Semi-blind Channel Estimation Algorithm for an FPGA Based 2x2 MIMO System

Konstanty Bialkowski*† and Adam Postula*†
*School of Information Technology and Electrical Engineering, University of Queensland, St Lucia, Brisbane, Australia
†National ICT Australia - Queensland Laboratory
Email: {ksb,adam}@itee.uq.edu.au

***Abstract*— This paper reports on the development of a semi-blind channel estimation algorithm that is aimed for inclusion in a 2x2 multiple input multiple output (MIMO) system. The system employs a Field Programmable Gate Array (FPGA) for fast processing of transmitted data. Its RF part is built using commercially available components and in-house developed antennas. FPGA hardware is used for fast processing of IF signals. The performance of the developed channel estimation algorithm is tested assuming the Alamouti coding scheme. An in-house developed MIMO channel emulation algorithm is applied during these tests.**

***Keywords*— MIMO systems, Field programmable gate arrays, Maximum likelihood decoding**

I. Introduction

It has been proved both theoretically and experimentally that multiple antennas with suitable signal processing algorithms at transmitter and receiver can significantly enhance the performance of a wireless communication system [1], [2]. Such multiple input multiple output (MIMO) systems can offer two kinds of benefits. One, by employing Spatial Multiplexing they can significantly increase the data throughput (capacity) of a wireless communication system without the need of increasing its operational bandwidth. Two, by using Spatial Diversity they can improve the quality of signal transmission (decrease Bit Error Rate). Such potentials of MIMO systems are of paramount importance with respect to the next generation of wireless communications that have to meet the demand of high date rate transmission.

The benefits of Spatial Multiplexing or Spatial Diversity can be achieved only under the condition that accurate channel state information (CSI) is available to the MIMO receiver. In practice, this condition can be met by applying suitable channel estimation techniques [3]–[5].

The most common channel estimation method involves the use of data sequences which are known at both the transmitter and receiver. These training sequences (also named a pilot signal) accompany the message signal. Their use relies on the assumption that the channel properties remain static during the transmission of a data packet that includes the training and message data sequence. However, they can vary from packet to packet. One shortfall of the training-based MIMO channel estimation approach is that it reduces the throughput of the MIMO channel. In order to counter count this situation, another approach to channel estimation can be applied. It does not involve known data sequences and thus is called blind MIMO channel estimation. As no overhead is involved, this approach maintains high data throughput. However, in practice it requires a long sequence of data to accurately estimate the channel. In order to maintain a high data throughput and at the same time avoid using very long data sequences, a hybrid approach can be used. This new approach is called a semi-blind approach [5]. It involves the use of a training sequence to start each packet and then it exploits properties of the data to adapt channel estimation during transmission. The investigations reported in this paper concern semi-blind channel estimation for a 2x2 MIMO system, which employs the Alamouti coding scheme. The goal is to work out a suitable balance between the training-based and blind channel estimation schemes for a gradually varying MIMO channel. As the entire MIMO system is aimed to be implemented in Field Programmable Gate Array (FPGA) technology [6], real-time implementation of the proposed scheme is studied. Advantages of the devised algorithm over the traditional training-based channel estimation algorithm are demonstrated.

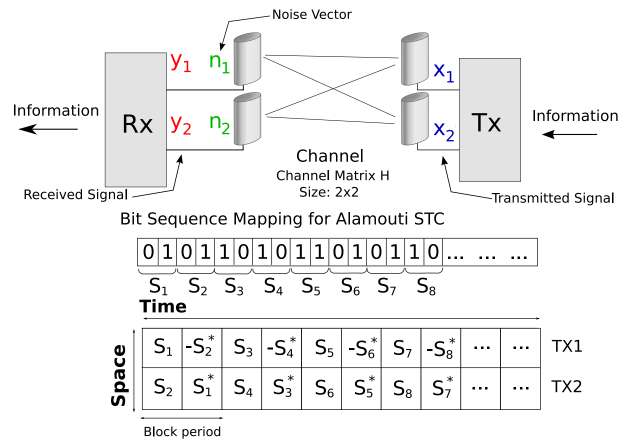

Fig. 1. System Configuration

II. System Description

The configuration of a 2x2 MIMO system, for which channel estimation is to be devised, is shown in Fig. 1.

