International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 05 | May-2018 www.irjet.net p-ISSN: 2395-0072

A Fast convergent frequency-domain MIMO equalizer for few-mode fiber communication systems

Kovvuri Rakesh Reddy[1], Prof. S.Revathi[2]

[1] *Student, Dept. of Electronics and communication Engineering, VIT University, INDIA,*
[2] *Prof, Dept. of Electronics and communication Engineering, VIT University, INDIA.*

Abstract - *Polarization-division multiplexing (PDM) has emerged as a next-generation technology to sustain the continuous traffic growth, in order to keep up with the future of Internet bandwidth requirement and one of the fundamental challenges in FMF transmission systems is the random inter-modal crosstalk between any two polarization modes. Another significant challenge is the large accumulated PMD, which can induce significant inter-symbol interference (ISI) on each polarized mode signal in PDM systems. The large accumulated PMD and an increasing number of multiplexed channels need very complex DSP hardware for MIMO processing and an urgent efficient solution is needed to mitigate the impact of booming internet penetration. We compare different mainstream blind and adaptive algorithms in order to find the algorithm that have better error convergence performance and efficient computational complexity.*

Key Words: ***Equalizers; Adaptive; Polarization Division Multiplexing, Chromatic Dispersion, Polarization Mode Dispersion***

1. INTRODUCTION

The data rates of optical communication networks have been widely increased but at data rates of more than 10 Gb/s, the performance of long-haul high-capacity optical fiber communication systems is significantly decreased by transmission impairments such as residual chromatic dispersion (CD), polarization-mode dispersion (PMD), laser phase noise and Kerr fiber nonlinearities. Generally, these linear impairments are compensated for in the optical domain, CD is compensated using dispersion compensating fiber or fiber Bragg gratings and PMD is avoided through fiber selection or compensated with an optical PMD compensator.

Over the past four decades with the introduction and development of coherent detection, advanced modulation formats, and digital signal processing techniques these advancements promoted the growth of optical communication towards high-capacity and long-distance transmissions. With the entire capture of the amplitude and phase of the signals using coherent optical detection, the compensation and mitigation of the transmission impairments can be implemented using the digital signal processing in electrical domain this technique is generally called equalization which deals with inter-symbol interference in communication systems.

There is no principal difference between a fiber optic channel and e.g. a radio channel in terms of ISI; the received baseband signal is distorted in a similar manner in both systems, i.e. symbols spread out over neighboring symbols as they propagate through the channel. Consequently, equalizer techniques used for radio and other systems should in essence be viable for fiber optic links as well. However, one important difference is that while a radio channel can usually be considered as linear, a fiber optic channel exhibits nonlinear characteristics which degrades the signal over transmission.

So digital signal processing in optical communication enabled next-generation optical communication networks to achieve a performance close to the Shannon capacity limit to which we are closer than ever before. Long before we know we are touching the Shannon limit in order to solve this problem Polarization-division multiplexing (SDM) has emerged as a next-generation technology to sustain the continuous traffic growth, in order to keep up with the future of Internet bandwidth requirement. Among PDM technologies, PDM using few-mode fiber (FMF) transmission has been extensively explored [1]. and with the help of advanced DSP components it offered huge gains in data capacities that can be carried over optical networks [2].

In the digital coherent transmission systems, the output from the photodiodes are sampled and transformed into the discrete signals using high-speed analogue-to-digital convertors (ADCs), which can be further processed by the DSP algorithms.

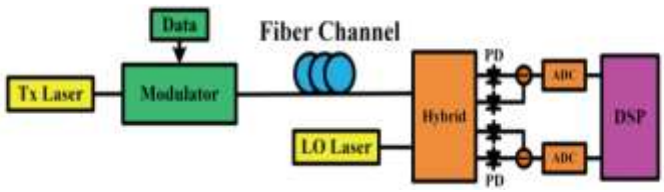


Figure (1.1): Schematic of coherent optical communication system with digital signal processing.

In the early era of optical communication networks various methods have been developed to increase the communications system performance by reducing the effects of the ISI. In this project we compare and analyze the performance of different algorithms blind and adaptive that aid the DSP in 100 Gbps DP-QPSK optical communication system in reducing the ISI and the Polarization Mode Dispersion from the received de-multiplexed data that reaches from the DP-QPSK coherent receiver after passing through long spans of fiber. The 100 Gbps DP-QPSK system can be divided into five main parts: DP-QPSK Transmitter, Transmission Link, Coherent Receiver, Digital Signal Processing, and Detection & Decoding (which is followed by direct-error-counting). Digital Signal Processing module consists of a DP-QPSK DSP MIMO equalizer module compensates for the linear impairments of the fiber through the following processes of Electrical signal amplification and filtering, Analog to Digital conversion (down sampling), Dispersion Compensation, Polarization De-multiplexing, Carrier phase estimation, Constellation Diagram Generation [15].

1.1 Analog to Digital conversion:

The analog to digital conversion here is basically a down sampling process and we have chosen a 2-bit sampling however sampling rate can be changed. Number of samples per symbol is defined by multiplying the length of the input signal divided by the number of symbols.

1.2 Dispersion compensation:

Dispersion compensation essentially cancels the chromatic dispersion of some optical element(s). However, the term is often used in a more general sense of dispersion management, meaning the control the overall chromatic dispersion of some system. The goal can be, e.g., to avoid excessive temporal broadening of ultra-short pulses and/or the distortion of signals. Dispersion compensation is applied mainly in mode-locked lasers and in telecommunication systems, but also sometimes in optical fibers transporting light e.g. to or from some fiber-optic sensor [3].

1.3 Polarization De-Multiplexing:

The dual-polarization (DP) transmission scheme has been introduced into practical optical communication systems for the first time by using recently-developed digital coherent receivers. Controlling the state of polarization (SOP) of the DP signal in the digital domain, such receivers can de-multiplex two polarization tributaries in an adaptive manner. The efficient SOP control based on digital signal processing (DSP) depends on the phase information of the DP signal, which is obtained from coherent detection employing phase and polarization diversities. First we will use the Constant Modulus Algorithm (CMA) for the blind estimation of the filter weights and later use the LMS and RLS algorithms and finally compare the performance of the latter two [4].

