Comparison of Alamouti and STS implementations using a software defined radio test bed

Shinhan Wee  
*University of Wollongong*

Montserrat B. Ros  
*University of Wollongong*, montse@uow.edu.au

Peter James Vial  
*University of Wollongong*, peterv@uow.edu.au

Follow this and additional works at: [https://ro.uow.edu.au/infopapers](https://ro.uow.edu.au/infopapers)

Part of the Physical Sciences and Mathematics Commons

**Recommended Citation**  
Wee, Shinhan; Ros, Montserrat B.; and Vial, Peter James: Comparison of Alamouti and STS implementations using a software defined radio test bed 2009, 1-8.  

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Comparison of Alamouti and STS implementations using a software defined radio test bed

Abstract
A Software Defined Radio test bed using the Gnu Radio project was installed on Unix computers and modified so that estimates of the channel state coefficients were taken for a Multiple Input Multiple output (MIMO) system to take advantage of space time transmission at a frequency of 2.4GHz. Two MIMO techniques were implemented, one based on Alamouti and the other on Space Time Spreading. The technique used to estimate channel coefficients is described. The Bit Error Rate performance was compared, when using Binary Phase Shift Keying, between the two systems and it was found that the STS system performed better when one of the two transmit channels experienced a large fading event.

Keywords
Comparison, Alamouti, STS, implementations, using, software, defined, radio, test, bed

Disciplines
Physical Sciences and Mathematics

Publication Details

This conference paper is available at Research Online: https://ro.uow.edu.au/infopapers/1455
Comparison of Alamouti and STS implementations using a Software Defined Radio Test Bed

Shinhan Wee, Montse Ros, Peter James Vial
University of Wollongong

Abstract
A Software Defined Radio test bed using the Gnu Radio project was installed on Unix computers and modified so that estimates of the channel state coefficients were taken for a Multiple Input Multiple output (MIMO) system to take advantage of space time transmission at a frequency of 2.4GHz. Two MIMO techniques were implemented, one based on Alamouti and the other on Space Time Spreading. The technique used to estimate channel coefficients is described. The Bit Error Rate performance was compared, when using Binary Phase Shift Keying, between the two systems and it was found that the STS system performed better when one of the two transmit channels experienced a large fading event.

Index Terms—Gnu Radio, Alamouti, Space Time Spreading, MIMO, Software Defined Radio

1. Introduction
In a wireless communication system it is difficult to provide reliable data transmission as the wireless channel is prone to time-varying multi-path fading. A technique often employed is the use of multiple receive and / or transmit antennas (providing space diversity if the antennas are separated by approximately ten wavelengths distance) [1].

A hardware and software platform which can be used to provide a MIMO based test bed is the Gnu Radio project [2,3]. This paper describes the implementation of a MIMO Software Define Radio (SDR) system which uses the open sourced software provided by the Gnu Radio project [2,3]. The Gnu radio hardware platform is provided from www.ettus.com [2] where the Gnu radio software can be used [2,3] either on a windows or unix based operating system. The project described in this paper used this equipment to implement the Alamouti and the Space Time Spreading (STS) techniques using Binary Phase Shift Keying (BPSK) proposed in [1] and [4] respectively. It then compares these two systems to each other.

STS is classified as an open loop transmit-diversity system, where there is no knowledge at the transmitter about the state of the channel between the transmitter and the receiver. In both STS and Alamouti systems this channel state information is only needed at the receiver, and estimates of the complex channel coefficients can be found from associated pilot signals sent by the transmitter to the receiver [5, 6]. Pilot signals are implemented using other orthogonal spreading codes from the Walsh Hadamard spreading sequence family.

Section 2 and 3 briefly describes the Alamouti and Space Time Spreading (STS) techniques, respectively. Section 4 briefly describes the Software Defined Radio equipment used in the test bed. Section 5 describes the technique employed to estimate the channel coefficients. Section 6 outlines the design flows of the Alamouti and STS implementations. Section 7 provides results comparing the BER performance of Alamouti, STS and a normal MIMO Space Time system which transmits data on both antennas at different times. Section 8 concludes the paper and provides future work.

2. Alamouti
The Alamouti scheme [1] is a transmit diversity technique that may be applied to MIMO systems. The scheme can be generalized to two transmit antennas and M receive antennas to provide a diversity order of 2M. The Alamouti scheme can utilize both space and time diversity at the transmitter and is considered to be the simplest available space-time diversity scheme to be implemented into a MIMO wireless system.

