การจำลองการเข้ารหัสช่องสัญญาณโดยใช้วิธีการเข้ารหัสแบบคอนโวลูชัน และการถอดรหัสแบบไวเทอร์บี

Simulation of Channel Encoding using Convolutional Code with Viterbi Algorithm

นายพนมกร ทองพัฒนกุล รหัส 46363313

i508 1324 ค.1
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 17 พ.ย. 2550
เลขทะเบียน 5000116
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร
ปร.
ฟ187ก.
2549.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2549

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**การจำลองการเข้ารหัสช่องสัญญาณโดยใช้วิธีการเข้ารหัสแบบคอนโวลูชัน และการถอดรหัสแบบไวเทอร์บี**
ผู้ดำเนินโครงงาน	**นายพนมกร ทองพัฒนกุล รหัส 46363313**
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา
ภาควิชา	วิศวกรรมไฟฟ้า และคอมพิวเตอร์
ปีการศึกษา	2549

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ
(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ
(ดร.สมยศ เกียรติวนิชวิไล)

หัวข้อโครงงาน	การจำลองการเข้ารหัสช่องสัญญาณโดยใช้วิธีการเข้ารหัสแบบคอนโวลูชัน และการถอดรหัสแบบไวเทอร์บี
ผู้ดำเนินโครงงาน	นายพนมกร ทองพัฒนกุล รหัส 46363313
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา
ภาควิชา	วิศวกรรมไฟฟ้า และคอมพิวเตอร์
ปีการศึกษา	2549

.....

บทคัดย่อ

การเข้ารหัสเพื่อแก้ไขความผิดพลาดของข้อมูลดิจิตอลระหว่างการส่งนั้น เป็นสิ่งที่มีความสำคัญอย่างยิ่งในระบบการสื่อสารข้อมูล เนื่องจากในระหว่างเส้นทางการส่งข้อมูล มักมีสัญญาณรบกวน ซึ่งจะทำให้ข้อมูลที่ได้รับมีความผิดเพี้ยนเกิดขึ้น ดังนั้นเพื่อลดความผิดเพี้ยนของข้อมูลจึงมีความจำเป็นที่ต้องทำการเข้ารหัสเพื่อแก้ไขข้อมูลให้มีความถูกต้อง

โครงงานนี้เป็นการสร้างแบบจำลองการเข้ารหัสช่องสัญญาณ ด้วยโปรแกรม MATLAB ซึ่งแสดงอยู่ในรูปแบบของ Graphic User Interfaces โดยใช้การเข้ารหัสแบบ Convolutional Code และการถอดรหัสแบบ Viterbi Decoder อีกทั้งยังเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาดในแต่ละอัตราการเข้ารหัสช่องสัญญาณแบบต่าง ๆ

จากผลการดำเนินโครงงานพบว่า สามารถสร้างโปรแกรมจำลองการเข้ารหัสและการถอดรหัส อีกทั้งยังสามารถแสดงการแก้ไขข้อมูลผิดพลาด และเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาดในแต่ละอัตราการเข้ารหัสช่องสัญญาณแบบต่าง ๆ ซึ่งจะเห็นได้ว่าการเข้ารหัสช่องสัญญาณที่มีอัตราการเข้ารหัสที่มาก จะสามารถแก้ไขข้อมูลผิดพลาดได้ดีที่สุด

Project Title	Simulation of Channel Encoding using Convolutional Code with Viterbi Algorithm.
Name	Mr. Panomkorn Thongpatthanakun. ID. 46363313.
Project Advisor	Assistant Professor Surachet Kanprachar , Ph.D.
Major	Electrical Engineering.
Departmant	Electrical and Computer Engineering.
Academic Year	2006

.....

ABSTRACT

This project is about creating Channel Coding model by MATLAB which is presented in a form of Graphic User Interfaces and encodes with Convolutional Encoder and decodes with Viterbi Decoder. Moreover, there is a comparison of the effectiveness in error correction of each encoding.

The result shows that the program of encoding and decoding model has been completed. Then the error correction has been shown and compared with the effectiveness for each encoding rate. It can be seen that encoding at the high rate can improve error correction.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี ด้วยการช่วยเหลือและสนับสนุนจากหลายๆท่านด้วยกัน ซึ่งผู้เขียนขอขอบพระคุณดังต่อไปนี้

ขอขอบคุณ ผศ.ดร.สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาโครงงาน ในการให้ความรู้ คำแนะนำและคำปรึกษาที่เป็นประโยชน์อย่างสูงในการทำโครงงานนี้ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์ที่ดีเยี่ยมและขอกราบขอบพระคุณอย่างสูง

ขอขอบคุณ ดร.ชัยรัตน์ พินทอง และ ดร.สมยศ เกียรติวนิชวิไล ที่ให้คำปรึกษาในเรื่องของการจัดทำรายงาน ตลอดจนเสียสละเวลาอันมีค่า ตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ในรายงานฉบับนี้

ขอขอบคุณ ผศ.ดร.ธนิต มาลากร ที่ให้คำปรึกษาในการดำเนินโครงงานทุกขั้นตอน

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ประสิทธิ์ประสาทวิชาความรู้ และอบรมสั่งสอนให้ผู้จัดทำเป็นคนที่ดีของสังคม

ขอขอบคุณเพื่อนทุกคนที่คอยให้ความช่วยเหลือ ให้กำลังใจ และให้คำปรึกษาในการทำโครงงานนี้จนสำเร็จลุล่วงได้เป็นอย่างดี

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิด และส่งเสริมให้ผู้จัดทำเดินทางมาถึง ณ จุดนี้ ได้อย่างราบรื่น

คุณค่า และประโยชน์อันพึงมีจากโครงงานนี้ ทางผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

ผู้จัดทำโครงงาน
พนมกร ทองพัฒนกุล

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูปภาพ ..... ซ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการดำเนินโครงงาน 3

1.5 ผลที่คาดว่าจะได้รับ 4

1.6 งบประมาณที่ต้องใช้ 4

บทที่ 2 หลักการ และทฤษฎีของระบบสื่อสาร

2.1 หลักการพื้นฐานของระบบสื่อสาร 5

2.2 หลักการพื้นฐานของการเข้ารหัสช่องสัญญาณ 7

2.2.1 FEC (Forward Error Correction) 7

2.2.2 ARQ (Automatic Repeat Request) 7

2.3 รหัสช่องสัญญาณ 7

2.3.1 การเข้ารหัสช่องสัญญาณแบบบล็อก (Block Codes) 8

2.3.2 การเข้ารหัสช่องสัญญาณแบบคอนโวลูชัน (Convolutional Codes) 8

2.4 รหัสคอนโวลูชัน 9

2.4.1 การเข้ารหัสคอนโวลูชัน 10

2.4.2 วิธีการทำงานของวงจรเข้ารหัสคอนโวลูชัน 14

สารบัญ (ต่อ)

หน้า

2.4.3 รูปแบบการถอดรหัสแก้ไขความผิดพลาดแบบคอนโวลูชัน 19

2.4.4 วิธีการทำงานของวงจรถอดรหัสแบบ Viterbi 20

2.4.5 การถอดรหัสด้วย Viterbi Algorithm 21

2.5 การถอดรหัส Viterbi โดยใช้แผนภาพเทรลลิส 22

2.5.1 แผนภาพการเปลี่ยนสถานะ 24

2.5.2 แผนภาพเทรลลิส 25

2.5.3 การถอดรหัสคอนโวลูชัน 26

บทที่ 3 การออกแบบโครงงาน และวิธีการดำเนินงาน

3.1 ขั้นตอนการออกแบบโปรแกรม 32

3.1.1 สร้างสัญญาณ 32

3.1.2 เข้ารหัสสัญญาณ 32

3.1.3 สร้างสัญญาณ 32

3.1.4 รวมสัญญาณที่เข้ารหัสกับสัญญาณรบกวน 32

3.1.5 การแก้ไขข้อมูลที่ผิดพลาดตรงปลายทาง 33

3.2 การออกแบบ Graphic User Interfaces และขั้นตอนการดำเนินงาน 34

3.2.1 Type of Convolutional Code 34

3.2.2 Application 36

3.2.3 Display Convolutional Code 37

3.2.4 BER 39

3.2.5 Compare of BER 41

บทที่ 4 ผลการดำเนินโครงงาน

4.1 โปรแกรมแสดงการแก้ไขข้อมูลผิดพลาด 44

4.1.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม 44

4.1.2 ตัวอย่างการแก้ไขข้อมูลผิดพลาด 47

สารบัญ (ต่อ)

หน้า

4.2 โปรแกรมแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด 54

4.2.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม 54

4.2.2 ตัวอย่างการลดระดับการเกิดข้อมูลผิดพลาด 56

4.3 โปรแกรมแสดงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ 63

4.3.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม 63

4.3.2 ตัวอย่างการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ 64

บทที่ 5 สรุปผลการดำเนินการ

5.1 ผลการดำเนินโครงงาน 68

5.2 ปัญหาที่พบขณะดำเนินโครงงาน 68

5.3 ข้อเสนอแนะ 69

เอกสารอ้างอิง 70

ประวัติผู้ดำเนินโครงงาน 71

สารบัญตาราง

ตารางที่ หน้า

2.1 การกำหนดค่าสำหรับแต่ละสถานะ (State) 15

2.2 รายละเอียดการเปลี่ยนสถานะ (State) 16

2.3 รายละเอียดการเปลี่ยนสถานะของการถอดรหัสแบบ Viterbi 25

2.4 การหาระยะแฮมมิ่งระหว่างชุดบิตที่รับได้ 101000001
เฉพาะ 9 บิตแรกกับรหัสทั้ง 8เส้นทาง 27

2.5 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 10 – 12 ซึ่งมีค่าเท่ากับ 011 27

2.6 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 13 – 15 ซึ่งมีค่าเท่ากับ 101 28

2.7 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 16 – 18 ซึ่งมีค่าเท่ากับ 011 29

2.8 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 19 – 21 ซึ่งมีค่าเท่ากับ 101 29

2.9 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 22 – 24 ซึ่งมีค่าเท่ากับ 100 30

4.1 แสดงการเปรียบเทียบค่า Pe,new ของอัตราการเข้ารหัสทั้ง 7 แบบ
โดยที่พิจารณา ณ ค่า Pe,set = 0.05 66

สารบัญภาพ

รูปที่ หน้า

2.1 ต้นแบบของระบบสื่อสาร 6

2.2 ประเภทของรหัสควบคุมความผิดพลาด 9

2.3 วงจรเข้ารหัสคอนโวลูชันที่ค่า R เท่ากับ 1/2 และค่า K เท่ากับ 3 10

2.4 Convolutionally encoding a message sequence with rate 1/2, K = 3 encoder 13

2.5 วงจรเข้ารหัสจากรูปที่ 2.3 เพื่อใช้ในการกำหนดค่าสถานะ (State) 15

2.6 แผนภาพสถานะ (State Diagram) ของวงจรเข้ารหัสที่ rate=1/2, K=3 16

2.7 แผนภาพต้นไม้ (Tree diagram) ของภาคเข้ารหัสที่ rate=1/2, K=3 18

2.8 แผนภาพเทรลลิส (Trellis diagram) ของวงจรเข้ารหัสที่ rate=1/2, K=3 19

2.9 การตัดสินใจแบบ Hard-Decision 20

2.10 ตัวอย่างวงจรเข้ารหัสคอนโวลูชัน 22

2.11 ตัวอย่างขั้นตอนการเข้ารหัสของข้อมูล 110101 โดยใช้วงจรเข้ารหัสในรูปที่ 2.10 24

2.12 แผนภาพการเปลี่ยนสถานะของวงจรเข้ารหัสคอนโวลูชัน
ตามโครงสร้างในรูปที่ 2.11 25

2.13 โครงสร้างแผนภาพเทรลลิสสำหรับวงจรเข้ารหัสคอนโวลูชันของรูปที่ 2.12 26

3.1 Flowchart การทำงานของโปรแกรม 33

3.2 Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสช่องสัญญาณ 35

3.3 Graphic User Interfaces ในการเลือกแสดงตัวอย่าง หรือคำนวณ Bit Error Rate 36

3.4 Graphic User Interfaces ในการแสดงการแก้ไขบิตผิดพลาด 38

3.5 Graphic User Interfaces ในการเปรียบเทียบ Bit Error Rate 40

3.6 Graphic User Interfacesในการเปรียบเทียบ Bit Error Rate
ของอัตราการเข้ารหัสแบบต่าง ๆ 42

4.1 หน้าต่างสำหรับเลือกอัตราการเข้ารหัสช่องสัญญาณชนิดต่าง ๆ 45

4.2 หน้าต่าง Application ใช้ในการเลือกระหว่างหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด
และหน้าต่างแสดงการลดอัตราการเกิดข้อมูลผิดพลาด 46

4.3 หน้าต่างแสดงการแก้ไขบิตผิดพลาด 46

สารบัญภาพ (ต่อ)

รูปที่ หน้า

4.4 ตัวอย่างในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัส 1/2 47

4.5 กราฟแสดงบิตข้อมูลอัตราการเข้ารหัส 1/2 49

4.6 ตัวอย่างในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัส 1/7 51

4.7 กราฟแสดงบิตข้อมูลอัตราการเข้ารหัส 1/7 53

4.8 หน้าต่างแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด 55

4.9 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/2 56

4.10 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/3 57

4.11 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/5 58

4.12 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/7 59

4.13 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/3 60

4.14 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/5 61

4.15 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/7 62

4.16 หน้าต่างแสดงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของแต่ละอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 อัตรา 63

4.17 แสดงตัวอย่างการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของแต่ละอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 อัตรา 64

4.18 ภาพขยายของการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของแต่ละอัตราการเข้ารหัส โดยใช้ตัวถอดรหัสแบบ Hard Decision ในรูปที่ 4.17 65

4.19 ภาพขยายในส่วนที่เป็นวงรี ในรูปที่ 4.18 65

4.20 การเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด
ของอัตราการเข้ารหัสทั้ง 7 แบบ โดยกำหนดค่า Pe,set 66

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบัน ระบบสื่อสาร (Communication System) มีบทบาทและความสำคัญต่อวิถีการดำรงชีวิตของมนุษย์ในแทบทุกด้าน ตั้งแต่ สังคม วัฒนธรรม การติดต่อธุรกิจ การส่งผ่านข่าวสาร รวมไปถึงกิจกรรมทางการเมืองของมนุษย์เป็นอย่างมาก

วัตถุประสงค์หลักของระบบสื่อสาร คือ การส่งข่าวสารจากแหล่งกำเนิดสัญญาณไปยังอุปกรณ์ภาครับให้ได้ในปริมาณที่มาก รวดเร็ว และถูกต้องที่สุด โดยผ่านทางช่องสัญญาณสื่อสารที่มีความผิดพลาด และสัญญาณรบกวนซึ่งส่งผลกระทบต่อคุณภาพของการรับส่งข่าวสาร ด้วยเหตุนี้ การศึกษาถึงระบบสื่อสาร จึงมักจะสนใจถึงสมรรถนะ และประสิทธิภาพของการรับส่งข่าวสารว่าจะดีมากน้อยเพียงใด

ในระบบสื่อสารนั้น สัญญาณรบกวนเป็นปัญหาสำคัญปัญหาหนึ่งที่ทำให้การรับส่งข้อมูลเกิดความผิดพลาด ซึ่งระบบสื่อสารเหล่านี้จำเป็นที่จะต้องได้รับการป้องกัน และแก้ไขปัญหาดังกล่าว หนึ่งในวิธีการป้องกัน และลดความผิดพลาดเหล่านี้ คือ การเข้ารหัสคอนโวลูชัน (Convolutional Code) และการถอดรหัสแบบ Viterbi Decoder ซึ่งเป็นวิธีการป้องกันความผิดพลาดล่วงหน้า (Forward Error Correction: FEC) การเข้ารหัสคอนโวลูชัน และการถอดรหัสแบบ Viterbi Decoder นี้ได้ถูกนำมาใช้งานในระบบสื่อสารแบบดิจิตอลในปัจจุบันอย่างแพร่หลาย อาทิ ระบบสื่อสารดาวเทียม ระบบโทรศัพท์เคลื่อนที่ และระบบ Wireless LAN เป็นต้น

ดังนั้น โครงงานนี้จึงจะนำเสนอการศึกษา และการสร้างแบบจำลองการเข้ารหัสช่องสัญญาณด้วยโปรแกรม MATLAB โดยทำการเข้ารหัสช่องสัญญาณ (Channel Coding) แบบคอนโวลูชัน (Convolutional Codes) และการถอดรหัสแบบ Viterbi Decoder อีกทั้งยังศึกษาความแตกต่าง และเปรียบเทียบประสิทธิภาพระหว่าง Soft Decision และ Hard Decision เพื่อที่จะทำการตัดสินใจเลือกใช้การถอดรหัสที่มีประสิทธิภาพสูงสุด และนำไปใช้ในการสื่อสารข้อมูลที่ต้องการความถูกต้องแม่นยำ อีกทั้งยังสามารถแก้ไขข้อผิดพลาดที่อุปกรณ์ภาครับได้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาประสิทธิภาพของการเข้ารหัสช่องสัญญาณ (Channel Coding) แบบคอนโวลูชัน (Convolutional Code)
2. เพื่อศึกษาระบบการทำงานการถอดรหัสแบบ Viterbi Decoder

3. เพื่อศึกษาความแตกต่าง และเปรียบเทียบประสิทธิภาพของ Soft Decision และ Hard Decision
4. เพื่อศึกษาความแตกต่าง และเปรียบเทียบประสิทธิภาพของอัตราการเข้ารหัสช่องสัญญาณแบบต่าง ๆ
5. เพื่อสร้างแบบจำลองของระบบสื่อสาร และสร้างโปรแกรมสำเร็จรูปในการเข้ารหัสแบบคอนโวลูชัน (Convolutional Code)
6. เพื่อศึกษาการเขียนโปรแกรมด้วย MATLAB
7. เพื่อนำความรู้ที่ได้ไปประยุกต์ใช้งานในระบบสื่อสาร

1.3 ขอบเขตของโครงงาน

1. สร้างโปรแกรมการเข้ารหัสของสัญญาณแบบคอนโวลูชัน (Convolutional Code) และการถอดรหัสแบบ Viterbi Decoder
2. เปรียบเทียบประสิทธิภาพของ Soft Decision และ Hard Decision
3. สามารถตัดสินใจเลือกการเข้ารหัสช่องสัญญาณที่มีประสิทธิภาพสูงสุด
4. สร้างโปรแกรมสำเร็จรูปในการเข้ารหัสแบบคอนโวลูชัน (Convolutional Code)

1.4 ขั้นตอนการดำเนินโครงงาน

กิจกรรมการดำเนินโครงงาน	2548		2549												2550				
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ศึกษาความรู้พื้นฐานเกี่ยวกับการเข้ารหัส และการถอด รหัส	←	→																	
2. ศึกษาการใช้งานโปรแกรม MATLAB เบื้องต้น			←	→															
3. ศึกษาความรู้และออกแบบโปรแกรม MATLABในการคำนวณการเข้ารหัสสัญญาณแบบ Convolutional Codes					←	→													
4. ออกแบบโปรแกรมการคำนวณการถอดรหัสแบบ Viterbi Decoder, Soft Decision, Hard Decision								←	→										
5. เปรียบเทียบประสิทธิภาพและความแตกต่างของการเข้ารหัสแบบต่าง ๆ											←	→							
6. จัดทำรูปเล่มโครงงาน															←	→			
7. ตรวจสอบ และปรับปรุงแก้ไขรูปเล่มโครงงาน																	←	→	
8. นำเสนอโครงงาน																			← →

1.5 ผลที่คาดว่าจะได้รับ

1. โปรแกรมคอมพิวเตอร์ที่สามารถคำนวณการเข้ารหัสช่องสัญญาณ และการถอดรหัสช่องสัญญาณได้
2. โปรแกรมคอมพิวเตอร์ที่สามารถเปรียบเทียบลักษณะของการเข้ารหัสช่องสัญญาณประเภท ต่าง ๆ ได้
3. สามารถเปรียบเทียบความแตกต่าง และประสิทธิภาพของการเข้ารหัสช่องสัญญาณได้
4. สามารถเลือกการเข้ารหัสช่องสัญญาณที่มีประสิทธิภาพสูงสุดได้
5. มีความรู้และทักษะในการเขียนโปรแกรมด้วย MATLAB

1.6 งบประมาณที่ต้องใช้

1.	ค่าเอกสารประกอบการทำโครงงาน	700 บาท
2.	ค่าแผ่นโปรแกรม	100 บาท
3.	ค่าจัดทำรูปเล่มโครงงาน	200 บาท
	รวม	1000 บาท

(หนึ่งพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

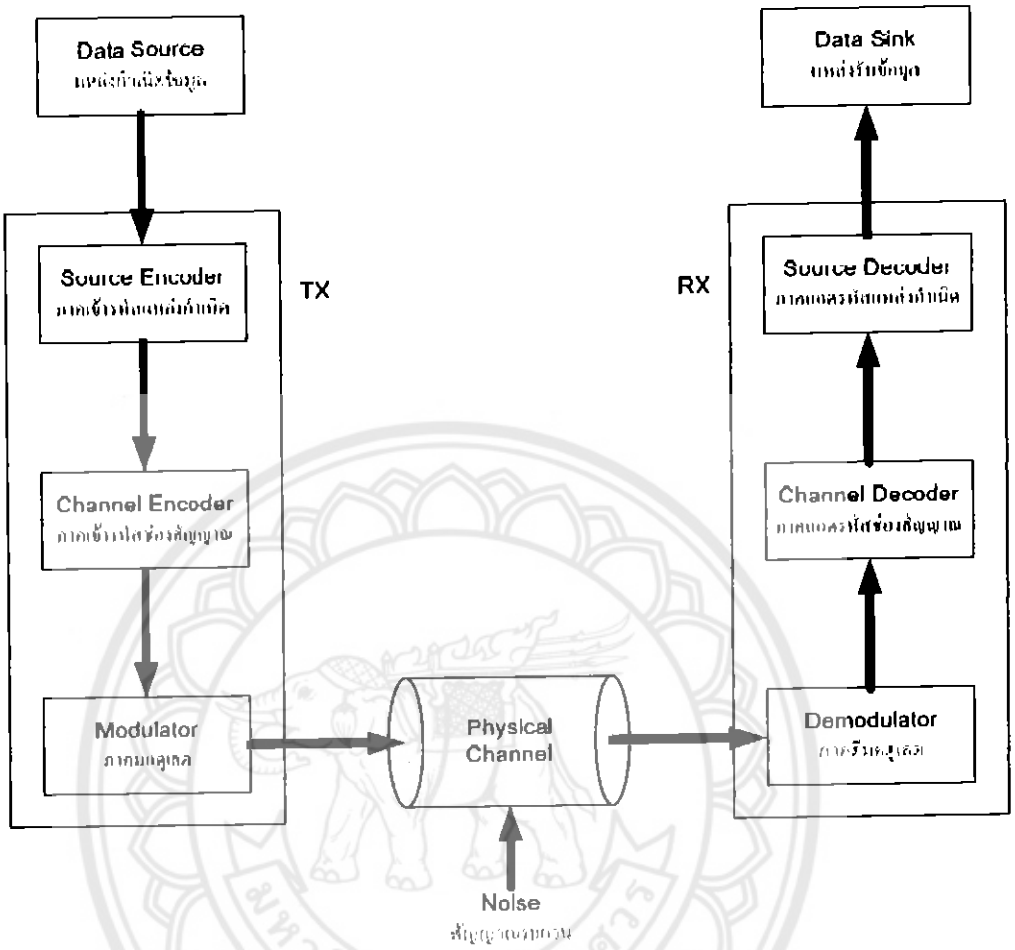
บทที่ 2
หลักการและทฤษฎีของระบบสื่อสาร

ในปัจจุบันการติดต่อสื่อสารระหว่างกันกลายเป็นสิ่งที่ขาดไม่ได้สำหรับสังคมมนุษย์ การสื่อสารในปัจจุบันนี้ถือได้ว่ามีความจำเป็นอย่างยิ่ง และสามารถพบเห็นได้โดยทั่วไปทั้งนี้เป็นผลเนื่องมาจากเทคโนโลยีโทรคมนาคมมีความก้าวหน้า และพัฒนาอย่างต่อเนื่องไม่หยุดยั้ง โดยเฉพาะอย่างยิ่งการพัฒนาทางด้านเทคโนโลยีการสื่อสาร ซึ่งมีประสิทธิภาพในการส่งข้อมูลเพิ่มขึ้น เช่น การรับส่งข้อมูลภาพ การรับส่งข้อมูลเสียง และการรับส่งข้อมูลภาพและเสียง เป็นต้น ซึ่งภายในบทนี้จะกล่าวถึงหลักการพื้นฐานของการสื่อสาร การเข้ารหัสช่องสัญญาณเพื่อแก้ไขข้อมูลผิดพลาดต่าง ๆ ที่ใช้กันในระบบสื่อสารปัจจุบัน

