Polar Codes:

A New Way Of Achieving Shannon Capacity

Prepared by Lab Group 6 - Group 1

Honor Code

We declare that

→ The work that we are presenting is our own work.

→ We have not copied the work (the code, the results, etc.) that someone else has done.

→ Concepts, understanding and insights we will be describing are our own.

→ We make this pledge truthfully. We know that violation of this solemn pledge can carry grave consequences.

Attested by:

DESAI ADITYA VEERAL
202201451

Jainam Ajaykumar Patel
202201514

Variya Hitanshu J.
202201510

Nandini Mandaviya
202201487

Anuj Valambhiya
202201481

Dev Vyas
202201453

Harsh Popatiya
202201463

Hardi Naik
202201477

Srushti Makwana
202201458

Heer Shah
202201525

Dhruvin Chaudhary
202101239

Group Members

Aditya Desai (202201451)

Dev Vyas (202201453)

Srushti Makwana (202201458)

Harsh Popatiya (202201463)

Hardi Naik (202201477)

Anuj Valambhiya (202201481)

Nandini Mandaviya (202201487)

Hitanshu Variya (202201510)

Jainam Patel (202201514)

Heer Shah (202201525)

Dhruvin Chaudhary (202101239)

Table of Contents

Introduction

Polar Codes were discovered by Erdal Arikan in 2008. They have become the most favourable codes for Forward Error Correction since then. They are notable for being first in their types with provable capacity-achieving property. Polar codes have especially found their application in the 5G wireless communication.

An Analogy to Understand Polar Codes

Why Polar Codes?

Polar codes propose a new approach to achieve the Shannon capacity. Instead of encoding data bits like traditional methods, polar codes polarize the channel into two extreme channels: highly reliable and less reliable. They then utilize the highly reliable channels to transfer the data bits, unlike different pre-existing coding techniques that focus on encoding the data bits. Thus, Polar Codes provide ease by working directly over channels rather than the data bits.

Analyzing the Differences Between Polar Codes and LDPC Codes

Factors:

1. **Shannon capacity:**
- LDPC: Operates near the Shannon limit for high reliability.
- Polar: Can match Shannon capacity in ideal scenarios.

2. **BER**:
- Polar: Low BER for short messages.
- LDPC: Superior BER for long messages.

3. **Throughput**:
- LDPC: High throughput, ideal for large data volumes.
- Polar: Potential throughput limitations for extensive data streams.

4. **Latency**:
- Polar: Lower latency, suitable for real-time applications.
- LDPC: Slightly higher latency, especially for time-sensitive tasks.

Polar Transformation

Let X be a Bernoulli Random Variable $X \sim p_x$ on $x = \{0, 1\}$
Let X_1, X_2, ..., X_n be iid $\sim X$
For $N = 2^n$, $n \geq 1$

$U \rightarrow$ a new transformed Random vector $U^N = X^N F^{\otimes n}$, F = $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

Where $U^N = (U_1 U_2 ...... U_N)$
$X^N = (X_1 X_2 ...... X_N)$

For example,
Let's consider $n = 1 \implies N = 2^1 = 2$
So, $U^2 = (U_1, U_2)$ and $X^2 = (X_1, X_2)$

$$(U_1, U_2) = (X_1, X_2) \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes 1}$$

$$(U_1, U_2) = (X_1 + X_2, X_2)$$

For $n = 2,\ N = 2^2 = 4$

$$
\begin{aligned}
(U_1\ \ U_2\ \ U_3\ \ U_4) &= (X_1X_2X_3X_4)F^{\otimes 2} \\
&= (X_1X_2X_3X_4)F^{\otimes 1} \otimes F \\
&= (X_1X_2X_3X_4)\begin{bmatrix} F^{\otimes 1} & [0]_{2\times 2} \\ F^{\otimes 1} & F^{\otimes 1} \end{bmatrix} \\
&= (X_1X_2X_3X_4)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

$$(U_1\ \ U_2\ \ U_3\ \ U_4) = (X_1 + X_2 + X_3 + X_4\ \ X_2 + X_4\ \ X_3 + X_4\ \ X_4)$$

For a general n, we define kronecker product as

$$F^{\otimes n} = F^{\otimes(n-1)} \otimes F$$

$$F^{\otimes n} = \begin{bmatrix} F^{\otimes(n-1)} & O \\ F^{\otimes(n-1)} & F^{\otimes(n-1)} \end{bmatrix}, \text{ where } O = [0]_{n\times n}$$

Theorem of Polarization

How Polar Codes work?

For Symmetric Channels, the capacity is given by:

$$I(W) \triangleq I(X;Y)$$

with $X \sim \text{unif}\{0,1\}$.
Using base 2 algorithms:

$$0 \leq I(W) \leq 1$$

Perfect Channel: $I(W) = 1$, Useless Channel: $I(W) = 0$

Erikan's polar transformation converts ordinary channels to such extreme channels by first combining them and then splitting them:

$$X_1 = U_1 \oplus U_2$$

$$X_2 = U_2$$

If U_1 and U_2 are distributed uniformly over $\{0, 1\}$, then X_1 and X_2 are also independent and uniformly distributed over $\{0, 1\}$.

$$I(X_1X_2; Y_1Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = 2I(W)$$

$$I(U_1U_2; Y_1Y_2) = I(X_1X_2; Y_1Y_2) = 2I(W)$$

$$2I(W) = I(U_1U_2; Y_1Y_2)$$

$$2I(W) = I(U_1; Y_1Y_2) + I(U_2; Y_1Y_2|U_1)$$ (Using the chain rule)

$$2I(W) = I(U_1; Y_1Y_2) + I(U_2; Y_1Y_2U_1)$$

$$2I(W) = I(W^-) + I(W^+)$$

So,

$$I(W^+) = I(U_2; Y_1Y_2|U_1) \geq I(U_2; Y_2) = I(X_2; Y_2) = I(W)$$

$$2I(W) = I(W^-) + I(W^+)$$

Therefore,

$$I(W^-) \leq I(W) \leq I(W^+)$$

Let us take a look at a binary erasure channel defined as BEC(e).
Consider W^-

$$U_1 \to (Y_1, Y_2)$$

Input is U_1 and output is given as : (Y_1, Y_2)

$$(Y_1, Y_2) = (U_1 \oplus U_2, U_2) \text{ with } p = (1-e)^2$$

$$(Y_1, Y_2) = (?, U_2) \text{ with } p = (1-e)e$$

$$(Y_1, Y_2) = (U_1 \oplus U_2, ?) \text{ with } p = (1-e)e$$

$$(Y_1, Y_2) = (?, ?) \text{ with } p = e^2$$

Erasure probability $= 1 - (1-e)^2 = 2e - e^2$.

Consider W^+

$$U_2 \to (Y_1, Y_2, U_1)$$

Input is U_2 and output is given as : (Y_1, Y_2, U_1)

$(Y_1, Y_2, U_1) = (U_1 \oplus U_2, U_2, U1)$ with $p = (1-e)^2$

$(Y_1, Y_2, U_1) = (?, U_2, U1)$ with $p = (1-e)e$

$(Y_1, Y_2, U_1) = (U_1 \oplus U_2, ?, U1)$ with $p = (1-e)e$

$(Y_1, Y_2, U_1) = (?, ?, U1)$ with $p = e^2$

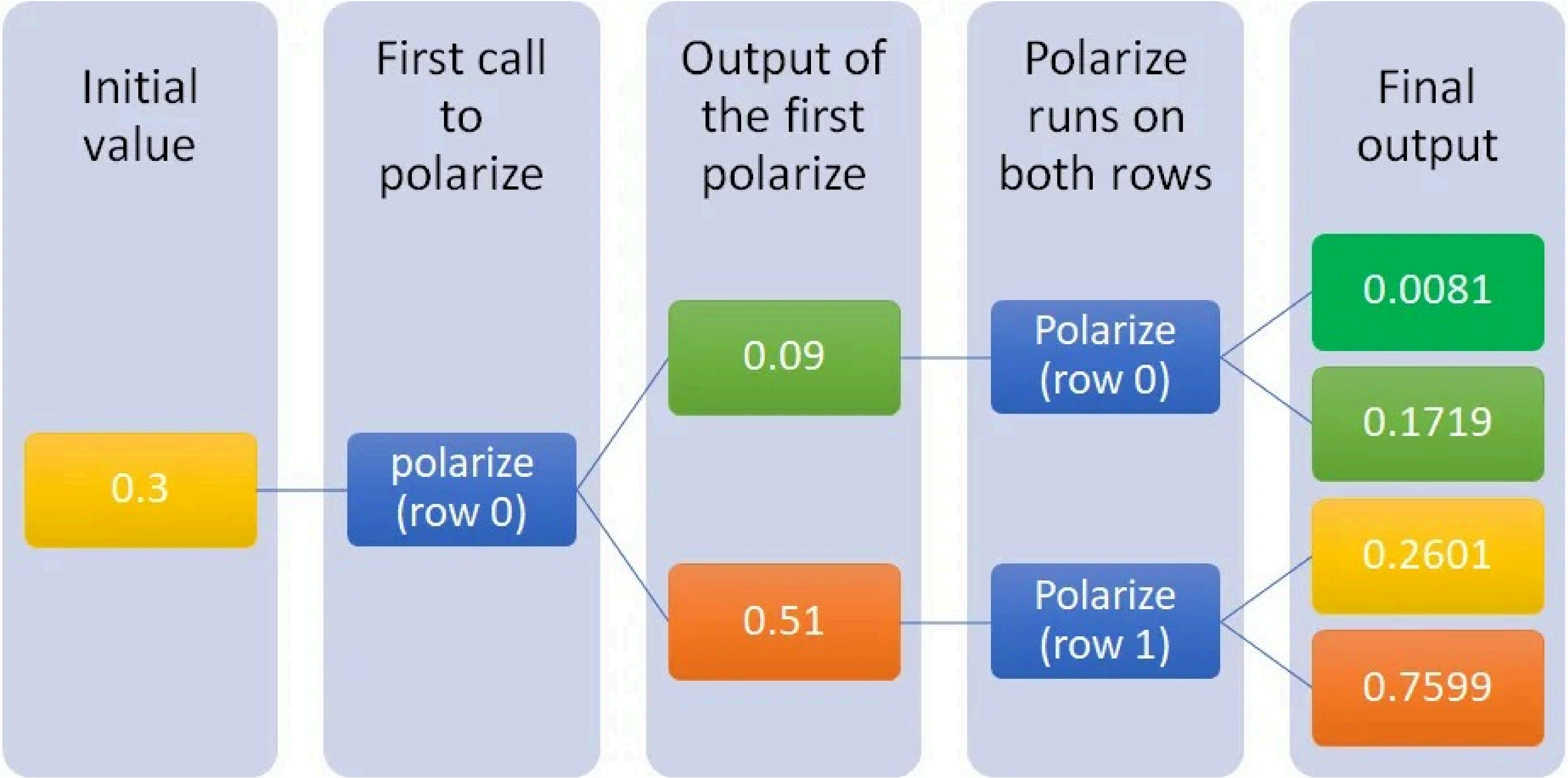
Erasure probability $= e^2$.

The channel is split into W^+ and W^-.

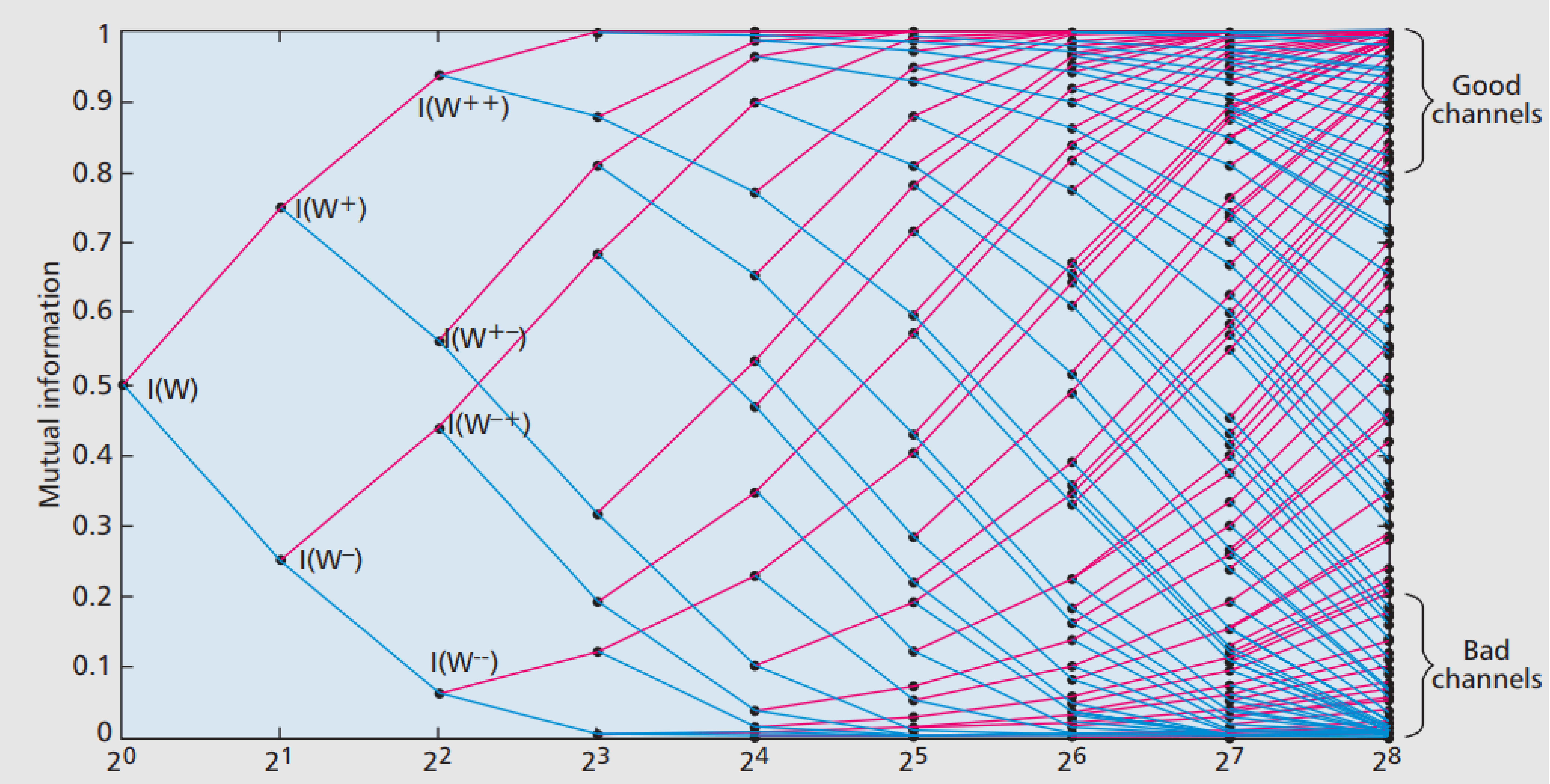
The channel W^+ has an erasure probability(e^+) of e^2. The channel W^- has an erasure probability(e^-) of $2e - e^2$.

Now for Binary Symmetric channel with error probability p the same equation will hold, only the erasure probability will be replaced by the error probability. Further notice that value of $e^+ + e^- = 2e$.
This holds from

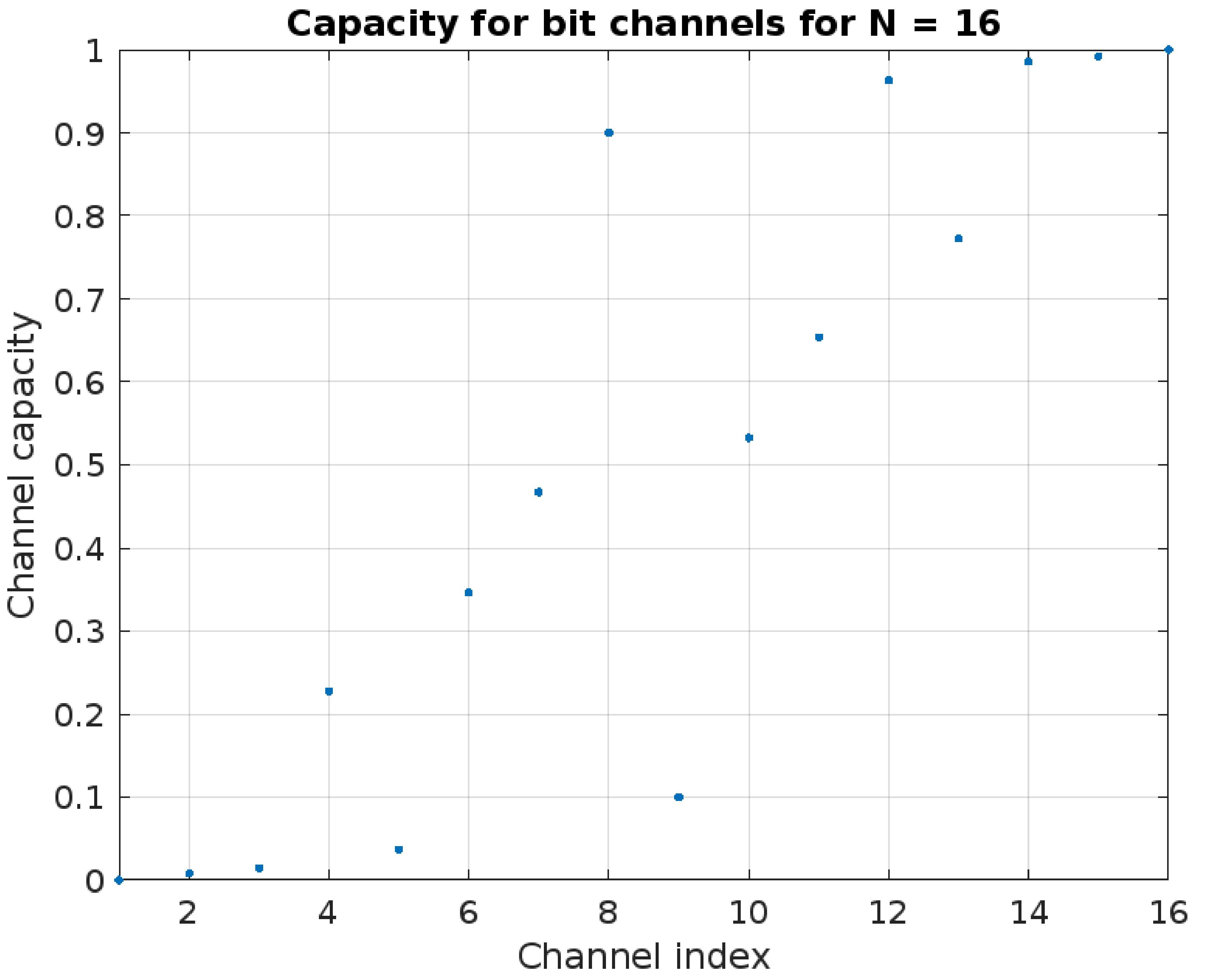
$$2I(W) = I(W^-) + I(W^+)$$


Initial value
First call to polarize
Output of the first polarize
Polarize runs on both rows
Final output
0.3
polarize (row 0)
0.09
0.51
Polarize (row 0)
Polarize (row 1)
0.0081
0.1719
0.2601
0.7599

