Maximum Likelihood Sequence Estimation(MLSE) from Lattice Viewpoint

Author: Saifur Rahman

School of Engineering, The University of British Columbia

3333 University Way, Kelowna, BC, Canada, V1V1V7

E-mail: saifur.rahman@ieee.org

Abstract: **In digital transmission systems Viterbi is a very well-known algorithm to use it for sequence estimation problems. Implementing Viterbi-algorithm was an optimal way to remove the hindrance of digital data transmission due to inter-symbol interference (ISI). This course project focuses on the research paper which identifies the complication in data detection when there is an additive AWGN noise and ISI. In the referred literature problem was dealt with by transforming the problem of MLSE into finding the closest lattice vector. Two key features relating to lattice algorithms such as the nearest plane algorithm and the enumeration algorithm was discussed to solve the lattice close vector problem (CVP). Based on the mentioned algorithm a very effective sequence estimator was proposed for multi-level PAM and QAM systems. Improvisation was made on the literature by inspecting SEP (symbol error probability) for Rician and Rayleigh channel with a higher-order modulation scheme.**

Index Terms – Lattice Sequence estimation, Close Vector Problem, Nearest Plane algorithm, MLSE

I. Literature Review

In digital communication transmission with high-data-rate is a major challenge as the signal pathway must encounter ISI (inter-symbol interference). In 1972, Forney **[1]** introduced an optimal solution to the sequence estimation problem using the Viterbi **[2]** algorithm to suppress the effect of ISI. Increasing the signal power to solve the problem by SSD (symbol by symbol detector) would be an in-efficient and impracticable method. In Forney's paper, he proposed an estimation technique that minimizes the probability of sequence-error, hence thought to be more efficient than conventional SSD. Over the past decades, **[3-6]** sequence estimator scheme proposed by the researcher was merely a variant of the VA. In the referred paper **[7]**, the focus was on MLSE when we have ISI in the channel for a multi-level lattice-type modulation scheme, e.g. M-ary QAM. The main objective was to implement a lattice structure into the modulation scheme as opposed to the Markov channel. LSE was proven to be simple in implementation and execution but suboptimal due to relaxation in its scheme. The following improvisation has been made in this project in contrast to the referred literature. LSE has been derived for a QAM-system with both real and complex-valued channel impulse response. This implies that the channel has both baseband and passband properties. Along with the AWGN channel set example, the other channel type e.g. Rayleigh and Rician channel were examined. The tightness of the bound for eq (11) and eq (13) in the literature was investigated using Simulink and was linked to the BER-analysis tool of MATLAB **[9]** for Monte-Carlo Simulation. Monte Carlo simulation provides a meaningful interpretation for the tightness of the Symbol Error rate of eq (11) & (13) of **[7]** for a large sample size. As a result, for large symbol size 'm' & large 'SNR' it was found to provide more meaningful reasoning using Monte-Carlo Simulation.

This paper has been outlined as follows. In Section II, we have the basic terminologies and notation used throughout this paper and a brief description of the Channel model. In Section III, we have Problem statement and enunciation of Lattice and close vector problem (CVP) using two key algorithms explained in the literature. In Section IV, we have equations on performance bound to simulate the objective figure, **fig. 9** of the literature. In Section V, the algorithm and code structure has been summarized into a Pseudocode for ease of understanding the motive. In section VI, we have the simulation results, replicating the paper output and improvised outcome for Lattice CVP. A comparison has been made on modulation schemes and different channel types. Finally, in section VII, a summary of this project analysis and a few open-ended problems were stated which were reviewed from the later published literature **[2-7]**.

II. Basic Notations and Channel Model

MLSE for lattice interpretation works for any modulation scheme having lattice-type signal constellation. It includes schemes e.g. PAM and QAM. Fig.1 below shows an example of a data transmission system. The transmitting filter $f_{trans}(t)$, converts the message sequence 'x' into an electrical signal for ease of transmission. The channel *C(t)* is the interim medium to transmit message 'x' to the receiver 'z'. Additive Gaussian noise is denoted as *n(t)*. The received filter $f_R(t)$ suppresses the noise power outside the channel BW (bandwidth). The whitened matched filter **[4]** $f_w(t)$, focuses on the symbol 'z' at the receiver to provide sufficient statistical to the received signal. We can model the transmitting filter $f_{trans}(t)$, $f_R(t)$, $f_w(t)$, *C(t)*, and the symbol rate sampler of fig.1 as a discrete-time channel. The model of discrete-time channel/ filter is given in fig.2 and stated in eq (1).