2.1 หลักการพื้นฐานของระบบสื่อสาร [1]

การสื่อสารข้อมูล คือการแลกเปลี่ยนข้อมูลกันระหว่างสองฝ่าย การสื่อสารข้อมูลโดยส่วนใหญ่ในปัจจุบันนั้นใช้ระบบดิจิตอล (Digital System) โดยมีรูปแบบแสดงดังรูป 2.1 ซึ่งเป็นระบบที่มีลักษณะของสัญญาณแบบไม่ต่อเนื่อง (Discrete Signal) ซึ่งมีข้อดีมากกว่าระบบอนาล็อก (Analog System) ที่มีลักษณะของสัญญาณที่ต่อเนื่องดังนี้

- ระบบดิจิตอลนั้นมีเสถียรภาพมากกว่าเมื่อเทียบกับอุปกรณ์ของระบบอนาล็อกที่สามารถเปลี่ยนแปลงไปตามสิ่งแวดล้อม เช่น อุณหภูมิ ความชื้น เป็นต้น
- ระบบดิจิตอลนั้นมีความยืดหยุ่นสูงกว่า สามารถแก้ไข ปรับปรุง และพัฒนาได้อย่างมีประสิทธิภาพมากกว่า สามารถพัฒนาสัญญาณ ตรวจสอบ และแก้ไขข้อผิดพลาด สามารถเข้ารหัสเพื่อรักษาความลับ สามารถบีบอัดข้อมูลโดยลดขนาดของข้อมูลที่เกินความจำเป็นด้วยอัลกอริทึม สามารถรองรับสัญญาณได้หลายประเภทไม่ว่าจะเป็นภาพ เสียง วีดีโอ ตัวอักษร ฯลฯ
- ระบบดิจิตอลนั้นจะส่งสัญญาณได้ระยะไกลกว่าระบบอนาล็อก



รูปที่ 2.1 ต้นแบบของระบบสื่อสาร [1]

การทำงานของภาคส่ง (TX: Transmitter) ในระบบสื่อสารดิจิตอลนั้น ข้อมูลข่าวสารที่ออกมาจากแหล่งกำเนิดข้อมูล (Data Source) จะถูกเข้ารหัสที่ภาคเข้ารหัสแหล่งกำเนิด (Source Encoder) เพื่อทำให้ข้อมูลนั้นมีรูปแบบที่เหมาะสมต่อการส่งข้อมูล ต่อจากนั้นข้อมูลก็จะถูกเข้ารหัสช่องสัญญาณที่ภาคเข้ารหัสช่องสัญญาณ (Channel Encoder) เพื่อควบคุมความผิดพลาดของข้อมูลระหว่างการส่ง และขั้นตอนสุดท้ายของการส่ง คือการมอดูเลตสัญญาณที่ภาคมอดูเลต (Modulator) เพื่อให้สัญญาณที่ต้องการส่งเหมาะสมกับช่องสัญญาณที่ใช้ก่อนที่จะส่งเข้าสู่ช่องสัญญาณ

ส่วนการทำงานของภาครับ (RX: Receiver) นั้นจะเป็นการทำงานย้อนกลับจากภาคส่งด้วยวิธีการที่สอดคล้องกันโดยเริ่มตั้งแต่การดีมอดูเลตที่ภาคดีมอดูเลเตอร์ (Demodulator) เพื่อให้ข้อมูลที่ได้กลับมาสู่ความถี่พื้นฐาน (Base band) จากนั้นจึงเข้าในภาคถอดรหัสช่องสัญญาณ (Channel Decoder) เพื่อ

ตรวจสอบ และแก้ไขข้อมูลที่เกิดความผิดพลาดให้ถูกต้อง และขั้นตอนสุดท้ายของการรับ คือการถอดรหัสแหล่งกำเนิด (Source Decoder) ข้อมูลก็จะส่งไปยังแหล่งรับข้อมูล (Data Sink)

2.2 หลักการพื้นฐานของการเข้ารหัสช่องสัญญาณ [2]

การส่งสัญญาณข้อมูลที่เป็นข้อมูลดิจิตอลในระบบสื่อสารนั้น มักจะเกิดการผิดเพี้ยนของรูปสัญญาณ อาจเป็นผลมาจากคุณสมบัติที่ไม่เป็นอุดมคติของช่องสัญญาณ หรือเป็นผลมาจากสัญญาณรบกวนในรูปแบบต่าง ๆ ซึ่งปัญหาเหล่านี้เองที่อาจจะส่งผลให้ข้อมูลที่ภาครับมีความผิดพลาดเกิดขึ้น ด้วยเหตุนี้ ในระบบสื่อสารปัจจุบันซึ่งมีความต้องการความถูกต้อง และความแน่นอนของข้อมูลสูง จึงได้มีการนำสัญญาณข้อมูลที่ต้องการส่งนั้น ผ่านกระบวนการเข้ารหัสช่องสัญญาณ (Channel Coding) ก่อนที่จะส่งสัญญาณข้อมูลออกไป เพื่อให้ข้อมูลที่ได้รับในภาครับมีความผิดพลาดน้อยลง และอยู่ในระดับที่สามารถยอมรับได้ ในการเข้ารหัสช่องสัญญาณนั้นจะต้องมีการเพิ่มจำนวนบิตที่จะส่งออกไป โดยบิตที่เพิ่มเข้ามาจะช่วยให้ภาครับสามารถที่จะตรวจจับบิตผิดพลาด (Error Detection) หรือถ้ามีการเพิ่มจำนวนที่เหมาะสมแล้วก็จะสามารถแก้ไขบิตที่ผิดพลาดได้ (Error Correction) ของข้อมูลได้ การเข้ารหัสช่องสัญญาณจะทำให้อัตราบิตของข้อมูลที่ต้องส่งจริงมีค่ามากขึ้น ซึ่งก็คือต้องการช่องสัญญาณที่ใช้ส่งมีแบนด์วิดท์ที่กว้างขึ้น แต่ถ้าหากมีช่องสัญญาณแบนด์วิดท์ที่แคบหรือจำกัด และต้องการที่จะให้ข้อมูลที่ความถูกต้อง ก็จะต้องลดอัตราการส่งบิตข้อมูลลง

การเข้ารหัสช่องสัญญาณแบ่งออกเป็น 2 รูปแบบ คือ

2.2.1 FEC (Forward Error Correction)

วิธีการ FEC นั้นภาครับจะสามารถตรวจจับว่ามีบิตผิดพลาดเกิดขึ้นในระหว่างการส่งสัญญาณหรือไม่ ถ้ามีก็จะต้องสามารถระบุได้ว่าบิตที่ผิดพลาดนั้นเกิดขึ้นที่ตำแหน่งใด ต่อจากนั้นภาครับจึงทำการแก้ไขบิตผิดพลาดดังกล่าว

2.2.2 ARQ (Automatic Repeat Request)

วิธีการ ARQ ภาครับจะตรวจจับว่ามีบิตผิดพลาดหรือไม่เท่านั้น ไม่สามารถที่จะแก้ไขบิตผิดพลาดได้ ถ้าหากว่าตรวจจับได้ว่ามีบิตผิดพลาดแล้วภาครับจะส่งสัญญาณร้องขอไปที่ภาคส่ง ให้ภาคส่งทำการส่งข้อมูลชุดเดิมกลับมา

2.3 รหัสช่องสัญญาณ [2]

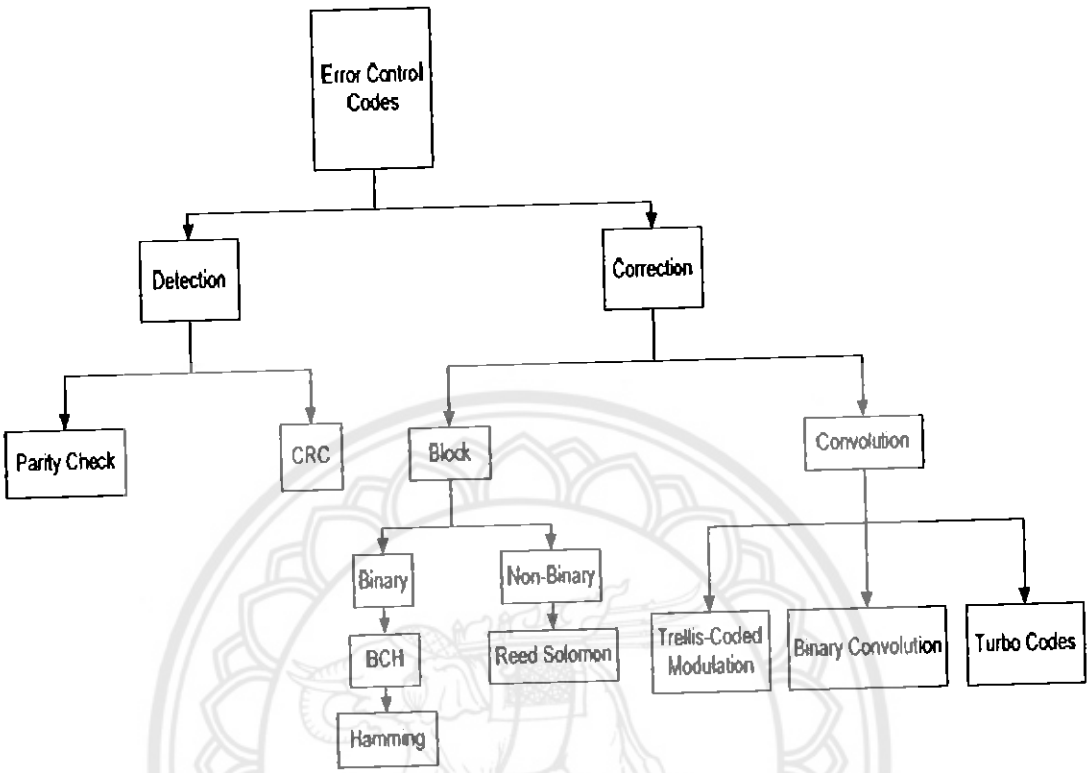
การเข้ารหัสช่องสัญญาณนั้นสามารถแบ่งออกได้เป็น 2 ประเภท คือ

2.3.1 การเข้ารหัสช่องสัญญาณแบบบล็อก (Block Codes)

การเข้ารหัสช่องสัญญาณแบบบล็อกนี้ จะทำการแบ่งบิตข้อมูลออกเป็นกลุ่ม ก่อนจะนำส่งเข้าสู่การเข้ารหัสช่องสัญญาณ ซึ่งเรียกว่าบล็อกมีขนาด k บิต จากนั้นจะทำการแปลงบิตข้อมูลในแต่ละบล็อกให้กลายเป็นคำรหัส (Codeword) ที่มีความยาวเท่ากับ n บิต โดยที่ $n > k$ อาจเรียกการเข้ารหัสนี้ว่า (n,k) ชุดของรหัสที่เข้ารหัสแล้วนั้นจะมีข้อมูลเดิม คือ k บิต และมีส่วนของข้อมูลพิเศษที่เพิ่มเข้ามาอีกจำนวนเท่ากับ $n - k$ บิต ซึ่งจะเรียกว่า Check Bit ในส่วนนี้จะใช้ในการตรวจสอบว่ามีความผิดพลาดในข้อมูลระหว่างการส่งผ่านช่องสัญญาณหรือไม่ ที่ภาครับก็จะมีวงจรในการทำหน้าที่ถอดรหัสของสัญญาณเพื่อดึงบิตข้อมูลเดิมออกมา

2.3.2 การเข้ารหัสช่องสัญญาณแบบคอนโวลูชัน (Convolutional Codes)

การเข้ารหัสช่องสัญญาณคอนโวลูชันนี้มีความแตกต่างกับการเข้ารหัสช่องสัญญาณแบบบล็อก คือ ข้อมูลที่จะเข้ารหัสช่องสัญญาณนั้นไม่ต้องนำมาแบ่งเป็นบล็อก การเข้ารหัสช่องสัญญาณแบบคอนโวลูชันนี้สามารถที่จะป้อนข้อมูลเข้าไปในวงจรเข้ารหัสได้เลย กระบวนการเข้ารหัสนี้ก็จะดำเนินต่อไปเรื่อย ๆ จนกว่าจะหยุดป้อนข้อมูล คุณสมบัติของการเข้ารหัสช่องสัญญาณแบบคอนโวลูชันจะแสดงอยู่ในรูปของอัตราส่วนการเข้ารหัสช่องสัญญาณ เช่น $1/n$ อย่างเช่น เมื่อเราป้อนข้อมูลจำนวน 1 บิตเข้าสู่วงจรเข้ารหัสช่องสัญญาณ ก็จะได้รหัสที่มีความยาวเพิ่มขึ้นเป็นจำนวน n เท่า อัตราส่วนในการเข้ารหัสจะมีค่าแตกต่างกันไปแล้วแต่ข้อมูล เช่น ข้อมูลเข้า 2 บิต และผลเป็นคำรหัสมีความยาว 3 บิต ดังนั้นจะได้อัตราส่วนการเข้ารหัสเท่ากับ 2/3



รูปที่ 2.2 ประเภทของรหัสควบคุมความผิดพลาด [1]

2.4 รหัสคอนโวลูชัน [2]

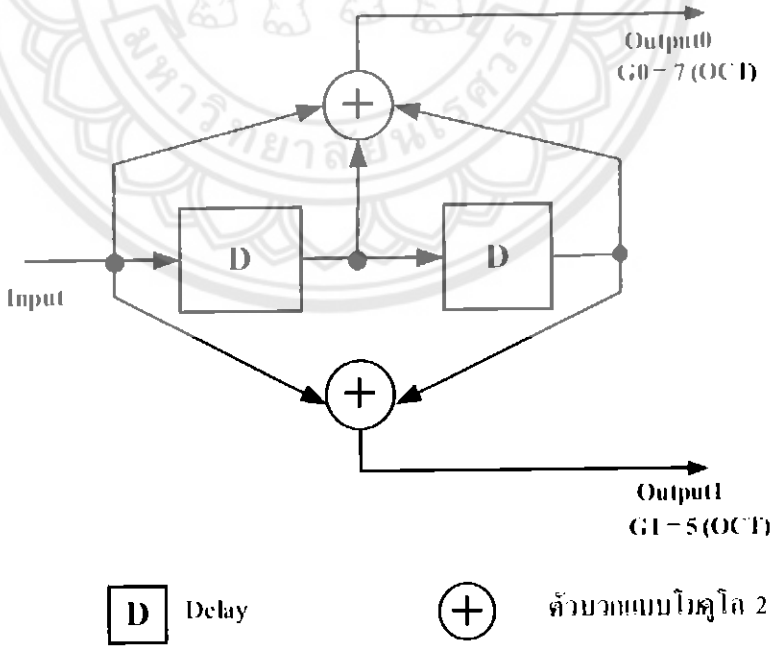
รหัสคอนโวลูชัน (Convolution Code) ได้รับการพัฒนาขึ้นครั้งแรกเมื่อปี ค.ศ. 1955 โดย P.Elias ในบทความชื่อ Coding for noisy channel ซึ่งได้รับการตีพิมพ์ใน IRE Conv. Record, Part 4, หน้า 37 -47 โครงสร้างพื้นฐานของรหัสคอนโวลูชันจัดว่ามีคุณลักษณะที่แตกต่างไปจากรหัสบล็อกที่ได้อธิบายไว้อย่างเห็นได้ชัด ในกรณีของการเข้ารหัสบล็อก บิตข้อมูลที่จะนำมาเข้ารหัสจะได้รับการแบ่งออกเป็นบล็อกที่มีขนาดตายตัวเท่ากันทุกบล็อกก่อน จากนั้นจึงนำบิตข้อมูลของแต่ละบล็อกไปผ่านกระบวนการเข้ารหัสเพื่อให้ได้เป็นคำรหัสที่มีความยาวมากขึ้น โดยความยาวของคำรหัสที่ได้จะมีขนาดคงที่ตายตัวเท่ากันหมดเช่นกัน ฉะนั้นการเข้ารหัสบล็อกจึงเปรียบได้กับการแปลงบล็อกของบิตข้อมูลขนาด k บิต ให้ได้เป็นคำรหัสที่มีความยาว n บิต นั่นเอง ในทางตรงกันข้าม รหัสคอนโวลูชันไม่จำเป็นต้องมีการแบ่งบิตข้อมูลออกเป็นบล็อกแต่อย่างใด เราสามารถป้อนชุดบิตข้อมูลขนาดความยาวเท่าใดก็ได้เข้าสู่วงจรเข้ารหัสคอนโวลูชัน และผลลัพธ์ที่ได้ทางด้านออกจะเป็นชุดคำรหัสที่มีความยาวเป็นจำนวนเท่าของชุดบิตข้อมูลที่ป้อนเข้า ความ

แตกต่างทางโครงสร้างพื้นฐานของรหัสทั้งสองประเภทนี้ทำให้การศึกษา และการวิเคราะห์ขีดความสามารถของรหัสจึงต้องอาศัยกรรมวิธีที่แตกต่างกันด้วย

2.4.1 การเข้ารหัสคอนโวลูชัน [4]

โดยทั่วไปของตัวเข้ารหัสคอนโวลูชันจะมี Shift Register k ตัว (k-Stage Shift Register) เพื่อเลื่อนบิตของข้อมูลข่าวสารสำหรับทำการเข้ารหัสและมีตัววงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) จำนวน n ตัว เพื่อทำการเข้ารหัสโดยที่ K คือค่า Constraint Length เป็นค่าที่แสดงถึงจำนวน Shift Register ทั้งหมด โดยเอาท์พุทที่ออกมาจะได้ตามจำนวนของวงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) จะได้จำนวนข้อมูลข่าวสารที่เข้ารหัสต่อจำนวนข้อมูลคำรหัส (Code Rate) เท่ากับ k/n เมื่อ $k < n$

รูปที่ 2.3 แสดงให้เห็นว่าที่ค่า Constraint Length เท่ากับ 3 และ n เท่ากับ 2 ดังนั้น Code Rate เท่ากับ 1/2 ซึ่งจำนวนบิตของข้อมูลข่าวสารจะวิ่งเข้าไปใน Shift Register จากซ้ายไปขวาแล้ว ตัววงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) จะเป็นตัวเข้ารหัสโดยสลับเอาท์พุทที่ออกมาได้เป็นสองทางจะเห็นว่าตัววงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) จะเป็นตัวกำหนดคุณสมบัติของรหัสเนื่องจากถ้ามีการเปลี่ยนการเชื่อมต่อระหว่างตัวบวก (Adder) กับ Shift Register จะทำให้เกิดรหัสที่ต่างกัน จากรูปแสดงให้เห็นว่าข้อมูลข่าวสารเวกเตอร์ m = "1 0 1" เมื่อทำการเข้ารหัสจะเห็นว่ามีข้อมูลข่าวสารสามบิต เข้าตัวเข้ารหัส (Encoder) ที่เวลา t_1, t_2, t_3



รูปที่ 2.3 วงจรเข้ารหัสคอนโวลูชันที่ค่า R เท่ากับ 1/2 และค่า K เท่ากับ 3 [3]

รูปที่ 2.4 เมื่อข้อมูลข่าวสารเข้าไปในตัวเข้ารหัสจนครบจะมีการใส่บิต "0" เข้าไปในตัวเข้ารหัสเพื่อเคลียร์บิตในตัวเข้ารหัสเพื่อกลับไปอยู่ในสถานะ (State) เริ่มต้นที่เวลา t_4, t_5, t_6จะได้ข้อมูลคำรหัสที่สมบูรณ์ดังนี้ "1 0 0 0 1 1 1 1 1 0" ซึ่งการเขียนรูปแบบของภาคเข้ารหัสจะระบุเป็นเซตของ n Connection Vector และ n Modulo-2 Adder โดยเวกเตอร์มีมิติเท่ากับ K จากนั้นก็ให้ค่า บิต "1" แสดงตำแหน่งที่มีการเชื่อมต่อกันระหว่าง Shift Register กับวงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) และ บิต "0" กับตำแหน่งที่ไม่ได้มีการเชื่อมต่อกันระหว่าง Shift Register กับวงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) ดังนั้นจะได้ Connection Vector G_0 สำหรับการเชื่อมต่อด้านบน และ G_1 สำหรับการเชื่อมต่อด้านล่างดังสมการที่ (2.1) และ (2.2) [4] ตามลำดับ

