ANNEE 2015 / 2016

RAPPORT de « MISSION en ENTREPRISE »

présenté par

"Minakhmetov Artur"

du Master of science : **«M2 Réseaux Optiques & Systèmes Photoniques»**

Mission effectuée du 07/03/2016 au 31/08/2016 chez :

Nokia Bell Labs

sujet de la mission :

"Etude du codage pour la compensation d'effets non-linéaires"

Directeur de stage : **"LAYEC Patricia"**

Conseiller de stage : **"DUBREUIL Nicolas"**

Contents

Summary

This document report on the study of the compensation of non-linear effect called Cross-Polarization Modulation (XPolM) in optical communications. Existing compensation technique called Generalized Maximum Likelihood for Cross Polarization Modulation (GML for XPolM) was investigated. Named technique delivers hard detection, thus making a possibility on it basis to deliver soft detection, which uses probabilistic approach. A soft detection let to use Forward Error Correcting (FEC) Codes based on with Low-Density Parity-Check (LDPC) codes. The codes enable to correct the errors that were introduced by XPolM effect. The soft-detection model of the channel perturbed by XPolM was proposed. It was implemented in computer simulations with help of a number of different algorithms. Each algorithm was reviewed from points of view of performance and complexity. Between such algorithms one were recommended for further development and investigation in real-life cases.

As well the document briefly covers the questions of choice of the place of internship, review of the company of the internship; report on internship experience and off-work activities connected with internship place.

0. Presentation of the internship

0.1. Presentation of Nokia and Bell Labs

The place of the stage is Nokia Bell Labs – subsidiary company of Nokia. Bell Labs has been acquired by Nokia in 2016 along with previous owner Alcatel Lucent.

Bell Labs is known for its invention in the world, among them most known are:

- Transistor – invented by Walter Houser Brattain in 1947.
- Laser – invented by Arthur Schawlow in 1957.
- Charge Coupled Device – CCD – invented by Willard Boyle and George Smith in 1969.
- UNIX OS – developed by Ken Thompson, Dennis Ritchie, and others in 1970.
- C programming language – was originally developed by Dennis Ritchie between 1969 and 1973.

Currently Bell Labs perform scientific work in different domains connected with telecommunication whether with fixed or mobile networks. Its course of action perfectly serves as a scientific base for innovations for equipment developed and provided to the market by Nokia – the equipment to work with mobile and fixed networks on the limits of speed – 4G and 5G.

The mother company – Nokia were founded in 1865 year in Finland. Interesting to notice that there is a town in Finland called Nokia, exactly in that town the second factory of Nokia was founded and took a name as company. The company was famous for different domains of expertise – paper industry, energy sector, tires and heavy and light machinery. At the second part of the 20^{th} century Nokia started to deal with telecommunication business. At the end of 20^{th} century Nokia became known for its mobile phones around the world. However at 2013 Nokia sold its Mobile Phone business to Microsoft and stayed on the marked of telecommunication equipment. In 2016 Nokia acquired French-American Company Alcatel Lucent together with its Bell Labs research division.

The company performs very well in financial way as reported below for the year 2015:

- Revenue 23 Billion EUR.
- Profit 1.68 Billion EUR.

The company offers to its employee's variety of off-job activities making the work more interesting – sports (optical department plays team Frisbee), music festival – during a week in a year employees do musical performance during lunch time.

The site of the internship – the optical communication department in Villarceaux, Nozay – occupies itself with scientific work in optical communication domain in long-haul communications and networks. They study different optical effects and manage them in order to increase bit-rate and distance of transmission. One of the themes was destined to study the possibilities of compensation of cross-polarization modulation (XPolM) effect in order to increase the bit-rate of the connection over the distance.

0.2.The motivation for this internship

The thematic of M2 Master program ROSP – Réseaux Optiques et Systèmes Photoniques – specifically turned around optical communications domain. That is why I turned my view on the proposals of internships in Bell Labs in optical communication domain. I thought that the internship experience would be a great step in understanding the domain and be crucial in my education in France in newly created University of Paris Saclay.

0.3.Personal experience of the internship

The internship gave the chance to:

- Get to know what it is like to be an engineer in an international company – get in collaboration with bright, supportive and kind engineers.
- Share the team spirit, listen, interpret, understand and execute the required task.
- Share the ideas, receive the critique and use it to deliver the best of mine.
- Have the difficulties in front of me (understanding the state of the art of the problem) and after successful resolution to feel the joy of achieving something.

During the internship I enhanced my hard and soft skills, such as:

- Matlab and C Programming – all of the simulations performed on Matlab and C language.
- Time Management – performing a lot of work in limited time.
- Human relations – find friends and establish good relations with company staff.

0.4.Corporate social responsibility

Nokia issues each year a report on Corporate Social Responsibility theme. Most recent one could be found here:
http://company.nokia.com/sites/default/files/download/nokia_people_and_planet_report_2015.pdf

We citing the key points Nokia achieved in the RSE at 2015:

- Environmental:
    - Decreased power consumption by 7%,
    - Decreased gas emission by 12%,
    - 51% of electricity went from renewable resources.
- Society:
    - 100% of employees are engaged.
    - Nokia has no concerns for unethical behavior.
    - Improve people's lives with technology.
- Economic:
    - Creating business opportunities and jobs in India – Nokia employed 11 588 people in India.
    - Contribute to the development of China's economy, technology and culture.
    - Connect the unconnected – "According to the World Bank, a 10% increase in high-speed internet connections leads to a 1.3% increase in economic growth. In 2015, we had sales in approximately 130 countries and agreed on improving network capacity and coverage in various emerging countries such as Colombia, Kenya, Vietnam, China and India."

1. Introduction to the optical telecommunications

1.1. Optical transmission

Fast and reliable long-haul communications are possible thanks to optical networks based on optical fibers, but to respect all three aspects of communication – speed, reliability and distance – is a hard work. The distance limits of reliable transmission are changing with bit-rate.

- When we say fast, we talk about bit-rate – number of bits send over a second. The range of bit-rate varies from applications:
  - The standard bit-rate for an optical communication channel used by one person (fiber to the home or FTTH) is 10 Gbit/s.
  - The multichannel that carries several individual channels could go up to 100 Gbit/s and even up to 400 Gbit/s per fiber. Such multichannels are used in metropolitan area networks and in intercontinental communication.

- When we say reliable, we talk about Bit Error Rate (BER) – part of bits received with error with respect to the number of bits sent. The systems in general should have 10^{-15}level of bit error rate. The optical systems always use forward error correcting (FEC) codes, such codes help to correct the errors and thus decrease the power of signal sent.

- When we say long-haul communications, we talk about sending data over thousands of kilometer.

The bigger the bit-rate – the better, and to increase the bit-rate there are several well known techniques, such as (illustrated at Figure 3):

- Use Polarization Division Multiplexing (PDM) – modulate the signal on two possible state of polarization of light – vertical and horizontal

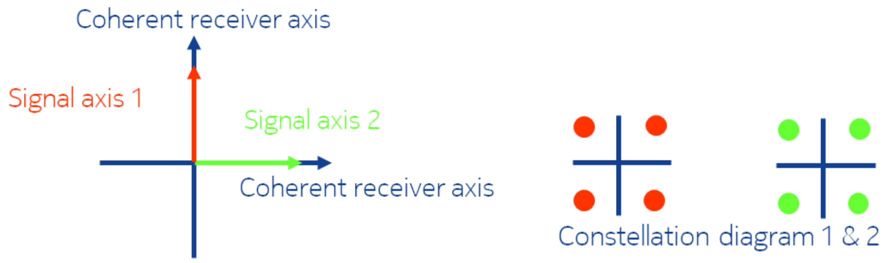


Figure 1 Representation of polarization multiplexing on the example of QPSK

- Use phase modulation as well as amplitude modulation of light with biggest number of states possible, or in other words – use constellation of higher order. In the context of this work we will use QPSK (Quadrature Phase-Shift Keying) modulation and 16 QAM (16 Quadrature Amplitude Modulation). The first one can transmit 4 distinct symbols and 16 QAM has a possibility to transmit 16 ones.

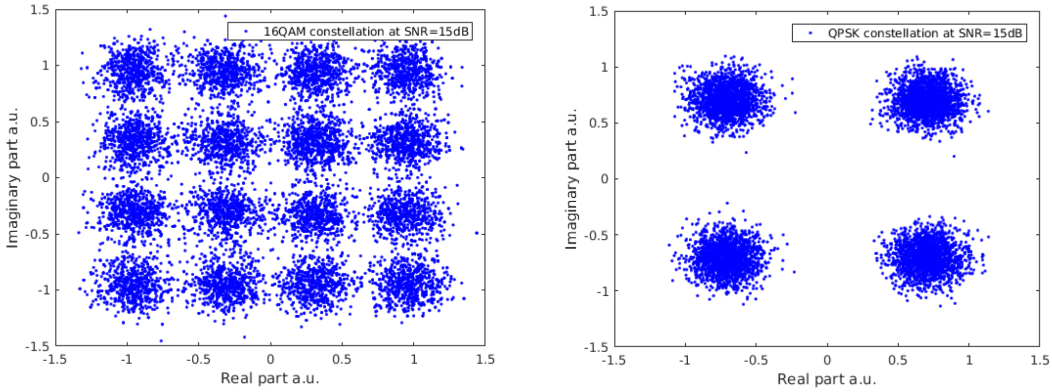


Figure 2 Example of constellations (16 QAM left and QPSK right)

- Use Wavelength Division Multiplexing (WDM)– send the signal on several wavelengths. Each wavelength transports two polarization states modulated with different constellation schemes. The number of wavelength could go up to 100, but generally in long haul communication networks the number of wavelength is typically 80.

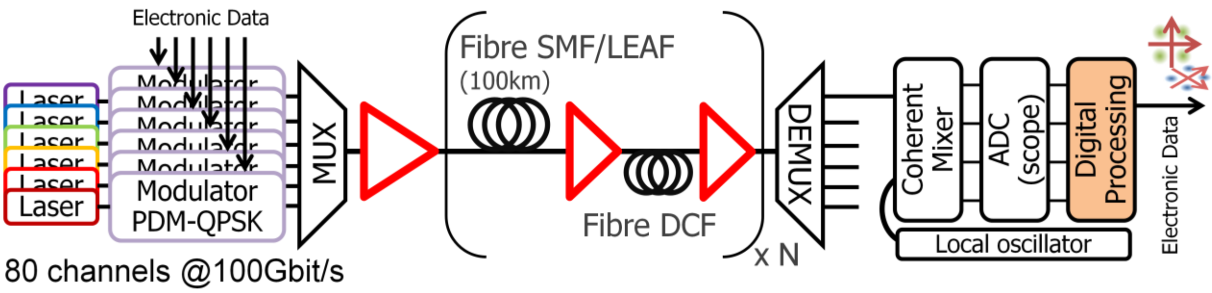


Figure 3 Optical Transmission model with wavelength multiplexing

1.2. Optical impairments

In order to send the signal for long distances, we have to take into account the effect of attenuation of the signal – if one want to reach long distances of transmission the usage of signal amplifiers is necessary, but the less possible number of amplifiers on routes is preferable, nevertheless, currently in Europe the existing scheme of amplifiers is used, and it might not be optimized for contemporary use This leads to the fact that we need to send signal with sufficient power in order to keep distinguishable shape of the modulated signals at the receiver side.

Increase of the power sent is natural necessity of high order modulation formats – for example for 16 QAM modulations we have to sent more power per symbol on the board of constellation than in QPSK. Increasing the power of sent signal results in a fact that signal sent becomes perturbed by non-linear effects, such as Kerr effects which include:

- Four-wave mixing – the channels on different wavelengths perturb each other, resulting in conversion of some part of the signal of the channel on original wavelength to some other wavelength.
- Self phase modulation – modulation of the phase of the signal under varying refractive index of the fiber, induced by high power of the same signal.

- Cross phase modulation – same effect as in the previous case but refractive index variation caused by high power of adjacent channel.
- Cross polarization modulation (XPolM) – modulation not only of the phase, but also of the amplitude of two polarizations of the signal under the influence of the adjacent channel through the Kerr Effect.

These effects result in deterioration of BER or increase of number or errors. When compensation takes place we can increase the distance of reliable transmission with required bit-rate (Figure 4).

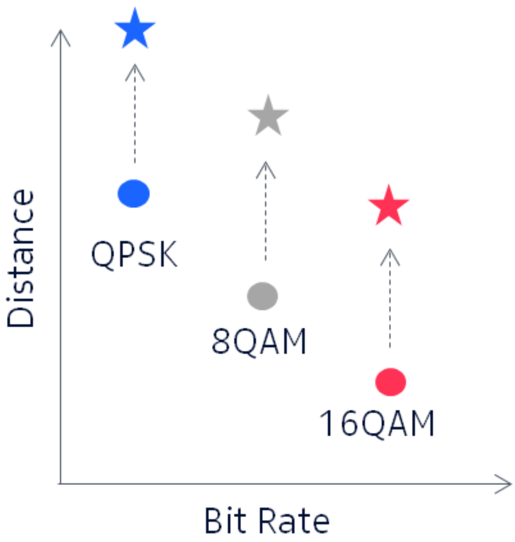


Figure 4 Result of impairments compensation

We consider the classic wavelength and polarization division multiplexing system with modulation formats QPSK and 16 QAM. To increase the throughput of these systems (bit-rate over distance), we have a look during the internship on the means of compensation of nonlinear effects, in particular the effects of XPolM.

2. Objectives of the internship – introduction of Soft-Output

Different techniques exist to mitigate XPolM effect but all of them deliver hard decision – constellation symbols directly translated to the bits. Decoded bits do not contain probabilistic information on them.

The objective of this work is to introduce the decoder that will deliver the probabilistic information on the bits. Such approach is called delivering Soft Decision or Soft-Output. As it will be shown later, such decoders will be based on Hard Output Generalized Maximum Likelihood Method for Compensation of XPolM (HO GML XPolM).

Soft Decision is needed for Forward Error Correcting (FEC) codes are currently used in commercial product as they significantly improve the quality of transmission, or BER. Such FEC are based on bitwise log-likelihood-ratio (LLR) with their application on Low-Density Parity-Check (LDPC) code decoder

We will introduce next changes in Digital Processing Architecture of XPolM compensation:

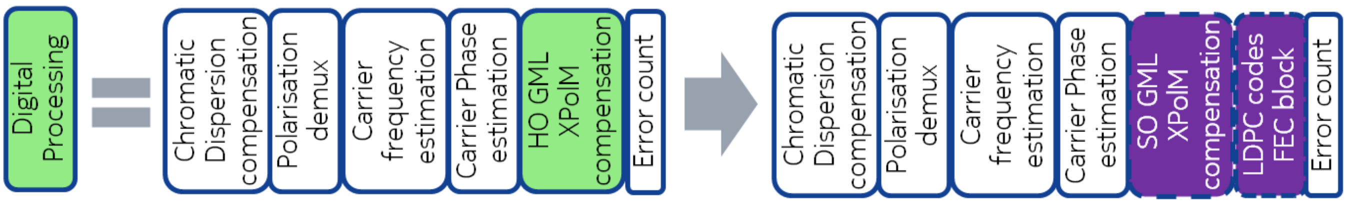


Figure 5 Introduction of the Soft-Output Algorithms.

Thus, the main objective of internship work is to adapt block "HO GML XPolM" to FEC block by introduction of Soft Output GML XPolM (SO GML XPolM) with application of LDPC codes in order to improve detection characteristics.

3. Compensation of the XPolM

3.1. XPolM channel model

The effect of XPolM on sent signals for kth channel use in the absence of all other effects – is described by the next model:

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x} + \boldsymbol{z}_k$$

Where $\boldsymbol{x}_k = \begin{pmatrix} x_{1,k} \\ x_{2,k} \end{pmatrix}$ – is the column vector of sent signal – the signal is sent on two polarizations, and $x_{1,k}$ and $x_{2,k}$ – represent amplitude/phase modulated symbol from constellation – BPSK, QPSK, 16 QAM, etc. The $\boldsymbol{z}_k = \begin{pmatrix} z_{1,k} \\ z_{2,k} \end{pmatrix}$ – is the column vector of complex additive white Gaussian noise with zero mean on two polarizations added to the sent signal and generated by the amplifiers on route and represented by amplified spontaneous emission ASE of amplifiers on route. The $\boldsymbol{H}_k$ – is a 2x2 matrix representing the XPolM effect. This matrix could be modeled as:

$$\boldsymbol{H}_k = \begin{bmatrix} \sqrt{1-\alpha_k{}^2} & \beta_k \\ \alpha_k & \sqrt{1-\beta_k{}^2} \end{bmatrix}$$

Where α_k and β_k – cross-talk coefficients – are zero-mean complex Gaussian processes, independent of symbols and ASE. These coefficients have temporal correlation, and their auto-correlation function is defined by:

$$R[k] =< \alpha_n^* \alpha_{n+k} >=< \beta_n^* \beta_{n+k} >$$

Where $<.>$ stands for ensemble average and * stands for complex conjugate. The XPolM effect is represented by maximum value $R_{max} = R[0]$ - at the origin of the autocorrelation function. Such model of **H** matrix is proposed in [1], more general model is proposed and verified experimentally in [2]. The variations of the channel are slower than the Baud-rate of the channel, as it is shown in [1]; in [2]it is said that we can model autocorrelation length by 10 symbols in average, thus making the compensation of such nonlinear effect feasible.

Finally $\boldsymbol{y}_k = \binom{y_{1,k}}{y_{2,k}}$ - the signal that was received by receiver on two polarizations.

The model of such channel is considered to be 2x2 MIMO, thus we can inherit all the techniques that are used for MIMO detection in wireless channel and apply them when needed.

3.2.State of the art

There are several techniques used for XPolM compensation, at first we consider next ones:

- **Nonlinear Polarization Crosstalk Canceller (NPCC)** presented in [1].
  In this model next noiseless channel model is considered:
  $$\begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix} = \begin{bmatrix} \sqrt{1-\alpha_k{}^2} & \beta_k \\ \alpha_k & \sqrt{1-\beta_k{}^2} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}$$
  The coefficients α_k, β_k are calculated as follows:
  $$\begin{cases} \beta_k = \dfrac{y_{1,k} - \widehat{x_{1,k}}}{\widehat{x_{2,k}}} \\ \alpha_k = \dfrac{y_{2,k} - \widehat{x_{2,k}}}{\widehat{x_{1,k}}} \end{cases}$$
  Where $\widehat{x_{1,k}}$ and $\widehat{x_{2,k}}$ are detected sent symbols directly from $y_{1,k}$ and $y_{2,k}$. After coefficients recovery, these coefficients are later used to recover $x_{1,k}$ and $x_{2,k}$ based on model described previously.
- **XPolM compensator based on Minimum Mean Square Error (MMSE)** estimation of vertical and horizontal polarization states perturbed by XPolM in [3].
  The Idea of such technique is next: use coherent detection and CMA algorithm to recover average slow polarization over long sequence of symbols, and then XPolM compensator recovers average fast polarization and orients axes (with the help of MMSE estimations) of PDM demultiplexer along the average polarization axes.
  The authors try to make a parallel with Poincare sphere. They put as a reference to transmitter constellation a constellation on a Poincare sphere, and then they say that in presence of XPolM the points in such sphere start to scatter in form of balls around

constellation points, then for each symbol (or it could be sequence) they orient the axes of demultiplexer to set this point in alignment with average position.

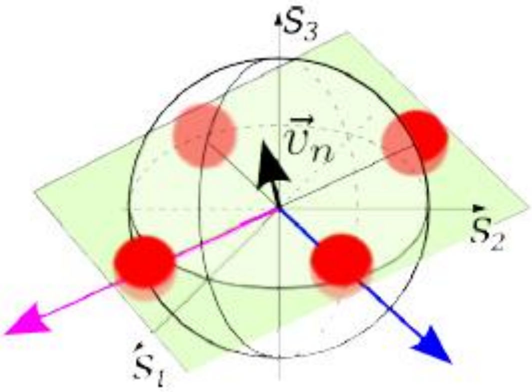


Figure 6 Poincare sphere with constellation representation on it [3].

The disadvantage of such compensation technique lies in the fact that this algorithm is adapted only for constant modulus modulation (16 QAM will work, but not as efficiently as for QPSK), as well as this compensation technique not always better than the case with absence of compensation.

- **Adaptation of Trellis-Coded High-Dimensional Modulation** for mitigation of the cross polarization modulation effect in [4].
  The authors propose to use Viterbi coding of sent differentially coded symbols, recover them after with GLRT algorithm (Generalizes Likelihood Ratio Test) - which takes correlation between unitary modulations and differentially rotated receiving signals.
  The authors claim they their GLRT achieve ML performance.
  The authors use only unitary space-time constellations.

Regarding at the results and the implementation complexity described in the sources these methods seems to be less attractive in compensation of the XPolM effect than presented in next chapter one.

3.3.Generalized Maximum Likelihood method for treatment of XPolM (GML for XPolM) [5]

The goal of the internship is to use the GML formulation to estimate the XPolM channel and propose a novel soft-output fast algorithm to compute it. Thus we will consider that method more precisely. We begin description of this method by highlighting three foundation points:

- The channel is considered to be constant during P[1] (could be variable) number of symbols.
- The estimation of the channel is performed in a developed closed form as a function of received signal and detected sequence, using the prior distribution of crosstalk coefficients.
- The fast tree-search algorithm solving the GML formulation is constructed on Branch-Estimate-and-Bound Algorithm presented in [6].