2.1. The Alamouti Transmitter
In a given symbol period, two data symbols are simultaneously transmitted from the two antennas. The symbol transmitted from antenna0 and antenna1 are denoted by $s_0$ and from $s_1$ respectively. During the next symbol period $(t + T)$, the signal $-s_1^*$ is transmitted from antenna0 and signal $s_0^*$ is transmitted from antenna1.
where * is the complex conjugate operator and T is the symbol duration \([1, 7]\).

The complex conjugate operation in the second symbol period ensures that orthogonal signals are transmitted from the two antennas. Since the signals are orthogonal to one another the chance of correlation between the two signals is minimized during transmission. The sequence is shown in Table 1.

<table>
<thead>
<tr>
<th>Symbol Period</th>
<th>Time</th>
<th>Antenna 0</th>
<th>Antenna 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(t)</td>
<td>(s_0)</td>
<td>(s_1)</td>
</tr>
<tr>
<td>2</td>
<td>(t + T)</td>
<td>(-s_1^*)</td>
<td>(s_0^*)</td>
</tr>
</tbody>
</table>

**Table 1** Transmission sequence of Alamouti scheme

From Table 1, it can be observed that two symbols are transmitted over two symbol periods and effectively only one data symbol is transmitted per symbol period. Therefore, for a packet length of four bits, four symbol periods will be needed to complete the transmission using the Alamouti scheme.

### 2.2. The Alamouti Receiver

In the receiver at each symbol period, a superposition signal of \(s_0\) and \(s_1\) will be received. This signal can be represented by the following equations \([1, 7]\):

At time \(t\): 
\[
    r_0 = h_0 s_0 + h_1 s_1 + n_0 \tag{1}
\]

At time \(t + T\): 
\[
    r_1 = -h_0 s_1^* + h_1 s_0^* + n_1 \tag{2}
\]

Where \(s_0\) and \(s_1\) are the signals sent by antenna0 and antenna1, \(r_0\) and \(r_1\) are the received signals at symbol period 1 and period 2, \(h_0\) and \(h_1\) are the estimated channel coefficients of antenna0 and antenna1, \(n_0\) and \(n_1\) are the channel noise for channel 0 and channel 1 respectively and * denotes the complex conjugate.

Assuming a good signal to noise ratio (SNR), (e.g. by transmitting at a higher power) the channel noise can be ignored:

At time \(t\): 
\[
    r_0 = h_0 s_0 + h_1 s_1 \tag{3}
\]

At time \(t + T\): 
\[
    r_1 = -h_0 s_1^* + h_1 s_0^* \tag{4}
\]

The output from the combiner, \(\tilde{s}_0\) and \(\tilde{s}_1\) will finally be put through a Maximum Likelihood detector which, for each of the signals, uses a decision rule to recover the original \(s_0\) and \(s_1\) data. Since the modulation scheme used in this project is Binary Phase Shift Keying (BPSK), a bit zero or a bit one will be represented as \(-1 + 0j\) or \(1 + 0j\) as illustrated in Figure 1.

**Figure 1** BPSK constellation diagram

Finally, the original bits \(s_0\) and \(s_1\) can be reconstructed at the receiver using the following decision rules:

\[
\text{if } (\Re(\tilde{s}_0)) \geq 0 \text{ then } s_0 = \text{ bit } 1 \\
\text{if } (\Re(\tilde{s}_0)) < 0 \text{ then } s_0 = \text{ bit } 0 \\
\text{if } (\Re(\tilde{s}_1)) \geq 0 \text{ then } s_1 = \text{ bit } 1 \\
\text{if } (\Re(\tilde{s}_1)) < 0 \text{ then } s_1 = \text{ bit } 0
\] \(\tag{7}\)

### 3. Space Time Spreading

The STS technique is outlined in [8] which describes the situation for the test bed when the STS system is in the presence of MAI (Multiple Access Interference) due to other STS users. In this section, we re-introduce the equations provided in [8] comparison between Alamouti and STS. This paper does not consider situations where MAI is experienced but does consider the effect of deep fades on one of the antenna branches. As indicated in [8] it is expected that as the STS technique includes information about the two data bits transmitted on both antennas, it can still transmit this data with a moderate degradation in BER performance when one of the antenna paths experiences a deep fade. This contrasts with the Alamouti schema \([1]\) considered in this paper where one or more data bits would be lost depending on the length in time of the fade. In practical implementations of the Alamouti system encountering a deep fade it is expected that forward error correction coding would be used to recover the data or the data would need to be
retransmitted, which is an inefficient use of scarce wireless resources (bandwidth).