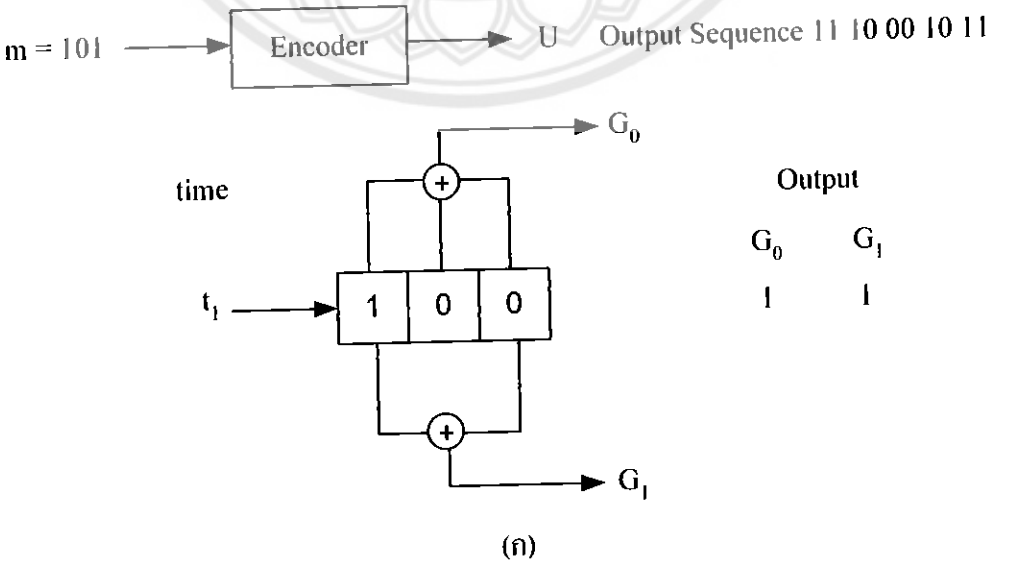
$$G_0 = \text{'111'} \tag{2.1}$$

$$G_1 = \text{'101'} \tag{2.2}$$

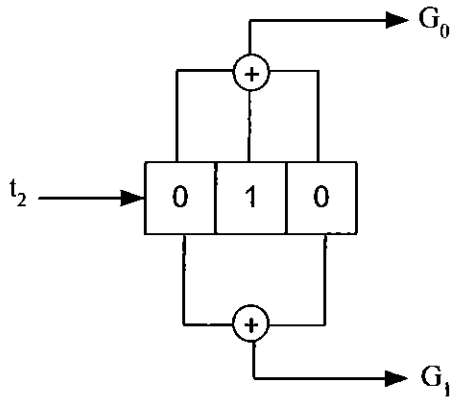
ซึ่งบางครั้งอาจใช้โพลีโนเมียล (Polynomial) ในการแสดงภาคเข้ารหัสโดยให้ดีกรีสูงสุดของโพลีโนเมียลเท่ากับ K - 1 และแทนสัมประสิทธิ์เป็น "1" เมื่อ Shift Register กับตัววงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) ต่อเข้าด้วยกัน และ "0" เมื่อ Shift Register กับตัววงจรบวกแบบมอดูโล 2 (Modulo-2 Adder) ไม่ได้ ต่อกันจากรูปที่ 2.3 เขียน $G_0(x)$ แทนการเชื่อมต่อด้านบนและ $G_1(x)$ แทนการเชื่อมต่อด้านล่างดังสมการที่ (2.3) และ (2.4) [4] ตามลำดับ

$$G_0(x) = 1 + x + x^2 \tag{2.3}$$

$$G_1(x) = 1 + x^2 \tag{2.4}$$

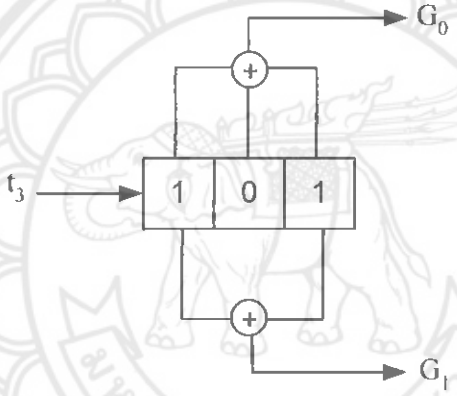


(ก)



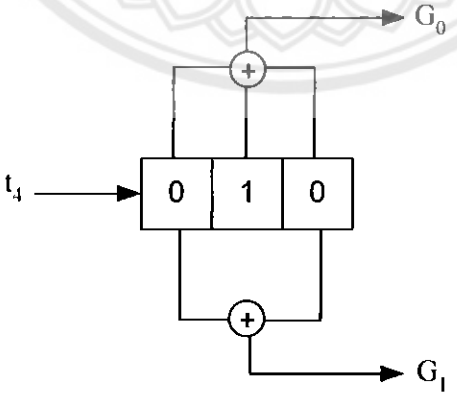
G_0	G_1
1	0

(ข)



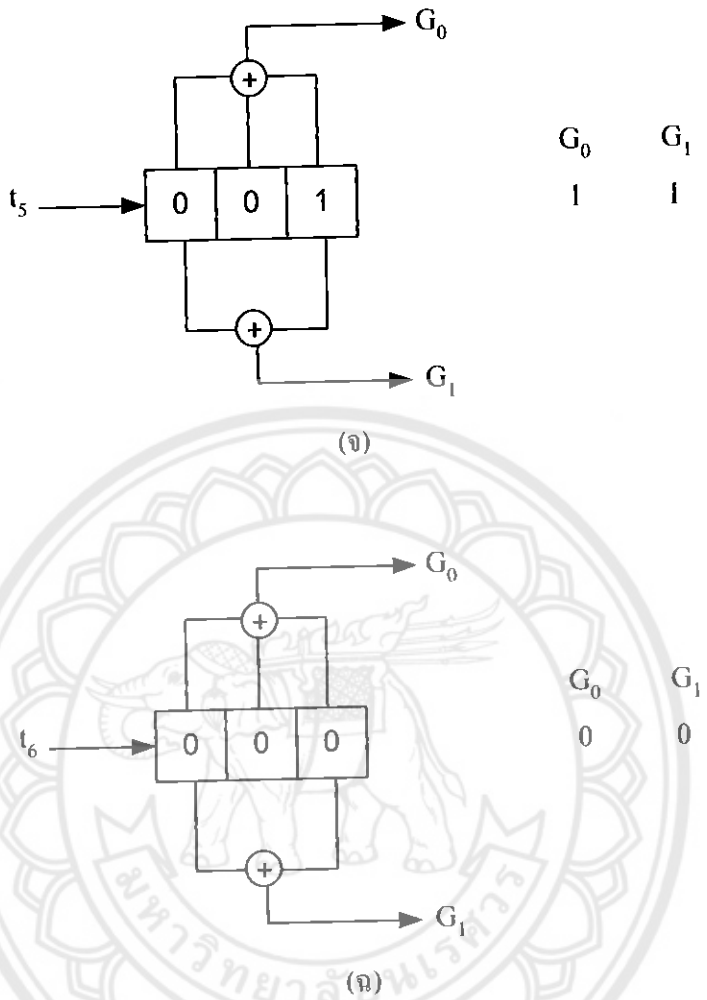
G_0	G_1
0	0

(ค)



G_0	G_1
1	0

(ง)



รูปที่ 2.4 Convolutionally encoding a message sequence with rate 1/2, K = 3 encoder [4]

ในตอนแรกสถานะของ Shift Register ทั้งหมดจะเป็นบิต "0" เมื่อมีข้อมูลเข้ามา คือ m = 101 ข้อมูลบิตแรกเป็น 1 จะเข้าสู่ Shift Register ด้านซ้ายมือสุดดังรูปที่ 2.4 (ก) จากนั้น G_0 ก็จะทำการมอดูโล 2 กันออกมาได้ Output เป็นบิต "1" เช่นเดียวกันกับ G_1 ซึ่งก็จะทำการมอดูโล 2 แล้วจะได้ Output ออกมาเป็นบิต "0"

จากรูปที่ 2.4 (ข) ข้อมูลที่ส่ง m จะเลื่อนไปเรื่อย ๆ จะเห็นได้ว่า บิต "1" ที่อยู่ใน Shift Register ด้านซ้ายมือสุดจะเลื่อนมาอยู่ที่ Shift Register ช่องกลาง แล้วข้อมูลที่ส่งบิตที่ 2 ซึ่งก็คือ บิต "0" จะเข้ามาแทนที่ใน Shift Register ด้านซ้ายมือสุดแทน จากนั้น G_0และ G_1จะทำการมอดูโล 2 ออกมาเป็น Output ได้เท่ากับ "0" และ "1" ตามลำดับ

จากรูปที่ 2.4 (ค) ข้อมูลที่ส่ง m จะเลื่อนเข้ามีอีก ซึ่งข้อมูลที่เข้ามานี้เป็นบิต "1" เมื่อข้อมูลเข้ามาแล้ว G_0 และ G_1 ก็จะทำการมอดูโล 2 ออกมาเป็น Output ได้เท่ากับ "0" และ "0" ตามลำดับ

จากรูปที่ 2.4 (ง), 2.4 (จ) และ 2.4 (ฉ) เมื่อไม่มีข้อมูลที่จะส่งแล้ว จะนำบิต "0" เข้ามาใน Shift Register แทน จนกว่าจะได้ค่า Output ออกมาเป็นบิต "0" ทั้งหมด จึงหยุดการเข้ารหัส

แล้วข้อมูลคำรหัส $U_0(x) = m(x)G_0(x)$ (2.5)

และ $U_1(x) = m(x)G_1(x)$ (2.6)

จากข้อมูลข่าวสาร m = '101' แทนด้วยโพลิโนเมียล $m(x) = 1 + x^2$ (2.7)

จะได้

$$m(x)G_0(x) = (1 + x^2)(1 + x + x^2) = 1 + x + x^3 + x^4 \quad (2.8)$$

$$m(x)G_1(x) = (1 + x^2)(1 + x^2) = 1 + x^4 \quad (2.9)$$

$$m(x)G_0(x) = 1 + x + 0*x^2 + x^3 + x^4 \quad (2.10)$$

$$m(x)G_1(x) = 1 + 0*x + 0*x^2 + 0*x^3 + x^4 \quad (2.11)$$

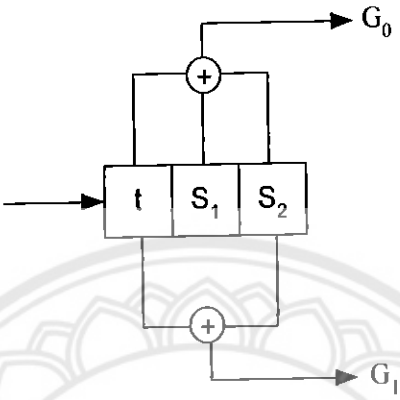
ดังนั้น ข้อมูลคำรหัสจะเป็น $U(x) = (1,1) + (1,0)x + (0,0)x^2 + (1,0)x^3 + (1,1)x^4$ (2.12)

2.4.2 วิธีการทำงานของวงจรเข้ารหัสคอนโวลูชัน [5]

ในการวิเคราะห์การทำงานของวงจรเข้ารหัสแบบคอนโวลูชันนั้นจะแสดงลักษณะ และวิธีการทำงานของวงจรเข้ารหัส โดยใช้รูปภาพแสดงความสัมพันธ์ระหว่างข้อมูลที่ถูกเก็บไว้ในวงจรเข้ารหัส ข้อมูลที่ถูกป้อนเข้ามา ณ เวลานั้น และค่าของคำรหัสที่จะถูกส่งออกไปเมื่อมีข้อมูลในกรณีต่างๆป้อนเข้ามา ซึ่งรูปแบบของภาพที่ใช้แสดงการทำงานของวงจรเข้ารหัสนั้นได้แก่ แผนภาพสถานะ (State Diagram) แผนภาพต้นไม้ (Tree Diagram) และ แผนภาพเทรลลิส (Trellis Diagram) โดยมีรายละเอียดดังต่อไปนี้

- **แผนภาพสถานะ (State Diagram)** สำหรับการแสดงการทำงานของวงจรเข้ารหัสข้อมูลแบบคอนโวลูชันโดยใช้แผนภาพสถานะนั้นจะเป็นการแสดงถึงความสัมพันธ์ระหว่างข้อมูลที่ถูกเก็บไว้ในวงจรเข้ารหัส ข้อมูลที่ถูกป้อนเข้ามา ณ เวลานั้นๆ และผลลัพธ์ที่ได้หลังจากมีการป้อนข้อมูลรูปแบบต่าง ๆ กับวงจร ซึ่งจะแสดงค่าอยู่ในตารางที่ 2.2 และแผนภาพสถานะดังรูปที่ 2.5 เป็นค่า State และ แผนภาพสถานะของวงจรเข้ารหัสข้อมูลตัวอย่างของรูปที่ 2.3 โดยในกรณีของวงจรตัวอย่างนั้นมีข้อมูลเก็บไว้ในวงจรเท่ากับสองบิต ดังนั้นจำนวนของข้อมูลทั้งหมดที่เป็นไปได้จะมีค่าเท่ากับ 2^2 หรือ 4 สถานะและสำหรับลูกศรที่อยู่ในรูปนั้นจะแสดงถึงลักษณะของการเปลี่ยนแปลงสภาวะของข้อมูลในวงจรเข้ารหัสจากสภาวะหนึ่งไปเป็นอีกสภาวะหนึ่งเมื่อมีข้อมูลป้อนเข้ามาในวงจร และเลข I/O ที่อยู่เหนือลูกศรนั้นจะแสดงถึงข้อมูลที่ถูก

ป้อนเข้ามา (I) โดยเส้นทึบแทนข้อมูลที่ถูกป้อนเข้ามาเป็น “0” เส้นประแทนข้อมูลที่ถูกป้อนเข้ามาเป็น “1” และ คำรหัสที่ได้หลังจากการทำงาน (O)



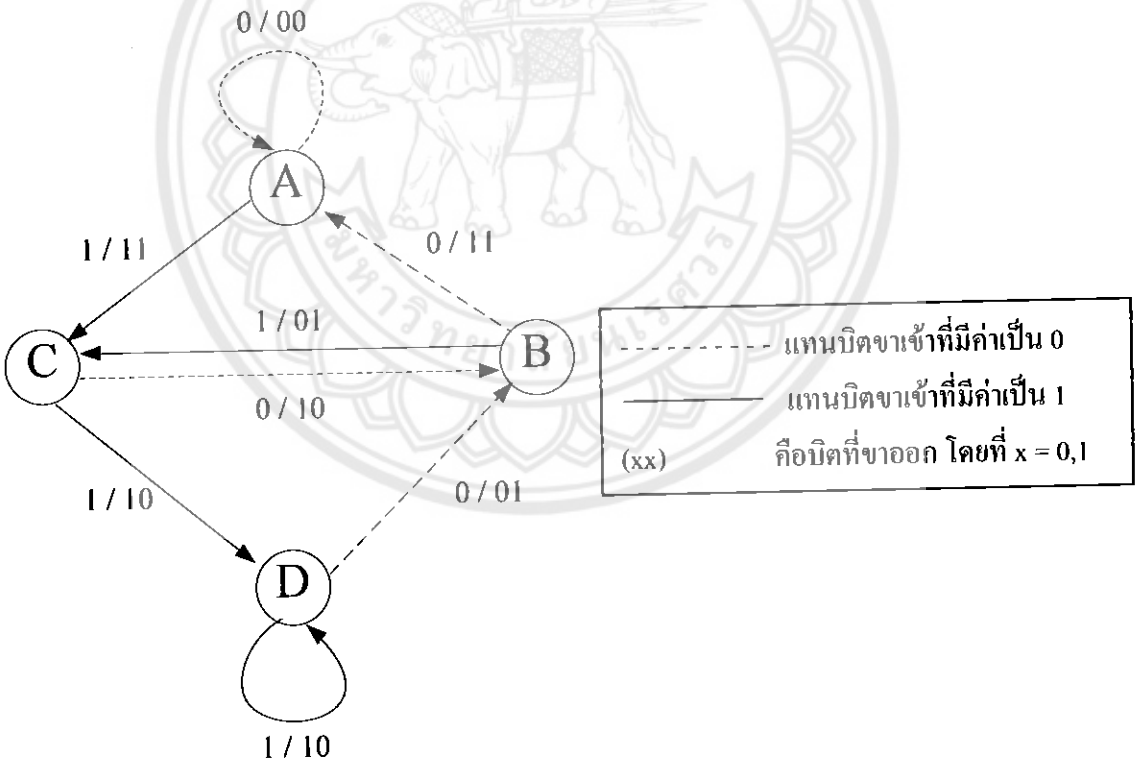
รูปที่ 2.5 วงจรเข้ารหัสจากรูปที่ 2.3 เพื่อใช้ในการกำหนดค่าสถานะ (State)

ตารางที่ 2.1 การกำหนดค่าสำหรับแต่ละสถานะ (State)

สถานะ	S_1	S_2
A	0	0
B	0	1
C	1	0
D	1	1

ตารางที่ 2.2 รายละเอียดการเปลี่ยนสถานะ (State)

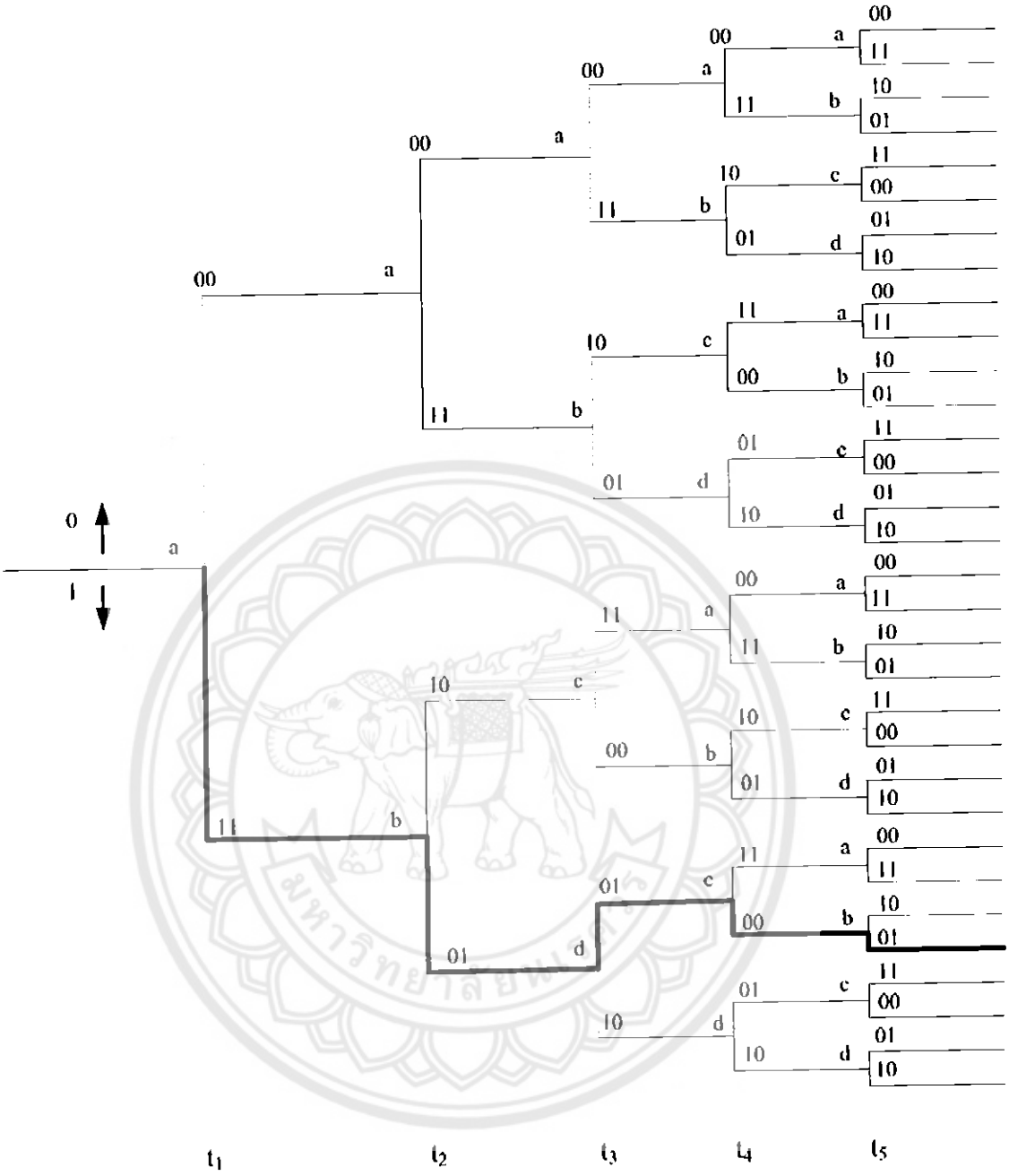
สถานะเดิม	Input	S_1	S_2	Output	สถานะใหม่
A	0	0	0	00	A
	1	0	0	11	C
B	0	0	1	11	A
	1	0	1	00	C
C	0	1	0	10	B
	1	1	0	01	D
D	0	1	1	01	B
	1	1	1	10	D



รูปที่ 2.6 แผนภาพสถานะ (State Diagram) ของวงจรเข้ารหัสที่ rate=1/2, K=3

ในช่วงเริ่มต้นของการเข้ารหัสตามรูปที่ 2.5 นั้น วงจรเข้ารหัสจะอยู่ในสถานะ A ($S_1 = 0, S_2 = 0$) เมื่อได้รับการป้อนรหัสข้อมูลบิตแรกเข้ามาในวงจรซึ่งมีค่าเท่ากับ 1 จะทำให้วงจรเปลี่ยนสถานะจาก A ไปเป็น C ($S_1 = 1, S_2 = 0$) ดูรูปที่ 2.4 และตารางที่ 2.2 ประกอบ เมื่อได้รับการป้อนข้อมูลบิตที่สองเข้ามาในวงจรซึ่งมีค่าเท่ากับ 0 จะทำให้วงจรเปลี่ยนสถานะจาก C ไป D ($S_1 = 1, S_2 = 1$) เมื่อได้รับการป้อนข้อมูลบิตที่สามเข้ามาในวงจรซึ่งมีค่าเท่ากับ 1 จะทำให้วงจรเปลี่ยนสถานะจาก D ไป B ($S_1 = 0, S_2 = 1$) และถ้านำลำดับการเปลี่ยนสถานะมาเรียงลำดับตามเวลา จะได้ผลดังรูปที่ 2.6

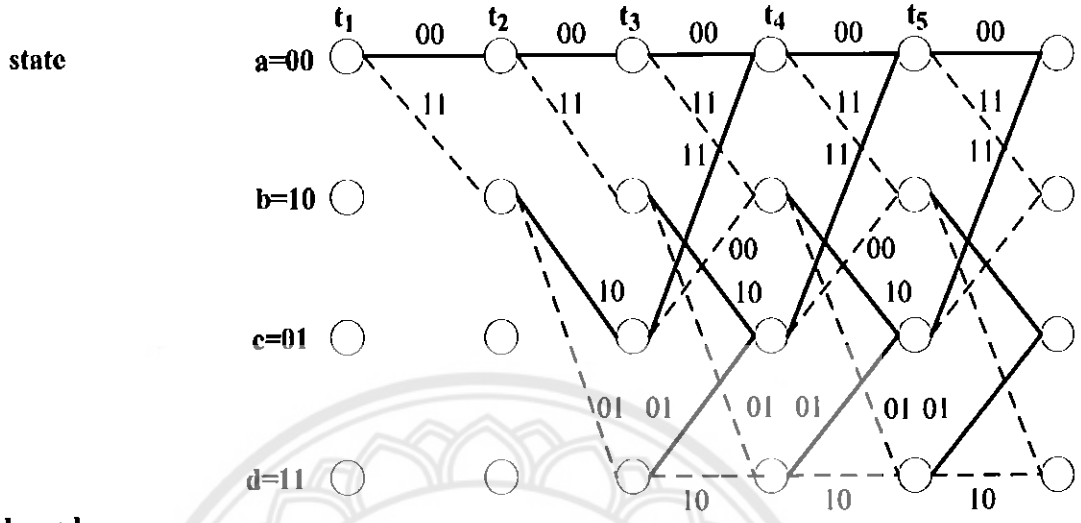
• แผนภาพต้นไม้ (Tree Diagram) สำหรับแผนภาพต้นไม้นั้นจะเป็นการพิจารณาถึงลักษณะของการทำงานของวงจรเข้ารหัสข้อมูลโดยที่จะมีการพิจารณาถึงค่าผลลัพธ์ที่ได้หลังจากการป้อนข้อมูลต่าง ๆ เข้าไปในวงจรเข้ารหัสเป็นหลักซึ่งในการพิจารณานั้นจะเริ่มต้นจาก ณ ตำแหน่งรากของแผนภาพต้นไม้ซึ่งจะมีการนำข้อมูลที่ถูกป้อนเข้ามาภายในวงจรเข้ารหัสเป็นตัวกำหนดทิศทางการเดินทางของข้อมูลในแผนภาพต้นไม้ ซึ่งจากตัวอย่างจะกำหนดให้มีการเลื่อนตำแหน่งไปด้านบนเมื่อมีการรับบิต "0" เข้ามา และจะเลื่อนลงด้านล่างเมื่อรับบิต "1" เข้ามาซึ่งหลังจากมีการเลื่อนตำแหน่งที่ใช้พิจารณาแล้ว จะมีการพิจารณาถึงข้อมูลที่จะถูกส่งออกไปจากภาคเข้ารหัส ณ เวลานั้นๆ จากข้อมูลที่อยู่เหนือเส้นทางในตำแหน่งที่มีการพิจารณา ซึ่งจากรูปที่ 2.7 จะมีอินพุต "1 1 0 1 1" ซึ่งจะได้เอาต์พุตเป็น "1 0 0 0 1 0 1 0 1 1"



