Co.

Polar Codes

Members:
Raj Shah - 202201403
Joshi Dev Jayantbhai - 202201405
Patel Herik Kiranbhai - 202201407
Mavadiya Sujal Pravinbhai - 202201040
Aryankumar Panchasara - 202201056
Gandhi Ayush Chiragbhai - 202201057
Patel Nipurnakumari Laxmanbhai - 202201061
Zinzuvadia Zenith Mehul - 202201082
Kanishk Kunj - 202201080
Vaghasiya Sahil Pareshbhai - 202201083
Rahulkumar Galchar - 202201100

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.

Signed by: all the members of the project group

Context

Introduction

Theoretical Concepts

Channel polarization, Encoding
Decoding

Implementations And Results

Summary

Co.

Introduction

Polar codes are linear block error correcting codes invented by Erdal Arikan in 2008. Polar codes help us to either make a channel “good” (highly reliable) or “bad”(with very low reliability). So by choosing appropriate channels we can send messages over a noisy channel with less probability of error.
Other important properties of polar codes are that it has very high capacity achieving property and low complexity of the encoding and decoding algorithms.

Channel Capacity theorem:
For a wide range of communication channels, there exists a channel capacity, denoted as C, such that it is possible to achieve reliable transmission at any rate below C (i.e., when R < C).
This theorem addresses the trade-off between transmission rate and reliability, confirming that reliable communication can be attained within certain rate limits.

However, the theorem does not discuss the complexity involved in achieving this balance. Practical coding schemes that are easy to implement are necessary for real-world applications.
Polar codes can reach the channel capacity and offer low-complexity encoding and decoding algorithms. This makes polar codes an attractive option for efficient and effective data transmission.

Advantages of Polar Coding:

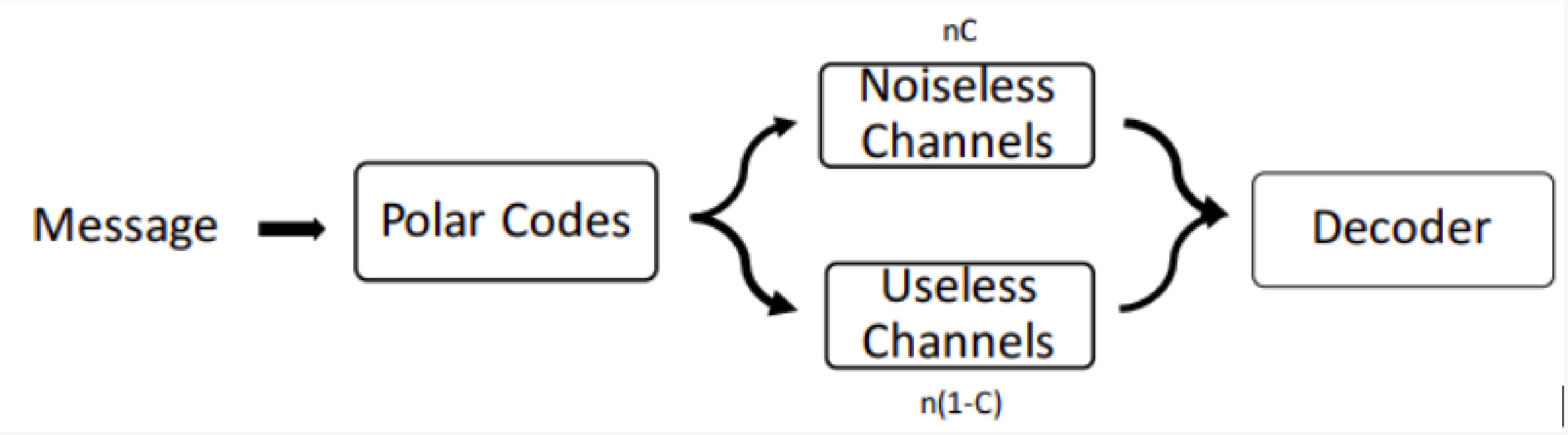
1)Capacity achieving performance: The performance of the polar code reaches the limit of Shannon channel capacity and hence is more optimal.
2)Low Complexity of decoding algorithms: The complexity of the SC decoding algorithm is O(nlogn) making it more suitable for practical usage.
3)Versatility: Polar codes provide a universal solution for all sorts of channels like BSC, BEC, etc

Disadvantages of Polar Coding:

1) Successive cancellation decoder used by Polar codes is poor in performance as compared to LDPC and turbo code techniques.
2) It offers O(n) higher latency.
3) At finite N, polar codes perform best with advanced decoders but at a higher price as compared to LDPC.

Applications:

1)5G wireless communication
2)Satellite communication
3)Storage Systems(like SSDs)
4)IoT
5)Cloud computing

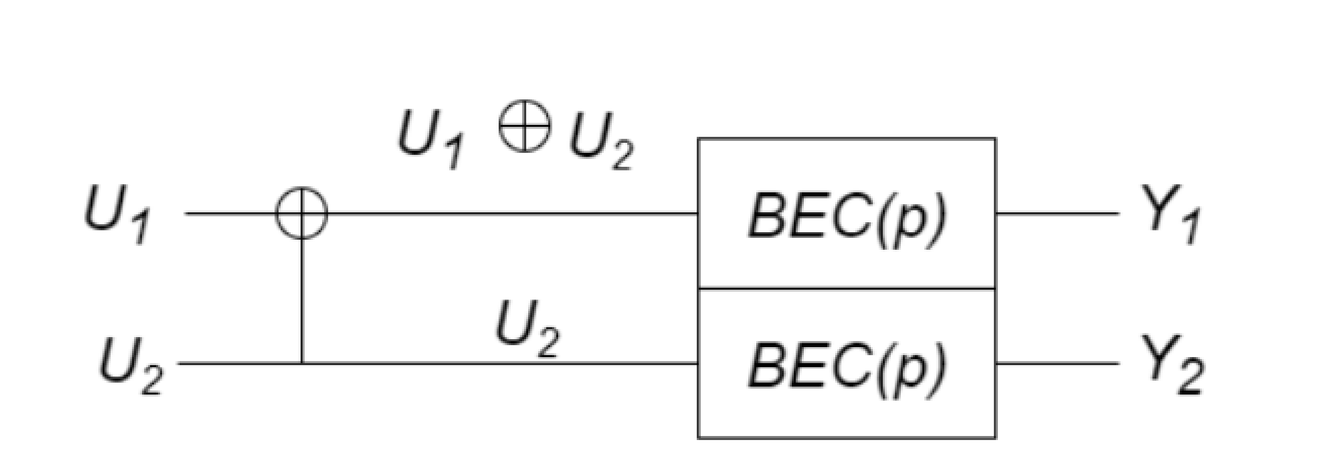


Co.

Theoretical Concepts

Channel Polarization:

Let there be a BEC channel with p as the error probability. So 1-p is the probability by which the receiver receives the bits correctly.Sending each bit separately on a BEC(p) channel would actually yield two identical channels through which both bits are sent.



So the polar transformation creates 2 new channels i.e W^- and W^+ .

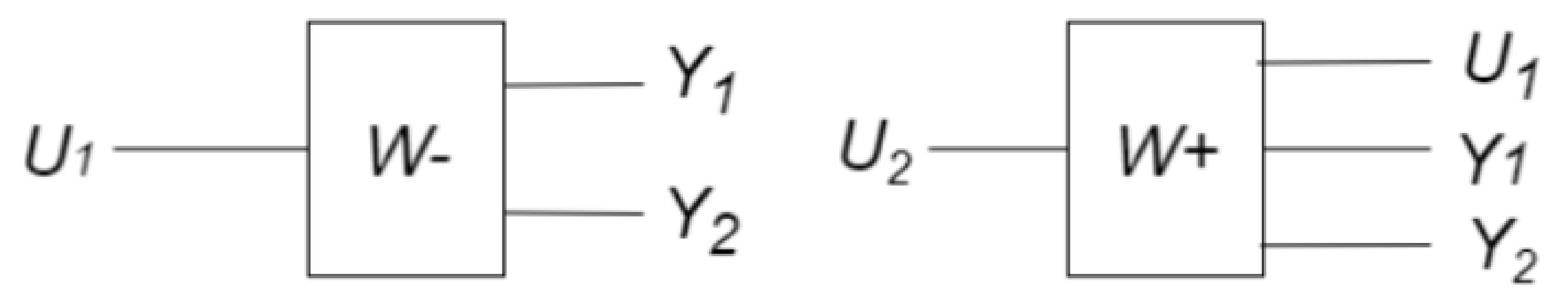
Y_1 and Y_2 are used to decode U_1 in the W^- channel. So we get the estimated value of U_1. Assuming that the estimated value of U_1 is correct using U_1, Y_1 and Y_2 we will decode U_2 in the W+ channel.

Suppose W is the BEC(p),

$$
(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-p)^2 \\ (\ ?\ , U_2) & \text{w.p. } p(1-p) \\ (U_1 \oplus U_2, ?\ ) & \text{w.p. } (1-p)p \\ (\ ?\ , ?\ ) & \text{w.p. } p^2 \end{cases}
$$

$$
Y = \begin{cases} X & \text{with probability } 1-p, \\ ? & \text{with probability } p \end{cases}
$$

For W^-, input is U_1 , and output :



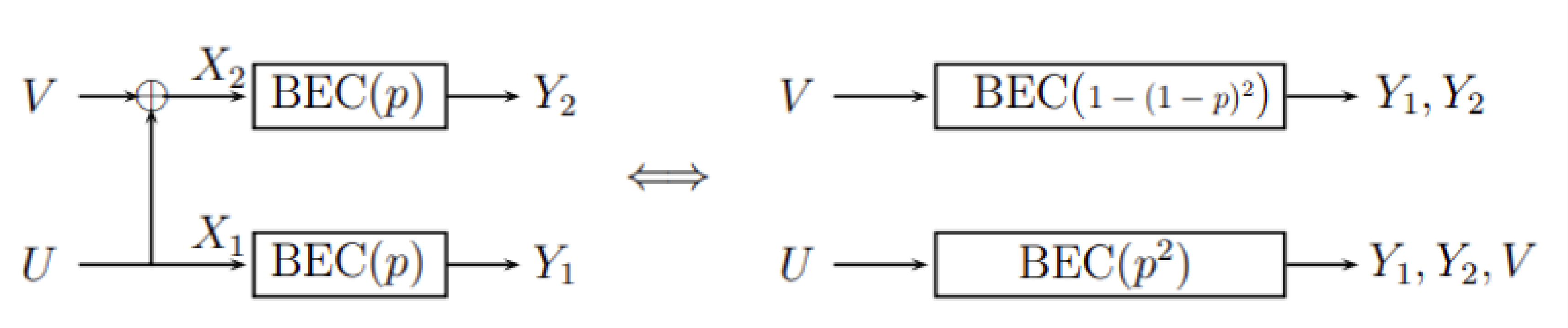
$$
(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-p)^2 \\ (\ ?\ , U_2) & \text{w.p. } p(1-p) \\ (U_1 \oplus U_2, ?\ ) & \text{w.p. } (1-p)p \\ (\ ?\ , ?\ ) & \text{w.p. } p^2 \end{cases}
$$

In the W^- channel, the probability that we receive Y_1 an Y_2 correctly is 1-p each. We know that $Y_1 = U_1 \oplus U_2$ and $Y_2 = U_2$
So if we receive Y_1 and Y_2 correctly we can get the probability for which we can decode U_1 correctly which is $(1-p)^2$. So we can write that the error for U_1 is the function of BEC$(1-(1-p)^2)$.
W^- is BEC$(2p-p^2)$.