1.3.1 Polarization de-multiplexing using the BLIND CMA MIMO algorithm:

The Jones matrix of the fiber for transmission can be written as:

$$T = \begin{pmatrix} \sqrt{\alpha}e^{i\delta} & -\sqrt{1-\alpha} \\ \sqrt{1-\alpha} & \sqrt{\alpha}e^{-i\delta} \end{pmatrix} \qquad 1.1$$

Where α and δ denote the power splitting ratio and the phase difference between the two polarization modes. The SOP of the output signal can be written as [11]:

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = T \begin{pmatrix} E_{in,x} \\ E_{in,y} \end{pmatrix} \qquad 1.2$$

So if we can find the inverse of matrix T, we can do polarization de-multiplexing. The constant modulus algorithm (CMA) is used first. Following figure shows the DSP circuit for channel expression and the corresponding equation:

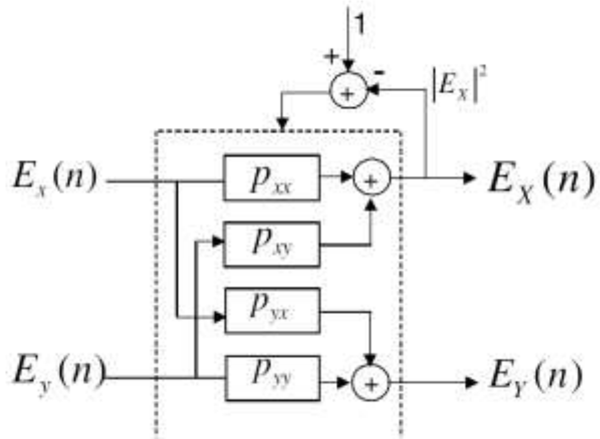


Figure (1.2): DSP circuit for channel

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix} \cdot \begin{pmatrix} E_x \\ E_y \end{pmatrix} \qquad 1.3$$

The matrix elements are updated symbol by symbol according to:

$p_{xx}(n+1) = p_{xx}(n) + \mu(1- ||E_{x(n)}||^2) E_x(n).E_x^*(n)$ 1.4

$p_{xy}(n+1) = p_{xy}(n) + \mu(1- ||E_{x(n)}||^2) E_x(n).E_y^*(n)$ 1.5

µ is the step-size parameter and n the number of symbol. The p matrix is basically an adaptive FIR filter and we use CMA for blind estimation. The initial values for $p_{xx}(0)$ and $p_{yy}(0)$ are:

$p_{xx}(0) = [00…010..00];$ 1.6

$p_{yy}(0) = [00…010..00];$ 1.7

$p_{xy}(0)=p_{yx}(0)= [00…000..00];$ 1.8

In our simulation we have chosen a 3-tap FIR filter, however the order can be changed.

1.3.2 Polarization De-multiplexing using the ADAPTIVE FD-LMS MIMO algorithm:

Adaptive MIMO algorithms can be implemented in both time domain and frequency domain. However, compared with time domain method, frequency domain MIMO equalizer could dynamically compensate the large accumulated DMGD and ISI with much lower hardware complexity by taking advantage of fast Fourier transform (FFT) and block processing. In Few Mode Fiber systems, except for hardware complexity, convergence speed is another major consideration in the design of adaptive MIMO equalizers. Adaptive MIMO equalizer usually uses data-aided adaptive frequency domain least mean square (FD-LMS) algorithm for the initial convergence[2].

The simplicity of the LMS algorithm and ease of implementation means that it is the best choice for many real-time systems. According to this method, the M tap weights of the filter are padded with an equal number of zeros, and an N-point FFT is used for the computation, where

$$N=2M \qquad 1.10$$

Thus let the N-by-l vector W(k) denote the FFT coefficients of the zero-padded, tap-weight vector w(k), as follows

$$\bar{W}(k) = FFT[\bar{w}(k), 0] \qquad 1.11$$

Where 0 is the M-by-1 null vector and FFT [ ] denotes fast Fourier transformation and the frequency-domain weight vector $\bar{W}(k)$ is twice as long as the time-domain weight vector $\bar{w}(k)$ Correspondingly, let U(t) denote an N-by-N diagonal matrix derived from the input data as follows

$$U(k) = diag\{ FFT[u(kM - M),\ldots \ldots, u(kM - 1), u(kM),\ldots \ldots u(kM + M - 1)]\} \qquad 1.12$$

For the kth block, define the M-by-1 desired response vector

$$d(k) = [d(kM), d(kM + 1), \ldots d\{kM + M - 1 \}]^T \qquad 1.13$$

and the corresponding M-by-1 error signal vector

$$e(k) = [e(kM), e\{kM + 1), \ldots, e(kM + M - l)]^T \qquad 1.14$$

We may transform the error signal vector e(k) into the frequency domain as follows:

$$E(k) = FFT[0,e(k)] \qquad 1.15$$

Next, recognizing that a linear correlation is basically a "reversed" form of linear convolution, we find that applying the overlap-save method to the linear correlation yields

$$\Phi(k) = \text{first } M \text{ elements of } IFFT[U^H(k).E(k)] \qquad 1.16$$

Finally we get the tap weight vector of the filter by

$$\bar{W}(k+1) = \bar{W}(k) + \mu.FFT[\Phi(k), 0] \qquad 1.17$$

Equations (2.11) to (2.17), in that order, define the Adaptive LMS algorithm. And the complexity of the algorithm with an increase in filter taps is given by (2N+1), where N is the number of filter taps.

1.3.3 Polarization de-multiplexing using the Adaptive RLS MIMO algorithm:

The standard RLS algorithm performs the following operations to update the coefficients of an adaptive filter [13].

