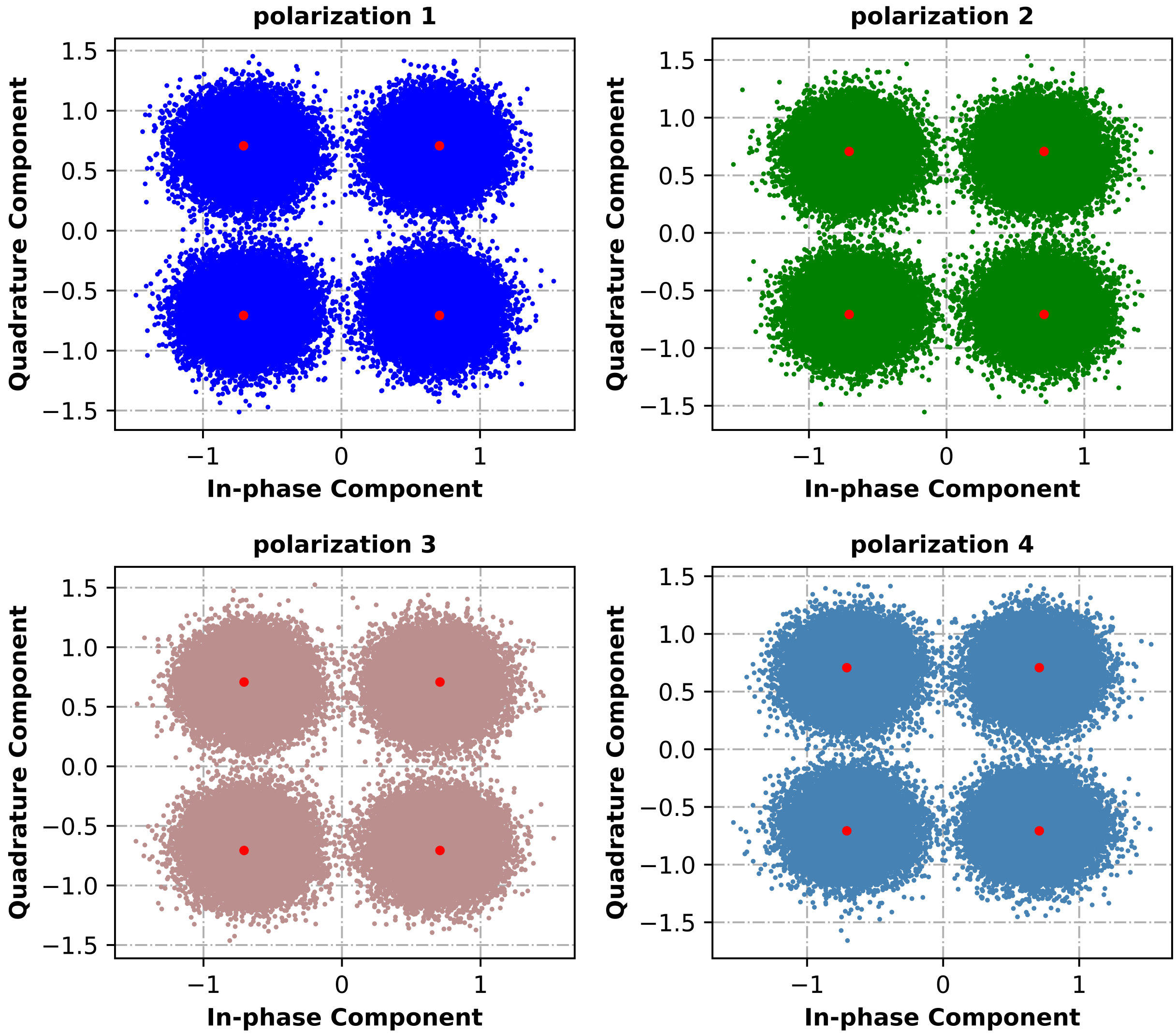


# Finite Precision Digital Signal Processing for Space Division Multiplexing

Master's thesis in Embedded Electronic System Design

Yingshi Jin

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's thesis 2018

# Finite Precision Digital Signal Processing for Space Division Multiplexing

Yingshi Jin

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Finite Precision Digital Signal Processing for Space Division Multiplexing
Yingshi Jin

© Yingshi Jin, 2018.

Supervisor: Jochen Schröder, Microtechnology and Nanoscience MC2
Examiner: Per Larsson-Edefors, Computer Science and Engineering CSE

Master's Thesis 2018
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: 4x4 CMA equalizer constellation diagram floating point

Typeset in LaTeX
Gothenburg, Sweden 2018

Finite Precision Digital Signal Processing for Space Division Multiplexing
Yingshi Jin
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

# Abstract

Optical fibers are widely used for data transmission in modern data networks, but the development of fiber systems is approaching some fundamental limits in terms of data capacity. Space division multiplexing (SDM) can help to overcome some of these fundamental limits. The most scalable SDM approach is so-called mode-division multiplexing (MDM) in multi-mode fiber (MMF). In MDM, each mode in principle will propagate independently and can be easily separated at the receiver. However in a real system, small perturbations of the fiber, such as manufacturing imperfections, bends or vibrations will cause mixing between the modes. Therefore at the receiver, an adaptive equalizer is needed to separate the orthogonal states and recover the signals.

So far, most demonstrations have relied heavily on offline digital signal processing using *floating point* representation. To achieve the required high data rates, real systems would need to employ purpose designed application-specific integrated circuits (ASICs) with finite precision. The goal of this project is to investigate how limited fixed point precision affects the performance of constant modulus algorithm (CMA) equalizer for SDM data transmission.

In this report, we will introduce how to design and implement adaptive CMA equalizers, perform system simulations initially using *floating point* and then reducing its precision to *fixed point* to evaluate performance penalty. Evaluations will be done with 2x2 mode with two polarizations and 4x4 mode with four polarizations.

Keywords: CMA equalizer, adaptive FIR filter, digital signal processing (DSP), space division multiplexing (SDM), multiple-input multiple-output (MIMO), fixed point finite precision.

# Acknowledgements

Firstly, I would like to thank Dr. Jochen Schröder in Microtechnology and Nanoscience MC2 for being my supervisor and Prof. Per Larsson-Edefors in Computer Science and Engineering CSE for being my examiner. Thank you for the great coaching and guidance during this project, which helped me to have better understanding on the digital signal processing in the optical communications.

Secondly, I would like to thank Mikael Mazur in Microtechnology and Nanoscience MC2 and Christoffer Fougstedt in Computer Science and Engineering CSE, thank you for all the valuable feedback and inputs during this project.

Finally, I would like to thank my family and my friends for encouraging and supporting me to work on this project.

Yingshi Jin, Gothenburg, June, 2018

# Contents

# List of Figures

# 1

# Introduction

The internet, the global engine for innovation, economic growth and social progress is powered by optical fibers, making optical fiber communication the backbone of modern data networks. In data centers, between cities and continents and from the cell-phone tower to the central office, any data connection will most certainly have traversed an optical fiber on the way to the user.

However, while the global demand for data continues to grow at a staggering rate of 40% a year [1], the optical communication systems are approaching some of the fundamental limits for the data capacity of the fiber. Space division multiplexing (SDM) is a new approach that uses the spatial dimension as a new data-carrying degree of freedom, and promises to overcome some of these fundamental limits [2].

## 1.1 Context

While different avenues are being pursued, arguably the most scalable SDM approach is the so-called mode-division multiplexing (MDM) in multi-mode fiber (MMF) [2]. In contrast to a single mode fiber, which has essentially two spatial eigen states (X and Y polarizations), a multi-mode fiber has many orthogonal spatial eigen states. In principle, each of these modes will propagate independently of all the other modes and thus can be easily separated at the end of the fiber. However, in a real long-haul system, small perturbations of the fiber, such as fabrication imperfections, bends or vibrations will cause mixing between the modes. Therefore, at the receiver we need to employ digital signal processing (DSP) to separate the orthogonal states. This is achieved using multiple-input multiple-output (MIMO) techniques, similar to what is used in wireless multi-antenna systems [3, 4].

However, in contrast to wireless systems, the DSP needs to operate at symbol rates above 10 Gbaud [2, 4, 5, 6, 7]. So far, most demonstrations have relied heavily on offline DSP using *floating point* math [8]. To achieve the required data rates, real systems would need to employ purpose designed application-specific integrated circuits (ASICs) with finite precision. It is unclear if and how DSP algorithms scale in such a scenario.

## 1.2 Goals and Challenges

The goal of this project is to investigate how limited *fixed point* precision affects the performance of constant modulus algorithm (CMA) equalizer for SDM transmission. We will develop an implementation of one specific CMA equalizer algorithm in Python [9], that can be tuned to different precisions and use this to process simulated MDM data to evaluate performance as a function of signal-to-noise ratio (SNR), number of spatial dimensions and precision.

## 1.3 Related Work

One of the challenges in multi-mode systems is that the required processing scales quadratically with mode count. A 10-mode transmission system would therefore require $10 \times 10$ MIMO processing, which becomes difficult to implement. In addition, more modes broaden the timing spread of different components of signal propagation delay to each other, which creates a need for more memory and more taps to equalize. So far, higher-order MIMO is only a research topic in optical communication, there is no commercial implementation more than $2 \times 2$ [2, 6, 8, 9]. Most research approaches were based on offline DSP on more modes with *floating point* precision in experiments [8]. But in order to implement it at above 10 Gbaud symbol rates with ASICs, *fixed point* precision is needed.

## 1.4 Thesis Outline

The report begins with introduction of the theory of DSP-based coherent transceiver and adaptive CMA equalizer in Chapter 2, followed by the methodology of DSP in transmitter and receiver in Chapter 3, with focus on the adaptive CMA equalizer. The results are presented in Chapter 4, the discussions and conclusions drawn from the results are discussed in Chapter 5, and recommended parameters are listed in Appendix.

# 2
# Theory

The big capacity increase of new generation fiber links greatly benefit from the fast development of DSP. Working in the digital domain with DSP offers much more flexibility than working in analog domain, and provides the ability to process data with many mathematical theories and techniques. Thus, DSP has been widely used in the whole range of optical communication systems, from long-distance backbone networks to shorter dynamic access networks [10].

At the receiver, DSP is used for signal equalization, which can recover signals out of the noisy, dispersed and nonlinearly distorted received signals [11]. DSP equalization includes both static and dynamic equalization. Static equalization compensates static channel impairments using fixed-tap filter, and less computational power. Dynamic equalization applies filter with adaptive algorithms, such as CMA, able to follow signal changes and recover signals dynamically in real-time [11]. DSP equalization also synchronizes the transmitter and receiver oscillators both in optical and electrical domains, to cancel any difference between them.

This chapter is organized as follows. Section 2.1 introduces the basics of DSP-based coherent transceiver in optical communications, which include transmitter, channel and receiver. Section 2.2 covers adaptive CMA equalizer. Section 2.4 compares the "infinite" precision in *floating point* and finite precision in *fixed point*. Section 2.5 contains some common ways to measure the performance in optical communication.

## 2.1 DSP-Based Coherent Transceiver

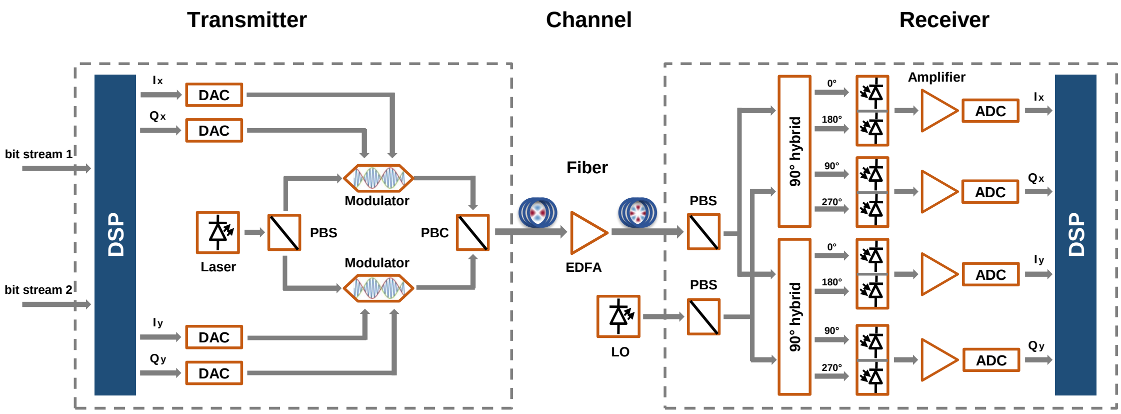


**Figure 2.1:** DSP-Based Coherent Transceiver

Fig. 2.1 depicts the general setup of the DSP-based coherent link, which is built up in three main components. The transmitter modulates the input digital data streams into the analog signals and transmit these through the optical channel with an optical carrier. The receiver demodulates the received optical analog signals and outputs digital signals. The data throughput is limited by the modulation format [12] and sampling clock frequency of the ADC [13]. High throughput can be achieved by pipelining in the architecture [14] and parallel processing algorithms in the equalizer at the receiver [13].

### 2.1.1 Transmitter

The DSP block at the transmitter side includes functions for encoding, modulation, precompensation of linear and nonlinear channel impairments and pulse shaping filtering. Four digital-to-analog converters (DACs) convert signals from digital to analog, each for in-phase and quadrature field components of bit stream 1 and 2 polarizations. One laser is split into two parts by the polarization beam splitter (PBS) and used for two IQ modulators (IQMs), which carry the output signals from DAC, combine them with the polarization beam combiner (PBC), and then transmit them through the fiber.

At the transmitter, oversampling is used to sample a signal at a sampling frequency higher than the Nyquist rate[1]. Oversampling helps to improve resolution, reduces noise and avoids aliasing and phase distortion when transmitting data through channel. At the receiver, the received signals will be filtered and downsampled to the desired sampling frequency.

[1]In signal processing, a bandwidth-limited signal can be reconstructed if sampled at the Nyquist rate or above it. The Nyquist rate is twice of the signal frequency.

In addition to the oversampling, pulse shaping is used to change the waveform or limit the bandwidth of the transmitted signals to minimize the intersymbol interference (ISI) in channel. When the linear modulation is applied to transmit symbols through channel continuously, the impulse response of the channel will spread the transmitted symbols in the time domain. So, the previously transmitted symbols have an impact on the currently received symbols, which cause ISI. Normally pulse shaping filters are used to control the ISI, such as sinc shaped filter, raised-cosine (RC) filter and Gaussian filter.

In communications, the Nyquist ISI criterion defines some conditions. When the overall response of filter transmit, channel response and filter receive all satisfy the Nyquist ISI criteria, symbols can be transmitted over a channel with flat response within a limited frequency band without ISI. If a root-raised-cosine (RRC) filter is used in both transmitter and receiver, and the channel response can also be estimated, then the overall response is Raised-cosine filter (RC) filter. The RC filter has filter response satisfying this Nyquist ISI criteria, so it is often used in transmitter and receiver in optical communication systems to minimize ISI. Split RC filter in both transmitter and receiver will maximize the signal-to-noise (SNR) ratio [15]. So, normally two RRC filters are used in optical systems, one used at the transmitter for spectral shaping, and another one used at the receiver as a matched filter (MF) [16].

One important parameter to characterize RRC filter is the roll-off factor $\beta$, which shows the steepness of a transmission function with frequency [17]. To have pulse shape close to the ideal Nyquist pulse shape, the roll-off factor $\beta$ needs to be as small as possible, but it requires more filter taps for accurate shaping [18].

### 2.1.2 Channel

The optical channel consists of fiber spans, which include optical fiber and optical amplifier. The typical span length is about 50-150 km with high-performance amplifiers [19]. The Erbium doped fiber amplifier (EDFA) is often used as optical amplifier. When signals propagate through systems, they will be distorted by linear and nonlinear channel impairments and polluted by different kinds of noise [20]. EDFA can help to compensate fiber losses, suppress noise, and ensure signals have high enough power to be detected at the receiver.

#### 2.1.2.1 Noise Sources

The sources of noise include shot noise, thermal noise, amplified spontaneous emission (ASE) noise from fiber amplifiers, residual beating of the local oscillator (LO) with relative intensity noise (LO-RIN), quantization noise from ADC etc. Short distance unamplified systems mainly suffer from shot noise, while long haul systems with inline amplifiers are mainly impaired by the ASE noise [13].

#### 2.1.2.2 Channel Impairments and ISI

**Channel impairments** have linear effects and nonlinear effects. The linear effects, such as chromatic dispersion (CD) and polarization mode dispersion (PMD). The nonlinear effects, such as self-phase modulation (SPM) and cross phase modulation (XPM) [11]. In addition, the defects in the fabrication process, which increase the material absorption, are still the main factor of the fiber power losses [13].

**ISI** is generated when transmitting signals at high modulation rate through a band-limited channel, where one symbol interferes with subsequent symbols. An increased modulation rate will increase the signal bandwidth, and when the signal's bandwidth is larger than the channel bandwidth, the signal will be distorted by channel. Such kind of distortion is called ISI. Channel impairments, such as rotation, polarization-mode dispersion (PMD) and differential group delay (DGD), are some sources of ISI [21].

*Rotation* means x- and y-polarizations are rotated at angle $\theta$ when signals are propagating through channel. Even after rotation, the x- and y-polarizations are still orthogonal.

*PMD* is a linear distortion that affects the performance of optical systems. In single-mode fiber (SMF) communication, PMD will cause DGD between two polarizations, which typically but not necessarily causes pulse spreading less than one symbol interval [22].

*DGD* is the 1st order PMD, which means when signals are transmitted through the channel, various polarization might have different time delay, which changes the shape of signals and result in signal distortion. Group delay will cause ISI when demodulating the digital signals from an analog carrier, which needs to be compensated with adaptive filters with large number of taps and cause high computational load [13].

### 2.1.3 Receiver

At the receiver, the optical signal is mapped to four electrical signals with two PBSs and a pair of 90° optical hybrids, which correspond to the in-phase and quadrature field components of signals in x- and y-polarizations [13]. Four balanced photodiodes are used to detect the output signals from 90° optical hybrids and generate the in-phase and quadrature of x- and y-polarizations based on the real and imaginary components of the electric fields of the x- and y-polarizations and LO. Then, signals are amplified with trans-impedance amplifiers (TIAs). After that, four analog-to-digital converters (ADCs) convert the electrical analog signals into digital signals and transfer them into DSP [11, 13].

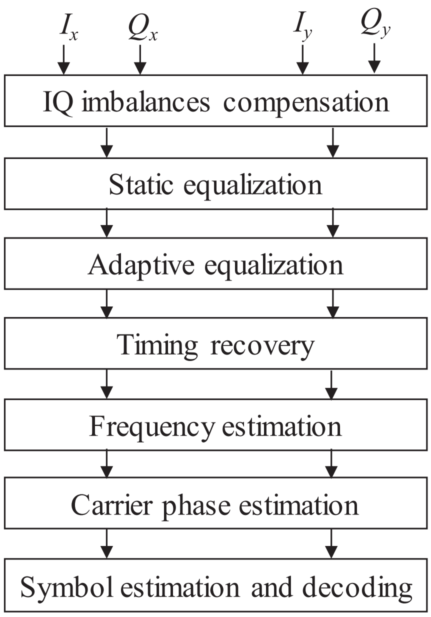


**Figure 2.2:** Typical DSP chain in fiber receiver [11]

Fig. 2.2 shows typical DSP chain in fiber receiver to demodulate digitized signal [11], which includes IQ imbalance compensation, static equalization, adaptive equalization, timing recovery, frequency estimation, carrier phase estimation, symbol estimation and decoding. In this project, we focus on the adaptive equalization with CMA.

