

https://dspace.org/

Soft-decision decoding of permutation codes in AWGN and fading channels

Kolade, Oluwafemi Ibrahim

2017

http://hdl.handle.net/10539/22995

Kolade, Oluwafemi Ibrahim (2017) Soft-decision decoding of permutation codes in AWGN and fading channels, University of the Witwatersrand, Johannesburg, <http://hdl.handle.net/10539/22995>

Downloaded from a DSpace repository. https://dspace.org/



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

SOFT-DECISION DECODING OF PERMUTATION CODES IN AWGN AND FADING CHANNELS

A Dissertation submited in fulfillment of the requirements for the degree of Master of Science in the
School of Electrical and Information Engineering
by

Kolade Oluwafemi Ibrahim

Supervisor: **Prof. DJJ Versfeld**

January, 2017

The financial assistance of the Center for Telecommunications Access and Services (CeTAS) and National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the CeTAS or NRF.

Declaration

I, KOLADE Oluwafemi Ibrahim, declare that this dissertation titled, 'Soft-Decision Decoding of Permutation Codes in AWGN and Fading Channels' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

Acknowlegdments

I am grateful to my family for their encouragement, love and support. Dr. Yinka Ogundile for his priceless advice and my supervisor for his invaluable guidance and support. I would also like to thank Exolve Technologies and above all, give all the glory to the Lord almighty.

UNIVERSITY OF THE WITWATERSRAND

Abstract

Engineering and the Built Environment
School of Electrical and Information Engineering
Master of Science

SOFT-DECISION DECODING OF PERMUTATION CODES IN AWGN AND FADING CHANNELS

by Kolade Oluwafemi Ibrahim
Supervisor: Prof. DJJ Versfeld

Permutation codes provide the required redundancy for error correction in a noisy communication channel. Combined with MFSK modulation, the outcome produces an efficient system reliable in combating background and impulse noise in the communication channel. Part of this can be associated with how the redundancy scales up the amount of frequencies used in transmission.

Permutation coding has also shown to be a good candidate for error correction in harsh channels such as the Powerline Communication channel. Extensive work has been done to construct permutation code books but existing decoding algorithms become impractical for large codebook sizes. This is because the algorithms need to compare the received codeword with all the codewords in the codebook used in encoding.

This research therefore designs an efficient soft-decision decoder of Permutation codes. The decoder's decision mechanism does not require lookup comparison with all the codewords in the codebook. The code construction technique that derives the codebook is also irrelevant to the decoder.

Results compare the decoding algorithm with Hard-decision plus Envelope Detection in the Additive White Gaussian Noise (AWGN) and Rayleigh Fading Channels. The results show that with lesser iterations, improved error correction performance is achieved for high-rate codes. Lower rate codes require additional iterations for significant error correction performance. The decoder also requires much less comuptational complexity compared with existing decoding algorithms.

Contents

Declaration ii

List of Figures ix

List of Tables xii

Abbreviations xiii

1 Introduction 1
- 1.1 Research Question . . . 2
- 1.2 Research Objectives . . . 3
- 1.3 Research Significance . . . 4
- 1.4 Author's Contribution . . . 5

2 Background Review 6
- 2.1 Modulation . . . 6
  - 2.1.1 Frequency Modulation . . . 6
  - 2.1.2 M-ary Frequency Shift Keying (MFSK) . . . 8
- 2.2 Communication Channel . . . 10
  - 2.2.1 AWGN Channel . . . 10
    - 2.2.1.1 Coherent Detection in AWGN Channels . . . 11
    - 2.2.1.2 Noncoherent Detection in AWGN Channels . . . 12
  - 2.2.2 Rayleigh Fading Channel . . . 15
    - 2.2.2.1 Frequency-nonselective Slowly Fading Channel . . . 15
- 2.3 Permutation Codes . . . 18
  - 2.3.1 Definition 1 . . . 19
  - 2.3.2 Definition 2 . . . 19
  - 2.3.3 Definition 3 . . . 19
  - 2.3.4 Definition 4 . . . 20
  - 2.3.5 Types of Permutation Codes . . . 20
    - 2.3.5.1 Variant 1 . . . 20
    - 2.3.5.2 Variant 2 . . . 20
- 2.4 The Assignment Problem . . . 21

2.4.1 Example 2.1 . . . 21
2.4.2 The Hungarian Algorithm . . . 22
2.4.3 Definition 5 . . . 23
2.4.4 Example 2.2 . . . 23
2.5 Murty's Algorithm . . . 25
2.6 The k-th Assignment Problem . . . 25
2.6.1 Definition 6 . . . 27
2.6.2 Example 2.3 . . . 27
2.7 Maximum Likelihood . . . 28

3 Research Methodology 30
3.1 Experimental Setup . . . 32
3.2 System Description . . . 32
3.3 The Hungarian Algorithm as a Soft-Decision Decoder . . . 34
3.4 Murty's Algorithm as a Soft-Decision Decoder . . . 37

4 Results 41
4.1 4FSK in AWGN Channel - Coherent Detection . . . 43
4.1.1 4 Codewords . . . 43
4.1.2 8 Codewords . . . 43
4.1.3 16 Codewords . . . 44
4.1.4 24 Codewords . . . 46
4.2 8FSK in AWGN Channel - Coherent Detection . . . 49
4.2.1 8 Codewords . . . 49
4.2.2 305 codewords . . . 50
4.2.3 1417 codewords . . . 50
4.2.4 20160 codewords . . . 51
4.2.5 40320 codewords . . . 51
4.3 MFSK in AWGN Channel - Noncoherent Detection . . . 54
4.3.1 4 Codewords . . . 55
4.3.2 8 Codewords . . . 55
4.3.3 16 Codewords . . . 56
4.3.4 24 Codewords . . . 56
4.4 8FSK . . . 59
4.4.1 8 Codewords . . . 60
4.4.2 305 codewords . . . 60
4.4.3 1417 codewords . . . 61
4.4.4 20160 codewords . . . 62
4.4.5 40320 codewords . . . 62
4.5 Noncoherent MFSK in AWGN and Rayleigh Fading Channels . . . 64
4.5.1 4 Codewords . . . 65
4.5.2 8 Codewords . . . 65
4.5.3 16 Codewords . . . 66
4.5.4 24 Codewords . . . 66

4.6 8FSK 67
4.6.1 8 Codewords 69
4.6.2 305 codewords 70
4.6.3 1417 codewords 70
4.6.4 20160 codewords 71
4.6.5 40320 codewords 71

5 Conclusion and Recommendations **78**
5.1 Conclusion 78
5.2 Recommendations 79

Bibliography **81**

List of Figures

2.1 Analog Signal Cosine Waveform 7
2.2 Sample 2FSK transmitting 0111101111 8
2.3 Block Diagram of System with AWGN Channel Model 11
2.4 Envelope Detector in AWGN Channel 13
2.5 Noncoherent Detection of MFSK - Symbol Error Rate vs SNR for Theoretical and Monte Carlo 14
2.6 Block Diagram of System with AWGN & Rayleigh Fading Channels . 16
2.7 Block Diagram of MFSK Noncoherent Detector 16
2.8 Trial and Error Method of Computing Costs for given $\sqrt{n}$ 29

3.1 Experiment Block Diagram 33

4.1 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, , Coherent Detection. $d_{min} = 4$, code rate = 0.25 44
4.2 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN, Coherent Detection. $d_{min} = 3$, code rate = 0.375 45
4.3 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, , Coherent Detection. $d_{min} = 2$, code rate = 0.5 46
4.4 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 2$, code rate = 0.57 47
4.5 SNR (dB) Gain vs $\frac{|C|}{|P|}$ for 4 Iterations for $|P| = 24$, 4FSK, Coherent Detection 48
4.6 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 8$ 50
4.7 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 5$ 51
4.8 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 4$ 52
4.9 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 3$ 53
4.10 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 2$ 54

4.11 SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, Noncoherent Detection. 55
4.12 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding. $d_{min} = 4$, code rate = 0.25 56
4.13 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 3$, code rate = 0.375 57
4.14 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 2$, code rate = 0.5 58
4.15 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 2$, code rate = 1 59
4.16 SNR (dB) Gain vs $\frac{|C|}{|P|}$ for First 4 Iterations for $|P| = 24$, 4FSK Modulation, Noncoherent Detection 60
4.17 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 8$ . . . 61
4.18 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 5$ . . . 62
4.19 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 4$ . . . 63
4.20 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 3$ . . . 64
4.21 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Noncoherent Detection. $d_{min} = 2$ 65
4.22 SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, Noncoherent Detection. 66
4.23 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 4$, code rate = 0.25 67
4.24 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 3$, code rate = 0.25 68
4.25 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$, code rate = 0.25 69
4.26 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$, code rate = 0.25 70
4.27 SNR (dB) Gain vs $|C|/|P|$ for First 4 Iterations for $|P| = 24$, 4FSK Modulation, AWGN and Rayleigh Fading Channels 71
4.28 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 8$ 72

4.29 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 5$ 73
4.30 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 4$ 74
4.31 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 3$ 75
4.32 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$ 76
4.33 SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, AWGN and Rayleigh Channels 77

List of Tables

4.1 Performance of Soft-decision Decoder using 4FSK in AWGN Channel, Coherent Detection 47
4.2 Performance of Soft-decision Decoder using 8FSK in AWGN Channel, Coherent Detection 53
4.3 Performance of Soft-decision Decoder using 4FSK in AWGN Channel, Noncoherent Detection 57
4.4 Performance of Soft-decision Decoder using 8FSK in AWGN Channel, Noncoherent Detection 63
4.5 Performance of Soft-decision Decoder using 4FSK in AWGN and Rayleigh Fading Channels 68
4.6 Performance of Soft-decision Decoder using 8FSK in AWGN and Rayleigh Fading Channels 73

Abbreviations

AWGN	**A**dditive **W**hite **G**aussian **C**hannel
CENELEC	**Comite E**uropeen de **N**ormalisation **Elec**trotechnique
dB	**D**ecibel
ED	**E**nvelope **D**etection
FEC	**F**orward **E**rror **C**orrection
FM	**F**requency **M**odulation
HA	**H**ungarian **A**lgorithm
HD	**H**ard **D**ecision
JVA	**J**onker-**V**olgenant **A**lgorithm
MA	**M**urty's **A**lgorithm
M**FSK**	**M**-ary **F**requency **S**hift **K**eying
PLC	**P**ower **L**ine **C**ommunication
SD	**S**oft **D**ecision
SER	**S**ymbol **E**rror **R**ate
SNR	**S**ignal to **N**oise **R**atio

Chapter 1

Introduction

In Communication systems, quality of information at the receiver is largely dependent on the conditions in the channel used during the transmission process. The Powerline Communication (PLC) channel for example, is a harsh channel affected by fading, background noise, narrowband and broadband noise. M-ary Frequency Shift Keying (MFSK) modulation spreads the information message over more timeslots in the frequency spectrum. When MFSK is combined with coding, the frequency spreading and time spreading properties provide better resistance to errors in channels with frequency disturbance and impulse noise [1]. MFSK provides constant envelope modulation [2] and assuming all the possible transmitted signals are of equal energy, Envelope Detection is used to detect and demodulate the signal. This requires a bank of M correlators to correlate the received signal with all possible transmitted signals. The output of the correlator will be the signal with the highest correlation value [2].

MFSK has proved to be a good candidate for modulation in PLC with Mengi and Vinck using Reed-Solomon coding with MFSK (that conforms with the CENELEC Band) to achieve considerably large SNR gain [3]. While Vinck [4] showed the effects of different noise conditions on MFSK signals, Vinck, Haring and Wadayama [5] also showed that Permutation coding with MFSK can handle narrowband, impulse and background noise within a certain distance of transmission. The combination of

MFSK with Permutation codes increases the number of frequencies mapped to each message in the sequence, thus providing frequency spreading that helps avoid bad portions of the channel. This has been shown to be an efficient way of combating impulse and narrowband noise [6].

Shum [7] showed that Permutation codes give better performance than when the message is not encoded, with a gain of about 2dB at a bit error rate of 10^{-5}. He further showed that Permutation block codes give better bit error rate performance when compared with convolutional codes soft-decoded with the Viterbi algorithm in some signal-to-noise ratio regions. Vinck [4] was also able to show the error correction capabilities of Permutation codes and used a simple soft-decision decoder that produces a 1 or 0 for values above a threshold or otherwise respectively. Chee and Purkayastha [8], Swart and Ferreira [9] further showed efficient ways of decoding Permutation codes but codes must have been constructed from distance-preserving or distance-increasing mapping algorithms.

With extensive construction of Permutation codes found in [10] [11] [12] [13] [14], there is no efficient soft-decision decoder of permutation codes. The decoder presented in Chee and Purkayastha [8] depends on the encoding algorithm while Bali and Rebai [15] present the maximum likelihood decoding performance of permutation codes. The soft-decision decoder designed in this research is however relevant irrespective of the code construction algorithm and remains relatively efficient for large codebooks.

1.1 Research Question

In order to design a soft-decision decoder with improved computational complexity that can decode large Permutation codebooks, this research aims to answer the following question:

1. Can a soft-decision decoder of Permutation Codes be designed using the Hungarian Algorithm?

- How can the performance of the Hungarian algorithm as a soft-decision decoder be improved?

- For codebook C, if the maximum assignment solution $A \notin C$, what intelligent decision can be made in order to correct more errors at the receiver?

- Can the next highest assignment cost improve the performance of the Hungarian Algorithm?

- If the above is true, at what point of iteration does the next highest assignment stop improving the decoder's performance?

1.2 Research Objectives

In this study, we introduce an efficient soft-decision decoder of Permutation block codes in a one-to-one symbol-to-codeword mapping system. Each codeword is a Permutation of k-different positive integers, each integer appearing only once in each codeword. Given a Permutation codebook of all possible codewords P, we select C, a subset of P to encode the message. We modulate with MFSK and evaluate the performance of a soft-decision decoder that implements the Hungarian Algorithm [16] for maximum assignment to decode the received noisy signals. The channel noise for this research is assumed to be either the Additive White Gaussian Noise (AWGN) and Rayleigh Fading or a combination of both channels.

The AWGN channel, known to be one of the first channel impairments is a model useful in studying deep space channels [2]. Noise is also common in many communication channels and therefore important to understand its contribution to communication systems.

The Rayleigh Fading channel on the other hand, is a multipath medium with properties that are applicable in radio communication channel models that experience ionospheric and tropospheric scattering [2]. The impairments experienced in this channel are found in everyday activities which include scattering as a result of moving and stationary objects such as vehicles and trees respectively.

The objectives of this research are:

1. Design a soft-decision decoder for efficiently decoding Permutation codes with codebooks of relatively large sizes
2. Reduce computational complexity of decoding Permutation codebooks.
3. Improve the error correction performance of Permutation codes in channels such as the AWGN and Rayleigh Fading Channel by improving on the coding gain at the receiver.
4. Investigate the Hungarian algorithm for maximum assignment and its characteristics in order to determine if the algorithm can be adapted to decode Permutation block codes.
5. Investigate and determine if the next cost assignments can improve the performance of the maximum assignment algorithm.
6. Recommend the point/iteration at which the decoder stops improving in performance, putting into consideration different code rates and codebooks.

1.3 Research Significance

This significance of this research is to contribute:

1. A soft-decision decoder that efficiently decodes and achieves considerably large coding gain of Permutation codes in the AWGN and Rayleigh Fading channels.
2. Current decoding algorithms for Permutation codes quickly become impractical as the codebook size increases. The soft-decision decoder remains practical for large codebooks with considerable computational complexity.
3. An improvement in the error correction performance and a reduction in the complexity involved in systems when encoding with Permutation codes.

1.4 Author's Contribution

Experiments were carried out by the author to solve the research objectives. The solution by the author produced a soft-decision decoder that combines the Hungarian algorithm and Murty's algorithm. The author leveraged on existing modulation systems, encoding methods and mathematical algorithms in order to achieve these objectives. The research is built upon the following systems and algorithms which were modelled using MATLAB:

1. MFSK
2. Permutation Codes construction
3. AWGN Channel
4. Rayleigh Fading Channel
5. Envelope Detector
6. Minimum Distance Decoder
7. Maximum Likelihood Decoder
8. The Hungarian Algorithm Decoder
9. Murty's Algorithm Decoder

In this dissertation, Chapter 2 discusses existing literature upon which the research is built such as MFSK modulation, Permutation codes, Hungarian and Murty's algorithm. Chapter 3 explains the experiment's methodology, including how the adopted methods solve the research problem and eventually describes the author's contribution. The results from the simulations are discussed in Chapter 4 while the conclusion and recommendations are discussed in Chapter 5.

Chapter 2

Background Review

2.1 Modulation

In analog and digital transmission of data through channels, the data is converted to a form that is suitable for the channel to transmit the data, a process generally known as modulation. The type of modulation technique used usually depends on the nature of the channel, the overall design of the system among other factors. The amount of bandwidth available for example, can influence the choice of the modulation technique.

2.1.1 Frequency Modulation

Frequency Modulation is an analog process transmitting data through a channel. Generally, a signal waveform can be represented as

$$s(t) = Acos\left(2\pi f_c t + \theta\right), \tag{2.1}$$

with amplitude A, carrier frequency f_c and phase θ. A simple signal waveform from 2.1 is shown in Fig. 2.1

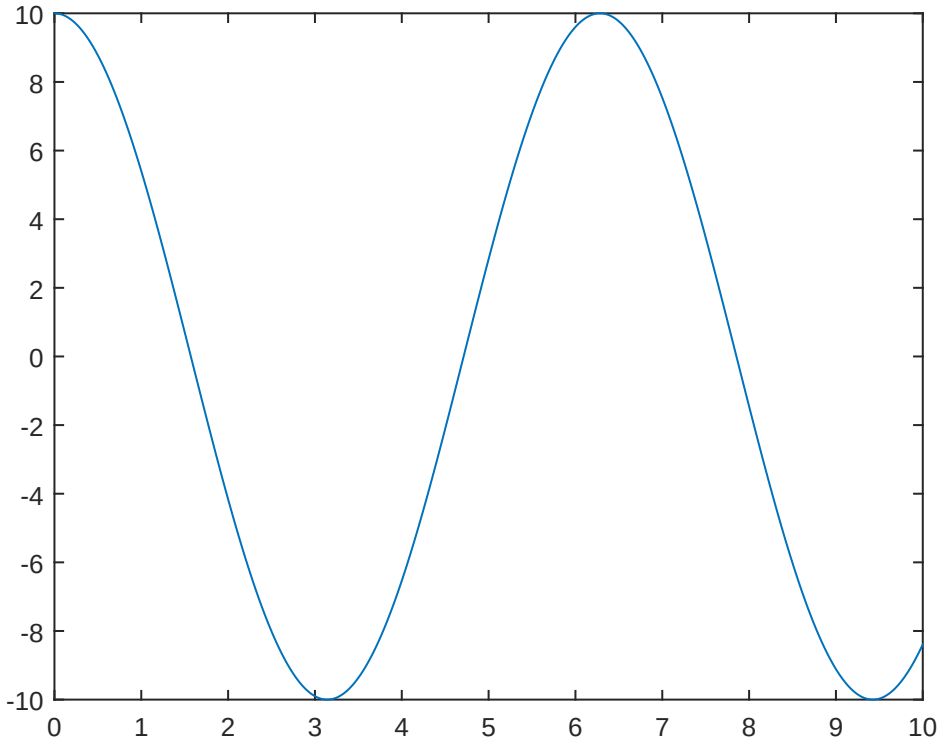

FIGURE 2.1: Analog Signal Cosine Waveform

The frequency of the signal to be transmitted is varied while keeping the amplitude constant in frequency modulation. The information message is mapped onto a carrier frequency thereby using the carrier frequency to transmit the message from point to point. The selected carrier frequency depends on the message that is transmitted at the instance with respect to the mapping between the frequencies and the message. The carrier frequency is given by

$$f_i(t) = f_c + k_f m(t), \tag{2.2}$$

where $m(t)$ is the baseband transmitted signal and k_f is a constant or scaling factor to determine the change in frequency in the process of selecting frequency for transmission.

The digital equivalent of analog frequency modulation can be simply explained by using 2 frequencies, f_1 and f_2. A sample signal waveform of a digital message

0111101111 is shown in Fig. 2.2. The mapping here is such that the symbol mapped to frequency f_1 is mapped to 0 while the symbol mapped to frequency f_2 is mapped to 1. Therefore, the signal changes with respect to the incoming message from the transmitter.

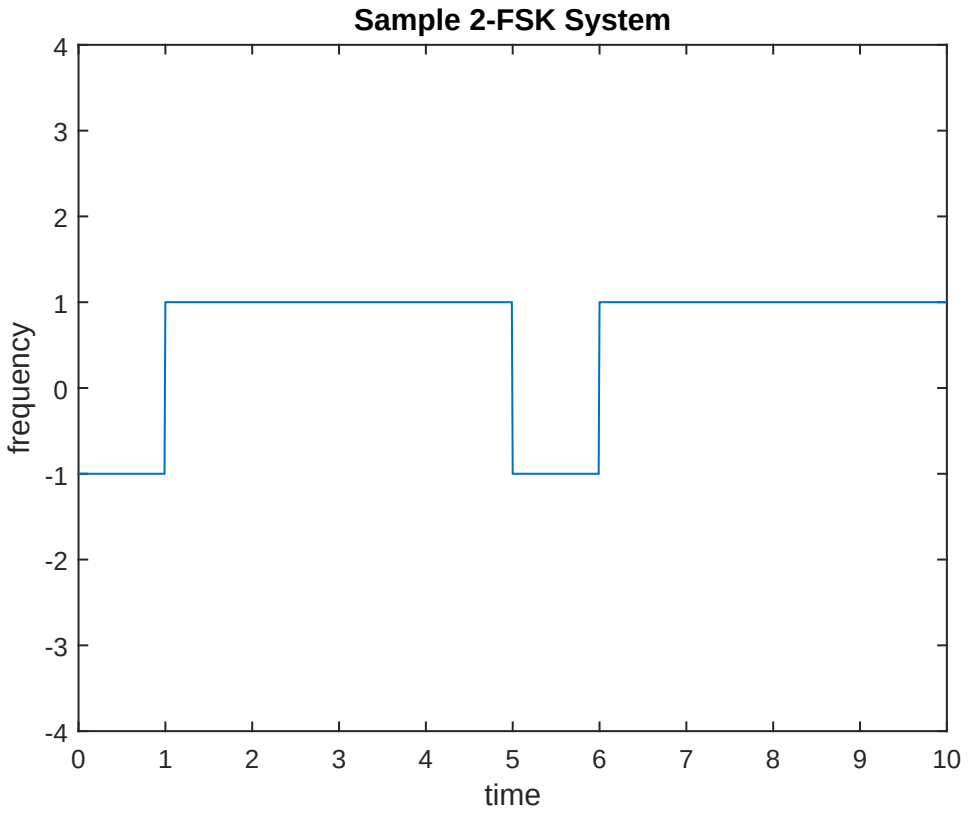

FIGURE 2.2: Sample 2FSK transmitting 0111101111

2.1.2 *M*-ary Frequency Shift Keying (*M*FSK)

M-ary Frequency Shift Keying (*M*FSK) is the digital equivalent of FM. Therefore, it is suitable for modulating digital signals that require transmission in digital channels. The same concept as FM however remains. A set of selected frequencies are mapped onto the set of messages to be transmitted. Frequency switching therefore happens each time a different message is to be transmitted. It can be seen as similar to frequency hopping in its simplest form. The M in *M*FSK is the number of frequencies used and is 2^k.

Given that the desired frequency was transmitted with energy E_s, the function of the noncoherent M-FSK detector is to choose the most likely frequency from a set of M frequencies, by choosing the one with the highest energy present at a sampling instance T. The SNR for such a system is calculated as $SNR = E_s/N_0$ (refer to [2]). The noncoherent M-FSK detector consists of a bank of M pairs of quadrature correlators, one pair for each frequency to be detected.