For W^+, input is U_2 , output is:

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-p)^2 \\ (\quad ? \quad, U_2, U_1) & \text{w.p. } p(1-p) \\ (U_1 \oplus U_2, \; ? \;, U_1) & \text{w.p. } (1-p)p \\ (\quad ? \quad, \; ? \;, U_1) & \text{w.p. } p^2 \end{cases}$$

In W^+ channel, decoding of U_2 depends on Y_1, Y_2 and U_1. Only the last case lacks U_2, rest other cases has both U_1 and U_2, so the error for U_2 is p^2.
W^+ is BEC(p^2).



Channel Splitting:
Channel splitting is the recursive process which aims to polarize the channel. It splits the channel into good channels (high reliability & high capacity/ low error rates) and bad channels (low capacity & low reliability / high error rates).
This polarization effect is what gives polar codes the ability to achieve capacity.
In previous channel polarization, we consider W^+ as a good/ reliable channel and W^- as a bad/unreliable channel.

BEC error rate for W^+ is p^2 and for W^- is $2p-p^2$, so the error rate for W^+ is less than W^- and hence it is W^+ is a good channel and W^- is a bad channel.

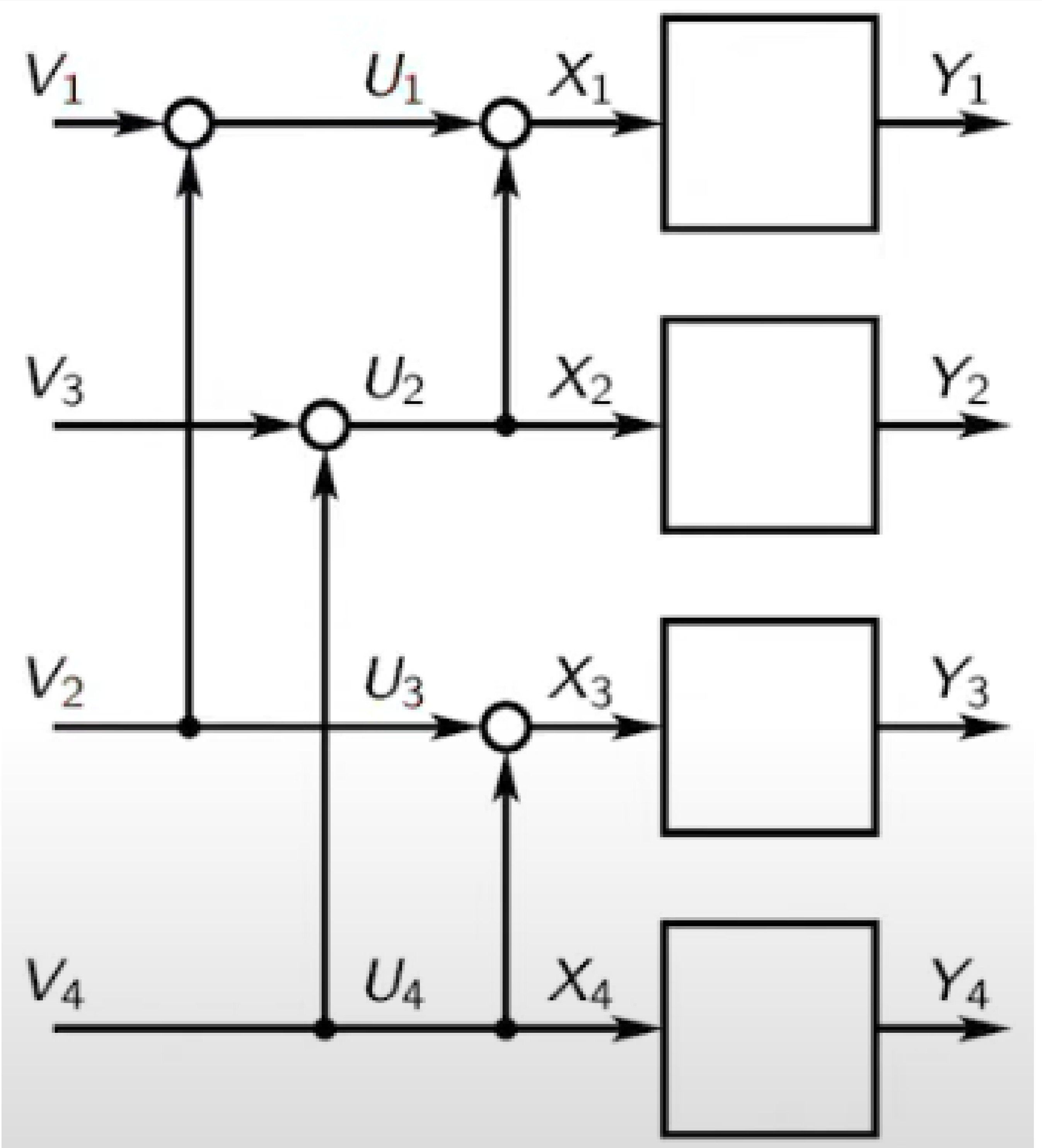
Duplicate W^- and W^+ to obtain:

$W^{--}: V_1 \rightarrow Y_1\ Y_2\ Y_3\ Y_4$

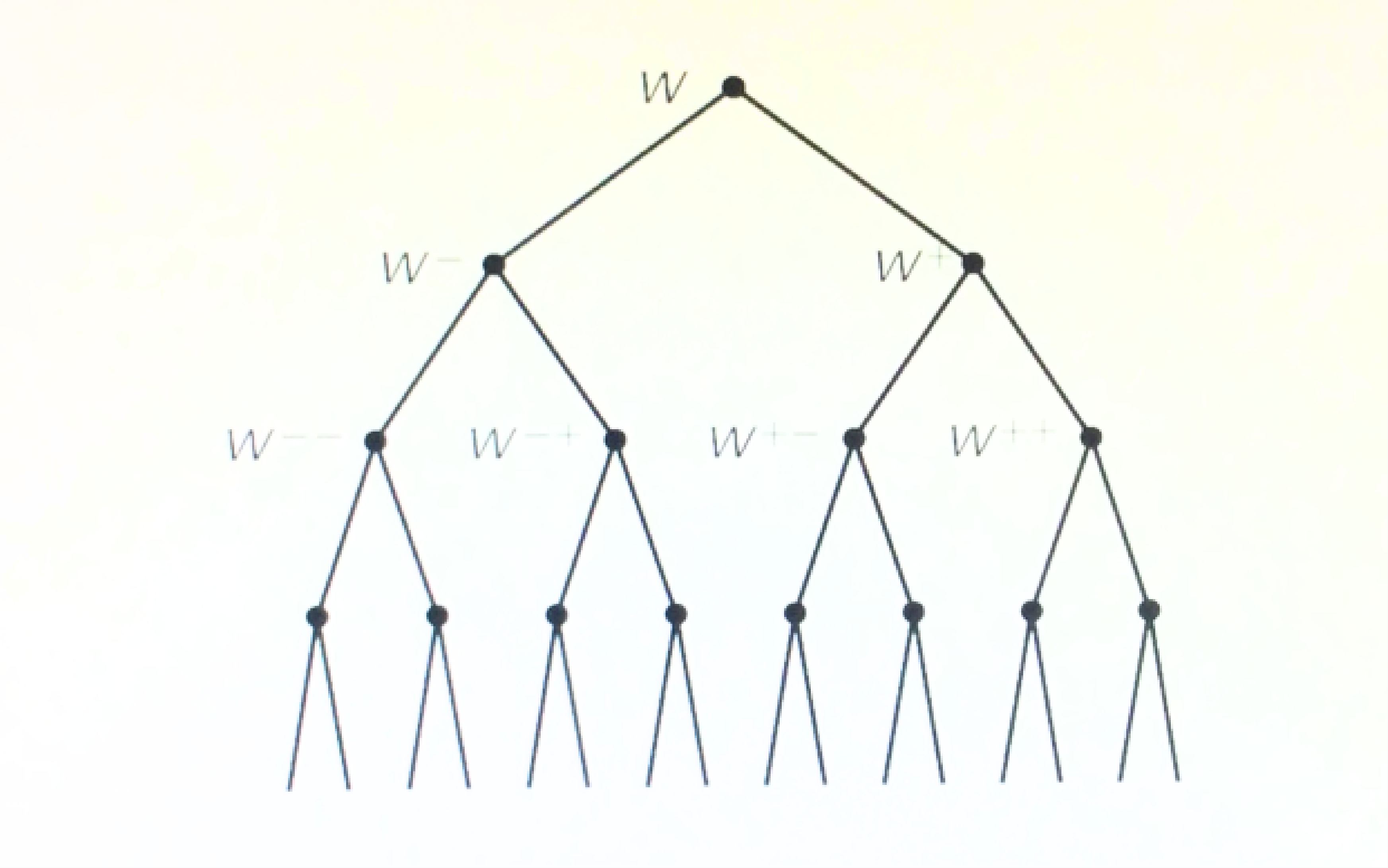
$W^{-+}: V_2 \rightarrow Y_1\ Y_2\ Y_3\ Y_4\ V_1$

$W^{+-}: V_3: \rightarrow Y_1\ Y_2\ Y_3\ Y_4\ V_1\ V_2$

$W^{++}: V_4: \rightarrow Y_1\ Y_2\ Y_3\ Y_4\ V_1\ V_2\ V_3$



As the number of duplication increases the channels tend to polarize in good and bad channels so there are very less number of channels present in middle ground so we can use the good channels to encode our message bit and the rest of the channels are used to input the frozen bits.



Encoding:

Polar Transform is defined using the principle of channel polarization. The G_2 matrix shown below is used for N=2 polar code.

$$G2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

So, $[u1 \;\; u2][G2] = [u1 + u2 \quad u2]$

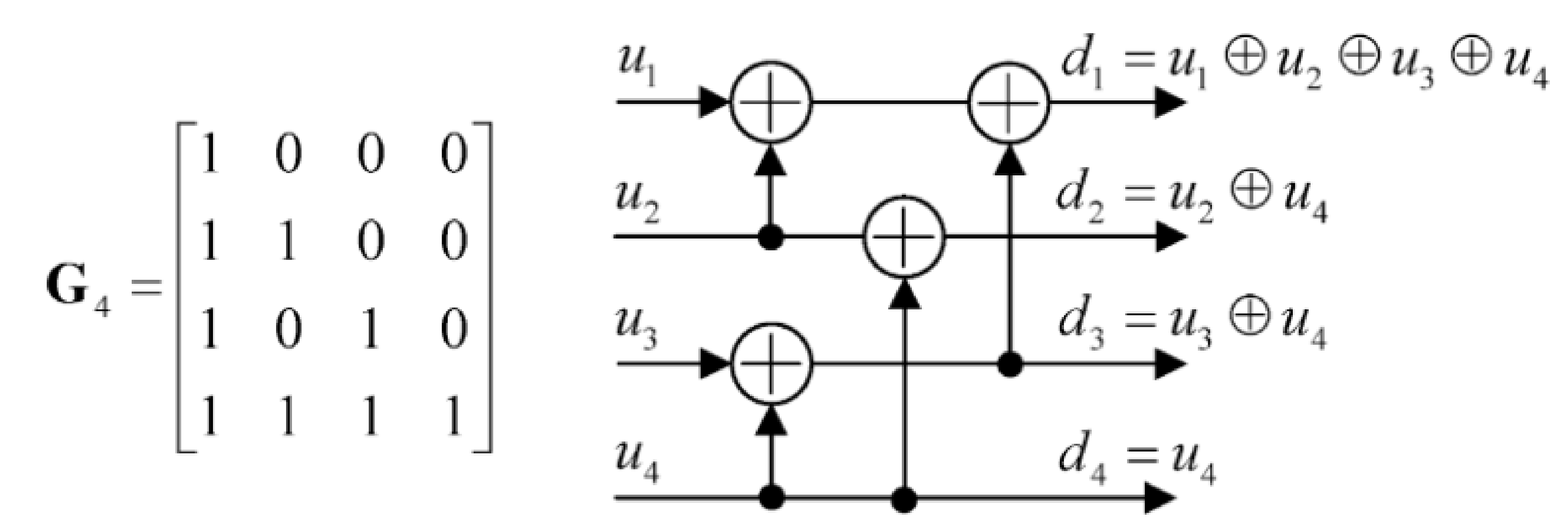
For larger polar codes we Find the Kronecker product of G_2 with itself N times for N Polar code. In the Kronecker product we replace each element of the matrix with the other matrix multiplied by the element.

$$G_{2^n} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n}$$

$N=2^n$

G_n: N x N matrix, Kronecker product of 2 x 2 kernel.

