STUDY AND DEVELOPMENT OF BLIND EQUALIZATION FOR DEMODULATION OF DYNAMIC IONOSPHERE CUBESAT EXPERIMENT SATELLITE DATA

by

Ravi Kant

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Jacob Gunther Major Professor	Dr. Todd Moon Committee Member

Dr. Charles Swenson
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2013

Abstract

Study and Development of Blind Equalization for Demodulation of Dynamic Ionosphere Cubesat Experiment Satellite Data

by

Ravi Kant, Master of Science

Utah State University, 2013

Major Professor: Dr. Jacob Gunther
Department: Electrical and Computer Engineering

The project presented in this report is for combating the narrowband interference present in the downlink data of Dynamic Ionosphere Cubesat Experiment (DICE) satellites. The interference prevents the Phase Locked Loop (PLL) and timing synchronization loops from performing efficiently and the data is not demodulated properly. The samples are corrupted and correct decisions about the signal constellation cannot be made. Here, blind equalization techniques are used to tackle this problem. In this thesis, Constant Modulus Algorithm (CMA) and Multi-Modulus Algorithm (MMA) algorithms are extended to offset-Quadrature Phase-Shift Keying (QPSK) modulation and then implemented. When applied to simulated and real data from DICE satellites, CMA and MMA perform similarly if no phase offset is present. However, the CMA algorithm requires an external PLL in presence of a phase offset. MMA, however, performs similar or better than CMA without any need for an external PLL.

(67 pages)

Public Abstract

Study and Development of Blind Equalization for Demodulation of Dynamic Ionosphere Cubesat Experiment Satellite Data

by

Ravi Kant, Master of Science

Utah State University, 2013

Major Professor: Dr. Jacob Gunther
Department: Electrical and Computer Engineering

Space Dynamics Lab (SDL) has an ongoing mission called Dynamic Ionosphere Cubesat Experiment (DICE) satellite program wherein two spacecrafts are put in orbit for scientific measurements. These relay their measurements to ground stations in Wallops Island, USA. Because of very low signal strength or Signal-to-Noise Ratio (SNR), the receiving antennas must have large gain to receive these signals. But there are many sources of signals nearby these antennas which have very high SNR as compared to that of DICE satellite spacecrafts. Thus, the processing being done to receive the signal and extract data is getting hampered by these interfering signals. The main problem is in estimation of when to correctly sample or measure the incoming signal to provide optimum readings to the receiving processing chain for correct demodulation.

In this report blind equalizers are implemented, which help overcome the problems mentioned above. The equalizers extract the inherent characteristics of the signal being sent and then adapt to combat the errors introduced by interference. Two different approaches are studied: the Constant Modulus Algorithm (CMA) and the Multi-Modulus Algorithm (MMA). It is found that MMA is a more robust method and works better or at least on par with CMA approach.

To my parents and brother

Acknowledgments

I would like to thank and express gratitude towards my advisor, Dr. Jacob Gunther, who made completion of this project possible. He has always been an immense source of guidance, help, and inspiration for me. He revealed the beauty of digital signal processing and communications to me. I also would like to acknowledge Dr. Todd Moon whose teaching of various concepts of mathematical application in signal processing has been most helpful. He has taught me many important lessons of life which surpass the academic realm. I would also thank Dr. Charles Swenson who has always been very helpful with his suggestions, opinions, and questions.

I would also thank my parents, Dr. S.D. Singh and Mrs. Rita Singh, and my brother, Rishi Sumit, for all their love, affection, teachings, and values. I will always be grateful to them for all they have done for me. I would also thank Varun Joshi, Shantanu, Varun Thakur, Omkar, Tejas, Satyajit, Neeraj, Swadesh, Nagarvind, Yogi, and all my friends and seniors, both in Logan and India, who have always helped me and been there for me no matter what.

Ravi Kant

Contents

List of Tables

Table Page

List of Figures

Figure Page

Chapter 1

Introduction

1.1 Problem Statement

The Space Dynamics Laboratory has launched the Dynamic Ionosphere Cubesat Experiment (DICE) satellites with the mission to map geomagnetic Storm Enhanced Density (SED) plasma bulge and plume formations in Earth's ionosphere [1]. There are two aircraft, Farkle and Yahtzee, which send the data to the ground station at Wallops Island in Virginia. The data modulation is offset-Quadrature Phase-Shift Keying (QPSK) constellation with two samples per symbol.

At the ground station the receiving antenna receives the desired signal from the satellites. But there is narrowband interference due to use of radios by civil and security personnel in the vicinity. This interference is of higher power level than that of desired signal from satellite. The radiation pattern of the receiving antenna is not able to reject these high power narrowband interferences. In Figures 1.1 and 1.2 the spikes are the narrowband interferences which are about 10-15 dbs above the desired signal. Due to this, the Phase Locked Loop (PLL) and especially the timing loop is not able to work well and thus the demodulation process is not being completed successfully. To combat this problem equalizers are needed. In the above mentioned scenario, we cannot use data-aided equalizers because the satellites do not transmit any training data sequence. Thus, blind equalizers are a good option to be used here [2–4]. In blind equalizers, Constant Modulus Algorithm (CMA) [5–11] and Multi-Modulus Algorithm (MMA) [12–15] type equalizers are appropriate for a QPSK type constellation. This thesis reports on research that investigates blind adaptive equalization algorithms for offset-QPSK. The DICE spacecraft transmits offset-QPSK. While blind adaptive equalization algorithms exist for QPSK and Quadrature Amplitude Modulation (QAM) techniques, very little has been published on extending these

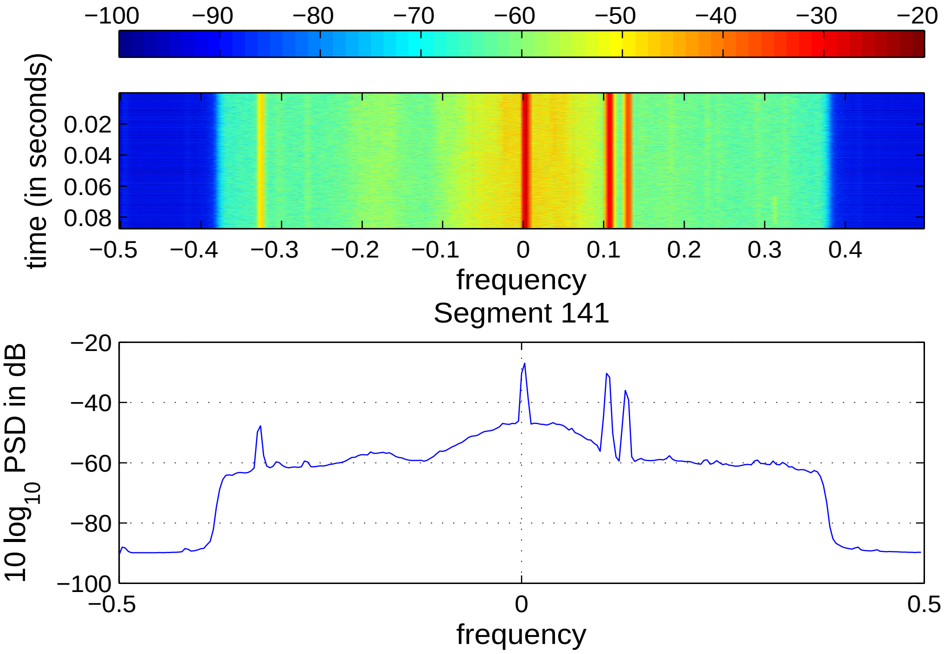


Fig. 1.1: Narrowband interference for data segment 141.

algorithms to the offset case. The rest of the thesis is organized as follows. The rest of this chapter reviews QPSK and offset-QPSK modulation as a background. The next chapter contains discussion about the blind adaptive equalization techniques. Also here, CMA and MMA algorithm filter update equations are extended for offset-QPSK. The last chapter discusses the implementation of the algorithms on simulated and real satellite data. The results and plots of these are also presented in the last chapter.

1.2 Quadrature Phase Shift Keying

QPSK is a modulation scheme in which the symbols have the same energy, i.e. at equal distance from origin in a coordinate system and spaced equally on the unit circle. The position of these points is equally spaced with phase angles as $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, and $\frac{7\pi}{4}$. The main information is in the phase change as the amplitudes of all the points are equal. A block diagram of QPSK modulator is shown in Figure 1.3 and QPSK signal constellation is shown in Figure 1.4. Also, two sinusoids are used which are 90° out of phase with each

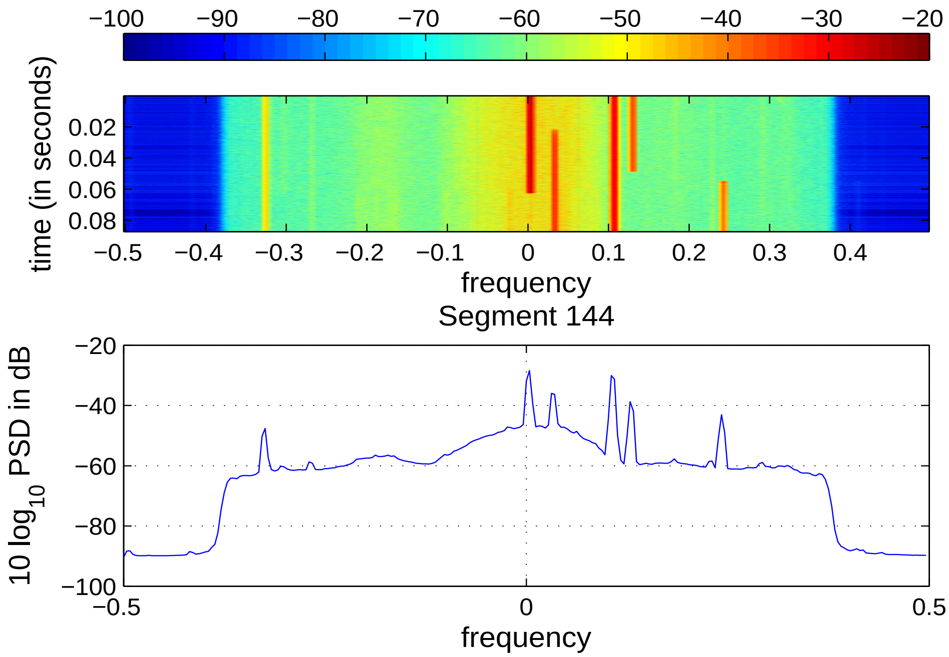


Fig. 1.2: Narrowband interference for data segment 144.

other [16, 17]. The data bits are generated at interval of T_b. These data bits are mapped to symbol coordinates using the constellation table. Using the serial to parallel converter, two bits are used to generate one symbol $(a_0(k), a_1(k))$ as shown in Table 1.1.

In Table 1.1, $A = 1$ for further analysis. If $A \neq 1$ then it can always be applied as a constant to further equations. The symbols, or indirectly bits, are split into into In-Phase(I) branch and Quadrature(Q) branch. These are named so because of the carrier sinusoids are

Table 1.1: Bits-to-symbol assignment.

bit0(k)	bit1(k)	$a_0(k)$	$a_1(k)$
0	0	$\sqrt{2}A$	$\sqrt{2}A$
0	1	$-\sqrt{2}A$	$\sqrt{2}A$
1	1	$-\sqrt{2}A$	$-\sqrt{2}A$
1	0	$\sqrt{2}A$	$-\sqrt{2}A$

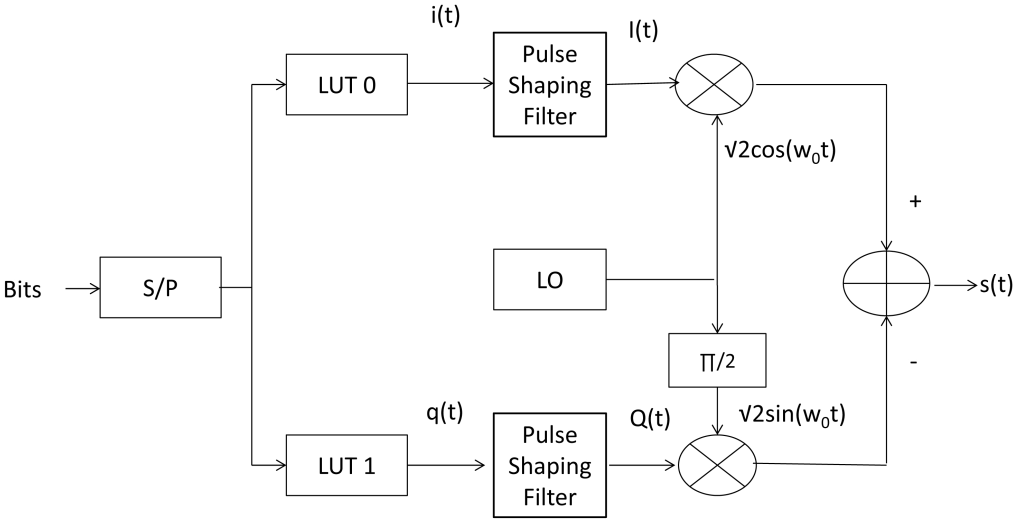


Fig. 1.3: Block diagram QPSK modulator.

90° out of phase. The signal on the In-Phase branch is

$$i(t) = \sum_k a_0(k)\delta(t - kT_s), \tag{1.1}$$

and on the quadrature branch is

$$q(t) = \sum_k a_1(k)\delta(t - kT_s). \tag{1.2}$$

Considering a continuous time system, these symbols are then passed through a pulse shaping filter with impulse response $h(t) = p(t)$. The pulse shaping filter is present to avoid the Inter-Symbol Interference (ISI) and fulfills Nyquist Criterion for zero ISI. The resultant signal on individual branches can be represented as

$$It(t) = \sum_k a_0(k)p(t - kT_s), \tag{1.3}$$

$$Qt(t) = \sum_k a_1(k)p(t - kT_s). \tag{1.4}$$

The signal is then modulated with a cosine and sin function being generated from a Local

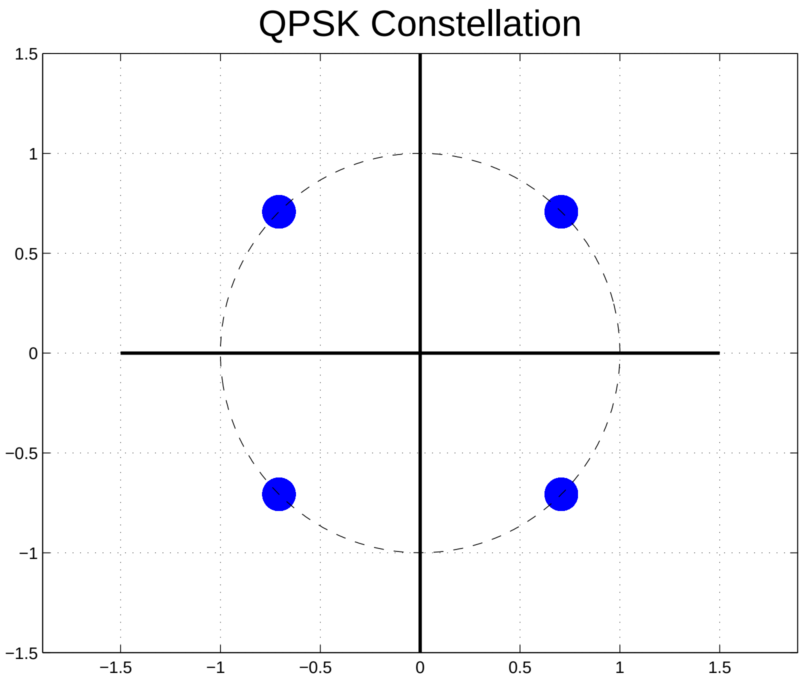


Fig. 1.4: QPSK signal constellation.

Oscillator (LO) at the same frequency. The cosine carrier sinusoid is used by I-branch symbols and the sine by Q-branch. These are added together to make the signal to be transmitted.

$$s(t) = It(t)\sqrt{2}\cos(w_0 t) - Qt(t)\sqrt{2}\sin(w_0 t) + w(t) \tag{1.5}$$

A QPSK receiver is shown in Figure 1.5. The received signal $r(t)$ can be expressed as sum of $Ir(t)$, $Qr(t)$ and additive white Gaussian noise (AWGN) component $w(t)$.

$$r(t) = Ir(t)\sqrt{2}\cos(w_0 t) - Qr(t)\sqrt{2}\sin(w_0 t) + w(t) \tag{1.6}$$

Here $Ir(t)$ and $Qr(t)$ are the In-phase and Quadrature signal streams.