3.1. The STS Transmitter

Before transmitting two data bits, the STS scheme [6, 9] separates them into odd and even symbols, identified as $b_1$ and $b_2$. This is then encoded in the following manner [8]:

Data to be transmitted from antenna0:

$$ t_1 = \left( \frac{1}{\sqrt{2}} \right) (b_1 c_1 + b_2 c_2) $$ (8)

Data to be transmitted from antenna1:

$$ t_2 = \left( \frac{1}{\sqrt{2}} \right) (b_2 c_1 - b_1 c_2) $$ (9)

Here the symbols $c_1$ and $c_2$ represent the orthogonal spreading codes used. The factor $1/\sqrt{2}$ used in both equations allows for signal power normalization so that the total transmission power of the transmit antennas is directly comparable to that required for one transmit antenna.

3.2. The STS Receiver

At the receiver, a superposition of the signals of $t_1$ and $t_2$ will be received in each symbol period. The following notations are used in [8][10]:

$$ b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad n = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} $$ (11)

Here, $(\bullet)^H$ stands for the Hermitian or Conjugate transpose, $n$ is a N x 1 vector representing the channel noise, and $h_0$ and $h_1$ are the channel complex coefficients of antenna0 and antenna1. Referring to equations 8 and 9, the vector $d$ (received signal) is given by[8]:

$$ d = \frac{1}{\sqrt{2}} H b + n $$ (12)

Expanding this matrix equation results in:

$$ d_1 = \frac{1}{\sqrt{2}} \left( h_1 b_1 + h_2 b_2 \right) + \xi_1^H n $$ (13)

$$ d_2 = \frac{1}{\sqrt{2}} \left( - h_1 b_1 + h_2 b_2 \right) + \xi_2^H n $$ (14)

For a system with no noise (or very high signal power) we can ignore the noise term and get:

$$ d_1 = \frac{1}{\sqrt{2}} \left( h_1 b_1 + h_2 b_2 \right) $$ (15)

$$ d_2 = \frac{1}{\sqrt{2}} \left( - h_1 b_1 + h_2 b_2 \right) $$ (16)

From equations 15 and 16 we then get $\tilde{b}_1$ and $\tilde{b}_2$ (in complex form):

$$ \tilde{b}_1 = \frac{1}{\sqrt{2}} \left( h_1 d_1 - h_2 d_2 \right) $$ (17)

$$ \tilde{b}_2 = \frac{1}{\sqrt{2}} \left( h_2 d_2 - h_1 d_1 \right) $$ (18)

Lastly, $b_1$ and $b_2$ at every symbol period can be reconstructed using the decision rules:

If $(\text{Re} \{ \tilde{b}_1 \}) \geq 0$ then $b_1 = \text{bit} 1$
If $(\text{Re} \{ \tilde{b}_1 \}) \leq 0$ then $b_1 = \text{bit} 0$
If $(\text{Re} \{ \tilde{b}_2 \}) \geq 0$ then $b_1 = \text{bit} 1$
If $(\text{Re} \{ \tilde{b}_2 \}) \leq 0$ then $b_1 = \text{bit} 0$ (19)

4. The Software Defined Radio

Universal Software Radio Peripheral (USRP) is an FPGA-based hardware component that creates the possibility of developing software defined radio (SDR) by acting as an RF frontend for a computer running GNU Radio. The USRP converts radio waves picked up by an antenna into a digital form suitable for processing on the host computer [9].

The USRP is produced by ETTUS Research LLC especially for the use of GNU Radio software and contains four high-speed, 64 mega samples-per-second (MS/s), 12-bit analog-to-digital (ADC) converters and four high-speed, 128 MS/s, 14-bit digital-to-analog (DAC) converters. Support circuitry, including a high-speed USB 2.0 interface, is also included in a USRP [2, 3].

The RFX2400 Transceiver daughter boards are specialized boards used to hold the RF transmitter and receiver interface of the USRP. These boards could be used for any software defined radio operating in the frequency range of 2.3-2.9 GHz. This paper uses a frequency range of 2.4 to 2.483 GHz.
The antennas used by the SDR are the PATCH2400, manufactured specifically for use with the RFX2400 transceivers. The PATCH2400 is a vertically polarized antenna; rated to have a gain of 7dBi for ISM band frequencies between the frequency ranges of 2400-2480 MHz [11,12]. Table 2 lists the hardware components required for a MIMO system using the SDR hardware.