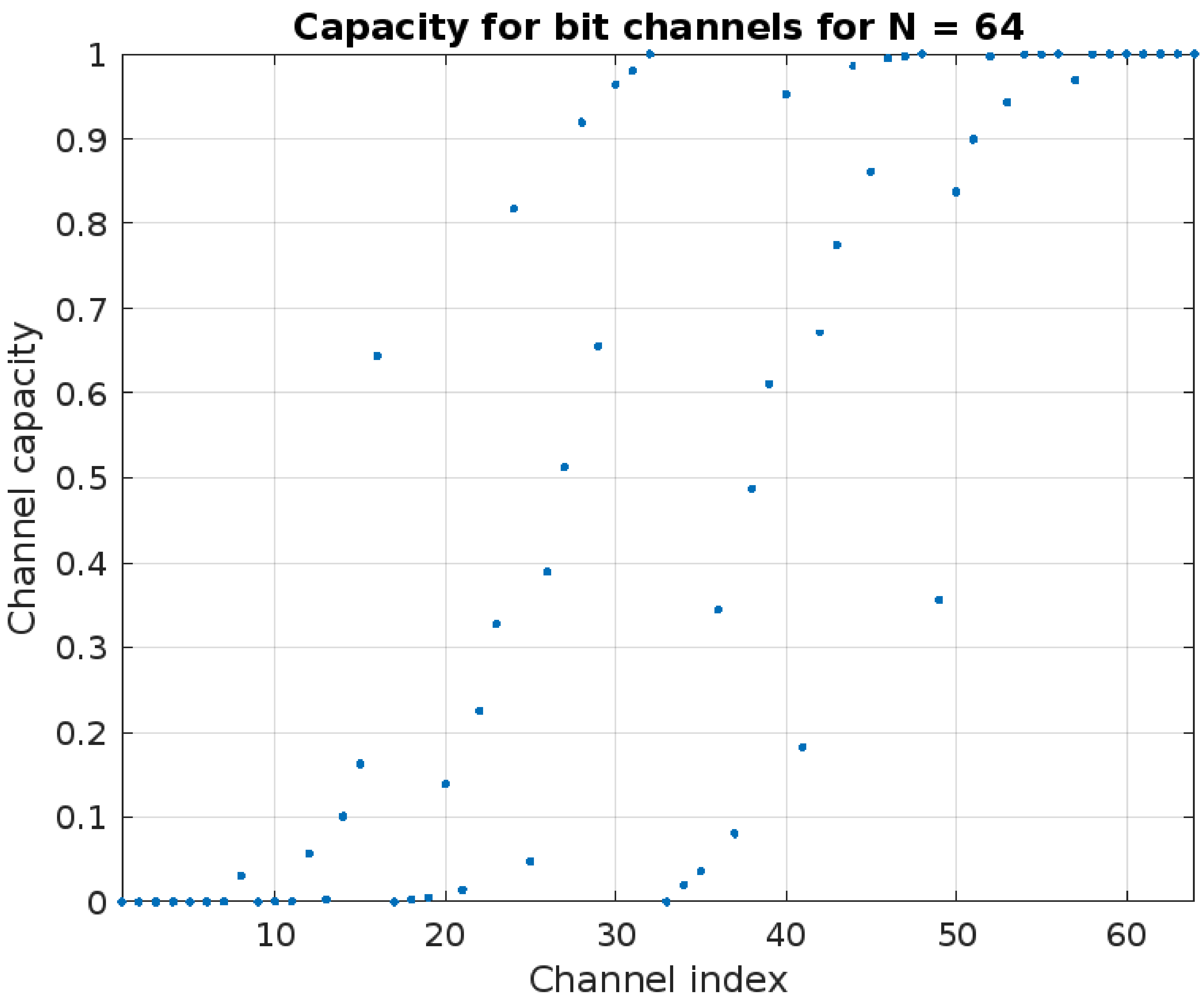
Polarization Martingale



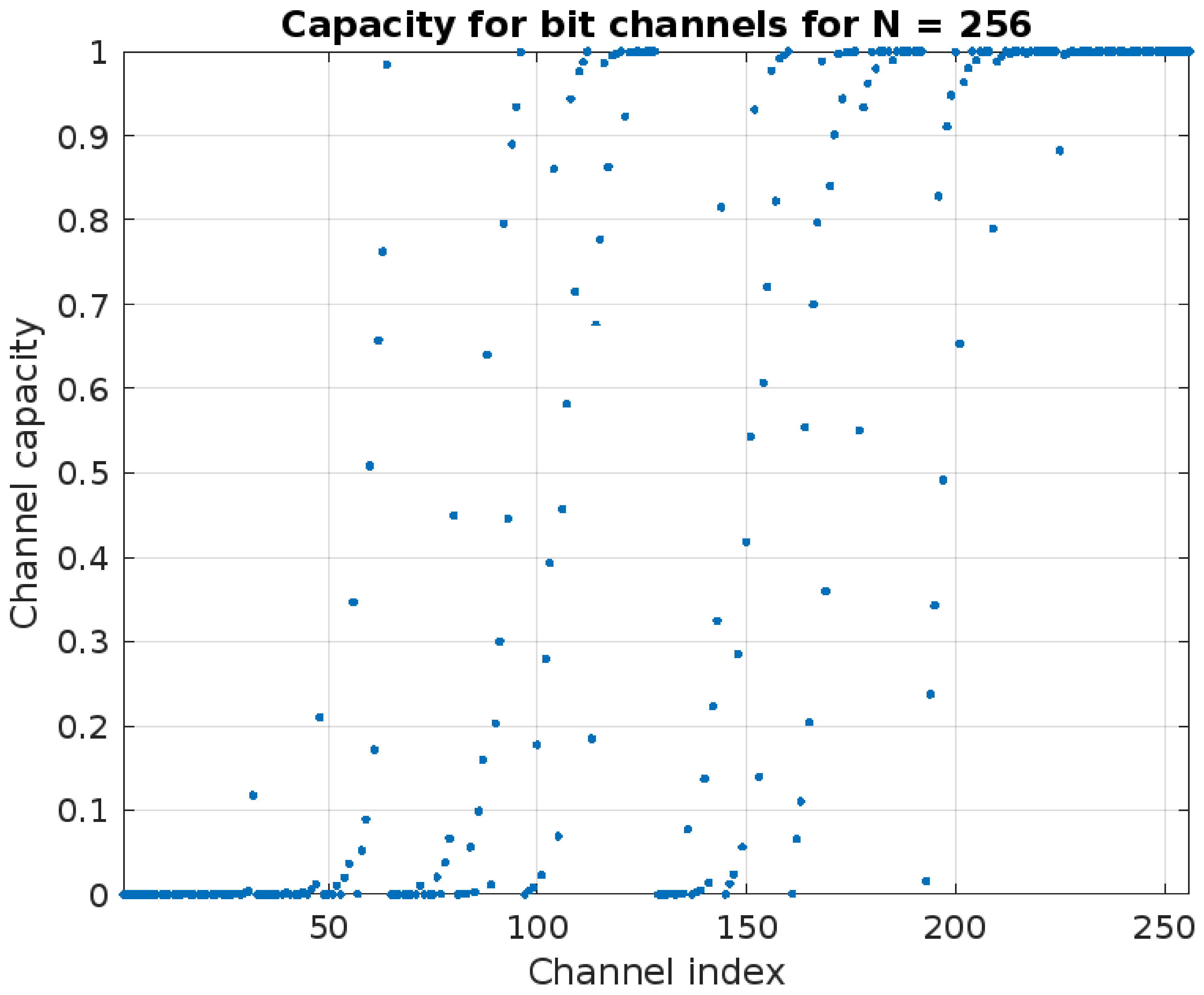
Polarization of BEC(1/2) for N=16



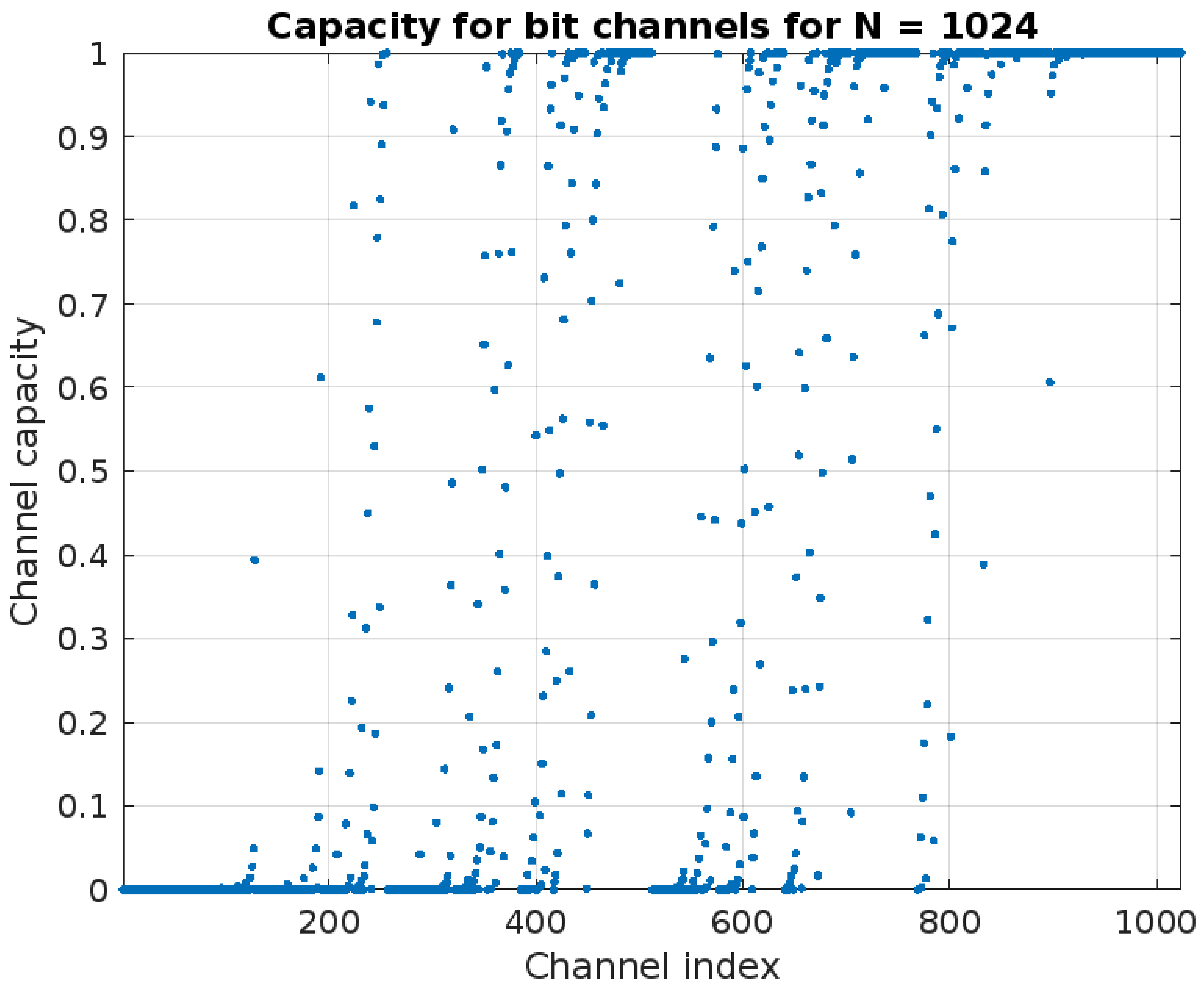
Polarization of BEC Rate (1/2) N=64

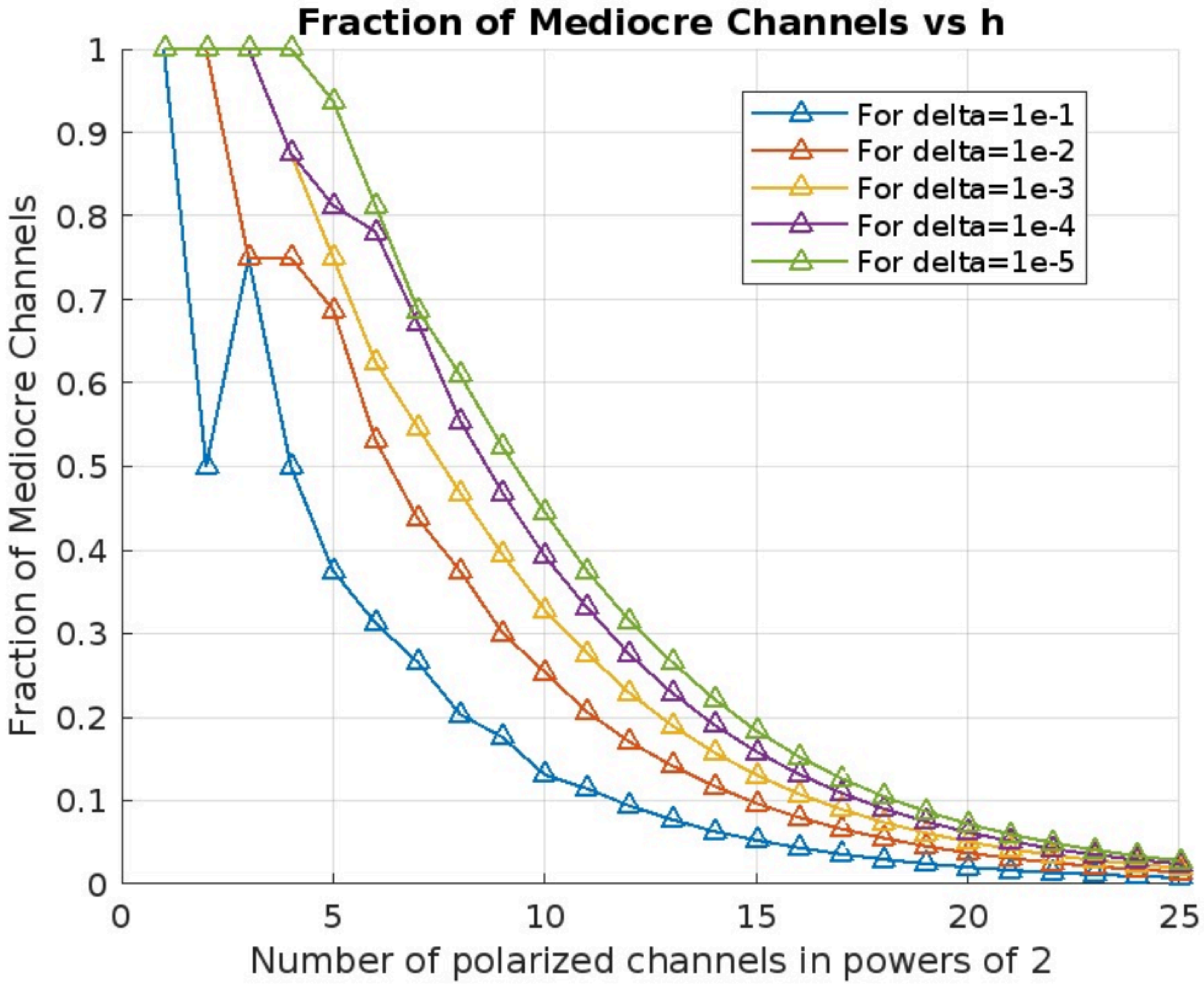


Polarization of BEC Rate (1/2) N=256



Polarization of BEC(1/2) for N=1024




Fraction of Mediocre Channels vs h
Fraction of Mediocre Channels
Number of polarized channels in powers of 2
For delta=1e-1
For delta=1e-2
For delta=1e-3
For delta=1e-4
For delta=1e-5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

Observations :

- In polar codes, as the value of N (representing the number of channels) increases, the number of channels approaching capacity extremes (0 or 1) also increases.
- We observe a greater quantity of extreme channels compared to mediocre ones.
- Polar codes strategically allocate frozen bits to channels with poor performance (the "worst" channels) and reserve efficient channels for transmitting message bits.
- This allocation minimizes error probability by leveraging channel characteristics.
- As N tends towards infinity, the error probability diminishes significantly, converging towards Shannon capacity.
- This demonstrates the remarkable efficiency and effectiveness of polar codes in maximizing transmission reliability.

Encoding Algorithm

Reliability Sequence:

- It is a sequence which gives a sorted order of channels starting from worst channels to the best channels according to the efficiency of the output.
- For **N=16** the reliability sequence is:

  1 2 3 5 9 4 6 10 7 11 13 8 12 14 15 16
- The reliability sequence is derived using the **Bhattacharya Parameter**, which is defined as below:

$Z(W) \triangleq \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}$, where $Z(W)$ takes values in $[0, 1]$ and W is a B-DMC (Binary Discrete Memoryless Channel).

Suppose we have a message **m** having K bits. We need to generate a codeword **C** having N ($N=2^n$) bits.