At the receiver carrier sinusoids are again multiplied, before the signal is passed on, the signals take the following form:

$$r(t)\sqrt{2}\cos(w_0 t) = I_r(t) + I_r(t)\cos(2w_0 t) - Qr(t)\sin(2w_0 t) + w(t), \tag{1.7}$$

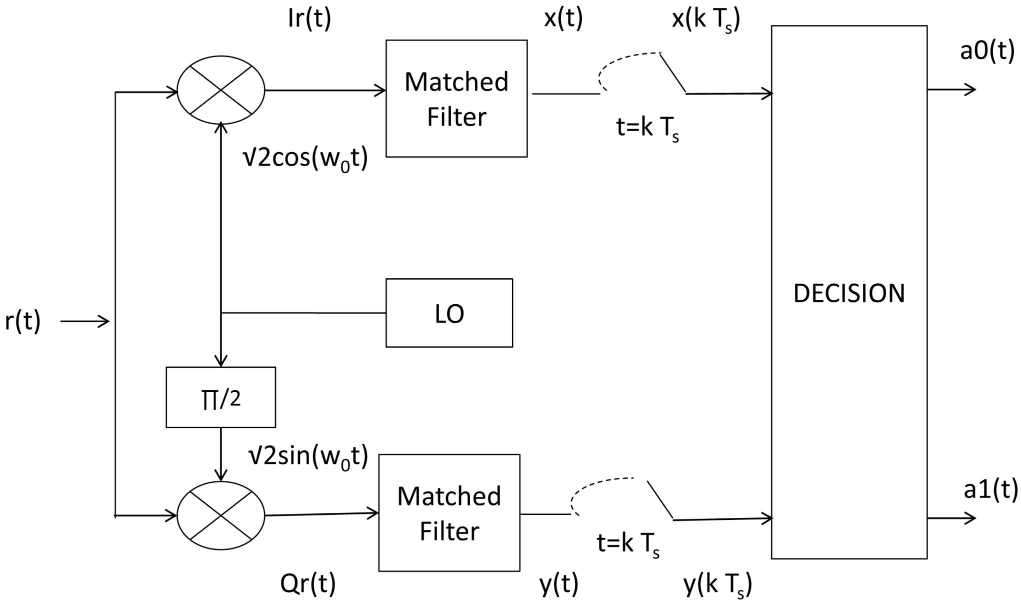


Fig. 1.5: Block diagram of continuous time QPSK receiver.

$$r(t)\sqrt{2}\sin(w_0t) = -Q_r(t) + I_r(t)\cos(2w_0t) + Qr(t)\sin(2w_0t) + w(t). \tag{1.8}$$

Inputs shown in (1.7) and (1.8) are passed to the matched filter. Matched filters have low-pass filtering property, and thus they reject the double frequency terms only $Ir(t)$ and $Qr(t)$ are passed. Also, the additive noise is also filtered through. The matched filter output for I branch is called $x(t)$ and that of Q branch is called $y(t)$.

$$x(t) = \int_{T_1+t}^{T_2+t} Ir(\lambda - t)h(\lambda)\, d\lambda + v_0(t) \tag{1.9}$$

$$y(t) = \int_{T_1+t}^{T_2+t} Qr(\lambda - t)h(\lambda)\, d\lambda + v_1(t) \tag{1.10}$$

Considering no phase, frequency, and timing offset,

$$x(t) = \sum_m a_0(m)r_p(mT_s - t) + v_0(t), \tag{1.11}$$

$$y(t) = \sum_{m} a_1(m) r_p(mT_s - t) + v_1(t). \tag{1.12}$$

Here, $v_0(t)$ and $v_1(t)$ are the noise terms which have been filtered through the matched filter. And $r_p(mT_s)$ is defined as

$$r_p(mT_s) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0. \end{cases}$$

Then the matched filter outputs are sampled to at $t = kT_s$, which produces the outputs $x(kT_s)$ and $x(kT_s)$.

$$x(kT_s) = \sum_{m} a_0(m) r_p((m-k)T_s) + v_0(kT_s) \tag{1.13}$$

$$y(kT_s) = \sum_{m} a_1(m) r_p((m-k)T_s) + v_1(kT_s) \tag{1.14}$$

Now if the pulse shape is in accordance with Nyquist Non-ISI condition, then there is no contribution to the k$^{\text{th}}$ sample from any other samples. But there is a noise component which is present. Therefore the expression becomes

$$x(kT_s) = a_0(k) + v_0(kT_s), \tag{1.15}$$

$$y(kT_s) = a_1(k) + v_1(kT_s). \tag{1.16}$$

This is then passed onto the decision making block where the symbols are mapped back to constellation points. These can then be converted back to the bits using a look-up table.

1.3 Discrete Time Realization

In the section above, the QPSK model was shown in continuous time. For Discrete Time Model, the block diagram shown in Figure 1.6 is used. Here the procedure to get symbols is same as that of continuous time. Once bits have been mapped to the constellation symbols then it is upsampled by a factor of N and thus we have symbol at I branch (and similarly at Q branch) as

$$a_0\left(\frac{m}{N}\right) = \begin{cases} a_0(\frac{m}{N}) & m = integer \times N \\ 0 & \text{otherwise.} \end{cases} \tag{1.17}$$

The upsampled symbols are then given as input to pulse shaping filter to give the I and Q streams

$$It(n) = \sum_k a_0(\frac{m}{N})p(n - mN), \tag{1.18}$$

$$Qt(n) = \sum_k a_1(\frac{m}{N})p(n - mN). \tag{1.19}$$

If it is assumed that the pulse shaping filter fulfills the required band pass and Nyquist requirements, then the output of this filter is used to modulate the carrier sinusoids. These two streams are then added together to form the signal $s(nT)$.

$$s(nT) = It(nT)\sqrt{2}\cos(\Omega_0 n) - Qt(nT)\sqrt{2}\sin(\Omega_0 n) \tag{1.20}$$

A discrete time QPSK receiver or demodulator is shown in Figure 1.7. The demodulator, on reception of the signal, first removes the sinusoids similar to that of continuous time implementation. The received signal is

$$r(nT) = Ir(nT)\sqrt{2}\cos(\Omega_0 t) - Qr(nT)\sqrt{2}\sin(\Omega_0 t) + w(nT). \tag{1.21}$$

This signal is then multiplied by sinusoids, like in continuous time case, to give following terms.

On the I branch

$$r(nT)\sqrt{2}\cos(\Omega_0 n) = I_r(nT) + I_r(nT)\cos(2\Omega_0 n) - Qr(nT)\sin(2\Omega_0 n) + w(nT). \tag{1.22}$$

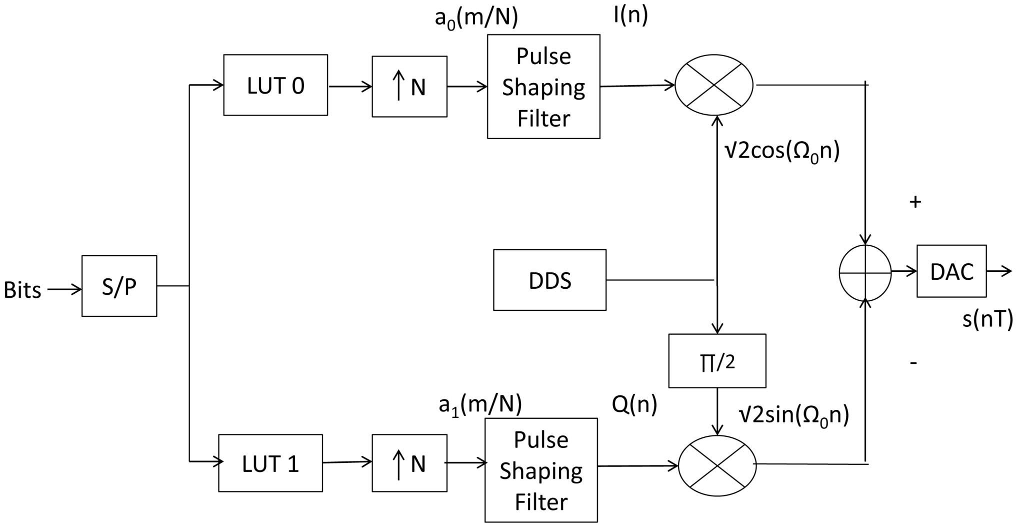


Fig. 1.6: Block diagram of discrete time QPSK modulator.

On the Q branch

$$r(nT)\sqrt{2}\sin(\Omega_0 n) = -Q_r(nT) + I_r(nT)\cos(2\Omega_0 n) + Qr(nT)\sin(2\Omega_0 n) + w(nT). \quad (1.23)$$

So, the match filter rejects high frequency terms the double frequency terms are rejected matched filter output are

$$x(nT) = \sum_{l=\frac{T_1}{T}+n}^{\frac{T_2}{T}+n} I_r(lT)p(lT - nT) + w_0(nT), \quad (1.24)$$

$$y(nT) = \sum_{l=\frac{T_1}{T}+n}^{\frac{T_2}{T}+n} Q_r(lT)p(lT - nT) + w_1(nT). \quad (1.25)$$

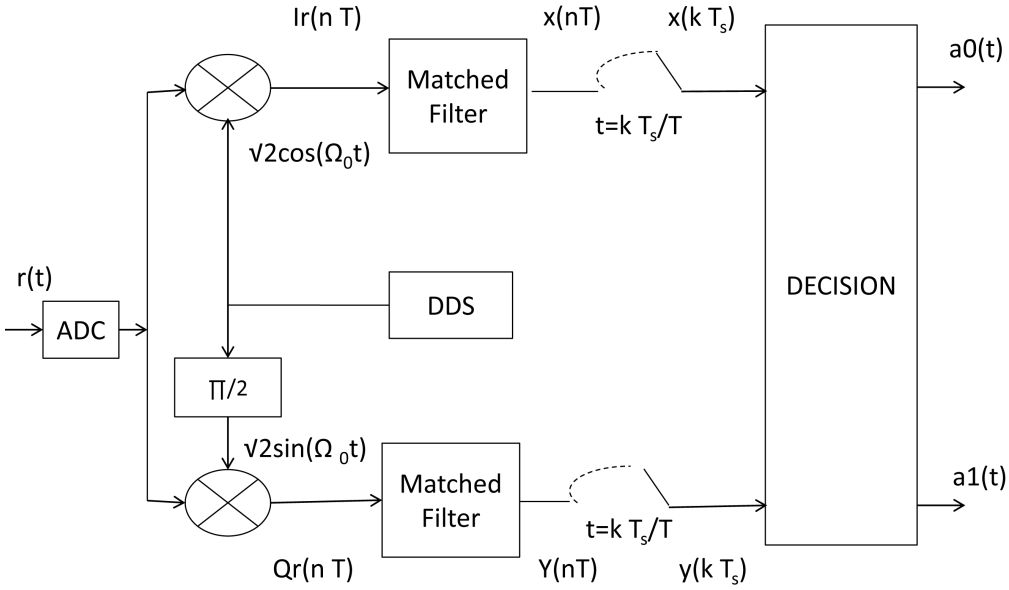


Fig. 1.7: Block diagram of discrete time QPSK receiver.

As in case of continuous time, it is assumed there is perfect phase, frequency, and timing synchronization give

$$x(nT) = \frac{1}{T}\sum_m a_0(m) r_p(mT_s - nT) + w_0(nT), \tag{1.26}$$

$$y(nT) = \frac{1}{T}\sum_m a_1(m) r_p(mT_s - nT) + w_1(nT). \tag{1.27}$$

If these are sampled at $n = kt_s/T$ and Nyquist-Non ISI condition is assumed, then

$$x(kT_s) = \sum_m a_0(m) r_p((m-k)T_s) + w_0(kT_s), \tag{1.28}$$

$$y(kT_S) = \sum_m a_1(m) r_p((m-k)T_s) + w_1(kT_s). \tag{1.29}$$

If we assume no ISI, then the samples become

$$x(kT_s) = a_0(m) + w_0(kT_s), \tag{1.30}$$

$$y(kT_S) = a_1(m) + w_1(kT_s). \tag{1.31}$$

The decision block then uses these to map the received symbols to constellation point and then converted to bits.

1.4 Offset-QPSK

Offset-QPSK is a modulation technique which is a variant of QPSK technique. Here the Q branch is delayed by half a symbol period. Thus the changes in phase on I and Q branch are not simultaneous. This is useful in preventing the amplifiers used in communication to being driven in saturation and cause an increase in bandwidth, called spectral regrowth. This can be attributed to amplitude changes which are not on the circle and changes passing through zero [16, 18].

The half symbol delay causes the Q branch to change at multiples of $\frac{T_s}{2}$ seconds. Hence in offset-QPSK system, both of the shortcomings are avoided and the spectral regrowth is reduced. The offset-QPSK system is represented in Figure 1.8 and the phase trajectories of the two techniques are shown in Figure 1.9. Here, a continuous time system is considered. The signal at output of pulse shaping filter is same as that of QPSK indicated in (1.3) and (1.4). Now a delay of half a symbol period is introduced in Q branch. The signals are then used to modulate the carrier sinusoids. On adding these two $s(t)$ becomes as

$$s(t) = \sum_k a_0(k)p(t - kT_s)\sqrt{2}\cos(w_0 t) - a_1(k)p(t - kT_s - \frac{T_s}{2})\sqrt{2}\sin(w_0 t). \tag{1.32}$$

The offset-QPSK detector is shown in Figure 1.10. The carrier is first removed and then signal passed through the matched filter which removes the double frequency terms. Also, if Nyquist Non-ISI condition is assumed, then the summation becomes

$$x(t) = \sum_m a_0(m)r_p(mT_s - t) + \text{noise}, \tag{1.33}$$

$$y(t) = \sum_m a_1(m)r_p(mT_s - t - \frac{T_s}{2}) + \text{noise}. \tag{1.34}$$

Now the signal is sampled. Since the Q Branch signal was delayed by $\frac{T_s}{2}$ thus the sample

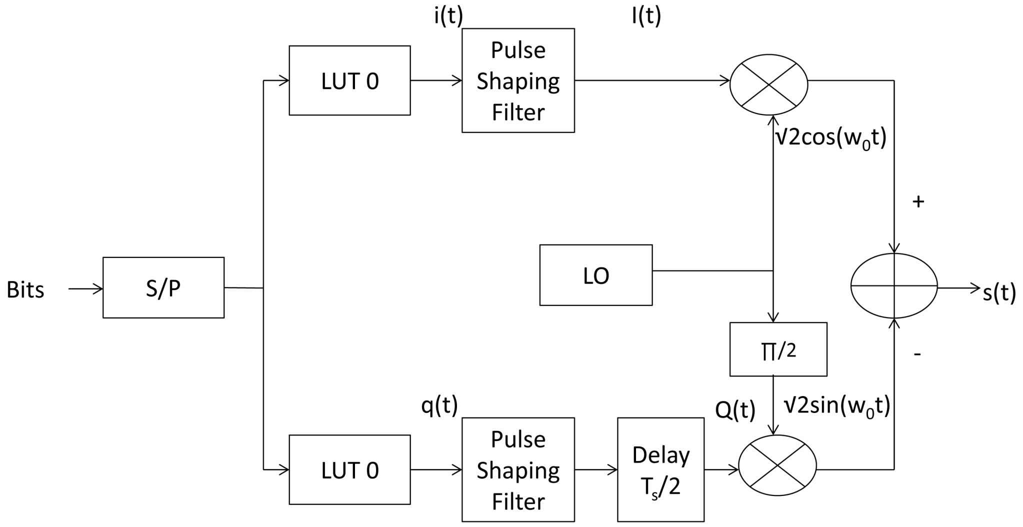
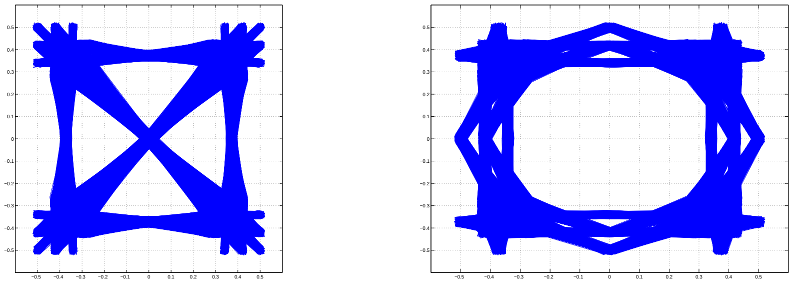


Fig. 1.8: Block diagram of offset-QPSK modulator.

used for decision block from Q stream is $y(nTs+Ts/2)$ and that in I stream is $x(nTs)$. Now these samples are used by decision block to map the sample to constellation points which are then converted back to bits. The system analysis for discrete time system proceeds in same way as for QPSK and has the Ts/2 delay discussed above.

1.5 Timing and Phase Offset

Any QPSK modulated signal, either offset or normal, when received for demodulation will have some phase, frequency, and timing offset. This may be due to limitation or



(a) QPSK phase trajectory.

(b) Offset-QPSK phase trajectory.

Fig. 1.9: Comparison of phase trajectories of offset-QPSK and QPSK modulation techniques.

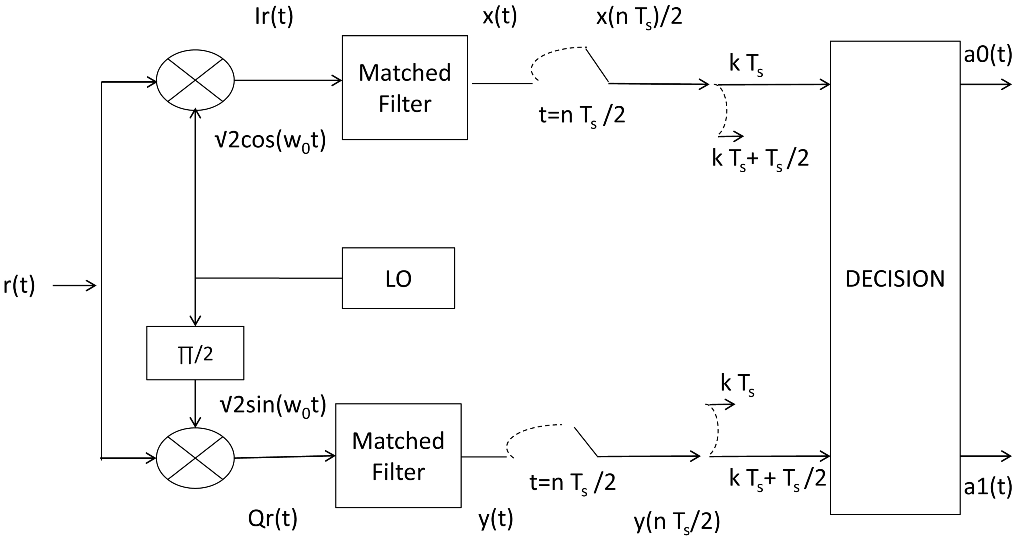


Fig. 1.10: Block diagram of offset-QPSK receiver.

synchronization error of the receiver components. If we have

$$r(t) = I(t)\sqrt{2}\cos(w_0 t + \theta) - Q(t)\sqrt{2}\sin(w_0 t + \theta). \tag{1.35}$$

Here, $I(t)$ and $Q(t)$ are as given in (1.3) and (1.4). The parameter θ is the phase difference or offset between received carrier and demodulator LO. Then using trigonometric identities for $\cos(A + B)$ and $\sin(A + B)$, $r(t)$ can be expanded as

$$r(t) = [I(t)\cos\theta - Q(t)\sin\theta]\sqrt{2}\cos(w_0 t) - [I(t)\cos\theta + Q(t)\sin\theta]\sqrt{2}\sin(w_0 t). \tag{1.36}$$

Now using these, as in Section 1.2, QPSK development,

$$x(kT_s) = \sum_m [a_0(m)\cos\theta - a_1(m)\sin\theta] r_p(mT_s - t) + \text{noise}, \tag{1.37}$$

$$y(kT_s) = \sum_m [a_0(m)\sin\theta - a_1(m)\cos\theta] r_p(mT_s - t) + \text{noise}. \tag{1.38}$$

Using the assumed no-ISI Nyquist properties it can be simplified to

$$x(kT_s) = a_0(k)\cos\theta - a_1(k)\sin\theta, \tag{1.39}$$

$$y(kT_s) = a_1(k)\sin\theta + a_1(k)\cos\theta. \tag{1.40}$$

There is a rotated component which hampers demodulation of received signal. There also may be some carrier offset in transmitted and receiver LO which is manifested in form of time-varying phase offset. To tackle this problem a PLL is used in the demodulator.

