.edu

Abstract—Underwater acoustic (UWA) multi-user network algorithms can only be fully tested through experimental studies where communication signals are transmitted through a real underwater channel. Reconfigurable UWA modem provides a flexible environment for the testing of different communication algorithms including networking protocols. In this paper we present design, implementation, and testing of a physical layer algorithm on reconfigurable acoustic modem. This physical layer is very flexible; therefore it can easily be modified in order to be employed for the testing of different UWA networking algorithms.

I. INTRODUCTION

UWA sensor networks have many applications including environmental data collection, pollution monitoring, offshore exploration, and military surveillance. The protocols developed for terrestrial sensor networks are not directly applicable to these networks due to very different channel properties. In particular, acoustic channels are characterized by long propagation delays, low communication bandwidth, and high channel error rate. Therefore recent efforts have been focusing on finding algorithms to provide reliable and high throughput connections through UWA channels [1], [2], [3].

Performance analysis of candidate algorithms can be achieved using an accurate channel model. However to date no single statistical channel model is widely accepted for any of the UWA channel. Modified radio channel models are used in simulations for preliminary evaluation of physical and network layer algorithms. However, a new algorithm can only be fully tested through experimental studies where communication signals are transmitted through a real underwater channel. Experiments are usually performed over a point-to-point link by recording acoustic signals transmitted through a real channel. The received data are then processed off-line. This type of experiments can not be performed for the validation of multi-user network algorithms.

Off-the-shelf acoustic modems can be employed for the experiment with UWA sensory networks [4]; however these modems have hard-coded parameters that cannot be modified to suit an application. These modems usually do not provide functions beyond the basic physical layer. Those that do are limited to rudimentary methods based on polling, or sub-optimal medium access control (MAC). Efficient networking algorithms may require information and services from the physical layer, such as packet error rate and power control.

This work was supported by NOAA Sea Grant NA060AR4170019 NSF "ORTUN" project 0708420.

Proprietary software changes would be needed to extract such information from commercial modems, or even hardware modifications may be needed to provide accurate, high resolution power control.

Most of the off-the-shelf modems are intended to be used as point-to-point links, and therefore lack resources to implement optimized networking algorithms. Network layer algorithms can be implemented on a host computer that communicates to the modem through a serial connection. However, these algorithms would have to utilize the packet structure and the modulation method dictated by the underlying physical layer.

A flexible UWA modem was presented for testing of different communication algorithms including networking protocols in [5]. Due to its flexibility, this modem is called reconfigurable modem or r-modem. Another important feature of r-modem is employing rapid prototyping environment of The MathWorks. Rapid prototyping environment reduces the time spend between simulation and real time implementation by allowing us to convert the simulation code with little effort into real-time code.

In this paper, we present design, implementation, and testing of a flexible physical layer on r-modem platform. This physical layer features quadrature phase shift keying (QPSK) modulation, convolutional coding, and recursive least squares (RLS) decision feedback (DFE) equalizer with phase locked loop (PLL). In the following section, we present the r-modem hardware. R-modem software model is presented in Section III. Test bed and experiment results are given in Sections IV and V, respectively. The paper is concluded in Section VI.

II. R-MODEM HARDWARE

R-modem hardware consists of the main board, UART daughter card, and transducer interface card. Figure 1 presents r-modem hardware.

For the main board we use off-the-shelf board from Spectrum Digital called C6713 DSP Starter Kit (C6713 DSK). This board features Texas Instruments TMS320C6713 DSP chip, 16 Mbytes of synchronous DRAM, 512 Kbytes of non-volatile Flash memory, and TLV320AIC codec.