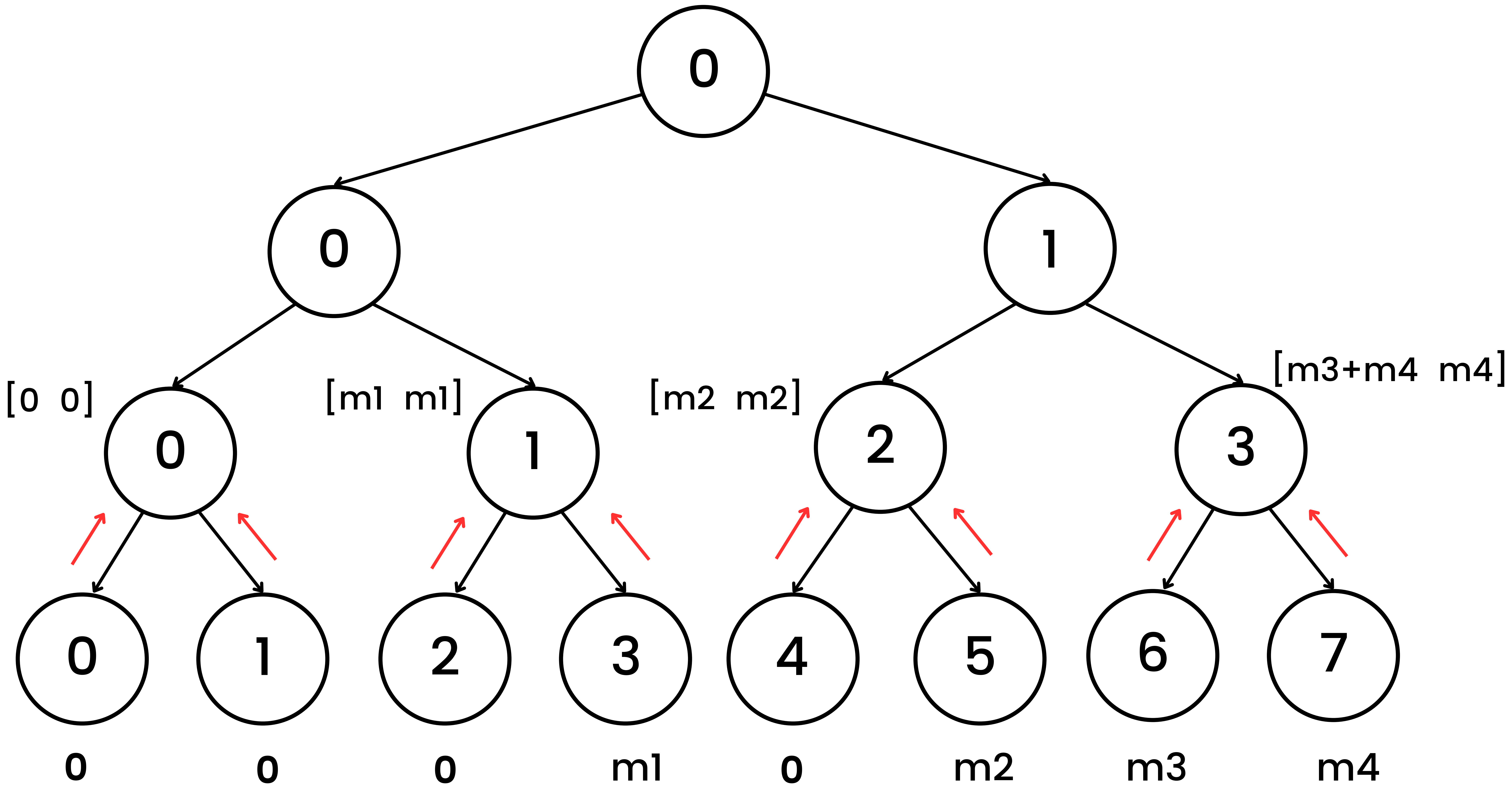
Algorithm:

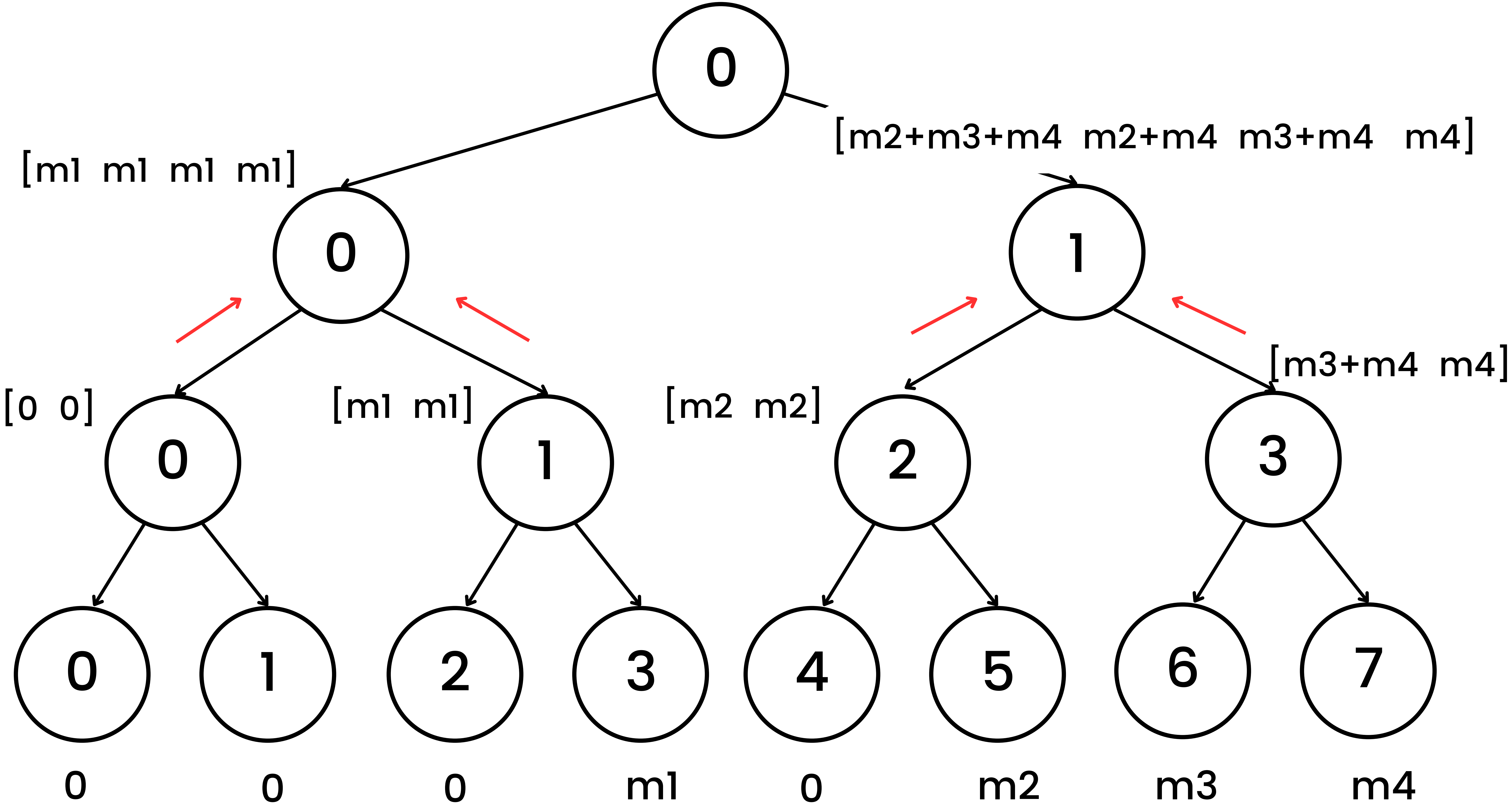
1. Find first N-K worst channels for N based on the reliability sequence.
2. For all the N-K positioned indexes in the vector **u** we set the bits to zero. These positions are also known as **Frozen Positions.**
3. Set the remaining K bits of vector **u** as the message bits.
4. Now, to generate the final codeword or the encoded message we need to multiply vector **u** with the Generator Matrix $\mathbf{G_n}$ based on the polar transform defined earlier.

$$\text{Codeword } C = u.G_n$$

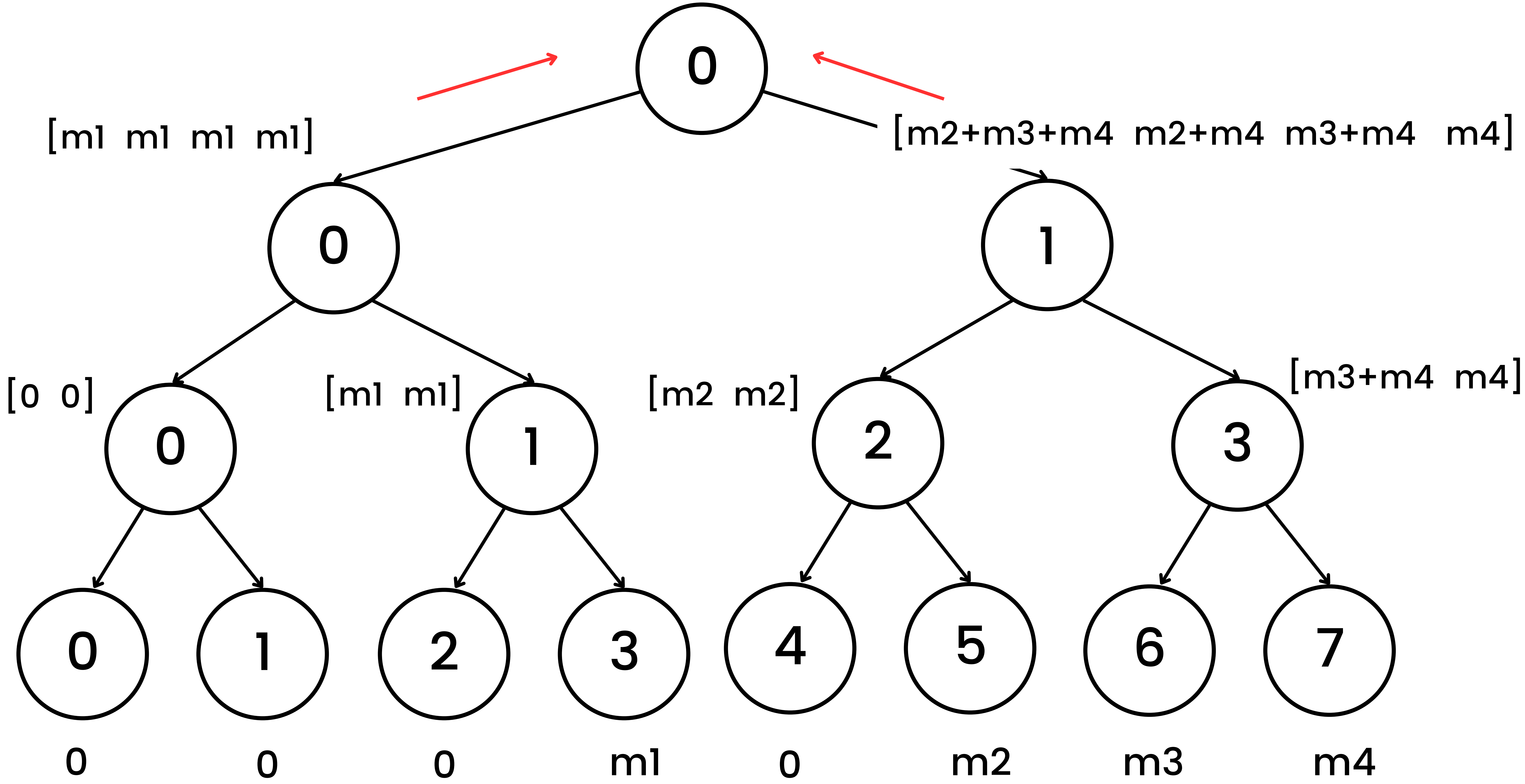
Example:

- Message m: 4 Bits (K=4)
- Codeword C: 8 Bits ($N=2^3$)
- Reliability Sequence: 1 2 3 5 4 6 7 8

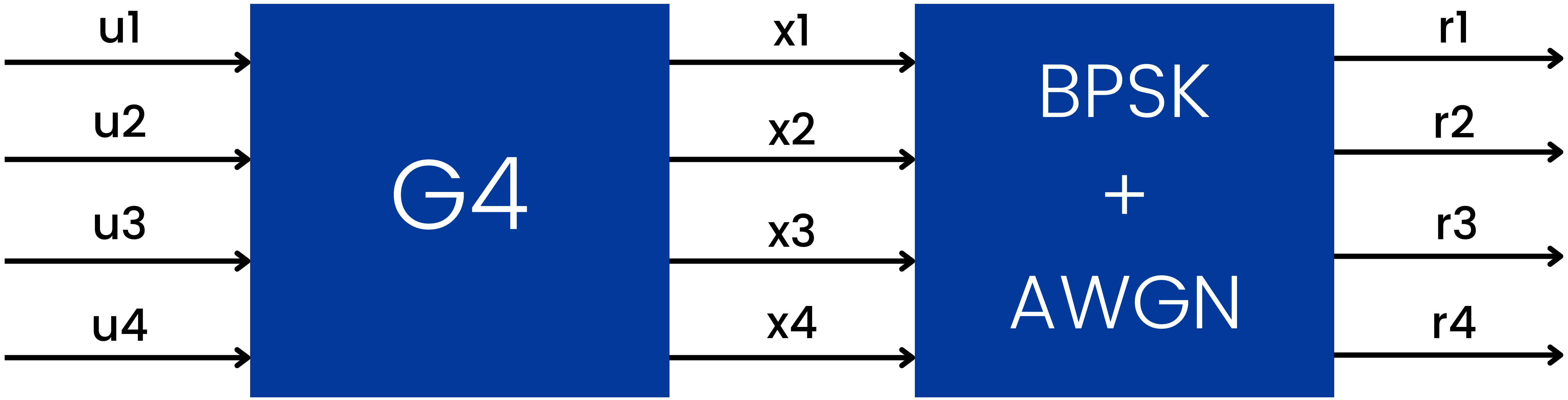

0
0
1
[0 0]
[m1 m1]
[m2 m2]
[m3+m4 m4]
0
1
2
3
0
1
2
3
4
5
6
7
0
0
0
m1
0
m2
m3
m4


0
[m1 m1 m1 m1]
[m2+m3+m4 m2+m4 m3+m4 m4]
0
1
[0 0]
[m1 m1]
[m2 m2]
[m3+m4 m4]
0
1
2
3
0
1
2
3
4
5
6
7
0
0
0
m1
0
m2
m3
m4

C=[m1+m2+m3+m4 m1+m2+m4 m1+m3+m4 m1+m4 m2+m3+m4 m2+m4 m3+m4 m4]



AWGN and BPSK



For a BPSK Channel the bit conversion scheme is

1 is converted to -1

0 is converted to 1

AWGN is added during transmission

Decoding Algorithm

Successive Cancellation Decoding

Basic Building Block N=2 SC Decoder

There are two main functions we require for Successive Cancelation Decoder. They can also be described as two types of SISO decoders used in different parts of the SC decoding algorithm. They are as described below:

SISO Decode for u_1 (SPC)

$$L(u_1) = f(r_1, r_2) = sgn(r_1)sgn(r_2)min(|r_1|, |r_2|)$$

$$\hat{u}_1 = \begin{cases} 0, & \text{if } L(u_1) \geq 0. \\ 1, & \text{if } L(u_1) < 0. \end{cases}$$

Given $\hat{u}_1$, Decode u_2 (Rep)

If $\hat{u}_1 = 0, \;\; L(u_2) = r_2 + r_1 \;\; (x = [u_2 \;\; u_2])$

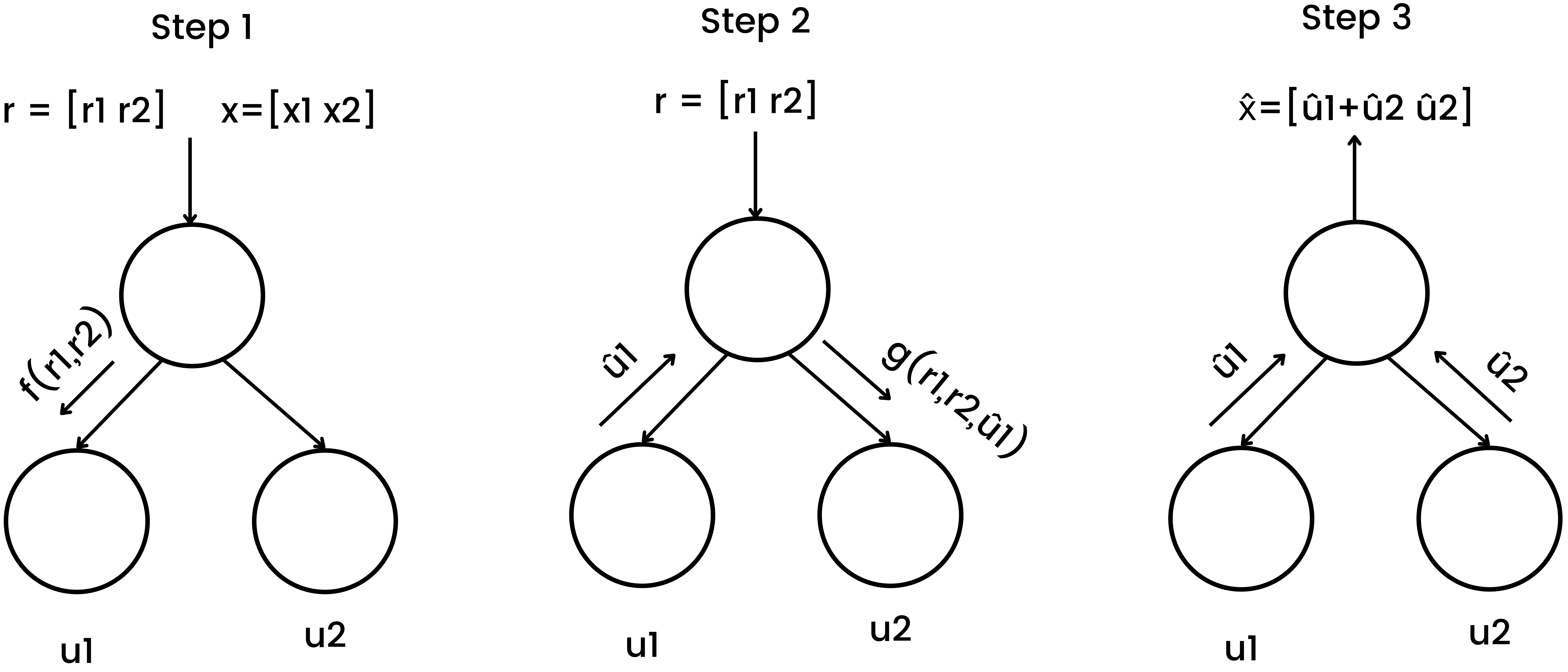
If $\hat{u}_1 = 1, \;\; L(u_2) = r_2 - r_1 \;\; (x = [\bar{u}_2 \;\; u_2])$