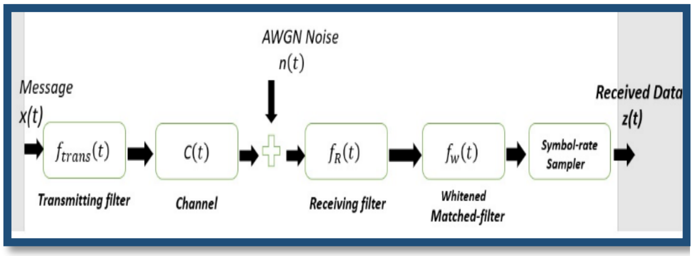

Fig 1: Data Transmission System

$$h(tT) = f_{trans}(t) * f_R(t) * f_w(t) * C(t) \qquad \boldsymbol{eq.(1)}$$

where, T- is the symbol rate

The noise sequence can be defined as:

$$w(tT) = n(t) * f_R(t) * f_w(t) \qquad \boldsymbol{eq.(2)}$$

The assumptions on the discrete-time channel model followed in the literature are stated below.

Assumption 1: Channel 'h' is linear and has a finite impulse response (FIR) of length $'v+1'$.

Assumption 2: Noise sequence $n(t)$ as in fig.1 is AWGN noise and is denoted by $w(t)$.

Assumption 3: Message symbol $'x_i'$ follows i.i.d distribution.

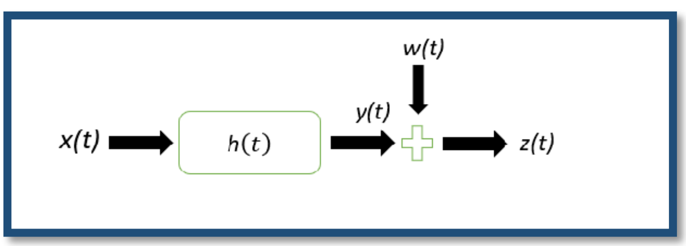

Fig 2: Discrete-time channel model

This channel can be viewed as a convolutional encoder over a real number field. In simulation $'h'$ experimented with both real and complex values in MATLAB. In table 1. We have a description of the basic symbol and their corresponding length in terms of the channel memory. Note the incoming message sequence $'x'$ is independent of the channel memory as these are drawn randomly at i.i.d.

Symbol	*Description*	*Length*
$\boldsymbol{h}$	*channel impulse response*	v
x	*message sequence symbol*	l
y	*transmitted symbol sequence*	$l+v-1$
z	*received symbol sequence*	$l+v-1$
w	*noise symbol sequence*	$l+v-1$
$\boldsymbol{\hat{x}}$	*detected symbol sequence*	$-$
$\boldsymbol{\hat{y}}$	*output symbol sequence*	$-$

- *where v is the channel memory length*

Table I: Basic Notations and their corresponding memory length

From Table I. we have the following equations as given below.

$$y = h * x \qquad \boldsymbol{eq.(3)}$$
$$\hat{y} = h * \hat{x} \qquad \boldsymbol{eq.(4)}$$
$$z = y + w \qquad \boldsymbol{eq.(5)}$$

The distortion in the channel is thought to be due to the ISI. If we take ISI to the above equation, we can rewrite the modified equation as:

$$y_i = h_o\, x_i + \sum_{j=1}^{v} h_j\, x_{(i-j)} \qquad \boldsymbol{eq.(6)}$$

The summation term in eq.6 is the ISI and the only distortion parameter considered in the literature channel model. The above equation paves the way to the finite-state description of the channel. In *fig.3*, contents of the storage elements are denoted by the state of the machine and each output symbol denotes the state transition output.

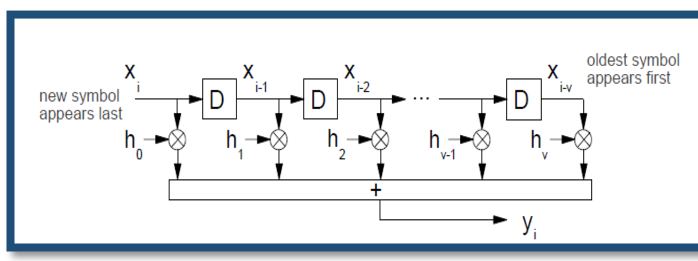

Fig 3: Finite State Machine Model [7]

Since there is a state dependence on the channel, it follows a Markov process. In fig 3. Elements above are ordered in such a way that the oldest symbol appears first, and the latest symbol appears last. The main objective of the literature was to identify the MLSE problem given the received sequence $'z'$, obtain a detected sequence $\boldsymbol{\hat{x}}$. In a way that minimizes the Euclidean distance $'d'$ to $d_{min} = \| h * \bar{x} - z \|$, between the output and received sequence.

