ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη αλγορίθμου για την τυφλή εκτίμηση του σηματοθορυβικού λόγου σε δυναμικά οπτικά δίκτυα

Algorithm development for the blind signal to noise ratio estimation in dynamic optical networks

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Ι. Καστρίτης

Επιβλέπων : Ηρακλής Β. Αβραμόπουλος

Καθηγητής Ε.Μ.Π

Αθήνα, Μάιος 2013

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη αλγορίθμου για την τυφλή εκτίμηση του σηματοθορυβικού λόγου σε δυναμικά οπτικά δίκτυα

Algorithm development for the blind signal to noise ratio estimation in dynamic optical networks

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Ι. Καστρίτης

Επιβλέπων : Ηρακλής Β. Αβραμόπουλος

Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Μάιου 2013.

....	....	....
Η. Αβραμόπουλος	Χ. Καψάλης	Ν. Ουζούνογλου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π

Αθήνα, Μάιος 2013

....

Δημήτριος Ι. Καστρίτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Abstract

As we are approaching the physical capacity limit of standard optical fiber, we are in a great demand of achieving high utilization-efficiency of the available resources in order to accommodate the geometrical increasing telecommunication traffic. The static nature of the conventional networks makes current networks inappropriate for satisfying future consumers' needs and expectations.

During the last decades, new, flexible optical network architectures have been proposed to overcome the inabilities of the static ones. The main advantage of these networks is the capability of adapting their operation on the link's varying existing physical and traffic conditions. The most important procedure that takes place in these networks is the optical performance monitoring (OPM) with witch the network controller is provided with the essential information about system's current state, in order to dynamically allocate the available resources and therefore, to maximize the utilization-efficiency.

In this thesis, we examine optical networks which use coherent detection technology for the transmission schemes. The available modulation formats of the networks are 4-QAM (QPSK), 16-QAM, 64-QAM and 256-QAM. A small reference in digital multilevel modulation formats is made. Moreover, an algorithm based on digital signal processing is proposed to reliably estimate the existing signal to noise ratio (SNR) of the link. In addition, we test the performance of various dynamic networks for different transmission parameters. For the OPM of these networks the proposed algorithm is used. Statistical results have been added for the evaluation of the introduced method and it is proven that the estimations of the algorithm are very accurate.

Keywords: spectral efficiency, quadrature amplitude modulation, constellation diagram, signal to noise ratio, modulation format

Acknowledgments

I would like to thank Prof. Hercules Avramopoulos for giving me the chance to work on this extremely interesting subject. Moreover, I would like to thank Stefanos Dris and Christos Spatharakis, both members of the Photonics Communications Research Laboratory of National Technical University of Athens, for their essential advice and the decisive help they offered me during the project.

In addition, I thank all my friends for their mental support and especially Anthie and Dimitris for their additional technical support.

Finally, I would most like to express my deepest gratitude to my parents for their valuable support during all my school and university years. None of this would exist without their contribution.

Table of Contents

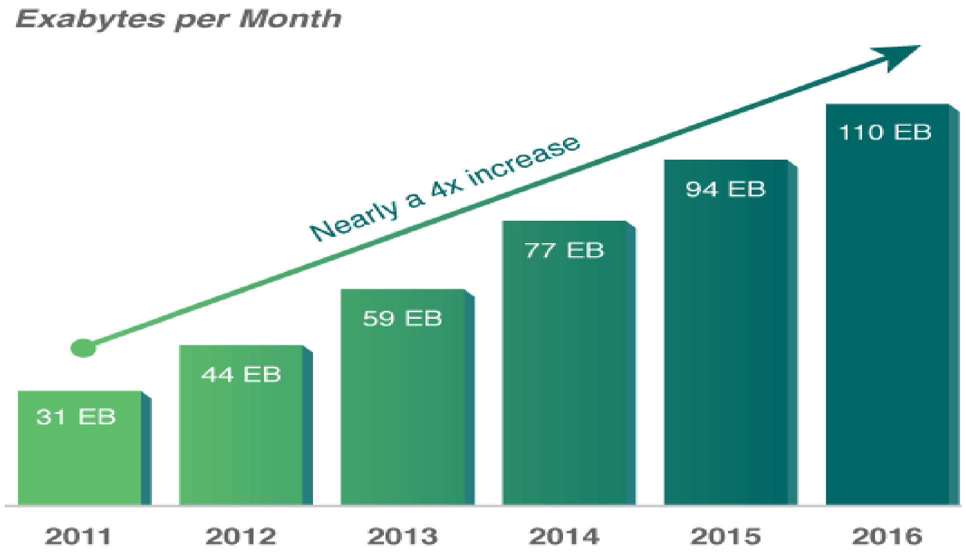
List of Figures

1 Introduction

1.1 The need for a new network architecture

As years go by, more and more people and companies from all over the world use electronic devices and software applications in order to exchange information or enjoy new services such as high definition TV, online gaming and online shopping. Future systems must be capable of accommodating the telecommunication traffic which doubles almost every two or three years as we can see in the figure below. Therefore, there is an increasing demand for high data rates that forces scientists and manufacturers to design new, advanced networks which should reliably perform close to channel's capacity.

Optical networks have become widely spread and have assumed a role as mission critical infrastructures for our information society and plans for further development are continually being made. The usage of optical fibers offer substantial advantages such as huge bandwidth, which leads to fast transmission times, immunity to electromagnetic interference in contrast to the Cu-wired or wireless networks and low attenuation (almost 0.5 dB/km [1]) and thus successful interconnections in very long distances (~10000km [1]). Furthermore, the cables are quite thin, light and flexible with low production cost as they are made of silica, a cheap material that can be easily found. In addition, the operation of optical networks is cost-efficient as the low power loss minimizes both the number of amplifiers needed and the transmitter's power required. Information about technical and physical characteristics of optical fiber systems are extensively discussed in [1] [2] [3].



Cisco VNI Forecasts 110 Exabytes per Month of IP Traffic in 2016

Source: Cisco VNI, 2012

Figure 1.1.1 IP Traffic forecasts for the next years by Cisco [4]

Traffic patterns have changed significantly. In contrast to client-server applications where the bulk of the communication occurs between one client and one server, today's applications access different databases and servers, creating a flurry of “east-west” machine-to-machine traffic before returning data to the end user device in the classic “north-south” traffic pattern. Moreover, many enterprise data centre managers are contemplating a utility computing model, which might include a private cloud, public cloud, or some mix of both, resulting in additional traffic across the wide network area. In addition, personal mobile devices such as smartphones, tablets and notebooks have powerfully entered the internet world providing users the chance to access easily the internet anywhere at any moment and thus, the required bandwidth has increased [5].

Existing network architectures were not designed to meet the requirements of today's users, enterprises and carriers as their static nature is in stark contrast to the dynamic nature of today's server environment. Many limitations [5] are imposed such as complexity, inconsistent policies, inability to scale, as well as inability to adapt when network conditions change. Unless new architectures are proposed, we will face capacity crunch in the not so distant future.

1.2 Elastic – Dynamic networks

The incremental improvement in transmission technology cannot meet the pace of traffic growth on its own. Considering that the capacity of conventional optical fibers is not limitless, a practical strategy for accommodating the ever-increasing traffic demands is to take advantage of synergetic effects [6], *i.e.*, continuous incremental innovation for higher capacity and spectral-efficiency-conscious networking evolution. Next generation networks should be capable of adapting their operation to the continually changing conditions and dynamically allocating the available resources. Proposals of elastic-dynamic optical network concepts are demonstrated in [6]. Comparisons between conventional and elastic networks are also made in this paper. Next figures show some implementation differences between the conventional static networks and elastic-dynamic networks.

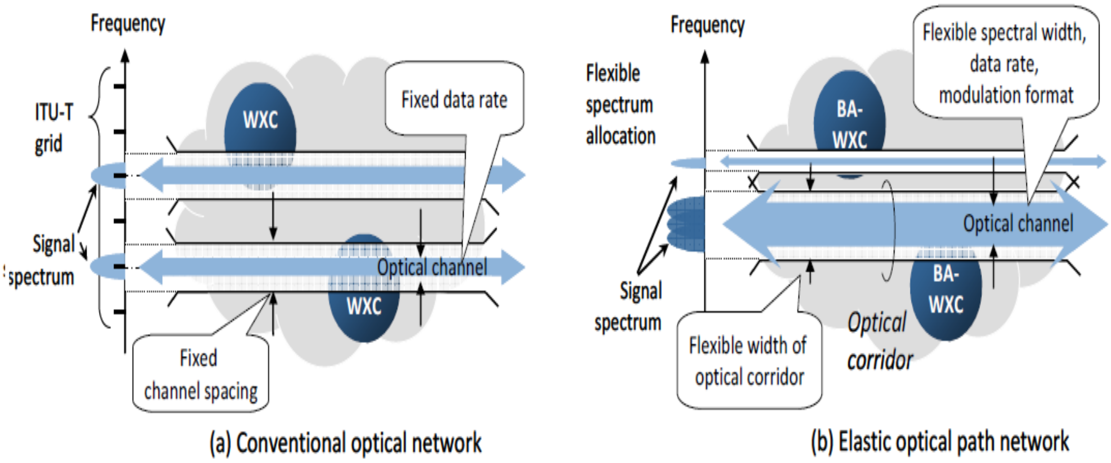

Frequency
ITU-T grid
Signal spectrum
WXC
Fixed data rate
Optical channel
WXC
Fixed channel spacing
(a) Conventional optical network
Frequency
Flexible spectrum allocation
Signal spectrum
BA-WXC
Flexible spectral width, data rate, modulation format
Optical channel
BA-WXC
Flexible width of optical corridor
Optical corridor
(b) Elastic optical path network

Figure 1.2.1 Elastic optical path network concept and explicit designation of optical corridor [6]

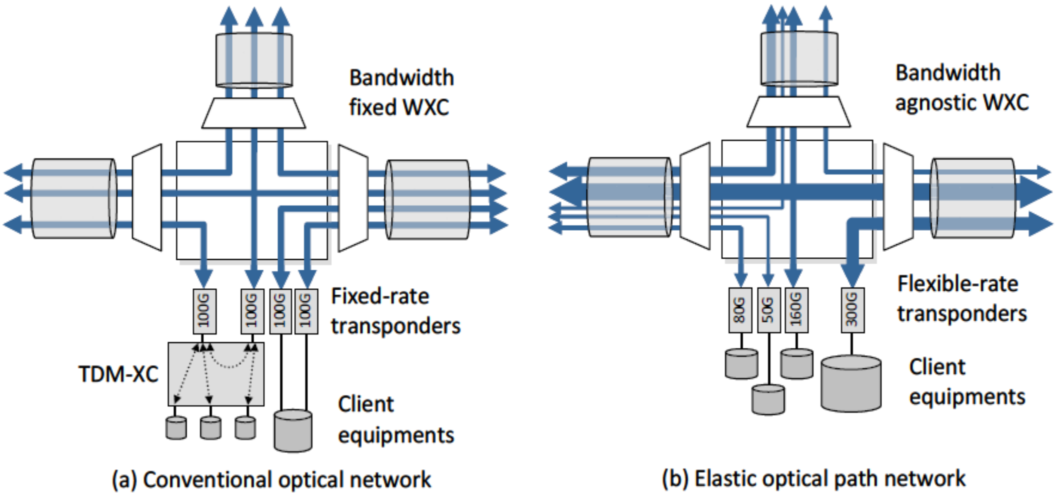


Figure 1.2.2 Rate adaptive spectrum allocation concept [6]

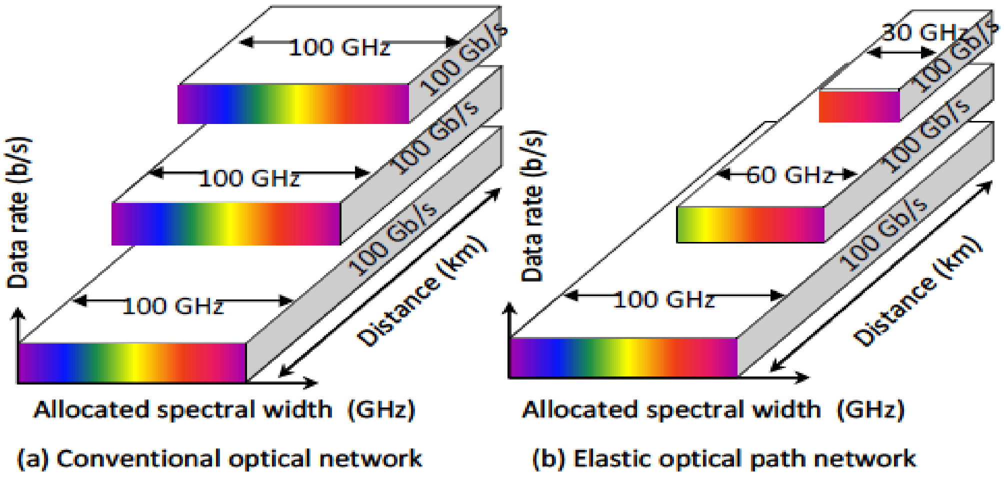


Figure 1.2.3 Distance adaptive spectrum allocation concept [6]

The network utilization efficiency in existing optical networks is limited due to the rigid nature of the networks. The available bandwidth per wavelength is fixed regardless of the capacity requirements of the clients. So, in the case that the total client traffic volume is not sufficient to fill the entire capacity of the wavelength, we cannot reach the maximum throughput. Current optical systems mitigate this issue by aggregating and grooming low-bit rate data flow with electrical time division multiplexing cross-connectors (TDM-XCs) or packet transport switches as shown in Figure 1.2.2(a), resulting in raised power consumption and extra financial cost. Moreover, in the case that the total client traffic volume is larger than that the wavelength can handle, great delays occur and the spectral efficiency is low, even if other channels are idle. In contrast, dynamic systems can accommodate this traffic by dynamically allocating the available resources to the various network points according to each point's requirements, achieving in this way high spectral efficiency.

Another significant advantage of the dynamic networks is the adaptability to the physical conditions on the route. The maximum length of the link is physically limited by chromatic dispersion [1][2][3] and nonlinear effects caused by the usage of the optical repeaters. As the distance between two nodes increases, the spectral width of the channels should decrease for a satisfying enough transmission. In conventional networks all links use the same spectral width (Figure 1.2.3(a)) which is appropriate only for the longest connections resulting in inefficient operation for the shorter ones whereas elastic networks allocate the available bandwidth according to each link's length (Figure 1.2.3(b)).

Furthermore, dynamic systems are also capable of adapting their operation to the varying noise conditions. For example, for high noise values it is recommended to use lower modulation formats in order to maintain the bit error rate low. Dynamic networks do adapt the modulation format to the existing noise in contrast to the static systems where the format remains constant.

Last, it is important to mention that elastic networks offer also the advantage of automatically diagnose and overcome link failures. When a link failure occurs and the detour route cannot provide sufficient capacity, network can reallocate the available resources to achieve the minimum acceptable quality of service.

1.3 Optical performance monitoring

Optical performance monitoring (OPM) is the procedure, thanks to which the dynamic network can check the condition of the links and adapt its operation whenever is needed. A successfully designed flexible optical network must automatically monitor and isolate all degrading effects as well as nimbly take actions to optimally allocate the resources in order to satisfy users' expectations. Some interesting monitoring techniques are discussed in [7] and [8].

Most of the monitoring techniques aim to estimate some of the following parameters:

i) **Signal to noise ratio (SNR)**

Signal-to-noise ratio (often abbreviated SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. Mathematically is defined:

$$SNR = \frac{Signal\ Power}{Noise\ Power}$$

or

$$SNR = 10 * log_{10}(\frac{Signal\ Power}{Noise\ Power}) \qquad \text{(dB)}$$

ii) **Bit error rate (BER)**

The bit error rate represents the number of the average number of bit errors occuring during a transmission divided by the total number of bits. It is the most common metric used to verify the quality of the transmission and evaluate the performance of the network.

iii) **Quality of service (QoS)**

The quality of service (QoS) refers to several related metrics of telecommunication networks that allow the transport of data with special requirements. Such metrics are the BER and the SNR as well as the response time, the connection failure probability etc.

iv) **Chromatic dispersion (CD)**

Chromatic dispersion is the phenomenon in which the different frequencies of the spectrum of the signal propagate with different velocities, due to the fact that the refractive index depends on the wavelength, causing pulse broadening in the time domain and thus, bit overlaps.

v) **Polarization mode dispersion (PMD)**

The polarization mode dispersion is a phenomenon in which the two orthogonal polarizations of the light propagate with different velocities in the optical fiber. Degradations imposed by PMD are similar to those imposed by CD. More details about PMD and CD can be found in [1][2]

OPM techniques used in current optical networks can be classified in two main groups. In the first one, electrical devices such as spectrum analyzers, are used to measure the previously mentioned parameters. Techniques of this group are not cost efficient and this can be huge drawback for budget limited systems. To maximize the benefit of the deployed equipment, the problem becomes a matter of finding the optimum locations for these devices.

Such problems do not exist in the techniques of the second group, the operation of which is relied on the digital signal processing (DSP) that takes place at receiver's edge. Signals are distorted due to the imperfections of the link and so, we can estimate link's condition by simply measuring this distortion. OPM techniques with DSP operate cost efficiently because no additional electrical device is required. As electronic technology is more mature than the optical technology, the DSP is electronically integrated in the receiver's side. However, it is estimated that in the future, DSP could also be integrated in the optical domain, allowing the derivation of relevant information at the mid-points of a network as well, for example at optical amplifiers.

1.4 Objectives

In this report, the main objectives are to:

a. Develop an algorithm which will estimate the existing signal to noise ratio (SNR) of a link using DSP techniques. The accuracy of the algorithm will be statistically evaluated at the end of the relevant chapter.(Chapter 4).

b. Experimentally examine and monitor the performance of an optical network with varying noise conditions and various transmission parameters (Chapter 5). Interesting deductions will be made during the chapter.

For both objectives, it is assumed that the optical network use quadrature amplitude modulation (QAM) formats for the transmission, especially 4-QAM, 16-QAM, 64-QAM, 256-QAM. Details about digital modulations are extensively described in chapter 2.

2 Digital Modulation Formats

Digital technology offers a variety of significant advantages such as high reliability, flexibility, simple multiplexing, capability in diagnosing and correcting errors, compatibility with other digital networks as well as signal processing storage and reuse. Furthermore, a digital signal is less susceptible to the noise than an analogue one.

In the next paragraphs we will show and compare some of the most common digital modulation techniques.

2.1 Most common formats – Quick overview

a) **Amplitude Shift Keying (ASK)**

Transmitted mathematical form of the transmitted signal is given in the following equation:

$$s(t) = Am(t)\cos(\omega_c t + \varphi) \qquad (1)$$

Information is encoded in function m(t) which is:

$$m(t) = \begin{cases} 1, & if\ current\ bit\ is\ 1 \\ 0, & if\ current\ bit\ is\ 0 \end{cases} \qquad (2)$$

The waveform a random ASK signal is illustrated in the next figure.

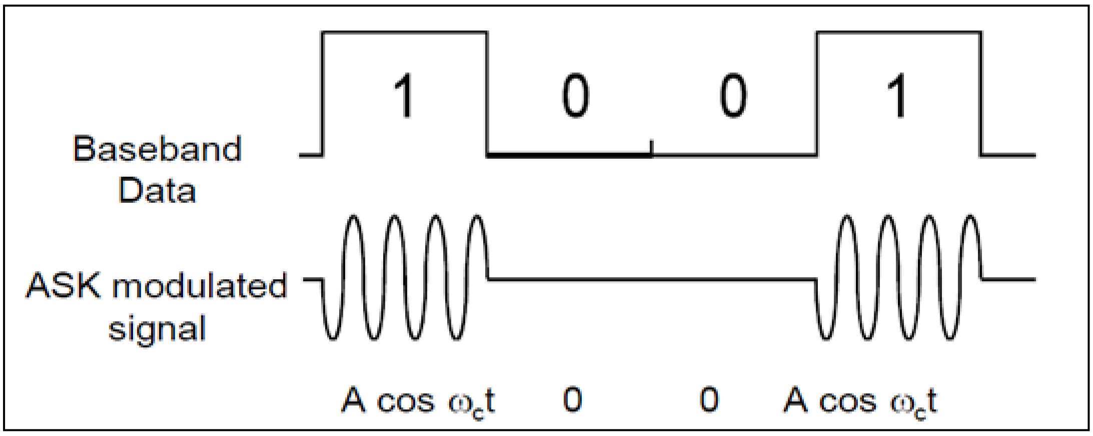


Figure 2.1.1 ASK signal waveform [9]

This is the binary ASK (1bit/symbol) modulation and it is well known as On-Off Keying (OOK). There are also the M-ary ASK modulations in which there are more than 2 levels of amplitude.

Despite the fact that ASK is characterized by simplicity and low-cost implementation, it demonstrates poor performance, as it is heavily affected by noise and interference. In addition, the fact that it is not a spectral efficient technique, makes it appropriate only for networks with low data rates and short distances.

b) **Frequency Shift Keying (FSK)**

In this scheme, data are transmitted through fluctuations of the carrier frequency. There are at least two frequencies. Signal's waveform is:

$$s(t) = \begin{cases} Acos[(\omega c + \Delta\omega)t + \varphi], & if\ current\ bit\ is\ 1 \\ Acos[(\omega c - \Delta\omega)t + \varphi], & if\ current\ bit\ is\ 0 \end{cases} \tag{3}$$

The waveform of a random FSK signal is shown in the following figure.

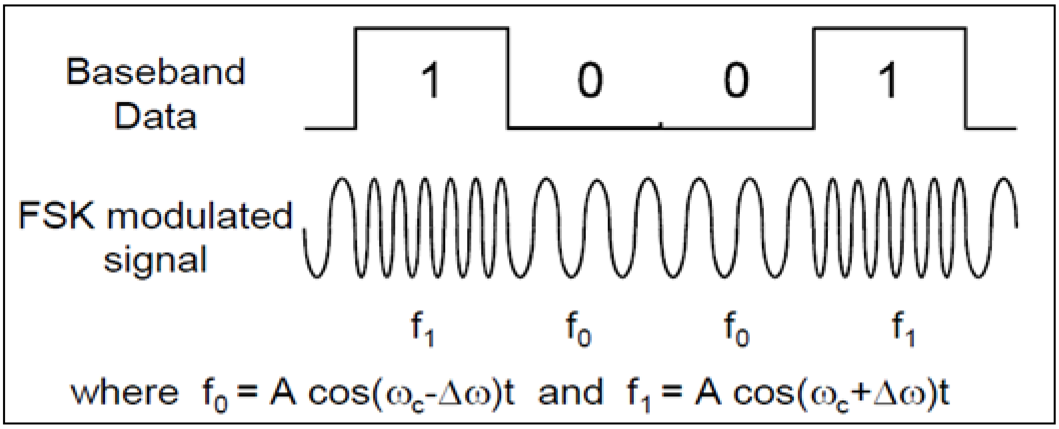

Baseband Data
1
0
0
1
FSK modulated signal
f_1
f_0
f_0
f_1
where $f_0 = A\cos(\omega_c-\Delta\omega)t$ and $f_1 = A\cos(\omega_c+\Delta\omega)t$

Figure 2.1.2 FSK signal waveform [9]

Equation 3 describes the 2-FSK modulation where every symbol conveys only one bit. In fact we can encode data in symbols with $\log_2 M$ bits per each by using M-FSK (M>1). For example a 4-FSK signal is illustrated in the next plot:

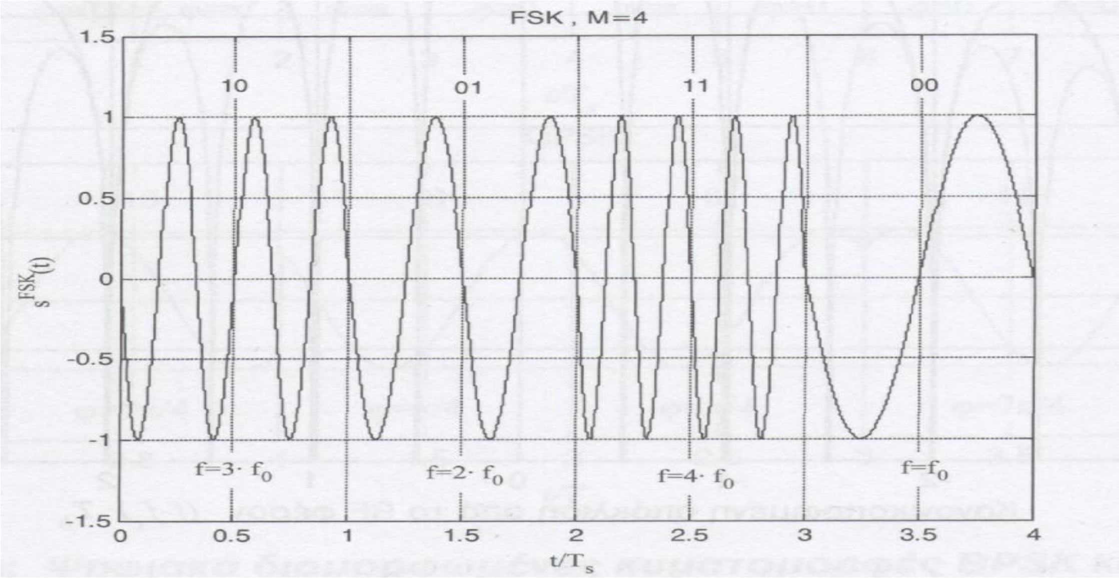


Figure 2.1.3 4-FSK signal waveform [10]

While M increases, the Bit Error Rate of MFSK decreases, but on the other hand the spectral usage becomes less efective because more freequency states are needed (Figure 2.13). The bandwidth efficiency for the MFSK is equal to $\log_2 M/(2M)$-decreasing function. Both FSK modulator and demodulator can be simply and inexpensively implemented.

c) **Phase Shift Keying (PSK)**

In PSK, information is included in signal's phase changes while the amplitude and the frequency of the waveform remain constant. The simplest PSK modulation is the BPSK one (Binary PSK) and its mathematical form is:

$$s(t) = A\cos(\omega_c t + \pi/2*(1-m(t))) \qquad (4)$$

where m(t) represents the information signal converted in bits.

A BPSK modulated signal is shown in the next figure where we can easily observe the phase's shift every when two consecutive bits are different.

