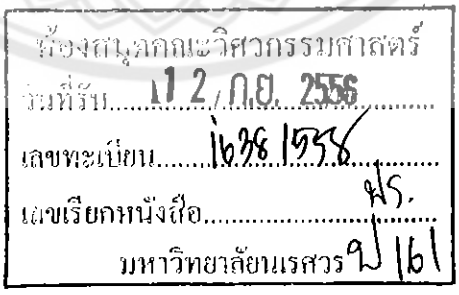
การจำลองการเข้ารหัสแบบ BCH CODES และถอดรหัสแบบ VITERBI ALGORLITHM

SIMULATION OF BCH ENCODING WITH VITERBI ALGORLITHM DECODING

นางสาวปนัดดา เอี่ยมชาวนา รหัส 50364669



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm
ผู้ดำเนินโครงงาน	นางสาวปนัดดา เอี่ยมชาวนา รหัส 50364669
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้า และคอมพิวเตอร์
ปีการศึกษา	2555

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวรอนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ
(ดร. ชัยรัตน์ พินทอง)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. อัครพันธ์ วงศ์กังแห)

ชื่อหัวข้อโครงงาน	การจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm
ผู้ดำเนินโครงงาน	นางสาวปนัดดา เอี่ยมชาวนา รหัส 50364669
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้า และคอมพิวเตอร์
ปีการศึกษา	2555

บทคัดย่อ

ในระบบการสื่อสารข้อมูล การเข้ารหัส-ถอดรหัสเพื่อแก้ไขความผิดพลาดของข้อมูลที่เกิดขึ้นระหว่างการส่งข้อมูลนั้นมีความสำคัญ เนื่องจากการส่งข้อมูลจากภาคส่งไปยังภาครับนั้นมักจะมีสัญญาณรบกวนเกิดขึ้น ซึ่งส่งผลให้ข้อมูลที่ได้รับมีความผิดพลาด ดังนั้นเพื่อลดการเกิดความผิดพลาดในการส่งข้อมูล จึงมีความจำเป็นต้องทำการเข้ารหัส-ถอดรหัส เพื่อให้สามารถแก้ไขข้อมูลที่ผิดพลาดให้มีความถูกต้อง

โครงงานนี้เป็นการสร้างแบบจำลองของการเข้ารหัส-ถอดรหัสข้อมูลด้วยโปรแกรม MATLAB โดยใช้การเข้ารหัสแบบ BCH Codes และทำการวิเคราะห์ประสิทธิภาพในการแก้ไขความผิดพลาดโดยการถอดรหัสแบบ Viterbi Algorithm ในกรณีการถอดรหัสแบบ Hard-decision

Project title	Simulation of BCH Encoding with Viterbi Algorithm Decoding
Name	Miss Panadda Eamchawna ID. 50364669
Project advisor	Assistant Professor Surachet Kanprachar, Ph.D.
Major	Electrical Engineering
Departmant	Electrical and Computer Engineering
Academic year	2012

Abstract

In data communication systems, encoding and decoding for correcting the errors occurring in the data transmission is significant. There is always noise which causes an error to the data transmitted between the transmitter and the receiver. To reduce the number of errors in the transmission, it is necessary to perform encoding and decoding to the signal. The corrupted signal received at the receiver can then be interpreted correctly.

In this project, a computer simulation of BCH encoding and decoding is done by using MATLAB. The error-correction performance of Hard-decision decoding with Viterbi algorithm is studied and analyzed.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงด้วยดี ด้วยการช่วยเหลือและสนับสนุนจากหลายๆท่านด้วยกัน ซึ่งผู้เขียนขอขอบพระคุณดังต่อไปนี้

ขอขอบพระคุณ ผศ.ดร.สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาโครงงาน ในการให้ความรู้ คำแนะนำ และคำปรึกษาที่เป็นประโยชน์อย่างสูงในการทำโครงงานนี้ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์ที่ดี และขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ ดร.ชัยรัตน์ พินทอง และ ผศ.ดร.อัครพันธ์ วงศ์กังแห ที่เสียสละเวลาอันมีค่าตรวจสอบแก้ไขข้อบกพร่องต่างๆ ในโครงงานนี้

ขอขอบพระคุณภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ประสิทธิ์ประสาทวิชาความรู้ และอบรมสั่งสอนให้ผู้จัดทำเป็นคนที่ดีของสังคม

ขอขอบคุณเพื่อนทุกคน น้องทุกคน ที่คอยให้ความช่วยเหลือ ให้กำลังใจ และให้คำปรึกษาในการทำโครงงานนี้จนสำเร็จลุล่วงได้เป็นอย่างดี

ขอกราบขอบพระคุณ คุณแม่ ผู้ให้กำเนิด และส่งเสริมให้ผู้จัดทำเดินทางมาถึง ณ จุดนี้ ได้อย่างราบรื่น

คุณค่า และประโยชน์อันพึงมีจากโครงงานนี้ ทางผู้จัดขอมอบแด่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

ปนัดดา เอี่ยมชาวนา

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์.....ก
บทคัดย่อภาษาไทย.....ข
บทคัดย่อภาษาอังกฤษ.....ค
กิตติกรรมประกาศ.....ง
สารบัญ.....จ
สารบัญตาราง.....ช
สารบัญรูป.....ซ

บทที่ 1 บทนำ.....1
1.1 ที่มาและความสำคัญของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....1
1.3 ขอบเขตของโครงงาน.....2
1.4 ขั้นตอนการดำเนินโครงงาน.....2
1.5 ผลที่คาดว่าจะได้รับ.....3
1.6 งบประมาณที่ต้องใช้.....4

บทที่ 2 หลักการ และทฤษฎีของระบบสื่อสาร.....5
2.1 การเข้ารหัสช่องสัญญาณ.....5
2.2 รหัสแบบบอส-โชดูรี-ฮอคเคนเกม (Bose-Chudhuri-Hocquenghem Codes: BCH).....6
2.3 การเข้ารหัสแบบ BCH.....6
2.4 การถอดรหัสแบบ Viterbi Decoder.....15

บทที่ 3 การออกแบบโครงงาน และวิธีการดำเนินงาน.....27
3.1 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm.....27
3.2 ขั้นตอนการออกแบบโปรแกรม.....27
3.3 การออกแบบ Graphic User Interfaces และขั้นตอนการดำเนินงาน.....28

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการดำเนินโครงงาน....33
4.1 โปรแกรมแสดงการแก้ไขข้อมูลผิดพลาด....33
4.2 โปรแกรมแสดงระดับอัตราการเกิดข้อมูลผิดพลาด....48

บทที่ 5 สรุปผลการดำเนินโครงงาน
5.1 ผลการดำเนินโครงงาน....55
5.2 ปัญหาที่พบขณะดำเนินโครงงาน....55
5.3 ข้อเสนอแนะ....55

เอกสารอ้างอิง....56
ประวัติผู้ดำเนินโครงงาน....57

สารบัญตาราง

ตารางที่ หน้า

2.1 Shift Register รหัสในรูป Systematic ของรหัส BCH(7, 4).....8

2.2 ตารางแสดงรายละเอียดการเปลี่ยนสถานะเข้ารหัส BCH(7, 4).....17

2.3 ตารางแสดงเส้นทางในการถอดรหัสแบบ Hard-decision.....20

2.4 ตารางแสดงการถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น $m_i = [1\ 1\ 0\ 1]$
บิตผิดพลาด 0 บิต.....23

2.5 การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น $m_i = [1\ 1\ 0\ 1]$
บิตผิดพลาด 1 บิต.....24

2.6 การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น $m_i = [1\ 1\ 0\ 1]$
บิตผิดพลาด 2 บิต.....25

สารบัญรูป

รูปที่ หน้า

2.1 รูปแบบของรหัสในรูป Systematic ของรหัส BCH มีขั้นตอนของ *(n- k)* shift-register.....7

2.2 รูปแบบของรหัสในรูป Systematic ของรหัส BCH(7, 4).....8

2.3 แผนภาพสถานะ (State Diagram) ของวงจรเข้ารหัส BCH(7, 4).....18

2.4 เส้นทางในการถอดรหัสแบบ Hard-decision.....19

2.5 การถอดรหัสแบบ Hard-decision ซึ่งเกี่ยวข้องกับรหัส BCH (7, 4).....21

3.1 Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสและถอดรหัส.....28

3.2 Graphic User Interfaces ในการเลือกแสดงตัวอย่าง หรือคำนวณ Bit Error Rate.....29

3.3 รูปแบบของ Graphic User Interfaces แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดของ
การเข้ารหัสสัญญาณ BCH Codes และการถอดรหัสแบบ Hard-decision.....30

3.4 Graphic User Interfaces เพื่อแสดงระดับการเกิดข้อผิดพลาดของ Hard-Decision.....31

4.1 หน้าต่างสำหรับการเลือกอัตราการเข้ารหัสช่องสัญญาณ.....33

4.2 หน้าต่าง Application ใช้ในการเลือกระหว่างหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด
และหน้าต่างแสดงระดับอัตราการเกิดข้อมูลผิดพลาด.....34

4.3 หน้าต่างการแก้ไขข้อมูลที่ผิดพลาด.....35

4.4 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1.....36

4.5 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2.....36

4.6 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3.....37

4.7 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1.....38

4.8 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2.....38

4.9 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 3.....39

4.10 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 1.....40

4.11 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2.....40

4.12 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3.....41

4.13 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1.....42

4.14 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2.....42

4.15 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3.....43

4.16 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1.....44

4.17 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2.....44

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.18 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 3.....45
4.19 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 1.....46
4.20 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2.....46
4.21 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3.....47
4.22 หน้าต่างแสดงการหน้าต่างที่ใช้แสดงระดับอัตราการเกิดข้อมูลผิดพลาด.....48
4.23 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100.....50
4.24 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 1000.....51
4.25 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 10000.....52
4.26 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100000.....53

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในการออกแบบหรือใช้งานระบบสื่อสารแบบดิจิตอลนั้น ต้องมีการพิจารณาถึงองค์ประกอบในหลายส่วนด้วยกัน โดยสิ่งหนึ่งที่ต้องมีการพิจารณาคือ ข้อมูลที่ถูกส่งจากต้นทางไปยังปลายทางนั้น มีข้อมูลที่เกิดความผิดพลาดขึ้นหรือไม่ ซึ่งสาเหตุหลักที่ทำให้เกิดความผิดพลาดคือการที่ระบบสื่อสารนั้นถูกรบกวนจากสัญญาณรบกวนต่างๆ ถ้าหากว่าขนาดของสัญญาณรบกวนที่เกิดขึ้นในระบบสื่อสารนั้นมีค่าสูง จะส่งผลให้อัตราการเกิดความผิดพลาดของข้อมูล (Bit Error Rate) มีค่าสูงตามไปด้วย ในการลดอัตราการเกิดความผิดพลาดของข้อมูลให้มีค่าที่ลดลง สามารถทำได้หลายรูปแบบด้วยกัน เช่น เพิ่มกำลังของเครื่องส่ง, การทำให้ขนาดของสัญญาณรบกวนมีค่าน้อยลง หรือการเข้ารหัสข้อมูล

การเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm เป็นการเข้ารหัสข้อมูลในระบบสื่อสารรูปแบบหนึ่ง ที่มีวัตถุประสงค์ในการทำงานเพื่อลดความผิดพลาดของข้อมูลที่เกิดขึ้นระหว่างการส่งผ่านระบบสื่อสารที่มีผลมาจากการถูกสัญญาณรบกวนต่างๆ การเข้ารหัสแบบ BCH Codes ถูกค้นพบในปี พ.ศ. 2502-2503 (ค.ศ.1959-1960) โดย Hocquenghem และพัฒนาต่อโดย Bose และ Chaudhuri ซึ่งการถอดรหัส Viterbi ได้ถูกพัฒนาจากรหัสแบบคอลโวลูชัน เริ่มเป็นครั้งแรกใน พ.ศ.2498 (ค.ศ.1955) โดย P.Elias ผลที่ได้จากการเข้ารหัสนี้ทำให้ข้อมูลต่างๆที่ถูกส่งผ่านระบบสื่อสารนั้นมีอัตราการเกิดความผิดพลาดที่น้อยลง

ในโครงงานนี้ เป็นการนำเสนอการศึกษาหลักการพื้นฐานและแบบจำลองการทำงานการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm ด้วยโปรแกรม MATLAB เพื่อนำไปประยุกต์ใช้งานให้เกิดประโยชน์ต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.2.2 เพื่อสร้างแบบจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.2.3 เพื่อศึกษาการเขียนโปรแกรมด้วย โปรแกรม MATLAB

1.3 ขอบเขตของโครงงาน

1.3.1 เพื่อศึกษาทฤษฎีเกี่ยวกับการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.3.2 เพื่อสร้างแบบจำลองการทำงานของการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.3.3 เพื่อศึกษาประสิทธิภาพของการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm ในการป้องกันความผิดพลาดของข้อมูลที่ถูกส่งผ่านระบบสื่อสาร

1.4 ขั้นตอนและแผนการดำเนินงาน

1.4.1 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.4.2 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการใช้ โปรแกรม MATLAB

1.4.3 ออกแบบโดยใช้โปรแกรม MATLAB ในการสร้างแบบจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.4.4 สรุปผลการทำงาน

1.4.5 จัดทำปริญญานิพนธ์และนำเสนอผลงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินโครงงาน

ขั้นตอนการดำเนินงาน	ปี 2555					ปี 2556		
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.4.1 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm	←	—	→					
1.4.2 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการใช้ โปรแกรม MATLAB				←	→			
1.4.3 ออกแบบโดยใช้โปรแกรม MATLAB ในการสร้างแบบจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm						←	→	
1.4.4 สรุปผลการทำงาน								← →
1.4.5 จัดทำปริญญานิพนธ์และนำเสนอผลงาน								← →

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1.5.1 มีความรู้ความเข้าใจการทำงานของการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.5.2 มีโปรแกรมคอมพิวเตอร์ที่สามารถแสดงผลการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

1.5.3 มีความรู้และทักษะในการเขียนโปรแกรมด้วยโปรแกรม MATLAB

1.6 งบประมาณ

1.6.1 ค่าเอกสารประกอบการทำปริญญานิพนธ์	700 บาท
1.6.2 จัดทำรูปเล่มปริญญานิพนธ์	300 บาท
รวมเป็นเงินทั้งสิ้น	1,000 บาท
	(หนึ่งพันบาทถ้วน)

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีของระบบสื่อสาร

ในปัจจุบันการสื่อสารโทรคมนาคมและสารสนเทศได้เข้ามามีบทบาท และเป็นปัจจัยที่สำคัญในการดำรงชีวิตประจำวัน ความสะดวกสบายที่เกิดจากการใช้เทคโนโลยีสื่อสารโทรคมนาคมและสารสนเทศ อาทิ ระบบโทรศัพท์เคลื่อนที่ ระบบอินเตอร์เน็ตความเร็วสูง และการกระจายหรือรับข่าวสารผ่านสื่อวิทยุและโทรทัศน์ ทำให้มีความต้องการการใช้งานที่เพิ่มขึ้นอย่างต่อเนื่อง ดังนั้นการพัฒนาศักยภาพของระบบสื่อสารให้สามารถรองรับการใช้งานรูปแบบต่างๆ ได้อย่างเหมาะสมและมีประสิทธิภาพมากขึ้น โดยเฉพาะการพัฒนาประสิทธิภาพการรับส่งข้อมูลในระบบสื่อสารเชิงดิจิตอล

ในการส่งข้อมูลในรูปแบบของข้อมูลดิจิตอลผ่านระบบสื่อสาร จะต้องมีการพิจารณาถึงองค์ประกอบต่างๆที่จะมีผลกับระบบสื่อสาร โดยที่สิ่งหนึ่งที่มีการพิจารณาได้แก่ การเกิดความผิดเพี้ยนของสัญญาณข้อมูลที่ถูกส่งมาจากภาคส่งถึงภาครับ เนื่องจากมีการถูกรบกวนจากสัญญาณรบกวน และเกิดการแทรกสอดระหว่างสัญลักษณ์ขึ้น โดยจะเป็นสาเหตุสำคัญที่ส่งผลทำให้ข้อมูลที่ตรวจจับได้นั้น เกิดความผิดพลาดตัวอย่างเช่นในกรณีของระบบสื่อสารไร้สาย ทั้งในปัจจุบันและในอนาคตนั้น จะเป็นระบบสื่อสารที่มีความต้องการในการส่งข้อมูลด้วยอัตราเร็วที่สูงขึ้น เพื่อให้สามารถรองรับการทำงานในรูปแบบต่างๆได้ ดังนั้นเพื่อทำให้ระบบสื่อสารที่ใช้งานนั้นมีความน่าเชื่อถือมากยิ่งขึ้น จึงได้มีการพัฒนาวิธีการขึ้นมาเพื่อใช้สำหรับลดความผิดเพี้ยนของสัญญาณที่ถูกส่งผ่านช่องสัญญาณให้มีค่าลดลง เพื่อให้สามารถส่งข้อมูลผ่านระบบสื่อสารได้ด้วยอัตราเร็วสูงได้ ซึ่งภายในบทนี้จะกล่าวถึงหลักการพื้นฐานของการสื่อสาร การเข้ารหัสเพื่อแก้ไขข้อมูลผิดพลาดต่างๆ ที่ใช้กับระบบสื่อสารในปัจจุบัน

2.1 การเข้ารหัสช่องสัญญาณ [1]

แบ่งได้เป็น 2 ประเภทคือ

2.1.1 Block codes

Block codes จะแบ่งข้อมูลออกเป็นกลุ่มหรือบล็อกขนาด k บิต และเข้ารหัสข้อมูลแต่ละบล็อกเป็นข้อมูล codeword ขนาด n บิต โดยที่ $n > k$ ใช้สัญลักษณ์ (n,k) โดย codeword จะประกอบด้วยข้อมูลเดิมขนาด k บิต และบิตพิเศษ (check bit) ขนาด $n-k$ ซึ่งเป็นบิตที่ใช้ตรวจสอบว่ามีความผิดพลาดหรือไม่ โดยที่ภาครับจะทำการถอดรหัส codeword ได้เป็นข้อมูลเดิม และค่าซินโดรม (syndrome) จะเป็นค่าที่บอกว่ามีความผิดพลาดเกิดขึ้นหรือไม่

2.1.2 Sequential codes

Sequential codes เป็นการเข้ารหัสข้อมูลโดยที่ไม่จำเป็นต้องแบ่งข้อมูลออกเป็นบล็อก โดยสามารถเข้ารหัสข้อมูลได้อย่างต่อเนื่อง โดยใช้สัญลักษณ์ $1/n$ ซึ่งเป็นขนาดข้อมูลหารด้วยขนาด codeword โดย codeword ที่ได้จะเป็นข้อมูลบิตใหม่ ซึ่งไม่ได้ประกอบด้วยข้อมูลบิตเดิมและบิตพิเศษ

2.2 รหัสแบบบอส-โชดูรี-ฮอคเคนเกม (Bose-Chudhuri-Hocquenghem Codes: BCH) [1, 2]

รหัส BCH สำหรับจำนวนเต็ม m และ t ใดๆ $(t < 2^{m-1})$ โดยมีพารามิเตอร์ดังต่อไปนี้

$$\begin{aligned} &\text{Block length:} && n = 2^m - 1 \\ &\text{Message Size (bits):} && k \geq n - m \\ &\text{Minimum Distance:} && d_{min} = 2t + 1 \end{aligned} \tag{1}$$

จะเห็นได้ว่า รหัสนี้สามารถแก้ไขบิตที่ผิดเท่ากับหรือน้อยกว่า t บิต ในบล็อกที่มีความยาวของรหัส $n = 2^m - 1$ บิต บางครั้งรหัสนี้ถูกเรียกว่า t – error correcting ของรหัส BCH