รูปที่ 2.7 แผนภาพต้นไม้ (Tree diagram) ของภาคเข้ารหัสที่ rate=1/2, K=3 [1]

• **แผนภาพเทรลลิส (Trellis Diagram)** การแสดงการทำงานของวงจรเข้ารหัส โดยใช้แผนภาพเทรลลิสนั้นจะเป็นการแสดงการทำงานของวงจรเข้ารหัส โดยใช้แผนภาพสถานะมาทำการเปลี่ยนแปลงรูปแบบให้อยู่ในอีกลักษณะหนึ่งที่แสดงถึงการเปลี่ยนแปลงของข้อมูลต่าง ๆ ภายในวงจรเข้ารหัสข้อมูลที่ป้อนเข้ามา และคำรหัสที่ส่งออกไป ณ เวลาต่าง ๆ โดยที่จะมีลักษณะของแผนภาพเทรลลิส ดังรูปที่ 2.8 เป็นการแสดงการทำงานของวงจรดังตัวอย่างในรูปที่ 2.4 ที่มีการนำข้อมูลในอดีตจำนวนสองบิตมาทำการประมวลผลร่วมกับข้อมูล ณ เวลานั้น (จำนวนสถานะทั้งหมดในแผนภาพเทรลลิสจะมีค่าเท่ากับสี่สถานะ) และจะมีข้อมูลป้อนเข้ามาภายในวงจรครั้งละหนึ่งบิต ซึ่งเส้นทางต่าง ๆ

ที่อยู่ภายในแผนภาพเทรลลิสนั้น จะแสดงถึงลักษณะการเปลี่ยนแปลงสภาวะของวงจร และตัวเลข x/c ที่อยู่เหนือทางเดิน



รูปที่ 2.8 แผนภาพเทรลลิส (Trellis diagram) ของวงจรเข้ารหัสที่ rate=1/2, K=3 [1]

ในแต่ละเส้นทางนั้นจะแสดงถึงข้อมูลที่ป้อนเข้ามา ซึ่งเมื่อพิจารณาลักษณะของแผนภาพเทรลลิสแล้วจะพบว่า รูปแบบของแผนภาพเทรลลิสในแต่ละสถานะการทำงานนั้นจะมีลักษณะที่คล้ายกันแต่จะมีความแตกต่างกันเฉพาะส่วนหัว และท้ายซึ่งเป็นผลมาจากข้อมูลที่เก็บอยู่ในวงจรนั้นจะมีค่าเริ่มต้นจากสภาวะที่ข้อมูลทั้งหมดเป็นศูนย์ และจบลงที่สภาวะข้อมูลเป็นศูนย์เช่นกันดังนั้นเส้นทางอื่น ๆ ที่ไม่ผ่านจุดที่มีข้อมูลเป็นศูนย์ทั้งหมด ณ จุดเริ่มต้นและจุดสุดท้ายนั้นจะไม่ถูกนำมาพิจารณา ดังนั้นขนาดของความยาวในแผนภาพเทรลลิสนั้นจึงขึ้นอยู่กับข้อมูลที่ถูกนำมาเข้ารหัส

2.4.3 รูปแบบการถอดรหัสแก้ไขความผิดพลาดแบบคอนโวลูชัน [6]

สำหรับการถอดรหัสข้อมูลเป็นการนำข้อมูลที่รับได้ปลายทางที่ยังเป็นข้อมูลคำรหัสมาทำการประมวลผลเพื่อเปลี่ยนข้อมูลที่รับได้นั้น ให้กลับมาเป็นข้อมูลข่าวสารที่ต้องการ พร้อมทั้งทำการแก้ไขข้อมูลให้มีความถูกต้องมากที่สุด ซึ่งวิธีในการถอดรหัสนั้นจะมีอยู่หลาย ๆ วิธีด้วยกันแต่ที่นิยมที่ใช้งานจะมีอยู่สามวิธีได้แก่ Sequential Decoding, Threshold Decoding และ Viterbi Decoding

Sequential Decoding จะเป็นการถอดรหัสคอนโวลูชัน โดยมีการนำแผนภาพต้นไม้มาใช้ในการออกแบบการทำงานของวงจรถอดรหัส ซึ่งจะเป็นการค้นหาเส้นทางที่มีลักษณะที่เหมือนกับข้อมูลที่ปลายทางได้รับมามากที่สุด

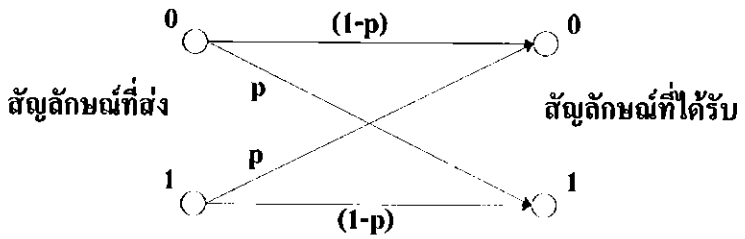
Threshold Decoding จะเป็นการถอดรหัสที่เริ่มมีการนำการคำนวณทาง Topological มาใช้ในการออกแบบการทำงานของวงจร ซึ่งจะทำให้การออกแบบต่าง ๆ นั้นง่ายขึ้นแต่จะทำให้ประสิทธิภาพในการทำงานลดลง

Viterbi Decoding เป็นการถอดรหัสคอนโวลูชันที่นิยมใช้มากที่สุด ซึ่งเป็นการถอดรหัสที่มีการนำแผนภาพเทรลลิสมาใช้ในการออกแบบการทำงานของวงจรซึ่งจะมีลักษณะการทำงานแบบ Maximum-Likelihood Decoding Algorithm ซึ่งจะหมายความว่าในการทำงานของวงจรถอดรหัสนั้นจะเป็นการนำข้อมูลที่รับได้ปลายทางนั้นไปทำการประมวลผลเพื่อค้นหาเส้นทางในแผนภาพเทรลลิสที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่ส่งมาจากต้นทางมากสุดซึ่งผลลัพธ์จากการทำงานนั้น จะได้เส้นทางเพียงเส้นทางเดียวจากเส้นทางทั้งหมดในแผนภาพเทรลลิส โดยจะมีกระบวนการที่เรียกว่า Viterbi Algorithm มาใช้ในการค้นหาเส้นทาง

2.4.4 วิธีการทำงานของวงจรถอดรหัสแบบ Viterbi

การถอดรหัสแบบ Viterbi นั้นจะมีรูปแบบสำหรับการถอดรหัสที่ใช้งานอยู่สองแบบได้แก่ การตัดสินใจแบบหยาบ (Hard-Decision) และการตัดสินใจแบบละเอียด (Soft-Decision) ซึ่งมีรายละเอียดดังนี้

การตัดสินใจแบบหยาบ (Hard-Decision) สำหรับการทำงานของวงจรถอดรหัสที่ใช้กระบวนการตัดสินใจแบบหยาบนั้นจะเป็นการพิจารณาข้อมูลที่รับเข้ามาโดยมีการพิจารณาข้อมูลที่รับเข้ามาในแต่ละบิตนั้น มีค่าของข้อมูลที่น่าจะเป็นบิต "0" หรือบิต "1" เท่านั้น โดยที่จะมีการพิจารณาถึงช่องสัญญาณที่ใช้งานในรูปแบบของ Binary Memoryless Channel ซึ่งจะมีลักษณะการพิจารณาดังรูปที่ 2.9 โดยตัวแปร p นั้นจะเป็นค่าความน่าจะเป็นที่จะเกิดความผิดพลาดในการส่งข้อมูลในระบบสื่อสารซึ่งจะทำให้มีการรับข้อมูลที่ผิดพลาดจากบิต "1" ไปเป็นบิต "0" และจากบิต "0" ไปเป็นบิต "1" ซึ่งในการทำงานจะมีการนำข้อมูลที่รับได้ ณ ปลายทางนั้น มาทำการตัดสินใจว่าเป็นบิต "0" หรือ "1" จากนั้นจึงนำข้อมูลที่รับได้ในแต่ละช่วงมาทำการเปรียบเทียบกับข้อมูลในแต่ละเส้นทาง ณ เวลานั้น โดยการคำนวณค่าที่เรียกว่าเมทริกซึ่งจะมีรูปแบบในการคำนวณอยู่หลายรูปแบบด้วยกันเช่น จำนวนของบิตที่แตกต่างกันหรือเหมือนกันเป็นต้น แล้วตัวแปรเมทริกนั้นจะถูกนำไปใช้ในการพิจารณาเส้นทางในการทำงานขั้นต่อไป



รูปที่ 2.9 การตัดสินใจแบบ Hard-Decision [1]

การตัดสินใจแบบละเอียด (Soft-Decision) สำหรับในกรณีของการถอดรหัสที่มีการใช้กระบวนการตัดสินใจแบบละเอียด จะเป็นการพิจารณาถึงข้อมูลที่รับเข้ามาได้โดยตัดสินใจระดับของข้อมูลที่รับเข้ามาได้โดยการแบ่งระดับสัญญาณที่ใช้ในการคำนวณหาค่าเมทริกที่มากกว่าสองระดับ เช่น 4 ระดับ (2 บิต) หรือ 8 ระดับ (3 บิต) เป็นต้น ซึ่งผลลัพธ์ที่ได้นั้นจะได้ข้อมูลรายละเอียดของข้อมูลที่ส่งมามีมากกว่ากรณีใช้การตัดสินใจแบบหยาบ ซึ่งข้อมูลที่ได้จากการตัดสินใจนั้นจะถูกนำมาใช้ในการคำนวณหาค่าเมทริกเพื่อเปรียบเทียบข้อมูลที่รับเข้ามา ณ เวลานั้น ๆ กับข้อมูลที่อยู่เส้นทางต่าง ๆ ณ เวลานั้นโดยรูปแบบที่ใช้ในการคำนวณจะมีรูปแบบที่แตกต่างกันออกไป

2.4.5 การถอดรหัสด้วย Viterbi Algorithm

การทำงานของการถอดรหัสด้วย Viterbi Algorithm เป็นการหาเส้นทางในแผนภาพเทรลลิส ซึ่งมีลักษณะที่ใกล้เคียงกับข้อมูลที่ถูกส่งมามากที่สุด โดย Viterbi Algorithm นั้น มีขั้นตอนในการทำงานแบ่งเป็นสามขั้นตอน ดังต่อไปนี้

ขั้นตอนที่ 1 Branch Metric Generation

ขั้นตอนนี้คำนวณหาค่า Branch Metric จากข้อมูลอินพุตที่รับเข้ามา r กับค่าเอาต์พุตของการเข้ารหัส C การคำนวณหาค่า Branch Metric ต้องคำนวณทุกๆ สาขาหรือ Branch ตามสถานะ การทำงานโดย Branch เท่ากับ 2^K การคำนวณหาค่า Branch Metric แสดงดังสมการที่ (2.13) [1]

$$BM_{i,j,n} = (r_n - C_{i,j})^2 \tag{2.13}$$

โดยที่ ค่า BM แทนค่า Branch Metric ระหว่างสถานะ i ไปยังสถานะ j ณ เวลา n

ค่า r แทนค่า ข้อมูลอินพุตที่รับเข้ามา ณ เวลา n

ค่า C แทนค่าเอาต์พุตของการเข้ารหัสระหว่างสถานะ i ไปยังสถานะ j ณ เวลา n

ขั้นตอนที่ 2 Survivor Path and Path Metric Update

ขั้นตอนนี้คำนวณหาค่า Survivor Path และ Path Metric จากจำนวนสถานะการทำงานทั้งหมด ค่า Path Metric ที่เลือกไว้เพื่อใช้ในการหาค่า Path Metric ครั้งต่อไป (Update) ส่วนค่า Survivor Path เป็นค่าที่ใช้ในการตัดสินใจหาค่าเอาต์พุต โดยการคำนวณหาค่า Survivor Path และ Path Metric นั้นค่าของ Branch Metric และ Path Metric จะถูกรวมเข้าด้วยกัน ซึ่งผลการบวกนั้นมีสองค่าที่เข้ามาในแต่ละจุดเชื่อมต่อ (Trellis Node) ของแผนภาพเทรลลิสโดยค่า Path Metric เป็นค่าที่เลือกจากค่าผลบวกที่น้อยกว่า ส่วนค่า Survivor Path เป็นสถานะการทำงานที่น้อยกว่าจากการเลือก Path Metric ซึ่งแสดงดังสมการที่ (2.14) [1]

$$PM_{j,n} = min(PM_{i,n-1} + BM_{i,j,n-1}, PM_{i+1,n-1} + BM_{i+1,j,n}) \qquad (2.14)$$

โดย ค่า *PM* แทน Path metric ระหว่างสถานะ *i* ไปยังสถานะ *j* ณ เวลา *n*

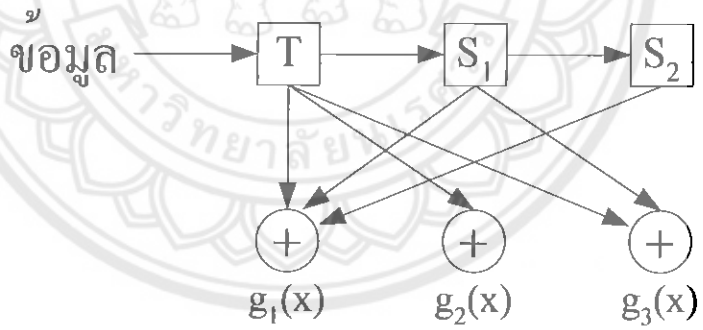
ค่า *BM* แทนค่า Branch metric ระหว่างสถานะ *i* ไปยังสถานะ *j* ณ เวลา *n*

ขั้นตอนที่ 3 Optimum Paths Trace Back

ขั้นตอนนี้เป็นขั้นตอนการตัดสินใจหาค่าเอาต์พุต โดยใช้ค่า Survivor Path ในแต่ละ สถานะที่บันทึกไว้มาตัดสินใจเลือกเส้นทางของข้อมูล โดยการตัดสินใจหาเส้นทางของข้อมูลจะเริ่มจาก Survivor Path ในอดีต (Trace Back) โดยในการเริ่มต้นจากเวลาผ่านไป *L* (Latency) โดยในทางปฏิบัติการ *L* ต้องมีค่ามากกว่าห้าเท่าของค่า *K* (Constraint Length) จึงทำให้ข้อมูลที่ถูกถอดรหัสมีความถูกต้องสูง

2.5 การถอดรหัส Viterbi โดยใช้แผนภาพเทรลลิส [7]

2.5.1 ขั้นตอนการเข้ารหัส



รูปที่ 2.10 ตัวอย่างวงจรเข้ารหัสคอนโวลูชัน [2]

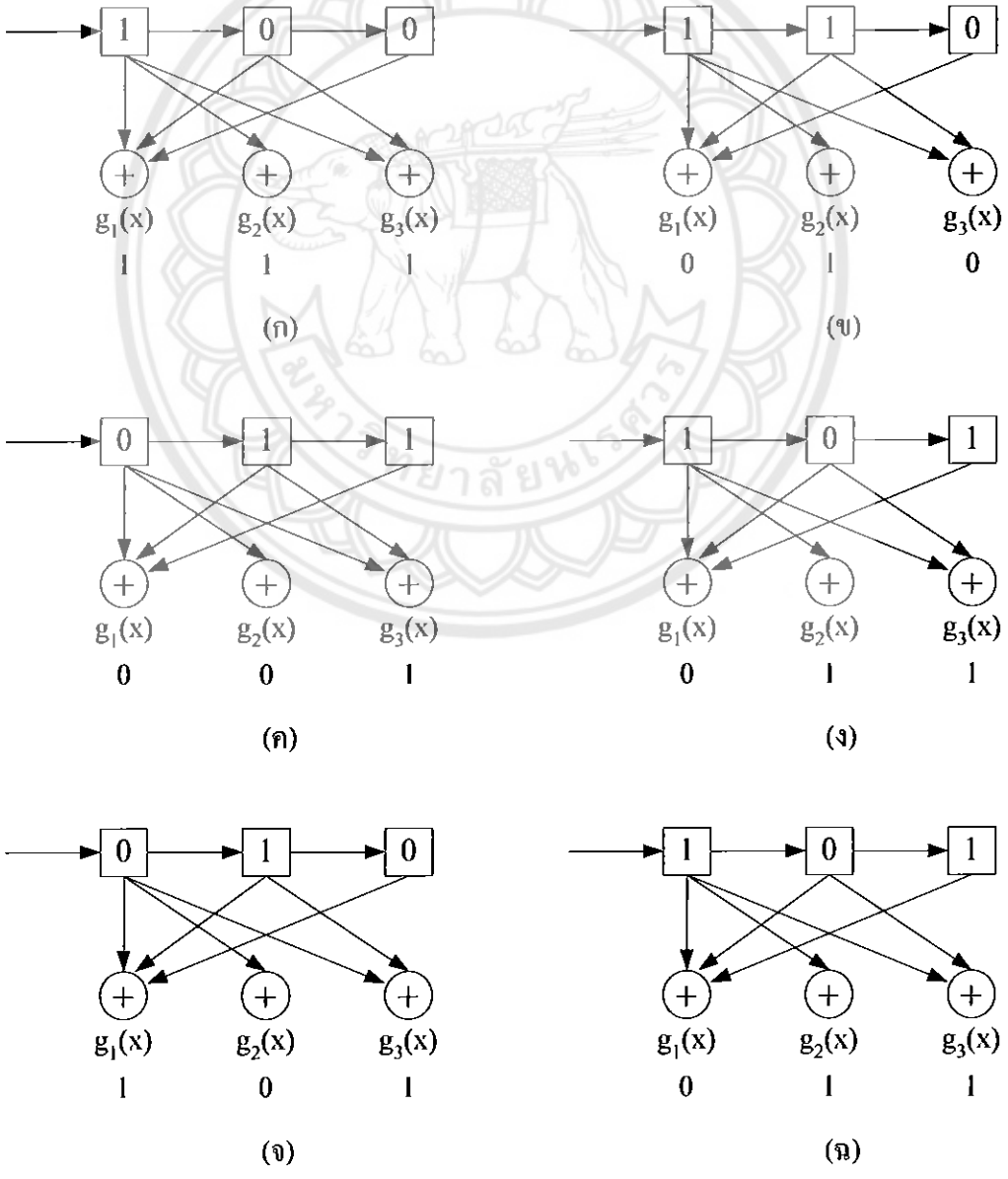
ในการทำความเข้าใจกับกลไกการทำงานของวิธีการเข้ารหัสคอนโวลูชัน เราจะอาศัยวงจรเข้ารหัสที่มีค่าอัตราการเข้ารหัสเท่ากับ 1/3 , K = 3 และ k = 1 ในรูปที่ 2.10 เป็นตัวอย่างในการอธิบาย จากวงจรในรูปจะเห็นว่า k = 1 ดังนั้นจำนวนบิตที่จะเลื่อนเข้าสำหรับการเข้ารหัสแต่ละครั้งมีค่าเพียง 1 บิต ค่า K = 3 และ k = 1 หมายความว่าต้องใช้ชิฟต์รีจิสเตอร์จำนวน 3 ชุด และในรูปประกอบด้วยวงจรพหุนามตัวกำเนิด 3 ชุด คือ $g_1(x)$, $g_2(x)$ และ $g_3(x)$ ดังสมการที่ (2.15), (2.16) และ (2.17) [2]

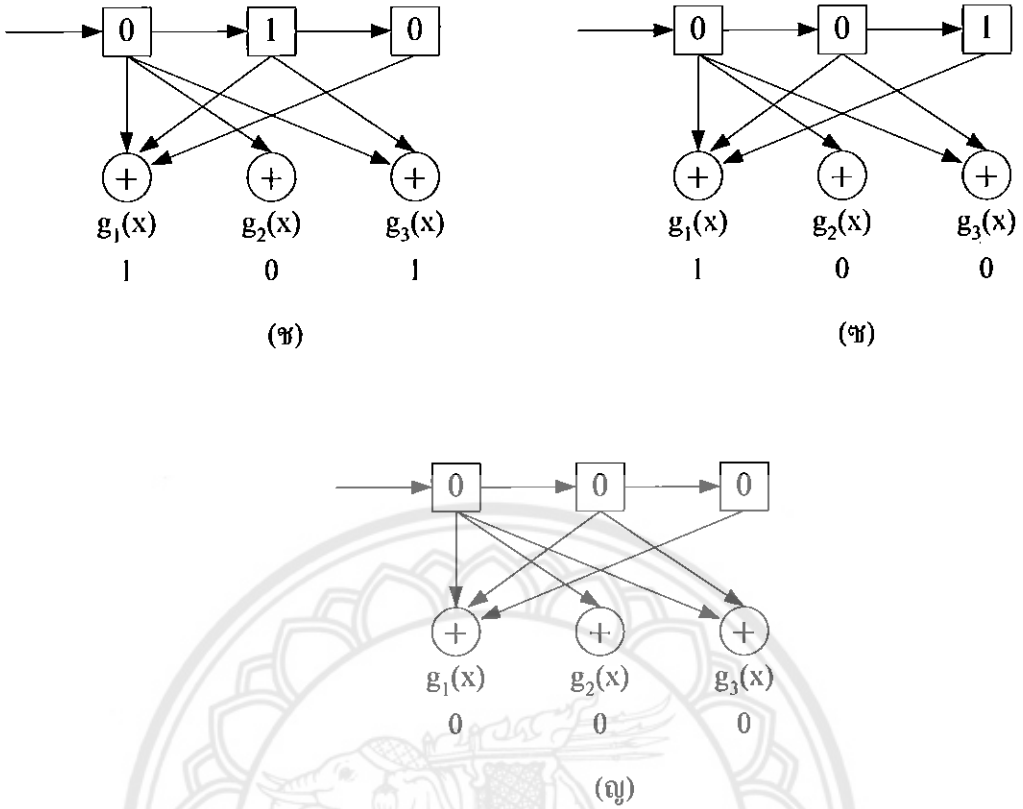
$$g_1(x) = 1+x+x^2 \quad (2.15)$$

$$g_2(x) = 1 \quad (2.16)$$

$$g_3(x) = 1+x \quad (2.17)$$

การเข้ารหัสข้อมูลแต่ละบิตจำให้เป็นคำรหัสที่มีจำนวน 3 บิต ซึ่งก็คือ ค่าที่ได้จาก $g_1(x)$, $g_2(x)$ และ $g_3(x)$ นั่นเอง โดยที่กระบวนการอ่านค่าเหล่านี้จะเป็นการวนสลับกันไป อย่างเช่น ถ้าข้อมูลที่ใช้ในการส่งเป็น 110101 คำรหัสที่ได้จากวงจรนี้ก็จะมีค่าเท่ากับ 111 010 001 011 101 011 101 100 ในส่วนของรายละเอียดการเข้ารหัสนั้นจะแสดงให้ดูในรูปที่ 2.11 จะเห็นได้ว่าในการเข้ารหัสบิตข้อมูลแต่ละชุดจะมีการเพิ่มบิตพิเศษเข้ามา เราเรียกบิตเหล่านี้ว่า บิตหาง (Tail bit) ต่อท้ายบิตข้อมูล