[1] Further in simulations we will always use P=4, if other case isn't mentioned.

These three points dictate the strategy behind the method and we will review the details of this method fulfilling that strategy.

In order to ease the computation complexity we will consider slightly modified model of the channel under XPolM – due to the fact that cross-talk coefficients α and β are much less than 1, the channel could be modeled without much loss in detector side as:

$$\widetilde{\boldsymbol{H}}_k = \begin{bmatrix} 1 & \beta_k \\ \alpha_k & 1 \end{bmatrix}$$

Taking advantage of the hypothesis of quasi-static channel, it is possible to rewrite the channel model for k^{th} sequence of symbols of length **P**:

$$\boldsymbol{Y}_k = \widetilde{\boldsymbol{H}}_k \boldsymbol{X}_k + \boldsymbol{Z}_k$$

Where $\boldsymbol{X}_k = \begin{pmatrix} \underline{\boldsymbol{x}}_1 \\ \underline{\boldsymbol{x}}_2 \end{pmatrix}_k = \begin{pmatrix} x_{1,1} & \cdots & x_{1,P} \\ x_{1,2} & \cdots & x_{2,P} \end{pmatrix}_k$ - sent sequence and $x_{i,j}$ – symbol from any arbitrary constellations such as BPSK, QPSK and 16 QAM, thus $\boldsymbol{Y}_k = \begin{pmatrix} \underline{\boldsymbol{y}}_1 \\ \underline{\boldsymbol{y}}_2 \end{pmatrix}_k = \begin{pmatrix} y_{1,1} & \cdots & y_{1,P} \\ y_{1,2} & \cdots & y_{2,P} \end{pmatrix}_k$ - received sequence. Further for simplicity the k index is omitted.

In the source article the Generalized Maximum Likelihood (GML) decision criteria is formulated so that the optimal solution is found as estimate of the channel with corresponding sequence:

$$\widehat{\boldsymbol{X}} = \arg\min_{X,H} \|\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{X}\|^{\mathbf{2}}$$

This is equal to the criteria:

$$\left(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{H}}\right) = \left(\widehat{\boldsymbol{X}}, \hat{\alpha}, \hat{\beta}\right) = \arg\max_{\alpha,\beta,\mathbf{X}} \{\Pr(\mathbf{Y}|\mathbf{X}, \alpha, \beta) \Pr(\alpha) \Pr(\beta)\}$$

With assumption of Gaussian a priori probability distribution of noise $\boldsymbol{Z}$ and channel cross-talk coefficients α, β we have:

$$\hat{X} = \arg\min_{X} \left\{ \min_{\alpha} \left\{ \left\| \underline{y_2} - \underline{x_2} - \alpha \underline{x_1} \right\|^2 + \gamma R_{max}^{-1} \|\alpha\|^2 \right\} + \min_{\beta} \left\{ \left\| \underline{y_1} - \underline{x_1} - \beta \underline{x_2} \right\|^2 + \gamma R_{max}^{-1} \|\beta\|^2 \right\} \right\}$$

Where $\gamma = \sigma_n^2$ - regularization factor equal to the power of noise over one symbol and R_{max} - maximum value of autocorrelation function of crosstalk coefficients α, β.

Thus it becomes possible to get a closed form solution for cross-talk coefficients:

$$\begin{cases} \hat{\alpha} = \left(\underline{y_2} - \underline{x_2}\right) \underline{x_1^H} \left(\underline{x_1 x_1^H} + \sigma_n^2 R_{max}^{-1} \boldsymbol{I}\right)^{-1} \\ \hat{\beta} = \left(\underline{y_1} - \underline{x_1}\right) \underline{x_2^H} \left(\underline{x_2 x_2^H} + \sigma_n^2 R_{max}^{-1} \boldsymbol{I}\right)^{-1} \end{cases}$$

Equation 1 Closed form solution of the cross-talk coefficients

Instead of the exhaustive search for the Generalized Maximum Likelihood (GML) solution, we Implement on Branch-Estimate-and-Bound Algorithm presented in [6] with use of partial Euclidian distances (PED):

$$D_{\Delta j} = \min_{\alpha}\left\{\left\|\underline{\boldsymbol{y}}_2^{\Delta j} - \underline{\boldsymbol{x}}_2^{\Delta j} - \alpha \underline{\boldsymbol{x}}_1^{\Delta j}\right\|^2 + \sigma_n^2 R_{max}^{-1}\|\alpha\|^2\right\} + \min_{\beta}\left\{\left\|\underline{\boldsymbol{y}}_1^{\Delta j} - \underline{\boldsymbol{x}}_1^{\Delta j} - \beta \underline{\boldsymbol{x}}_2^{\Delta j}\right\|^2 + \sigma_n^2 R_{max}^{-1}\|\beta\|^2\right\}$$

Equation 2 Definition of the partial distance

Where partial sent symbol sequence and received signal sequence with $\Delta j \in [1..P]$ represented as:

$$\boldsymbol{X}_{\Delta j} = \begin{pmatrix} \underline{\boldsymbol{x}}_1^{\Delta j} \\ \underline{\boldsymbol{x}}_2^{\Delta j} \end{pmatrix} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,\Delta j} \\ x_{1,2} & \cdots & x_{2,\Delta j} \end{pmatrix}$$

$$\boldsymbol{Y}_{\Delta j} = \begin{pmatrix} \underline{\boldsymbol{y}}_1^{\Delta j} \\ \underline{\boldsymbol{y}}_2^{\Delta j} \end{pmatrix} = \begin{pmatrix} y_{1,1} & \cdots & y_{1,\Delta j} \\ y_{1,2} & \cdots & y_{2,\Delta j} \end{pmatrix}$$

The GML decision is given by the least (minimum) Partial Euclidian Distance for j=P – sequence length.

The idea to solve the GML formulation is to represent the exhaustive search as a tree-search, with estimation of PED corresponding to each partial received sequence which cannot be bigger than a GML (maximum likelihood) distance of the vector **X** (Lemma in source article is provided):

$$D_{\Delta 1} < D_{\Delta 2} < \cdots < D_{\Delta j} < D_{\Delta P} \leq D_{GML} \leq D$$

We set as the initial GML distance as D, and then update it, if some GML candidates were found. Also we perform pruning of the tree, stating that partial received sequence cannot be bigger than a GML distance as it was already said previously. If no candidate were found, we then increase D. The choice of the initial distance D is suggested in [1] to be:

$$D = \lambda^2 N \sigma_n^2$$

Where λ - scaling factor, N – degrees of freedom of Additive White Gaussian Noise (AWGN).[2]

The algorithm of GML search is adapted from algorithm described in [6]. The GML algorithm works with big constellation **Ψ**, corresponding to possible vectors of all pair combination for two polarizations of alphabet constellation symbols. Example for PDM-QPSK, where 1,2,3,4 – is a alphabet symbols denoted for QPSK constellation:

$$\Psi = \begin{bmatrix} 1\,1\,1\,1\,2\,2\,2\,2\,3\,3\,3\,3\,4\,4\,4\,4 \\ 1\,2\,3\,4\,1\,2\,3\,4\,1\,2\,3\,4\,1\,2\,3\,4 \end{bmatrix}$$

One uses a position search vector **v** of size 1xP to denote the point (vector) from big constellation Ψ in the estimated sequence:

$$v = [\text{some position of col. in } \Psi \quad \ldots \quad \text{some position of col. in } \Psi \quad \ldots \quad \text{some position of col. in } \Psi]$$

The GML algorithm in pseudo-code and inflow-chart could be presented as:

0. Set initial Euclidian distance to compare with as $D = D_0 = \lambda^2 N \sigma_n^2$. Generate the big constellation Ψ. Create index vector v of size 1xP.

[2] We will use in simulations next parameters: $\lambda^2 = 1.6$, $N = 2$ if it is not mentioned.

1. Set layer $j = 1$, then $v(j) = 1$and $\boldsymbol{X}_{\boldsymbol{\Delta j}} = \boldsymbol{\Psi}(\boldsymbol{v}(\boldsymbol{j}))$. Go to step #2.
2. Compute the partial metric based on Equation 2. For $\boldsymbol{X}_{\boldsymbol{\Delta j}}$:
   - If $D_{\Delta j} > D$ then go to step #3 to search over other branches;
   - Else go to step #4.
3. Find the largest k such $1 \le k \le j$ and $v(k) \le |\Omega|^M$ – cardinality of Ψ constellation.
4. Condition on j value:
   - If j=P, store the current final solution and update D with the new Euclidian distance and go to step #3.
   - Else set $j = j + 1$, $v(j) = 1$ and $\boldsymbol{X}_{\boldsymbol{j}} = \boldsymbol{\Psi}(\boldsymbol{v}(\boldsymbol{j}))$, then add this column to existing sequence $X_{\Delta j-1}$ to form $X_{\Delta j}$, then go to step #2.
5. Set $v(j) = v(j) + 1$ and $\boldsymbol{X}_{\boldsymbol{j}} = \boldsymbol{\Psi}(\boldsymbol{v}(j))$, then add this column to existing sequence $\boldsymbol{X}_{\boldsymbol{\Delta j-1}}$ to form $\boldsymbol{X}_{\boldsymbol{\Delta j}}$, then go to step #2.
6. If a final solution is found in step#4, output the solution, Else increase D to start again and go back to step #1.

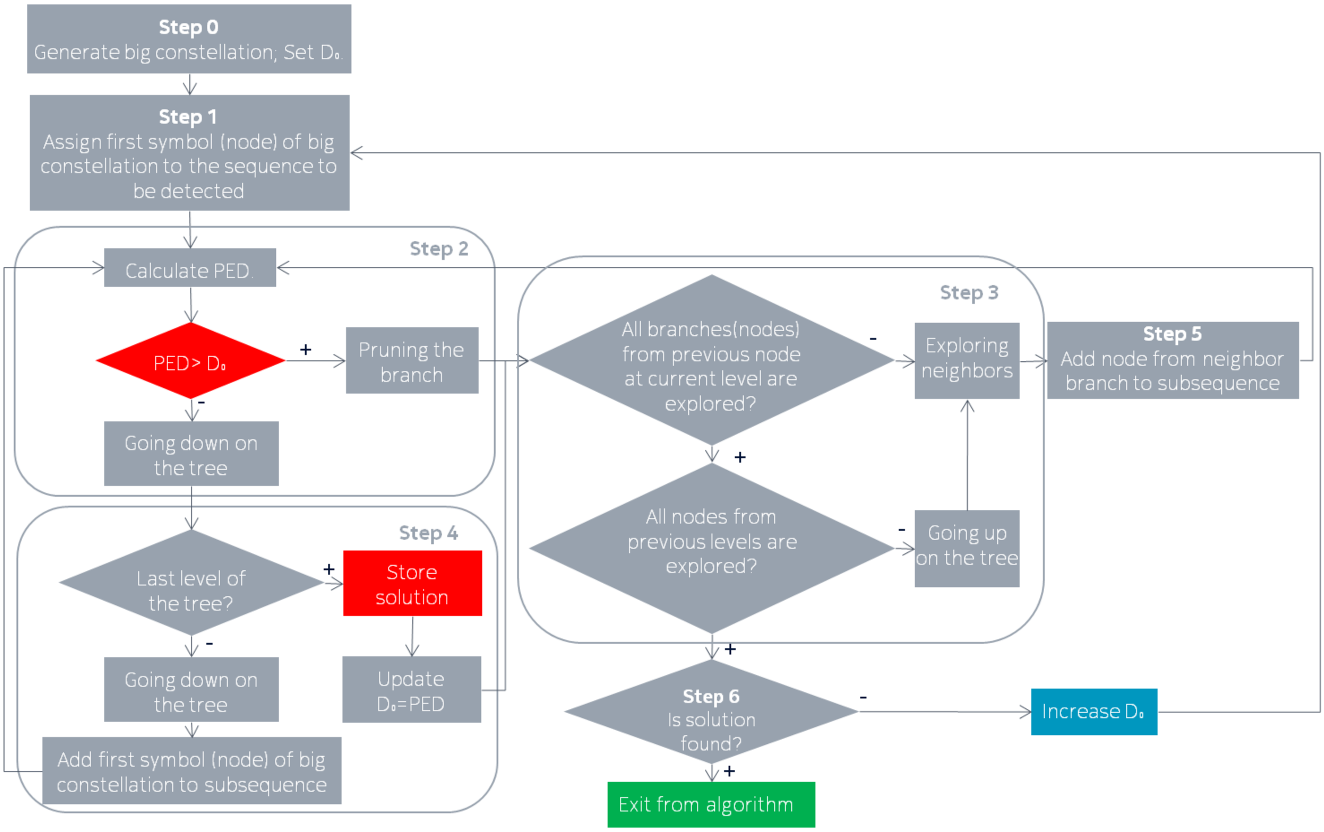


Figure 7 GML XPolM Compensation Algorithm – Break Down[3]

Presented algorithm above for the BPSK on single polarization for P=4 could be illustrated with next figure:

[3]Levels of tree = Number of columns in sequence
Several level of tree = Subsequence = Several columns
Symbols from big constellation = columns in detected sequence=nodes
Red steps – subject to change in Repeated Tree Search Strategy and Single Tree Search Strategy GML.
Blue step – subject to change in RTS (Repeated Tree Search) Strategy
Green step – subject to change in STS (Single Tree Search) Strategy

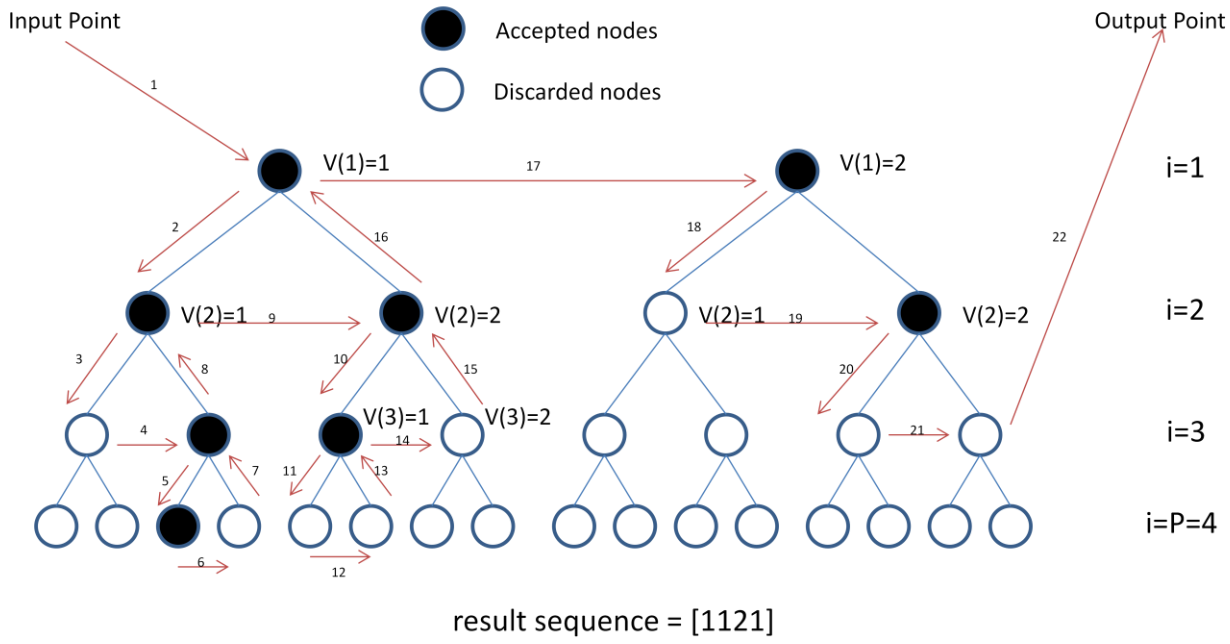


Figure 8 GML for XPolM – BPSK for Single Polarization for P=4 – Representation of algorithm as a tree search.

The presented hard output algorithm is not compatible with FEC. Therefore in the following, modifications of this algorithm will be done to derive a soft-output version of the fast tree-search of the GML formulation. The elaboration of that theme is done in chapter 5.

4. On the Capacity of XPolM Channel

In order to estimate the performance of the decoder with implemented FEC one needs to know the limits of possible BER waterfall (the signal to noise ratio (SNR) where the BER tends to zero) to be achieved. To find that limit the second Shannon theorem (channel coding theorem) comes to the help. It states that there are codes that allow transmitting information with rate less that the information capacity of channel with the probability of error tending to zero. The information capacity of the channel is defined as [7]:

$$C = \max_{p(x)} I(X;Y)$$

Where X and Y are random variables that represent symbols sent and signal received p (x) – probability distribution of the random variable X[4].

Thus, to find the limits of BER achievable one needs to get the capacity of XPolM channel, choose a rate of the code, and see the SNR that corresponds to that rate. The theoretically limited BER waterfall will be achieved at such SNR. The distance from such SNR will enable us to evaluate the power of the code and the power of the estimation of the channel.

Generally, information capacity of channel is achieved when p(x) is Gaussian, but we consider the uniform distribution p(x) corresponding to QPSK and 16 QAM constellations. Thus, we need to know the mutual information of the XPolM channel in case of these particular constellations. However, on the contrary to the Gaussian distribution probability p(x) case, there is no closed form expression for the mutual information in case of uniform distribution p(x) – we need to use specific calculus methods of which we speak later.

This chapter is devoted to the results of finding the capacity of the channel under XPolM.

4.1.Different cases of Mutual Information

I(X;Y) – memoryless unknown channel hypothesis.

We start from the hypothesis that the channel has no memory, or that is to say – does not have autocorrelation length and each state of the channel is independent from the previous one. At the output of the channel we do not have the information about its state – only the received signal Y is known. In this case the mutual information will be the lower bound of the achievable capacity and will correspond to a case when we do not use special properties of the channel to detect the symbols, such as:

- Slowly varying (approximation that during 4 symbols sent it has the same state) properties.
- Autocorrelation length around 14 symbols.

As we mentioned earlier, capacity is the maximized mutual information between X and Y. In case of defined constellation we should calculate the mutual information for specific exact p(x), using next expression [7]:

[4] Small x as well as y represent a particular realization of random variable in form of column vector.

$$I(X;Y) = E_{x,y} \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Where X and Y are random variables, x and y – specific values from X and Y alphabets, $p(x) = \frac{1}{4}\ or \frac{1}{16}\ or \frac{1}{256}$ in case of BPSK, QPSK and 16 QAM modulation respectively on two polarizations for all x, p(y) – probability distribution for random variable Y, and p(x,y) – joint probability distribution of two random variables X and Y.

The problem to calculate such mutual information comes from the absence of knowledge of p(y), p(x,y) and from the fact that Y is a continuous random variable. To overcome the problem of continuity we introduce discrete approximation of Y. To overcome the problem of absence of knowledge of p(y) and p(x,y) we start building statistical model of distribution of Y, X and Y jointly – we simulate a big number of transmissions and catch the distribution of p(y) and p(x,y) during it. We use "binning" technique to discretisize Y and get the statistical properties of each "bin".

I(X;Y|H) – known channel hypothesis.

The case of known channel hypothesis corresponds to the knowledge of exact state of the channel at its output with detected signal Y. We can consider this case as an upper bound of the achievable capacity. This case could be easily calculated and with more precision than the case with unknown channel. That case gives the general idea where the BER waterfall should occur.

Several models of definition of capacity for different type channels with known state are presented in [8]. They differ with allocation of power among the polarizations. Thanks to uniform allocation of power used in case XPolM – due to the absence of different gain on two polarizations in average in time and in polarization – we use model that is defined in [9] and mentioned in [8].

The mutual information for known channel is described in [9] [8] , and could be seen as conditional mutual information [9] over three random variables X,Y and H which represent symbols sent, signal received, and channel state:

$$I(X;Y|H) = E_{x,y,h} \log\left(\frac{p(h)p(x,y,h)}{p(x,h)p(y,h)}\right)$$

$$I(X;Y|H) = \sum_{x\in X}\sum_{y\in Y}\sum_{h\in H} p(x,y,h) \log\left(\frac{p(h)p(x,y,h)}{p(x,h)p(y,h)}\right)$$

Where p(x), p(h)[5] and p(y) – probability distributions of random variables X,H,Y and p(x,y,h) – joint probability distribution. Here, as in previous case, the problems with continuity of y and h arise.

However, the closed form expression of joint probability density function of random variables X, Y and H exists. That helps us to define probability p(x,y,h) for some "bin" of infinitely small size $dx \times dy \times dh$ for certain values of x, y and h [5] for noise of power N_0:

[5] Small h represent particular realization of channel matrix and is defined in form of matrix 2x2.

$$f(x,y,h)dxdydh = \exp\left\{-\frac{1}{N_0}\|y - hx\|^2\right\}dxdydh \sim p(y|x,h)$$

$$p(x,y,h) = p(y|x,h)p(x)p(h)$$

$$p(x,y,h) \sim f(x,y,h)p(x)p(h)$$

In order to overcome the problem of dealing with continuous random variables we can use Monte-Carlo simulations (produce M realizations of data transmission) and Big Number Law to estimate expected value of mutual information:

$$I(X;Y|H) = E_{x,y,h} \log\left(\frac{p(h)p(x,y,h)}{p(x,h)p(y,h)}\right) = \frac{1}{M} \sum_{M\ realisations} \log\left(\frac{p(h)p(x,y,h)}{p(x,h)p(y,h)}\right)$$

The random variables X and H are independent, thus $p(x,h) = p(x)p(h)$. Due to uniformity of probability distribution of random variable and its discrete nature we have:

$$p(x) = \frac{1}{16}\ or\ \frac{1}{256}$$

in case of QPSK and 16 QAM modulation on two polarizations for any x. After rearrangements we get:

$$I(X;Y|H) = \frac{1}{M} \sum_{Number\ of\ realisation} \log\left(\frac{f(x,y,h)}{p(x)\sum_{x=X} f(x,y,h)}\right)$$

That key formula let us estimate the conditional mutual information for XPolM channel.