2.3 การเข้ารหัสแบบ BCH [1, 5]

ที่มาของเจนเนอเรเตอร์โพลิโนเมียลของรหัส BCH คือ ให้ α เป็นไพรมีทีฟอีลีเมนท์ของ $GF(2^m)$ พิจารณาการยกกำลังของ α คือ $\alpha, \alpha^2, \alpha^3, ..., \alpha^{2t}$ ให้ $m_1(X)$ เป็นโพลิโนเมียลต่ำสุด ดังนั้นเจนเนอเรเตอร์โพลิโนเมียลของ t-error correcting ของรหัส BCH ซึ่งได้จาก Least common multiple (LCM) ของ $m_1(X), m_2(X), \ldots\ldots, m_{2t}(X)$ รหัสถูกสร้างจาก

$$g(X) = LCM(m_1(X), m_2(X), \ldots\ldots, m_{2t}(X)) \tag{2}$$

โดยที่ $\alpha, \alpha^2, ..., \alpha^{2t}$ เป็นรากของ $g(X)$ นั้นคือ $g(\alpha^i) = 0$ สำหรับ $i = 1,2,\ldots,2t$ ดังนั้นกำลังคู่และกำลังคี่ของ α จะมีโพลิโนเมียลต่ำสุดเหมือนกัน ทำให้ผลลัพธ์เจนเนอเรเตอร์โพลิโนเมียลของรหัสลดลง เป็นสมการ

$$g(X) = LCM(m_1(X), m_3(X), \ldots\ldots, m_{2t-1}(X)) \tag{3}$$

บางครั้งเรียกสมการข้างต้นว่า narrow sense BCH Code เนื่องจากลำดับของแต่ละโพลิโนเมียลต่ำสุดจะเท่ากับหรือน้อยกว่า m อันดับของ $g(x)$ มากที่สุดคือ mt หรือบิทของพาริตีที่เช็ค $n-k$ จะไม่เกิน mt การเข้ารหัสข้อมูล $u(X)$ โดยการใช้รหัส BCH กับเจอเนอเรเตอร์โพลิโนเมียล $g(X)$ ข้างต้น สามารถหาสัญลักษณ์พาริตี้ตามสมการ

$$b(X) = X^{n-k}u(X) \bmod g(X) \tag{4}$$

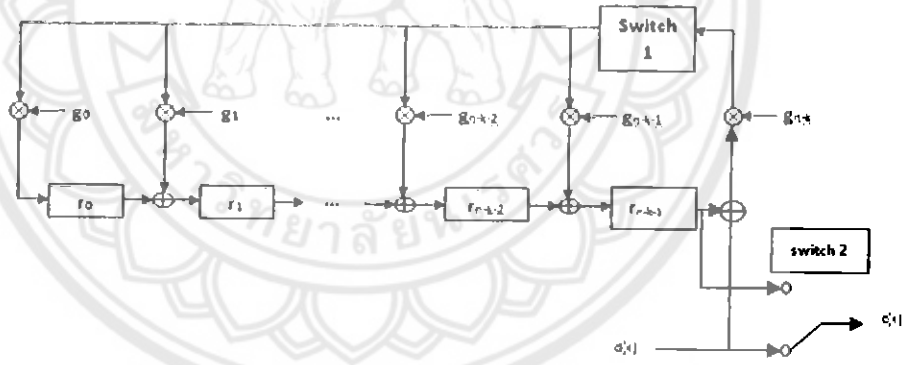
ดังนั้น รหัสคำข้อมูลที่ผ่านการเข้ารหัส BCH ได้เป็น

$$v(X) = X^{n-k}u(X) + b(X) \tag{5}$$

ในกรณี systematic codes การทำงานต่างๆของวงจร Generator polynomial, $g(X)$ เขียนได้เป็น

$$g(X) = g_0 + g_1x + g_2x^2 + \ldots + g_{n-k-1}x^{n-k-1} + g_{n-k}x^{n-k} \tag{6}$$

และแสดงในรูปที่ 2.1



รูปที่ 2.1 รูปแบบรหัสในรูป Systematic ของรหัส BCH มีขั้นตอนของ $(n-k)$ shift-register [1]

โดยในกรณีของวงจรเข้ารหัสตัวอย่างในรูปที่ 2.1 จะสามารถแสดงการทำงานต่างๆของวงจรได้ด้วย Generator polynomial ต่อไปนี้

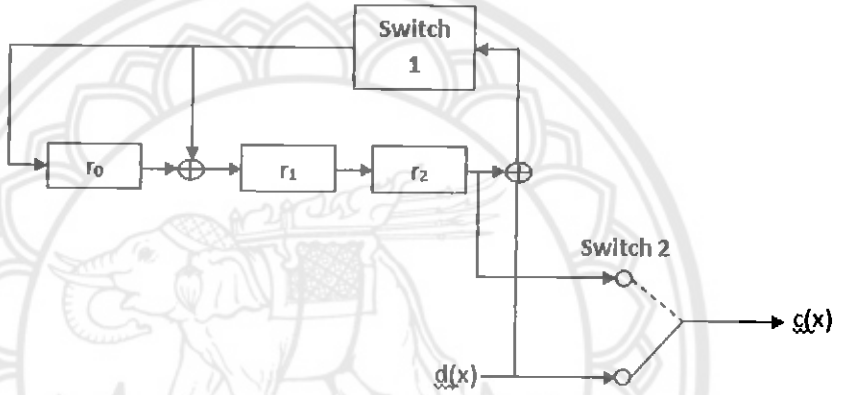
<u>ตัวอย่าง</u> การเข้ารหัสแบบ BCH(7,4) ซึ่งประกอบไปด้วยค่า Generator polynomial = 13_{octal} และ *n-k* = 3, 4-*bit information data* คือ *d* = 1 0 1 1

$$g = 13_{octal}$$

$$= 1011_{bin}$$

$$g(X) = X^3 + X + 1$$

นำ Generator polynomial มาแสดงเป็นโครงสร้างการเข้ารหัสได้ดังรูปที่ 2.2



รูปที่ 2.2 รูปแบบของรหัสในรูป Systematic ของรหัส BCH(7, 4) [1]

จากรูปที่ 2.2 มีขั้นตอนดังนี้

ตารางที่ 2.1 Shift Register รหัสในรูป Systematic ของรหัส BCH(7, 4) [1]

Input queue	Shift index	Shift register $r_0r_1r_2$	Codeword $c_0c_1c_2c_3c_4c_5c_6$
1 0 1 1	0	0 0 0	- - - - - - -
1 0 1	1	1 1 0	1
1 0	2	1 0 1	1 1
1	3	1 0 0	0 1 1
	4	1 0 0	1 0 1 1
	5	0 1 0	0 1 0 1 1
	6	0 0 1	0 0 1 0 1 1
	7	0 0 0	1 0 0 1 0 1 1

การทำงานในรูปที่ 2.2 เมื่อ state เริ่มต้นเป็น 0 0 0 บิตแรกที่ถูกส่งเข้าในวงจรคือบิต "1" บิต "1" จะผ่าน switch 1 และเคลื่อนมายัง r_0 ซึ่ง r_0 จะถูกแทนด้วยบิต "1" บิต "0" เดิมที่อยู่ใน r_0 จะ *exclusive* or กับบิต "1" เท่ากับ "1" และเลื่อนบิต "1"มายัง r_1 และบิต "0" จาก r_1 เดิม จะถูกเลื่อนไปที่ r_2 ค่าที่ได้จากกการอินพุตบิต "1" เข้าไป ทำให้ state ข้างต้นเปลี่ยนเป็น 1 1 0 และค่าจาก r_2 จะ *exclusive* or กับบิต "1" เท่ากับ "1" ซึ่ง Codeword แรกที่ได้คือ "1" จากข้อมูลที่ส่งเข้ามาจะถูก Shift Register ไปเรื่อยๆ จนเมื่อไม่มีข้อมูลที่จะส่งแล้ว จะนำบิต "0" เข้ามาใน Shift Register แทน จนกว่าจะได้ค่า Output ออกมาเป็นบิต "0" ทั้งหมด จึงหยุดการเข้ารหัส จากตัวอย่างได้ codeword เป็น $c = 1\,0\,0\,1\,0\,1\,1$ ทำให้อยู่ในรูปของ polynomial ได้เป็น $c(X) = 1 + x^3 + x^5 + x^6$

เรายังสามารถคำนวณการเข้ารหัสแบบ BCH โดยใช้ทฤษฎีจากการเข้ารหัสแบบ RS codes ดังนี้ i(x) คือ information polynomial, g(x) คือ generator polynomial แสดงในสมการดังต่อไปนี้

$$c(x) = i(x) \cdot g(x) \tag{7}$$

$$i(x) = i_1x + i_2x^2 + .. + i_kx^k = \sum_{j=1}^{k} i_j x^j \tag{8}$$

$$c(x) = c_1x + c_2x^2 + .. + c_nx^n = \sum_{j=1}^{n} c_j x^j \tag{9}$$

ผลกระทบการเข้ารหัส จะต้องมีการบวกเพิ่มของ error คือ c(x)

$$r(x) = c(x) + e(x) \quad = c_1x + c_2x^2 + .. + c_nx^n + e_1x + e_2x^2 + .. + e_nx^n \tag{10}$$

เมื่อ r(x) คือ โพลินอเมียลที่ได้รับเกิดความเสียหาย สามารถใช้สมการ(11) ในการถอดรหัสอย่างง่าย

$$i(x) = \frac{c(x)}{g(x)} \tag{11}$$

ในระบบ systematic RS codes สัญลักษณ์แรกที่เข้ารหัส คือ k ซึ่งจะนำคูณกับ information polynomial, i(x) ด้วย x^{n-k} เพื่อที่จะเปลี่ยนเป็นอันดับสูงสุดของ codeword โดยคำตอบที่ได้ใช้สัญลักษณ์ p(x) และเป็นไปตามกฎการเข้ารหัส เพื่อที่จะส่งผลในการ codewordที่ถูกต้อง ถูกต้องในแง่นี้หมายความว่า

$$c(x) = x^{(n-k)} \cdot i(x) + p(x) \tag{12}$$

และเป็นไปตามข้อกำหนดข้างต้นจาก

$$Rem\left\{\frac{c(x)}{g(x)}\right\} = 0 \tag{13}$$

$$Rem\left\{\frac{x^{(n-k)}\cdot i(x)+p(x)}{g(x)}\right\}=0 \tag{14}$$

$$Rem\left\{\frac{x^{(n-k)}\cdot i(x)}{g(x)}\right\}+Rem\left\{\frac{p(x)}{g(x)}\right\}=0 \tag{15}$$

เนื่องจากลำดับของ parity พหุนาม p(x) มีค่าน้อยกว่า (n-k) และ ลำดับของ g(x) คือ (n-k) จะได้

$$Rem\left\{\frac{p(x)}{g(x)}\right\}=p(x) \tag{16}$$

โดยการแทนสมการ(16) ในสมการ(15) จะได้

$$-Rem\left\{\frac{x^{(n-k)}\cdot i(x)}{g(x)}\right\}=p(x) \tag{17}$$

ดังนั้นเมื่อแทนสมการ(17)ลงในสมการ(12) จะได้สมการการเข้ารหัสดังนี้

$$c(x)=x^{(n-k)}\cdot i(x)+Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\} \tag{18}$$

ตัวอย่าง การเข้ารหัสแบบ BCH (7,4) ซึ่งประกอบไปด้วยค่า Generator polynomial = 13_{octal} และ (data bits): d = 1 1 0 1

$$g = 13_{octal}$$

$$= 1011_{bin}$$

$$g(x) = x^3 + x + 1$$

data bits = 1 1 0 1 แปลงให้อยู่ในรูปของ $d(x) = i(x) = x^3+x^2+1$

จากสมการ(18): $c(x) = x^{(n-k)} \cdot i(x) + Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$

หาค่าในส่วนของ $Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$ แทนค่า

$$= Rem\left\{\frac{x^6+x^5+x^3}{x^3+x+1}\right\}$$

$$\begin{array}{r} x^3 + x^2 + x + 1 \\ x^3 + x + 1 \;\overline{)\; x^6 + x^5 + 0 + x^3} \\ \underline{x^6 + 0 + x^4 + x^3} \\ x^5 + x^4 \\ \underline{x^5 + 0 + x^3 + x^2} \\ x^4 + x^3 + x^2 \\ \underline{x^4 + 0 + x^2 + x} \\ x^3 + 0 + x \\ \underline{x^3 + 0 + x + 1} \\ 1 \end{array}$$

จากสมการ(7)แทนค่า: $c(x) = x^3(x^3+x^2+1)+1$

$$= x^6 + x^5 + x^3 + 1$$

แปลงให้อยู่ในรูปของเลขฐานสอง ฉะนั้นคำตอบของ codeword คือ 1 1 0 1 0 0 1

<u>ตัวอย่าง</u> การเข้ารหัสแบบ BCH (7,4) ซึ่งประกอบไปด้วยค่า Generator polynomial = 13_{octal} และ (data bits): d = 1 0 0 1

$$g = 13_{octal}$$

$$= 1011_{bin}$$

$$g(x) = x^3 + x + 1$$

data bits = 1 0 0 1 แปลงให้อยู่ในรูปของ $d(x) = i(x) = x^3 + 1$

จากสมการ(7): $c(x) = x^{(n-k)} \cdot i(x) + Rem\left\{\frac{x(n-k) \cdot i(x)}{g(x)}\right\}$

หาค่าในส่วนของ $Rem\left\{\frac{x(n-k) \cdot i(x)}{g(x)}\right\}$ แทนค่า

$$= Rem\left\{\frac{x^6+x^3}{x^3+x+1}\right\}$$

$$= \begin{array}{r} x^3 + x \\ x^3 + x + 1 \overline{)\, x^6 + 0 + 0 + x^3} \\ \underline{x^6 + 0 + x^4 + x^3} \\ x^4 \\ \underline{x^4 + 0 + x^2 + x} \\ x^2 + x \end{array}$$

จากสมการ(7)แทนค่า: $c(x) = x^6 + x^3 + x^2 + x$

แปลงให้อยู่ในรูปของเลขฐานสอง ฉะนั้นคำตอบของ codeword คือ 1 0 0 1 <u>1 1 0</u>

การเข้ารหัสในรูปของ BCH (7,4) ส่งข้อมูลจำนวน 4 บิตในหนึ่งบล็อก **เมื่อนำเข้ารหัสจะได้รับบิตเพิ่มขึ้นอีก 3 บิต เป็นบิตพาริตี้ ฉะนั้นบิตที่ได้รับจากการเข้ารหัสมีทั้งหมด 7 บิต ซึ่งยังมี** การเข้ารหัส BCH (15,11) ดังตัวอย่างที่จะแสดงต่อไปนี้

<u>ตัวอย่าง</u> การเข้ารหัสแบบ BCH (15,11) ซึ่งประกอบไปด้วยค่า Generator polynomial = 23_{octal} และ (data bits): d = 1 0 0 1 0 0 1 0 0 1 1

$$g = 23_{octal}$$

$$= 10011_{bin}$$

$$g(x) = x^4 + x + 1$$

data bits = 1 0 0 1 0 0 1 0 0 1 1 แปลงให้อยู่ในรูปของ $d(x) = i(x) = x^{10} + x^7 + x^4 + x + 1$

จากสมการ(7): $c(x) = x^{(n-k)} \cdot i(x) + Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$

หาค่าในส่วนของ $Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$ แทนค่า

$$= Rem\left\{\frac{x^4(x^{10}+x^7+x^4+x+1)}{x^4+x+1}\right\}$$

$$= \frac{x^{14}+x^{11}+x^8+x^5+x^4}{x^4+x+1}$$

$$= x^4 + x + 1 \overline{)x^{14} + 0 + 0 + x^{11} + 0 + 0 + x^8 + 0 + 0 + x^5 + x^4}$$

$$x^{10} + x^6 + x^4 + x^3 + x^2 + 1$$

$$\underline{x^{14} + 0 + 0 + x^{11} + x^{10}}$$

$$x^{10} + 0 + x^8 + 0 + 0 + x^5 + x^4$$

$$\underline{x^{10} + 0 + 0 + x^7 + x^6}$$

$$x^8 + x^7 + x^6 + x^5 + x^4$$

$$\underline{x^8 + 0 + 0 + x^5 + x^4}$$

$$x^7 + x^6$$

$$\underline{x^7 + 0 + 0 + x^4 + x^3}$$

$$x^6 + 0 + x^4 + x^3$$

$$\underline{x^6 + 0 + 0 + x^3 + x^2}$$

$$x^4 + 0 + x^2$$

$$\underline{x^4 + 0 + 0 + x + 1}$$

$$\underline{x^2 + x + 1}$$

จากสมการ(7) แทนค่า: $c(x) = x^{14} + x^{11} + x^8 + x^5 + x^4 + x^2 + x + 1$

แปลงให้อยู่ในรูปของเลขฐานสอง ฉะนั้นคำตอบของ codeword คือ 1 0 0 1 0 0 1 0 0 1 1 <u>0 1 1 1</u>

<u>ตัวอย่าง</u> การเข้ารหัสแบบ BCH (15,11) ซึ่งประกอบไปด้วยค่า Generator polynomial = 23_{octal} และ (data bits): d = 1 0 0 0 1 0 0 0 0 1 0

$$g = 23_{octal}$$

$$= 10011_{bin}$$

$$g(x) = x^4 + x + 1$$

data bits = 1 0 0 0 1 0 0 0 0 1 0 แปลงให้อยู่ในรูปของ $d(x) = i(x) = x^{10} + x^6 + x$

จากสมการ(7): $c(x) = x^{(n-k)} \cdot i(x) + Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$

หาค่าในส่วนของ $Rem\left\{\frac{x(n-k)\cdot i(x)}{g(x)}\right\}$ แทนค่า

$$= Rem\left\{\frac{x^4(x^{10}+x^6+x)}{x^4+x+1}\right\}$$

$$= \frac{x^{14}+x^{10}+x^5}{x^4+x+1}$$

$$= x^4 + x + 1 \overline{)\, x^{14} + 0 + 0 + 0 + x^{10} + 0 + 0 + 0 + 0 + x^5}$$

$x^{10} + x^7 + x^4 + x^3$

$x^{14} + 0 + 0 + x^{11} + x^{10}$

$x^{11} + 0 + 0 + 0 + 0 + 0 + x^5$

$x^{11} + 0 + 0 + x^8 + x^7$

$x^8 + x^7 + 0 + x^5$

$x^8 + 0 + 0 + x^5 + x^4$

$x^7 + 0 + 0 + x^4$

$x^7 + 0 + 0 + x^4 + x^3$

x^3

จากสมการ(7) แทนค่า: $c(x) = x^{14} + x^{10} + x^5 + x^3$

แปลงให้อยู่ในรูปของเลขฐานสอง ฉะนั้นคำตอบของ codeword คือ 1 0 0 0 1 0 0 0 0 1 0 <u>1 0 0 0</u>

การเข้ารหัสในรูปของ BCH (15,11) ส่งข้อมูลจำนวน 11 บิตในหนึ่งบล็อก เมื่อนำเข้ารหัสจะได้รับบิตเพิ่มขึ้นอีก 4 บิต เป็นบิตพาริตี้ ฉะนั้นบิตที่ได้รับจากการเข้ารหัสมีทั้งหมด 15 บิต

2.4 การถอดรหัสแบบ Viterbi Decoder [1, 5]

สำหรับการถอดรหัส แบบ Viterbi Decoding จะมีรูปแบบสำหรับการถอดรหัสที่ใช้งานอยู่ 2 ลักษณะด้วยกัน ได้แก่