รูปที่ 2.11 ตัวอย่างขั้นตอนการเข้ารหัสของข้อมูล 110101 โดยใช้วงจรเข้ารหัสในรูปที่ 2.10 [2]

จากรูปที่ 2.11 ในตอนแรกค่าใน Shift Register จะมีค่าเป็น 000 แต่เมื่อทำการส่งข้อมูลเข้ามาสู่วงจรแล้ว บิตแรกที่ป้อนเข้ามาคือ 1 ทำให้ได้คำรหัส คือ 111 ดังรูปที่ 2.11 (ก) ต่อมาทำการป้อนบิตข้อมูลบิตที่สอง ซึ่งมีค่าเท่ากับ 1 จะทำให้ข้อมูลใน Shift Register เดิมนั้นเลื่อนออกไปทางขวา ส่งผลให้ข้อมูลบิตแรกเลื่อนมาอยู่ใน Shift Register ตัวกลาง และทำให้ได้คำรหัส คือ 010 ดังรูปที่ 2.11 (ข) ต่อมาก็จะทำการป้อนบิตข้อมูลบิตที่ 3 ซึ่งมีค่าเท่ากับ 0 เข้าสู่วงจร แล้วข้อมูลใน Shift Register ก็จะเลื่อนไปทางขวาอีก ทำให้ได้คำรหัส คือ 001 ดังรูปที่ 2.11 (ค) ทำการส่งข้อมูลอย่างนี้ต่อไปเรื่อย ๆ จนถึงบิตสุดท้ายของข้อมูลที่ต้องการส่ง เมื่อส่งบิตสุดท้ายแล้วต่อไปเราจะทำการส่งบิต 0 เข้ามาในวงจรจนกว่าจะทำให้ได้ค่าของคำรหัสที่มีค่าเป็น 0 ทั้งหมด จึงหยุดทำการส่ง ดังรูปที่ 2.11 (ช), 2.11 (ซ) และ 2.11 (ญ)

2.5.1 แผนภาพการเปลี่ยนสถานะ

แผนภาพการเปลี่ยนสถานะ เป็นการนำข้อมูลการเข้ารหัสช่องสัญญาณมาแสดงให้อยู่ในรูปแผนภาพเพื่อให้ดูง่ายต่อความเข้าใจ และสามารถนำไปใช้เป็นแผนภาพเทรลลิสได้ในอนาคต

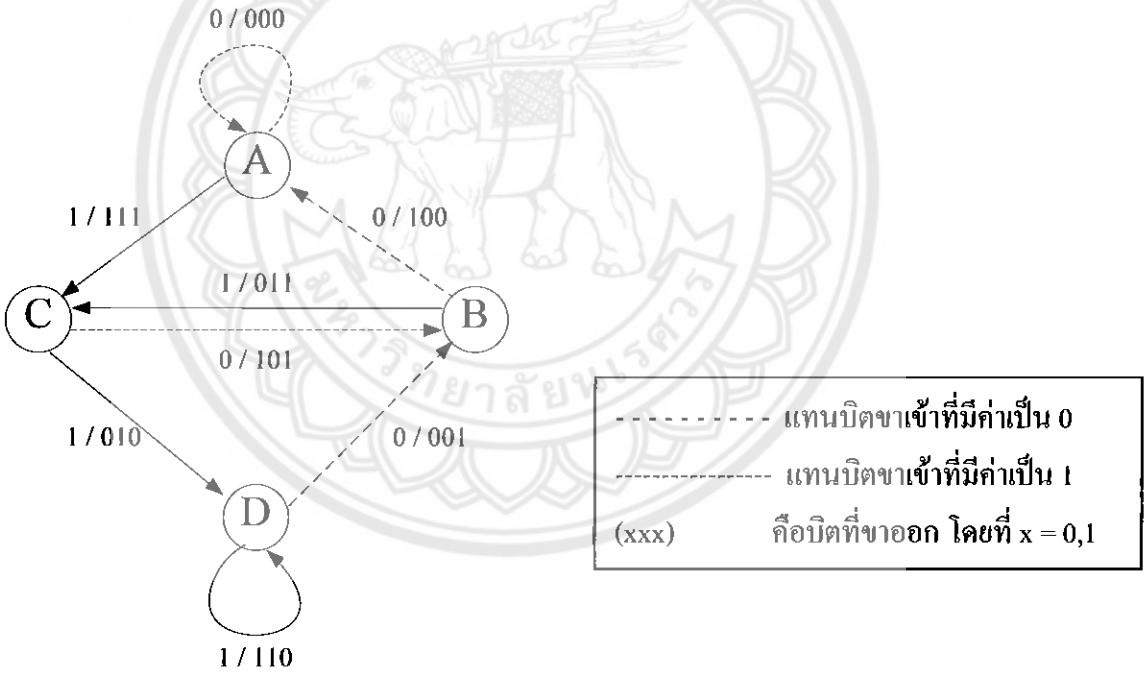
โดยกำหนดให้สถานะหมายถึงค่าของชิฟต์รีจิสเตอร์ S_1 S_2 นั่นเอง ดังนั้นจึงมีสถานะได้ทั้งสิ้น 4 สถานะ เราจะนำรายละเอียดการกำหนดสถานะในตารางที่ 2.1ในหัวข้อ 2.4.2 มาใช้

ห้องสมุดคณะวิศวกรรมศาสตร์

ตารางที่ 2.3 รายละเอียดการเปลี่ยนสถานะของการถอดรหัสแบบ Viterbi

สถานะเดิม	Input	S_1	S_2	Output	สถานะใหม่
A	0	0	0	000	A
	1	0	0	111	C
B	0	0	1	100	A
	1	0	1	011	C
C	0	1	0	101	B
	1	1	0	010	D
D	0	1	1	001	B
	1	1	1	110	D

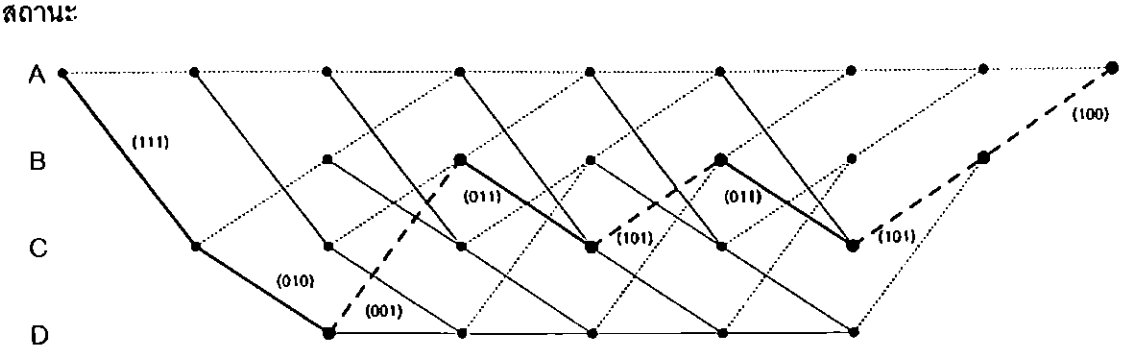
5000116
I 5081324 .C.2
วร.
พ187ก
2549



รูปที่ 2.12 แผนภาพการเปลี่ยนสถานะของวงจรเข้ารหัสคอนโวลูชันตามโครงสร้างในรูปที่ 2.11 [2]

2.5.2 แผนภาพเทรลลิส

เราจะนำแผนภาพการเปลี่ยนสถานะมาเขียนในทางแกนเวลาจะทำให้ได้ แผนภาพเทรลลิส โดยที่จุดเริ่มต้นของแผนภาพอยู่ที่สถานะ A เสมอ จากนั้นเมื่อทำการป้อนบิตข้อมูลจะทำให้สถานะเปลี่ยนไป แต่หลังจากบิตที่ 3 ไปโครงสร้างของแผนภาพจะเริ่มทำซ้ำ และเมื่อป้อนบิตข้อมูลจนครบแล้วก็จะเติมบิตที่มีค่าเป็น 0 เข้าไปจนทำให้กลับไปสู่สถานะ A อีกครั้ง



รูปที่ 2.13 โครงสร้างแผนภาพเทรลลิสสำหรับวงจรเข้ารหัสคอนโวลูชันของรูปที่ 2.12 [2]

2.5.3 การถอดรหัสคอนโวลูชัน [2]

การถอดรหัสที่ภาครับสามารถทำได้หลายวิธี แต่วิธีที่จะนำเสนอในที่นี้คือวิธีของ Viterbi เพราะเป็นวิธีที่มีประสิทธิภาพและนำมาใช้งานได้ดี การทำงานของวิธี Viterbi อาศัยแผนภาพเทรลลิสเป็นกลไกพื้นฐานในกระบวนการถอดรหัส จะอาศัยตัวอย่างการถอดรหัสของคำรหัสที่มีขนาดสั้น ๆ เพื่อช่วยประกอบในการอธิบายกรรมวิธีการถอดรหัสแต่ละขั้นโดยละเอียด คำรหัสตัวอย่างที่จะใช้ในการถอดรหัส คือ 111 010 001 011 101 011 101 100 ซึ่งได้จากการเข้ารหัสข้อมูล 110101 โดยอาศัยวงจรเข้ารหัสในรูปที่ 2.10 เพื่อให้เห็นภาพว่ารหัสคอนโวลูชันมีขีดความสามารถในการแก้ไขความผิดพลาดที่อาจเกิดขึ้นระหว่างการส่งผ่านช่องสัญญาณ ในที่นี้จะสมมติให้คำรหัสที่ได้รับมีความผิดพลาดเกิดขึ้น 2 บิต คือ บิตที่ 2 และ 5 ฉะนั้นชุดบิตที่จะนำไปผ่านการถอดรหัสจึงมีค่าเท่ากับ 101 000 001 011 101 011 101 100

ในการถอดรหัสเราจะต้องเริ่มพิจารณาที่สถานะ A เสมอ ขั้นแรกให้มองลึกเข้าไปในแผนภาพเทรลลิส 3 ระดับ และพิจารณาดูเส้นทางแต่ละเส้นทางซึ่งมีทั้งหมด 8 เส้นทางว่ามีความคล้ายคลึงกับชุดบิตที่ได้รับมากน้อยเพียงใด โดยอาศัยการวัดค่าระยะแฮมมิ่งระหว่างเส้นทางเหล่านั้นกับชุดบิตที่รับได้ 9 บิตแรก รายละเอียดของการเปรียบเทียบทั้งหมดได้แสดงไว้ในตารางที่ 2.4

ตารางที่ 2.4 การหาระยะแฮมมิ่งระหว่างชุดบิตที่รับได้ 101000001 เฉพาะ 9 บิตแรกกับรหัสทั้ง 8 เส้นทาง [2]

เส้นทาง	รหัส	ระยะแฮมมิ่ง	Survivor?
AAAA	000000000	3	Yes
ACBA	111101100	5	No
ACDB	111010001	2	Yes
AACB	000111101	6	No
AAAC	000000111	4	No
ACBC	111101011	4	Yes
ACDD	111010110	5	Yes
AACD	000111010	7	No

จากค่าระยะแฮมมิ่งที่ได้ก็นำมาเปรียบเทียบเป็นคู่ทั้งหมด 4 คู่ เช่น คู่แรกที่สิ้นสุดที่สถานะ A เส้นทาง AAAA มีขนาดของระยะแฮมมิ่งเท่ากับ 3 ซึ่งสั้นกว่าเส้นทาง ACBA ดังนั้น เส้นทางที่สั้นกว่านี้จะถูกเลือก หรือกล่าวว่า เส้นทางนี้เป็นเส้นทาง Survivor คู่ที่สองที่ไปสิ้นสุดที่สถานะ B เส้นทาง ACDB มีค่าของระยะแฮมมิ่งที่สั้นกว่าเส้นทาง AACB อย่างชัดเจน ดังนั้นเส้นทางที่จะเลือกเป็น Survivor คือเส้นทาง ACDB ในกรณีของคู่ที่สิ้นสุดที่สถานะ C เส้นทางทั้งสองมีค่าระยะแฮมมิ่งที่เท่ากันคือ 4 ให้เลือกเส้นทางใดเส้นทางหนึ่งเป็นเส้นทาง Survivor ก็ได้ ในตัวอย่างได้เลือกเส้นทาง ACBC ส่วนกรณีสุดท้ายได้เลือกเส้นทาง ACDD ไว้เพราะมีค่าระยะแฮมมิ่งที่สั้นกว่าเส้นทาง AACD ผลลัพธ์ที่ได้คือ มี 4 เส้นทางที่เป็นเส้นทางที่เป็นไปได้

ตารางที่ 2.5 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 10 – 12 ซึ่งมีค่าเท่ากับ 011 [2]

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
AAAAA	3	AA	2	5	No
ACDBA	2	BA	3	5	Yes
ACBCB	4	CB	2	6	No
ACDDB	4	DB	1	5	Yes
AAAAC	3	AC	1	4	No
ACDBC	2	BC	0	2	Yes
ACBCD	4	CD	1	5	Yes

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
ACDDD	5	DD	2	7	No

ในขั้นที่สองเราจะมาดูข้อมูลทีละ 3 บิตหรือ เพียงหนึ่งขั้นของการเปลี่ยนสถานะจาก เส้นทาง 4 เส้นทางที่ได้คัดเลือกเป็น Survivor จากขั้นแรก โดยเราจะนำสถานะสุดท้ายของแต่ละเส้นมาเป็นจุดเริ่มต้นเพื่อใช้ในการพิจารณาหาเส้นทางต่อไป ค่าระยะแฮมมิ่งของเส้นทางใหม่ที่ได้ให้นำไปรวมกับค่าระยะแฮมมิ่งของค่าเดิมที่คำนวณได้ในขั้นก่อน แล้วก็ทำการตัดสินใจเก็บเส้นทางที่สั้นกว่าไว้ รายละเอียดของขั้นตอนนี้สำหรับการพิจารณา 3 บิตถัดมาคือ บิตที่ 9 – 12 ได้แสดงในตารางที่ 2.5 กระบวนการที่ใช้ในขั้นตอนนี้ให้นำมาใช้ซ้ำในการพิจารณาต่อคราวละ 3 บิตจนกระทั่งชุดบิตที่ต้องการถอดรหัสหมดไป สำหรับการพิจารณาบิตที่ 13 – 15, 16 – 18, 19 – 21 และ 22 – 24 สามารถดูได้จากตารางที่ 2.6 – 2.9 ตามลำดับ

ตารางที่ 2.6 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 13 – 15 ซึ่งมีค่าเท่ากับ 101 [2]

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
ACDBAA	5	AA	2	7	No
ACDDBA	5	BA	1	6	Yes
ACDBCB	2	CB	0	2	Yes
ACBCDB	5	DB	1	6	No
ACDBAC	5	AC	1	6	Yes
ACDDBC	5	BC	2	7	No
ACDBCD	2	CD	3	5	Yes
ACBCDD	5	DD	2	7	No

ตารางที่ 2.7 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 16 – 18 ซึ่งมีค่าเท่ากับ 011 [2]

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
ACDDBAA	6	AA	2	8	No
ACDBCBA	2	BA	3	5	Yes
ACDBACB	6	CB	2	8	No
ACDBCDB	5	DB	1	6	Yes
ACDDBAC	6	AC	1	7	No
ACDBCBC	2	BC	0	2	Yes
ACDBACD	6	CD	1	7	Yes
ACDBCDD	5	DD	2	7	No

ตารางที่ 2.8 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 19 – 21 ซึ่งมีค่าเท่ากับ 101[2]

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
ACDBCBAA	5	AA	2	7	No
ACDBCDBA	6	BA	1	7	Yes
ACDBCBCB	2	CB	0	2	Yes
ACDBACDB	7	DB	1	8	No
ACDBCBAC	5	AC	1	6	Yes
ACDBCDBC	6	BC	2	8	No
ACDBCBCD	2	CD	3	5	Yes
ACDBACDD	7	DD	2	9	No

ตารางที่ 2.9 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิตที่ 22 – 24 ซึ่งมีค่าเท่ากับ 100 [2]

เส้นทาง	ระยะแฮมมิ่งของ Survivor เดิม	ส่วนของเส้นทางใหม่	ระยะแฮมมิ่งเฉพาะส่วนของเส้นทางใหม่	ระยะแฮมมิ่งของเส้นทางทั้งหมด	Survivor?
ACDBCDBAA	7	AA	1	8	No
ACDBCBCBA	2	BA	0	2	Yes
ACDBCBACB	6	CB	1	7	Yes
ACDBCBCDB	5	DB	2	7	No
ACDBCDBAC	7	AC	2	9	No
ACDBCBCBC	2	BC	3	5	Yes
ACDBCBACD	6	CD	2	8	No
ACDBCBCDD	5	DD	1	6	Yes

เมื่อได้ทำตามขั้นตอนที่ 2 จนครบทุกบิตแล้วจะได้เส้นทาง Survivor ทั้งหมด 4 เส้นทาง ดังนี้คือ ACDBCBCBA, ACDBCBACB, ACDBCBCBC, ACDBCBCDD ซึ่งแต่ละเส้นทางมีระยะแฮมมิ่งเท่ากับ 2, 7, 5 และ 6 ตามลำดับ เมื่อเปรียบเทียบเส้นทางทั้ง 4 เส้นพบว่าเส้นทางที่สั้นที่สุดก็คือเส้นทาง ACDBCBCBA ซึ่งมีค่าระยะแฮมมิ่งเท่ากับ 2 ฉะนั้น เราจึงตัดสินใจว่าเส้นทางดังกล่าวนี้คือเส้นทางแทนคำรหัสที่ถูกต้อง และเมื่อแปลงเส้นทางนี้ให้อยู่ในรูปของชุดบิตก็จะได้เป็น 111 010 001 011 101 011 101 100 ถูกต้องตรงกับคำรหัสที่ต้นทางได้ส่งออก ซึ่งหมายความว่าบิตตำแหน่งที่ 2 และ 5 ซึ่งผิดไปในระหว่างการส่งผ่านช่องสัญญาณได้รับการแก้ไขให้ถูกต้องแล้ว

ในบทที่ 2 นี้ได้ทำการศึกษาหลักการพื้นฐานของระบบสื่อสาร และหลักการพื้นฐานของการเข้ารหัสช่องสัญญาณ ซึ่งสามารถแบ่งออกได้เป็น 2 กรณี คือ ARQ (Automatic Repeat Request) เป็นการตรวจจับบิตข้อมูลที่ผิดพลาด มีความสามารถในการระบุได้ว่ามีบิตข้อมูลผิดพลาดตรงตำแหน่งใด แต่ไม่มารถแก้ไขได้ และ FEC (Forward Error Correction) การตรวจจับบิตข้อมูลที่ผิดพลาด สามารถระบุได้ว่าเกิดบิตข้อมูลที่ผิดพลาดตรงตำแหน่งใด และยังสามารถแก้ไขข้อมูลผิดพลาดนั้นได้ตรงปลายทาง

โดยที่ FEC (Forward Error Correction) ยังสามารถแบ่งออกได้เป็นอีก 2 ประเภท คือ การเข้ารหัสช่องสัญญาณแบบบล็อก (Block Codes) และ การเข้ารหัสช่องสัญญาณแบบคอนโวลูชัน (Convolutional Codes) ซึ่งในโครงงานนี้จะทำการศึกษาการเข้ารหัสช่องสัญญาณแบบคอนโวลูชัน โดยที่ในบทนี้ได้ทำการศึกษาเกี่ยวกับการเข้ารหัสคอนโวลูชัน วิธีการทำงานของวงจรเข้ารหัสคอนโวลู-

ชัน ซึ่งสามารถแบ่งออกได้เป็น แผนภาพสถานะ (State Diagram), แผนภาพต้นไม้ (Tree Diagram) และแผนภาพเทรลลิส (Trellis Diagram) ศึกษาการทำงานของวงจรถอดรหัส Viterbi Decoder และยังสามารถแบ่งออกได้เป็น 2 ชนิดได้แก่ Hard Decision และ Soft Decision ในช่วงสุดท้ายได้ยกตัวอย่างการถอดรหัส Viterbiโดยใช้แผนภาพเทรลลิส จากการศึกษาพบว่าถ้ามี Redundancy bit จำนวนมากจะทำให้การแก้ไขบิตข้อมูลที่ผิดพลาดได้ดีขึ้น

ในบทต่อไปจะอธิบายถึงวิธีการออกแบบดำเนินโครงงาน และการดำเนินโครงงาน โดยได้แสดงขั้นตอนการเขียนโปรแกรม และการสร้าง Graphic User Interfaces อย่างละเอียด

บทที่ 3
การออกแบบโครงงาน และวิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงโครงสร้างการทำงาน และการออกแบบการเขียนโปรแกรมจำลองการเข้ารหัสสัญญาณแบบ Convolutional Code และการถอดรหัสแบบ Viterbi Decoder อีกทั้งยังได้อธิบายวิธีการสร้าง Graphic User Interfaces เพื่อใช้ในการแสดงค่าต่าง ๆ โดยจะแบ่งออกเป็นขั้นตอนดังนี้ คือ

3.1 ขั้นตอนการออกแบบโปรแกรม [8]

3.1.1 สร้างสัญญาณ

ในขั้นแรกต้องทำการสร้างสัญญาณขึ้นมา เพื่อนำสัญญาณนี้มาเข้ารหัส Convolutional Code ก่อนที่จะทำการส่งเข้าไปในระบบสื่อสาร ซึ่งการสร้างสัญญาณนี้ทำได้โดยใช้วิธีการสุ่มรหัสของสัญญาณขึ้นมา

3.1.2 เข้ารหัสช่องสัญญาณ

ขั้นที่สองจะนำสัญญาณที่สร้างขึ้นมา ไปเข้ารหัส Convolutional Code โดยที่มีอัตราการเข้ารหัสแตกต่างกันออกไป 7 แบบ ได้แก่ อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 1/3, อัตราการเข้ารหัส 1/5, อัตราการเข้ารหัส 1/7, อัตราการเข้ารหัส 2/3, อัตราการเข้ารหัส 2/5, อัตราการเข้ารหัส 2/7

