1. ABSTRACT

In this research paper, a hardware emulator will be implemented for an adaptive Minimum Bit Error Rate (MBER) beamforming algorithm using Altera's Stratix EP1S25 DSP Development Board interfaced with TI's TMS320C6701 DSP Board. A two-element antenna array will be used as the receiver. The performance of the MBER algorithm will be compared to the performance of the widely-used Least Mean Squares (LMS) algorithm.

The MBER algorithm directly minimizes the bit error rate (BER) of a communications system instead of the mean square error (MSE). Studies have shown that the MBER approach achieves better performance than the classical Minimum MSE approach. In this paper, a stochastic gradient based implementation will be used, which allows a sample-by-sample adaptation with a computational complexity similar to the LMS algorithm.

For this testbed, BPSK signals are generated by a vector signal generator and are sampled using the EP1S25's onboard A/D converter. The effects of AWGN, co-channel interference and multipath are simulated using the FPGA board. Digital downconversion is also performed using the FPGA board. The complex weights adjustments for the antenna array are computed by the TI DSP board. Real-time adaptation will be implemented for both the MBER and MMSE approach. The actual BER will be the basis of comparison for the two approaches.

2. INTRODUCTION

2.1 Statement of the Problem

When Chen introduced the Adaptive Minimum Bit-Error-Rate (MBER) solution as an alternative to the Minimum Mean-Square-Error (MMSE) solution for beamforming applications, simulations showed that the MBER solution offers potentially significant performance gains over the MMSE solution. One of the major advantages of the MMSE solution is that it can be easily implemented using the Least Mean Squares (LMS) algorithm or one of its variants. In [10], Chen derived a stochastic gradient-based algorithm, called the Approximate Least Bit-Error-Rate (ALBER) algorithm, which has a computational complexity similar to that of the LMS algorithm, thus further increasing the attractiveness of the MBER solution. However, although the MBER algorithm theoretically has the same complexity as the LMS algorithm, the feasibility of real-time implementation of the algorithm in hardware still needs to be tested. In particular, the use of exponential functions in the weight update equations of the ALBER algorithm might prove to be a limiting factor in the data rates achievable by this algorithm.

2.2 Objectives

In this thesis, the problem of implementing the ALBER algorithm in real-time on existing hardware will be addressed. The objectives of this thesis may be broken down into the following:

1. Compare the memory requirements of the ALBER and the LMS algorithms. Both the data and program memory required for both algorithms will be investigated.

2. Compare the processing power required to achieve the same output data rates for the ALBER and the LMS algorithms using the same hardware setup. In relation to this, the maximum data rates achievable for both algorithms using the same hardware setup will also be determined.

3. Compare the performance of the two algorithms in the presence of different wireless channel distortion effects. The BER will be the primary basis of comparison for the two algorithms.

3. REVIEW OF RELATED LITERATURE

3.1. Characteristics of the Wireless Channel

The wireless channel operates through electromagnetic radiation from the transmitter to the receiver. Ideally, a signal originating from the transmitter would travel to the receiver in a single straight line path. The signal at the receiver would then be a perfect copy of the signal from the transmitter. In the real world, however, this is generally not the case. In passing through the wireless channel, the signal is modified and what arrives at the receiver is the combination of attenuated, reflected, refracted and diffracted versions of the original. The channel also adds noise to the signal, further distorting it.

3.1.1 Noise

Like all other practical communications channels, the wireless channel is corrupted by ambient noise. This noise may be caused by thermal vibrations in the transmitting and receiving antenna, black body radiation from the earth and other warm objects or from celestial sources such as the sun. This noise may be modeled as having a very wide bandwidth and random in nature. [1, 2]

One widely-used model is that of Additive White Gaussian Noise (AWGN), or simply white noise. This noise is characterized as having a flat power spectral density (at least, in the frequency band we are interested in), and induces a Gaussian probability distribution at the output of a linear filter that it is used as an input to. This type of noise is additive to the other signals at the receiver, as its name implies.

3.1.2 Attenuation

In a noiseless, lossless and obstruction-free wireless channel, signals emanating from the transmitter will still experience attenuation as it travels to the receiver. For an isotropic antenna, the power at the receiver is inversely proportional to the square of the distance between the receiver and transmitter, as a direct consequence of the law on conservation of energy. This type of attenuation, also called diffusive attenuation, cannot be remedied and is present at all frequencies.

For the non-ideal channel, attenuation is aggravated by shadowing. Shadowing occurs when there are obstructions, such as buildings and walls, in the signal path between the transmitter and receiver. The signals may diffract off the obstruction boundaries (if the obstructions are not too large), thus preventing total shadowing. The amount of diffraction is dependent on the radio frequency, with lower frequencies diffracting more than higher frequencies, which is why shadowing is more detrimental to high frequency signals. If diffraction is not possible, the signals may be partially absorbed by the material: the larger the depth of the material, the greater the attenuation. Shadow fades may last for several seconds and even minutes, which is why it is also called slow fading. [1]

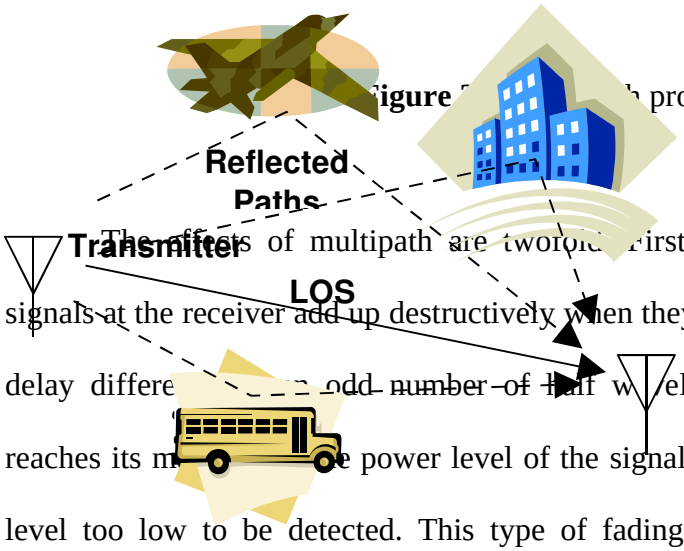
A useful model for representing the path loss effects of the environment is the lognormal shadowing model. This model represents the path loss versus distance relationship with the equation:

$$P_{loss}(d) = 10n\log\left(\frac{d}{d_0}\right) + X \quad \text{(in dB)} \tag{3.1}$$

where P_{loss} is the path loss in dB, d is the distance, d_0 is the distance associated with a reference measurement. The distance power exponent, n, and the parameter X are dependent on the environment. [1]