## 2.2 Adaptive CMA Equalizer

When signals are propagating through optical channel, the signals will be contaminated by noise and different kinds of channel impairments. The chromatic dispersion (CD) can be compensated in digital domain with intradyne detection method [23]. For optical systems running at 40 Gbit/s, the more critical channel impairment is dynamic polarization mode dispersion (PMD), which requires an adaptive equalizer, such as MIMO CMA equalizer. The CMA attempts to minimize error *en*, which is the difference of the desired reference signals and the sample signals [24, 25, 26]. The constant amplitude makes it possible to perform polarization demultiplexing even for higher order modulation formats [27].

A simplified block diagram of an adaptive CMA equalizer is presented in Fig. 2.3. Before applying CMA equalizer to equalize signals, we must first apply it to some sample signals to train the adaptive filter to get the optimal filter taps $wn$. Both CMA training block and CMA filter block use same value for filter taps, which is defined by user as a variable, can be one tap or multi taps.

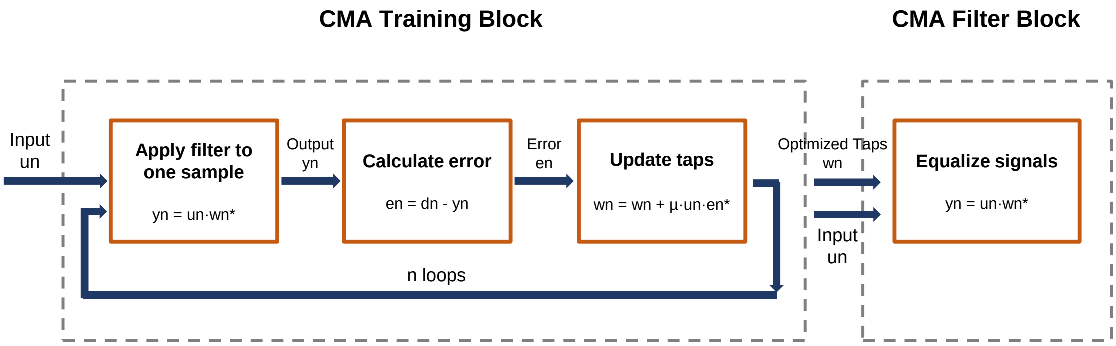


**Figure 2.3:** Adaptive CMA equalizer

### 2.2.1 CMA Training Block

The CMA equalizer first invokes the training block: here we apply the filter to one sample, calculate the error $en$ based on the difference between the desired reference $d$ and output $y$. Then we use $en$ to update filter taps $wn$.

For 2x2 mode CMA equalizer with filter inputs $u_x$, $u_y$, and filter taps:

$$w_x = \begin{bmatrix} w_{xx} & w_{xy} \end{bmatrix}, \tag{2.1}$$

$$w_y = \begin{bmatrix} w_{yx} & w_{yy} \end{bmatrix}, \tag{2.2}$$

first calculate the filter outputs:

$$y_x = [w_{xx}^* u_x + w_{xy}^* u_y], \tag{2.3}$$

$$y_y = [w_{yx}^* u_x + w_{yy}^* u_y]. \tag{2.4}$$

In equations 2.1 and 2.2, for a filter with one tap, filter taps $w_{xx}$, $w_{xy}$, $w_{yx}$ and $w_{yy}$ are just one number representing one filter tap; for a filter with N number of taps, each of them representing length N of filter taps. In equations 2.3 and 2.4, superscript $(\cdot)^*$ denotes the complex conjugate operation. For the filter with N number of taps, one output is calculated with N number of inputs.

Then, derive the filter error function:

$$en = (d\_energy - y\_energy) * y. \tag{2.5}$$

Finally, conclude with the filter taps update equations:

$$w(n+1)_x = [wn_x + \mu \cdot en_x^* \cdot un_x], \tag{2.6}$$

$$w(n+1)_y = [wn_y + \mu \cdot en_y^* \cdot un_y], \tag{2.7}$$

where $\mu$ is the filter step size.

### 2.2.2 CMA Filter Block

After training, the CMA equalizer is used to equalize all signals with the optimal filter taps $w$, which are the output from the filter training block.

$$y_x = [w_{xx}^* u_x + w_{xy}^* u_y], \tag{2.8}$$

$$y_y = [w_{yx}^* u_x + w_{yy}^* u_y], \tag{2.9}$$

where $w_{xx}, w_{xy}, w_{yx}, w_{yy}$ are the optimal filter taps after training.

2x2 mode CMA equalizer has 2 inputs and 2 outputs, while 4x4 mode CMA equalizer has 4 inputs and 4 outputs, which are the multiple-input multiple-output (MIMO) with butterfly-structured filter taps.

## 2.3 Signals Correlation

The CMA equalizer is used to equalize the received signals to the CMA ring, while the correlation is used to find polarization of the equalized signals and align the equalized signals with the transmitted symbols. As shown in Fig. 2.4, cross-correlation moves one signal through another to check the degree of similarity of two signals. When two signals are aligned, the peak of correlation is in the center of data array. According to (Savory 2017), the adaptive equalizer is not limited with respect to its outputs, so the equalizer might converge both dimensions of the received signals to the same polarization, which is known as the polarization singularity problem [9].

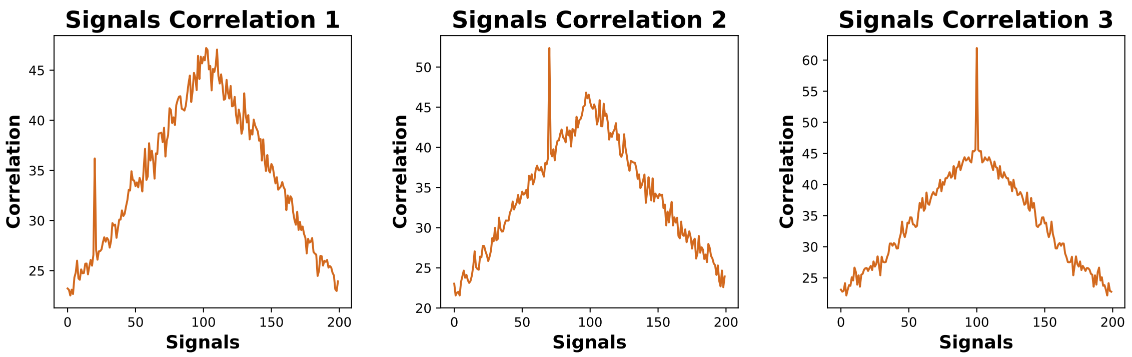


**Figure 2.4:** Signals Correlation

## 2.4 Floating Point and Finite Precision

The conversion of an analog signal into digital format has three steps [28]:

- **Sampling.** Take samples at evenly spaced time intervals.
- **Quantization.** Map continuous-amplitude samples into a finite number of amplitude levels.

- **Coding.** Encode quantized (finite precision) samples into digital codewords.

In quantization, in order to keep acceptable performance, we should select the finite set of amplitudes with small enough spacing between adjacent amplitude levels. The minimum-to-maximum (full-scale) amplitude range of the input signal is split into $M$ partition or threshold intervals. The mapping of input signals in the partition interval into an amplitude is called quantization. The interval between adjacent partition levels is called step size. Dividing the full-scale range into same-length partition intervals is called uniform quantization. Normally, the quantization level count $M$ is selected as power of 2.

In coding, the input signal is mapped to the closest approximation values, so it can be represented by an m-bit codeword. Coding means assigning binary codewords to the finite number of quantization levels.

Quantization is a noninvertible process, original analog signal cannot be exactly recovered with the quantized signal. The quantization process introduces two kinds of errors:

- **Quantization noise.** Caused by applying approximation algorithm (*round, ceiling, floor*) over the range of output levels, which can be minimized by selecting a small enough quantization step size.
- **Saturation (peak clipping).** When the input signal is bigger than the maximum level, it will result in overflow. This can be avoided by selecting the appropriate quantization range to match to the input signal amplitude range.

The conversion of a signal from "infinite" precision in *floating point* to finite precision in *fixed point* can be done in similar way.

## 2.5 Performance Metrics

Below are some commonly used methods to define transmission quality and receiver sensitivity in optical transmission systems:

### 2.5.1 BER, SER and SNR

In optical transmission systems, the signal-to-noise ratio (SNR) of received signals, the bit-error rate (BER), and the symbol-error rate (SER) at receiver output are the most commonly used parameters to define transmission quality. The SNR defines the difference of signal and noise levels of samples. BER and SER are related to SNR, which define the probability of mistaken "digital signal space 0" as "digital signal mark 1" ($0 \rightarrow 1$) and vice versa ($1 \rightarrow 0$) [29], as shown in Fig. 2.5.

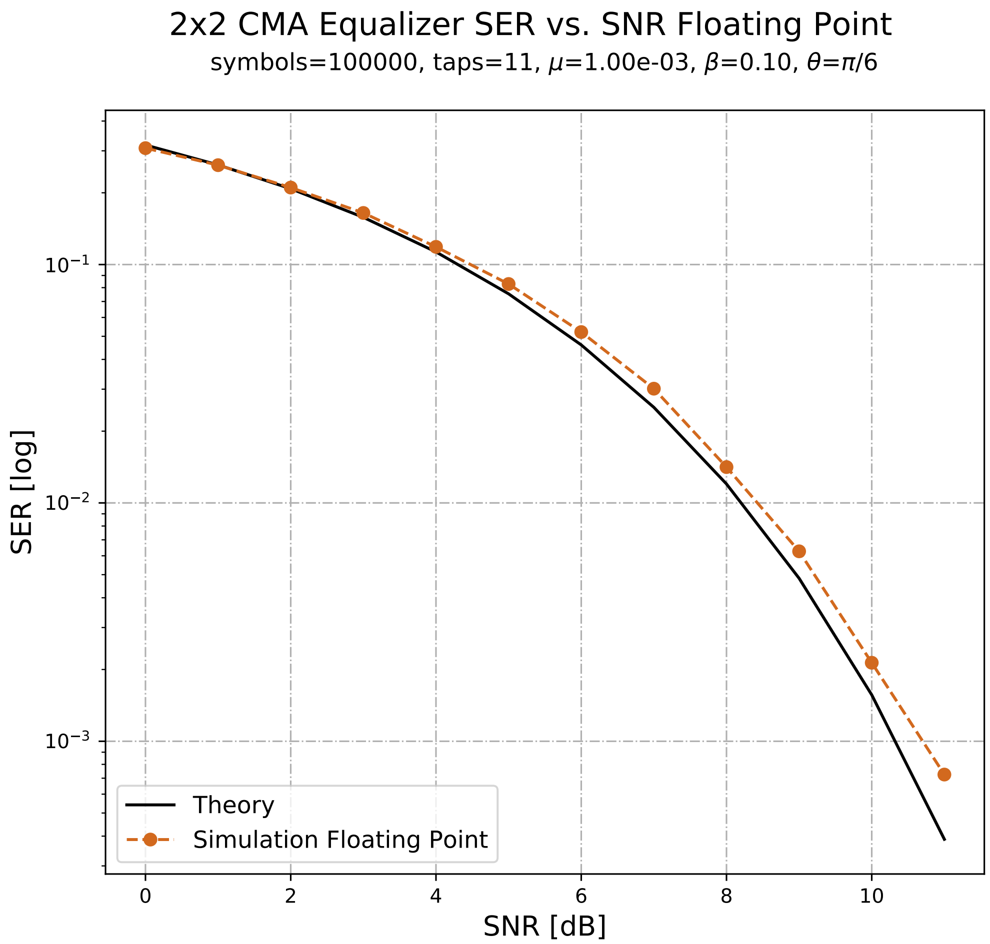


**Figure 2.5:** 2x2 CMA equalizer performance curve SER vs. SNR

$$\text{BER} = \frac{\text{received} \quad \text{bit} \quad \text{errors}}{\text{total} \quad \text{transmitted} \quad \text{bits}}, \tag{2.10}$$

$$\text{SER} = \frac{\text{received} \quad \text{symbol} \quad \text{errors}}{\text{total} \quad \text{transmitted} \quad \text{symbols}}, \tag{2.11}$$

$$\text{SNR} = \frac{P_{signal}}{P_{noise}}, \tag{2.12}$$

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_{signal}}{P_{noise}} \right). \tag{2.13}$$

In equations 2.12 and 2.13, $P_{signal}$ is the power of signal and $P_{noise}$ is the power of noise. If measure in amplitude, equation 2.13 is represented in squared to be proportional to power.

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left[ \left( \frac{A_{signal}}{A_{noise}} \right)^2 \right]. \tag{2.14}$$

### 2.5.2 EVM

The error vector magnitude (EVM) is used to measure the quality of the digital transmitter or receiver. Ideally, the signals sent by transmitter or received by receiver should be all constellation points and locating at the ideal locations. But normally signals deviate from the ideal locations due to different kinds of impairments or imperfections in the process of material fabrication or during signal transmission. EVM, as depicted in Fig. 2.6, shows how far signals deviate from the ideal locations. Since EVM is independent of the noise distribution, it provides more flexibility than SNR [30, 31]. EVM can be calculated in dB or in percentage %.

$$\mathrm{EVM_{dB}} = 10\log_{10}\Big(\frac{P_{error}}{P_{reference}}\Big), \tag{2.15}$$

$$\mathrm{EVM_{\%}} = \sqrt{\frac{P_{error}}{P_{reference}}} \times 100\%. \tag{2.16}$$

In equations 2.15 and 2.16, $P_{error}$ is the power of error and $P_{reference}$ is the power of reference signal. Equation 2.16 uses root mean square (RMS).

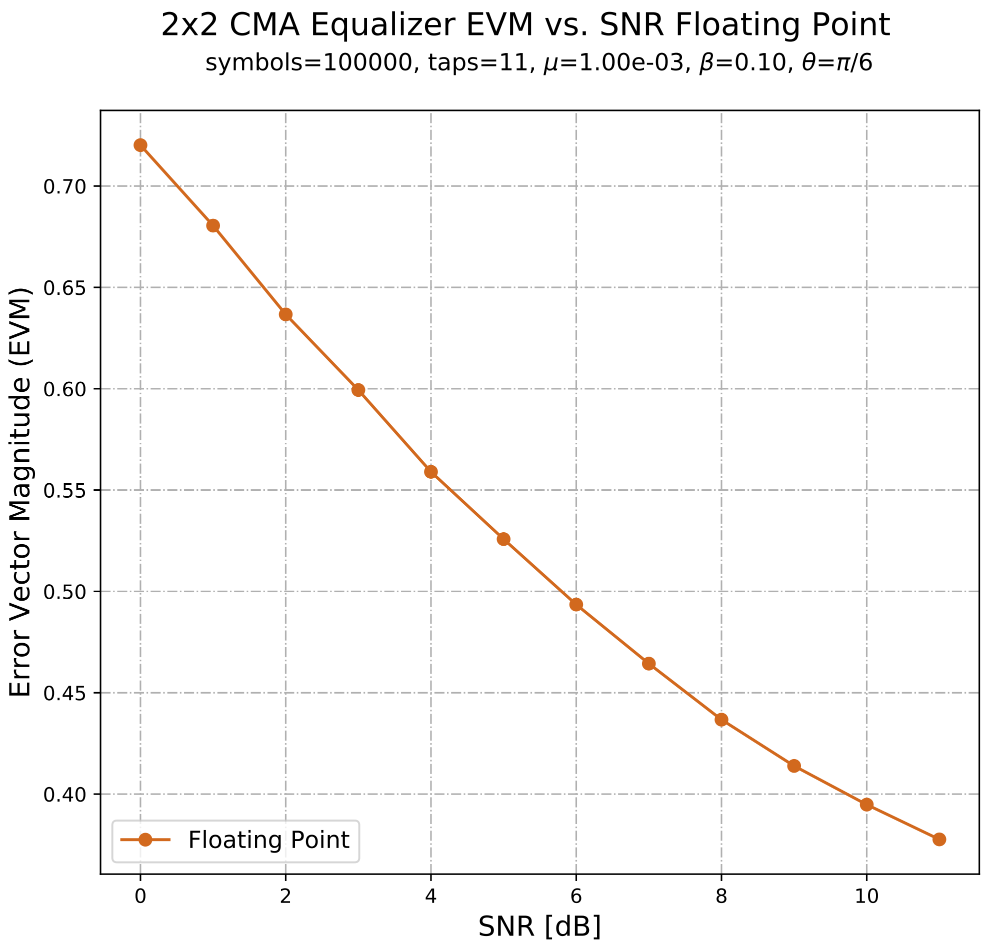


**Figure 2.6:** 2x2 CMA equalizer performance curve EVM vs. SNR

# 3
# Methods

The approach of this project is to perform offline processing using simulated data on a desktop computer. We first test in floating point, and then reduce precision of equalizer with different quantization methods and quantization bits to evaluate the impact to the overall performance.

Detailed tasks include:

- Implement data simulation in the transmitter, which includes generate bits and map to symbols.
- Implement data simulation in the channel, which includes generate noise and add noise to symbols, oversampling and pulse shaping, simulate polarization mode dispersion (PMD), differential group delay (DGD) and polarization rotation.
- Implement data simulation in the receiver. Design MIMO CMA equalizer with adaptive FIR filter.
- Perform basic simulations of 2x2 mode transmission in fiber and test the equalizer in *floating point* first and then reduce precision by quantization to *fixed point*.
- Test with different quantization methods, which include *ceiling, floor and round*.
- Test with different quantization bits, in range of 3-14.
- Increase the mode count to 4x4 and test in *floating point* and *fixed point* respectively.

Fig. 3.1 represents the data simulation for optical communication, which includes DSP in transmitter, data simulation in channel and DSP in receiver. All data simulation was implemented with Python programming language in PyCharm 2018.1.1 (Edu) [32].

DSP in transmitter is covered in section 3.1, DSP simulation for channel impairments are introduced in section 3.2, DSP in receiver is covered in section 3.3, which focus on CMA equalizer, signals correlation and symbol decision.

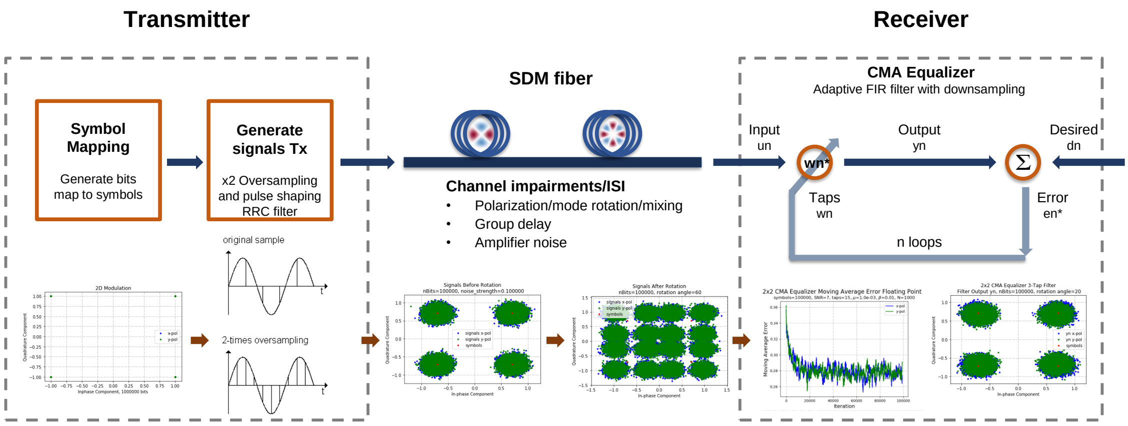


**Figure 3.1:** Data Simulation for Optical Communication

## 3.1 DSP in Transmitter

In the transmitter, a random generator was used to generate data streams in bits. Then symbols were generated based on bits, and moved to symbol locations by symbol mapping with Gray coding. Gray coding means two adjacent symbols only differ by one bit, to minimize the probability of error in symbol mapping and symbol decision [33]. The simulated system uses QPSK modulation with two bits per symbol, which has four symbols locating at (1+1j, -1+1j, -1-1j, 1-1j), as shown in Fig. 3.2.