3.1.3 สร้างสัญญาณรบกวน

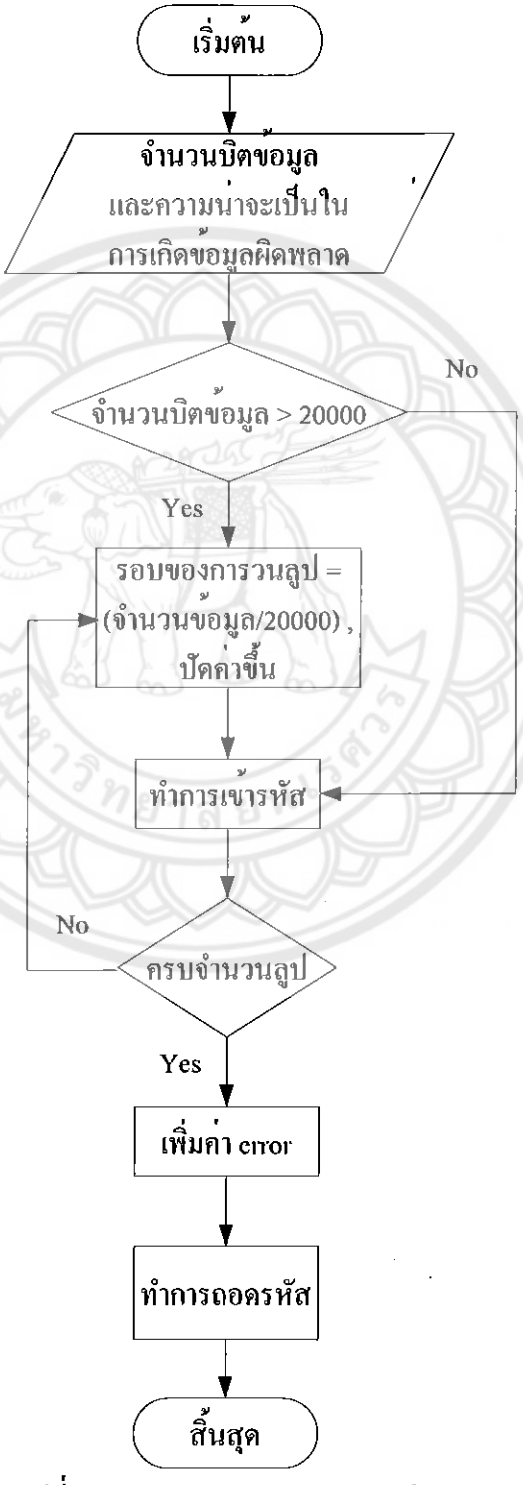
ขั้นที่สามจะทำการสร้างสัญญาณรบกวนจำลองขึ้นมา โดยใช้วิธีการสุ่มเช่นเดียวกันกับการสร้างสัญญาณที่ใช้ในการส่ง การเพิ่มสัญญาณรบกวนเข้าไปในข้อมูลที่ทำการเข้ารหัส Convolutional Code แล้วก็เพื่อใช้เป็นข้อพิสูจน์ในการแก้ไขบิตที่ผิดพลาดตรงปลายทาง ว่ามีความสามารถในการแก้ไขหรือไม่

3.1.4 รวมสัญญาณที่เข้ารหัสกับสัญญาณรบกวน

ขั้นที่สี่จะนำข้อมูลที่ทำการเข้ารหัส Convolutional Code แล้วมารวมกับสัญญาณรบกวนที่สร้างขึ้นมา ซึ่งในขั้นตอนนี้จะเป็นขั้นตอนที่สมมติว่าในกระบวนการส่งข้อมูล ของระบบสื่อสารนั้นเกิดข้อผิดพลาดขึ้นมา เพื่อให้ในส่วนของการถอดรหัสที่ตรงปลายทาง ทำการแก้ไขข้อมูลที่ผิดพลาดนี้ให้มีความถูกต้องได้

3.1.5 การแก้ไขข้อมูลที่ผิดพลาดตรงปลายทาง

ในขั้นสุดท้ายทำการออกแบบโปรแกรมในการแก้ไขบิตผิดพลาด โดยใช้วิธีการของ Viterbi Decode ซึ่งสามารถแบ่งออกได้เป็น 2 ชนิด คือ Hard Decision และ Soft Decision รายละเอียดได้กล่าวไว้แล้วในบทที่ 2



รูปที่ 3.1 Flowchart การทำงานของโปรแกรม

3.2 การออกแบบ Graphic User Interfaces และขั้นตอนการดำเนินงาน [9]

3.2.1 Type of Convolutional Code

ในขั้นตอนนี้จะทำการสร้างหน้าต่างในการเลือกชนิดของการเข้ารหัสสัญญาณ Convolutional Code แบบต่าง ๆ ซึ่งได้แก่

1. Convolutional Code 1/2 (อัตราการเข้ารหัสช่องสัญญาณ 1/2)
2. Convolutional Code 1/3 (อัตราการเข้ารหัสช่องสัญญาณ 1/3)
3. Convolutional Code 1/5 (อัตราการเข้ารหัสช่องสัญญาณ 1/5)
4. Convolutional Code 1/7 (อัตราการเข้ารหัสช่องสัญญาณ 1/7)
5. Convolutional Code 2/3 (อัตราการเข้ารหัสช่องสัญญาณ 2/3)
6. Convolutional Code 2/5 (อัตราการเข้ารหัสช่องสัญญาณ 2/5)
7. Convolutional Code 2/7 (อัตราการเข้ารหัสช่องสัญญาณ 2/7)
8. Compare Bit Error Rate เป็นการเปรียบเทียบประสิทธิภาพในการลดระดับการเกิดข้อผิดพลาดโดยวิธีการเข้ารหัสในแบบต่าง ๆ

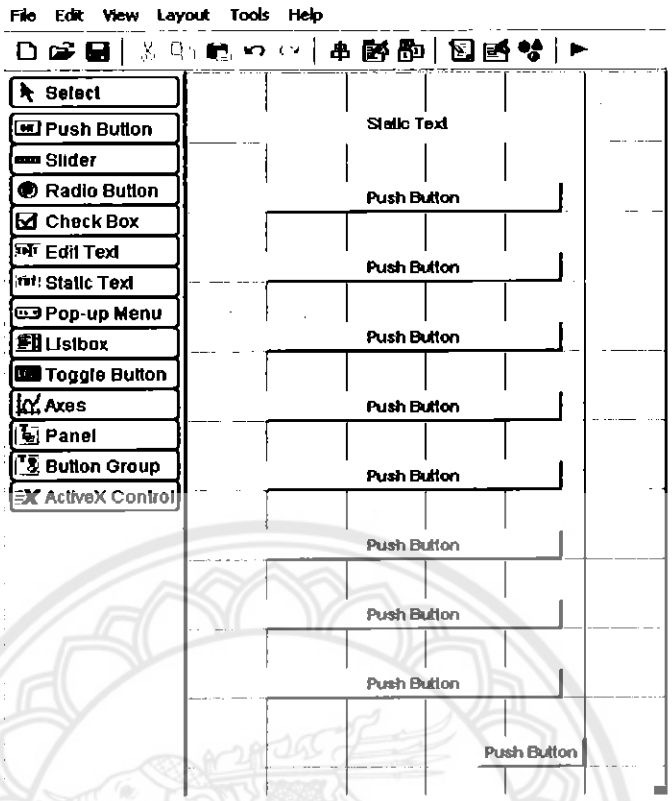
ขั้นตอนการดำเนินงาน

1. ทำการสร้างปุ่ม Push Button เพื่อใช้ในการเลือกค่าอัตราการเข้ารหัสแบบต่าง ๆ ได้แก่ อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 1/3, อัตราการเข้ารหัส 1/5, อัตราการเข้ารหัส 1/7, อัตราการเข้ารหัส 2/3, อัตราการเข้ารหัส 2/5, อัตราการเข้ารหัส 2/7 รวมทั้งหมด 7 ปุ่ม

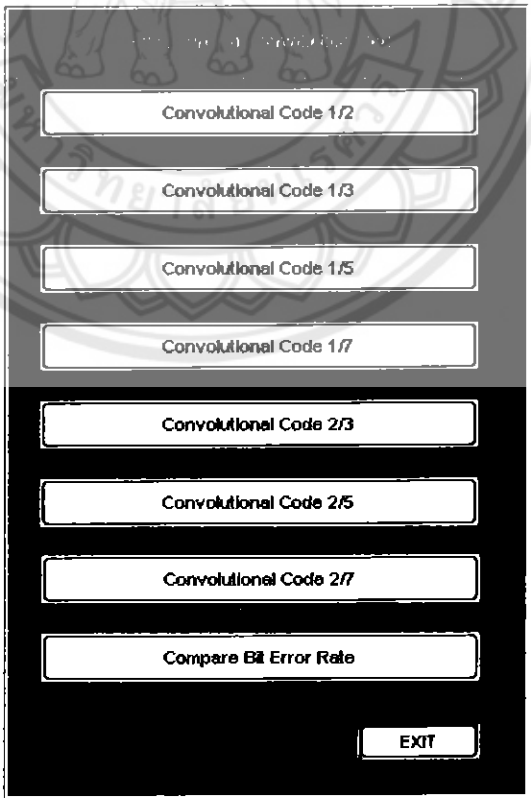
2. ทำการสร้างปุ่ม Push Button ไว้สำหรับเปรียบเทียบประสิทธิภาพในการลดระดับการเกิดข้อผิดพลาด ด้วยอัตราการเข้ารหัสแบบต่าง ๆ

3. ทำการสร้างปุ่ม Push Button ไว้สำหรับออกจากโปรแกรม

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.2 (ก)



(ก)



(ข)

รูปที่ 3.2 Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสช่องสัญญาณ

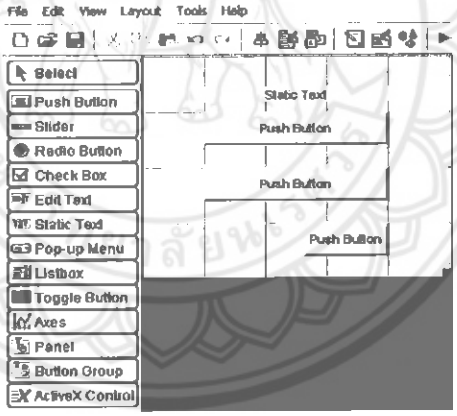
รูปที่ 3.2 (ข) เป็นรูปแบบของ Graphic User Interfaces ในการเลือกอัตราการเข้ารหัสช่องสัญญาณที่เสร็จสมบูรณ์แล้ว ซึ่งเมื่อคลิกที่ปุ่ม Convolutional Code แบบต่าง ๆ ก็จะขึ้นหน้าต่าง Application ขึ้นมาซึ่งแสดงอยู่ในหัวข้อที่ 3.2.2 แต่ถ้าคลิกที่ปุ่ม Compare Bit Error Rate ก็จะขึ้นหน้าต่าง Compare of BER ซึ่งแสดงอยู่ในหัวข้อที่ 3.2.5

3.2.2 Application

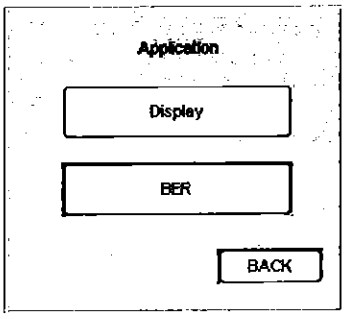
ในขั้นตอนนี้จะทำการสร้างหน้าต่างในการแสดงผลของการเข้ารหัสสัญญาณ Convolutional Code แบบต่าง ๆ มีขั้นตอนดังนี้ คือ

1. ทำการสร้างปุ่ม Push Button เพื่อใช้เลือกที่จะแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด
2. ทำการสร้างปุ่ม Push Button ไว้สำหรับแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด
3. ทำการสร้างปุ่ม Push Button ไว้สำหรับย้อนกลับไปสู่หน้าต่างการเลือกอัตราการเข้ารหัสช่องสัญญาณ

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.3 (ก)



(ก)



(ข)

รูปที่ 3.3 Graphic User Interfaces ในการเลือกแสดงตัวอย่าง หรือคำนวณ Bit Error Rate

จากรูปที่ 3.3 (ข) จะเห็นได้ว่ามี 2 ปุ่มให้เลือก คือ

1. Display เป็นการแสดงตัวอย่างการเข้ารหัสที่กำหนดให้มีจำนวนของบิตข้อมูลที่เป็นสัญญาณในการส่งไม่มากนัก เพื่อใช้ตรวจสอบตัวถอดรหัสสัญญาณที่ปลายทางว่ามีความสามารถในการแก้ไขบิตผิดพลาดหรือไม่ เมื่อทำการคลิกที่ปุ่มนี้แล้วจะนำไปสู่หน้าต่าง Display Convolutional Code ซึ่งแสดงไว้ในหัวข้อที่ 3.2.3

2. BER เป็นการเปรียบเทียบ Bit error rate ของตัวถอดรหัสสัญญาณที่ปลายทาง ระหว่าง Hard Decision และ Soft Decision ในการใช้งานจริง ซึ่งก็หมายความว่าต้องมีบิตข้อมูลจำนวนมากที่ใช้ในการส่งเข้ารหัสช่องสัญญาณ เมื่อทำการคลิกที่ปุ่มนี้แล้วจะนำไปสู่หน้าต่าง BER ซึ่งแสดงไว้ในหัวข้อที่ 3.2.4

3. BACK เป็นปุ่มที่ใช้ในการนำกลับไปสู่หน้าต่างการเลือกอัตราการเข้ารหัสช่องสัญญาณ

3.2.3 Display Convolutional Code

<u>ขั้นตอนการดำเนินงาน</u>

1. ทำการสร้างช่อง Edit Text สำหรับใส่บิตข้อมูลที่มีจำนวนไม่มาก เพื่อใช้ในการแสดงการตรวจสอบตัวถอดรหัสสัญญาณที่ปลายทางว่ามีความสามารถในการแก้ไขบิตผิดพลาดหรือไม่

2. ทำการสร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด ซึ่งก็คือ Probability of bit error ขึ้นในตอนแรก ซึ่งค่านี้จะนำไปคำนวณหาค่า Noise ที่เกิดขึ้นภายในระบบสื่อสาร

3. ทำการสร้างปุ่ม Push Button เพื่อนำค่าที่บิตข้อมูล และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดไปคำนวณ

4. ทำการสร้างช่อง Edit Text สำหรับแสดงข้อมูลดังนี้ คือ

   4.1 บิตข้อมูลที่ได้จากการสุ่ม

   4.2 บิตข้อมูลที่ทำการเข้ารหัส Convolutional Code แล้ว

   4.3 บิตข้อมูลผิดพลาดที่ได้จากการสุ่ม

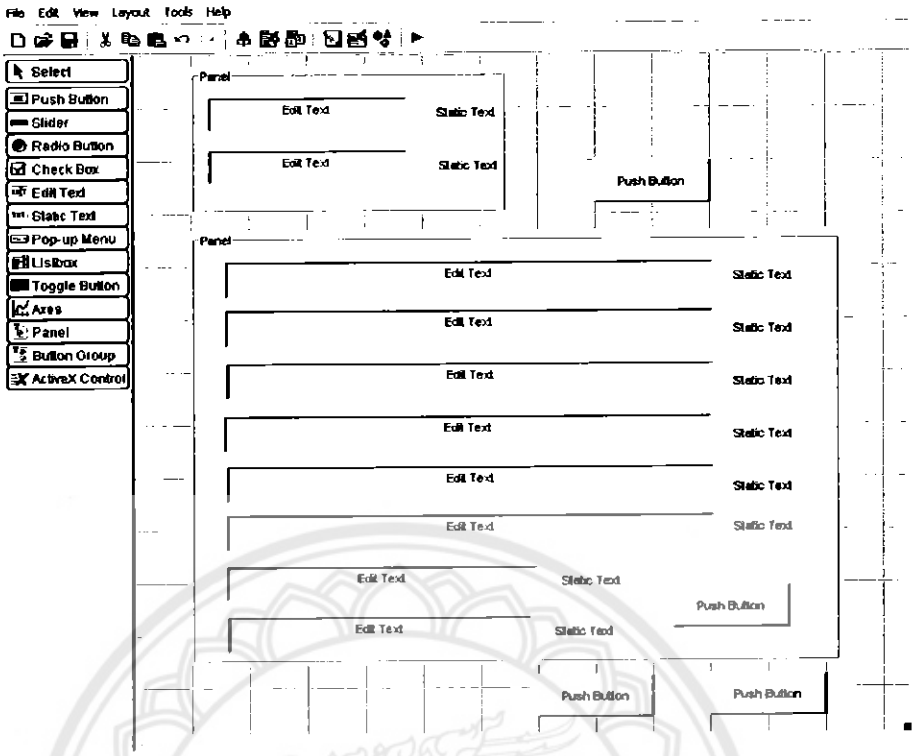
   4.4 บิตข้อมูลที่ได้จากการเข้ารหัส Convolutioanl Code แล้ว รวมกับบิตข้อมูลที่ผิดพลาด

5. ทำการสร้างปุ่ม Push Button สำหรับวาดกราฟเพื่อให้เห็นสัญญาณที่ชัดเจนขึ้น

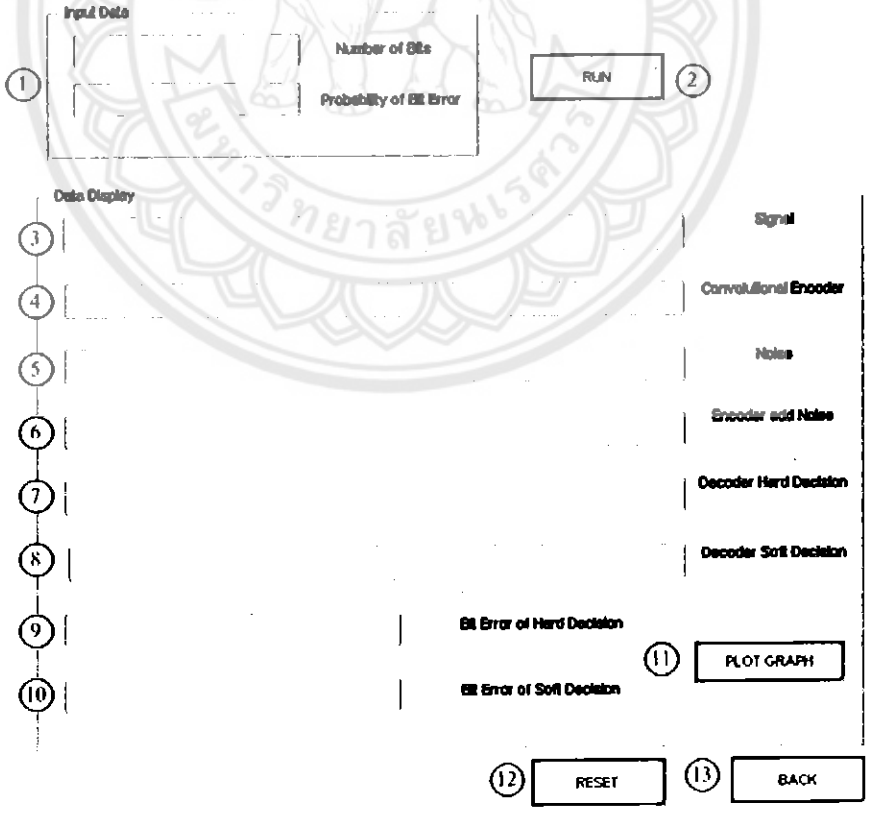
6. ทำการสร้างปุ่ม Push Button สำหรับ Reset ค่า

7. ทำการสร้างปุ่ม Push Button สำหรับ Back ไปสู่หน้าต่าง Application

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.4 (ก)



(ก)



(ข)

รูปที่ 3.4 Graphic User Interfaces ในการแสดงการแก้ไขบิตผิดพลาด

จากรูปที่ 3.4 (ข)

① ช่อง Input Data เป็นส่วนที่ใส่ข้อมูลจำนวนบิตที่ใช้ในการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด

② ปุ่ม Display สำหรับกดรัน เพื่อนำค่าข้อมูลในช่อง Input Data ไปคำนวณ และแสดงค่า

③ แสดงบิตข้อมูลที่ได้จากการสุ่มขึ้นมาตามจำนวน Number of Bit ในช่อง Input Data

④ แสดงบิตข้อมูลที่ได้จากการเข้ารหัส Convolutional Code

⑤ แสดงการสุ่ม Noise

⑥ แสดงบิตข้อมูลที่ได้จากการเข้ารหัส Convolutional Code รวมกับ Noise ซึ่งบิตข้อมูลนี้เป็นบิตข้อมูลที่เกิดข้อผิดพลาดขึ้น

⑦ แสดงการถอดรหัสข้อมูล และแก้ไขข้อมูลผิดพลาดแบบ Hard Decision

⑧ แสดงการถอดรหัสข้อมูล และแก้ไขข้อมูลผิดพลาดแบบ Soft Decision

⑨ แสดง Bit Error Rate ของ Hard Decision เมื่อเทียบกับข้อมูลที่ใช้ส่ง

⑩ แสดง Bit Error Rate ของ Soft Decision เมื่อเทียบกับข้อมูลที่ใช้ส่ง

⑪ ปุ่มสำหรับ Plot graph เพื่อใช้แสดงข้อมูลให้ดูได้ง่ายขึ้น

⑫ ปุ่มสำหรับ Reset ค่าใน Input Data และส่วนที่แสดงบิตข้อมูลต่าง ๆ

⑬ ปุ่มสำหรับ Back กลับไปสู่หน้าต่าง Application (รูปที่ 3.3 (ข))

3.2.4 BER

ขั้นตอนการดำเนินงาน

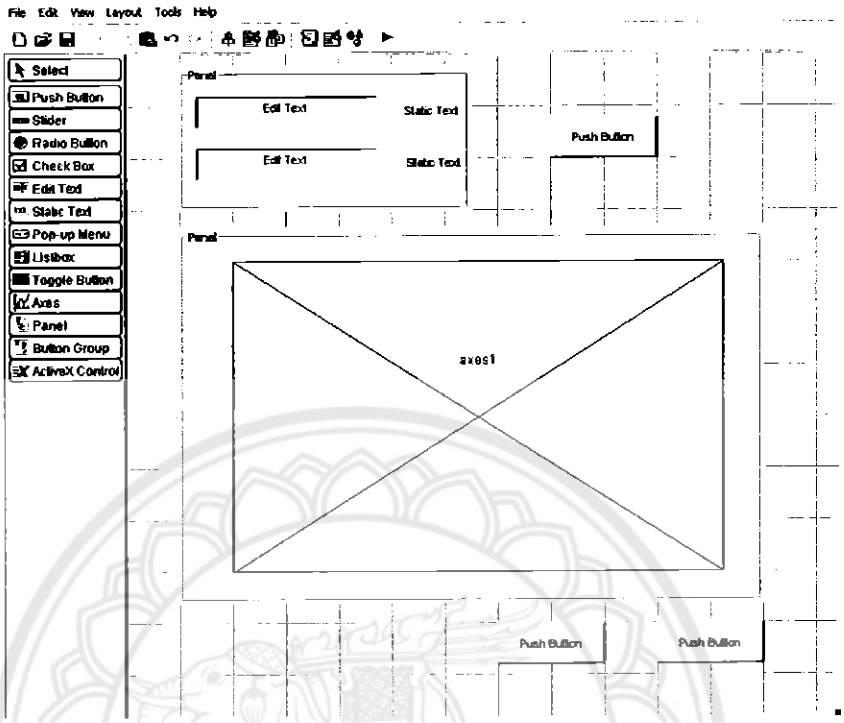
1. ทำการสร้างช่อง Edit Text สำหรับใส่บิตข้อมูล ในส่วนนี้จะเป็นบิตข้อมูลที่ใช้ในการส่งสัญญาณจริง จึงเป็นบิตข้อมูลที่มีจำนวน

2. ทำการสร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด ซึ่งก็คือ Probability of bit error ขึ้นในตอนแรก ซึ่งค่านี้จะนำไปคำนวณหาค่า Noise ที่เกิดขึ้นภายในระบบสื่อสาร