The output of each quadrature pair is a metric, which is calculated using the square law. Each metric corresponds to each possible frequency. The most likely transmitted symbol for sampling instance T is determined based on these M metrics. The symbol corresponding to the metric with the highest value is chosen as the candidate for envelope detection [2].

MFSK is an orthogonal signaling system and in this research, we assume the signals are of equal energy. Therefore, the signals can be represented using vectors as follows:

$$\begin{aligned} s_1 &= (\sqrt{E}, 0, 0, 0) \\ s_2 &= (0, \sqrt{E}, 0, 0) \\ \vdots &= \quad \vdots \\ s_M &= (0, 0, 0\sqrt{E}) \end{aligned}, \tag{2.3}$$

or

$$s_i = \begin{cases} \sqrt{E}, & \text{if } f = f_i \\ 0, & \text{otherwise,} \end{cases}, \tag{2.4}$$

where E is the symbol energy. Energy is found only in the frequency that is transmitted while other frequencies are 0 at that instance.

Consider a sample random message sequence $i = (2, 3, 4, 1, 2)$ to be modulated before transmitting across a channel. Assuming the energy in the signal is 1, the MFSK representation using (2.3) is

$$
\mathrm{S} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{array}
\begin{array}{c}
\begin{array}{ccccc} t_1 & t_2 & t_3 & t_4 & t_5 \end{array} \\
\left(\begin{array}{ccccc}
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{array} \right)
\end{array}
$$

2.2 Communication Channel

In the encoding decoding paradigm, knowledge of the channel conditions for transmission is key. The effect of noise of transmitted signals helps to estimate the expectations at the receiver. This knowledge largely informs the processes required to recover the message. A good decoding algorithm therefore must cater for the channel conditions in its design.

2.2.1 AWGN Channel

This research investigates the decoder's performance with Additive White Gaussian Noise (AWGN) channel characteristics. Additive noise is very common in electronic communication systems and exist in most channel designs as thermal noise [2]. Thermal noise is commonly generated from electronic components in the communication system. It is a random process and considered a power signal. The random process is non-finite and therefore is modelled to have infinite energy.

The signals transmitted across the channel can be defined by vectors distinct in R^n vector space. Statistically, the noise values of an AWGN channel are mutually independent random variables with mean $\mu = 0$ and same variance σ and power spectral density $N_0/2$. The variance of the noise values is expressed as $\sigma^2 = N_0/2$.

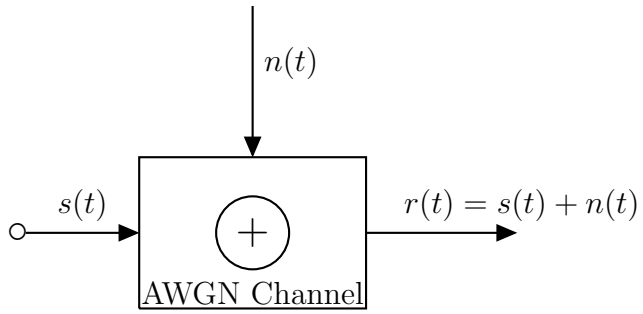

FIGURE 2.3: Block Diagram of System with AWGN Channel Model

2.2.1.1 Coherent Detection in AWGN Channels

If the transmitter and receiver are perfectly synchronised, then we assume the signal arrives at the receiver without a phase change. This implies the signal can be detected coherently [2] and the output of the channel as shown in Fig. 2.3 can be represented as

$$r(t) = s(t) + n(t). \tag{2.5}$$

Therefore, using 2.5 to introduce AWGN to the vector representation in (2.4) and assuming f_1 was transmitted, a corresponding element-wise addition of $s(t)$ to $n(t)$ is

$$r_1 = (\sqrt{E} + n_1, n_2, n_3, n_4), \tag{2.6}$$

where $n_1, \ldots, n_4$ are Gaussian random variables with zero-mean μ and equal variance σ^2.

In order for the detector to make a decision, the received signal is correlated with all possible transmitted signals $s_{1\ldots M}$ as shown in block diagram Fig. 2.4 adapted from [2]. The receiver selects the signal with the largest correlation value as the likely transmitted signal which satisfies the condition

$$\hat{m} = arg\,max\,(r\,.\,s_m), \quad \text{for} \quad 1 \leqslant m \leqslant M, \tag{2.7}$$

where r is the received signal.

With the assumption that f_1 was transmitted, correlation of the received signal with all possible transmitted signals s_m is

$$R_1 = r_1.s_m \quad \text{for} \quad f = 1. \tag{2.8}$$

The vector representation of the receiver output values to be correlated with all possible transmitted signals can therefore be represented as

$$R_i = \begin{cases} \sqrt{E_s} + n_i, & \text{if } f = f_i \\ n_i, & \text{otherwise.} \end{cases} \tag{2.9}$$

In order to generate AWGN noise values for each chosen SNR, we determine the variance σ as follows [2]:

$$N_0 = \frac{E}{SNR}, \tag{2.10}$$

and therefore,

$$\sigma = \sqrt{\frac{N_0}{2}}. \tag{2.11}$$

2.2.1.2 Noncoherent Detection in AWGN Channels

Unlike coherent detection which assumes the transmitter and receiver are in perfect synchronisation for all time intervals, this is often times not practical in real-time

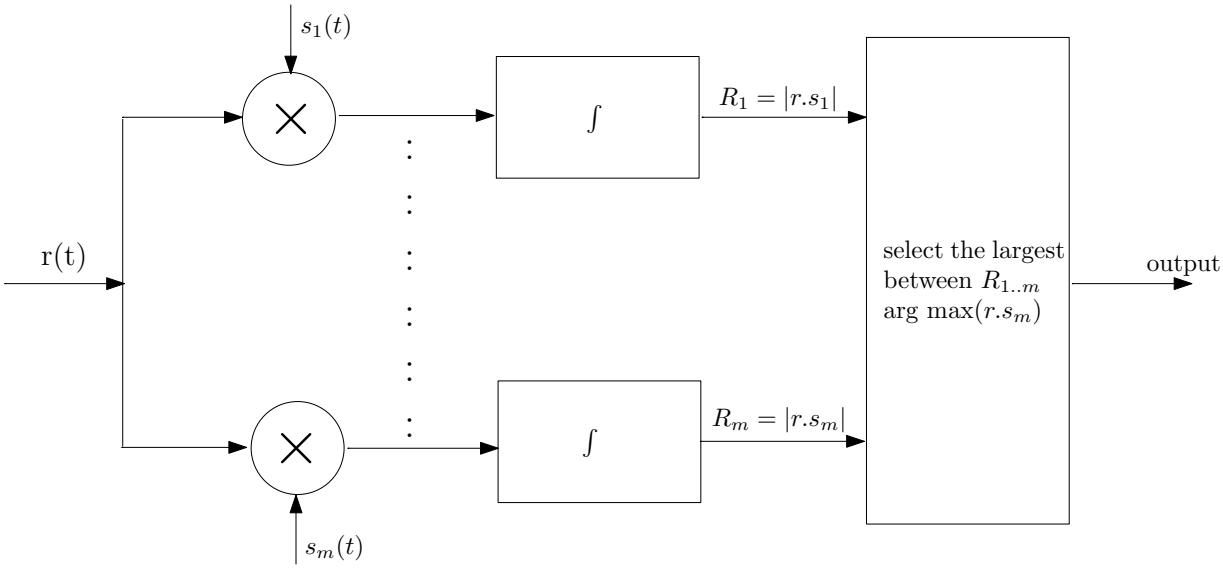

FIGURE 2.4: Envelope Detector in AWGN Channel

communication systems. Imperfect synchronisation of transmitter and receiver implies random phase shifts and time delays of the transmitted signal at the receiver. Although the time delay at each interval is negligible, the multiplicative effect of a large carrier frequency means the shift $\phi = 2\pi f_c t$ becomes noticeable [2]. To the user, the effect of the phase shift on the signal is random. However, ϕ can be modelled as a uniformly distributed random variable between 0 and 2π.

The equivalent lowpass signal detected at the receiver at each interval is represented as

$$r(t) = s(t)e^{j\phi} + n(t), \tag{2.12}$$

where $e^{j\phi} = cos\phi + jsin\phi$ and n(t) are complex-valued zero-mean Gaussian random variables. The condition for the receiver to make a correct decision is obtained using (2.7) and a block diagram in Fig. 2.7 adapted from [2] to illustrate the components of the Envelope Detector. The receiver however calculates the magnitude of the correlated values for Envelope Detection. Therefore,

$$R_m = |r.s_m| \quad \text{for} \quad 1 \leqslant m \leqslant M, \tag{2.13}$$

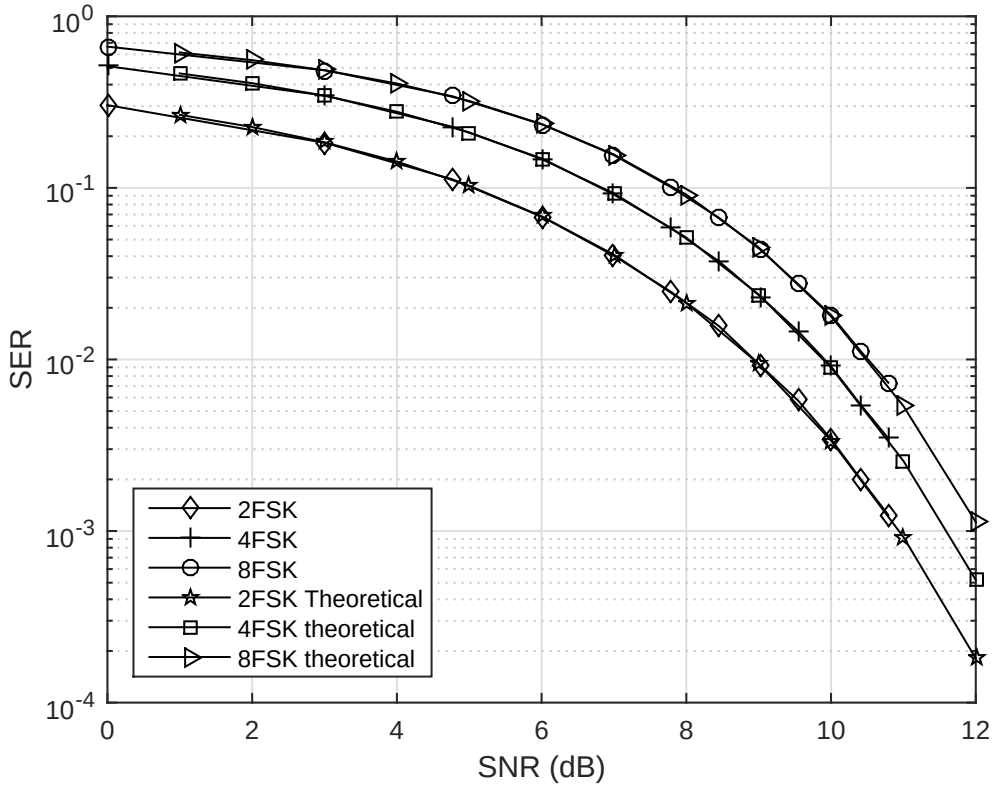

FIGURE 2.5: Noncoherent Detection of MFSK - Symbol Error Rate vs SNR for Theoretical and Monte Carlo

where R_m are independent random variables. Assuming f_1 is transmitted, then R_1 has a Ricean distribution with $\sigma = \sqrt{2E_sN_0}$ and R_m, for $2 \leqslant m \leqslant M$ have Rayleigh distribution [2] also with $\sigma = \sqrt{2E_sN_0}$.

The probability of a symbol error for orthogonal signaling noncoherently detected in AWGN channel is given by [2]

$$P_c = \sum_{n=0}^{M-1} \frac{(-1)^n}{n+1} \binom{M-1}{n} e^{-\frac{n}{n+1}\frac{E}{N_0}}. \tag{2.14}$$

Fig. 2.5 shows the plot of (2.14) and compared with a Monte Carlo simulation of a 100,000 random messages selected with equal probability P_m and detected noncoherently.

2.2.2 Rayleigh Fading Channel

A transmitted signal experiences multipath propagation when received from the channel that has signal fading characteristics. The effect of signal fading on the received signal includes varying arrival times with each path having its delay parameters. Each path also has an attenuation factor that scales the signal transmitted along the path. The combination of the attenuation factor and transmission delay on the transmitted signal produces a received lowpass signal $r(t)$ and can be expresses as

$$r(t) = \sum_{n} \alpha_n(t) e^{j\theta_n(t)}, \tag{2.15}$$

where α_n is the attenutation factor for path n and θ_n is $-2\pi f_c \tau_n(t)$. $\tau_n(t)$ represents the time delay for the n-th path. A change of τ_n by $\frac{1}{f_c}$ changes $\theta_n(t)$ by 2π rad.

The time delays τ_n are assumed as random and can be modelled statistically as a complex-valued random Gaussian process [2]. A channel is assumed to have a Rayleigh distribution if the process is Gaussian with zero-mean. This model is as a result of scattering in the ionosphere and troposphere during signal propagation in both mediums. For such zero-mean Gaussian process, the phase is random and distributed between 0 and 2π [2].

2.2.2.1 Frequency-nonselective Slowly Fading Channel

Signals transmitted via radio communication channels experience scattering in the ionosphere and troposphere. For example, if the same impulse signal is transmitted at two different time intervals, the received signals will however not be the same. The reflection can be caused by moving reflectors such as ions in motion or a moving vehicle. The reflection can also be stationary such as signals reflecting off buildings.

This scattering process can be modelled as a set of random events that can be represented statistically. The receiver design is such that is capable of combining the scattered signals. When the signals are combined at the receiver, the effect of the

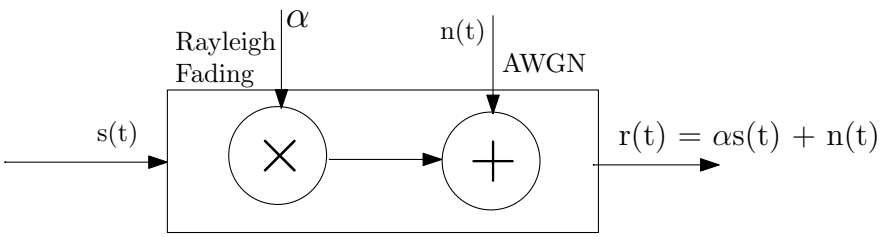

FIGURE 2.6: Block Diagram of System with AWGN & Rayleigh Fading Channels

combination can either attenuate the signal constructively or destructively. This effect on the signal is characterised as fading and the fading constant which attenuates the signal has a multiplicative effect on the signal. If a fading constant applies over a period of intervals before it changes, this is considered as a slowly fading channel.

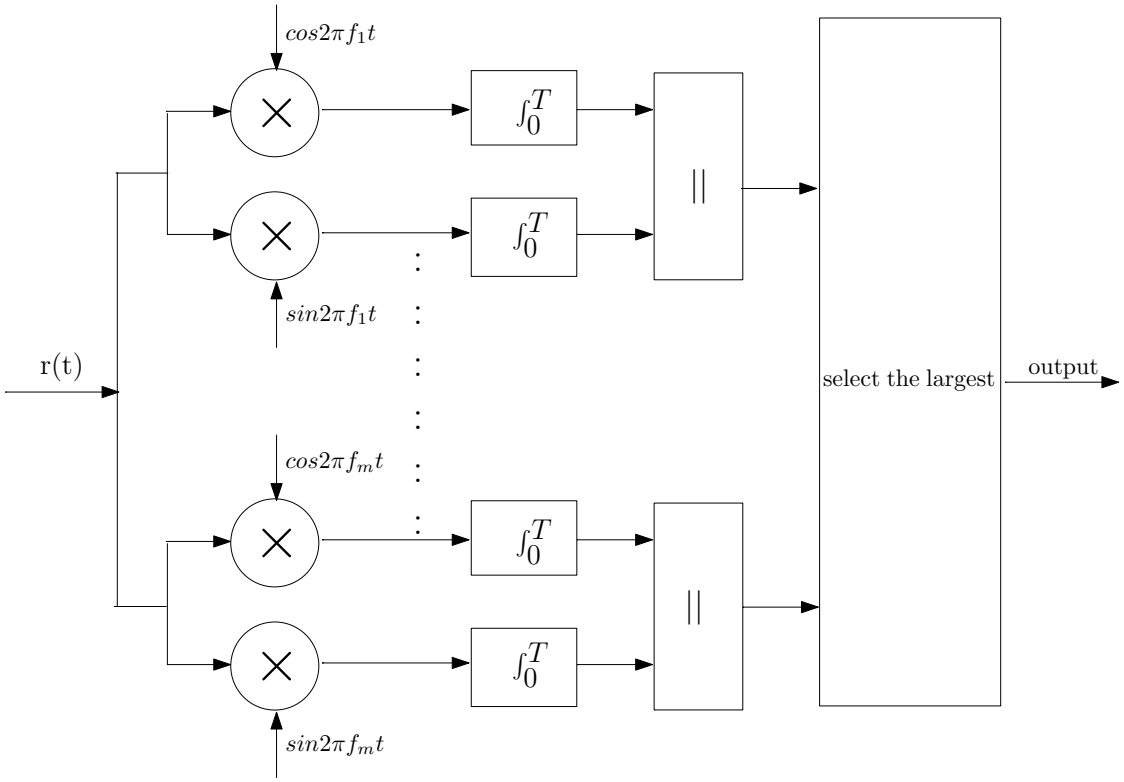

FIGURE 2.7: Block Diagram of MFSK Noncoherent Detector

For a signal transmitted over a frequency-nonselective slowly fading channel, the equivalent lowpass signal detected at the receiver at each interval as shown in Fig. 2.6 is represented as

$$r(t) = \alpha\, s(t)e^{j\phi} + n(t), \tag{2.16}$$

where α is the fading constant and equals $x + jy$, $\theta = 2\pi f_i t$ and $\psi = arctan(y/x)$ and n(t) is a complex-valued white gaussian random variable.

The fading constant α can be modelled as zero-mean complex-valued Gaussian distribution in order to be Rayleigh-distributed or otherwise Rayleigh Fading Channel.

Therefore,

$$r(t) = \alpha\, s(t)(cos\phi + jsin\phi) + n(t). \tag{2.17}$$

Assuming f_1 is transmitted and using $s(t) = Acos(2\pi f_1 t + \phi)$, correlation therefore will be

$$r(t) = |\alpha|\, cos\psi\, Acos(2\pi f_1 t + \phi)\, (cos2\pi f_1 t + jsin2\pi f_1 t) + n(t) \tag{2.18}$$

The real part of the eq. 2.18 becomes

$$Re[r(t)] = \alpha Acos(2\pi f_1 t + \phi)(cos2\pi f_1 t) + n(t). \tag{2.19}$$

Using cosAcosB = $\frac{1}{2}$[cos(A - B) + cos(A + B)] and expanding the real part of (2.19),

$$Re[r(t)] = \frac{\alpha A}{2}[\int_0^T cos\, 4\pi f t\, dt + \int_0^T cos\, \phi\, dt] + Re[n(t)] \tag{2.20}$$

$$\int_0^T cos\, 4\pi f t\, dt = 0, \tag{2.21}$$

therefore

$$Re[r(t)] = \frac{\alpha A}{2} \int_0^T cos\phi\, dt + Re[n(t)] \tag{2.22}$$

$$Re[r(t)] = \frac{|\alpha| A}{2}\, cos\psi\, cos\phi + Re[n(t)] \tag{2.23}$$

$$Im[r(t)] = \alpha A cos(2\pi f_1 t + \phi)(sin 2\pi f_1 t) + n(t). \tag{2.24}$$

Using cosAsinB = $\frac{1}{2}$[sin(A + B) - sin(A - B)] and expanding the imaginary part of (2.24),

$$Im[r(t)] = \int_0^T \frac{\alpha A}{2}[sin(4\pi f_1 t + \phi) - sin\phi]\, dt + Im[n(t)] \tag{2.25}$$

$$\int_0^T sin\, 4\pi f t\, dt = 0, \tag{2.26}$$

therefore

$$Im[r(t)] = -\frac{\alpha A}{2}\int_0^T sin\phi\, dt + Im[n(t)] \tag{2.27}$$

$$Im[r(t)] = -\frac{|\alpha| A}{2}\int_0^T cos\psi\, sin\phi\, dt + Im[n(t)] \tag{2.28}$$

The real part in (2.23) and imaginary part in (2.28) are modelled for MATLAB simulations. The outcome of the model produces noise signals that can be decoded using Hard-decision or Soft-decision decoding.

2.3 Permutation Codes

Given a set of integers $i = \{i_1, i_2, \ldots, i_K\}$, a permutation codebook C is defined as a subset of permutations of the integers i, such that the minimum Hamming distance d_{min} of the codebook C is the largest, i.e., the minimum of the Hamming distances between any two permutations for the codebook C is optimised. A permutation codebook is therefore a $K \times |C|$ matrix such that each row of the matrix is a codeword. The first codeword being a set of integers such that each symbol appears only once,

the remaining codewords in the codebook are formed by reordering the symbols of the first codeword [10]. Generally, a permutation array with N codewords can be written as PA(N, d_{min}) and have the following properties:

The coderate of the code C is defined as [7]

$$R = \frac{\log_2(|C|)}{K \log_2(K)}, \tag{2.29}$$

where K is the length of each codeword. The coderate in the case of permutation codes is a measure of the size of data that can be encoded. For example, a codebook with 12 codewords can map onto more unique messages compared with a codebook with 4 codewords.

2.3.1 Definition 1

The **Hamming distance** d_m between any two vectors v and w, denoted by $d_m(v, w)$, is defined as the number of positions where the values are not the same [17].

2.3.2 Definition 2

The **minimum Hamming distance** d_{min} of a permutation codebook is defined as [17]

$d_{min} = min\{d(\bar{v}, \bar{w}) : \bar{v}, \bar{w} \in C, \bar{v} \neq \bar{w}\}$

2.3.3 Definition 3

An **invalid codeword** is referred to as a codeword A_i not in C i.e. $A_i \notin C$. Such codeword might either be a valid or invalid permutation codeword.

2.3.4 Definition 4

A **hard-decision decoder** of permutation codes is a decoder that finds the codeword in C that has the smallest d_m with a received codeword A_i by performing a one-to-one comparison of A_i with each codeword in C.

Consider a set of messages mapped onto a permutation codebook and transmitted via an AWGN channel, the received codeword may arrive at the receiver as invalid. A hard-decision decoder therefore performs a maximum likelihood operation by choosing the codeword with the smallest d_m with the decoded vector as the likely transmitted codeword. A hard-decision decoder can detect $d_m - 1$ errors and correct $\frac{d_m - 1}{2}$.

2.3.5 Types of Permutation Codes

Permutation modulation, introduced in [10] showed the concept of substituting known modulation methods such as FM, AM with a permutation code book. Permutation code books are simply formed by reordering the symbols that form the first codeword. Slepian [10] classifies permutation codes as follows:

2.3.5.1 Variant 1

Variant I permutation codes include positive and negative real integers [10]. The symbols of each codeword do not necessarily have to be unique. Equation 2.30 defines the range of each codeword in the codebook.

$$i = \{i_1, i_2, \ldots, i_K\} \qquad -\infty < K < \infty \tag{2.30}$$

2.3.5.2 Variant 2

Variant II permutation codes include positive non-zero integers [10]. The symbols of each codeword do not also necessarily have to be unique. The equation in (2.31)

defines the range of each codeword in the codebook.

$$i = \{i_1, i_2, \ldots, i_K\} \qquad -1 < K < \infty \tag{2.31}$$

2.4 The Assignment Problem

In linear programming, if the cost A is such that we minimize [16]

$$A = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} c_{ij} x_{ij}, \tag{2.32}$$

subject to

$$\begin{aligned} \sum_{i=1}^{i=n} x_{ij} &= 1(i = 1, \ldots, n) \\ \sum_{i=1}^{i=n} x_{ij} &= 1(j = 1, \ldots, n), \end{aligned} \tag{2.33}$$

where $x_i j \geqq 0$, then A_i is the cost of the assignment and $X = (x_i j)$ is a $n \times n$ permutation matrix in which all n assigned row-column pair is 1 while unassigned $n^2 - n$ row-column pairs are 0.