2.4.1 การตัดสินใจแบบ Hard-Decision สำหรับการทำงานของวงจรถอดรหัสที่ใช้กระบวนการตัดสินใจแบบ Hard Decision นั้น จะเป็นการพิจารณาข้อมูลที่รับเข้ามา โดยการพิจารณาว่าข้อมูลที่รับเข้ามาในแต่บิตนั้น มีค่าของข้อมูลเป็น "0" หรือ "1" เท่านั้น โดยพิจารณาค่าความน่าจะเป็นที่จะเกิดความผิดพลาดในการส่งข้อมูลระบบสื่อสารซึ่งทำให้มีการรับข้อมูลที่ผิดพลาดจากบิต "0" ไป "1" และบิต "1" ไป "0"

2.4.2 การตัดสินใจแบบ Soft-Decision การถอดรหัสที่มีการใช้กระบวนการตัดสินใจแบบ Soft Decision จะเป็นการพิจารณาถึงข้อมูลที่รับเข้ามาได้โดยการทำการตัดสินใจระดับของข้อมูลที่รับเข้ามา โดยการแบ่งระดับของสัญญาณที่ใช้ในการคำนวณหาค่า metric ที่มากกว่า 2 ระดับ ซึ่งผลลัพธ์ที่ได้นั้น จะได้ข้อมูลรายละเอียดของข้อมูลที่ส่งมาที่มากกว่า กรณีของ Hard-Decision ซึ่งข้อมูลที่ได้จากการตัดสินใจ (soft-output) นั้น จะถูกนำมาใช้ในการคำนวณหาค่า metrics เพื่อเปรียบเทียบข้อมูลที่รับเข้ามา ณ เวลานั้นๆกับข้อมูลที่อยู่ในเส้นทางต่างๆ ณ เวลานั้น ซึ่งจะมีรูปแบบที่ใช้ในการคำนวณที่แตกต่างกันไป

Viterbi Algorithm [1]

โดยในการทำงานต่างๆนั้น จะต้องมีการคำนวณหาความแตกต่างระหว่างข้อมูลที่รับเข้ามาและค่าที่อยู่ในเส้นทางต่างๆ เพื่อใช้ในกระบวนการตัดสินใจโดยกระบวนการที่ใช้ในการทำการหาเส้นทางที่ดีที่สุดนั้น จะใช้วิธีการทำงานที่มีชื่อว่า Viterbi Algorithm ซึ่งจะเป็นกระบวนการที่ใช้ในการค้นหาเส้นทางที่อยู่ใน Trellis Diagram ที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับได้มากที่สุด เพื่อที่จะนำข้อมูลในเส้นทางนั้นมาคำนวณหาค่าของข้อมูลที่ถูกส่งมา โดยที่ในกระบวนการค้นหาเส้นทางที่เหมาะสมที่สุดโดยใช้ Viterbi Algorithm นั้น จะมีขั้นตอนในการทำงานดังต่อไปนี้

1. พิจารณาแบ่งข้อมูลที่รับเข้ามาออกเป็นข้อมูลย่อยๆ จำนวน m ช่วง ซึ่งแต่ละช่วงนั้นมีขนาดของข้อมูลเท่ากับ n บิต

2. ทำการวาด Trellis diagram ที่มีจำนวน state ในการทำงานเท่ากับ m state โดยจะมีการพิจารณาเฉพาะเส้นทางที่มีความเป็นไปได้ว่าจะถูกส่งมาเท่านั้น โดยสำหรับที่ state ของTrellis diagram ตั้งแต่ L-1 ขึ้นไปนั้น(L คือค่า Constraint Length) ให้วาดเฉพาะเส้นทางที่จะพุ่งเข้าหาสภาวะของวงจรที่มีข้อมูลเป็น 0 ทั้งหมด

3. กำหนดค่าตัวแปร l = 1 และทำการกำหนดค่าเริ่มต้นของตัวแปร metric ในสภาวะเริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมด ให้มีค่าของ metric เท่ากับ 0

4. ทำการคำนวณหาค่าความแตกต่างของข้อมูล (distance) ระหว่างข้อมูลที่รับได้ชุดที่ 1 กับข้อมูลในเส้นทางในการเปลี่ยนแปลงสถานะใน Trellis diagram จาก state ที่ l ไปเป็น l+1

5. นำค่าที่คำนวณได้ไปบวกกับค่า metric สะสมของ state l เพื่อคำนวณหาค่า ของ metric สะสมใน State ที่ l+1 เพื่อใช้ในการตัดสินใจเลือกเส้นทางที่เหมาะสมที่สุด ในการเปลี่ยนแปลงข้อมูลไปยัง state นั้นๆ โดยในแต่ละ stateนั้น จะมีจำนวนเส้นทางทั้งหมดจำนวน 2^k เส้นทางที่จะพุ่งเข้า state เดียวกัน

6. พิจารณา ณ ตำแหน่งใน state ที่ l+1 ในแต่ละstateนั้น ทำการเลือกเส้นทางที่มีค่า metric สะสมที่มีค่าน้อยที่สุดที่พุ่งเข้าหาในแต่ละ state โดยทีเส้นทางที่ถูกเลือกนั้นจะถูกเรียกว่า "Survivor" ซึ่งจะเป็นเส้นทางที่ถูกเก็บไว้ทำการคำนวณใน state ต่อไป และสำหรับเส้นทางอื่นๆที่ไม่ได้ถูกเลือกนั้นจะถูกเรียกว่า "Forgetting" โดยจะถูกลบทิ้งออกไปจากระบบการตัดสินใจ

7. ถ้าหากว่า l นั้นมีค่าเท่ากับ m แล้วให้ทำงานในขั้นตอนต่อไปได้ แต่ถ้ายังมีค่าน้อยกว่าจะมีต้องมีการเพิ่มค่า l ขึ้นอีก 1 จากนั้นจึงกลับไปทำงานที่ขั้นตอนที่ 4 ใหม่

8. เริ่มต้นพิจารณา ณ state ที่ m+1 ที่มีสภาวะของข้อมูลสถานะเป็น 0 ทั้งหมด ทำการเลือกเส้นทางที่เป็น "Survival" ซึ่งเป็นเส้นทางที่ถูกเลือกที่เหลืออยู่ย้อนกลับไปจนกระทั่งถึงสภาวะเริ่มต้นของการทำงานที่มีสภาวะในการทำงานเป็น 0 ทั้งหมด ซึ่งเส้นทางที่ได้นั้นจะเป็นเส้นทางที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับเข้ามามากที่สุด ซึ่งจะถูกนำไปใช้ในการคำนวณหาข้อมูลข่าวสารที่ถูกส่งมา โดยข้อมูลข่าวสารที่จะถูกส่งออกไปจากภาคถอดรหัสนั้น จะเป็นการส่งข้อมูลทั้งหมดที่อยู่ในเส้นทางส่งออกไป ยกเว้นข้อมูล 0 จำนวน k_0 (L-1) บิต ที่อยู่ท้ายสุดนั้นจะถูกตัดทิ้งไป

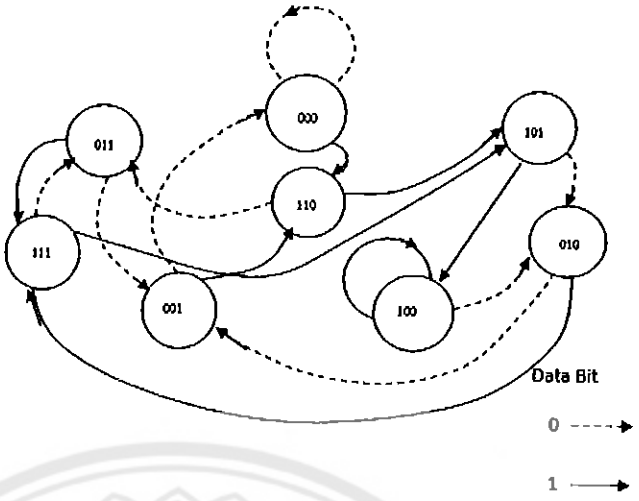
การคำนวณเส้นทางของการถอดรหัสแบบ Hard-decision โดยอาศัยแผนภาพเทรลลิส สำหรับการถอดรหัสจะเป็นวิธีที่จะนำเสนอและแสดงตัวอย่าง เพื่อช่วยประกอบในการอธิบายกรรมวิธีการถอดรหัสในแต่ละขั้นตอนโดยละเอียด

การเข้ารหัสแบบ BCH (7,4) มีรูปแบบของรหัสในรูป Systematic ของรหัส BCH(7,4) ดังแสดงในรูปที่ 2.2 ในรูปของการเข้ารหัสนี้สามารถนำมาแสดงเป็นตารางการเปลี่ยนสถานะเข้ารหัส BCH(7,4) ได้ดังนี้

ตารางที่ 2.2 ตารางแสดงรายละเอียดการเปลี่ยนสถานะเข้ารหัส BCH(7,4) [1]

State	r_0	r_1	r_2	Input bit	Codeword
0	0	0	0	0	0 0 0
				1	1 1 0
1	0	0	1	0	0 0 0
				1	1 1 0
2	0	1	0	0	0 0 1
				1	1 1 1
3	0	1	1	0	0 0 1
				1	1 1 1
4	1	0	0	0	0 1 0
				1	1 0 0
5	1	0	1	0	0 1 0
				1	1 0 0
6	1	1	0	0	0 1 1
				1	1 0 1
7	1	1	1	0	0 1 1
				1	1 0 1

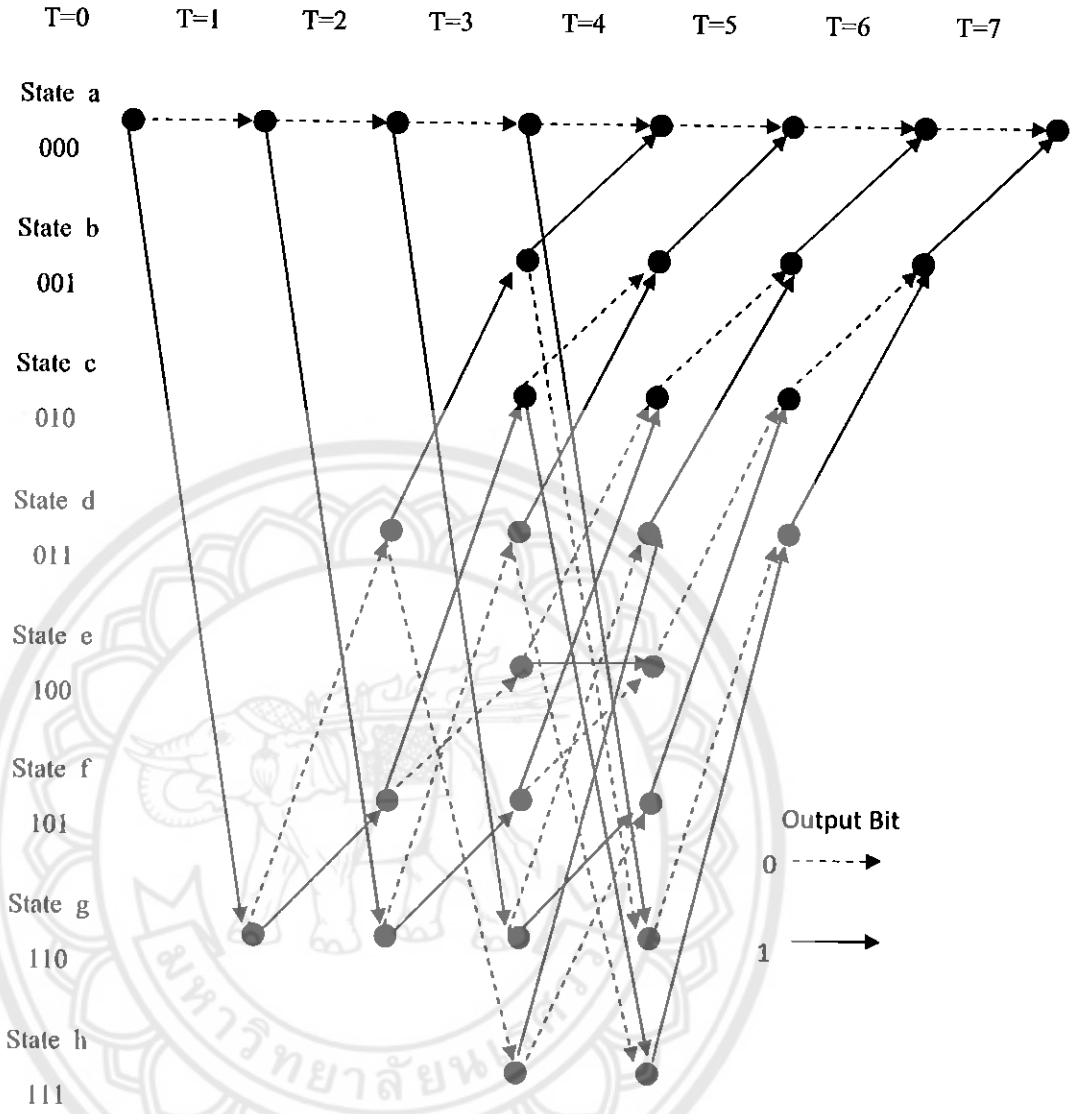
จากตารางที่ 2.2 สามารถนำมาเขียนเป็นแผนภาพสถานะ(State Diagram) ของวงจรเข้ารหัส BCH(7,4) ดังนี้ จากตารางเริ่มต้นที่ state 000 เมื่ออินพุตบิต "0" state ยังคงที่อยู่ที่ 000 เมื่ออินพุตบิต "1" state เปลี่ยนเป็น 110 ซึ่งนำมาแสดงในรูปที่ 2.3 คือเมื่ออินพุตบิตที่เป็นบิต "0" state ยังคงวนอยู่ที่ 000 และเป็นเส้นปะ แต่เมื่ออินพุตบิต "1" state เปลี่ยนเป็น 110 และเป็นเส้นทึบ ทำไปเรื่อยๆจนครบทุก state แผนภาพที่ได้จะวนเป็น cycle ดังรูปที่ 2.3



รูปที่ 2.3 แผนภาพสถานะ (State Diagram) ของวงจรเข้ารหัส BCH(7,4) [1]

นำรายละเอียดในตารางที่ 2.2 ตารางแสดงรายละเอียดการเปลี่ยนสถานะเข้ารหัส BCH(7,4) และแผนภาพสถานะ (State Diagram) ของวงจรเข้ารหัส BCH(7,4) มาหาเส้นทางในการถอดรหัสแบบ Hard-Decision ได้ดังนี้

จากรูปที่ 2.4 ที่ T=0 state เริ่มต้นคือ state 000 เมื่อเราอินพุตบิต "0" จะมีเส้นทางที่เป็นเส้นประ ที่ขยับไปยัง T=1 ที่ state 000 เมื่ออินพุตบิต "1" จะมีเส้นทางเดินเป็นเส้นทึบ และขยับไปยัง T= 1 ที่ state 110 ต่อมาที่ T=1 ประกอบไปด้วย 2 state ด้วยกันคือ state 000 และ state 110, ที่ state 000 แบ่งออกเป็น 2 ทางคือไปยัง state 000 และstate 110, ในทางเดียวกันstate 110 แบ่งออกเป็น 2 ทางคือ state 011 และ state 101 ซึ่งทั้ง 4 state ใหม่นี้เคลื่อนที่ไปที่ละ T=1 ทำไปเรื่อยๆจน state สุดท้ายกลับมาที่ state 000



รูปที่ 2.4 การถอดรหัสแบบ Hard-decision [1]

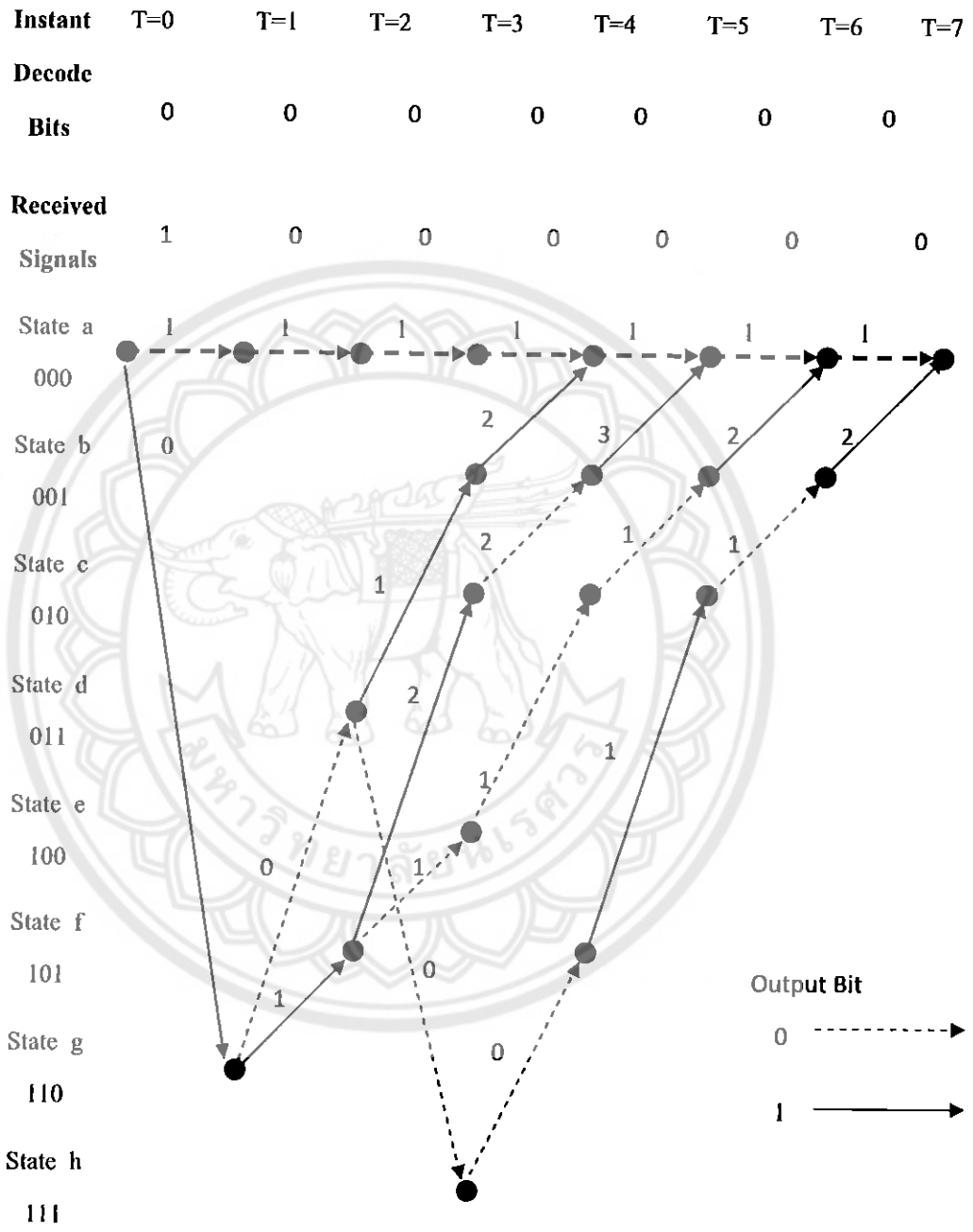
จากรูปที่ 2.4 สามารถนำมาวิเคราะห์หาเส้นทางเดินได้ดังนี้ จากรูปที่ 2.4 เรากำหนดให้เส้นประคือ บิต "0" และเส้นทึบคือ บิต "1" ในเส้นทางที่ 1 เราเลือกให้เป็นบิต 0 0 0 0 0 0 0 คือเป็นเส้นตรงที่ state 0 เส้นทางที่ 2 คือ 1 0 1 1 0 0 0 คือที่ state 000 มาที่ state 110 มาที่ state 011 มาที่ state 001 มาที่ state 000 เลือกให้ครบทุกเส้นทาง เราจะได้เส้นทางทั้งหมด 16 เส้นทาง ดังแสดงในตารางที่ 2.3