3.1.3 Multipath

In wireless channels, there exists, most of the time, numerous objects that may reflect the transmitted signals. These reflective surfaces provide multiple paths between the transmitter and the receiver. The propagation delay and loss for each of these paths is generally not identical to that of the other paths. The signal seen by the receiver is the vector sum of all these multipath signals. Figure 3.1 shows an illustration of multipath.



igure propagation

The effects of multipath are twofold. First, multipath may cause fading. The signals at the receiver add up destructively when they are out of phase. In particular, for a delay differe odd number of half w elengths, the destructive interference reaches its m e power level of the signal at the receiver may be reduced to a level too low to be detected. This type of fading is often called frequency-selective fading. The Rayleigh distribution is commonly used to describe the statistical time varying nature of the received signal power when there is no line of sight (LOS) existing between the transmitter and the receiver. If a LOS exists, or if there is a dominant multipath component present, the resulting fluctuations in the signal power are modeled by a Ricean distribution .[1, 2, 3]

Figure 3.2 shows a graph of typical Rayleigh fading as a function of distance. It also shows the effect of slow fading. The received signal *r(t)* is the sum of *m(t)*, which shows slow fading due to shadowing effects, and $r_o(t)$, which illustrates Rayleigh fading. In comparison to slow fading, Rayleigh fading occurs only over very short distances.

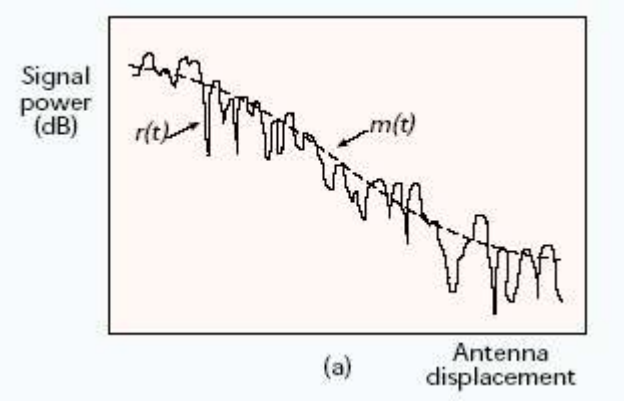


Figure 3.2. Slow fading and Rayleigh fading [4].

The second effect of multipath is to cause delay spread. Delay spread is the time dispersion of the channel. If the delay difference between the multiple paths is on the order of the data symbol intervals, this can result in intersymbol interference (ISI). ISI occurs when the contributions from multiple symbols arrive at the receiver at the same time. The effect of multipath on ISI is particularly significant in high data-rate systems. Correct demodulation may be impossible if there is too much ISI present. [2]

3.1.4 Interference

Interference may be classified into two types. Multiple access interference (MAI) is a result of other signals using the same network. Co-channel interference (CCI) is a result of signals from other communications systems using the same frequency band as the signal transmitted. Like ISI, these interfering signals may add up at the receiver and possibly distort the desired signal beyond recovery. [2]

3.2. Antenna Array Beamforming

Antenna arrays may be used to combat the effects of multipath fading and interfering signals to improve the performance and capacity of wireless communications systems. An antenna array consists of two or more antenna elements connected to a feed network. Each element may be fed identical signals. In this case, the central beam may be made narrow and the sidelobes reduced by appropriately choosing the number of antenna elements and the spacing between these elements. Figure 3.3 shows such an arrangement for a 4-element array and the corresponding radiation pattern.

Alternatively, the phase fed to each element may be varied to optimize the received signal. By using appropriate phase shifts in the array elements, the central beam may be steered in any direction. Figure 3.4 shows such an array with 8 elements and the corresponding radiation pattern. The central beam for this configuration is steered 45 degrees to the left. [5]

In beamforming, both the phase and the amplitude of each antenna element may be varied to optimize the received signal. This is achieved by multiplying the signals at each antenna element by a complex weight. In this configuration, the side lobes and nulls are controlled better. [5]