I(X;Y|$H_{estimated}$) – unknown channel with memory approximation.

The mutual information for the case of unknown channel with no memory is relatively easy to calculate, however when it comes to calculate the mutual information for the channel with memory it occurs that there are hard problems to resolve – because we have to treat the X and Y random variables not only for one channel use, but for several time uses, in general for infinity time uses. In [10] the capacity of channel with its unknown state but with presence of memory was defined:

$$C = \lim_{n\to\infty} \sup_{p(X^n)} \frac{1}{n} I(X^n;Y^n)$$

This result directly corresponds to the Mutual information of the channel with X – random variable of infinite length (previously we encountered column vector 2x1, but here we should view a vector 2xn where n tends to infinity).

The channel perturbed by XPolM is considered to be as a channel with memory due to presence of autocorrelation of current symbol with previous ones. Thus, in fact we are mostly interested in mutual information of the case of unknown channel with memory.

Being said that it is difficult to calculate we can make an approximation – sought mutual information could be modeled as conditional mutual information with partial knowledge of the channel, where the channel state is given by GML XPolM decoder.

$$\frac{1}{n} I(X^n; Y^n) \approx I(X; Y|H_{estimated})$$

That approximation is justified because it uses some properties of channel in time – we consider that the channel does not change during P symbols.

This approximation will let to get the best evaluation of the SNR waterfall possible in reality – when we do not know the exact state of the channel.

Capacity of XPolM channel

In previous chapters we considered mutual information in presense of uniform probability distribution of $p(x)$. Now we propose to find out, what limits of mutual information, called capacity could be achieved in presence of Gaussian $p(x)$. This capacity will show the theoretical limits of the channel, which could be never surpassed.

The sought capacity of the XPolM channel is defined as a capacity of the channel in case of known channel for Gaussian random variable X [9]:

$$C = \max_{p(x)} I(X; (Y, H)) = I(X; Y|H)_{P(x)-\, gaussian} = \log_2\left(det\left(\boldsymbol{I} + SNR \times \boldsymbol{H}^{\dagger}\boldsymbol{H}\right)\right)$$

where **I** is identity matrix of size 2x2. Before exploring the relation $\boldsymbol{H}^{\dagger}\boldsymbol{H}$ we must note one particular property of the possible channel model: $\boldsymbol{\alpha} = -\boldsymbol{\beta}^{*}$. That hypothesis was developed in [11]. Thus:

$$\boldsymbol{H}^{\dagger}\boldsymbol{H} = \begin{bmatrix} \sqrt{1-\alpha^2} & \alpha^* \\ -\alpha & \sqrt{1-\alpha^2} \end{bmatrix} \begin{bmatrix} \sqrt{1-\alpha^2} & -\alpha^* \\ \alpha & \sqrt{1-\alpha^2} \end{bmatrix} = \boldsymbol{I}$$

The matrix **H** is unitary, thus we can get next expression for capacity:

$$C = 2\log_2(1 + SNR)$$

This simple relation let us describe the theoretical limit that could be called XPolM channel capacity

4.2.Results and discussion

In this chapter we present the results of calculation of different kind of mutual information for 2x2 channels under the effect of XPolM and its Capacity. We point out those next inequalities that take place:

$$C \geq I(X; Y|H) \geq \lim_{n\to\infty} \sup_{\mathrm{p}(X^n)} \frac{1}{n} I(X^n; Y^n) \geq I(X; Y|H_{est}) \geq I(X; Y)$$

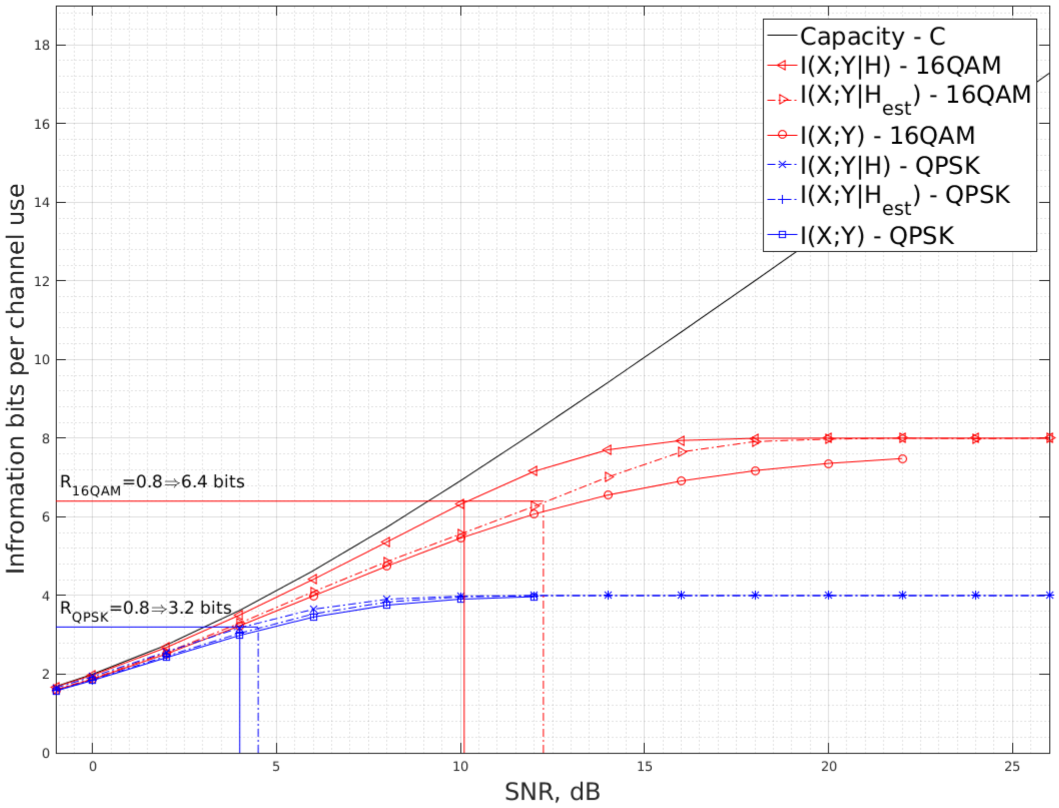


Figure 9 Mutual information of the channel under XPolM for different cases.

We can conclude on next interesting points:

- The XPolM effect is more evident and show itself in the case of modulation formats of higher orders – for example in 16 QAM. That is seen from the fact that the curves for QPSK modulation for different cases are almost superimposed but for 16 QAM each case is distinct.
- We see that for 16 QAM case the estimation of the channel starts to work at SNR for polarization or SNR for symbol=11 dB. That fact is seen from the point of separation of the curves $I(X;Y|H_{est})$ and I(X;Y) binning. After that point with GML XPolM compensation scheme we start to use memory properties of the XPolM channel. (We suppose that curves $I(X;Y|H_{est})$ and I(X;Y) binning are not superimposed before due to only the discretization of Y variable for binning).
- For the usual code rate used in optical networks R=0.8(which is 6.4 bits/p.c.u for 16 QAM and 3.2 bits/p.c.u for QPSK) we can find next theoretical values achievable by FEC code block:
  - $SNR_{waterfall}(QPSK_{known\ channel})$=4 dB
  - $SNR_{waterfall}(QPSK_{estimated\ channel})$=4.5 dB
  - $SNR_{waterfall}(16\ QAM_{known\ channel})$=10.1 dB
  - $SNR_{waterfall}(16\ QAM_{estimated\ channel})$=12.25 dB
- For the code rate R=0,5(which is 4 bits/p.c.u for 16 QAM and 2 bits/p.c.u for QPSK) we can find next theoretical values achievable by FEC code block:

- $SNR_{waterfall}(QPSK_{known\ channel})$=0.2 dB
- $SNR_{waterfall}(QPSK_{estimated\ channel})$=0.4 dB
- $SNR_{waterfall}(16\ QAM_{known\ channel})$=5.05 dB
- $SNR_{waterfall}(16\ QAM_{estimated\ channel})$=5.7 dB

- The found SNRs let us conclude that for 16 QAM there are exist a need to introduce a new technique of channel estimation, due to 2.1 dB of potential gain for R=0.8 and 0.65 dB of potential gain for R=0.5, but for QPSK there are only 0.5 dB gain for R=0.8 and 0.2 dB for R=0.5 and GML XPolM technique already fulfills the need.

5. Novel Soft-Output detectors based on GML for XPolM

There is a need to introduce a decoding block that will be adapted to the FEC block. The said adaptation consists in delivering to the output of the decoder and input of FEC block the sequence of log likelihood ratios (LLRs) – soft representation of hard type bit sequence.

LLR should be defined for each bit in decoded sequence of symbols. We will use max-log approximation ($\log \sum_k \exp(a_k) \approx \max_k a_k$) in order to calculate LLR and define its metric based on next expression [12]:

$$L(x_{i,j,k}) = \min_{s \in x_{i,j,k}^{(0)}} \|\boldsymbol{y} - \boldsymbol{Hs}\|^2 - \min_{s \in x_{i,j,k}^{(1)}} \|\boldsymbol{y} - \boldsymbol{Hs}\|^2$$

Where: $x_{i,j,k}^{(0)}$ and $x_{i,j,k}^{(1)}$ – are the sets of symbol vectors that have the k-th bit in the symbol of the i,j-th scalar symbol equal to 0 and 1, respectively. Index “i” – in our case will define the polarization position of the symbol and “j” – the column in the block of length of P.

We can rearrange this expression to be more suitable with the help of introduction of notion of Generalized Maximum Likelihood Counterhypothesis - the least distant symbol in constellation from received signal that contains opposite bit to the bit that in GML decision:

$$\lambda_{i,j,k}^{\overline{ML}} = \min_{s \in \overline{x_{i,j,k}}} D_{\Delta P}$$

Where $x_{i,j,k}$ - a bit in GML detected sequence, and $\overline{x_{i,j,k}}$ - the opposite to such bit. $D_{\Delta P}$ - is defined by Equation 2 as a function of received signal $\boldsymbol{y}$ and counterhypothesis $\boldsymbol{s}$.

We note in such case Euclidian distance of GML detected sequence:

$$\lambda^{ML} = D_{ML}$$

In other way LLR could be described as:

$$L(x_{i,j,k}) = \begin{cases} \lambda^{ML} - \lambda_{i,j,k}^{\overline{ML}}, x_{i,j,k}^{ML} = 0 \\ \lambda_{i,j,k}^{\overline{ML}} - \lambda^{ML}, x_{i,j,k}^{ML} = 1 \end{cases}$$

Where λ^{ML} - Euclidian distance for GML solution, $\lambda_{i,j,k}^{\overline{ML}}$ – Minimum Euclidian distance of the counterhypothesis to i,j,k bit of GML decision.

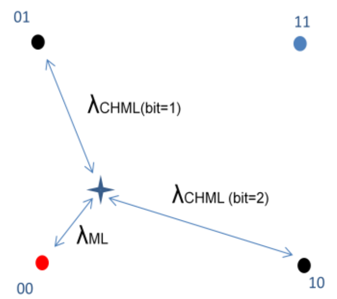


Figure 10 Example of LLR calculus for sequence of length P=1, on one polarization and coded on QPSK constellation: For 00-th received symbol: 1-st bit: LLR=λ_{ML} -λ_{CHML}(bit=1) 2-nd bit: LLR=λ_{ML} -λ_{CHML}(bit=2).

We note that in such metric LLRs are normalized by factor $-\frac{1}{N_0}$, where N_0 - noise power on each polarization in channel. In order to use the LLRs later in FEC block one have to multiply such normalized LLR with that factor.

Thus, implementing the soft-output GML XPolM decoder mean to use the algorithm not only to find the GML decision with least Euclidian distance (metric), but also find all the counterhypothesis' least Euclidian distances corresponding to GML decision bits.

After the deliverance of the LLRs the bit sequence will correct the errors with the help of LDPC decoder. We remind that the sequence was coded at the beginning of scheme with FEC coder. The principle is described in Chapter 2. We recognize that there exist other codes, more powerful in terms of BER and that let us to reach the near Shannon limit. In the simulation we will use iterative LDPC decoder with variable parameters of the number of iterations. Currently in long-haul optical networks number of iterations are considered less than 10 [13]. However, in order to fully use the potential of the coding, we proceed by 50 iterations, if it doesn't said contrary. Also we use clipping of absolute LLR values at the input of FEC block at the level of 20, if it nothing other précised.

5.1.Optimal Decoders

In this chapter we consider the "optimal" decoders algorithms – those that deliver LLRs limited only by max-log approximation.

Repeated Tree Search Algorithm Approach

The most simple in logic and in implementation algorithm is "Repeated Tree Search"(RTS) - we use GML XPolM in for-cycle to find GML counterhypothesis to each (i,j,k) bit of GML detected sequence at the beginning of RTS

We modify GML algorithm by:

- Adding to pruning criteria at step 2 of original XPolM algorithm additional condition – prune those branches that contains bits identical to bits of GML decision, because we are searching for counterhypotheses – those for sure will have opposite bit to detected one, or counterbits.
- Introduction LLR calculation in step 4.

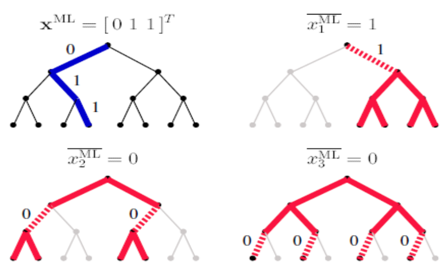


Figure 11 Illustration of additional pruning criteria [12]

The GML XPolM algorithm is modified and put in for-loop, as illustrated below:

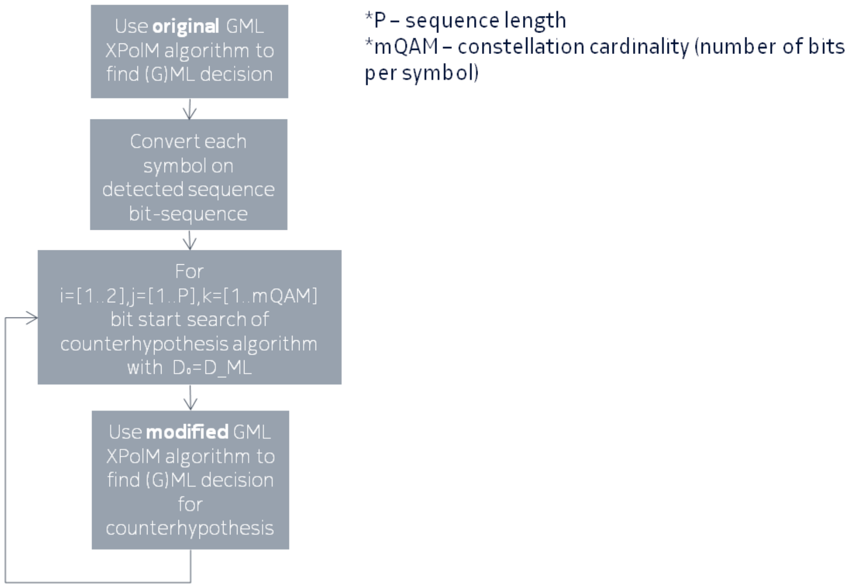


Figure 12General flow chart of RTS GML algorithm

The modified steps in original GML XPolM algorithm to new RTS GML XPolM represented in the figure below:

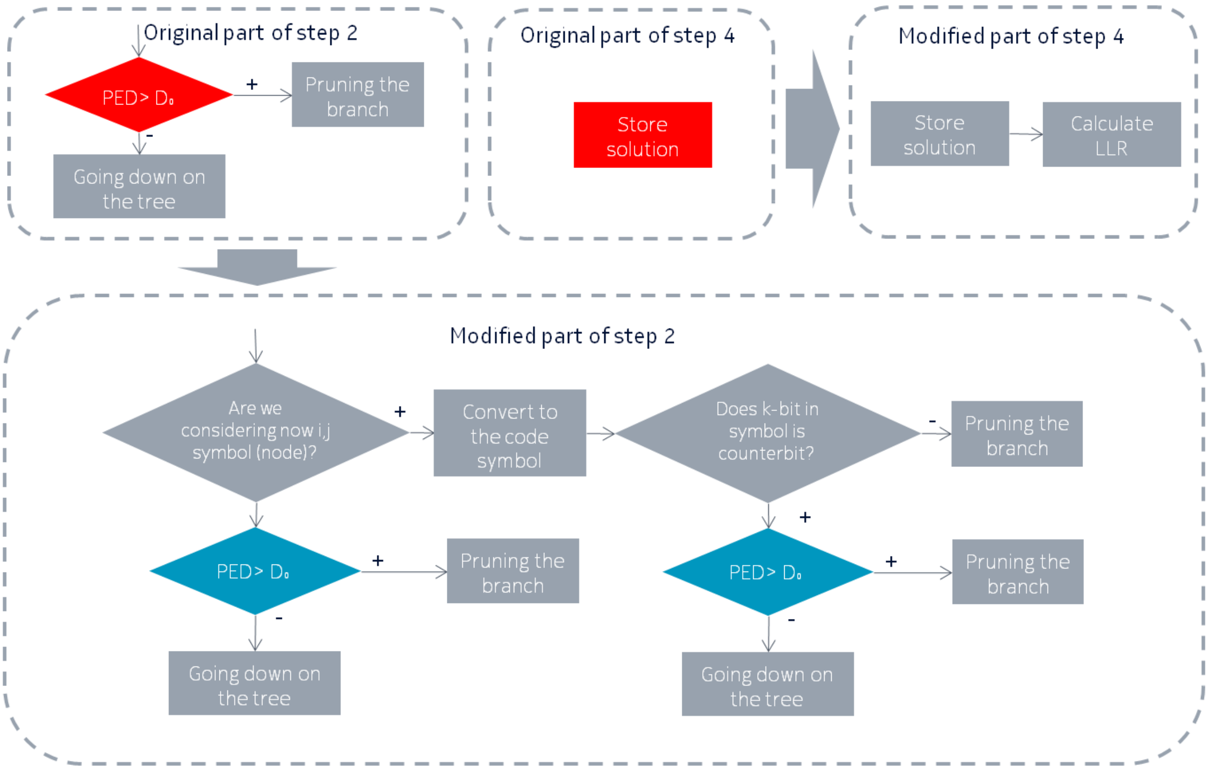


Figure 13 Modification of Steps in original GML algorithm (Blue steps – Steps of later LLR Clipping introduction see "Log-likelihood clipping technique in application to Repeated and Single Tree Search Algorithms")

For further notice – we will measure the performance of our algorithm using two metrics – Bit Error Rate – BER and number of visited nodes per channel realization, or per column transmitted – number of calculation of Euclidian distance per channel use (p.c.u).

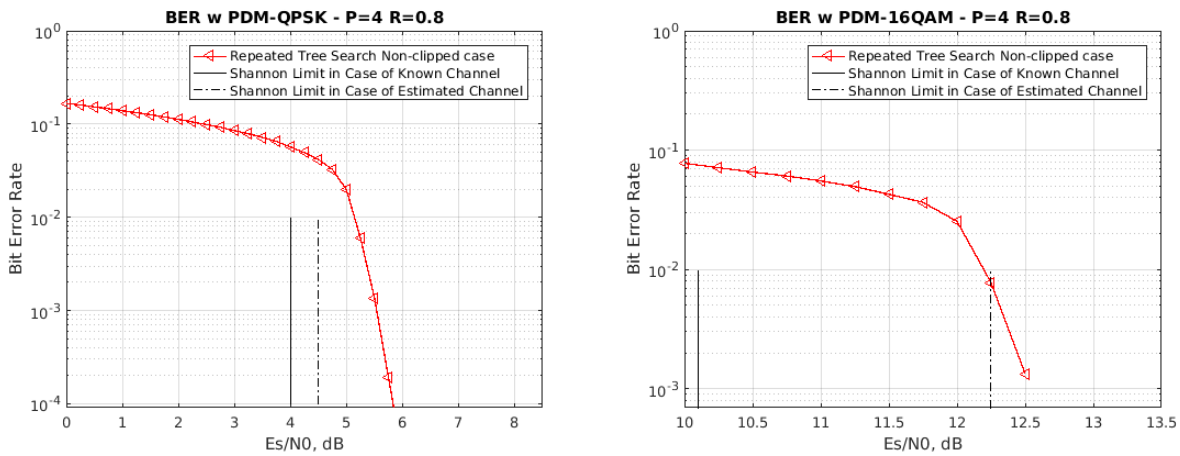


Figure 14 BER for RTS algorithm for QPSK case(left) and for 16 QAM case(right)

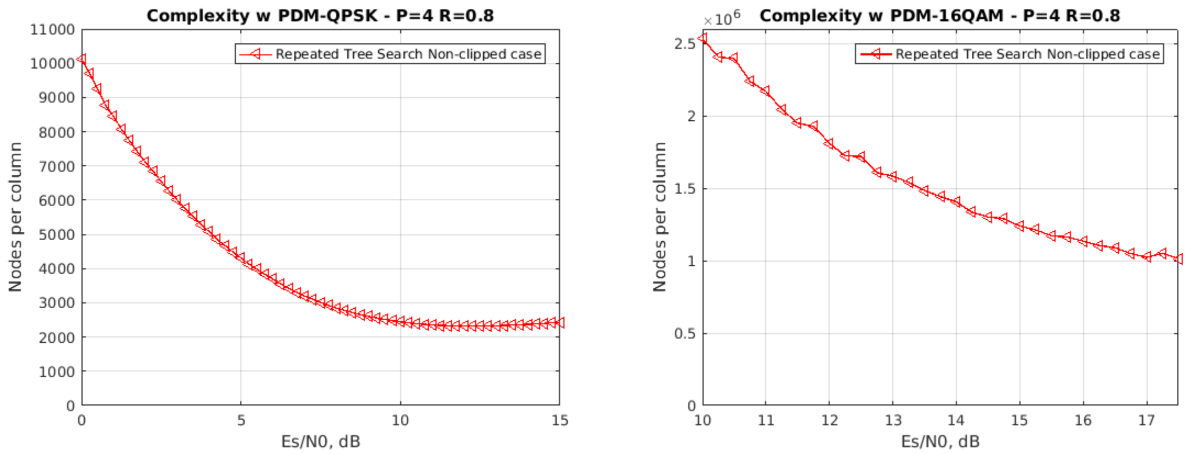


Figure 15 Visited nodes for RTS algorithm for QPSK case (left) and for 16 QAM case(right)

Single Tree Search Algorithm Approach

Here we propose a single tree search algorithm, based on a technique represented in [12] – the single tree search (STS) performs search for counterhypotheses and for GML decision together at the same time.