The second issue is timing synchronization which is the mismatch of sampling instant because of the time delay, τ, introduced.

$$r(t) = \sum_m [a_0(m) - a_1(m)] r_p(mT_s - t - \tau) + \text{noise}. \tag{1.41}$$

Here the carrier sinusoid term is not shown, Ts is the symbol time, τ is the time delay. Considering there is a timing estimate, $\hat{\tau}$ used and sampling done at $t = kT_s + \hat{\tau}$. Output of matched filter can be expressed as

$$x(kT_s + \hat{\tau}) = \sum_m a_0(m) r_p((m-k)T_s - t - \hat{\tau}) + \text{noise}, \tag{1.42}$$

$$y(kT_s + \hat{\tau}) = \sum_m a_1(m) r_p((m-k)T_s - t - \hat{\tau}) + \text{noise}. \tag{1.43}$$

Since this time delay is unknown quantity, another loop for timing synchronization is used in demodulator.

Chapter 2

Blind Equalization Techniques

Figure 2.1 shows a block diagram of a Data Aided equalizer [19]. It is seen that the data aided equalizers estimate the channel effects on the signal by comparing it with a reference signal. Generally, a training sequence is transmitted by the transmitter and a similar sequence of data is stored on receiver side. The data aided equalizers compare the two and estimate the channel effects. But the DICE satellites do not transmit any training sequence. Therefore, the data aided equalizers are not an option to be used. The blind equalizers, as shown in Figure 2.2, do not require any training data. These can be used

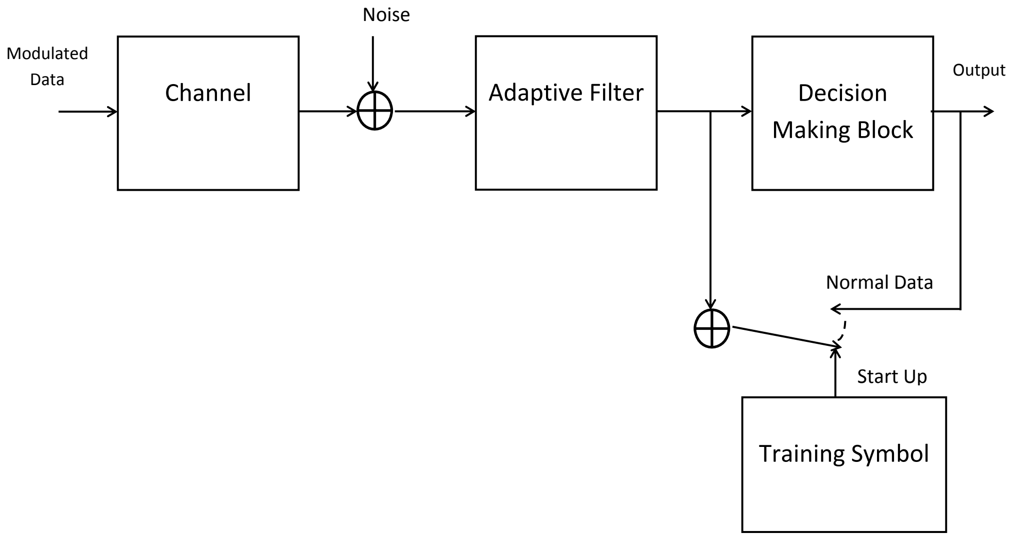


Fig. 2.1: Block diagram of data aided filter.

for the project being presented here. The blind equalizers generally exploit some invariant property of the signal being sent which gets affected by the interference or the transmission losses. The blind algorithms try to correct these properties and in process also correct the signal being received so that it can be demodulated. In the types of algorithms being used here, generally the constellation symmetry and shape is the property being used.

2.1 Constant Modulus Algorithm

Constant Modulus Algorithm (CMA) is based on the cost function in which the existing symbol's absolute value is compared with the constellation symbol's magnitude and the difference, called error, is minimized [2, 5, 19, 20]. Since it takes into account the absolute value or magnitude, it is named so. Consider

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k), \tag{2.1}$$

where y(k) is the k$^{\text{th}}$ ouput of filter, $\mathbf{w}(k) = [w_k(0), w_k(1), ..., w_k(L-1)]^T$ is the filter coefficients at k$^{\text{th}}$ iteration, L is the length of filter and $\mathbf{x}(k) = [x(k-L+1), x(k-L+2), ..., x(k-1), x(k)]^T$ is the input data vector.

The cost function of CMA algorithm [5, 20] is

$$J(k) = E[(|y(k)|^2 - R_2)^2]. \tag{2.2}$$

This cost function minimizes the deviation of the received symbol absolute value from the dispersion constant R_p. The value of dispersion constant is dependent on the constellation. This cost function is used to develop the update equation for updating the adaptive blind equalizer. When the cost function is minimized and the error is tending to zero

$$\begin{aligned} e(n) =& \frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = 0 \\ &\Rightarrow [(|y(k)|^2 - R_2)^2] = 0 \\ &\Rightarrow y(k) = \pm R_2. \end{aligned} \tag{2.3}$$

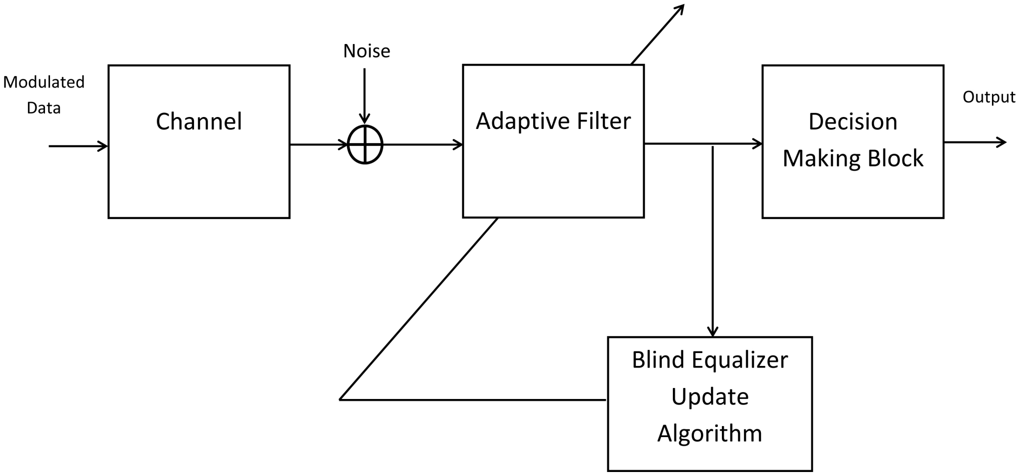


Fig. 2.2: Block diagram of blind equalizer filter.

But $y(k) = \sqrt{\text{real}(y(k))^2 + \text{imag}(y(k))^2}$. Hence, (2.3) becomes

$$\text{real}(y(k))^2 + \text{imag}(y(k))^2 = R_2^2. \tag{2.4}$$

This is an equation of circle. This shows that the CMA algorithm minimizes the error along a circle shown in Figure 2.3. The update equation is obtained using stochastic gradient of the cost function $J(k)$ [5, 7, 19]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda \frac{\partial J(k)}{\partial \mathbf{w}^*(k)}. \tag{2.5}$$

Solving for $\frac{\partial J(k)}{\partial \mathbf{w}^*(k)}$ from (2.5)

$$\frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = \frac{\partial(|y(k)|^2 - R_2)^2}{\partial \mathbf{w}^*(k)},$$

$$\frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = 2(|y(k)|^2 - R_2)\frac{\partial(|y(k)|)^2}{\partial \mathbf{w}^*(k)}. \tag{2.6}$$

Now solving for $\frac{\partial(|y(k)|)^2}{\partial \mathbf{w}^*(k)}$ and using identities from Moon and Stirling [21] we get,

$$\frac{\partial(|y(k)|)^2}{\partial \mathbf{w}^*(k)} = y(k)\mathbf{x}^*(k).$$

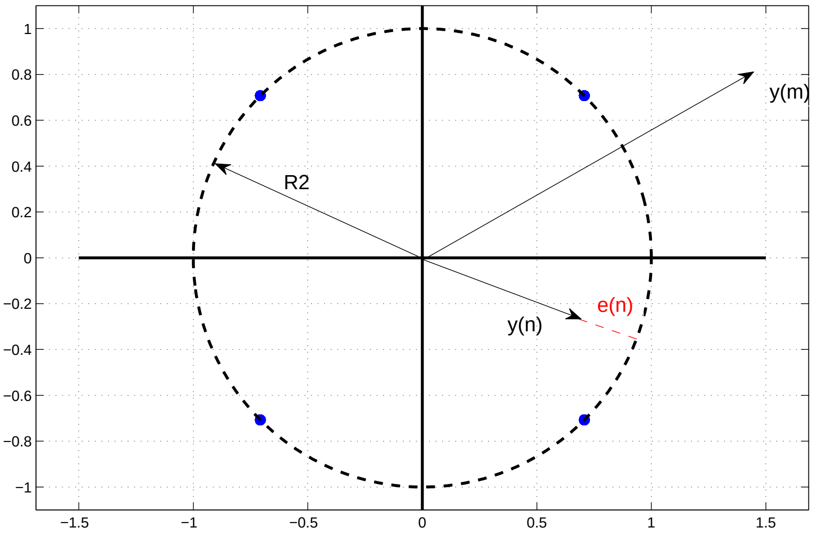


Fig. 2.3: Representation of error $e(n)$ of CMA algorithm.

Substituting this in (1.4) the update equation becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda(2(|y(k)|^2 - R_2)y(k)\mathbf{x}^*(k)).$$

The factor of 2 can be absorbed in the constant λ to give to give following update equation

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda(|y(k)|^2 - R_2)y(k)\mathbf{x}^*(k). \tag{2.7}$$

2.2 CMA Algorithm for Offset-QPSK

The CMA can readily be extended to offset-QPSK case. Here the real part and imaginary part of symbol, in other words, I and Q branch, differ in symbol timing by $\frac{T_s}{2}$ = half a symbol period. The block diagram is shown in Figure 2.4. The real part of symbol is $y(k)$ and the imaginary part is $y(k + \frac{1}{2}T_s)$. Here

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{2.8}$$

$$y(k + \frac{1}{2}) = \mathbf{w}^T(k + \frac{1}{2})\mathbf{x}(k + \frac{1}{2}).$$

$\mathbf{w}(k)$ and $\mathbf{x}(k)$ are as defined in (2.1). And the cost function to be minimizes becomes

$$\begin{aligned} J(k) =& E[(|\text{real}(y(k)) + j\text{imag}(y(k+\frac{1}{2}))|^2 - R_2)^2] \\ &= E[(|c(k)|^2 - R_2)^2]. \end{aligned}$$

Here $c(k) = \text{real}(y(k)) + j\text{imag}(y(k+\frac{1}{2}))$. The update equation is derived using the gradient descent update equation as (2.5).

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda \frac{\partial J(k)}{\partial \mathbf{w}^*(k)} \tag{2.9}$$

Solving for $\frac{\partial J(k)}{\partial \mathbf{w}^*(k)}$

$$\frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = \frac{\partial(|c(k)|^2 - R_2)^2}{\partial \mathbf{w}^*(k)},$$

$$\frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = 2(|c(k)|^2 - R_2)\frac{\partial(|c(k)|)^2}{\partial \mathbf{w}^*(k)}. \tag{2.10}$$

Now solving for $\frac{\partial(|c(k)|)^2}{\partial \mathbf{w}^*(k)}$ and using identities from Moon and Stirling [21] and Hjorungnes and Gesbert [22]

$$\begin{aligned} \frac{\partial(|c(k)|)^2}{\partial \mathbf{w}^*(k)} =& 2|c(k)|.\frac{\partial(|c(k)|)}{\partial \mathbf{w}^*(k)} \\ &= 2|c(k)|.\frac{\partial(\sqrt{\text{real}(c(k))^2 + \text{imag}(c(k))^2}}{\partial \mathbf{w}^*(k)} \\ &= 2|c(k)|.\frac{1}{2.\sqrt{\text{real}(c(k))^2 + \text{imag}(c(k))^2}}.\frac{\partial(\sqrt{\text{real}(c(k))^2 + \text{imag}(c(k))^2}}{\partial \mathbf{w}^*(k)} \\ &= 2|c(k)|.\frac{1}{2.|c(k)|}.\frac{\partial(\sqrt{\text{real}(c(k))^2 + \text{imag}(c(k))^2}}{\partial \mathbf{w}^*(k)} \\ &= 2.\text{real}(y(k)).\frac{\partial \text{real}(y(k))}{\partial \mathbf{w}^*(k)} + 2.\text{imag}(y(k+\frac{1}{2})).\frac{\partial \text{imag}(y(k+\frac{1}{2}))}{\partial \mathbf{w}^*(k+\frac{1}{2})} \\ &= 2.\text{real}(y(k)).\frac{\partial \frac{(y(k)+y(k)^*)}{2}}{\partial \mathbf{w}^*(k)} + 2.\text{imag}(y(k+\frac{1}{2})).\frac{\partial \frac{(y(k+\frac{1}{2})-y(k+\frac{1}{2})^*)}{2j}}{\partial \mathbf{w}^*(k+\frac{1}{2})} \end{aligned} \tag{2.11}$$

$$
\begin{aligned}
&= \text{real}(y(k)).\frac{\partial(\mathbf{w}^T(k)\mathbf{x}(k) + (\mathbf{w}^T(k)\mathbf{x}(k))^*)}{\partial\mathbf{w}^*(k)} + \\
&\frac{\text{imag}(y(k+\frac{1}{2}))}{j}.\frac{\partial(\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}) - (\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}))^*)}{\partial\mathbf{w}^*(k+\frac{1}{2})} \\
&= \text{real}(y(k)).\frac{\partial(\mathbf{w}^T(k)\mathbf{x}(k) + \mathbf{x}^H(k)\mathbf{w}(k)^*)}{\partial\mathbf{w}^*(k)} + \\
&\frac{\text{imag}(y(k+\frac{1}{2}))}{j}.\frac{\partial(\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}) - (\mathbf{x}^H(k+\frac{1}{2})\mathbf{w}(k+\frac{1}{2})^*))}{\partial\mathbf{w}^*(k+\frac{1}{2})} \\
&= \text{real}(y(k)).\mathbf{x}^H - \frac{\text{imag}(y(k+\frac{1}{2}))}{j}.\mathbf{x}^H(k+\frac{1}{2}).
\end{aligned}
$$

Here it should be noticed that for making the vector to be appropriate for usage in update equation, the transpose of $\mathbf{x}^H(k)$ should be taken to give $\mathbf{x}^*(k)$. This gives the following update equation for the equalizer

$$
\begin{aligned}
\mathbf{w}(k+1) =& \mathbf{w}(k) - \lambda(|(c(k)|^2 - R_2)(\text{real}(c(k))\mathbf{x}^*(k) + \text{imag}(c(k))\mathbf{x}^*(k+\frac{1}{2})) \\
&= \mathbf{w}(k) - \lambda(|(c(k)|^2 - R_2)((p(k)). \qquad (2.12)
\end{aligned}
$$

Here $p(k) = \text{real}(c(k))\mathbf{x}^*(k) + \text{imag}(c(k))\mathbf{x}^*(k+\frac{1}{2})$. The CMA algorithm does not compensate for frequency and phase offset. So a PLL is needed to combat these.The algorithm used for CMA blind equalization is presented below.