Both the cases together can be written as :

$L(u_2) = g(r_1, r_2, \hat{u}_1) = r_2 + (1 - 2\hat{u}_1)r_1$

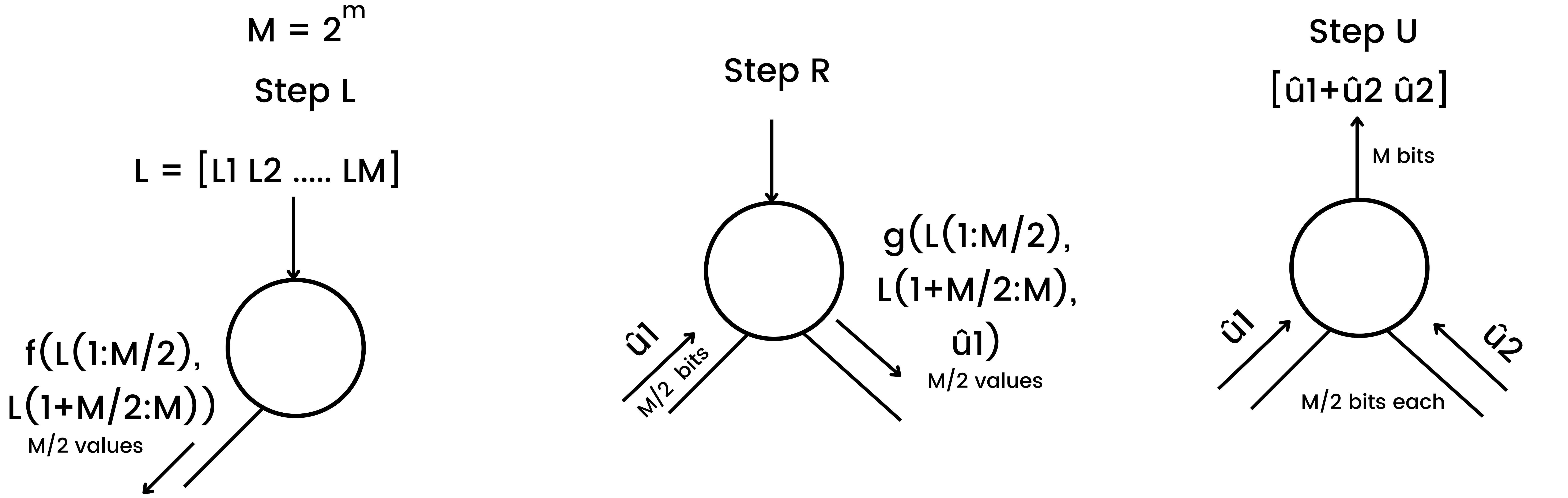
Proof of SISO SPC decoder and SISO Repetition decoder in Appendix

Successive Cancellation Decoding(N=2)



We repeat the above three steps till we traverse the whole tree (For N>2)

Successive Cancellation Decoding(for any N)



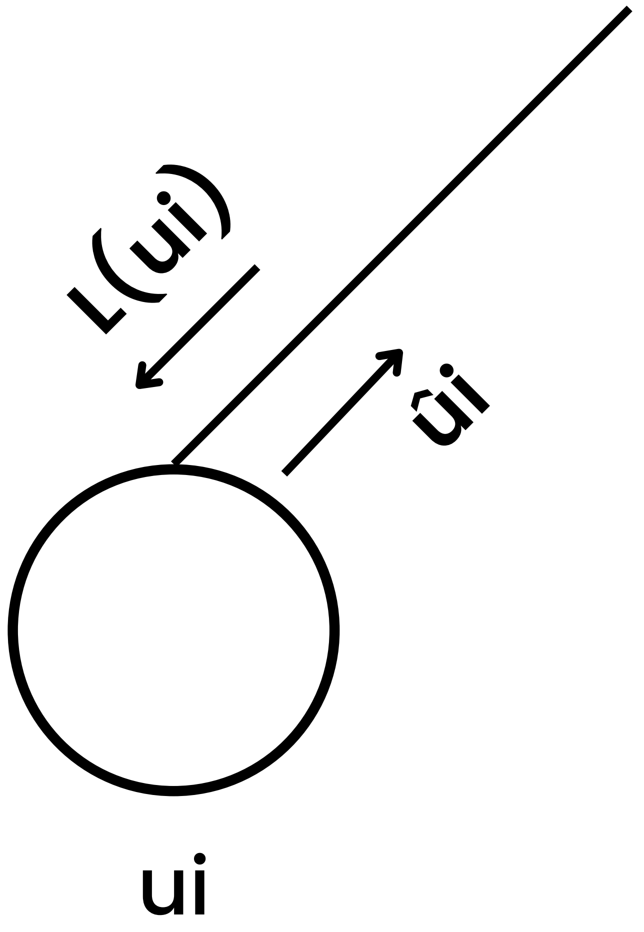
f(a(1:p),b(1:p))=[f(a1,b1)f(a2,b2) ...f(ap,bp)]

g(a(1:p),b(1:p),c)=[g(a1,b1,c1)g(a2,b2,c2) ...g(ap,bp,cp)]

Here f and g functions are the same as defined earlier

We repeat the above three steps till we traverse the whole tree (For N>2)

Successive Cancellation Decoding(for any N)



If 'i' is a Frozen Position: $\hat{u}_i=0$

If 'i' is a message Position: $\hat{u}_i=0$, if $L(u_i) \geq 0$

$\hat{u}_i=1$, if $L(u_i) < 0$

We repeat the above step for all the leaf nodes till we traverse the whole tree (For N>2)

Successive Cancellation Decoding

Algorithm:

Start at the root.

For intermediate nodes

1. Do step L and go to the left child.

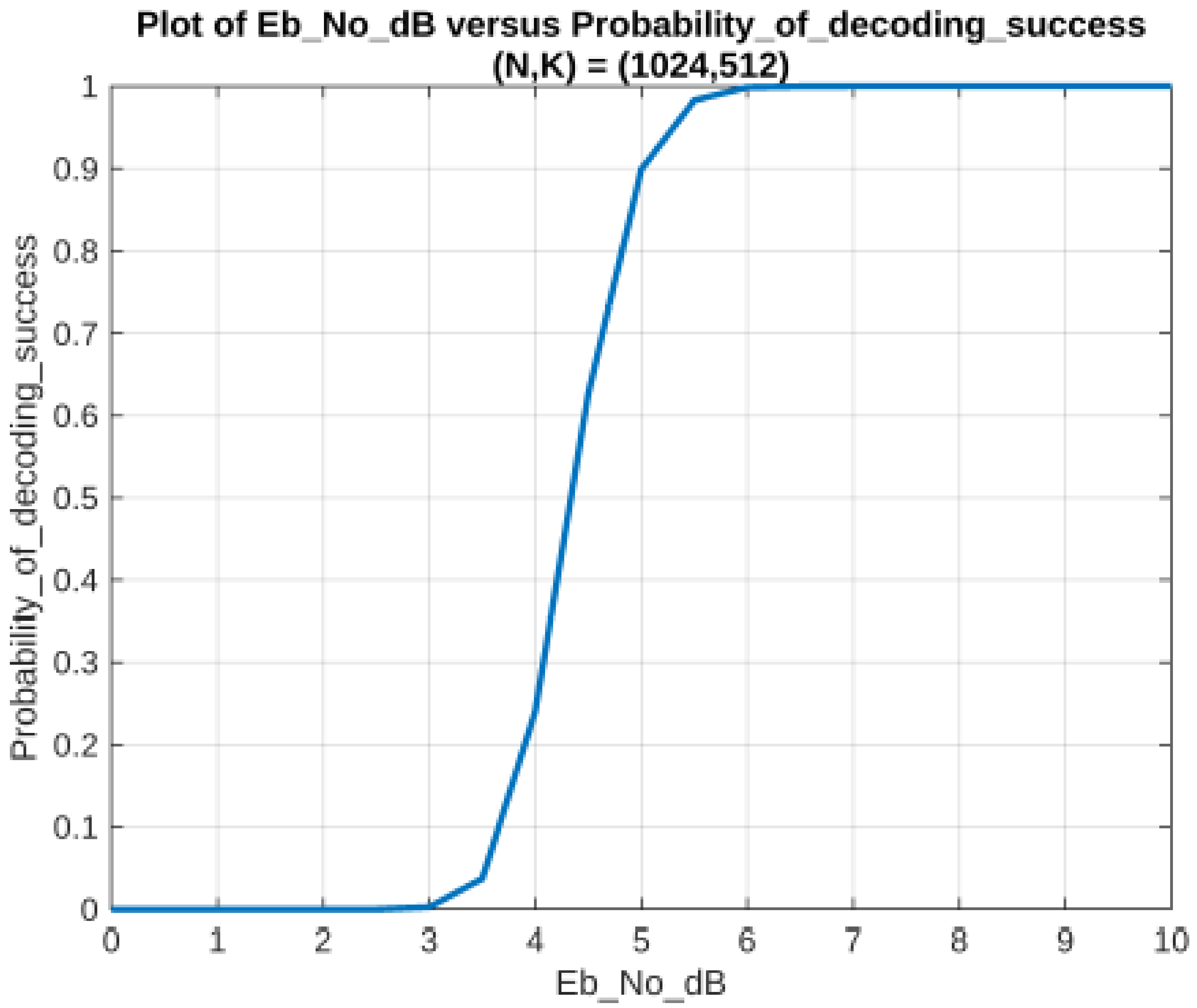
2. Once the decision ($\hat{u}1$) is recieved from the left child, do step R and go to the right child

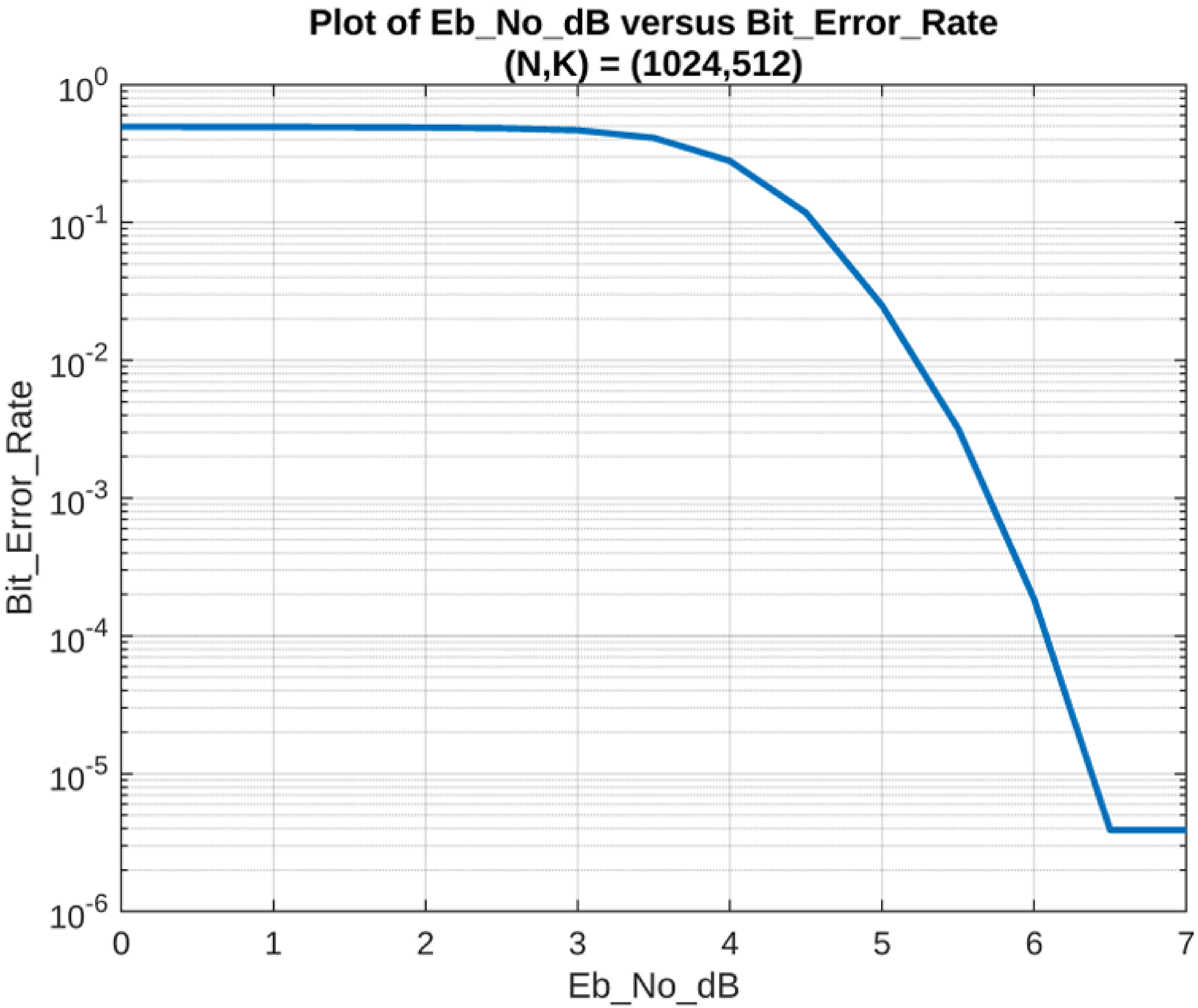
3. When decision is received from the right child ($\hat{u}2$), do step U and go to the parent.

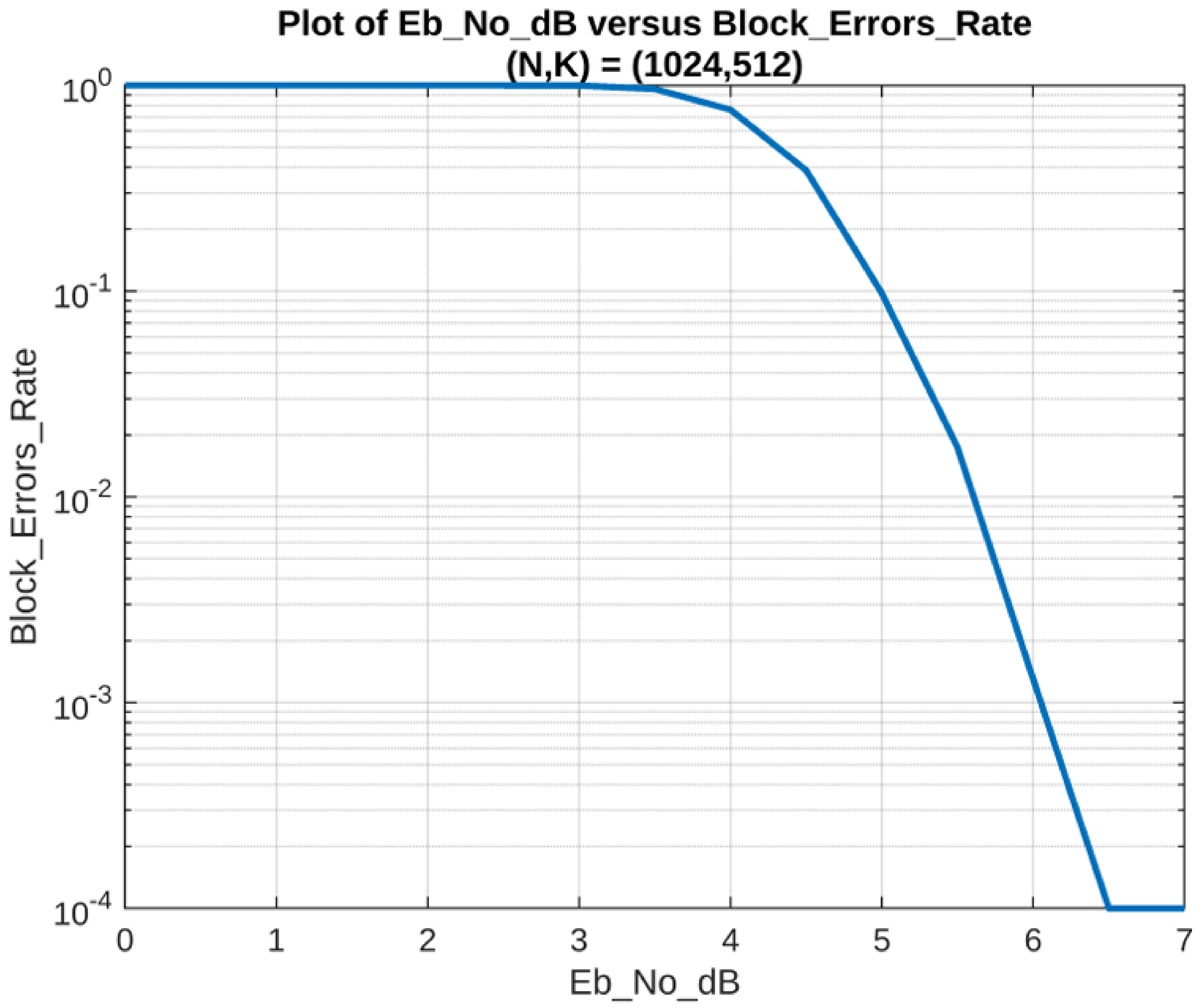
For Leaf nodes

Leaf must make a decision and send it to the parent

Simulating Outputs


Plot of Eb_No_dB versus Probability_of_decoding_success
(N,K) = (1024,512)
Probability_of_decoding_success
Eb_No_dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7
8
9
10

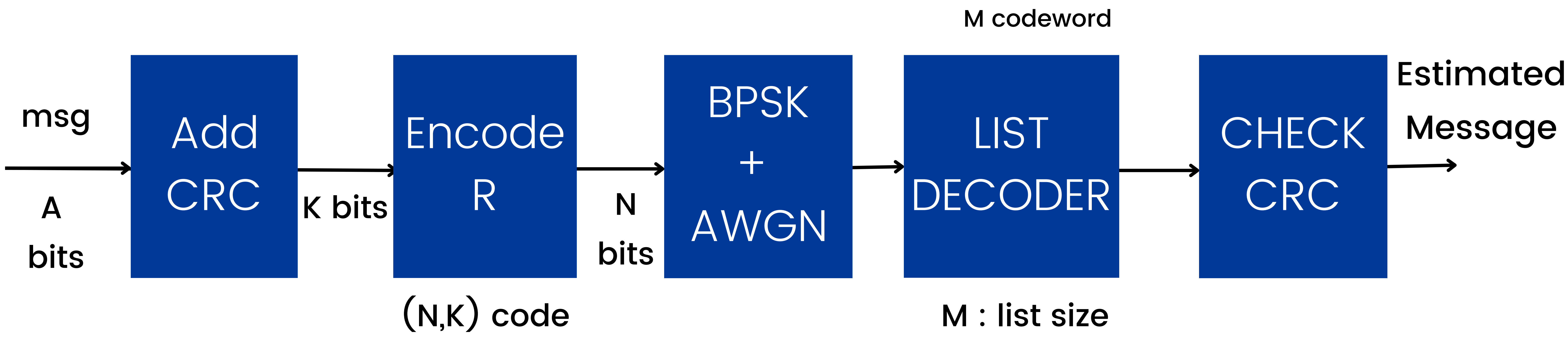

Plot of Eb_No_dB versus Bit_Error_Rate
(N,K) = (1024,512)
Bit_Error_Rate
Eb_No_dB
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
0
1
2
3
4
5
6
7


Plot of Eb_No_dB versus Block_Errors_Rate
(N,K) = (1024,512)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block_Errors_Rate
0
1
2
3
4
5
6
7
Eb_No_dB

Successive Cancellation Decoding : Disadvantages

- SC requires the decoding process to advance bit by bit. This results in high latency and low throughput when implemented in hardware.
- Polar codes decoded with SC only achieve the channel capacity when the code length tends toward infinity. For practical polar codes of moderate length, SC decoding cannot provide a reasonable error-correction performance.
- SCL Decoder is about 1dB better than SC decoder.