<table>
<thead>
<tr>
<th>Hardware components</th>
<th>Quantity</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computers (Ubuntu OS)</td>
<td>2</td>
<td>Both computers must have GNU Radio installed. One PC used as transmitter and another PC used as receiver</td>
</tr>
<tr>
<td>USRP main boards</td>
<td>2</td>
<td>One transmitting USRP and one receiving USRP</td>
</tr>
<tr>
<td>RFX2400 boards</td>
<td>3</td>
<td>Two TX slots for two transmit antennas and one RX slot for one receive antenna</td>
</tr>
<tr>
<td>PATCH2400 antennas</td>
<td>3</td>
<td>Two transmit antennas and one receive antenna</td>
</tr>
</tbody>
</table>

Table 2: MIMO systems hardware requirements [8]

GNU Radio has libraries for many common software radio needs, including various modulations (DBPSK was modified to BPSK for this test bed), signal processing constructs (optimized filters, FFTs, equalizers, timing recovery), and scheduling. GNU Radio is a very flexible system and utilizes two programming languages, C++ and Python [13]. Three layers are defined in GNU Radio, the Application layer, the Python layer, and the C++ layer. These were previously described in [8]. All modifications were performed by modifying the Python code layer.

The block diagram shown in Figure 5 shows the complete data flow path between the different signal processing blocks used in the transmitter SDR. The arrows representing the signaling links connect each block using the `fg.connect` flow chart function in the Python layer.
The operation is fully described in [8], though one should note that the BPSK modulation scheme was achieved by removing the difference algorithm employed in the original DBPSK transmission technique which was standard with the Gnu Radio software.

The SDR receiver is expected to receive a superpositioned payload (payload0 and payload1) and attempts to reverse the signal processing that was done in the transmitter SDR. Figure 6, reproduced from [8] shows the block diagram of the connections between the different signal processing blocks.

In the developed space-time-diversity channel estimator, each antenna takes turns to transmit its pilot signal which is also known to the receiver. As a result, the estimator would require two symbol periods to successfully estimate the coefficient of both channels. Since GNU radio applies zero padding to fill up data packets, the pilot signals are chosen as long strings of bit zero to avoid synchronization problems at the receiver. Therefore, at instances where the pilot samples are not synchronized at the receiver, the channel estimation would remain relatively accurate. At the USRP source, a complex vector sink was implemented to collect all complex samples passing through the USRP source to an array called $fs\_usrp\_data$. This complex vector sink is shown in Figure 6 which is reproduced from [8]. All received pilot complex samples are stored in the array $fs\_usrp\_data$ and each bit in the GNU radio platform is represented by 16 complex samples. Therefore, the channel estimation algorithm used in this project can be graphically represented as shown in Figure 7.

In the developed space-time-diversity channel estimator, each antenna takes turns to transmit its pilot signal which is also known to the receiver. As a result, the estimator would require two symbol periods to successfully estimate the coefficient of both channels. Since GNU radio applies zero padding to fill up data packets, the pilot signals are chosen as long strings of bit zero to avoid synchronization problems at the receiver. Therefore, at instances where the pilot samples are not synchronized at the receiver, the channel estimation would remain relatively accurate. At the USRP source, a complex vector sink was implemented to collect all complex samples passing through the USRP source to an array called $fs\_usrp\_data$. This complex vector sink is shown in Figure 6 which is reproduced from [8]. All received pilot complex samples are stored in the array $fs\_usrp\_data$ and each bit in the GNU radio platform is represented by 16 complex samples. Therefore, the channel estimation algorithm used in this project can be graphically represented as shown in Figure 7.

5. STS channel estimator

For STS, in particular, the reconstruction process at the receiver requires an accurate estimation of channel fading coefficient. As such an effective channel estimator must be developed to realize the implementation of the proposed systems. We found that there was no available module in the Gnu Radio Software open source system that allowed these channel estimates. We thus needed to develop our own estimation technique.