Algorithm 2.2.0: CMA ALGORITHM FOR OFFSET-QPSK($\mathbf{w}, \mathbf{x}$)

$$
\textbf{Initialization}
\begin{cases}
\mathbf{w} = \text{delta impulse} \\
\lambda = \text{convergence factor} \\
R_2 = \frac{E[|a(n)|^4]}{E[|a(n)|^2]} \\
\text{index} = 0
\end{cases}
$$

```
for index ← 0 to end of data samples
do
    update input data vector $\mathbf{x}$ with new data
    calculate filter output $y(index) = \mathbf{w}^T(index)\mathbf{x}(index)$

    if index $= m(index + \frac{1}{2}T_s), m \in \mathbb{Z}$
    then
        form the symbol $c(index) = \text{Real}\ \{y(index - \frac{1}{2}T_s)\} + j\ \text{Imag}\ \{y(index)\}$
        update the filter $\mathbf{w}(index + 1) = \mathbf{w}(index) - \lambda(|c(index)|^2 - R_2)(p(k))$
    calculate and update error vector
    index = index + 1
```

2.3 Multi-Modulus Algorithm

Multi-Modulus Algorithm (MMA) is based on comparing the absolute value of real and imaginary part of the equalizer output separately to dispersion constants. This means that the I branch and Q branch are separately equalized to minimize the deviation of their absolute values from the dispersion constants. The MMA type algorithms also are able to compensate for phase offset. The cost function is [6, 7, 15, 23, 24]

$$J(k) = E[(|y_R|^2 - R_R)^2] + E[(|y_I|^2 - R_I)^2]. \tag{2.13}$$

Here, y_R and y_I are the real and imaginary part of the equalizer output $y(k)$ as defined in (2.1). R_R and R_I are dispersion constants. As it is seen, the real and imaginary part are separated in cost function, giving it different properties from CMA type algorithms. When the cost function is minimized and the error is tending to zero

$$\begin{aligned} e(n) =& \frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = 0 \\ &\Rightarrow [(|y_R|^2 - R_R)^2] = 0 \text{ and } [(|y_R|^2 - R_I)^2] = 0 \\ &\Rightarrow y_R = \pm R_R \text{ and } y_I = \pm R_I. \end{aligned} \tag{2.14}$$

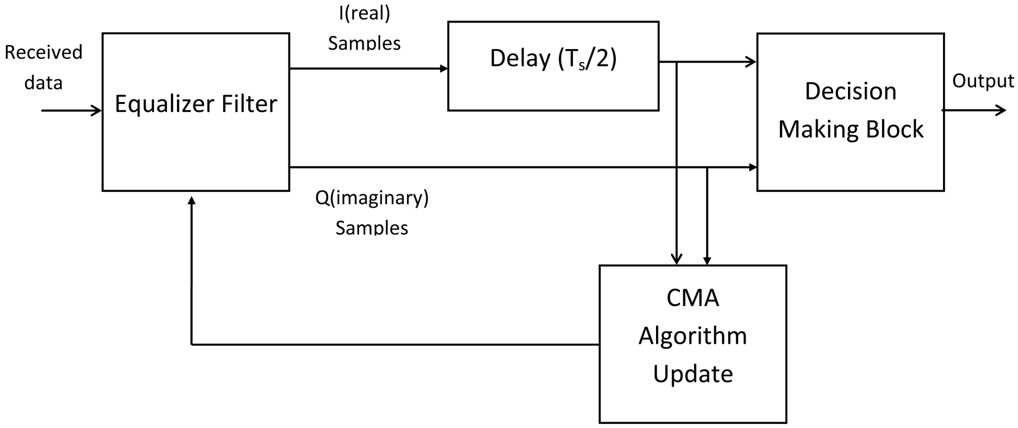


Fig. 2.4: Block diagram CMA equalizer algorithm for offset-QPSK.

This is shown in Figure 2.5. Here too, a stochastic gradient method is used to derive update equation for the filter equations

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda \frac{\partial J(k)}{\partial \mathbf{w}^*(k)}. \tag{2.15}$$

Solving for $\frac{\partial J(k)}{\partial \mathbf{w}^*(k)}$ from

$$\frac{\partial J(k)}{\partial \mathbf{w}^*(k)} = \frac{\partial (|y_R|^2 - R_R)^2}{\partial \mathbf{w}^*(k)} + j\frac{\partial (|y_I|^2 - R_I)^2}{\partial \mathbf{w}^*(k)}. \tag{2.16}$$

Solving for real part and using equations from Moon and Stirling [21] and Hjorungnes and Gesbert [22]

$$\begin{aligned} \frac{\partial (|y_R|^2 - R_R)^2}{\partial \mathbf{w}^*(k)} &= 2(|y_R|^2 - R_R)\frac{\partial (|y_R)^2}{\partial \mathbf{w}^*(k)} \\ &= 2(|y_R|^2 - R_R)(|y_R|)\frac{(\mathbf{x}^*(k)}{2} \\ &= (|y_R|^2 - R_R)(|y_R|)(\mathbf{x}^*(k)). \end{aligned} \tag{2.17}$$

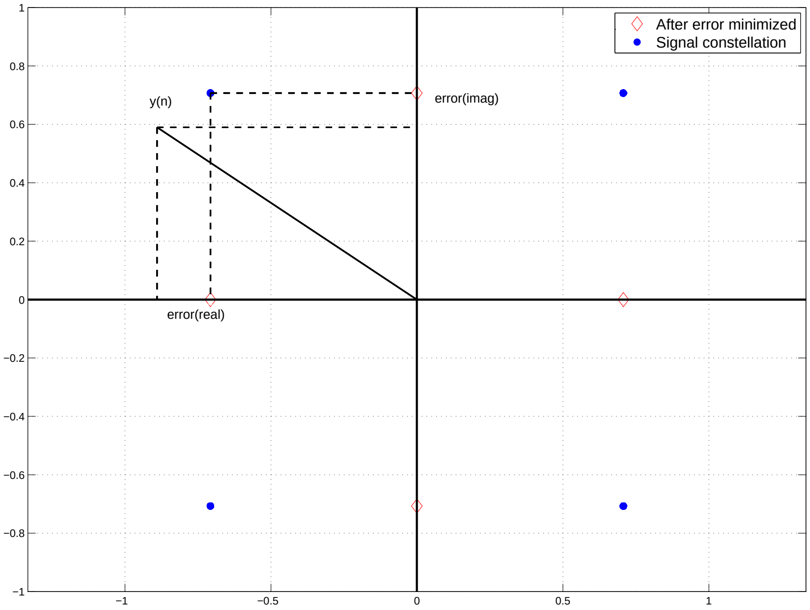


Fig. 2.5: Representation of minimized error of MMA algorithm.

Similarly, the imaginary part of (2.16)

$$\begin{aligned} \frac{\partial(|y_I|^2 - R_I)^2}{\partial \mathbf{w}^*(k)} &= 2(|y_I|^2 - R_I)\frac{\partial(|y_I)^2}{\partial \mathbf{w}^*(k)} \\ &= 2(|y_I|^2 - R_I)(|y_I|)(j)\frac{(\mathbf{x}^*(k))}{2}. \end{aligned} \tag{2.18}$$

Substituting (2.17) and (2.18) in (2.16)

$$\begin{aligned} \frac{\partial J(k)}{\partial \mathbf{w}^*(k)} &= (|y_R|^2 - R_R)(|y_R|)(\mathbf{x}^*(k)) - j(|y_I|^2 - R_I)(|y_I|)(\mathbf{x}^*(k)) \\ &= [(|y_R|^2 - R_R)(|y_R|)) - j(|y_I|^2 - R_I)(|y_I|)](\mathbf{x}^*(k)). \end{aligned} \tag{2.19}$$

And substituting this expression in (2.15) the filter update equation is

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \lambda[(|y_R|^2 - R_R)(|y_R|)) - j(|y_I|^2 - R_I)(|y_I|)](\mathbf{x}^*(k)). \tag{2.20}$$

2.4 MMA Algorithm for Offset-QPSK

Extending MMA for offset-QPSK algorithm needs the equalization of the real and imaginary streams of symbols separately. Also, since the samples are separated by half symbol delay, it is easier to equalize the real or imaginary part of the filter output appropriately. Here no PLL is needed to combat the phase offset and the MMA inherently compensates for such effects. The block diagram for algorithm is shown in Figure 2.6. For offset-QPSK case the y_R and y_I terms are separated by $\frac{T_s}{2}$ time period and after sampling appropriately can be represented as

$$y_R = \text{Real}\{y(k)\} = \frac{y(k) + y(k)^*}{2}, \tag{2.21}$$

$$y_I = \text{Imag}\{y(k + \frac{1}{2})\} = \frac{y(k+\frac{1}{2}) - y(k+\frac{1}{2})^*}{2j}. \tag{2.22}$$

The filter is updated two times in one symbol period. First it is updated when the real or I branch sample is received and next when the imaginary or Q branch sample is received. When the real update is to be done, $\frac{\partial(|y_R|^2 - R_R)^2}{\partial \mathbf{w}^*(k)}$ to be used in update equation is derived as following,

$$\begin{aligned}
\frac{\partial(|y_R|^2 - R_R)^2}{\partial \mathbf{w}^*(k)} &= 2.(|y_R|^2 - R_R).\frac{\partial(|y_R|^2 - R_R)}{\partial \mathbf{w}^*(k)} \\
&= 2.(|y_R|^2 - R_R).2|y_R|.\frac{\partial \frac{y(k)+y(k)^*}{2}}{\partial \mathbf{w}^*(k)} \\
&= 2.(|y_R|^2 - R_R).|y_R|.\frac{\partial(y(k) + y(k)^*)}{\partial \mathbf{w}^*(k)}. \\
&= 2.(|y_R|^2 - R_R).|y_R|.\frac{\partial(\mathbf{w}^T(k)\mathbf{x}(k) + (\mathbf{w}^T(k)\mathbf{x}(k))^*)}{\partial \mathbf{w}^*(k)} \\
&= 2.(|y_R|^2 - R_R).|y_R|.\frac{\partial(\mathbf{w}^T(k)\mathbf{x}(k) + \mathbf{x}^H(k)\mathbf{w}(k)^*)}{\partial \mathbf{w}^*(k)} \\
&= 2.(|y_R|^2 - R_R).|y_R|.\mathbf{x}^H(k).
\end{aligned} \tag{2.23}$$

Here noise term is neglected and transpose of $\mathbf{x}^H(k)$ taken to give proper dimension of the vector. This can be mathematically represented as, for real or I branch sample, when

$t = kT_s$,

$$\begin{aligned}\mathbf{w}(k+\frac{1}{2}) =& \mathbf{w}(k) - \lambda(|y_R|^2 - R_R)(|y_R|)\mathbf{x}^*(k)\\ =& \mathbf{w}(k) - \lambda(\text{Real}\{y(k)\}^2 - R_R)|\text{Real}\{y(k)\}|\mathbf{x}^*(k).\end{aligned} \tag{2.24}$$

And now doing similar analysis for imaginary or Q branch sample

$$\begin{aligned}\frac{\partial(|y_I|^2 - R_I)^2}{\partial\mathbf{w}^*(k+\frac{1}{2})} =& 2.(|y_I|^2 - R_I).\frac{\partial(|y_I|^2 - R_I)}{\partial\mathbf{w}^*(k+\frac{1}{2})}\\ &= 2.(|y_I|^2 - R_I).2|y_I|.\frac{\partial\frac{y(k+\frac{1}{2})-y(k+\frac{1}{2})^*}{2j}}{\partial\mathbf{w}^*(k+\frac{1}{2})}\\ &= \frac{2}{j}.(|y_I|^2 - R_I).|y_I|.\frac{\partial(y(k+\frac{1}{2}) - y(k+\frac{1}{2})^*)}{\partial\mathbf{w}^*(k+\frac{1}{2})}.\\ &= \frac{2}{j}.(|y_I|^2 - R_I).|y_I|.\frac{\partial(\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}) - (\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}))^*)}{\partial\mathbf{w}^*(k+\frac{1}{2})}\\ &= \frac{2}{j}.(|y_I|^2 - R_I).|y_I|.\frac{\partial(\mathbf{w}^T(k+\frac{1}{2})\mathbf{x}(k+\frac{1}{2}) - \mathbf{x}^H(k+\frac{1}{2})\mathbf{w}(k+\frac{1}{2})^*)}{\partial\mathbf{w}^*(k+\frac{1}{2})}\\ &= 2j.(|y_I|^2 - R_I).|y_I|.\mathbf{x}^H(k+\frac{1}{2}).\end{aligned} \tag{2.25}$$

Here noise term is neglected and transpose of $\mathbf{x}^H(k)$ taken to give proper dimension of the vector. This can be mathematically represented as, for imaginary or Q branch sample, when $t = k + \frac{1}{2}T_s$,

$$\begin{aligned}\mathbf{w}(k+1) =& \mathbf{w}(k+\frac{1}{2}) + j\lambda(|y_I|^2 - R_I)(|y_I|)\mathbf{x}^*(k+\frac{1}{2})\\ =& \mathbf{w}(k+\frac{1}{2}) + j\lambda(\text{Imag}\{y(k+\frac{1}{2})\}^2 - R_I)|\text{Imag}\{y(k+\frac{1}{2}\}|\mathbf{x}^*(k+\frac{1}{2}).\end{aligned} \tag{2.26}$$

The algorithm used in the thesis for MMA blind equalization is presented below.

Algorithm 2.4.0: MMA ALGORITHM FOR OFFSET-QPSK($\mathbf{w}, \mathbf{x}$)

```
Initialization
    $\mathbf{w} =$ delta impulse
    $\lambda =$ convergence factor
    $R_R = \frac{E[|a(n)|^4]}{E[|a(n)|^2]}$
    $R_I = R_R$
    index $= 0$
for index $\leftarrow 0$ to end of data samples
    do
        update input data vector $\mathbf{x}$ with new data
        calculate filter output $y(indx) = \mathbf{w}^T(k)\mathbf{x}(k)$

        if index $= mk, m \in \mathbb{Z}$
            then
                assign real part of sample $y_R =$ Real $y(indx)$
                update the filter $\mathbf{w}(k+\frac{1}{2}) = \mathbf{w}(k) - \lambda((|y_R|^2 - R_R)(|y_R|)).\mathbf{x}^*(k)$
        if index $= m(k+\frac{1}{2}T_s), m \in \mathbb{Z}$
            then
                assign imaginary part of sample $y_I =$ Imag $y(indx)$
                update the filter $\mathbf{w}(k+1) = \mathbf{w}(k+\frac{1}{2}) - \lambda(|y_I|^2 - R_I)(|y_I|).\mathbf{x}^*(k)$
        calculate and update error vector
        index $=$ index $+ 1$
```

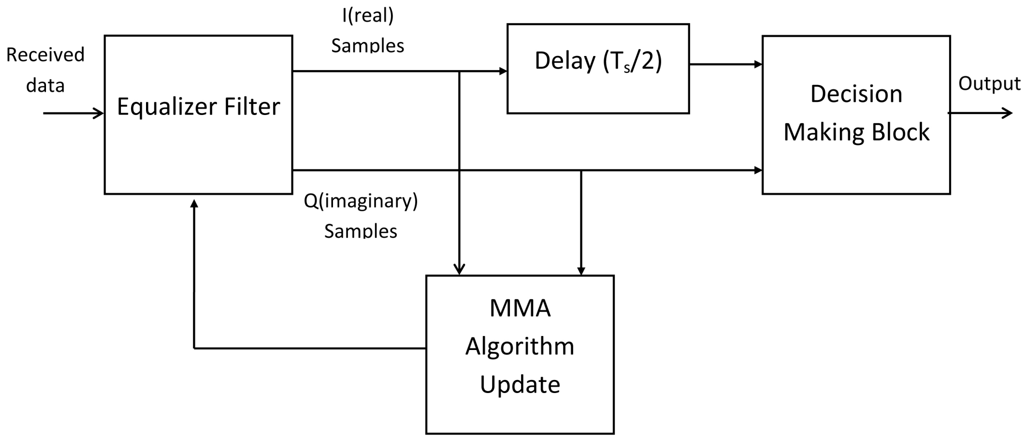


Fig. 2.6: Block diagram of MMA equalizer algorithm for offset-QPSK.

Chapter 3

Implementation and Results

In this section, the implementation and the results of the algorithms described earlier are presented. The CMA and MMA algorithms were coded in MATLAB program and plotted. The DICE data files were also used to test the algorithm code and results are presented. For CMA, the algorithm developed in Section 2.1 was implemented with and without PLL. The MMA algorithm, as shown in Section 2.3, too was developed with and without PLL and compared appropriately with other. The algorithms were first tested on simulated data and then used on real DICE data files.

3.1 Comparison of CMA and MMA Without PLL

First implementation of the CMA and MMA algorithms is done on simulated data. Here there is no PLL present, and thus it compares the performances of the two algorithms without any aid for combating phase and frequency offset.

3.1.1 No Phase Offset Present

In the first simulation there is no phase offset in the simulated data. The parameters of the SRRC pulse shaping filter are 70% excess bandwidth, the time truncation factor $L_p = 12$ and symbol time $T_s = 1$. The modulation type is offset-QPSK. Here the SNR is at 12 db. Comparing the plots given in Figures 3.1, 3.2, and 3.3, it can be seen that CMA and MMA have comparable performance. The MMA equalizer starts off with high error but converges later to have slightly smaller steady state value as compared to CMA. The channel [6] response and the resulting equalizer responses are also shown in Figures 3.4 - 3.13.

3.1.2 With Phase Offset

In second case there is a phase offset present. The pulse shaping filter characteristics and SNR remain the same. Here on comparing the plots as shown in Figures 3.14, 3.15, and 3.16, it is seen that CMA equalizer does not converge. The scatter plot does not show any well defined constellation from where accurate decisions can be made. The error thus does reduce to a lower value. MMA algorithm works well in this case and the scatter plor plot show well separated symbol constellation regions. The error converges to a very low value. The convergence of the error takes slightly longer time as compared to previous case. The plots of channel used and equalizer parameters are shown in Figures 3.17 - 3.26.

3.2 Comparison of CMA and MMA With PLL

Now in this case real data from DICE satellite used to compare the performances of the two equalizers. Here, both equalizers have a PLL which helps in combating the small phase offset, if any present, in data files of DICE satellite. There is no matched filter used here because of some ambiguity of the pulse shaping filter being implemented in the satellite communication module. Also, additional processing is done to remove any residual carrier frequency removal.