Successive Cancellation List : Decoding Block Diagram



Cyclic Redundancy Check (CRC)

- CRC is a method to detect error in digital networks. It is used to change accidental changes in data.
- In this method, we divide the message by a selected polynomial to get a remainder polynomial which we then append it to the end of message to form our new codeword which we then transmit.
- On the receiver side, after decoding, we divide decoded codeword by the selected CRC polynomial.
- If the remainder on division comes out to be zero then it is a valid codeword.

Decision Metric

The decision metric can be described as follows:

If $L(u_i) \geq 0$, then:

- $\hat{u}_i = 0$ has DM $= 0$,
- $\hat{u}_i = 1$ has DM $= |L(u_i)|$.

If $L(u_i) < 0$, then:

- $\hat{u}_i = 1$ has DM $= 0$,
- $\hat{u}_i = 0$ has DM $= |L(u_i)|$.

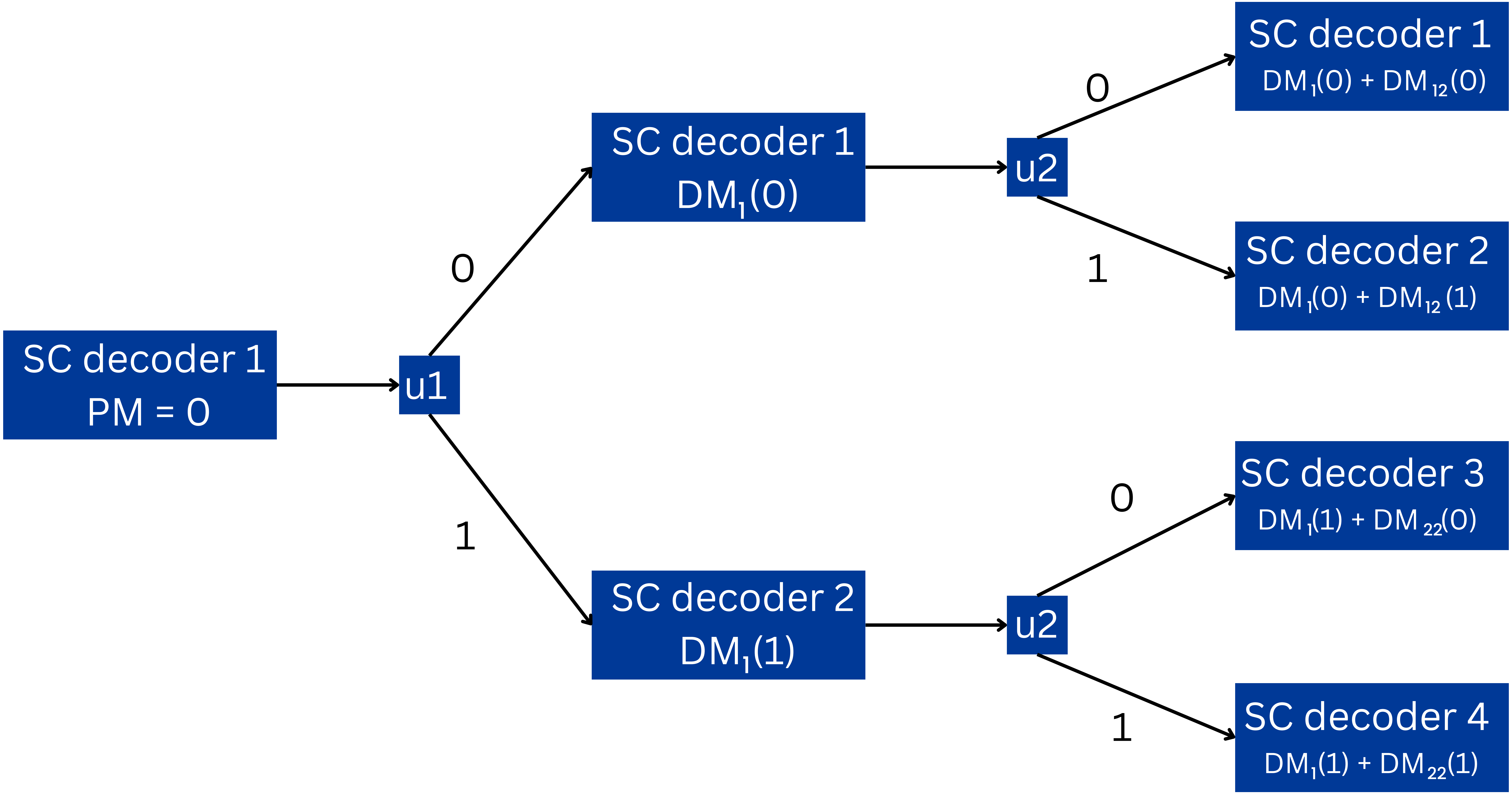
Note: If u_i is a frozen bit and $L(u_i) < 0$, then DM is also assigned.

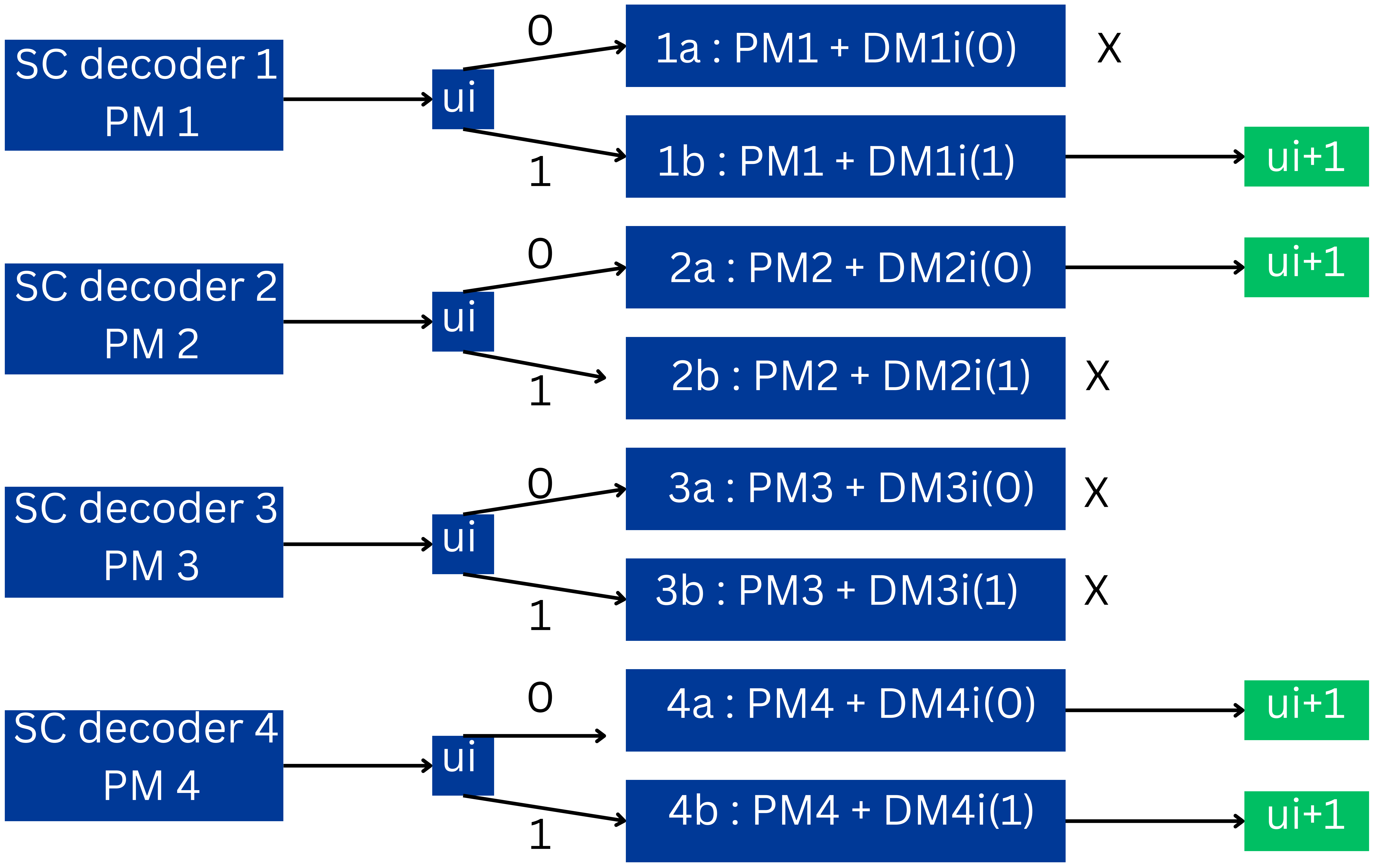
Path metric:

Sum of decision metrics for every path is called Path metric.

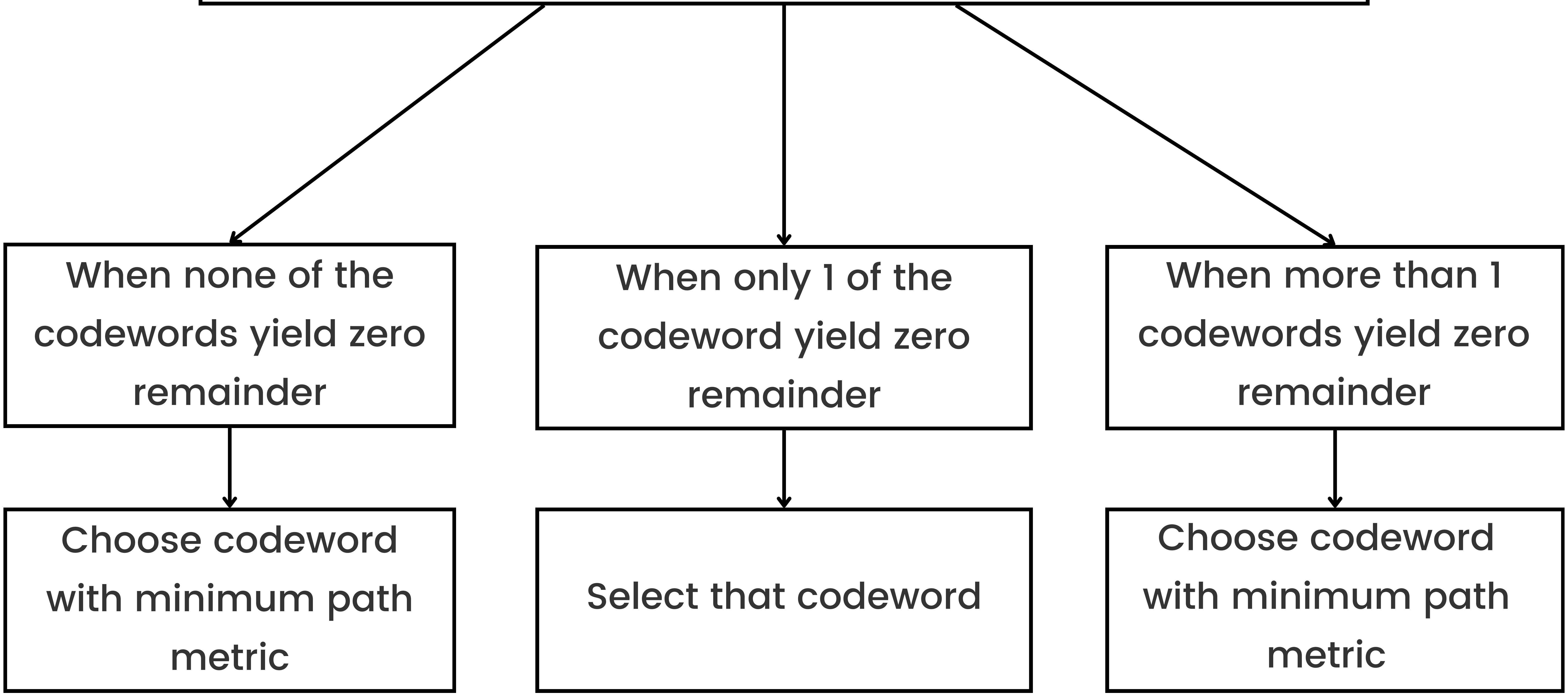
Algorithm:

- Here we use as many SC decoders as our list size (L) and run them together.
- We follow the same procedure as SC decoder for all the concurrent decoders except at the leaf.
- At the leaf we do following extra procedures as follows:
- We consider all possible estimated bits for a given belief
- We keep track of path metric in which we assign a penalty (absolute value of belief) for taking an opposite decision for a given belief.
- Then we sort and take the first L decoders having minimal path metrics.
- After we receive the L decoded codewords we perform a CRC check.

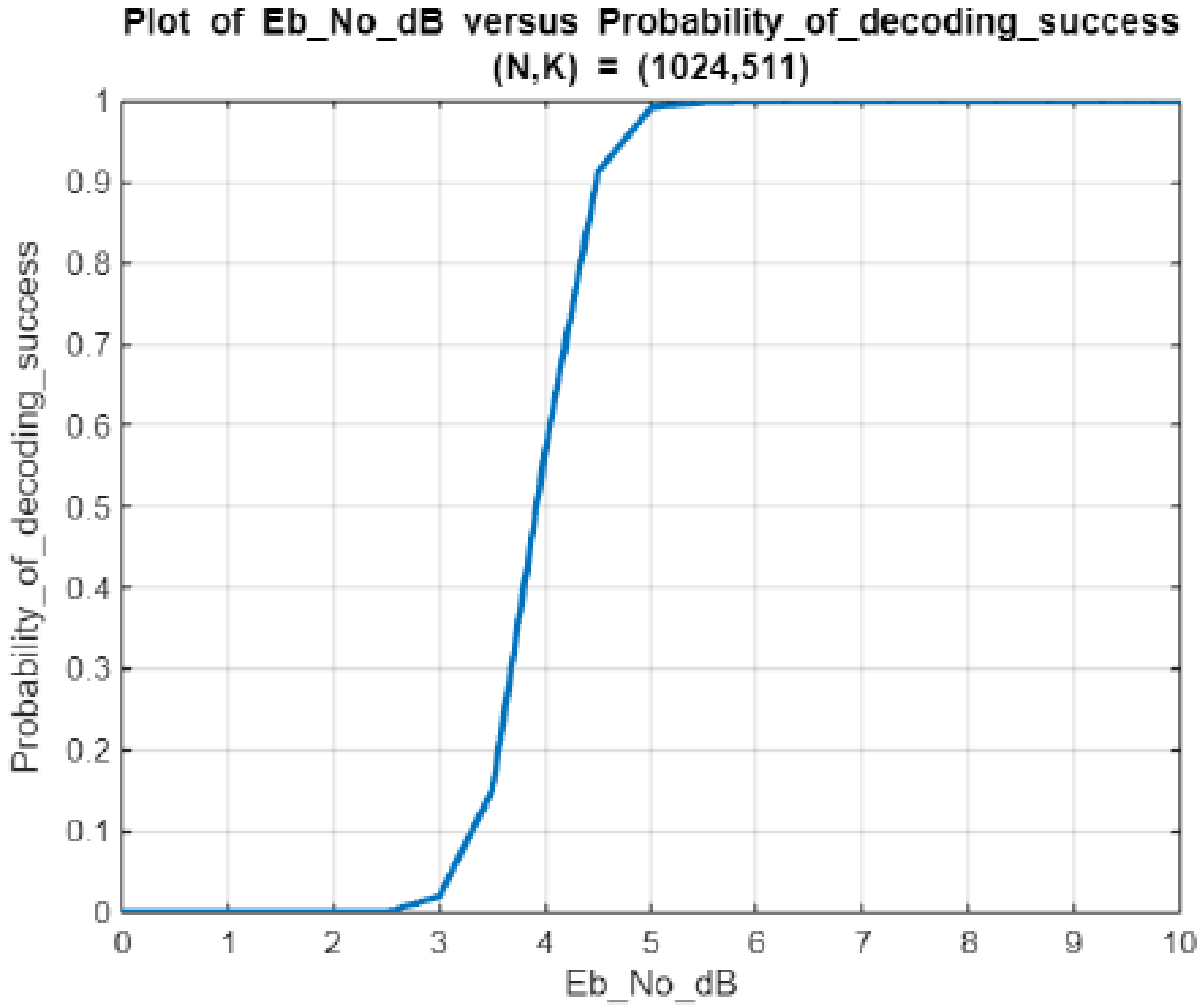

SC decoder 1
PM = 0
u1
0
1
SC decoder 1
DM1(0)
SC decoder 2
DM1(1)
u2
u2
0
1
0
1
SC decoder 1
DM1(0) + DM12(0)
SC decoder 2
DM1(0) + DM12(1)
SC decoder 3
DM1(1) + DM22(0)
SC decoder 4
DM1(1) + DM22(1)