In the developed space-time-diversity channel estimator, each antenna takes turns to transmit its pilot signal which is also known to the receiver. As a result, the estimator would require two symbol periods to successfully estimate the coefficient of both channels. Since GNU radio applies zero padding to fill up data packets, the pilot signals are chosen as long strings of bit zero to avoid synchronization problems at the receiver. Therefore, at instances where the pilot samples are not synchronized at the receiver, the channel estimation would remain relatively accurate. At the USRP source, a complex vector sink was implemented to collect all complex samples passing through the USRP source to an array called $fs\_usrp\_data$. This complex vector sink is shown in Figure 6 which is reproduced from [8]. All received pilot complex samples are stored in the array $fs\_usrp\_data$ and each bit in the GNU radio platform is represented by 16 complex samples. Therefore, the channel estimation algorithm used in this project can be graphically represented as shown in Figure 7.

Let the array $fs\_usrp\_data$ be an $m$ by 1 matrix, $B_T$ where the first complex sample in the first sample period is stored in $B_0[0]$, the second complex sample is stored in $B_0[1]$ and the last complex sample is stored in $B_0[m-1]$.

At symbol period 1, only antenna0 is allowed to transmit its pilot signal ($x_0$). In this case:

$$y_0(n) = h_0(n)x_0(n) \quad (20)$$

Where $n=1,2,\ldots,32$. Rearranging:

$$h_0(n) = \frac{y_0(n)}{x_0(n)} \quad (21)$$

And we note that the pilot signal $x_0$ is known at the receiver. Because we choose:

$$x_0(n) = -1 + 0j \quad (22)$$

Where $j$ is the complex number the square root of -1, then the complex samples of $h_0(n)$ (the complex coefficient estimates) are given by:

$$h_0(n) = -y_0(n) \quad \text{(complex samples)} \quad (23)$$

Thus the average channel 0 coefficient ($h_0$) estimated over 32 bits of pilot signal is given by:

$$h_0(n) = \frac{\sum_{i=1}^{32} y_0(n)}{n} \quad (24)$$
Where:
\[ y_0(n) = \frac{\sum_{m=0}^{(n+1)/2} \sqrt{\rho_0[m]}}{n} \]  \hspace{1cm} (25)

Which is the average over 16 complex samples.

The calculations required at symbol period two are the same but applied to antenna1 instead of antenna0. Antenna1 is then allowed to transmit its pilot signal \((\chi_1)\) resulting in the average channel 1 coefficient \((h_1)\) estimated over 32 bits of pilot signal given by:
\[ h_1(n) = \frac{\sum_{n=0}^{2^{16}-1} y_1(n)}{n} \]  \hspace{1cm} (26)

Where:
\[ y_1(n) = \frac{\sum_{m=0}^{(n+1)/2} \sqrt{\rho_1[m]}}{n} \]  \hspace{1cm} (27)

Which is also averaged over 16 complex samples. After the reception of the pilot signals, the channel estimation algorithm will read the complex array \(f_s\) and apply equation 24 to the first symbol period to get an estimate of \(h_0\) and equation 26 to the second symbol period to get an estimate of \(h_1\).

6. SDR implementation

In this section we look at the implementation of the Alamouti and STS systems on the SDR platform.

6.1 Alamouti transmitter

The two data symbols to be transmitted must be user-defined. These data symbols may be modified by the Alamouti transmit algorithm and will be the actual payloads that are finally transmitted through the two antennas. The Alamouti transmit algorithm coded into the transmitter SDR is presented in the flowchart shown in Figure 8.

6.2 Alamouti receiver

Utilizing the channel coefficients estimated by the channel estimator, the original symbols transmitted by both antennas can be recovered using the technique presented by the flowchart shown in Figure 9.

6.3 STS transmitter

The STS transmitter is considered an upgrade to that of the Alamouti transmitter. The STS algorithm is more complex and requires additional processing of the original data stream. The STS scheme also performs spreading of the original data, after which a superimposed signal of the spreaded data will be transmitted through both transmit antennas. Therefore, for the STS transmitter to be working correctly, at least two orthogonal spreading codes are required.

Eight-chip Walsh Hadamard spreading codes were used for the first implementation of the STS algorithm which was then upgraded to incorporate the use of 32-chip spreading codes. The STS transmit algorithm implemented in the transmitter SDR is as follows (see Figure 10):

1) The original data stream defined by the user was split into two separate streams, \(b_1\) and \(b_2\) where \(b_1\) represents the odd bits of the original data stream and \(b_2\) represents the even bits.
2) Two sets of 32 bit zeros are transmitted as pilot signals.
3) After the transmission of the pilot signals, the transmit algorithm was applied as per equation 8 and equation 9, after which the encoded payloads \(f_1\) and
were interleaved and transmitted from antenna0 and antenna1 simultaneously.