1. Calculates the output signal y (n) of the adaptive filter.
2. Calculates the error signal $e(n)$ by using the following equation:

$$e(n) = d(n) - y(n). \qquad 1.18$$

3. Updates the filter coefficients by using the following equation:

$$\bar{w}(n+1) = \bar{w}(n) + e(n).k(n) \qquad 1.19$$

Where w(n) is the filter coefficients vector and $\bar{k}(n)$ is the gain vector. $\bar{k}(n)$ is defined by the following equation:

$$k(n) = [(p(n).\bar{u}(n)) / (\lambda + \bar{u}^T(n).p(n).\bar{u}(n))]. \qquad 1.20$$

Where λ is the forgetting factor and $P(n)$ is the inverse correlation matrix of the input signal. $P(n)$ has the following initial value $P(0)$:

$$p(0) = \delta^{-1}. [I]. \qquad 1.21$$

Where I is an identity vector and δ is the regularization factor. The standard RLS algorithm uses the following equation to update this inverse correlation matrix.

$$p(n+1) = \lambda^{-1}.p(n) - \lambda^{-1}\bar{k}(n).\bar{u}^T(n).p(n). \qquad 1.22$$

RLS algorithms calculate $J(n)$ by using the following equation

$$J(n) = \frac{1}{N}\sum_{i=0}^{N-1}\lambda^i e^2(n-i) \qquad 1.23$$

Where N is the filter length and λ is the forgetting factor.

This algorithm calculates not only the instantaneous value $e^2(n)$ but also the past values, such as $e^2(n-1)$, $e^2(n-2)$... $e^2(n-N+1)$. The value range of the forgetting factor is (0, 1]. When the forgetting factor is less than 1, this factor specifies that this algorithm places a larger weight on the current value and a smaller weight on the past values. The resulting $E[e^2(n)]$ of the RLS algorithms is more accurate than that of the LMS algorithms.

The LMS algorithms require fewer computational resources and memory than the RLS algorithms. However, the eigenvalue spread of the input correlation matrix, or the correlation matrix of the input signal, might affect the convergence speed of the resulting adaptive filter. The convergence speed of the RLS algorithms is much faster than that of the LMS algorithms. However, the RLS algorithms require more computational resources than the LMS algorithms.

1.4 Carrier phase estimation:

Phase locking in the hardware domain can be replaced by phase estimation in digital domain by DSP [5]. The received QPSK signal can be presented by

$E(t) = A \exp\{ j [\theta_s(t) + \theta_{s(t)}] \}$ 1.26

We have used the following algorithm to estimate the phase of the QPSK signal in digital domain:

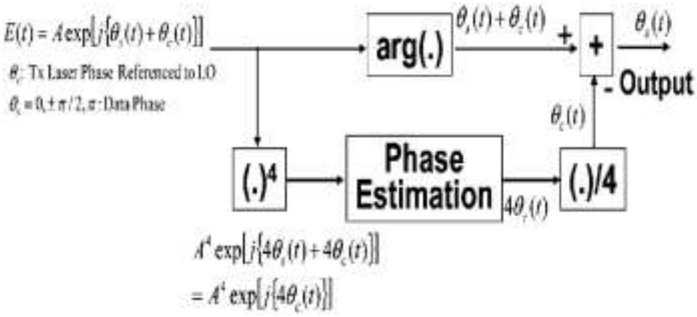


Figure (1.3): Visual explanation of Carrier Phase Estimation

1.5 Further Analysis into Adaptive LMS variants

1.5.1 Least Mean Square (LMS) algorithm:

It has become the most frequently used algorithm for the training of adaptive finite impulse response (FIR) filters.

The LMS minimises the cost function J(k) =0.5. e^2 (k), and is given by

$e(k) = d(k) – x^T (k).w(k)$ 1.27

$w(k + 1) = w(k) + \mu.e(k).x(k)$ 1.28

Where e(k) denotes the instantaneous error at the output of the filter, d(k) is the desired signal,x(k) = [x(k − 1),...,x(k − N)] T is the input signal vector, N is the length of the filter, (·) T is the vector transpose operator, and w(k) = [w 1 (k),...,w N (k)] T is the filter coefficient(weight) vector. The parameter μ is the step size (learning rate), which is critical to the performance, and defines how fast the algorithm is converging. The main drawback of the "pure" LMS algorithm is that it is sensitive to the scaling of its input. This makes it very hard to choose a learning rate μ that guarantees stability of the algorithm. Below here we discuss various variants of LMS that offer better performance.

1.5.2 Variable Step Size LMS algorithms:

Normalised LMS (NLMS) is a variant of the LMS which basically normalises the step size based on the power of the signal in the filter memory in order to improve stability and convergence. The NLMS update equation is given by

$w(n+1) = w(n) + [\beta / (\epsilon + ||x(n)||^2)] .e(n).x(n).$ 1.29

$w(n + 1) = w(n) + \mu. e_p (n).x(n)$ 1.30

$e_p(n) = d(n) – x^T (n).w(n + 1)$ 1.31

Here ϵ is a small “regularization” constant, added to avoid division by 0 for small values of input or signal power is too low

1.5.3 Gradient Adaptive Step-Size (GASS) Algorithms

The standard LMS uses a fixed step size (μ) and hence there is a trade-off between speed of convergence and steady state error variance as previously discussed. Ideally we wish to have a large step when the error is large and then reduce the step size as the estimate approaches the true values, thus allowing for both fast convergence and small steady state error. Variable step size algorithms aim to simultaneously minimize the cost function with respect to the weights as well as minimising the cost function with respect to the step size. The update equation for the step size is given by equation μ(n+1) in which $\psi(n)$ can take various forms [6]. Step Size update equation:

$\mu(n+1) = \mu(n) – [\rho . \nabla\mu .J(n)]$

$= \mu(n) – \rho.e(n).x(n). \psi(n)$

$= \mu(k) + \rho.e(k).x^T(k). \psi(n)$ 1.32

Where (**ρ**) is the step size learning rate)

Weight update equation of conventional LMS:

$w(k + 1) = w(k) + \mu.e(k).x(k)$ 1.33

A gradient adaptive learning rate μ(n) can be introduced into the LMS algorithm

a.) Benveniste’s is rigorous and evaluates the sensitivity

$\psi (n) = [I − (\mu(n − 1) \times x(n − 1)x^T (n − 1))] \psi(n − 1) + e(n − 1).x(n − 1)$ 1.34

b.) Ang & Farhang algorithm is based on a recursive calculation of ψ from Benveniste’s Equation in the form