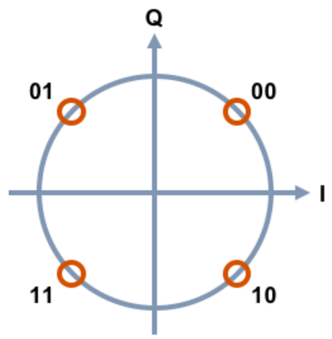


**Figure 3.2:** Symbols in Gray coding

Then, RRC filter is used for two times oversampling and pulse shaping. In data simulation, the sampling rate was increased from 20 GHz to 40 GHz. To achieve better performance, the filter roll-off factor $\beta$ was adjusted together with adaptive CMA filter taps. A bigger RRC filter roll-off factor $\beta$ was used with smaller CMA filter taps, and vice versa.

## 3.2 Channel

The optical signals are transmitted through the optical fiber in analog domain. In data simulation, we simulated signals propagated through optical fiber by adding noise and channel impairments to the signals, which include rotation, PMD and DGD. According to the equation 2.12

$$\mathrm{SNR} = \frac{P_{signal}}{P_{noise}}$$

and given signals, noise was generated with SNR (0-11) and random generator. Rotation angle $\theta$ between the principal axis to the observed axis was a variable, which could be adjusted during testing to check if signals converged at different rotation angles. The DGD between x- and y-polarization axes was tested at 40 ps. The simulation for noise and simulation for channel impairments both used 40 GHz sampling rate.

## 3.3 DSP in Receiver

For DSP in receiver, we focused on adaptive CMA equalizer, signals correlation, and symbols decision.

### 3.3.1 Adaptive CMA equalizer

The adaptive CMA equalizer first run in floating point, and then run in fixed point. The result of floating point was taken as the reference to check the penalty of fixed point. The floating point values were quantized to fixed point with methods round, ceiling, and floor. The fixed point algorithm was first applied in the CMA filter block on filter optimal taps $wn$, and then applied in the CMA training block on filter output $yn$, error $en$, and taps $wn$.

#### 3.3.1.1 Data Simulation In Floating Point

Data simulation started in *floating point* for reasons outlined below:

- This allows us to verify correct functionality of CMA equalizer algorithm including proper signal convergence.
- Fine tuning parameters of CMA equalizer for best performance.
- This enables us to save *floating point* data for further use in scaling the *fixed point* algorithm.

**Check signals convergence**

- One way to check signals convergence is plot moving average over the CMA equalizer adaptive filter error. The moving window $N = 1000$, which means the first calculation of the absolute mean average value is based on data 1 to 1000, and the second calculation is based on data 2 to 1001, and the third calculation is based on data 3 to 1002, and so on. The moving average error curve shows how fast the CMA equalizer converges and stabilizes.
- Another way to check convergence is plot the constellation diagram. Quadrature Phase Shift Keying (QPSK) has four symbols located in four quadrants. The received signals include noise and channel impairments, which were added onto signals while transmitting through the channel. The CMA equalizer moved signals close to the CMA circle with constant radius to the origin, and surrounding four symbols symmetrically.

- The third way to check signals convergence is histogram, which is similar to the constellation diagram, and which shows signals surrounding symbols.

**Fine tuning parameters**
To have stable and reliable test result, 100,000 symbols were used for data simulation. SNR was tested in range (0 - 11 dB). For initialization, all filter taps were set to zeros except a central peak (1) at real part of $w_{xx}$ and $w_{yy}$. In filter error function, same as equation 2.5:

$$en = (d\_energy - y\_energy) * y,$$

where $d\_energy$ is the normalized power of desired reference $d$.

In data simulation, the following parameters were tested with different values to find the combination of parameters which have the best performance.

- Transmitter RRC filter roll-off factor $\beta$, tested in range 0.01 to 0.5.
- Receiver CMA equalizer filter taps, tested in range 1 to 101.
- Receiver CMA equalizer filter step size $\mu$, tested in range $1 \cdot 10^{-1}$ to $1 \cdot 10^{-10}$. $\mu$ is very sensitive, so it should be tuned slowly in testing.
- Channel impairment rotation angle $\theta$, tested in range $\pi/18$ to $\pi$.

Performance curves SER vs. SNR and EVM vs. SNR for SNR = 0-11 were used to check the overall performance of the CMA equalizer. The parameters with simulation curve more close to the theory curve are recommended.

**Save data and plottings**
All variables defined or generated in the floating point were saved as .npz files, one file for each SNR value, with total 12 files for SNR = 0-11. The saved floating point data was imported into the fixed point program for scaling the fixed point logic with same floating point values. It also speed up the simulation with less computation.

All plottings for the moving average error, constellation diagram, histogram, signals correlation, performance curves SER vs. SNR and EVM vs. SNR were saved as PDF files for quick searching and archiving.

#### 3.3.1.2 Data Simulation In Fixed Point

The CMA equalizer *fixed point* was first tested with different quantization methods, which included *round, ceiling and floor.* The fixed point quantization was applied in the different locations in the CMA equalizer as follows:

- **Fixed point v1, Fig. 3.3**:
  - CMA filter block filter taps *wn*

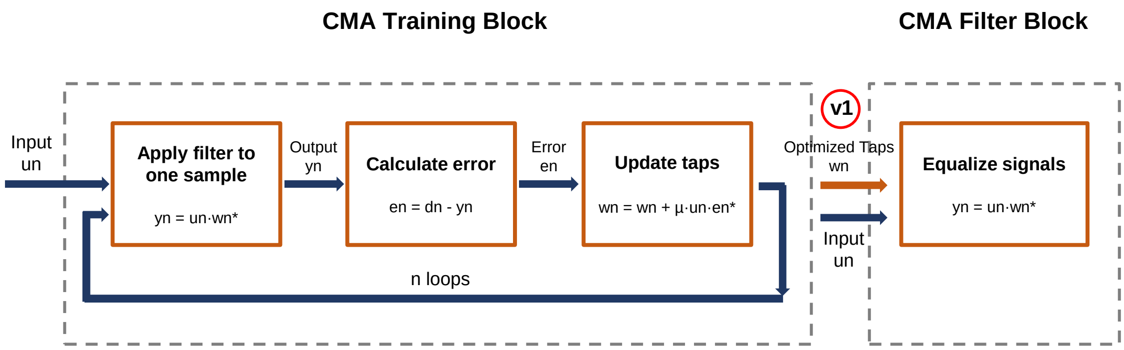


**Figure 3.3:** CMA equalizer fixed point v1 block diagram

- **Fixed point v2, Fig. 3.4**:
  - CMA filter block filter taps *wn*
  - CMA training block output *yn*

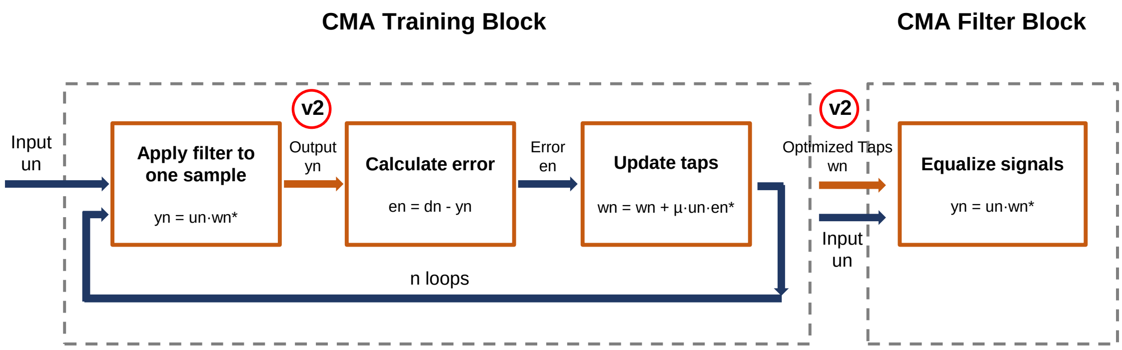


**Figure 3.4:** CMA equalizer fixed point v2 block diagram

- **Fixed point v3, Fig. 3.5**:
  - CMA filter block filter taps *wn*
  - CMA training block output *yn*
  - CMA training block error *en*

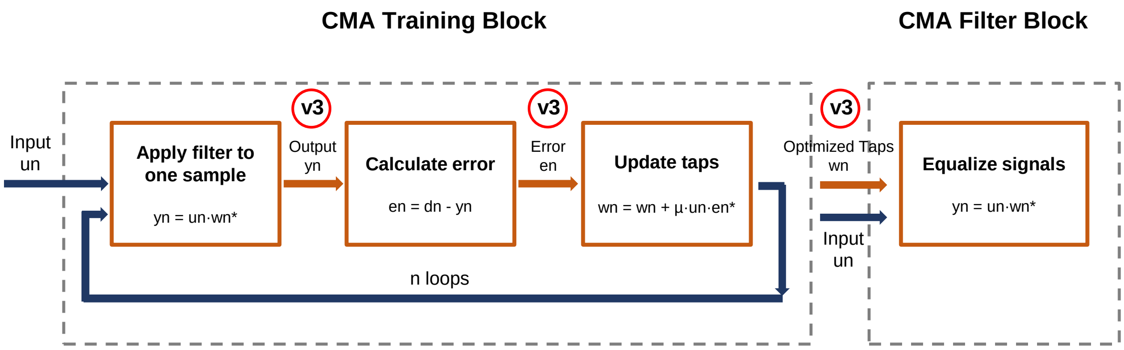


**Figure 3.5:** CMA equalizer fixed point v3 block diagram

- **Fixed point v4, Fig. 3.6**:
  - CMA training block output $yn$
  - CMA training block error $en$
  - CMA training block filter taps $wn$

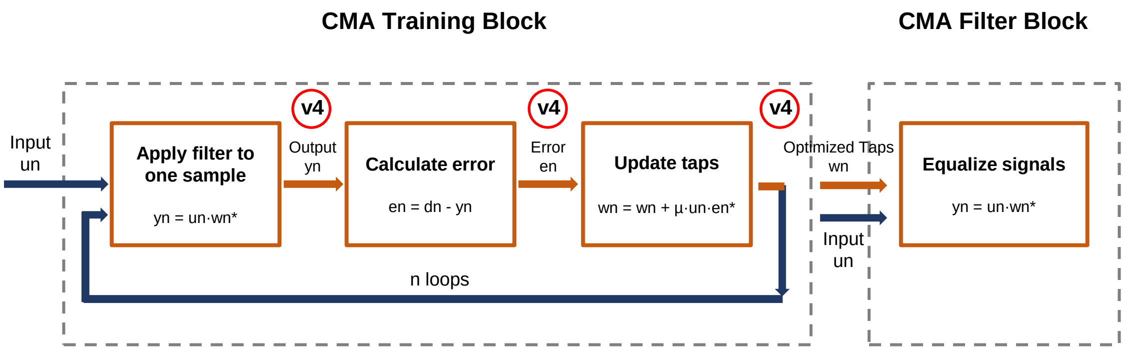


**Figure 3.6:** CMA equalizer fixed point v4 block diagram

The performance of CMA equalizer fixed point was measured with performance curves SER and EVM as a function of number of quantization bits. In the end, the performance difference of 2x2 and 4x4 fixed point was compared with SER vs. quantization bits.

### 3.3.2 Signals Correlation

After checking the CMA adaptive filter was working properly to have signal converged, correlation function was used to check if signal correlation logic was working properly.

#### 3.3.2.1 Polarization Identification

We first checked how closely two signals were matched. The correlation function returned peak value when two signals were the most similar. First, the unknown equalized signal was compared with the known transmitted symbols to identify the polarization of the equalized signal. The equalized signal output from the CMA equalizer had multi dimensions. We first took one dimension of data and compared it with all polarization of the transmitted symbols separately. The comparison returned maximum value means that two compared signals were the most similar to each other, which defined this data dimension had the same polarization as the compared polarization of the transmitted symbols. And then, repeat comparison with the transmitted symbols to identify the polarization for all data dimensions. The steps to find polarization were illustrated in Fig. 3.7, 3.8, 3.9 and 3.10.

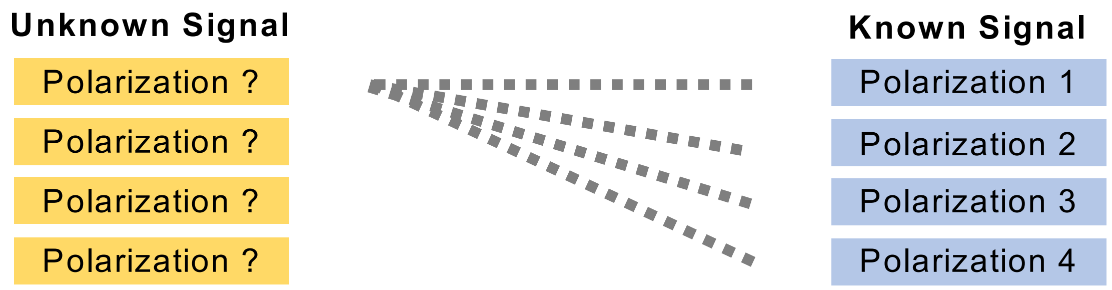


**Figure 3.7:** Signals correlation polarization identification 1

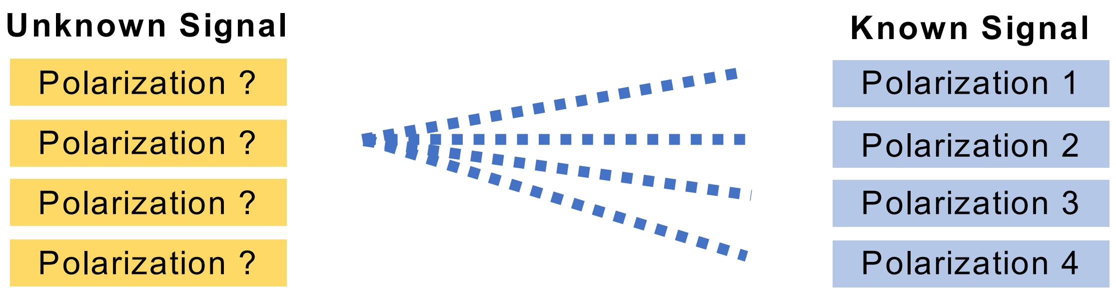


**Figure 3.8:** Signals correlation polarization identification 2

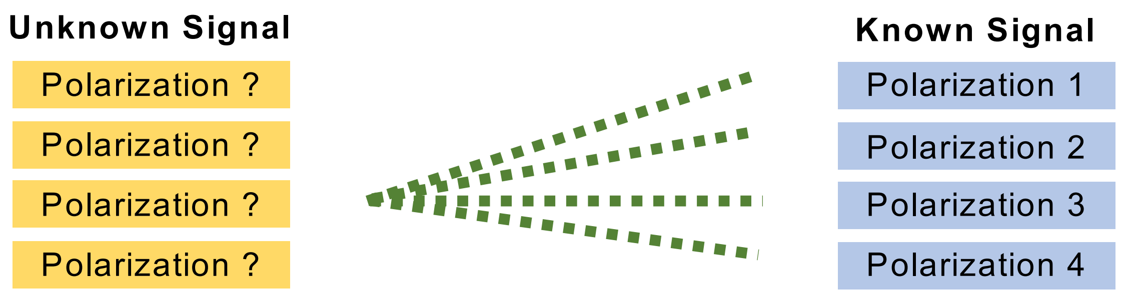


**Figure 3.9:** Signals correlation polarization identification 3


**Figure 3.10:** Signals correlation polarization identification 4

Different dimension of the equalized signal should map to different polarization of the transmitted symbols. Fig. 3.11 shows the case when two data dimensions mapped to the same polarization, which is known as the polarization singularity error. This case reported ValueError: converged onto the same polarization!

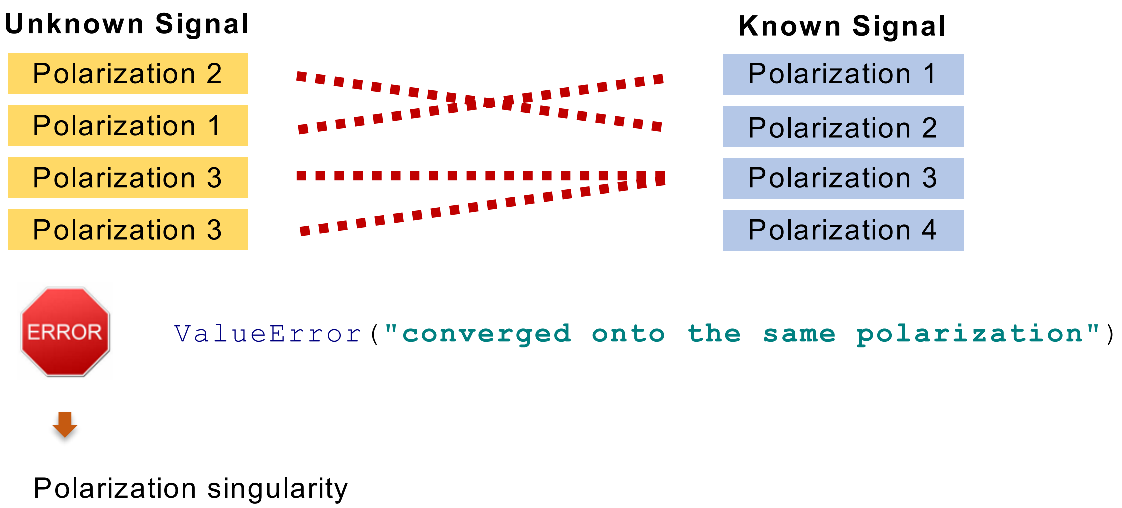


**Figure 3.11:** Signals correlation polarization singularity error

The occurrence of the singularity error means that either the filter logic or the correlation logic is wrong. Fig. 3.12 shows that after correcting the logic, all data dimensions mapped to one polarization each, and the singularity error disappeared.