ตารางที่ 2.3 ตารางแสดงเส้นทางในการถอดรหัสแบบ Hard-decision

เส้นทาง	Codeword						
1	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0
3	1	0	0	1	1	1	0
4	1	0	0	0	1	0	1
5	1	0	1	0	0	1	1
6	1	1	1	0	1	0	0
7	1	1	0	0	0	1	0
8	1	1	0	1	0	0	1
9	0	1	0	1	1	0	0
10	0	1	1	1	0	1	0
11	0	1	1	0	0	0	1
12	0	0	1	0	1	1	0
13	0	0	1	1	1	0	1
14	0	0	0	1	0	1	1
15	0	1	0	0	1	1	1
16	1	1	1	1	1	1	1

จากเส้นทางเดินดังกล่าว สามารถนำมาถอดรหัสเพื่อเปรียบเทียบเส้นทางเดินที่มีบิตผิดพลาดน้อยที่สุด เพื่อที่ปลายทางจะสามารถรับข้อมูลที่ถูกต้องมากที่สุด

<u>ตัวอย่าง</u> การถอดรหัสแบบ Hard-decision [1] บิตที่รับเข้ามาจากการเข้ารหัสคือ 1 0 0 0 0 0 0 แสดงการถอดรหัสได้ดังรูปที่ 2.5



รูปที่ 2.5 การถอดรหัสแบบ Hard-decision ซึ่งเกี่ยวข้องกับรหัส BCH (7,4) [1]

จากรูปที่ 2.5 บิตที่ได้รับมาคือ 1 0 0 0 0 0 0 ผ่านการเข้ารหัสแบบ BCH Codes ซึ่งการเข้ารหัส และส่งสัญญาณมายังปลายทางอาจประกอบไปด้วยค่าของ noise เมื่อนำมาถอดรหัสแบบ Hard-decision จะมีขั้นตอนการคิดดังนี้ ในการคำนวณเพื่อถอดรหัส จะประกอบไปด้วย state 7 states เริ่มตั้งแต่ 000, 001, 010, 011, 100, 101, 110, และ 111 และประกอบไปด้วย Instant T=0 ถึง T=7 เราจะเริ่มจากหาเส้นทางที่เป็นไปได้ เส้นปะ คือทางเดินของ "0" เส้นทึบคือทางเดินของ "1" เราจะนำมาเทียบกับบิตที่ได้รับมาแล้วนำมาเลือกเส้นทาง ที่ T=0 เริ่มต้นที่state 000 บิตที่รับมาคือ "1" ทางเลือกที่ 1 หากเดินไปเส้นปะจะทำให้มีบิตที่ผิดสะสมคือ 1 บิต ทางเลือกที่2 หากเดินไปยังเส้นทึบ จะทำให้มีบิตที่ผิดสะสมคือ 0 บิต ที่ T=1 บิตที่รับมาคือ "0" หากเลือกเดินในเส้นปะ จะมีบิตที่ผิดสะสมเป็น 1 เท่าเดิม เนื่องจากในตำแหน่งที่เป็นเส้นปะคือ "0" บิตที่รับก็คือ "0" เพราะฉะนั้นบิตที่ผิดคือ "0" ทำให้บิตที่ผิดสะสมยังคงเป็น 1 ซึ่งในทางกลับกันหากเลือกเส้นทางที่เป็นเส้นทึบ จะมีบิตสะสมจาก 0 บิตเป็น 1 บิต ทำไปเรื่อยจนค่าสุดท้ายเข้าสู่ state 000 เมื่อปลายทางทราบรูปแบบทางเดินและบิตที่ผิดสะสม ปลายทางจะสามารถเลือกทางเดินที่มีโอกาสผิดที่น้อยสุด ซึ่งจากรูปที่ 2.5 ปลายทางสามารถ decode bits ได้เป็น 0 0 0 0 0 0 0

อีกตัวอย่างเป็นวิธีการถอดรหัสโดยการนำเส้นทางในตารางที่ 2.3 มาคำนวณ

<u>ตัวอย่าง</u> การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น $m_1 = [1\ 1\ 0\ 1]$ Codeword ที่ผ่านการเข้ารหัสคือ $c_1 = [1\ 1\ 0\ 1\ 0\ 0\ 1]$ กำหนดให้บิตผิดพลาด 0 บิต

นำ c_1 มา *exclusive* or กับเส้นทางในตารางที่ 2.3 แล้วรวมบิต 1 ในแต่ละแถว และเลือกเส้นทางที่มีบิตที่ผิดพลาดน้อยที่สุด เมื่อได้เส้นทางจากบิตที่ผ่านการคำนวณแล้ว จะนำเส้นทางนั้นไปเลือกเส้นทางจากตารางที่ 2.3 ค่าจากเส้นทางนั้นคือ Codeword

ตารางที่ 2.4 ตารางแสดงการถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น $m_1 = [1\ 1\ 0\ 1]$ บิตผิดพลาด 0 บิต

เส้นทาง	C_1 *exclusive* or กับเส้นทางในตารางที่ 2.3							Sum bit (1)
1	1	1	0	1	0	0	1	4
2	0	1	1	0	0	0	1	3
3	0	1	0	0	1	1	1	4
4	0	1	0	1	1	0	0	3
5	0	1	1	1	0	1	0	4
6	0	0	1	1	1	0	1	4
7	0	0	0	1	0	1	1	3
8	0	0	0	0	0	0	0	0
9	1	0	0	0	1	0	1	3
10	1	0	1	0	0	1	1	4
11	1	0	1	1	0	0	0	3
12	1	1	1	1	1	1	1	7
13	0	0	1	1	1	0	1	4
14	1	1	0	0	0	1	0	3
15	1	0	0	1	1	1	0	4
16	0	0	1	0	1	1	0	3

จากตารางที่ 2.4 เส้นทางที่มีบิตผิดพลาดน้อยที่สุดคือ เส้นทางที่ 8 นำไปเปรียบเทียบในตารางที่ 2.3 ค่า Codeword ในตารางที่ 2.3 แถวที่ 8 คือ 1 1 0 1 0 0 1 ตัด 3 บิตสุดท้ายที่เป็นพาริตี้บิต จะได้ค่า Codeword ที่ปลายทางได้รับคือ 1 1 0 1 ปลายทางรับข้อมูลที่ถูกต้อง

<u>ตัวอย่าง</u> การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น m_i = [1 1 0 1] Codeword ที่ผ่านการเข้ารหัสคือ c_i = [1 1 0 1 0 0 1] กำหนดให้บิตผิดพลาด 1 บิต ,c_i +error =[1 1 0 1 1 0 1]

นำ c_i มา *exclusive* or กับเส้นทางในตารางที่ 2.3 แล้วรวมบิต 1 ในแต่ละแถว และเลือกเส้นทางที่มีบิตที่ผิดพลาดน้อยที่สุด เมื่อได้เส้นทางจากบิตที่ผ่านการคำนวณแล้ว จะนำเส้นทางนั้นไปเลือกเส้นทางจากตารางที่ 2.3 ค่าจากเส้นทางนั้นคือ Codeword

ตารางที่ 2.5 การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น m_i = [1 1 0 1]

บิตผิดพลาด 1 บิต

เส้นทาง	C_i *exclusive* or กับเส้นทางในตารางที่ 2.3							Sum bit(1)
1	1	1	0	1	1	0	1	5
2	0	1	1	0	1	0	1	4
3	0	1	0	0	0	1	1	3
4	0	1	0	1	0	0	0	2
5	0	1	1	1	1	1	0	5
6	0	0	1	1	0	0	1	3
7	0	0	0	1	1	1	0	3
8	0	0	0	0	1	0	0	1
9	1	0	0	0	0	0	1	2
10	1	0	1	0	1	1	1	5
11	1	0	1	1	1	0	0	4
12	1	1	1	1	0	1	1	6
13	1	1	1	0	0	0	0	3
14	1	1	0	0	1	1	0	4
15	1	0	0	1	0	1	0	3
16	0	0	1	0	0	1	0	2

จากตารางที่ 2.5 เส้นทางที่มีบิตผิดพลาดน้อยที่สุดคือ เส้นทางที่ 8 นำไปเปรียบเทียบในตารางที่ 2.3 ค่า Codeword ในตารางที่ 2.3 แถวที่ 8 คือ 1 1 0 1 0 0 1 ตัด 3 บิตสุดท้ายที่เป็นพาริตี้บิต จะได้ค่า Codeword ที่ปลายทางได้รับคือ 1 1 0 1 ปลายทางรับข้อมูลที่ถูกต้อง

<u>ตัวอย่าง</u> การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น m_i = [1 1 0 1] Codeword ที่ผ่านการเข้ารหัสคือ c_i = [1 1 0 1 0 0 1] กำหนดให้บิตผิดพลาด 2 บิต ,c_i +error = [1 1 1 0 0 0 1]

นำ c_i มา *exclusive* or กับเส้นทางในตารางที่ 2.3 แล้วรวมบิต 1 ในแต่ละแถว และเลือกเส้นทางที่มีบิตที่ผิดพลาดน้อยที่สุด เมื่อได้เส้นทางจากบิตที่ผ่านการคำนวณแล้ว จะนำเส้นทางนั้นไปเลือกเส้นทางจากตารางที่ 2.3 ค่าจากเส้นทางนั้นคือ Codeword

ตารางที่ 2.6 การถอดรหัสแบบ Hard-decision ให้บิตที่เข้ารหัส BCH เป็น m_i = [1 1 0 1] บิตผิดพลาด 2 บิต

เส้นทาง	C_i *exclusive* or กับเส้นทางในตารางที่ 2.3							Sum bit(1)
1	1	1	1	0	0	0	1	4
2	0	1	1	0	1	0	1	3
3	0	1	0	0	0	1	1	6
4	0	1	0	1	0	0	0	3
5	0	1	1	1	1	1	0	2
6	0	0	1	1	0	0	1	2
7	0	0	0	1	1	1	0	3
8	0	0	0	0	1	0	0	2
9	1	0	0	0	0	0	1	6
10	1	0	1	0	1	1	1	4
11	1	0	1	1	1	0	0	1
12	1	1	1	1	0	1	1	5
13	1	1	1	0	0	0	0	4
14	1	1	0	0	1	1	0	5
15	1	0	0	1	0	1	0	4
16	0	0	1	0	0	1	0	3

จากตารางที่ 2.6 เส้นทางที่มีบิตผิดพลาดน้อยที่สุดคือ เส้นทางที่ 11 นำไปเปรียบเทียบในตารางที่ 2.3 ค่า Codeword ในตารางที่ 2.3 แถวที่ 11 คือ 0 1 1 0 0 0 1 ตัด 3 บิตสุดท้ายที่เป็นพาริตี้บิต จะได้ค่า Codeword ที่ปลายทางได้รับคือ 0 1 1 0 ปลายทางตัดสินบิตผิดพลาด

จากตัวอย่างเมื่อบิตมีการผิดพลาดมากกว่า 1 บิต ปลายทางจะตัดสินผิดพลาด เนื่องจากการเป็นข้อจำกัดของการถอดรหัสนี้ที่สามารถแก้ไขข้อผิดพลาดที่ผิดได้เพียงบิตเดียว

จากตัวอย่างอย่างข้างต้นจะเห็นว่า การถอดรหัสแบบ Viterbi Algorithm เมื่อมีการเข้ารหัสที่มี state ที่มากขึ้น เช่น รหัส BCH(15,7) จะประกอบด้วย state 256 states, BCH(31,21) จะประกอบด้วย state 1024 states ทำให้มีความซับซ้อนที่มากขึ้น การเข้ารหัสเช่นนี้ส่งผลที่ผลที่เกิดความยุ่งยาก และซับซ้อน จึงเป็นเหตุผลที่การถอดรหัสด้วยวิธีของ Viterbi Algorithm ไม่เหมาะสมกับการถอดรหัสที่มี state มากเกินไป จึงเป็นเหตุผลให้ในโครงงานชิ้นนี้ทำการศึกษาเฉพาะรหัส BCH (7,4) และถอดรหัสแบบ Viterbi Algorithm ในกรณีของ Hard-decision

ในบทที่ 2 นี้เป็นการแสดงทฤษฎีต่างที่ใช้ในงานวิจัย และได้แสดงตัวอย่างการทำงานเพื่อที่จะแสดงให้ผู้สนใจในการศึกษางานวิจัยฉบับนี้ทำความเข้าใจได้ง่ายขึ้น ซึ่งข้อมูลในบทนี้จะเป็นพื้นฐานในการสร้างโปรแกรมที่ใช้ในการทดลองในบทที่ 3 บทที่ 4 และบทที่ 5 ฉะนั้นทฤษฎีและวิธีการต่างที่ใช้ได้แสดงไว้ในบทที่ 2 นี้

บทที่ 3
การออกแบบโครงงาน และวิธีการดำเนินงาน

ในบทนี้กล่าวถึงโครงสร้างการทำงาน และการออกแบบการเขียนโปรแกรมจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm อีกทั้งยังอธิบายวิธีการสร้าง Graphic User Interfaces เพื่อความสะดวกในการใช้งาน โดยแบ่งออกเป็นขั้นตอนดังนี้

3.1 ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm ซึ่งรายละเอียดดังกล่าวได้กล่าวไว้ในบทที่ 2

3.2 ออกแบบด้วย โปรแกรม MATLAB ในการสร้างแบบจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

ขั้นตอนการออกแบบโปรแกรม

ออกแบบด้วย โปรแกรม MATLAB ในการสร้างแบบจำลองการเข้ารหัสแบบ BCH Codes และถอดรหัสแบบ Viterbi Algorithm

3.2.1 การสร้างสัญญาณ

ทำการสร้างรหัสสัญญาณขึ้นมา เพื่อนำสัญญาณนี้มาเข้ารหัสแบบ BCH ก่อนที่จะทำการส่งเข้าไปในระบบสื่อสาร

3.2.2 การเข้ารหัสช่องสัญญาณ

นำสัญญาณที่สร้างขึ้นมา เข้ารหัส BCH โดยมีอัตราการเข้ารหัส คือ อัตราการเข้ารหัส (7,4)

3.2.3 การสร้างสัญญาณรบกวน

จำลองการสร้างสัญญาณรบกวน โดยใช้วิธีการสุ่ม การเพิ่มสัญญาณรบกวนเข้าไปในข้อมูลที่ทำการเข้ารหัส BCH เพื่อใช้เป็นข้อพิสูจน์ในการแก้ไขบิตที่ผิดพลาดที่ภาครับ ว่ามีความสามารถในการแก้ไขบิตที่ผิดพลาดหรือไม่

3.2.4 การรวมสัญญาณที่เข้ารหัสกับสัญญาณรบกวน

นำข้อมูลที่ทำการเข้ารหัส BCH รวมกับสัญญาณรบกวนที่ถูกจำลองขึ้นมา ซึ่งในขั้นตอนนี้เป็นขั้นตอนที่สมมุติว่าในกระบวนการส่งข้อมูลของระบบสื่อสารนั้นเกิดข้อผิดพลาด เพื่อให้ส่วนของการถอดรหัสที่ปลายทาง ทำการแก้ไขข้อมูลที่ผิดพลาดนี้ให้มีความถูกต้อง

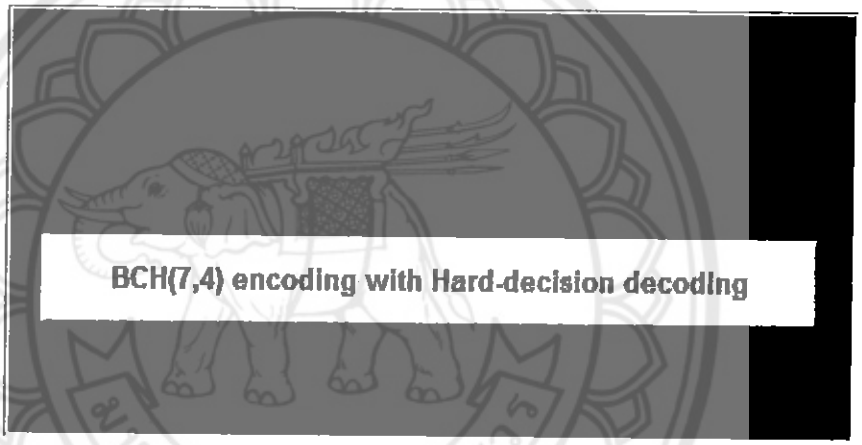
3.2.5 การแก้ไขข้อมูลที่ผิดพลาดที่ภาครับ

ขั้นตอนสุดท้าย ทำการออกแบบโปรแกรมในการแก้ไขบิตที่ผิดพลาด โดยใช้วิธีการของ Viterbi Algorithm ในกรณีของ Hard-decision ซึ่งรายละเอียดได้กล่าวไว้ในบทที่ 2

3.3 การออกแบบ Graphic User Interfaces และขั้นตอนการดำเนินงาน

3.3.1 Type of BCH Codes

ขั้นตอนนี้ทำการสร้างหน้าต่างในการเลือกชนิดของการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision ดังแสดงไว้ในรูปที่ 3.1



รูปที่ 3.1 Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสและถอดรหัส

จากรูปที่ 3.1 เป็นรูปแบบของ Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสและถอดรหัส ที่พร้อมสำหรับการใช้งาน เมื่อทำการคลิกที่ปุ่ม BCH(7,4) encoding with Hard-decision decoding จะปรากฏหน้าต่างของ Application ของรหัสที่ทำการเลือก ดังแสดงในหัวข้อที่ 3.3.2

3.3.2 Application

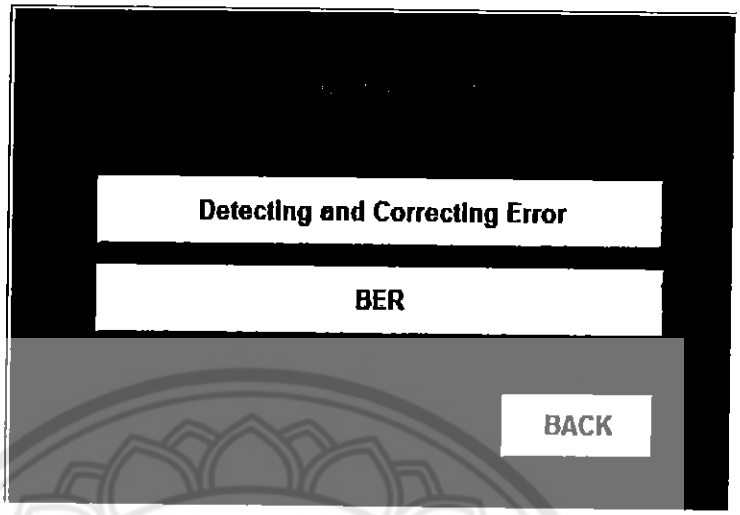
ขั้นตอนนี้จะทำการสร้างหน้าต่าง ในการแสดงผลของการเข้ารหัสสัญญาณ BCH Codes แบบต่างๆ มีขั้นตอนดังนี้ คือ

3.3.2.1 สร้างปุ่ม Push Button เพื่อแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด

3.3.2.2 สร้างปุ่ม Push Button เพื่อแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

3.3.2.3 สร้างปุ่ม Push Button สำหรับย้อนกลับไปสู่หน้าต่างการเลือกอัตราการเข้ารหัสและถอดรหัส

ซึ่งขั้นตอนดังกล่าวได้แสดงไว้ในรูปที่ 3.2



รูปที่ 3.2 Graphic User Interfaces ในการเลือกแสดงตัวอย่าง หรือคำนวณ Bit Error Rate

จากรูปที่ 3.2 เป็นรูปแบบของ Graphic User Interfaces ในการเลือกแสดงตัวอย่าง หรือคำนวณ Bit Error Rate เมื่อคลิกที่ปุ่ม Detecting and Correcting Error ของรหัส จะปรากฏหน้าต่างของการตรวจสอบการแก้ไขข้อมูลที่ผิดพลาดของรหัสที่เลือก ซึ่งแสดงในหัวข้อที่ 3.3.3 เมื่อคลิกที่ปุ่ม BER จะปรากฏหน้าต่างเพื่อดูกราฟแสดงระดับอัตราการเกิดข้อมูลผิดพลาด ซึ่งแสดงในหัวข้อที่ 3.3.4

