The Effect of Using Channel Equalizer in The SDR Modem

Dr. Sabah N. Hussein
Department of computer engineering techniques
College of Electrical and Electronic Engineering Techniques
Baghdad, Iraq
drsabah2004@yahoo.com

Raghad S. Majeed
Department of computer engineering techniques
College of Electrical and Electronic Engineering Techniques
Baghdad, Iraq
raghad_eng89@yahoo.com

Abstract — The channel equalization is a technique allowing to remove the inter-symbol interference (ISI) in the SDR receiver caused by the radio-mobile channel. In this paper, present a design and implementation of modem with 16-QAM modulation, convolution, interleaved circuit, differential coding and adaptive equalizer using LMS algorithm based of SDR, using MATLAB system generator model. The results show that LMS equalizer gives a good estimation on the proposed channel model. The hardware implementation is done in FPGA board kit, which has shown a promising foundation for developing coding, modulating and other circuit of modems circuits.

Keywords — FPGA; LMS equalizer; ISI; SDR modem; 16-QAM.

I. INTRODUCTION

Software defined radios (SDR) are highly configurable hardware platforms that provide the technology for realizing the rapidly expanding future generation digital wireless communication infrastructure [1]. In many practical communication systems, data is transmitted over a channel with inter symbol interference (ISI). To reduce ISI in proposed SDR system, two methods are suggested that are convolution code and least mean square error algorithms [2]. Convolution code known also as error correcting code added redundant bits to the information transmitted bits to allow the receiver to detect and correct a limited number of errors occurring anywhere in the transmitter signal. Different studies that are combined convolution code in SDR system like in [3]. Equalizer gives the inverse of the channel to the received signal and combination of channel and equalizer gives a flat frequency response and linear phase [4]. The LMS algorithm is a type of the adaptive filter used to discover the filter coefficients in the adaptive manner that is used to model the inverse channel and overcome ISI problem. There are different studies that are combined LMS equalizer in SDR system like in [5]. The reasons for choice LMS algorithm are simplicity; low computational complexity and better performance in may run environment [6].

The paper is organized as following: Section II provides an overview of 16-qam modulation/de-modulation, Section III provides an overview of the convolutional codes, Section IV the interleaver and de-interleaver, Section V provides an overview of the differential coding, Section VI provides an overview of the LMS equalizer algorithms, Section VII provides the block diagram of the 16-QAM SDR system, Section VIII provides the simulation results, Section IX conclusion.

II. THE 16-QAM MODULATION/DEMODULATION

QAM is one of the widely used modulation techniques because of its efficiency in power and bandwidth. The constellation diagram of 16-QAM in contain 16 different symbols each having a different real and imaginary component. Each constellation point can represent four bits, with two bits on the I axis and two on the Q axis. The 4 bits Gray coded that represent one point in the constellation diagram can be regarded also as two of two bits words on I-axis, and Q-axis respectively as shown in Fig (1) [7].
The received complex coded sequence is:
\[ y = x + n \]  (1)

\( x \) is the data complex sequence in the form of:
\[ \alpha_{16 \text{QAM}} = \{-1 \pm j, -1 \pm 3j, 3 \pm j, 3 \pm 3j\} \]  (2)

In a demodulation the Maximum A posteriori Probability (MAP) method has been used as soft bit detection for 16-QAM. This method usually maximizes the probability that assume the bit \( b_m \) was transmitted given \( y \) received:
\[ P(b_m|y) = \frac{P(y|b_m)P(b_m)}{P(y)} \]  (3)

The detail description of soft bit detection is summarizing in [8].

The soft bit for bit \( b_0 \) is:
\[ sb(b_0) = \begin{cases} 2(y_r + 1), & y_r < -2 \\ y_r, & -2 \leq y_r < 2 \\ 2(y_r - 1), & y_r > 2 \end{cases} \]  (4)

And for bit \( b_1 \) is:
\[ sb(b_1) = \begin{cases} y_r + 2, & y_r \leq 0 \\ -y_r + 2, & y_r > 0 \end{cases} \]  (5)

The soft bit for \( b_2 \) is similar to soft bit for \( b_0 \) except the resolutions are based on the imaginary component and \( b_3 \) is similar to \( b_1 \) but also are based on the imaginary component.

**III. THE CONVOLUTIONAL CODES**

Coding is a technique where redundancy is added to the original bit sequence to increase the reliability of the communication [9]. The following parameters that can be used for convolution code are summarized as: code rate is \( 1/2 \), constraint length \( K = 3 \), and Generator polynomial is \( G = [7 \ 6] \). Fig.2 shows convolution encoder with given generator polynomial.