2.4.1 Example 2.1

A sample permutation matrix is given as

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The row-column pair of an assignment solution which can be represented as

$$a_k = \{(1, j_1), \ldots, (n, j_n)\} \tag{2.34}$$

solves $a_k = 2134$ for X.

2.4.2 The Hungarian Algorithm

The Hungarian algorithm [16] falls under a branch of combinatorial mathematics to solve the assignment problem. In 1955, Harold Kuhn [16] was able to extend the work of D. Konig and E. Egervary in linear programming to derive a solution to the assignment problem. Before then, linear programming was able to solve small assignment problems that can be arranged in a relatively small-sized matrix. The solution however grows in complexity exponentially for linear programming methods when required to solve larger sizes from 12×12 matrices which presents 144 values for the computer to process.

The order of complexity of the Hungarian algorithm is $O(N^3)$ [18]. Depending on the complexity of the cost matrix, the required number of steps, iterations and recursions to arrive at a solution varies. The complexity required to solve different assignment problems is therefore not always at its maximum.

Given the cost for each of n workers to perform n tasks is known, the Hungarian algorithm solves (2.36) by assigning all n tasks to n workers such that

1. A worker can only perform not more than a single task.

2. If a task is assigned to a worker, such task can no longer be assigned to any other worker. Using matrix column-row terms, once a cost is assigned, no other cost can be selected from that row or column.

3. The sum of all the costs for each worker to perform a task each is the minimum cost the employer has to pay for all the tasks to be assigned as described above. Therefore, the algorithm's outcome should give the minimum total cost

and does not necessarily have to select the least cost for each row or column to determine the minimum cost.

4. There are as many number of workers as number of tasks. Therefore, arranging the costs to satisfy this condition gives a square cost matrix. If there are not as many costs as workers or otherwise, least additional rows and columns that are needed to complete the cost matrix as square are added to the end of the matrix.

2.4.3 Definition 5

The **first iteration** A_1 of the soft-decision decoder is the solution of the Hungarian algorithm.

2.4.4 Example 2.2

A sample cost matrix is given as

$$C = \begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{bmatrix},$$

Applying the Hungarian algorithm to solve for the minimum assignment to C, the solution to C therefore is:

$$C = \begin{bmatrix} 90 & 75 & 75 & \boxed{80} \\ 35 & 85 & \boxed{55} & 65 \\ 125 & \boxed{95} & 90 & 105 \\ \boxed{45} & 110 & 95 & 115 \end{bmatrix}.$$

Using (2.34),

$$a_1 = \{(1,4),(2,3),(3,2),(4,1)\}. \tag{2.35}$$

Algorithm 1: Hungarian Algorithm

Data: Square cost $(n \times n)$ matrix

Result: Minimum cost row-column assignment

```
initialization;
subtract from each item of each row, the row's lowest item;
subtract from each item of each column, the column's lowest item;
if minimum number of horizontal and vertical lines to cover zeros is less than n
then
    repeat
        subtract from all items of each uncovered row, the smallest item uncovered
        by a line;
        add to all items of each covered column, the smallest item uncovered by a
        line;
    until minimum number of horizontal and vertical lines to cover zeros is less
    than n;
else
    return
end
```

Although, the above conditions use tasks and workers for illustration, the assignment problem can be found in wider applications. For example, a coach needs to assign players to positions such that the entire team is effective as a whole or optimising employees' skills without overworking them.

2.5 Murty's Algorithm

Murty's algorithm [19] extends the minimum assignment solution by ranking the costs of a square matrix in order of increasing costs. Using the Hungarian algorithm or some other algorithm such as Jonker-Volgenant algorithm, the next best assignment $a_{i+1} \dots a_k$ can be solved.

The Hungarian algorithm solves the minimum cost required to assign jobs to workers with each worker performing a job. What if however, the employer is willing to know the next minimum cost?

Murty's algorithm [19] was derived to find the 2nd up to the k-th cost in order of increasing costs.

2.6 The k-th Assignment Problem

The Hungarian algorithm stops at the minimum cost of the assignment problem. In order to be able to find the second, third up to the k-th minimum cost of the assignment problem, Murty's algorithm was derived with order of complexity $O(kN^4)$ [20]. Part of this can be attributed to its high dependence on the Hungarian algorithm while it solves for the k-th assignment. Its starting point is the solution of the first assignment A_1. Unlike the Hungarian algorithm, all steps of Murty's algorithm have to be completed in order to arrive at the final solution and the Hungarian algorithm has to be used $N-1$ times for a $N \times N$ matrix.

The row-column constraint is also kept for the k-th assignment. Therefore, every output of Murty's algorithm is also definitely a permutation code. This maintains the higher probability of correctly decoding the received signal.

Consider the solution a_1 in (2.34), the algorithm extracts $n-1$ non-empty subsets of C_s into nodes $N_1 \dots N_{n-1}$, a process referred to as partitioning. Nodes in this case are defined as:

$$\begin{aligned} N_1 &= \{\overline{(i_1, j_1)}\}, \\ N_2 &= \{(i_1, j_1); \overline{(i_2, j_2)}\}, \\ &\vdots, \\ N_r &= \{(i_1, j_1), \ldots, (i_{r-1}, j_{r-1}); \overline{(i_r, j_r)}\}. \end{aligned} \quad (for \quad r = 1, \ldots, n-1) \tag{2.36}$$

The row-column pair without the bar implies the elements on the row and column are removed from C_s for that node while the row-column pair with the bar implies the item at that row-column position is replaced with a very large value or infinity. The minimum cost is then solved for each node. The node with the least cost forms the next assignment which can be represented with (2.34) and is used to partition for the next assignment.

The minimum assignment costs of the nodes can be arranged in a row-vector as

$(N_1 \dots N_r)$. The node with the lowest minimum assignment cost is the next assignment cost A_{1+i}.

Algorithm 2: Murty's Algorithm

Data: Square cost $(n \times n)$ solution matrix of the Hungarian algorithm

Result: row-column assignment vector

initialize n to row length of matrix;
initialize i to 1;
initialize $1 \times (n-1)$ row-vector $\hat{v}$;
generate $n-1$ nodes from the $n \times n$ solution matrix;
in
while *i is less than or equal to $n-1$* **do**
    generate node N_i;
    determine minimum cost of node N_i;
    add minimum assignment cost of N_i to $\hat{v}$;
    add 1 to i;
end
find v_{min} which is the smallest cost value in $\hat{v}$;
the node that produces v_{min} produces the next assignment solution a_{1+k}

2.6.1 Definition 6

The **second iteration** A_2 of the soft-decision decoder is the first solution of Murty's algorithm while the decoder's **third iteration** A_3 is Murty's second solution. The k-th iteration of the decoder therefore produces a solution A_k which is the $k-1$ solution of Murty's algorithm.

2.6.2 Example 2.3

Using C and a_1, the list for the next stage are formed using the nodes from a_1 as follows:

$$N_1 = \{\overline{(1,4)}\} = \begin{bmatrix} 40 & \boxed{0} & 5 & \infty \\ 0 & 25 & 0 & \boxed{0} \\ 55 & 0 & \boxed{0} & 5 \\ \boxed{0} & 40 & 30 & 40 \end{bmatrix},$$

with minimum cost of 0,

$$N_3 = \{(1,4); \overline{(2,3)}\} = \begin{bmatrix} 0 & \boxed{25} & \infty \\ 55 & 0 & \boxed{0} \\ \boxed{0} & 40 & 30 \end{bmatrix},$$

with minimum cost of 25 and

$$N_3 = \{(1,4),(2,3); \overline{(3,2)}\} = \begin{bmatrix} \boxed{55} & \infty \\ 0 & \boxed{40} \end{bmatrix},$$

with minimum cost of 95.

Since node N_3 has the lowest cost from the 3 nodes, node N_3 solves for a_2 as {(1, 3), (2, 1), (3, 4), (4, 3)}. a_2 is then used to partition N_3 in order to find a_3.

2.7 Maximum Likelihood

Since the cost matrix includes the actual costs each worker requires to carry out a task, ranking the costs of all possible permutations of the matrix also produces a solution.

The Maximum Likelihood (ML) decoder uses the brute-force method of ranking of all costs of a square matrix and is equivalent to the combination of the Hungarian Algorithm and Murty's algorithm. The brute-force method computes the costs of all possible codewords $|P|$ and then sorts in descending order. The cost ranking of $|C|$ is a subset of the cost ranking of $|P|$ and can therefore be extracted. This ML decoder's process is irrespective of the codebook size.

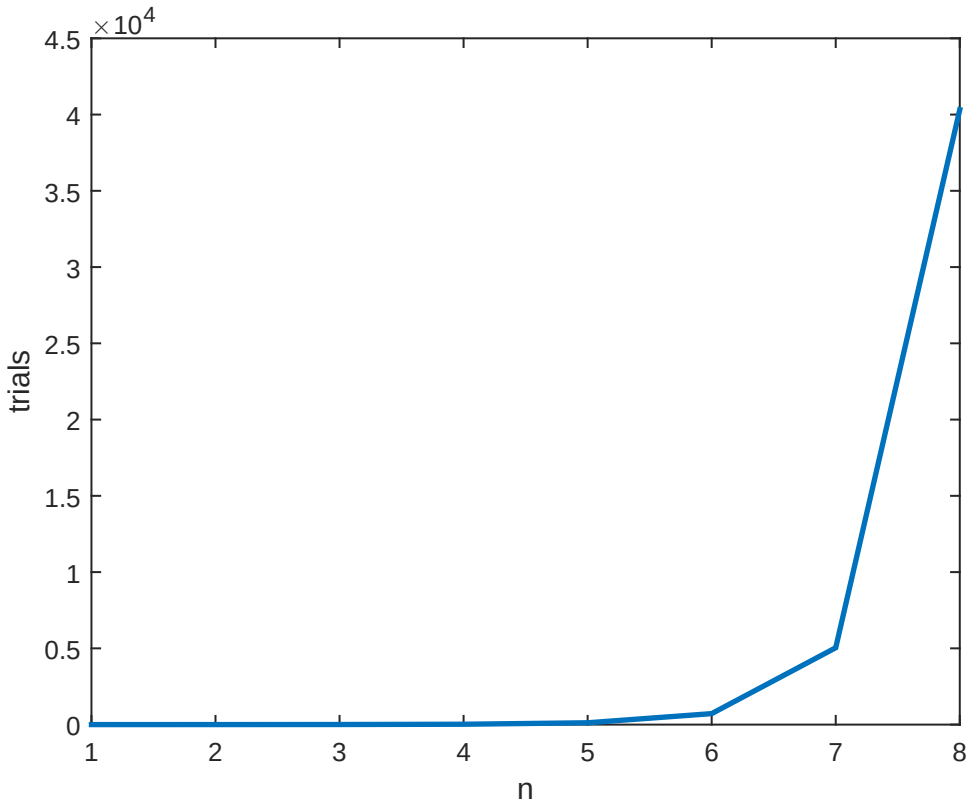

FIGURE 2.8: Trial and Error Method of Computing Costs for given $\sqrt{n}$

The process of computing the cost of all possible codewords requires $n!$ trials [21] or $O(n!)$. For small n, the computational complexity is trivial. However, as n increases, the number of computations required to rank all the costs becomes large at an exponential rate as shown in Fig. 2.8. The computational complexity required to decode for large n is too high and renders the decoding method infeasible. This method however gives the maximum likelihood performance for decoding Permutation codes using the cost ranking method.

Chapter 3

Research Methodology

This chapter investigates the characteristics of the assignment problem that makes the Hungarian algorithm and Murty's algorithm suitable for a soft-decision decoder of permutation codes.

An efficient way of decoding permutation codes was derived in [8]. The decoding algorithm however depends on the algorithm used in the code construction process. Soft-decoding described in [15] relies on knowledge of the message sent at the decoder. Practicality of such decoding method reduces quickly as code rate increases.

Similar to the assignment problem, generation of permutations is a concept in combinatorial mathematics as is the assignment problem. An analysis of the Hungarian algorithm and Murty's algorithm is done in order to link the characteristics that make the soft-decision decoder and permutation codes compatible. Experimental simulations are carried out to discover to what extent this algorithm can efficiently decode in the AWGN and Rayleigh fading channels. The performance is also judged based on the complexity of the algorithm because it is important the algorithm remains considerably practical while the size and complexity of the codes increase.

Simulations are done using engineering computing software, MATLAB. Discrete random messages will be generated, each with equal probability of occurrence. This helps introduce some level of unpredictability into the system. Each symbol in the

message is then encoded by mapping each message to a codeword in the codebook such that similar messages are mapped to the same codeword. Each message is assumed to be transmitted at regular time intervals. The implication is that each message will be encoded for modulation with n frequencies where n is the length of each codeword.

The channel model in the simulations include coherent and noncoherent detection in AWGN. Rayleigh fading is also added to the noncoherent detection in AWGN. The AWGN channel is modelled by generating a vector of random integers with zero mean, equal variance relative to the signal-to-noise ratio [2]. Noncoherent detection has the same statistical distribution but is modelled as complex-valued. First, the message is recovered from the noisy signal using hard-decision which uses envelope detection and a lookup with the codebook. Secondly, the soft-decision decoder comprises of the Hungarian algorithm for maximum assignment and Murty's algorithm that ranks costs in descending order. Both the hard-decision and soft-decision methods are compared to each other using Symbol Error Rate (SER) versus Signal to noise (SNR) plots.

This simulation will be done for different codebooks and at different code rates. The research investigates ways to recognise codewords that fall outside the set of codewords used in encoding the message. These codewords introduce errors in the system and will therefore find ways of making an intelligent decision in decoding some of these codewords correctly.

The success criterion therefore is to achieve a significant dB gain using the SER vs SNR plot. The performance of the soft-decision decoder is compared with hard-decision. The Hungarian algorithm is considered as the first iteration and is the first point-of-call in the decoding process.

Should the first iteration produce a decoded vector $A_i \notin C$, then the next set of iterations are done using Murty's algorithm. The aim is to run enough iterations to

find the next highest assignment cost A_{i+1} of the square matrix for only outputs

$$A_i \notin C, (i > 1).$$

Each step to solve A_{i+1} will be applied as a means of complementing the A_i outputs. If these iterative steps improve the algorithm, the research will investigate the saturation point of the performance of the algorithm. Simulations will consider different code rates and codebooks in order to recommend the iteration beyond which further iteration will not necessarily improve the performance algorithm.

3.1 Experimental Setup

An end-to-end setup of the system consists of a message transmitter, Rayleigh slow Fading channel, AWGN channel and receiver in respective order as shown in Figure 3.1. The transmitter comprises of a random message generator, an encoder and MFSK modulator while the receiver includes correlators and a decoder which decodes and demodulates simultaneously the signals at the output of the channel.

3.2 System Description

The message generator generates a sequence of random messages, each message is a positive integer v_i such that $1 \leq v_i \leq |C|$. This sequence is assumed to represent messages at regular time intervals. Consider a codebook P containing all possible permutations $|P|$, the encoder selects a subset C from P for a chosen code rate and Hamming distance. One-to-one mapping operation is then performed at the encoder, the outcome which is fed into the MFSK modulator. In the modulator, the transmitted frequency is represented as 1 while other unused $M - 1$ frequencies at that timeslot remain 0.

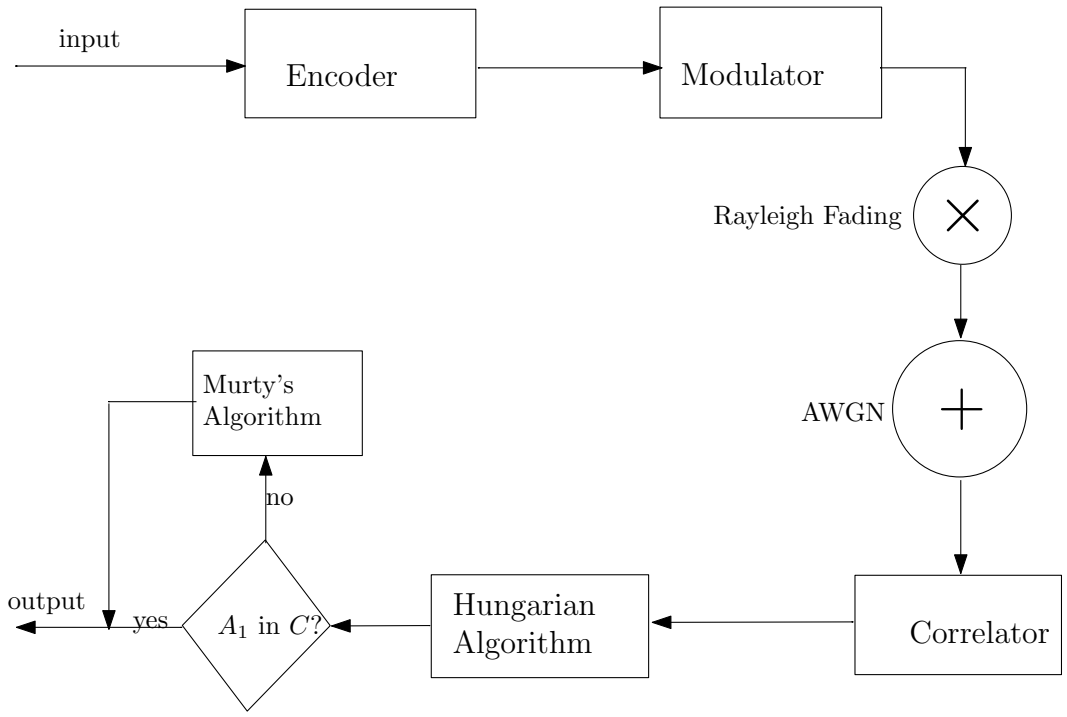

FIGURE 3.1: Experiment Block Diagram

The Rayleigh Fading channel is modelled as a block fading channel. It is multiplicative in nature and therefore attenuates the signal either constructively or destructively. The AWGN channel on the other hand is additive and is an $M \times N$ matrix that can be added to the signal from the modulator.

The Hungarian algorithm for maximum assignment and Murty's algorithm for the k-th assignment always require square matrices to operate. Therefore, in the decoder and demodulator, the noisy signal at the output of the channel is broken into blocks of square matrices, each block equivalent to the message at that interval. The outcome of the Hungarian algorithm is always a permutation codeword and in this case considered the first iteration. If the output codeword does not exist in the codebook this outputs an error at the receiver. The second iteration up to the k-th iteration employs Murty's algorithm for the k-th assignment by ranking the costs of the square matrix in order of decreasing costs.

For each message to be decoded at the receiver, the algorithms run iteratively until either the k-th iteration produces a valid codeword or reaches the maximum k set for the simulation.

3.3 The Hungarian Algorithm as a Soft-Decision Decoder

It was defined earlier that Variant II permutation codes include positive non-zero integers, the row-column constraint of the assignment solution implies that the solution to an assignment problem will always be a Variant II permutation code.

Consider a set of randomly generated positve integers $\hat{v} = \{v_1, v_2, \ldots, v_E\}$ of length E. Each message at each symbol period is randomly selected from $1 \leq |C|$. Using one-to-one mapping onto a permutation codebook C and modulating with MFSK, the signal, with timeslots increased to $M \times E$ is fed to the AWGN channel. At the output of the AWGN channel, the received signal using Equation 2.9 is

$$R = \begin{bmatrix} n_{11} & n_{12} & n_{13} & \ldots & \sqrt{E} + n_{1L} \\ n_{21} & \sqrt{E} + n_{22} & n_{23} & \ldots & n_{2L} \\ \sqrt{E} + n_{31} & n_{32} & n_{33} & \ldots & n_{3L} \\ n_{41} & n_{42} & \sqrt{E} + n_{43} & \ldots & n_{4L} \end{bmatrix}, \tag{3.1}$$

or

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1L} \\ r_{21} & r_{22} & r_{23} & \ldots & r_{2L} \\ r_{31} & r_{32} & r_{33} & \ldots & r_{3L} \\ r_{41} & r_{42} & r_{43} & \ldots & r_{4L} \end{bmatrix}, \tag{3.2}$$

where $L = M \times E$, the length of the entire message sent at the transmitter and n_{ij} could be either positive or negative noise values.

To decode this signal using the Hungarian algorithm, the signal is divided into L/M blocks of square $M \times M$ matrices such as

$$R_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}. \tag{3.3}$$

In a case where the last block in the matrix is not a square, additional columns filled with zeros are added to the matrix in order to convert to a square matrix as shown in Equation 3.4

$$R_{L-1} = \begin{bmatrix} r_{1L-3} & r_{1L-2} & r_{1L-1} & 0 \\ r_{2L-3} & r_{2L-2} & r_{2L-1} & 0 \\ r_{3L-3} & r_{3L-2} & r_{3L-1} & 0 \\ r_{4L-3} & r_{4L-2} & r_{4L-1} & 0 \end{bmatrix}. \tag{3.4}$$

The Hungarian algorithm is then solved for each block in order to decode the signals. Although, the steps described above in solving the algorithm are for minimum assignment solution, the maximum assignment solution is rather of interest here. In order to calculate the maximum assignment solution, the cost matrix is negated as shown in Equation 3.5

$$R'_{Ni} = -1 \times [R_{Ni}]. \tag{3.5}$$

By applying the same steps to the negated matrix, the outcome will be the maximum cost solution.

An example below uses a one-to-one mapping of messages to the cyclically rotated codebook C with $d_{min} = 4$ [7]

$$C = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \\ 2 & 3 & 4 & 1 \end{bmatrix}.$$

A sample block of a signal from an AWGN output is:

$$R_1^{'} = \begin{bmatrix} -0.0048 & 0.1998 & -0.6410 & 0.6190 \\ 1.4252 & -1.3300 & 0.7402 & -0.6016 \\ 0.7114 & 0.3415 & -0.1570 & 0.8115 \\ -0.4108 & 1.1389 & 1.6880 & 0.5057 \end{bmatrix}.$$

Using Equation 3.5,

$$R_1^{'} = \begin{bmatrix} 0.0048 & -0.1998 & 0.6410 & -0.6190 \\ -1.4252 & 1.3300 & -0.7402 & 0.6016 \\ -0.7114 & -0.3415 & 0.1570 & -0.8115 \\ 0.4108 & -1.1389 & -1.6880 & -0.5057 \end{bmatrix}.$$

Applying the Hungarian algorithm reduces R_1 to

$$R_1^{'} = \begin{bmatrix} 0.6237 & \boxed{0} & 1.2599 & 0 \\ \boxed{0} & 2.3360 & 0.6849 & 2.0268 \\ 0.1001 & 0.0509 & 0.9685 & \boxed{0} \\ 2.0988 & 0.1300 & \boxed{0} & 1.1824 \end{bmatrix},$$

produces an assignment solution $A_1 = 2143$, which is also a permutation codeword. Its row-column representation

$$a_1 = \{(1,2),(2,1),(3,4),(4,3)\}. \tag{3.6}$$

Applying this step to each signal matrix always decodes to a permutation codeword. The decoder however needs to search the codebook for the decoded codeword. An assignment solution that is not a valid codeword is interpreted as an error.

An important edge the Hungarian algorithm gives in decoding permutation codes is that while Envelope Detection always relies on one-to-one comparison with every codeword in order to determine the minimum distance, the outcome of the algorithm is always a codeword. This largely reduces the complexity of the decoder. The decoder then needs to check if the received codeword is valid or a member of the codebook. Efficient algorithms in programming are in existence to solve this without having to perform a one-to-one comparison with each codeword in the codebook.