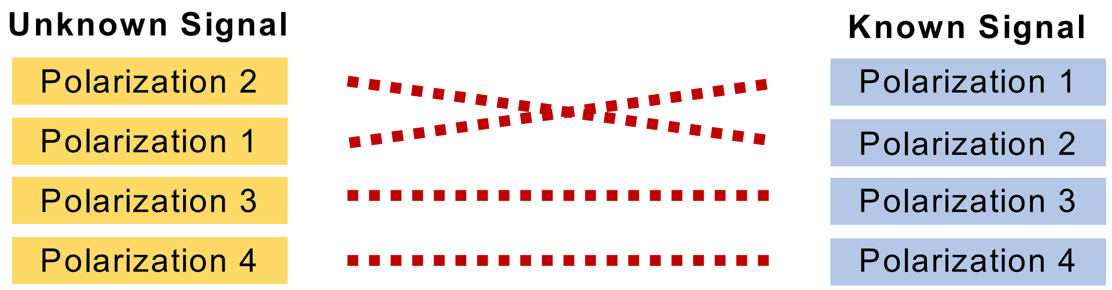


**Figure 3.12:** Signals correlation polarization identification

#### 3.3.2.2 Align Signals

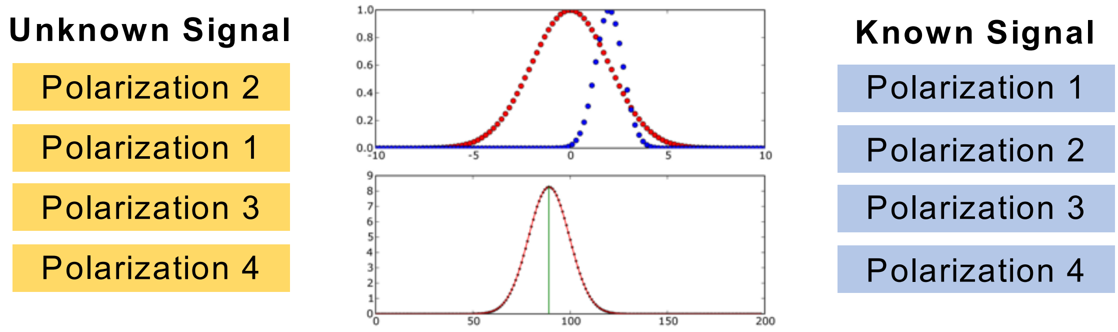


**Figure 3.13:** Signals correlation align signals

After the polarizations have been identified, the next step was to align the equalized signal with the transmitted symbols. When two signals were aligned, the peak was in the center of data array. If the peak has offset to the center, this means that the two signals are not aligned. If two signals were not aligned, we moved the known transmitted symbols to align with the unknown equalized signal, as shown in Fig. 3.13.

### 3.3.3 Symbols Decision

After correlation, signals were moved to the symbol locations by symbols decision. In case of QPSK with four symbols located at 1+1j, -1+1j, -1-1j and 1-1j, the symbols decision can be done in two ways:

1. By sign: The signs of real and imaginary components of signals were compared with the signs of real and imaginary components of symbols. Signals were mapped to symbols with same signs of real and imaginary components.
2. By distance: Check the distance from signals to four symbols and map signals to symbols with the shortest distance.

After symbols decision, measured performance of CMA equalizer with BER, SER, SNR and EVM.

# 4
# Results

Results of 2x2 CMA equalizer floating point and fixed point are covered in Sections 4.1 and Sections 4.2, and results of 4x4 CMA equalizer floating point and fixed point are covered in Sections 4.3 and 4.4, which include data simulation test result, recommended parameters and conclusions.

## 4.1 2x2 CMA Equalizer Floating Point

In this section, we fine tuned parameters to verify the floating point algorithm of the 2x2 CMA equalizer and identify the optimal parameters to deliver the best performance. The plottings of moving average error, constellation diagram, histogram and signals correlation were used to evaluate performance. When tuning one parameter, other parameters kept same without change. After finding the optimal parameters in the floating point, same parameters were used in the fixed point v1, v2, v3 and v4.

- Transmitter RRC filter roll-off factor $\beta$, tested in range 0.01 to 0.5. The simulation results showed that the CMA equalizer was easier to stabilize at bigger $\beta$ value. Since $\beta = 0.1$ is used in the laboratory experiments, we followed this value in the simulation.

- Receiver CMA equalizer filter taps, tested in range 1 to 101. When there was a big gap between the simulation curve and the theory curve, increasing the number of filter taps greatly improved the performance and the simulation curve moved quickly close to the theory curve. But when the simulation curve was very close to the theory curve, increasing the number of taps didn't improve the performance in any discernible way. In simulation, we found that both the even and odd number of taps had similar performance. For example, filter $taps = 3, 4, 5, 6, 7, 8, 9, 10, 11$ were all working. For the 2x2 mode CMA equalizer, filter $taps = 11$ showed good enough performance.

- Receiver CMA equalizer filter step size $\mu$, tested in range $1 \cdot 10^{-1}$ to $1 \cdot 10^{-10}$. Big $\mu$ caused polarization singularity error and small $\mu$ took long time to converge. So, we need to find the biggest $\mu$ value to converge, but without the polarization singularity error. $\mu$ and filter *taps* were working together. Convergence could be reached either with small *taps* and big $\mu$, or with big *taps* and small $\mu$.

  The filter step size $\mu$ was tested from $1 \cdot 10^{-1}$ and for decreasing step sizes at big steps, i.e. $1 \cdot 10^{-2}$, $1 \cdot 10^{-3}$, until $1 \cdot 10^{-10}$. Bigger step sizes $1 \cdot 10^{-1}$ and $1 \cdot 10^{-2}$ had polarization singularity error. Smaller step sizes $1 \cdot 10^{-3}$ to $1 \cdot 10^{-10}$ were working without error. Then, we fine tuned $\mu$ in the range $1 \cdot 10^{-2}$ to $1 \cdot 10^{-4}$ at smaller steps 0.001 and 0.0001, and finally found that filter $taps = 11$ and step size $\mu = 1 \cdot 10^{-3}$ delivered the best performance.

- Channel impairment rotation angle $\theta$, tested in range $\pi/18$ to $\pi$. With the defined CMA equalizer algorithm, $\theta$ working at $\pi/3, \pi/4, \pi/5, \pi/6, \pi/7$ with similar performance, and $\theta$ not working at $\pi/2, \pi/8, \pi/9, \pi/18$. $\pi/6$ was used in 2x2 CMA equalizer simulation.

**Recommended parameters for 2x2 CMA equalizer simulation:**
- Transmitter RRC filter roll-off factor $\beta = 0.1$
- Receiver CMA equalizer filter $taps = 11$
- Receiver CMA equalizer filter step size $\mu = 1 \cdot 10^{-3}$
- Channel impairment rotation angle $\theta = \pi/6$

Below are the plottings of the moving average error, constellation diagram, histogram and signals correlation with the optimal parameters.

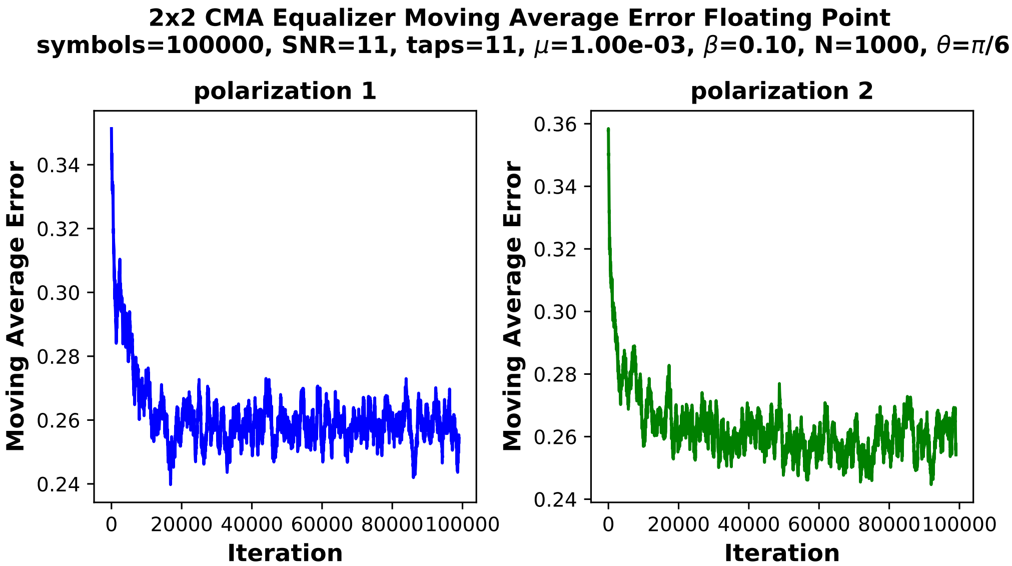


**Figure 4.1:** 2x2 CMA equalizer floating point moving average error

Fig. 4.1 is the plotting for the moving average error. The *error* was the difference between the filter desired response and the filter output in the CMA equalizer. The *moving average error* was calculated at the moving window $N = 1000$. From Fig. 4.1 we can see that the moving average error dropped quickly from around 0.35 to 0.25 in the first 20000 symbols, and the 2x2 CMA equalizer converged when symbols > 20000 with the moving average error stabilized around 0.25.

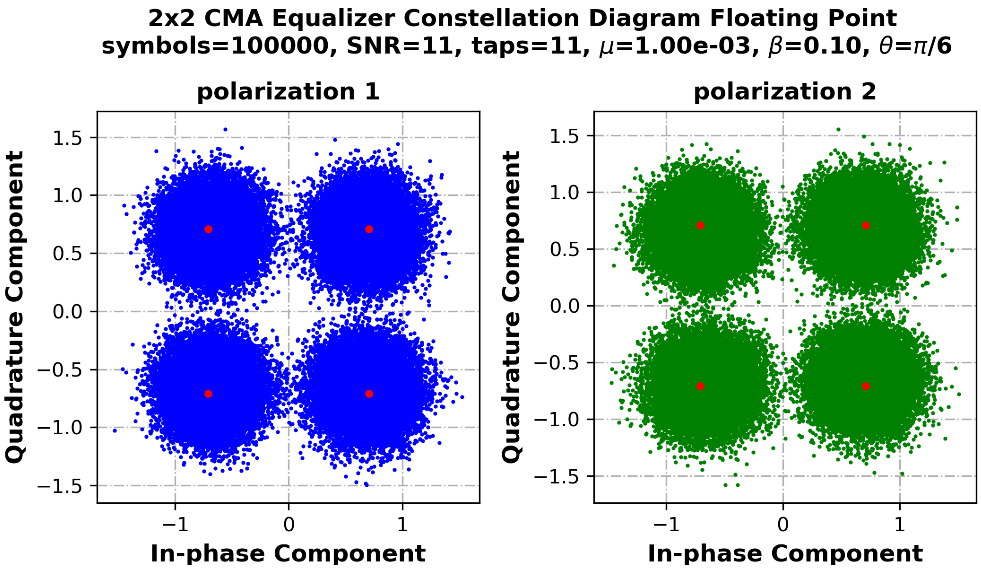


**Figure 4.2:** 2x2 CMA equalizer floating point constellation diagram

From Fig. 4.2 we can see that two polarizations of 2x2 CMA equalizer were converged.

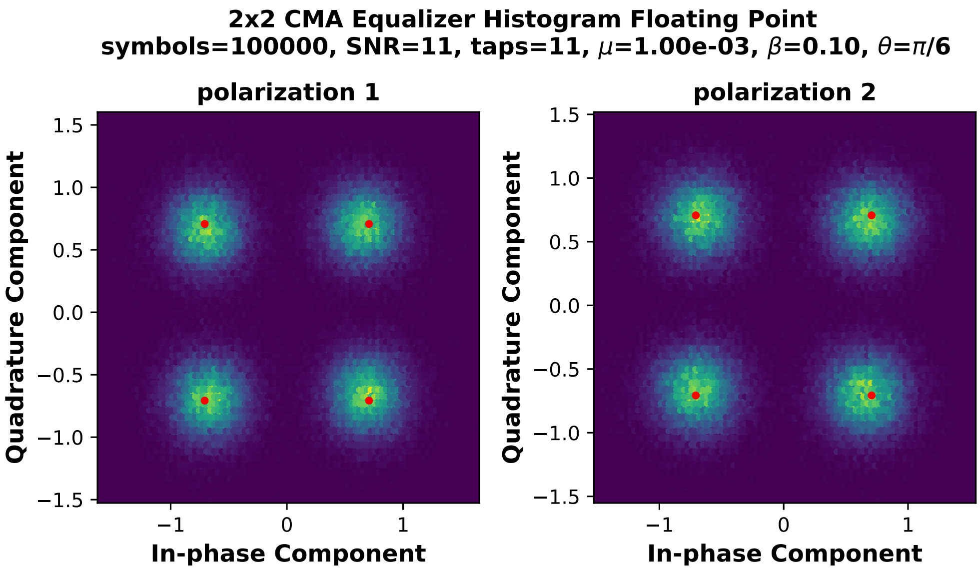


**Figure 4.3:** 2x2 CMA equalizer floating point histogram

Fig. 4.3 depicts both polarization 1 and 2 converged to symbols.

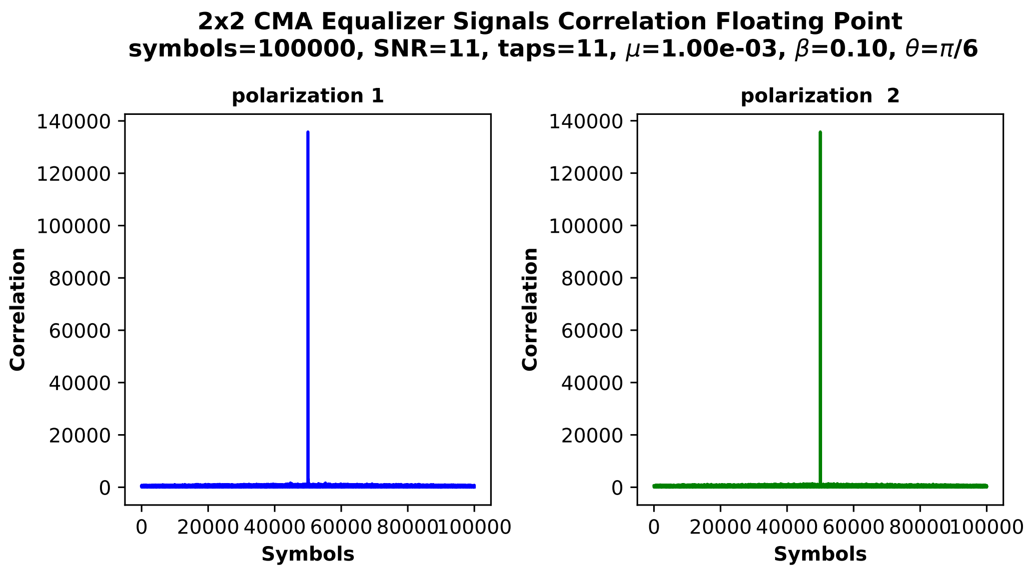


**Figure 4.4:** 2x2 CMA equalizer floating point signals correlation

Fig. 4.4 represents that the signals after CMA equalized were aligned with the signals transmitted with the peak in the middle of data array.

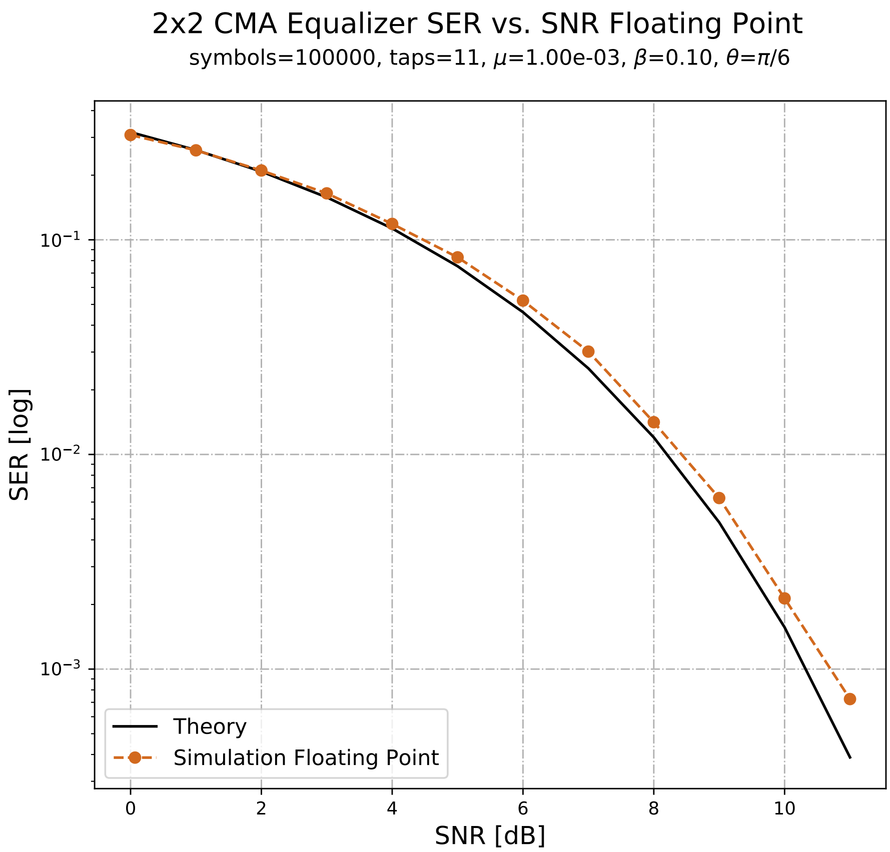


**Figure 4.5:** 2x2 CMA equalizer floating point SER vs. SNR

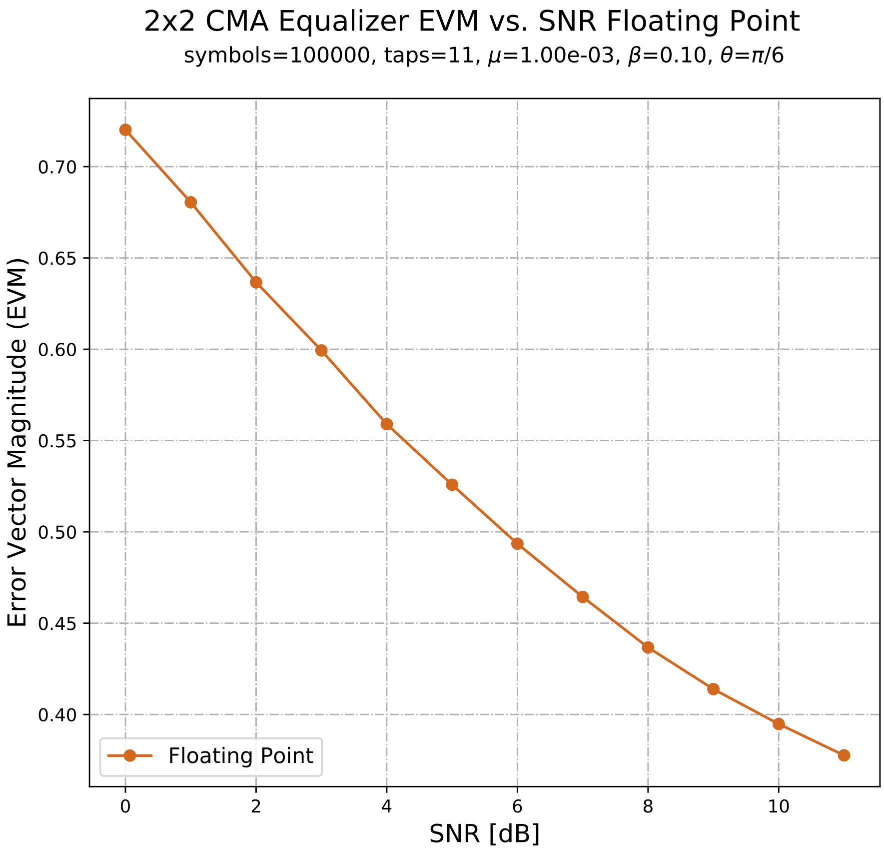


**Figure 4.6:** 2x2 CMA equalizer floating point EVM vs. SNR

In SER vs. SNR performance curve as shown in Fig 4.5, by comparing the simulation curve with the theory curve, we found the combination of parameters that deliver the best performance, which had the simulation curve most aligned to the theory curve.