It uses two antennas both at the transmitter and receiver sites (developed in house) accompanied by RF hardware and signal

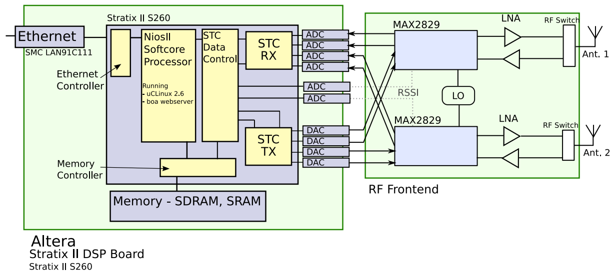

Fig. 2. The block diagram of the complete 2x2 MIMO system.

processing circuitry that uses commercially available components. The modules are controlled by in-house developed software. The system is assumed to use QPSK modulation and the Alamouti encoding scheme. The block diagram for this FPGA system when completed is shown in Fig. 2. The up and down conversion modules make use of MAX2829 chips which are able to perform direct up conversion from base band to an RF frequency of 2.4GHz or 5GHz. Currently the focus is on the development of a 2.4GHz system. Direct up-conversion requires 2 analogue signals, an in-phase (I) and quadrature (Q) signal. This means that for a 2x2 MIMO system, 4 analogue to digital converters (ADC) and 4 digital to analogue converters (DAC) are required. The MAX2829 processor also gives some useful variables such as the current received signal strength indicator (RSSI).

At the current stage, the functioning of this system is investigated at the baseband (IF) level by employing a 2x2 MIMO channel emulator.

A. System Components

The main sections which are implemented in both simulation and FPGA, are the Transmitter module, Channel Emulator and Receiver module. To this purpose, Stratix II 2S60 FPGA, and high-speed Analogue to Digital converters (ADC), capable of 125MSmp./sec (12bits) and Digital to Analogue converters (DAC), capable of 165MSmp./sec (14bits) are used. Their operation is described below.

1) Transmitter Module: takes the input bit stream and prepends the training sequence to every packet. When transmitting each packet the data sequence is split between the two antennas, encoded using the Alamouti encoding scheme and modulated using QPSK modulation. The implementation of this module in both the simulation platform and FPGA is quite easy, as only simple encoding operations are required. The block diagram for this module is shown in Fig. 3.

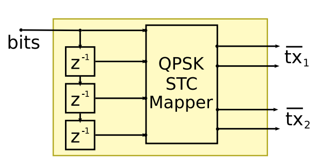

Fig. 3. STC TX Module

2) Receiver Module: has two modes as described in the algorithm description. The first mode involves the use and detection of the training sequence and the second involves adapting the channel estimation using the new incoming data. This is performed by taking the ‘I‘ and ‘Q‘ for both received channels and performing a set of ML decoding and ML estimation. When implemented in FPGA the Maximum Likelihood (ML) decoder and estimator run in parallel and are controlled by the STC scheduler. The two modes differ by what values are used for the ML estimator block. Also in the FPGA design proper synchronization for both symbols and data are required. This is assisted by the training sequence initially being only transmitted on 1 channel. This operation reduces the MIMO system to its SIMO equivalent. The block diagram for the receiver module is shown in Fig. 4.