$\psi(n) = \alpha.\psi(n − 1) + e(n − 1).x(n − 1), \alpha \in (0,1).$ 1.35

c.) Mathews’ algorithm is a simplification of the algorithm by Farhang and Ang, where α = 0, that is, it uses noisy instantaneous estimates of the gradient, resulting in the learning rate update

$\psi(n) = e(n − 1)x(n − 1)$ 1.36

The term in the square brackets for the Benveniste algorithm represents a time varying low pass filter which smoothes the instantaneous gradient $e(n − 1)\mathbf{x}(n − 1)$ and hence increases robustness to statistical variations in the input. Ang & Farhang is a simplification in which the low pass filter term is fixed and Matthews & Xie is the same as Ang & Farhang for the case when $\alpha = 0$, i.e. the filter time constant is zero and so there is no smoothing. This is the least robust meaning a small step (ρ) will be necessary but its advantage is in its low computational complexity [9].

1.5.4 Introducing robustness into NLMS: Adaptive Regularization of NLMS Algorithm

The purpose of ϵ in NLMS update equation is to prevent the step size from being too large (and hence causing instability) when the signal power in the filter memory ($\|x(n)\|^2$) is small. It also prevents divide by zero errors when $\|x(n)\|^2 = 0$. By adapting ϵ with time, we can make the gain of the NLMS adaptive and hence more optimal. We choose an update equation for ϵ such that the cost function is minimized.

$$\epsilon(n+1) = \epsilon(n) - \rho.\nabla\epsilon J(n). \quad 1.37$$

Where (ρ) is the step for the adaption of regularisation factor. The weight update formula is given by

$$w(n+1) = w(n) + [\, 1 / (\epsilon(n) + \|x(n)\|^2) \,]\, e(n).x(n) \quad 1.38$$

The Adaptive Regularization factor ϵ update equation

$$\epsilon(n+1) = \epsilon(n) - \rho.\mu.[\,(e(n)e(n-1)x^T(n)x(n-1)) / (\epsilon(n-1) + x^T(n-\ldots\ldots\ldots 1)x(n-1))\,] \quad 1.39$$

1.5.5 Comparing the Complexity of Benveniste and the Adaptive Regularisation Algorithm.

Digital Signal Processing (DSP) algorithms are mostly implemented on special purpose Digital Signal Processors. Recently, advanced hybrid microcontrollers and Field Programmable Gate Arrays (FPGAs) are also evolving as the suitable platforms for realizing DSP algorithms. Here in table below we compare the algorithms for the number of multipliers they need for the implementation of algorithm.

	Addition	Multiplication	Division
Benveniste Weight update	q	2q	0
Benveniste μ update	4q	$2q^2 + 3q + 2$	0
AR weight update	2q	2q + 1	1
AR ϵ update	2q	2q + 4	1

Table 1. Comparison of complexity between Benveniste and AR algorithm.

2. Simulation and results

The signal generated by an optical DP-QPSK Transmitter is then propagated through the fiber loop where dispersion and polarization effects occur. It then passes through the Coherent Receiver and into the MATLAB DSP component for post processing where several scripts are used to reduce the effects of ISI and PMD induced by the fiber non-linear impairments. The fiber dispersion is compensated using a simple transversal digital filter, and the adaptive polarization de-multiplexing is realized by applying the constant-modulus algorithm (CMA), Adaptive Least Mean Squares (LMS) algorithm and Adaptive Recursive Least Square (RLS) algorithm. A modified Viterbi-and-Viterbi phase estimation algorithm is then used to compensate for phase and frequency mismatch between the transmitter and local oscillator (LO). After the digital signal processing is complete, the signal is sent to the detector and decoder, and then to the BER Test Set for direct-error-counting. Even though RLS equalizer converges quickly than the LMS equalizer it requires much more multipliers for an increase in number of filter taps. We further simulated different types of Gradient Adaptive Step Size LMS variants to achieve a faster convergence rate at the same time maintaining the complexity of the system. Finally we compared GASS algorithms with custom NLMS variant which has an adaptive regularization factor that adapts to the input signal which provided better convergence performance than the GASS algorithms while maintaining similar complexity of pure LMS.

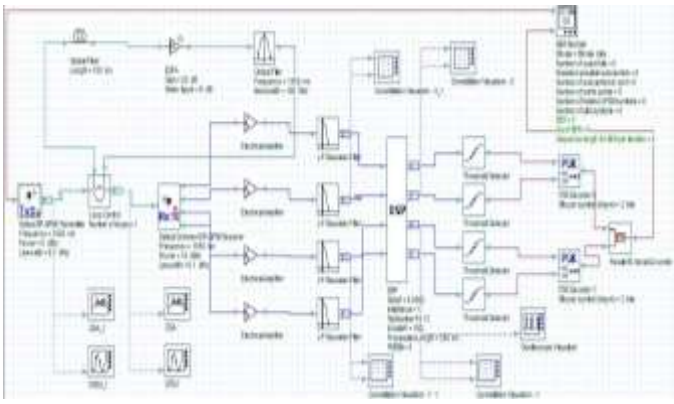


Figure 2. Schematic of 100 Gbps DP-QPSK optical communication system with DSP in optisystem.

2.1 DSP Equalizer using Blind CMA algorithm:

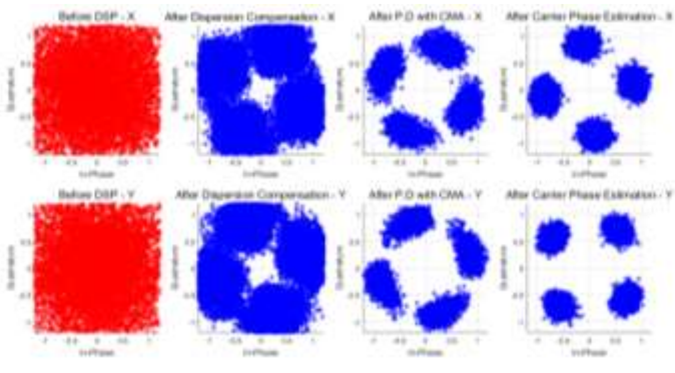


Figure (2.1): DSP Equalizer using blind CMA algorithm

After receiving the data from the fiber, optical DP-QPSK receiver De-multiplexes the ISI induced noisy multiplexed data and sends the individual data streams to the MATLAB-OPTISYSTEM DSP component where it runs the user defined scripts to reduce the noise error. First the data individual data streams are down sampled, then sent to Dispersion Compensation module to cancel the Chromatic Dispersion error induced by the fiber non linearity. Afterwards it's sent into the Polarization De-multiplexing module to compensate the error induced by phase delay that occurs over the fiber length using the Constant Modulus Algorithm (CMA) which is used for the estimation of filter weights. Finally the filtered data is sent into Carrier phase Estimation module to perform Phase lock between the X and the Y polarization to further decrease phase differences between a received signal's wave and the receiver's local oscillator. And the constellation diagrams after each module operation are generated and we can observe that the error levels reached the optimum value.