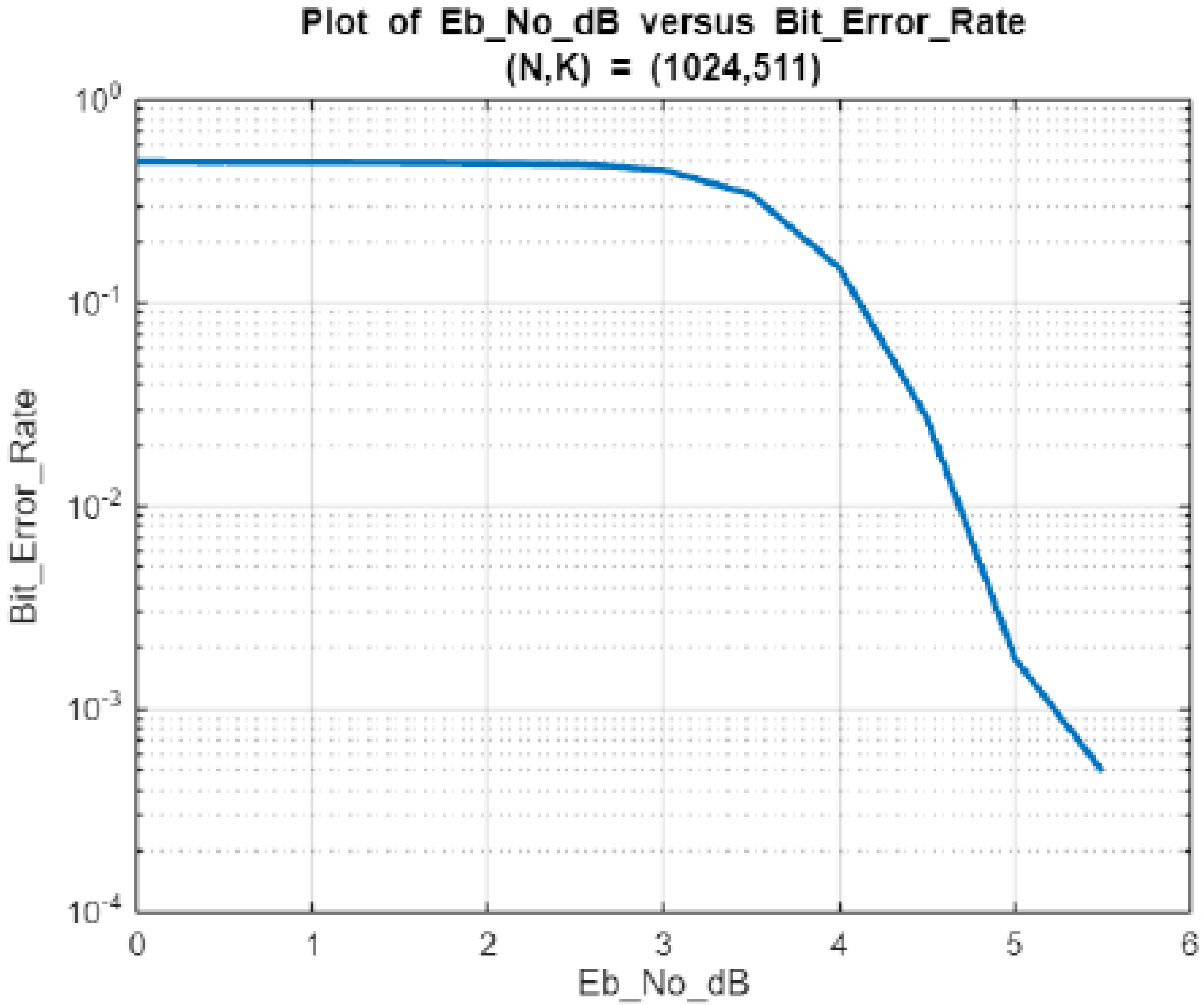

SC decoder 1
PM 1
ui
0
1
1a : PM1 + DM1i(0)
X
1b : PM1 + DM1i(1)
ui+1
SC decoder 2
PM 2
ui
0
1
2a : PM2 + DM2i(0)
ui+1
2b : PM2 + DM2i(1)
X
SC decoder 3
PM 3
ui
0
1
3a : PM3 + DM3i(0)
X
3b : PM3 + DM3i(1)
X
SC decoder 4
PM 4
ui
0
1
4a : PM4 + DM4i(0)
ui+1
4b : PM4 + DM4i(1)
ui+1

Possible scenarios during CRC check on L decoded codewords

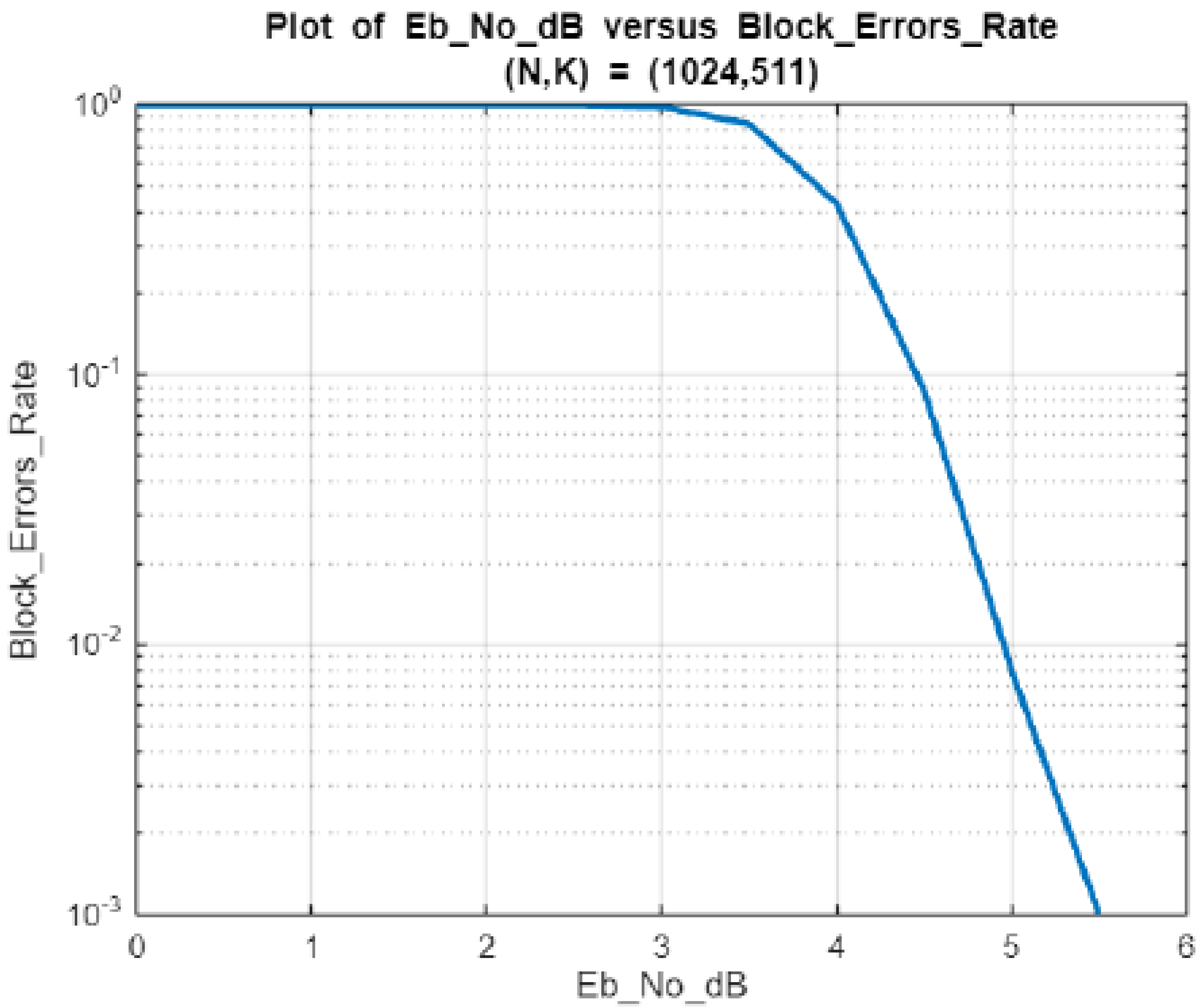


Simulating Outputs

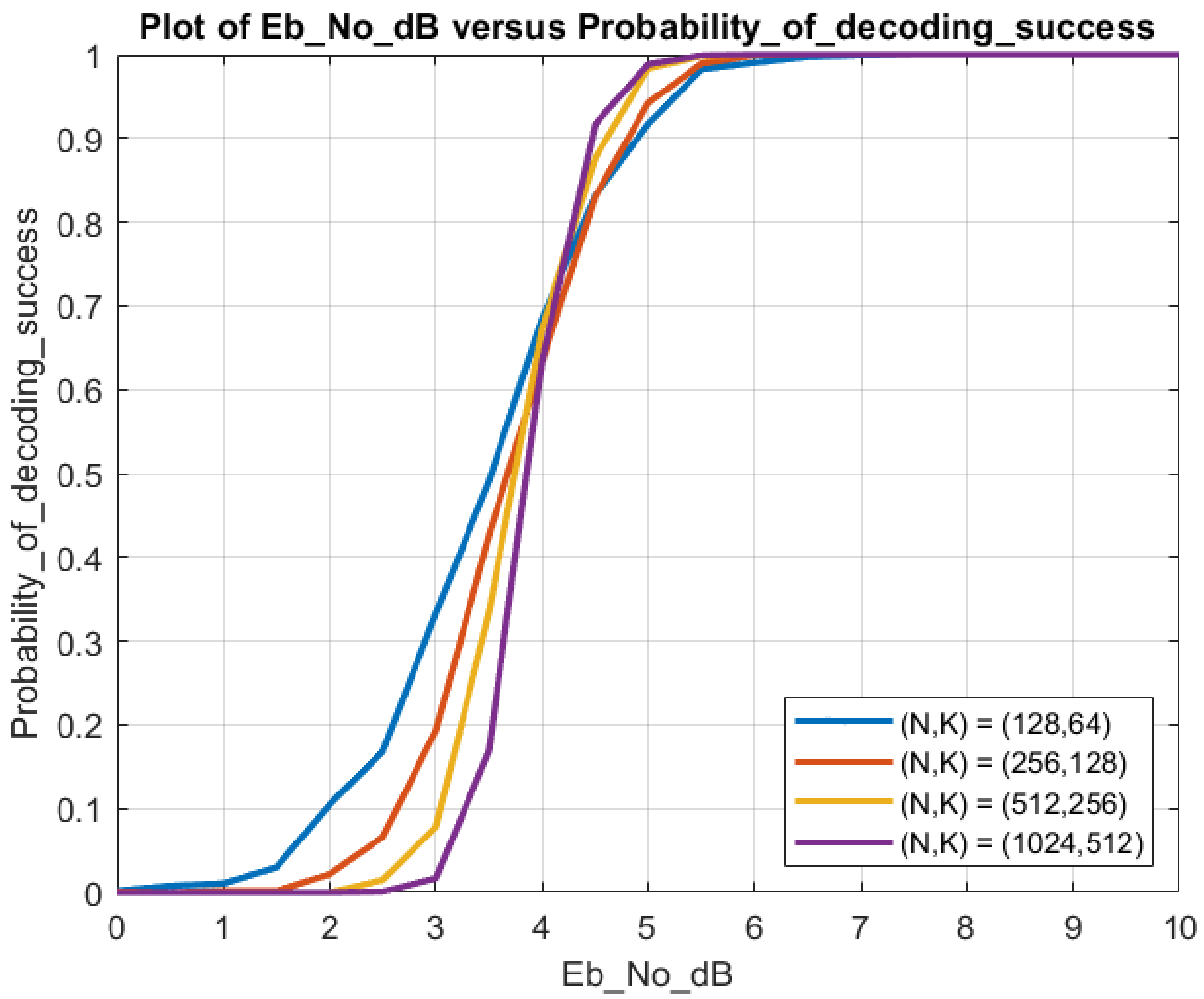

Plot of Eb_No_dB versus Probability_of_decoding_success
(N,K) = (1024,511)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Probability_of_decoding_success
0
1
2
3
4
5
6
7
8
9
10
Eb_No_dB

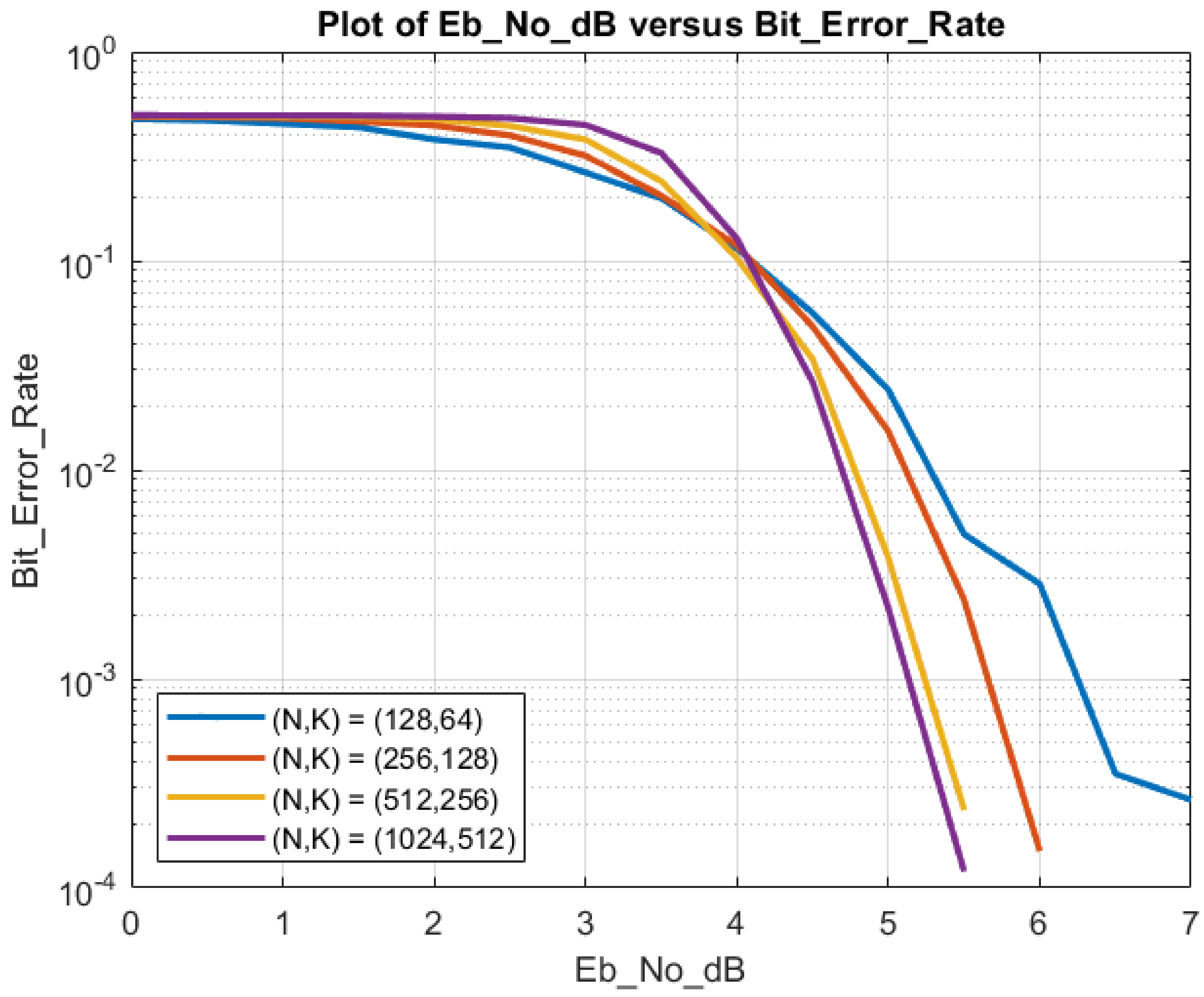


Plot of Eb_No_dB versus Bit_Error_Rate
(N,K) = (1024,511)
Bit_Error_Rate
Eb_No_dB
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
1
2
3
4
5
6