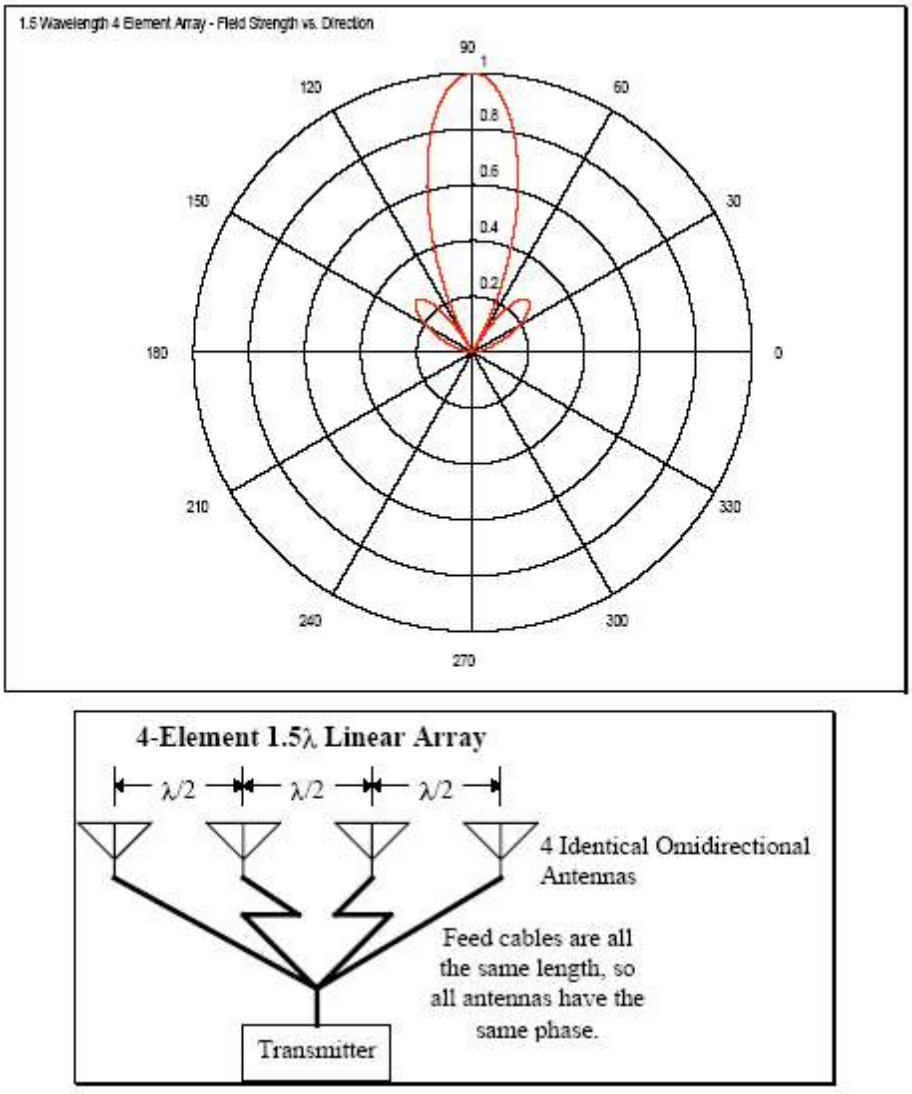
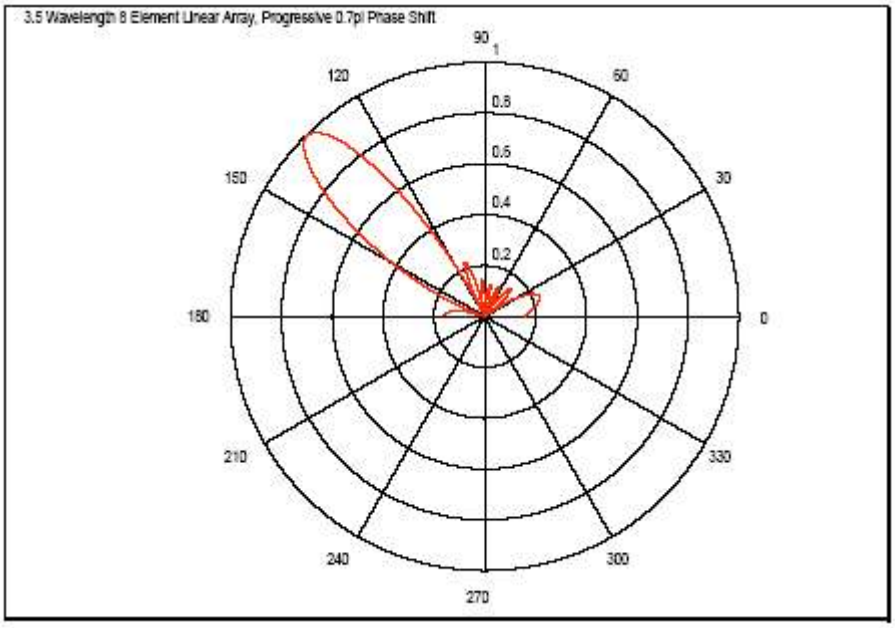


Figure 3.3. Fixed phase antenna array [5].



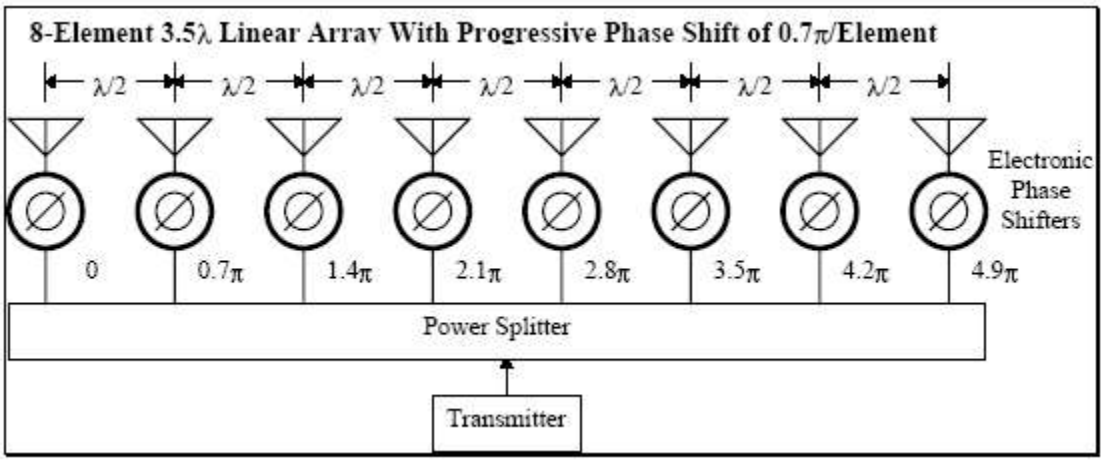


Figure 3.4. Electronically steered array [5].