The TMS320C6713 is a 225 MHz floating-point DSP. Floating-point processor is utilized in order to reduce the time required to convert simulation software into real-time code. The processing power of this DSP is enough to minimize the hand optimization effort for rapid prototyping. The disadvantage of floating point functionality is high power consumption

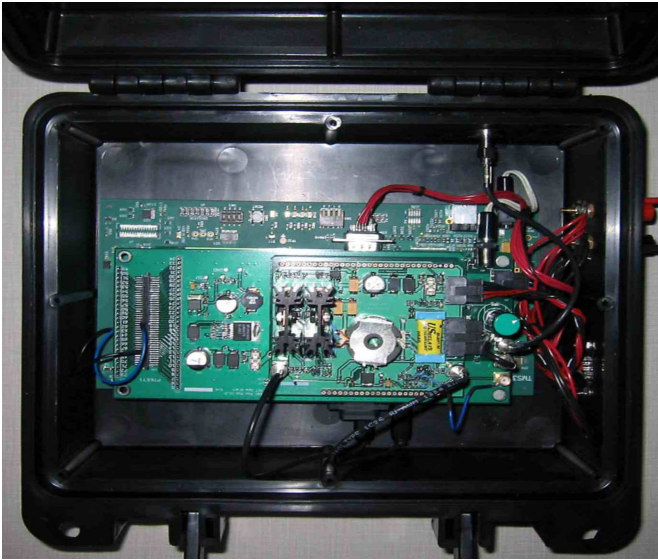


Fig. 1. R-modem hardware

as compared to the C5000 series low power DSP chips. R-modem is intended as a research-based rapid prototyping environment; therefore ease of programming is preferred over low power consumption.

Flash Memory does not lose its contents when the power is off. During the implementation of an algorithm on the r-modem, the DSP program for the algorithm is verified by loading the program to RAM memory. After the verification, the program is burned to flash memory to create a stand alone acoustic modem.

Transducer interface card is utilized to drive acoustic transducers. This card features a class B power amplifier and a two stage op-amp pre-amplifier. The power amplifier can be turned off to save power. Also, the half duplex operation is controlled by selecting receive or transmit modes. general-purpose input/output (GPIO) modules in TMS320C6713 DSP are utilized to turn off the power amplifier and to select the transmit or receive mode during the run time of the r-modem.

The C6713 DSK uses a Texas Instruments AIC23 stereo codec for input and output of acoustic signals. The codec samples analog signals coming from preamplifier on transducer interface card, and converts them into digital data so it can be processed by the DSP. When the DSP has completed data processing, codec converts the samples back into analog signals to feed it to amplifier on the transducer interface card.

UART daughter card provides up to four UART serial communication ports, with RS-232-C or RS-422 interfaces. UART ports are currently used to communicate with a host computer.

III. R-MODEM SOFTWARE

R-modem employs rapid prototyping environment and model based design tools from The MathWorks, namely Simulink, Real Time Workshop, and Target for TIC6000. Together these tools are utilized to develop and validate candidate algorithms from simulation through real time implementation

by performing automatic code generation on TIC6000 processors.

In order to verify an algorithms on r-modem, we first develop a model in Simulink. Simulink features a graphical environment and set of block libraries for design, simulation, and implementation. In order to create the Simulink model, we use both the custom blocks and the blocks from Simulink libraries. After verifying the model on Simulink simulation environment, we update it using peripheral blocks which are required to configure the peripherals on C6713 DSK for the code generation. Peripheral blocks include analog to digital converter (ADC), digital to analog converter (DAC), UART transmit, and UART receive blocks. Finally we generate and verify the code on r-modem hardware.

A communication packet is defined as a collection of a preamble, dead time, training symbols, and data symbols [6]. At the receiver, signals are frame-synchronized before any processing. Preamble is a short sequence with good auto-correlation properties and useful for frame synchronization. Adaptive DFE equalizer is used at the receiver. Predetermined training symbols are transmitted in all frames to train this equalizer.

Figure 2 presents the transmitter Simulink model. In this figure, the Simulink blocks with label "MIT SG" are custom designed at MIT Sea Grant. The data bits, which are received from UART receive block, and the training bits are passed to base band processing subsystem. In the base band processing subsystem, the training bits and the data bits are concatenated to create a frame. Then this frame is passed to the convolutional encoder. The encoded bits are then passed to the baseband modulator. After that, preamble symbols are appended to data symbols to create a communication packet. The communication packet is passed through an interpolation filter. Interpolation filter Simulink block is implemented using polyphase structure. Rate-changing is computationally expensive, since for both decimation and interpolation the lowpass filter is implemented at the higher rate. Polyphase filters integrate rate changing with low pass filtering to reduce the complexity. After interpolation filter, the signals are carrier modulated and sent to DAC.