Every solution to the assignment problem obeys the row-column constraint. Therefore, the solution will always be a permutation codeword. This similarity can be said to make the Hungarian algorithm suitable for decoding permutation codes.

3.4 Murty's Algorithm as a Soft-Decision Decoder

Murty's algorithm is dependent on the first assignment from the Hungarian algorithm, the algorithm is therefore not always executed at every instance of the decoding process. The algorithm is triggered in the decoder when the output of the Hungarian algorithm results in a codeword $A_1 \notin C$. Because the resulting codeword is definitely invalid, the decoder probes further into the next highest cost to determine if $A_i \in C$.

Murty's algorithm therefore helps in finding the next reliable codeword using its ability to determine the next highest cost. Again, although the algorithm solves for the next minimum cost, negating the cost matrix solves for the next highest cost up to the k-th highest cost (lowest cost).

The solution a_1 in Equation 3.6 represents a decoded codeword $A_1 \notin C$. A_1 is therefore an error decoded message. This triggers Murty's algorithm to find the next reliable codeword from the received signal.

The row-column pair without the bar implies the values on the row and column are removed from R_1 for that node while the row-column pair with the bar implies the item at that row-column position is replaced with a very large value or infinity. The minimum cost is then solved for each node. The node with the least cost forms the next assignment a_2.

Using the solution matrix of $R_1^{'}$,

$$R_1^{'} = \begin{bmatrix} 0.6237 & 0 & 1.2599 & 0 \\ 0 & 2.3360 & 0.6849 & 2.0268 \\ 0.1001 & 0.0509 & 0.9685 & 0 \\ 2.0988 & 0.1300 & 0 & 1.1824 \end{bmatrix},$$

$n-1$ nodes of $R_1^{'}$ are

$$N_1 = \begin{bmatrix} 0.6237 & \infty & 1.2599 & \boxed{0} \\ \boxed{0} & 2.3360 & 0.6849 & 2.0268 \\ 0.1001 & \boxed{0.0509} & 0.9685 & 0 \\ 2.0988 & 0.1300 & \boxed{0} & 1.1824 \end{bmatrix},$$

with an assignment cost of 0.0508,

$$N_2 = \begin{bmatrix} \infty & \boxed{0.6849} & 2.0268 \\ \boxed{0.1001} & 0.9685 & 0 \\ 2.0988 & 0 & \boxed{1.1824} \end{bmatrix},$$

with an assignment cost of 1.9674 and

$$N_3 = \begin{bmatrix} \boxed{0.9685} & \infty \\ 0 & \boxed{1.1824} \end{bmatrix},$$

with an assignment cost of 2.1508.

Therefore, for $N = (N_1 \dots N_i)$, the minimum value in N_{min} is N_1. The next assignment solution A_2 is therefore 4123. Since $A_2 \in C$, the decoder operation concludes for the current signal and proceeds to the next signal in the sequence.

If however $A_2 \notin C$, the decoder proceeds to find A_3 using the solution matrix of A_2 and applying the same steps that solve A_2. Unlike the Hungarian algorithm that solves for the first assignment, Murty's algorithm can be repeated for $k-1$ iterations to solve up to the $k-th$ assignment.

Combining the Hungarian algorithm and Murty's algorithm to form a soft decision decoder, the algorithm's pseudocode can be represented as

Algorithm 3: Soft-decision Decoder

Data: Signal output of Correlator

Result: Soft Decision Decoder: Hungarian Algorithm and Murty's Algorithm

```
initialization;
if A_1 ∈ C then
    r̂_l ← A_1;
else
    if A_2 ∈ C then
        r̂_l ← A_2;
    else
        if A_3 ∈ C then
            r̂_l ← A_3;
        else
            ⋮
        end
    end
end
```

Chapter 4

Results

This section discusses the outcome of simulations carried out using MATLAB. The aim is to investigate how many errors the algorithms can correct at the receiver.

Given a sequence of message symbols of length E comprising positive integers $\hat{s} = \{s_1, s_2, \ldots, s_l\}$ and set of decoded messages $\hat{r} = \{r_1, r_2, \ldots, r_l\}$. The non-zero items in $\hat{r} - \hat{s}$ is the number of messages decoded in error. The performance of the decoder is therefore a measure of how much we reduce the non-zero items in $\hat{r} - \hat{s}$ especially at low signal to noise ratio (SNR). The data transmission is done via an AWGN channel with zero-mean and equal variance of $\sigma^2 = \sqrt{\frac{N_0}{2}}$ and detection is done both coherently and noncoherently. Rayleigh Fading is further introduced into the channel conditions. The effect of Rayleigh Fading on the transmitted signals are then shown and analysed.

At the output of the channel using an MFSK system, the received signal is a cost matrix $R = (r_{ij})$ of order $M \times M$. Every assignment in R is a permutation matrix and its column-wise index is therefore a permutation codeword. A permutation matrix is formed such that only one element in each column and row is set to 1 while other elements on the same column and row are set to 0 [19]. An example of a permutation matrix is

$$P_m = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The column-wise index of P_m is a permutation codeword vector $\{2134\}$. There are $M!$ possible permutation matrices and therefore $M!$ possible assignments for every received signal matrix R. By ranking all possible assignments A_i in R in order of decreasing costs, simulation results show that the error-correction performance improves. We define an iteration of the decoding process as the rank of an assignment.

Unlike the maximum likelihood method, the size of M in MFSK determines the size of the cost matrix at the input and output of the channel. M is therefore equivalent to n, the sample size of the cost matrix. For example, a 4FSK system produces a 4 $\times$ 4 matrix for the decoder while an 8FSK produces an 8 $\times$ 8 matrix.

The combination of the Hungarian algorithm is $O(n^3)$ [21] and Murty's algorithm is $O(n^4)$ [22]. A combination of both algorithms in the decoder is $O(n^3) + O(n^4)$ which gives a worst case complexity of $O(n^4)$.

Consider a matrix comprising of the set of all permutations P, a codebook C is defined as $C \subseteq P$ for all codewords or $C \subsetneq P$. In the experiment, each iteration compares the assignment of the iteration with the codebook C. Each iteration after the first is a combination of the performances of all previous iterations. For example, decoding a message with the third iteration means the first and second iteration failed to find $A \in C$. However, decoding the next message may stop at the first iteration but still counts as performance of the third iteration. The next iteration is only activated if $A \notin C$.

We discuss each iteration's performance for different code rates, codebook sizes and d_{min}. The decoder's performance at each iteration is also compared with Envelope detection (ED). We refer to Hard-decision decoding as a combination of ED and minimum distance decoding.

We determine the code rates using Equation 2.29 to generate codebooks. Minimum Hamming distance is also of concern in analysing the performance of the algorithm. The first iteration uses the Hungarian algorithm decoding while the second up to the n-th iteration uses Murty's algorithm.

4.1 4FSK in AWGN Channel - Coherent Detection

A 4FSK system, combined with a permutation codebook with $K = 4$ is a simple system that can be used to analyse the decoder's performance. It is also easier to compare the performance of codebooks with same d_{min} but different codeword composition. The codebook mapped to the 4FSK system has $|P| = 24$ and the decoder input receives 4×4 matrix in this case.

4.1.1 4 Codewords

A simple codebook with $|C| = 4$ and $K = 4$ can be formed by cyclically rotating the first codeword 1234 in order to create the remaining 3 codewords. The outcome of this produces a codebook with $d_{min} = 4$. From Figure 4.1, the first iteration gives similar performance compared with hard-decision. However, a significant gain is observed at the second iteration with more than 1dB gain. The third and fourth iteration produce similar performance but improve the performance of the second iteration by additional 0.6dB.

4.1.2 8 Codewords

Increasing $|C|$ to 8 also increases the code rate slightly to 0.375 and $\frac{|C|}{|P|}$ is 0.33. Figure 4.2 shows the performances of the first three iterations. First iteration still remains similar in performance with hard-decision although slight but negligible improvement

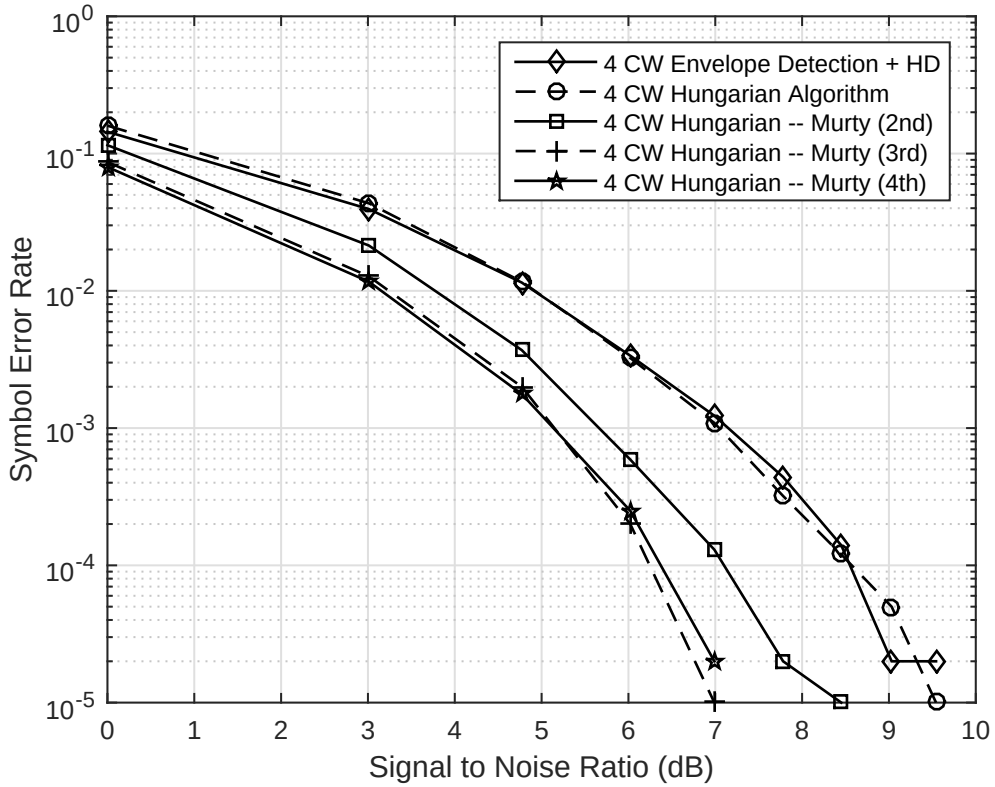

FIGURE 4.1: 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, , Coherent Detection. $d_{min} = 4$, code rate = 0.25

is observed in some SNR regions. The second iteration however improves performance with up to 1.8dB gain. The third and fourth iteration both produce performance similar to the second iteration with minimal 0.1 dB gain improvement in some SNR regions.

4.1.3 16 Codewords

In Figure 4.3, results of the first three iterations are shown for increased code rate of 0.5. With $\frac{|C|}{|P|} = 0.67$, there is reduced probability the decoder will output a codeword in the remaining 33% of codewords in P but not in C. This is why the performance of the first iteration improves compared with hard-decision. Another reason for improved performance is that d_{min} has also reduced, thereby reducing the error correction performance of minimum distance decoding. The first iteration gives

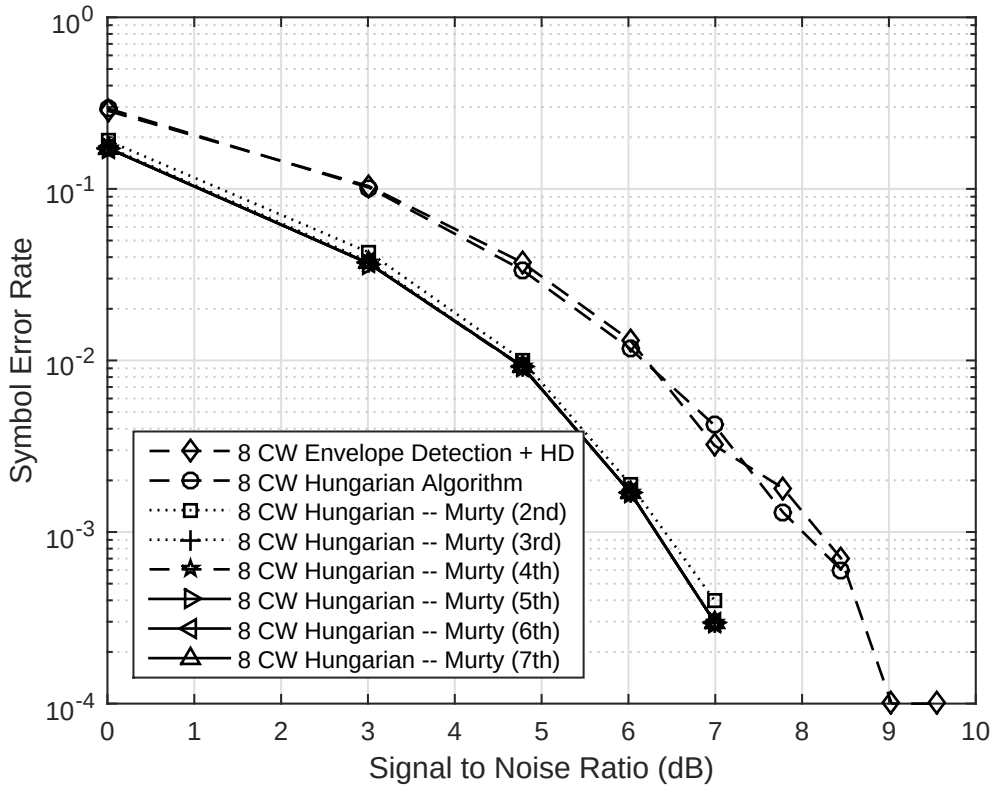

FIGURE 4.2: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN, Coherent Detection. $d_{min} = 3$, code rate = 0.375

more than 2dB gain with up to 2.5dB in some SNR regions. Performance is similar when codebook consists of a different subset of P but with the same d_{min}.

With the exception of matrices with same costs at different iterations, the second iteration only has to produce a codeword in $|P| - 1$ and the third, $|P| - 2$. Therefore, the percentage of codewords that can result in decoding error reduces for each iteration. However, the probability of producing a codeword in C is higher for 16 codewords and this probability increases for the next iteration. Producing a codeword in C does not however guarantee the codeword is correctly decoded. The second iteration therefore does not improve in performance compared with the first iteration as it appears in Figure 4.3 to be nearing its performance limit. The performance of the third iteration is almost exactly the same as the second iteration. In terms of computational complexity, decoding may not be necessary beyond the second iteration. However, between the first and second iteration, the computational complexity

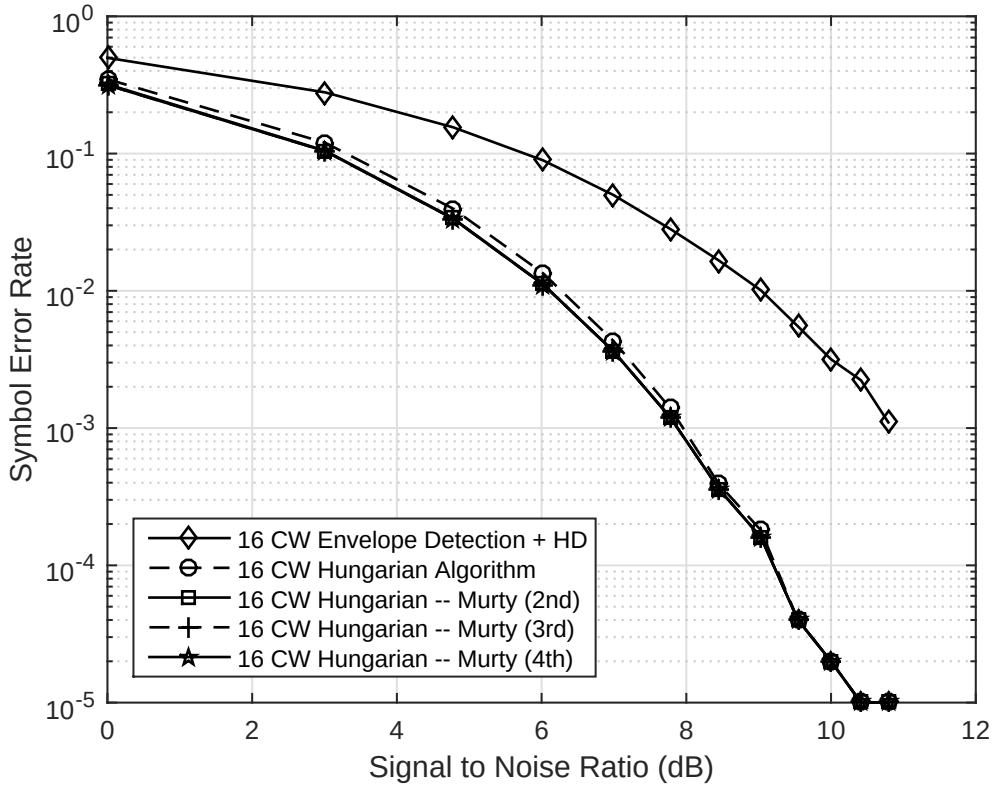

FIGURE 4.3: 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, , Coherent Detection. $d_{min} = 2$, code rate = 0.5

required may be too high compared with the achievable gain.

4.1.4 24 Codewords

Simulation for a code rate of 0.57 as shown in Figure 4.4 uses 100% of the codewords in P. The performance of the Hungarian algorithm gives up to 3dB gain. As mentioned earlier, the more codewords in C, the better the performance of the algorithm. The other iterations after the first are not needed because the condition $A \in C$ is always satisfied in the decoding process because the outcome of the Hungarian algorithm is always a permutation codeword. Therefore, the next iteration is never triggered.

Table 4.1 contains the properties and performance results of each codebook used in the 4FSK system described. It also shows the performance of each iteration up

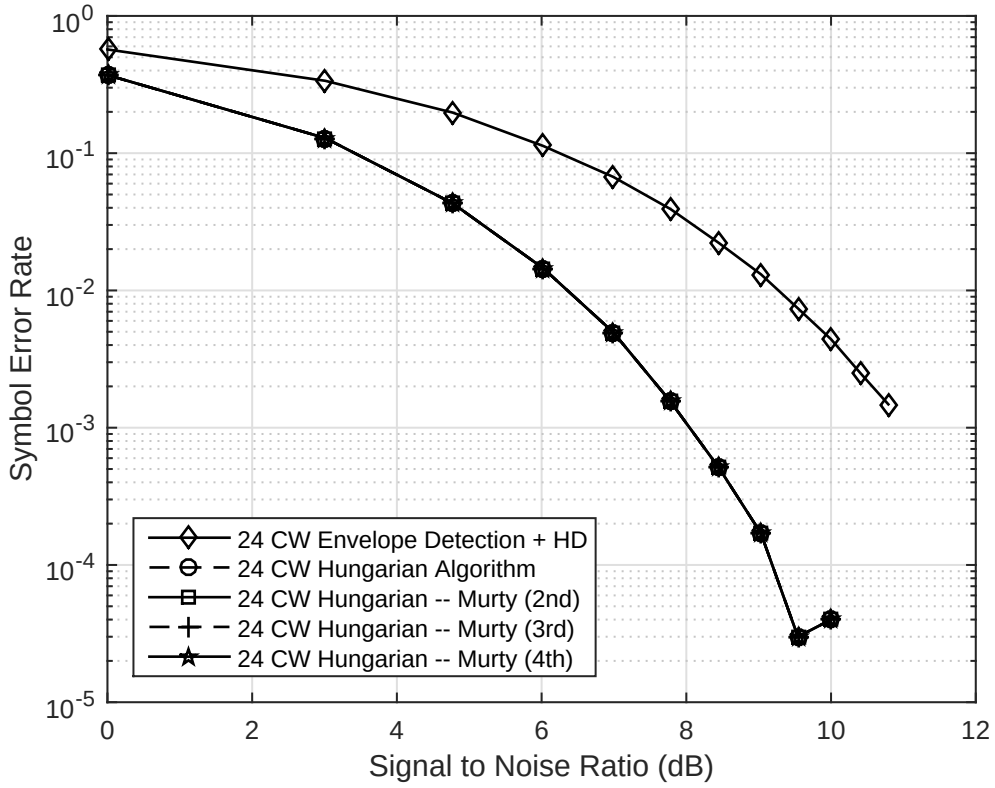

FIGURE 4.4: 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 2$, code rate = 0.57

TABLE 4.1: Performance of Soft-decision Decoder using 4FSK in AWGN Channel, Coherent Detection

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4
4	0.25	0.1667	4	0.2	1.0	1.8	1.8
8	0.375	0.33	3	0.2	1.6	1.7	1.7
12	0.45	0.5	3	1.0	2.4	2.4	2.4
12	0.45	0.5	2	2.8	3.0	3.0	3.0
16	0.5	0.67	2	2.8	3.0	3.0	3.0
24	0.57	1	2	3.0	3.0	3.0	3.0

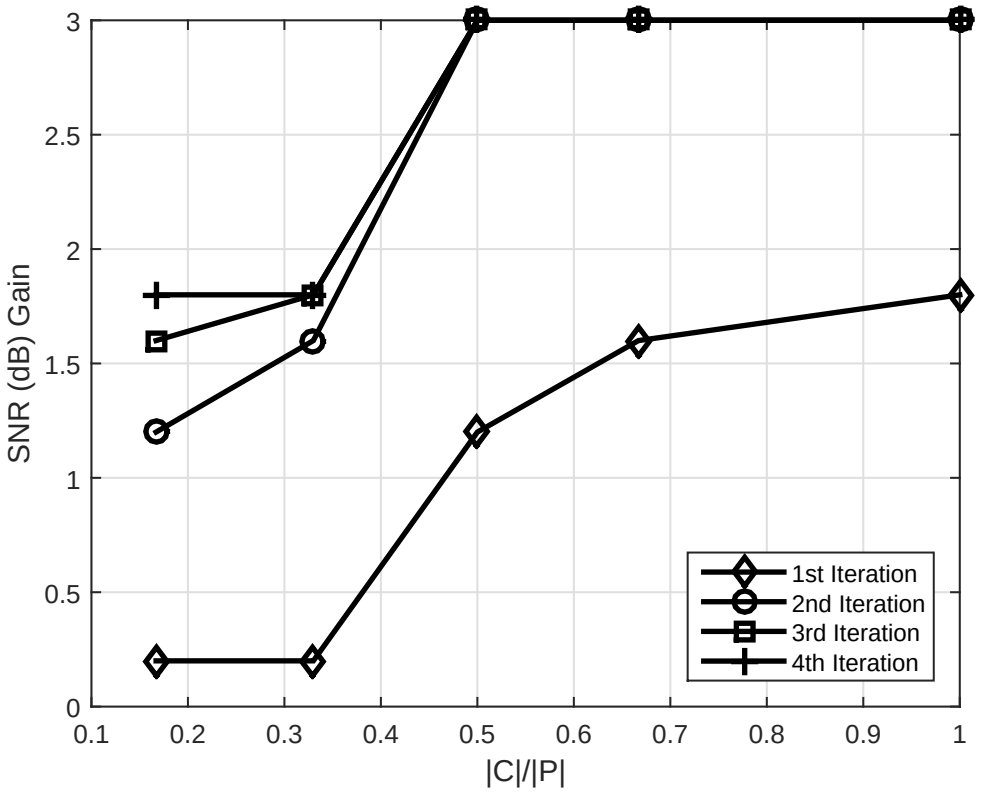

FIGURE 4.5: SNR (dB) Gain vs $\frac{|C|}{|P|}$ for 4 Iterations for $|P| = 24$, 4FSK, Coherent Detection

to the fourth iteration. This performance is compared with the combination of the Envelope Detection plus hard-decision.