To tackle residual carrier offset here, the incoming signal is multiplied with a sinusoid. The frequency of this is estimated using Fourier transform of the incoming signal [14]

$$z(k) = \sum_{n=0} L_f - 1 y^{N_M}(n) e^{-j \frac{2\pi}{L_f} nk}. \tag{3.1}$$

This is done using fast Fourier transform. The maximum of the absolute value of the Fourier transform indicates where the residual frequency offset is present.

$$\Delta \hat{f} = \frac{\hat{k}}{N_M L_f}, \tag{3.2}$$

where

$$\hat{k} = \underset{0 \leq k \leq K}{\operatorname{argmax}} |z(k)|.$$

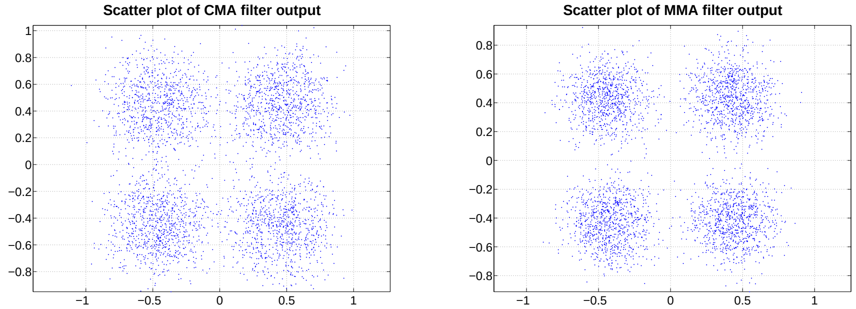
The received signal stream is then multiplied with an exponential having negative of the frequency estimated, thus removing the residual carrier offset. PLL is also implemented along with the equalizers. The PLL phase update equation [14] is

$$\theta_{n+1} = \theta_n + \alpha[\text{imag}(y(n)y^*(n)) + \gamma \sum_{i=0}^{n-1} \text{imag}(y(i)y^*(i))], \tag{3.3}$$

and the block diagram of implementation is shown in Figure 3.27. The performance of the equalizers can be evaluated based on plots shown in Figures 3.28, 3.29, and 3.30. The figures show the scatter plot of MMA equalizer is tighter than the CMA equalizer. Though CMA equalizer scatter plot show sufficient separation of constellation symbol groups for decision making. But the MMA equalizer gives a tighter constellation grouping of symbols. This leads to lower mean squared error, as shown in Figure 3.30. Also, it can be observed that MMA equalizer converges slightly faster. Thus, here superiority of MMA equalizer algorithm over CMA equalizer algorithm, for demodulating real data of DICE satellite, is illustrated. The plots of equalizer parameters are shown in Figures 3.31 - 3.34.

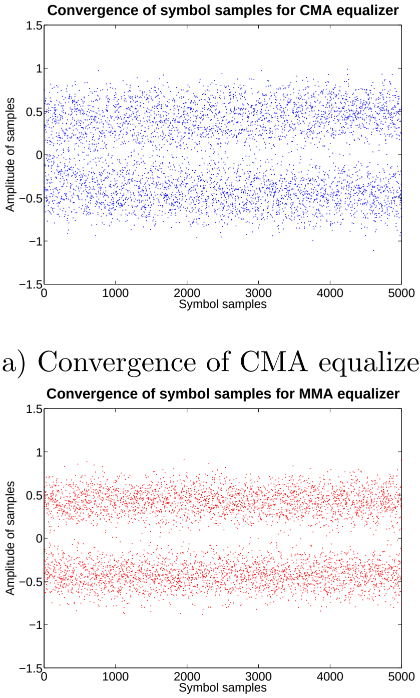
3.3 Comparison of MMA With and Without PLL

It is seen in previous section that MMA performs better than CMA for our data files. In this section, performance of MMA equalizer is evaluated with and without PLL. The data files used are of DICE satellite. And as explained before, no matched filtering is done but residual carrier offset is removed using the Fourier transform method. Figure 3.35 shows the scatter plot of the constellation after the MMA equalizer (with and without) PLL. Both the constellation groupings are very similar and are well separated for decisions to be made. The better performance of MMA equalizer for demodulation can be seen again here. The MMA equalizer performs equally well with and without PLL. The convergence and mean squared error plot, illustrated in Figures 3.36 and 3.37, also shows similar performance for both the cases. The DICE satellite data is demodulated with low error values and absence of PLL neither affects the speed of convergence nor the steady state error rate. The equalizer parameters are shown in Figures 3.38 - 3.41.



(a) Scatter plot of CMA equalizer. (b) Scatter plot of MMA equalizer.

Fig. 3.1: Comparison of scatter plots obtained by different blind equalizer algorithms without PLL and no phase offset.



(a) Convergence of CMA equalizer.

(b) Convergence of MMA equalizer.

Fig. 3.2: Convergence of symbol samples by different blind equalizer algorithms without PLL and no phase offset.

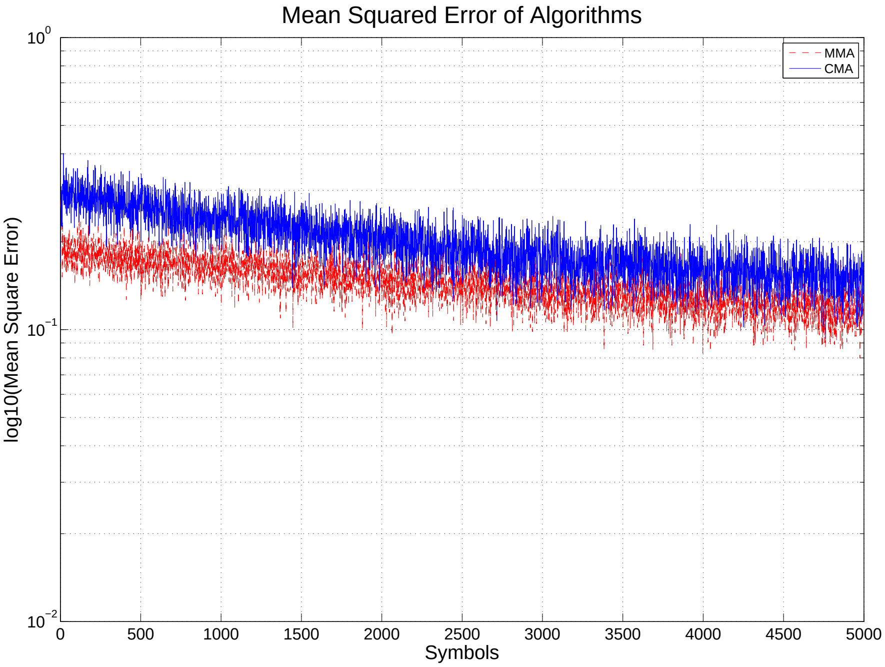
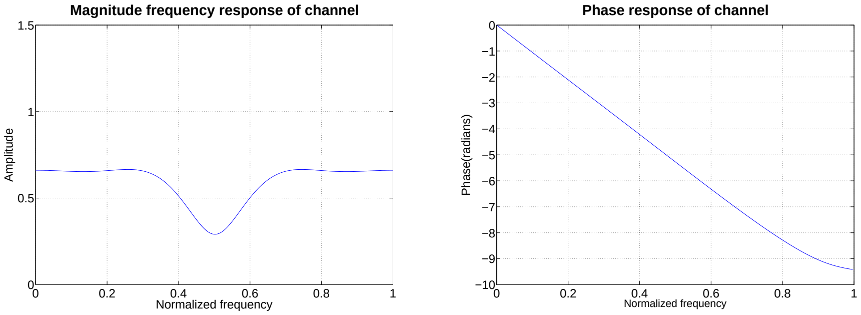


Fig. 3.3: MSE of different blind equalizer algorithms without PLL and no phase offset present.



(a) Frequency response.

(b) Phase response.

Fig. 3.4: Channel frequency and phase response used for simulated data without PLL and no phase offset present.

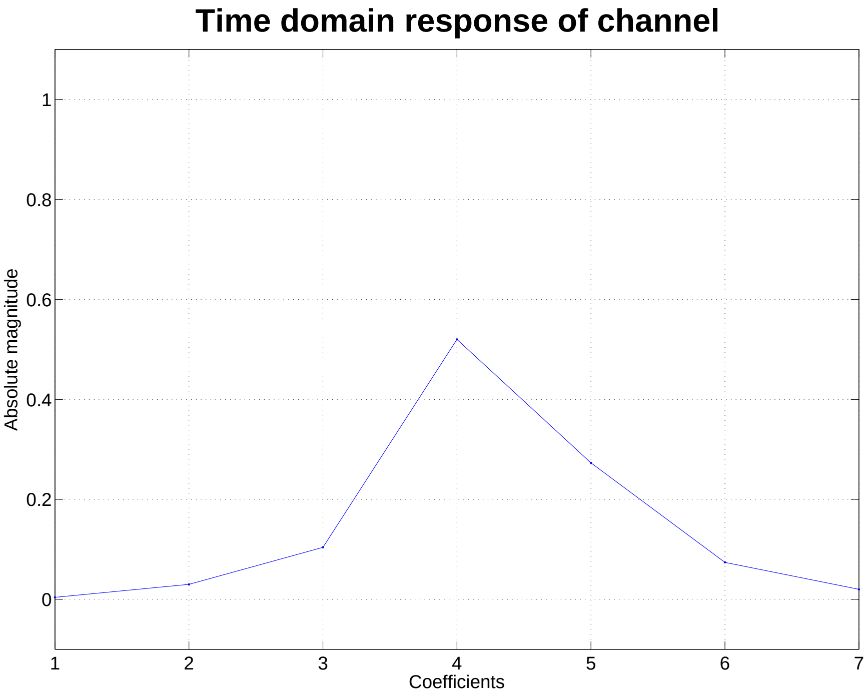
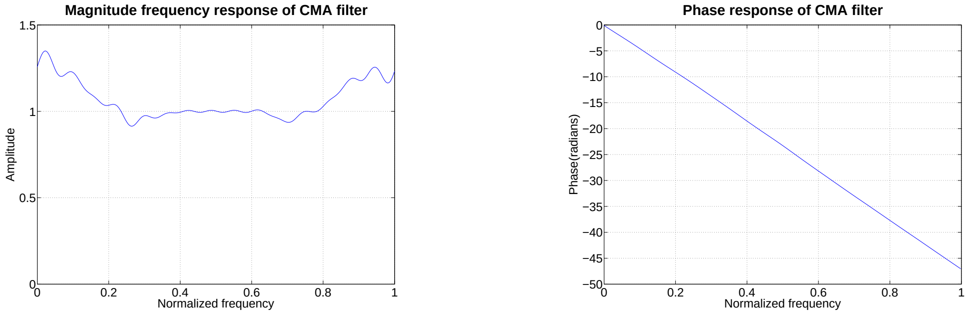


Fig. 3.5: Channel time domain coefficients used for simulated data without PLL and no phase offset present.



(a) Frequency response of the CMA equalizer.

(b) Phase response of the CMA equalizer.

Fig. 3.6: CMA equalizer frequency and phase response for simulated data without PLL and no phase offset present.

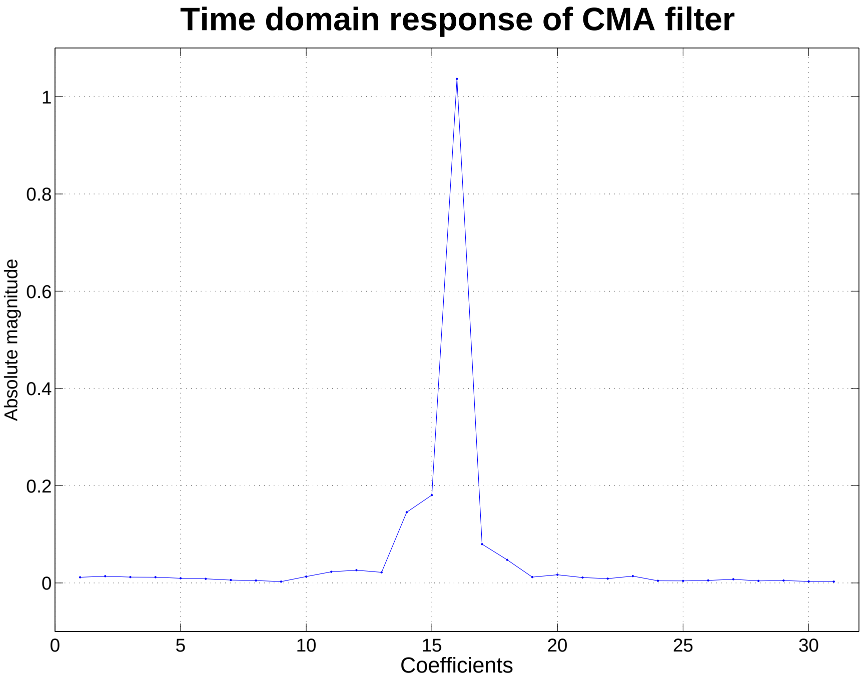
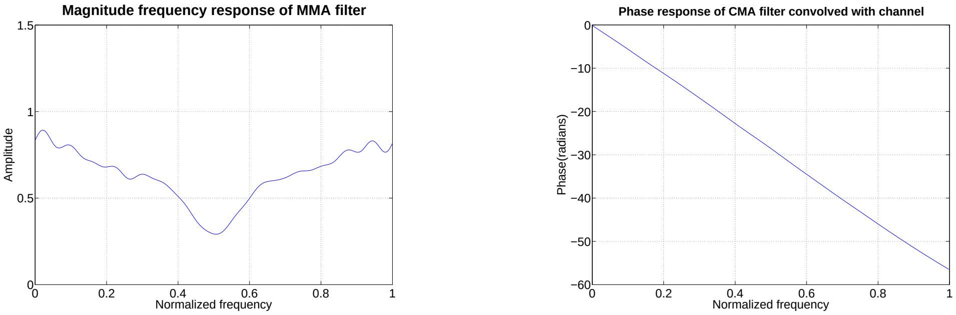


Fig. 3.7: CMA equalizer time domain coefficients for simulated data without PLL and no phase offset present.



(a) Frequency response of the CMA equalizer convolved with channel.

(b) Phase response of the CMA equalizer convolved with channel.

Fig. 3.8: CMA equalizer, convolved with channel, frequency, and phase response for simulated data without PLL and no phase offset present.

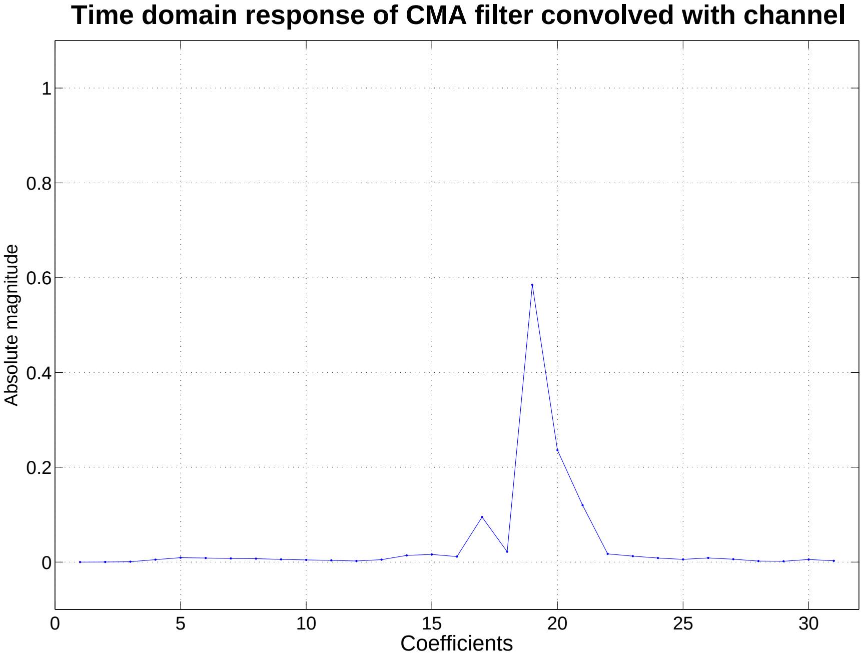
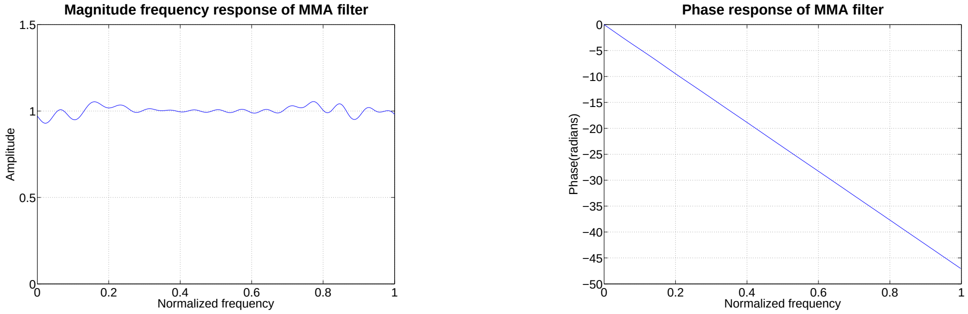


Fig. 3.9: CMA equalizer, convolved with channel, time domain coefficients for simulated data without PLL and no phase offset present.



(a) Frequency response of the MMA equalizer.

(b) Phase response of the MMA equalizer.

Fig. 3.10: MMA equalizer frequency and phase response for simulated data without PLL and no phase offset present.