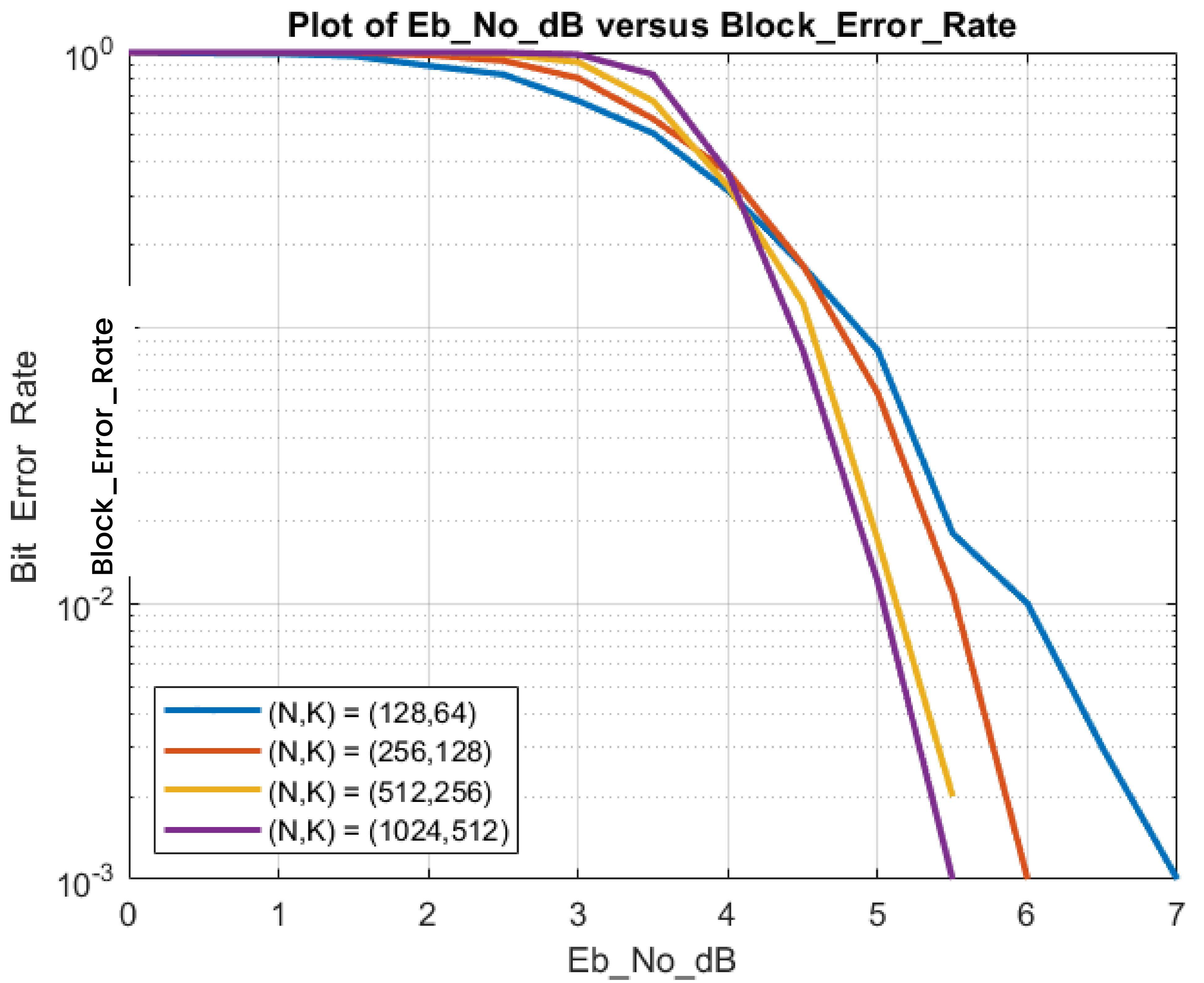

Plot of Eb_No_dB versus Block_Errors_Rate
(N,K) = (1024,511)
Block_Errors_Rate
10^0
10^-1
10^-2
10^-3
0
1
2
3
4
5
6
Eb_No_dB

Simulating for different (N,K)’s

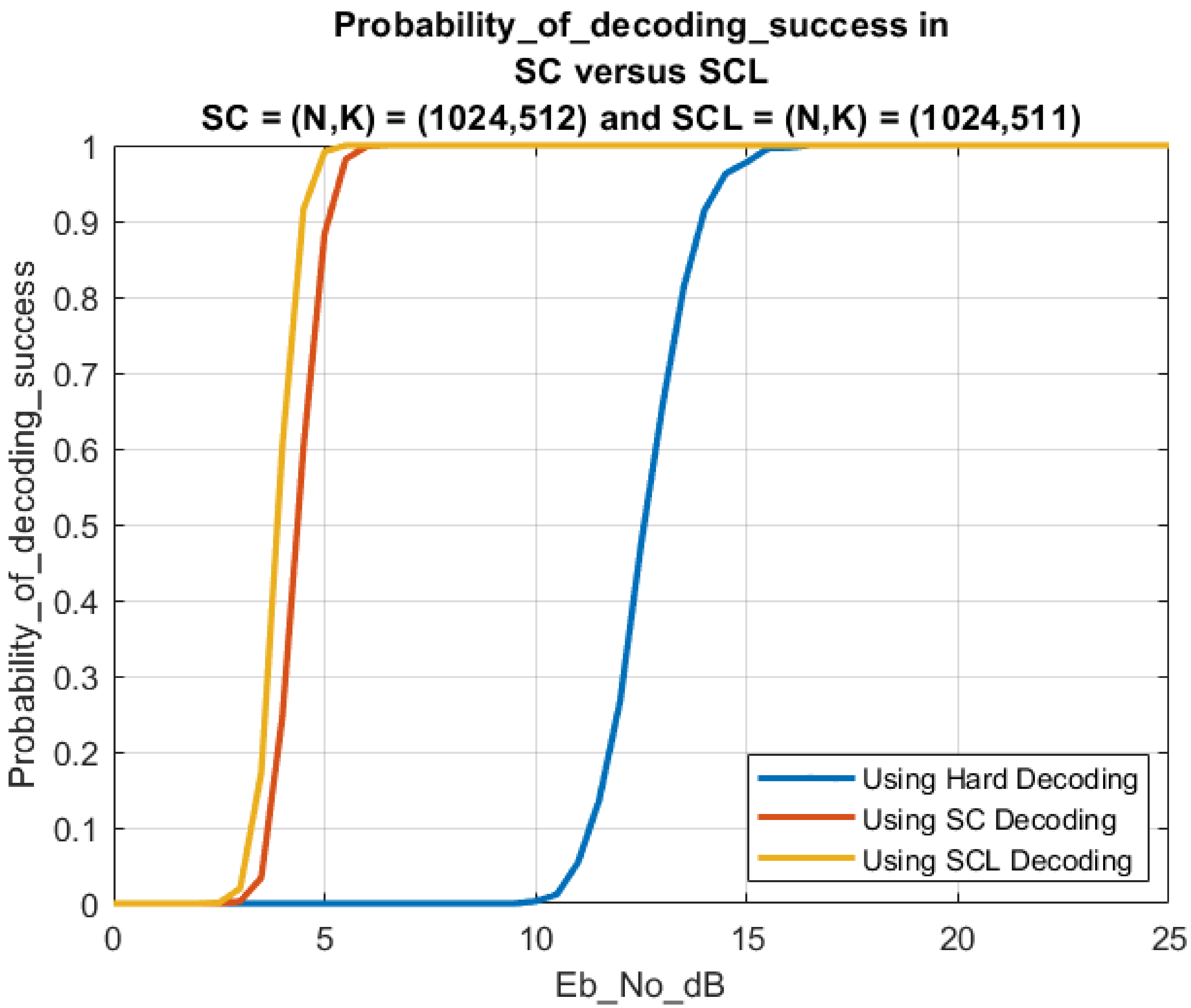

Plot of Eb_No_dB versus Probability_of_decoding_success
Probability_of_decoding_success
Eb_No_dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7
8
9
10
(N,K) = (128,64)
(N,K) = (256,128)
(N,K) = (512,256)
(N,K) = (1024,512)

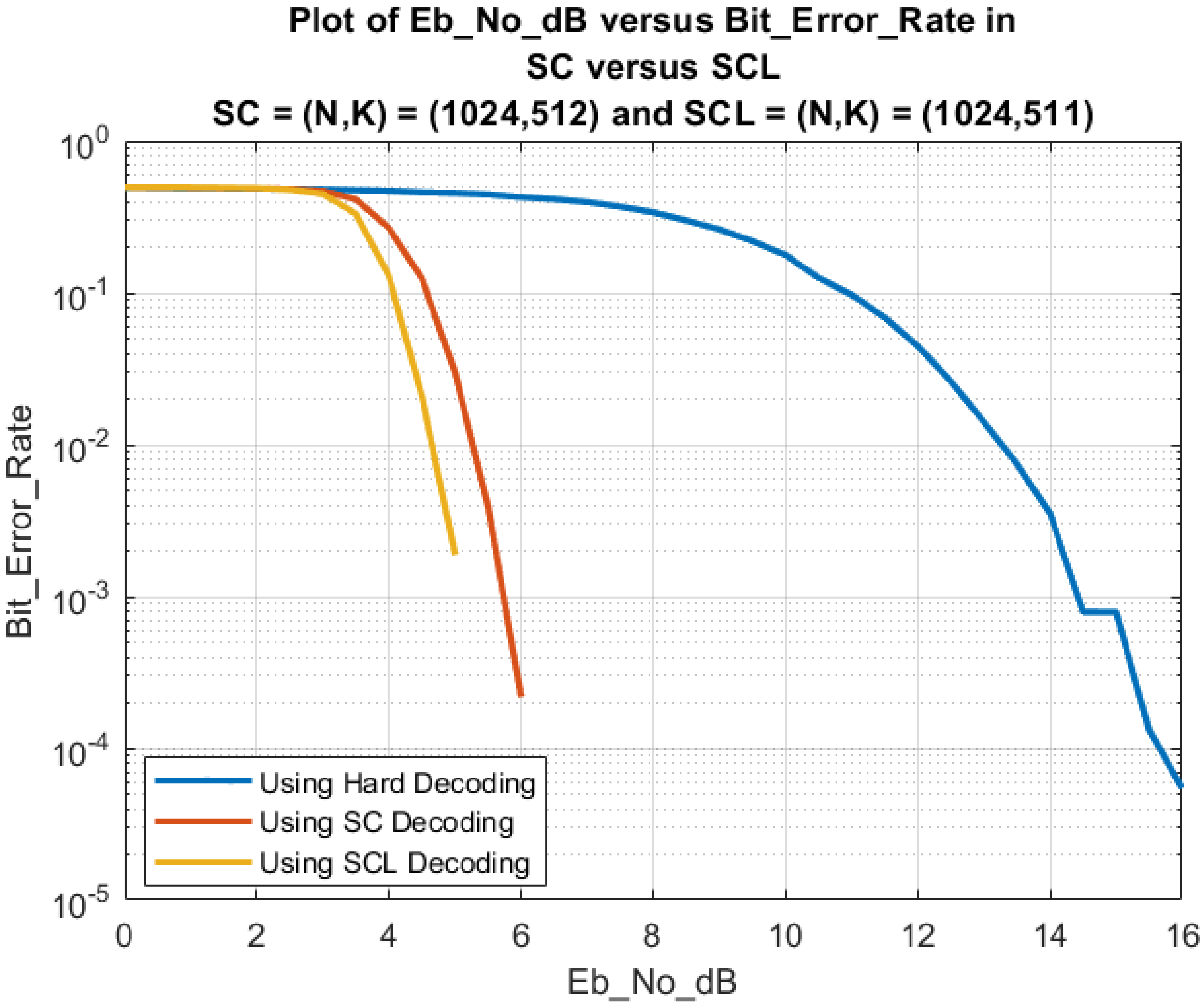


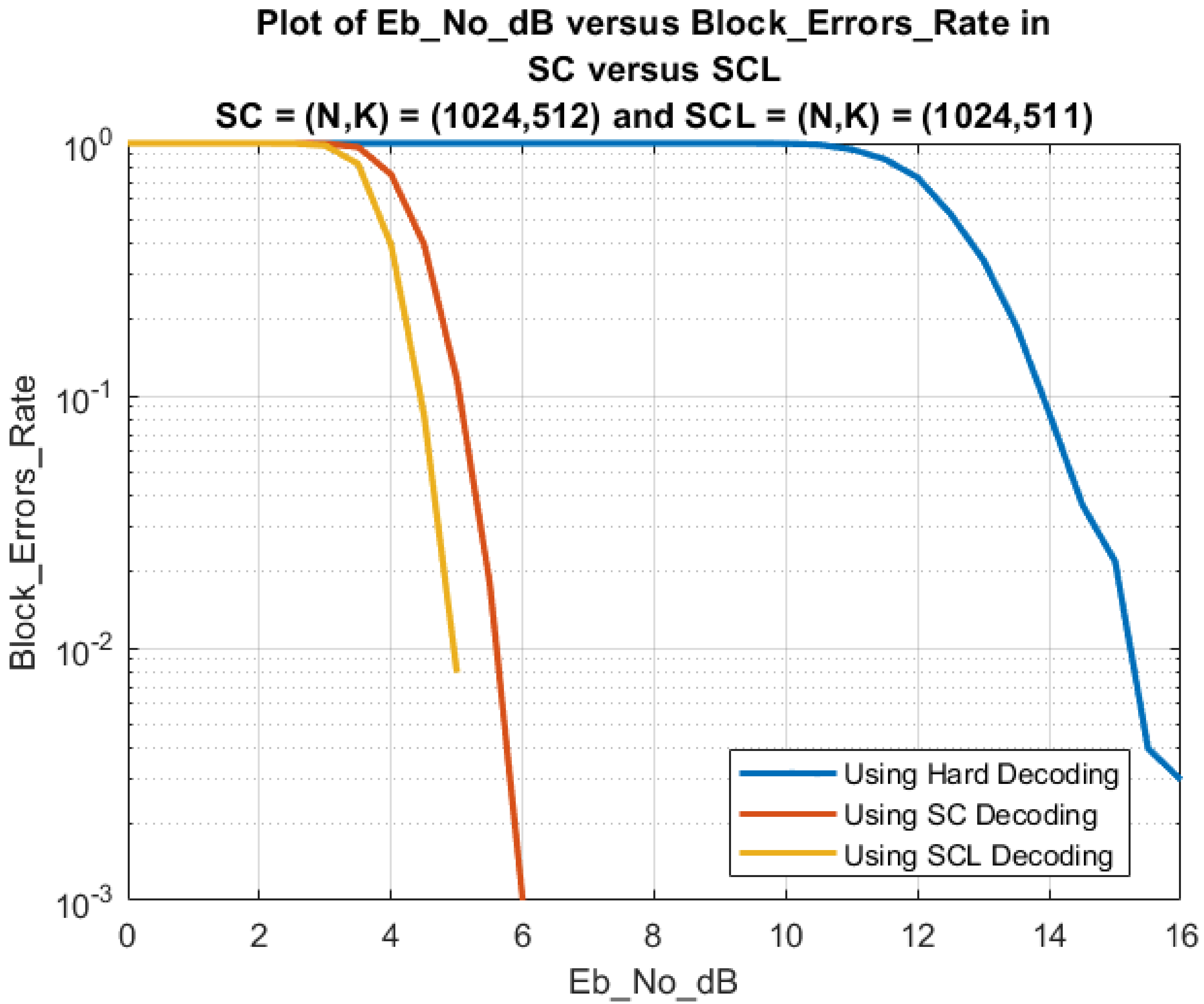
Plot of Eb_No_dB versus Bit_Error_Rate
Bit_Error_Rate
Eb_No_dB
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
1
2
3
4
5
6
7
(N,K) = (128,64)
(N,K) = (256,128)
(N,K) = (512,256)
(N,K) = (1024,512)


Plot of Eb_No_dB versus Block_Error_Rate
Bit Error Rate
Block_Error_Rate
10^{0}
10^{-2}
10^{-3}
0
1
2
3
4
5
6
7
Eb_No_dB
(N,K) = (128,64)
(N,K) = (256,128)
(N,K) = (512,256)
(N,K) = (1024,512)