From Fig.4.6 we can see that the Error Vector Magnitude (EVM) reduced while increasing SNR, which also proved that signals were more converged with smaller distance to the reference symbols.

## 4.2 2x2 CMA Equalizer Fixed Point

Fig. 4.7 represents the initial testing of the fixed point 2x2 CMA equalizer. The test result showed that the quantization method *round* in blue color was better than *ceiling* and *floor*, and the performance was more close to the *floating point*. So, in the following tests, we focused on quantization method *round* only.

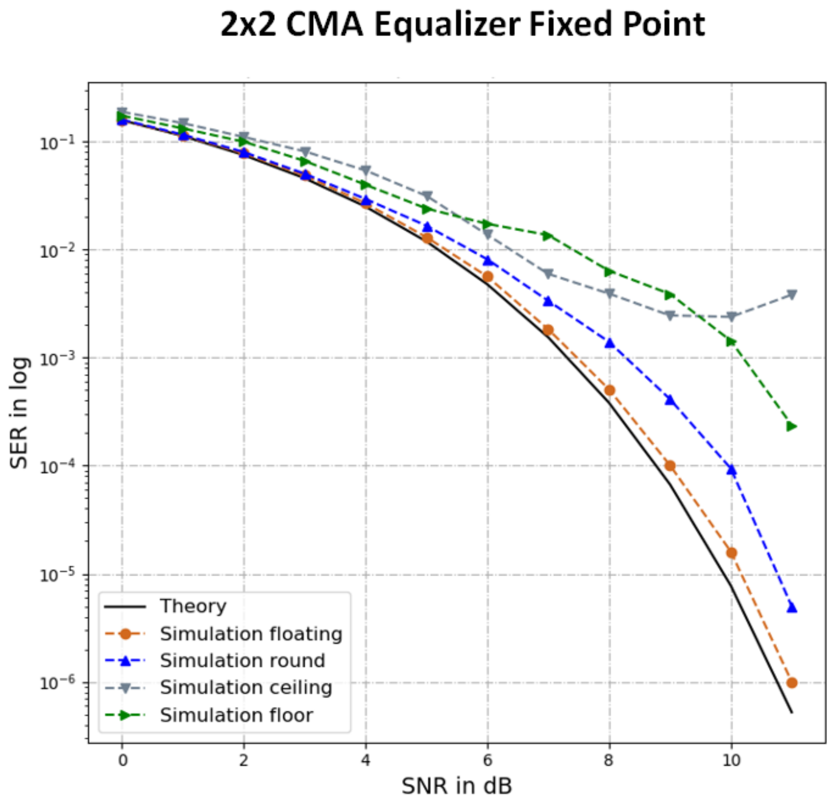


**Figure 4.7:** 2x2 CMA equalizer fixed point quantization methods

In simulation for the fixed point CMA equalizer, filter output *yn*, error *en* and taps *wn* were quantized to fixed point. The quantization values were calculated based on the *min* and *max* of *yn*, *en* and *wn* and the quantization bits. For example, when quantization bit = 3, the difference between the *min* and *max* were split into $2^3 = 8$ steps.

In the first round of simulation, the *min* and *max* of *yn*, *en* and *wn* were the average of 36 sample values. During simulation, fixed *min* and *max* values were used to calculate the quantization values. Fixed *min* and *max* had an error in the SER vs. SNR performance curve that the *fixed point* curve lower than the *floating point* curve.

To solve this issue, the *min* and *max* of *yn*, *en* and *wn* were updated in each run of simulation, leading to correct SER vs. SNR performance curve with the *fixed point* curve higher than the *floating point* curve.

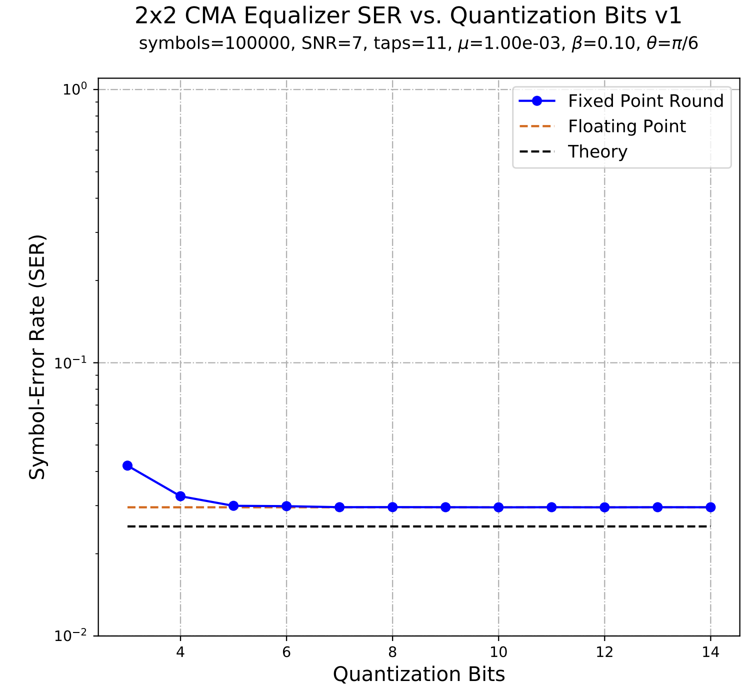


**Figure 4.8:** 2x2 CMA equalizer fixed point v1 SER vs. quantization bits

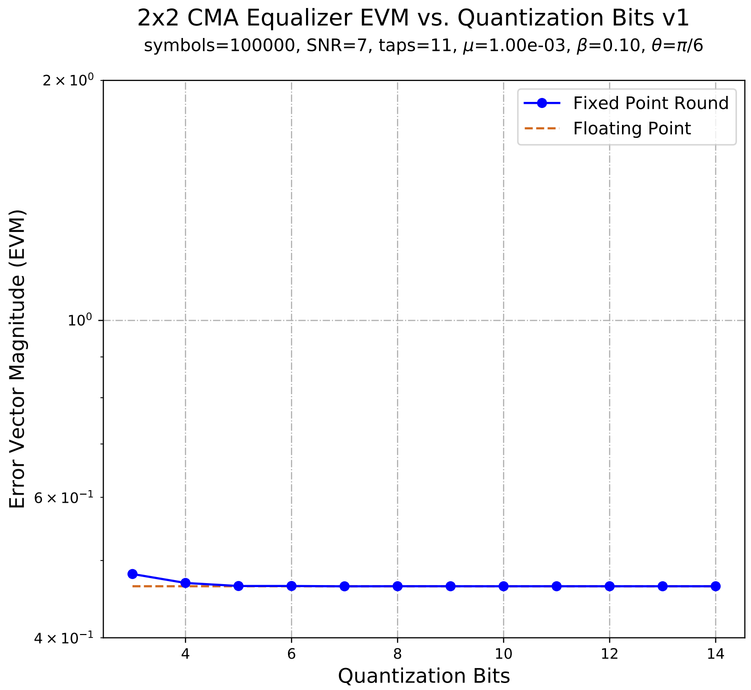


**Figure 4.9:** 2x2 CMA equalizer fixed point v1 EVM vs. quantization bits

Fig. 4.8 and 4.9 show that 2x2 fixed point v1, when filter taps *wn* in the CMA filter block quantized to fixed point at bits $\geq 5$, *fixed point round* had same performance as *floating point*.

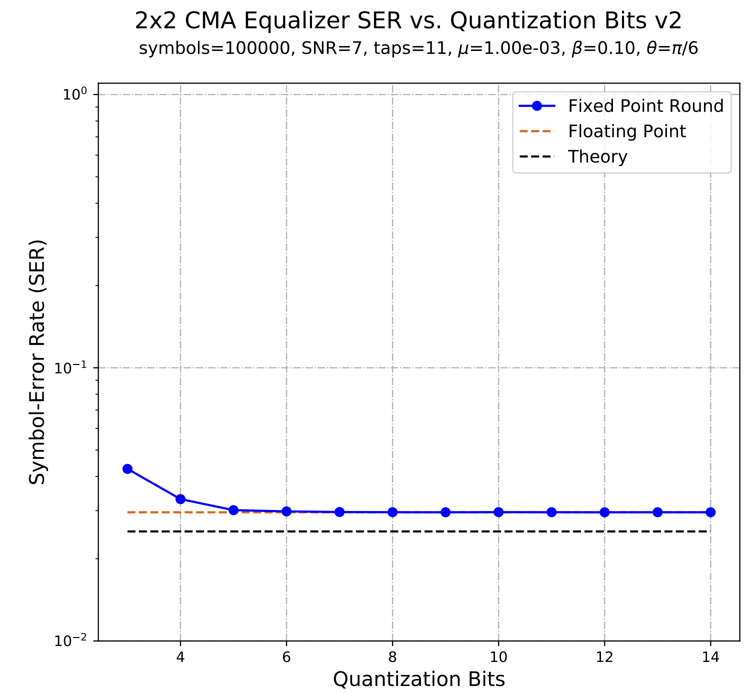


**Figure 4.10:** 2x2 CMA equalizer fixed point v2 SER vs. quantization bits

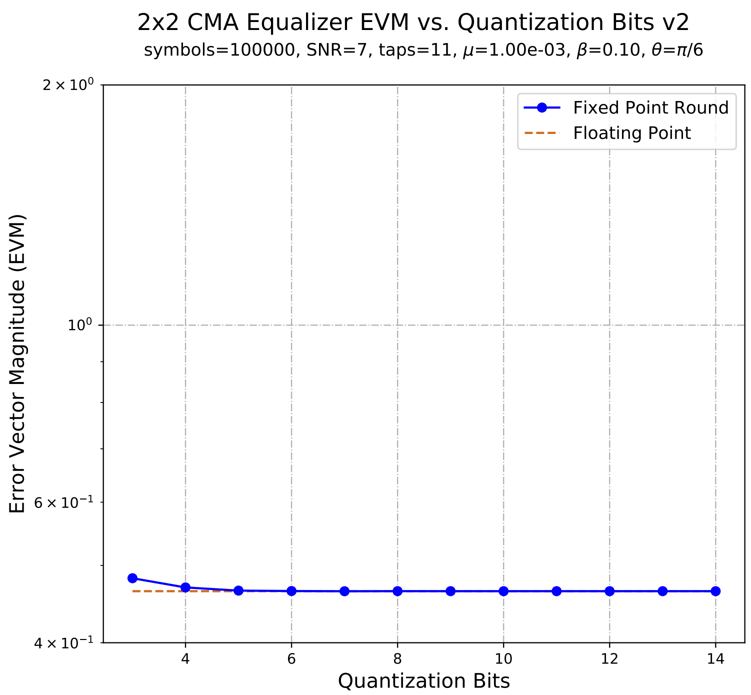


**Figure 4.11:** 2x2 CMA equalizer fixed point v2 EVM vs. quantization bits

Fig. 4.10 and 4.11 show that 2x2 fixed point v2, when filter taps $wn$ in the CMA filter block and filter output $yn$ in the CMA training block quantized to fixed point at bits $\geq 5$, *fixed point round* had same performance as *floating point*.

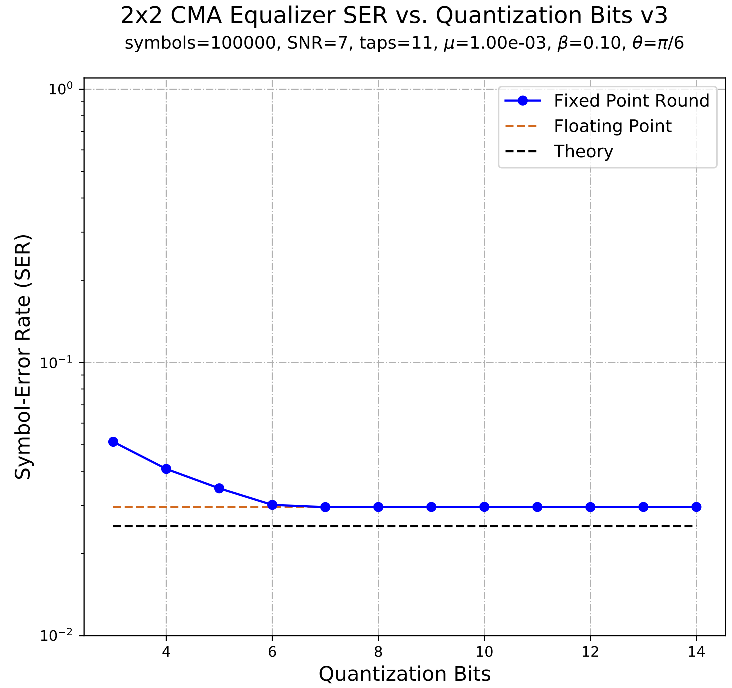


**Figure 4.12:** 2x2 CMA equalizer fixed point v3 SER vs. quantization bits

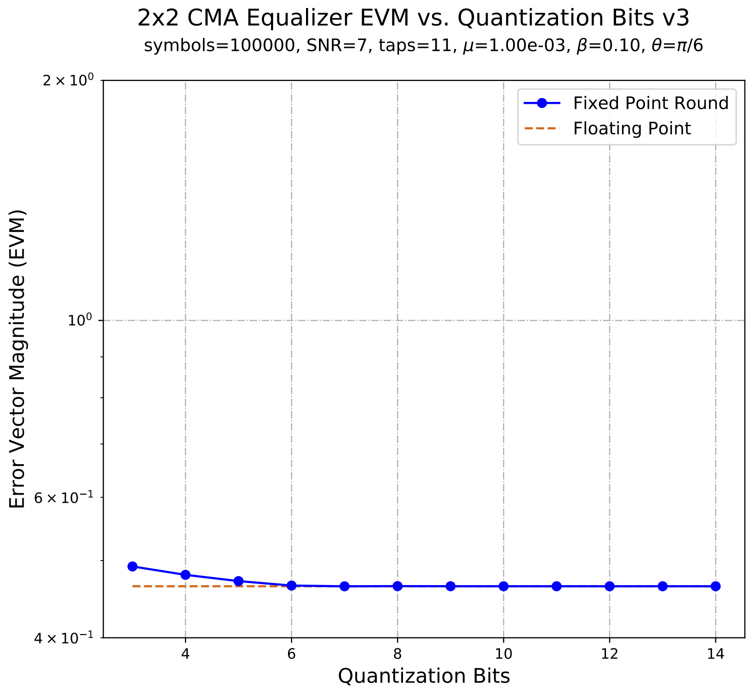


**Figure 4.13:** 2x2 CMA equalizer fixed point v3 EVM vs. quantization bits

Fig. 4.12 and 4.13 show that 2x2 fixed point v3, when filter taps *wn* in the CMA filter block and filter output *yn* and error *en* in the CMA training block quantized to fixed point at bits $\geq 6$, *fixed point round* had same performance as *floating point*.

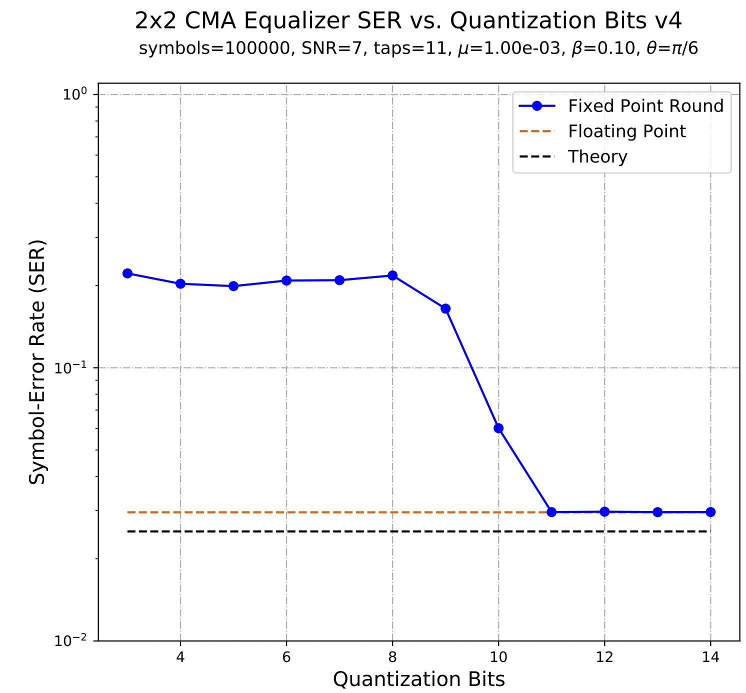


**Figure 4.14:** 2x2 CMA equalizer fixed point v4 SER vs. quantization bits

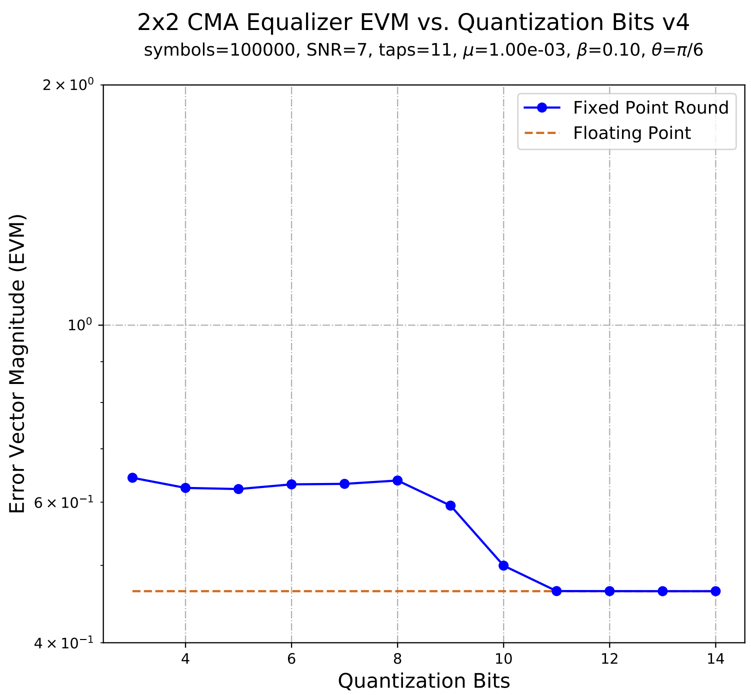


**Figure 4.15:** 2x2 CMA equalizer fixed point v4 EVM vs. quantization bits

Fig. 4.14 and 4.15 show that 2x2 fixed point v4, when filter output $yn$, error $en$ and taps $wn$ in the CMA training block quantized to fixed point at bits $\geq 11$, *fixed point round* had same performance as *floating point*.

**2x2 CMA Equalizer Conclusions:**

- Performance penalty mainly came from quantization error and filter algorithm.
- Fixed point quantization method *round* had better performance than *ceiling* and $floor$.
- Fixed point v1 and v2:
  *quantization bit* $\geq 5$, *fixed point round* had same performance as *floating point*.
- Fixed point v3:
  *quantization bit* $\geq 6$, *fixed point round* had same performance as *floating point*.
- Fixed point v4:
  *quantization bit* $\geq 11$, *fixed point round* had same performance as *floating point*.

## 4.3 4x4 CMA Equalizer Floating Point

In this section, we used same methods as 2x2 CMA equalizer to verify the floating point algorithm of the 4x4 CMA equalizer and identify the optimal parameters.