2.2 DSP Equalizer using Adaptive FD-LMS algorithm:

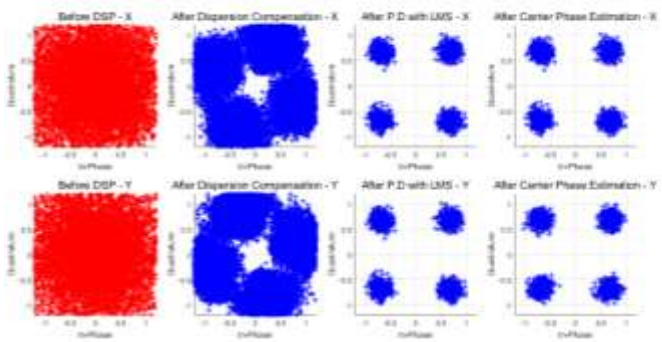


Figure (2.2): Constellation Diagrams of processed data DSP MIMO Equalizer using Adaptive FD-LMS algorithm.

In this simulation every parameter is same as the above with the exception of Polarization Multiplexing done by the Adaptive Frequency Domain LMS algorithm which estimates the filter weights by adaptive converging them toward desired optimum values and we can observe that the error in the data stream drops to optimum values right after the Polarization De-multiplexing as carrier recovery and phase matching between the transmitter and local oscillator (LO) is already done by adaptive LMS algorithm in Polarizing De-multiplexing module. And we can see observe that constellation diagram of Adaptive LMS aided DSP is more concentrated than the CMA aided DSP.

2.3 DSP Equalizer using Adaptive RLS algorithm:

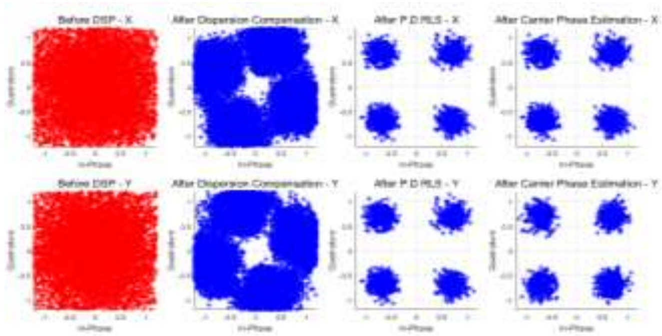


Figure (2.3): DSP Equalizer using Adaptive RLS algorithm.

In this simulation Adaptive LMS algorithm will be used in the Polarization De-multiplexing module which estimates the filter weights by adaptive converging them toward desired optimum values and we can observe that the error in the data stream drops to optimum values right after the Polarization De-multiplexing as carrier recovery is already done by adaptive RLS algorithm in Polarizing De- multiplexing module. And we can see observe that constellation diagram of Adaptive RLS aided DSP is similarly concentrated as the LMS aided DSP and more than concentrated than the Blind CMA version.

2.4 Comparing the Adaptive Algorithms Error convergence Performance:

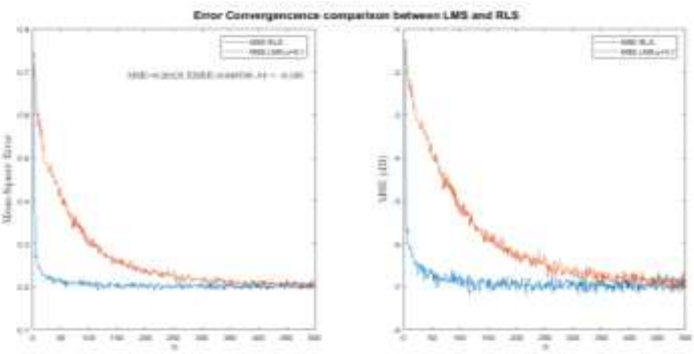


Figure (2.4): Error convergence comparison of LMS and RLS algorithm.

In the above graphs we can observe that the RLS aided DSP can achieve faster convergence rate toward the optimum value after taking few samples compared to the LMS aided DSP which requires more samples to converge. In the Optical communication it is required that the system compensate the error at faster rate in order to avoid data loss. So here RLS is a clear winner in this competition.

2.5 Analyzing the LMS In-Phase, Quadrature and constellation diagrams of input and output data:

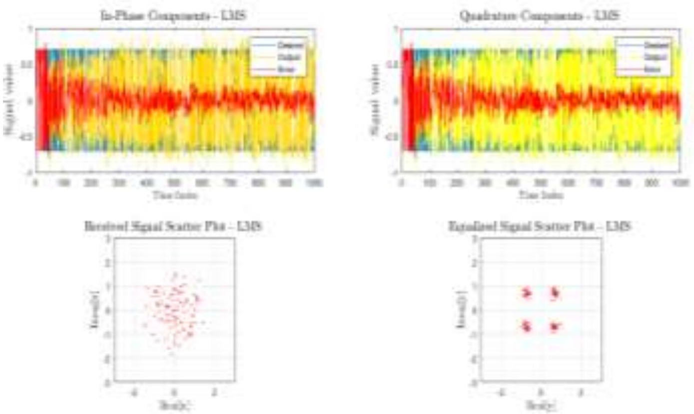


Figure (2.5): Visualization of In-Phase, Quadrature and constellation diagram components of LMS DSP system.