The Simulink model for receiver is presented in Figure 3. The samples received from the ADC block are passed to carrier demodulator to form baseband signal. The resulting baseband signal is given to decimation filter to reduce the sample rate. Polyphase structure is utilized also for the implementation of decimation filter. The output of the decimator filter is fed to baseband processing subsystem. Baseband processing subsystem includes frame synchronization subsystem and equalization and decoding subsystem.

At the frame synchronization subsystem, the received samples are further down sampled before correlating with the known preamble to reduce the computational cost. Resulting correlation vector can be utilized using different methods to detect the preamble in the input signal. The simplest method would be to compare the maximum correlation value to a fixed threshold. If the maximum correlation value is greater than this threshold, issue a detection signal. The location of the preamble is the location of the maximum correlation value in

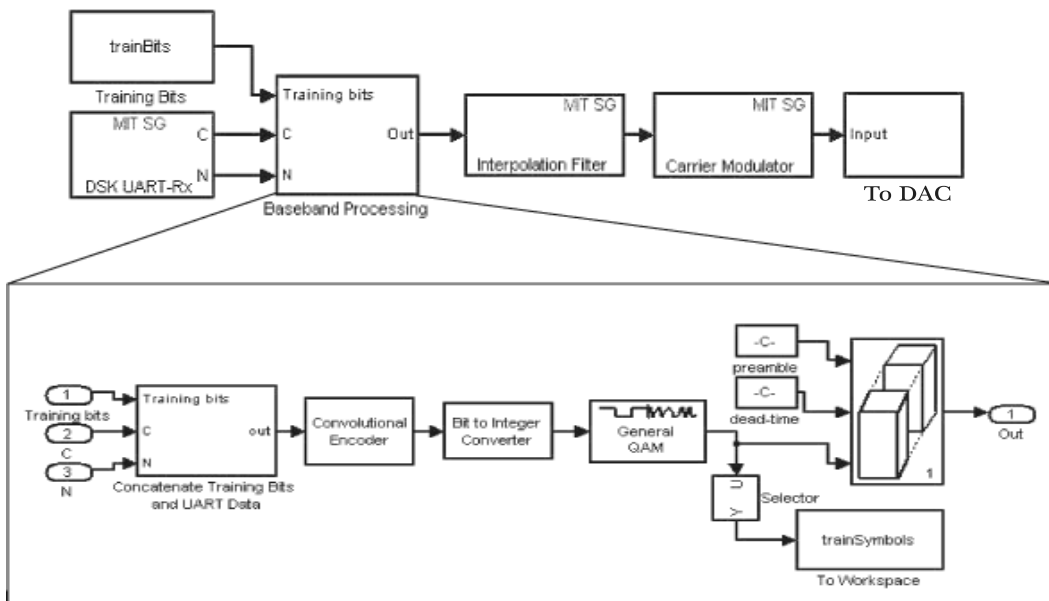


Fig. 2. Simulink model for transmitter

the correlation vector. However the maximum correlation value varies depending on the transmission power, noise level, and the distance between transmitter and receiver. In this case since the threshold is fixed, channel variations can cause receiver to miss the communication packets. In particular this method can not be employed for the applications with mobile transmitter and/or mobile receiver. Therefore, we apply another method which employs adaptive threshold value to detect the preamble signal. In this method, m highest correlation values are found from correlation vector. Then if the maximal correlation value is greater than scaled version of the m th maximum value, a detection signal is issued. The location of maximum correlation shows the location of the preamble. The value of m should be greater than maximum number of multipaths in the channel. This method utilizes the ratio between the correlation values to detect the preamble. Our experiments show that $m = 7$ gives good results in the test tank.

Frame synchronization enables the equalization and decoding subsystem, when preamble is detected. In equalization and decoding subsystem, the data and training symbols are extracted from input signal and applied to the equalizer. The output of the equalizer is demodulated and given to the Viterbi decoder. Decoded bits are then sent to UART transmit block. For equalization, we utilize an adaptive DFE equalizer that combines RLS and a PLL algorithms. We call this equalizer RLS DFE with PLL. Here DFE equalizer is selected due to its good trade-off between the performance and complexity. RLS is preferred as adaptive algorithm over least mean squares (LMS) due to better convergence property of RLS. Also LMS algorithm is hard to optimize due to its sensitivity to step size. PLL is used in the equalizer to jointly perform equalization and synchronization.