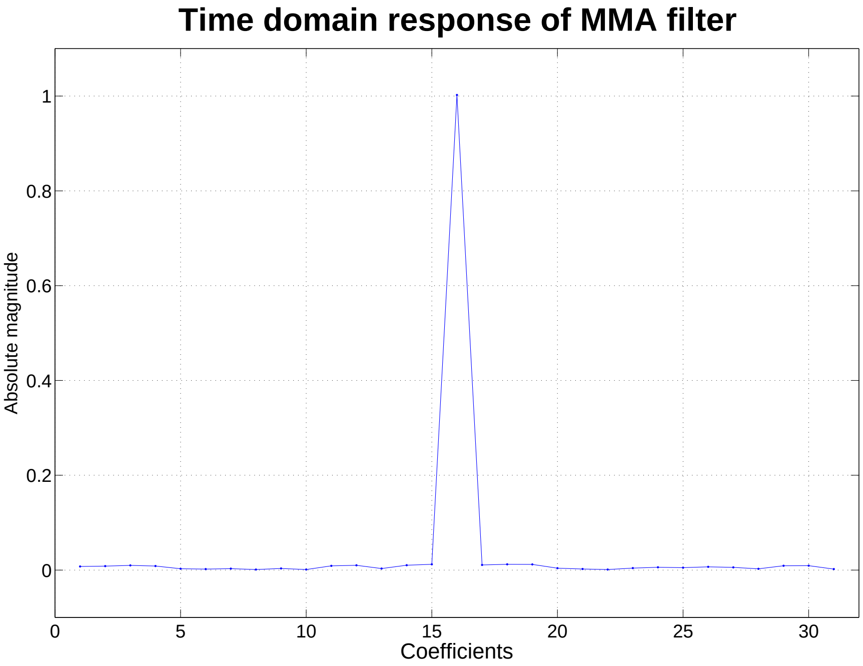
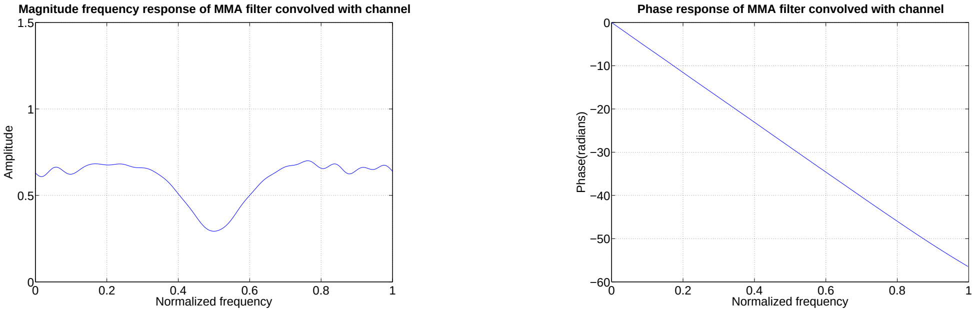


Fig. 3.11: MMA equalizer time domain coefficients for simulated data without PLL and no phase offset present.



(a) Frequency response of the MMA equalizer convolved with channel.

(b) Phase response of the MMA equalizer convolved with channel.

Fig. 3.12: MMA equalizer, convolved with channel, frequency, and phase response for simulated data without PLL and no phase offset present.

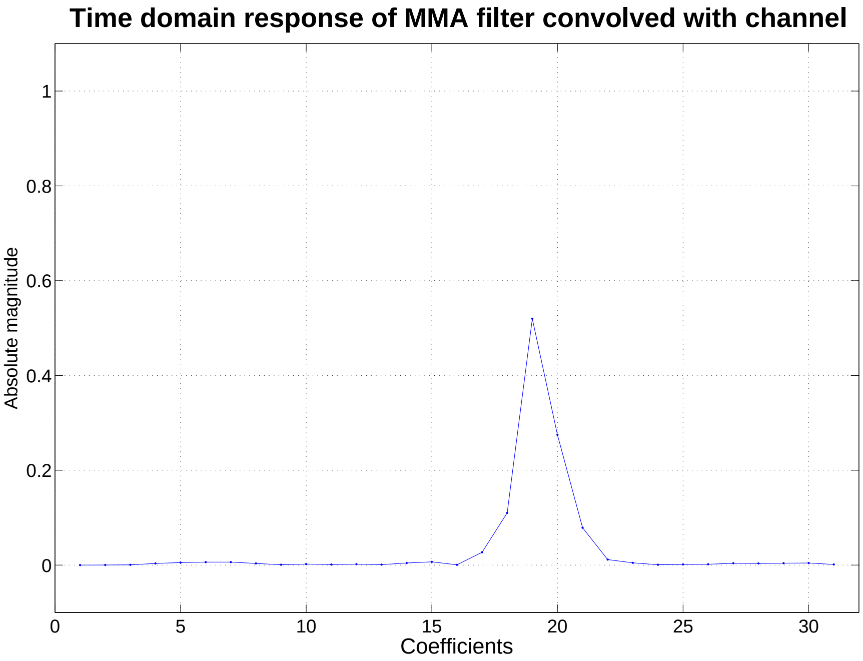
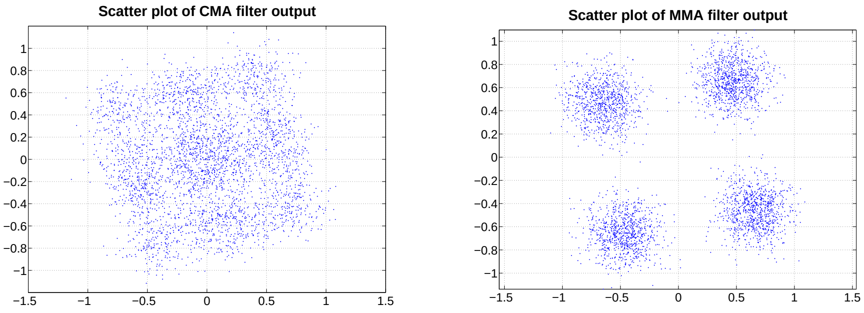
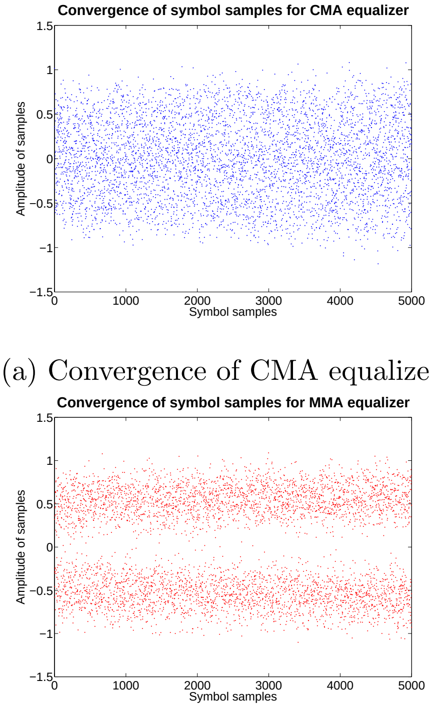


Fig. 3.13: MMA equalizer, convolved with channel, time domain coefficients for simulated data without PLL and no phase offset present.



(a) Scatter plot of CMA equalizer. (b) Scatter plot of MMA equalizer.

Fig. 3.14: Comparision of scatter plots obtained by different blind equalizer algorithms without PLL and with phase offset present.



(a) Convergence of CMA equalizer.

(b) Convergence of MMA equalizer.

Fig. 3.15: Convergence of symbol samples by different blind equalizer algorithms without PLL and with phase offset present.

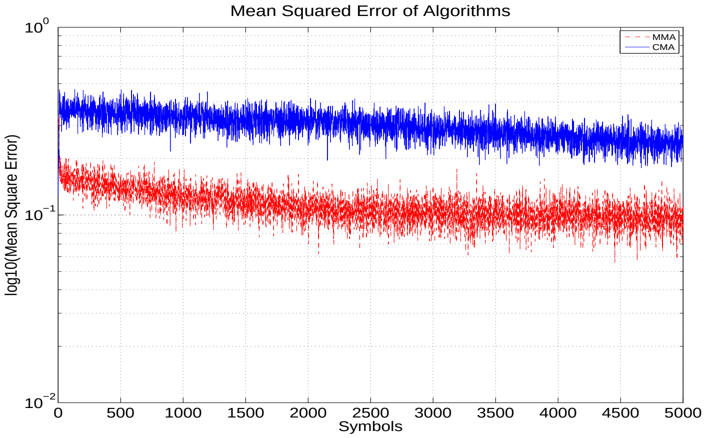
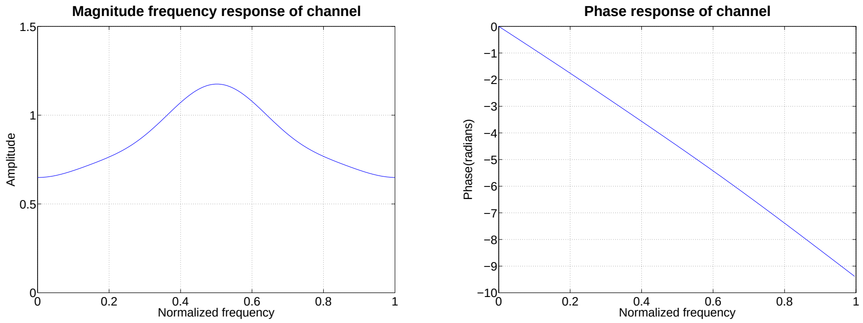


Fig. 3.16: MSE of different blind equalizer algorithms without PLL and with phase offset present.



(a) Frequency response.

(b) Phase response.

Fig. 3.17: Channel frequency and phase response used for simulated data without PLL and with phase offset present.

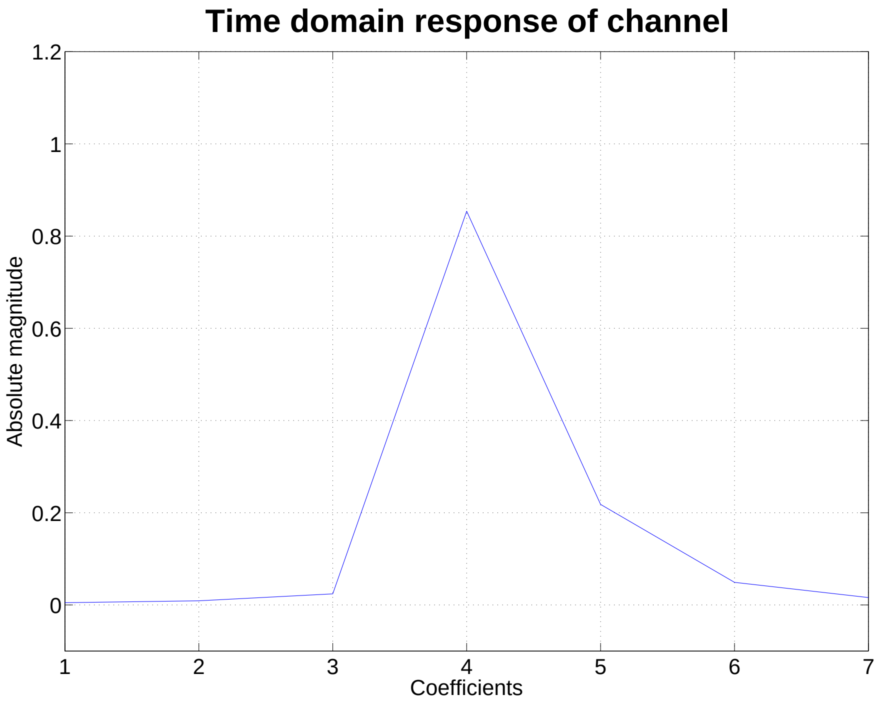
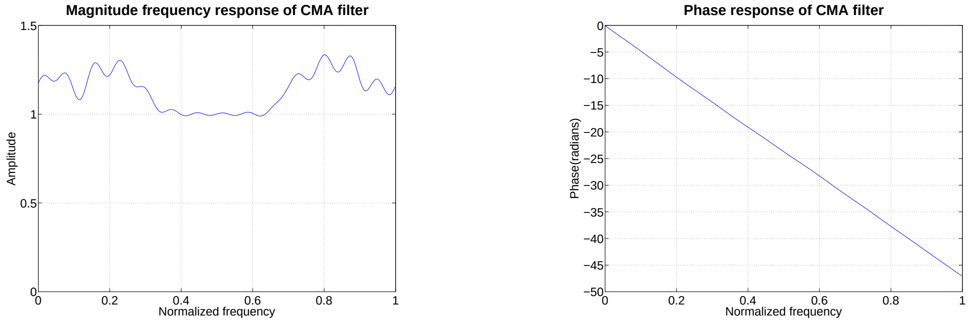


Fig. 3.18: Channel time domain coefficients used for simulated data without PLL and with phase offset present.



(a) Frequency response of the CMA equalizer.

(b) Phase response of the CMA equalizer.

Fig. 3.19: CMA equalizer frequency and phase response for simulated data without PLL and with phase offset present.

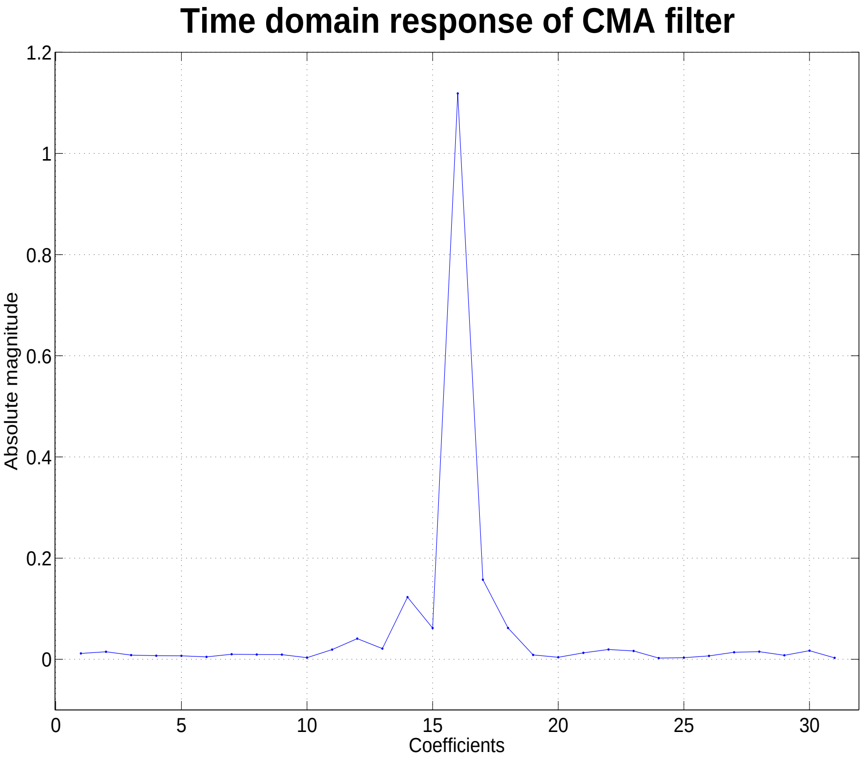
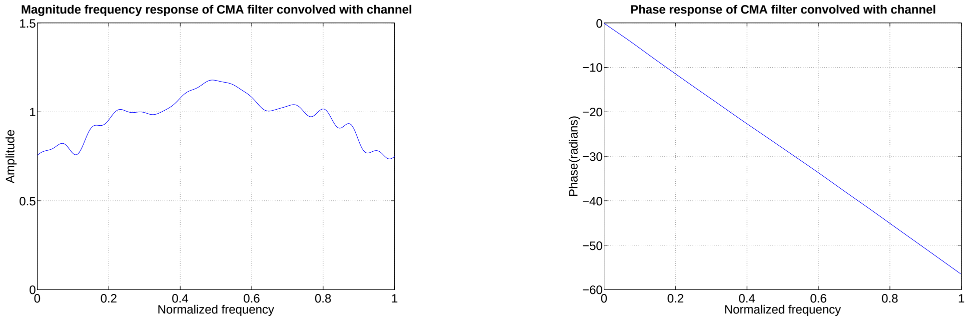


Fig. 3.20: CMA equalizer time domain coefficients for simulated data without PLL and with phase offset present.



(a) Frequency response of the CMA equalizer convolved with channel.

(b) Phase response of the CMA equalizer convolved with channel.

Fig. 3.21: CMA equalizer, convolved with channel, frequency, and phase response for simulated data without PLL and with phase offset present.

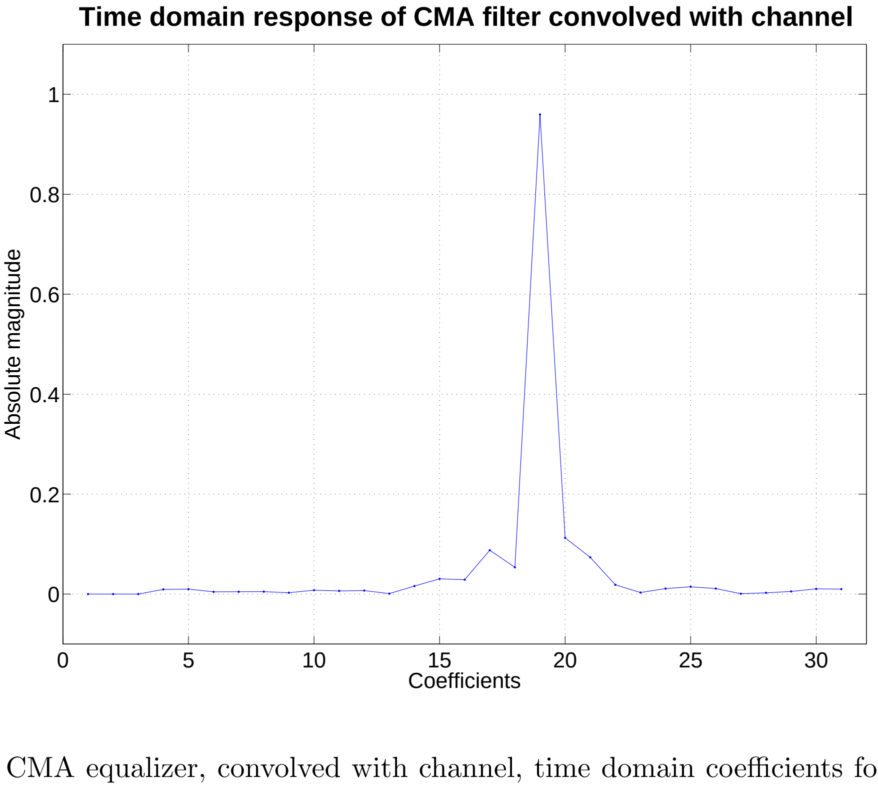
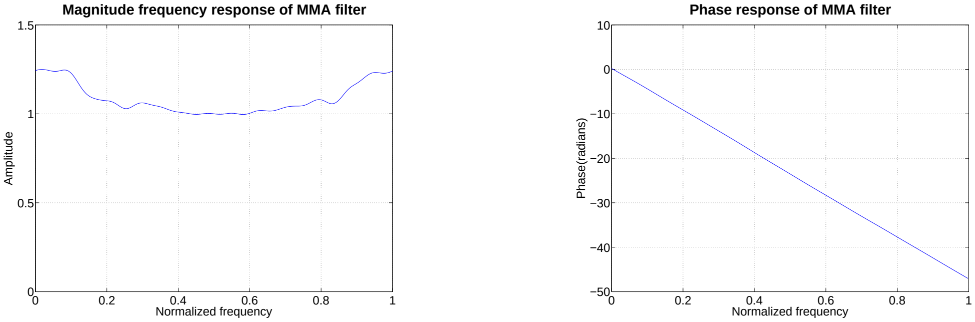


Fig. 3.22: CMA equalizer, convolved with channel, time domain coefficients for simulated data without PLL and with phase offset present.