![Convolution Encoder](image)

**A. Viterbi Decoding of Convolution Codes**

Viterbi Decoder algorithm is used to recover the information sequence at the receiver side. Hard decision decoder with Hamming distance measure has been used in this paper. The receiver employs a trellis based maximum likelihood Viterbi decoder which decodes the input bits to obtain the information bits. The trellis length is chosen to be 5 times the constraint length [10]. Fig (3) shows an example of Viterbi algorithm and how the information is recovered.

![Viterbi Algorithm Example](image)

**IV. THE INTERLEAVER / DE-INTERLEAVER**

A method for making data recovering more efficient by rearranging or renumbering. The interleaver/de-interleaver is used to reduce the effects of long burst errors. Simple random error correction code by rearranges the elements of its input vector using a random permutation [11].

**V. THE DIFFERENTIAL CODING**

Bit streams through transmitter can be un-intentionally inverted. Most signal processing circuits cannot know if the stream bit is inverted or not. Differential Encoding is used to protect against this possibility [12]. It is one of the simplest forms of error protection coding done on a baseband sequence prior to modulation. Supposing that \( x_i \) is a bit intended for transmission, and \( y_i \) is a bit actually transmitted (differentially encoded) [12], if
\[ y_i = y_{i-1} + x_i \]  (6)
Is transmitted, then on the decoding side,

\[ x_i = y_{i-1} + y_i \]

(7)
can be reconstructed, where + is modulo-2 addition.

VI. THE LMS EQUALIZER

The structure of the adaptive filter is shown in Fig. (4). Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that is related to producing the least mean squares of the error signal [14]. The LMS algorithm performs the following operations to update the coefficients of the FIR filter:

1. Calculate the output signal \( Y(K) \) of the FIR filter. The output of the filter represents an estimate of the desired response. \( Y(K) \) is calculated as the convolution of the weight vector and the input vector:

\[ Y(k) = \sum_{n=0}^{L-1} W_n(k)x(k-n) = W^T(k)x(k) \]

(8)

2. The error signal \( e(k) \), is estimation error defined as the difference between the estimated response and the desired response:

\[ e(k) = d(k) - y(k) \]

(9)

3. The error signal and the input signal are applied to the weight update algorithm to updates the filter coefficients [13].

VI. THE 16-QAM SDR BLOCK DIAGRAM

The design of blocks of Fig. 5 can be explain based on system generator of Xilinx that work under the environment of MATLAB Simulink for FPGA design. The Past experience with Xilinx FPGA or Hardware properties Languages (HDLs) is not needed when using System Generator. All of the downstream FPGA implementation procedures including synthesis, position automatically executed to generate an FPGA programming.

A. The main blocks of transmitter section are:

- Random Binary Signal Generation
  The Random integer block in MATLAB Simulink is used as a stream binary signal with 2 M-ary number.
- Convolution Encoder
The convolution encoder Xilinx IP core of system generator has been used that have native rate of 1/2, a constraint length equal to 3, and generator polynomials codes G1=110 and G2=111. Fig. 6 shows convolution encoder system [15].

- **Parallel to Serial**
  - The parallel to serial conversion has been done using a special converter serial to parallel available in system generator whose block layout is shown in Fig. 7. In this Block, the parallel two bits output data of convolution encode are converted to serial streams bits.

- **Interleaver encoder**
  - The main idea of the Random Interleaver is rearranges the elements of its input vector using a random permutation. The output of convolution code is firstly converted into serial bits and then passing through a random interleaver that is a one-to-one permutation map according to random labeling sequence after the data converted into vector mode.

- **Differential Encoder**
  - This encoder has been done using one delay with the logical (exclusive OR) components as shown in Fig. 9.

- **Serial to Parallel converter**
  - This block has been used to convert the Din serial bits to D4 bit parallel 4 bits of data in the form d (3:0) using shift registers as shown in Fig. 10.

- **16-QAM Mapping**
  - Each parallel four bits generated from the serial to parallel section are mapped using the 16-QAM constellation. The four
coding values (±1 to ±3) are stored in a ROM memory block. The block diagram of 16-QAM mapping is demonstrated in Fig. 11 [6].

Fig. 11. The 16 QAM Mapping.

B. The main blocks of channel model:

To modeling the multipath channel two taps of coefficients of 0.1 and 0.9 respectively have been used with AWGN. The signal to noise ratio has been set to (SNR= 40 dB).

Fig. 12. The block diagram of channel model.

C. The main blocks of equalizer filter:

The LMS adaptive equalizer algorithm is shown in Fig. 13, in this Fig. x represent the desired signal that used for training, and y complex represent the actual samples. The stream sample pass through shift register to select five samples as shown in Fig. 14, the five samples are entered in parallel to the LMS filter to update weight. To update weight firstly multiplied Yk with Mu by complex multiplication then adding to the older weight to produce the new weight as shown in Fig. 15. Output of LMS filter that is feedback to subtract with the next complex desired signal to produce the error.