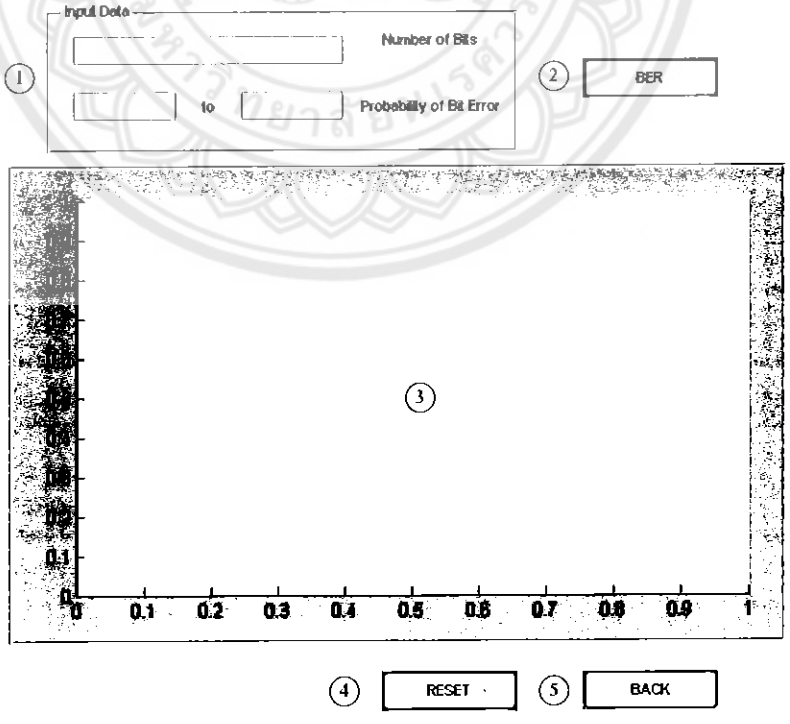
3. ทำการสร้างปุ่ม Push Button เพื่อนำค่าที่บิตข้อมูล และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดไปคำนวณ

4. ทำการสร้าง Axes เพื่อแสดงการเปรียบเทียบประสิทธิภาพในการลดระดับการเกิดข้อผิดพลาดของ Hard Decision และ Soft Decision

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.5 (ก)



(ก)



(ข)

รูปที่ 3.5 Graphic User Interfaces ในการเปรียบเทียบ Bit Error Rate

จากรูปที่ 3.5 (ข)

① ช่อง Input Data เป็นส่วนที่ใส่ข้อมูลจำนวนบิตที่ใช้ในการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด

② ปุ่ม BER สำหรับกดรัน เพื่อนำค่าข้อมูลในช่อง Input Data ไปคำนวณ และแสดงค่าอัตราการลดระดับการเกิดข้อมูลผิดพลาด

③ ส่วนแสดงเส้นกราฟการลดระดับการเกิดข้อมูลผิดพลาด

④ ปุ่มสำหรับ Reset ค่าใน Input Data

⑤ ปุ่มสำหรับ Back กลับไปสู่หน้าต่าง Application (รูปที่ 3.3 (ข))

3.2.5 Compare of BER

ขั้นตอนการดำเนินงาน

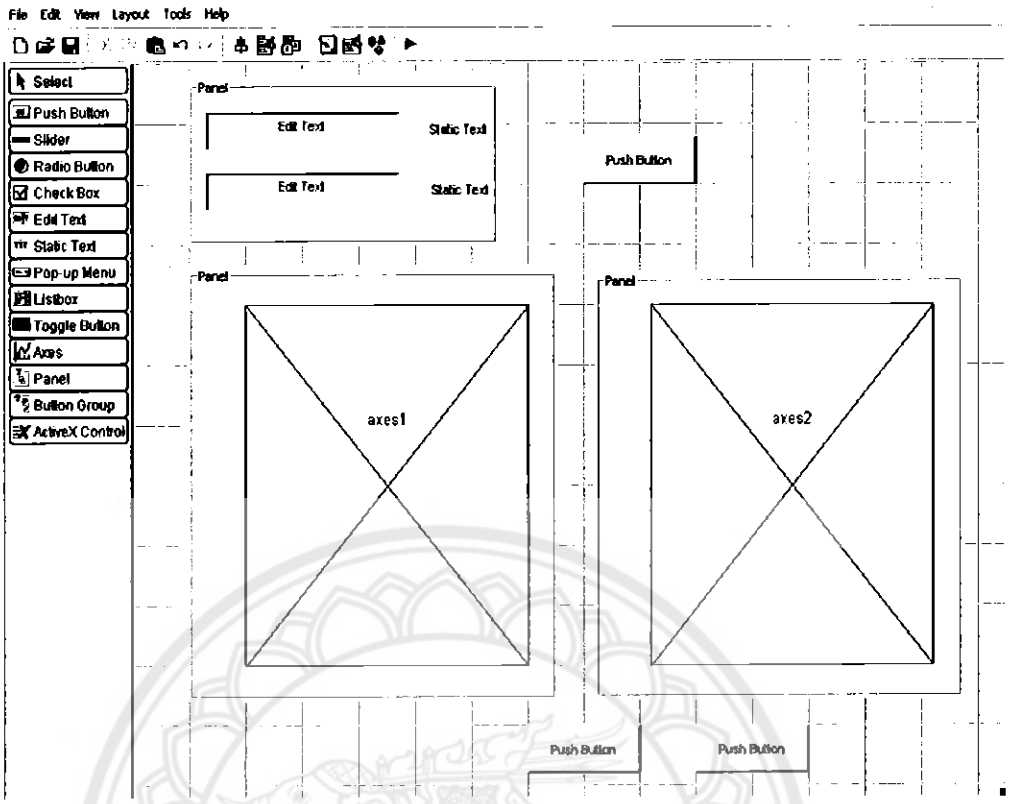
1. ทำการสร้างช่อง Edit Text สำหรับใส่บิตข้อมูล ในส่วนนี้จะเป็นบิตข้อมูลที่ใช้ในการส่งสัญญาณจริง จึงเป็นบิตข้อมูลที่มีจำนวน

2. ทำการสร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด ซึ่งก็คือ Probability of bit error ขึ้นในตอนแรก ซึ่งค่านี้จะนำไปคำนวณหาค่า Noise ที่เกิดขึ้นภายในระบบสื่อสาร

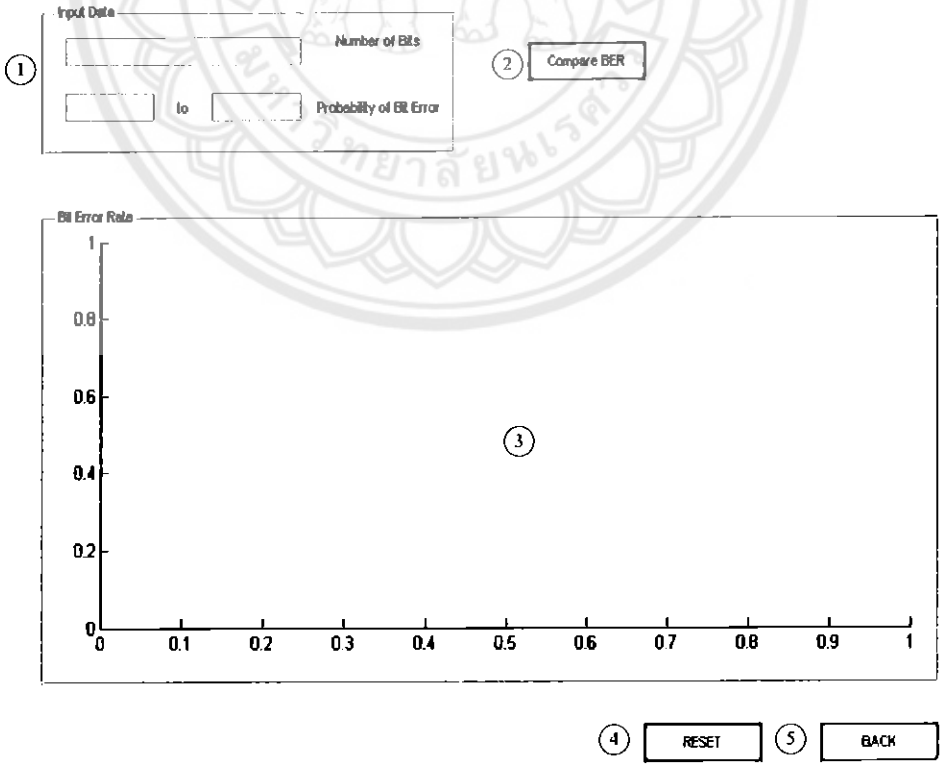
3. ทำการสร้างปุ่ม Push Button เพื่อนำค่าที่บิตข้อมูล และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดไปคำนวณ

4. ทำการสร้าง Axes เพื่อแสดงการเปรียบเทียบประสิทธิภาพในการลดระดับการเกิดข้อผิดพลาดของ Hard Decision ในอัตราการเข้ารหัสแบบต่าง ๆ ซึ่งได้แก่ อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 1/3, อัตราการเข้ารหัส 1/5, อัตราการเข้ารหัส 1/7, อัตราการเข้ารหัส 2/3, อัตราการเข้ารหัส 2/5, อัตราการเข้ารหัส 2/7 และ Soft Decision ในอัตราการเข้ารหัสแบบต่าง ๆ ซึ่งได้แก่ อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 1/3, อัตราการเข้ารหัส 1/5, อัตราการเข้ารหัส 1/7, อัตราการเข้ารหัส 2/3, อัตราการเข้ารหัส 2/5, อัตราการเข้ารหัส 2/7 เพื่อเป็นการเปรียบเทียบว่าอัตราการเข้ารหัสในแบบใดที่จะมีความสามารถในการแก้ไขบิตที่ผิดพลาด (ลดอัตราการเกิดข้อผิดพลาด) ได้ดีกว่ากัน

ซึ่งขั้นตอนดังกล่าวข้างต้นได้แสดงไว้ดังรูปที่ 3.6 (ก)



(ก)



(ข)

รูปที่ 3.6 Graphic User Interfacesในการเปรียบเทียบ Bit Error Rateของอัตราการเข้ารหัสแบบต่าง ๆ

จากรูปที่ 3.6 (ข)

① ช่อง Input Data เป็นส่วนที่ใส่ข้อมูลจำนวนบิตที่ใช้ในการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด

② ปุ่ม BER สำหรับกดรัน เพื่อนำค่าข้อมูลในช่อง Input Data ไปคำนวณ และแสดงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัส

③ ส่วนแสดงกราฟเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด โดยใช้การถอดรหัสและการแก้ไขข้อมูลที่ผิดพลาดแบบ

④ ปุ่มสำหรับ Reset ค่าใน Input Data

⑤ ปุ่มสำหรับ Back กลับไปสู่หน้าต่าง Application (รูปที่ 3.3 (ข))

ในบทที่ 3 นี้ ได้แสดงหลักการทำงานของโปรแกรม และได้อธิบายถึงการสร้าง Graphic User Interfaces เพื่อใช้ในการแสดงการทำงานของโปรแกรม ในรูปแบบที่เข้าใจง่าย และสะดวกต่อการใช้งาน ซึ่งสามารถแบ่งได้เป็น

1. หน้าต่างที่ใช้แสดงการเลือกอัตราการเข้ารหัสช่องสัญญาณ
2. หน้าต่างที่ใช้แสดงการแก้ไขข้อมูลผิดพลาด
3. หน้าต่างที่ใช้แสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด
4. หน้าต่างที่ใช้แสดงการเปรียบเทียบประสิทธิภาพในการลดระดับอัตราการเกิดข้อมูลผิดพลาด ในแต่ละอัตราการเข้ารหัสช่องสัญญาณ

ในบทต่อไปจะแสดงผลการทดลองของโปรแกรมการคำนวณ และแสดงค่าต่าง ๆ ที่ได้สร้างไว้แล้วในบทนี้ โดยที่ทดลองใส่ค่าบิตข้อมูลไปในโปรแกรม เพื่อให้แสดงค่าต่าง ๆ ที่ต้องการ รวมถึงกราฟที่แสดงการลดระดับอัตราการเกิดข้อผิดพลาดด้วย

บทที่ 4
ผลการดำเนินโครงงาน

ในบทนี้จะทำแสดงผลการทดลองในการใส่บิตข้อมูลเข้าไปในโปรแกรมจำลองการเข้ารหัสช่องสัญญาณที่ได้สร้างไว้ โดยใช้การเข้ารหัสแบบ Convolutional Code และการถอดรหัสแบบ Viterbi Decoder ซึ่งสามารถแสดงการแก้ไขข้อมูลผิดพลาด การลดระดับอัตราการเกิดข้อมูลผิดพลาด และการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ในแต่ละอัตราการเข้ารหัสช่องสัญญาณ โดยที่จะแสดงผลการทดลองนี้ในรูปแบบของ Graphic User Interfaces

4.1 โปรแกรมแสดงการแก้ไขข้อมูลผิดพลาด

4.1.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1. ในขั้นตอนแรกต้องทำการเปิดหน้าต่างการเลือกอัตราการเข้ารหัสชนิดต่าง ๆ ซึ่งได้แก่

1.1 Convolutional Code 1/2 (อัตราการเข้ารหัสช่องสัญญาณ 1/2)

1.2 Convolutional Code 1/3 (อัตราการเข้ารหัสช่องสัญญาณ 1/3)

1.3 Convolutional Code 1/5 (อัตราการเข้ารหัสช่องสัญญาณ 1/5)

1.4 Convolutional Code 1/7 (อัตราการเข้ารหัสช่องสัญญาณ 1/7)

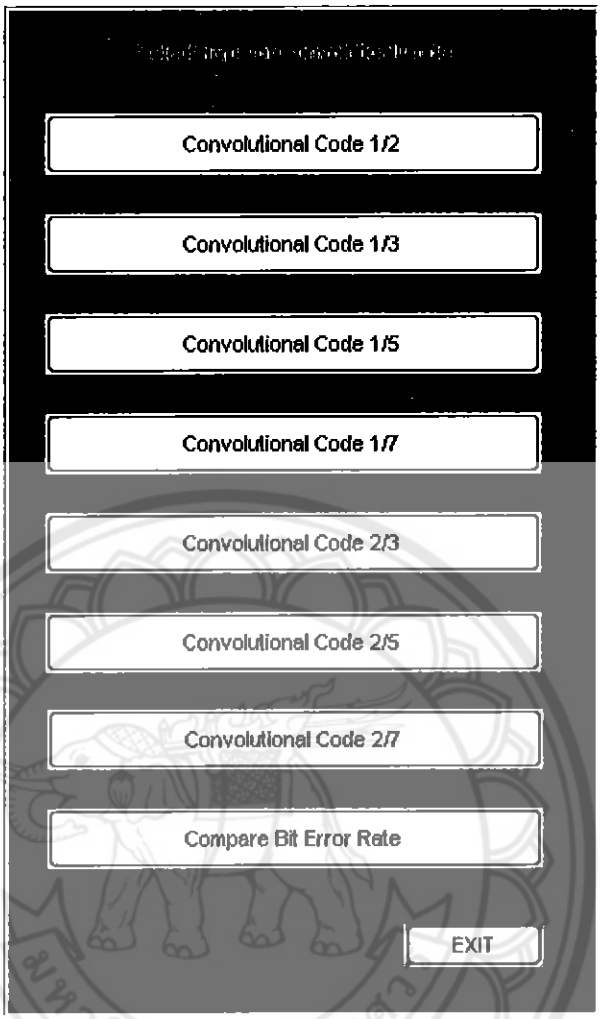
1.5 Convolutional Code 2/3 (อัตราการเข้ารหัสช่องสัญญาณ 2/3)

1.6 Convolutional Code 2/5 (อัตราการเข้ารหัสช่องสัญญาณ 2/5)

1.7 Convolutional Code 2/7 (อัตราการเข้ารหัสช่องสัญญาณ 2/7)

1.8 Compare Bit Error Rate

มีอัตราการเข้ารหัสช่องสัญญาณให้เลือก 7 ชนิด ดังรูปที่ 4.1



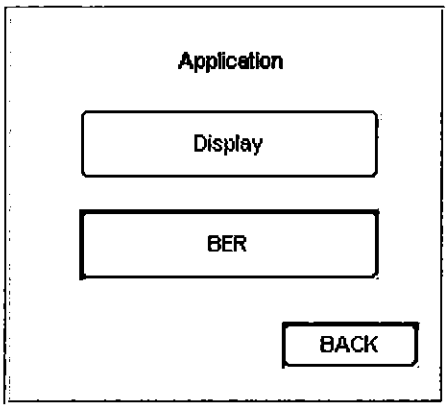
รูปที่ 4.1 หน้าต่างสำหรับเลือกอัตราการเข้ารหัสช่องสัญญาณชนิดต่าง ๆ

2. เมื่อทำการเลือกโดยคลิกที่อัตราการเข้ารหัสช่องสัญญาณที่ต้องการได้แล้ว จะทำให้หน้าต่าง Application ปรากฏขึ้นมา ซึ่งหน้าต่างนี้จะมีปุ่มให้เลือกขึ้นมา 2 ปุ่มได้แก่

2.1 Display คือ หน้าต่างแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด

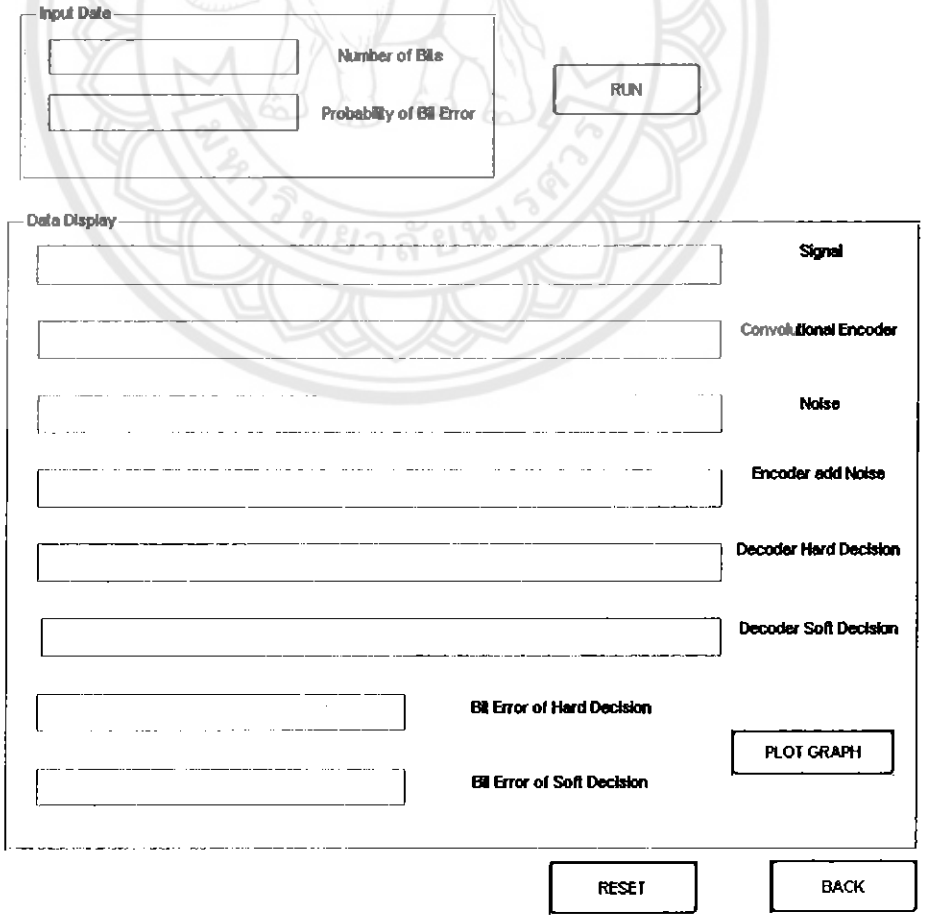
2.2 BER คือ หน้าต่างแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด

ซึ่งหน้าต่าง Application ได้แสดงไว้ในรูปที่ 4.2



รูปที่ 4.2 หน้าต่าง Application ใช้ในการเลือกระหว่างหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด และหน้าต่างแสดงการลดอัตราการเกิดข้อมูลผิดพลาด

3. กดคลิกที่ Display เพื่อเลือกหน้าต่างแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด เมื่อคลิกแล้วจะปรากฏหน้าต่างแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาดขึ้น ดังรูปที่ 4.3



รูปที่ 4.3 หน้าต่างแสดงการแก้ไขบิตผิดพลาด

4. เมื่อปรากฏหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาดแล้ว จากนั้นทำการใส่ค่า จำนวนข้อมูลที่ต้องการใช้ในการส่ง (ในที่นี้เป็นข้อมูลที่มีจำนวนไม่มากนัก) และเลือกความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error)

4.1.2 ตัวอย่างการแก้ไขข้อมูลผิดพลาด

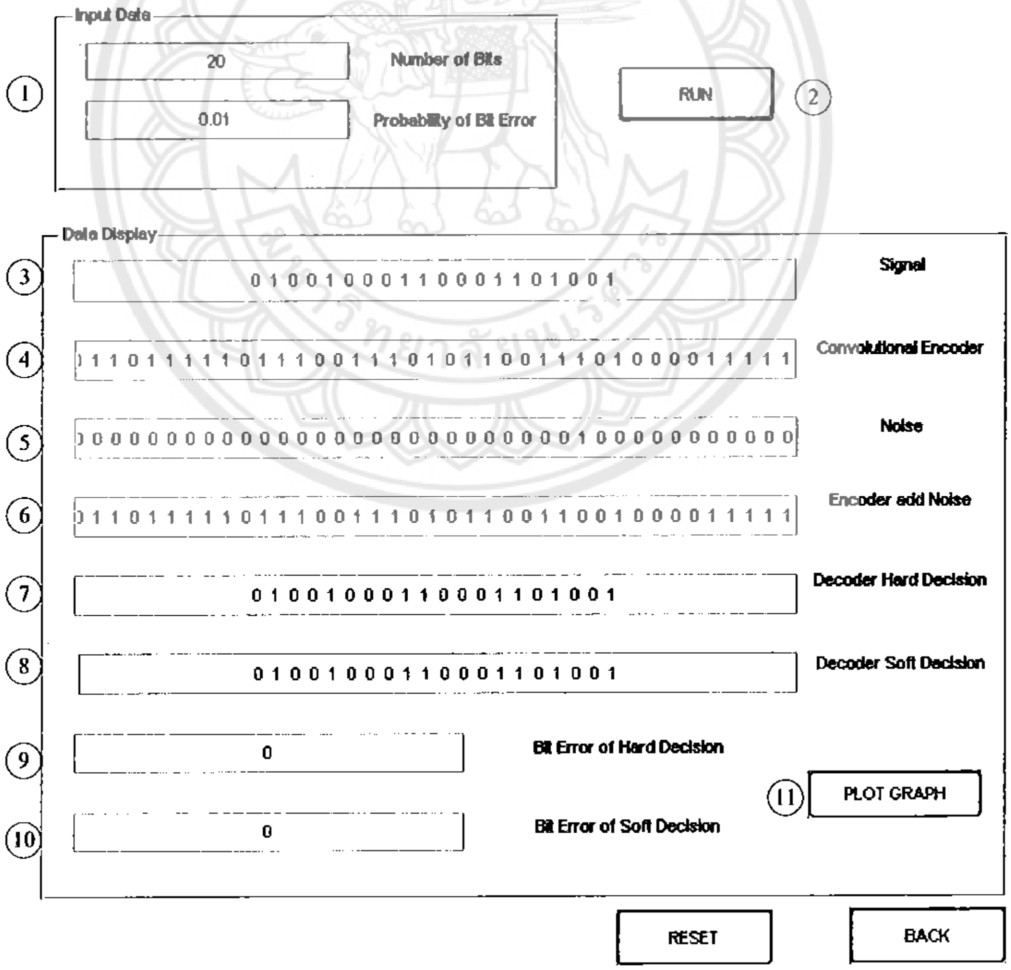
1. อัตราการเข้ารหัสช่องสัญญาณ 1/2

1.1 คลิกเลือก Convolutional Code 1/2 (อัตราการเข้ารหัส 1/2) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