3.2.1 Filter and Sum Beamforming

In this approach to beamforming, the signals in each element are delayed, multiplied with weights then added. The beamformer delays compensate for the different propagation delays of each multipath so that the signals at the receiver may be added synchronously, while the weights compensate for the different path losses in each

multipath. In this way, the desired signal contributions are added synchronously, resulting in a higher SNR. [6]

3.2.2 Constrained Beamforming

In constrained beamforming, there is prior knowledge about the operating environment and the weights are constrained accordingly. For example, the signals from certain directions may be suppressed or the signal from a desired direction may be maximized by applying the appropriate constraints to the beamforming array complex weights. However, this approach limits the degrees of freedom of the array and involves time-consuming operations, making it unsuited to real-time applications. [6]

3.2.3 Other Applications of Antenna Arrays

Aside from beamforming, antenna arrays may be used in a wide variety of applications where spatial and temporal-spatial filtering is required. When it is desired to know how many signal sources there are and where they are located, super-resolution techniques for antenna arrays may be utilized. Blind deconvolution may be used to reduce the effects of unwanted sources and noise in an unknown system. [6]

3.3. Adaptive Beamforming Arrays

An adaptive beamforming array consists of an antenna array whose weights are adjusted adaptively and is capable of tracking the time-variation of the operating environment (provided it is not too fast-changing) and optimizing the received signal accordingly. [7]

The simplest form of adaptive antennas is the side-lobe canceler. When an interfering signal is present along with the desired signal, this interfering signal may be canceled exactly in the absence of receiver noise. If receiver noise does exist, the side-lobe canceler still performs satisfactorily as long as the desired signal is strong compared to the receiver noise. [7]

The adjustment of the complex weights of each antenna element is done through an adaptive algorithm. Several algorithms for finding the optimum weight coefficients have been developed over the past years. One of the most widely-used of these algorithms is the Least Mean Square (LMS) algorithm. Its popularity is due to its simplicity and satisfactory performance in non-stationary environments. Furthermore, prior knowledge of the operating environment is not needed for adaptation.

Figure 3.5 shows an LMS implementation of an adaptive antenna array with four elements. The input signals to the antenna elements are given by the vector **x**, and the weight coefficients are contained in the weight vector **w**. The outputs of the weight multipliers are added linearly and subtracted from a desired response, *d*, to give the error signal, *e*. For the LMS algorithm, the goal is to minimize the mean square error (MSE), E $[e^2]$.

The optimum weight vector $\mathbf{w}_{opt}$ which gives the minimum mean square error can be computed by utilizing the equation

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{S} \qquad (3.2)$$

where **R** is the autocorrelation matrix of the input and **S** is the cross correlation vector between the input and the desired output. However, accurate estimates of **R** and **S** are generally not available, and the computation of the inverse of **R** is a resource-hungry task,

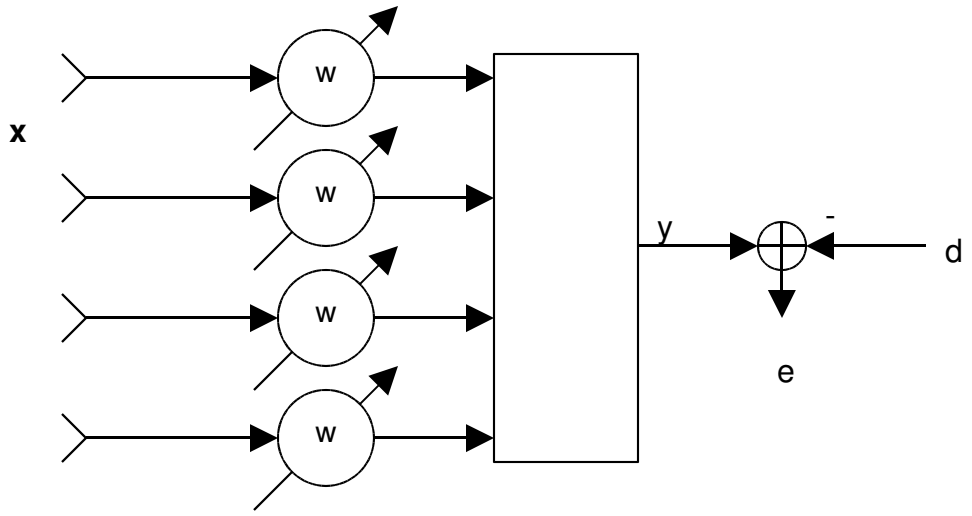


Figure 3.5. LMS implementation. [8]

particularly when the number of antenna elements is large. The LMS algorithm remedies these problems by updating the weight coefficients as each sample of the input arrives. The update equation for the weights is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2?e\mathbf{x}(k) \quad (3.3)$$

where k indicates the iteration number and ? is an appropriately chosen step size.

The above update equation is valid only if the weights are scalars (for example, if we are dealing with BPSK signals). If the weights are complex valued, the modified update equation is

$$\mathbf{w}(k+1) = \mathbf{w}(k)+ 2?e^*\mathbf{x}(k) \quad (3.4)$$

where * indicates conjugation. [7, 8, 9]

It has been shown that this algorithm, despite its simplicity and efficiency, provides near optimum performance in signal environments with slow time variations. The rate of convergence is primarily governed by the step size, which is adjusted to fit the signal environment. Smaller step sizes allow the algorithm to approach the theoretical MMSE more closely at the expense of slower convergence rate.

3.4. Adaptive Minimum Bit Error Rate Approach to Beamforming

Classically, adaptive beamformer algorithms have focused on minimizing the MSE. Their goal is to approach as closely as possible the theoretical MMSE. Due to its simplicity and ease of implementation, the LMS algorithm which was discussed in the previous section, or a variation of it, has been widely employed for adaptive beamformer antenna arrays.

However, in practical communications systems, the achievable bit error rate (BER) is more important than the MSE. The system should be designed to minimize the BER instead of the MSE. Chen *et al* have shown in their studies that the MBER approach offers potentially significant improvements over the classical MSE approaches in terms of BER. [10, 11]

Assume that the communications system consists of M signal sources, transmitting BPSK signals. Without loss of generality, source 1 is assumed to be the desired signal source and the other (M-1) sources are considered to be interference. The linear array consists of L uniformly spaced elements. The input to the array can be expressed as:

$$\mathbf{x}(k) = \bar{\mathbf{x}}(k) + \mathbf{n}(k) = \mathbf{P}\mathbf{b}(k) + \mathbf{n}(k) \tag{3.5}$$

where $\mathbf{n}(k)$ is the complex-valued Gaussian, zero-mean noise vector with variance $2?_n^2$. The output of the array is

$$y(k) = \mathbf{w}^H \mathbf{x}(k) + \mathbf{n}(k) = \mathbf{w}^H \bar{\mathbf{x}}(k) + \mathbf{n}(k) \tag{3.6}$$