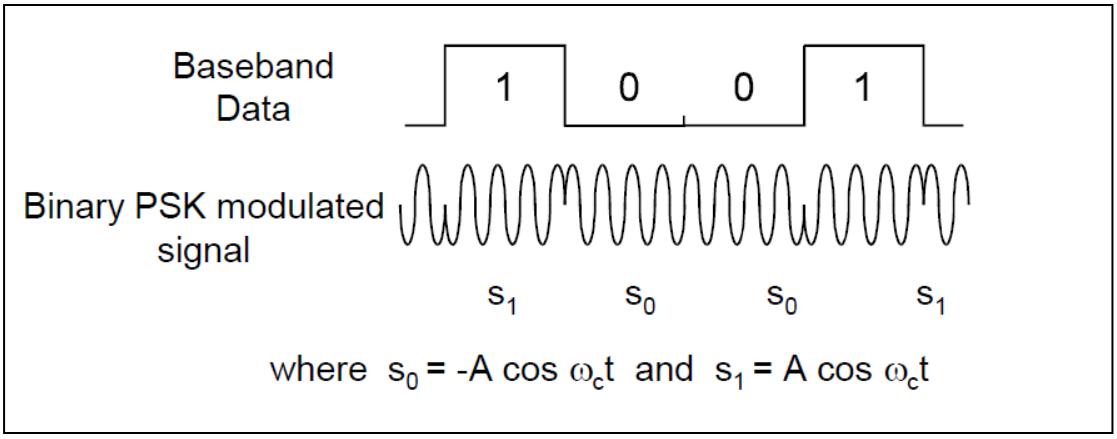

Baseband Data
1
0
0
1
Binary PSK modulated signal
s_1
s_0
s_0
s_1
where $s_0 = -A\cos\omega_c t$ and $s_1 = A\cos\omega_c t$

Figure 2.1.4 PSK signal waveform [9]

PSK can be effectively expanded to a M-ary scheme, employing multiple phases as different state and so we have the MPSK modulated signals described by the next general equation.

$$s(t) = A\cos(\omega_c t + \theta_k(t)) \qquad (5.1)$$

$$\theta_k(t) = 2k\pi/M + \pi/M \text{ , } k = 0,1,....M-1 \qquad (5.2)$$

In MPSK every symbol conveys $\log_2 M$ bits whilst there is no need for wider spectrum, because in contrast with MFSK, no more frequencies are needed. Therefore, we can obtain $\log_2 M$ times higher spectral efficiency. For example, in BPSK spectral efficiency is equal to 0.5bps/Hz while in QPSK (4-PSK) spectral efficiency is equal to 1bps/Hz. Hence, we can either double the data rate while given the same bandwidth or 50% reduce the required bandwidth without decreasing the data rate. Unfortunately, there is a Bit Error Rate penalty when we use higher-order schemes, because the phase margin of each state becomes smaller as we can observe in the constellation diagrams shown in next page, where in QPSK the angle space between two **successive points is 90° whereas in 8-PSK it is 45°.**

Next plot shows a QPSK waveform. In a symbol's period, two bits are transmitted concurrently.

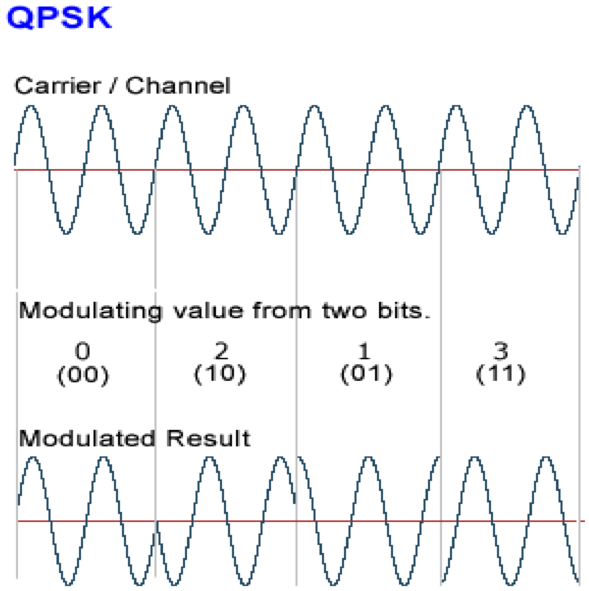


Figure 2.1.5 QPSK signal waveform [11]

There is a special kind of polar diagrams for plotting PSK signals called "**Constellation diagrams**" that helps us to examine different PSK-network performances. PSK signals can be expressed in a polar form with constant amplitude A :

$$s(t) = A*\mathrm{Re}\{e^{j\theta}\} \qquad (6)$$

(Information is involved in variable angle θ).

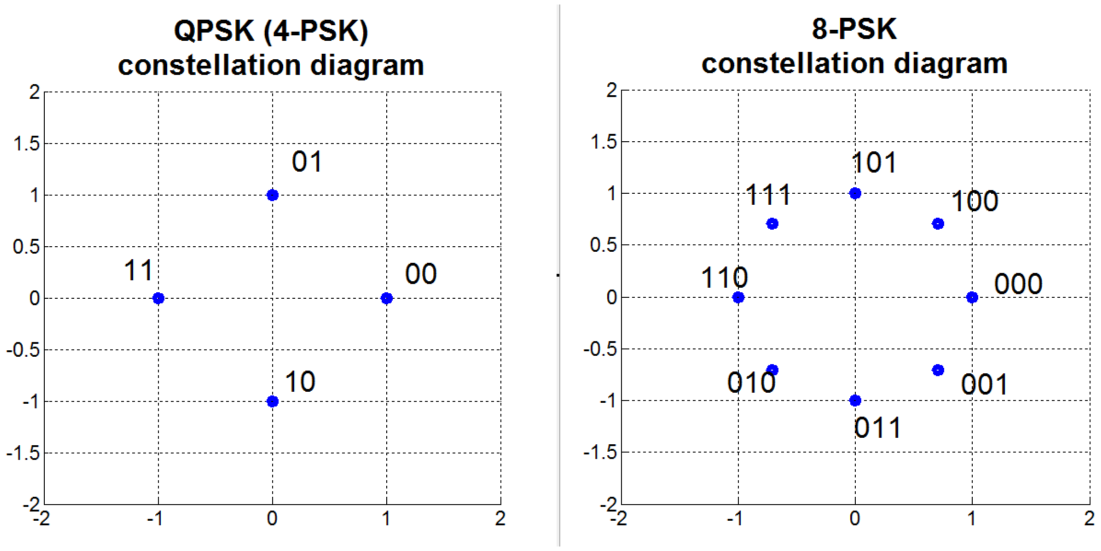


Figure 2.1.6 QPSK and 8PSK constellation diagrams

Each point mapped on the constellation diagram represents one symbol and each symbol is associated with one unique phase. For example, in the 8-PSK constellation diagram, the point with angle 135° represents the symbol that carries the 3-bit information 101. In this report, we are going to use extensively the constellation diagrams.

d) *Quadrature Amplitude Modulation (QAM)*

Quadrature Amplitude Modulation combines ASK and PSK techniques to transmit effectively more bits per symbol and each symbol is represented **by a particular combination amplitude-phase**. QAM technology is used in bandwidth limited channels, where it is beneficial to trade off error rate for gaining fruitful spectral utilization.

In this report we will examine systems with M-QAM modulations so it is important to mention some of the QAM fundamentals. We are going to discuss about squared QAM (even number of bits/symbol) formats because they can be implemented more easily than the star QAM (odd number of bits per symbols).

i)QAM Signal Generation

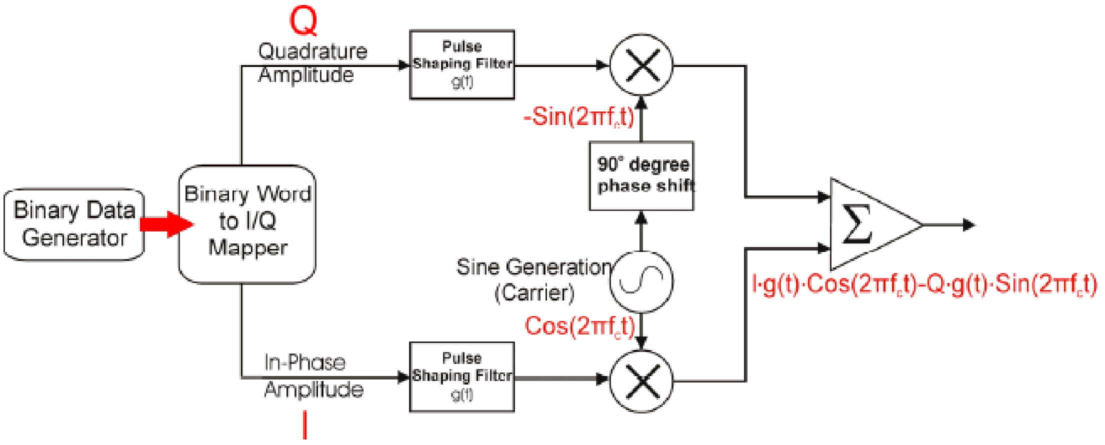

Q
Quadrature Amplitude
Pulse Shaping Filter g(t)
-Sin(2πf_ct)
90° degree phase shift
Binary Data Generator
Binary Word to I/Q Mapper
Sine Generation (Carrier)
Σ
I·g(t)·Cos(2πf_ct)-Q·g(t)·Sin(2πf_ct)
Cos(2πf_ct)
In-Phase Amplitude
Pulse Shaping Filter g(t)
I

Figure 2.1.7 Illustrating QAM signal generation

Information data are divided in two separated sequences. Then, the one sequence is translated to in-phase amplitude I, and the other to the quadrature amplitude Q. After pulse shaping, these sequences are modulated on the sinus and cosinus carriers respectively, and finally summed at the output.

ii) QAM signal waveform

The mathematical form of a QAM signal is:

$$s(t) = Ip(t)\cos(\omega_c t) + \; Qq(t)\sin(\omega_c t) \quad (7)$$

where p(t), q(t) the pulse waveforms. Usually p(t) = q(t) = g(t). Thus we have

$$s(t) = Ig(t)\, cos(\omega_c t) + \; Qg(t)\, sin(\omega_c t) = \; Re\{(I \; + \; jQ)g(t)e^{\,j\omega_c t}\}$$

Setting :$\sqrt{I + jQ} \; = A$,

$$\arctan\left(-\tfrac{Q}{I}\right) = \theta \;,$$

the last equation is transformed to:

$$s(t) = Re\{Ae^{\,j\theta} g(t)e^{\,j\omega_c t}\} = Re\{Ag(t)e^{\,j(\omega_c t+\theta)}\}$$

$$=Ag(t)Re\{cos(\omega_c t) + j\,sin(\omega_c t)\}$$

$$\rightarrow s(t) = \; Ag(t)\cos\,(\omega_c t + \theta) \qquad (8)$$

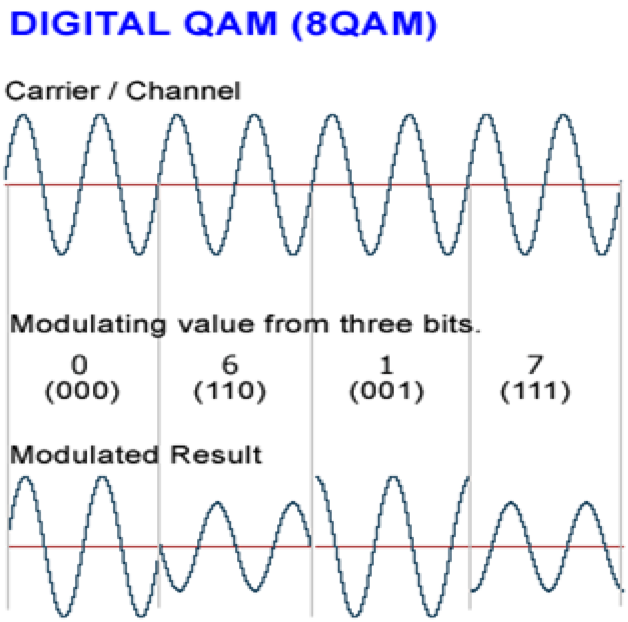


Figure 2.1.8 8-QAM signal waveform [11]

iii) QAM constellation diagram

It is convenient to view QAM symbols on a polar (or IQ) plot. Every symbol can be plotted to only one specific point on the constellation diagram determined by its particular amplitude-phase combination as drawn in the next picture:

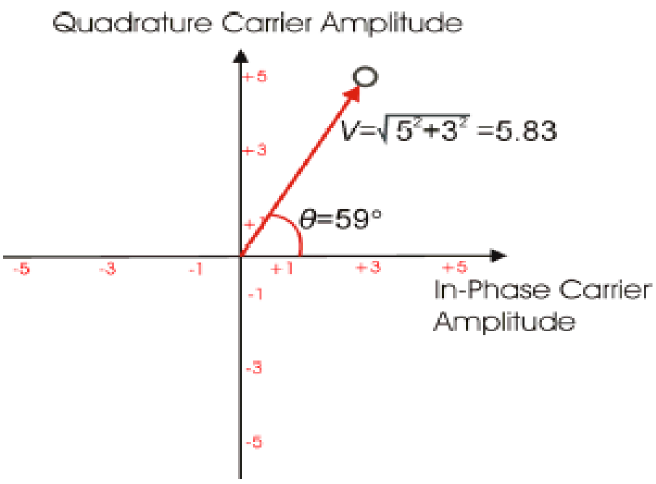


Figure 2.1.9 Illustrating a single symbol from an M-QAM system plotted in the IQ plane [12]

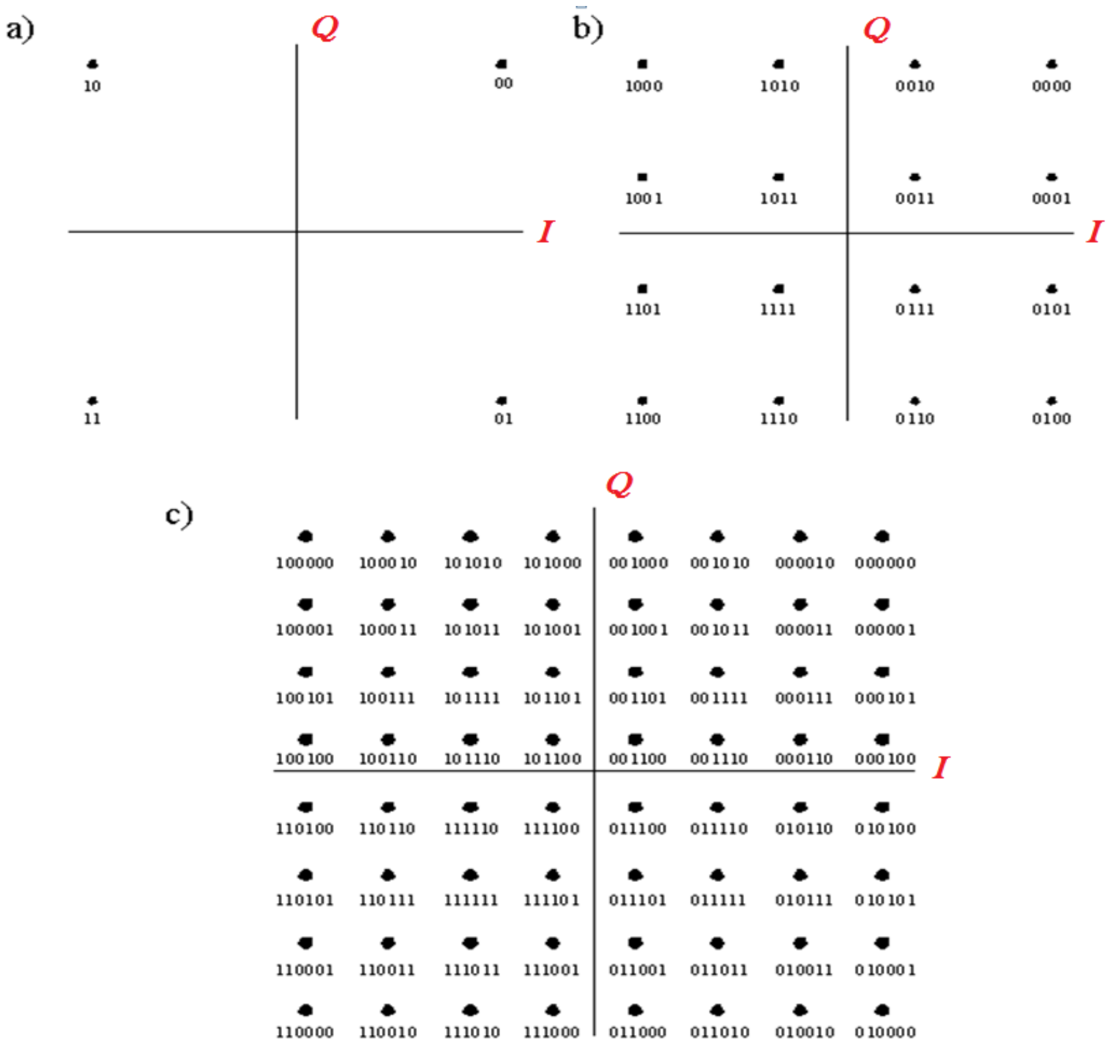


Figure 2.1.10 Constellation diagram with Gray encoding for a) 4-QAM, b) 16-QAM, c)64-QAM [13]

2.2 Comparisons

Higher-level QAM modulations are more susceptible to noise effects because the adjacent points of the constellation diagram “approach” each other, and so, the Noise tolerance interval becomes smaller (Figures 2.11 2.12). The Symbol Error rate (SER) as well as the Bit Error rate (BER) are given in [1]:

$$SER^{max} \approx 2Erfc\sqrt{\frac{1.5*SNR}{(M-1)}} \quad , \qquad (9)$$

where Erfc is complementary error function :

$$Erfc(x) = \frac{1}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \qquad (10)$$

For SNR≫1 and Gray encoding (Adjacent symbols differ by only one bit):

$$BER^{max} \approx \frac{SER}{log_2 M} \qquad (11)$$

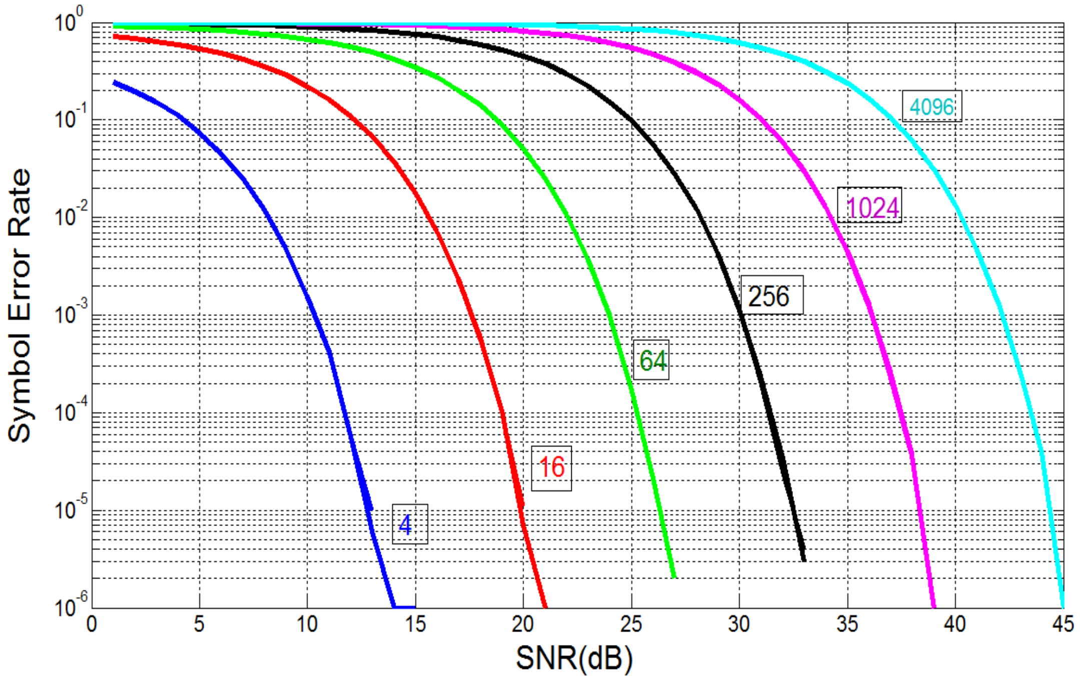


Figure 2.2.1 Symbol error rate for M-QAM schemes

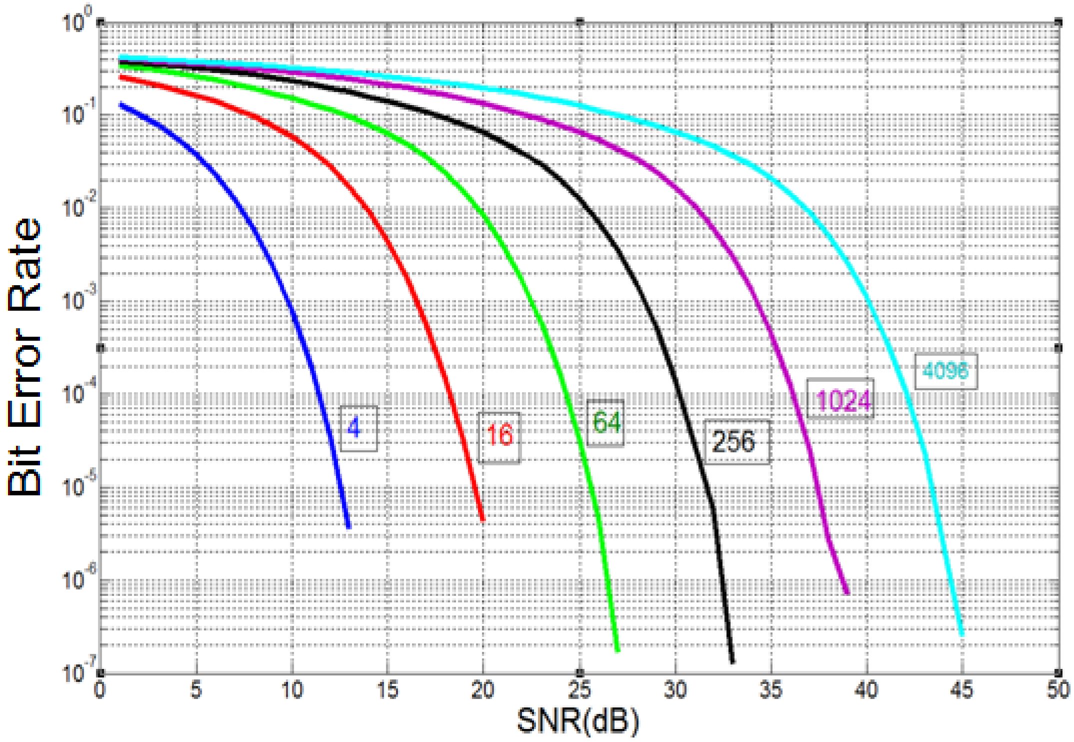


Figure 2.2.2 Bit error rate for M-QAM schemes

It can be readily observed that as M grows, greater signal to ratio values is needed to achieve the same BER or SER. For example, the required SNR to achieve BER<10^{-3} by using 16-QAM is about 17dB whereas for 64-QAM the SNR needed for the same goal is 23dB. This is a huge drawback for networks limited by power constraints.On the other hand, as far as the spectral efficiency is concerned, it can be increased by modulating multiple bits per symbol. Consequently, M-ary QAM techniques are more appropriate when there is a strict constraint for the used spectrum. Since required minimum bandwidth is equal to the Baud rate, we can reach the same bit-rate transmission by using less channel bandwidth.

Bandwidth efficiency (or normalized data rate), measured in bits/s/Hz, is simply the bit rate to bandwidth ratio (equation taken from [1]):

$$R/W = \log_2(M) \quad (12)$$

where R represents the bit rate and W the offered bandwidth.

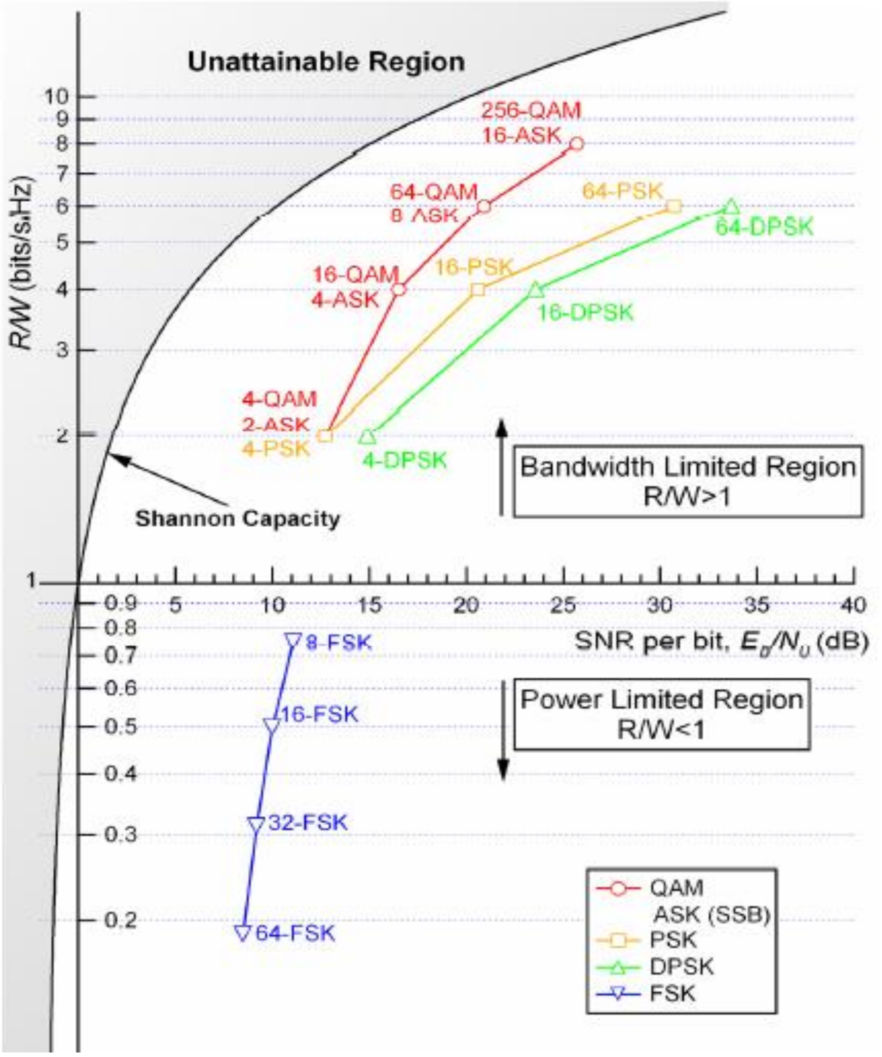


Figure 2.2.3 Comparison of several modulation schemes for a BER of 10^{-9} [12]

To sum up, we infer that:

1) QAM modulation is the most spectral efficient method
2) In QAM and PSK modulation, as M increases, so does the spectral efficiency in a lognormal way.
3) M-QAM and M-PSK reach the same spectral efficiency for each M but reduced minimum SNR is required for M-QAM to achieve the same BER with M-PSK.
4) Bit loading higher than 4-QAM causes a rapid growth in the OSNR penalty, while further increase in the launced signal power results in serious impairment due to nonlinear effects.
5) M-FSK is a completely spectral inefficient modulation while more bandwidth is needed to accommodate M-ary schemes. However, less power per bit is required while M increases. Thus FSK is more appropriate for power-limited

3 Error Vector Magnitude (EVM)

3.1 EVM definition

In a M-ary scheme, data are encoded in symbols with $\log_2 M$ bits per each before transmitted over the optical fiber link. Those symbols can be represented by points on constellation diagrams as described in the previous section. In an ideal situation, the constellation diagram of the transmitted data would be precisely the same at the two edges of the network. However, imperfections cannot be avoided and hence, there will be deviations between transmitter's and receiver's constellation diagrams.