- Transmitter RRC filter roll-off factor $\beta = 0.1$, followed the $\beta$ value used in the laboratory experiments.

- Receiver CMA equalizer filter $taps = 11$. For 4x4 mode CMA equalizer, we found that filter $taps = 4*n+1$ (n=0,1,2,3,...) and $taps = 4*n+3$ had different SER vs. SNR performance curve trend. In case filter $taps = 4*n+1$, the simulation curve aligned to the theory curve at smaller SNR values, and moved far away from the theory curve at bigger SNR values. Filter $taps = 4*n+3$ had opposite trend, which had bigger gap between the simulation curve and the theory curve at smaller SNR values and aligned to the theory curve at bigger SNR values. For the 4x4 mode CMA equalizer, filter $taps = 11$ showed good enough performance.

- Receiver CMA equalizer filter step size $\mu$ was tested from $1 \cdot 10^{-1}$ and first reduced in big step size, i.e. $1 \cdot 10^{-2}$, $1 \cdot 10^{-3}$, until $1 \cdot 10^{-10}$. Bigger step sizes $1 \cdot 10^{-1}, 1 \cdot 10^{-2}$ and $1 \cdot 10^{-3}$ had polarization singularity error. Smaller step sizes $1 \cdot 10^{-4}$ to $1 \cdot 10^{-10}$ were working without error. Then, we fine tuned $\mu$ in the range $1 \cdot 10^{-3}$ to $1 \cdot 10^{-6}$ at smaller steps $10^{-6}$, and finally found that filter $taps = 11$ and step size $\mu = 2.35 \cdot 10^{-5}$ delivered the best performance.

- Channel impairment rotation angle $\theta$, followed 2x2 mode and used $\pi/6$ in 4x4 CMA equalizer simulation.

**Recommended parameters for 4x4 CMA equalizer simulation:**

- Transmitter RRC filter roll-off factor $\beta = 0.1$
- Receiver CMA equalizer adaptive filter $taps = 11$
- Receiver CMA equalizer adaptive filter step size $\mu = 2.35 \cdot 10^{-5}$
- Channel impairment rotation angle $\theta = \pi/6$

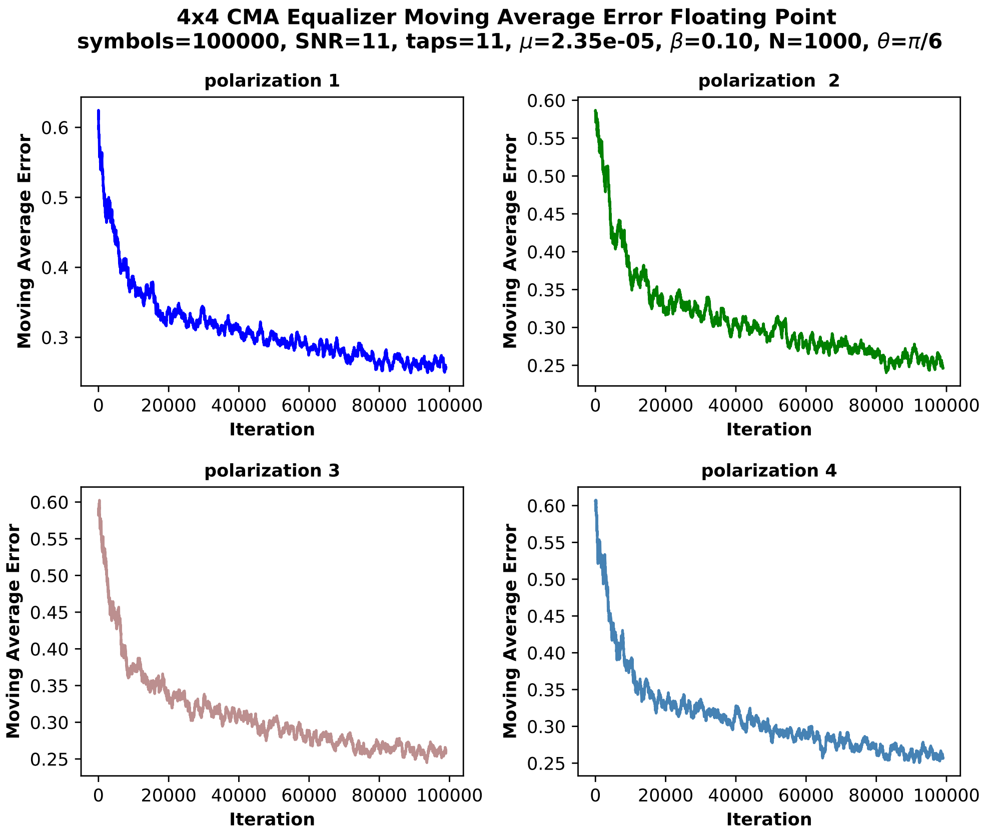


**Figure 4.16:** 4x4 CMA equalizer floating point moving average error

Fig. 4.16 is the plotting for the moving average error. The *error* was the different between the filter desired response and the filter output in the CMA equalizer. The *moving average error* was calculated at the moving window $N = 1000$. The 4x4 CMA equalizer moving average error curve in Fig. 4.16 shows that the equalizer was working properly in four polarizations and signals converged at around 20000 symbols.

Compare the moving average error of 4x4 mode (Fig. 4.16) and 2x2 mode (Fig. 4.1), we can see that:

- 4x4 mode initial error was almost doubled as 2x2 mode
- 2x2 mode stabilized more quickly than 4x4 mode
- both 2x2 and 4x4 modes stabilized at the moving average error around 0.25

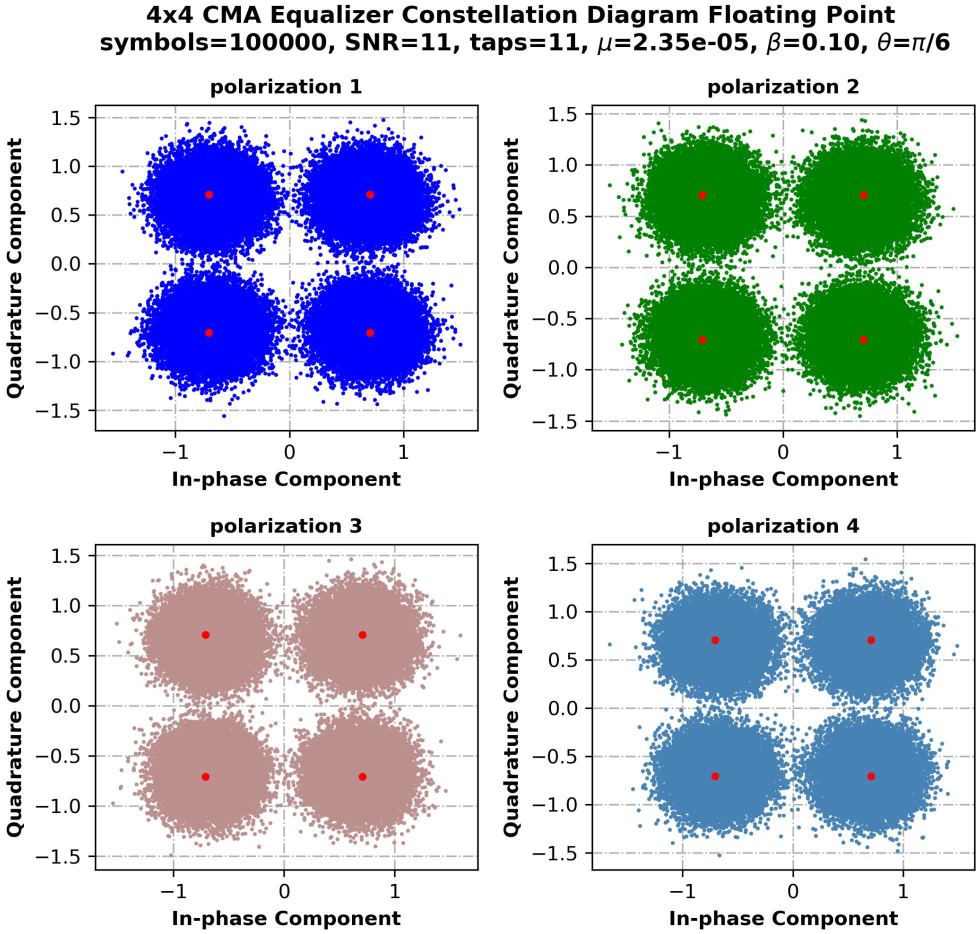


**Figure 4.17:** 4x4 CMA equalizer floating point constellation diagram

4x4 CMA equalizer constellation diagram in Fig. 4.17 shows that signals were converged in four polarizations.

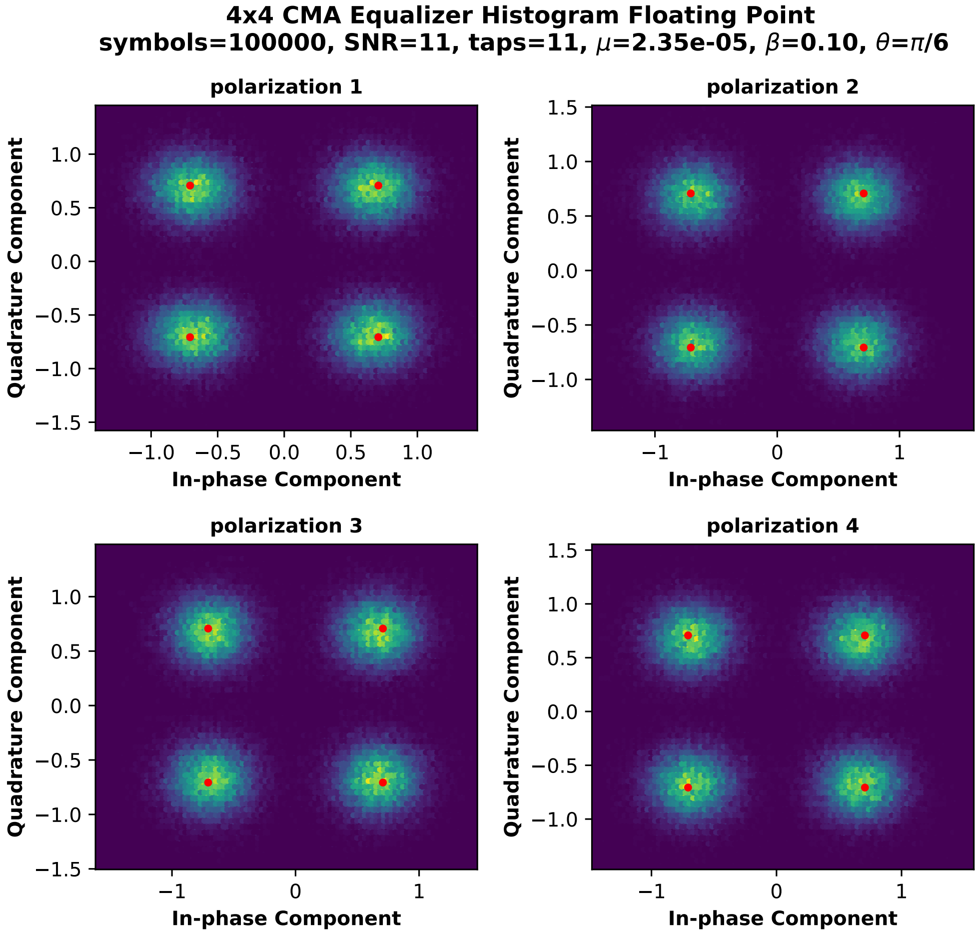


**Figure 4.18:** 4x4 CMA equalizer floating point histogram

The 4x4 CMA equalizer histogram in Fig. 4.18 was used to double check that signals were converged in four polarizations.

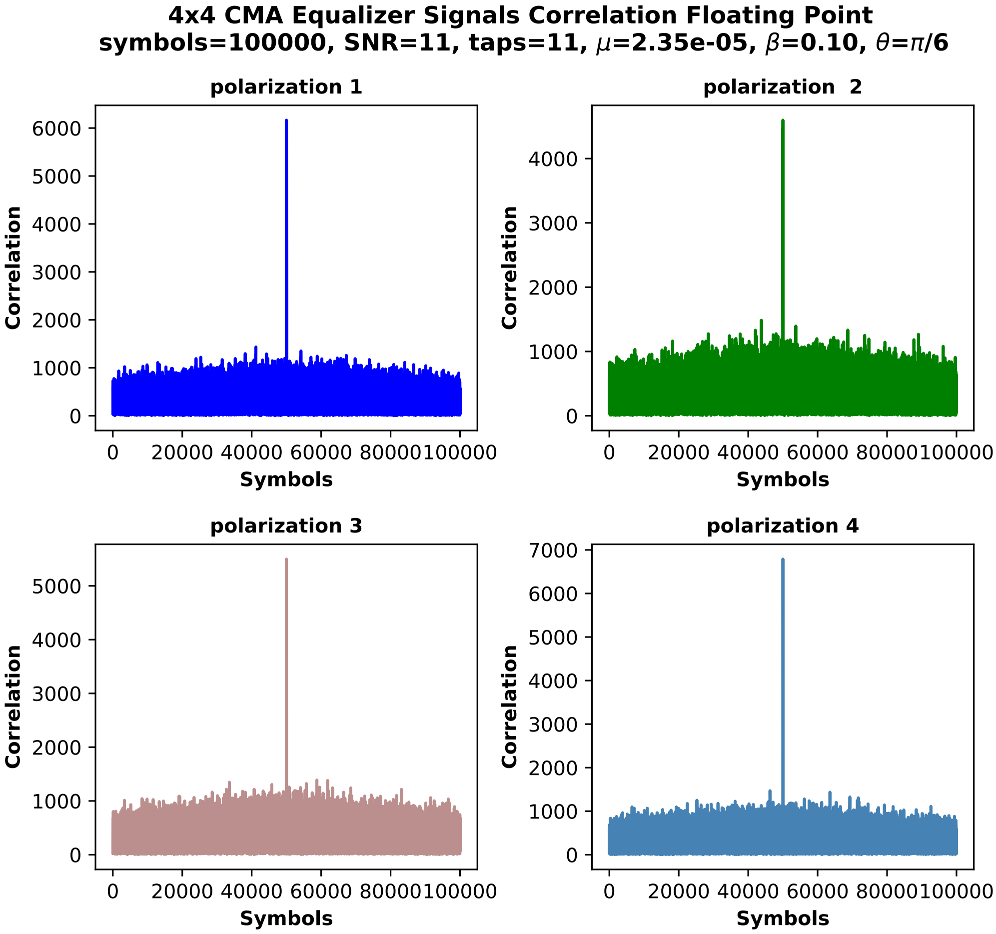


**Figure 4.19:** 4x4 CMA equalizer floating point signals correlation

The 4x4 CMA equalizer signals correlation in Fig. 4.19 shows that the CMA equalized signals were aligned with the transmitted symbols in four polarizations.

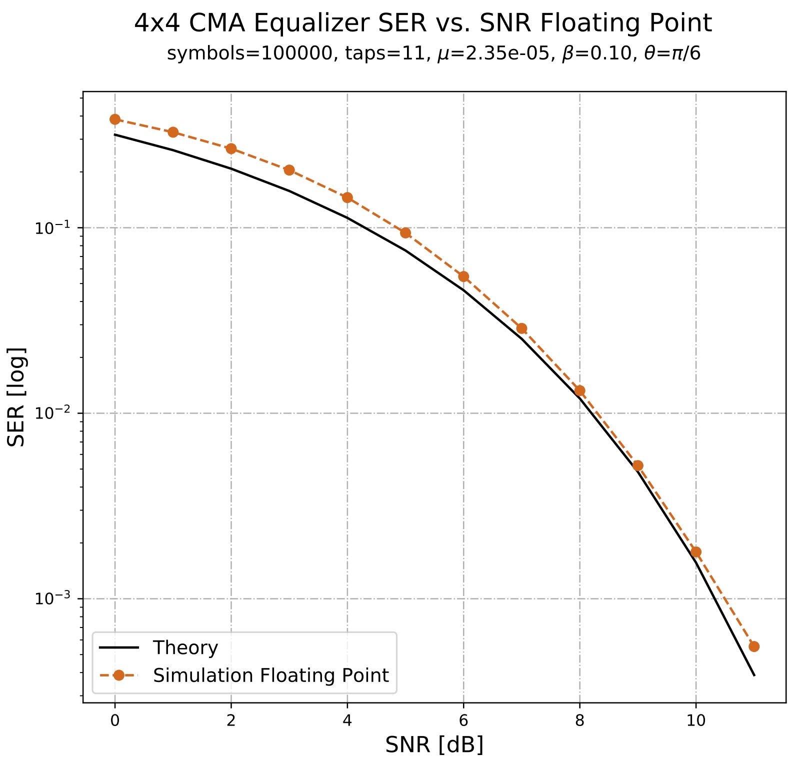


**Figure 4.20:** 4x4 CMA equalizer floating point SER vs. SNR

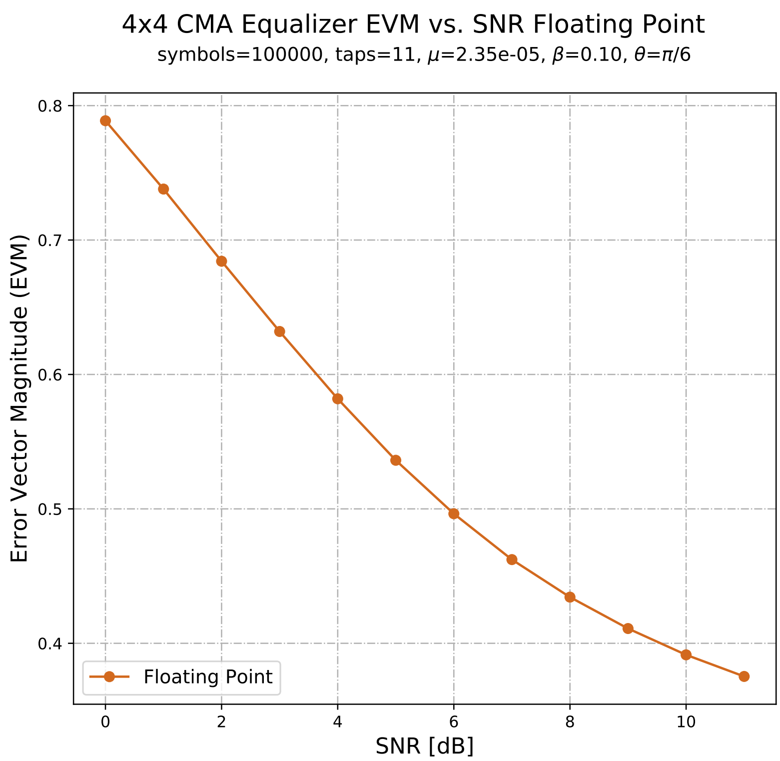


**Figure 4.21:** 4x4 CMA equalizer floating point EVM vs. SNR

In SER vs. SNR performance curve as shown in Fig 4.20, by comparing the simulation curve with the theory curve, we found the combination of parameters that deliver the best performance, which had the simulation curve most aligned to the theory curve.

From Fig.4.21 we can see that the Error Vector Magnitude (EVM) reduced while increasing SNR, which also proved that signals were more converged with smaller distance to the reference symbols.