III. Problem Statement and Enunciation

Forney **[1],** explained in his paper that due to the truncation effect there is a negligible loss in performance of 'VA'. In the referred literature a decomposition of MLSE was proposed which is, in fact, a variant of the truncated VA. However, each decomposed VA can be thought of as an MLSE with its dimension truncated in correspondence to the original MLSE. Therefore, decomposition is independent of MLSE implementation. This independence was illustrated in fig.6 of the literature. The stated MLSE problem can be transformed into a 'nearest lattice point (NLP) problem'. NLP states that given a $\delta - vector,\ 'q'$ finds a $\delta - vector\ \bar{x}$ in the lattice H{0,1……m-1} of order δ such that the chosen vector is closest to the query point $'q' - in\ Euclidean\ distance$. Fig. 4 illustrates that the closest lattice vector to the test query point $'q'$ is (1,0) in the 2-D constellation diagram.

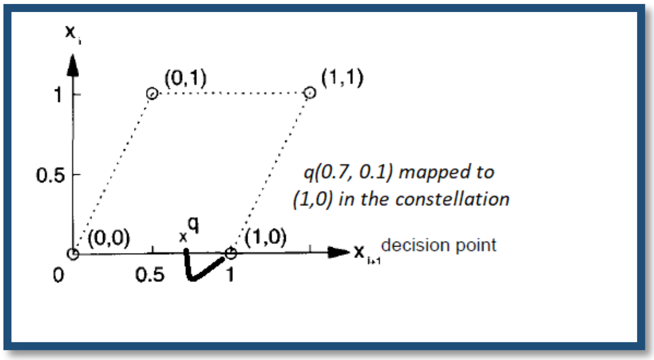

Fig 4: Lattice interpretation of 2-D MLSE [7]

CVP (closest vector problem) also termed as the nearest lattice-point problem is a typical algorithm broadly studied in computational geometry **[5]**. Computational geometry is a study of geometrical aspects of solving an analytical problem. It has a vast range of applications in areas of computer science and engineering, such as pattern recognition, computer graphics, image processing, statistics, CAD-design, robotics, etc. Kannan **[8]** provided a general solution to CVP. A standard CVP identifies the radius of a small spherical region that inscribes the required lattice vector. The latter process involves enumerating an algorithm to determine all the lattice vectors in the inscribed region. The problem statement and its enunciation are outlined below.

Step I: Search for a lattice vector $'x'$, which is closest to the query point $'q'$ and both $'x\ and\ q'$ obeys the following inequality:

$$||\hat{x} - q|| \leq ||x - q|| \qquad \boldsymbol{eq.(7)}$$

It was shown in the literature that inequality has the closest lattice vector $'\ \hat{x}\ '$ under lattice interpretation.

Step II: Enumerate all the lattice points inside a spherical region to speculate the closest vector $\hat{x}$. In the literature, Babais algorithm was used to formulate the nearest plane algorithm, and Fincke and Pohst **[5]** algorithm was used to formulate the enumeration algorithm.

The two algorithms are stated in brief below:

A. Nearest Plane algorithm: Expressing the query vector $'q'$ in terms of $b_i{}^*$ the algorithm has been derived in **[14]**. The simplified expression is given below.

$$||q - x||^2 \leq \frac{1}{4}\sum_{i=1}^{n} \quad ||\,b_i{}^*||^2 \qquad \boldsymbol{eq.(8)}$$

From eq.8, one can easily deduce that the lattice vector $'x'$ is indeed a close vector to the test query vector $'q'$.

B. Enumeration algorithm: Suppose that $r = ||q - x||$ denotes the radius of the sphere to be enumerated for n-tuples. The enumeration algorithm can be simplified to the following inequality.

$$\sum_{i=1}^{n} \quad ||\,(n_i - \ v_i)b_i||^2 = \sum_{i=1}^{n} \left\{\sum\nolimits_{j=1}^{n}(n_j - v_j)\ u_{j,i}\right\}^2 ||\,b_i{}^*||^2 \leq r^2 \qquad \boldsymbol{eq.(9)}$$