Example for N = 4 → G_4:

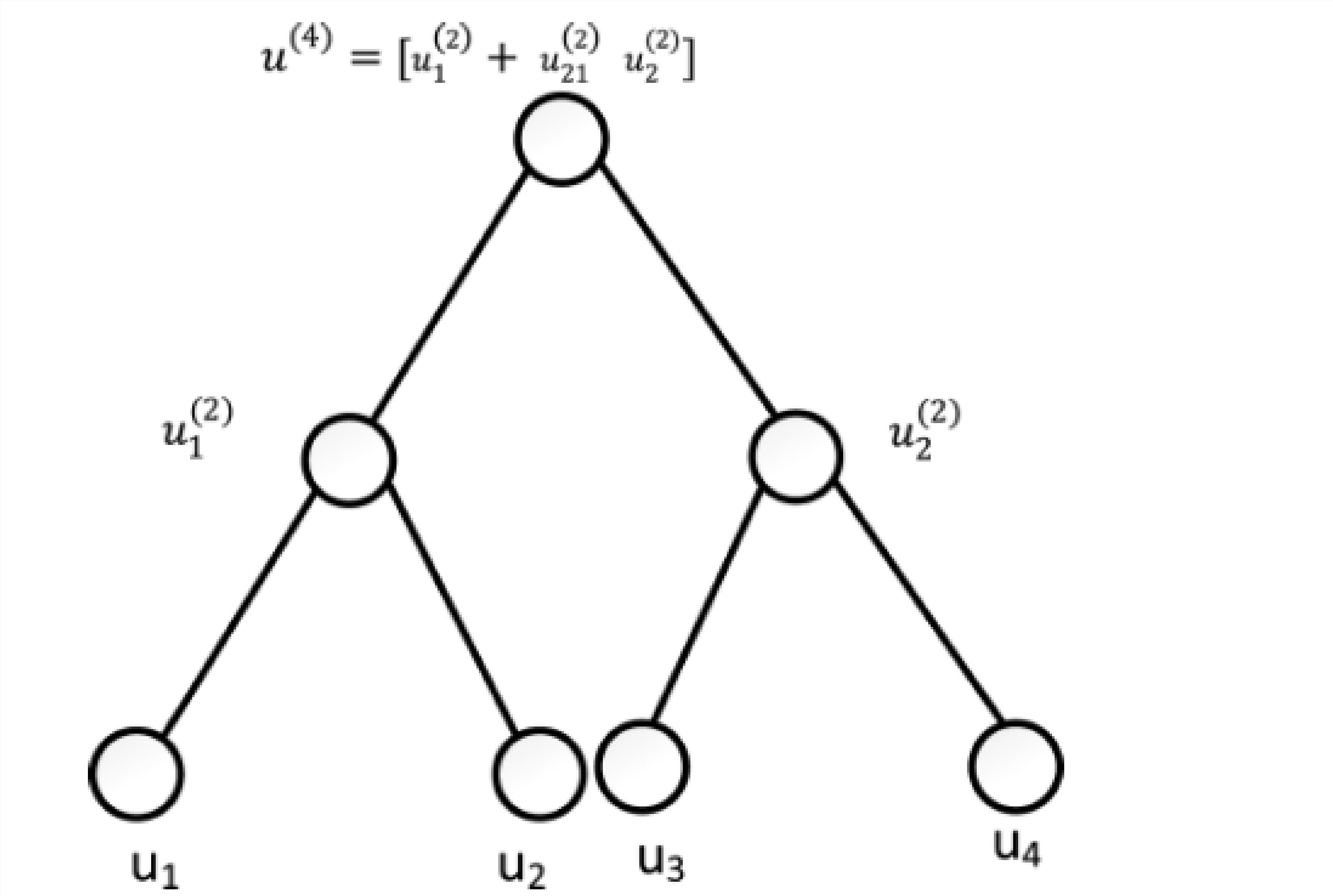


Using this N=4 polar code, the encoded message is formed as:

$[u_1 , u_2 , u_3 , u_4][G_4] = [u_1+u_2+u_3+u_4 , u_2+u_4 , u_3+u_4 , u_4]$

Note that the plus operation is the XOR operation or modulo-2 addition.

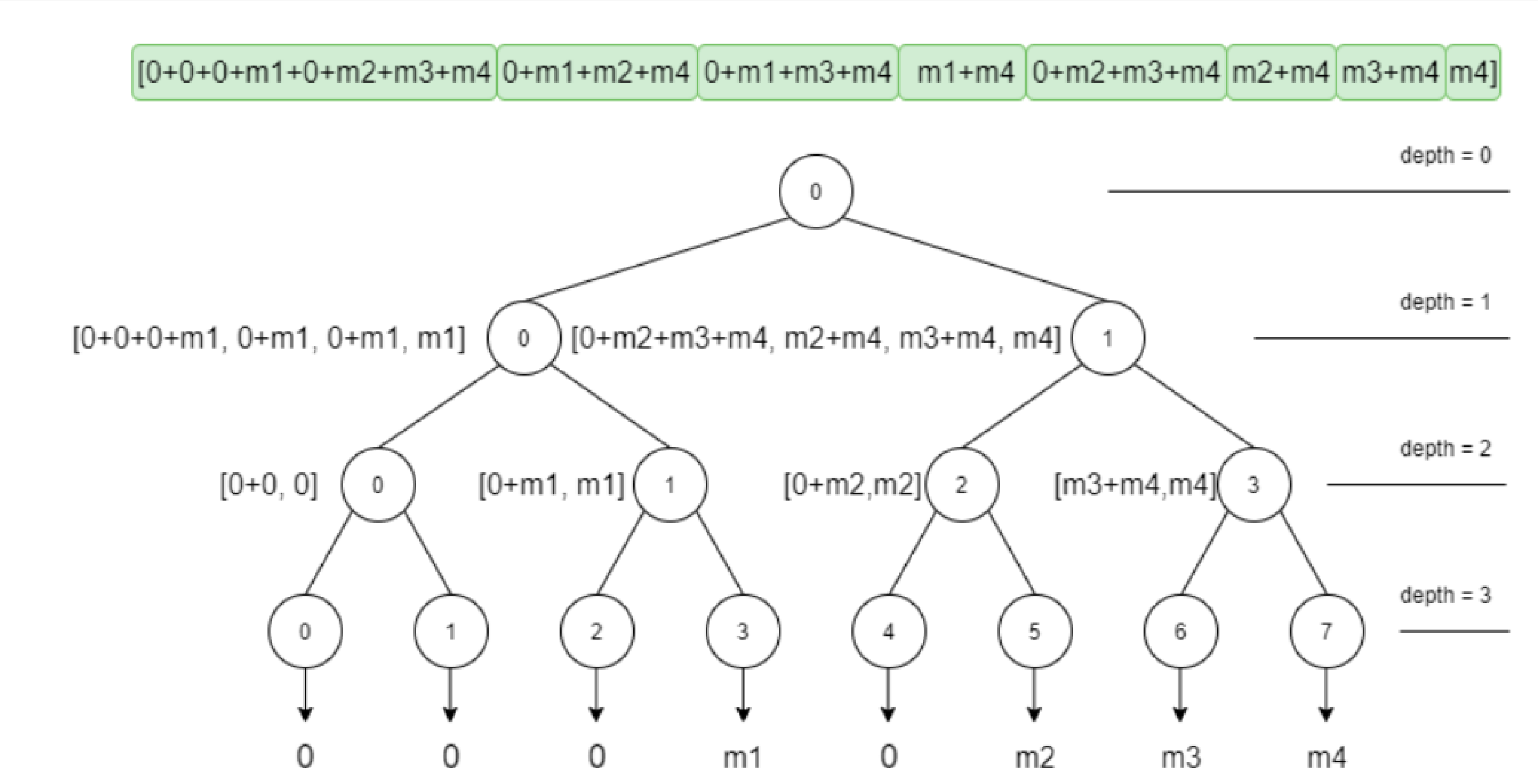
For simplification we can use the binary tree representation for the same:



In the binary tree representation the root of the tree represents the encoded message and the leaf nodes are fed with either the input message or 0 depending on its reliability.
Consider a case where we have a k bit message input which is encoded into N bit codeword.
$N = 2^n$. Here n will be the depth of the binary tree representing this encoding method.
So the N leaf nodes of this binary tree are fed with information bits. Now the k bits which are assigned with the information bits are decided by the reliability sequence.

Reliability Sequence:
The sequence of bit-channel positions ordered in increasing order of reliability is called the reliability sequence. According to this sequence, we can determine which positions are more reliable and consequently have probabilities of erasure smaller than other positions. Therefore, we allocate data bits to these positions and frozen bits to the rest of the positions in which the probability of erasure is higher. In fact, we consider all frozen bits as zero and when we want to do decoding, we already know that frozen bits are zero and we just have to do decoding for the rest of the positions.

The Reliability Sequence of N=8 is {1, 2, 3, 5, 4, 6, 7, 8} (channel number 1 is the worst channel and channel number 8 is the best one)



In every step of the encoding process starting from the node positions the bits of 2 consecutive positions are combined as: $[u_1 + u_2 , u_2]$ and the process continues till we reach the root node so the process starts at depth n till root node.