Fig. 13. The block diagram of the equalizer filter.

Fig. 14. The block diagram of the shift register
D. The main blocks of receiver section:

- **16 QAM De-Mapping**
  The De-Mapping has been performed by assigning the received I-Q signals location to the nearest point in the I-Q constellation using soft bit algorithm. Figs. 16 and 17 show the soft bit decision circuit for $b_0$ and $b_1$ respectively. Soft bit decision of $b_2$ and $b_3$ has been built in the same manner as $b_0$ and $b_1$ respectively [7].

- **Parallel to Serial Conversion**
  The parallel four bits output from 16 QAM De-mapping has been converted to serial stream bits using parallel to serial converter as shown in Fig. 18.

- **Differential Decoder**
  Fig. 19 shows the differential decoder circuit. Using one bit delay block, and one block exclusive OR components [3].

- **Random De-interleaver**
  The output of the differential decoder is stream bit and each four bits serial that convert to the parallel with re-arrange the sequence for each bit it's the inverse for the interleaver encoder and the output is serial bit.

- **Serial to parallel**
  The serial to parallel conversion has been used to convert serial data to parallel two bits streams $d(1:0)$. The slice block has been used to select one bit, the upper slice select $d(0)$ and the lower slice select $d(1)$ as shown in Fig. 21.

- **Viterbi decoder**
  Viterbi decoder Xilinx IP core version7 has been used to recover information bits as shown in Fig. 22. Viterbi decoder has the same parameter setting of convolution encoder to be consistent with it [15].
VIII. THE SIMULATION RESULTS

The verification of the implementation has been done via system generator. Fig. 22 shows the time waveform for the error signal between the actual and desired signal during the training mode for \( \mu = 0.001 \) and \( \mu = 0.006 \) respectively. From this Fig. it can be seen that increasing the step size (\( \mu \)) will decrease the time of training mode and converge the signal to the desired value in very fast. In other word increasing the step size will decrease the resolution of the recovered signal and hence decreased the performance of the error rate.

Fig. 23. The time waveform of the error between the actual and desired signals during the training mode for \( \mu = 0.001 \) and \( \mu = 0.006 \) respectively.

Figs. 24 and 25 shows the constellation diagram of 16-QAM for transmitter and receiver side respectively.

Fig. 24. The time waveforms of transmitter side.

Fig. 25. The time waveform of the error between the actual and desired signals during the training mode for \( \mu = 0.001 \) and \( \mu = 0.006 \) respectively.

Fig. 26 and 27 show the simulation results at a transmitter and receiver sides respectively.

Fig. 26. The time waveforms of receiver side.

Fig. 27. constellation diagram of 16 QAM at the output of the equalizer.
Table (I) shows resource utilization and operating frequency.

Table (I) Resource utilization and operating frequency

<table>
<thead>
<tr>
<th>Device Utilization Summary</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of 32-bit Multipliers</td>
<td>2,119</td>
<td>30,770</td>
<td>7%</td>
</tr>
<tr>
<td>Number of multipliers</td>
<td>2,115</td>
<td>30,770</td>
<td>7%</td>
</tr>
<tr>
<td>Number of RAM Chips</td>
<td>1,767</td>
<td>6,346</td>
<td>11%</td>
</tr>
<tr>
<td>Number of RAM芯片 containing only relational logic</td>
<td>1,767</td>
<td>6,346</td>
<td>11%</td>
</tr>
<tr>
<td>Number of RAM芯片 containing only relational logic</td>
<td>0</td>
<td>1,767</td>
<td>0%</td>
</tr>
<tr>
<td>Total number of LUTs</td>
<td>3,707</td>
<td>20,720</td>
<td>7%</td>
</tr>
<tr>
<td>Number used as logic</td>
<td>1,488</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as inputs to combiner</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as inputs to combiner</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>4</td>
<td>44</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>1</td>
<td>0%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>3</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Number of embedded cells</td>
<td>0</td>
<td>22</td>
<td>1%</td>
</tr>
<tr>
<td>Average Power of Non-Calculators</td>
<td>0.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IX. CONCLUSION

In this work a proposed SDR communication system, the system has been designed and implemented using system generator tools to check the effect of using convolution code and efficient LMS linear equalizer. The SDR system generator gives flexibility and optimal in communication system design. The hardware has been implemented on the Xilinx Virtex-4 FPGA using VHDL. A comparison of our proposed work with a conventional LUT-based method and also with a recent work show significant improvement on resource utilization and operating frequency as shown in Table (I). The simulation results show that the LMS filter is good estimation for the channel and the error is zero between the desired and actual signal. Also the results show that the system is synchronized between each component and can be realized in life day as SDR system.

REFERENCES