Below we present next key points of the GML XPolM STS algorithm:

- At the beginning we create a 3 dimensional array for counterhypothesis distances for each bit in P length block sequence.
- We initialize the algorithm with the GML distance and counterhypothesis distances with value *infinity*.
- As in RTS algorithm we modify only the steps 2 and 4 of GML algorithm.
- The objective point of algorithm could be described in next two principle points:
    - When a leaf(sequence) of tree is reached, we distinguish two cases:
        - If new GML hypothesis is found, i.e., D(current sequence under consideration)<D(current GML hypothesis), then

- If bits in new GML sequence were flipped, then in the table of counterhypothesis we assign to counterhypothesis distance of the respected position of flipped bit the former GML hypothesis distance.
- If bits were not flipped, then the counterhypothesis in table is not changed.

- If no update of GML hypothesis is required, then we update only those counterhypotheses distances, which correspond to different bits (counterbits) of current leaf and current GML hypothesis.

- The pruning criteria organized as follow:
  - The branch is cut If PED is bigger than altogether
    - The counterhypotheses distances that corresponds to the counterbits of already explored part of subsequence and GML hypotheses.
    - The counterhypotheses distances that correspond to unexplored bits of subsequence
  - Otherwise branch is not pruned.

- When all the counterhypothesis distances and GML hypothesis distance are calculated we calculate LLR output.

Illustration of pruning process is provided in scheme below:

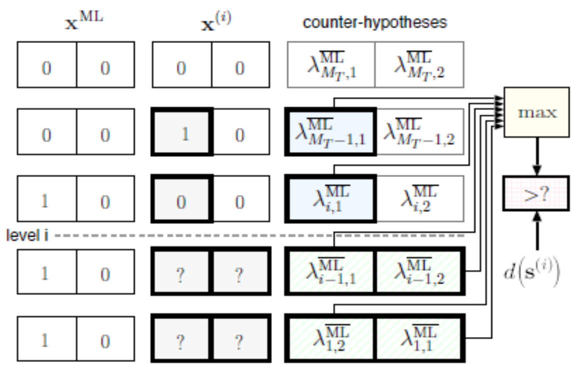


Figure 16 Example of pruning process [12]

Where X^{ML} – current GML decision hypothesis, $X^{(i)}$ – partial sequence under consideration. Explicative note – we put in comparison list of the distances for explored current counterbits (outlined with black) as well as we put in comparison list of the distances of unexplored counterhypotheses distances. If current PED is bigger than all of the distances considered, then the branch is pruned.

We present modification of steps of GML XPolM algorithm in chart-flow:

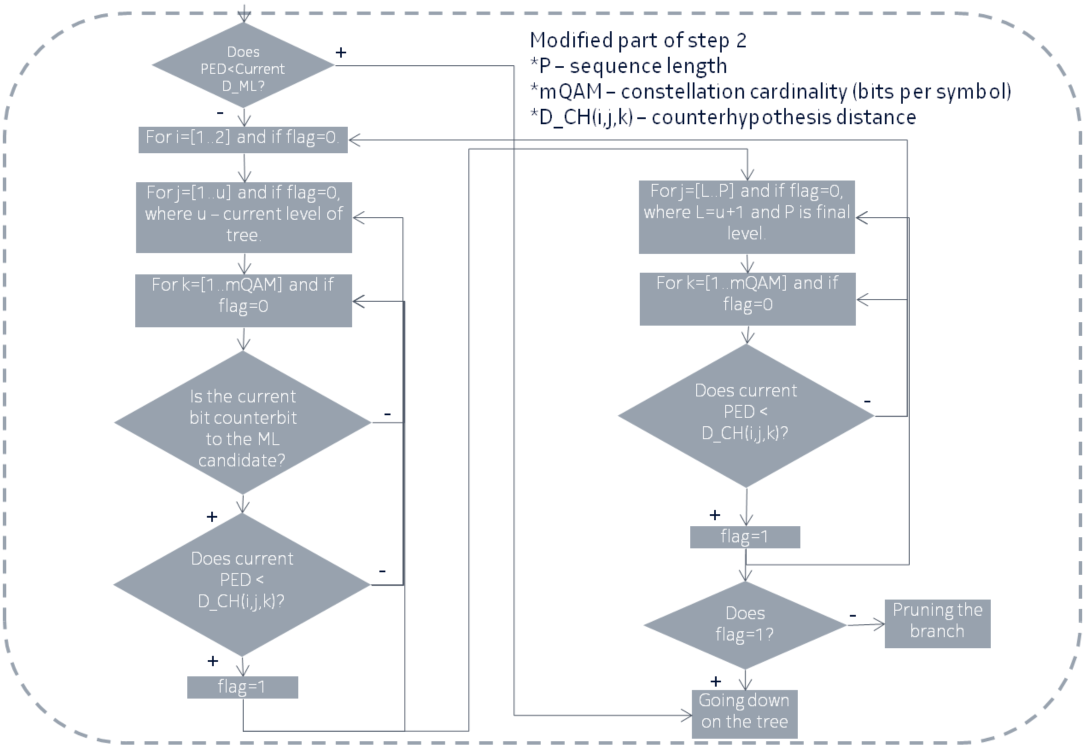


Figure 17 Modification of Steps in original GML algorithm – Step 2

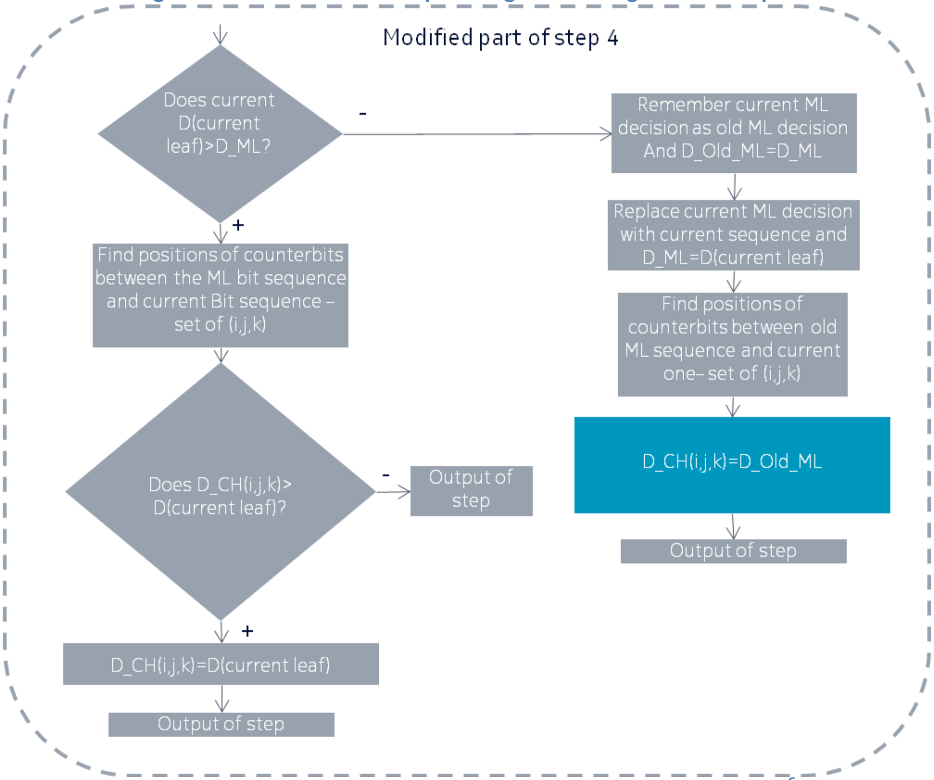


Figure 18 Modification of Steps in original GML algorithm – Step 4[6]

[6] *Blue steps – Steps of later LLR Clipping introduction

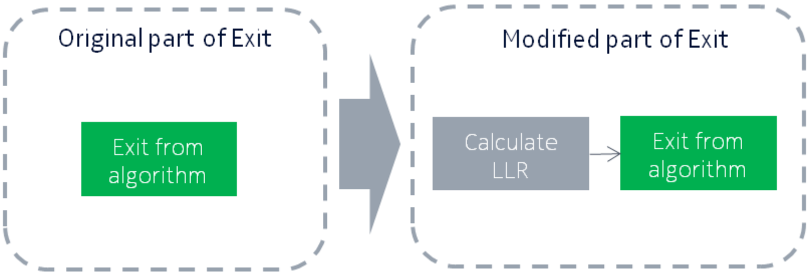


Figure 19 Modification of exit step in original GML algorithm for STS strategy

We present below the results of simulation of Single Tree Search strategy (algorithm):

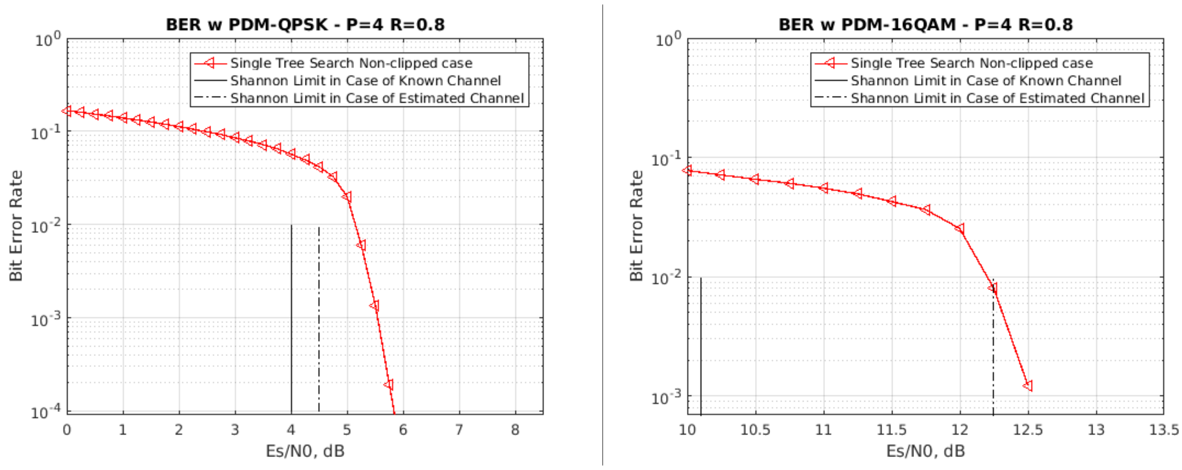


Figure 20 BER for STS algorithm for QPSK case(left) and for 16 QAM case(right)

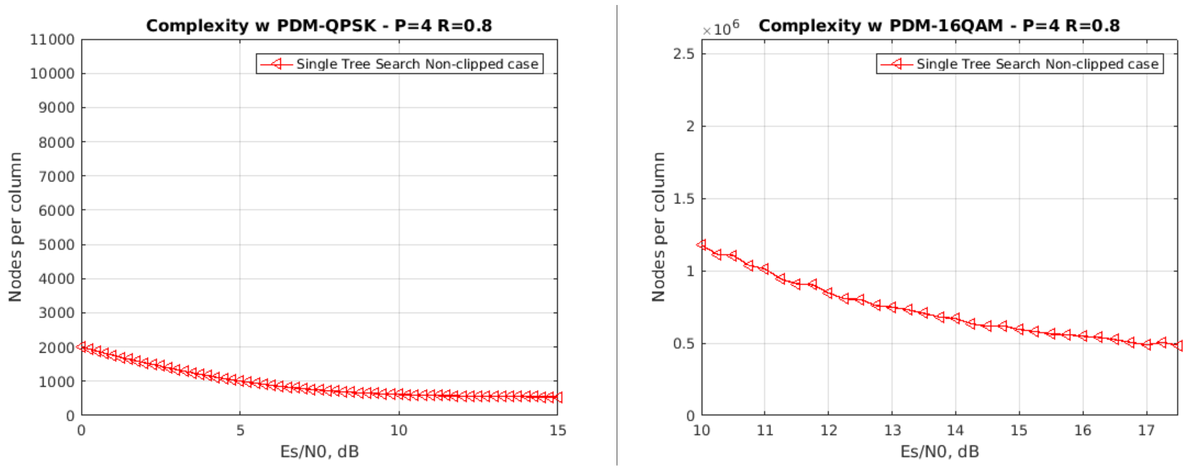


Figure 21 Visited nodes for STS algorithm for QPSK case (left) and for 16 QAM case(right)

We see that the results of STS and RTS GML XPolM in terms of BER are the *same*, and there is no surprise, they should be the same, because they search for same metrics distances, but only with different approach. However we may note that the number of nodes visited differ significantly:

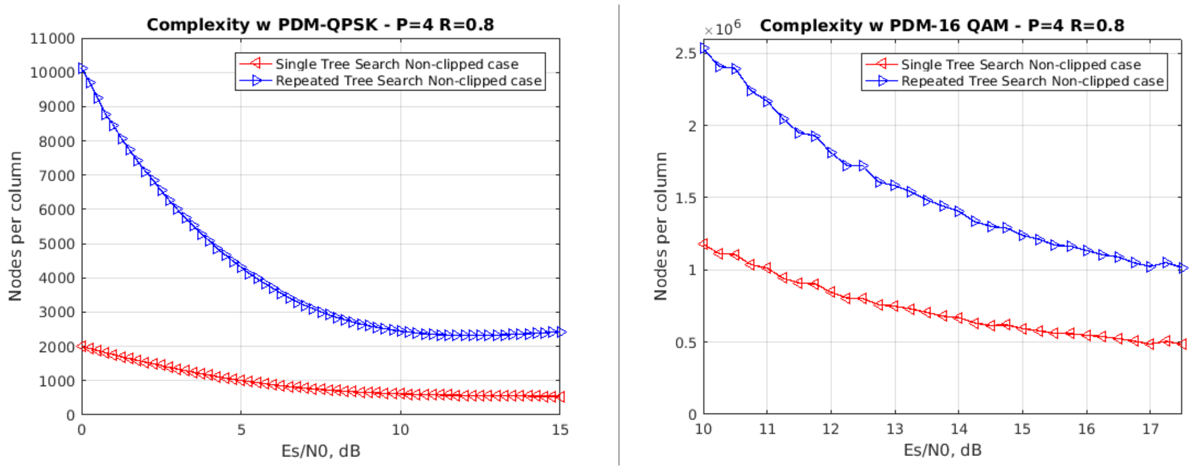


Figure 22 Comparison of visited nodes p.c.u of STS and RTS strategy for QPSK (left) and 16 QAM(right)

We see that for QPSK constellation Single Tree Search is 5 times more efficient than Repeated Tree Search, and for 16 QAM case that ratio decrease to the factor 2.

5.2.Genie aided, artificial LLR estimation, Perfect channel estimation

In addition to SNR waterfall that give possible SNR limit for a FEC code, we can also estimate upper bound of GML equalization at given code. If we need to exclude the influence of the particular LDPC code, we have to recognize next two cases:

- Genie aided decoder – the decoder that will know the exact state of the channel at any instant and perform the detection with exact knowledge of channel. Thanks to knowledge of state of the channel at any instant we can perform decoding column-by-column, which is much faster than decoding block-by-block of length P. The Genie aided decoder based on single tree search algorithm on the hypothesis that it is more performing than repeated tree search. One may refer to that type decoder as "**known H**".
- Perfect Channel Estimation (PCE or "data-aided") – the decoder that will know the exact information that was sent and on the basis of the information sent will execute the best possible channel estimation (thanks to Equation 1) and then apply it to decode the sequence generated with same channel state but with different noise added. One may refer to that type decoder as "**known X**".

Thus, genie aided case help us to exclude the influence of the code and compare different Soft Output decoders and PCE case will show us the limits of estimation model, also excluding the influence of code.

There is another special case to consider – generation of artificial LLRs on the basis of hard output detector GML XPolM. That case will let find out if there was a gain apported with introduction of the soft-output GML XPolM. We realize such technique according to the next scheme:

- Get the detected symbols by XPolM GML and convert them to bit sequence.

- Generate the artificial LLRs corresponding to a bit sequence according to the next scheme :

$$L(x_{i,j,k}) = \begin{cases} 20\,, x_{i,j,k}^{ML} = 0 \\ -20, x_{i,j,k}^{ML} = 1 \end{cases}$$

- Send then such sequence to the FEC decoder to correct the errors.

We add these two new limits to the STS GML case in order to be able to estimate how well are done the detection and XPolM mitigated. We also add the case of artificial LLR.

We present below the BER for QPSK and 16 QAM cases:

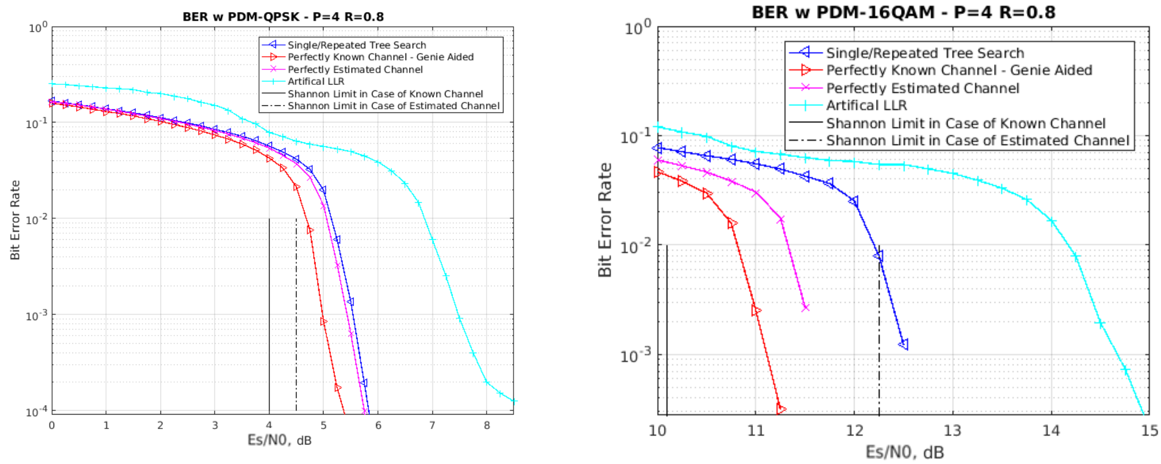


Figure 23 BER for special cases for QSPK (left) and for 16 QAM (right).

Based on the results of simulation we can conclude that:

- For QPSK case:
  - The interval between "Genie Aided" case and "Soft Output GML XPolM" compensation technique is about **0.5 dB**, which shows that we have almost compensated the effect of XPolM. This result could be directly related to the Shannon limits in case of known and estimated channel. We can cover distance with introducing of new techniques of channel estimation.
  - The interval between "Perfect Channel Estimation" and Soft Output GML XPolM" compensation technique is about **0.1 dB**, which shows that we are almost on the limit of performance of channel estimation.
  - The case of artificial LLR shows that we still can use original Hard Output GML XPolM compensation technique with FEC codes without introducing LLR calculations.
- For 16 QAM case:
  - The interval between "Genie Aided" case and "Soft Output GML XPolM" compensation technique is about **1.5 dB**, which is less than the interval between Shannon limits. That could be explained by the fact that the code performs better in "Soft Output GML XPolM" than in "Genie Aided" case and help to decrease that gap. Nevertheless such gap show us that the need of new approach to compensate the channel exists.

- The interval between "Perfect Channel Estimation" and Soft Output GML XPolM" compensation technique is about **0.85 dB**. That fact tells us that with existing model of channel estimation there is some gain to be explored. Such gain could be explored by turbo algorithms, which will estimate the channel several times based on corrected sequence.
- The case of artificial LLR shows that we still can use original Hard Output GML XPolM compensation technique with FEC codes without introducing LLR calculations, however it is reasonably far from achieved results.