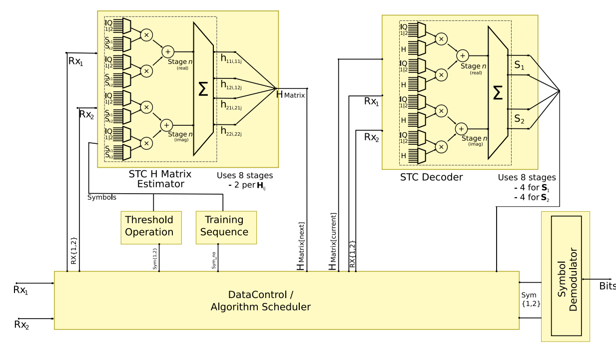

Fig. 4. STC RX Module

3) Channel Emulator: has 2 parameters, the first is the current channel matrix, and the second is the amount of noise. This channel matrix is varied to emulate slow and block fading, which is regulated by adjusting the period that a channel matrix remains constant. For implementation in FPGA, the channel emulator is implemented in the time domain. The set of channel matrices are pre-generated and stored in a look-up table, and randomly selected at run time.

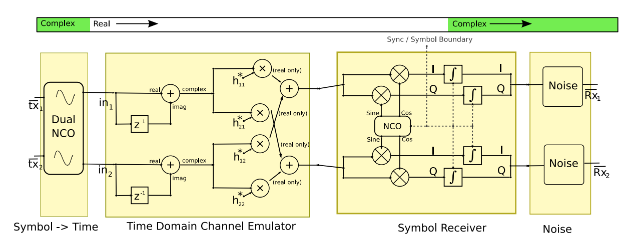

Fig. 5. Channel Emulator

B. Semi-Blind Estimation Algorithm

The Alamouti coding scheme involves a space time block which has a pair of symbols sent in the first block period, and then a modified version of these symbols in the second block period. The details are shown in the lower part of Fig. 1. The channel emulator under a fixed channel matrix uses a Rayleigh propagation model. When the channel emulator emulates a block fading environment, the channel is kept fixed for a certain number of symbol periods [5]. Fast fading is not considered. For the 2x2 MIMO system, the channel matrix H representation is used, as given in Equation (1).

$$y = Hx + n;$$

$$\begin{bmatrix} y_1(t) & y_1(t+1) \\ y_2(t) & y_2(t+1) \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \times \begin{bmatrix} x_1(t) & x_1(t+1) \\ x_2(t) & x_2(t+1) \end{bmatrix} + \begin{bmatrix} n_1(t) & n_1(t+1) \\ n_2(t) & n_2(t+1) \end{bmatrix} \quad (1)$$

The receiver has no a priori information about the channel and uses the training sequence to estimate the channel. The algorithm is based on the Maximum Likelihood (ML) decoding and estimation technique [5]. Being semi-blind it involves two different modes. That is the training mode and the adaptive mode. During training mode, a known training sequence is used to estimate the channel matrix. For purposes of synchronization the first 4 symbols (2 blocks) of the training sequence are sent on only one antenna, to make the problem of synchronization simple, like in a SISO system. After this, an additional 12 symbols/6 blocks (or 28 symbols/14 blocks) are sent using both antennas.

Following the training sequence, the receiver enters an adaptive mode. In this mode, the received symbols are decoded using the previous valid channel matrix, and the result is quantized to the nearest correct symbol and used to estimate the new channel matrix. In order to reduce the effects of noise, the channel matrix used for the next iteration, is a weighted average of the old and new channel matrix estimations. In the investigations the ratio varies between a ratio of 1:8 and 1:32, along the training sequence. During data transmission the ratio depends on how big a change the previous and current estimations were from their previous values. The estimation which is less of a change is given a larger weight.

C. Simulation Platform

Simulation during the design of this algorithm was performed using ITPP [7]. It is used due to its many toolboxes which resemble MATLAB. Being in C++, it is well suited to alternate number representations such as integer types. This helps the eventual implementation in FPGA. Such an approach is necessary because only fixed point numbers (and not floating point) are used in FPGA. Because of this fixed point representation, further simulations are performed to investigate the effect of quantization on the performance of the algorithm in various stages of the receiver module.