3.3.3 Detecting and Correcting Error

เป็นการแสดงตัวอย่างการเข้ารหัสและถอดรหัส ที่กำหนดให้ใส่บิตข้อมูลที่เป็นสัญญาณในการส่งไม่มากนัก เพื่อใช้ในการตรวจสอบการถอดรหัสสัญญาณที่ปลายทาง ว่ามีความสามารถในการแก้ไขข้อผิดพลาดหรือไม่ เมื่อทำการคลิกที่ปุ่ม Detecting and Correcting Error จะปรากฏหน้าต่างดังแสดงในรูปที่ 3.3

ในขั้นตอนนี้จะทำการสร้างหน้าต่างในการแสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดของการเข้ารหัสสัญญาณ BCH Codes และการถอดรหัสแบบ Hard-decision ซึ่งมีขั้นตอนดังต่อไปนี้

3.3.3.1 สร้างช่อง Edit Text สำหรับใส่บิตข้อมูลที่มีจำนวนไม่มาก เพื่อใช้ในการแสดงการตรวจสอบตัวถอดรหัสสัญญาณ ว่ามีความสามารถในการแก้ไขบิตที่ผิดพลาดหรือไม่

3.3.3.2 สร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด ซึ่งคือ Probability of bit error ขึ้นในขั้นตอนแรก ซึ่งค่านี้จะนำไปคำนวณหาค่า Noise ที่เกิดขึ้นภายในระบบสื่อสาร

3.3.3.3 สร้างปุ่ม Push Button เพื่อนำค่าที่บิตข้อมูล และความน่าจะเป็นในกรณีเกิดข้อมูลผิดพลาดไปคำนวณ

3.3.3.4 สร้างช่อง Edit Text สำหรับแสดงข้อมูลดังนี้ คือ

1.บิตข้อมูลที่ได้จากการใส่ค่า

2.บิตที่ข้อมูลที่ทำการเข้ารหัส BCH Codes แล้ว

3.บิตข้อมูลผิดพลาดที่ได้จากการสุ่ม

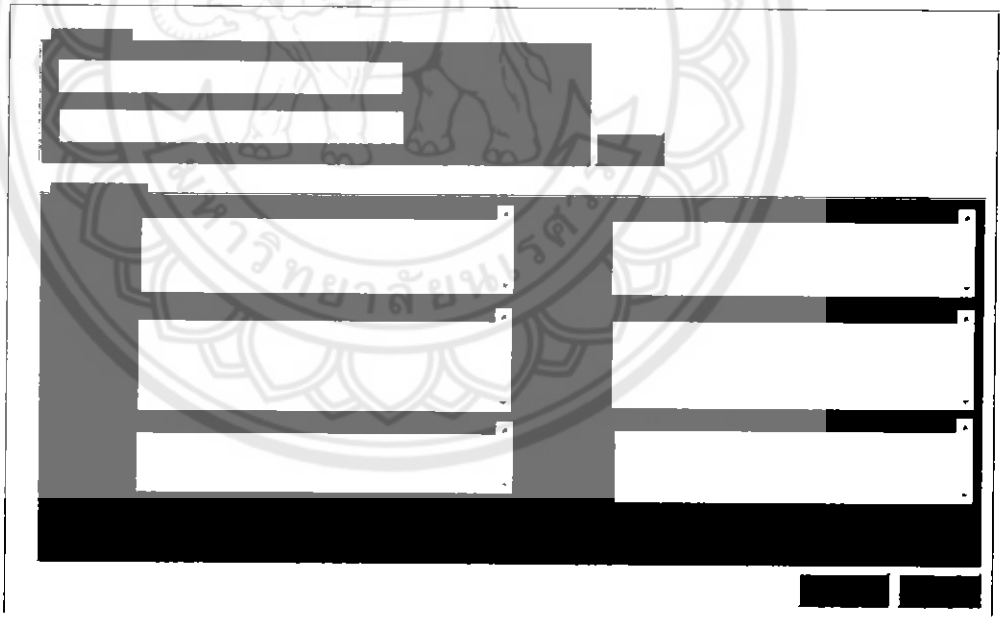
4.บิตข้อมูลที่ได้จากการเข้ารหัส BCH Codes รวมกับบิตข้อมูลที่ผิดพลาด

5.บิตข้อมูลที่ผ่านการถอดรหัส

3.3.3.5 สร้างปุ่ม Push Button สำหรับ Reset ค่า

3.3.3.6 สร้างปุ่ม Push Button สำหรับ Back ไปสู่หน้าต่าง Application

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.3



รูปที่ 3.3 รูปแบบของ Graphic User Interfaces แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดของการเข้ารหัสสัญญาณ BCH Codes และการถอดรหัสแบบ Hard-decision

3.3.4 BER

BER (Bit error rate) คืออัตราส่วนของข้อมูลที่ผิดพลาดต่อข้อมูลทั้งหมด โดยค่า BER ที่ได้ต้องมีค่าน้อยๆ หรือมีค่าเป็นศูนย์ เพื่อปลายทางสามารถรับข้อมูลที่มีความถูกต้อง

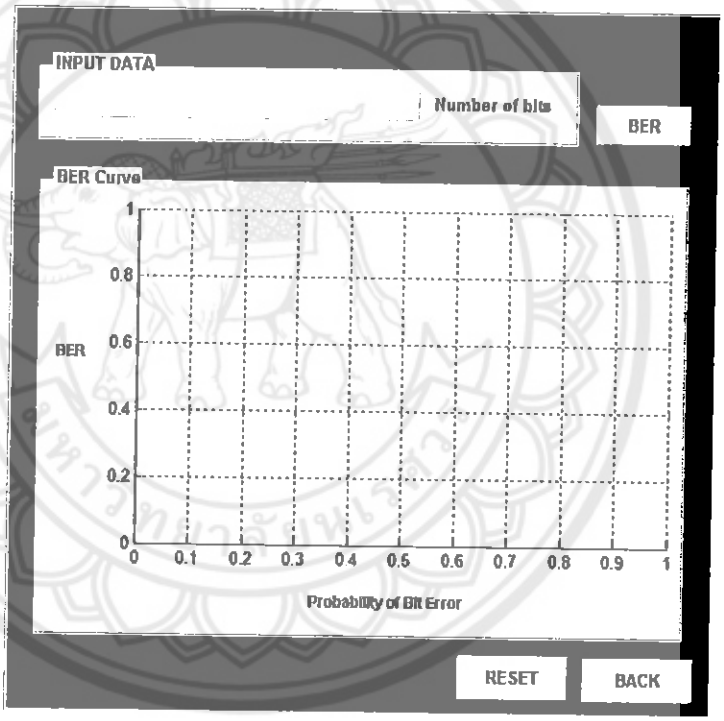
ในขั้นตอนนี้จะทำการสร้างหน้าต่างแสดงอัตราส่วนของข้อมูลที่ผิดพลาดต่อข้อมูลทั้งหมด

3.3.4.1 สร้างช่อง Edit Text สำหรับใส่บิตข้อมูล ในส่วนนี้จะเป็นบิตข้อมูลที่ใช้ในการส่งสัญญาณ

3.3.4.2 สร้างปุ่ม Push Button เพื่อนำข้อมูลที่ต้องการส่งไปคำนวณ

3.3.4.3 สร้าง Axes เพื่อแสดงระดับการเกิดข้อผิดพลาดของ Hard-Decision

ขั้นตอนดังกล่าวแสดงในรูปที่ 3.4



รูปที่ 3.4 Graphic User Interfaces เพื่อแสดงระดับการเกิดข้อผิดพลาดของ Hard-Decision

ในบทที่ 3 ได้แสดงหลักการทำงานของโปรแกรม และได้อธิบายวิธีการสร้าง Graphic User Interfaces เพื่อใช้ในการแสดงการทำงานของโปรแกรม ในรูปแบบที่เข้าใจง่ายและสะดวกต่อการใช้งาน ซึ่งสามารถแบ่งได้เป็น

1. หน้าต่างที่ใช้แสดงการเลือกการเข้ารหัสสัญญาณ BCH Codes และการถอดรหัสแบบ Hard-decision

2. หน้าต่างที่ใช้เลือกระหว่างแสดงการแก้ไขข้อมูลที่ผิดพลาดกับหน้าต่างที่ใช้แสดงระดับอัตราการเกิดข้อมูลผิดพลาด

3. หน้าต่างที่ใช้แสดงการแก้ไขข้อมูลที่ผิดพลาด

4. หน้าต่างที่ใช้แสดงระดับอัตราการเกิดข้อมูลผิดพลาด

ในบทต่อไปจะแสดงผลการทดลองของโปรแกรมการคำนวณ และแสดงค่าต่างๆที่ได้สร้างไว้ในบทนี้ โดยทดลองใส่ค่าบิตข้อมูลลงไปในโปรแกรม เพื่อให้แสดงค่าต่างๆที่ต้องการ

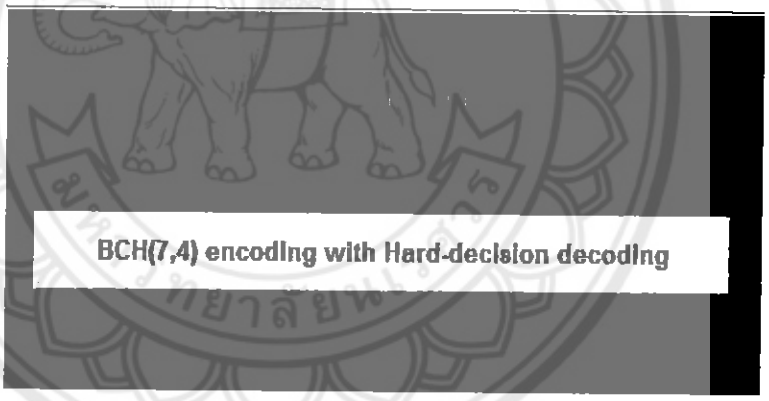
บทที่ 4
ผลการดำเนินโครงงาน

ในบทนี้จะทำการแสดงผลการทดลองในการใส่บิตข้อมูลเข้าไปในโปรแกรมจำลองการเข้ารหัสสัญญาณแบบ BCH Codes และการถอดรหัสแบบ Hard-decision ซึ่งสามารถแสดงการแก้ไขข้อมูลผิดพลาด และแสดงระดับอัตราการเกิดข้อมูลผิดพลาด โดยแสดงผลการทดลองในรูปแบบของ Graphic User Interfaces

4.1 โปรแกรมแสดงการแก้ไขข้อมูลผิดพลาด

4.1.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1.ในขั้นตอนแรก เมื่อทำการเปิดหน้าต่างการเลือกชนิดของการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision ดังรูปที่ 4.1



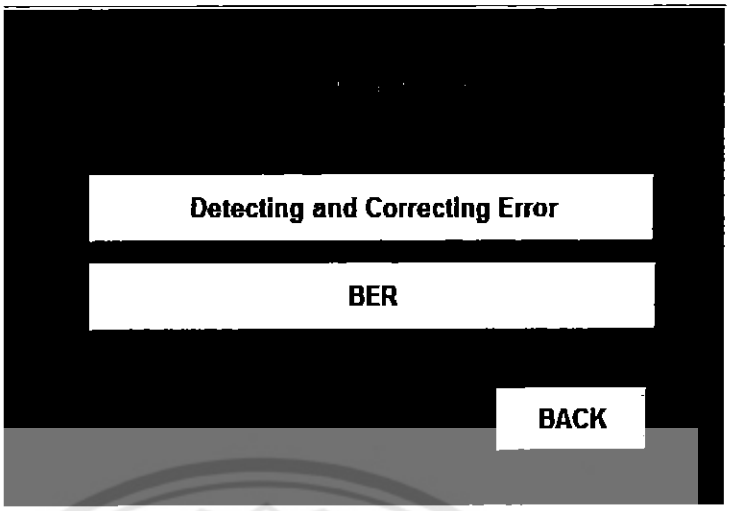
รูปที่ 4.1 หน้าต่างสำหรับการเลือกอัตราการเข้ารหัสช่องสัญญาณ

2.เมื่อทำการคลิกที่ช่องสัญญาณแล้ว จะทำให้หน้าต่าง Application ปรากฏขึ้นมา ซึ่งหน้าต่างนี้จะมีปุ่มให้เลือกสองปุ่ม ได้แก่

2.1 Detecting and Correcting Error คือ หน้าต่างแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด

2.2 BER คือ หน้าต่างแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

ซึ่งหน้าต่าง Application แสดงไว้ในรูปที่ 4.2



รูปที่ 4.2 หน้าต่าง Application ใช้ในการเลือกระหว่างหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด และหน้าต่างแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

3.เมื่อทำการเปิดหน้าต่าง Detecting and Correcting Error จะปรากฏหน้าต่างดังรูปที่ 4.3 ซึ่งรายละเอียดต่างๆในการใส่ค่าต่างๆมีดังนี้

3.1 ช่อง Data Bits คือช่องใส่ค่าบิตที่ต้องการส่ง

3.2 ช่อง Probability of Bit error คือช่องใส่ความน่าจะเป็นในการเกิดบิตผิดพลาด

3.3 ปุ่ม RUN จะนำค่าบิตที่ต้องการส่งและความน่าจะเป็นในการเกิดบิตผิดพลาดเข้าสู่โปรแกรมการคำนวณ และแสดงผล

3.4 ช่อง Data bits ในส่วนของ Data Display จะแสดงผลการใส่ค่าบิตที่ต้องการส่ง

3.5 ช่อง BCH Encoder จะนำบิตที่ต้องการส่งมาเข้ารหัส BCH

3.6 ช่อง Noise จะแสดงผลการสุ่มของบิตผิดพลาด

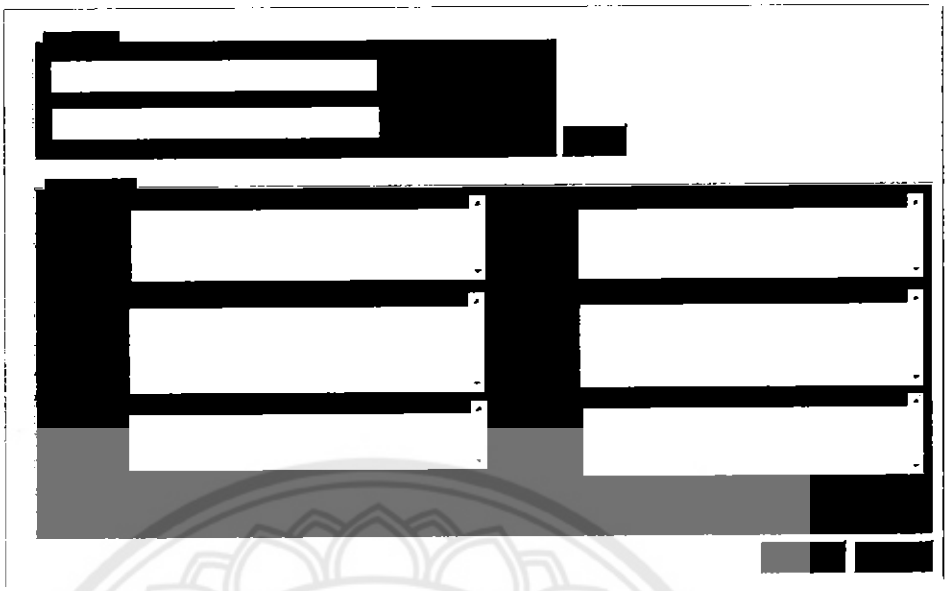
3.7 ช่อง BCH Encoder + Noise จะนำค่าสัญญาณที่ได้จากการเข้ารหัส BCH มารวมกับ Noise ที่เกิดขึ้น เพื่อทำให้เกิดการผิดพลาดของข้อมูล

3.8 ช่อง Decoder of hard-decision แสดงการถอดรหัส ทำให้ได้สัญญาณที่ส่งกลับมา

3.9 ช่อง Bit error of Hard-decision แสดงผลของบิตที่แก้ไขผิดพลาด เกิดเป็น error

3.10 ปุ่ม Reset เมื่อกดปุ่มนี้จะทำการรีเซ็ตค่าในหน้าต่างแสดงการทำงาน เพื่อพร้อมสำหรับทำงานต่อไป

3.10 ปุ่ม Back เมื่อกดปุ่มนี้ โปรแกรมจะกลับไปหน้า Application ซึ่งแสดงดังรูปที่ 4.3



รูปที่ 4.3 หน้าต่างการแก้ไขข้อมูลที่ผิดพลาด

4.1.2 ตัวอย่างการแก้ไขข้อมูลผิดพลาด

1.คลิกเลือกชนิดของการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision ดังรูปที่ 4.1

2.เมื่อทำการคลิกที่ช่องสัญญาณแล้ว จะทำให้หน้าต่าง Application ปรากฏขึ้นมาให้เลือก Detecting and Correcting Error

3.ใส่ค่าต่างๆมีดังนี้

3.1 ช่อง Data Bits คือช่องใส่ค่าบิตที่ต้องการส่ง

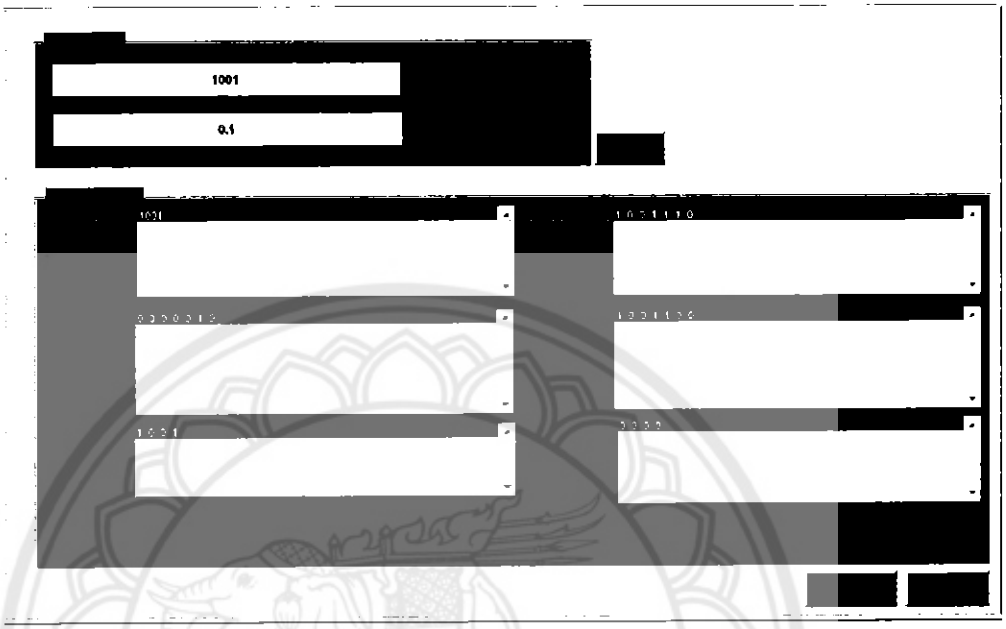
3.2 ช่อง Probability of Bit error คือช่องใส่ความน่าจะเป็นในการเกิดบิตผิดพลาด

3.3 ปุ่ม RUN จะนำค่าบิตที่ต้องการส่งและความน่าจะเป็นในการเกิดบิตผิดพลาดเข้าสู่โปรแกรมการคำนวณ และแสดงผล