In this graph we graphically compare the Adaptive LMS In-Phase and Quadrature components of the error, output and the desired signals in a single subplot and can infer that the error gradually decreases after taking in the few hundred samples so by that time the filter weights will be adaptively adjusted by the LMS algorithm to the desired values.

2.6 Analyzing the RLS In-Phase, Quadrature and constellation diagrams of input and output data:

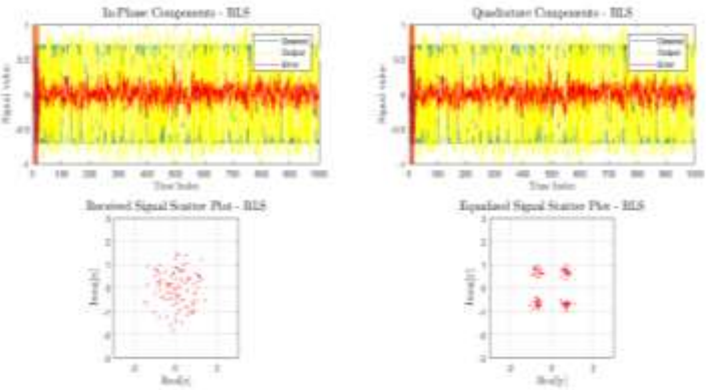


Figure (2.6): Visualization of In-Phase, Quadrature and constellation diagram components of LMS DSP system.

In this graph we graphically compare the adaptive RLS In-Phase and Quadrature components of the error, output and the desired signals in a single subplot and can infer that the error gradually decreases after taking in the few tens of samples so by that time the filter weights will be adaptively adjusted by the RLS algorithm to the desired values. And once again we can infer with confidence that the adaptive RLS algorithm convergence rate to that of the LMS algorithm is faster and superior.

2.7 Comparison of complexity between LMS and RLS:

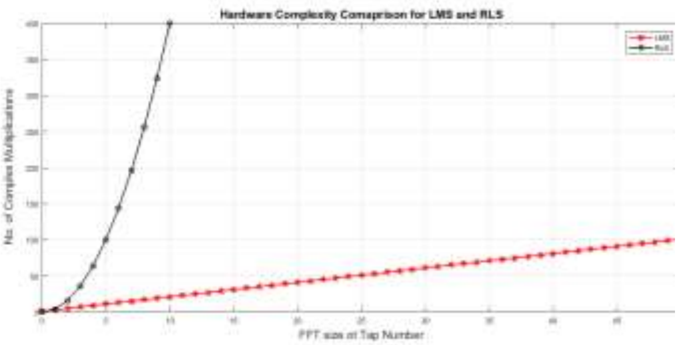


Figure (2.7): Hardware complexity comparison of LMS and RLS.

Even though the convergence rate of the Adaptive RLS algorithm is higher than the LMS algorithm the number of complex multiplications required for the number increasing filter taps exponentially. Whereas the adaptive LMS algorithm requires far less multiplication cycles for the increasing Filter weights. So even if the RLS provides better convergence rate it's practicality is endangered with complexity it produces to compensate the error. So here LMS is better equipped to compensate the huge amount of error with comparatively less complexity and slower convergence rate. So further simulation are done on the LMS variants which offer better convergence rate with similar complexity structure.

2.8 Error performance comparison of LMS and NLMS algorithms:

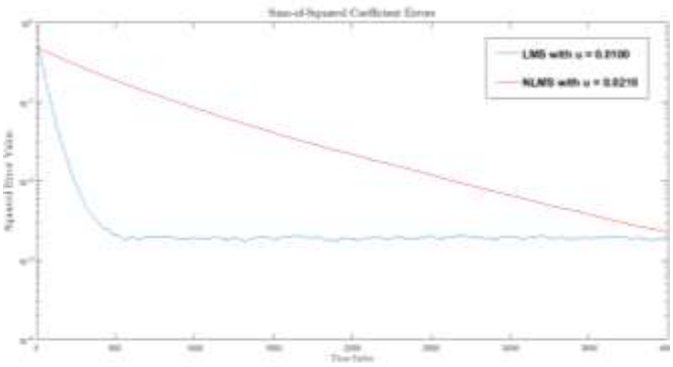


Figure (2.8): Error performance comparison of LMS and NLMS.

The above figures compares the squared error performance of LMS an NLMS algorithm and we know that higher step sizes (μ) can lead to faster convergence rate.

Even with step size (μ) = 0.210 NLMS algorithm convergence performance is worse than that of the pure LMS with step size (μ) = 0.0100. Due to the normalization of step size in NLMS variant its effective step size varies inversely over time to the signal power so for low power inputs the effective (μ) of NLMS will be much smaller and need more samples before it can converge to the optimum value although it remains stable, but its convergence rate needs improvement.

2.9 Error performance comparison of GASS and LMS algorithms:

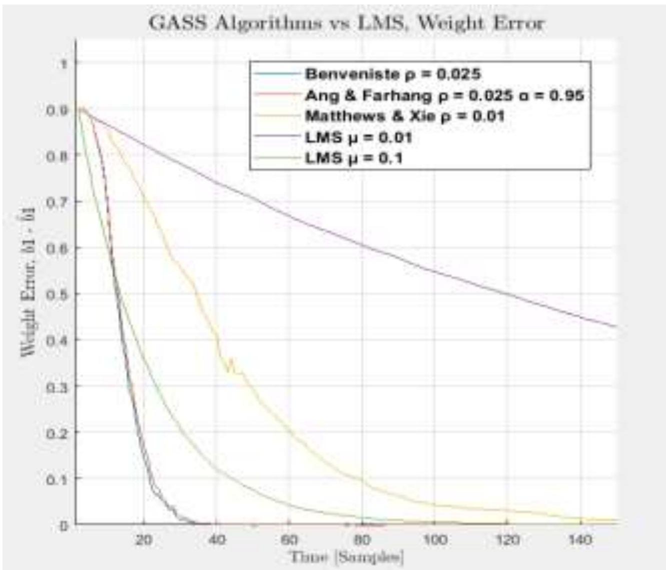


Figure (2.9 a): Weight Error convergence performance comparison of GASS and LMS.