By utilizing the conditional p.d.f. of the beamformer's output set, it was shown in [10] that the BER of the beamformer associated with the weight vector $\mathbf{v}$ is given by:

$$P_E(\mathbf{w}) = \frac{2}{N_b} \sum_{q=1}^{N_b/2} Q(g_{q,+}(\mathbf{w})) \tag{3.7}$$

where

$$g_{q,+}(\mathbf{w}) = \frac{\text{sgn}(b_{q,1})\bar{y}_{R,q}^{(+)}}{\sigma_n \sqrt{\mathbf{w}^H \mathbf{w}}} = \frac{\text{sgn}(b_{q,1})\Re[\mathbf{w}^H \bar{\mathbf{x}}_q^{(+)}]}{\sigma_n \sqrt{\mathbf{w}^H \mathbf{w}}} \tag{3.8}$$

The MBER beamforming solution is the one that minimizes the bit error rate:

$$\mathbf{w}_{\text{MBER}} = \arg\min P_E(\mathbf{w}) \tag{3.9}$$

3.4.1 Block-Data Based Gradient Adaptive MBER Algorithm

Several methods of finding the MBER solutions were proposed. One is a block-data based gradient adaptive algorithm. For this approach, the p.d.f is approximated using a Parzen window given a training block of *K* samples. From this estimate of the p.d.f., the BER can be estimated as:

$$\tilde{P}_E(\mathbf{w}) = \frac{1}{K} \sum_{k=1}^{K} Q(\tilde{g}_k(\mathbf{w})) \tag{3.10}$$

with

$$g_k(\mathbf{w}) = \frac{\text{sgn}(b_1(k))y_R(k)}{\rho_n \sqrt{\mathbf{w}^H \mathbf{w}}} \tag{3.11}$$

Simulations show that this MBER algorithm converges rapidly to the theoretical MBER solution for reasonably small blocks of data [10]. The downside to the above approach is that it requires a lot of resources in order to be implemented. In particular, the memory requirements become prohibitive due to the need for storing blocks of data in order to estimate the output's p.d.f.

3.4.2 Stochastic Gradient Based MBER Algorithms

An alternative approach to estimating the MBER optimum weights is to use a sample-by-sample adaptation algorithm. Two versions of this approach were presented in [10], the least bit error rate (LBER) and approximate LBER (ALBER) techniques. ALBER requires more approximations than LBER, but has a significantly lower computational complexity, which is comparable to the LMS algorithm.

For the ALBER algorithm, the probability of BER is simplified to

$$\tilde{P}_E(\mathbf{w}) = \frac{1}{K}\sum_{k=1}^{K} Q(\tilde{g}_k(\mathbf{w})) \tag{3.12}$$

with

$$g_k(\mathbf{w}) = \frac{\text{sgn}(b_1(k))y_R(k)}{\rho_n} \tag{3.13}$$

where K is the # of samples in the training block. This simplification leads to a weight update equation similar to that of the LMS algorithm and given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{\text{sgn}(b_1(k))}{2\sqrt{2\pi}\,\rho_n} \exp\left(-\frac{y_R^2(k)}{2\rho_n^2}\right)\mathbf{x}(k) \tag{3.14}$$

where the adaptive gain ? and the Parzen window kernel width $?_n$ are the two parameters that have to be adjusted appropriately.

Simulations show that LBER and ALBER algorithms have almost indistinguishable learning curves. The ALBER algorithm exhibits virtually the same convergence rate as the LBER algorithm although it involves more computations and has a lower complexity. Both algorithms are able to closely approach the optimum MBER solution.

3.4.3 Comparison of the Performance of the MMSE and MBER Optimum Solutions

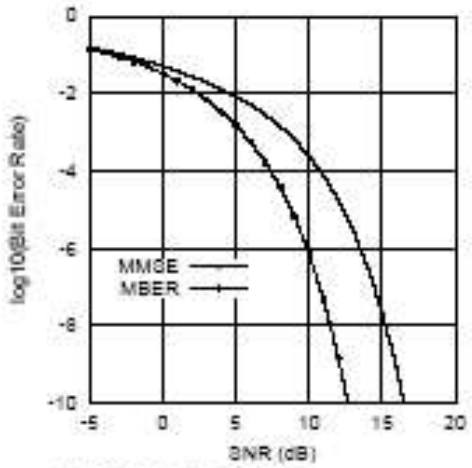
Figure 3.6 shows a comparison of the BER performance of the optimum MMSE and MBER beamformers. For this simulation, 5 signal sources were used, with 4 of them being interferers, and a two-element array was used. In (a), all the sources have equal power, in (b), one of the interferers has 6 dB higher power than the desired source, and in (c), all the interferers have 6 dB higher power than the desired source.

For all cases, the superior performance of the MBER algorithm over traditional MMSE algorithms can be seen. In particular, for cases (b) and (c), the MMSE approach is unable to achieve low BERs even at high SNRs while the MBER approach still exhibits lower BER as SNR increases.

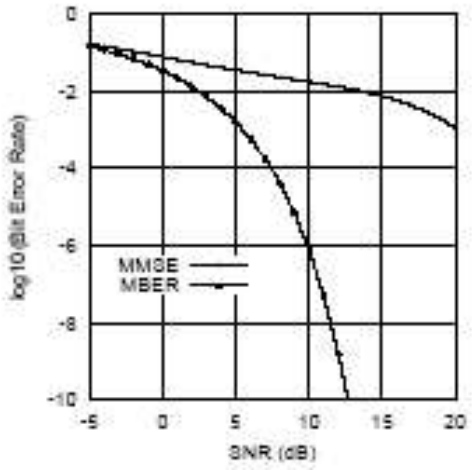
3.4.4 Normalized MBER Algorithms

The above MBER algorithms have so far assumed non-fading channels. In a realistic wireless environment, the presence of multipath components will invariably introduce fading effects.