ซึ่งผลลัพธ์ของการรันโปรแกรมแสดงในรูปต่างๆดังนี้

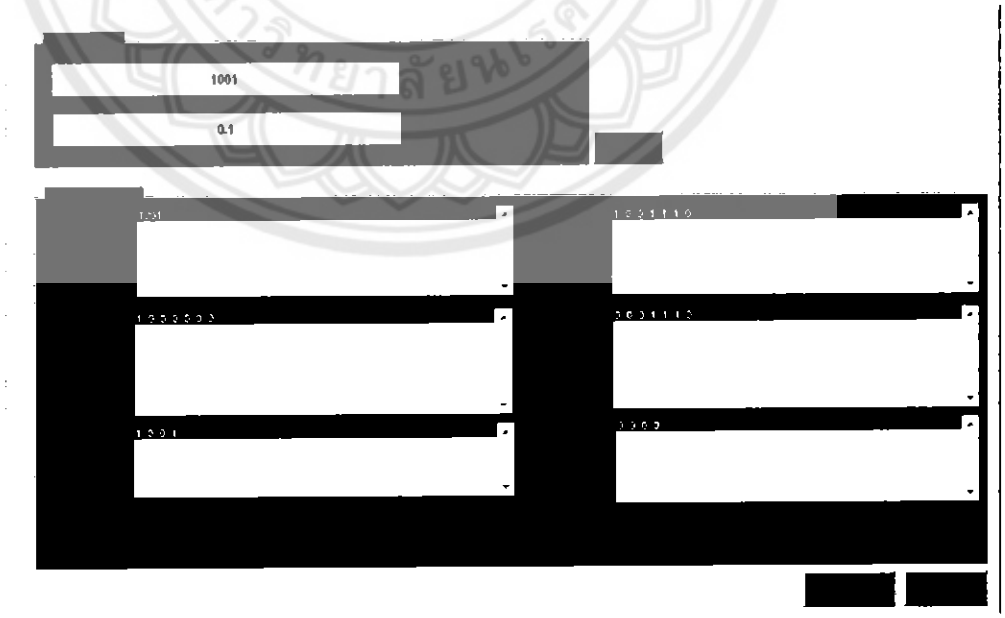
4.1.2.1 ตัวอย่างการแก้ไขข้อมูลที่ผิดพลาด เมื่อใส่ Data Bits 4 บิต คือ1001

<u>กรณีที่ 1</u>: ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1 แสดงในรูปที่ 4.4



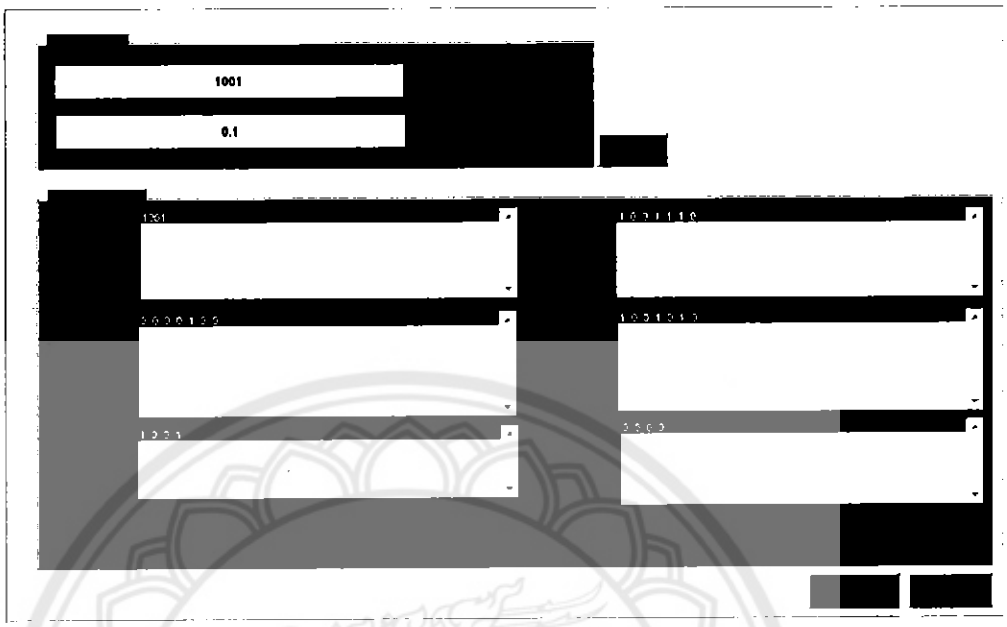
รูปที่ 4.4 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1

ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2 แสดงในรูปที่ 4.5



รูปที่ 4.5 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2

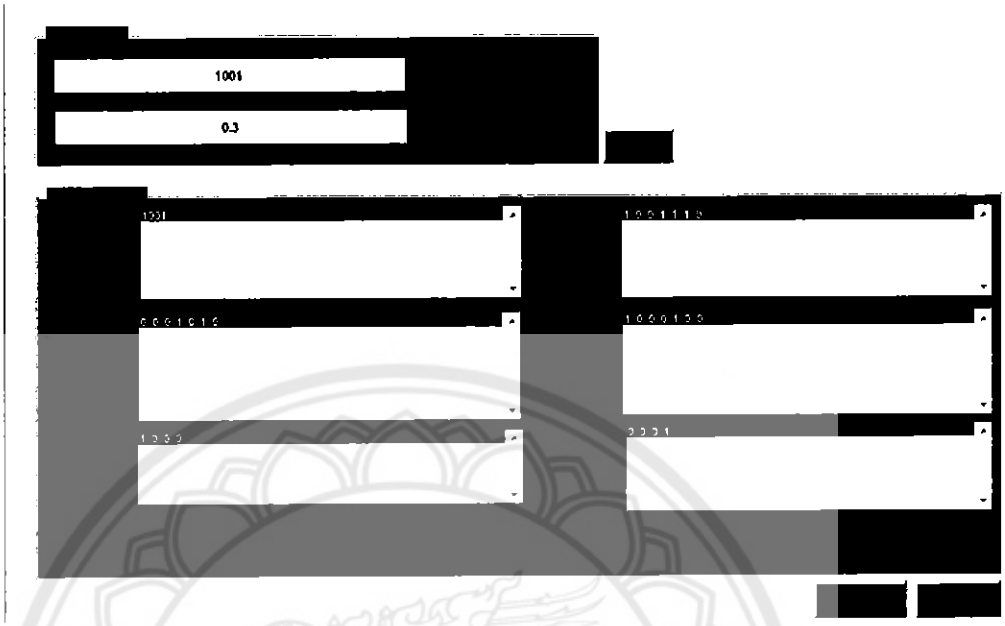
ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3 แสดงในรูปที่ 4.6



รูปที่ 4.6 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3

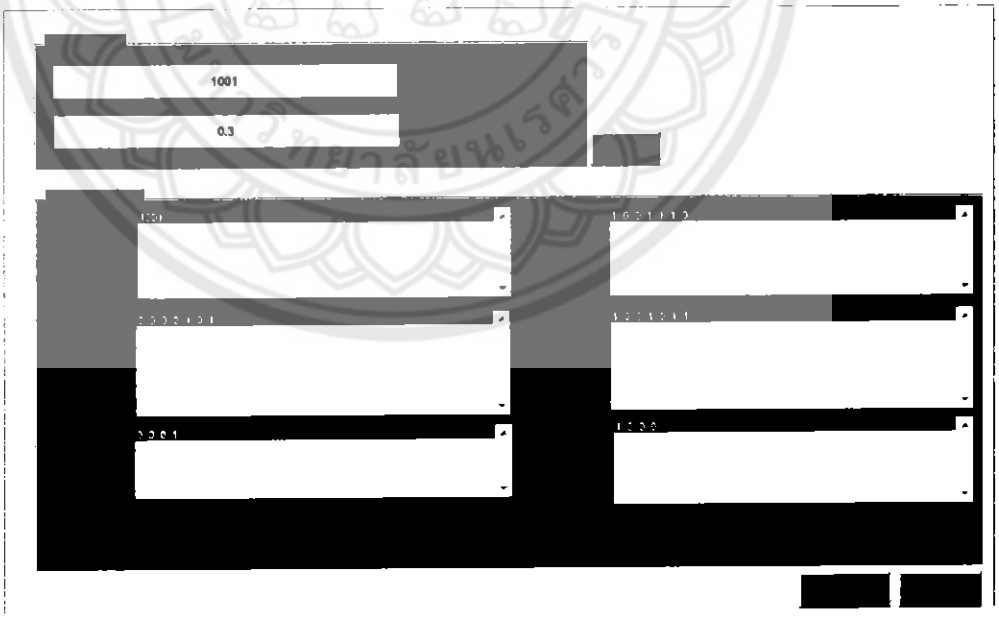
จากกรณีที่ 1 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 4 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.1 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.4 รูปที่ 4.5 และรูปที่ 4.6 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 4 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 4 บิตในหนึ่งบล็อก ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น 3 บิต ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.1 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมสามารถแก้ไขบิตที่ผิดพลาดได้

<u>กรณีที่ 2</u>: ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1 แสดงในรูปที่ 4.7



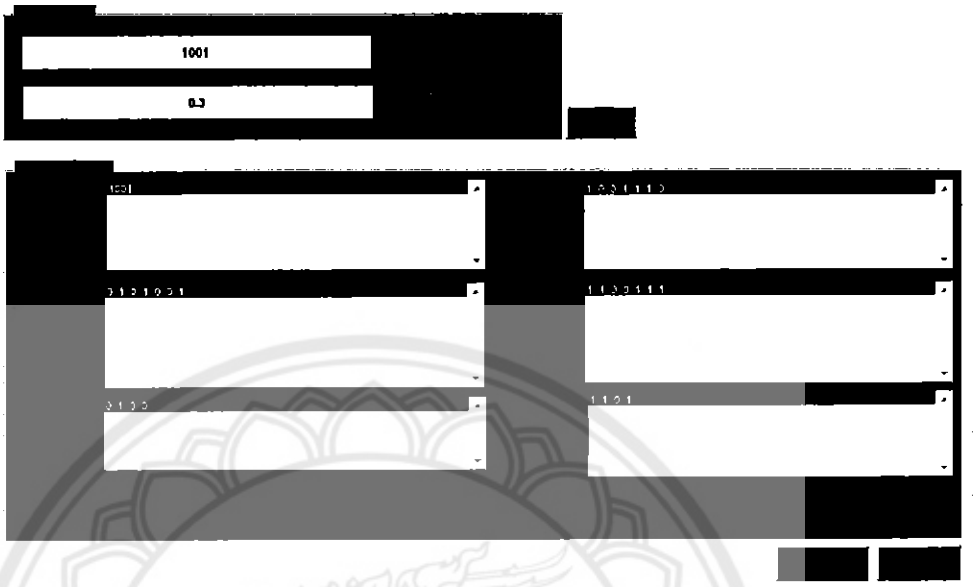
รูปที่ 4.7 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1

ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2 แสดงในรูปที่ 4.8



รูปที่ 4.8 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2

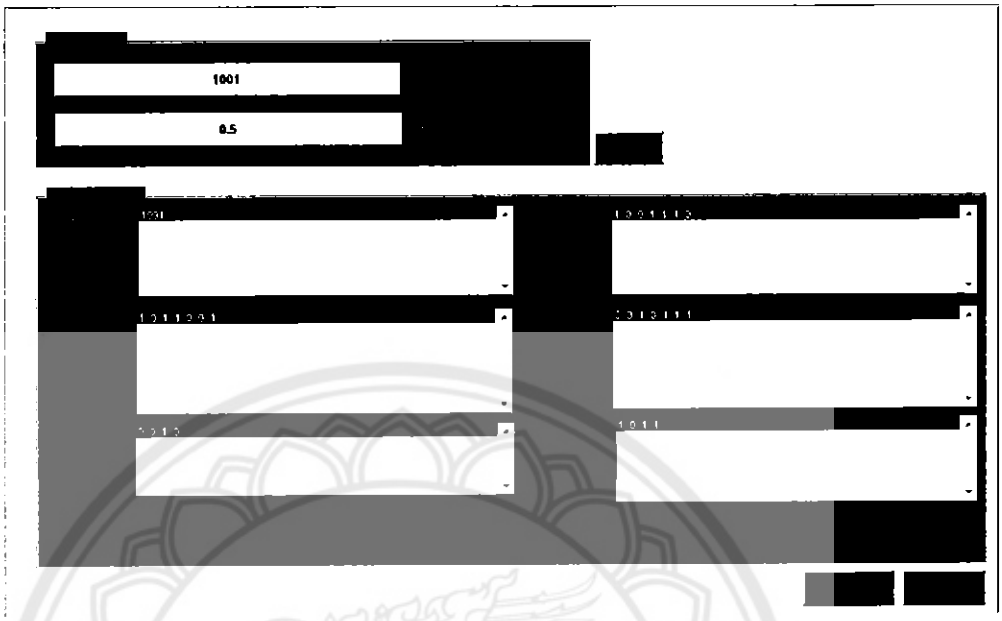
ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2 แสดงในรูปที่ 4.8



รูปที่ 4.9 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 3

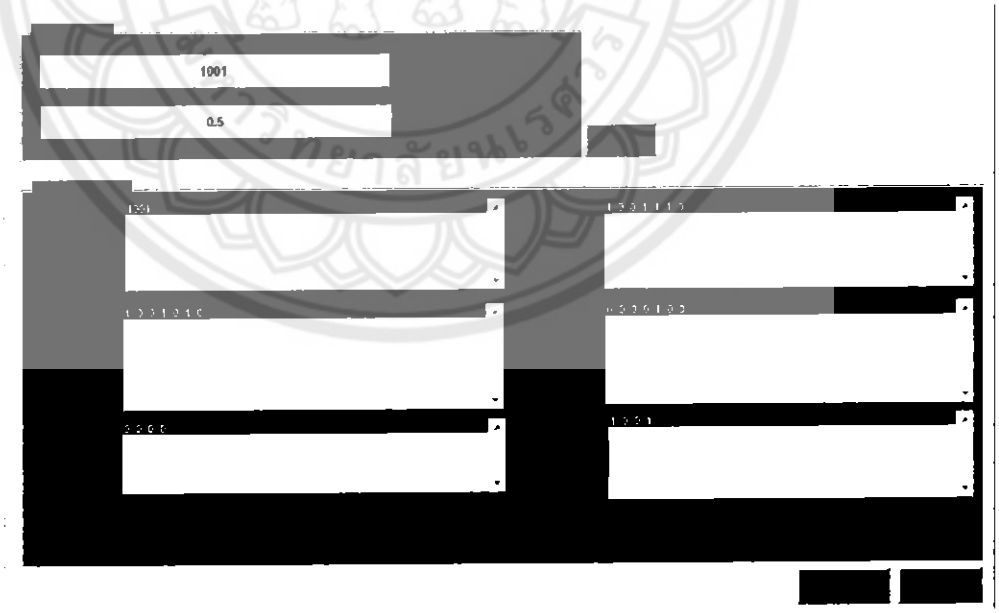
จากกรณีที่ 2 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 4 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.3 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.7 รูปที่ 4.8 และรูปที่ 4.9 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 4 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 4 บิตในหนึ่งบล็อก ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น 3 บิต ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.3 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมไม่สามารถแก้ไขบิตที่ผิดพลาดได้ เพราะมีขีดจำกัดในความสามารถแก้ไขบิตผิดพลาดได้เพียงหนึ่งบิต

<u>กรณีที่ 3</u>: ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 1 แสดงในรูปที่ 4.10



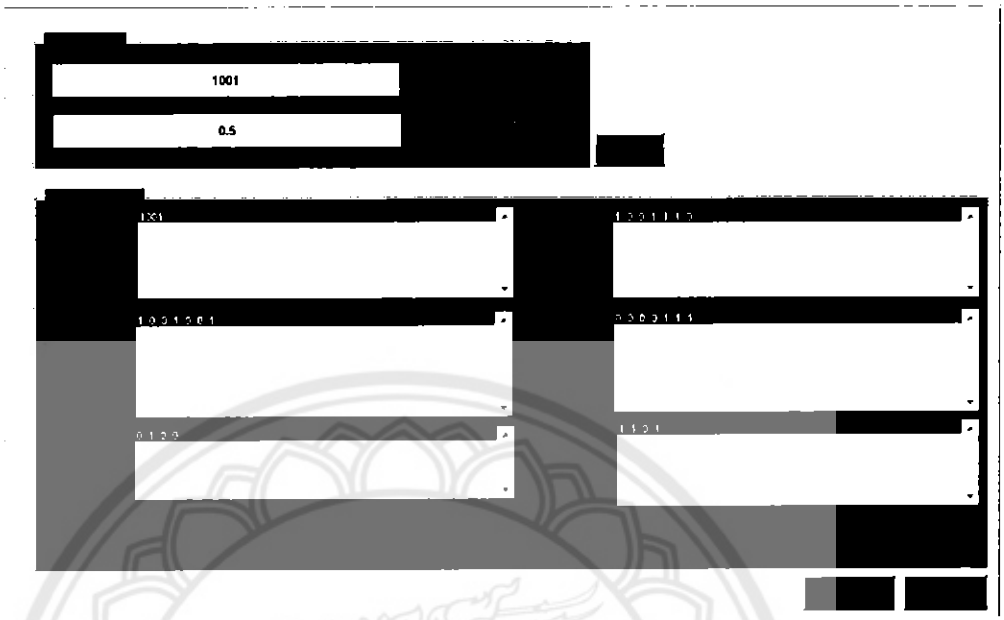
รูปที่ 4.10 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กด**รันครั้งที่** 1

ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2 แสดงในรูปที่ 4.11



รูปที่ 4.11 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2

ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3 แสดงในรูปที่ 4.12

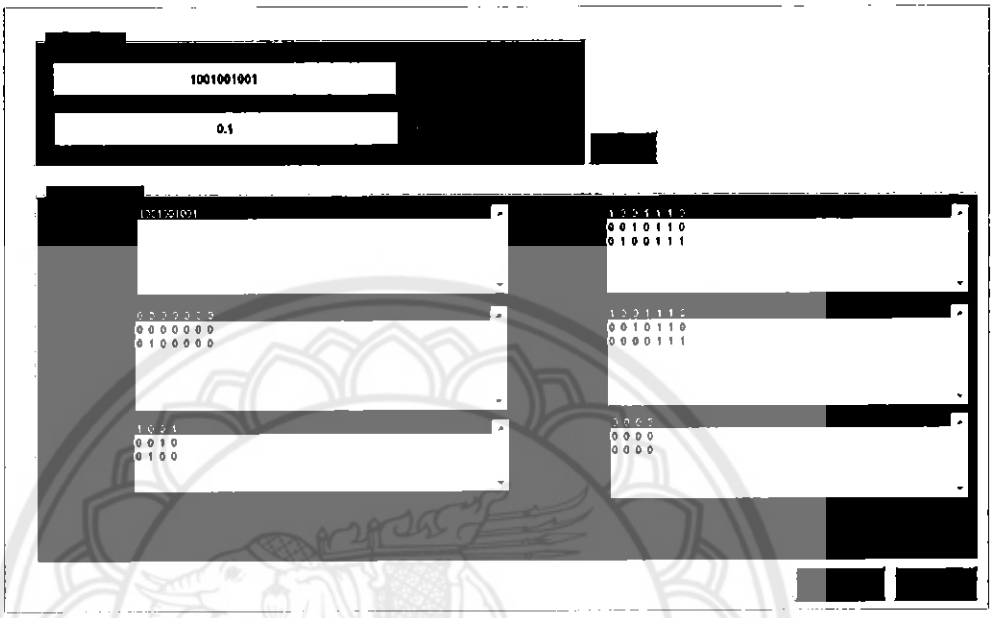


รูปที่ 4.12 แสดงตัวอย่าง การใส่ Data Bits 4 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3

จากกรณีที่ 3 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 4 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.5 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.10 รูปที่ 4.11 และรูปที่ 4.12 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 4 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 4 บิตในหนึ่งบล็อก ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น 3 บิต ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.5 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมไม่สามารถแก้ไขบิตที่ผิดพลาดได้ เพราะมีขีดจำกัดในความสามารถแก้ไขบิตผิดพลาดได้เพียงหนึ่งบิต