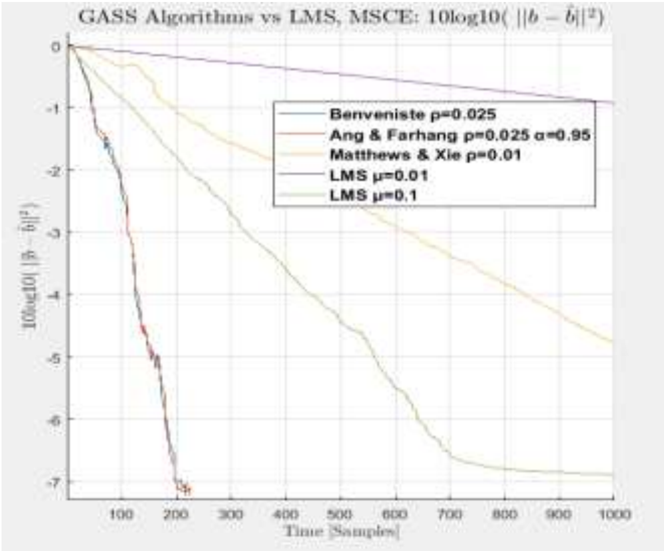
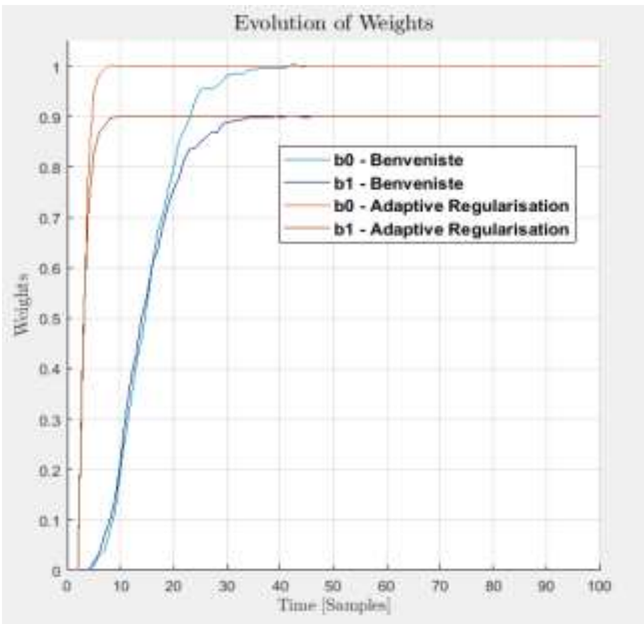


Figure (2.9 b): Weight Error convergence performance comparison of GASS and LMS.

The above figures compares the three GASS algorithms with their optimal parameters and the standard LMS for two different values of μ and we can observe the filter weight convergence of each variant. We already know that an adaptive LMS algorithm with higher step size (μ) converges faster than the one with lower step size. From the graphs we can infer that the LMS with (μ) = 0.1 has converged faster than the LMS variant with (μ)=0.01. And it shows that Benveniste and Ang & Farhang algorithms to significantly outperform the standard LMS in terms of both convergence speed and steady state error, Benveniste algorithm which is most efficient of the current batch is further compared with modified variant of NLMS.

2.10 Performance comparison of Benveniste and Adaptive Regularization variant of NLMS algorithm:



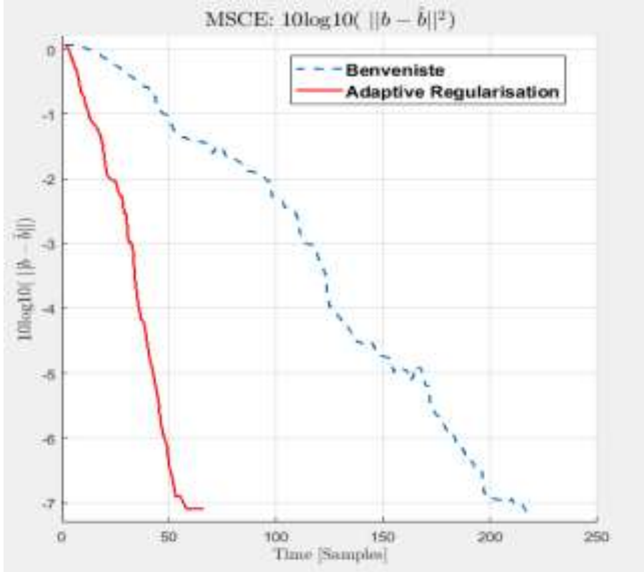


Figure (2.10): Performance comparison of Benveniste and Adaptive Regularization variant of NLMS algorithm over evolution of weights and Mean Square Convergence Error.

From above graphs we can observe that the evolution filter weights (b0, b1) adapted by Adaptive Regularisation variant of NLMS converges to their desired value at faster rate than the Benveniste algorithm and in the comparison of Mean Square Convergence Error we can clearly infer that Adaptive Regularization variant converges rapidly to optimum value with an intake of just around 60 samples whereas Benveniste algorithm needed over 200 samples to converge to same optimum value. Finally we can infer that Adaptive Regularization of NLMS offers better performance than its counterpart Benveniste with fewer intake of data samples.

2.11 Complexity comparison of Benveniste and Adaptive Regularization variant of NLMS algorithm:

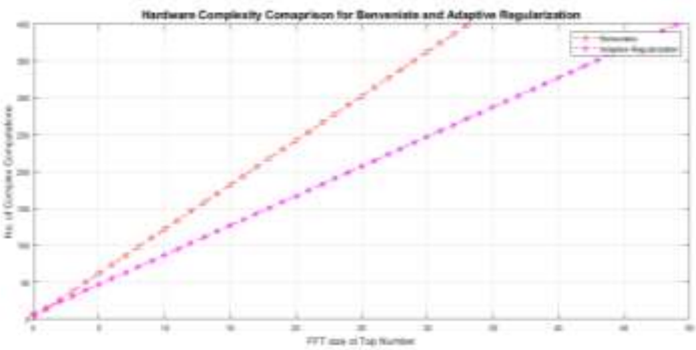


Figure (2.11): Complexity comparison of Benveniste and Adaptive Regularization variant of NLMS algorithm over evolution of weights and Mean Square Convergence Error.