D. FPGA Verification Platform

When performing experiments in the FPGA several other modules are included in order to visualize and export the obtained results. One of the main parts involves the use of a NIOS II soft-core processor which runs uCLinux and an embedded web server. uCLinux is selected due to its robust and high performance networking capability. On a normal browser an interface for the results is displayed over HTTP using scalable vector graphics (SVG).

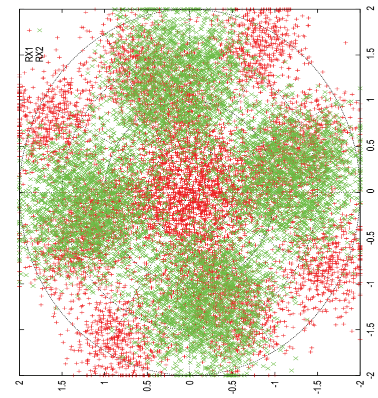

Fig. 6. Received Signal Constellation

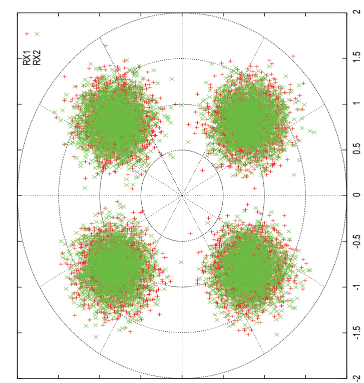

Fig. 7. Decoded Signal Constellation

On this interface various stages of the decoding and transmission stages can be displayed including the time domain signal, the received signal constellation, the decoded signal constellation and the estimated channel matrices. New data can be displayed every 60-100ms allowing for almost real-time visualization. Common duraction of each section of data is 10ms. Fig. 6 and 7 demonstrate the visualization capabilities of the developed system. In Fig. 6 the received signal constellation is shown (before ML decoding). In Fig. 7 the decoded signal constellation is presented. The presented results assume an SNR of 6dB.

In addition to visualization, the data for each plot can be fetched for further offline analysis. This is accomplished using the capabilities of the web server interface.

Also, the data for each plot can be fetched for further offline analysis, by retrieving the data used by the web browser interface.

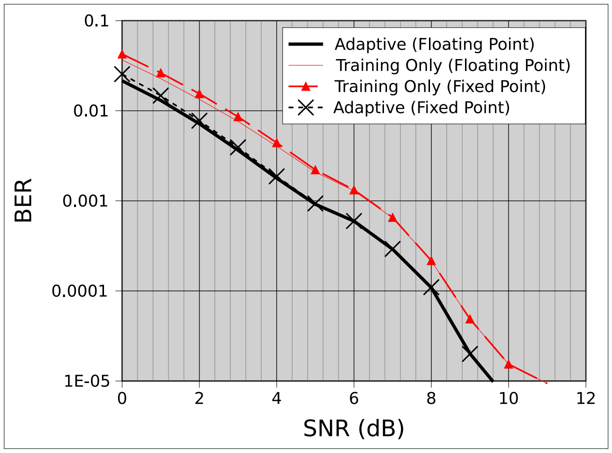

Fig. 8. BER Performance of training based only and adaptive algorithm with quantization

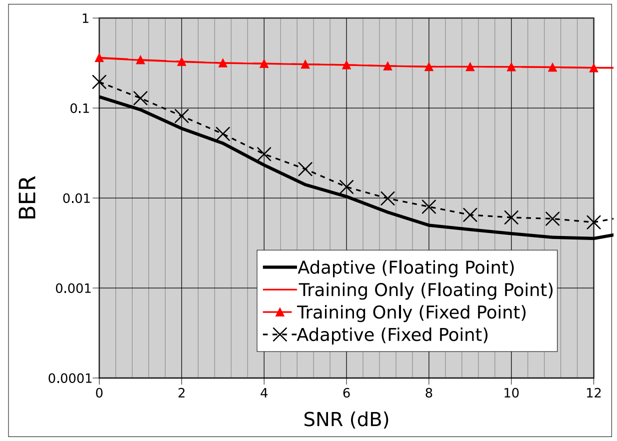

Fig. 9. BER Performance of training based only and adaptive algorithm under block fading (period = 256 symbols)