IV. Simulation Result of Literature

In this section, the methodology and simulation result of the literature would be discussed. An upper bound on Symbol Error Probability (SEP) of LSE was derived in section VII of the literature. At first error probability of two events was expressed in the form of Q-function. An expression for total probability was found by cascading the probability of these two events. SEP of LSE from eq. (11) of the literature is:

$$SEP_{LSE} \leq \lim_{m \to \propto} \ \sum\nolimits_{d\ \varepsilon\ D(m)} Q\left(\frac{d}{2\sigma}\right) \sum\nolimits_{\varepsilon\ \epsilon\ Ed(m)} W_H(\varepsilon)$$

$$= \lim_{m \to \propto} F(m) \qquad \boldsymbol{eq.(10)}$$

$where\ d(\varepsilon) = ||h *\ \varepsilon||;$

$W_H: is\ the\ hamming\ weight\ of\ the\ error\ event\ '\varepsilon\,';$

and F(m) denotes the Forney's upper bound of MLSE,

$$F(m) \triangleq \sum_{d\ \varepsilon\ D(m)} Q\left(\frac{d}{2\sigma}\right) \sum_{\varepsilon\ \epsilon\ Ed(m)} W_H(\varepsilon) \prod_{i=0}^{n-v} \frac{m - |\varepsilon_{xi}|}{m} \qquad \boldsymbol{eq.(11)}$$

As $m \rightarrow \propto$ *and* $\sigma \rightarrow 0$, *eq* .10 approaches eq. 11. The performance of LSE is asymptotically optimal in the sense that it is identical to MLSE when,

$$m \rightarrow \propto, \sigma \rightarrow 0, \; d \rightarrow d_{min}$$

The ratio of {eq.10: eq.11} approximates to '1'and eq (10) can be re-written as:

$$SEP_{LSE} \le \lim_{m \rightarrow \propto} F(m) \sim \lim_{m \rightarrow \propto} Q(\frac{d_{min}(m)}{2\sigma}) \sum_{\varepsilon \, \epsilon \, Ed(m)} W_H(\varepsilon) \quad \textbf{eq.(12)}$$

To achieve a meaningful response to the left-hand side of eq. (10) must be less than '1'. For all channel type F(m) doesn't converge, but for FIR (finite impulse response) and high SNR, F(m) converges to a finite limit.

The simulation was obtained for different values of $'SNR'$ and $'m'$. The channel impulse response considered was real-valued, where $h = (1, 0.5)$. Fig. 5 below is the objective simulation output for this project. From the figure, we have truncated version of the upper bound for MLSE and LSE taking four error events $'\varepsilon i' - into\ account.$

$$\varepsilon\,(1) = \pm 1; \quad m = 2\,'$$
$$\varepsilon\,(2) = \pm(1, -1); \quad m = 4\,'$$
$$\varepsilon\,(3) = \pm(1, -1, 1); \quad m = 8\,'$$
$$\varepsilon\,(4) = \pm(1, -1, 1, -1); \quad m = 16\,'$$

The sample size for each event was 1000, for a single error event. The truncation depth $\delta = 5(\vartheta + 1) = 10$ was chosen as a practical rule of thumb.

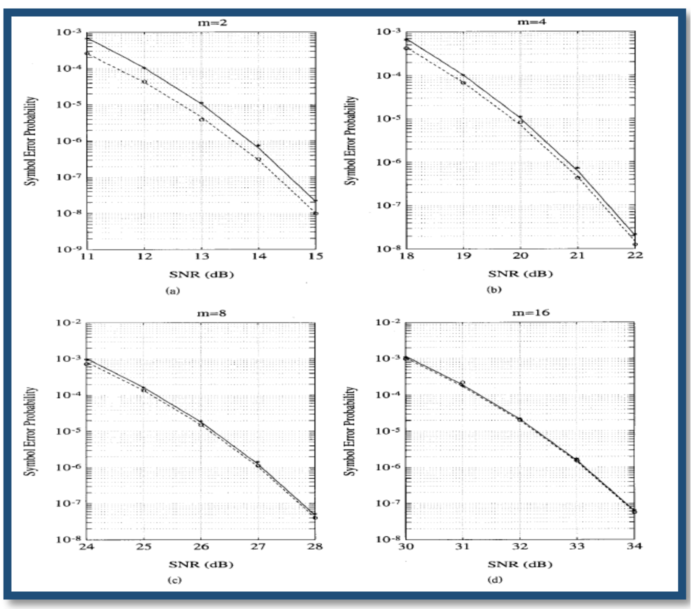

Fig 5: Objective figure- Simulated Performance bound for LSE & MLSE [7]

From figure 5. it is evident that LSE shows optimal performance for $m \geq 8$. Therefore, it can be proven that eq.10 approaches eq.11 for higher modulation order $'m'$ the SNR gap decreases between LSE & MLSE upper bound. As the value of $'SNR'$ *and* $'m'$ *increases the* boundary error in the Voronoi region of lattice decreases and SEP approaches to its optimal value. The performance degradation for m=2 is due to the difference in eq. (10) & eq. (11). So, for lower modulation order there exists a notable difference between the upper bound of SEP_{LSE} & $F(m)$.