The results shows us again that for QPSK the performance of decoder is nearly perfect, however for 16 QAM case the estimation of channel is not done perfectly, the XPolM effect is more perturbing than in QPSK case.

5.3.On the Log-likelihood distributions in case of XPolM

It is interesting to notice that the LLRs on the output of modified GML XPolM with random message have a particular distribution at relatively high SNR:

- Four symmetrical peaks for 16 QAM.
- Two symmetrical peaks for QPSK.

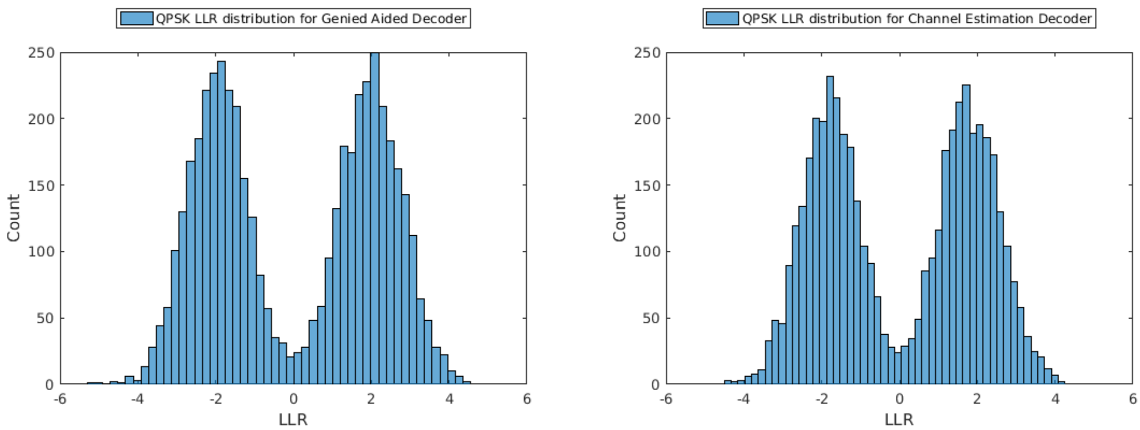


Figure 24 the LLR distribution in case of QPSK for Genie aided case (left) and GML XPolM STS detector (right)

We see that in case of QPSK modulation there is almost no difference between cases with XPolM effect compensated partially and XPolM effect compensated fully. We can conclude on the effective compensation of XPolM with modified STS or RTS GML XPolM algorithm is performed. That is in accordance with the conclusion of the chapter 3 – we use fully the capabilities of channel model estimation and we are able to mitigate the XPolM effect.

Range of the normalized LLR distribution: $[-6; +6]$

Peaks of the normalized LLR distribution: ± 2

For the 16 QAM modulations format the situation is a little bit different:

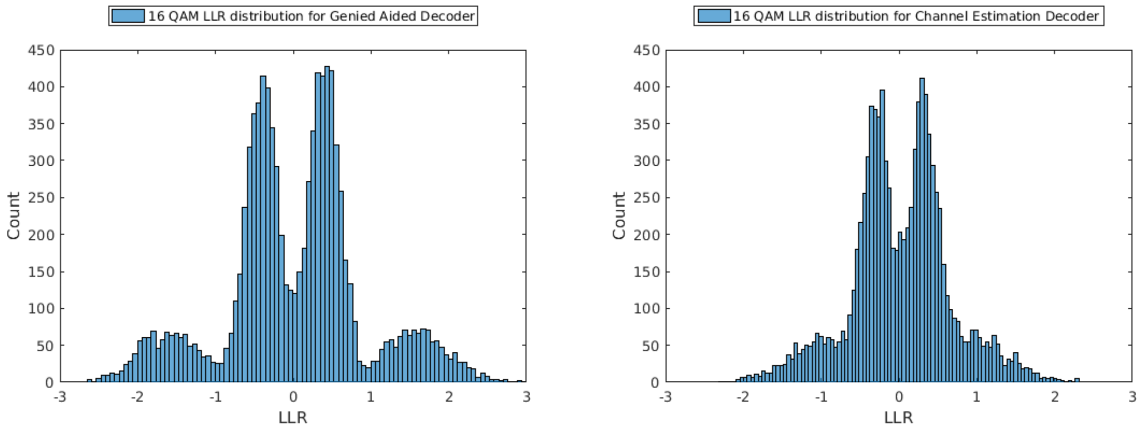


Figure 25 the LLR distribution in case of 16 QAM for Genie aided case (left) and GML XPolM STS detector (right)

We see here that the effect of XPolM is more evident in case of 16 QAM and we are not able fully compensate it – the sideway peaks are shifted and attenuated. That also corresponds to the fact that we established in the chapter “**Erreur ! Source du renvoi introuvable.**”.

Range of the normalized LLR distribution: $[-3; +3]$

Peaks of the normalized LLR distribution: $\pm 0.4;\ \pm 1.6$.

How that comportment could be explained? We remind that the LLR is calculated as difference between GML distance and Counterhypothesis Distance. Due to the fact that most of the points sent, arrive not far from symbol on constellation, the average LLR for these most point will be defined as:

$$< L(x_{i,j,k}) >= \begin{cases} < \lambda^{ML} > - < \lambda^{\overline{ML}}_{i,j,k} >, x^{ML}_{i,j,k} = 0 \\ < \lambda^{\overline{ML}}_{i,j,k} > - < \lambda^{ML} >, x^{ML}_{i,j,k} = 1 \end{cases}$$

$$< L >= \pm(0 - < D_{counter-hypothesis} >)$$

$$< L >= \pm < D_{counter-hypothesis} >$$

Thus peaks will be defined by nearest counterhypothesis distance. We will consider two cases – QPSK and 16 QAM, with average power on the polarization equal to 1.

For QPSK case the average counterhypothesis distance for any bit in constellation is:

$$< D_{counter-hypothesis} >= d^2_{min} = 2$$

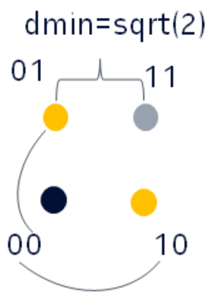


Figure 26 QPSK Constellation counterhypothesis average value demonstration for 00 Symbol case

For 16 QAM case the average counterhypotheses distances are:

$$< D_{counter-hypothesis} >= \begin{cases} d_{min}^2 = 0.4 \, for \, first \, and \, third \, bits \\ (2d_{min})^2 = 1.6 \, for \, second \, and \, forth \, bits \end{cases}$$

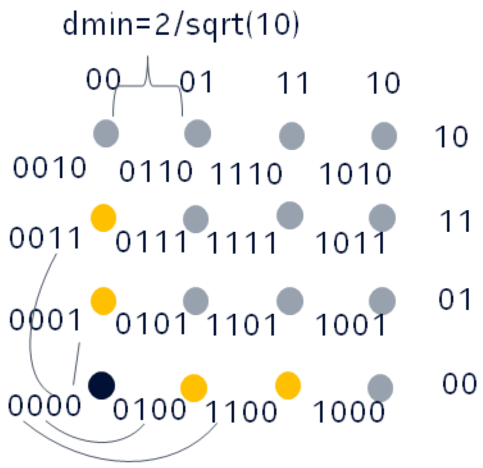


Figure 27 16 QAM Constellation counterhypothesis average value demonstration for 0000 Symbol case

In the 16 QAM case the power of two peaks is not the same – the cause of this is gray mapping – and for central points (0111, 0101, 1101, 1111) the average counterhypothesis distance for any bit is $< D_{counter-hypothesis} >= d_{min}^2 = 0.4$

Thanks to the shape of LLR distribution we can know if the algorithm works correctly or not.

5.4.Suboptimal decoders

With the introduction of the STS and RTS GML XPolM decoder we add a lot to the complexity – we must explore a lot of nodes in order to find the true counterhypothesis to the optimal decision. Even with the fact that we use "max-log" approximation of LLR calculus we consider such algorithms optimal in sense of the results delivered. However we can limit the complexity of algorithms with introduction of suboptimal methods.

We distinguish next cases of suboptimal decoders:

- Clipping of LLRs – introduce the limits of LLR values that could be achieved, solutions implemented described in [12].
- Two parts algorithm – perform calculation of LLR not "block by block of length P" but column by column. That will be possible to do if we use original GML XPolM algorithm to estimate the channel, and then use the state of estimated channel to calculate LLR.
- List Sphere decoder – during the search for optimal solution keep the list of solution candidates, and use these solutions to calculate LLR – with hypothesis that counterhypothesis solution will be in the list.
- List Sphere decoder with "bit flipping" at the end of algorithm – for those bits for which there were no counterhypothesis found use bit-flipping algorithm – flip bit in original sequence and calculate for it the Euclidian distance and keep that distance as counterhypothesis.

- Turbo decoder – introduce iterative algorithm that will let us to do estimation of channel, each iteration based on the effect of error correction.

Log-likelihood clipping technique in application to Repeated and Single Tree Search Algorithms

We can decrease the complexity of algorithm by introducing new condition of LLR clipping – the modulus of LLR value can be lower but not bigger than a certain limit:

$$|L(x_{i,j,k})| < L_{max}$$

where $x_{i,j,k}$ is a bit in a bit sequence, position of which defined by indexes i,j,k – Polarization state, Column in block of length P and bit number in symbol respectively.

Single Tree Search case

The part of modified XPolM algorithm and presented in Figure 18 is susceptible to change in a next way:

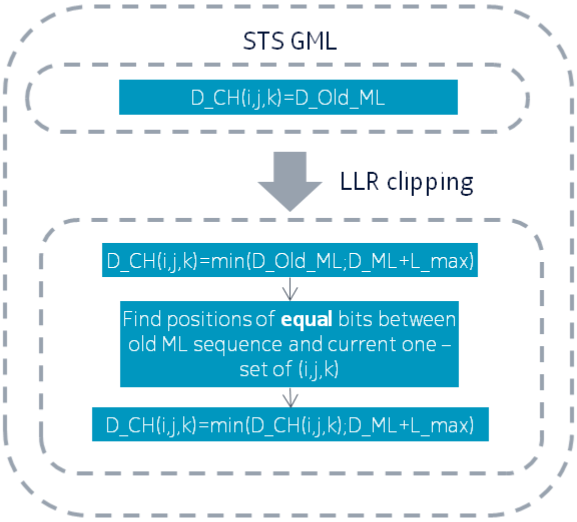


Figure 28 Introduction of LLR clipping to Single Tree Search algorithm

This step is an application of next rule:

$$\overline{\lambda^{\mathrm{ML}}_{\mathrm{i,j,k}}} \leftarrow \min\left\{\overline{\lambda^{\mathrm{ML}}_{\mathrm{i,j,k}}}, \lambda^{\mathrm{ML}} + \mathrm{L_{max}}\right\}$$

where λ^{ML} - Euclidian distance for GML solution, $\lambda_{i,j,k}^{\overline{ML}}$ – Minimum Euclidian distance of the counterhypothesis to i,j,k bit of GML decision and L_{max} - maximal value we could have in LLR sequence.

After implementation of previous steps calculation of LLR is done in same manner as non-clipped case.

Repeated Tree Search case

The part of modified XPolM algorithm and presented in Figure 13 is susceptible to change in a next way:

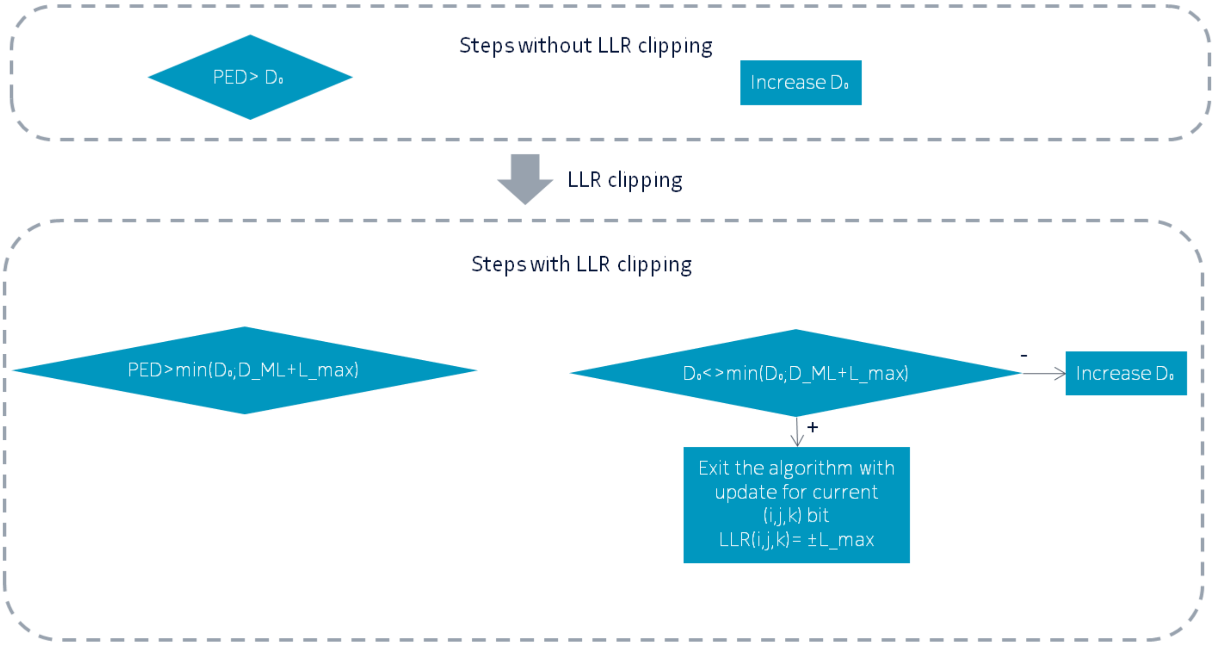


Figure 29 of LLR clipping to Repeated Tree Search algorithm

These changes are the application of a next rule:

$$D_0 \leftarrow \min\{D_0, \lambda^{\mathrm{ML}} + \mathrm{L}_{\max}\}$$

This rule directly corresponds to the fact that we limit the initial radius of search of the GML counterhypothesis in the tree by the range of distances $D_0 \in [\lambda^{\mathrm{ML}}, \lambda^{\mathrm{ML}} + \mathrm{L}_{\max}]$. If the counterhypothesis were not found we put as a final count hypothesis distance for a bit:

$$\overline{\lambda_{\mathrm{i,j,k}}^{\mathrm{ML}}} = \lambda^{\mathrm{ML}} + \mathrm{L}_{\max}$$

After implementation of previous steps calculation of LLR is done in same manner as non-clipped case.

The histograms of LLR clipped cases

As it was defined previously the range of absolute LLR values are next, depending on constellation:

- 16 QAM: $|L(x_{i,j,k})| \in [0; 3]$
- QPSK: $|L(x_{i,j,k})| \in [0; 6]$

Based on such ranges of absolute values we can arbitrary set uniformly distributed LLR-clipping values:

- 16 QAM: $L_{max} = [0.2\ 0.5\ 0.8\ 1.1\ 1.4\ 1.7\ 2.0\ 2.3]$
- QPSK: $L_{max} = [0.2\ 0.8\ 1.4\ 2.0\ 2.6\ 3.2\ 3.8\ 4.4]$

We represent example of LLRs distribution for zero message, which were at first passed through scrambler, then transmitted through the system, after LLRs of such scrambled message were computed and then descrambled.

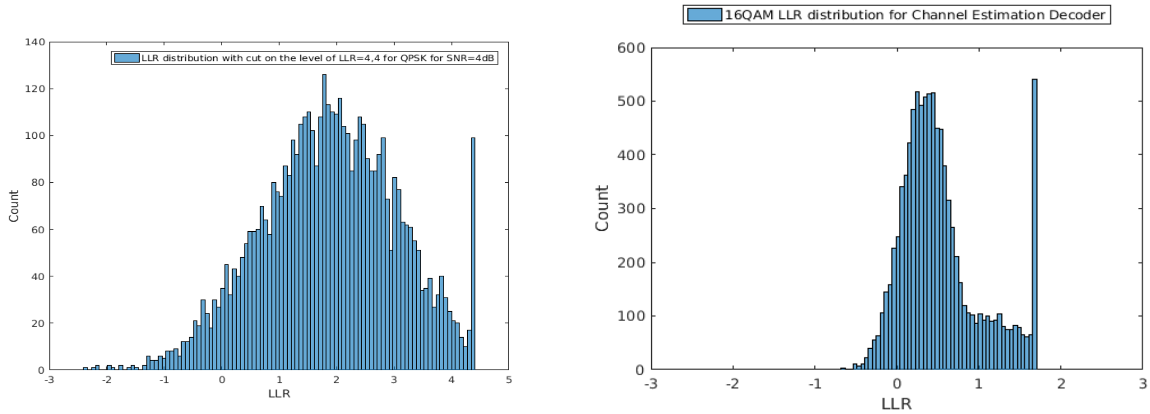


Figure 30 left) figure corresponds to the QPSK case with L_{max}=4.4; right) figure corresponds to the 16 QAM case with L_{max}=1.7

We present the results of simulation for LLR clipping for QPSK and 16 QAM:

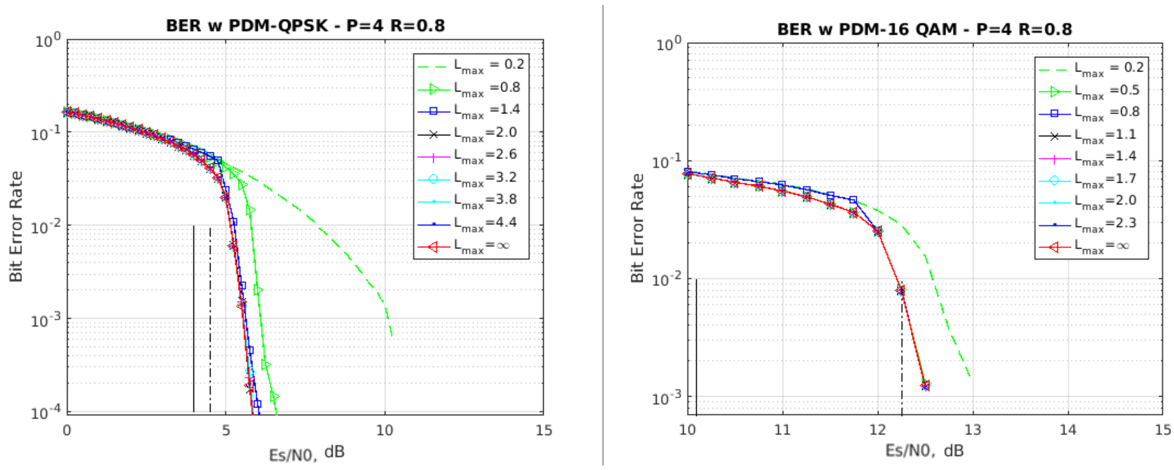


Figure 31 BER for clipping case QPSK (left) and for 16 QAM (right)

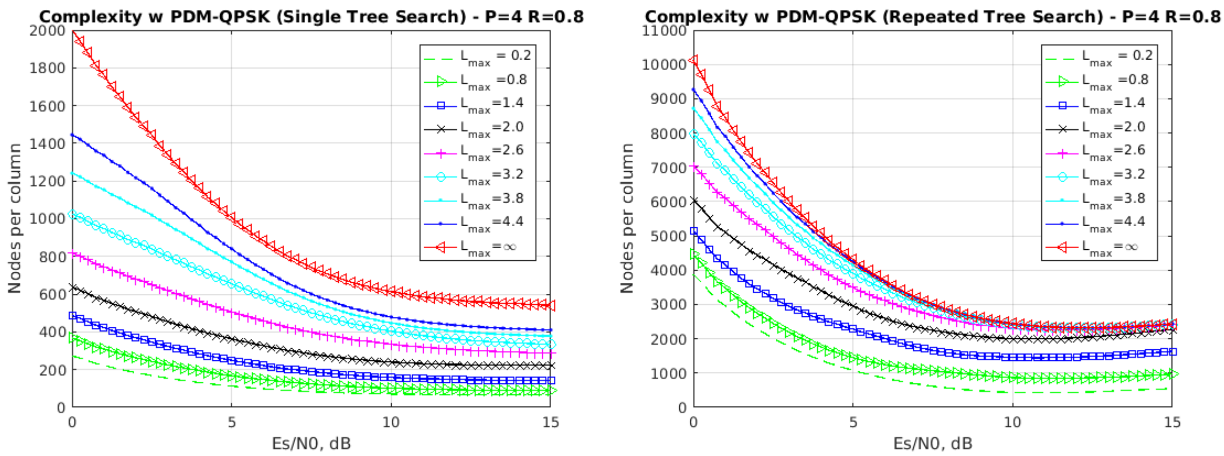


Figure 32 Complexity for STS (left) and for RTS (right) for QPSK

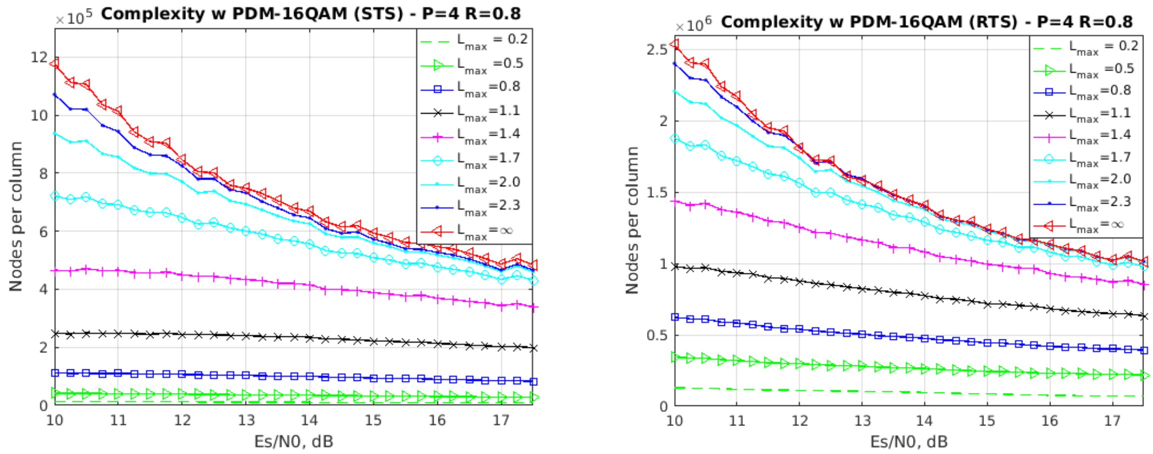


Figure 33 Complexity for STS (left) and for RTS (right) for 16 QAM

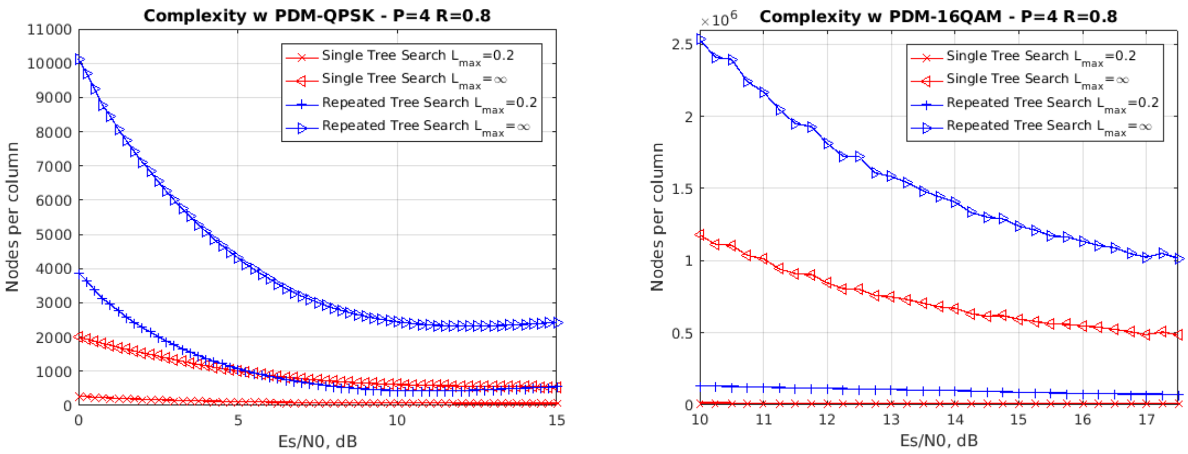


Figure 34 Comparison of the bounds of complexities for STS and RTS strategies in clipping.