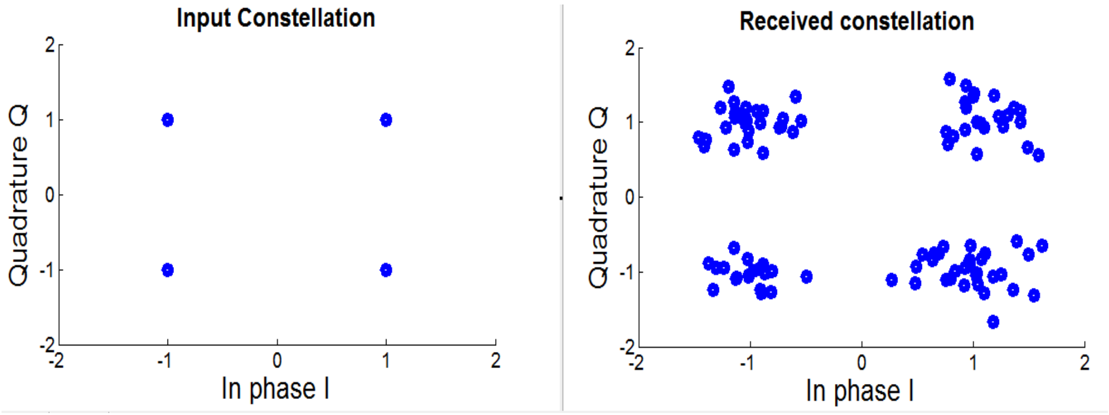


Figure 3.1.1 Input and output constellation diagrams for a 4-QAM, corrupted by Additive White Gaussian Noise (AGWN), transmission

Error Vector Magnitude(EVM) is a common performance metric for assessing the quality of communication. It describes the effective distance of the received complex symbol from its ideal position in the constellation diagram as drawn in the next picture.

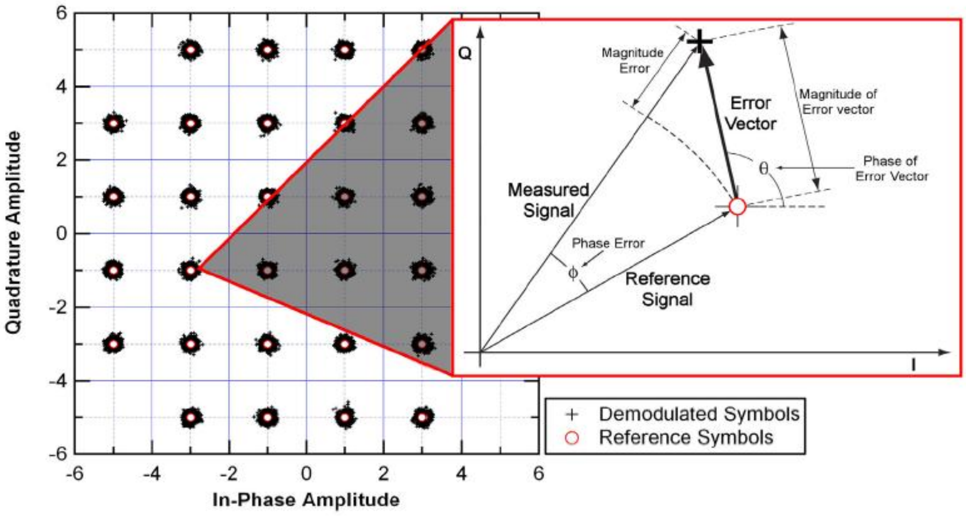


Figure 3.1.2 Illustrating the error vector and related quantities [12]

Mathematically, EVM can be expressed in the following two ways:

$$EVM_{max} = \sqrt{\frac{Error\ Vector\ Power}{Maximum\ inputed\ QAM\ Signal\ Power}} * 100\%$$

$$= \sqrt{\frac{\frac{1}{N}\sum_{i}^{N}\{(I_{i,in} - I_{i,out})^2 + (Q_{i,in} - Q_{i,out})^2\}}{\max\{I_{i,in}{}^2 + Q_{i,in}{}^2\}}} * 100\%$$

(3.1)

or

$$EVM_{avg} = \sqrt{\frac{Error\ Vector\ Power}{Average\ inputed\ QAM\ Signal\ Power}} * 100\%$$

$$= \sqrt{\frac{\frac{1}{N}\sum_{i}^{N}\{(I_{i,in} - I_{i,out})^2 + (Q_{i,in} - Q_{i,out})^2\}}{\frac{1}{N}\sum_{i}^{N}\{I_{i,in}{}^2 + Q_{i,in}{}^2\}}} * 100\%$$

(3.2)

It can be shown that $EVM_{max} = k * EVM_{avg}$ (3.3) where k is defined as the peak to average ratio.

3.2 SNR estimation using EVM

There are two EVM values to consider: the first corresponds to data-aided reception (sent data are known) and the other to nondata-aided reception (sent data are not known). To avoid confusion in remaining sections, we will refer to the first one as True EVM and to the second one as Blind EVM.

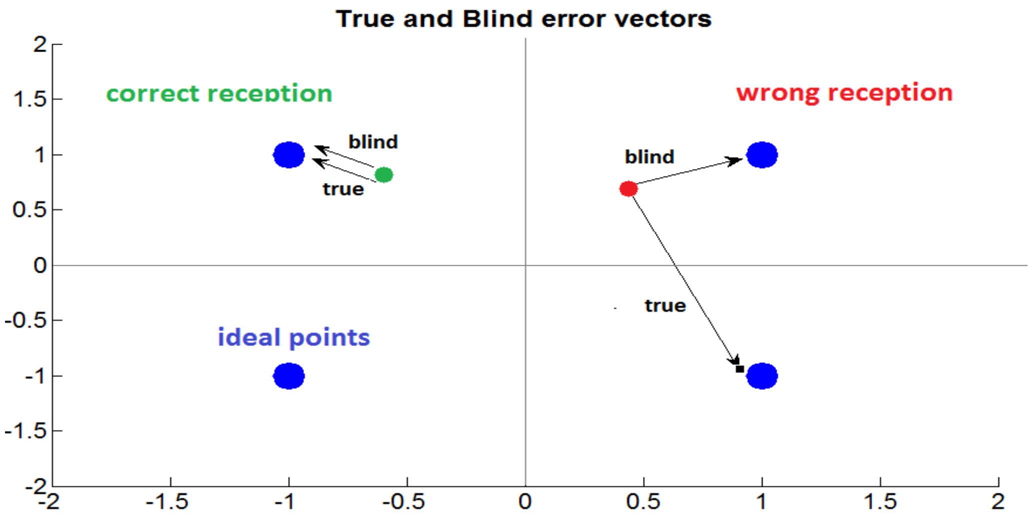


Figure 3.2.1 True and Blind error vectors

Regardless of the normalization method used (EVM$_{max}$ or EVM$_{avg}$), EVM can be related to the digital SNR. When normalized to the average symbol magnitude, assuming that only White Gaussian Noise corrupts the network, SNR can be estimated from EVM using the next formula:

$$SNR = \frac{E_s}{N_0} = log_2 M * \frac{E_b}{N_0} = \frac{1}{(EVM_{avg})^2} \qquad (3.3)$$

The plots bellow show the accuracy of the formula. We will use the Normalized Mean square error and the Absolute normalized bias for the evaluation. Error statistic metrics are described in appendix A.

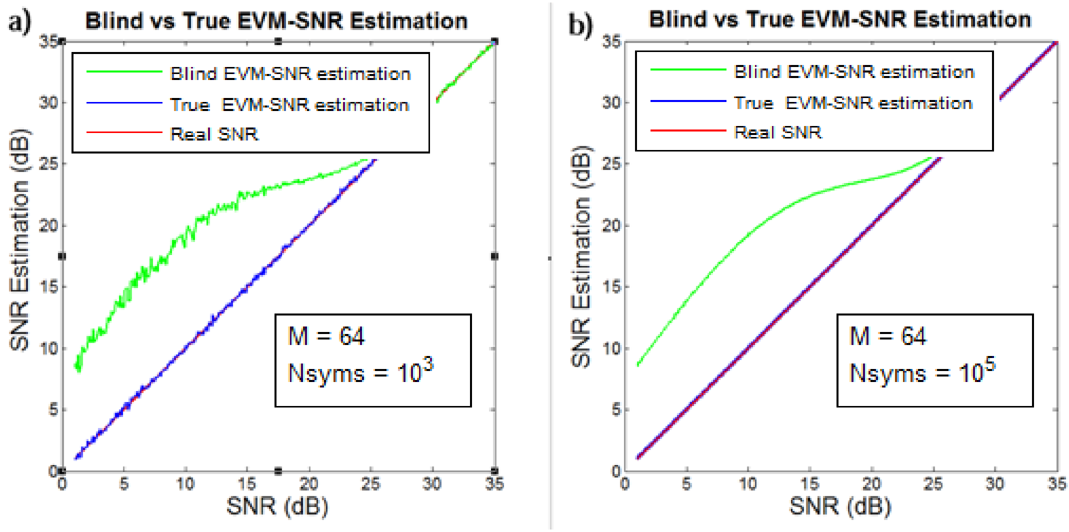


Figure 3.2.2 True(blue) vs Blind(green) EVM-SNR estimations. a) 64-QAM 10^3 symbols b) 10^5 symbols

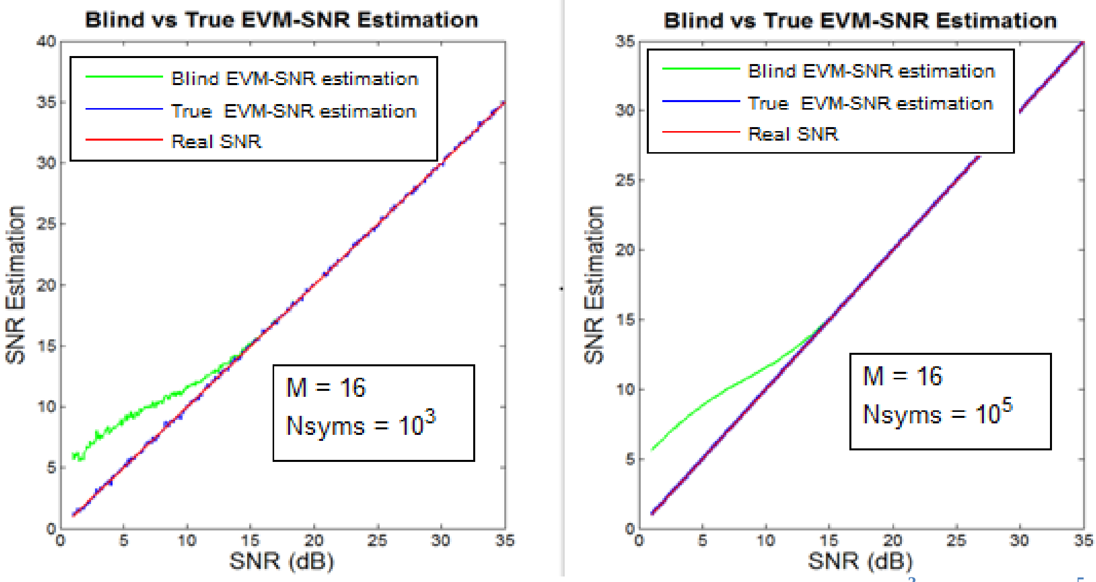


Figure 3.2.3 True(blue) vs Blind(green) EVM-SNR estimations. a) 16-QAM 10^3 symbols b) 10^5 symbols

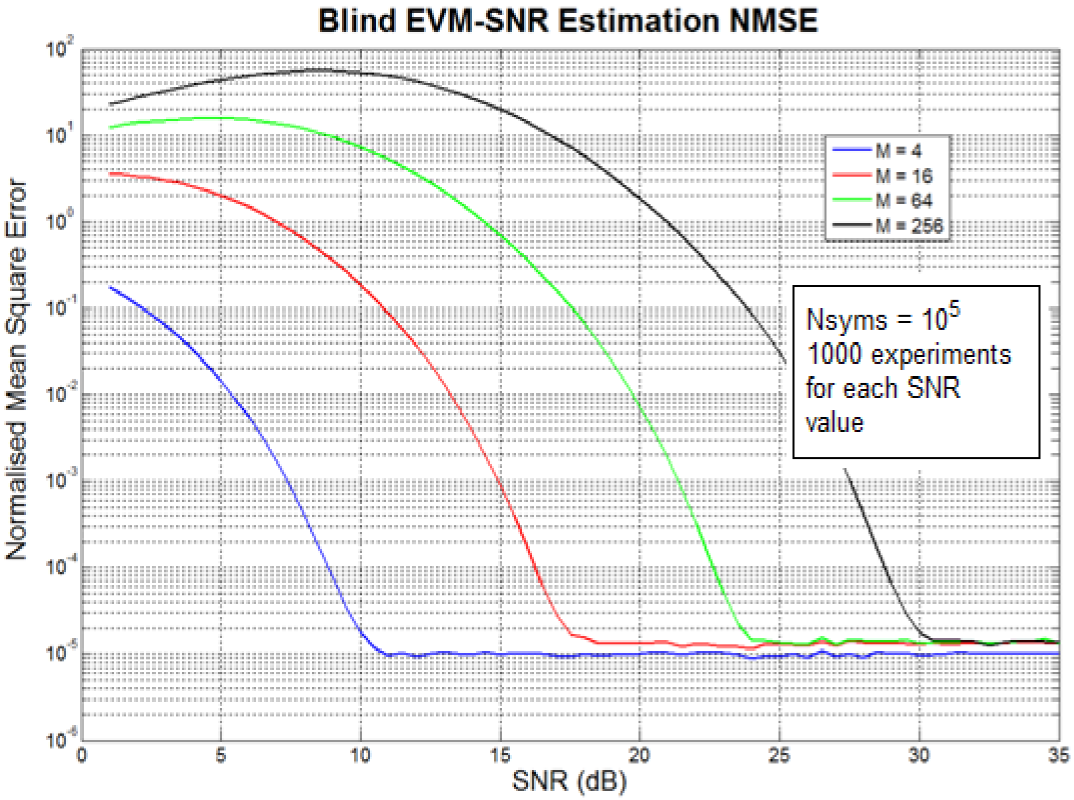


Figure 3.2.4 Normalized Mean Square Error of (3.3) formula's SNR estimations

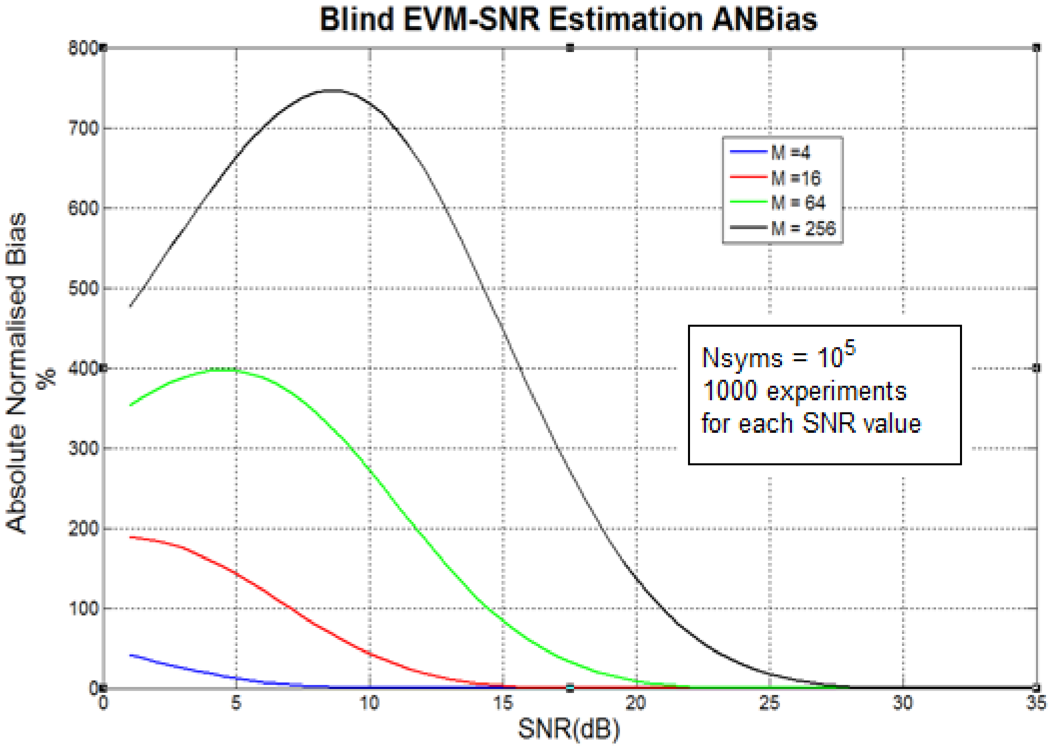


Figure 3.2.5 % Absolute Normalized Bias of (3.3) formula’s SNR estimations

Comments:

- The (3.3) Formula is extremely accurate for data-aided scenarios but as far as the nondata-aided scenarios are concerned, great deviations are observed. Figures 3.4 and 3.5 show that even when the number of transmitted symbols is 10^5, the estimations remain unacceptable for low SNR values.
- Blind EVM is virtually decreasing as M increasing because the receiver uses the closest ideal point to calculate the EVM whichever the real symbol is and so, the SNR is overestimated (figure 3.2.1 above).
- While SNR increases, there is a particular SNR threshold for every modulation over which the Blind EVM-SNR estimation is as much accurate as the True one. For example for M =16 this threshold's value is 16 dB and for M = 256 almost 27 dB. In addition, as M increases, higher SNR is required in order to estimate well enough the existing signal to noise ratio. This can be explained by the fact that, as M increases, the distances between the ideal points on the IQ plane become shorter and so symbol errors occur more often. In other words, data become more susceptible to noise effects.
- Larger quantity of transmitted symbols usually leads to more accurate estimations since the variance of the measured EVM diminishes. However, measurements with more symbols require more time.

It can be easily understood that real network's operation do not coincide with the data-aided scheme, so only the Blind EVM can be used to evaluate link's condition. A method that overcomes the above equation's problems , is tested in [7]. In the next chapter, an algorithm with which we can make satisfying estimations is proposed.

Finally, it is important to mention that Error Vector Magnitude can be also used for the BER or SER estimation as described in [14] and [15]. Despite the fact that we are not going to deal with BER or SER estimations in the current report, for complement reasons we show in the next figure the relation between SER and blind-EVM. It can be easily observed that for higher orders of QAM the EVM takes lower values but with higher symbol and bit error probability.

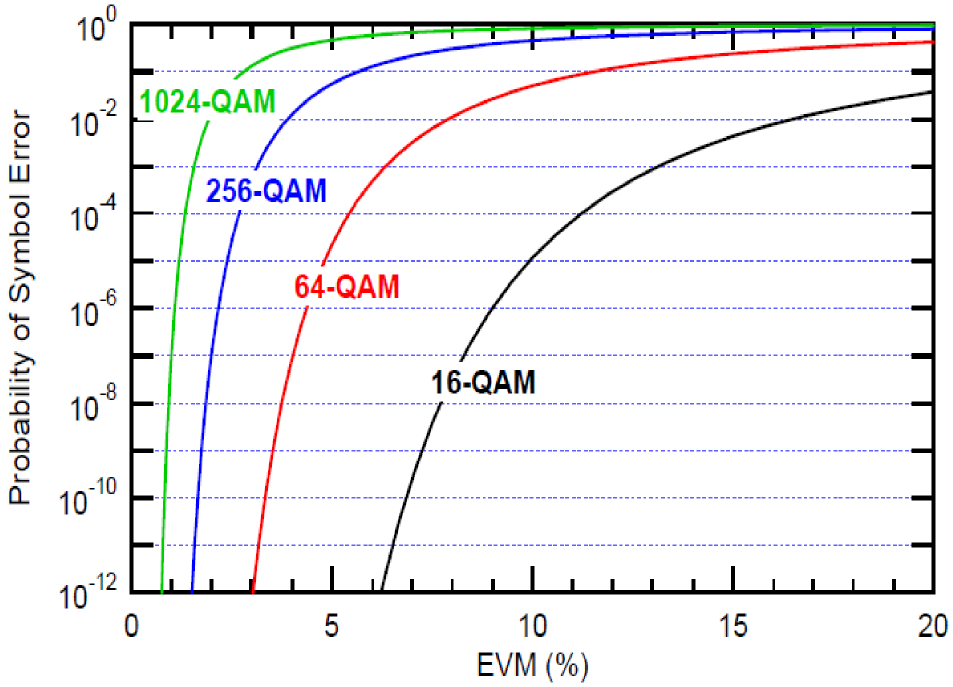


Figure 3.2.6 Probability of symbol error in QAM as a function of EVMavg [12]

4 Proposed Algorithm for Blind SNR Estimations

4.1 The route to reach the algorithm

As proved in the last chapter, the mathematical form (3.3) is not a sufficiently reliable formula to estimate signal to noise ratio when the real SNR is low. Therefore, it would be vital to develop a new algorithm that will estimate well enough the existing SNR by using the Blind Error Vector Magnitude. To achieve this goal, we notice two characteristic details of the graphical relation between SNR and Blind-EVM:

a) *Monotony*

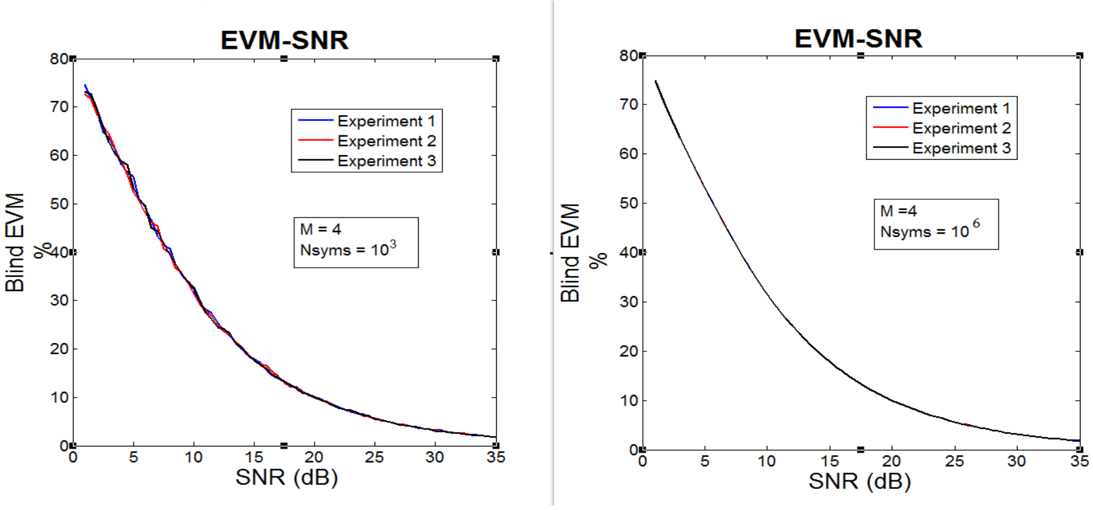


Figure 4.1.1 Blind EVM-SNR relation for 4-QAM 10^3 and 10^6 symbols

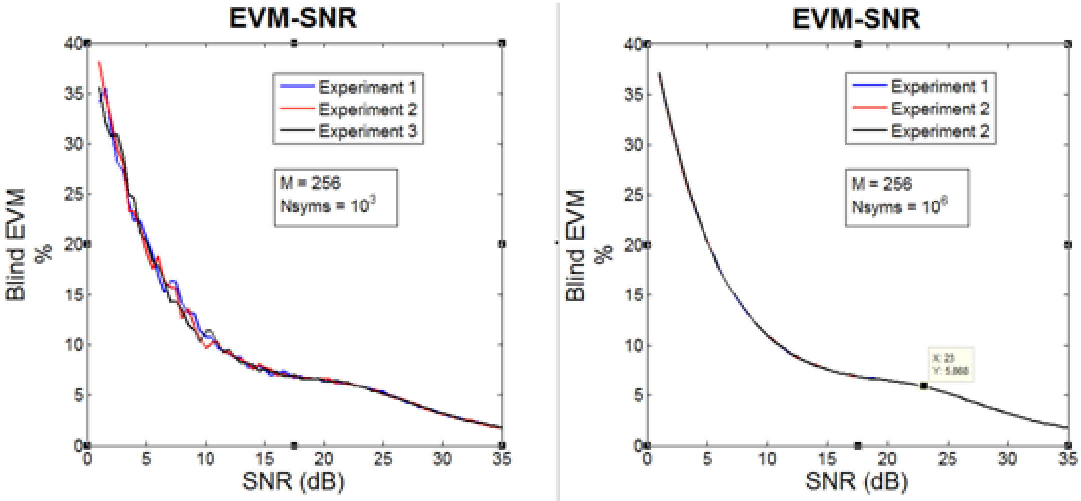


Figure 4.1.2 Blind EVM-SNR relation for 4-QAM 10^3 and 10^6 symbols

b) *Low Coefficient Of Variation (CV)*

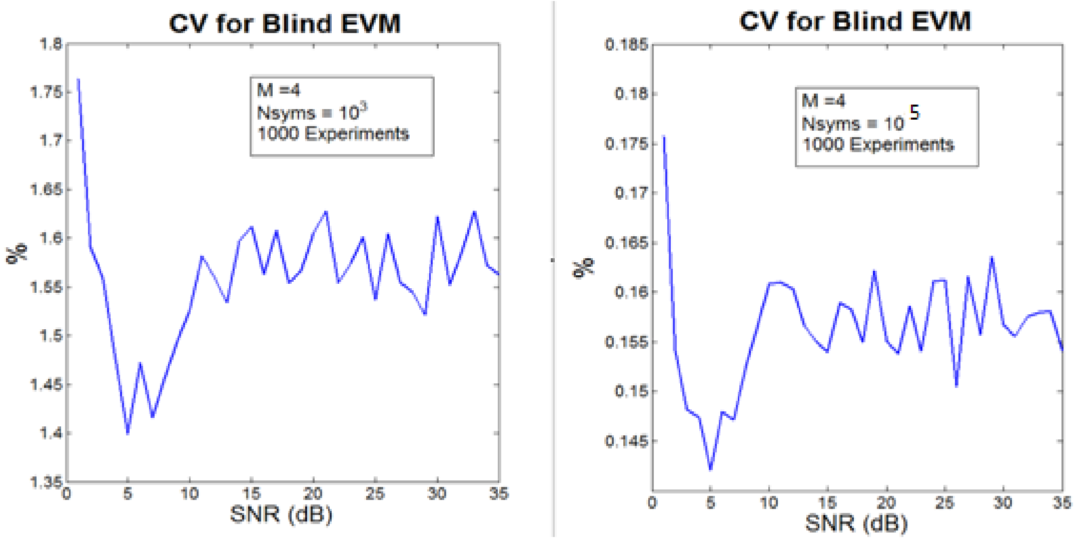


Figure 4.1.3 Coefficient of variation of Blind EVM for 4-QAM

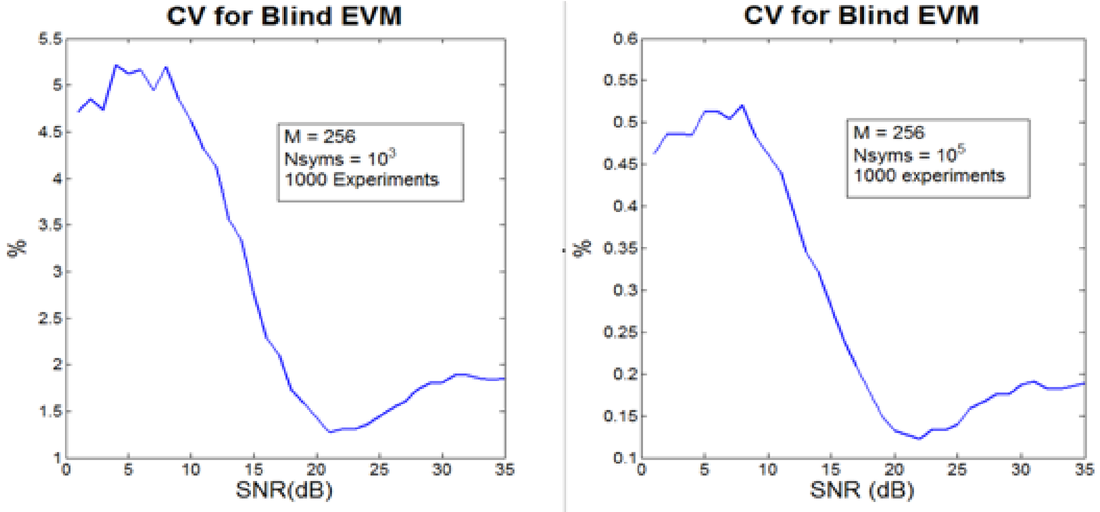


Figure 4.1.4 Coefficient of variation of Blind EVM for 256-QAM

From the plots above, we infer that if we transmit many symbols, we will extract strictly monotonic relation between SNR and EVM. Hence, there will be a strictly one to one correspondence between the two magnitudes. No EVM value can be derived by more than just one possible signal to noise ratios that differ a lot and no signal to noise ration can cause two totally different EVM values. Furthermore, we deduce that no matter how many times did we run the monte-carlo simulation program (appendix B program 1), we extract almost the same graphical relation (with a very small coefficient of variation). Combining those two clues (a,b), we finally decided to produce global, monotonic curves (EVM-SNR) and tables(“Look Up Table”) for every possible M that will effectively apply to all transmissions, regardless the number of transmitted symbols would be and so, we will be able to estimate the real SNR by just matching received blind EVM to the related saved in global tables SNR, without using any hardware device.