V. Pseudocode of the Algorithm

This section summarizes the coding algorithm of the objective simulation output, for ease of understanding the motive. Two key algorithms of CVP have been outlined below, the *Nearest plane algorithm* & *the Enumeration algorithm.*

Algorithm 1: Closest Vector Problem

for i = 1 to n do;

for j = (i – 1) to 1 do;

$u_{ij} = \{ b_i^T b_j - \sum_{k=1}^{j-1} u_{jk} \, u_{ik} \, \beta_k \} / \beta_k$;

if $|u_{ij}| > \frac{1}{2}$, *do* ;

$\eta = round\ (u_{i\,j});$

$b_i = \{b_i - \eta\ b_j\}\ mod\ \ M$; - (for M- order scheme)

$u_{ij} = \ u_{ij} - \eta;$

end if

end for

$\beta_i = ||b_i||^2 - \sum_{k=1}^{i-1} u_{ik}^2 \ \ \beta_k$;

end for

For an m-ary system, the Voronoi region of a lattice takes a parallelepiped structure. Given the multi-dimensional structure of a lattice, there is a high probability for a q-point to reside outside the inspected lattice and have the closest lattice point on the boundary region. To enumerate n-tuples lattice point in a sphere as stated in eq. (9), we can inspect for all n-tuples of the parallel face of the lattice and consider each hyperplane for which the q-point doesn't lie in its corresponding parallel face pair. A vector projection of $'q'is\ found\ in\ every\ such\ hyperplane.$ If a projected vector is found in the boundary region, it is substituted to the original q-point.

Algorithm 2: Enumeration Algorithm

1. $\bar{q} = H^{-1}q$;

2. $for\ i = 1\ to$ n, do;

3. $\bar{d_\iota} = \bar{q}_i - \frac{m-1}{2}$;

4. if $| d_i | > \frac{m-1}{2}$, do $a_i = 1; else\ a_i = 0$;

5. end for;

6. $d = q - \frac{m-1}{2}$ H (sign $\bar{d}$) +1);

7. $r = \infty$; $q' = q$;

8. $for\ i = 1\ to\ n,\ do$;

9. if $a_i = 1\ , do$;

10. $\bar{q} = H^{-1}\,(q - (d^T N_i) N_i)$; $\bar{q} = \min(max)\,(\bar{q}, 0),\ m - 1)$; p = H$\bar{q}$ (normalizing each column);

11. if $r > ||p - q||,\ do$;

12. $r = ||p - q||$; $q' = p$;

13. end if;

14. end if;

15. end for;

16. return q';

VI. Project Objective: Simulation

In this section, we have the simulation result replicating the outputs of fig. 5, using MATLAB & Simulink. Note the simulated output only follows the trend as the message 'x' is a random input in the code.

I have stated the key MATLAB command used to simulate the objective result.

a. *MLSE Equalizer (comm.MLSEEqualizer):* This object, in particular, uses a Viterbi – algorithm to linearize modulated signal through a dispersive channel. It processes input frames and returns MLSE of the message signal $'m'$. MLSE equalizer processes an estimate of our channel model using the mechanism of channel estimation and uses FIR as a default filter. Hence, this MATLAB tool will exactly serve the purpose of our simulation **[9]**.

MATLAB command:

```
Mlse_LSE =
comm.MLSEEqualizer('TracebackDepth',10,...

'Channel',chCoeffs,'Constellation',pammod(dat
a, M, Phase));
```

where, h → = `chCoeffs = (1,0.5);`

δ → TracebackDepth = 1 ;
M → { 2, 4, 8, 16 } ;

`Phase: offset was set to '0' in our example.`

b. *Modulation Scheme(qammod)*: it modulates input message 'x' by QAM, and the order of modulation is 'M' which is usually expressed in the power of '2'.

MATLAB command:

```
y = qammod(x,M); where x was i.i.d in my
simulation and M = {2,4,8,16};
```

c. *Semi-analytic function (*`semianalytic`*)* returns error probability after analyzing transmit and receive signal analytically after applying Gaussian distribution. This function averages error probability over the entire received signal to determine the overall error probability. The received signal is related to the transmitted signal by the eq. (6) for the channel with ISI. It returns bit-error-rate or upper-bound of BER which is SEP (symbol error probability) in our case.