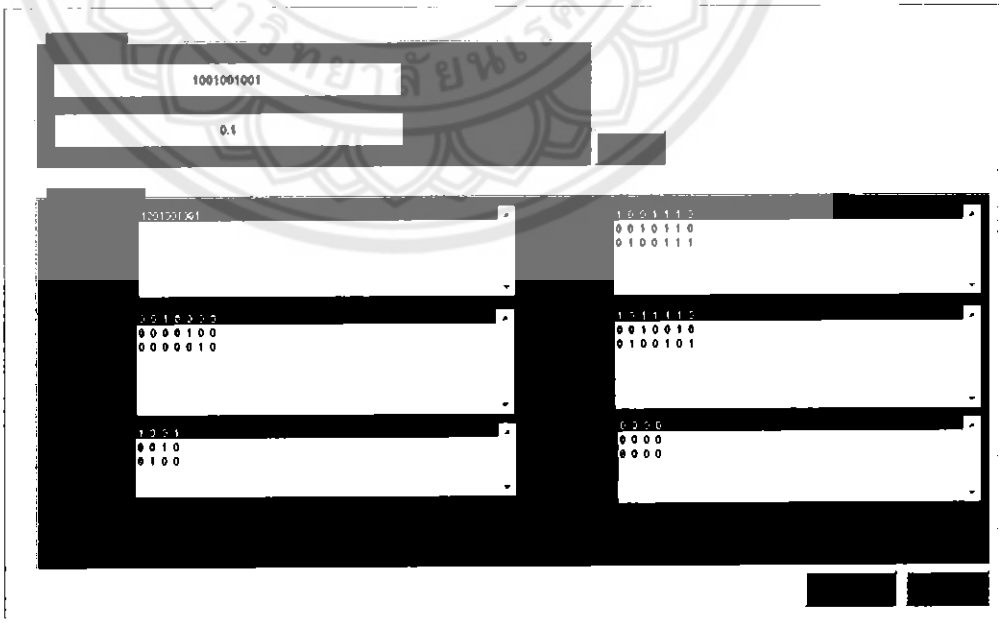
4.1.2.2 ตัวอย่างการแก้ไขข้อมูลที่ผิดพลาด เมื่อใส่ Data Bits 10 บิต คือ1001001001

<u>กรณีที่ 1</u>: ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1 แสดงในรูปที่ 4.13



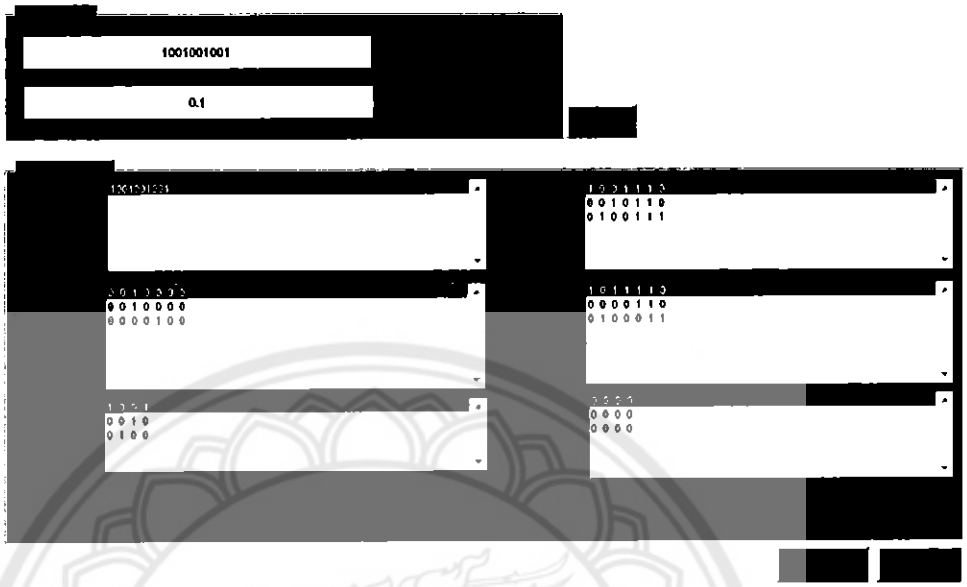
รูปที่ 4.13 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 1

ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2 แสดงในรูปที่ 4.14



รูปที่ 4.14 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 2

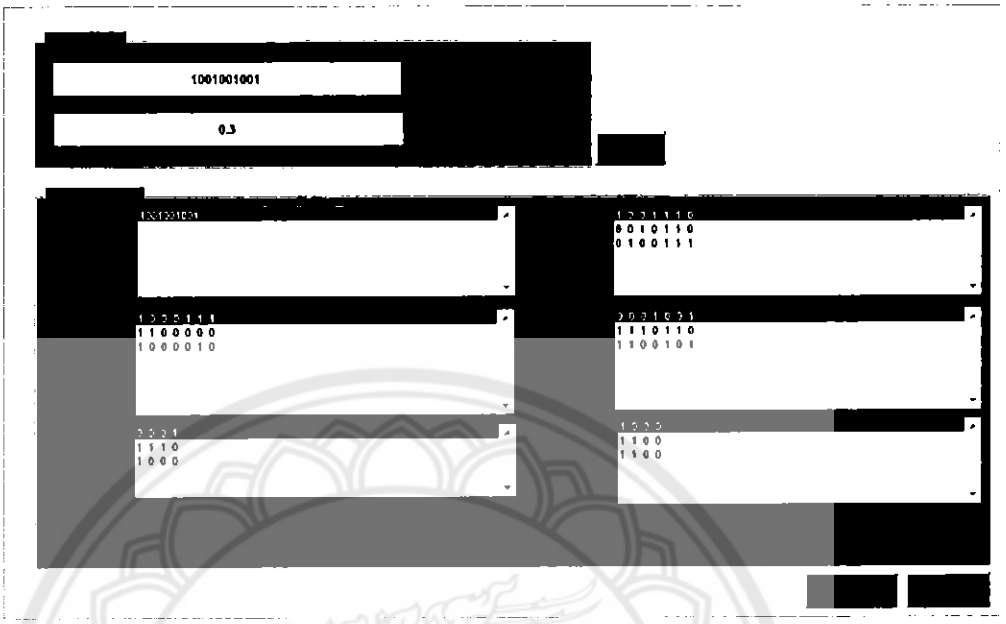
ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3 แสดงในรูปที่ 4.15



รูปที่ 4.15 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.1 กดรันครั้งที่ 3

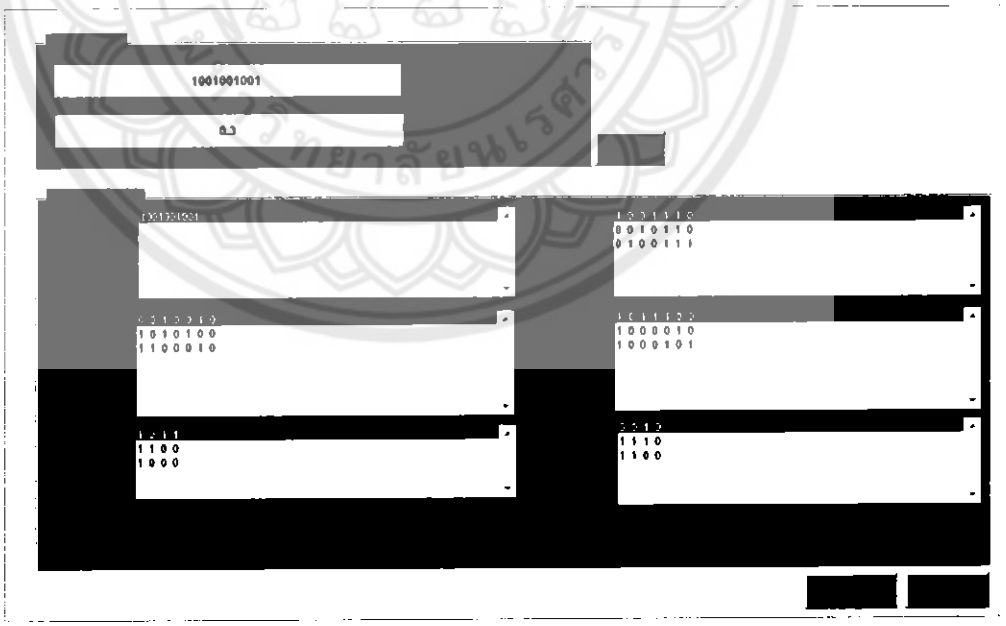
จากกรณีที่ 1 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 10 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.1 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.13 รูปที่ 4.14 และรูปที่ 4.15 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 10 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 12 บิต เนื่องจากการส่งข้อมูลครั้งละ 4 บิตในหนึ่งบล็อก ในกรณีนี้จะส่งทั้งหมดสามบล็อก ทำให้มีบิตในการส่งทั้งหมด 12 บิต โดยที่บิตที่เกินมาจากข้อมูลที่ส่งนั้น จะถูกกำหนดเพิ่มให้มีค่าเป็นบิตศูนย์ ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.1 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมสามารถแก้ไขบิตที่ผิดพลาดได้

<u>กรณีที่ 2</u>: ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1 แสดงในรูปที่ 4.16



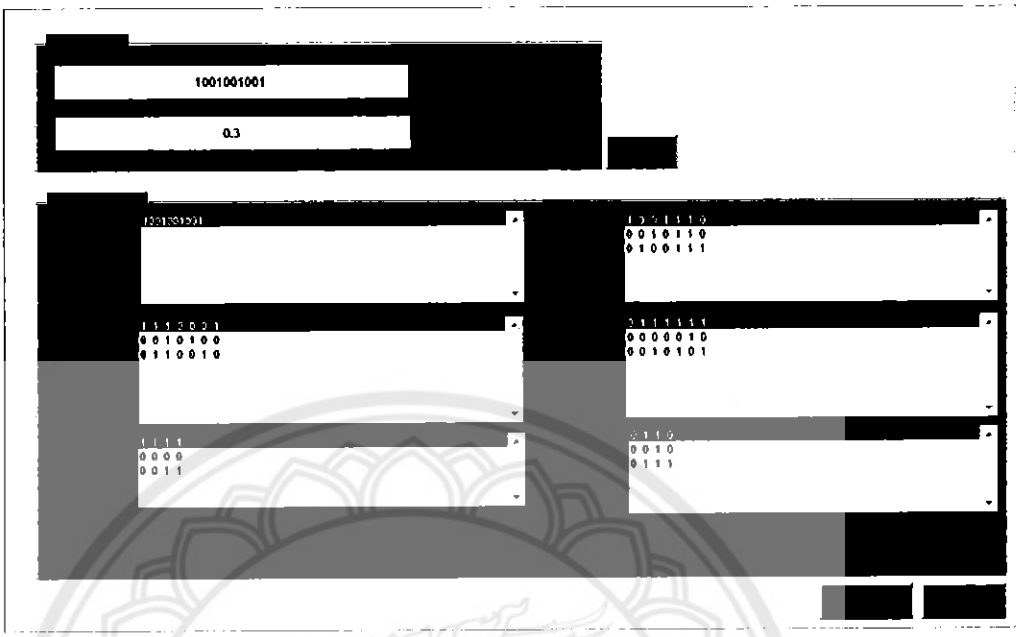
รูปที่ 4.16 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 1

ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2 แสดงในรูปที่ 4.17



รูปที่ 4.17 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 2

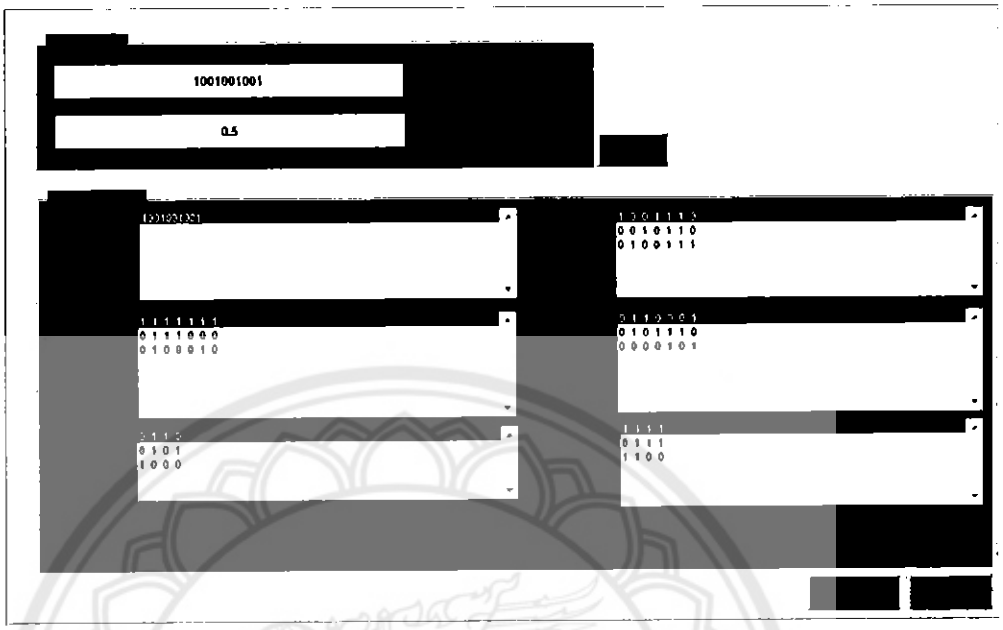
ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 3 แสดงในรูปที่ 4.18



รูปที่ 4.18 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.3 กดรันครั้งที่ 3

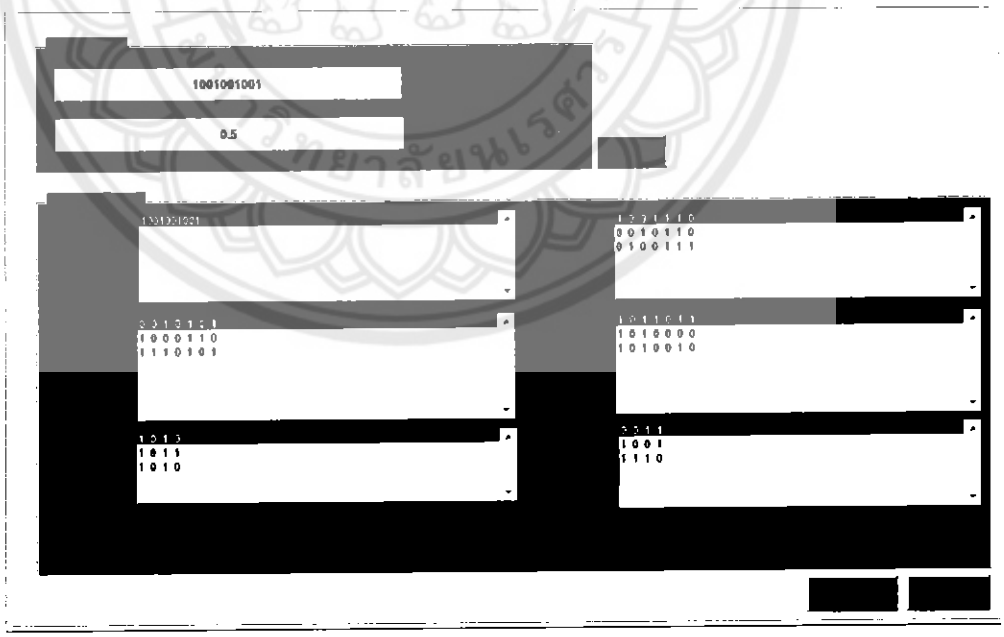
จากกรณีที่ 2 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 10 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.3 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.16 รูปที่ 4.17 และรูปที่ 4.18 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 10 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 12 บิต เนื่องจากการส่งข้อมูลครั้งละ 4 บิตในหนึ่งบล็อก ในกรณีนี้จะส่งทั้งหมดสามบล็อก ทำให้มีบิตในการส่งทั้งหมด 12 บิต โดยที่บิตที่เกินมาจากข้อมูลที่ส่งนั้น จะถูกกำหนดเพิ่มให้มีค่าเป็นบิตศูนย์ ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.3 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมไม่สามารถแก้ไขบิตที่ผิดพลาดได้ เพราะมีขีดจำกัดในความสามารถแก้ไขบิตผิดพลาดได้เพียงหนึ่งบิต

<u>กรณีที่ 3</u>: ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 1 แสดงในรูปที่ 4.19



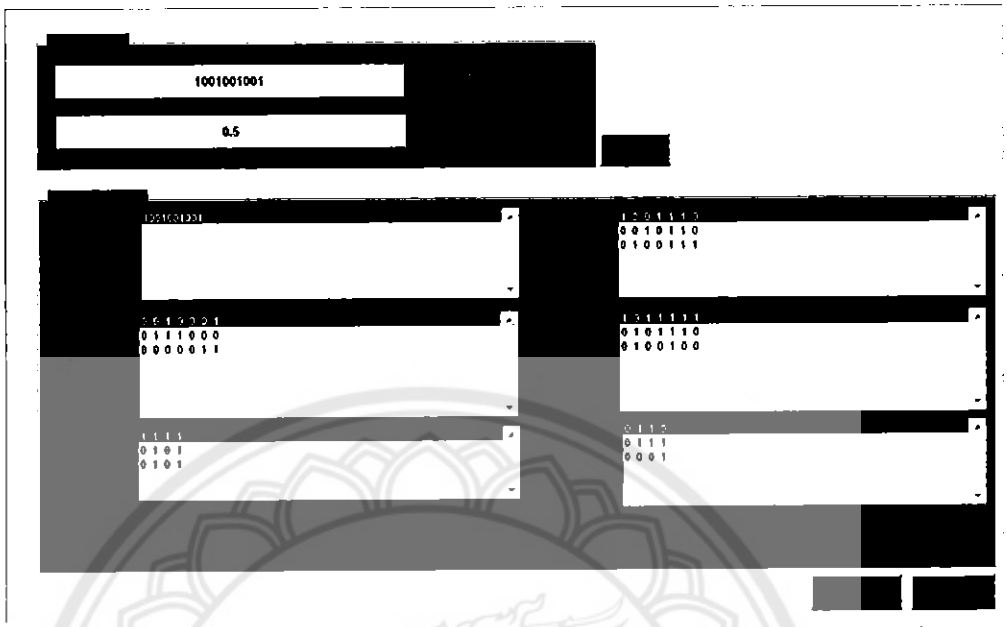
รูปที่ 4.19 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 1

ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2 แสดงในรูปที่ 4.19



รูปที่ 4.20 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 2

ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3 แสดงในรูปที่ 4.21



รูปที่ 4.21 แสดงตัวอย่าง การใส่ Data Bits 10 บิต ความน่าจะเป็นที่ 0.5 กดรันครั้งที่ 3