III. Algorithm Validation

Before implementing in FPGA various algorithms are verified by computer simulations. For each set of simulation results we evaluate the following:

(i) Performance of the training-based MIMO channel estimation algorithm
(ii) Performance of the semi-blind estimation algorithm
(iii) The effect of quantization (due to fixed point operation in FPGA).

Identical received signals are used and decoded using the two separate receiver blocks. For the first 2 simulations 64bits of training sequence, 32kbits of data and SNR values of -3dB-10dB are used. To calculate Bit Error Rate (BER) 20000 data packets are sent per simulation (total 640k bits of data). Fig. 8 shows a comparison between the performance of the training sequence only (dashed line) and the adaptive algorithm (continuous line), and quantization effects. In this simulation the channel remains static during each packet. The BER results indicate a better performance of the adaptive algorithm. This is due to the channel matrix averaging applied in this algorithm. In this figure we also show the same simulation, but with quantization for the input signals and fixed point calculations being used. There are two lines for each algorithm, the additional lines being solid with X quantized adaptive and the dashed with triangles being the quantized non adaptive algorithm. A quantization of 12 bits (real + imaginary) for both the received signals, as well as a quantization of 24 bits for the channel matrix was used. The quantized algorithms and inputs show results which are only slightly worse than the non-quantized and floating point versions.

In the next experiment we test the performance of the training based and semi-blind channel estimation algorithms under a block fading scenario. In this case we use a block fading period of 256 symbols. The BER of the non adaptive model is quite high, however the adaptive algorithm shows better performance. This is observed in Fig. 9, using the same line convention as for Fig. 8.

IV. Conclusions

This paper has reported on investigations into semi-blind channel estimation for a 2x2 MIMO system developed in FPGA that uses the Alamouti coding scheme. Its operation has been investigated using a channel emulator, for both fixed and fading signal cases. A special attention has been paid to quantization errors caused by fixed-point operation performed in the FPGA. In all of the investigated cases, the obtained results indicate that the proposed semi-blind algorithm is superior to the training-based MIMO channel estimation algorithm.

Acknowledgment

The authors acknowledge the financial support of National ICT Australia (NICTA). NICTA is funded by the Australian Governments *Backing Australias Ability initiative, in part through the Australian Research Council. The first author acknowledges the financial support in the form of Morris Gunn Scholarship.*

References

[1] *G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas,"* Wirel. Pers. Commun.*, vol. 6, no. 3, pp. 311–335, 1998.*

[2] *D. Gesbert, M. Shafi, D.-s. Shiu, P. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems,"* IEEE Selected Areas in Communications Journal*, vol. 21, no. 3, pp. 281–302, Apr. 2003.*

[3] *M. Biguesh and A. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals,"* IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 884–893, Mar. 2006.*

[4] *D. Zheng-cong, T. Bin, and L. Ke, "Adaptive MIMO time-varying channel equalization using particle filtering," -, vol. 1, pp. 123–126, Sept. 2005.*

[5] *B. Vuetic and J. Yuan,* Space-Time Coding. *Chichester, UK: Wiley, 2003.*

[6] *K. Bialkowski, U. Uthansakul, M. Bialkowski, and A. Postula, "Design of MIMO Testbed with an FPGA Board for Fast Signal Processing," in* International Conference on Wireless Broadband and Ultra Wideband Communications*, Sydney, Mar. 2006.*

[7] *"ITPP C++ library of mathematical, signal processing, speech processing and communication classes and functions," http://itpp.sourceforge.net/, 2007.*