Time complexity: O(NlogN)

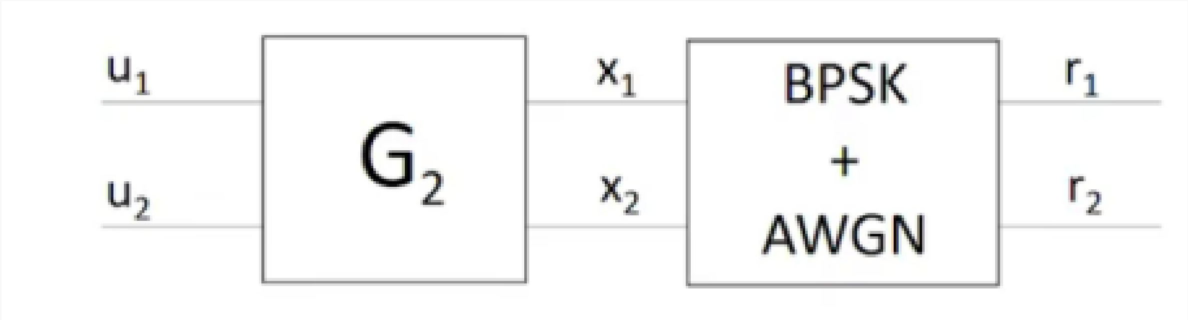
Proof: Let i_n be the complexity of computing polar transform $x = uG_n$. Using recursion,

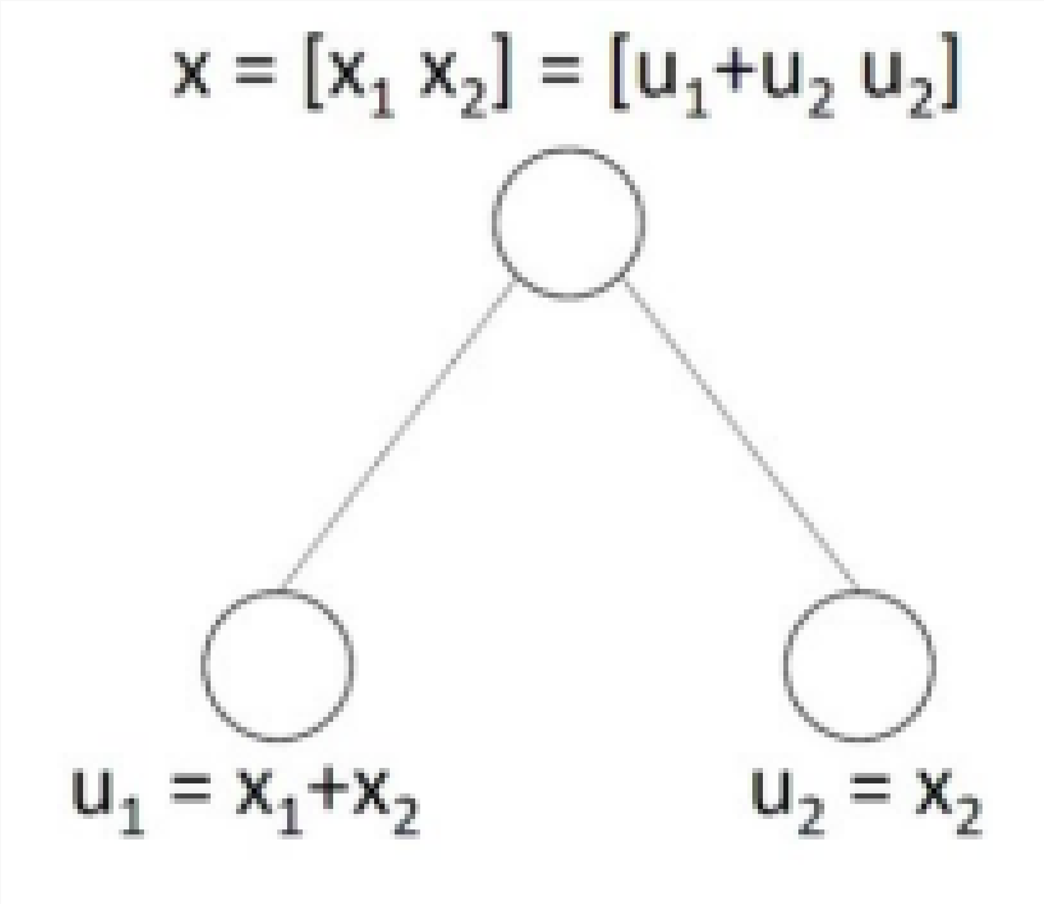
$$G_N = \begin{bmatrix} Gn/2 & 0 \\ Gn/2 & Gn/2 \end{bmatrix}$$ where Gn/2 is $G_{N/2}$.

We calculate that $i_n\ \ 2i_n/_2 + 3N/2$ with $i_2 = 3$. Therefore, $i_n <= (3N/2)\log N$.

Successive Cancellation(SC) Decoder:

It is a successive decoder so it decodes bits individually one after the other and uses the previous results in further decoding process. Using the reliability sequence we already know which bits are frozen so the decoder simply sets it to zero and moves ahead. If the bits are not frozen then we use the soft decoding technique where we first find the estimated value of u_1 using the $F(r_1,r_2)$ function and then assuming that u_1 is correct depending on the value of u_1 we find the estimated value of u_2.Consider the case for 2 bit decoder first:





The first figure shows the complete transmission and encoding process and we know that

$X_1 = u_1 + u_2$

$X_2 = u_2$

So we can conclude that during decoding process

$U_1 = x_1 + x_2$

$U_2 = x_2$

R_1 and r_2 are beliefs for x_1 and x_2 and to find the beliefs of u_1 we use function called min sum function defined as:

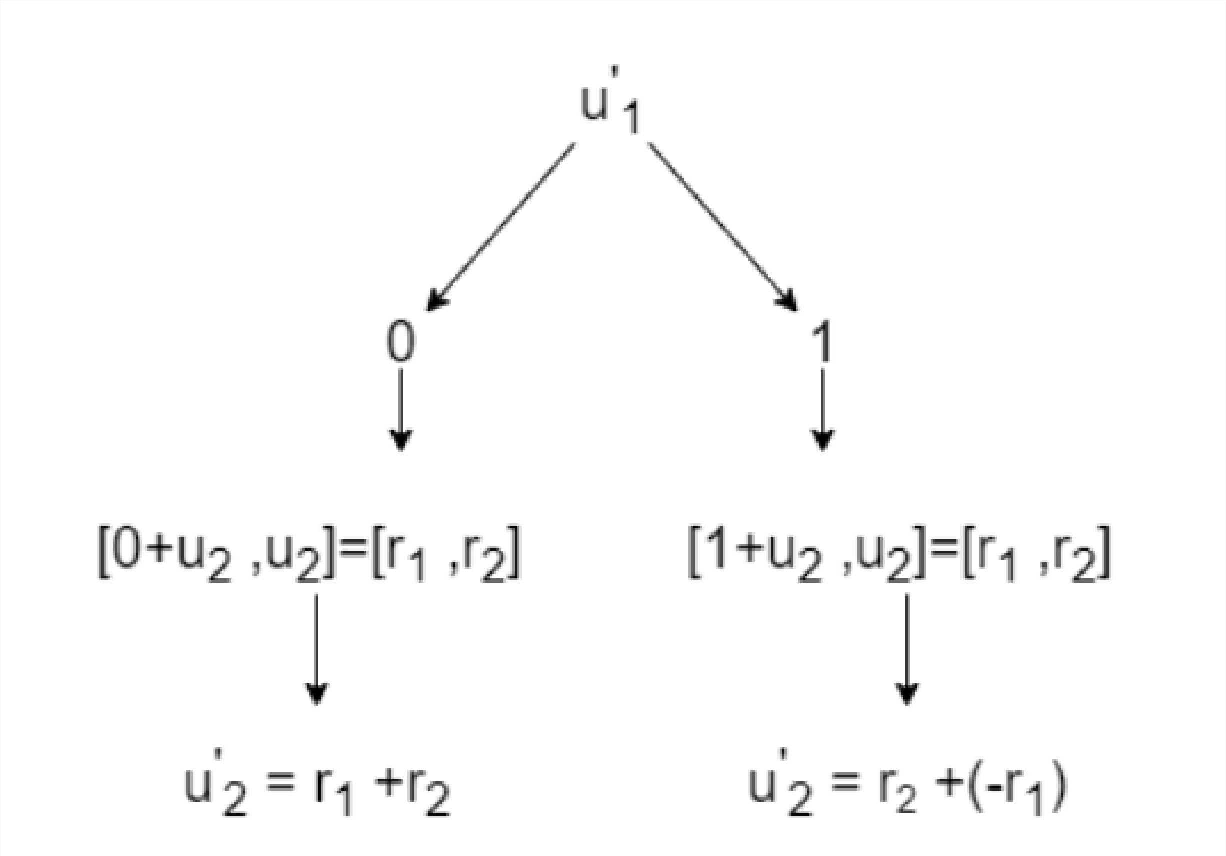
$f(r_1,r_2) = sgn(r_1) . sgn(r_2) . min(|r_1||r_2|)$

Where sgn(.) is the sign function, considering the sign of each belief, and min(.) is the minimum function which in this case is used to get the minimum absolute value of the two beliefs.

Once we get the estimated value of u_1 assuming it is correct,we use $g(r_1,r_2,u_1)$ function to get u_2:

$g(r_1, r_2, u_1) = r_2 + (1 - 2u_1)r_1$

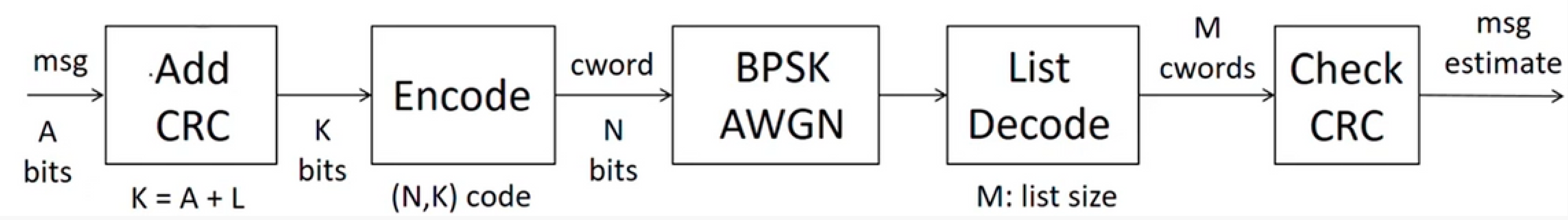
i.e. If $u_1 = 0$ then u_2 will be $r_1 + r_2$ and if $u_1 = 1$ then $u_2 = r_2 - r_1$.



So in the decoding process the complete tree is traversed starting from the root node till all the leaf nodes are decoded. At every node firstly keep going to the left child until leaf node is reached and then use the min sum function to decode the first bit and then pass that bit to its parent and then decode the second bit using the $g(r_1, r_2, u_1)$ function. Then both the beliefs from the left and right child are sent to the parent as $[u_1 + u_2 , u_2]$. These 3 steps are continued until we decode our complete message.

Successive cancellation list decoder:

SC list decoding is an advanced version of SC decoding technique used in decoding of polar codes. SC decoding needed an improvement in performance so SC list decoding produces a list of all possible list of codewords rather than producing a single codeword estimate.