6.4 STS receiver

Due to the spreading of $b_1$ and $b_2$ at the transmitter, the received packet at the STS receiver is expected to be a superpositioned stream of data bits. The original symbols $b_1$ and $b_2$ can be recovered every symbol period using the following procedures (see Figure 11):

1) The first and second payloads received at the receiver are the pilot signals. Using the channel estimation algorithm, it is assumed that good channel coefficients were estimated.
2) The next packet received is the actual data payload and the STS algorithm will start reconstructing the original data at every symbol period.
3) By substituting the estimated channel coefficients $h_0$, $h_1$ (complex numbers) and the spreading codes $c_1$, $c_2$ into equation 17 and equation 18, the complex representation of $\tilde{b}_1$ and $\tilde{b}_2$ can be obtained.
4) $\tilde{b}_1$ and $\tilde{b}_2$ are then put through the Maximum Likelihood Decision Rule described in equation (19). The output from the Maximum Likelihood Detector is the reconstructed symbols of $b_1$ and $b_2$.

7. Comparison of Alamouti and STS

The test beds for both Alamouti and STS systems were set up. In addition a normal MIMO system was constructed (which sends data bits at different times on different antennas – not simultaneously). As noise at the receiver is present in both systems, the signal amplitude was varied as the parameter of interest and the BER performance was measured as this signal amplitude was increased. For each point 200 data bits were transmitted. The amplitude levels chosen were 0, 10, 25, 50, 100, 200 and 250 (note that default amplitude in the SDR is set to 12000). Figure 12 shows the test results.

From Figure 12, it is noticeable that the STS system possesses the best performance out of the three tested systems in a noisy environment. This superior result is most likely due to the superimposed signal transmitted by each STS antenna. By superimposing $b_1$ and $b_2$ at the transmitter, the magnitude of the transmitted data is effectively doubled, leading to better resistance against noise. Furthermore, processing gain is obtained through the use of the Walsh Hadamard spreading codes. With each STS data bit represented by a 32-chip sequence, there is more resistance to a burst of errors. A simple bit counting algorithm (FOR loops) is implemented in the STS receiver to count the numbers of + and - in order to make a soft decision on the most probable bit if errors were detected in the reconstructed sequence. The Alamouti system on the other hand, transmits only a single bit of data from both antennas at each symbol.
period. As such, the magnitude of the transmitted data is half of the STS system and no processing gain was obtained since the data was not spread. Consequently, these properties of the Alamouti system make the data more susceptible to a noisy channel as compared to a STS system. However, the Alamouti system still outperforms the MIMO non-simultaneous transmit system, due to the ability of a diversity system to reconstruct the original bits, even if deep fading occurs at one of the channels.

A second experiment was set up to look at the effect of a deep fade on the three systems. The Alamouti and STS systems had deep fades programmed into the receiver (that is, in each case one of the antenna branches results was ignored or set to zero). Two hundred bits were transmitted using both algorithms at each pre-defined amplitude levels of 300, 250, 200, 100, 50, 25, 10, and 0. The results obtained are presented in Figure 13.

From Figure 13, the STS system performed better as compared to the Alamouti system when both systems suffered from a deep fade. The reasons for this are similar to that discussed for the first test results shown in Figure 12. Figure 13 displays the evidence of diversity between both antennas of the Alamouti and STS system. When an error bit occurred on one of the channels, it does not always mean a failure on the other wireless channel as evident from the different BER response of the antennas at the same amplitude level. This experiment shows that for a diversity system: given the same signal power, two channels will experience different fading effects at any time instance. Therefore, if a deep fade is present in one of the wireless channels, the other channel can still reconstruct the original bits leading to a better BER when the system is employed in a noisy environment.

8. Conclusion

A MIMO SDR Alamouti and STS system were constructed using the GNU Radio platform. A system was developed to estimate the individual channel coefficients in the Alamouti and STS test beds using the GNU Radio open source project. The Alamouti and STS systems were compared over two hundred transmitted bits and the STS system was found to provide a lower BER performance. The effect of a deep fade was also investigated and it was found that STS also has a better BER performance than the Alamouti system for the developed test bed. Future work may investigate the comparison of Space Time Block codes within the SDR test bed.

9. References