## 4.4 4x4 CMA Equalizer Fixed Point

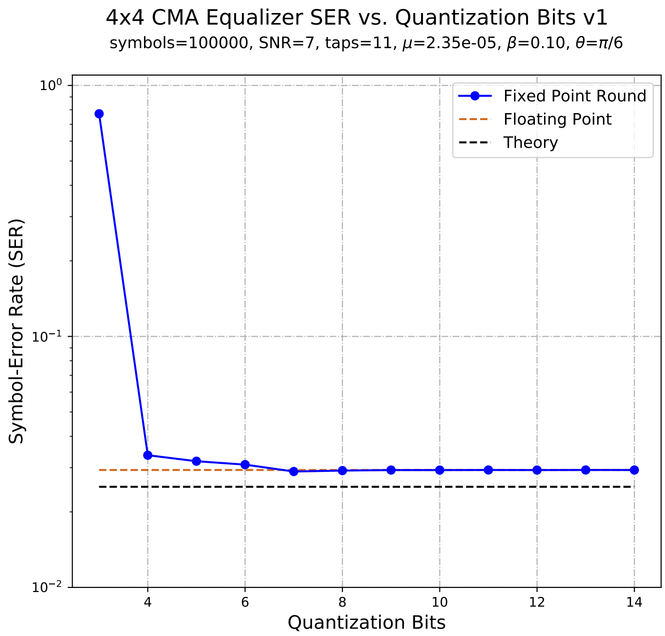


**Figure 4.22:** 4x4 CMA equalizer fixed point v1 SER vs. quantization bits

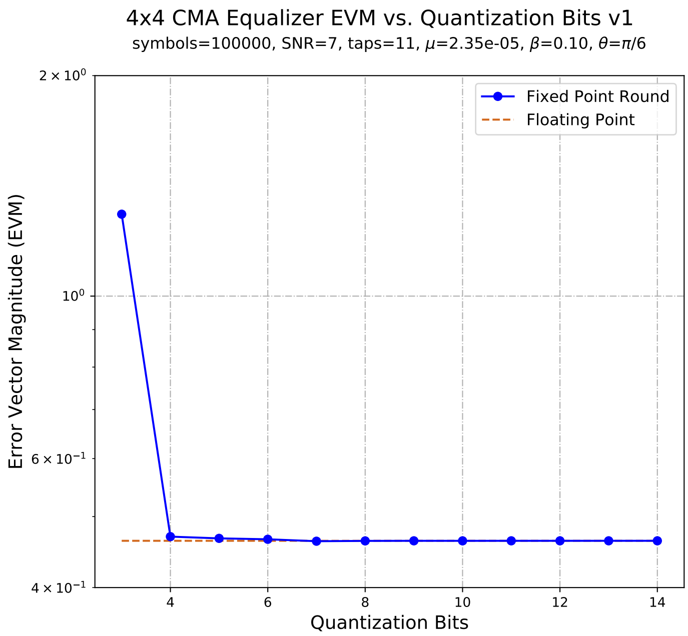


**Figure 4.23:** 4x4 CMA equalizer fixed point v1 EVM vs. quantization bits

Fig. 4.22 and 4.23 show that 4x4 fixed point v1, when filter taps $wn$ in the CMA filter block quantized to fixed point at bits $\geq 7$, *fixed point round* had same performance as *floating point*.

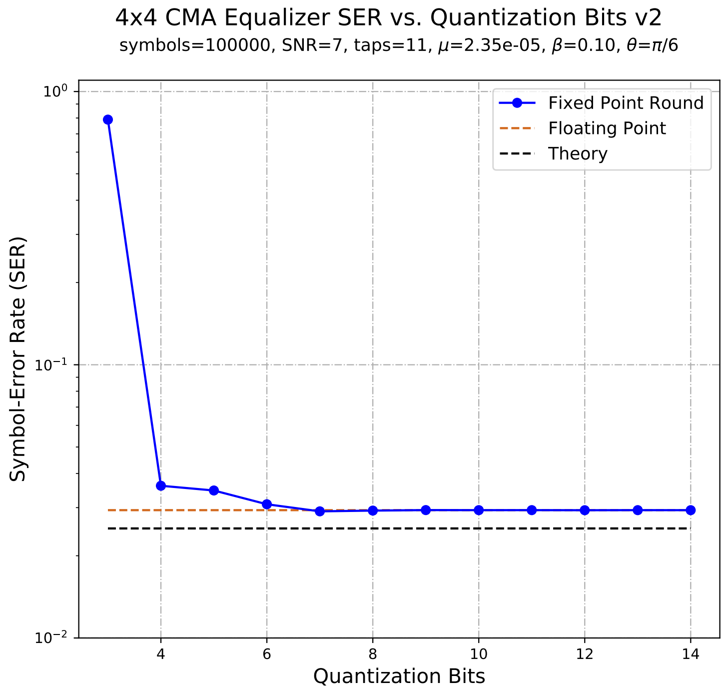


**Figure 4.24:** 4x4 CMA equalizer fixed point v2 SER vs. quantization bits

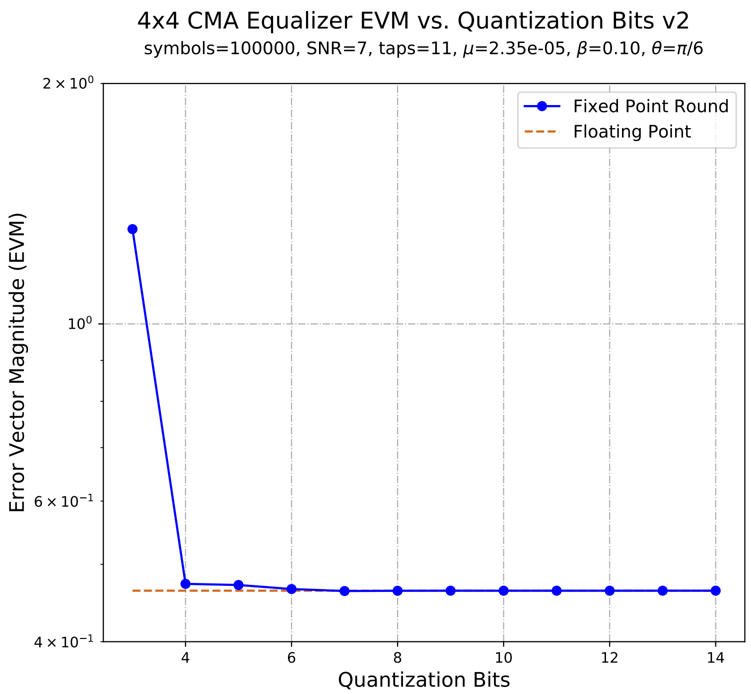


**Figure 4.25:** 4x4 CMA equalizer fixed point v2 EVM vs. quantization bits

Fig. 4.24 and 4.25 show that 4x4 fixed point v2, when filter taps $wn$ in the CMA filter block and filter output $yn$ in the CMA training block quantized to fixed point at bits $\geq 7$, *fixed point round* had same performance as *floating point*.

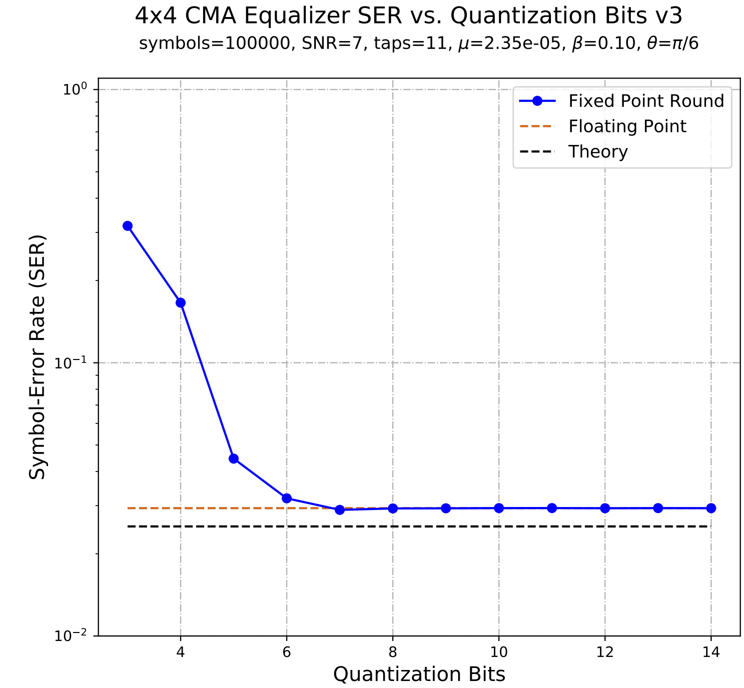


**Figure 4.26:** 4x4 CMA equalizer fixed point v3 SER vs. quantization bits

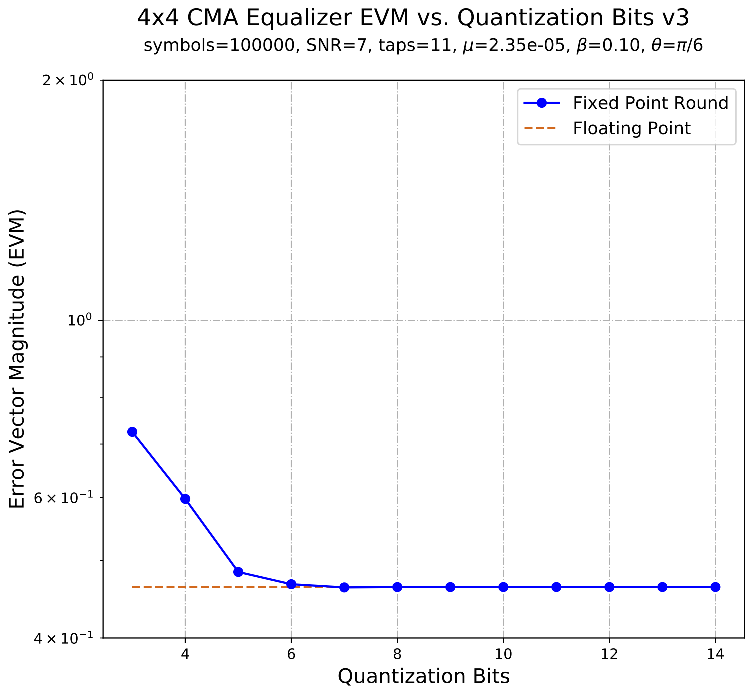


**Figure 4.27:** 4x4 CMA equalizer fixed point v3 EVM vs. quantization bits

Fig. 4.26 and 4.27 show that 4x4 fixed point v3, when filter taps *wn* in the CMA filter block and filter output *yn* and error *en* in the CMA training block quantized to fixed point at bits $\geq 7$, *fixed point round* had same performance as *floating point*.

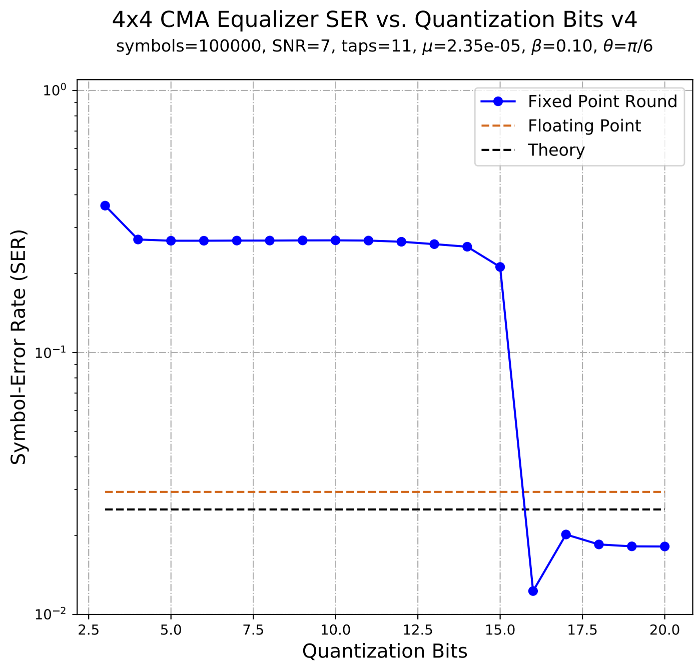


**Figure 4.28:** 4x4 CMA equalizer fixed point v4 SER vs. quantization bits

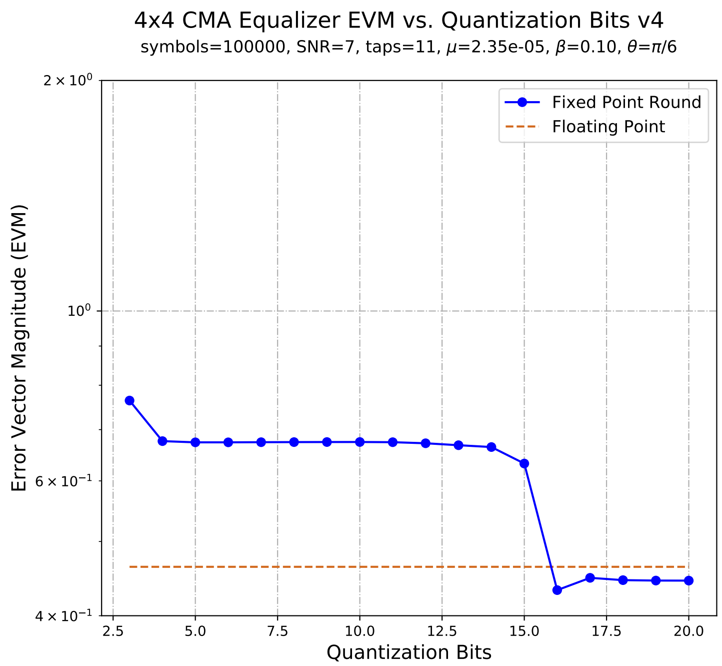


**Figure 4.29:** 4x4 CMA equalizer fixed point v4 EVM vs. quantization bits

Fig. 4.28 and 4.29 show that 4x4 fixed point v4, when filter output *yn*, error *en* and taps *wn* in the CMA training block quantized to fixed point, in *quantization bits* = 3-20, *fixed point round* didn't have same performance as *floating point*.

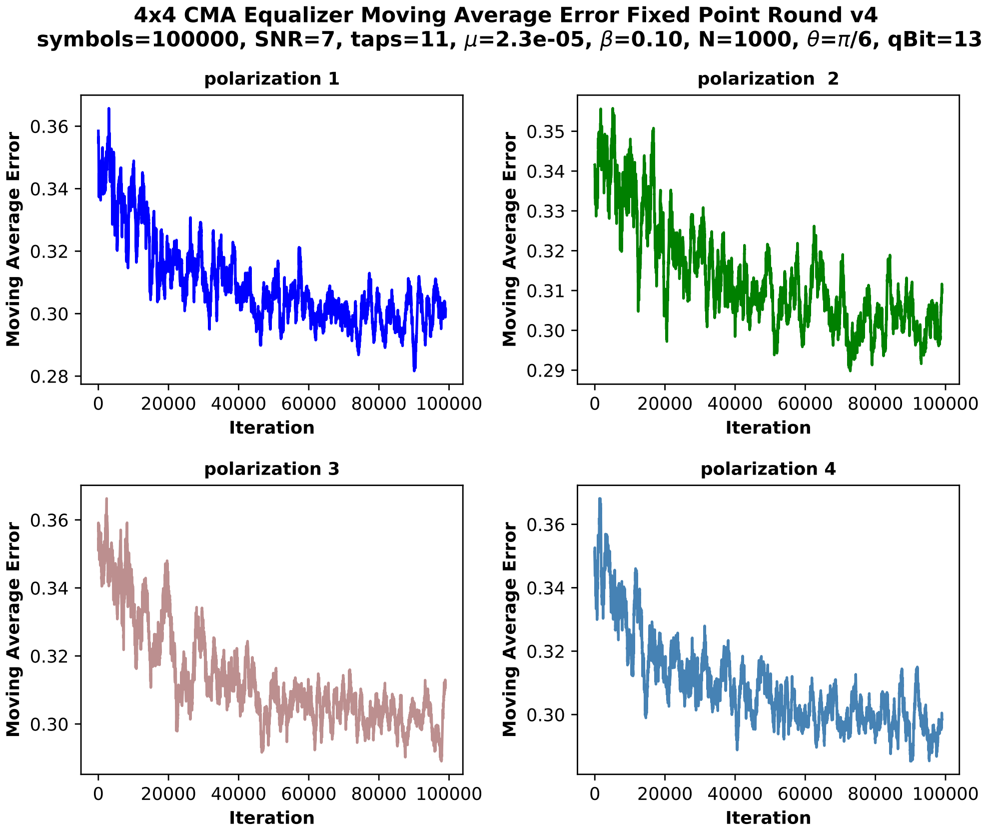


**Figure 4.30:** 4x4 CMA equalizer fixed point v4 moving average error at quantization bit=13

Fig. 4.30 shows that the 4x4 CMA equalizer fixed point v4 moving average error had obvious downward trend when *quantization bit* $\geq 13$.