4.2 Description of the Algorithm

After noticing the previously mentioned details for the EVM and the SNR relation, we are ready to explain one by one the steps of how we create the global curves and finally how to estimate the existing SNR.

STEP 1: Creation of the global curves.

The more symbols will transmit the smaller CV we have as shown in figure (4.3). Thus, it would be beneficial to trade off some simulation time in order to create the most appropriate curve for each modulation. We are going to produce three curves for each scheme. One with 1dB step, one with 0.1dB step, and one with 0.01 dB step at the SNR axis. In the remaining pages of this thesis we will refer to those curves as LUT_1, LUT_01, LUT_001 respectively. We run five times the program 2 (appendix B) for 9,000,000 bauds (the largest number of bauds that the computer memory can handle) and finally we calculate the mean EVM value for each SNR. Next figure shows the LUT_01 curves for M = 4 and M = 256

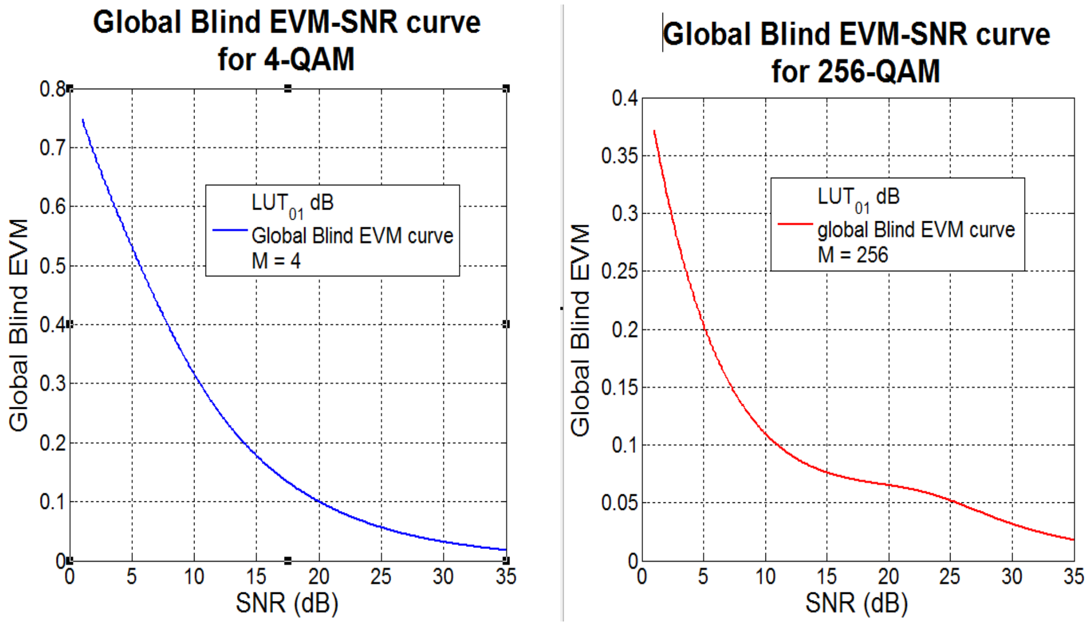


Figure 4.2.1 Global Blind EVM-SNR curves for 4-QAM and 16-QAM

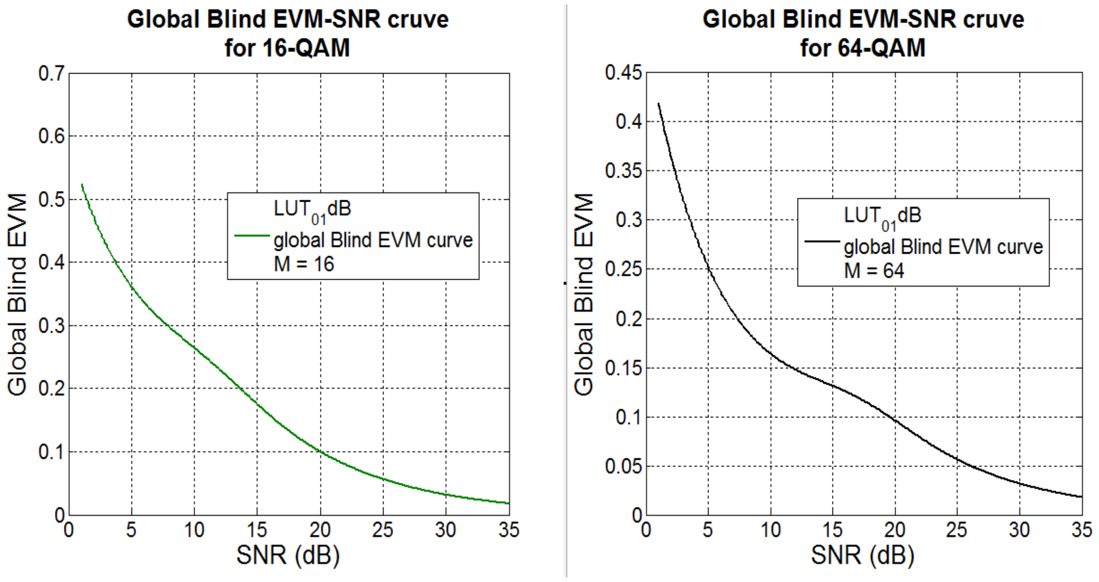


Figure 4.2.2 Figure 4.2.1 Global Blind EVM-SNR curves for 16-QAM and 64-QAM

STEP 2: Transformation to a strictly monotonic curve if the desired monotony is not achieved

For the curves stepped by 1dB and 0.1 dB this specific step can be avoided as they are already strictly “1-1” but for the LUT_001 this is an essential action because the created function is not strictly decreasing as desired. However, we can transform it to a strictly monotonic one without making significant errors. To achieve this, anywhere that there are **succesive** SNR values that give the same EVM we keep only the central one deleting all the others as illustrated in next picture and according to the algorithm that follows:

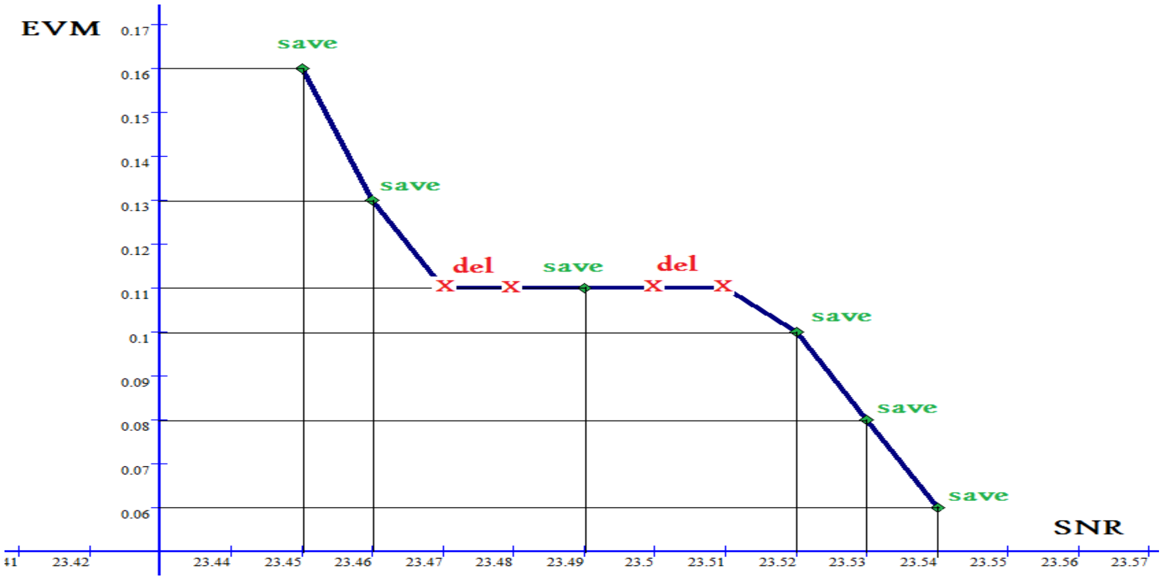


Figure 4.2.3 Illustrating how we transform to strictly monotonic a simly monotonic curve

```
If (number of SNR value that have the same EVM is n) then
        save only the round(n/2);
        ignore the others;
    end
```

For example, if we have five SNR values (SNR_{i-2} SNR_{i-1} SNR_i SNR_{i+1} SNRi$_{+2}$) we only save the SNR_i. The maximum additive error of this procedure is $(10\text{^}^{(0.02/10)} -1) *100\% = 0.46\%$.

STEP 3: EVM-SNR patterns storage

Now that the global monotonic functions have been created, we have to store all EVM-SNR matches using two tables ("Look Up Table") so that they can been accessed every time we need to make an estimation.

STEP 4: Estimate the Signal To Noise Ratio

Every time a transmission comes to its end, a blind EVM value is calculated at the receiver's edge. We seek this value in the stored global Blind-EVM table and with a simple matching to the SNR table we find the related SNR. Graphically, we try to find the projection of the calculated EVM on the SNR axis. Usually, the received EVM would not match precisely with any of the values included in the global tables so we can estimate the SNR with linear interpolation (can be easily programmed in a high-level program language such as JAVA). Although the EVM-SNR curve is polyonymal and not linear, the adjacent points at the SNR axis differ so little that very small errors are imposed with the linear interpolation. We can also just round the EVM to the nearest existing value of the EVM table and seek the related SNR.

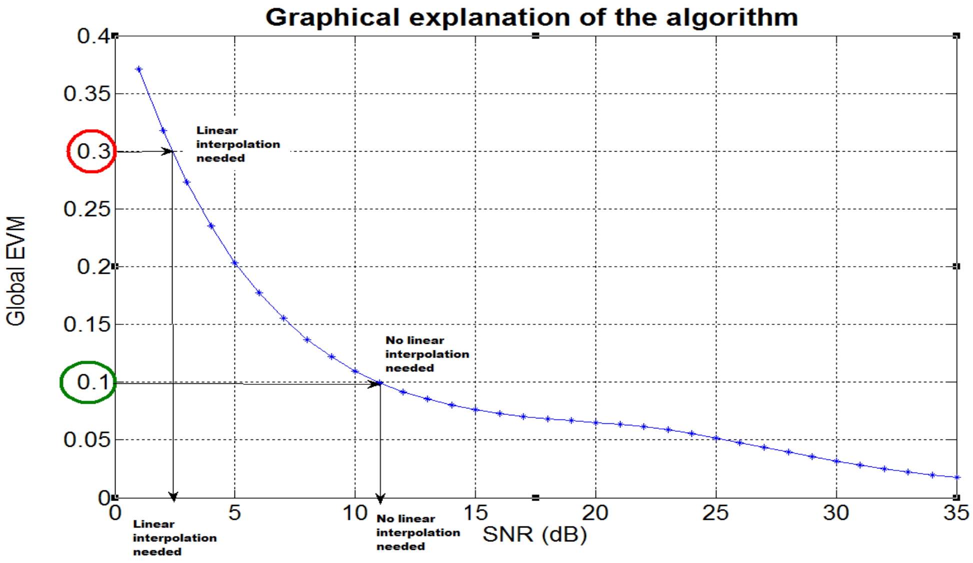


Figure 4.2.4 Graphical explanation of the proposed algorithm

STEPS 1,2,3 have to be executed only once and after that every time we want to estimate the SNR, we just perform step 4.

4.3 Statistical Evaluation

Mathematical details of the statistic measurements we are going to perform are given in appendix A. In this subchapter we examine proposed algorithm's accuracy and compare the estimation derived from the three created global curves(LUT_1, LUT_01, LUT_001). We will use the Normalized Mean Square Error and the Absolute Normalized Bias for the cases of linear interpolation, modulations M = 4 and M = 256, and number of transmitted symbols 10^3 ,10^5.

In Matlab, we use a random integer number generator (function **randi)** to create the symbols of the signal. Symbols' coordinates on the IQ-plane are equal to one of the M fixed ideal points. Transmission is corrupted by Additive White Gaussian Noise imposed with the function **awgn(**IQin,SNR,'measured'**)** where variable IQin represents the input signal and variable SNR the existing signal to noise ratio. After demodulation, the affected by noise symbols of the received signal (IQqwgn) would have been deviated from their starting positions on the constellation diagram and the blind Error Vector Magnitude will be measured by using the nearest ideal coordinates of each one. We will test the accuracy of the algorithm for SNR 1dB-35dB with a step of 0.5dB. 1000 experiments for each SNR value are going to be made. More results are shown in appendix C. Estimations are becoming better as the quantity of symbols is increasing. Therefore, to determine the overall accuracy of the algorithm we will use 100,000 bauds for the experiments.

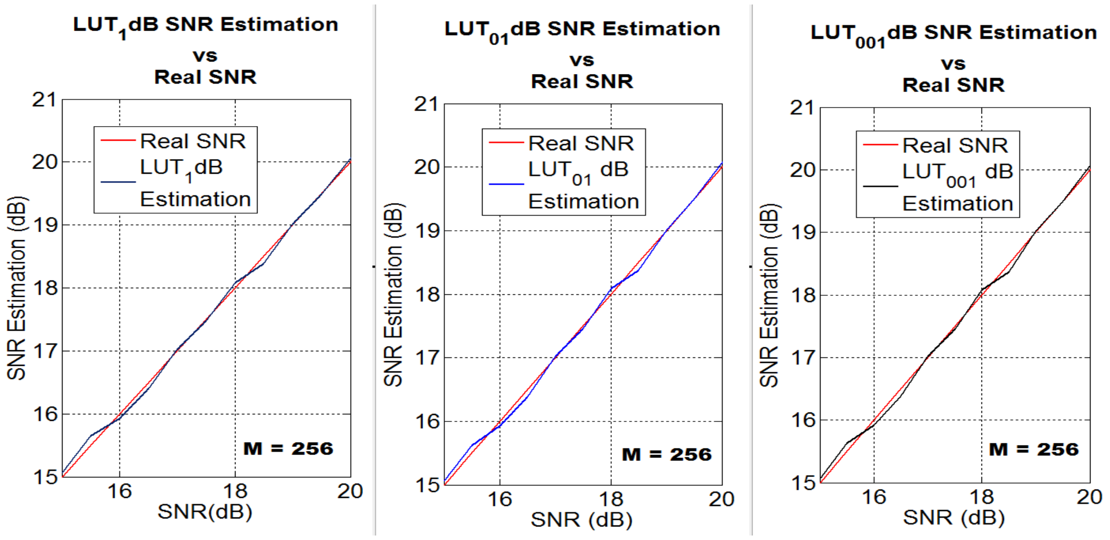


Figure 4.3.1 SNR estimations vs Real SNR for 256 QAM. 10^5 symbols

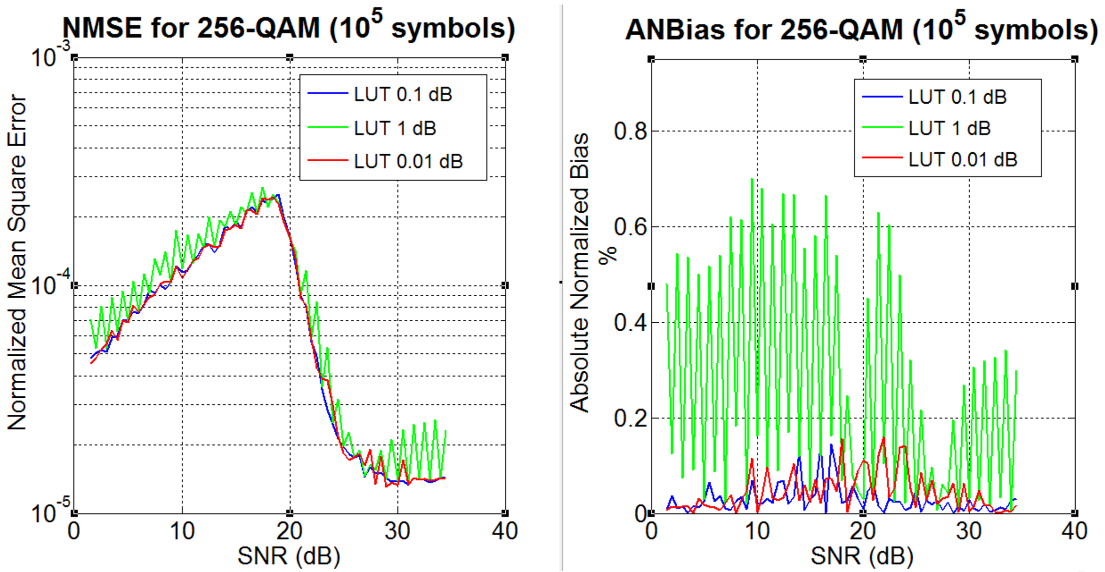


Figure 4.3.2 Normalized Mean Square Error and Absolute Normalized Bias for 256-QAM. 10^5 symbols

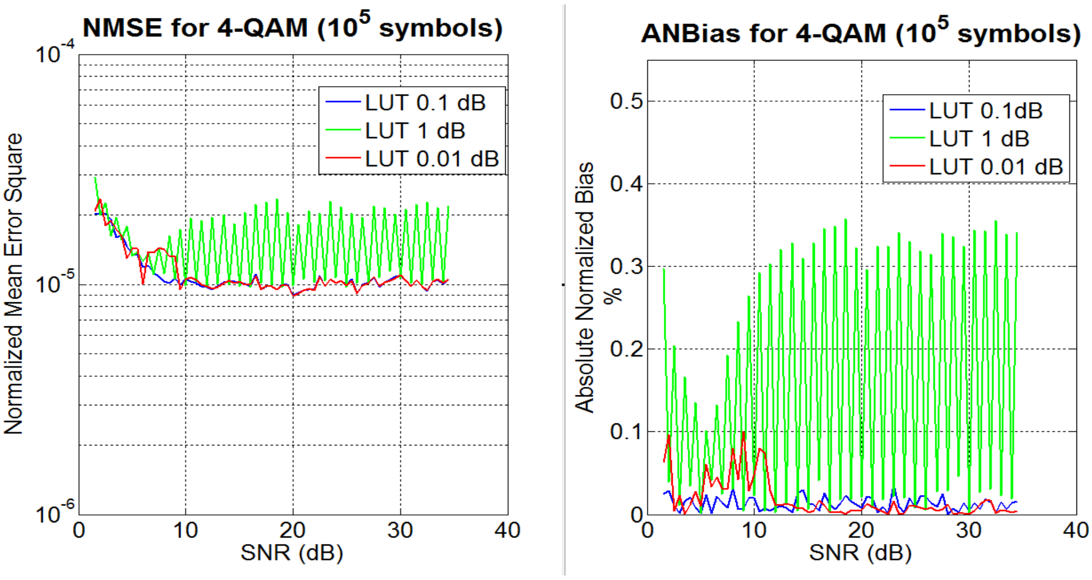


Figure 4.3.3 Normalized Mean Square Error and Absolute Normalized Bias for 256-QAM. 10^5 symbols

A variety of interesting inferences can be extracted from the previous plots:

- Generally, for every modulation level M, the proposed algorithm can provide quite satisfying estimations regardless the existing signal to noise ratio. For example, the maximum absolute normalized bias for the 256-QAM scheme is less than 1% whereas the corresponded error of formula (3.3) is almost 750%.
- The LUT_1 method is not as much accurate as the other two. Stored SNR values differ so much (1dB~25% in linear scale) that whenever linear interpolation needed (when the SNR is not precisely equal to any of the stored in global arrays SNR value) important estimation-errors cannot be avoided.

- LUT_01 and LUT_001 give similar errors in the estimations. As mentioned in STEP 2 of the algorithm in the previous subchapter, when the EVM-SNR curve is stepped by very small values, as the LUT_001 one, the desired strict monotony of the function is lost and we have to transform it in order to get a strictly monotonic graphical relation. Hence, there is no benefit from taking more points on the SNR axis for the global curve creation. In addition, larger memory is required to store the two global tables of LUT_001 and thus, more time is demanded to access them.
- Combining last two inferences is obvious that the optimal LUT method is the LUT_01 one because it is more accurate than the LUT_1 as well as more memory-efficient and time-efficient than the LUT_001. LUT_01 is chosen for the simulations in next chapters.
- As M increases, so does the algorithm's estimation errors due to the fact that the coefficient of variation is larger in higher order modulations as we can observe in figures 4.1.3 and 4.1.4.
- For every M, the coefficient of variation for SNR <5-10 dB reach its highest limits, so the falseness of the estimation for those set of values is greater. (figures 4.1.3 and 4.1.4)
- For most modulations, there is a special set of SNR values where the algorithm's ability to make satisfying estimations is slightly reduced. For example, as far as the 256-QAM is concerned, this set is 15-20 dB. In these sets, the derivative of the EVM(SNR) function is so low that the curve tends to be parallel to the SNR axis. Consequently, small deviations in y-axis(EVM) may lead to bid deviations in x-axis(SNR) as illustrated in the next picture.

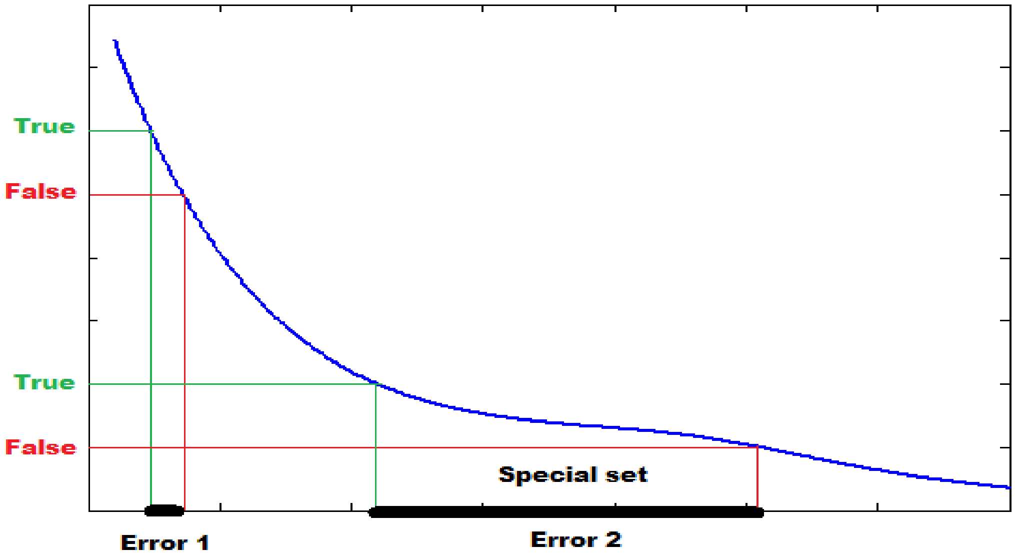


Figure 4.3.4 Errors for different SNR values using the proposed algorithm. For the special set, small deviations on y-axis lead to big deviations in x axis

As we deduce from the plots, we can use the proposed algorithm in order to reliably make estimations for the SNR. Unfortunately, there will be probably no chance to transmit such a large number of symbols to reach the mentioned accuracy. For example, for IP packets with size 1500 bytes = 12000 bits, modulation format 256-QAM and thus 8 bits per packet the number of symbols is 1500. So, we need to examine the correctness of the algorithm for fewer than 100,000 symbols, expecting of course degradations on its performance.