As mentioned earlier, the performance is irrespective of the subset of P chosen to form the codebook with $|C| = 16$. However, for codebook with $|C| = 12$, the codebook can be constructed with either $d_{min} = 2$ or $d_{min} = 3$. The performance of both codebooks differ and this is associated with the size of d_{min}. As shown in Table 4.1, the decoder performs better when the d_{min} is 2 than when d_{min} is 3 for $|C| = 12$. Therefore, the higher the d_{min}, the poorer the performance of decoder.

The plots in Figure 4.5 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain.

4.2 8FSK in AWGN Channel - Coherent Detection

The decoder solves for each 8×8 matrix that represents a message. Since both algorithms output a codeword, there is no need to lookup with the codebook. Total possible permutations $|P|$ is 8! or 40320. The increased size of K to 8 means the encoder has more codewords and code rates to select from. Simulations for this set of experiments run iterations up to the 7th best assignment.

$$C = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 8 & 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 7 & 8 & 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 & 8 & 1 & 2 & 3 \\ 3 & 4 & 5 & 6 & 7 & 8 & 1 & 2 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 1 \end{bmatrix}.$$

4.2.1 8 Codewords

The codebook used for this simulation is the cyclically rotated codebook C with $|C| = 8$, $K = 8$ and $d_{min} = 8$. This codebook is however only 0.02% of P. Most decoded codewords are far more likely to fall in the other 99.98% codewords not in C. As shown in Figure 4.6, the Hungarian algorithm decoding performs poorer than hard-decision. Probability of decoding the wrong codeword is very high. More iterations up to the 7th iteration does little to improve the performance. Although the performance at the 7th iteration can be seen to perform slightly better than hard-decision, the gain in decibels (dB) is quite minimal, the highest being about 0.1dB.

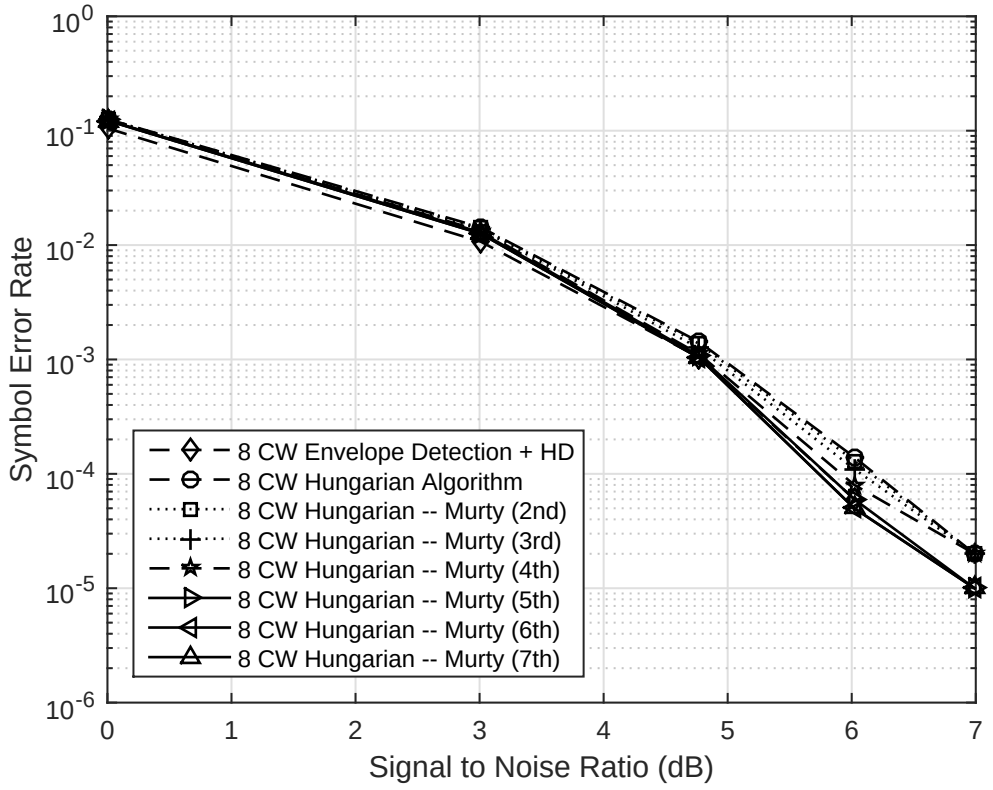

FIGURE 4.6: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 8$

4.2.2 305 codewords

$\frac{|C|}{|P|}$ increases to 0.0076 while d_{min} of the codebook is reduced to 5. There is still higher probability of decoding $A_i \notin C$ for each of the 7 iterations. While the Hungarian algorithm's performance remains similar with hard-decision, 1dB gain is observed at the 3rd iteration. Subsequent iterations' respective performances are similar to the performance at the 3rd iteration with negligible gain in between.

4.2.3 1417 codewords

With $\frac{|C|}{|P|} = 0.035$, the probability of the decoder producing an output $A_i \notin C$ is reduced compared with 305 codewords. The effect of this is evident in Figure 4.8 where performance is improved at the second iteration with up to 1dB gain over

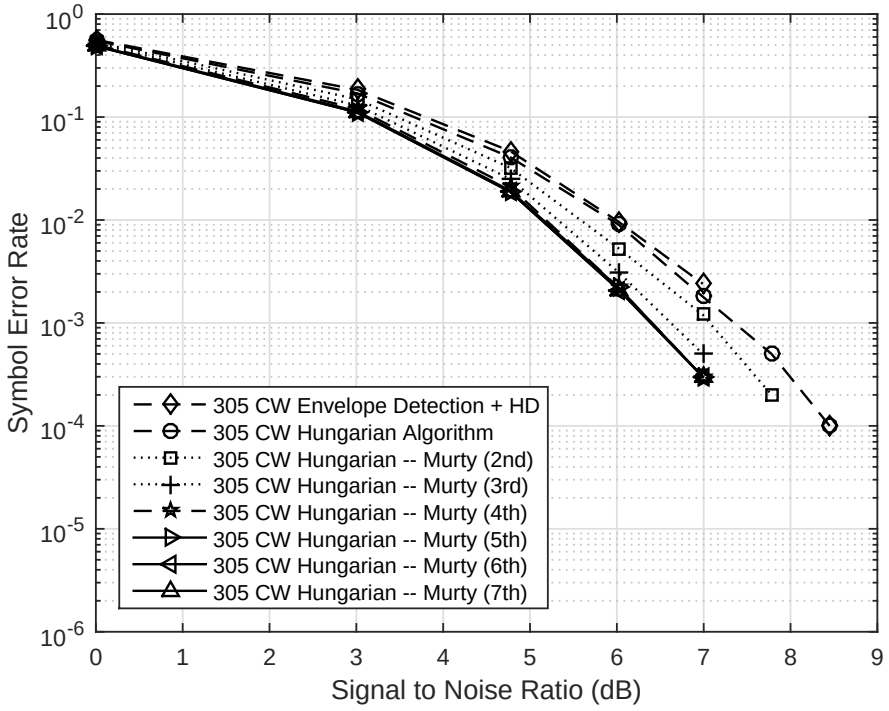

FIGURE 4.7: 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 5$

hard-decision. The algorithm however stops improving after the 3rd iteration which improves performance with 1.8dB gain.

4.2.4 20160 codewords

The d_{min} of this codebook is 3. As shown in Figure 4.9. The Hungarian algorithm's decoding performance out-performs hard-decision by 1dB unlike in codebooks with greater d_{min}. The second iteration further adds 1dB improvement to the Hungarian algorithm. The decoder however stops improving performance after the second iteration.

4.2.5 40320 codewords

The code rate of this codebook is 1.91, $d_{min} = 2$ and $|C| = |P|$. The performance of all iterations are equal as seen in Figure 4.10. This is also similar to the performance

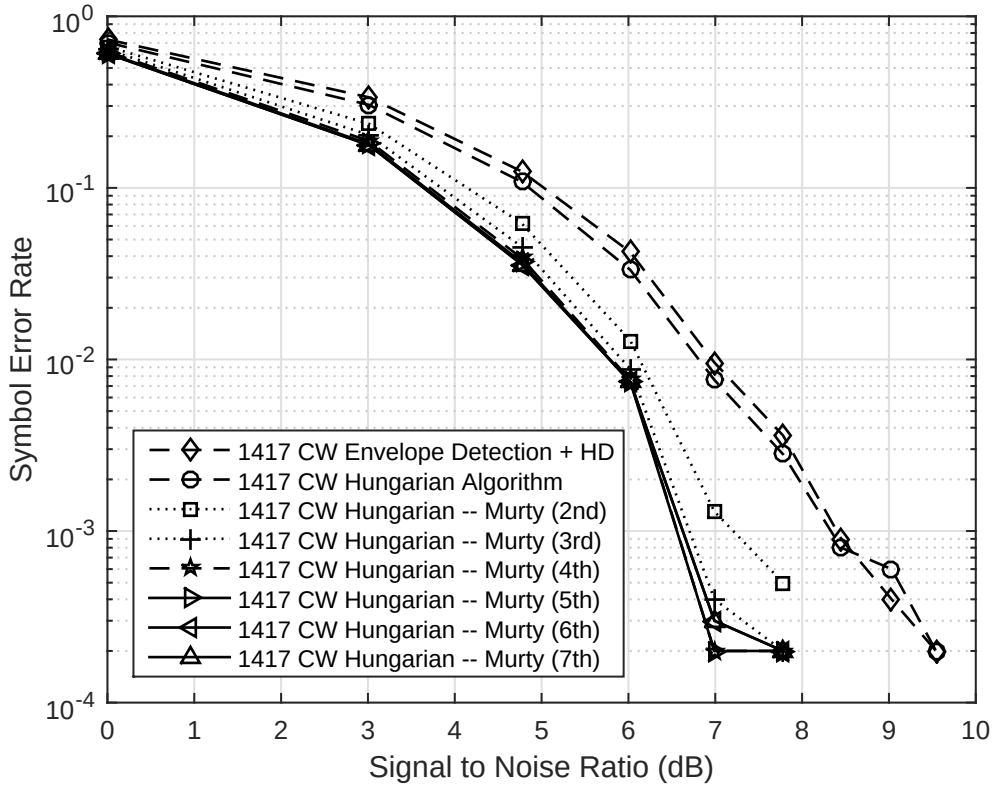

FIGURE 4.8: 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 4$

results observed in the 4FSK system in Figure 4.4. As explained earlier, the stopping condition of the decoder is to decode a codeword $A_i \in C$. This condition will always be satisfied at iteration of the Hungarian algorithm.

The performance of the Hungarian algorithm can therefore be used to represent the performance of all subsequent iterations in the decoder. Coding gain of the Hungarian algorithm compared with hard-decision is over 2dB in some SNR regions.

Table 4.2 summarises the properties of each codebook used in the 8FSK system. It also shows the performance of each iteration up to the seventh iteration. This performance is compared with the combination of the Envelope Detection and hard-decision. The plots in Figure 4.11 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain.

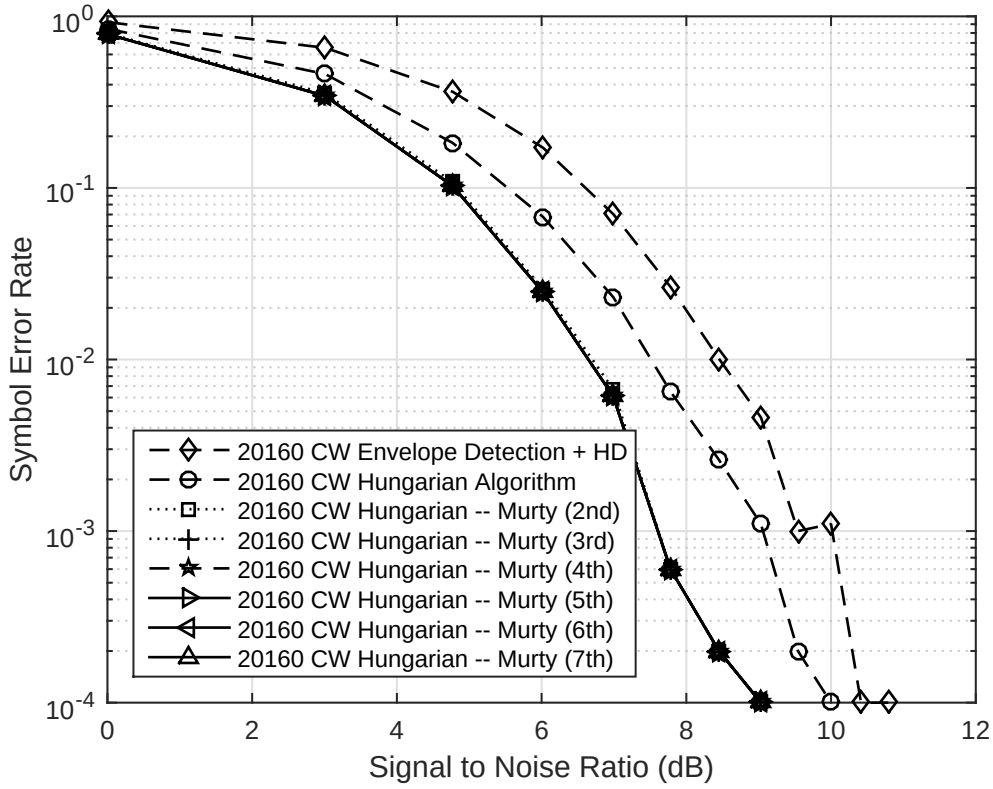

FIGURE 4.9: 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 3$

TABLE 4.2: Performance of Soft-decision Decoder using 8FSK in AWGN Channel, Coherent Detection

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4	**Gain (dB) at** A_5	**Gain (dB) at** A_6	**Gain (dB) at** A_7
8	0.125	0.0002	8	-0.2	0	0	0.1	0.1	0.2	0.2
305	0.344	0.0075	5	0.2	0.5	0.7	0.8	0.9	0.9	0.9
1417	0.436	0.035	4	0.2	0.8	1.2	1.4	1.4	1.4	1.4
5000	0.512	0.124	3	0.8	1.8	1.8	1.8	1.8	1.8	1.8
10000	0.554	0.0248	3	1.0	1.8	2.0	2.0	2.0	2.0	2.0
15000	0.578	0.372	3	1.0	1.8	2.0	2.0	2.0	2.0	2.0
20160	0.596	0.5	3	1.1	2.4	2.4	2.4	2.4	2.4	2.4
40320	0.637	1	2	2.5	2.4	2.4	2.4	2.4	2.4	2.4

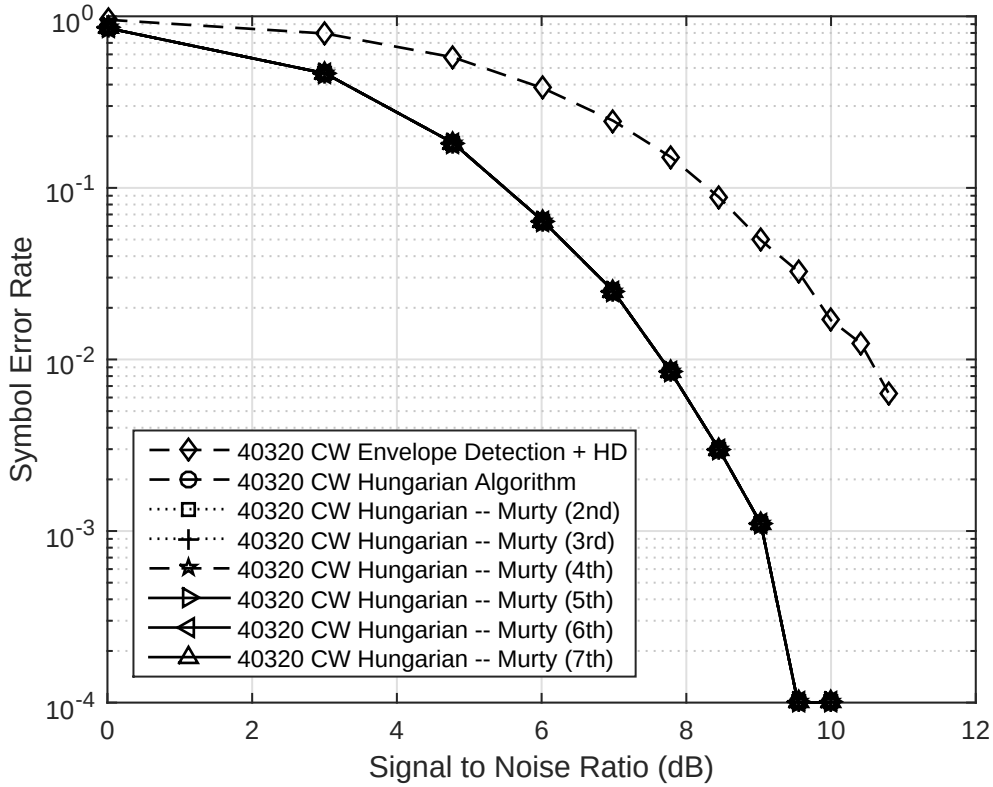

FIGURE 4.10: 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Coherent Detection. $d_{min} = 2$

4.3 MFSK in AWGN Channel - Noncoherent Detection

Due to the properties of noncoherent detection, it is expected that the performance of the decoder degrades as more uncertainty has been introduced by the random phase shifts. The system is therefore more random in nature unlike coherent detection because the receiver and transmitter are out of phase. Part of detection of the received signal is therefore predictive due to the random process involved. The following sections show the performance of the decoder when the received signal is not in phase with the transmitted signal.

The performance of each iteration remains similar to coherent detection in terms of whether or not the iteration out-performs hard-decision.

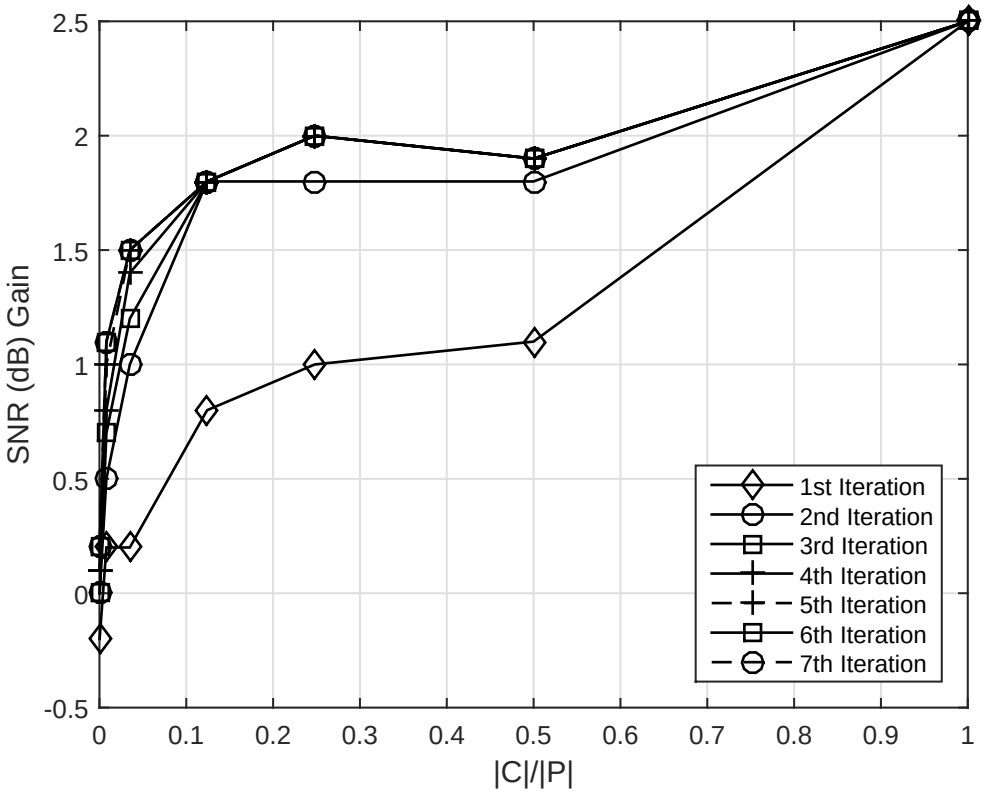

FIGURE 4.11: SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, Noncoherent Detection.

4.3.1 4 Codewords

As shown in Figure 4.12, the performance of the Hungarian algorithm remains similar in performance with hard-decision although negligible 0.1dB improvements are observed in some SNR regions. The second iteration adds 1dB gain to the decoder while the third and fourth iterations produce similar performance, adding some 0.5dB gain to the second iteration.

4.3.2 8 Codewords

Figure 4.13 shows that the second iteration improves the Hungarian algorithm by 1dB. Subsequent iterations however perform in very similarly manner with the second iteration.

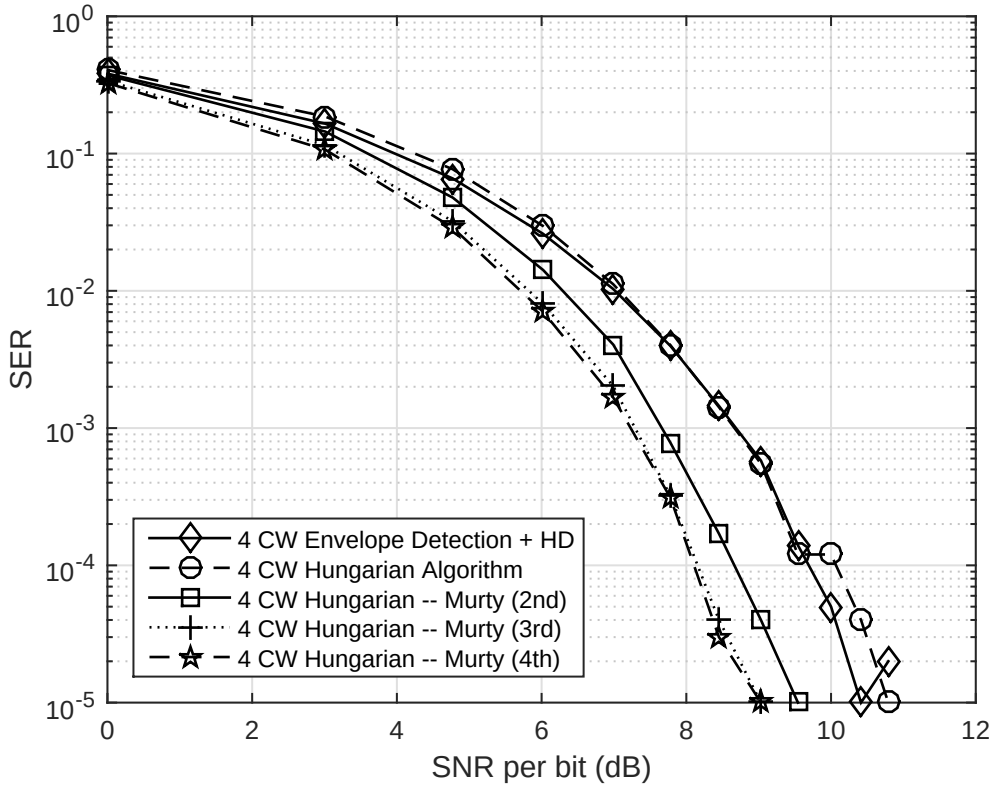

FIGURE 4.12: 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding. $d_{min} = 4$, code rate = 0.25

4.3.3 16 Codewords

Figure 4.14 shows a 2dB gain at the Hungarian algorithm iteration. Subsequent iterations however could only improve the Hungarian algorithm by 0.1dB. Similar to coherent detection, the performance is irrespective of the subset of P chosen to form the codebook $|C| = 16$.

4.3.4 24 Codewords

This codebook defines the optimum performance of the decoder. The best coding gain is observed right from the first iteration (Hungarian algorithm) which is about 2.3dB. Subsequent iterations are not necessary as shown in Figure 4.15.