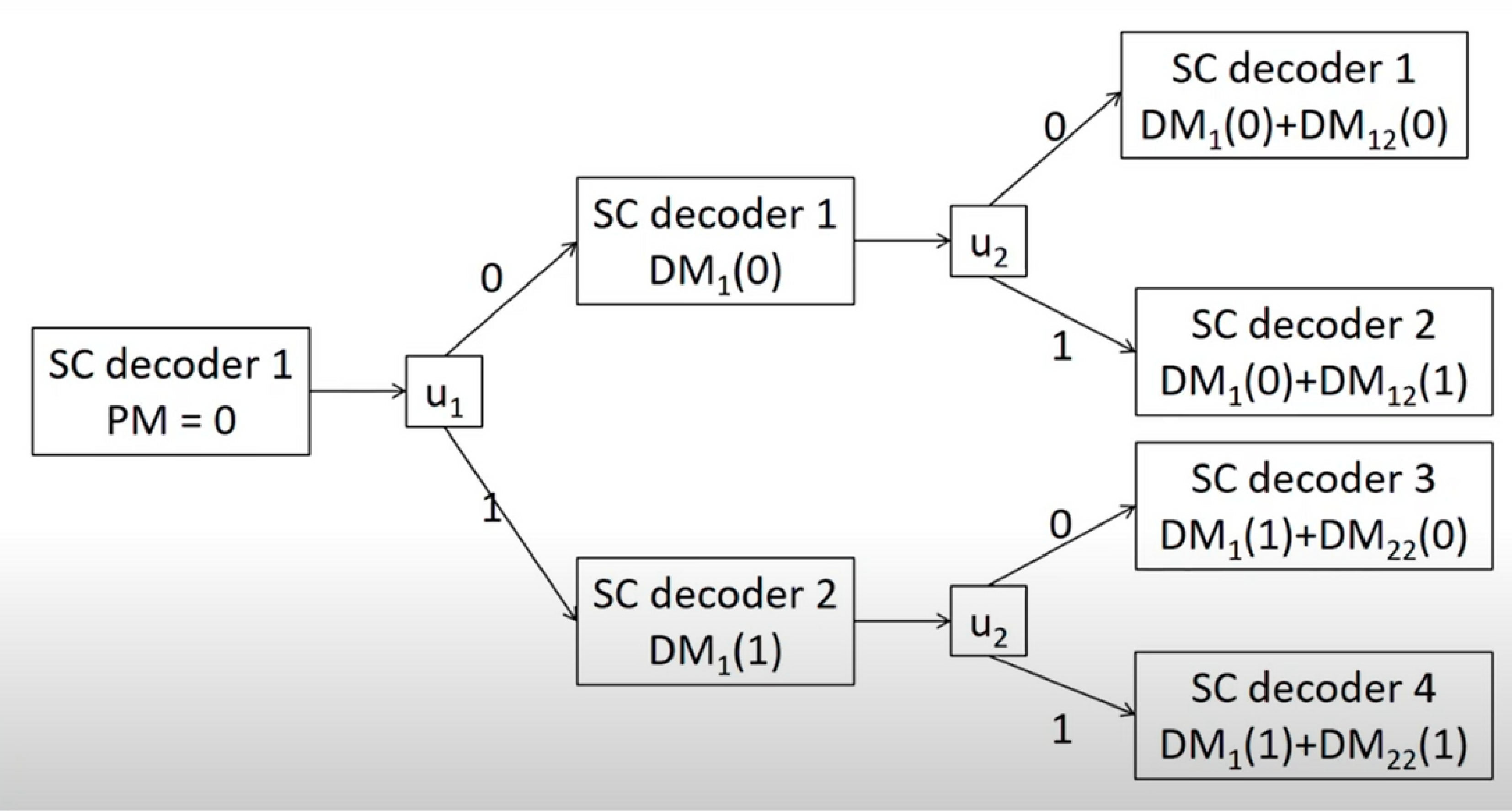


In this decoding algorithm firstly our ‘a’ bits message is appended with a ‘L’ bit CRC bits(cyclic redundancy check bits). The cyclic redundancy check bits are traditionally used to check for which of the M codewords is correct. So the codeword which passes the CRC check at last is the correct decoded message.Computing the CRC has a standard method and for instance the standard documents completely describe how to compute CRC. After adding CRC the message becomes of K bits and this message is then encoded by the standard methods.
List size M tells us that each bit is repeated M times. The greater the M lesser the possibilities of error but the downside is that it takes a lot of time in decoding and also larger memory is required.
In the decoding algorithm firstly there are 2 possibilities: either the bit is frozen or not.
If the bit is frozen then simply its estimated value will be 0 but we need to calculate it’s belief.
If the belief is greater than or equal to 0 then its path metric remains same but if belief is less than 0 and the bit is frozen that means its estimated value according to belief should be 1 but we know it is 0 so we need to add the penalty accordingly in the path metric.
Path metric is the sum of all the decision metric in that particular path.

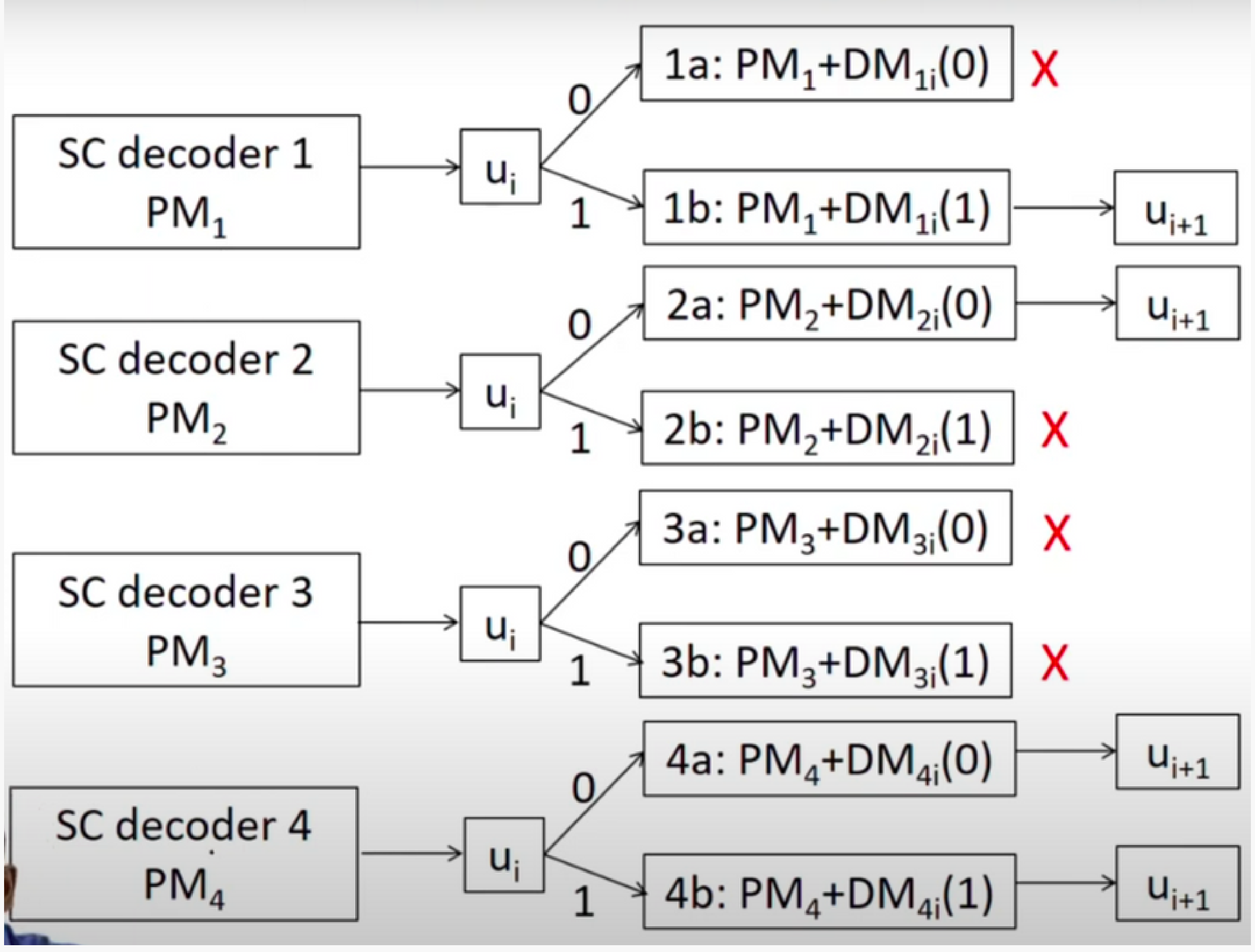
Decision metric is decided in the following way:

if $L(u_i) \geq 0$: $\hat{u}_i = 0$ has $DM_i = 0$, $\hat{u}_i = 1$ has $DM_i = |L(u_i)|$
if $L(u_i) < 0$: $\hat{u}_i = 1$ has $DM_i = 0$, $\hat{u}_i = 0$ has $DM_i = |L(u_i)|$
IMPORTANT: DM assigned even if 'i' is frozen

Let's take an example for list size M = 4:

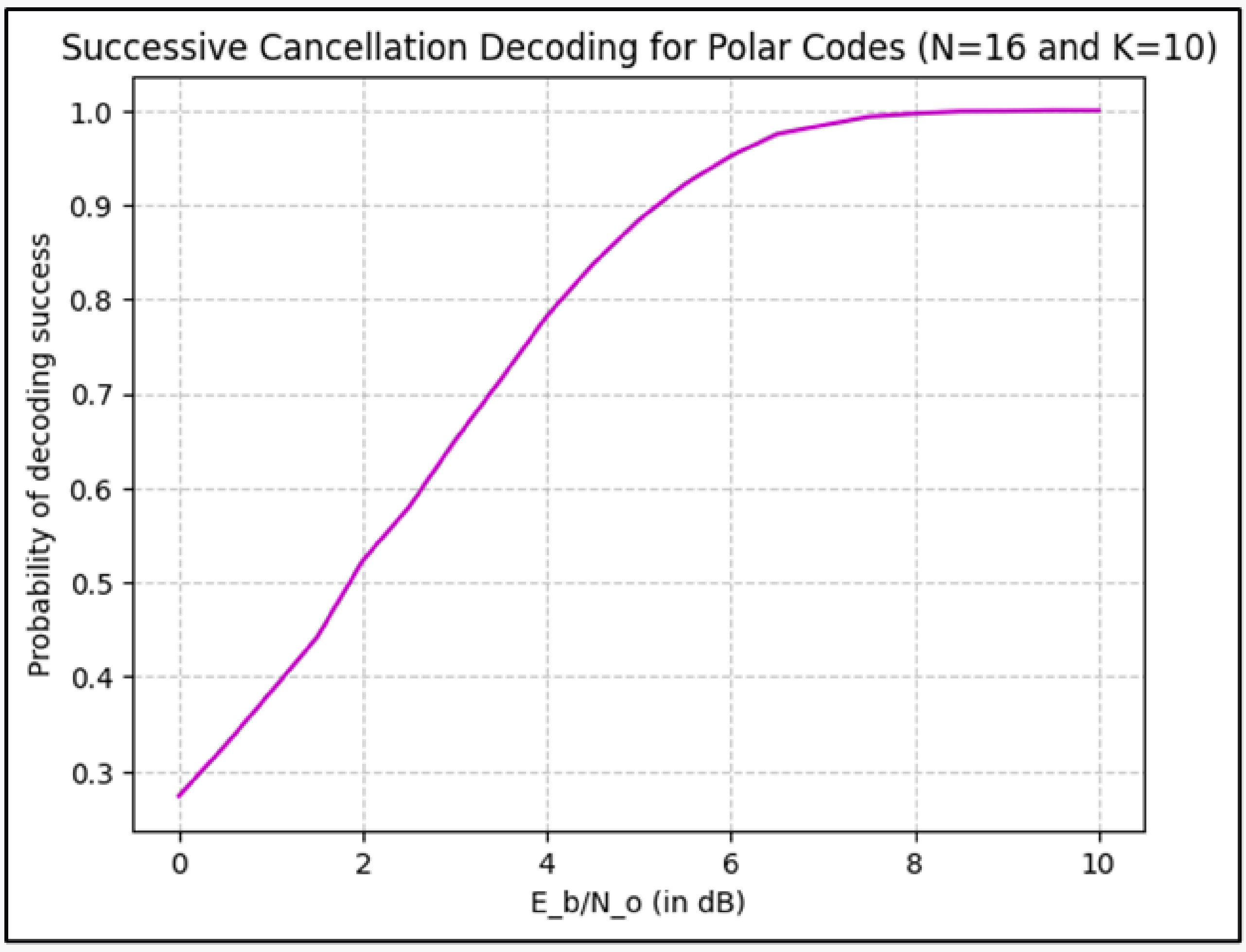


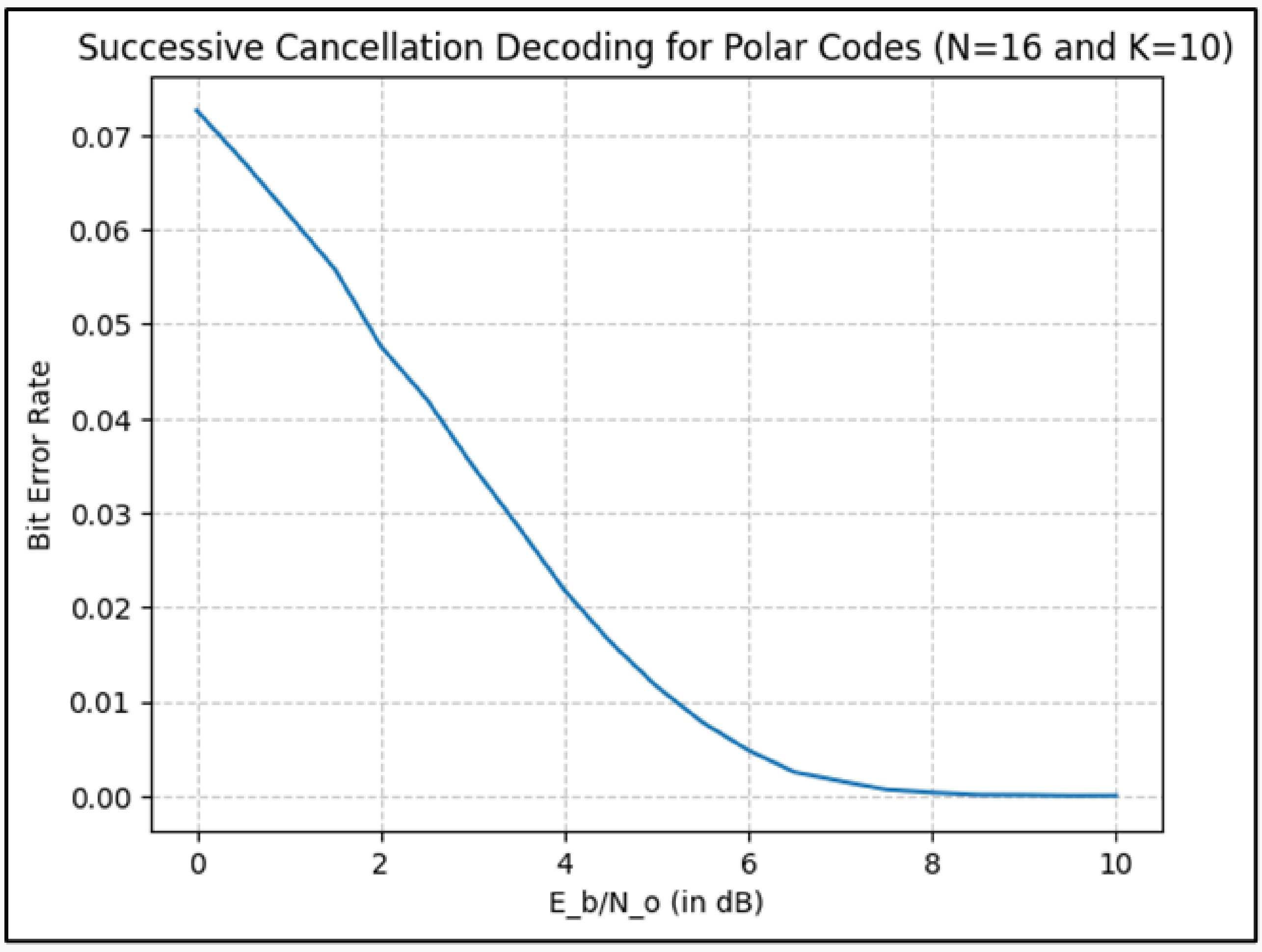
If the bit at any node is not frozen then as it is SC list decoder it considers all possible combinations of decoding and then selects M(here 4) cases with least PM.



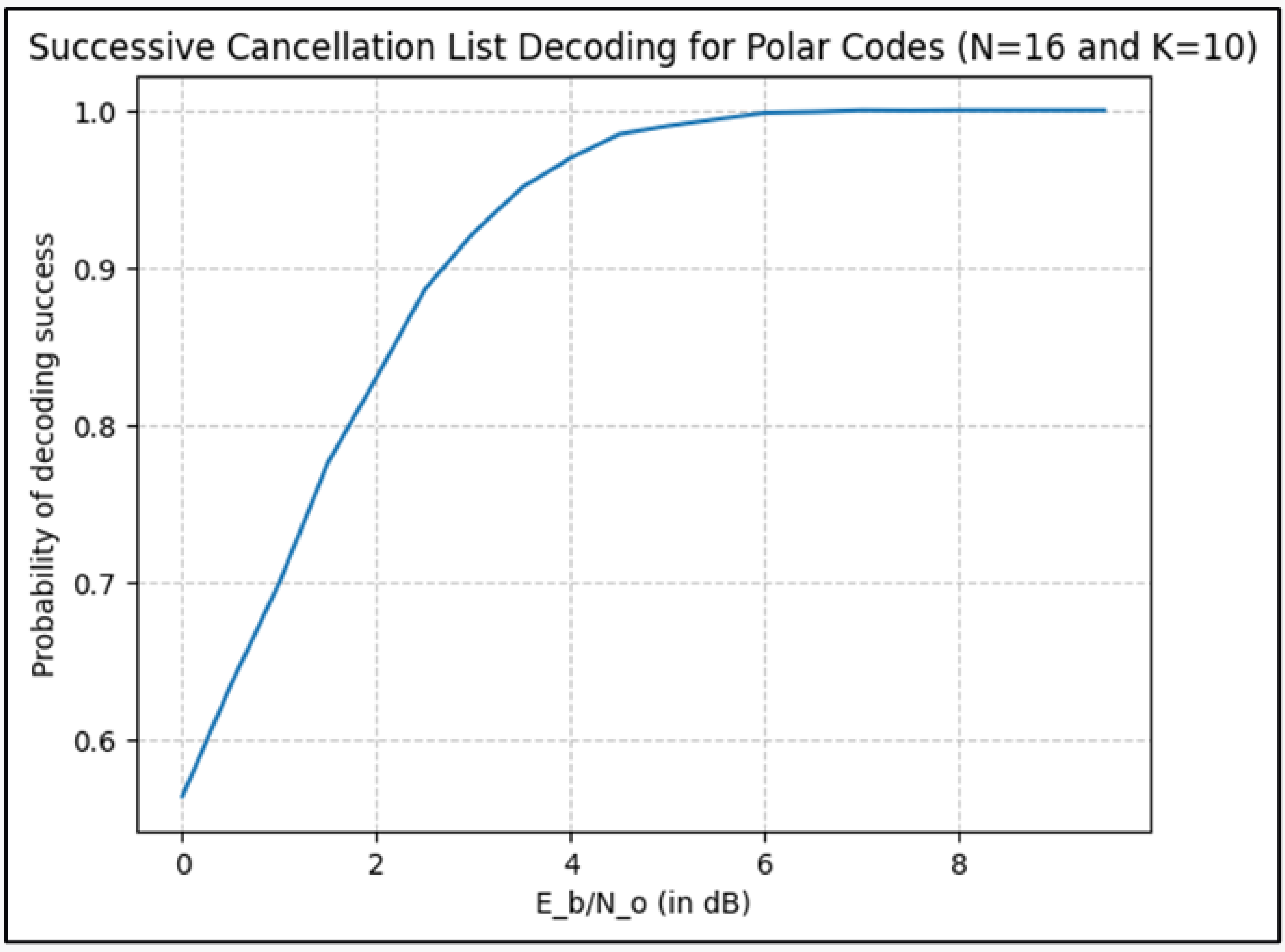
Implementations And Results

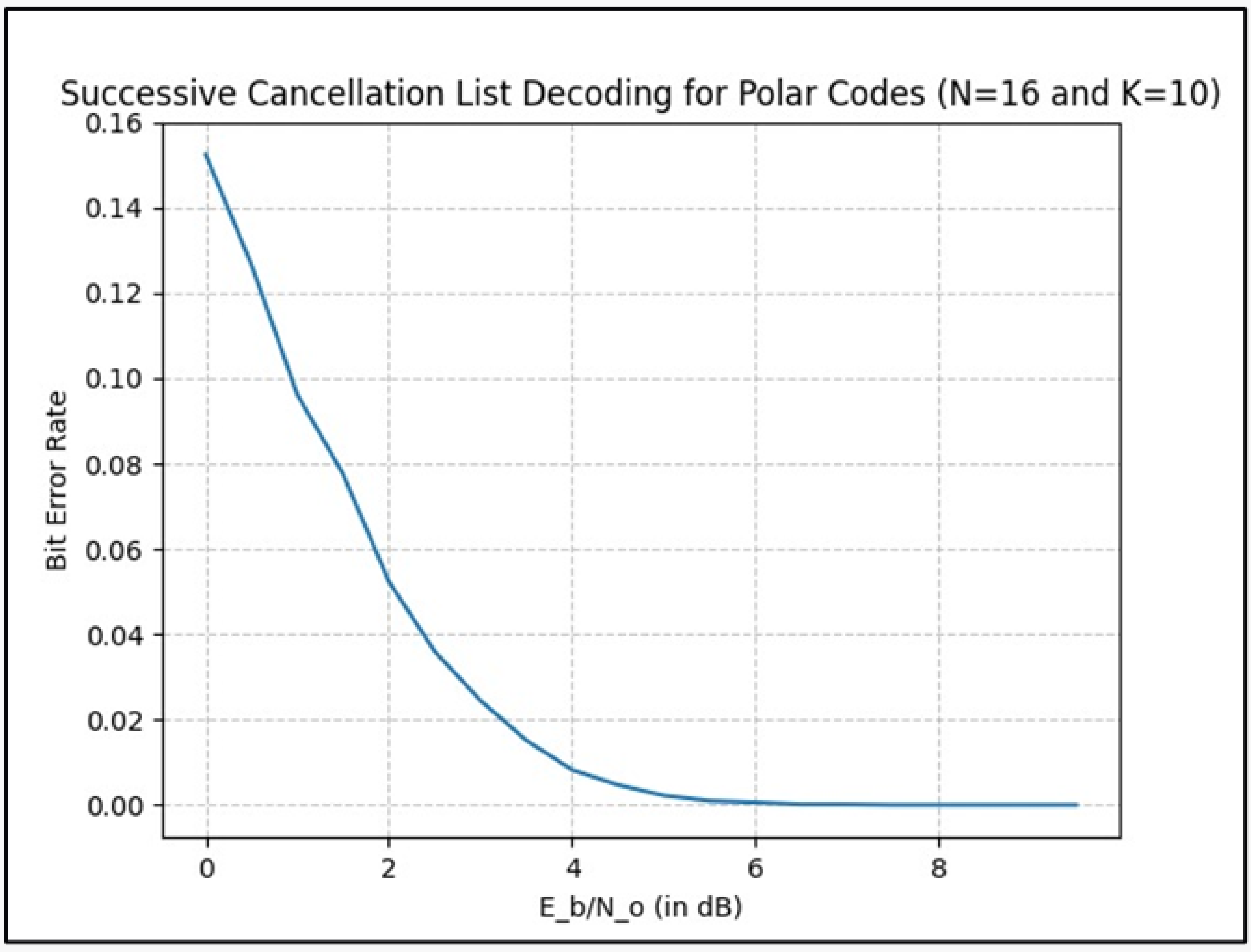
Successive Cancellation(SC) Decoder:



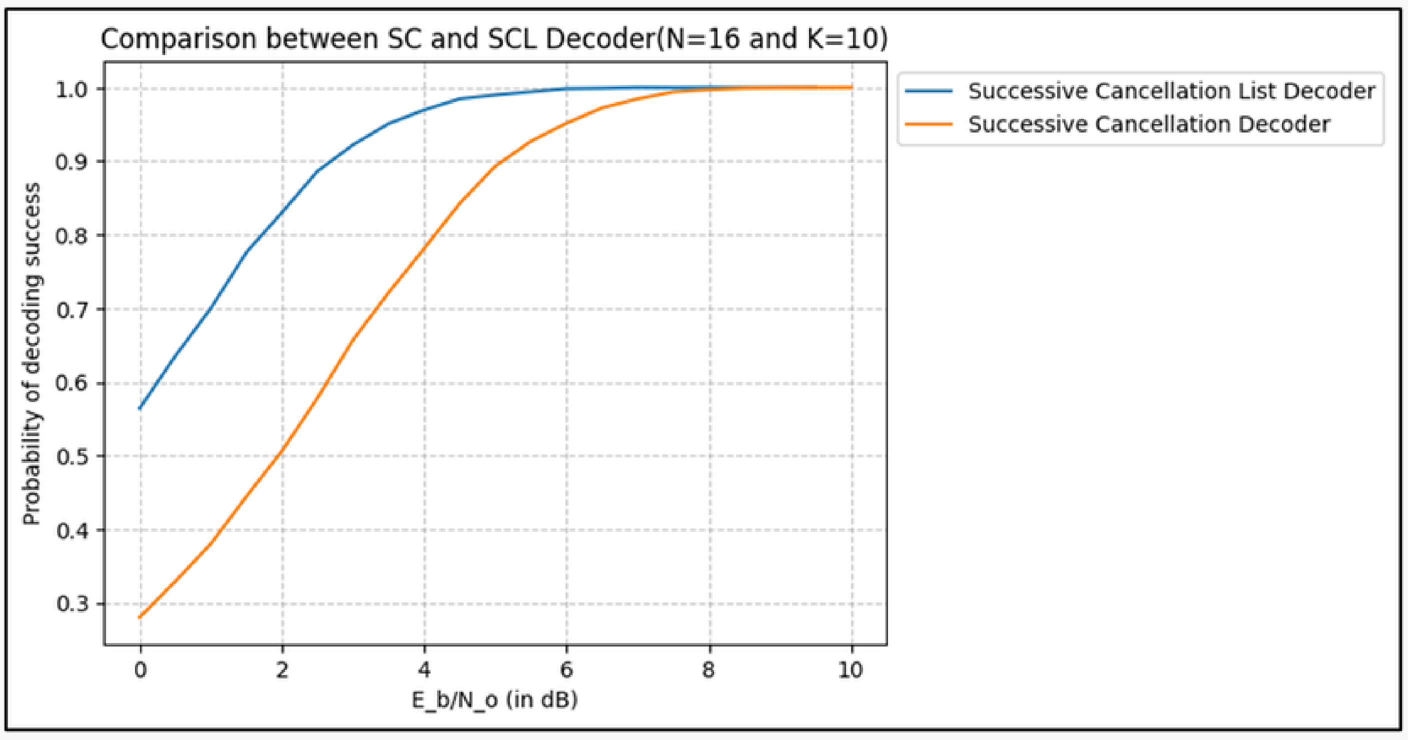


Successive Cancellation(SC) List Decoder:

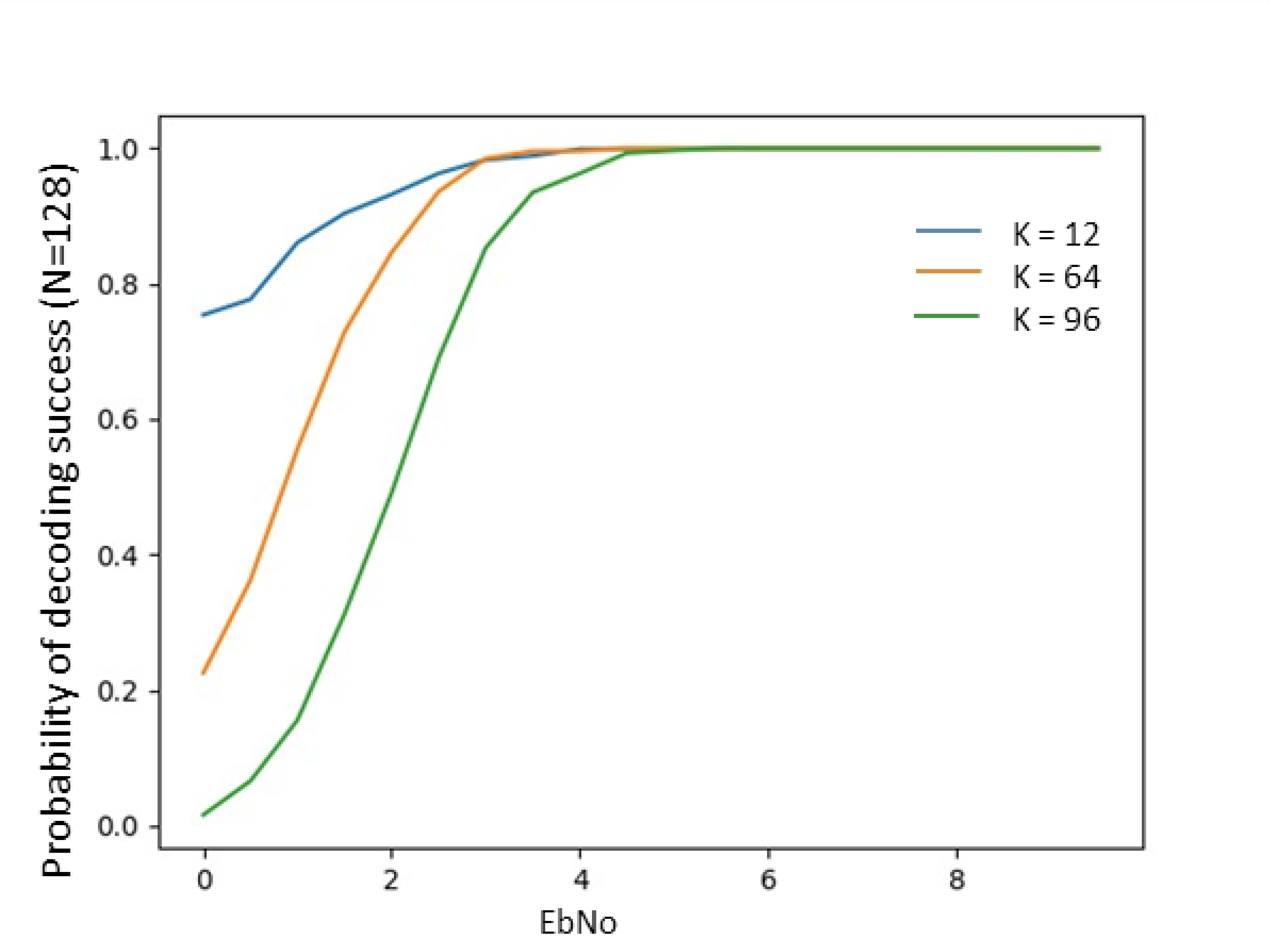


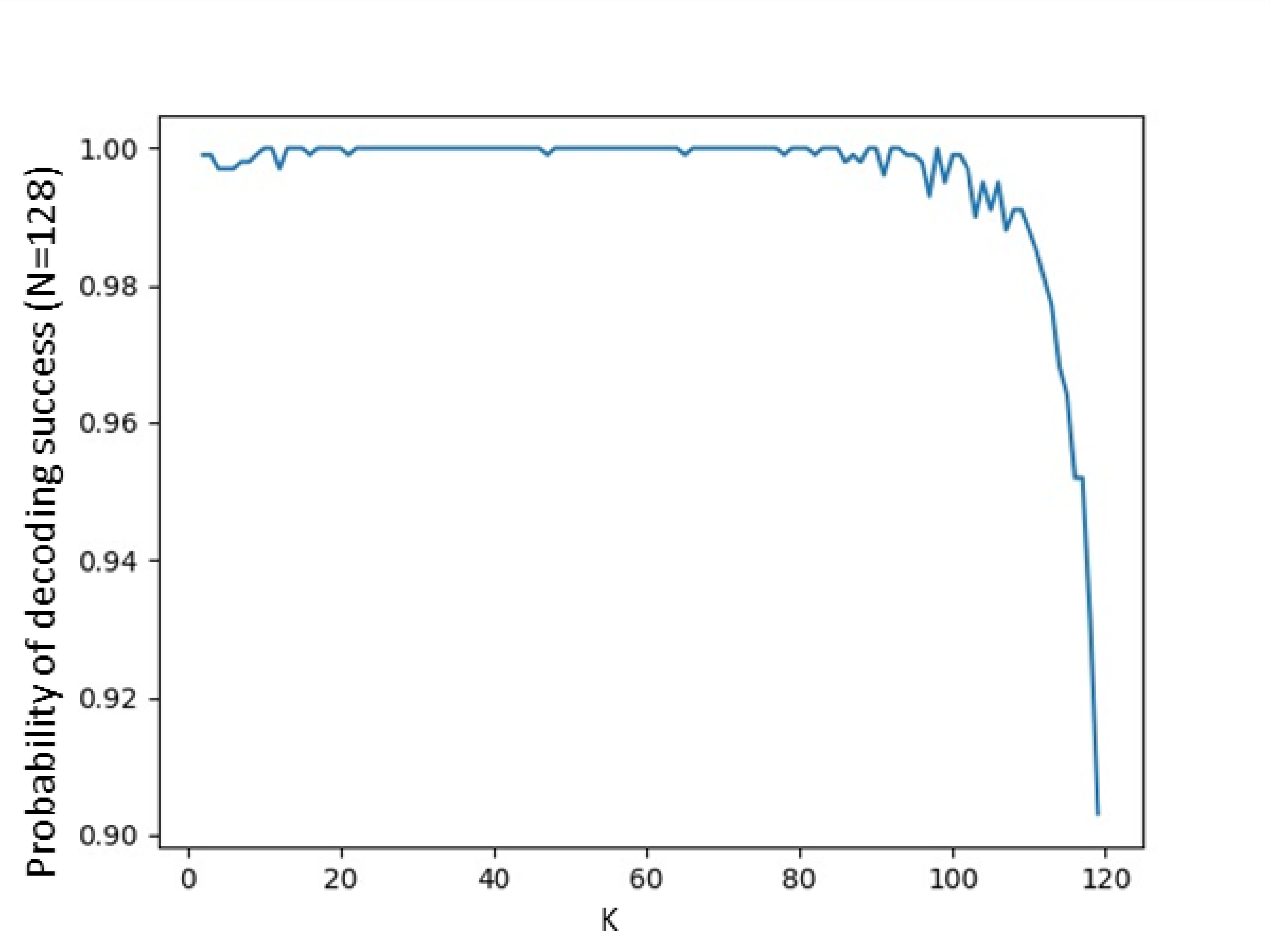


SC vs SC list Decoder



Comparison For Different Code Rates:

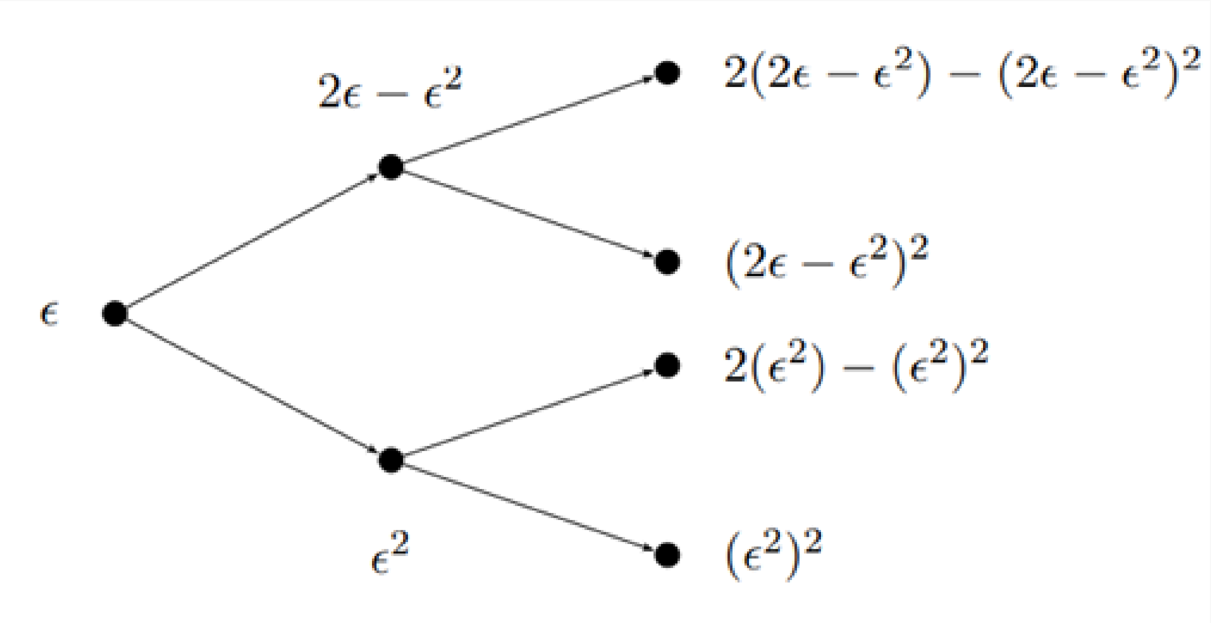




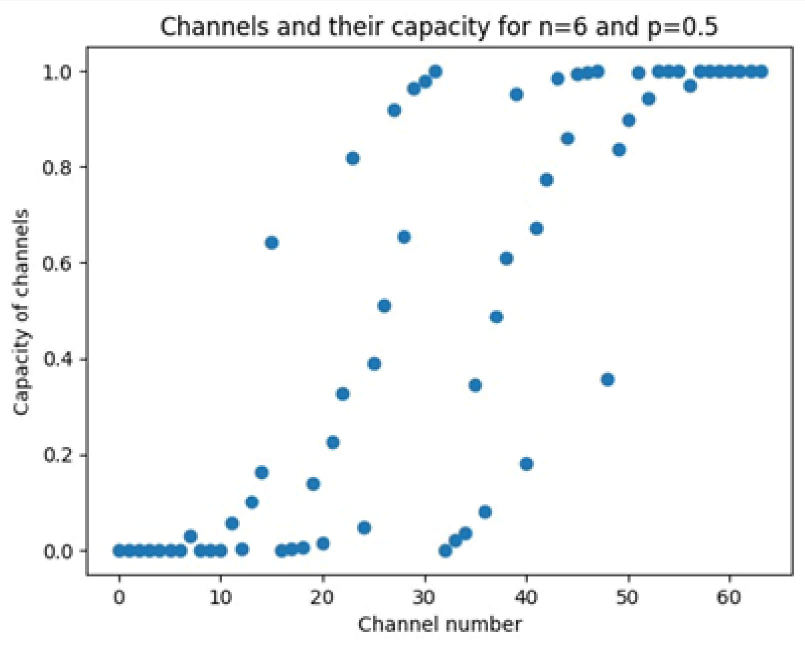
Co.

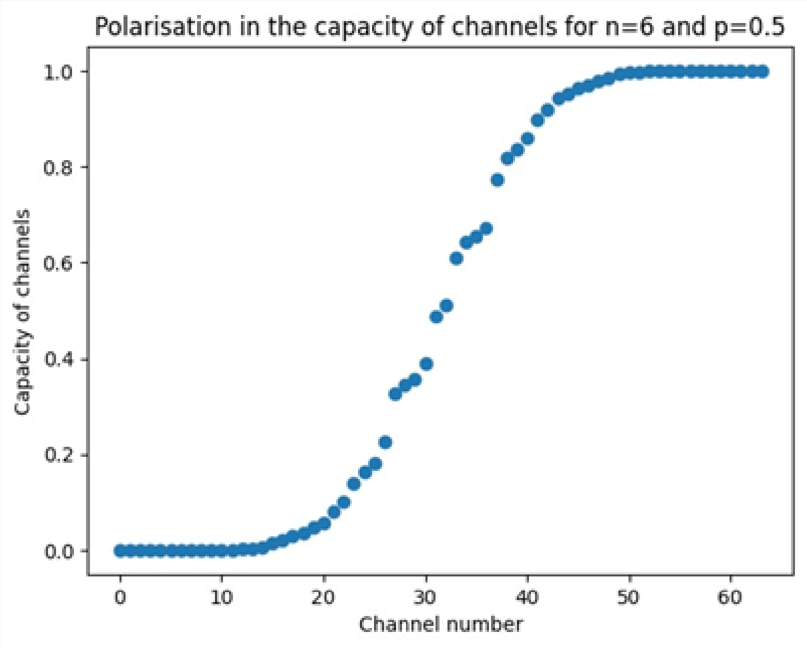
Summary

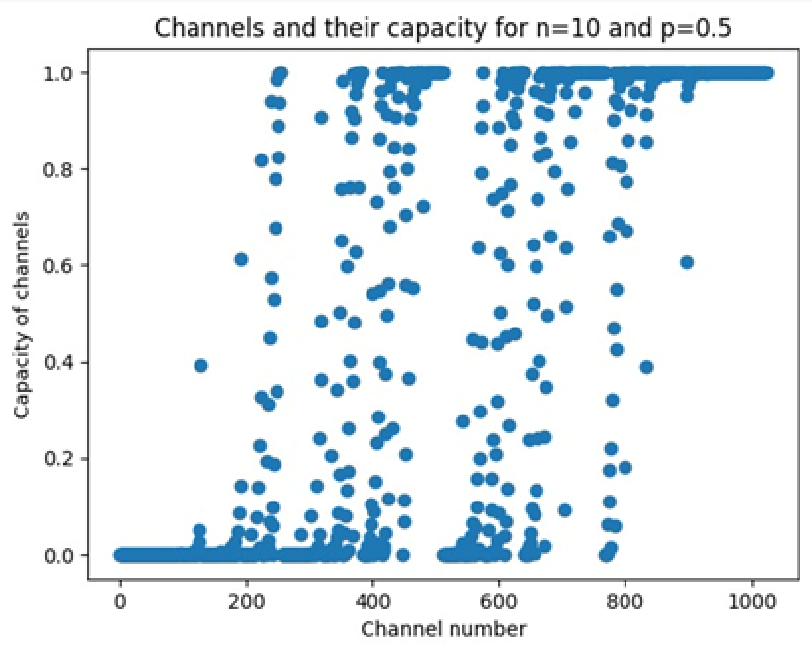
One of the biggest advantage of using Polar codes is that it is successful in achieving shanon's channel capacity bound.
The polar encoding process involves assigning the information bits to the most reliable (noiseless) synthesized channels and fixing the remaining bits (frozen bits) to predetermined values on the less reliable channels. This encoding philosophy contrasts with traditional codes that try to maximize the minimum Hamming distance.
Hence we will understand how polar codes achieves Shannon's Channel Capacity Bound through the effect of polarization
As we had discussed earlier in channel polarization and splitting part, we get two BEC channels w^- and w^+ with probability of error $2p-p^2$ and p^2 respectively. As we increase the number of channels the probability of error changes as shown in the image given:

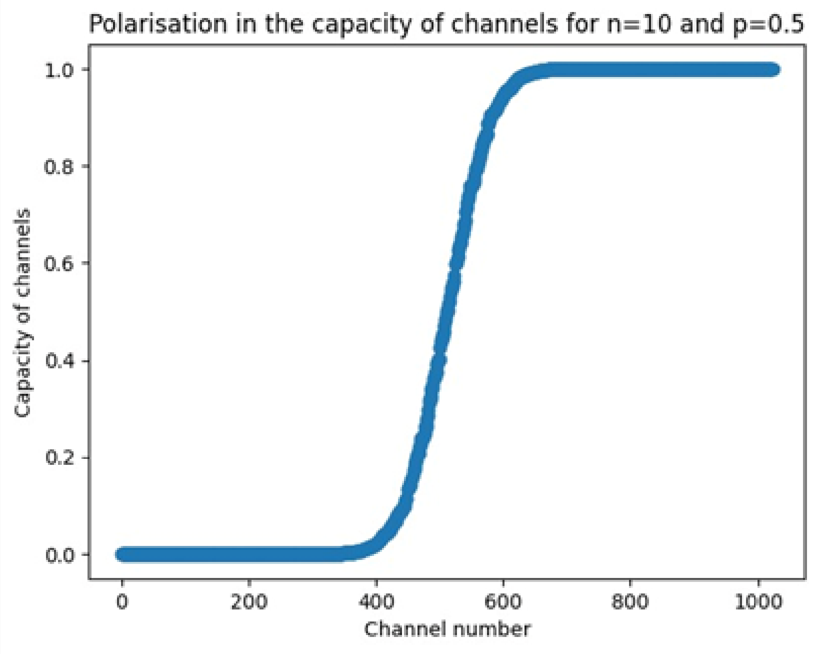


We will observe that as we increase the number of channels the probability of error tends to 1 or 0. Hence when N tends to infinity the channels will be either highly reliable(with probability of error 0) or highly unreliable(with probability of error 1).
We use the highly reliable bit to send message bits and unreliable bits to send predefined frozen bits to send the message reliably.
We know that information I for BEC(p) channel is given by:
$I = 1-p$
Hence as we increase the number of channels the information for reliable channel tends to one and for unreliable channels it tends to zero.
We have shown the simulations for the variation in capacity as we increase the number of channels which is available:









As we can observe from the graphs as number of channels increases the capacity of reliable channels tends to 1 or 0 and the mediocre channels(i.e. channels having capacity between 0 and 1) tend to diminish using Martingale's Convergence Theorem.
This shows that we are able to achieve Shannon's Channel Capacity Bound using polar codes.

Comparison with other codes:

	Encoding		Design and construction	
	Structure	**Complexity**	**Methods**	**Complexity**
Polar	Recursive encoder [1]	$O(N\log N)$ medium	DE [4]	High
			Tal and Vardy [5]	Medium
			GA[3]	Low
Turbo	Convolutional encoder [12]	$O(mN)$ low	Interleaver optimization [12]	High
LDPC	Matrix multiplication [12]	$O(N^2)$ high	Degree distribution optimization [12]	High

	Decoding		
	Algorithms	Complexity	Performance
Polar	SC[1]	$O(N\log N)$ low	Suboptimal
	SCL [8]	$O(LN\log N)$ medium	Approach ML
	BP[1]	$O(I_{max}N\log N)$ high	Suboptimal
	CA-SCL[7]	$O(LN\log N)$ medium	Outperform ML
Turbo	Iterative BCJR	$O(I_{max}(4N2^m))$ high	Approach ML
LDPC	BP	$O(I_{max}(N\bar{d}_v + M\bar{d}_c))$ High	Approach ML

REFERENCES:

https://biblioteca.sbrt.org.br/articlefile/2910.pdf
https://ieeexplore.ieee.org/iel7/6245522/10061655/10061665.pdf
https://archive.nptel.ac.in/courses/117/106/108106137/
http://essay.utwente.nl/82406/1/Miclea_BA_EEMCS.pdf
https://ieeexplore.ieee.org/document/9381073
https://ieeexplore.ieee.org/iel7/18/9760494/09693531.pdf
https://ieeexplore.ieee.org/document/9824746/
http://ieeexplore.ieee.org/document/6852102/
https://ieeexplore.ieee.org/document/9271700
https://www.youtube.com/@SimonsInstituteTOC
https://www.youtube.com/@ISIT-gq4nl