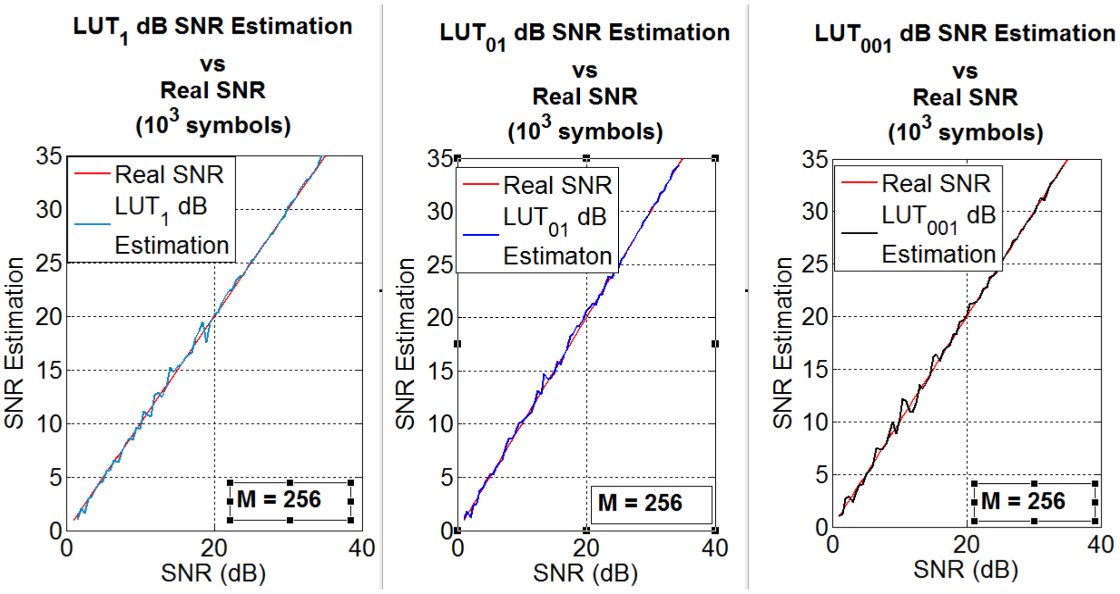


Figure 4.3.5 SNR estimations vs Real SNR for 256 QAM. 10^3 symbols

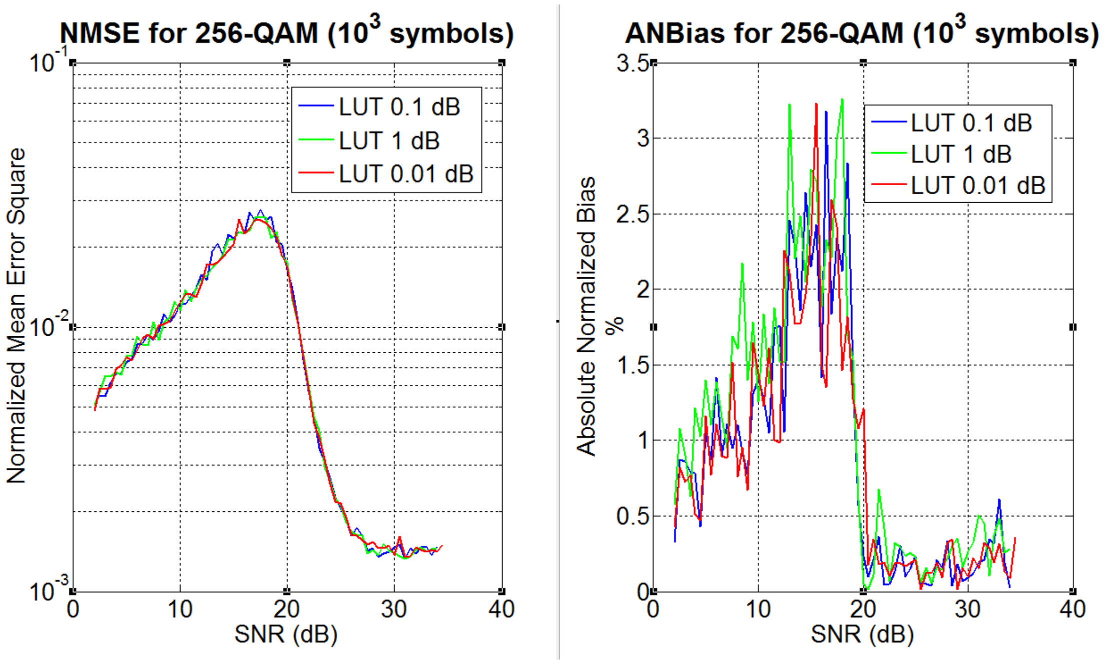


Figure 4.3.6 Normalized Mean Square Error and Absolute Normalized Bias for 256-QAM. 10^3 symbols

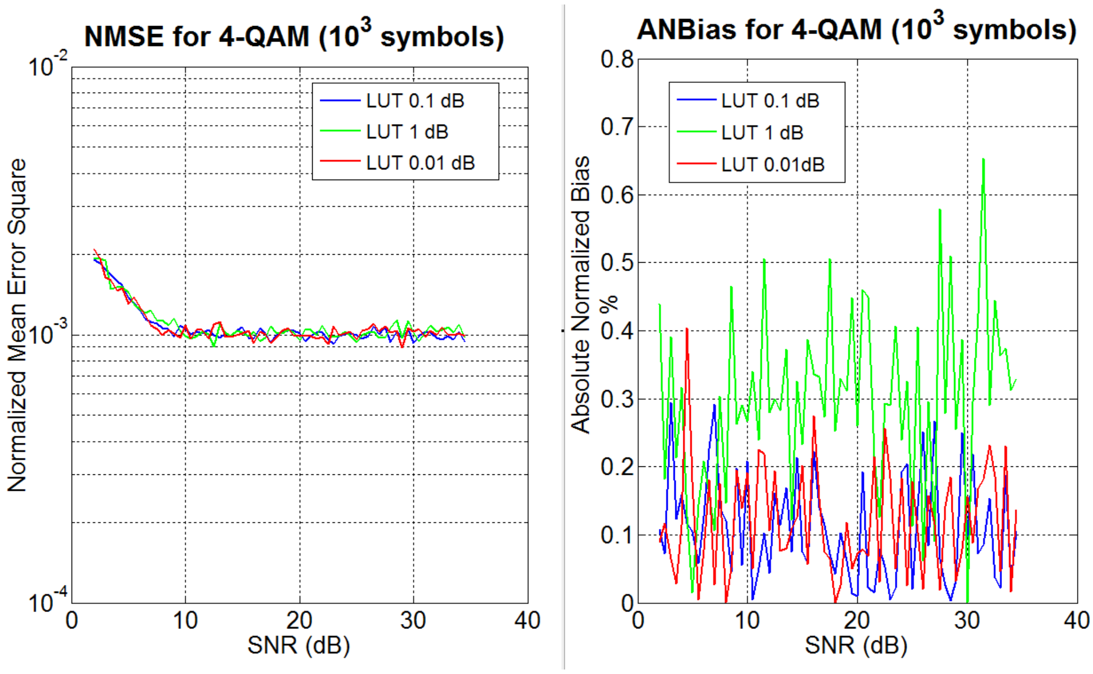


Figure 4.3.7 Normalized Mean Square Error and Absolute Normalized Bias for 4-QAM. 103 symbols

Comments:

- As we expected, when the transmitted symbols are fewer, the accuracy of the proposed method is reduced because coefficient of variance becomes bigger and thus, the stored global curve may not coincide totally with the current experimental one as we can observe in the next plot. Especially for the low SNR values big deviations occur.

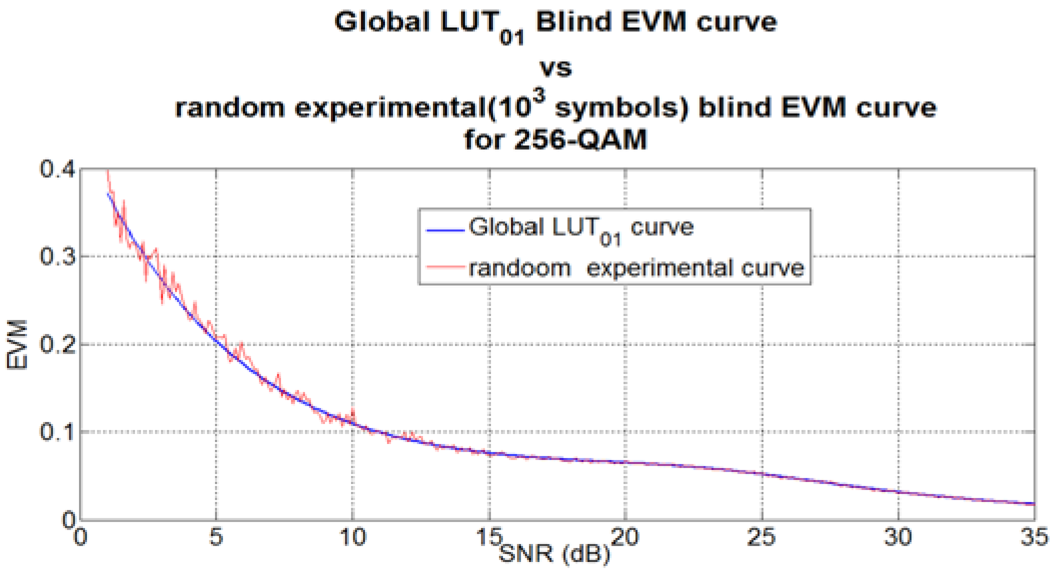


Figure 4.3.8 Global Blind EVM curve vs a random experimental curve for a transmission of 1000 symbols.

- However, the algorithm still can be used for making estimations reliably since the measured errors remain low. For example the maximum absolute normalized Bias for 256-QAM, which represents the worst case, is lower than 3.5%.
- Although the accuracy gap between the LUT_1 method and the other two (LUT_01 and LUT_001) is markedly reduced in relation to the transmission with large quantity of symbols, it still remains not as much reliable as the others. The best option remains the LUT_01 one as it combines low memory requirements and quite good estimations.

To sum up, the proposed algorithm overcomes efficiently the problems occurring with the usage of formula (3.3) when the existing signal to noise ratio is low. In addition, no specific hardware installation is needed in order to perform it. On the other hand, being an experimental method and not a mathematical one, there is a strict demand for software memory enough to store the arrays which contain the values of the global EVM and the correlated SNR. In fact, two vectors with size equal to $\frac{predicted\_SNR_{max} - predicted\_SNR_{min}}{step} + 1$ for each modulation need to be stored. That would be a serious drawback for memory limited systems. Never the less, proposed algorithm can be used to monitor the operation of a dynamic network and help to achieve the optimal allocation of the available resources.

5 System Monitoring

As mentioned in the introduction, the optical performance monitoring (OPM) is an extremely useful set of actions need to be done at most of the synchronous and future networks in order to optimally allocate the available resources and setup the transmission parameters. In this chapter, we will run various monte-carlo simulations to evaluate the performance of a dynamic optical network.

5.1 Test Setup

a. Network's topology

In a common network, most of the nodes are connected to more than one link in a way that there can be multiple optical paths between two of them. Each link is corrupted by noise with either constant or continually changing characteristics. The total physical length probably differs from path to path resulting to different signal to noise ratios as illustrated in the next picture.

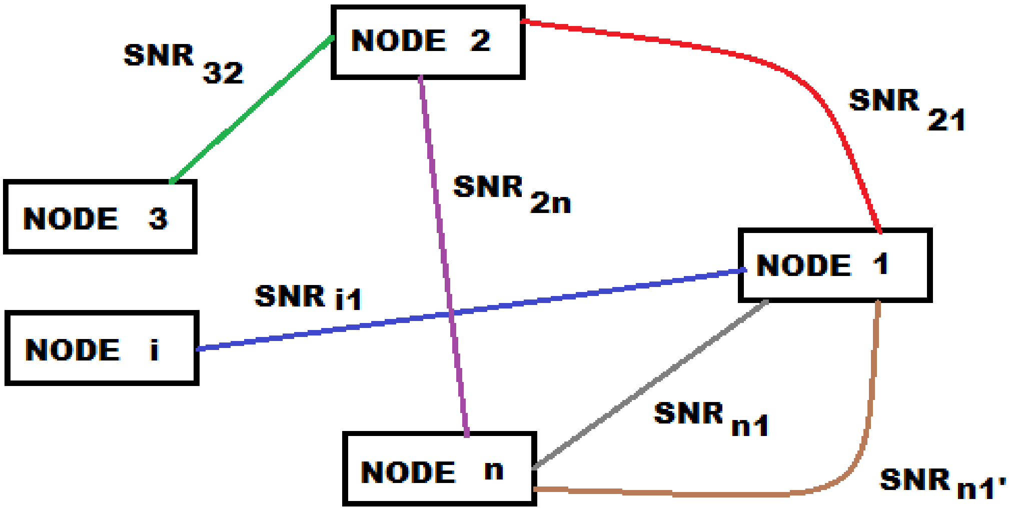


Figure 5.1.1 A real network's topology. Each colour represents a different path and each path is characterized by a different SNR at the receiver's side

b. Simulation topology

Monitoring takes place at the receiver's edge. To examine a random network's performance we will assume the following topology which is equivalent to the above one.

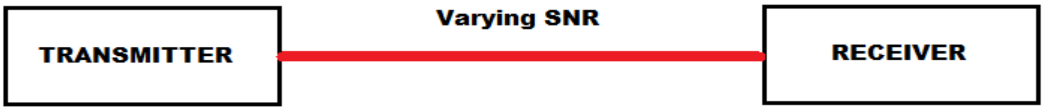


Figure 5.1.2 Topology of the simulated network

In this implementation, we care only for the digital procedures that take place at the receiver node and so, all transmitter nodes can be represented by only one. We also use only one link, the average noise of which does not remain constant, in order to represent all links. The receiver must be able to nimbly detect every SNR's change and trigger the procedures to inform the transmitter to change the transmission parameters in case conditions have changed drastically. The needed actions for informing the transmitter are made by higher-level layers.

c. Configuration of the transmission

We assume that the length of the transmitted packets is fixed during each experiment. The duration of the simulation is measured in bits and it is divided in units equal to the packet length (Time_length_unit).

Concerning the link's noise, we assume the existence of only Additive White Gaussian Noise. The existing signal to noise ratio is uniformly distributed in the interval 9.96dB-35-dB while the duration changes according to Poisson's law. It is important to notice that the minimum duration of a SNR value is the same for all experiments regardless the length of packets used (Noise_length_unit). The Lambda value of Poisson's formula represents the average expected SNR duration measured in Noise_length_units.

Another parameter we have to define is named "Block size". Block size represents the number of packets need to be transmitted before the monitoring procedures take place. For example, block size equal to five means that the receiver has to calculate the noise effects for five packets and thus five time units are needed before informing transmitter about the monitoring results.

The modulation formats we use are 4-QAM , 16-QAM, 64-QAM, 256-QAM. Every moment, network must be capable of choosing the most suitable to the existing SNR modulation. For each M-ary scheme there is a particular SNR threshold which is correlated with an upper boundary of acceptable BER for transmitted packets. The maximum desired BER is 10^{-3}. For safety and reliability reasons we use SNR thresholds corresponding to BER lower than $10^{-}3$. Next table contains the SNR thresholds for a BER lower than $0.85*10^{-3}$. To create this table we run program 2 of appendix B.

Modulation	**SNR(dB) threshold** **for BER < 0.85*10^{-3}**
4-QAM	9.95
16-QAM	16.7
64-QAM	22.7
256-QAM	28.57

Table 5.1.1 SNR thresholds for BER < 0.85*10-3

d. Metrics for the Performance evaluation

We would like to examine the above described network's performance for different packet lengths, block sizes, and average SNR durations and so we will run the simulations various times, using each time different parameters. For the evaluation of the experimental results we will use the following metrics:

i) <u>Packet failure probability:</u>

Link's existing noise may cause symbol errors and bit errors for the transmitted packets. We assume that a packet failure occurs every when the BER of the received packet is higher than 10^{-3}. The metric is mathematically defined as :

$$Packet\ failure\ prob\ = \frac{Packets\ with\ BER > 10^{-3}}{Total\ number\ of\ transmitted\ packets}$$

Perhaps it is the most significant metric and we would of course desire it to be low.

ii) <u>Average bit rate of successfully transmitted packets:</u>

Another important metric for the evaluation of the network is the average bit rate. For each successfully received packet we measure the transmission time, which may not be the same for all packets due to the fact that the modulation does not remain constant. Hence, we have to calculate the average transmission time for all non-failed packets. For the bit rate we have:

$$Average\_bit\_rate = \frac{Packet\ Length}{Average\ transmission\ time}$$

The higher this bit rate is, the better the network performs. It is important to notice that we should not take into consideration failed packets, as they are transmitted with a bit rate larger than that the network can handle.

iii) BER of successfully transmitted packets:

After the successful reception of a packet, we would like to know how many bits have been correctly transmitted. Consequently, we calculate the bit error rate. The correction of the heavily affected by noise bits may cause time-cost, so we would desire small BER for all packets.

iv) Format changes:

This metric represents the number of total format changes that take place in the experiment. When this number is large the network is capable of adapting agilely in varying conditions. On the other hand, a large quantity of format changes result to a cost of time, because important time intervals are needed for the transmitter to change modulations.

5.2 Instant scenario example

Before we start the experiments for various parameters, we run an “instant scenario”. We run program 3 of the appendix B (In this program, additional metrics have been also calculated). The parameters and the results of the transmission are shown in next lines:

Transmission Parameters:

Total size of data = 64 Megabytes
Packet length L = 2^{16} bits
Noise unit length = 2^{16} bits
Baud rate = 1 Megabaud/s
Expected duration for SNR (Poisson law): Lambda = 7
Block size = 1

Numerical Results:

Total number of packets = 8192
Packet failure Probability = 5.54%
BER of successfully transmitted packets = $9.52*10^{-5}$
Format changes = 929
Average bit rate of successfully transmitted packets = 3.67 Mbps

Graphs and notices:

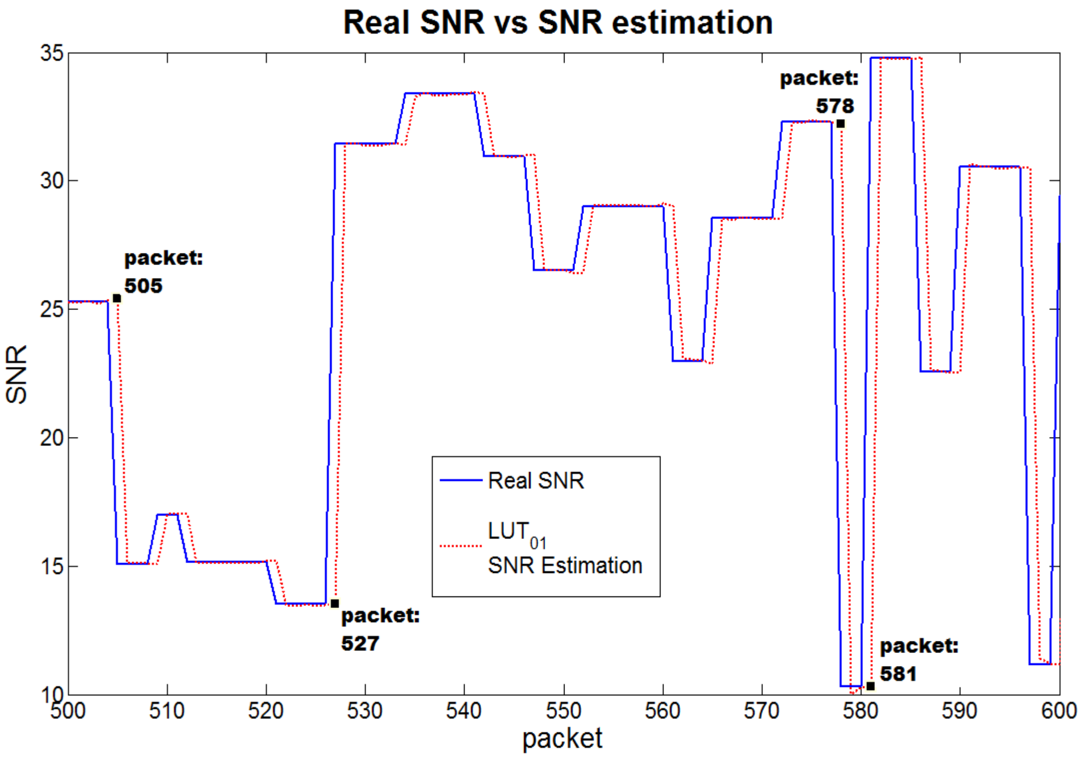


Figure 5.2.1 SNR estimations for the packets 500-600

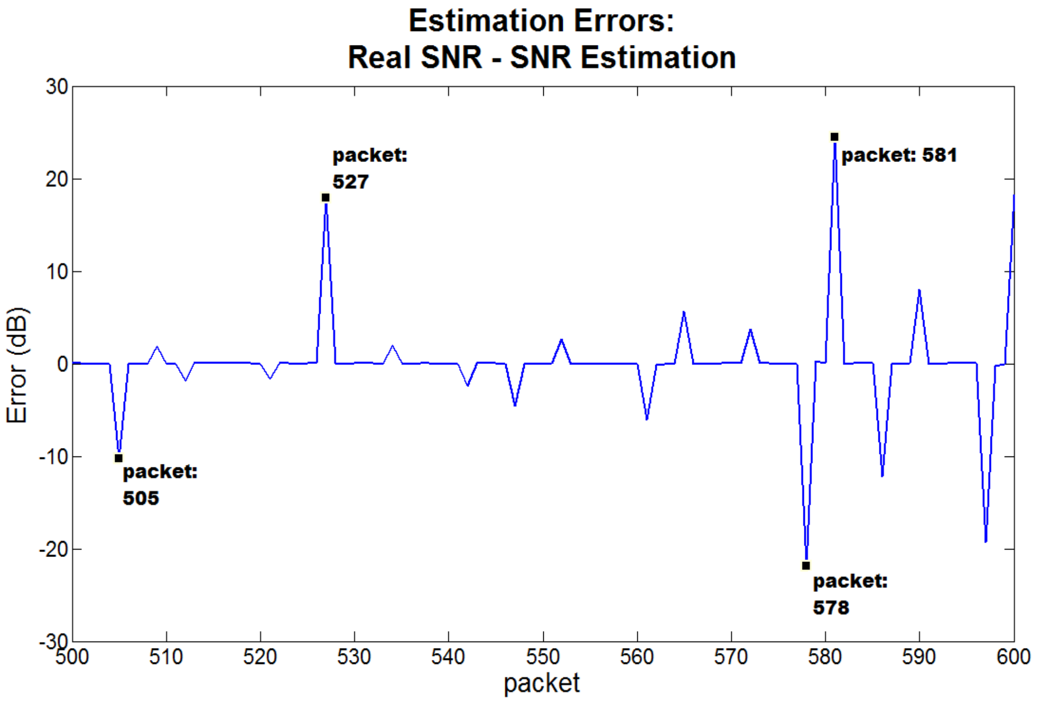


Figure 5.2.2 SNR estimation errors for packets 500-600

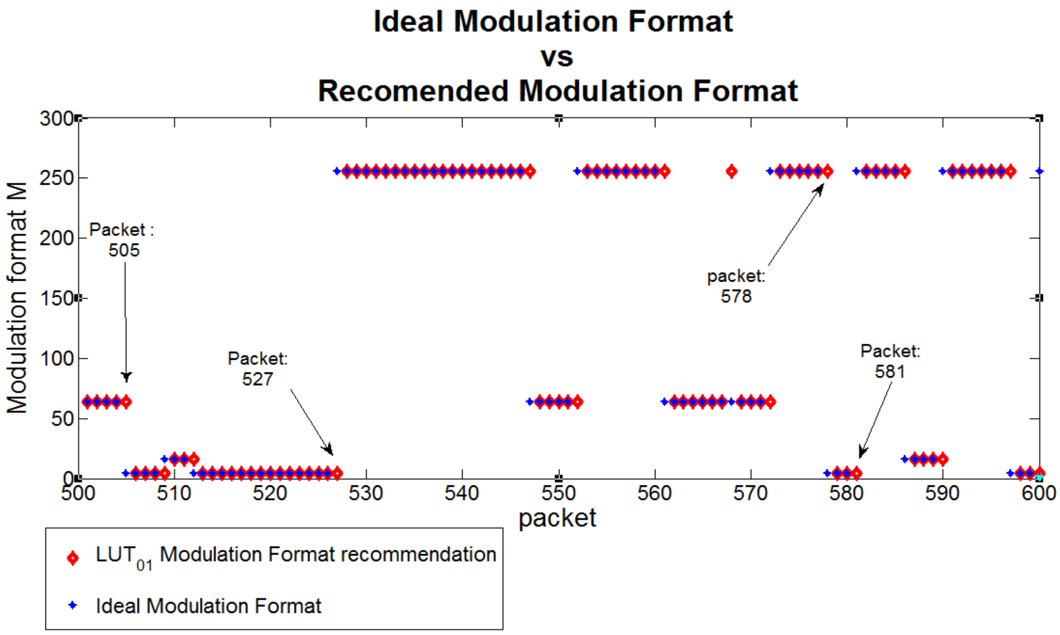


Figure 5.2.3 Ideal and recommended modulation format for packets 500-600

From figures 5.2.1, 5.2.2 and 5.2.3 it is obvious that the proposed algorithm is appropriate for monitoring a system and thus for choosing the optimal M-ary QAM modulation. The SNR estimations, represented by the red line, almost coincide with the real SNR values which are represented by the blue line. Most of the estimation errors are valued close to zero. In fact, big deviations occur only when the existing SNR changes, because the algorithm (here the LUT_01 method) is capable of estimating the SNR of the already received packets and not the SNR of the next ones. A time interval equal to Block size is required for the algorithm to adapt its estimations to the new SNR. To prove this, we marked four packets in the plots: 505 527,578 and 581. Each of these packets is the first packet corrupted by a new SNR and it can be easily observed that for these time units the estimation errors are great.

The Bit Error Rate of transmitted packets is strictly related with the existing SNR and the modulation format. So, when we have sudden reductions of the SNR, the BER increases rapidly. At those moments it is beneficial to change the modulation format because, as mentioned in chapter 2, higher level modulations cannot resist to the noise effects as effectively as the lower level modulations do. Block size time units required to inform the transmitter for the new link’s condition and until this moment transmission continues with the former, no longer ideal modulation. Of course, no BER problems occur when signal to noise ratio changes from low values to higher ones as the former ideal modulation format is at least equal to the current one. That can be observed in the next figure, where only the 505 and the 578 (from the previously marked packets) packets have been heavily affected by the SNR change.