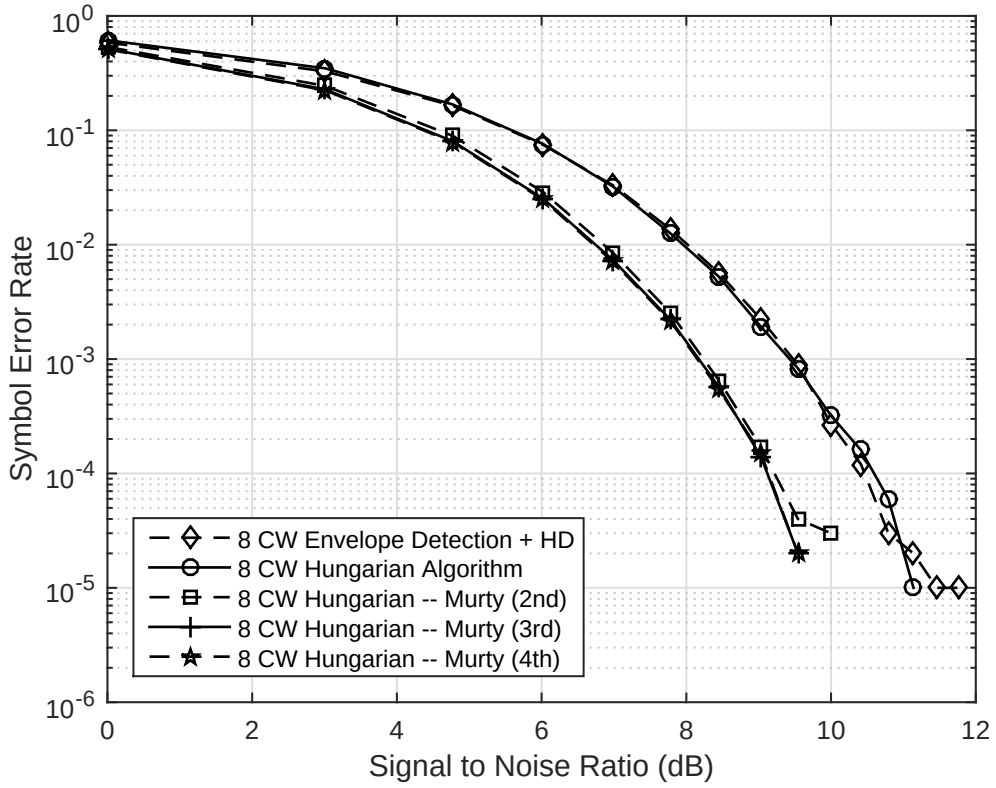

FIGURE 4.13: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 3$, code rate = 0.375

TABLE 4.3: Performance of Soft-decision Decoder using 4FSK in AWGN Channel, Noncoherent Detection

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4
4	0.25	0.1667	4	0.2	0.8	1.4	1.4
8	0.375	0.33	3	0.0	1.2	1.2	1.2
12	0.45	0.5	3	0.1	1.1	1.1	1.1
12	0.45	0.5	2	2	2.2	2.2	2.2
16	0.5	0.67	2	2.0	2.2	2.2	2.2
24	0.57	1	2	2.2	2.2	2.2	2.2

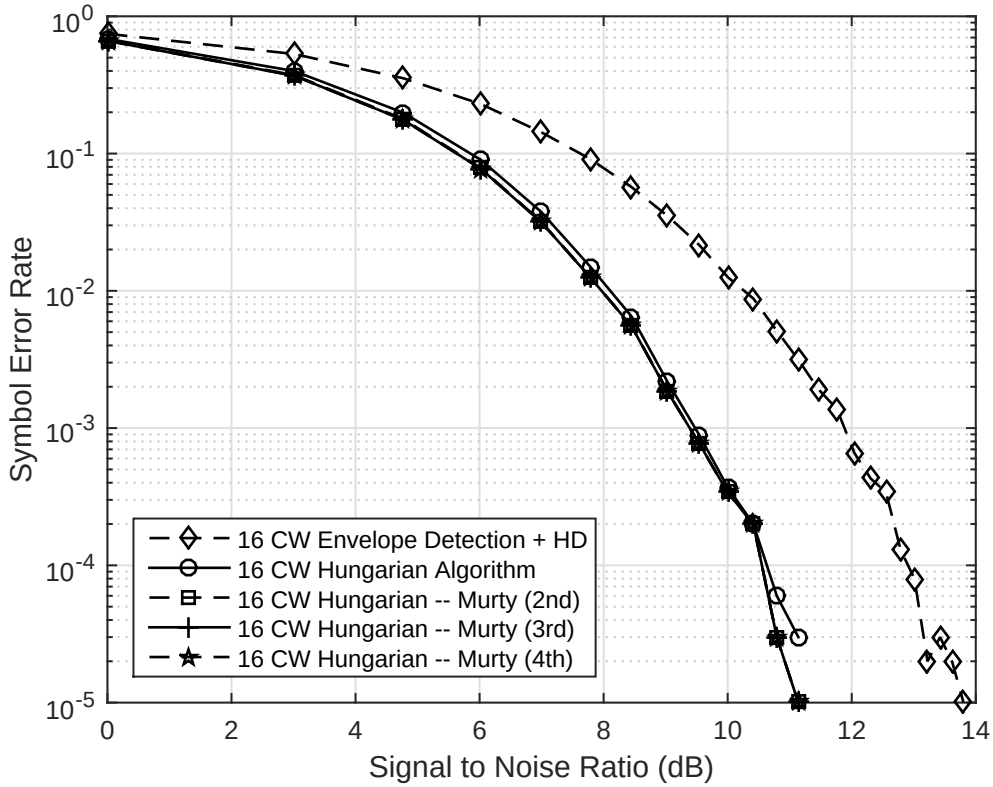

FIGURE 4.14: 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 2$, code rate = 0.5

Table 4.3 summarises the properties of each codebook used in the 4FSK noncoherent detection. It also shows the performance of each iteration up to the fourth iteration. This performance is compared with the combination of the Envelope Detection plus hard-decision.

The performance of codebook with $|C| = 16$ is irrespective of the subset of P chosen to form the codebook. Unlike the codebook with $|C| = 16$, the codebook with $|C| = 12$ on the other hand can be constructed with either $d_{min} = 2$ or $d_{min} = 3$. The performance of both codebooks differ and this is also associated with the size of d_{min} as observed in coherent detection. Table 4.3 shows that the decoder performs better when the d_{min} is 2 than when d_{min} is 3 for $|C| = 12$. The Hungarian algorithm's performance is negligible while considerable performance improvement is observed from the second iteration. This performance behaviour is quite similar to $|C| = 8$

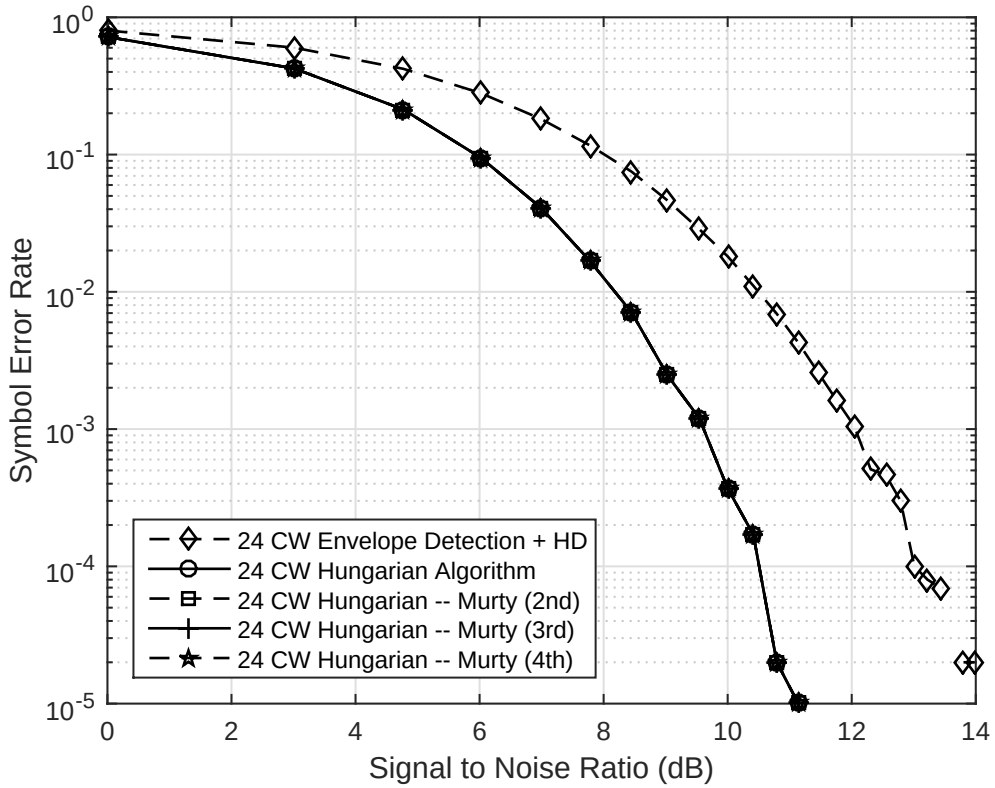

FIGURE 4.15: 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 2$, code rate = 1

codebook and the poorer performance of the decoder can be attributed to the higher d_{min}.

The plots in Figure 4.16 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain at each iteration.

4.4 8FSK

This section shows simulation results using 8FSK with codebooks each having $|C| = 8$. Performances are made among iterations up to the 7th iteration. It is also of interest to know how much the noncoherent property affects the decoder's performance compared with the noncoherent detection system.

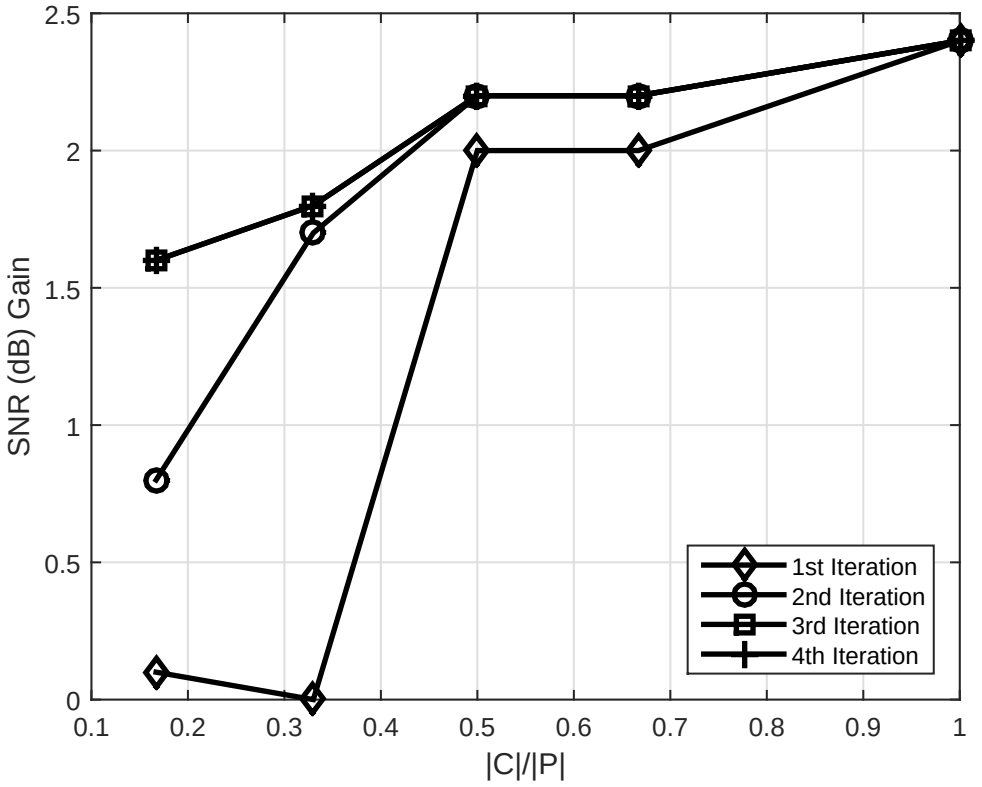

FIGURE 4.16: SNR (dB) Gain vs $\frac{|C|}{|P|}$ for First 4 Iterations for $|P| = 24$, 4FSK Modulation, Noncoherent Detection

4.4.1 8 Codewords

The codebook remains the cyclically rotated codebook in section 5.2, performance remains similar to coherent detection where hard-decision performs better than all other iterations as seen in Figure 4.17. This is as a result of the high d_{min} and low $\frac{|C|}{|P|}$ ratio which are advantages for hard-decision decoding. Hard-decision however is more computationally complex than the decoder because the hard-decision decoder relies on a lookup with all possible 40320 codewords in order to make decision.

4.4.2 305 codewords

The percentage of P used to form C increases to 0.75% while d_{min} of the codebook is reduced to 5. There is still higher probability of decoding $A_i \notin C$ for each of the 7

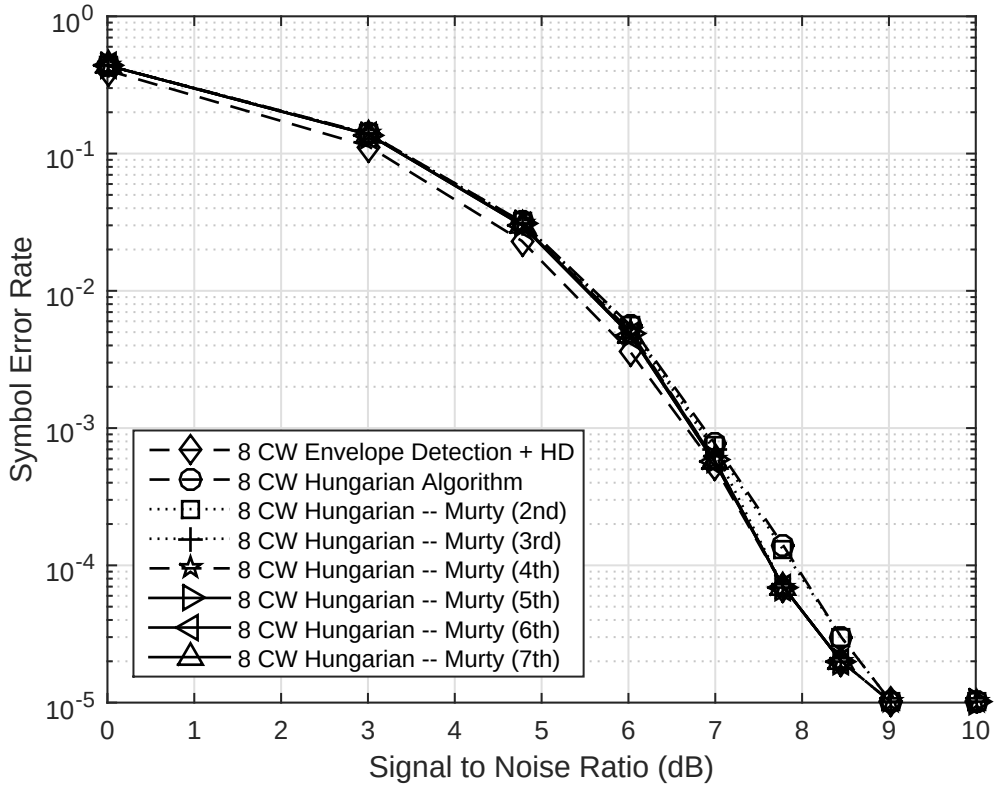

FIGURE 4.17: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 8$

iterations. While the Hungarian algorithm's performance remains similar with hard-decision, 1dB gain is achieved at the 3rd iteration. Subsequent iterations' respective performances are similar to the performance at the 3rd iteration.

4.4.3 1417 codewords

For a codebook P with each codeword of length $|C| = 8$, this codebook contains all possible codewords in P with $d_{min} = 4$. As shown in Figure 4.19, while the Hungarian algorithm's performance still remains similar with hard-decision, the second iteration improves performance up to 1dB. Third iteration improves the second iteration by up to additional 1dB. Further iterations up to the 7th iteration show minimal improvement in performance with an average of less than 0.1dB between them.

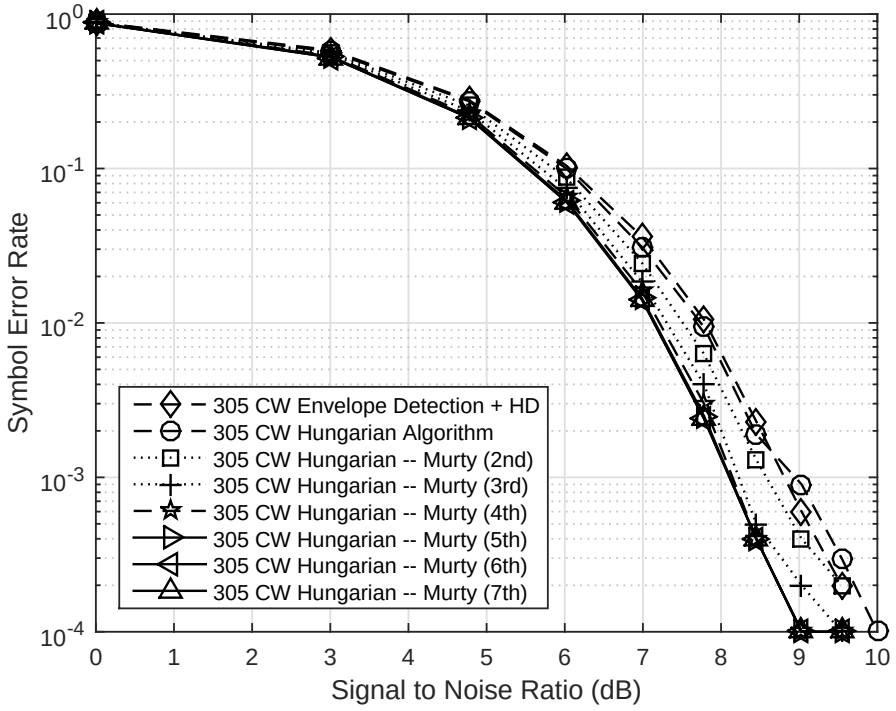

FIGURE 4.18: 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 5$

4.4.4 20160 codewords

This codebook utilises half of $|P|$. Therefore, the Hungarian algorithm still performs better than Envelope detection plus hard-decision but degrades to about 0.8dB unlike the 1dB gain in coherent detection. Subsequent iterations perform similarly but improve the Hungatrian algorithm with additional 1dB as shown in Figure 4.20.

4.4.5 40320 codewords

The optimum performance the highest iteration can produce in noncoherent detection is observed in this codebook. Figure 4.21. Approximately 2dB is observed from the Hungarian algorithm and as expected, further iterations do not add any improvements to the performance.

Table 4.4 summarises the properties of each codebook with different code rates for noncoherent detection of 8FSK. It also shows the performance of each iteration up

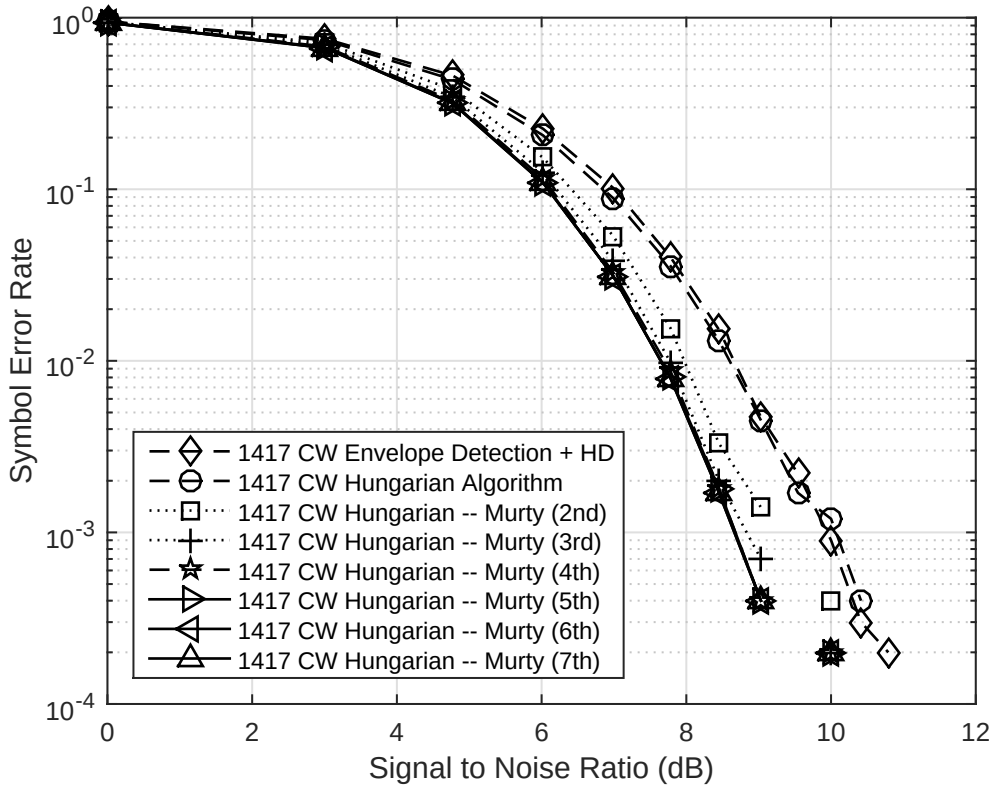

FIGURE 4.19: 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 4$

TABLE 4.4: Performance of Soft-decision Decoder using 8FSK in AWGN Channel, Noncoherent Detection

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4	**Gain (dB) at** A_5	**Gain (dB) at** A_6	**Gain (dB) at** A_7
8	0.125	0.0002	8	-0.3	-0.3	-0.2	-0.2	-0.2	-0.2	-0.2
305	0.344	0.0075	5	0	0.1	0.2	0.4	0.5	0.5	0.5
1417	0.436	0.035	4	0	0.8	1.0	1.0	1.0	1.0	1.0
5000	0.512	0.124	3	0.6	1.2	1.3	1.3	1.3	1.3	1.3
10000	0.554	0.248	3	0.6	1.3	1.4	1.4	1.4	1.4	1.4
15000	0.578	0.372	3	0.6	1.2	1.3	1.3	1.3	1.3	1.3
20160	0.596	0.5	3	0.8	1.5	1.6	1.6	1.6	1.6	1.6
40320	0.637	1	2	2.3	2.3	2.3	2.3	2.3	2.3	2.3

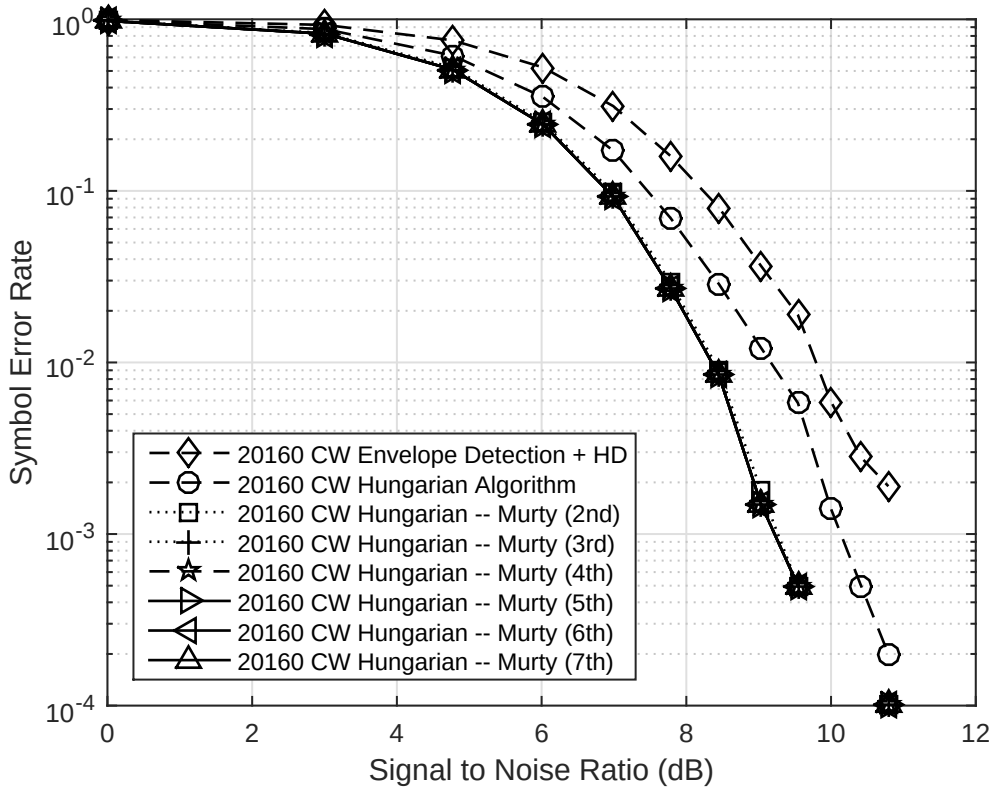

FIGURE 4.20: 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Channel, Noncoherent Detection. $d_{min} = 3$

to the seventh iteration. This performance is compared with the combination of the Envelope Detection and hard-decision. The plots in Figure 4.22 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain.