From above graphs we can observe that the number of complex arithmetic computations required for the increasing number of filter taps is more for the Benveniste algorithm and comparatively lesser for the Adaptive Regularization algorithm. So here Adaptive Regularization algorithm is better equipped to compensate the huge amounts of error with comparatively less complexity and faster convergence rate than the Benveniste algorithm.

3 .Conclusion

As we look back on our objective at the starting of the report we accomplished and generated the respective results.

Our first objective of comparing the performance of Blind CMA, Adaptive LMS and Adaptive RLS MIMO equalizers by using the distorted de-multiplexed has been accomplished by processing the data individually with above algorithms and we observed the error convergence performance of RLS equalizer is comparatively better than the LMS but in the complexity factor RLS required more multipliers for a the increasing amount filter taps. So the LMS was efficient choice.

Then the Adaptive LMS algorithm is further analysed compared by simulating Gradient Adaptive Step Size (GASS) algorithms and observed that optimal Benveniste and Ang & Farhang algorithms performed better than the pure LMS variant.

Finally we simulated the Mean Square Error Convergence performance, evolution of weights and the hardware complexity comparison of both the Benveniste and Adaptive Regularization algorithm and observed that the evolution filter weights (b0, b1) adapted by Adaptive Regularisation variant of NLMS converged to its desired value at faster rate than the Benveniste algorithm. In the comparison of Mean Square Convergence Error we can clearly infer that Adaptive Regularization NLMS variant converged rapidly to optimum value with an intake of just around 60 samples whereas Benveniste algorithm needed over 200 samples to converge to same optimum value. And in hardware complexity comparison number of complex arithmetic computations required for the increasing number of filter taps is more for the Benveniste algorithm and comparatively lesser for the Adaptive Regularization algorithm. Finally we can infer that Adaptive Regularization of NLMS offers better error convergence performance and efficient computational complexity than its counterpart Benveniste with fewer intake of data samples.

4. References

[1] XuanHe & YiWeng (2018), Fast convergent frequency-domain MIMO equalizer for few-mode fiber communication systems, Optics Communications, Volume 409, 15 February 2018, Pages 131-136.

[2] X. He, X. Zhou, J. Wang, Y. Weng, and Z. Pan (2013), "A Fast Convergence Frequency Domain Least Mean Square Algorithm for Compensation of Differential Mode Group Delay in Few Mode Fibers," in Optical Fiber Communication Conference/National Fiber Optic Engineers Conference, OSA Technical Digest (Optical Society of America, 2013), paper OM2C.4

[3] Junyi Wang, Digital Signal Processing in Coherent Optical Communications (2013), 99 pages; 3590082.

[4] Roland Ryf, Sebastian Randel, Alan H. Gnauck, Cristian Bolle, Alberto Sierra, Sami Mumtaz, Mina Esmaeelpour, Ellsworth C. Burrows, René-Jean Essiambre, Peter J. Winzer, David W. Peckham, Alan H. McCurdy, and Robert Lingle (2012), "Mode-Division Multiplexing Over 96 km of Few-Mode Fiber Using Coherent 6 × 6 MIMO Processing," J. Lightwave Technol. 30(4), 521-531.

[5] Danilo P. Mandic, Vanessa Su Lee Goh (2010), Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models.

[6] Ling Liu; Zhenning Tao; Weizhen Yan; Oda, S.; Hoshida, T.; Rasmussen, J.C. (2009); , "Initial tap setup of constant modulus algorithm for polarization de-multiplexing in optical coherent receivers," Optical Fiber Communication - incudes post deadline papers, 2009. OFC 2009. Conference on, vol., no., pp.1-3.

[7] Guifang Li (2009), "Recent advances in coherent optical communication," Adv. Opt. Photon. 1, 279-307.

[8] Seb J. Savory (2008), Digital filters for coherent optical receivers, Vol. 16, No. 2, OPTICS EXPRESS.

[9] Kikuchi, K (2008); "Polarization de-multiplexing algorithm in the digital coherent receiver," IEEE/LEOS Summer Topical Meetings, Digest of the, vol., no., pp.101-102.

[10] 12 Gb/s Co-DP-QPSK Optical Transmission Offline Digital Filtering Analysis. In: 14th Simpósio Brasileiro de Microondas e Optoeletrônica.

[8]. Neng Bai, Cen Xia, and Guifang Li (2012), "Adaptive frequency-domain equalization for the transmission of the fundamental mode in a few-mode fiber," Opt. Express **20**(21), 24010-24017

[9]. Md. Saifuddin Faruk and Kazuro Kikuchi (2011). "Adaptive frequency-domain equalization in digital coherent optical receivers," Opt. Express **19**(13), 12789-12798.

[11]. Kevin Banovi´c, Mohammed A. S. Khalid and Esam Abdel-Raheem (2007), "A Configurable Fractionally-Spaced Blind Adaptive Equalizer for QAM Demodulators," Digital Signal Processing, Vol. 17, No.6, pp. 1071- 1088, Nov.

[12]. J. Jensen, R. C. Hendriks, and J. S. Erkelens (2006), "MMSE Estimation of Discrete Fourier Coefficients with Generalized Gamma Priors," Tech. Rep., Delft Univ. Techn.,.

[13]. Antoinette Beasley and Arlene Cole-Rhodes (2005), "Performance of Adaptive Equalizer for QAM signals," IEEE, Military Communications Conference. MILCOM 2005, Vol.4, pp. 2373 – 2377

[14]. N. Bai, E. Ip, M. Li, T. Wang, and G. Li (2013), "Experimental demonstration of adaptive frequency-domain

equalization for mode-division multiplexed transmission," in Optical Fiber Communication Conference/National Fiber Optic Engineers Conference, OSA Technical Digest (Optical Society of America), paper OM2C.5.

[15]. Oumer, Berhan. (2015). Channel Equalization Linear, Nonlinear, Blind and Time Domain and Interferance Cancellation. 10.13140/RG.2.1.3546.256.