MATLAB command:

```
Sep_LSE =
semianalytic(txsig,rxsig,'qam',M,N_sample,num
,den, SNR);
```

```
where,

-txsig: is the i.i.d message sequence generated
randomly;
-rxsig- is the relation used in eq.6 between
'x & y');

- N_sample = 1000 (according to paper model);

- SNR: is the range of x-axis in the plot.
```

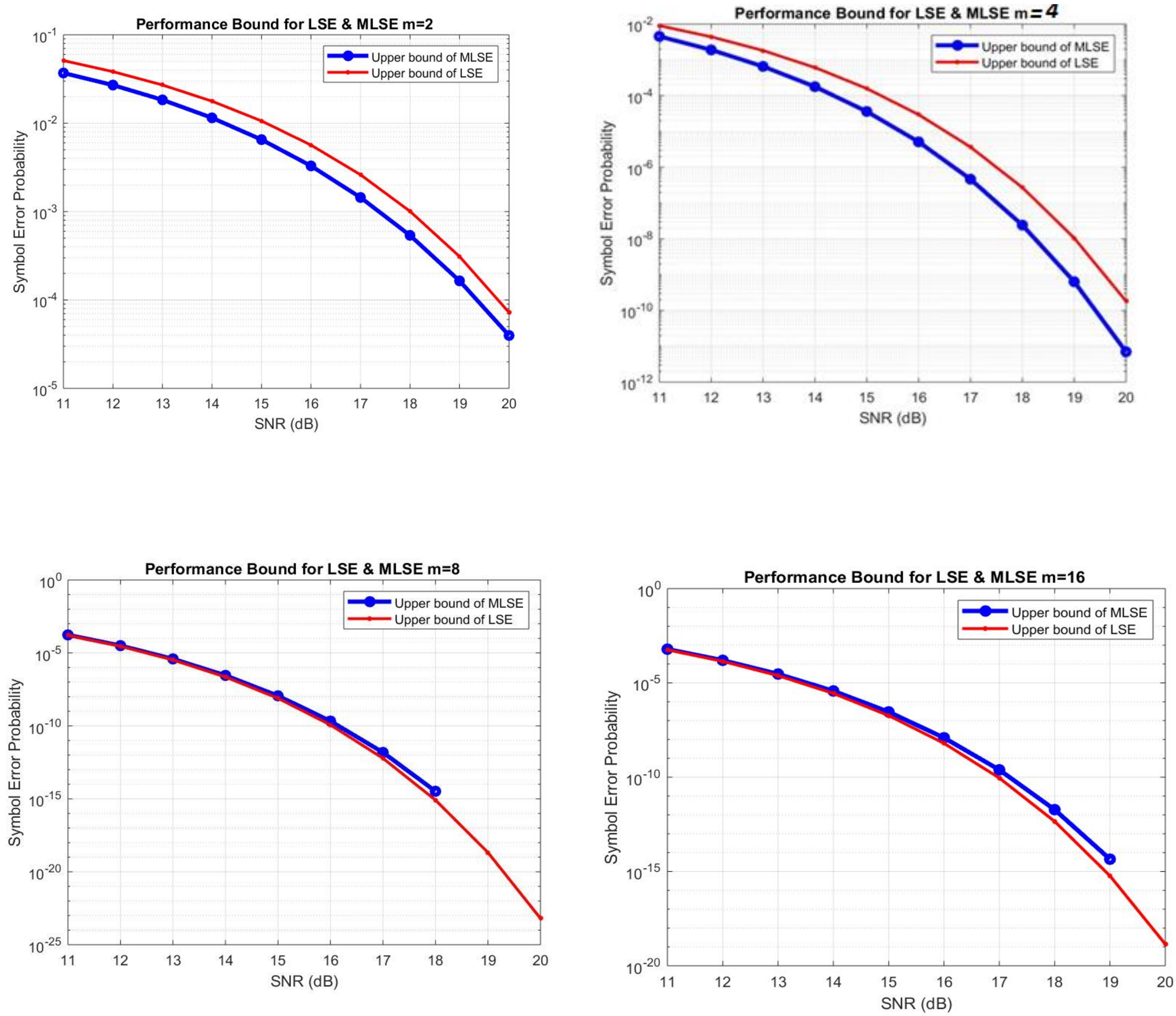

Fig 6: MATLAB simulation of Performance bound for LSE & MLSE for m={2,4,8,16}, $\delta = 10$, N= 1000

From the simulated fig.6 which resembles the objective figure of the literature, we can deduce that the upper bound of LSE is very tight for higher-order QAM-scheme. For $m \geq 8$ the upper-bound coincides for both LSE & MLSE. However, referring to other literature **[3-5]**, this trend might not agree with other channel types. For instance, a partial impulse response channel doesn't have a bounded Forney's upper bound limit F(m). Note that in the simulation the dependence of truncation depth $'\delta'$, and N-sample size were not examined in this project. The value of $'N = 100\ \&\ \delta = 10'$ was kept constant for later simulation as well. There are several ways to calculate the accuracy of the simulated figure, analytically. One of the common ways is to express the 'standard deviation (SD)' as a percentage of the estimated Symbol Error Probability.