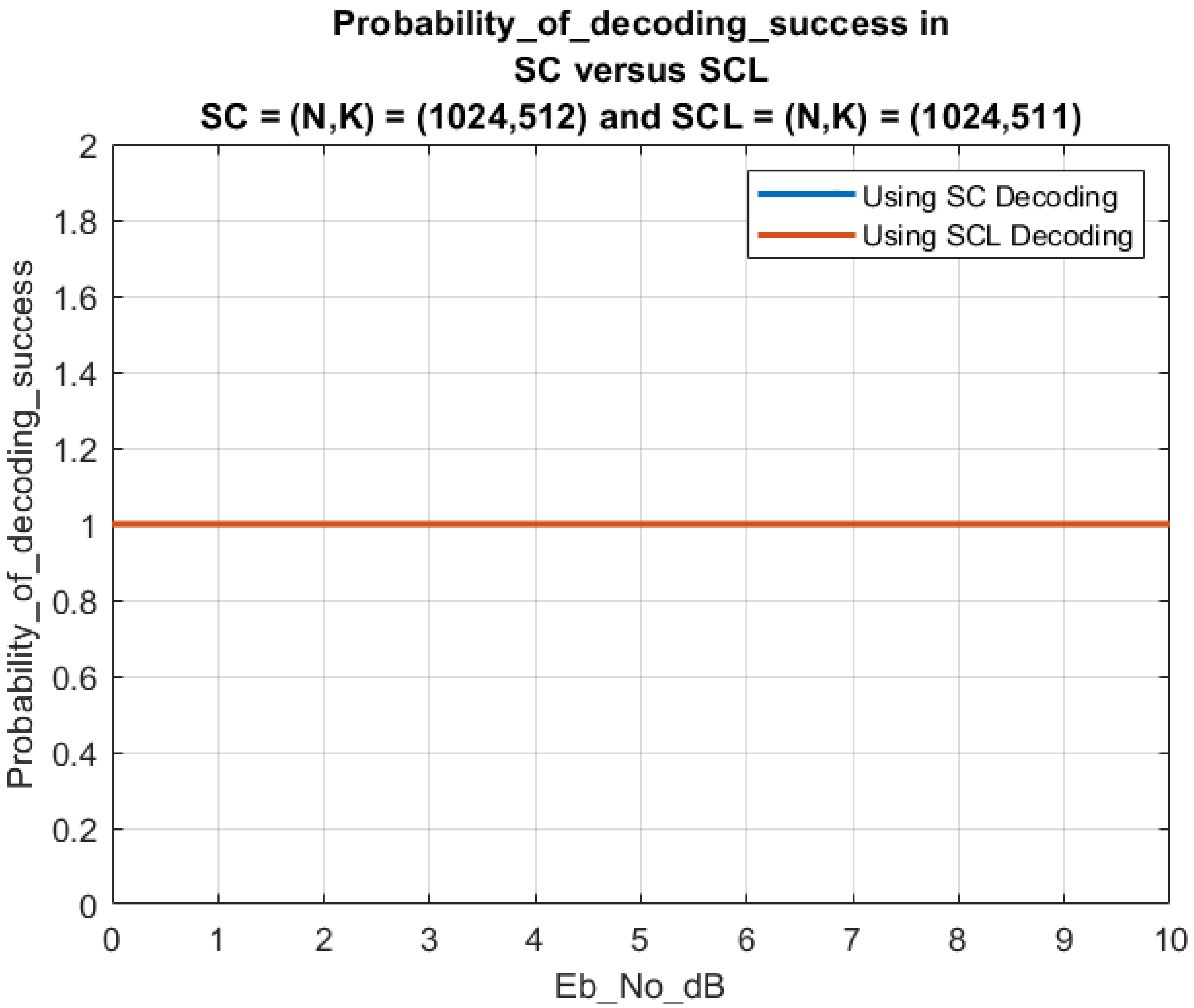
Decoder Comparison Outputs

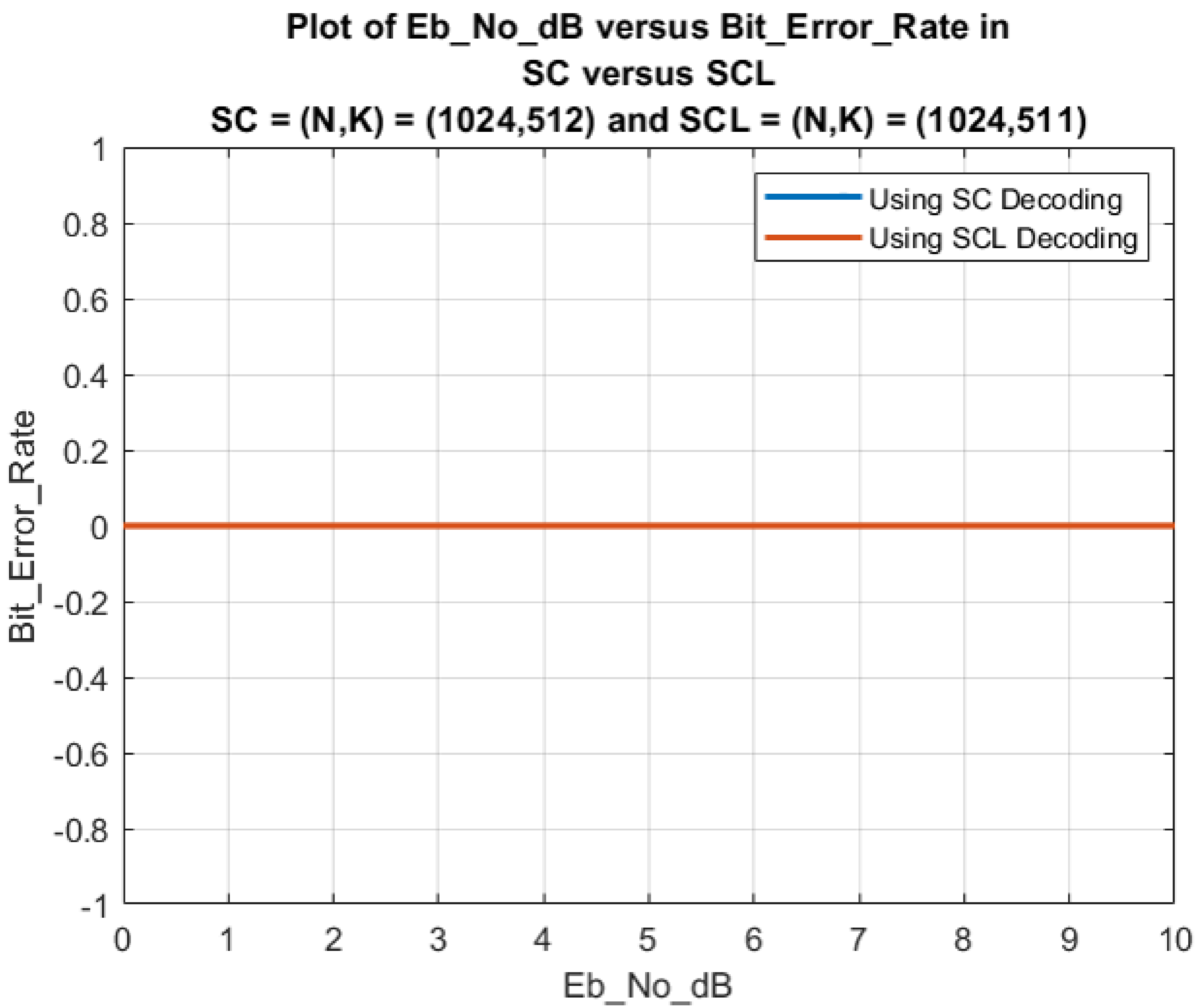

Probability_of_decoding_success in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Probability_of_decoding_success
Eb_No_dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
Using Hard Decoding
Using SC Decoding
Using SCL Decoding

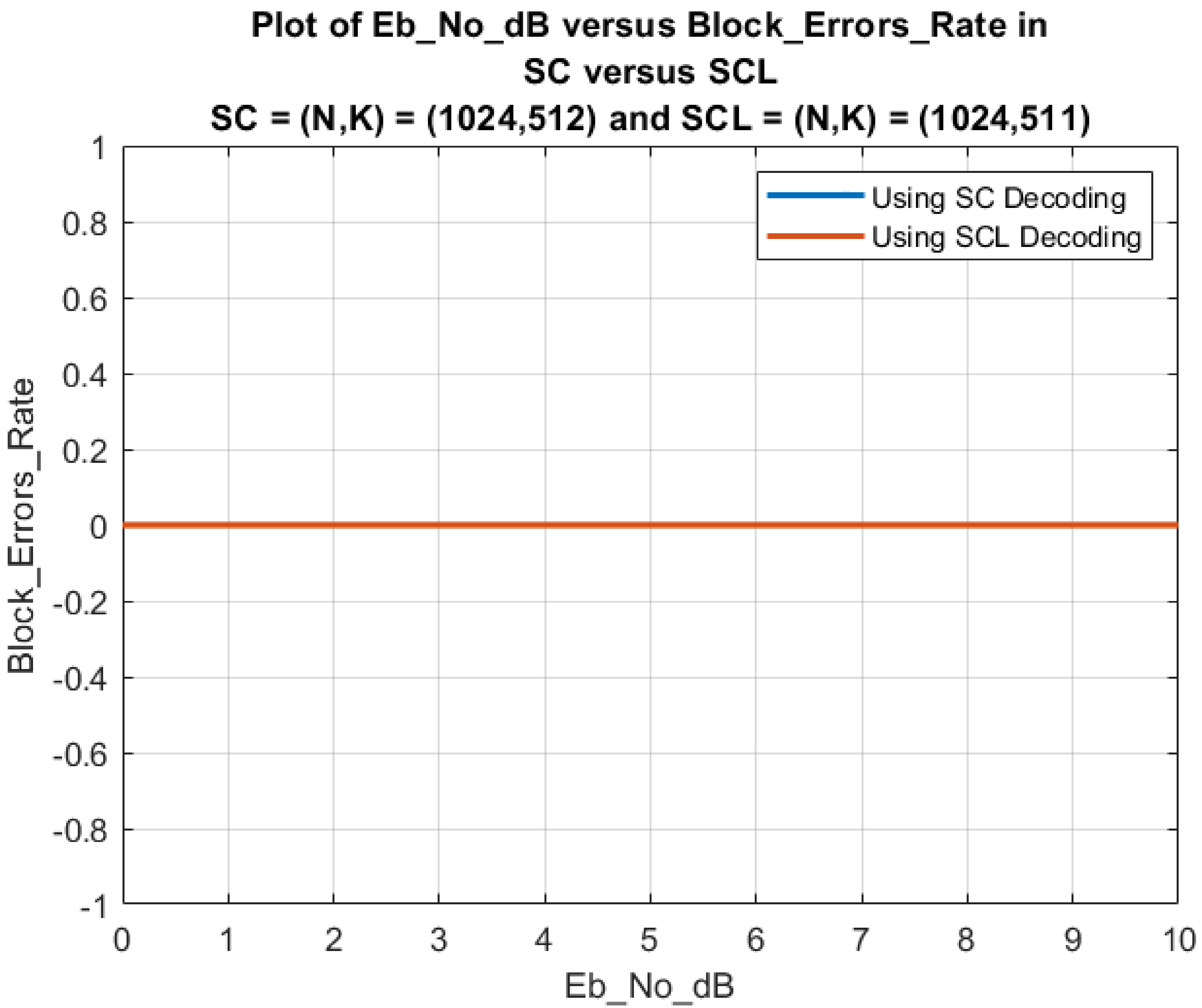

Plot of Eb_No_dB versus Bit_Error_Rate in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Bit_Error_Rate
Eb_No_dB
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
0
2
4
6
8
10
12
14
16
Using Hard Decoding
Using SC Decoding
Using SCL Decoding


Plot of Eb_No_dB versus Block_Errors_Rate in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Block_Errors_Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0
2
4
6
8
10
12
14
16
Eb_No_dB
Using Hard Decoding
Using SC Decoding
Using SCL Decoding

Comparing Functionality Test


Probability_of_decoding_success in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Using SC Decoding
Using SCL Decoding
Probability_of_decoding_success
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10
Eb_No_dB


Plot of Eb_No_dB versus Bit_Error_Rate in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Using SC Decoding
Using SCL Decoding
Bit_Error_Rate
Eb_No_dB
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
10


Plot of Eb_No_dB versus Block_Errors_Rate in
SC versus SCL
SC = (N,K) = (1024,512) and SCL = (N,K) = (1024,511)
Using SC Decoding
Using SCL Decoding
Block_Errors_Rate
Eb_No_dB
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
10

Appendix

SISO Decoder for SPC

Let us assume a Single Parity Check (SPC) Code with (N, K) = (3, 2)
where, K = Number of message bits
N = Number of encoded bits

Let message bits $m = [m_1\, m_2]$
Let code word $c = [c_1\, c_2\, c_3]$
Let Received bits $r = [r_1\, r_2\, r_3]$
Let us define the Beliefs for these Code word bits be, Beliefs = $[l_1\, l_2\, l_3]$

Now, In SPC Code, If we know c_2 and c_3, then the bit c_1 is given by,
$\therefore c_1 = c_2 \oplus c_3$

Now, We define the Likelihood Ratio for bit c_i, where $i \in N$ as,
$\therefore \lambda_i = \frac{\mathrm{pr}(c_i=0|r_i)}{\mathrm{pr}(c_i=1|r_i)}$

And so the Log-Likelihood Ratio for bit c_i, where $i \in N$ is given by,
$\therefore l_i = \ln(\lambda_i)$
$\therefore l_i = \ln\left(\frac{\mathrm{pr}(c_i=0|r_i)}{\mathrm{pr}(c_i=1|r_i)}\right)$

Now Assume that, we know the probability of c_2 being 1 is p_2 and c_3 being 1 is p_3.
So, The Probability of bit $c_1 = 1$ is given by,
$\therefore 1 - p_1 = (1 - p_2) \cdot p_3 + (1 - p_3) \cdot p_2$

and, The Probability of bit $c_1 = 0$ is given by,
$\therefore p_1 = p_2 \cdot p_3 + (1 - p_2) \cdot (1 - p_3)$

From the above two equations the Following Equation Holds true:
$\therefore \frac{p_1-(1-p_1)}{p_1+(1-p_1)} = \frac{p_2-(1-p_2)}{p_2+(1-p_2)} \cdot \frac{p_3-(1-p_3)}{p_3+(1-p_3)}$

$\therefore \frac{1-(\frac{1-p_1}{p_1})}{1+(\frac{1-p_1}{p_1})} = \frac{1-(\frac{1-p_2}{p_2})}{1+(\frac{1-p_2}{p_2})} \cdot \frac{1-(\frac{1-p_3}{p_3})}{1+(\frac{1-p_3}{p_3})}$

Now, Let us define Extrinsic Log-Likelihood ratio, $l_{\text{ext,i}} = \ln(\frac{p_i}{1-p_i})$, , where $i \in N$

Therefore, above Equation can be written as,

$\therefore \frac{1-e^{-l_{\text{ext},1}}}{1+e^{-l_{\text{ext},1}}} = \frac{1-e^{-l_2}}{1+e^{-l_2}} \cdot \frac{1-e^{-l_3}}{1+e^{-l_3}}$

Now, We know that, $\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{1-e^{-2x}}{1+e^{-2x}}$

Thus, the above Equation becomes,

$$\therefore \tanh\left(\frac{l_{\text{ext},1}}{2}\right) = \tanh\left(\frac{l_2}{2}\right) \cdot \tanh\left(\frac{l_3}{3}\right)$$

Function Analysis of $\tanh(x)$:

Let f(x) $= \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

Now replacing x by -x,

f(-x) $= \frac{e^{-x}-e^{x}}{e^{-x}+e^{x}}$

f(-x) $=$ - $\frac{e^{-x}-e^{x}}{e^{-x}+e^{x}}$

f(-x) $=$ - f(x)

f(x) $=$ - f(-x)

Hence, $\tanh(x)$ is an odd function:

Here, we observe that,

- $x < 0 \Rightarrow \tanh(x) < 0$
- $x > 0 \Rightarrow \tanh(x) > 0$

The above results shows that $\tanh(x)$ follows sgn(x) function
So, we have above set of Equations,

$$\therefore \operatorname{sgn}(l_{\text{ext},1}) = \operatorname{sgn}(l_2) \cdot \operatorname{sgn}(l_3)$$

$$\therefore \tanh\left(\frac{|l_{\text{ext},1}|}{2}\right) = \tanh\left(\frac{|l_2|}{2}\right) \cdot \tanh\left(\frac{|l_3|}{2}\right)$$

$$\therefore \ln\left(\tanh\left(\frac{|l_{\text{ext},1}|}{2}\right)\right) = \ln\left(\tanh\left(\frac{|l_2|}{2}\right)\right) + \ln\left(\tanh\left(\frac{|l_3|}{2}\right)\right)$$

Let, $\mathrm{f(x)} = \left|\ln\left(\tanh\frac{|x|}{2}\right)\right|$, where $x > 0$;

Hence, $f^{-1}(x) = \mathrm{f(x)}$

$$\therefore f(|l_{\text{ext},1}|) = f(|l_2|) + f(|l_3|)$$

$$\therefore |l_{\text{ext},1}| = f(f(|l_2|) + f(|l_3|))$$

Here, Intrinsic Beliefs are : l_1, l_2, l_3
Here, Extrinsic Beliefs are : $l_{\text{ext},1}, l_{\text{ext},2}, l_{\text{ext},3}$
Also, $f(x)$ is a non-linear function.

Thus, the Expression $f(l_2) + f(l_3)$ is approximately equal to $f(\min(|l_2|, |l_3|))$

So, The Min-sum approximation is given by :

$$\therefore |l_{\text{ext}_1}| = \min(|l_2|, |l_3|)$$

$$\therefore sgn(l_{\text{ext}_1}) = sgn(l_2) \cdot sgn(l_3)$$

SISO Decoder for Repetition Code

Log Likelihood Ratio for $n = 3$ Repetition code

$\mathbf{c} = [c_1\ c_2\ c_3]$ Code_word
$\mathbf{r} = [r_1\ r_2\ r_3]$ Received_word

$\mathrm{pr}(c_{1=0}|r_1) = \frac{f(r_1|c_{1=0})\cdot \mathrm{pr}(c_{1=0})}{f(r_1)}$

$\mathrm{pr}(c_{1=1}|r_1) = \frac{f(r_1|c_{1=1})\cdot \mathrm{pr}(c_{1=1})}{f(r_1)}$

Likelihood Ratio:

$$\frac{\Pr(c_{1=0}|r_1)}{\Pr(c_{1=1}|r_1)} = \frac{f(r_1|c_{1=0})}{f(r_1|c_{1=1})}$$

Where $f(x)$ is the probability density function for the Gaussian distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

When $c_{1=0}$, the symbol s will be $+1$ and hence the received bit $r_1 = s + n$:

$$r_1 = 1 + n$$

So:

$$\frac{f(r_1|c_{1=0})}{f(r_1|c_{1=1})} = \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_1-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_1+1)^2}{2\sigma^2}}}$$

Taking the logarithm on both sides:

$$\log\left(\frac{\Pr(c_{1=0}|r_1)}{\Pr(c_{1=1}|r_1)}\right) = \log\left(e^{\frac{2r_1}{\sigma^2}}\right) = \frac{2r_1}{\sigma^2}$$

Output Log-Likelihood Ratio (LLR) SISO Decoder:
Belief vector $\rightarrow [L_1\ L_2\ L_3]$
$L_i = \log\left(\frac{\Pr(c_{i=0}|r_1,r_2,r_3)}{\Pr(c_{i=1}|r_1,r_2,r_3)}\right)$
L_1:

$$\frac{f(r_1,r_2,r_3|c_{1=0}) \cdot \Pr(c_{1=0})}{\Pr(r_1,r_2,r_3)}$$

Likelihood ratio of L_1:

$$\frac{f(r_1, r_2, r_3 | c_{1=0})}{f(r_1, r_2, r_3 | c_{1=1})}$$

When $c_{1=0}$ and the output from BPSK is $[+1 \; +1 \; +1]$, then:

$$r_1 = 1 + N_1(0, \sigma^2)$$
$$r_2 = 1 + N_2(0, \sigma^2)$$
$$r_3 = 1 + N_3(0, \sigma^2)$$

$$\frac{e^{-\frac{(r_1-1)^2}{2\sigma^2}} \cdot e^{-\frac{(r_2-1)^2}{2\sigma^2}} \cdot e^{-\frac{(r_3-1)^2}{2\sigma^2}}}{e^{-\frac{(r_1+1)^2}{2\sigma^2}} \cdot e^{-\frac{(r_2+1)^2}{2\sigma^2}} \cdot e^{-\frac{(r_3+1)^2}{2\sigma^2}}}$$

Taking log:

$$L_1 = (r_1 + r_2 + r_3) \cdot \frac{2}{\sigma^2}$$

$L_i = (r_1 + r_2 + r_3) \cdot \frac{2}{\sigma^2}$, where we set the factor $\frac{2}{\sigma^2}$ as 1 for L_1.
For L_1, r_1 is intrinsic and r_2, r_3 are extrinsic.

References

- Polar Codes in 5G Standard by Prof. Andrew Thangaraj
- Essence of Polar Codes by Dr. Erdal Arikan
- Flesh of Polar Codes by Dr. Emre Telatar
- Summary on Polar codes by Prof. Tufail Ahmad
- Slides by Dr. Erdal Arikan
- Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels
- On the rate of Channel Polarisation
- Source Polarisation by Dr. Erdal Arikan
- A Brief Introduction to Polar Codes by Dr. Henry Pfister
- Simplified Calculation of Bhattacharyya Parameters in Polar Codes
- Polar codes: Primary concepts and practical decoding algorithms

Thank You!