4.5 Noncoherent MFSK in AWGN and Rayleigh Fading Channels

In this section, the Rayleigh Slow Fading channel is added to the channel conditions which already includes AWGN. Transmitted signals are detected noncoherently and simulation results are discussed for 4FSK and 8FSK using different codebooks at the encoder.

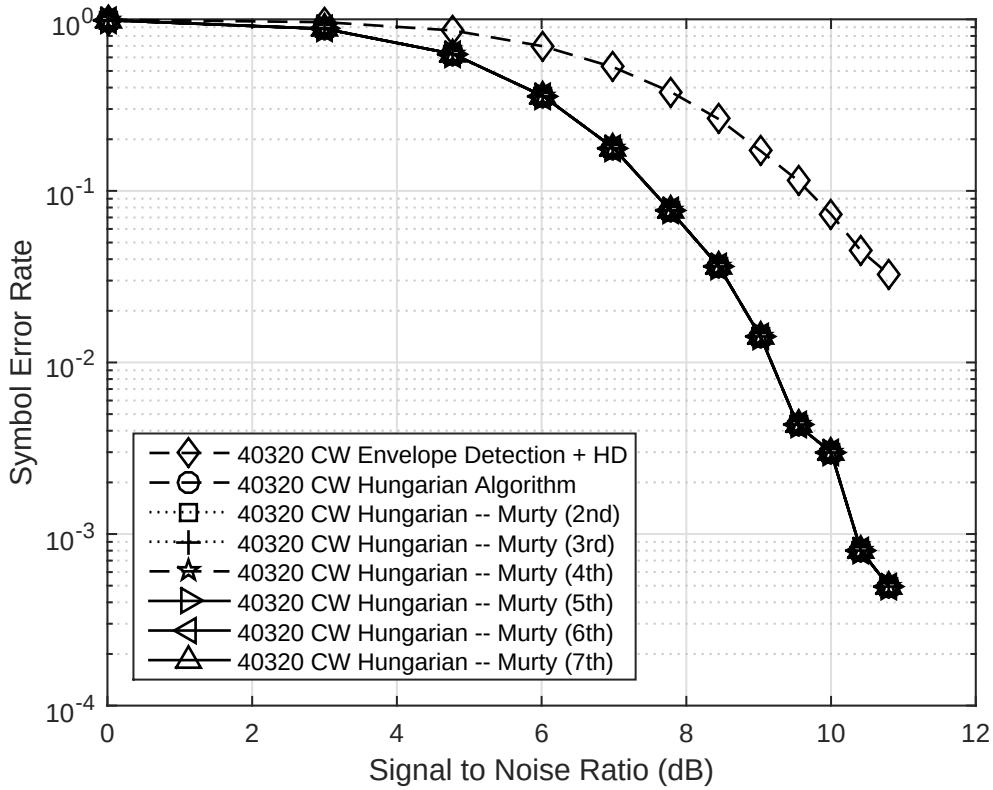

FIGURE 4.21: 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in AWGN Noncoherent Detection. $d_{min} = 2$

4.5.1 4 Codewords

Performance of the decoder improves significantly at the fourth iteration with approximately 1dB gain as shown in Figure 4.23. Previous iterations perform closely to Envelope Detection plus hard-decision with negligible dB gains between them.

4.5.2 8 Codewords

Increasing the code words and therefore the code rates improves the decoder's performance from the second iteration with about 1dB gain as shown in Figure 4.24. The decoder however fails to improve beyond the second iteration with subsequent iterations performing closely with the second iteration.

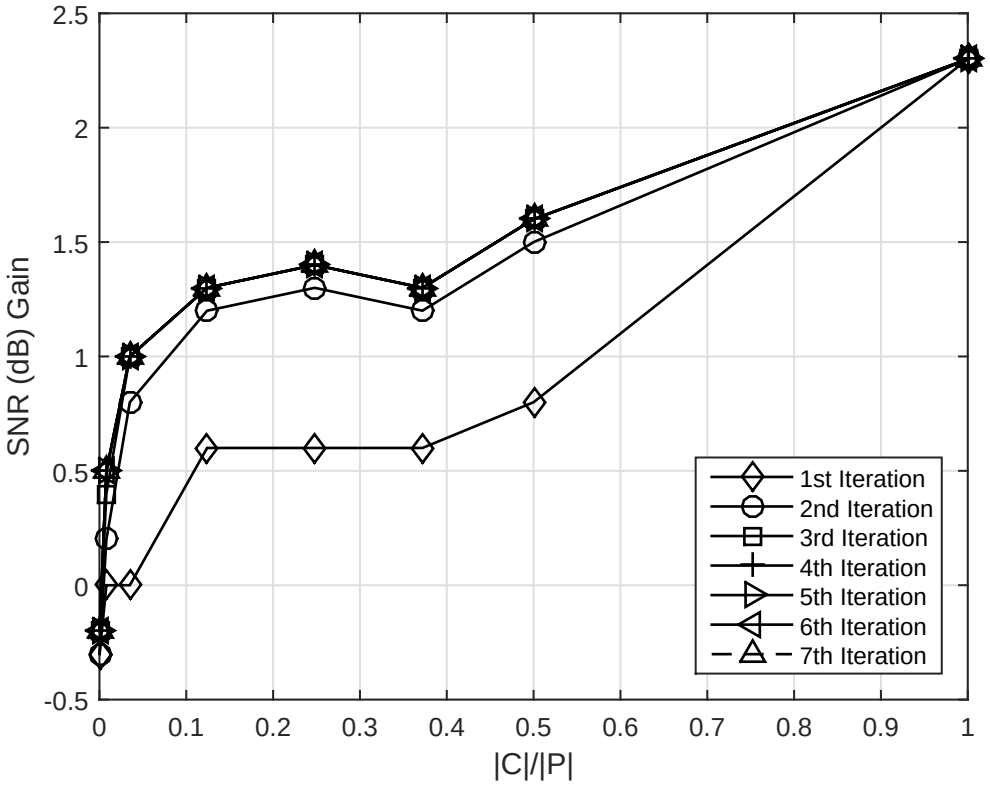

FIGURE 4.22: SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, Noncoherent Detection.

4.5.3 16 Codewords

With higher $\frac{|C|}{|P|}$, less codewords exist outside $|C|$. The Hungarian algorithm improves performance with more than 1dB gain compared with envelope detection plus hard-decision as shown in Figure 4.25. Subsequent iterations produce similar performance but improve the performance of the Hungarian algorithm, adding 0.2dB to the performance.

4.5.4 24 Codewords

The optimum performance of the decoder is observed in Figure 4.23 with about 2dB gain in some SNR regions.

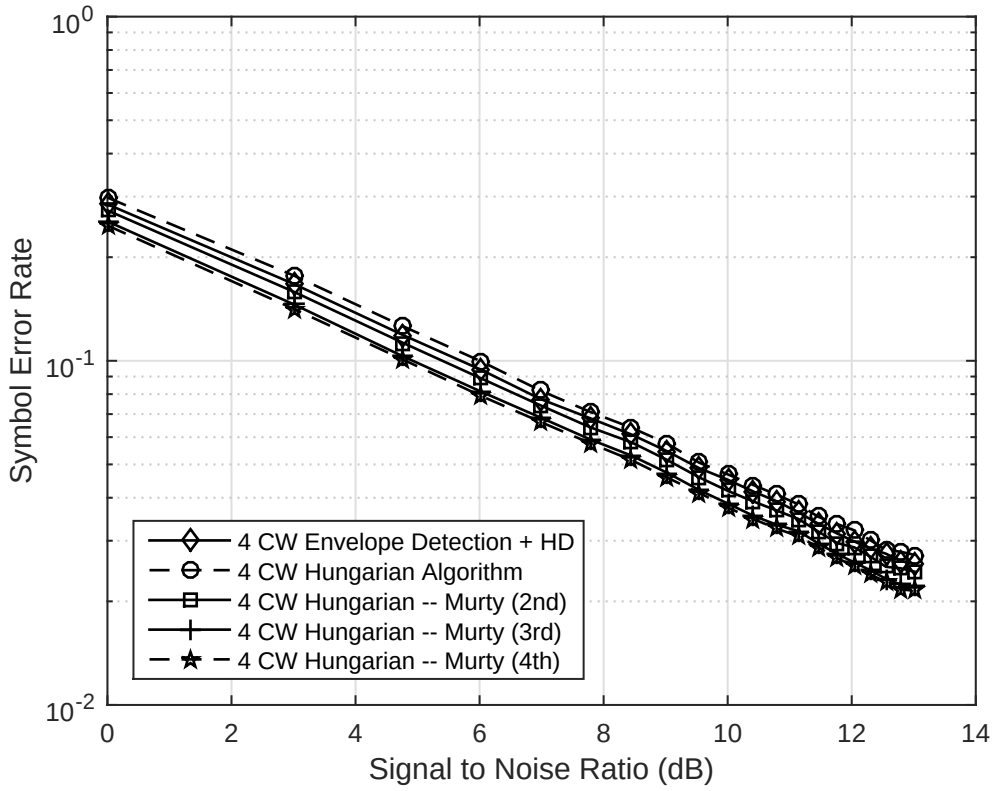

FIGURE 4.23: 4 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 4$, code rate = 0.25

Table 4.5 summarises the properties of each codebook used in the 4FSK system described. It also shows the performance of each iteration up to the third iteration. This performance is compared with the combination of the Envelope Detection and hard-decision.

The plots in Figure 4.27 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain.

4.6 8FSK

This section analyses the performance of the Soft-decision decoder in AWGN and Rayleigh Fading channels when the codebook size is increased, with $|C| = 8$. Generally, the performance of both soft-decision and hard-decision reduce in coding gain

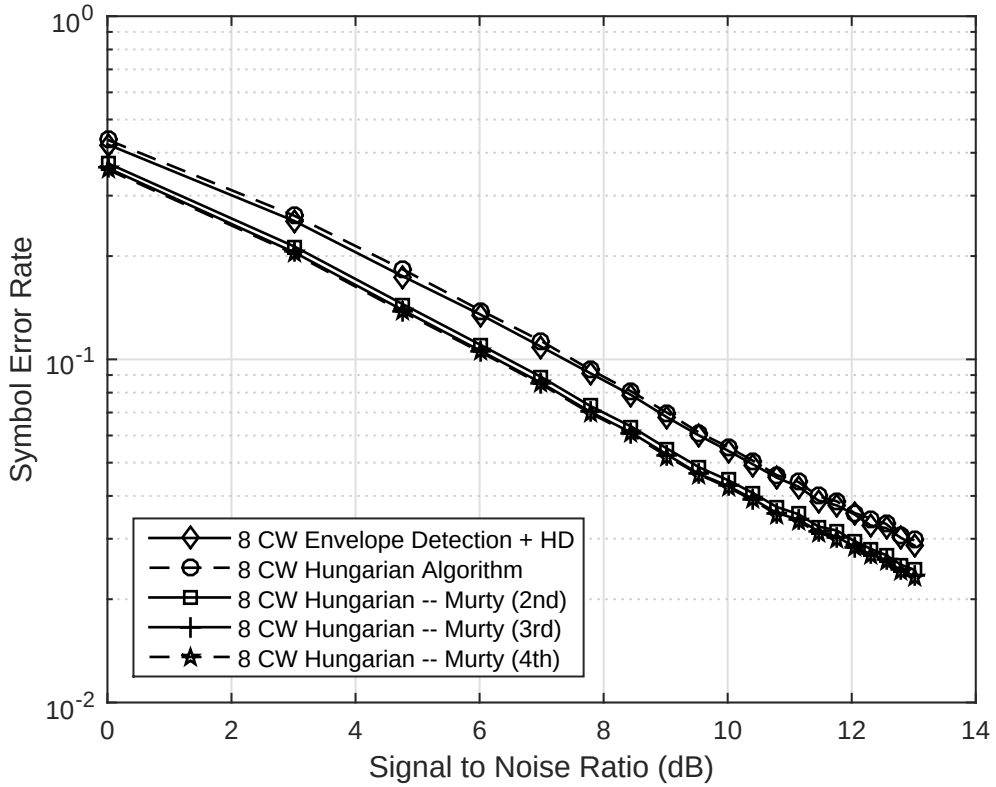

FIGURE 4.24: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 3$, code rate = 0.25

TABLE 4.5: Performance of Soft-decision Decoder using 4FSK in AWGN and Rayleigh Fading Channels

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4
4	0.25	0.1667	4	0.2	0.4	1.4	1.4
8	0.375	0.33	3	0.1	1.0	1.1	1.1
12	0.45	0.5	3	0.4	0.6	0.6	0.6
12	0.45	0.5	2	1.8	2.0	2.0	2.0
16	0.5	0.67	3	1.8	2.0	2.0	2.0
24	0.57	1	2	2.0	2.0	2.0	2.0

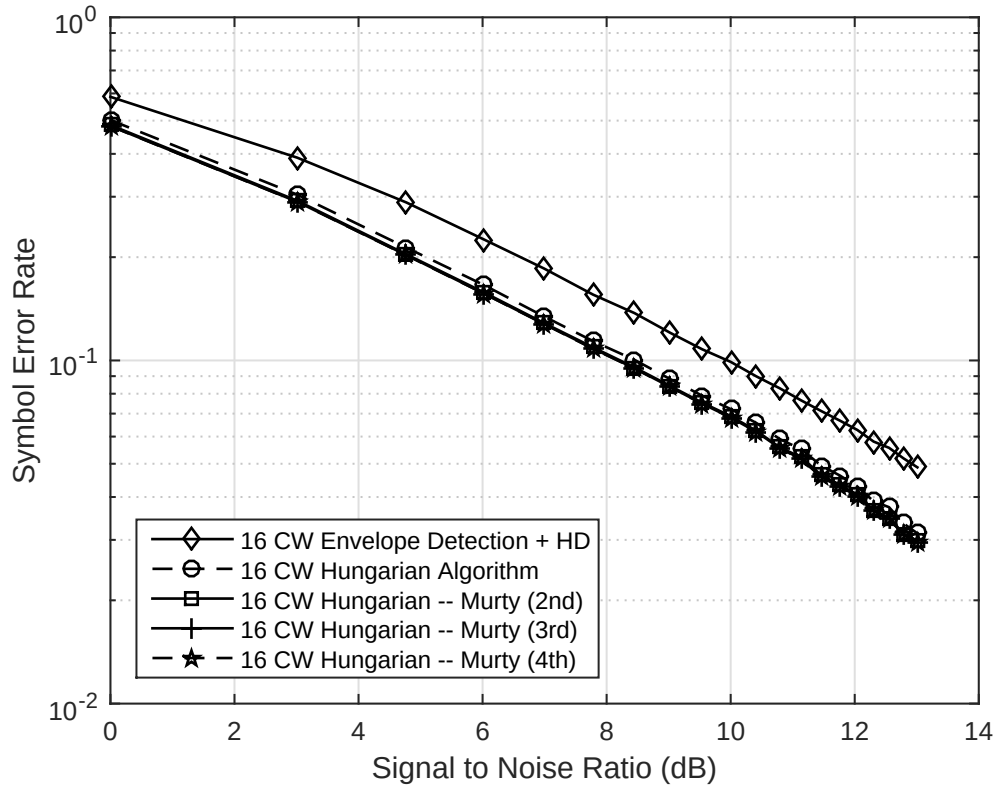

FIGURE 4.25: 16 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$, code rate = 0.25

compared with the performances obtained without the inclusion of Rayleigh Fading channel. However, the soft-decision decoder still out-performs the hard-decision decoder at high code rates. Hard-decision performs better than soft-decision when the code rate is very low.

4.6.1 8 Codewords

Hard-decision decoding out-performs all 7 iterations of the soft-decision decoder. At this code rate, $|C|/|P|$ is too low for the soft-decision decoder to out-perform hard-decision even after 7 iterations as shown in Figure 4.28. The computational complexity of hard-decision is also very low and therefore makes hard-decision a better decoding candidate at this code rate.

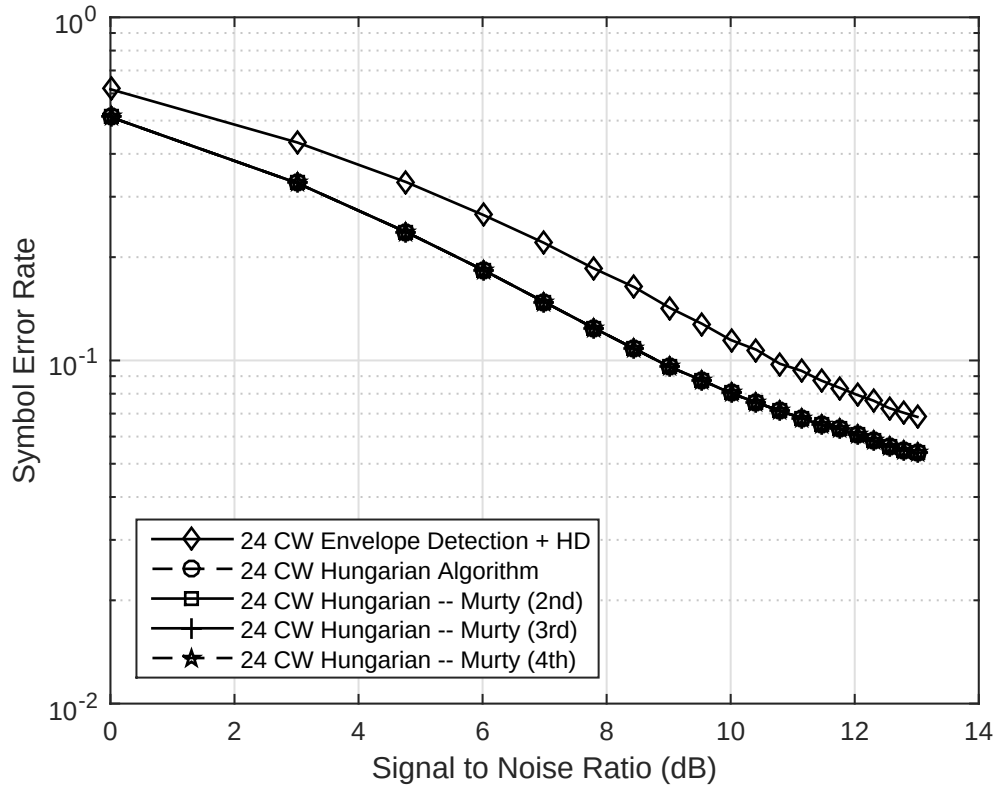

FIGURE 4.26: 24 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$, code rate = 0.25

4.6.2 305 codewords

The soft-decision out-performs the hard-decision decoder but only after the fourth iteration with 1dB gain as seen in Figure 4.29. Up until the second iteration, the hard-decision decoder still out-performs soft-decision decoder. $|C|/|P|$ is very low and therefore accounts for the poor performance of the soft-decision decoder.

4.6.3 1417 codewords

The performance of the first iteration and hard-decision are very similar because $|C|/|P|$ is very low. However, the performance becomes noticeable from the second iteration. An approximate dB gain of 1 is accounted for at the fourth iteration as seen in Figure 4.30.

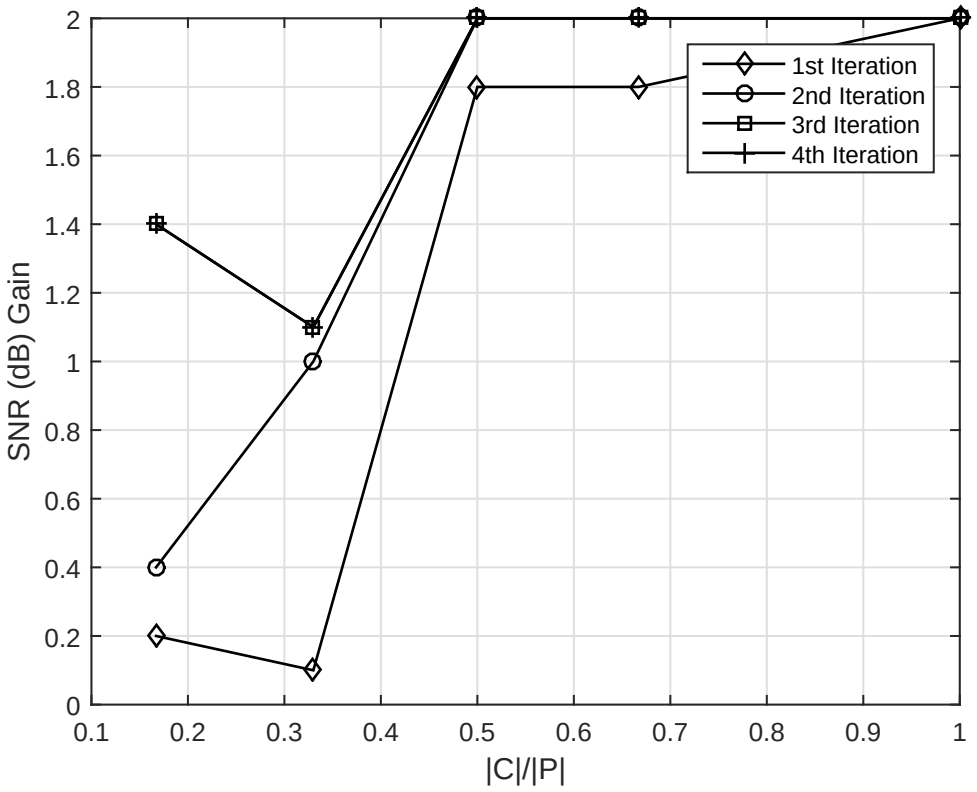

FIGURE 4.27: SNR (dB) Gain vs $|C|/|P|$ for First 4 Iterations for $|P| = 24$, 4FSK Modulation, AWGN and Rayleigh Fading Channels

4.6.4 20160 codewords

With half of the codewords used, the first iterations out-performs hard-decision with up to 2dB gain as shown in Figure 4.31. Subsequent iterations produce negligible improvement to the performance of the first iteration.

4.6.5 40320 codewords

The optimum performance of the soft-decision decoder is observed when all the codewords are used in encoding. The coding gain for all iterations are the same as seen in Figure 4.32 which is about 2dB gain in some SNR regions. It is however obvious that the Rayleigh channel dominates the channel noise, with a significant reduction in the performance of the decoders.