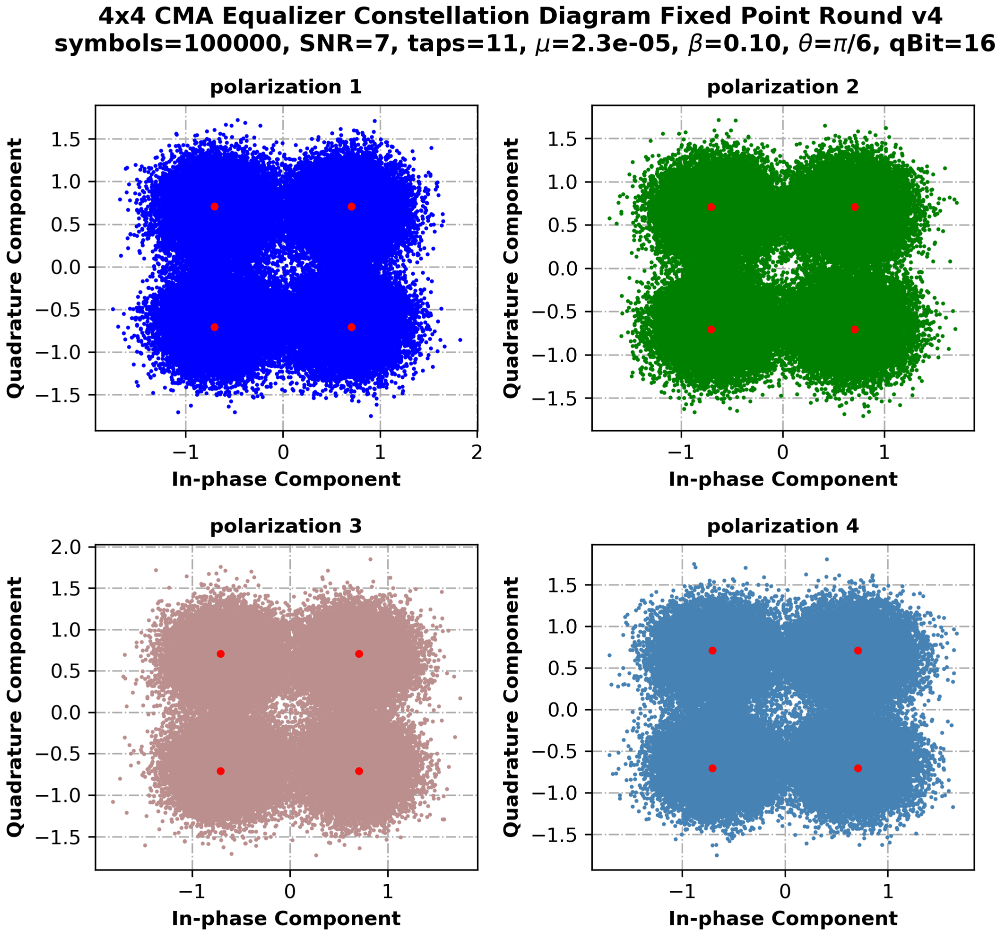


**Figure 4.31:** 4x4 CMA equalizer fixed point v4 constellation diagram at quantization bit=16

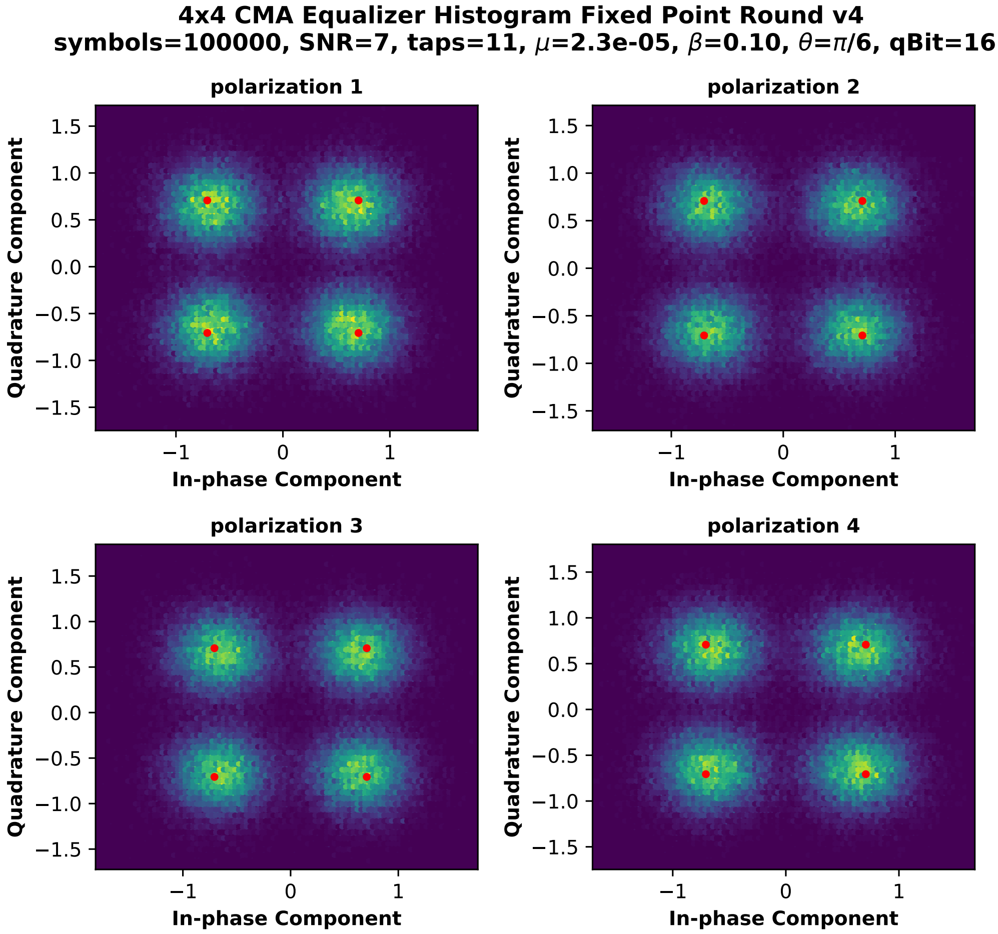


**Figure 4.32:** 4x4 CMA equalizer fixed point v4 histogram at quantization bit=16

Fig. 4.31 and Fig. 4.32 show that signals converged when $quantization\ bit \geq 16$.

**4x4 CMA equalizer conclusions:**

- Fixed point v1, v2, v3:
  $quantization\ bits \geq 7$, *fixed point round* had same performance as *floating point*.
- Fixed point v4:
  $quantization\ bits = 3 - 20$, *fixed point round* didn't have same performance as *floating point*.
  $quantization\ bits = 3 - 15$, the simulation curve of *fixed point round* was higher than the simulation curve of *floating point*.
  $quantization\ bits = 16 - 20$, the simulation curve of *fixed point round* was lower than the simulation curve of *floating point*.
  $quantization\ bits \geq 16$, signals converged.

In the end, we compared 2x2 mode and 4x4 mode CMA equalizer fixed point v1 SER vs. quantization bits at quantization bits = 4, 6, 8. From Fig. 4.33 we can see that in general, 2x2 mode and 4x4 mode had similar SER errors, even though sometimes 2x2 mode had a little bigger errors than 4x4 mode, and sometimes 4x4 mode had a little bigger error than 2x2 mode.

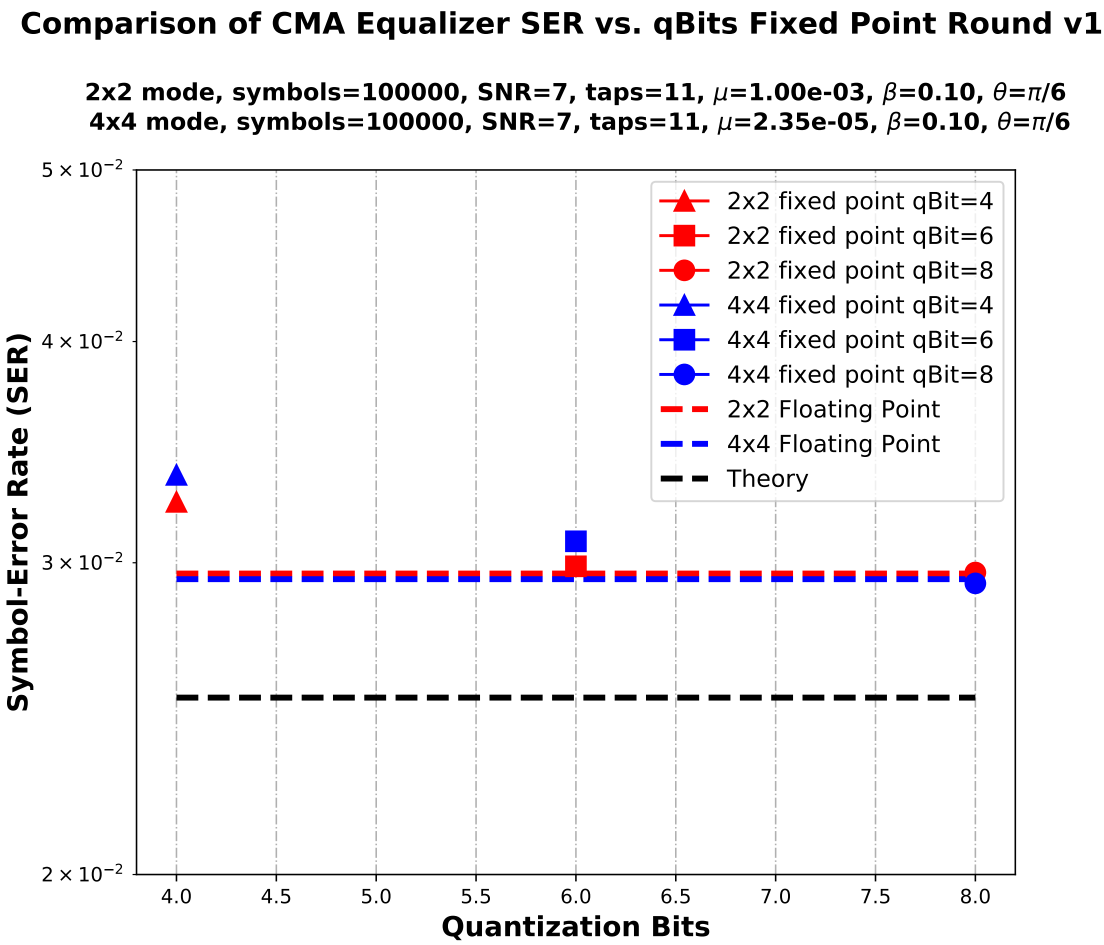


**Figure 4.33:** Comparison of 2x2 mode and 4x4 mode CMA equalizer

# 5 Conclusions

## 5.1 Discussions

The goal of this project was to investigate how limited fixed point precision affects the performance of CMA equalizer for SDM transmission. We developed an implementation of one specific algorithm in Python, and tuned it to different precision with different quantization methods and different number of quantization bits. This algorithm was used to process simulated MDM data to evaluate performance as a function of SNR, number of spatial dimensions and precision.

The simulation results showed that modifying CMA equalizer from the floating point to the fixed point increased SER errors at lower number of quantization bits, but had same SER errors at higher number of quantization bits, except 4x4 fixed point v4.

Comparing three kinds of fixed point quantization methods, we found that *round* had better performance than *ceiling* and *floor*. Among four fixed point quantization versions, v1 was the most important and effective one, because it had the lowest computational cost and applied in the CMA filter block directly to equalize signals without the time-and-power consumption looping process. V1 was also the foundation of v2, v3 and v4.

2x2 mode and 4x4 mode CMA equalizer had similar SER error, even though sometimes 2x2 mode had a little bigger error than 4x4 mode, and sometimes 4x4 mode had a little bigger error than 2x2 mode.

The simulation time increased exponentially when increased quantization bit. For example 4x4 mode fixed point v4, the simulation time increased from 4 min at quantization bit = 16 to 170 min at quantization bit = 20. This made 4x4 mode fixed point v4 impossible to be used for real-time data capture. It's uncertain that if the simulation time can be reduced by optimizing the CMA equalizer algorithm or use another simulation tool, such as MATLAB.

## 5.2 Conclusions

The CMA equalizer *fixed point* could reach the same performance as the *floating point* by increasing the number of quantization bits.

For 2x2 mode CMA equalizer, *fixed point* had same performance as the *floating point* when quantization bits $\geq 5$ for v1 and v2, $\geq 6$ for v3, and $\geq 11$ for v4.

For 4x4 mode CMA equalizer, *fixed point* had same performance as the *floating point* when quantization bits $\geq 7$ for v1, v2 and v3. In case of v4, in the range of quantization bits $= 3-20$, *fixed point* didn't have same performance as the *floating point*.

# Bibliography

[1] Turner, V., Gantz, J. F., Reinsel, D., & Minton, S. (2014). The Digital Universe of Opportunities: Rich Data and Increasing Value of the Internet of Things. IDC White Paper, (April), 1–5. https://doi.org/10.7790/ajtde.v2n3.47

[2] Richardson, D. J., Fini, J. M., & Nelson, L. E. (2013). Space-division multiplexing in optical fibres. Nature Photonics. https://doi.org/10.1038/nphoton.2013.94

[3] Ryf, R., Randel, S., Gnauck, A. H., Bolle, C., Essiambre, R.-J., Winzer, P., & Lingle, R. (2011). Space-division multiplexing over 10 km of three-mode fiber using coherent 6 × 6 MIMO processing. In Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011 (p. PDPB10). Washington, D.C.: OSA. https://doi.org/10.1364/NFOEC.2011.PDPB10

[4] Randel, S., Sierra, A., Mumtaz, S., Tulino, A., Ryf, R., Winzer, P. J., & Essiambre, R. J. (2012). Adaptive MIMO signal processing for mode-division multiplexing - OSA Technical Digest. Optical Fiber Communication Conference, OW3D.5. https://doi.org/10.1364/OFC.2012.OW3D.5

[5] Ragheb, A. M., Shoaib, M., Alshebeili, S., & Fathallah, H. (2015). Enhanced Blind Equalization for Optical DP-QAM in Finite Precision Hardware. IEEE Photonics Technology Letters, 27(2), 181–184. https://doi.org/10.1109/LPT.2014.2364265

[6] Shibahara, K., Lee, D., Kobayashi, T., Mizuno, T., Takara, H., Sano, A., & Morioka, T. (2016). Dense SDM (12-Core × 3-Mode) transmission over 527 km with 33.2-ns mode-dispersion employing low-complexity parallel MIMO frequency-domain equalization. Journal of Lightwave Technology, 34(1), 196–204. https://doi.org/10.1109/JLT.2015.2463102

[7] Ryf, R., Randel, S., Gnauck, A. H., Bolle, C., Sierra, A., Mumtaz, S., & Lingle, R. (2012). Mode-division multiplexing over 96 km of few-mode fiber using coherent 6×6 MIMO processing. Journal of Lightwave Technology, 30(4), 521–531. https://doi.org/10.1109/JLT.2011.2174336

[8] Randel, S., Corteselli, S., Badini, D., Pilori, D., Caelles, S., Chandrasekhar, S., & Winzer, P. J. (2015). First real-time coherent MIMO-DSP for six coupled mode transmission. In 2015 IEEE Photonics Conference, IPC 2015. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IPCon.2015.7323761

[9] Sun, H., Sahai, A., & Wawrzynek, J. (2017). Spectrum Access System: Design and Implementation of the Decision-Feedback Equalizer in Software. Master's Thesis in University of Cali-

fornia at Berkeley. Technical Report No. UCB/EECS-2017-112. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-112.html.

[10] Mazur, M. (2015). Fiber Nonlinearities in Coherent Optical Communication Systems. Master's Thesis in Chalmers University of Technology.

[11] Haykin, S. (2014). Adaptive Filter Theory. Wittenmark, B. (1993). Adaptive filter theory. https://doi.org/10.1016/0005-1098(93)90162-M

[12] Wu, M., Dick, C., Cavallaro, J. R., & Studer, C. (2016). High-Throughput Data Detection for Massive MU-MIMO-OFDM Using Coordinate Descent. IEEE Transactions on Circuits and Systems I: Regular Papers, 63(12), 2357–2367. https://doi.org/10.1109/TCSI.2016.2611645

[13] Faruk, M. S., & Savory, S. J. (2017). Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats. Journal of Lightwave Technology, 35(5), 1125–1141. https://doi.org/10.1109/JLT.2017.2662319

[14] Emeretlis, A., Kefelouras, V., Theodoridis, G., Nanou, M., Politi, C., Georgoulakis, K., & Glentis, G. (2015). FPGA implementation of a MIMO DFE IN 40 GB/S DQPSK optical links. 2015 23rd European Signal Processing Conference (EUSIPCO), Nice, pp. 1581-1585. doi: 10.1109/EUSIPCO.2015.7362650

[15] Proakis, J. G., & Salehi, M. (2007). Digital Communications. Chart (p. 1150). https://doi.org/10.1007/978-1-4684-0004-5

[16] Morioka, T., Awaji, Y., Ryf, R., Winzer, P., Richardson, D., & Poletti, F. (2012, February). Enhancing optical communications with brand new fibers. IEEE Communications Magazine. https://doi.org/10.1109/MCOM.2012.6146483

[17] Daumont, S., Rihawi, B., & Lout, Y. (2008). Root-raised cosine filter influences on PAPR distribution of single carrier signals. In 2008 3rd International Symposium on Communications, Control, and Signal Processing, ISCCSP 2008 (pp. 841–845). https://doi.org/10.1109/ISCCSP.2008.4537340

[18] Kikuchi, K., & Tsukamoto, S. (2008). Evaluation of sensitivity of the digital coherent receiver. Journal of Lightwave Technology, 26(13), 1817–1822. https://doi.org/10.1109/JLT.2007.913589

[19] Du, L. B., & Lowery, A. J. (2010). Improved single channel backpropagation for intra-channel fiber nonlinearity compensation in long-haul optical communication systems. Optics Express, 18(16), 17075–17088. https://doi.org/10.1364/OE.18.017075

[20] Kikuchi, K. (2015). Fundamentals of Coherent Optical Fiber Communications. Journal of Lightwave Technology, PP(99), 1–1. https://doi.org/10.1109/JLT.2015.2463719

[21] Dally, W. J., & Poulton, J. W. (2008). Digital Systems Engineering. Cambridge University Press. pp. 280–285. ISBN: 9780521061759.

[22] Ip, E., & Kahn, J. M. (2007). Digital equalization of chromatic dispersion and polarization mode dispersion. Journal of Lightwave Technology, 25(8), 2033–2043. https://doi.org/10.1109/JLT.2007.900889

[23] Leven, A., Kaneda, N., Klein, A., Koc, U.-V., & Chen, Y.-K. (2006). Real-time implementation of 4.4 Gbit/s QPSK intradyne receiver using field programmable gate array. Electronics Letters, 42(24), 1421. https://doi.org/10.1049/el:20063107

[24] Taylor, M. G. (2004). Coherent Detection Method Using DSP for Demodulation of Signal and Subsequent Equalization of Propagation Impairments. IEEE Photonics Technology Letters, 16(2), 674–676. https://doi.org/10.1109/LPT.2003.823106

[25] Betti, S., Curti, F., De Marchis, G., & Iannone, E. (1991). A Novel Multilevel Coherent Optical System: 4-Quadrature Signaling. Journal of Lightwave Technology, 9(4), 514–523. https://doi.org/10.1109/50.76666

[26] Godard, D. N. (1980). Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems. IEEE Transactions on Communications, 28(11), 1867–1875. https://doi.org/10.1109/TCOM.1980.1094608

[27] Lundberg, L., Fougstedt, C., Larsson-Edefors, P., Andrekson, P. A., & Karlsson, M. (2016). Power Consumption of a Minimal-DSP Coherent Link with a Polarization Multiplexed Pilot-Tone. ECOC 2016; 42nd European Conference on Optical Communication, Dusseldorf, Germany, 2016, pp. 1-3.

[28] Mesiya, M. F. (2013). Contemporary Communication System. Publishing House of Elec.

[29] Cvijetic, M., & Djordjevic, I. B. (2013). Advanced optical communication systems and networks. Artech House.

[30] Schmogrow, R., Nebendahl, B., Winter, M., Josten, A., Hillerkuss, D., Koenig, S., & Leuthold, J. (2012). Error vector magnitude as a performance measure for advanced modulation formats. IEEE Photonics Technology Letters, 24(1), 61–63. https://doi.org/10.1109/LPT.2011.2172405

[31] Mckinley, M. D., Remley, K. a, Myslinski, M., Kenney, J. S., & Nauwelaers, B. (2004). EVM Calculation for Broadband Modulated Signals. 64th ARFTG Conf Dig, 45–52. https://doi.org/10.1109/ICEMI.2007.4350769

[32] (2018). Download PyCharm Edu. JetBrains. https://www.jetbrains.com/pycharm-edu/download/

[33] Doran, R. W. (2007). The Gray Code. Journal of Universal Computer Science, 13(11), 1573–1597. https://doi.org/10.3217/jucs-013-11-1573

# A

# Appendix 1

This section includes the recommended parameters of 2x2 mode and 4x4 mode CMA equalizer.

## A.1 Recommended Parameters For 2x2 Mode CMA Equalizer Simulation

- Transmitter RRC filter roll-off factor $\beta = 0.1$
- Receiver CMA equalizer filter $taps = 11$
- Receiver CMA equalizer filter step size $\mu = 1 \cdot 10^{-3}$
- Channel impairment rotation angle $\theta = \pi/6$

## A.2 Recommended Parameters For 4x4 Mode CMA Equalizer Simulation

- Transmitter RRC filter roll-off factor $\beta = 0.1$
- Receiver CMA equalizer adaptive filter $taps = 11$
- Receiver CMA equalizer adaptive filter step size $\mu = 2.35 \cdot 10^{-5}$
- Channel impairment rotation angle $\theta = \pi/6$