Based on the results of simulations we can conclude that:

- For QPSK case:
  - We can see that after clipping value $L_{max} = 2$ the performance isn't changed while increasing the value of clipping, and is almost equal to the clipping value $L_{max} = \infty$, that is to say, without clipping, however the gain in complexity has factor more than **3** for STS case and around **2** for RTS case.
  - The single strategy search still stays more optimal one than repeated tree search strategy. The overall difference stays as in unclipped case on the level of factor **5**.
- For 16 QAM case:
  - We can see that after clipping value $L_{max} = 0.5$ the performance isn't changed while increasing the value of clipping, and is almost equal to the clipping value $L_{max} = \infty$, that is to say, without clipping, however the gain in complexity has factor more than **28** for STS case and around **8** for RTS case.
  - The single strategy search still stays more optimal one than repeated tree search strategy. The overall difference stays as in unclipped case on the level of factor **2**.
- In general – apparently the loss of performance in decoding starts to appear when the LLR values continue to be forcefully less than position of first peak – LLR_{QPSK}<2 and

LLR_{16QAM}<0.4.That is to say that we can safely clip LLRs that are bigger in modulus that provided before values.

Two parts algorithm

A novel complexity reduced algorithm is proposed in this chapter to deliver suboptimal LLR values. The idea of such algorithm lies in mixture of two approaches:

1. Decode the block of symbols of length P with hard-output using ordinary XPolM GML mitigation algorithm and derive the estimation of the channel.
2. Consequently compute LLRs values with algorithm based on Single Tree Search Strategy with sequence length of P=1, while feeding to the algorithm the estimated state of the channel from step #1.

Thus the resulted complexity will be near to the XPolM algorithm – the first contribution to the complexity makes the original GML for XPolM algorithm, and the second one comes from STS based algorithm which is constant for any SNR and equal to cardinality of joint constellation – modulation + polarization multiplexing.

We provide below the comparison of performance of algorithms in terms of BER[7]:

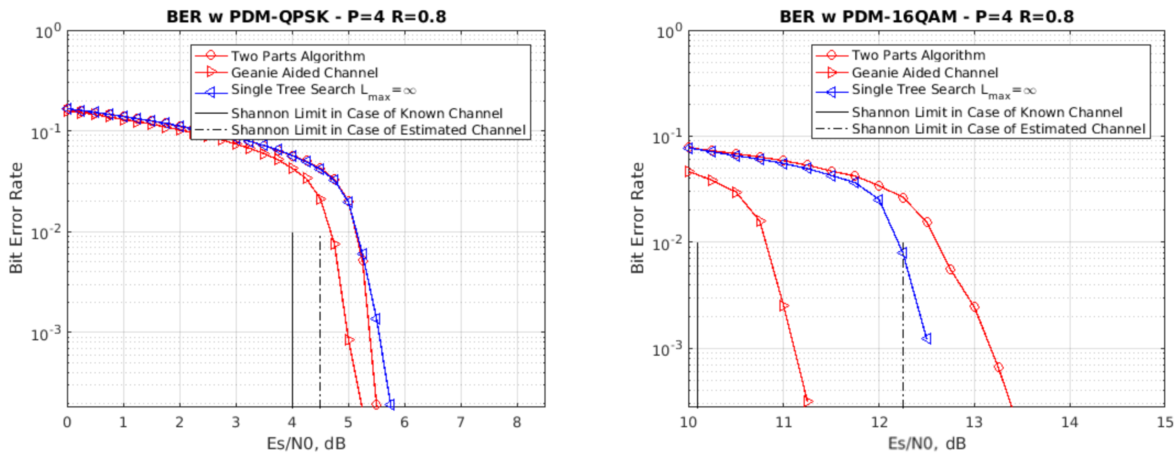
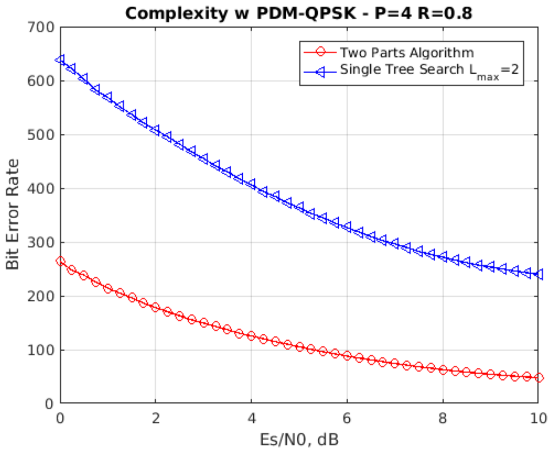


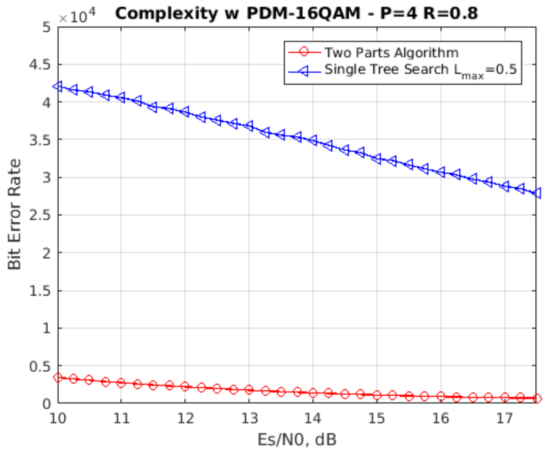
Figure 35 The BER of Two Parts Algorithm in comparison with Genie aided case and Single Tree Search without clipping – QPSK case (left) 16 QAM case (right)

We see that in terms of BER for QPSK case the algorithm performs in the same level as STS algorithm(last point in previous graph we consider as “statistical instability”), however for 16 QAM STS outperforms the Two Parts algorithm by **0,6 dB.**

Now we compare the complexity of two parts algorithm with Single Tree Search with LLR clipping at the level without loss of performance:

[7] The simulation of “Two Part algorithm” was performed with LDPC FEC clipping at level 30. That level is bigger than 20, but this doesn’t result in efficient gain in comparison – the estimated gain will be around **0,02dB.**





We see that "Two Parts" decoding approach offers huge gain in complexity – factor **2,5** in QPSK case and factor **10,5** in 16QAM, however in case of 16QAM we have to make a compromise and loose 0,6 dB in BER.

This algorithm could replace STS algorithm with LLR clipping in QPSK truly, but not in 16 QAM case.

List Sphere Decoder Algorithm Approach

The idea of List Sphere Decoder is described in [14]. We extend the idea of such decoder to the GML XPolM process by introducing a list of GML candidate solutions – N_{cand} with corresponding distances. While in original (hard output) GML XPolM we searched for only one best candidate, here we search for more than one candidate. The list should be sorted out with GML candidate on the first position. Some exact or nearby counterhypotheses to such GML decision are located in such list – thus list forcefully contains the soft information.

The original GML XPolM algorithm is modified in two parts:

- The modification of original algorithm is quite simple – the part "store solution" of step 4 in the Figure 7 should be changed with "store solution in the list" and "sort the list by Euclidian distances"
- The part of "Update D_0=PED" of step 4 in the Figure 7 should be changed to "Update D_0 with the Euclidian distance defined by $N_{assured}$ distance is the list".

The $N_{assured}$ index defines the size of the list that is sure to be filled – $N_{assured} \in [1; N_{cand}]$. One can imagine N_{cand} as a preallocated space to be filled with solutions, but **not forcefully** completely, and $N_{assured}$ is a sub-space of N_{cand}, which represent a space to be **forcefully** filled with solutions. This index directly tied with complexity – the number of nodes visited per column. The bigger $N_{assured}$ the bigger is the area of research of the tree. We can put $N_{cand} = 20$ and $N_{assured} = 1$ – then the complexity will be the same (not true in case of Bit-Flipping case reviewed further, however is close) as in the case GML XPolM original algorithm, but we do not guarantee that the whole list will be filled – the list at the end of search could contain only 10 candidates. In order to have more candidates we set the initial radius of search twice as it was in HO GML XPolM – ${D_0}_{initial} = 2.\lambda^2 N\sigma_n^2$ – in order to increase chances of having number of candidates for solutions more than $N_{assured}$.

While using List Sphere Decoder we could arrive to the case when between final GML candidates in the list there are no any counterhypotheses to a bit in GML solution. In such case we choose to proceed by one of the two options:

- Put as LLR a maximum value between LLRs that exist in the list due to presence of counterhypotheses – that would be the lower band of possible LLRs in such bit.
- Perform the bit-flipping technique – take the GML solution and by flipping the bit (0 changes to 1 and 1 changes to 0), for which there is no counterhypotheses, create artificial counterhypothesis and calculate its Euclidian Distance.

We provide the example of List Sphere Decoder for the cases with and without bit-flipping technique:

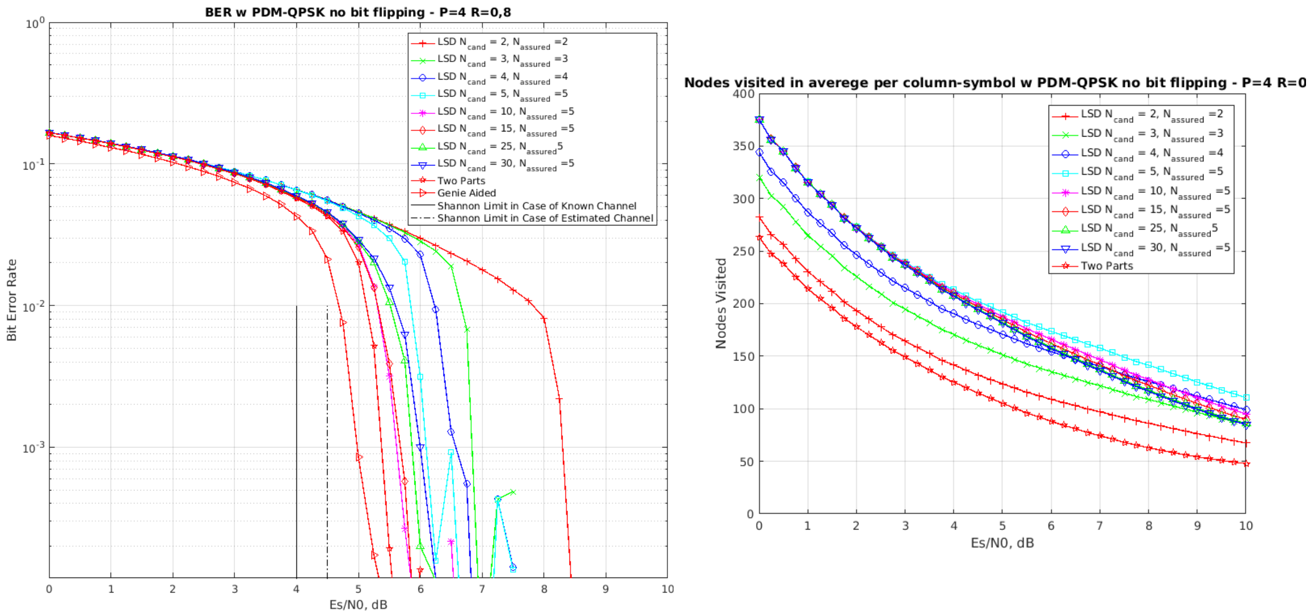


Figure 36 the List Sphere Decoder in case without Bit-Flipping case (left – BER right – Nodes visited) - QPSK

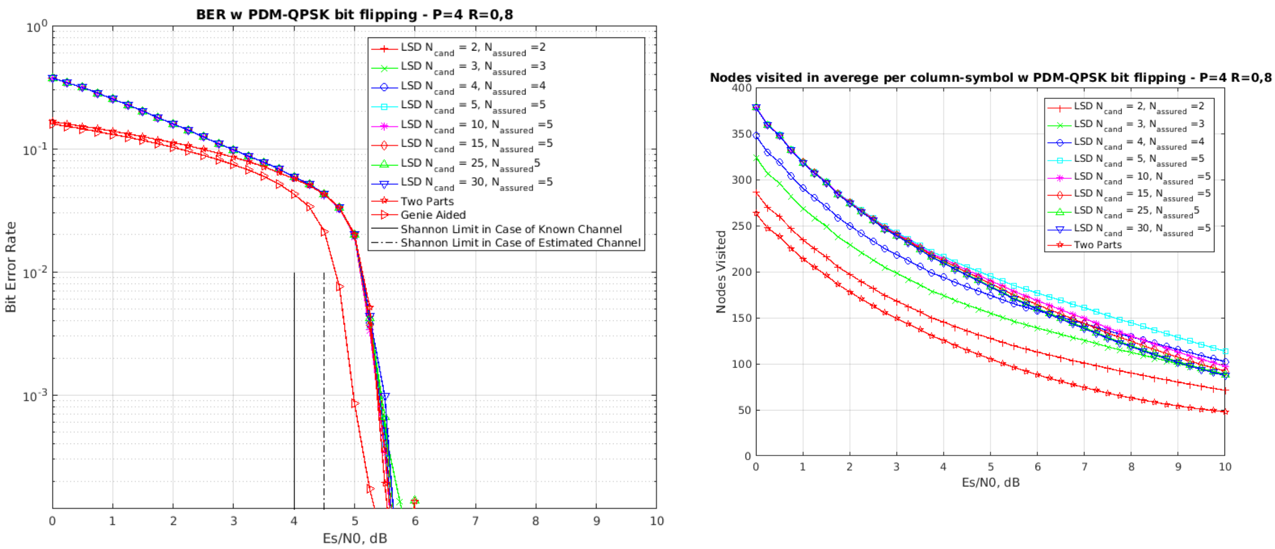
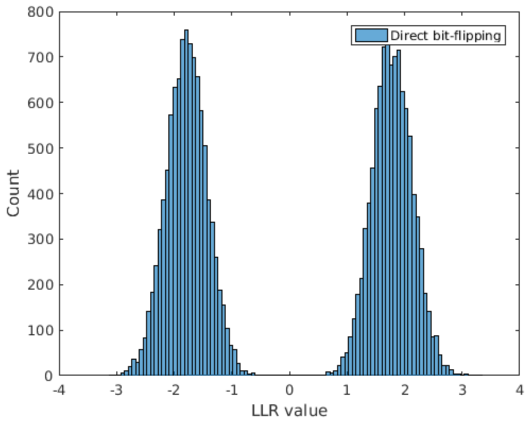


Figure 37 the List Sphere Decoder in case with Bit-Flipping case (left – BER right – Nodes visited) - QPSK

As we can see the Bit-Flipping technique is better than the technique without bit-flipping in List Sphere Decoder – the BER is significantly improved, however the increase in complexity is relatively low. That is why we opt for that solution to calculate the LLR in absence of its value at the end of List Sphere Decoder algorithm.

When we do the bit flipping, we must be careful in doing so. Here what happens when we apply direct bit-flipping (flipping only one bit) in terms of LLR histograms:



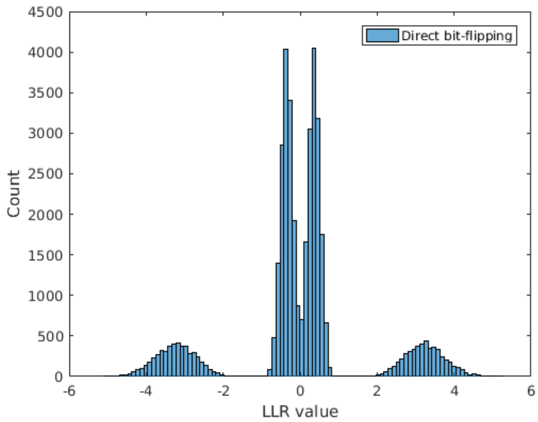


Figure 38 left) QPSK LLR Distribution right) 16 QAM LLR Distribution

We see that the position of the peaks for QPSK case correspond to the position of authentic LLR peaks provided in Figure 24, however side-peaks position for 16 QAM case does not correspond to the authentic ones – that is an evidence that we need to introduce "joint" bit-flipping – to calculate certain LLR we need to flip not only one corresponding bit, but also adjacent one.

For the gray mapping represented in Figure 27 we developed next algorithm of bit flipping:

- If we consider LLR value for 2nd or 4th bit, then we execute direct bit-flipping.
- If we consider LLR value for 1st bit, then
    - If the modulus of real part of considered symbol is the minimum possible value in constellation, then we execute direct bit-flipping
    - Else execute flipping of 1st and 2nd bits – joint bit-flipping.
- If we consider LLR value for 3rd bit, then
    - If the modulus of imaginary part of considered symbol is the minimum possible value in constellation, then we execute direct bit-flipping
    - Else execute flipping of 3rd and 4th bits – joint bit-flipping.

After introduction of joint-bit flipping we have next distribution of LLRs for 16 QAM case:

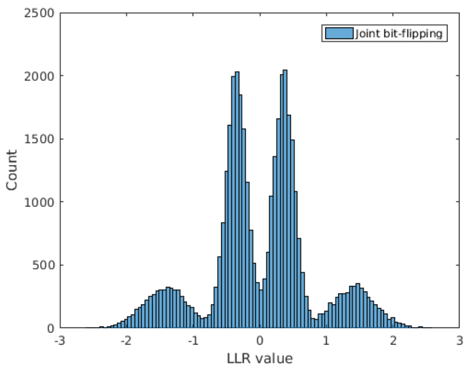


Figure 39 16 QAM LLR distribution in case of Joint Bit Flipping

We see that the LLRs range of possible values are in defined previously range and the position of the peaks correspond to the position of one's defined earlier. Also we can show the relevance of such correction in next figures:

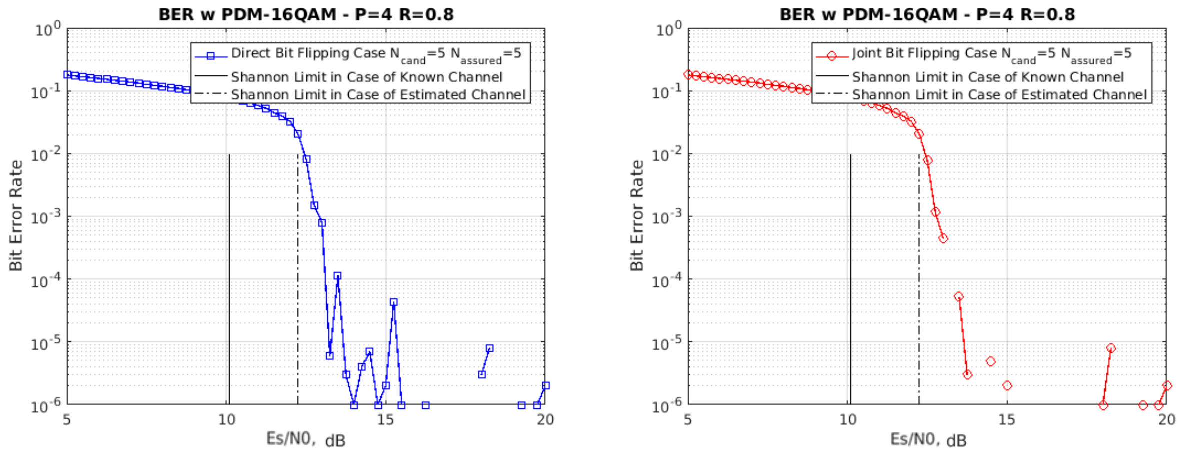


Figure 40 BER for Direct (left) and Joint (right) bit-flipping in case of 16 QAM.