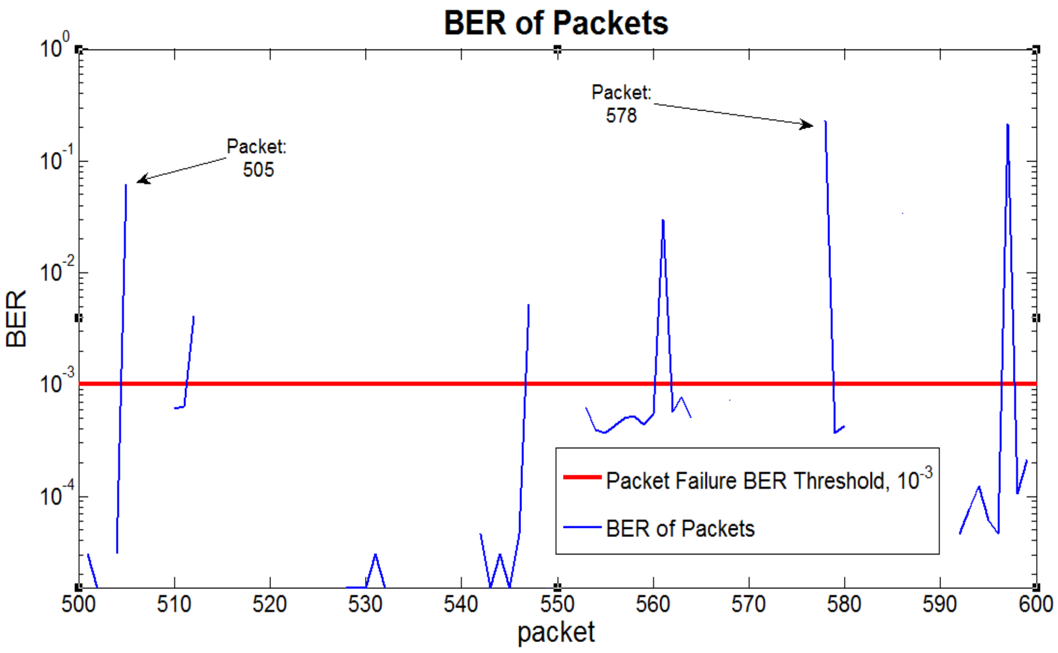


Figure 5.2.4 BER for packets 500-600. Packets with BER=0 have not been plotted. Packets with BER greater than 10^{-3} cannot be successfully received by the destination node

To visualize the steps of the system's adaption to a rapid SNR decrease, we show in the next plots the constellation diagrams for packets 577, 578, and 579. The constellation diagrams of 577 and 579 are quite "clear" whereas the diagram of 578, which is the first packet with the new SNR, is very "noisy". Notice that:

Packet 577: SNR = 32.23 dB Modulation: 256-QAM

Symbols per packet = L/$\log_2(M)$ = 8192

Packet 578: SNR = 10.29 dB Modulation: 256-QAM

Symbols per packet = L/$\log_2(M)$ = 8192

Packet 579: SNR = 10.29 dB Modulation: 4-QAM

Symbols per packet = L/$\log_2(M)$ = 32768

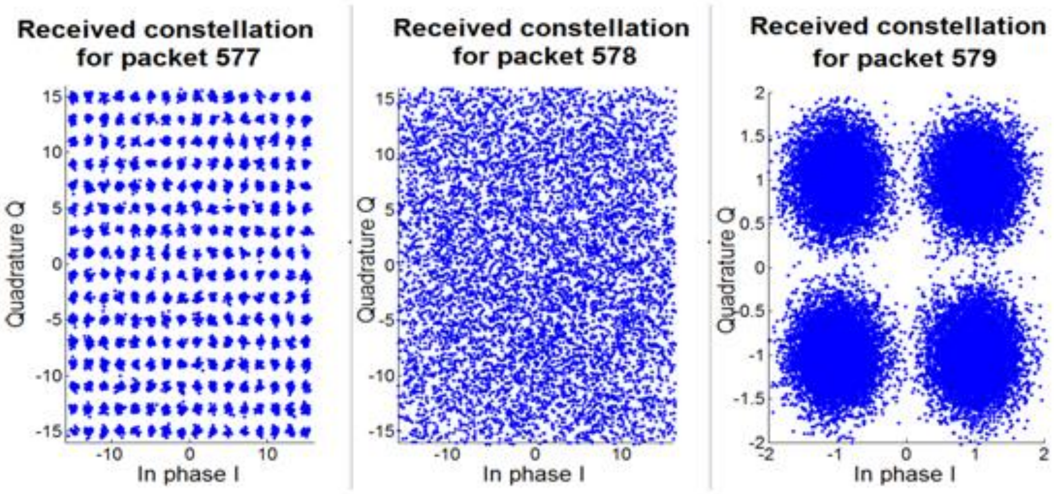


Figure 5.2.5 Constellation diagrams during a SNR rapid reduction

5.3 Network's Performance Evaluation

To evaluate how differently designed networks perform in continually changing noise conditions, we will do 5.2 paragraph's experiment for different values of packet Length, Block size and average expected duration of SNR. We will use the previously explained metrics to compare the results.

The total number of experiments can be categorized in two sets. In the first one, we will examine how a network with fixed Block_size(BS) behaves for different average SNR (Lambda) durations. The size of the total transmitted file for these simulations is 256 Megabytes. In the second one, the Lambda value will remain constant for various Block sizes. The size of the total transmitted file for these simulations is 512 Megabytes. We would like to make measurements for a wide range of values so we will use lognormal step for Lambda and Block size axis. For both categories three different packet lengths will be used:

i) $L_1 = 2^{14}$ bits. The total number of packets for a 256 Megabyte file is 131,072 and for a 512 Megabyte file 262,144

ii) $L_2 = 2^{12}$ bits The total number of packets for a 256 Megabyte file is 524,288 and for a 512 Megabyte file 1,048,576

iii) $L_3 = 2^{10}$ bits The total number of packets for a 256 Megabyte file is 2,097,151 and for a 512 Megabyte file 4,194,304

For all experiments, the quantity of transmitted packets is big enough to get reliable results. The Noise_length_unit is assumed equal to the length of the largest packet, which is the L_1, and the Baud rate is fixed and equal to 10^6 bauds/second. Next figures show the measured metrics for the kinds above described simulations

Graphs and notices:

Packet failure Probability

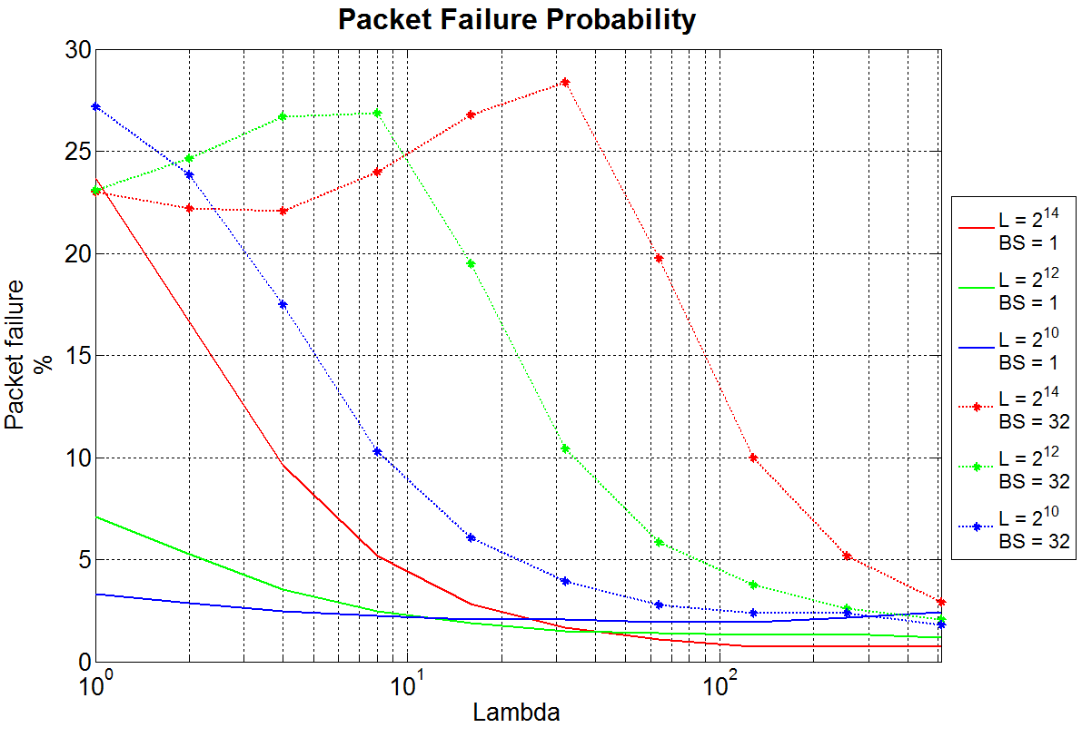


Figure 5.3.1 Packet failure probability, BS = 1 and BS = 32 (star marked)

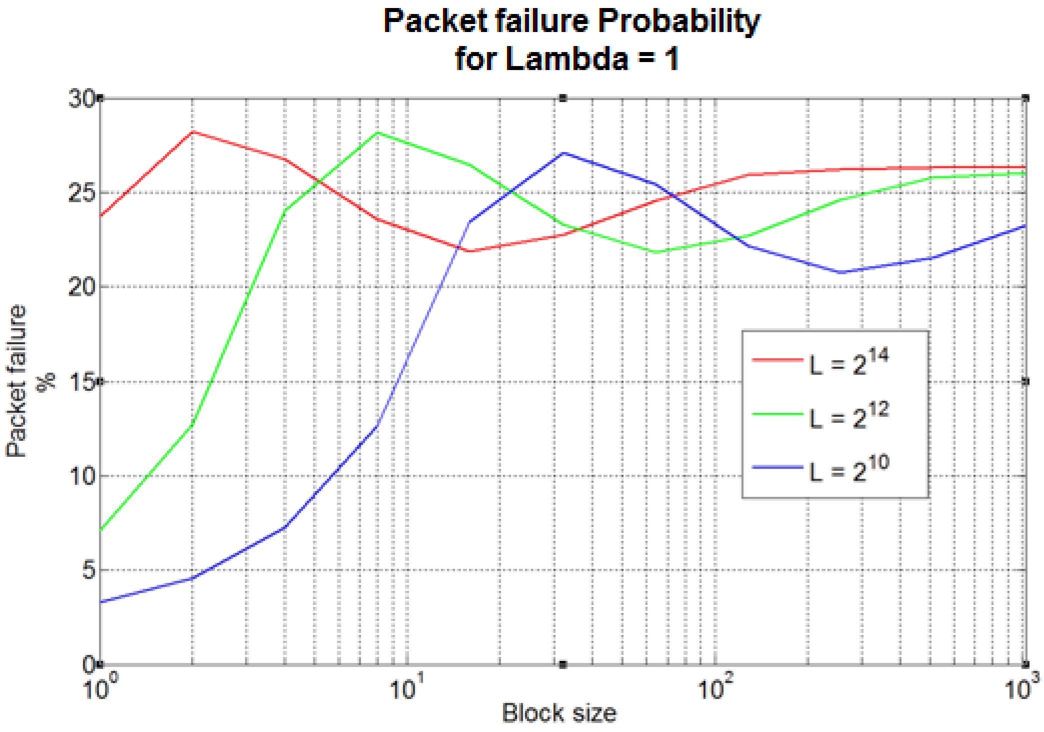


Figure 5.3.2 Packet failure probability, Lambda = 1

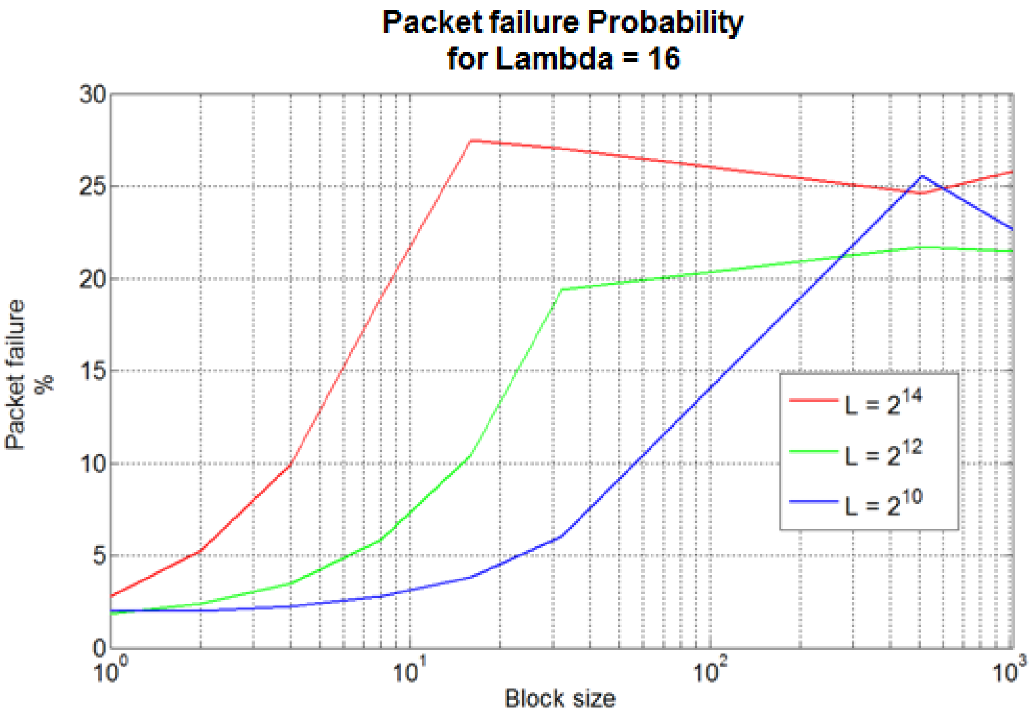


Figure 5.3.3 Packet failure probability, Lambda = 16

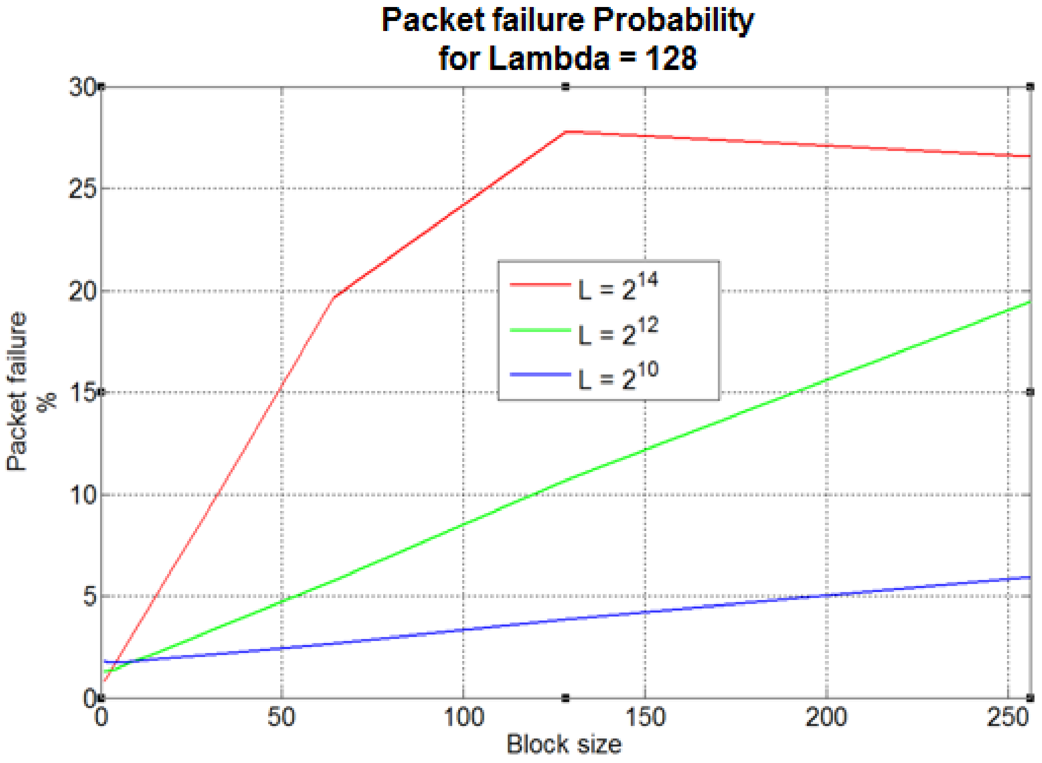


Figure 5.3.4 Packet failure probability, Lambda = 128

From the figures above, first of all we infer that the packet failure probability – Lambda graphical relation can be split in two parts. The first one consists of the measurements made for Lambda < BS*L/Noise_length_unit and the second one for Lambda > BS*L/Noise_length_unit.

In the first one, the network is not capable of following in time the continually changing link's conditions. Hence, the larger Block size we have the more different SNR values contribute to the final estimation and thus the final estimation would probably be stable and equal to the mean real SNR value (Figure 5.3.5). In our simulation, where we use uniformly distributed SNR values in the interval 9.95dB-35dB, the mean SNR value is (9.95+35)/2 = 22.48 dB. According to table 5.1, for SNR=22.48 dB the most appropriate modulation is the 16-QAM, so, if the ideal format is 256-QAM, 64-QAM or 16-QAM, few packet drops will occur, but if the corresponding ideal format is 4-QAM many packets will be lost due to noise effects. Therefore, there is a constant probability ¼ = 25% to unsuccessfully detect a packet. That can be observed in last figures where the Packet failure Probability is almost stable and equal to 25% . As far as the packet length is concerned, the only thing we have to notice for this area is, that smaller packets can perform better for the same Block size as their transmission time is quicker and thus the monitoring is performed more frequently. Next figure shows the SNR estimations made by the receiver for an instant scenario with Block size =256, Lambda=2, L = 2^{14} =Noise_length_unit. Notice that only 16-QAM is used for the transmission.

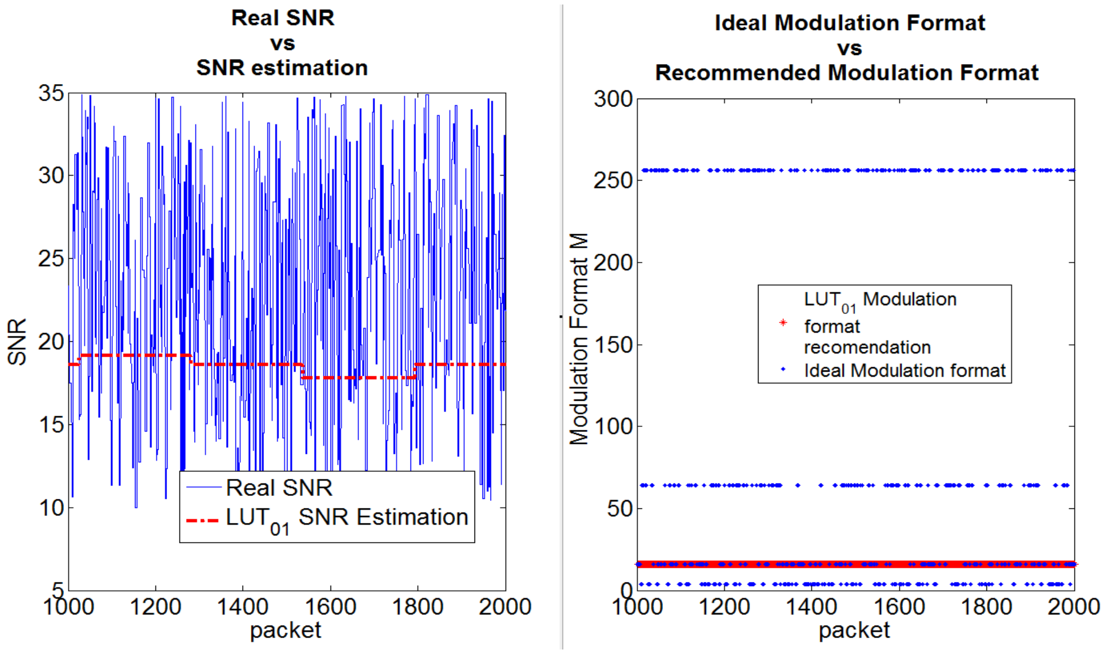


Figure 5.3.5 SNR estimation and recommended modulation format for a transmission with Lambda << L*BS/Noise_time_unit. Lambda = 2, L = 2^{14}, Noise_time_unit = 2^{14}, BS = 256

In the second part, where the Block size is smaller than the average duration of the SNR, the system can check the link's condition multiple times, before the SNR changes. Therefore, as the ratio BS/Lambda decreases the packet failure probability decreases too. For quite big values of this ratio, the metric remains constant. Generally, the majority of packet failures happens during a SNR's change and Block size time units are needed for the network to adapt its modulation. Example in subchapter 5.2 is a network which operates with such parameters. As we can see in the plots of that subchapter, the estimation errors are very small and all modulation formats are used with the same frequency as they should be. Moreover, for all transmissions, when BS/Lambda~1, it is more beneficial to use small packets because the system can adapt its operation more quickly. On the other hand, for BS/Lambda≪1, large packets can also give relatively flexible monitoring and considering that larger packets give more accurate estimations as they consist of more symbols, it would be preferable to use packets with bigger length. To understand the difference in the estimation accuracy between the systems that use short packets and the systems that use long packets, we run the same instant scenario with the scenario of the previous subchapter using this time packet size 2^{12} bits. In the following figure we can see that the SNR estimations are not as satisfying as when 2^{16}-bit packets are used and the deviations that occur may lead to an overestimation for the optimal recommended modulation format, resulting in unacceptably high BER.

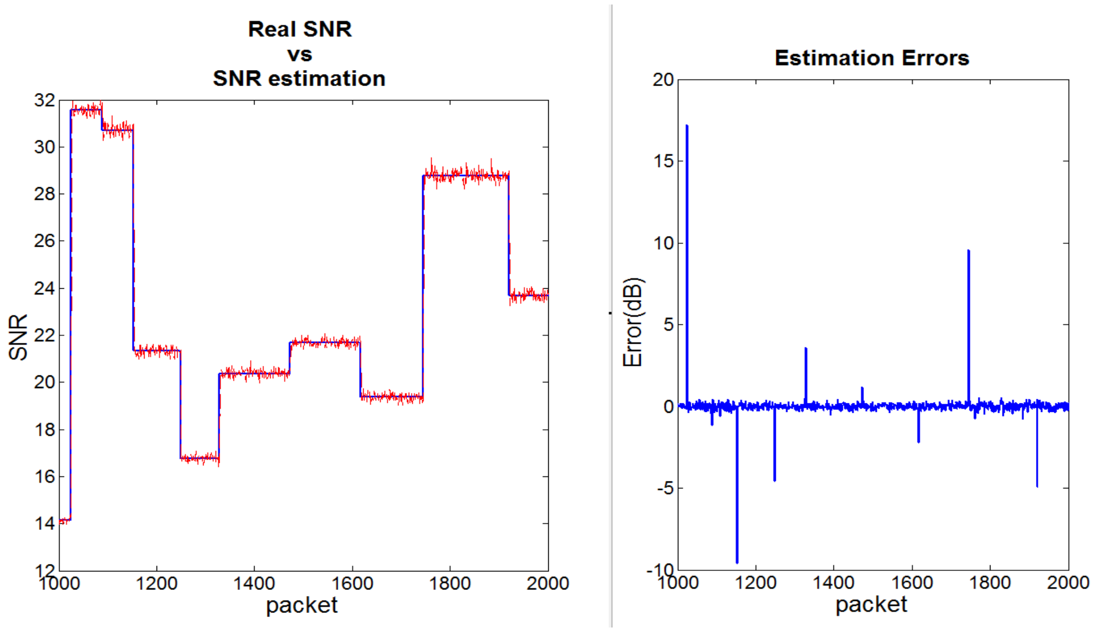


Figure 5.3.6 SNR estimation and estimation errors for a transmission with , Lamda >> BS, L = 2^{12}, Noise_length_unit = 2^{16}, Lambda =7, BS = 1

Packet failure probability reaches its peak value when Lambda = BS*L/Noise_length_unit. In this case, even if the receiver has estimated well the real SNR for the last Block size received packets, the real SNR will probably change in the next unit and so the monitoring results are not helpful.

Packet failure probability equal to 25% is an unacceptable ratio for every network. Dropped packets have to be transmitted again resulting in an additional time cost and traffic. For a satisfying enough system's operation, ratios BS/Lambda<1 are required. Consequently, for the next metrics we will emphasize mostly in networks with BS<Lamda. Besides, for a well designed network, SNR must change at most as frequently as the procedure of monitoring takes place.

Average Bit Rate of Successfully transmitted packets:

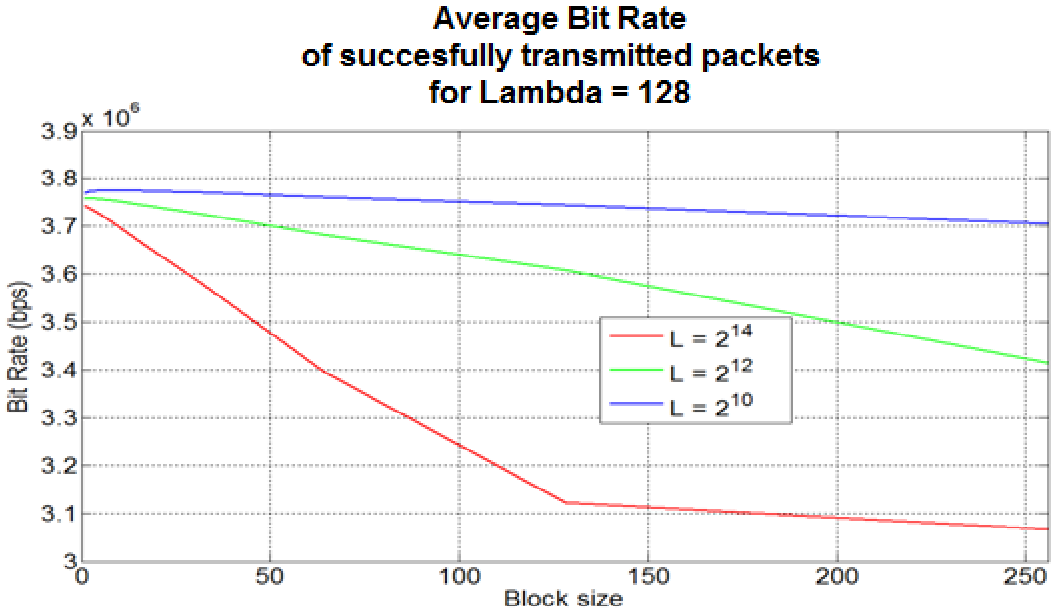


Figure 5.3.7 Average bit rate of successfully transmitted packets, Lambda = 128

In the figure above, we see how the average bit rate changes while the Block size increases for a network with BS*L/Noise_length_unit <Lambda. For every packet size, it is obvious that the smaller Block size we use the higher rates we achieve. For those values, the network can effectively adapt its operation to the existing noise conditions and therefore the chosen modulation is in most cases the optimal one. Furthermore, we observe that for the same Block size, the networks that use shorter packets perform better than those that use longer ones as they can faster increase the modulation format when favorable conditions exist. For extremely low BS relatively to Lambda and long enough experiment time, modulation formats are used with the same frequency. As mentioned in a previous page, the mean SNR for the experiment corresponds to the 16-QAM modulation. As a result, the average bit rate tends to be constant and equal to log2(16)*Baud rate = 4*Baud rate.