(a) Frequency response of the MMA equalizer.

(b) Phase response of the MMA equalizer.

Fig. 3.23: MMA equalizer frequency and phase response for simulated data without PLL and with phase offset present.

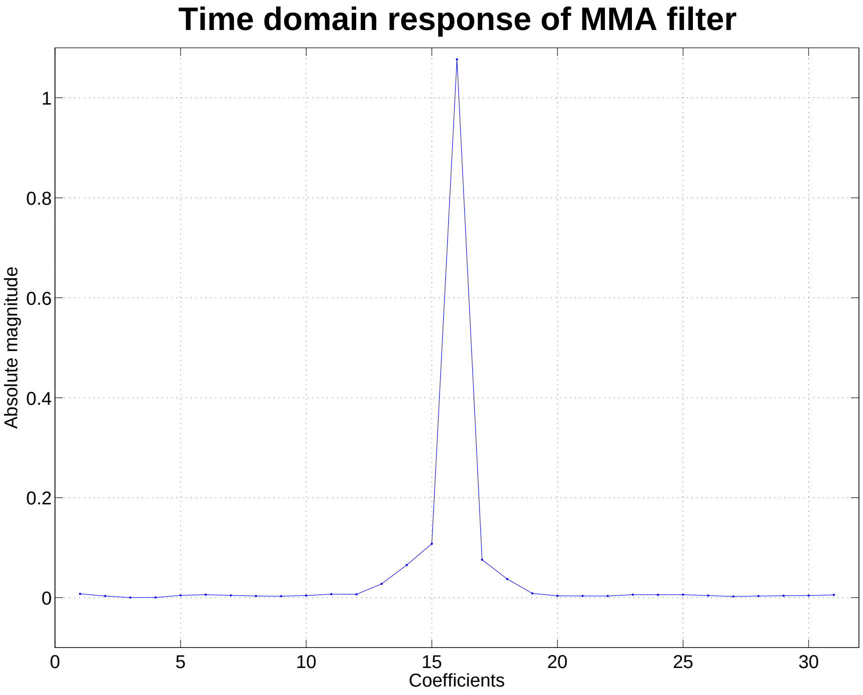
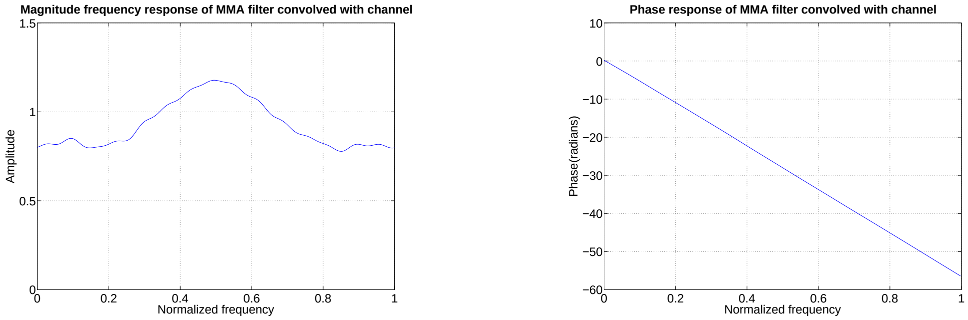


Fig. 3.24: MMA equalizer time domain coefficients for simulated data without PLL and with phase offset present.



(a) Frequency response of the MMA equalizer convolved with channel.

(b) Phase response of the MMA equalizer convolved with channel.

Fig. 3.25: MMA equalizer, convolved with channel, frequency, and phase response for simulated data without PLL and with phase offset present.

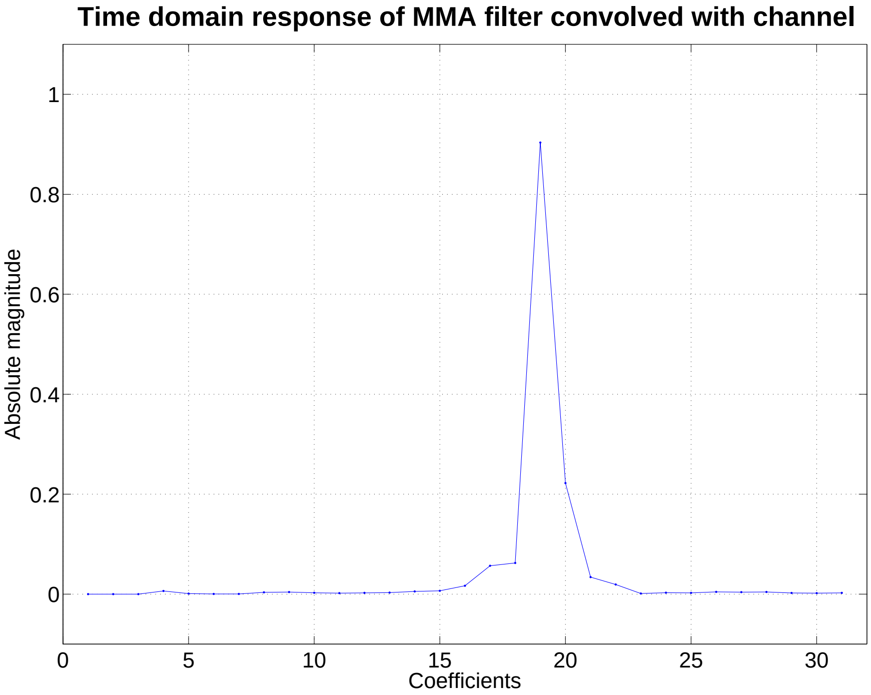


Fig. 3.26: MMA equalizer, convolved with channel, time domain coefficients for simulated data without PLL and with phase offset present.

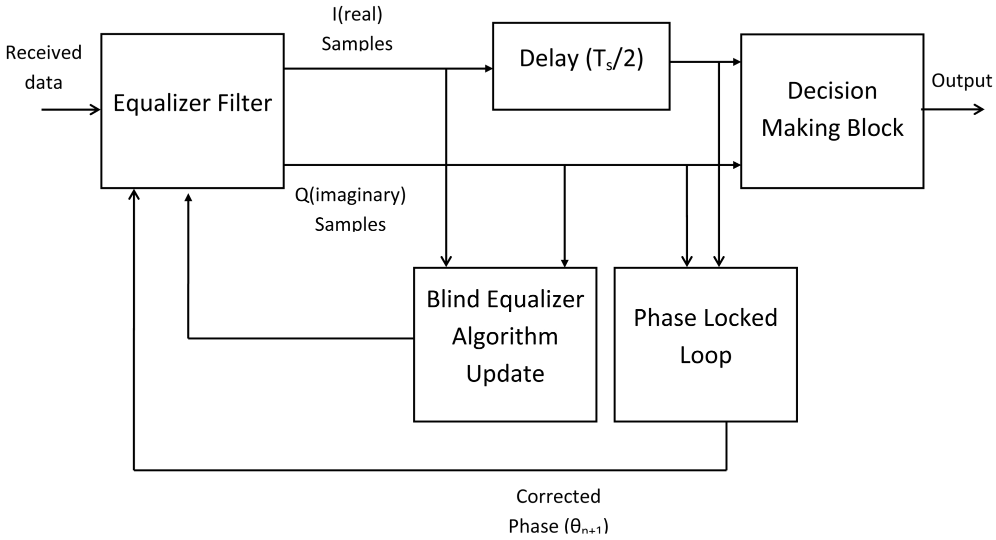
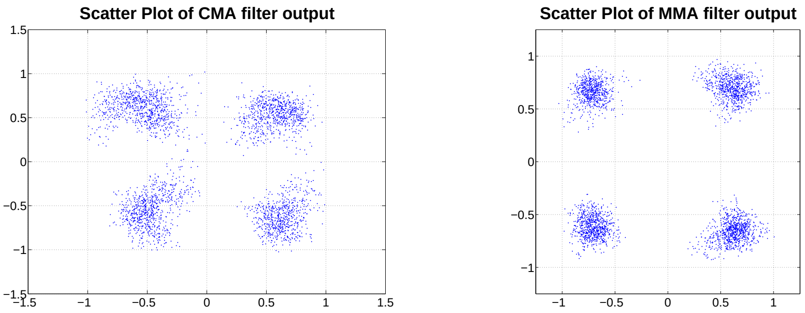
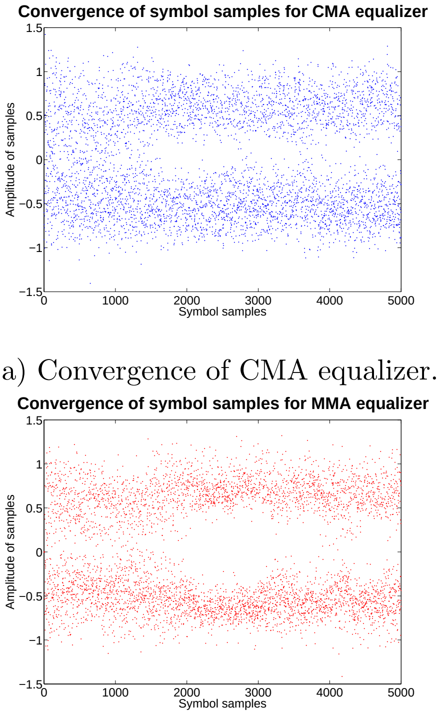


Fig. 3.27: PLL implemented in blind equalization algorithm.



(a) Scatter plot of CMA equalizer. (b) Scatter plot of MMA equalizer.

Fig. 3.28: Comparision of scatter plots obtained by different blind equalizer algorithms with PLL on real DICE satellite data.



(a) Convergence of CMA equalizer.

(b) Convergence of MMA equalizer.

Fig. 3.29: Convergence of symbol samples for CMA and MMA equalizer with PLL on real DICE satellite data.

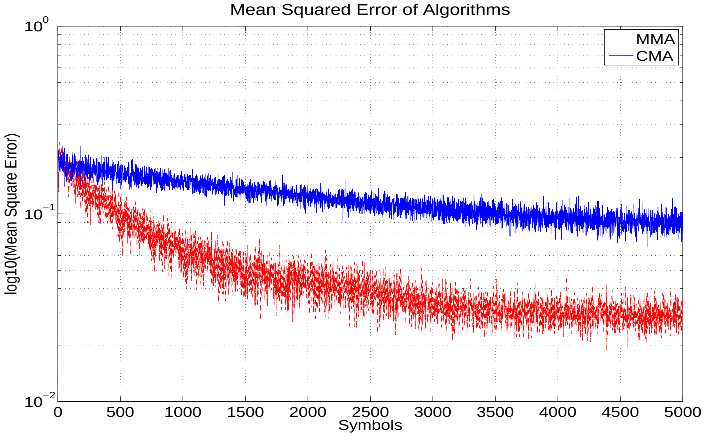
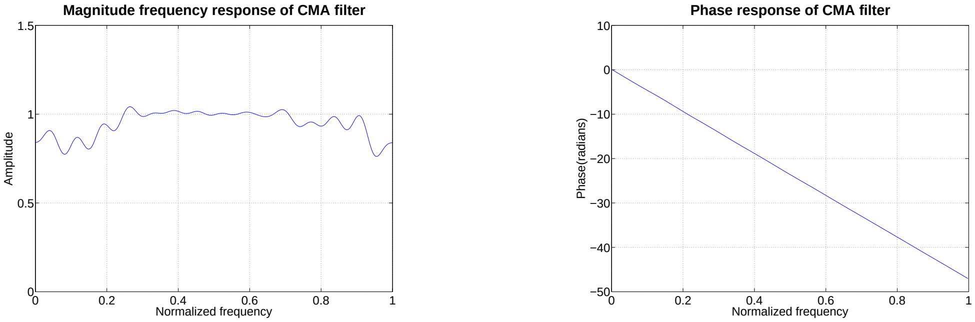


Fig. 3.30: MSE of different blind equalizer algorithms with PLL on real DICE satellite data.



(a) Frequency response of the CMA equalizer.

(b) Phase response of the CMA equalizer.

Fig. 3.31: CMA equalizer with PLL frequency and phase response for real DICE satellite data.

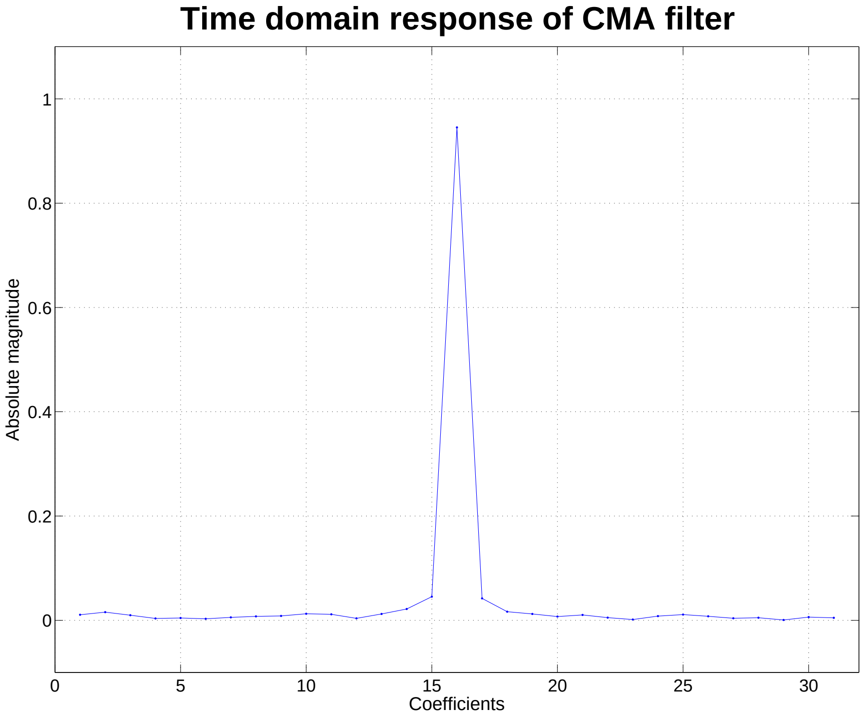
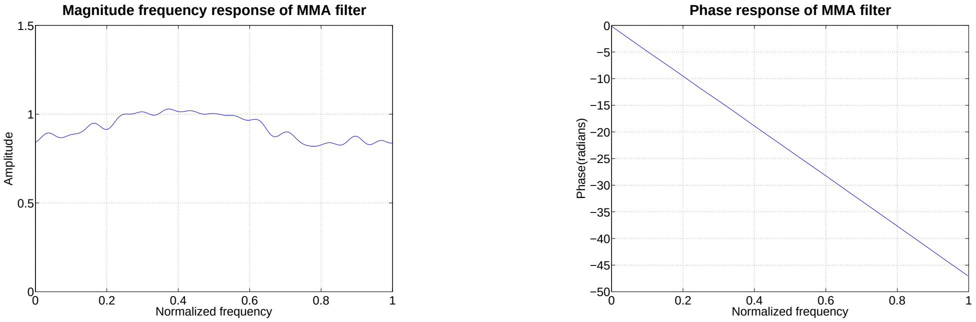


Fig. 3.32: CMA equalizer with PLL time domain coefficients for real DICE satellite data.



(a) Frequency response of the MMA equalizer.

(b) Phase response of the MMA equalizer.

Fig. 3.33: MMA equalizer with PLL frequency and phase response for real DICE satellite data.