We can see the difference only in those regions of BER (10^{-4}..10^{-6}) where it becomes difficult to simulate the transmission with reliable number of errors(200), however some errors arrive (40<). Nevertheless, such number of errors let us to see that some of them are corrected better in case of Joint Bit Flipping than in Direct Bit flipping case.

We present below the series of simulation for different $N_{candidates}$ and different $N_{assured}$ for QPSK case with "Two Parts" algorithm as a reference:

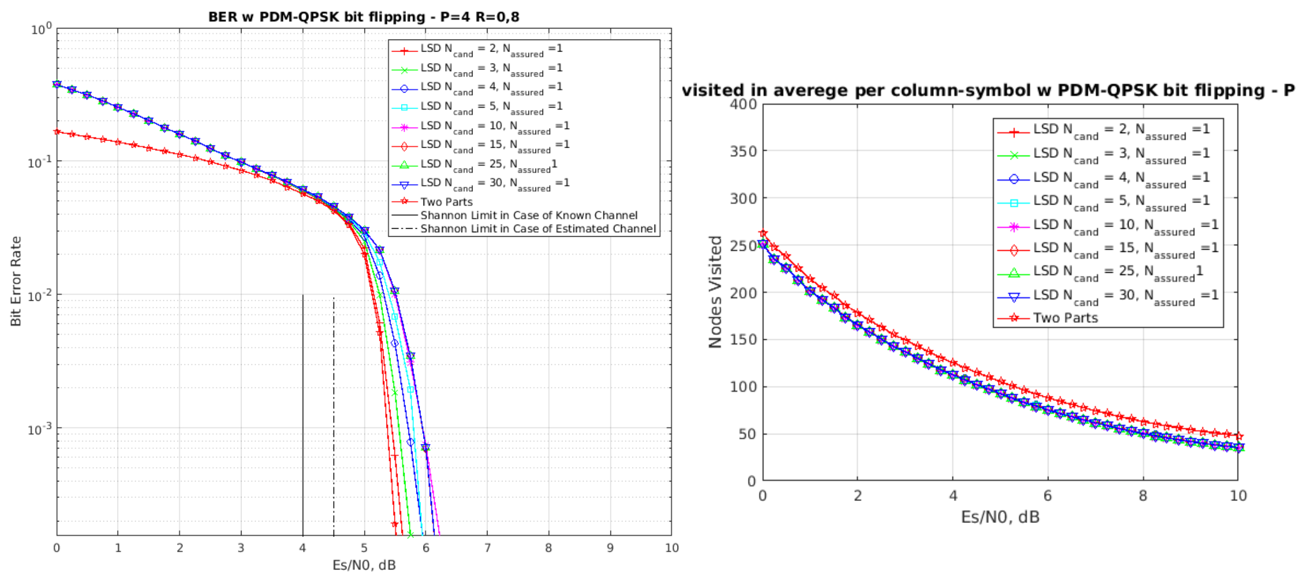


Figure 41 List Sphere Decoder with $N_{assured}$=1.

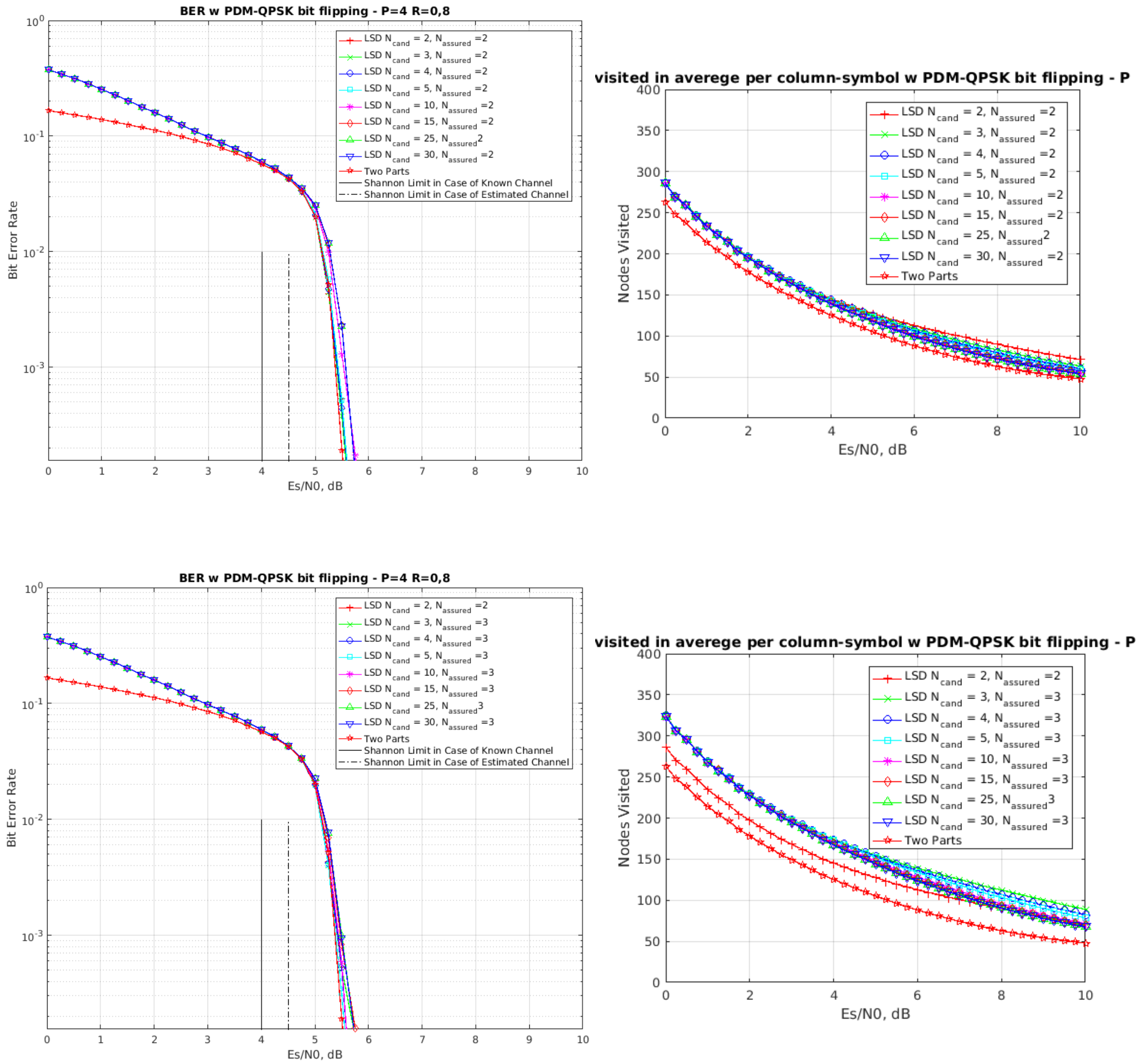


Figure 42 List Sphere Decoder with $N_{assured}$=2 and 3.

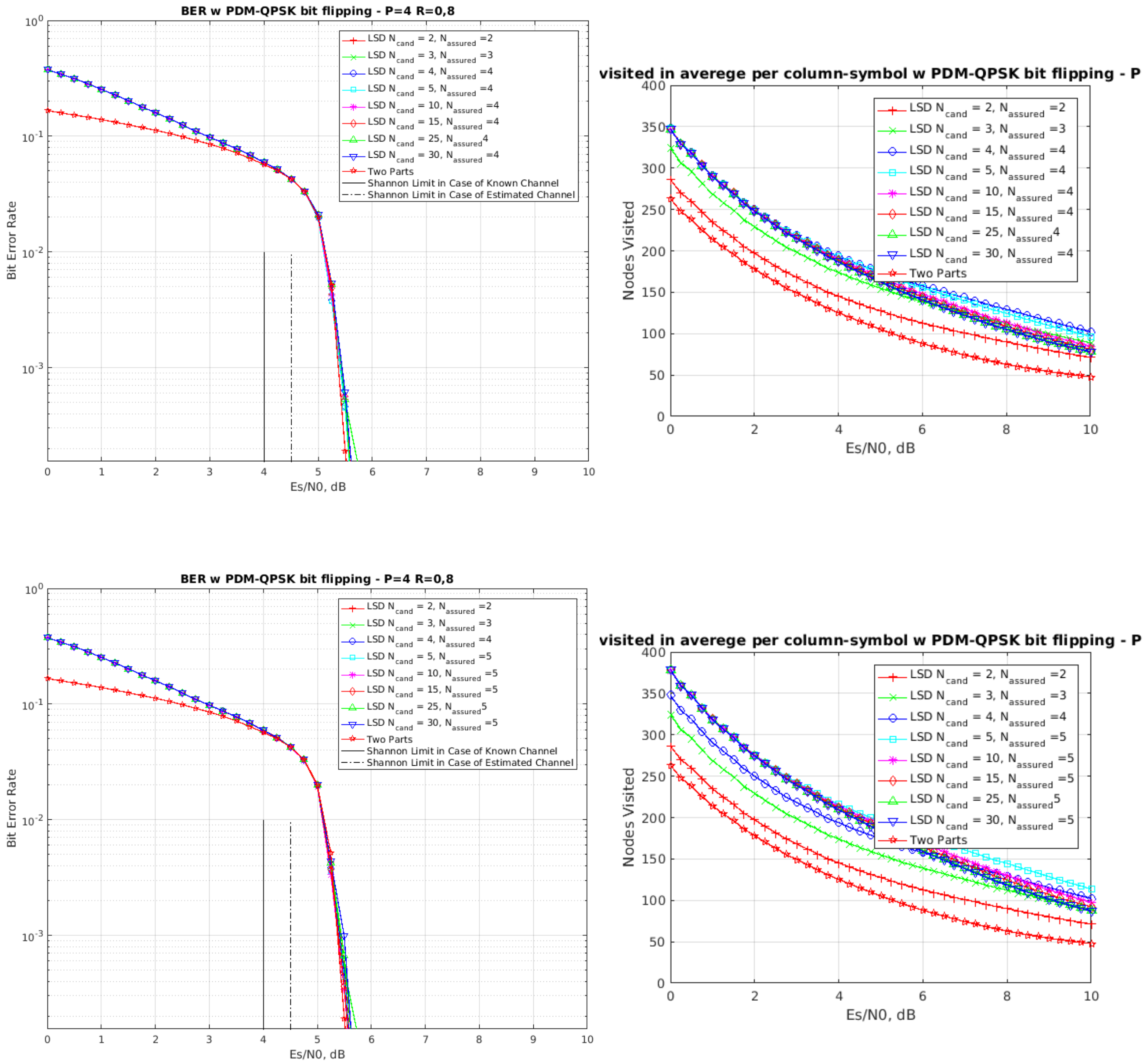


Figure 43 Different cases for List Sphere Decoder for QPSK case with $N_{assured}$=4 and 5.

In QPSK case we see that List Sphere Decoding with N_{cand}=1 algorithm bounds the performance of List Sphere Decoder and its complexity, thus making the application of other decoders for QPSK case obsolete.

We present the same simulations for 16 QAM case:

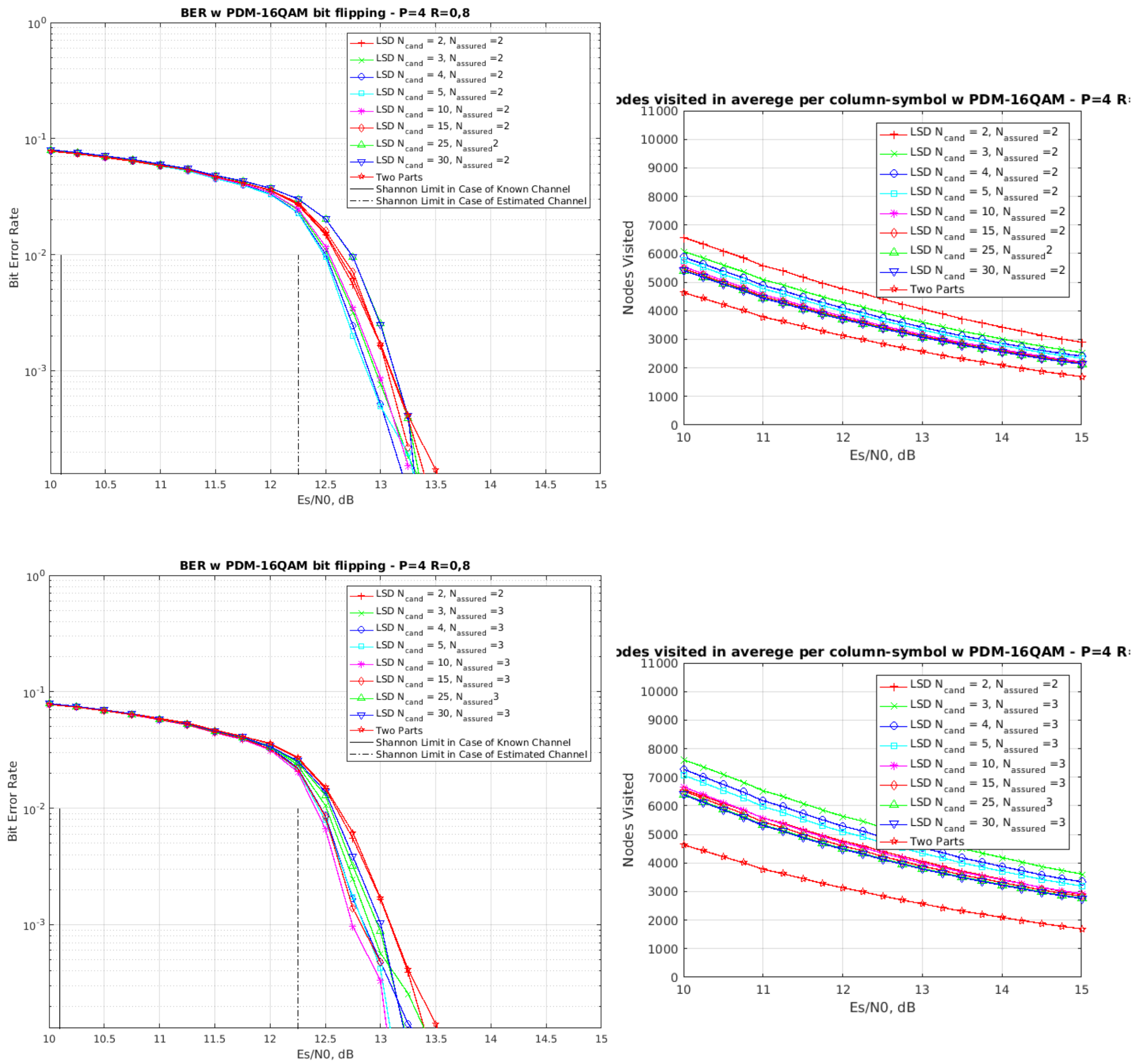


Figure 44 List Sphere Decoder with $N_{assured}$=2 and 3.

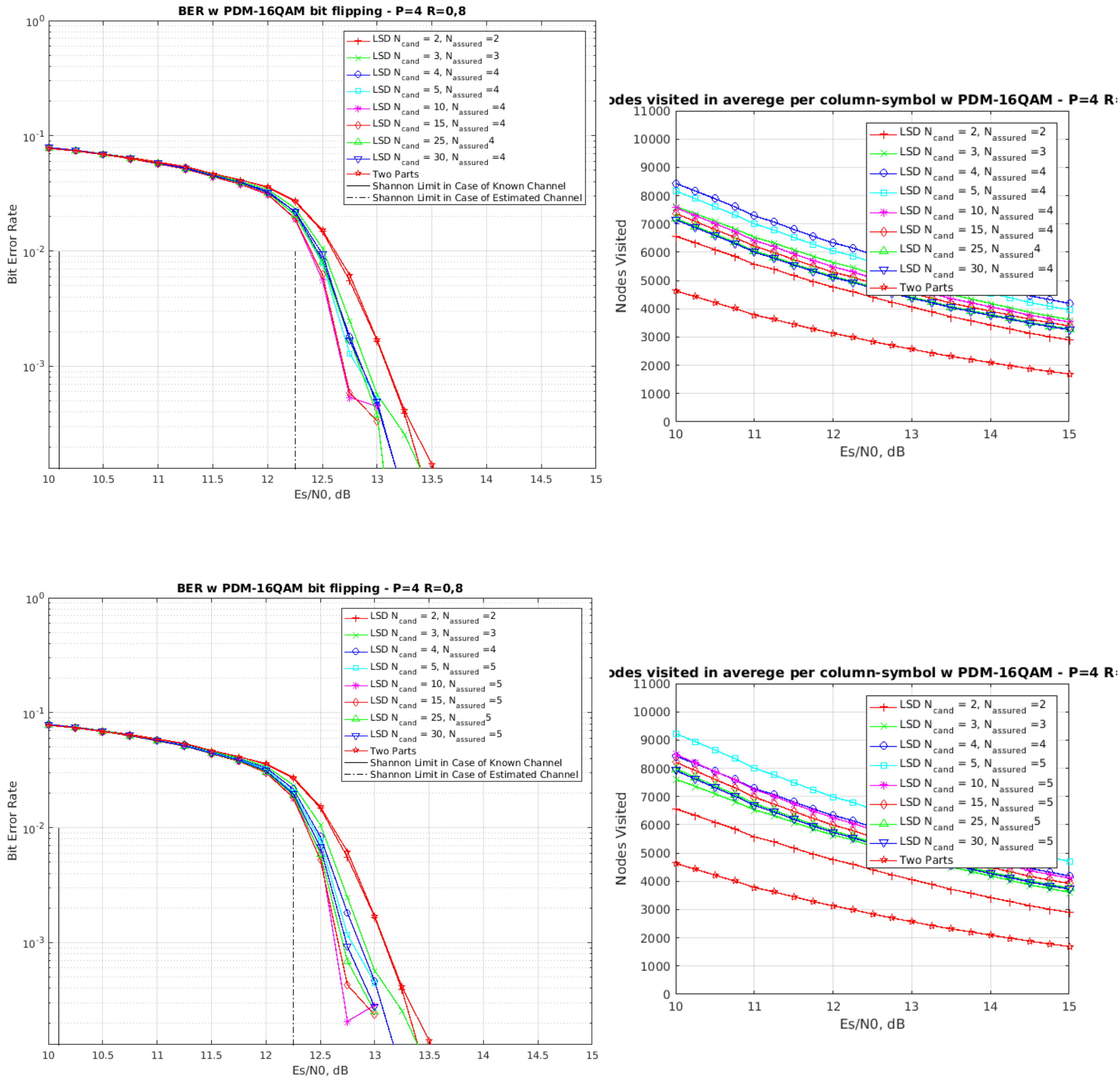


Figure 45 Different cases for List Sphere Decoder for 16 QAM case with $N_{assured}$=4 and 5.