Last, for complement reasons we have also measured the average bit rate for networks with very low Lambda. In this figure we observe that for BS*L/(Noise_length_unit*Lambda) =(1-10) there is a degradation in the bit rate of successfully packets because the majority of those packets is transmitted with 4-QAM modulation which is the safest option. Further increase for the Block size leads to a static performance with a fixed 16-QAM modulation and the bit rate reaches the 4Mbps again.

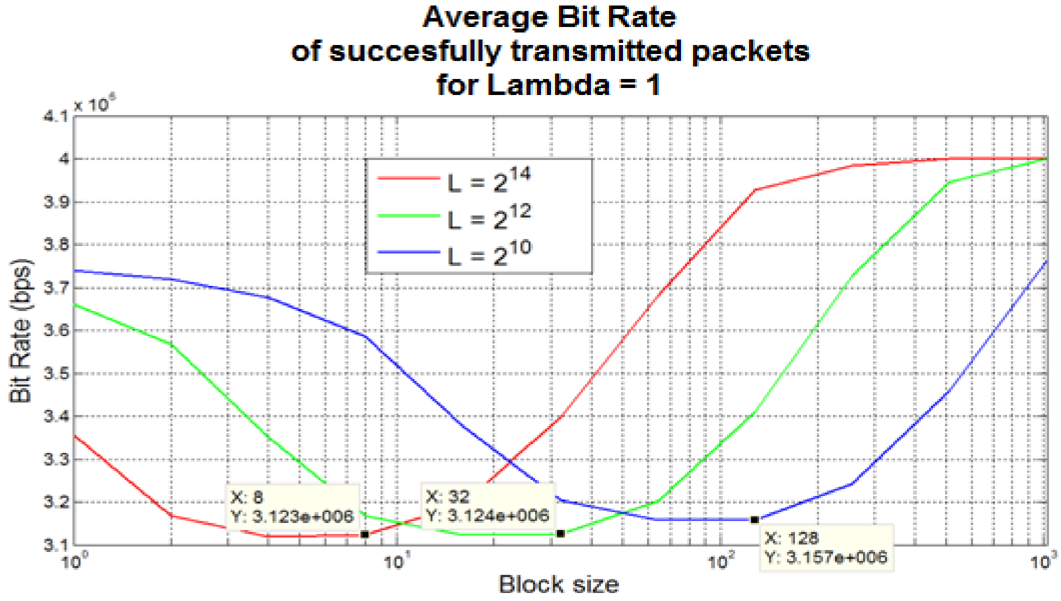


Figure 5.3.8 Average bit rate of successfully transmitted packets, Lambda = 1

BER of successfully transmitted packets:

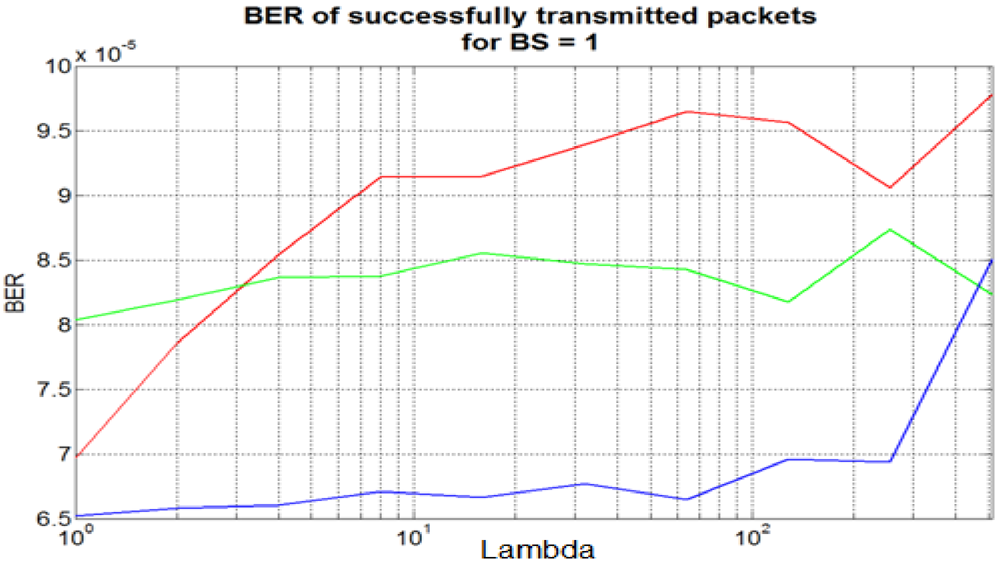


Figure 5.3.9 BER of successfully transmitted packets, BS = 1

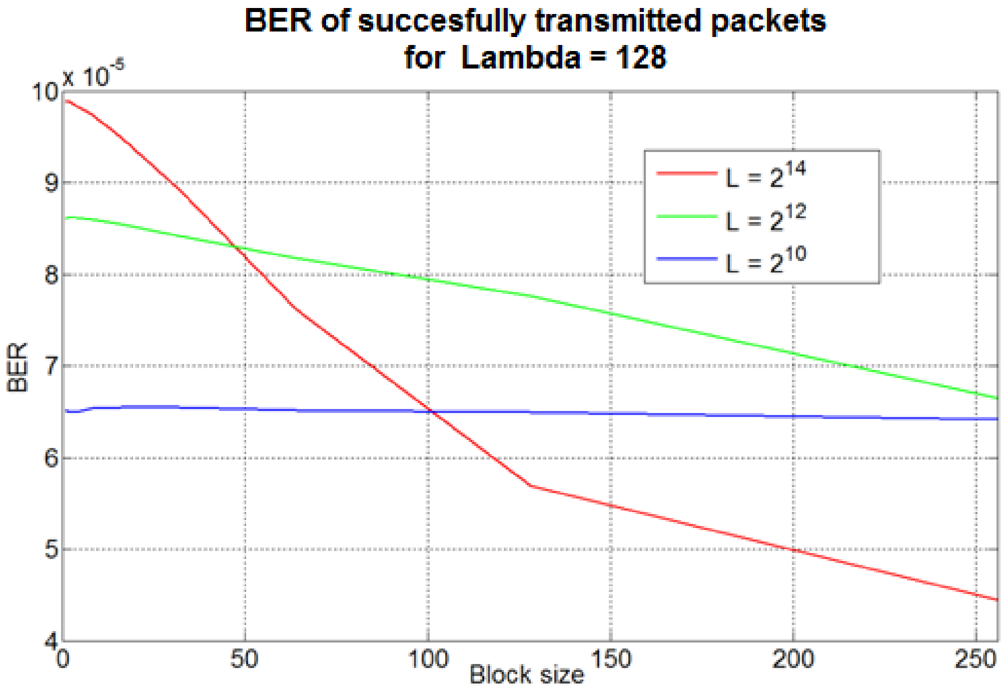


Figure 5.3.10 BER of successfully transmitted packets, Lambda = 128

An interesting notice can be extracted from the measurements of this metric. In the last two figures we observe that for the Lambda values, where the packet failure probability was greater than the others, the opposite happens for the bit error probability of correctly received packets. The inability to estimate accurate enough the existing signal to noise ratio, half times leads to overestimation of the recommended modulation format and half times to underestimation. In the first cases, we have higher than the acceptable limit BER while in the second ones we have successfully transmitted packets with quite low BER due to the fact that the used modulation is lower than the optimal and although this fact leads to reduced bit rates, it also causes very few symbol errors. Moreover, we observe that for quite low ratios L*BS/(Lambda*Noise_length_unit) the performance of the network according to this metric tends to be stable as we can infer mainly from the blue line which represents the smallest packets we use for the experimental.

Total format changes:

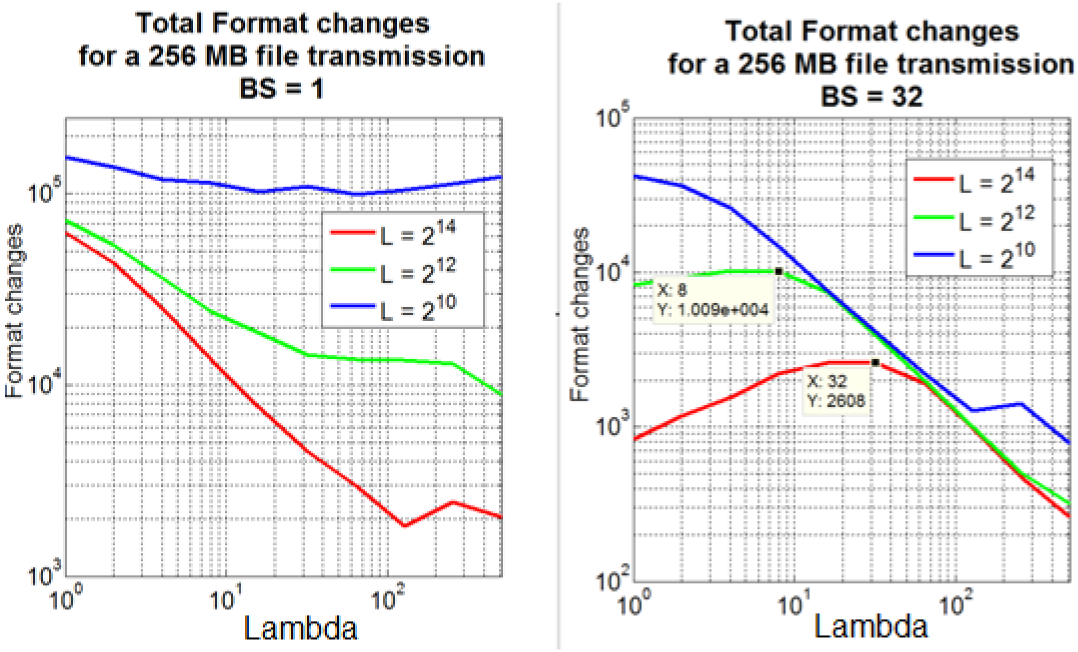


Figure 5.3.11 Total Format changes for a 256 MB file transmission, BS = 1 and BS = 32

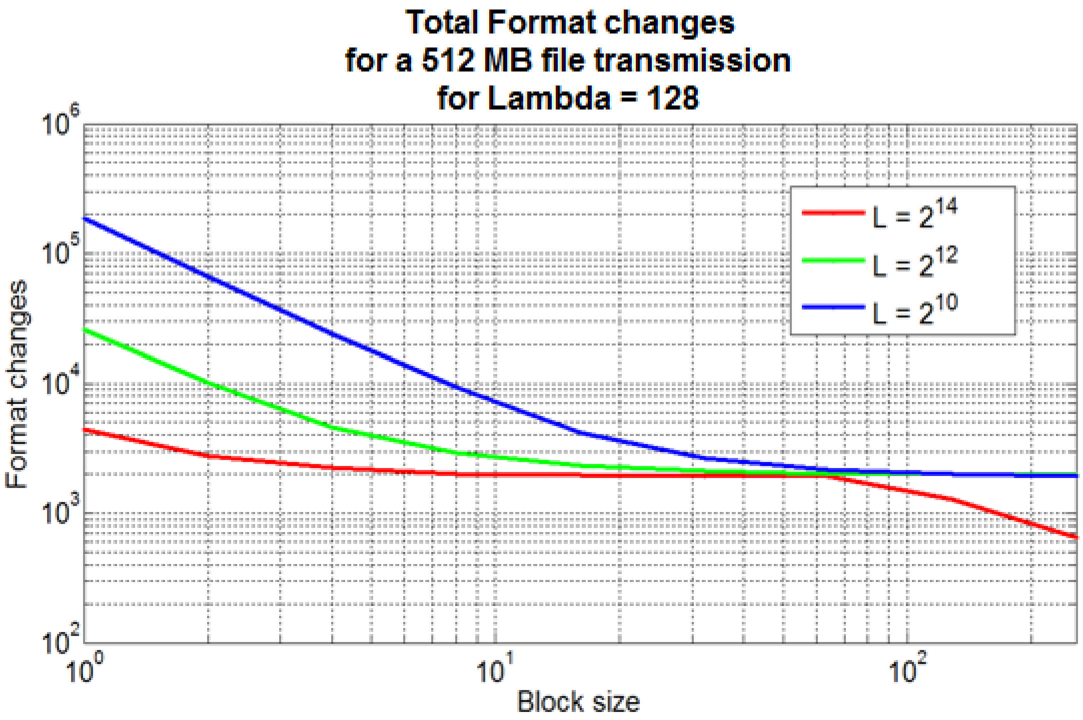


Figure 5.3.12 Total Format changes for a 512 MB file transmission, Lambda = 128

In the plots above, first of all we observe that for constant Block size and Lambda>BS*L/Noise_length_unit, the number of format changes decreases as Lambda increases due to the fact that the existing SNR changes less frequently. For Lambda<BS*L/Noise_length_unit the system cannot handle the link's very frequently changing noise and this results in ignoring the majority of the changes and hence, the number of format changes is reduced. Especially for BS>>Lambda this metric is equal to zero as we can infer from Figure 5.3.5 where the modulation format is fixed on the 16-QAM. In addition, for the same values of Lambda the number of format changes increases as the Block size decreases. As far as the packet length is concerned, we infer that the number of format changes is bigger for systems with smaller packet size because the estimation's variance around one SNR value is large enough to cause changes in the modulation formats even when not needed, as we can notice in figure 5.3.6. Most of these unnecessary changes are avoided when we use larger packet sizes.

For an efficient network's performance the total number of format changes should not be extremely high because additional time cost is imposed. If the recommended modulation format changes continually between two adjacent values, for example 4-QAM and 16-QAM, it is preferable to keep up with the lower level modulation instead of changing the modulation for short time intervals, trading off in this way part of the maximum achievable bit rate. It is important to keep an efficient balance between the flexibility and the extra delay.

5.4 Conclusion and combined metrics

To sum up, in this chapter we experimentally examined the operation of various dynamic networks in different noise conditions. From the simulation results we verified that the flexibility of the system increases while block size decreases relatively to the average duration of a SNR. The final deduction of which network scheme performs best depends on the metric we use to make the comparisons. A scheme may perform satisfying enough according to one metric but not so efficiently according to another. To extract general deductions, we can define formulas that combine two or more of the metrics we used. For example, if we care for the packet failure probability and the BER of successfully transmitted packets we can use the following factor:

$$n = \frac{1}{(Packet\ failure\ Probability * BER\ of\ successfully\ transmitted\ packets)}$$

If we would like to combine the average bit rate with the packet failure probability we can use the next factor:

$$n = \frac{Average\ bit\ rate\ of\ successfully\ transmitted\ packets}{Packet\ failure\ Probability}$$

The nominator of the formula should consist of the metrics that represent positive parameters of the network, such as the average bit rate of successfully transmitted packets, and the dominator should consist of the negative ones, such as the packet failure probability. Hence, the network scheme with the highest factor n is the optimal one. The provider and the manager of the network take into consideration consumers' needs and expectations in order to use the most appropriate factor.

6 Conclusion and Further Work

Future telecommunication needs are so demanding that they will not probably be satisfied only by the physical improvement of the standard optical fiber. Many elastic and flexible network architectures have been proposed from the scientists to overcome the disabilities of the conventional optical networks to handle the future traffic. These systems are capable of monitoring the network and achieving the optimal allocation of the available resources.

In this thesis, networks with digital coherent technology were examined. We introduced an algorithm for estimating the links' Signal to Noise Ratio of networks with 4-QAM, 16-QAM, 64-QAM and 256 QAM modulation formats, by applying DSP techniques on the received packets. A huge quantity of statistical results was added to show that the algorithm is accurate enough for monitoring reliably an optical network as very small estimation errors occur when use it. Moreover we tested the performances of differently configured dynamic networks, equipped with one link, in varying noise conditions and interesting conclusions were made.

Despite the fact that this thesis focuses on optical networks, the proposed algorithm can also be effectively used for all systems that used coherent multilevel digital technology, (for example some wireless systems). Furthermore, although the algorithm was developed for M-QAM modulation schemes, similar procedures can also be used for other M-ary formats such as M-PSK. We only have to perform the steps of the algorithm using the appropriate parameters' values. In addition, a similar method may be developed for the estimation of the Bit Error Rate.

An expansion of this research may contain the mathematical establishment of the proposed algorithm as in the current thesis, the method is based on experimental results. Moreover, we could experimentally test the effectiveness of the method when it is used as a tool to find the less noisy optical path within a multi-link designed network. Last, a related interesting research project would aim to find the optimal SNR step for the global curves (described in chapter 4) of the introduced method.

References

[1] “*Διαφάνειες για το μάθημα "Φωτονική Τεχνολογία" του 9ου Εξαμήνου της Σχολής Ηλεκτρολόγων Μηχανικών κ' Μηχανικών Υπολογιστών*,” available at http://photonics.ntua.gr/announcements.html

[2] Agrawal, P.G., “*Fiber Optic Communication System, Second edition,”* John Wiley & Sons, 1997

[3] Αλεξανδρής Α., “*ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΜΕ ΟΠΤΙΚΕΣ ΙΝΕΣ Θεωρία-Εφαρμογές-Πειράματα-Προσομοιώσεις,” ΤΖΙΟΛΑ, 2010*

[4] Cisco VNI 2012, available at http://www.cisco.com

[5] “*Software-Defined Networking: The New Norm for Networks*,” OPEN NETWORKING FOUNDATION report, 2012

[6] Jinno M., Ohara T., Sone Y., et al, “*Introducing Elasticity And Adaption Into The Optical Domain,”* NTT Network Innovation Laboratories, NTT Corporation

[7] Willner A.E., Yang J.U., WU X., “*Optical Performance Monitoring To Enable Robust And Reconfigurable Optical High-Capacity Networks*,” Military Communications Conference, 2009.

[8] “Optical performance monitoring techniques for heterogeneous transmission systems,” ICT-CHRON project deliverable D4.2.

[9]Dr Fitton M., “*Principals Of Digital Modulation*,” TOSHIBA TREL, available at http://www.berk.tc/combas/digital_mod.pdf

[10] Κανατάς Α., Κωνσταντίνου Φ., Πάντος Γ., “*Συστήματα Κινητών Επικοινωνιών*,” chapter-4, Παπασωτηρίου 2010.

[11]figure available at http://encyclopedia2.thefreedictionary.com/QAM4

[12] Dris S., "*Performance of the CMS Tracker Optical Links and Future Upgrade Using Bandwidth Efficient Digital Modulation*," PhD Thesis, Dept. of Physics, Imperial College London, 2006

[13] figure available at http://happy.emu.id.au/lab/tut/dttb/dtbtut5b.htm

[14] Schmogrow R., Nebendahl B.,Winter M., et al ,"*Error Vector Magnitude as a Performance Measure for Advanced Modulation Formats*," IEEE PHOTONICS TECHNOLOGY LETTERS, 2012

[15] Shafik R., Rahman S., AHMR Islam, "*On the Extended Relationships Among EVM, BER and SNR as Performance Metrics*," 4th International Conference on Electrical and Computer Engineering ICECE, 2006

[16] "HFTA-05.0: Statistical Confidence Levels for Estimation BER Probability," MAXIM Application Note 703: Oct 26, 2000.

[17] D. R. Pauluzzi, NC. Beaulieu, "*A Comparison of SNR Estimation Techniques for the AWGN Channel*," IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 48, NO. 10, OCTOBER,2000

[18] Κωνσταντίνου Φ., Καψάλης Χ.,Κωττής Π., "*Εισαγωγή Στις Τηλεπικοινωνίες*," Παπασωτηρίου, 1995

Appendix A - Statistic Metrics and Errors

Metrics

N = total number of experiments

x_i = value of the i experiment, i = 1,2,....,N

a) Mean Value

$$\bar{X} = \frac{\sum_{i}^{N} x_i}{N}$$

b) Standard deviation

$$std = \sqrt{\frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{x_i} - \bar{X})^2}{N}}$$

or

$$std = \sqrt{\frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{x_i} - \bar{X})^2}{\mathrm{N} - 1}}, \qquad for\ samlpes$$

c) Coefficient of variation

$$CV = \frac{std}{\bar{X}} = 100 * \frac{std}{\bar{X}}\%$$

Errors

$\widehat{\rho_\iota}$ = estimation value of i experiment, ρ = real value

a) Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^{N} |\widehat{\rho_\iota} - \rho|}{N}$$

b) Mean Square Error

$$MSE = \frac{\sum_{i=1}^{N} (\widehat{\rho_\iota} - \rho)^2}{N}$$

c) Root Mean Square Error

$$RMSE = \sqrt{MSE}$$

d) Normalized Mean Square Error

$$NMSE = \frac{1}{\rho^2} * \frac{\sum_{i=1}^{N} (\widehat{\rho_\iota} - \rho)^2}{N}$$

e) Normalized Root Mean Square Error

$$NRMSE = \sqrt{NMSE}$$

f) Bias

$$Bias = \frac{\sum_{i=1}^{N} (\widehat{\rho_\iota} - \rho)}{N}$$

g) Absolute Normalized Bias

$$ANBias = \left| \frac{1}{\rho} * \frac{\sum_{i=1}^{N} (\widehat{\rho_\iota} - \rho)}{N} \right| = \left| \frac{1}{\rho} * \frac{\sum_{i=1}^{N} (\widehat{\rho_\iota} - \rho)}{N} \right| * 100\ \%$$

Appendix B - MATLAB Programs

Program 1: EVM calculation

```
M = input(' Choose Modulation order (4,16,64....,4096) M =');
% Modulation order (M=4 for QPSK, M=16 for 16QAM, M=32 for 32QAM,
etc.)
noSyms = 1000;      % Number of symbols to simulate
% Calculate number of simulated bits from number of symbols
noBits = noSyms*log2(M);
% Create modulation and demodulation objects (see MATLAB help) with
gray encoding
modObj = modem.qammod('M', M, 'SymbolOrder', 'gray');
demodObj = modem.qamdemod('M', M, 'SymbolOrder', 'gray');
ideal_point = modulate(modObj,[0:M-1]);

%average_power_calculation
average_power_of_ideal_points = 0;

for i=1 : length(ideal_point)
    average_power_of_ideal_points = average_power_of_ideal_points
+real(ideal_point(i))^2 +imag(ideal_point(i))^2 ;
end
average_power_of_ideal_points  =
sqrt(average_power_of_ideal_points/length(ideal_point));

    k = 0;
for snr = 1:0.1:35
k = k+1;
SNR(k) =snr;
% Create random input symbols (symbol alphabet: 0-3)
Data_in = randi([0 M-1],noSyms,1);
% Input symbols converted to bits
Bits_in = dec2bin(Data_in);
% Perform baseband (NOT passband!) modulation using input symbols -
The result
% is a sequence of complex QAM symbols, each having a specific
amplitude and phase
IQin = modulate(modObj,Data_in);
% Add noise to modulated QAM signal to simulate transmission in a
noisy AWGN channel
IQawgn = awgn(IQin,snr,'measured');
% Use demodulation object to detect received, noisy symbols
Data_out = demodulate(demodObj,IQawgn);
% Convert detected symbols to bits
Bits_out = dec2bin(Data_out);

%EVM caclulation
ideal_point = modulate(modObj,[0:M-1]);


%True EVM
true_EVM = 0;
for i =1 : noSyms
    true_EVM = true_EVM + (real(IQin(i))-real(IQawgn(i)))^2 +
(imag(IQin(i))-imag(IQawgn(i)))^2;
```

```
end
    tEVM(k)= 100*sqrt(true_EVM/noSyms)/average_power_of_ideal_points;

%Blind EVM
nearest_ideal_point = ideal_point(Data_out+1);
received_EVM = 0;
for i =1 : noSyms
    received_EVM = received_EVM +(real(IQawgn(i))-
real(nearest_ideal_point(i)))^2 + (imag(IQawgn(i))-
imag(nearest_ideal_point(i)))^2;
end
    bEVM(k) =
100*sqrt(received_EVM/noSyms)/average_power_of_ideal_points;

end
```

Program 2: SNR thresholds calculation for BER <0.85*10^-3

```
safety_margin  = 0.15*(10^-3);

for i= 1 : 4
M = 4^i;  k=0;
noSyms = 9*10^3;     % Number of symbols to simulate
% Calculate number of simulated bits from number of symbols
noBits = noSyms*log2(M);
% Create modulation and demodulation objects (see MATLAB help) with
gray encoding
modObj = modem.qammod('M', M, 'SymbolOrder', 'gray');
demodObj = modem.qamdemod('M', M, 'SymbolOrder', 'gray');

for snr = 1:0.01:35
k=k+1;
SNR(k) = snr ;         % Set SNR

% Create random input symbols (symbol alphabet: 0-3)
Data_in = randi([0 M-1],noSyms,1);
% Input symbols converted to bits
Bits_in = dec2bin(Data_in);
% Perform baseband (NOT passband!) modulation using input symbols -
The result
% is a sequence of complex QAM symbols, each having a specific
amplitude and phase
IQin = modulate(modObj,Data_in);
% Add noise to modulated QAM signal to simulate transmission in a
noisy AWGN channel
IQawgn = awgn(IQin,snr,'measured');
% Use demodulation object to detect received, noisy symbols
Data_out = demodulate(demodObj,IQawgn);
% Convert detected symbols to bits
Bits_out = dec2bin(Data_out);
% Calculate bit error rate (BER)
Bit_error = Bits_out - Bits_in;                   % Compare detected
bits to input bits
Total_Bit_Errors = length(find(Bit_error));       % Count how many bits
are different
BER(k) = Total_Bit_Errors/noBits  ;                   % Divide bit
errors by total number of bits transmitted to get BER

[GeneralBER995(i,k), CL] = berconfint(Total_Bit_Errors,noBits,0.995);
GeneralBER995hi(i,k) = CL(2);
GeneralBER995lo(i,k) = CL(1);
end
end

for i=4:-1:1
    M=4^i;
    down_lim(i,1) = M;
    k = 3401;

    while (GeneralBER995(i,k) < (10^-3) -safety_margin)
        k = k-1;
    end
        down_lim(i,2) = SNR(k+1);
end
```