จากกรณีที่ 3 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการส่ง(Data Bits) เท่ากับ 10 บิต และความน่าจะเป็นในการเกิดข้อผิดพลาด(Probability of bit error) เท่ากับ 0.5 เมื่อกดปุ่มรัน ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างรูปที่ 4.19 รูปที่ 4.20 และรูปที่ 4.21 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 10 บิต ข้อมูลที่ทำการส่งที่ผ่านโปรแกรมจะมีจำนวน 12 บิต เนื่องจากการส่งข้อมูลครั้งละ 4 บิตในหนึ่งบล็อก ในกรณีนี้จะส่งทั้งหมดสามบล็อก ทำให้มีบิตในการส่งทั้งหมด 12 บิต โดยที่บิตที่เกินมาจากข้อมูลที่ส่งนั้น จะถูกกำหนดเพิ่มให้มีค่าเป็นบิตศูนย์ ข้อมูลที่ได้นี้จะนำไปเข้ารหัส โดยจากมีพาริตี้บิตเพิ่มขึ้น ดังแสดงในช่อง BCH Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดหรือ Noise ซึ่งการรันทั้ง 3 ครั้ง ความน่าจะเป็นที่ 0.5 ค่า Noise จะถูก Random แตกต่างกันออกไป เมื่อมีบิตผิดพลาดเกิดขึ้น โปรแกรมจะทำการรวมข้อมูลที่เข้ารหัสกับข้อมูลที่ผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดความผิดพลาดระหว่างการส่งข้อมูล จากนั้นเข้าสู่ขั้นตอนการถอดรหัสด้วยการถอดรหัสแบบ Hard-decision ในช่อง Decoder of Hard-decision และมีตัวเปรียบเทียบข้อมูลระหว่างข้อมูลที่ส่งกับข้อมูลที่ถูกถอดรหัสว่ามีความผิดพลาดเกิดขึ้นที่บิตใดในช่อง Bit Error of Hard-decision เมื่อเปรียบเทียบข้อมูลที่ภาคส่งและภาครับ พบว่าโปรแกรมไม่สามารถแก้ไขบิตที่ผิดพลาดได้ เพราะมีขีดจำกัดในความสามารถแก้ไขบิตผิดพลาดได้เพียงหนึ่งบิต

4.2 โปรแกรมแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

4.2.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1.ในขั้นตอนแรก เมื่อทำการเปิดหน้าต่างการเลือกชนิดของการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision ดังรูปที่ 4.1

2.เมื่อทำการคลิกที่ช่องสัญญาณแล้ว จะทำให้หน้าต่าง Application ปรากฏขึ้นมา ซึ่งหน้าต่างนี้จะมีปุ่มให้เลือกสองปุ่ม ได้แก่

2.1 Detecting and Correcting Error คือ หน้าต่างแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด

2.2 BER คือ หน้าต่างแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

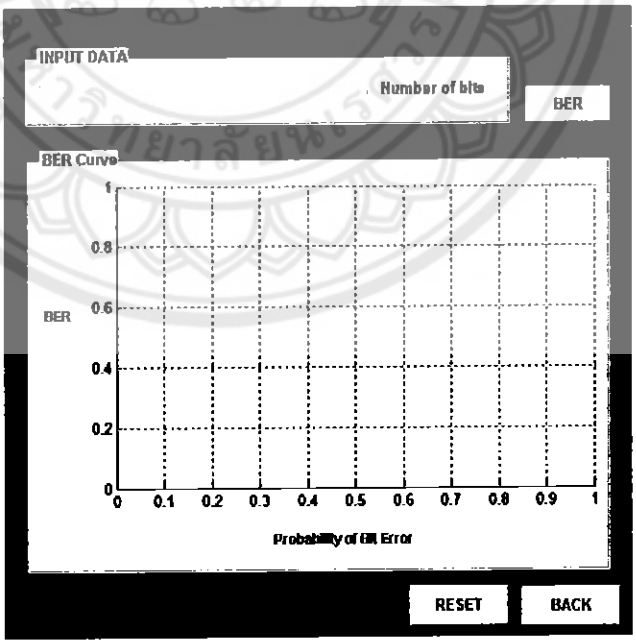
ซึ่งหน้าต่าง Application แสดงไว้ในรูปที่ 4.2

3.เมื่อทำการเปิดหน้าต่าง BER จะปรากฏหน้าต่างดังรูปที่ 4.22 ซึ่งรายละเอียดต่างๆในการใส่ค่าต่างๆมีดังนี้

3.1 ช่อง Number of Bits คือช่องใส่จำนวนบิตที่ต้องการส่ง

3.2 ปุ่ม RUN จะนำค่าบิตที่ต้องการส่งเข้าสู่โปรแกรมการคำนวณ และแสดงผลซึ่งแสดงดังรูปที่ 4.5

3.3 ส่วนแสดงเส้นกราฟระดับการเกิดข้อมูลผิดพลาด



รูปที่ 4.22 หน้าต่างแสดงการหน้าต่างที่ใช้แสดงระดับอัตราการเกิดข้อมูลผิดพลาด

4.2.2 ตัวอย่างแสดงระดับอัตราการเกิดข้อมูลผิดพลาด

1.คลิกเลือกชนิดของการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision ดังรูปที่ 4.1

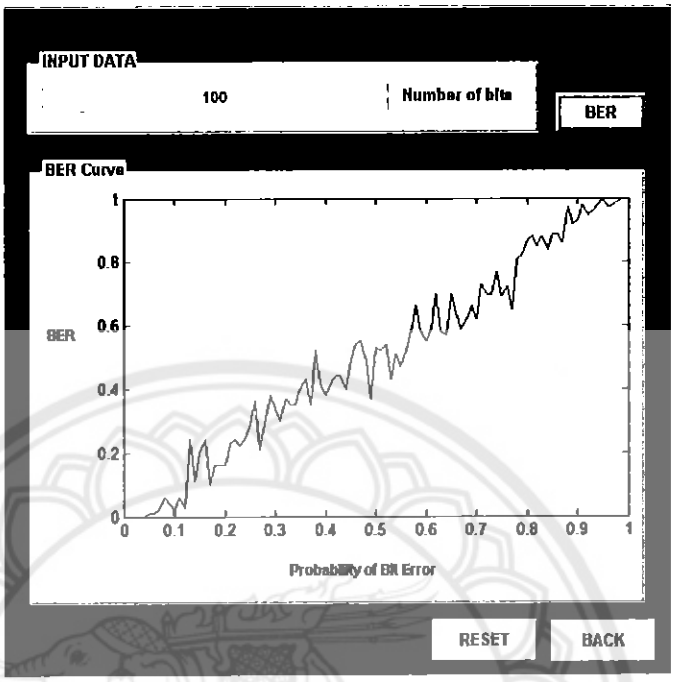
2.เมื่อทำการคลิกที่ช่องสัญญาณแล้ว จะทำให้หน้าต่าง Application ปรากฏขึ้นมาให้เลือก BER

3. ใส่ค่าต่างๆมีดังนี้

3.1 ช่อง Number of Bits คือช่องใส่จำนวนบิตที่ต้องการส่ง

3.2 กดปุ่ม BER จะนำค่าบิตที่ต้องการส่งเข้าสู่โปรแกรมการคำนวณ และแสดงผลซึ่งผลลัพธ์ของการรันโปรแกรมแสดงในรูปต่างๆดังนี้

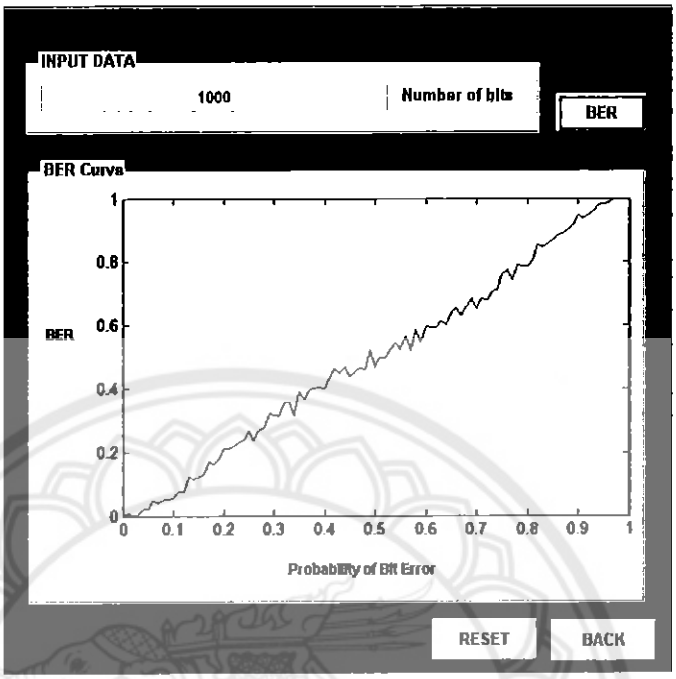
4.2.2.1 ระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100



รูปที่ 4.23 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100

จากรูปที่ 4.23 การเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision เมื่อทำการส่งข้อมูลจำนวน 100 บิต และกำหนดความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) ในช่วง 0 ถึง 1 แล้วจากกราฟจะเห็นได้ว่าค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่ผ่านโปรแกรมการเข้ารหัสเพื่อแก้ไขข้อมูลที่ผิดพลาด มีค่าลดลงจนเหลือศูนย์ในช่วงเริ่มต้น แต่เมื่อค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมีค่ามากขึ้น ก็ทำให้ BER เพิ่มขึ้นตามไปด้วย ซึ่งหมายความว่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมาก การเข้ารหัสสัญญาณก็ไม่สามารถแก้ไขข้อมูลผิดได้มากเท่านั้น และแสดงให้เห็นการว่าใส่ค่าจำนวนบิตที่น้อย ส่งผลให้ค่าความเบี่ยงเบนกราฟที่แสดงออกมามีค่าสูง

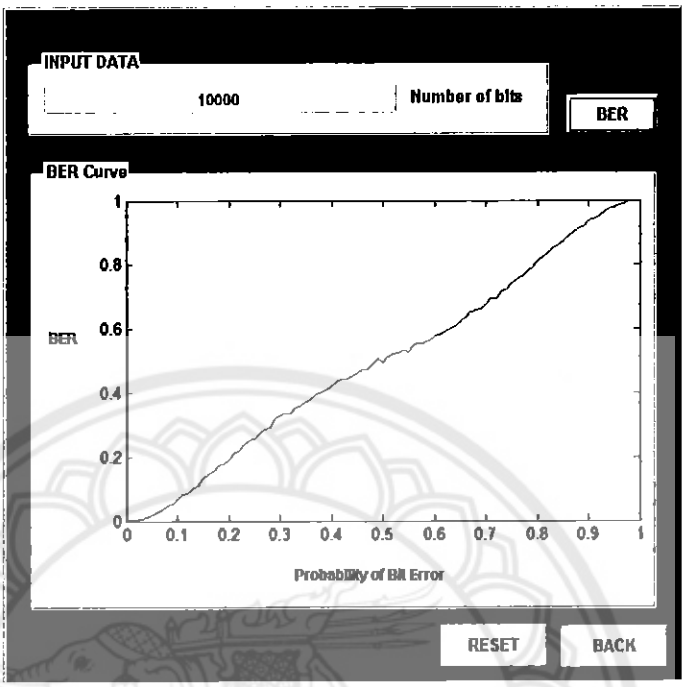
4.2.2.2 ระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 1000



รูปที่ 4.24 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 1000

จากรูปที่ 4.24 จะเห็นว่าการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision เมื่อทำการส่งข้อมูลจำนวน 1000 บิต และกำหนดความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) ในช่วง 0 ถึง 1 แล้วจากกราฟจะเห็นได้ว่า ค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่ผ่านโปรแกรมการเข้ารหัสเพื่อแก้ไขข้อมูลที่ผิดพลาด มีค่าลดลงจนเหลือศูนย์ในช่วงเริ่มต้น แต่เมื่อค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมีค่ามากขึ้น ก็ทำให้ BER เพิ่มขึ้นตามไปด้วย การใส่ค่าจำนวนบิตที่ 1000 บิต สามารถทำให้มองให้เห็นได้ว่ากราฟที่ได้เริ่มเป็นเส้นตรง ซึ่งหมายความว่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมาก การเข้ารหัสสัญญาณก็ไม่สามารถแก้ไขข้อมูลผิดได้มากเท่านั้น และจะเห็นได้ว่าจากรูปที่ 4.24 ค่าความเบี่ยงเบนกราฟลดลง เนื่องจากจำนวนบิตที่อินพุตที่มากขึ้น

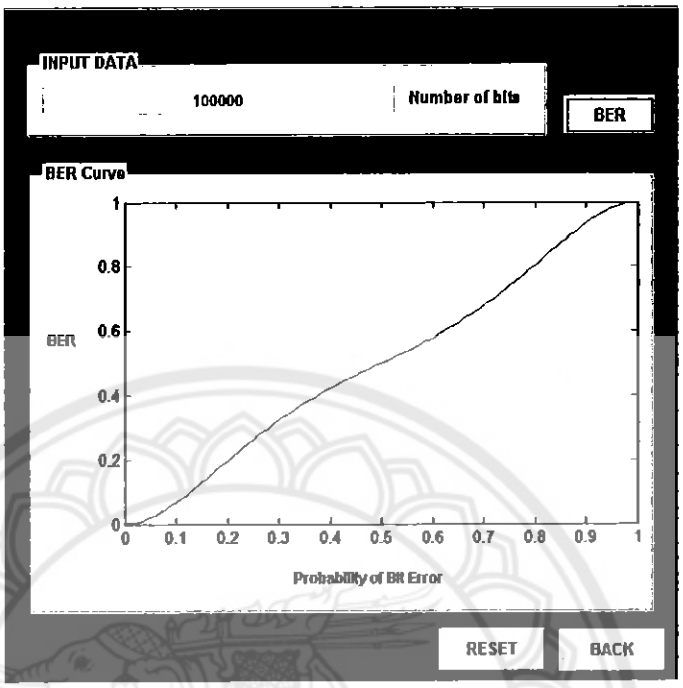
4.2.2.3 ระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 10000



รูปที่ 4.25 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 10000

จากรูปที่ 4.25 จะเห็นว่าการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision เมื่อทำการส่งข้อมูลจำนวน 10000 บิต และกำหนดความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) ในช่วง 0 ถึง 1 แล้วจากกราฟจะเห็นได้ว่า ค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่ผ่านโปรแกรมการเข้ารหัสเพื่อแก้ไขข้อมูลที่ผิดพลาด มีค่าลดลงจนเหลือศูนย์ในช่วงเริ่มต้น แต่เมื่อค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมีค่ามากขึ้น ก็ทำให้ BER เพิ่มขึ้นตามไปด้วย การใส่ค่าจำนวนบิตที่ 10000 บิต สามารถทำให้มองให้เห็นได้ว่ากราฟที่ได้เกือบเป็นเส้นตรง ซึ่งหมายความว่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมาก การเข้ารหัสสัญญาณก็ไม่สามารถแก้ไขข้อมูลผิดได้มากเท่านั้น และจากรูปที่ 4.25 แสดงให้เห็นว่าค่าความเบี่ยงเบนของกราฟลดลงกว่ารูปที่ 4.24 เนื่องจากจำนวนบิตที่อินพุตที่มากขึ้น

4.2.2.4 ระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100000



รูปที่ 4.26 ตัวอย่างระดับอัตราการเกิดข้อมูลผิดพลาดที่จำนวนบิต 100000

จากรูปที่ 4.26 จะเห็นว่าการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision เมื่อทำการส่งข้อมูลจำนวน 100000 บิต และกำหนดความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) ในช่วง 0 ถึง 1 แล้วจากกราฟจะเห็นได้ว่า เมื่อค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดในช่วงเริ่มมีค่าใกล้ 0 มากเท่าไร ค่า BER ก็ใกล้ 0 มากนั้น แต่เมื่อค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมีค่าสูงขึ้น BER ก็สูงตามขึ้นไปด้วย ซึ่งการแก้ไขข้อผิดพลาดด้วยวิธีนี้มีขีดจำกัด คือแก้ไขบิตผิดพลาดทำได้เพียงบิตเดียว และจากรูปที่ 4.26 แสดงให้เห็นว่าค่าความเบี่ยงเบนลดลงเกือบเป็นเส้นตรง เนื่องจากจำนวนบิตที่อินพุตที่มากขึ้น ซึ่งสามารถสรุปได้ว่าจำนวนบิตอินพุตแปรผกผันกับความเบี่ยงเบนของกราฟ หรืออาจกล่าวอีกในหนึ่งได้ว่า เมื่อใส่จำนวนบิตทดสอบที่มาก ผลการทดลองจะมีความเที่ยงตรงมากขึ้น

ในบทที่ 4 นี้ ได้แสดง Graphic User Interfaces ที่ใช้แสดงผลการเข้ารหัสสัญญาณ BCH Codes แบบอัตราการเข้ารหัสช่องสัญญาณ (7,4) และการถอดรหัสแบบ Hard-decision แสดงการแก้ไขข้อมูลที่ผิดพลาด อีกทั้งยังแสดงกราฟระดับอัตราการเกิดข้อมูลผิดพลาดของรหัสที่ทดสอบ

ในบทต่อไปจะเป็นการสรุปผลการดำเนินโครงงาน ปัญหาที่พบขณะดำเนินงาน และข้อเสนอแนะ

บทที่ 5
สรุปผลการดำเนินโครงงาน

5.1 ผลการดำเนินโครงงาน

โครงงานนี้เป็นการจำลองระบบสื่อสาร โดยการใช้การเข้ารหัสช่องสัญญาณเพื่อแก้ไขข้อมูลที่ผิดพลาด การเข้ารหัสช่องสัญญาณที่ใช้ในการดำเนินโครงงานนั้น คือ วิธีการเข้ารหัสสัญญาณ BCH Codes และการถอดรหัสแบบ Hard-decision ซึ่งใช้โปรแกรม MATLAB ในการดำเนินโครงงาน และแสดงออกมาในรูปของ Graphic User Interfaces โดยให้มีอัตราการเข้ารหัสช่องสัญญาณ BCH Codes (7,4) ซึ่งการเข้ารหัสด้วย BCH Codes (7,4) นี้สามารถแก้ไขผิดพลาดได้เพียงหนึ่งบิตเท่านั้น

โครงงานนี้ได้แสดงการแก้ไขข้อมูลที่ผิดพลาด และยังแสดงระดับอัตราการเกิดข้อมูลผิดพลาดในอัตราการเข้ารหัสช่องสัญญาณ โดยในการส่งข้อมูลผู้ใช้สามารถเลือกโครงงานนี้เพื่อเป็นตัวแปรในการเลือกใช้ว่าการเข้ารหัสนี้เหมาะสมกับการส่งข้อมูลนั้นมากน้อยเพียงใด เพื่อที่จะนำไปใช้ในระบบการสื่อสารจริง ส่งผลให้การส่งข้อมูลมีประสิทธิภาพสูงสุด

5.2 ปัญหาที่พบขณะดำเนินโครงงาน

เนื่องจากคำสั่งบางคำสั่งในโปรแกรม MATLAB ผู้ดำเนินโครงงานยังไม่มีความเข้าใจส่งผลให้เกิดความล่าช้าในขณะดำเนินโครงงาน

5.3 ข้อเสนอแนะ

1.ควรขอคำแนะนำในการดำเนินโครงงานจากอาจารย์ที่ปรึกษาโครงงาน เพื่อให้ได้งานที่มีคุณภาพ และเสร็จทันเวลาที่กำหนด

2.ในโปรแกรม MATLAB เมื่อมีปัญหาหรือคำสั่งที่ไม่เข้าใจ สามารถใช้ Help ช่วยในการเขียนโปรแกรมได้โดยจะมีคำอธิบายวิธีการเรียกใช้ วิธีการใช้งาน หรือเขียนคำสั่งต่างๆ

3.โครงงานนี้สามารถนำไปศึกษาเพื่อประกอบการเรียนหรือเป็นสื่อการเรียนการสอนสำหรับผู้ที่สนใจ

เอกสารอ้างอิง

[1] L.HANZO, T.H. LIEW, B.L. YEAP. **"Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over fading Channels".** pp.75-95.

[2] CHRISTIAN B. SCHLEGEL, LANCE C. PEREZ. **"Trellis and Turbo Coding".**

[3] A. Bruce Carlson, Paul B. Crilly, Janet C. Rutledge. **"Communication Systems".** 4 th edition. pp. 585-591.

[4] รศ.ดร. มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. **"คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์".** พิมพ์ครั้งที่ 1: สำนักพิมพ์อินโฟเพรส. 2543.

[5] DSL 102-M-6. **"BCH and RS Codes"**[Online]. from http://www.kmitl.ac.th/dslabs/Viterbi.