We see that in 16 QAM case the "Two Parts" algorithm is not bounding in BER, but bounding in complexity, thus the use of such algorithm could be justified – it has BER better than "Two Parts" but complexity lower than Single Tree Search with clipping. Also we see that generally, the bigger the $N_{assured}$ the more precise results we have. Unfortunately the increase of the list (N_{cand}) while fixing the $N_{assured}$ (and fixing the complexity)does not guarantee the gain in performance, as we can see in simulations – sometime in the list we could have good candidates, which improving the situation, sometimes very bad ones. Thus, we can establish the general rule:

$$N_{assured} = N_{cand}$$

We present the comparison of Single Tree Search strategy with LLR clipping in terms of BER and Nodes Visited:

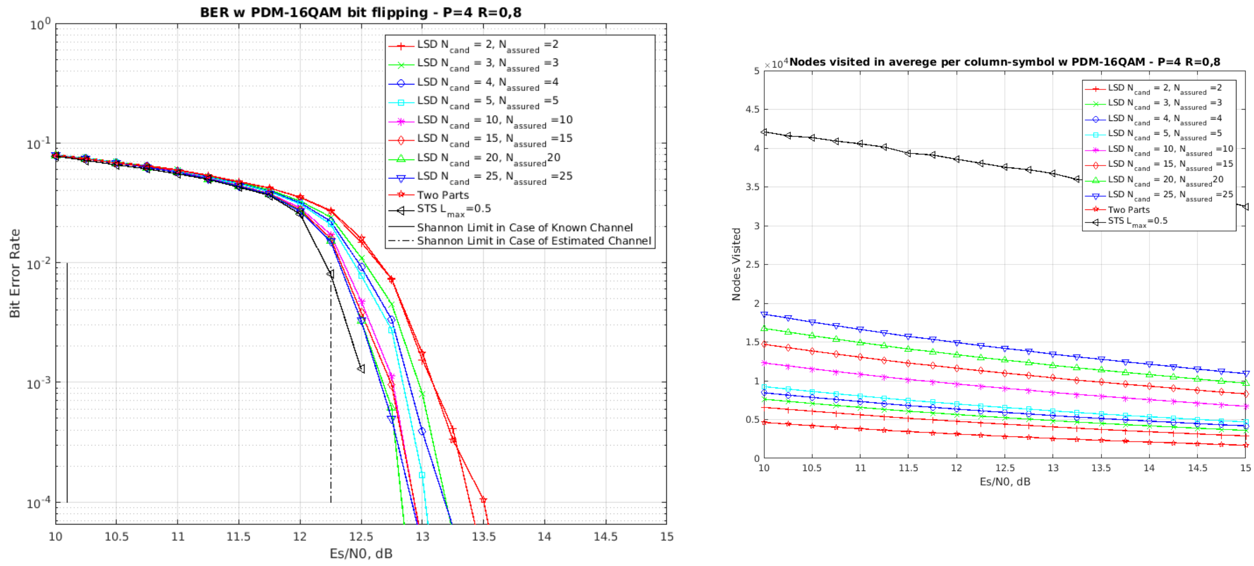


Figure 46 Comparison of different strategies in terms of BER and Complexity for 16 QAM

As we can witness, the List Sphere Decoder is bound by Two Parts Algorithm and Single Tree Search with clipping strategy both in complexity and performance. Thus, making next conclusion of choosing the strategy to implement:

- If the performance is more important than complexity – choose the Single Tree Search algorithm.
- If the complexity is more important that performance – choose the "Two Part Algorithm".
- If we want some trade-off between performance and complexity – choose the List Sphere Decoder with required number of list.

5.5. Turbo algorithm approach

We always have used the combination of two independent blocks – GML XPolM decoder that delivers sequence of LLR instead of sequence of bits, and a LDPC (FEC) block, that corrected the errors with some number of iterations. Now we propose next turbo scheme, which uses the GML blocks and LDPC correcting block consequently in cycle: estimation of LLR on the basis of estimated previously channel, correcting errors with LDPC block, estimation of channel on the basis of sequence with corrected errors, repeating the cycle N_{turbo} times . In order to be able to compare the complexity of such approach we have to decrease the number of LDPC iterations N_{FEC} by N_{turbo}. We present such evolution scheme on next figure:

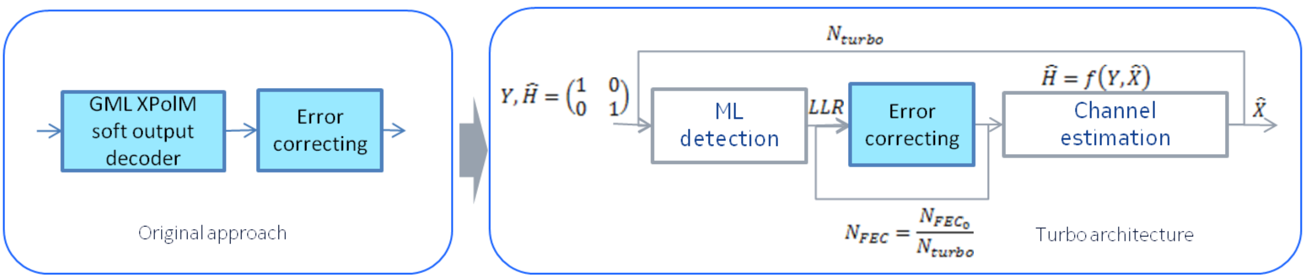


Figure 47 The scheme of Turbo Decoding Architecture.

Where N_{turbo} – number of total cycles, N_{FEC_0} – number of LDPC cycles, that were used initially, in our case it was 50. For next simulations we will put 48, to be divided by deferent spectrum of N_{turbo}.

We present the results of simulations for 16 QAM case:

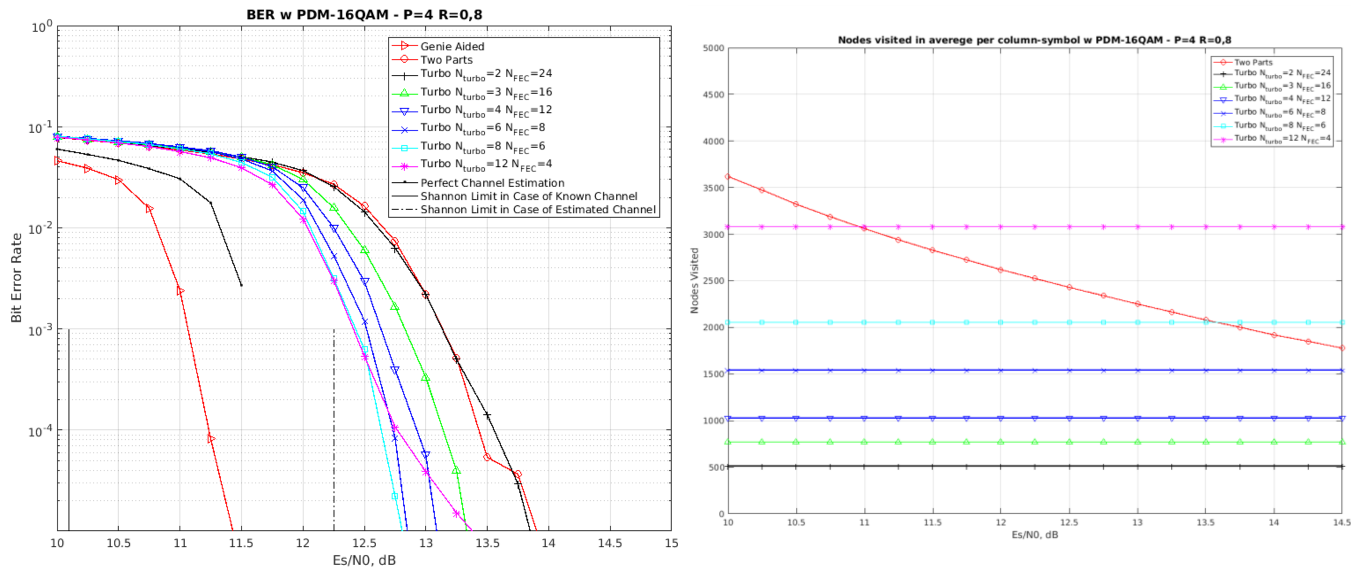


Figure 48 BER (left) and complexity (right) case for Turbo Algorithm for 16 QAM

We can witness that turbo algorithm with variable N_{FEC} has constant complexity for different N_{turbo}, which is generally lower than "Two Parts" complexity, and has superior performance to the "Two Parts" algorithm. We see that we can increase the N_{turbo} till $N_{turbo} = 8$ and still increase the performance – the BER is lower, however, when $N_{turbo} = 12$ the number of iteration $N_{FEC} = 4$ is very small, which start to deteriorate the performance after SNR=17,5 dB.

We present the simulation for QPSK:

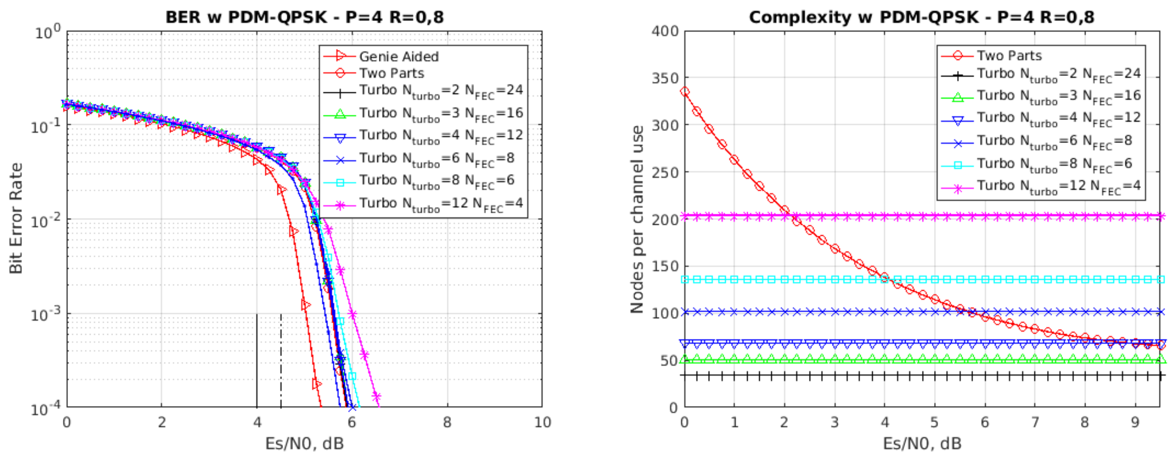


Figure 49 BER (left) and complexity (right) case for Turbo Algorithm for QPSK

We see that the performance in terms of BER approach the performance of Perfect Channel Estimation.

We also present the case for 16 QAM and QPSK without limiting the number of iterations N_{FEC} - we always put $N_{FEC} = 48$. We believe this is the best performance we can achieve with given code and given channel model and that way of its state estimation.

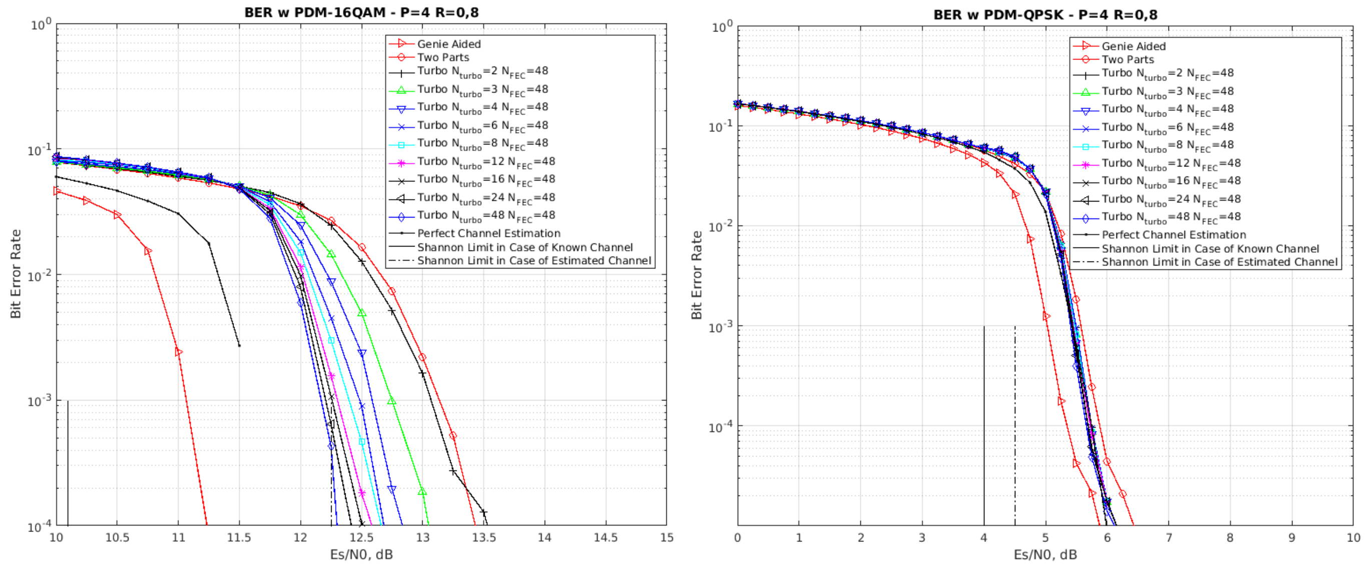


Figure 50 non limited turbo for 16 QAM(left) and for QPSK(right)

We see that for 16 QAM we can achieve the ultimate limit that we have never seen – **10.1 dB** in SNR waterfall point. And for QPSK – **5.5 dB**. Here we have general rule – the increase of N_{turbo} leads to increase of performance, but for 16 QAM it almost stop to increase after $\boldsymbol{N_{turbo}} = \mathbf{16}$ and for QPSK it stops to increase after almost immediately after $\boldsymbol{N_{turbo}} = \mathbf{2}$ .That also prove that for QPSK the effect of XPolM already compensated.

5.6.Comparison of different algorithms in terms of BER and nodes visited

We compare the best algorithms on our opinion in BER figure for 16 QAM and QPSK:

- Two Parts Algorithm
- Single Tree Search with LLR clipping with $L_{max} = 2$ for QPSK and $L_{max} = 0,5$ for 16 QAM.
- List Sphere Decoder with $N_{cand} = 1$ for QPSK and $N_{cand} = 5$ and $N_{cand} = 5$ for 16 QAM.

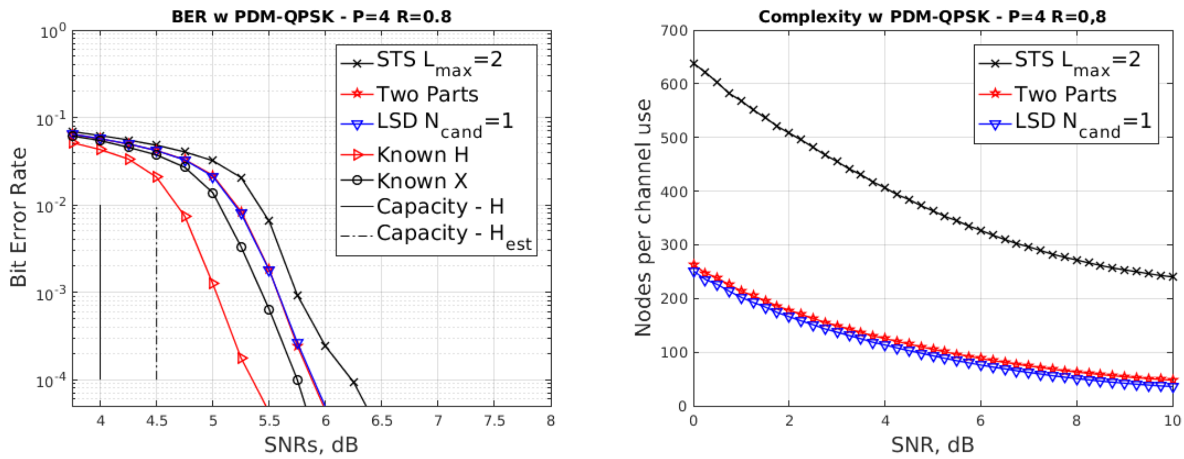


Figure 51 Comparison of algorithms for QPSK

For QPSK all algorithms in terms of BER perform almost on the same level, thus making the choice easy – choose the least complex algorithm. Thus we choose List Sphere Decoder Algorithm with $N_{cand} = 1$.

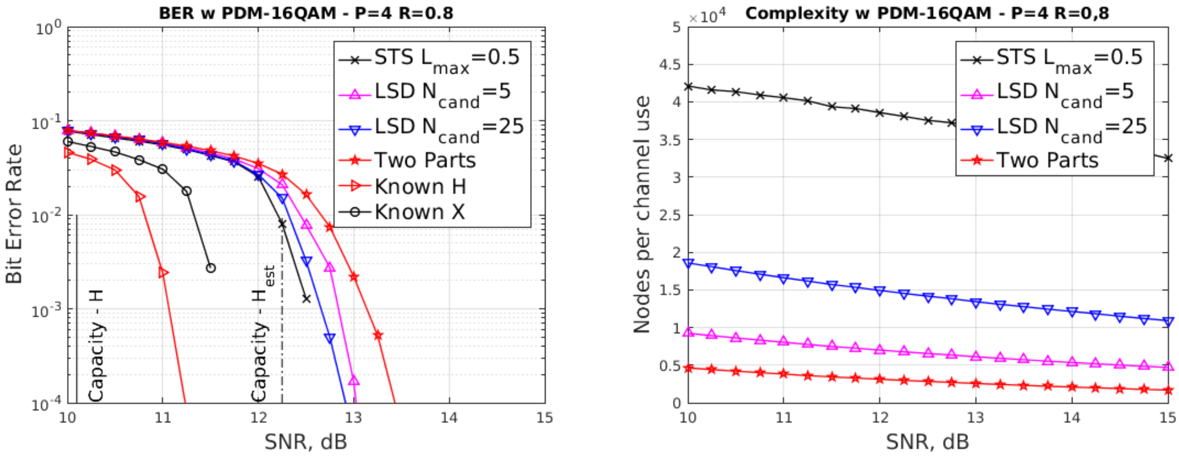


Figure 52 Comparison of algorithms for 16 QAM

The choice of algorithm for 16 QAM is less obvious – the best performance in terms of BER gives us the Single Tree Search Strategy with LLR clipping, while the best in complexity in terms of Nodes Visited gives us the Two Parts algorithm. The List Sphere Decoder is trade-off between these two algorithms, thus defining our choice – select the List Sphere Decoder with $N_{cand} = 25$. In terms of complexity it two times better that STS, but have performance near optimal.

6. Conclusion on the internship works performed

This internship focuses on the proposition of novel fast algorithms to compute the soft output GML formulation. To this end, we proposed different algorithms extended from the literature to include a general channel matrix and sequence length detection. This includes optimal decoders achieving optimal performances but at the cost of high complexity, sub-optimal decoders trading-off performance and complexity. We also investigated upper bounds with the so-called genie-aided bound (i.e. perfect channel estimation), perfectly estimated channel (i.e. perfect knowledge of the symbols sent) to assess the gap of our propositions to these upper limits. Finally, we evaluated these propositions in a number of simulations with QPSK and 16-QAM signals and discussed the performance vs. complexity trade-off.

After proposition and evaluation of different algorithms we can conclude that for specifically this impairment – XPolM – with presented channel estimation based on the GML formulation, the best algorithm to treat such non-desirable effect with introduction of Forward Error Correcting Codes is the proposed extended version of **List Sphere Decoder** for QPSK and 16 QAM with different parameters.

In addition, the turbo algorithm is a promising approach for quite low complexity but in practical systems where the number of maximum decoding iterations is 8 to 10, it reveals to lose significant BER performance compared to the other proposed methods. However, such a practical constraint may evolve if enhanced BER performances are required in next generation of systems.

Bibliography

[1] "L. Li, Z. Tao, L. Liu, W. Yan, S. Oda, T. Hoshida, and J. C. Rasmussen, "Nonlinear Polarization Crosstalk Canceller for Dual-Polarization Digital Coherent Receivers," *in Optical Fiber Communication Conference, OSA Technical Digest (CD) (Optical Society of America, 2010), paper OWE3.*

[2] " Z. Tao et al., "XPM induced nonlinear polarization crosstalk and mitigation in optical dual polarization systems," Photonics Society Summer Topical Meeting Series, 2010 IEEE, Playa del Carmen, 2010, pp. 8-9.".

[3] "P. Serena, A. Ghazisaeidi, and A. Bononi, "A New Fast and Blind Cross-Polarization Modulation Digital Compensator," in European Conference and Exhibition on Optical Communication, OSA Technical Digest (online) (Optical Society of America, 2012)," *paper We.1.A.5.*

[4] "] T. Koike-Akino, K. Kojima, and K. Parsons, "Trellis-Coded High-Dimensional Modulation for Polarization Crosstalk Self Cancellation in Coherent Optical Communications," *in Advanced Photonics 2015, OSA Technical Digest (online) (Optical Society of America, 2015), paper SpS3D.6..*

[5] P. Layec, A. Ghazisaeidi, G. Charlet, J.-C. Antona and S. Bigo, "Generalized Maximum Likelihood for Cross-Polarization Modulation Effects Compensation," *in Journal of Lightwave Technology, vol. 33, no. 7,* pp. pp. 1300-1307, April 1, 2015.

[6] "Weiyu Xu, M. Stojnic and B. Hassibi, "ON exact maximum-likelihood detection for non-coherent MIMO wireless systems: A branch-estimate-bound optimization framework," Information Theory, 2008. ISIT 2008. IEEE International Symposium on, Toronto, ON, 2008," pp. 2017 - 2021.

[7] J. A. T. Thomas M. Cover, *Elements of Information Theory,* Hoboken, New Jersey: A JOHN WILEY & SONS, INC., PUBLICATION, 2006.

[8] A. Goldsmith, S. A. Jafar, N. Jindal and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE Journal on Selected Areas in Communications,* vol. 21, no. 5, pp. 684-702, June 2003.

[9] E. Telatar, "Capacity of Multi-antenna Gaussian Channels," *Eur. Trans. Telecomm,* no. 10, p. 585–595, 1999.

[10] A. J. Goldsmith and P. P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Transactions on Information Theory,* vol. 42, no. 3, pp. 868-886, May 1996.

[11] P. Layec and A. Ghazisaeidi, "Apparatus, method and computer program operable to determine estimates". AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR Patent EP2720424 A1, 16 April 2014.

[12] C. Studer, A. Burg and H. Bolcskei, "Soft-Output Sphere Decoding: Algorithms and VLSI Implementation," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,* vol. 26, no. 2, pp. 290-300, February 2008.

[13] C. Dorize, O. Rival and C. Costantini, "Power scaling of LDPC decoder stage in long haul networks," in *2012 International Conference on Photonics in Switching (PS)* , Ajaccio, 2012.

[14] B. M. Hochwald and S. ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel," *IEEE TRANSACTIONS ON COMMUNICATIONS,* vol. 51, no. 3, pp. 389-399, March 2003.