Program 3: Monitoring of an optical network – Simulation

```
load('new_LUT_bEVM_SNR_estimations_01.mat') % contains all global
tables

% Create modulation and demodulation objects with gray encoding
 modObj(1) = modem.qammod('M', 4, 'SymbolOrder', 'gray');
 demodObj(1) = modem.qamdemod('M', 4, 'SymbolOrder', 'gray');
 modObj(2) = modem.qammod('M', 16, 'SymbolOrder', 'gray');
 demodObj(2) = modem.qamdemod('M', 16, 'SymbolOrder', 'gray');
 modObj(3) = modem.qammod('M', 64, 'SymbolOrder', 'gray');
 demodObj(3) = modem.qamdemod('M', 64, 'SymbolOrder', 'gray');
 modObj(4) = modem.qammod('M', 256, 'SymbolOrder', 'gray');
 demodObj(4) = modem.qamdemod('M', 256, 'SymbolOrder', 'gray');

Baud_rate = 10^6;
Noise_time_unit = 2^14; % Length in bits
L = 2^14;   % packet length = bits/packet
Packets_per_unit = Noise_time_unit/L;
Total_size_bytes = 4*2^20; % 4 MB
Time_units = 8*Total_size_bytes /Noise_time_unit;
Block_size = 1;
T_SNR = 10; % expected duration for SNR value, Poisson distribution
(Lamda value)
freq = zeros(1,4); % format frequency

counting = false;
j = 0;
while (j <= Time_units)

    Duration_Of_SNR =  poissrnd(T_SNR,1,1);
    Value_Of_SNR =   9.96 +  (34.9-9.96)*rand(1,1);
    for i = 1 : Duration_Of_SNR
        if (j+i) <= Time_units
            SNR_for_this_unit(j+i) = Value_Of_SNR;
        else
            break;
        end
    end
    j = j + Duration_Of_SNR;
end

i = 0;
for (j = 1 : length(SNR_for_this_unit))
    for jj = 1 : Packets_per_unit
    i = i+1;
        SNR_for_this_packet(i) = SNR_for_this_unit(j);
            if  (SNR_for_this_packet(i)> 28.57)
                ideal_M(i) = 256;

            elseif (SNR_for_this_packet(i) > 22.7)
                ideal_M(i) = 64;
            elseif (SNR_for_this_packet(i)> 16.7)
                ideal_M(i) = 16;
            elseif (SNR_for_this_packet(i) > 9.95)
                ideal_M(i) = 4;
            end

    end
end
Total_number_of_packets = length(SNR_for_this_packet);
```

```
Overall_Bit_Errors = 0;                 % Bit errors of all packets

M = 4;                                  % StartUp Modulation
LUT_01_SNR_ESTIMATION_linear = SNR_for_this_packet(1);
P_loss = 0;     %counter of failed packets
k = 0;          %counter of correct packets
Format_changes = 0;
TranTime = 0;
OverallTime = 0;
C = 0;       % simulation time (packet) counter;
BER_of_correct_packet = 0;
BER_of_failed_packet = 0;

while (C<Total_number_of_packets)
    Last_M = M;
    Block_EVM  = 0;
    choice = floor(log2(M)/2);
    noSyms = floor(L/log2(M)); %Symbols/packet

    packet_of_this_block = 1;

    ideal_point = modulate(modObj(choice),[0:M-1]);
    %average power calculation for the EVM
    average_power_of_ideal_points = 0;
    for i=1 : length(ideal_point)
        average_power_of_ideal_points = average_power_of_ideal_points
+real(ideal_point(i))^2 +imag(ideal_point(i))^2 ;
    end
    average_power_of_ideal_points  =
sqrt(average_power_of_ideal_points/length(ideal_point));

       EVM = 0;
    while (packet_of_this_block <= Block_size)
        C = C+1;
       error(C) = SNR_for_this_packet(C) -
LUT_01_SNR_ESTIMATION_linear;
        SNR_ESTIMATION(C) = LUT_01_SNR_ESTIMATION_linear;
        Data_in = randi([0 M-1],noSyms,1);                     %
Create random input symbols (symbol alphabet: 0-3)
        Bits_in = dec2bin(Data_in);                            %
Input data converted to bits
        IQin = modulate(modObj(choice),Data_in);
% input signal
        IQawgn = awgn(IQin,SNR_for_this_packet(C),'measured'); %
noisy output signal
        Data_out = demodulate(demodObj(choice),IQawgn);
% output data;
        Bits_out = dec2bin(Data_out);                          %
Output data converted to bits

        % Calculate bit error rate (BER)
        Bit_error = Bits_out - Bits_in;
        Total_Bit_Errors = length(find(Bit_error));
        BER = Total_Bit_Errors/L;
        BER_of_packet(C) = BER;

        if counting
            OverallTime = OverallTime + (L/log2(M))/Baud_rate;
            Md(C) = M;
            freq((log2(M)/2)) = freq((log2(M)/2))+1;
            if (BER> 10^-3)
                P_loss = P_loss +1;
                BER_of_failed_packet = BER_of_failed_packet+BER;
            else
                k = k+1;
                BER_of_correct_packet = BER_of_correct_packet+ BER;
                TranTime = TranTime + (L/log2(M))/Baud_rate;
            end
            Overall_Bit_Errors = Overall_Bit_Errors +
Total_Bit_Errors;
        end
        %EVM caclulation
        nearest_ideal_point = ideal_point(Data_out+1);

        for i =1 : noSyms
            EVM= EVM +(real(IQawgn(i))-
real(nearest_ideal_point(i)))^2 + (imag(IQawgn(i))-
imag(nearest_ideal_point(i)))^2;
        end

        packet_of_this_block  = packet_of_this_block +1;
    end

    if ~counting
        counting = true;
        C = 0;
    end
    Block_EVM  =
sqrt(EVM/(noSyms*Block_size))/average_power_of_ideal_points;

    %Start the calculation of SNR estimation with LUT_01

    if M == 4
        LUT_01_SNR_ESTIMATION_linear =
interp1(monotonic_Global_bEVM_M4_01,new_LUT_real_SNR_M4_01,Block_EVM,
'linear');
    end

    if M == 64
        LUT_01_SNR_ESTIMATION_linear =
interp1(monotonic_Global_bEVM_M64_01,new_LUT_real_SNR_M64_01,Block_EV
M,'linear');
    end

    if M == 16
        LUT_01_SNR_ESTIMATION_linear =
interp1(monotonic_Global_bEVM_M16_01,new_LUT_real_SNR_M16_01,Block_EV
M,'linear');
    end

    if M == 256
        LUT_01_SNR_ESTIMATION_linear =
interp1(monotonic_Global_bEVM_M256_01,new_LUT_real_SNR_M256_01,Block_
EVM,'linear');
    end
    %Finished the calculation of SNR estimation with LUT_01
    %Choose the optimal next M
    if  (LUT_01_SNR_ESTIMATION_linear > 28.57)
        M = 256;

    elseif (LUT_01_SNR_ESTIMATION_linear > 22.7)
        M = 64;
    elseif (LUT_01_SNR_ESTIMATION_linear > 16.7)
        M = 16;
    elseif (LUT_01_SNR_ESTIMATION_linear > 9.95)
        M = 4;
    end

    if Last_M ~= M
        Format_changes = Format_changes + 1;
    end
    C
end

Total_number_of_correct_packets = k
Total_number_of_failed_packets = P_loss
Total_number_of_packets = k + P_loss;
Packet_failure_Prob = 100*P_loss/Total_number_of_packets
Overall_BER = Overall_Bit_Errors/(Total_number_of_packets*L)
CV_BER = 100*std(BER_of_packet)/Overall_BER
Range = range(BER_of_packet)
Overall_BER_of_correct_packets = BER_of_correct_packet/k
Overall_BER_of_failed_packets = BER_of_failed_packet/P_loss;
Format_changes
Average_Suc_bit_rate = L*k/TranTime
Average_Overall_bit_rate = L*Total_number_of_packets/OverallTime
MEAN_ERROR = mean(error)
CV_ERROR = 100*std(error)/MEAN_ERROR
freq  = 100*freq/Total_number_of_packets
```

Appendix C - Statistical Evaluation of the Proposed Algorithm

The following plots show the Normalized Mean Square of the proposed algorithm's SNR estimations. We will use the LUT_01 method with linear interpolation and rounding to the nearest stored in global table point for SNR from 1dB to 35 dB with step 0.05 dB.

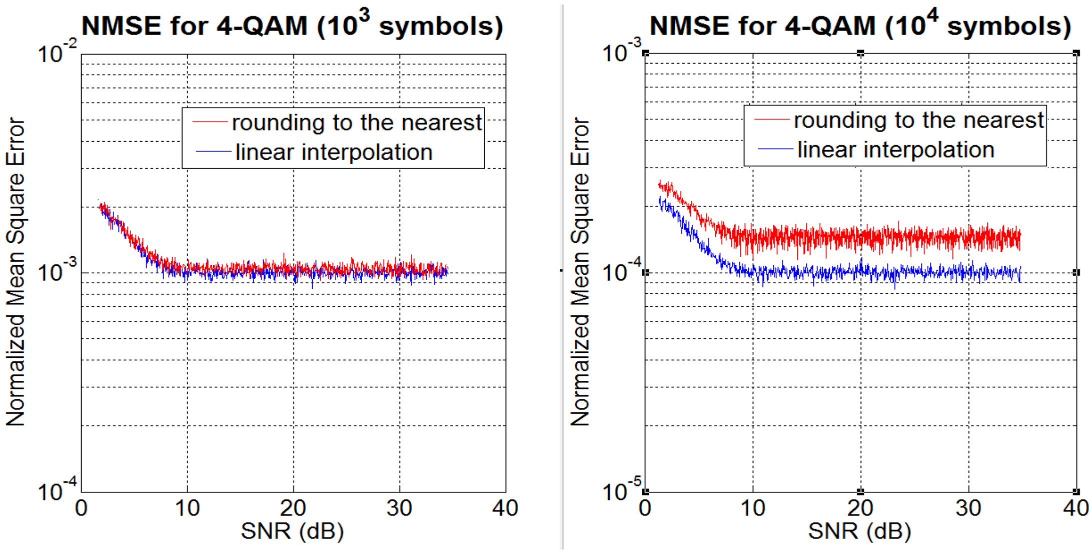


Figure C.1 Normalized Mean Square Error for 4-QAM, 10^3 symbols and 10^4 symbols

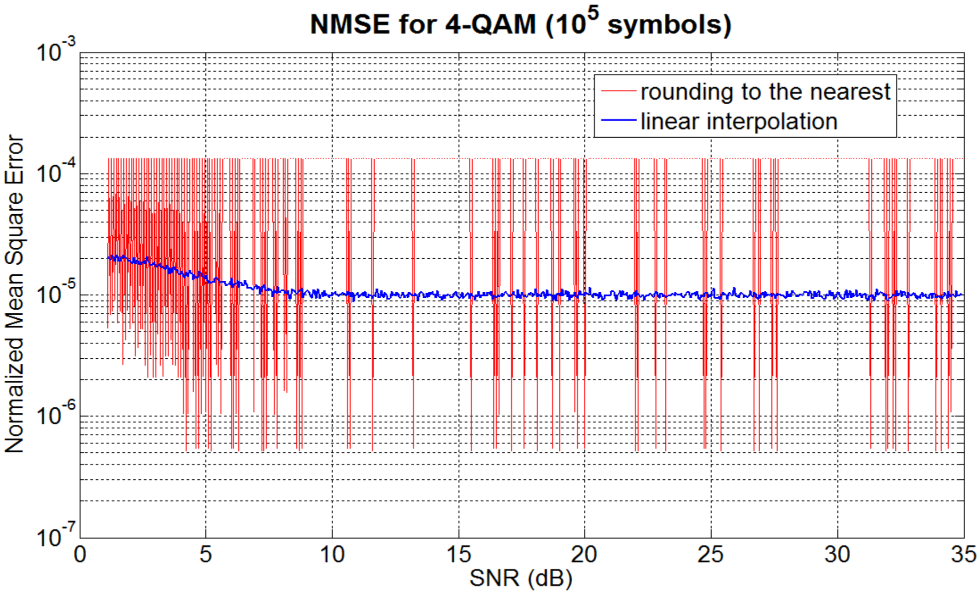


Figure C..2 Normalized Mean Square Error for 4-QAM, 10^5 symbols

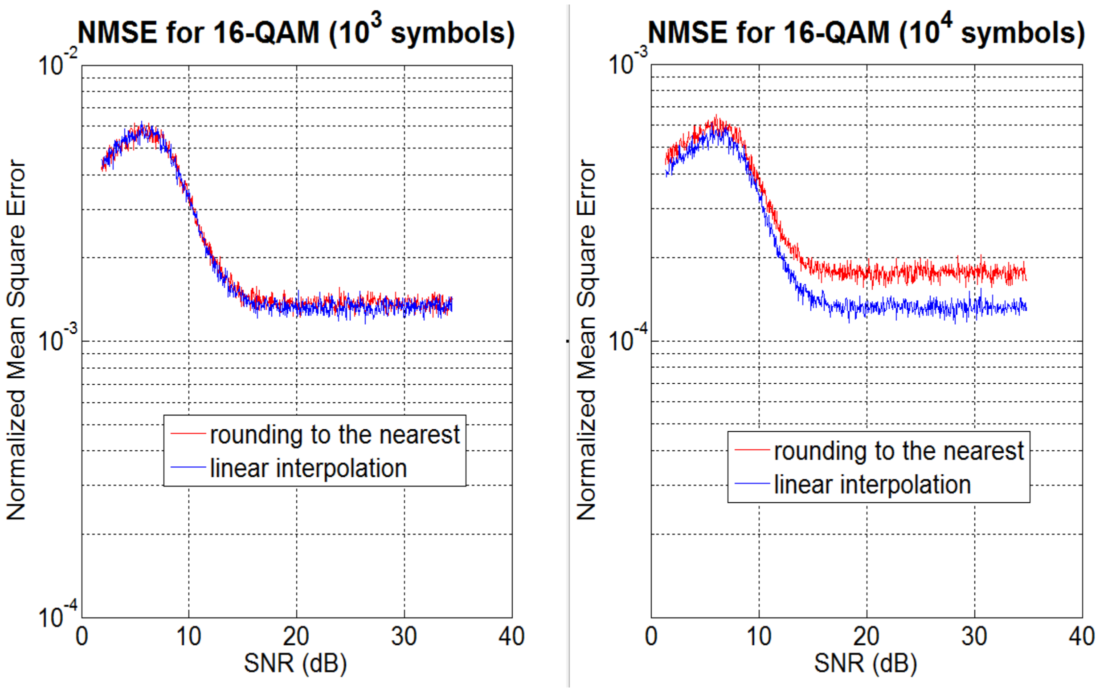


Figure C.3 Normalized Mean Square Error for 16-QAM, 10^3 symbols and 10^4 symbols

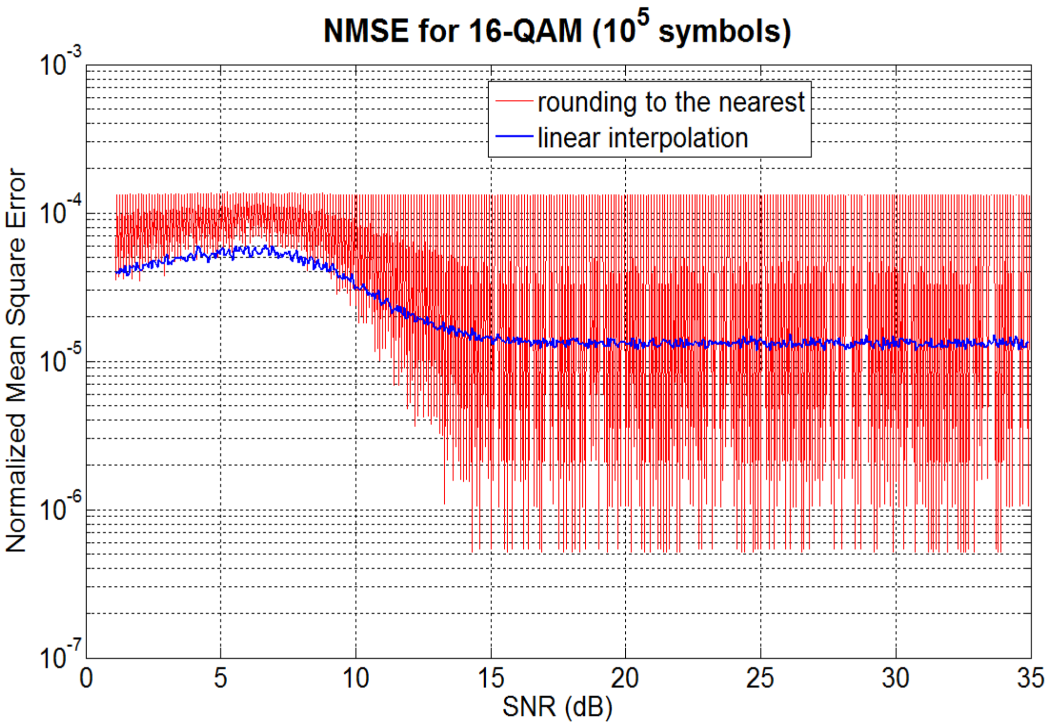


Figure C.4 Normalized Mean Square Error for 16-QAM, 10^5 symbols

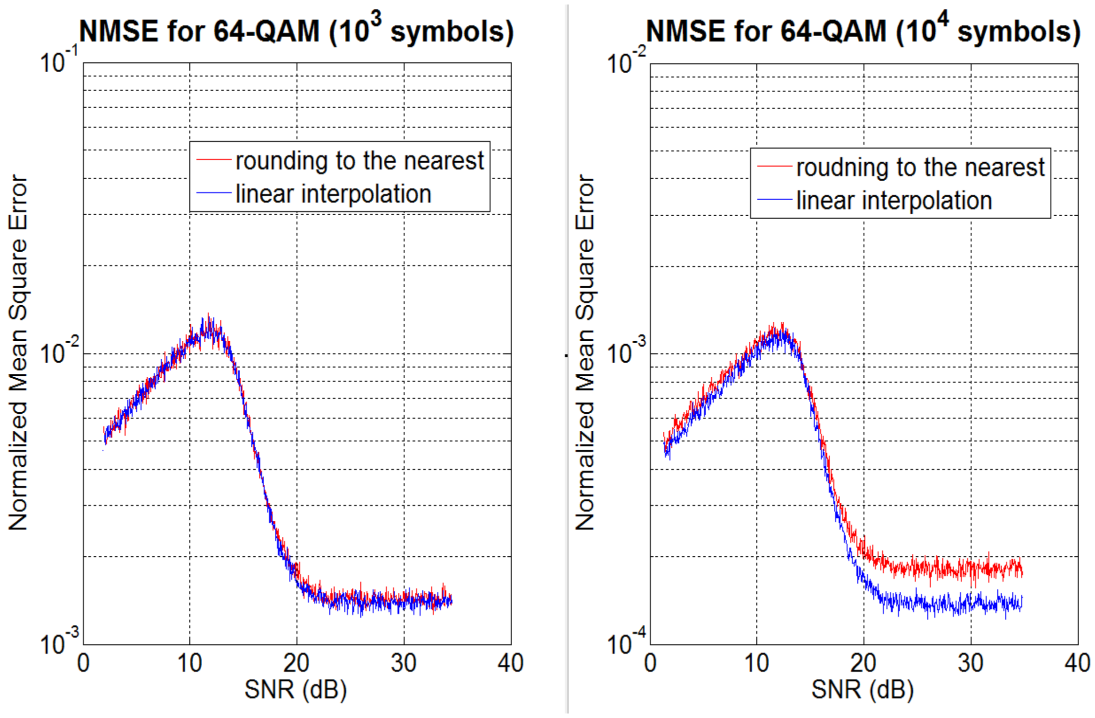


Figure C.5 Normalized Mean Square Error for 64-QAM, 10^3 symbols and 10^4 symbols

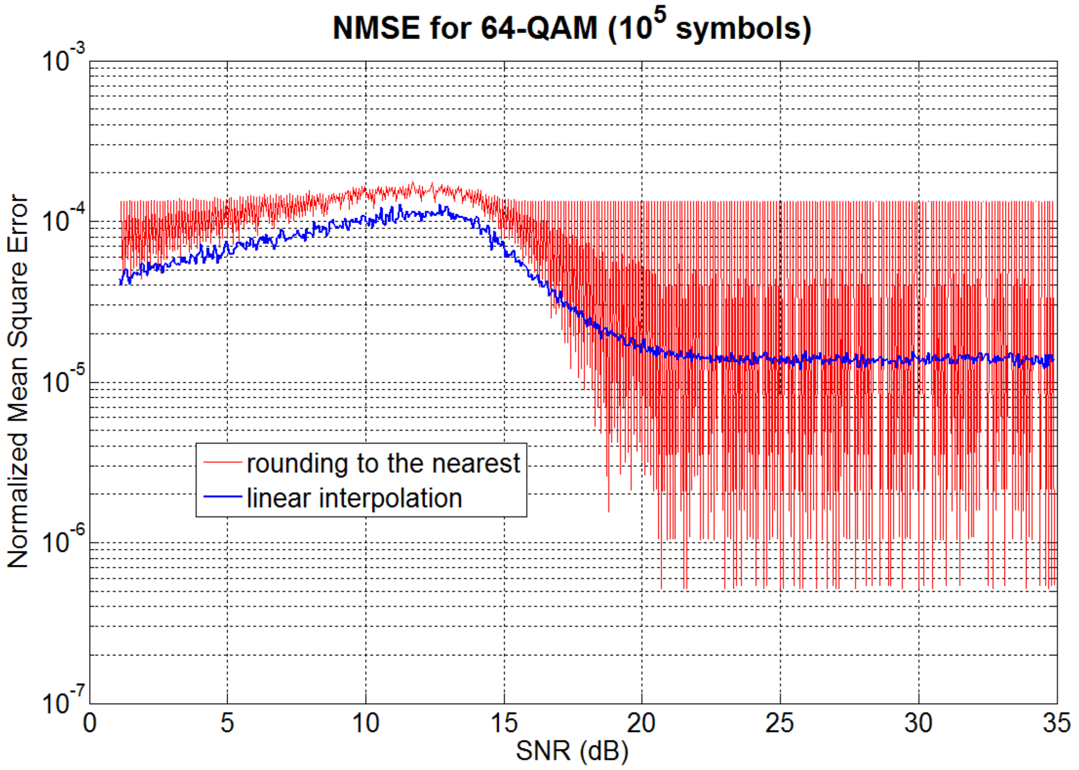
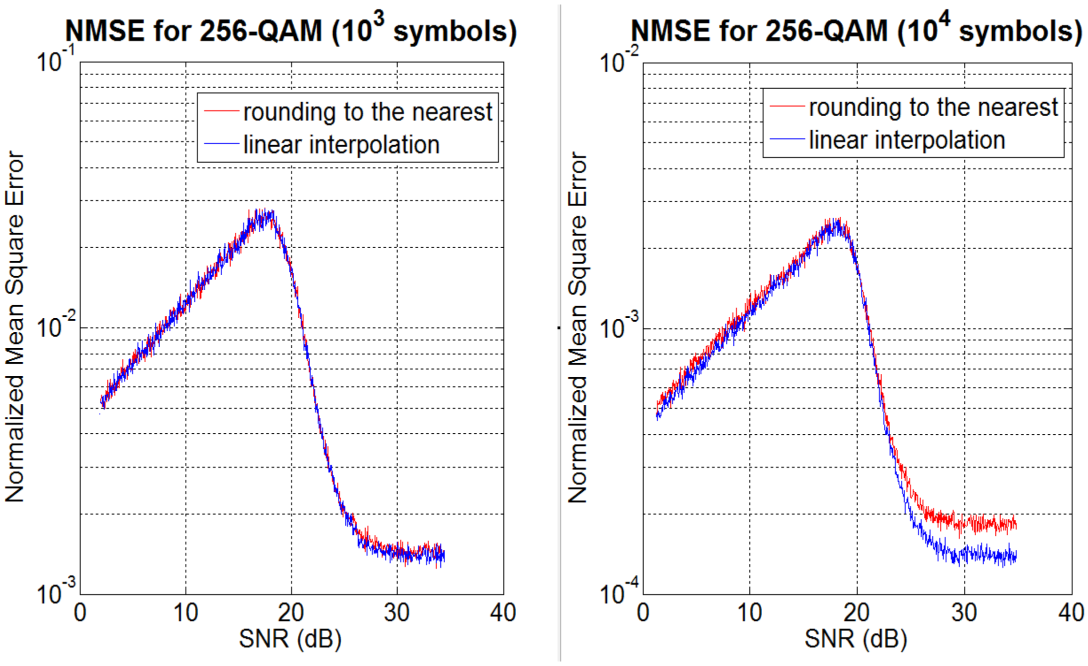


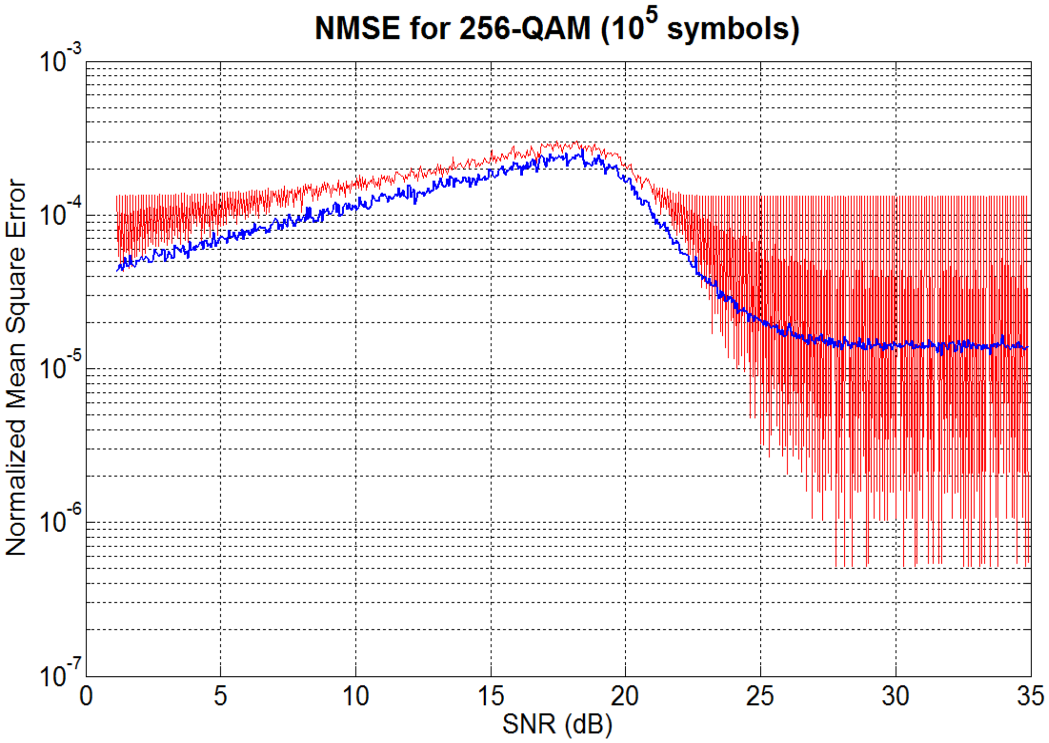
Figure C.6 Normalized Mean Square Error for 64-QAM, 10^5 symbols



NMSE for 256-QAM (10^3 symbols)

NMSE for 256-QAM (10^4 symbols)

Figure C.7 Normalized Mean Square Error for 256-QAM, 10^3 symbols and 10^4 symbols



NMSE for 256-QAM (10^5 symbols)

Figure C.8 Normalized Mean Square Error for 256-QAM, 10^5 symbols