Errors can also be due to symbols detected outside the input alphabet. As with the post-processing in the literature, the detector rounds off the output symbol into the nearest integer in the alphabet. For instance, if the detected symbol is $'m + 1'$ which is outside the range of input alphabet $A =$ {0,1………m-1}, then $'m - 1'$ is chosen to be the desired output. In fig. 7 we have a plot of SEP for an AWGN channel. It is evident that as $m - increases$, SEP drops sharply. MATLAB built-in BER-tool (fig.8) was used to obtain Monte-Carlo simulation for $m = 16$ of LSE (fig. 9) for complex $'h'$. A comparison has been made for the other channel type 'Rayleigh' & Rician for the same modulation order $m = 64$. We can conclude from the graph that the Rician channel is less subjected to interference than Rayleigh channel under similar channel constraints.

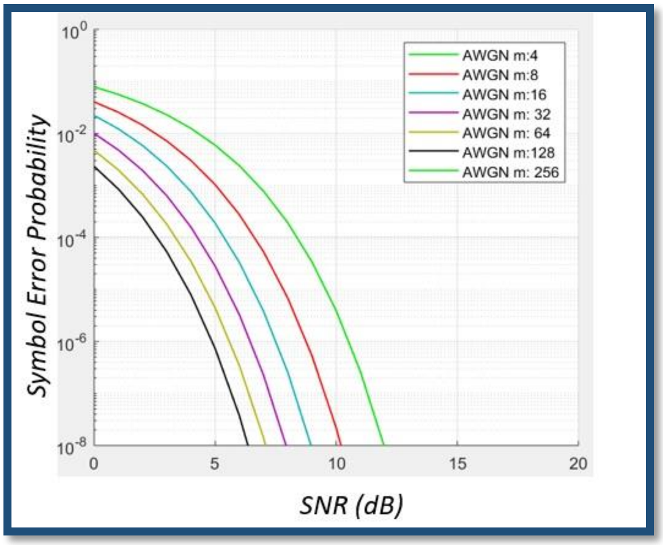

Fig 7: MATLAB simulation of SEP for AWGN Channel $'m' = \{4, 8, 16, 32, 64, 128, 256\}$