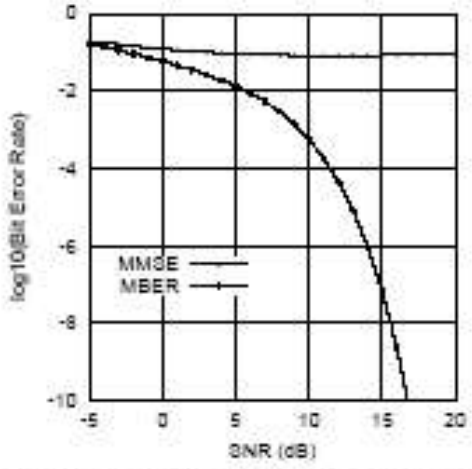
Garcia [13] investigated the performance of the above MBER algorithms in frequency-flat, directional fading channels with co-channel interference and developed MBER algorithms more suited to this type of signal environment. Simulations in his study have shown that in a fading channel, MBER beamformers may fail to achieve lower BERs compared to Wiener-solution-based beamformers.



(a) SNR=INR$_i$ for $i = 2, 3, 4, 5$.



(b) SNR=INR$_i$ for $i = 3, 4, 5$, and INR$_2$ =SNR+ 6 dB.



(c) INR$_i$ =SNR+ 6 dB for $i = 2, 3, 4, 5$.

Figure 3.6. Comparison of BER performance [10]

To improve the MBER performance in fading channels, normalization was performed on the MBER solution by removing the variation of the cost function with respect to the input signal power. It was proposed in [13] to modify the probability of bit error and change it to:

$$P_{E,Norm}(\mathbf{w}) = \frac{1}{N_b} \sum_{q=1}^{N_b/2} Q\left(\frac{\text{sgn}(b_{q,1}) \bar{y}_{R,q}^{(+)}}{\sigma_n \left\| \bar{\mathbf{x}}_q \right\| \sqrt{\mathbf{w}^H \mathbf{w}}} \right) \tag{3.15}$$

The $\|\mathbf{x}_r\|$ term was inserted in the denominator to minimize the variation of the bit error probability with respect to the input signal's power.

Simulations have shown that, in a fading channel, given the same mean SNR, the Normalized MBER (NMBER) algorithms are capable of achieving lower BERs compared to their respective un-normalized MBER counterparts as well as traditional MMSE techniques.

4. PROPOSED METHODOLOGY

4.1 Implementation

A simplified block diagram of the set-up used in this research is shown in Figure 4.1. The FPGA board to be used is Altera's Stratix EP1S25 DSP Development Board. The DSP board to be used is TI's TMS320C6701. The output will be displayed using a host computer.

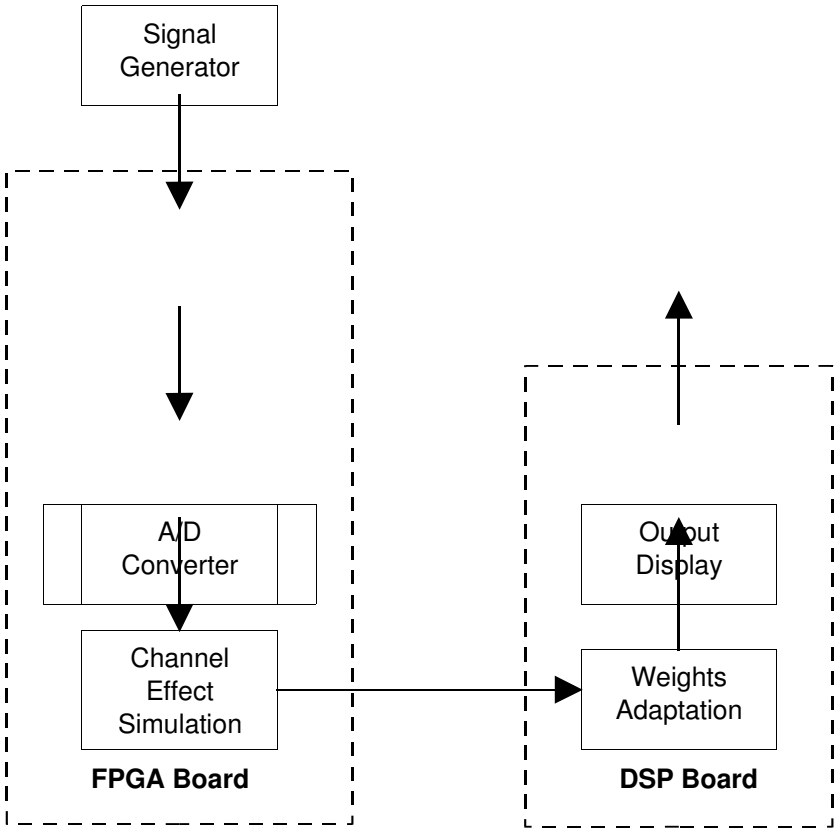


Figure 4.1. Block diagram

A vector signal generator will be used to generate BPSK modulated signals. This signal sequence will be used as the desired signal sequence. The power of this generated signal may be varied to produce different values of signal-to-noise ratios (SNR).

For this research, the MBER algorithm will be implemented for a two-element antenna array. However, to control the signal environment, the BPSK signals generated by the vector signal generator will already be in the intermediate frequency of 100 MHz and will be fed directly to one of the FPGA A/D converters. This ensures that the signals will still be uncorrupted by the channel when the A/D converter samples them at the rate of 80 mega-samples per second, using 12 bits per sample. Simulation of the second antenna element and effects of the channel will be added later on.

After reception, digital downconversion will be performed on the signal. The digital downconverter (DDC) will be implemented entirely in the FPGA board. Instead of using just a single, large decimating filter, the DDC will be broken down into several, smaller cascaded filters. This allows the filters to be relatively easier to design and reduces the resources consumed on the FPGA. After downconversion, the received signals will have been translated to their complex baseband representation. Further manipulations of the signal will now make use of complex arithmetic.

Effects of the channel will be added after the downconversion. The first channel effect to be added is interference. To simulate interference sources, independent sequences of baseband signals will be generated in the FPGA. The power of these signals may be controlled to produce different values signal-to-interference ratios (SIR).