IV. TEST BED

We use the MIT towing tank for the initial testing of r-modem. The tank is 108 ft. long, 8.5 ft. wide. The average

depth of the tank is 4.5 ft. The test bed consists of two r-modems, two laptops, and two transducers. The communication between the laptop and the r-modem is provided with serial ports. Figure IV illustrates the test bed.

V. EXPERIMENTS RESULTS

We conducted an experiment in the test bed using a configuration shown in Figure IV. Carrier frequency of 12 KHz was selected for this experiment. For convolutional code, coding rate and generator polynomial were 1/2 and (171, 133) (in octal notation), respectively. The modulation format was QPSK. Each communication blocks included 200 training bits and 300 data bits. The length of preamble was 28 symbols. The bit rate was 550 bits/s.

We achieved error free transmission with the following RLS DFE with PLL equalizer parameters:

- Number of sample per symbols was 2
- Number of feedforward and feedback coefficients were both 6
- Forgetting factor for RLS algorithm was 0.96
- PLL tracking constants denoted by K_{f1} and K_{f2} were 10^{-3} and 10^{-4} , respectively.

Figures V and VI present the real-time scatter plots using the data collected before and after the equalizer, respectively. Real

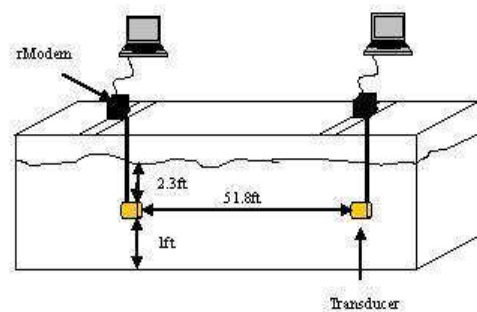


Fig. 4. Illustration of the test bed for r-modem

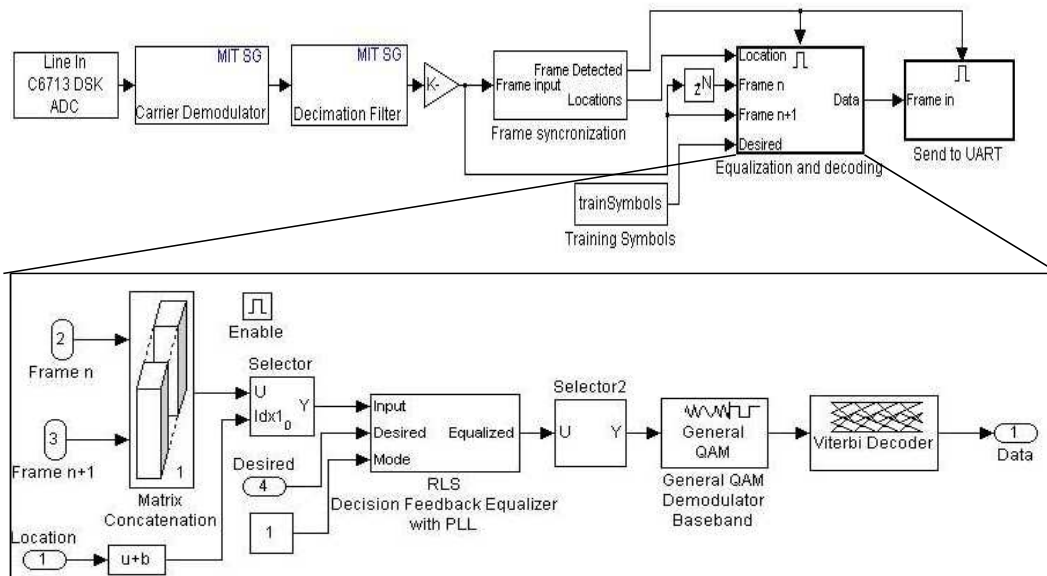


Fig. 3. Simulink model for receiver

time data collection from r-modem was achieved using real time data exchange (RTDX) blocks of Target for TIC6000 product from The MathWorks. These blocks provides data transfer between a host and a target DSP without stopping the applications. The bandwidth for the data transfer is limited due to real-time operation. Therefore 100 data samples in each communication packet were sent from r-modem board to host PC for the scatter plots during the experiment in the test bed.