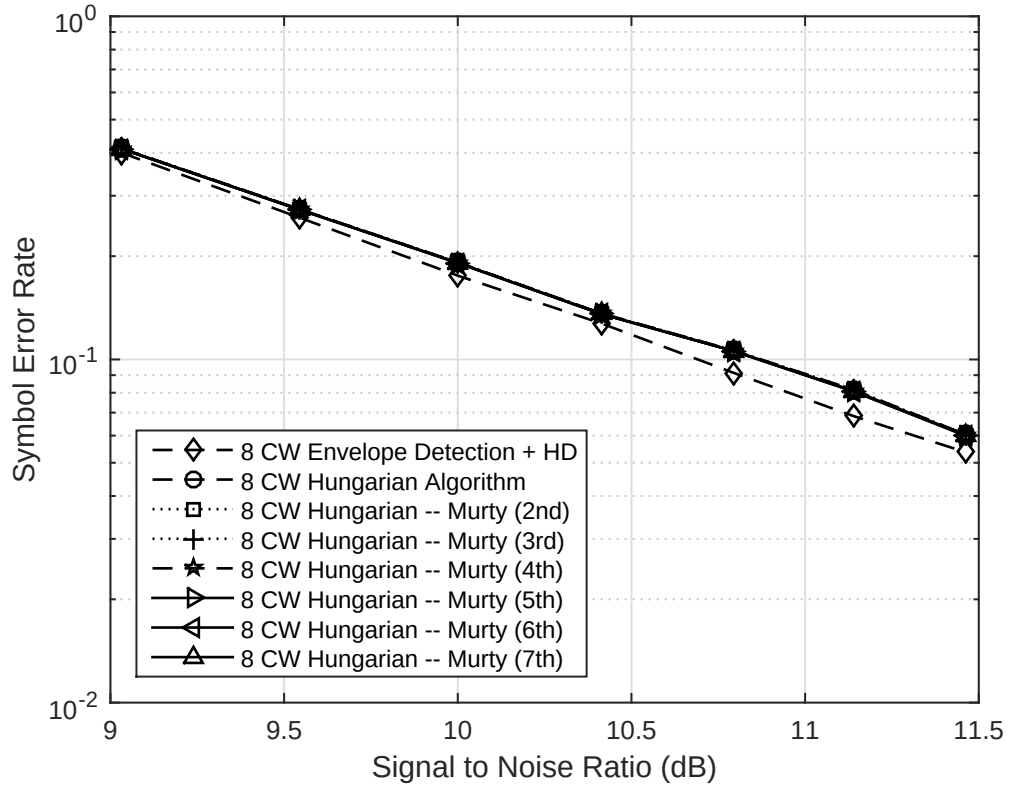

FIGURE 4.28: 8 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 8$

Table 4.6 summarises the performance of different codebooks selected from $|P| = 40320$. It also shows the performance of each iteration up to the seventh iteration. This performance is compared with the combination of the Envelope Detection and hard-decision. The plots in Figure 4.33 describe the results obtained by varying the $\frac{|C|}{|P|}$ ratio with the coding gain.

The reason why the next iteration tends to improve the performance of the decoder is that whenever an iteration produces an invalid codeword, the next iteration only has to find a codeword in $|C|$ and $|P| - 1$. The probability of $A_i \notin C$ is therefore reduced. The closer $\frac{|C|}{|P|}$ is to 1, the higher the probability of producing a codeword in C and therefore lessens the additional iterations required for improving coding gain performance.

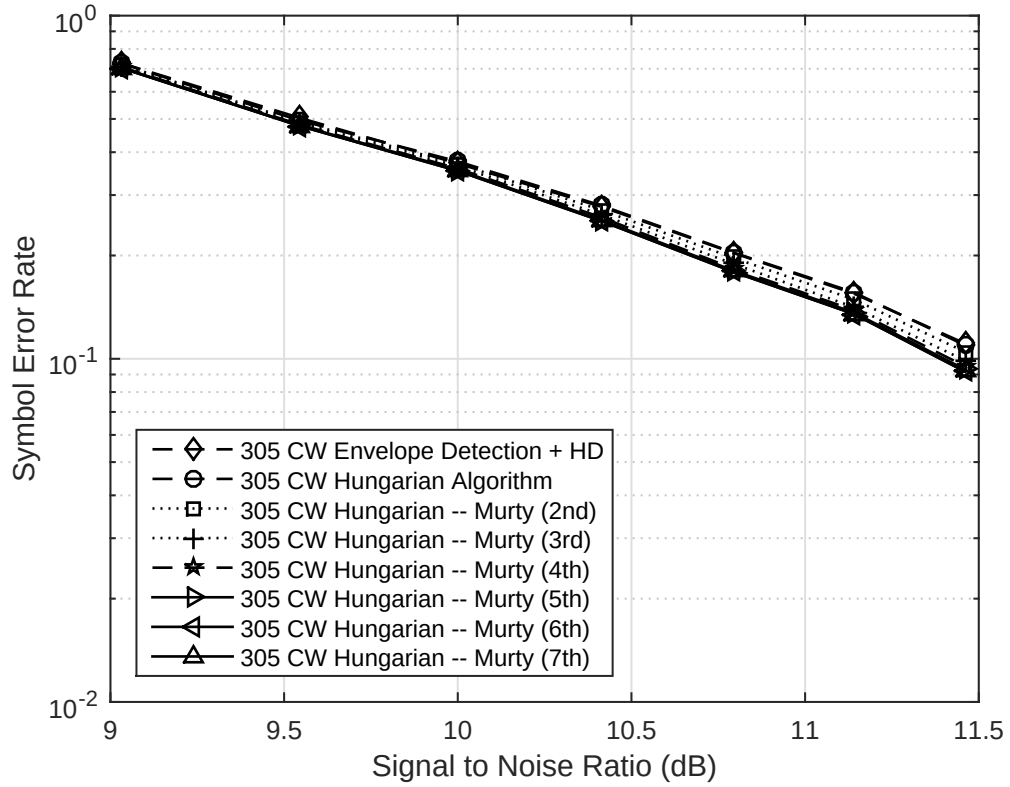

FIGURE 4.29: 305 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 5$

TABLE 4.6: Performance of Soft-decision Decoder using 8FSK in AWGN and Rayleigh Fading Channels

$\|C\|$	**Code rate**	$\frac{\|C\|}{\|P\|}$	d_{min}	**Gain (dB) at** A_1	**Gain (dB) at** A_2	**Gain (dB) at** A_3	**Gain (dB) at** A_4	**Gain (dB) at** A_5	**Gain (dB) at** A_6	**Gain (dB) at** A_7
8	0.125	0.0002	8	-0.3	-0.3	-0.2	-0.2	-0.2	-0.2	-0.2
305	0.344	0.0075	5	0	0.1	0.2	0.2	0.2	0.2	0.2
1417	0.436	0.035	4	0	0.4	0.5	0.5	0.5	0.5	0.5
5000	0.512	0.124	3	0.8	1.5	1.5	1.5	1.5	1.5	1.5
10000	0.554	0.248	3	1.0	1.5	1.5	1.5	1.5	1.5	1.5
15000	0.578	0.372	3	1.2	1.8	1.8	1.8	1.8	1.8	1.8
20160	0.596	0.5	3	1.8	1.8	1.8	1.8	1.8	1.8	1.8
40320	0.637	1	2	2.0	2.0	2.0	2.0	2.0	2.0	2.0

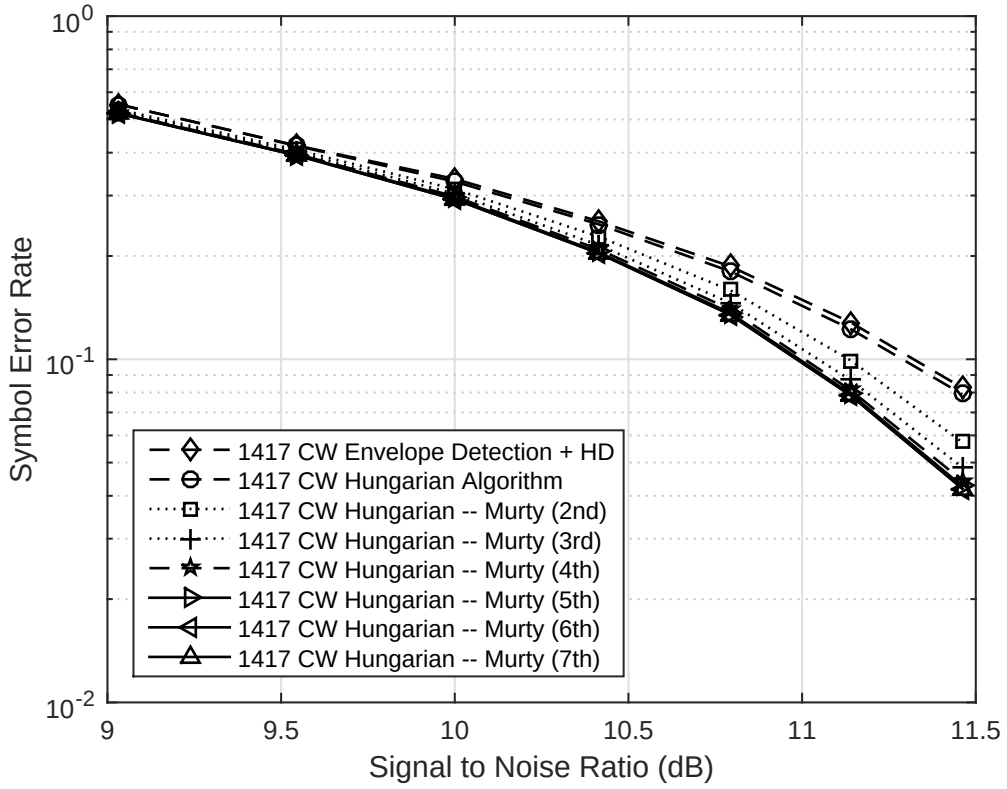

FIGURE 4.30: 1417 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 4$

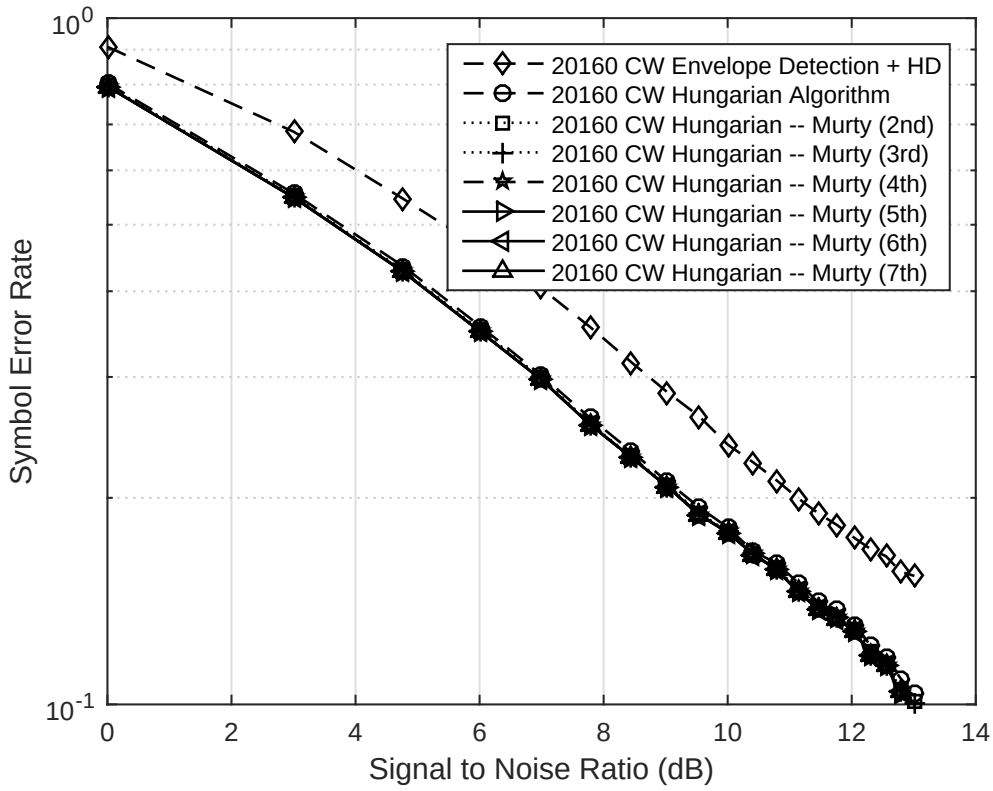

FIGURE 4.31: 20160 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 3$

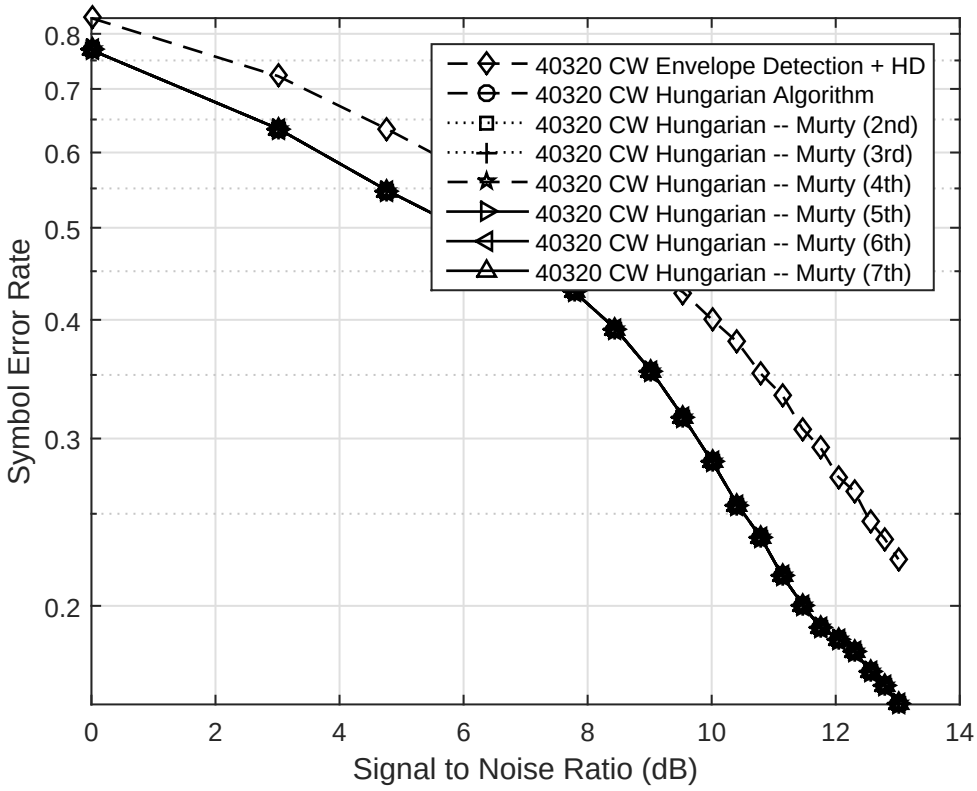

FIGURE 4.32: 40320 Codewords (CW): Performance of Hard-decision and Soft-decision Decoding in Noncoherent Detection, AWGN and Rayleigh Slow Fading Channels. $d_{min} = 2$

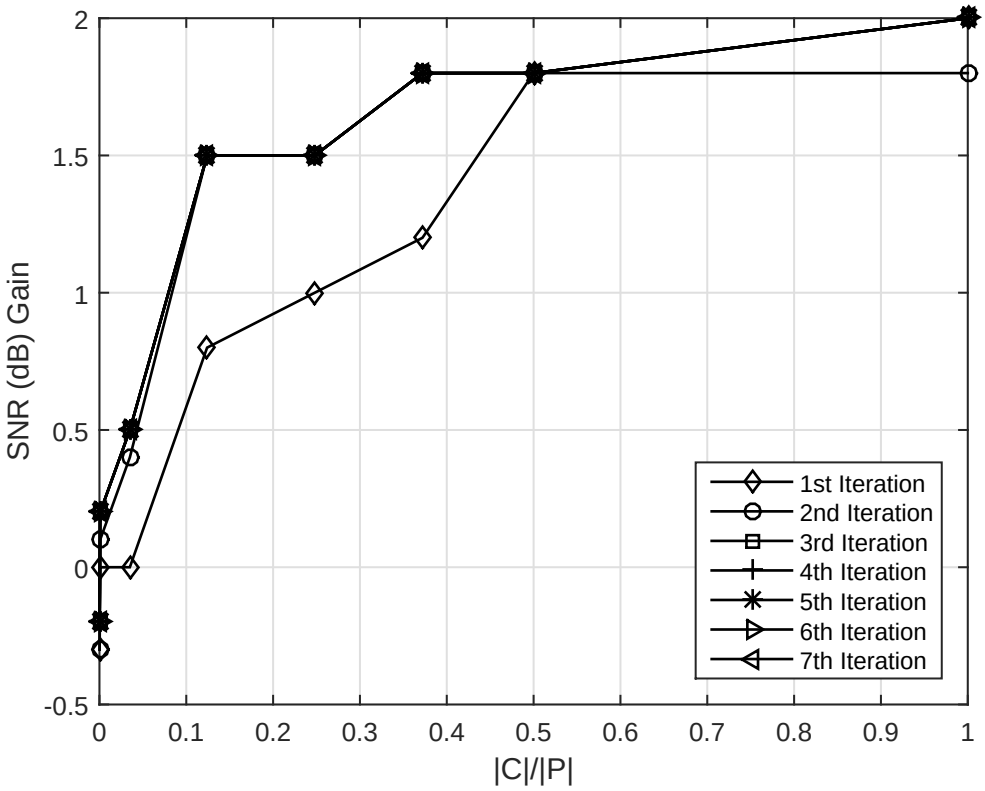

FIGURE 4.33: SNR (dB) Gain vs $|C|/|P|$ for First 7 Iterations for $|P| = 40320$, 8FSK Modulation, AWGN and Rayleigh Channels

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

We designed a soft-decision decoder to decode Permutation codes. The decoder combines the Hungarian algorithm (HA) for maximum assignment and Murty's algorithm (MA) for the k-th assignment. The Hungarian algorithm is the first attempt or first iteration by the decoder to correctly decode the noisy signal. Subsequent iterations are done by Murty's algorithm. Both algorithms were designed to solve the assignment problem. The solution to the assignment problem produces a permutation matrix which has similar characteristics with Permutation codes. The outcome of both algorithms in the decoder will therefore both produce a permutation codeword.

Simulations analysed the performance of the decoder using different codebook sizes compatible with 4FSK and 8FSK modulation. Each codebook differs from the other in terms of its coderate, therefore enables simulations be carried out for a variety of coderates. This analysis aims at understanding the performance of the decoder as we vary the amount of information that can be transmitted in the communication system.

For a set of all possible permutations P, a code book C can either be a subset of P or $C = P$. In either case, given a square signal matrix at the output of the AWGN

channel, the decoder produces the first assignment solution A_1 which is a Permutation codeword. Additional intelligence is built into the decoder such that it iteratively tries to find $A_i \in C \quad for \quad i > 1$. The performance of A_i is a combination of the performances of A_i and $A_{i-1} \quad for \quad i > 1$. Any codeword $A_i \notin C$ is an error and the decoder therefore proceeds to try decoding using the next iteration. Simulations carried out in this research stopped at the 4th iteration for 4FSK and 7th iteration for 8FSK.

Irrespective of the size of the codebook, the decoder will only require $M \times M$ signal matrix in order to decode a signal, M being the number of frequencies used for modulation. The computational complexity required to carry out this operation is $O(n^4)$. An alternative is a Maximum Likelihood (ML) soft-decision decoder that ranks all the costs in P for any given codebook C. The complexity required to achieve this is $O(n!)$. The soft-decision decoder derived from both the HA and MA algorithms therefore remains practical for code books of large sizes. The decoder only stops decoding a received signal once $A_i \in C$. It does not necessarily have to run all iterations.

Results compared the performance of the soft-decision decoder with Envelope Detection combined with minimum distance decoding by plotting SNR versus SER. The Hungarian algorithm improved coding gain at high rate codes. The largest gain and lowest complexity of the soft-decision decoder are observed when $C = P$. This is because the outcome of the HA always produces a Permutation codeword and therefore always satisfies the stopping condition $A \in C$.

5.2 Recommendations

This research is a able to determine the performance of the designed decoder compared with Hard decision. However, most simulations showed the decoder's performance did not increase significantly after the fourth iteration.

Simulations only stopped at the 7th iteration. It may be important to understand the performance for additional iterations. This research did not probe deep into the actual complexity each additional iteration adds to the decoder. It may be important to analyse the cost an additional iteration adds to the soft-decision decoder and determine if the coding gain justifies the added complexity. The result of this may help understand the computational cost required for each additional iteration.

Bibliography

[1] A. J. Vinck and J. Haring. "Coding and Modulation for Powerline Communications.", 2000.

[2] M. S. John G.Proakis. *Digital Communications*. McGraw-Hill, fourth ed., 1989.

[3] A. Mengi and A. J. H. Vinck. "Maximum-likelihood block decoding of noncoherent coded FSK for the CENELEC band." In *Power Line Communications and Its Applications, 2008. ISPLC 2008. IEEE International Symposium on*, pp. 117–122. April 2008.

[4] A. H. Vinck. "Coding for a Terrible Channel." *Telecommunications and Information Science and Technologies COST*, 2005.

[5] A. J. H. Vinck, J. Haering, and T. Wadayama. "Coded M-FSK for power line communications." In *Information Theory, 2000. Proceedings. IEEE International Symposium on*, pp. 137–. 2000.

[6] S. Huczynska. "Powerline communication and the 36 officers problem." *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 364, no. 1849, pp. 3199–3214, 2006. URL http://rsta.royalsocietypublishing.org/content/364/1849/3199.

[7] K. W. Shum. "Permutation coding and MFSK modulation for frequency selective channel." In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 5, pp. 2063–2066 vol.5. Sept 2002.

[8] Y. M. Chee and P. Purkayastha. "Efficient decoding of permutation codes obtained from distance preserving maps." In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 636–640. July 2012.

[9] T. G. Swart and H. C. Ferreira. "Decoding distance-preserving permutation codes for power-line communications." In *AFRICON 2007*, pp. 1–7. Sept 2007.

[10] D. Slepian. "Permutation modulation." *Proceedings of the IEEE*, vol. 53, no. 3, pp. 228–236, March 1965.

[11] M. Schroeder and N. Sloane. "New permutation codes using Hadamard unscrambling (Corresp.)." *IEEE Transactions on Information Theory*, vol. 33, no. 1, pp. 144–146, Jan 1987.

[12] R. Ahlswede and G. Dueck. "Good codes can be produced by a few permutations." *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 430–443, May 1982.

[13] A. J. H. Vinck and H. C. Ferreira. "Permutation trellis codes." In *Information Theory, 2001. Proceedings. 2001 IEEE International Symposium on*, pp. 279–. 2001.

[14] P. Keevash and C. Y. Ku. "A Random Construction for Permutation Codes and the Covering Radius." *Des. Codes Cryptography*, vol. 41, no. 1, pp. 79–86, Oct. 2006. URL http://dx.doi.org/10.1007/s10623-006-0017-3.

[15] M. C. Bali and C. Rebai. "Optimum receiver of coded M-FSK modulation for power line communications." In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6. June 2014.

[16] H. W. Kuhn. "The Hungarian method for the assignment problem." *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. URL http://dx.doi.org/10.1002/nav.3800020109.

[17] S. Lin and D. J. Costello. *Error Control Coding, Second Edition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.

[18] H. S. S. Matt L. Miller and I. J. Cox. "Optimizing Murty's Ranked Assignment Method." In *NEC Research Institute, Technical Report*. 1995-2008.

[19] K. G. Murty. "Letter to the EditorAn Algorithm for Ranking all the Assignments in Order of Increasing Cost." *Operations Research*, vol. 16, no. 3, pp. 682–687, 1968. URL http://dx.doi.org/10.1287/opre.16.3.682.

[20] L. Liu and D. Shell. "Assessing Optimal Assignment under Uncertainty: An Interval-based Algorithm." In *Proceedings of Robotics: Science and Systems*. Zaragoza, Spain, June 2010.

[21] Y. Bassil. "A Comparative Study on the Performance of Permutation Algorithms." vol. 1, no. 1, pp. 7–19, February 2012.

[22] I. J. Cox, M. L. Miller, R. Danchick, and G. E. Newnam. "A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 295–301, Jan 1997.