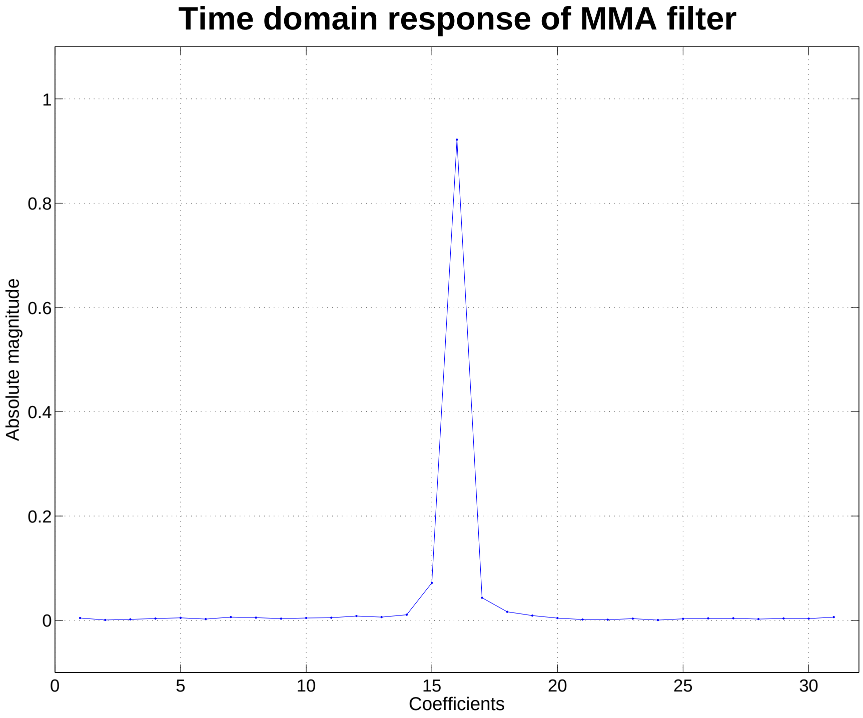
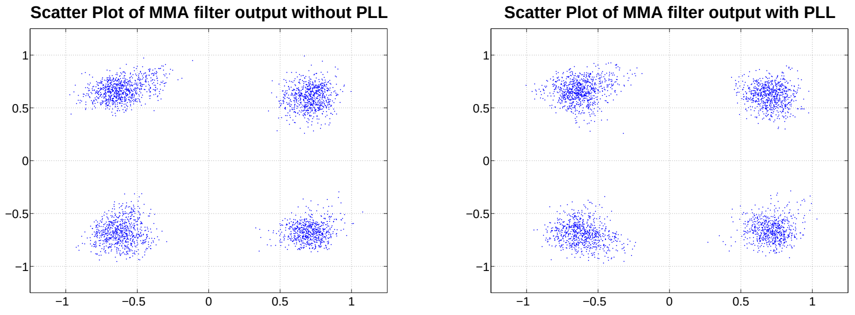
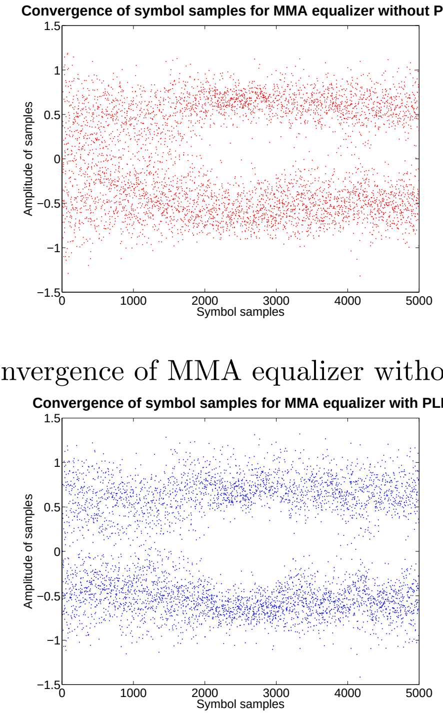


Fig. 3.34: MMA equalizer with PLL time domain coefficients for real DICE satellite data.



(a) Scatter plot of MMA equalizer. (b) Scatter plot of MMA equalizer.

Fig. 3.35: Comparision of scatter plots obtained by MMA blind equalizer algorithm with and without PLL on real DICE satellite data.



(a) Convergence of MMA equalizer without PLL.

(b) Convergence of MMA equalizer with PLL.

Fig. 3.36: Convergence of symbol samples for MMA equalizer with and without PLL on real DICE satellite data.

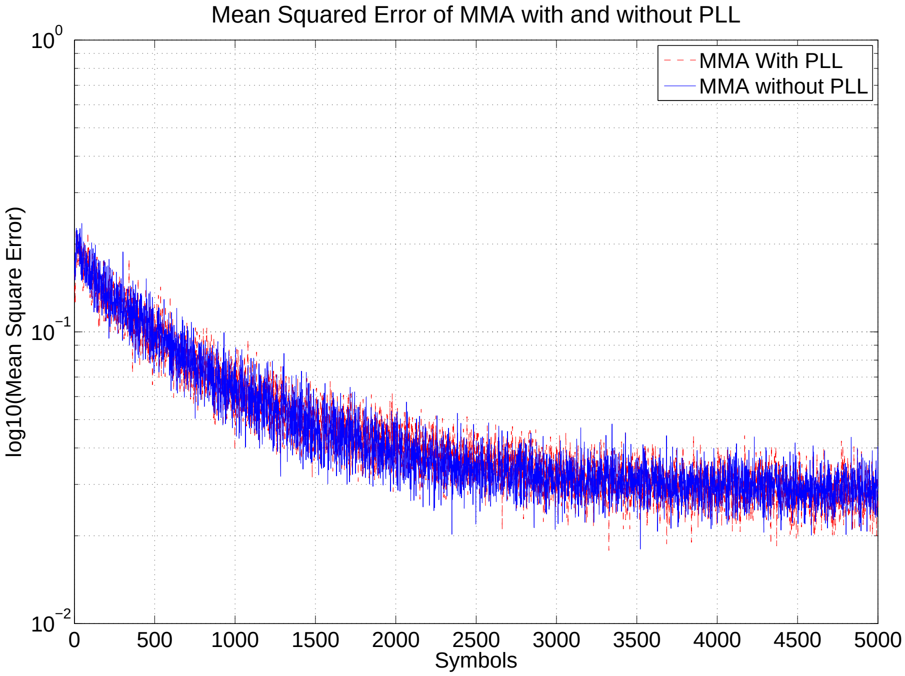
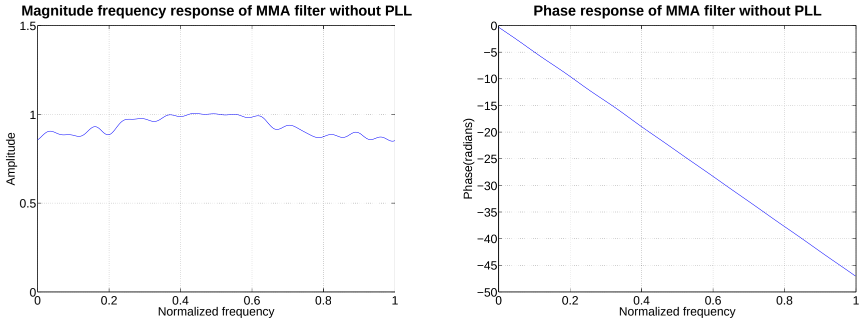


Fig. 3.37: MSE of MMA equalizer algorithm with and without PLL on real DICE satellite data.



(a) Frequency response.

(b) Phase response.

Fig. 3.38: Frequency and phase response of the MMA equalizer without PLL.

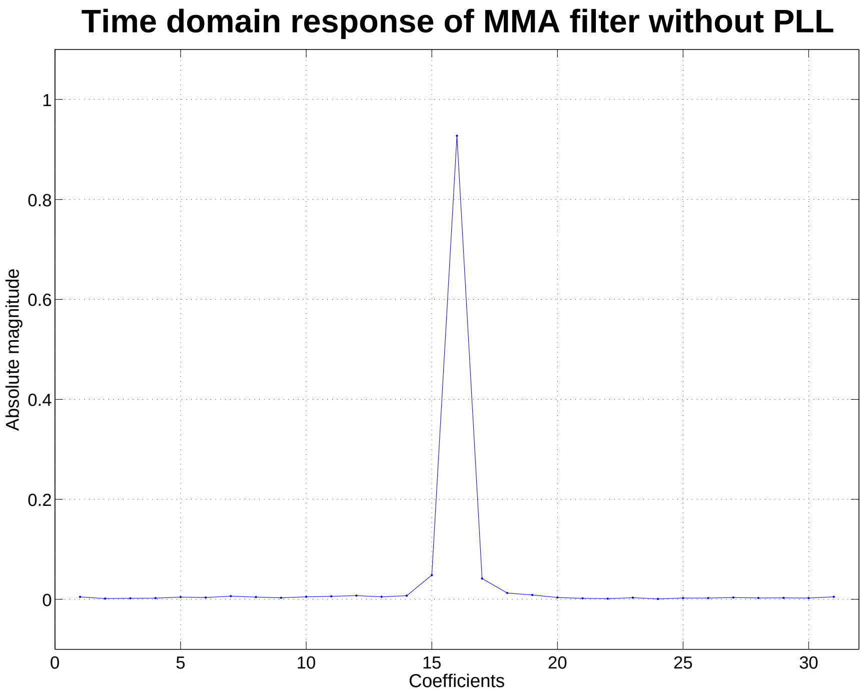
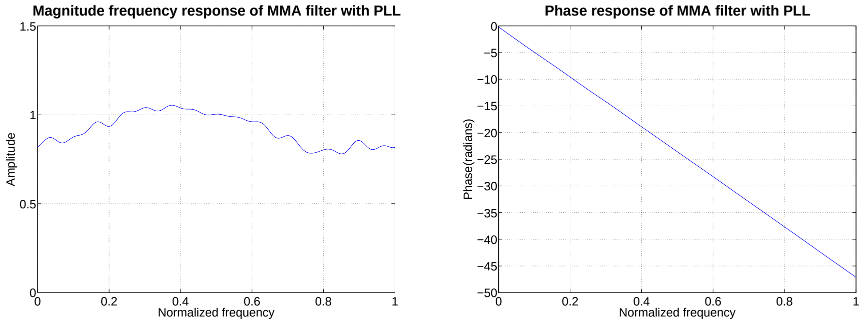


Fig. 3.39: Time domain coefficients of MMA equalizer without PLL.

‘



(a) Frequency response.

(b) Phase response.

Fig. 3.40: Frequency and phase response of the MMA equalizer with PLL.

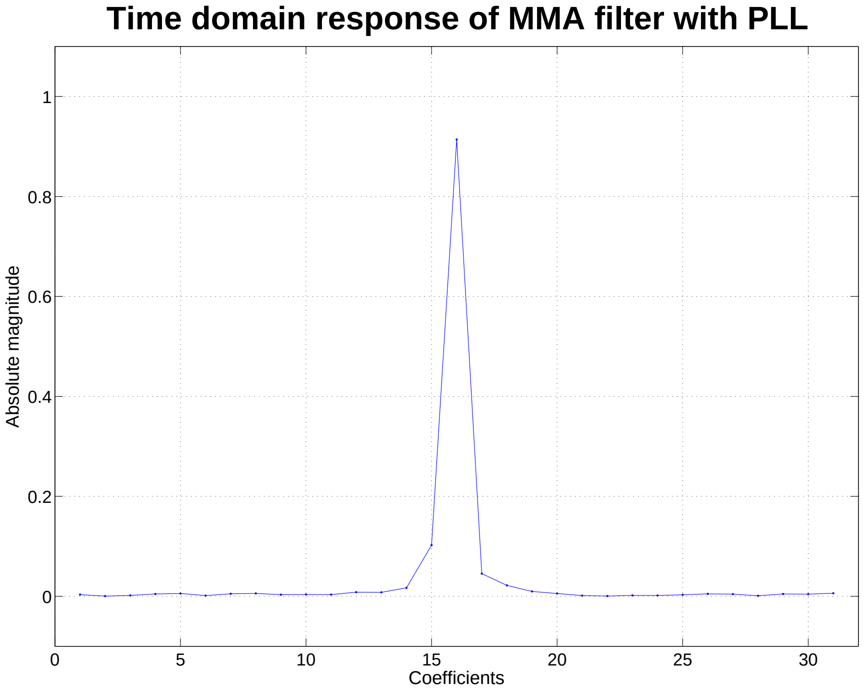


Fig. 3.41: Time domain coefficients of MMA equalizer with PLL.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this project work, blind equalizers were used for demodulation of QPSK and offset-QPSK signals. The aim was to implement algorithms to demodulate the downlink data stream of DICE satellite. The blind algorithms were selected because of no requirement of training data for equalizers to work. The blind algorithms selected are CMA and MMA algorithms. These algorithms utilize the inherent symmetry of the constellation and equalizes the incoming data stream. The algorithms are based on a gradient descent approach. The CMA minimizes the deviation of the absolute value of the symbol values from a constant, which is determined by signal constellation. MMA algorithm separately equalizes the real and imaginary part of the symbols. Here too, the deviation from a constant, determined by signal constellation, is restricted for equalization. Comparison of performances CMA and MMA algorithms, without any phase offset present in the received data signal, reveal that both perform similarly with almost equally low squared error and very close convergence time is required. However in presence of phase offset and without any external PLL present, the CMA algorithm performs very poorly against the good performance of MMA algorithm. MMA algorithm achieves low error witout any PLL present. The CMA equalizer hence does not compensate for the phase offset. An external PLL is required for CMA equalizer for achieving well separated signal constellation. The real DICE satellite data is used, after appropriate processing, to compare performance of CMA and MMA equalizer with an external PLL implemented. Here, both the algorithms show reasonable performance and scatter plots of constellation have well defined grouping of symbols. The MMA algorithm produces a more tighter and clear grouping. Thus, the error for MMA algorithm is lower than that of CMA algorithm, indicating its better performance for demodulation of DICE satellite downlinked data. Then MMA algorithm was implemented with and without external PLL on downlinked DICE satellite data to investigate if PLL was required for demodulation. Here, MMA algorithm performed almost in same manner. The scatter plot of signal constellation were almost similar in both cases. Also, the steady state error level and the speed of convergence was almost identical in both cases. This indicated independence of MMA algorithm performance from additional PLL to compensate for phase offset. Hence, we can use MMA blind equalizer for demodulation of DICE satellite data. It can be used without any external PLL being implemented. The output of this can be fed to decision block for further conversion to bits.

4.2 Future Work

In this project work, blind equalization techniques were identified to be appropriate for demodulation of DICE satellite data. CMA and MMA equalizers were implemented and satisfactory results were obtained. It could be exteded so that initially blind equalizers are used when the eye opening is very small. No training data will be required. Then once the eye is suffciently opened up then the equalization could be switched to normal data aided equalizers like LMS or RLS algorithms. Also, these implementations could be programmed in C language to make them execute faster and demodulation could be done in real time.

References

[1] Space Dynamics Laboratory. (2011) Dynamic Ionosphere Cubesat Experiment. Available: http://www.sdl.usu.edu/programs/dice.

[2] C. R. Johnson, P. Schniter, I. Fijalkow, L. Tong, A. Touzni, H. H. Zeng, M. Green, and J. R. Treichler, *The core of FSE-CMA behavior theory, in Unsupervised Adaptive Filtering, vol. 2: Blind Deconvolution, S. Haykin.* New York, NY: Wiley, 2000.

[3] J. R. Treichler, V. Wolff, and C. R. Johnson Jr., "Observed misconvergence in the constant modulus adaptive algorithm," *1991 Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 663–667, Nov. 1991.

[4] J. Treichler, M. Larimore, and J. Harp, "Practical blind demodulators for high-order QAM signals," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1907–1926, Oct. 1998.

[5] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *Communications, IEEE Transactions*, vol. 28, no. 11, pp. 1867–1875, Nov. 1980.

[6] G. Picchi and G. Prati, "Blind equalization and carrier recovery using a "stop-and-go" decision-directed algorithm," *Communications, IEEE Transactions*, vol. 35, no. 9, pp. 877–887, 1987.

[7] K. N. Oh and Y. O. Chin, "Modified constant modulus algorithm: blind equalization and carrier phase recovery algorithm," in *Seattle, 'Gateway to Globalization', IEEE International Conference on Communications*, vol. 1, pp. 498–502, June 1995.

[8] B. Agee, "The least-squares CMA: A new technique for rapid correction of constant modulus signals," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, vol. 11, pp. 953–956, 1986.

[9] S. Abrar, "Square-contour stop-and-go algorithm for blind channel equalization in QAM data communication systems," in *Engineering, Sciences and Technology, Student Conference*, pp. 73–76, 2004.

[10] J. Treichler and B. Agee, "A new approach to multipath correction of constant modulus signals," *Acoustics, Speech and Signal Processing, IEEE Transactions*, vol. 31, no. 2, pp. 459–472, 1983.

[11] J.-T. Yuan and T.-C. Lin, "Equalization and carrier phase recovery of CMA and MMA in blind adaptive receivers," *Signal Processing, IEEE Transactions*, vol. 58, no. 6, pp. 3206–3217, June 2010.

[12] S. Abrar, A. Roy, and J. Axford, "Sliced multi-modulus blind equalization algorithm," *Electronics and Telecommunications Research Institute Journal*, vol. 27, no. 3, pp. 257–266, June 2005.

[13] S. Abrar and A. Nandi, “Adaptive solution for blind equalization and carrier-phase recovery of square-QAM,” *Signal Processing Letters, IEEE*, vol. 17, no. 9, pp. 791–794, 2010.

[14] M. Niroomand, M. Derakhtian, A. Shahimaeen, and S. Gazor, “Adaptive blind equalization for QAM modulated signals in the presence of frequency offset,” in *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop*, pp. 223–226, 2011.

[15] J. Yang, ”Multimodulus algorithms for blind equalization”. Ph.D. dissertation, University of British Columbia, Vancouver, BC, Canada, Nov. 1997.

[16] R. Michael, *Digital Communications: A Discrete-Time Approach.* Upper Saddle River, NJ: Prentice Hall, 2008.

[17] S. Haykin and M. Micheal, *An Introduction to Analog and Digital Communications.* New York, NY: John Wiley and Sons, 2000.

[18] J. G. Proakis, *Digital Communication.* New York, NY: McGraw-Hill, 2002.

[19] J. R. Treichler, C. R. Johnson Jr., and M. G. Larimore, *Theory and Design of Adaptive FIlters.* Upper Saddle River, NJ: Prentice Hall, 2001.

[20] J. Johnson, R., P. Schniter, T. Endres, J. Behm, D. Brown, and R. Casas, “Blind equalization using the constant modulus criterion: a review,” *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1927–1950, 1998.

[21] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms for signal processing.* Upper Saddle River, NJ: Prentice Hall, 2000.

[22] A. Hjorungnes and D. Gesbert, “Complex-valued matrix differentiation: Techniques and key results,” *Signal Processing, IEEE Transactions*, vol. 55, no. 6, pp. 2740–2746, 2007.

[23] S. Abrar and A. Nandi, “Blind equalization of square-QAM signals: a multimodulus approach,” *Communications, IEEE Transactions*, vol. 58, no. 6, pp. 1674–1685, June 2010.

[24] J. Yang, J.-J. Werner, and G. Dumont, “The multimodulus blind equalization and its generalized algorithms,” *Selected Areas in Communications, IEEE Journal*, vol. 20, no. 5, pp. 997–1015, 2002.