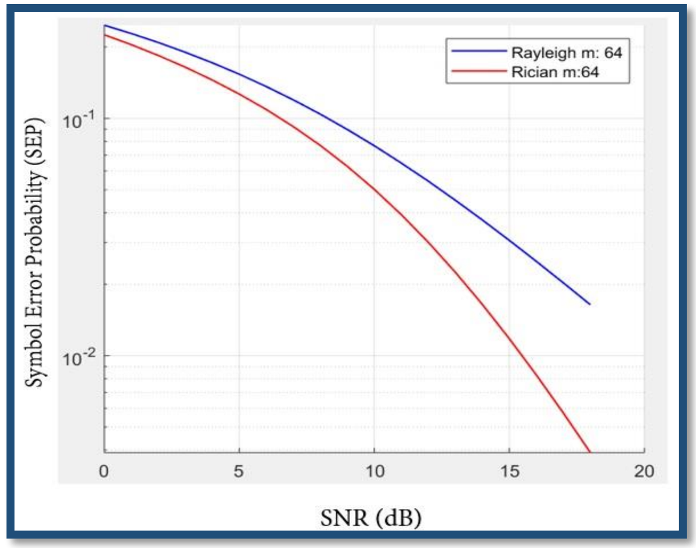

Fig 10: MATLAB simulation of Rayleigh & Rician Channel $\{'m = 64'\}$

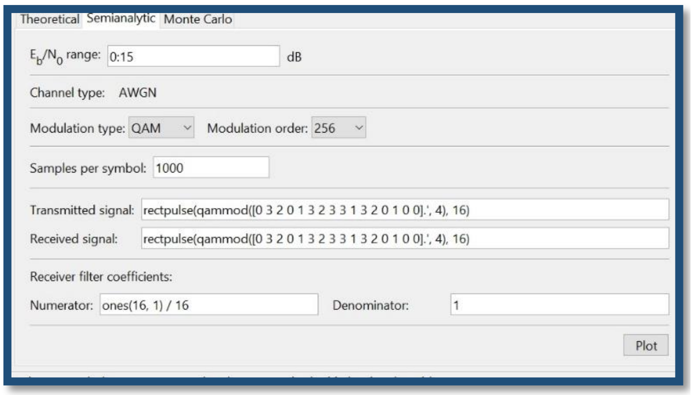

Fig 8: BER toolbox to obtain Monte Carlo Simulation as shown below

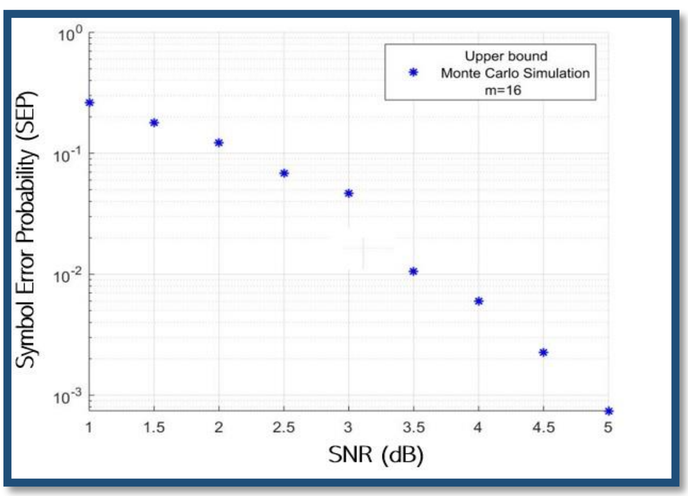

Fig 9: Monte Carlo Simulation of LSE- SEP Upper bound $\{'m = 16'\}$ with the complex channel response

VII. Conclusion

In this project, a tight upper-bound on the SEP of LSE was derived. The performance of LSE was contrasted with optimal MLSE in terms of SNR loss. The performance was proven to be optimal for a large signal level and was shown to be eventually ideal for a higher-level modulation scheme. The optimality might not follow for the class of channel in which Forney's upper bound doesn't converge. A slight improvisation was made experimenting AWGN channel with higher modulation order. A Monte-Carlo simulation was obtained in MATLAB, coded for complex $'h'$ & m =16. Finally, a comparison has been made between Rayleigh and Rician channel for the same modulation order and it was shown that the Rician channel was more robust to interference and had a better SEP response. The scope of this project could be extended for different channel impulse response and classify a class of channel for which Forney's upper bound F(m) exist. The effect of truncation depth δ on SEP could also be investigated.

References

[1] Forney, G. (1970). Convolutional codes I: Algebraic structure. *IEEE Transactions on Information Theory*, *16*(6), 720–738. doi: 10.1109/tit.1970.1054541

[2] Viterbi, A. (1973). Information theory in the sixties. IEEE Transactions on Information Theory, 19(3), 257–262. doi: 10.1109/tit.1973.1055000.

[3] Farhang-Boroujeny. (1995). Channel memory truncation for maximum likelihood sequence estimation. *Proceedings of GLOBECOM 95 GLOCOM-95*. doi: 10.1109/ctmc.1995.502954.

[4] Xiong, F. (n.d.). Sequential sequence estimation for ISI channels with convolutionally coded input sequence. *[Conference Record]*

SUPERCOMM/ICC 92 Discovering a New World of Communications. doi: 10.1109/icc.1992.267985.

[5] Fincke, U., & Pohst, M. (1985). Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, *44*(170), 463–463. doi: 10.1090/s0025-5718-1985-0777278-8

[6] Mow, W. H. (n.d.). Performance of the lattice sequence estimator. Proceedings of 1994 IEEE International Symposium on Information Theory. doi: 10.1109/isit.1994.394796

[7] Mow, W. H. (1994). Maximum likelihood sequence estimation from the lattice viewpoint. *IEEE Transactions on Information Theory*, *40*(5), 1591–1600. doi: 10.1109/18.333872

[8] Wesolowski, K. (1993). Adaptive Channel Estimation For Maximum Likelihood Sequence Estimation. Adaptive Systems in Control and Signal Processing 1992, 523–528. doi: 10.1016/b978-0-08-041717-2.50089-6

[9] Retrieved April 28, 2020, from https://www.mathworks.com/help/comm/ug/bit-error-rate-ber.html#a1058287176b1