VI. CONCLUSIONS

We presented design, implementation, and testing of a physical layer algorithm on reconfigurable UWA modem. The reconfigurable UWA (software-defined) modem is intended as a research tool for easy testing of communication algorithms and network protocols.

REFERENCES

- [1] E.M.Sozer, M.Stojanovic, and J.G.Proakis, "Underwater acoustic networks," *IEEE Journal of Oceanic Eng.*, vol. 25, no. 1, Jan 2000.
- [2] S. Singh J. Partan P. Koski K. Ball L. Freitag, M. Grund, "The whoi micro-modem: An acoustic communications and navigation system for multiple platforms,," in *IEEE Oceans Conf.* IEEE, 2005.
- [3] V.Rodoplu and M.K.Park, "An energy-efficient mac protocol for underwater wireless acoustic networks,," in *IEEE Oceans Conf.* IEEE, 2005.
- [4] P.A. Boxley C.L. Fletcher R.K. Creber, J.A. Rice, "Performance of undersea acoustic networking using rts/cts handshaking and arq retransmission,," in *MTS/IEEE Oceans Conf.* 2006.
- [5] E.M.Sozer and M. Stojanovic, "Reconfigurable acoustic modem for underwater sensor networks,," in *ACM WUWNet*, 2006.
- [6] Milica Stojanovic Robert S. H. Istepanian, *Underwater Acoustic Digital Signal Processing and Communication Systems*, Kluwer Academic Publishers, 2002.

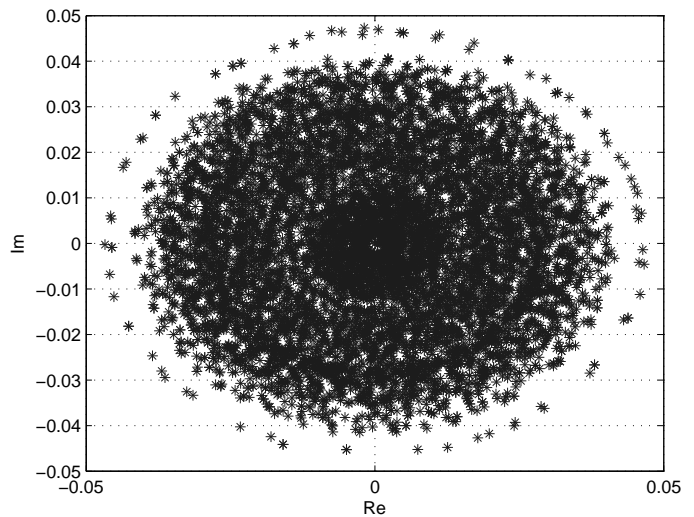


Fig. 5. Scatter plot before the equalization

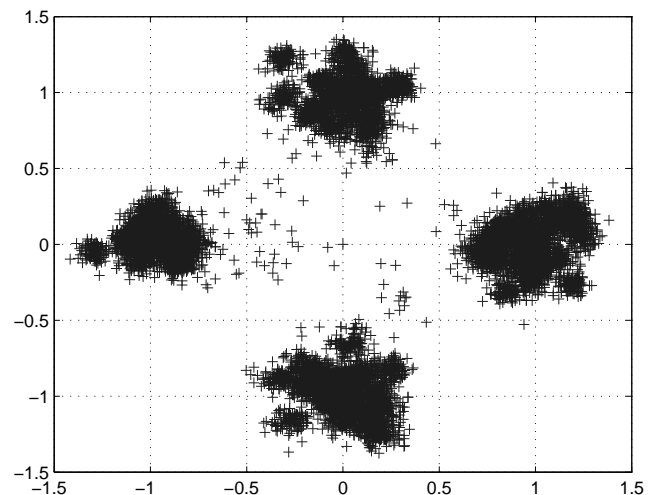


Fig. 6. Scatter plot after the equalization