Each interference source's direction of arrival (DOA) is set beforehand, as well as the desired signal's DOA. The 2-element array is modeled, without loss of generality, by using the desired and interference signals that arrive at element 1 as the reference signals. The desired signal component arriving at element 1 is the signal generated by the vector signal generator, while the interference signals components are those generated in the FPGA. To simulate the signals arriving at the second antenna element, appropriate phase

shifts which depend on the signals' DOA are applied to each signal sequence. To reduce usage of resources in the FPGA, the complex multipliers to be used for each signal sequence are computed beforehand and are implemented as constant values in the FPGA. An antenna element spacing of half a wavelength is used for the computations, and a 1.9GHz carrier is assumed for both the desired and interference signals.

The next step will be corruption of the signal by Additive White Gaussian Noise. This noise has an in-phase and quadrature component, which are added independently to the in-phase and quadrature components of the baseband signals at each antenna element. The signals at the antenna elements are already the sum of the desired signal and the interference signals. To simulate the effect of noise, a look-up table of Gaussian-distributed noise samples will be used.

The effects of multipath will be simulated in the time domain using a frequency-flat fading model. Delayed versions of the arriving desired and interference signals will be generated, each with a corresponding attenuation and DOA. These time-delayed sequences will be added to the signals at each antenna element, with the appropriate phase shifts taken into consideration. A minimum of one multipath component will be implemented for each signal source; the actual number of multipath components will depend on the resources available in the FPGA board.

After all the channel effects have been added and the array output has been computed, these data will be passed onto the TI DSP board. The interface between the TI TMS320C6701 board and the Altera EP1S25 board is the External Memory Interface (EMIF) port on the TI side and the TI-EVM connector on the Altera side. Figures 4.3 and 4.4 show the block diagrams of the TI (from EVM user's guide) and Altera (from

EP1S25 datasheet) boards, respectively, with the appropriate interface ports highlighted in red.

Computation of the weights will be done in the TI DSP board. Two algorithms for computation of the adaptive beamformer weights will be implemented separately. The first will be a Minimum-Mean-Square-Error solution which will be implemented using the Normalized Least Mean Square (NLMS) algorithm. The second will be a Minimum-Bit-Error-Rate solution which will be implemented using the ALBER algorithm introduced in section 3.4. The performance of the two algorithms shall be compared

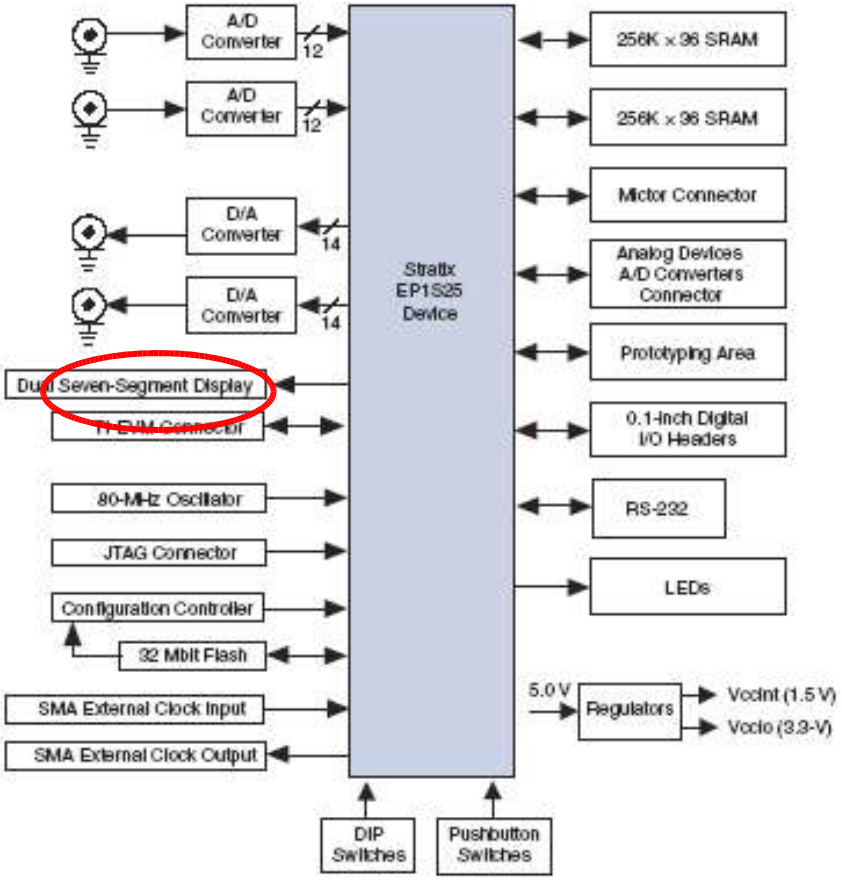


Figure 4.3. EP1S25 block diagram

using the achieved BER as the basis of comparison. The convergence speed of the algorithm will not be given much importance in both cases.

The actual BER will be computed in the TI board. In order to access the computed BERs, the TI board will be interface to a host computer using JTAG Emulation. The relevant interface port can also be seen in Figure 4.4 highlighted in blue.

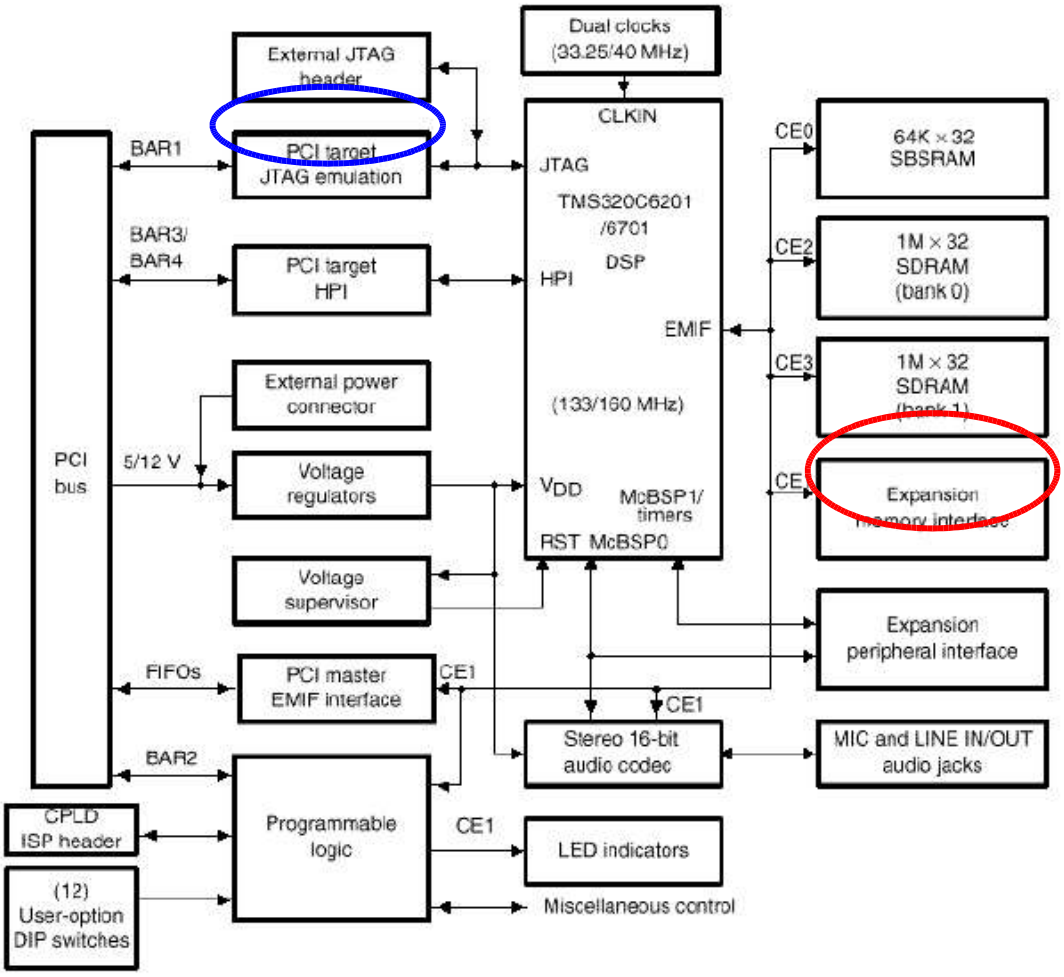


Figure 4.4 TMS320C6701 block diagram

4.2 Testing

As already mentioned, the performance of the MMSE and MBER solution based on the BER will be compared in this research using the hardware implementation detailed above. In addition, the resource usage of the two algorithms in terms of memory and processing power requirements will also be investigated.

For each algorithm, four sets of test cases will be implemented. For the first set of test cases, the desired signal will only be corrupted by AWGN. No interfering sources and multipath effects are present. The SNR will be varied by changing the input signal's power while maintaining the noise power at a constant level. Plots of BER versus SNR will be generated for both the algorithms and displayed using the host computer.

The second set of test cases will add the effect of co-channel interference. Between one to three interference sources will be added, each corrupted by AWGN. The performance of the algorithms will be tested for:

a) SIR < SNR

b) SIR = SNR

c) SIR > SNR

The performance of the algorithms when the separation between the desired signal source and interference source (in degrees) is varied will also be evaluated. Again, plots of BER will be generated and displayed on the host computer.

The third set of test cases investigates the effect of multipath on the performance of the algorithms. Interference sources are not present for these test cases.

Finally, the fourth set of test cases combines the AWGN, co-channel interference and multipath effects.

For all sets of test cases, the data and program memory requirements in the FPGA and DSP board will be compared for the two algorithms. For the FPGA, the number of logic elements used will be the basis of comparison, while for the DSP board, the amount of memory space needed by the program and look-up tables will be used as the basis of comparison.

Besides the memory requirements, the processing power needed by both algorithms will also be compared. The processing power needed will be quantified by the number of cycles used by non-NOP instructions in each iteration of both algorithms for the same output data rate. In addition, given the same set of parameters for a particular signal environment, the maximum achievable data rate will be determined.

The resource usage of the channel effect simulations will not be included in the comparisons since this should be identical for both algorithms.

4.3 Gantt Chart

Novembe				December				January				February				March				April				May			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
(1)																											
				(2)																							
												(3)															
																(4)											
																				(5)							
																				(6)							

(1) Design of Digital Downconverter

(2) Coding of Channel Effects

(3) Coding of MMSE and MBER algorithms

(4) Integration

(5) Testing

(6) Documentation

REFERENCES

[1] T. S. Rappaport, *Wireless Communications,* Prentice Hall, Upper Saddle River NJ, USA, 2002.

[2] X. Wang and H. Poor, *Introduction to Wireless Communication Systems: Advanced Techniques for Signal Reception,* Prentice Hall, 2003.

[3] L. Devlin et al, "Multipath Measurements for High Data-Rate Wireless Communications," *Wireless Symposium/Portable By Design Conference Spring 1999,* Feb. 1999.

[4] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems, Part 1: Characterization," *IEEE Communications Magazine,* vol. 35, no. 9, Sept. 1997, pp. 136-46.

[5] T. Haynes, "A Primer on Digital Beamforming," *Spectrum Signal Processing,* March 1998.

[6] D. Schobben, "Efficient Adaptive Multi-channel Concepts in Acoustics: Blind Signal Separation and Echo Cancellation," Eindhoven University of Technology, Sept. 1999.

[7] B. Widrow and S. Stearns, *Adaptive Signal Processing,* Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.

[8] J. Hudson, *Adaptive Array Principles,* Peter Peregrinus Ltd., London, UK, 1981.

[9] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice-Hall Inc., Englewood Cliffs, NJ, 1996

[10] S. Chen et al, "Adaptive Minimum Bit Error Rate Beamforming," *IEEE Transactions on Wireless Communications,* April 2002.

[11] Chen, S. et al, “Stochastic Gradient Adaptive Minimum Symbol-Error-Rate Equalization for Multi-Level Pulse-Amplitude Modulation,” *Proceedings of ICASSP*, pp. 2629-32, April 2002.

[12] S. Chen, “Adaptive Minimum Bit Error Rate Filtering,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 151, no. 9, pp. 76-85, 2004.

[13] I. Garcia, “Performance of Adaptive Minim Bit-Error-Rate Beamforming Algorithms for Diversity Combining and Interference Suppression Under Frequency-Flat Directional Fading Channels,” University of the Philippines, March 2004.