1.2 จากนั้นคลิกที่ Display จากหน้าต่าง Application ดังรูปที่ 4.2

1.3 จะปรากฏหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด ดังรูปที่ 4.3

1.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 20 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.01 จะได้ค่าต่าง ๆ ดังรูปที่ 4.4



รูปที่ 4.4 ตัวอย่างในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัส 1/2

จากรูปที่ 4.4

① ช่อง Input Data ใส่ค่าบิตที่ต้องการส่งจำนวน 20 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.01

② ปุ่ม RUN จะนำค่าบิตที่ต้องการส่งจำนวน 20 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.01 เข้าสู่โปรแกรมการคำนวณ และแสดงผล

③ สุ่มสัญญาณที่ใช้ในการส่งจำนวน 20 บิต

④ นำสัญญาณ 20 บิตในช่องที่ ③ มาเข้ารหัส Convolution Code จะได้ออกมาเป็น 40 บิต เนื่องจากว่าอัตราการเข้ารหัสที่ใช้ตอนนี้เป็น 1/2

⑤ สุ่ม Noise ขึ้นมาให้มีจำนวนบิตเท่ากับ สัญญาณที่เข้ารหัส Convolutional Code แล้ว คือ 40 บิต

⑥ สัญญาณที่เข้ารหัส Convolutional Code แล้วรวมกับ Noise เพื่อทำให้เกิดข้อผิดพลาด

⑦ การถอดรหัสแบบ Hard Decision ทำให้ได้สัญญาณเดิมที่ส่งกลับมา

⑧ การถอดรหัสแบบ Soft Decision ทำให้ได้สัญญาณเดิมที่ส่งกลับมา

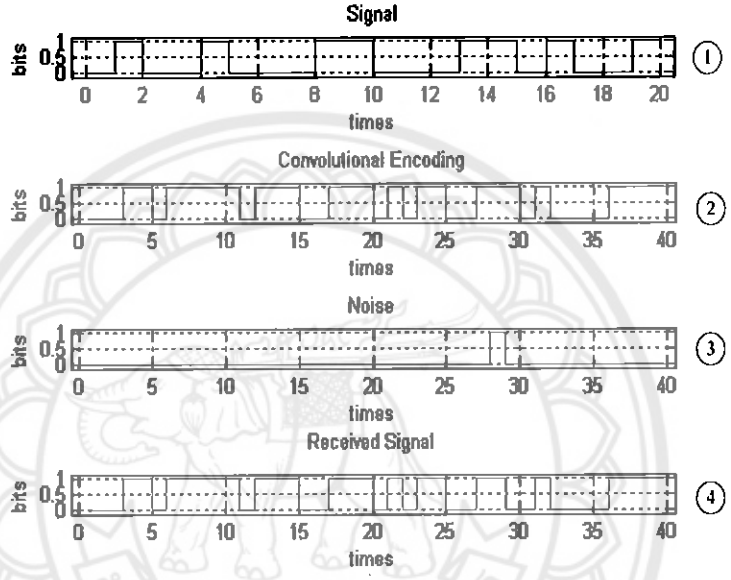
⑨ อัตราการเกิดข้อผิดพลาดเมื่อเทียบสัญญาณที่ส่งกับตัวถอดรหัส Hard Decision จะเห็นได้ว่าไม่มีการเกิดข้อผิดพลาดเลย

⑩ อัตราการเกิดข้อผิดพลาดเมื่อเทียบสัญญาณที่ส่งกับตัวถอดรหัส Soft Decision จะเห็นได้ว่าไม่มีการเกิดข้อผิดพลาดเลย

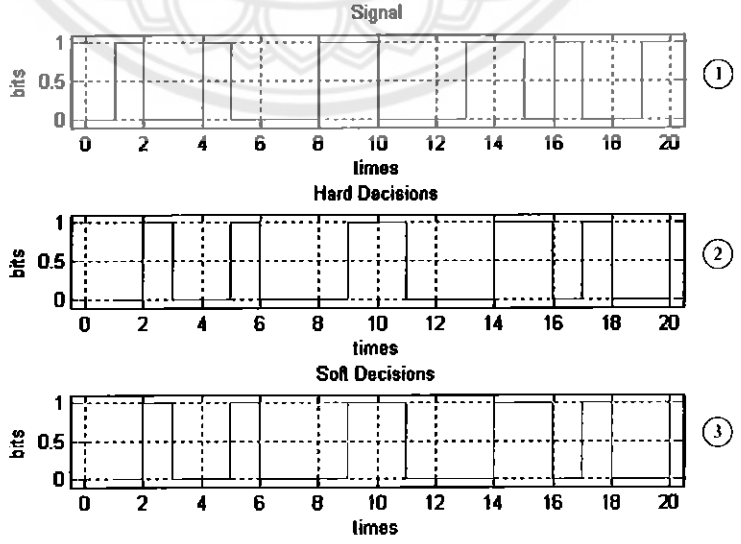
⑪ ปุ่ม Plot Graph เมื่อกดปุ่มนี้จะแสดงกราฟขึ้นมา ดังรูปที่ 4.5 (ก) และ 4.5 (ข)

จะเห็นได้ว่าเมื่อใส่ค่าบิตข้อมูลที่จะทำการส่ง 20 บิต และค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.01 ในช่อง Input Data แล้วทำการคำนวณค่าต่าง ๆ นี้ในโปรแกรมโดยการกดปุ่ม RUN จะได้ค่าบิตข้อมูลที่สุ่มขึ้นมา 20 ค่าดังแสดงไว้ในช่องที่ ③ แล้วก็นำบิตข้อมูลเหล่านี้มาเข้ารหัสช่องสัญญาณคอนโวลูชันซึ่งจะได้บิตข้อมูลออกมา 40 บิตดังแสดงไว้ในช่องที่ ④ ที่ได้ข้อมูลออกมา 40 บิต เนื่องจากอัตราการเข้ารหัสที่เราเลือกใช้ในขณะนี้มีค่าเท่ากับ 1/2 หลังจากนั้นก็ทำการสุ่มค่า Noise ออกมา 40 ค่าดังแสดงในช่องที่ ⑤ แล้วเราก็จะนำสัญญาณที่ผ่านการเข้ารหัสมารวมกับ Noise เพื่อให้ได้ค่าสัญญาณที่เกิดข้อผิดพลาดดังแสดงไว้ในช่องที่ ⑥ ซึ่งจะเห็นได้ว่าจะมีบิตที่แตกต่างจะสัญญาณที่เข้ารหัสแล้วในตำแหน่งที่ 28 แล้วหลังจากนั้นก็จะนำสัญญาณนี้ไปทำการถอดรหัส และ

แก้ไขข้อมูลที่ผิดพลาด โดยใช้วิธีการถอดรหัส 2 แบบ คือ Hard Decision และ Soft Decision ดังแสดงไว้ในช่องที่ ⑦ และ ⑧ ตามลำดับ จะเห็นได้ว่าข้อมูลที่ทำการถอดรหัสแล้วเหมือนกับข้อมูลที่ส่งทุกประการ ซึ่งก็แสดงว่าสามารถแก้ไขข้อมูลผิดพลาดได้ จึงทำให้อัตราการเกิดข้อมูลผิดพลาดในช่องที่ ⑨ และ ⑩ มีค่าเป็น 0 และเมื่อเรากดปุ่ม PLOT GRAPH ก็จะได้เส้นกราฟที่แสดงค่าต่าง ๆ ที่ได้กล่าวไว้ข้างต้น ดังรูปที่ 4.5 (ก) และ 4.5 (ข)



(ก)



(ข)

รูปที่ 4.5 กราฟแสดงบิตข้อมูลอัตราการเข้ารหัส 1/2

จากรูปที่ 4.5 (ก)

① บิตข้อมูลที่ใช้ในการส่งจำนวน 20 บิตที่ได้จากการสุ่ม

② นำบิตข้อมูลจำนวน 20 บิต มาเข้ารหัส Convolutioanl Code จะมีจำนวนบิตข้อมูลเท่ากับ 40 บิต เนื่องจากมีอัตราการเข้ารหัสช่องสัญญาณแบบ 1/2

③ Noise ที่ได้จากการสุ่ม โดยที่จะมีจำนวนบิตเท่ากับข้อมูลที่เข้ารหัส Convolutional Code แล้ว คือ 40 บิต

④ เป็นบิตข้อมูลที่นำข้อมูลที่เข้ารหัส Convolutioanl Code แล้วรวมกับ Noise เพื่อให้เกิดข้อผิดพลาด ก่อนทำการถอดรหัส

จากรูปที่ 4.5 (ข)

① แสดงบิตข้อมูลที่ใช้ในการส่ง จำนวน 20 บิต

② แสดงบิตข้อมูลที่ทำการถอดรหัสแล้ว โดยใช้วิธี Hard Decision

③ แสดงบิตข้อมูลที่ทำการถอดรหัสแล้ว โดยใช้วิธี Soft Decision

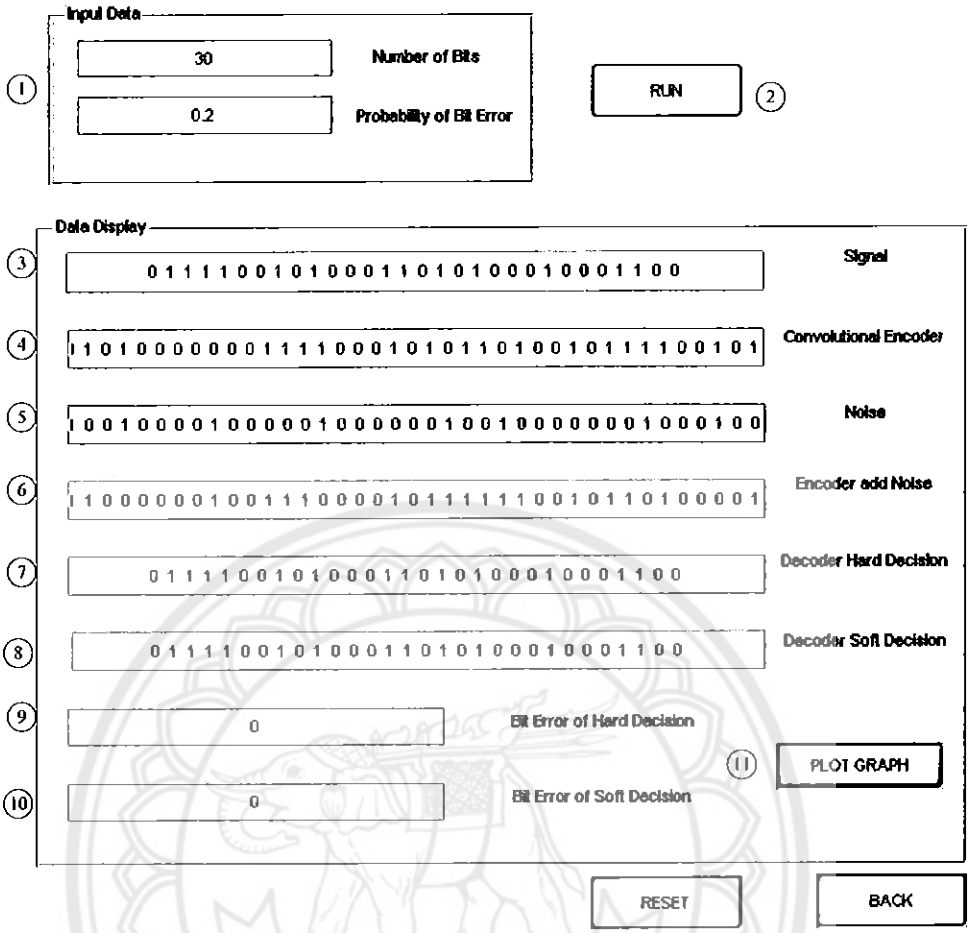
2. อัตราการเข้ารหัสช่องสัญญาณ 1/7

2.1 คลิกเลือก Convolutional Code 1/7 (อัตราการเข้ารหัส 1/7) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

2.2 จากนั้นคลิกที่ Display จากหน้าต่าง Application ดังรูปที่ 4.2

2.3 จะปรากฏหน้าต่างแสดงการแก้ไขข้อมูลผิดพลาด ดังรูปที่ 4.3

2.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 30 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.01 จะได้ค่าต่าง ๆ ดังรูปที่ 4.6



รูปที่ 4.6 ตัวอย่างในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัส 1/7

จากรูปที่ 4.6

(1) ช่อง Input Data ใส่ค่าบิตที่ต้องการส่งจำนวน 30 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.1

(2) ปุ่ม RUN จะนำค่าบิตที่ต้องการส่งจำนวน 30 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.1 เข้าสู่โปรแกรมการคำนวณ และแสดงผล

(3) สุ่มสัญญาณที่ใช้ในการส่งจำนวน 30 บิต

(4) นำสัญญาณ 20 บิตในช่องที่ (3) มาเข้ารหัส Convolution Code จะได้ออกมาเป็น 210 บิต เนื่องจากว่าอัตราการเข้ารหัสที่ใช้ตอนนี้เป็น 1/7

(5) สุ่ม Noise ขึ้นมาให้มีจำนวนบิตเท่ากับ สัญญาณที่เข้ารหัส Convolutional Code แล้ว คือ 210 บิต

(6) สัญญาณที่เข้ารหัส Convolutional Code แล้วรวมกับ Noise เพื่อทำให้เกิดข้อผิดพลาด

⑦ การถอดรหัสแบบ Hard Decision ทำให้ได้สัญญาณเดิมที่ส่งกลับมา

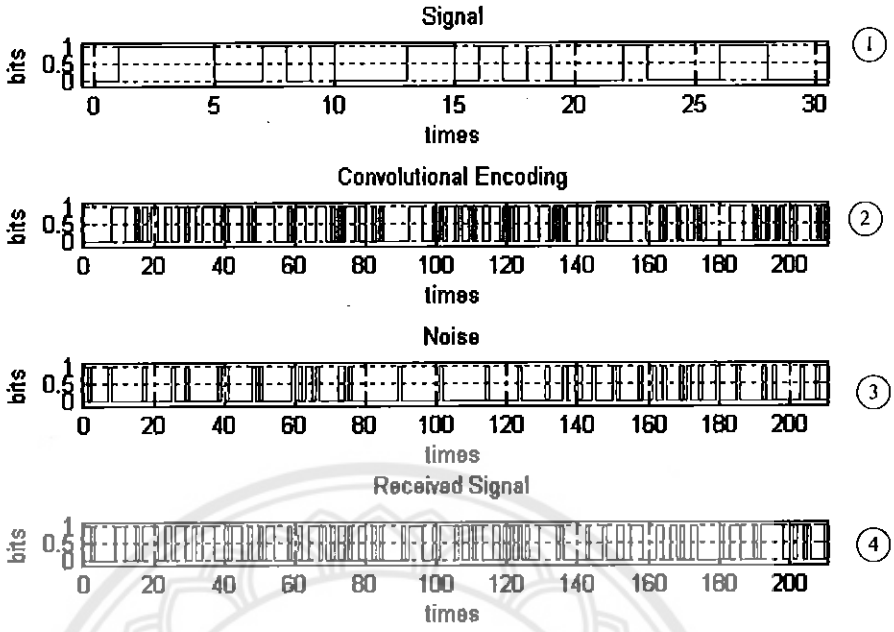
⑧ การถอดรหัสแบบ Soft Decision ทำให้ได้สัญญาณเดิมที่ส่งกลับมา

⑨ อัตราการเกิดข้อผิดพลาดเมื่อเทียบสัญญาณที่ส่งกับตัวถอดรหัส Hard Decision จะเห็นได้ว่าไม่มีการเกิดข้อผิดพลาดเลย

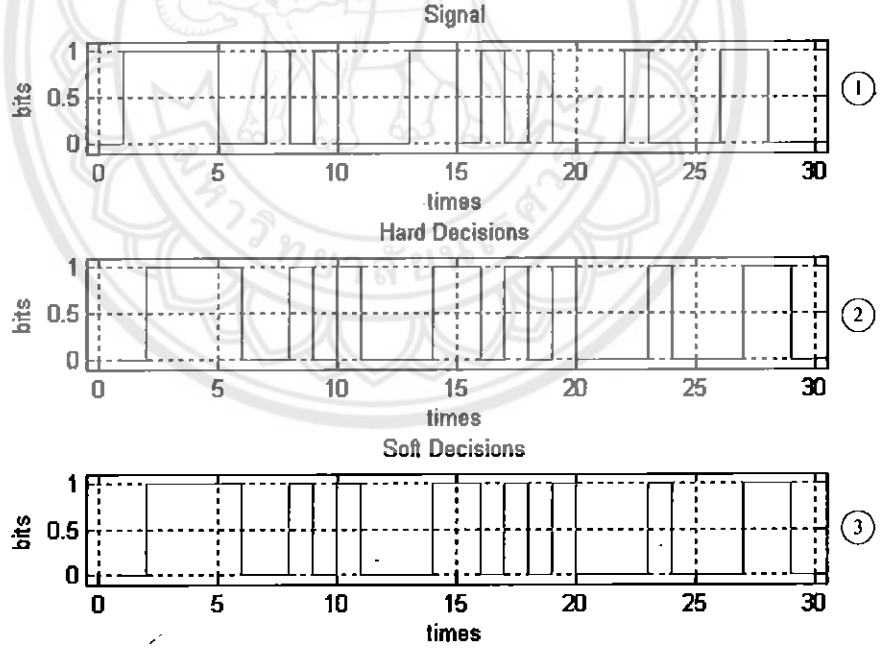
⑩ อัตราการเกิดข้อผิดพลาดเมื่อเทียบสัญญาณที่ส่งกับตัวถอดรหัส Soft Decision จะเห็นได้ว่าไม่มีการเกิดข้อผิดพลาดเลย

⑪ ปุ่ม Plot Graph เมื่อกดปุ่มนี้จะแสดงกราฟขึ้นมา ดังรูปที่ 4.7 (ก) และ 4.7 (ข)

เช่นเดียวกันกับอัตราการเข้ารหัสช่องสัญญาณ 1/2 เมื่อใส่ค่าจำนวนบิตข้อมูล 30 บิต และค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดเท่ากับ 0.2 ในช่อง Input Data แล้วทำการคำนวณค่าเหล่านี้ในโปรแกรมโดยการกดปุ่ม RUN จะได้ค่าบิตข้อมูลที่สุ่มขึ้นมา 30 บิตดังแสดงในช่องที่ ③ หลังจากนั้นก็จะนำบิตข้อมูลนี้ไปเข้ารหัสช่องสัญญาณแบบคอนโวลูชันที่มีอัตราการเข้ารหัส 1/7 ทำให้ได้บิตข้อมูลออกมา 210 บิต ดังแสดงไว้ในช่องที่ ④ ต่อไปก็จะทำการสุ่ม Noise ขึ้นมาจำนวน 210 บิตเท่ากัน ดังแสดงไว้ในช่องที่ ⑤ หลังจากนั้นจะนำสัญญาณที่ผ่านการเข้ารหัสแล้วมารวมกับ Noise เพื่อให้ได้สัญญาณที่เกิดข้อมูลผิดพลาดดังแสดงไว้ในช่องที่ ⑥ จากนั้นก็จะนำข้อมูลนี้ไปทำการถอดรหัสสัญญาณแบบ Hard Decision และ Soft Decision เมื่อถอดรหัสออกมาแล้วก็จะได้บิตข้อมูลที่มีลักษณะเหมือนกับบิตข้อมูลที่ทำการส่งทุกประการดังแสดงไว้ในช่องที่ ⑦ และ ⑧ ตามลำดับ ซึ่งแสดงให้เห็นว่าสามารถแก้ไขข้อมูลผิดพลาดได้ จึงทำให้อัตราการเกิดข้อมูลผิดพลาดในช่องที่ ⑨ และ ⑩ เป็น 0 และเมื่อกดปุ่ม PLOT GRAPH ก็จะได้เส้นกราฟที่แสดงค่าต่าง ๆ ที่ได้กล่าวไว้ข้างต้น ดังรูปที่ 4.7 (ก) และ 4.7 (ข)



(ก)



(ข)

รูปที่ 4.7 กราฟแสดงบิตข้อมูลอัตราการเข้ารหัส 1/7

จากรูปที่ 4.7 (ก)

① บิตข้อมูลที่ใช้ในการส่งจำนวน 30 บิตที่ได้จากการสุ่ม

② นำบิตข้อมูลจำนวน 30 บิต มาเข้ารหัส Convolutioanl Code จะมีจำนวนบิตข้อมูลเท่ากับ 210 บิต เนื่องจากมีอัตราการเข้ารหัสช่องสัญญาณแบบ 1/7

③ Noise ที่ได้จากการสุ่ม โดยที่จะมีจำนวนบิตเท่ากับข้อมูลที่เข้ารหัส Convolutional Code แล้ว คือ 210 บิต

④ เป็นบิตข้อมูลที่นำข้อมูลที่เข้ารหัส Convolutioanl Code แล้วรวมกับ Noise เพื่อให้เกิดข้อผิดพลาด ก่อนทำการถอดรหัส

จากรูปที่ 4.7 (ข)

① แสดงบิตข้อมูลที่ใช้ในการส่ง จำนวน 30 บิต

② แสดงบิตข้อมูลที่ทำการถอดรหัสแล้ว โดยใช้วิธี Hard Decision

③ แสดงบิตข้อมูลที่ทำการถอดรหัสแล้ว โดยใช้วิธี Soft Decision

สำหรับในอัตราการเข้ารหัสช่องสัญญาณแบบอื่น ๆ ได้แก่ อัตราการเข้ารหัสช่องสัญญาณ 1/3, อัตราการเข้ารหัสช่องสัญญาณ 1/5, อัตราการเข้ารหัสช่องสัญญาณ 2/3, อัตราการเข้ารหัสช่องสัญญาณ 2/5 และอัตราการเข้ารหัสช่องสัญญาณ 2/7 ก็จะได้หน้าต่างการแสดงการแก้ไขข้อมูลผิดพลาดในทำนองเดียวกัน

4.2 โปรแกรมแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด

4.2.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1. ในขั้นตอนแรกต้องทำการรันโปรแกรมการเลือกชนิดอัตราการเข้ารหัสช่องสัญญาณ ซึ่งมีอัตราการเข้ารหัสช่องสัญญาณที่แตกต่างกัน 7 ชนิด ได้แก่

1.1 Convolutional Code 1/2 (อัตราการเข้ารหัสช่องสัญญาณ 1/2)

1.2 Convolutional Code 1/3 (อัตราการเข้ารหัสช่องสัญญาณ 1/3)

1.3 Convolutional Code 1/5 (อัตราการเข้ารหัสช่องสัญญาณ 1/5)

1.4 Convolutional Code 1/7 (อัตราการเข้ารหัสช่องสัญญาณ 1/7)

1.5 Convolutional Code 2/3 (อัตราการเข้ารหัสช่องสัญญาณ 2/3)

1.6 Convolutional Code 2/5 (อัตราการเข้ารหัสช่องสัญญาณ 2/5)

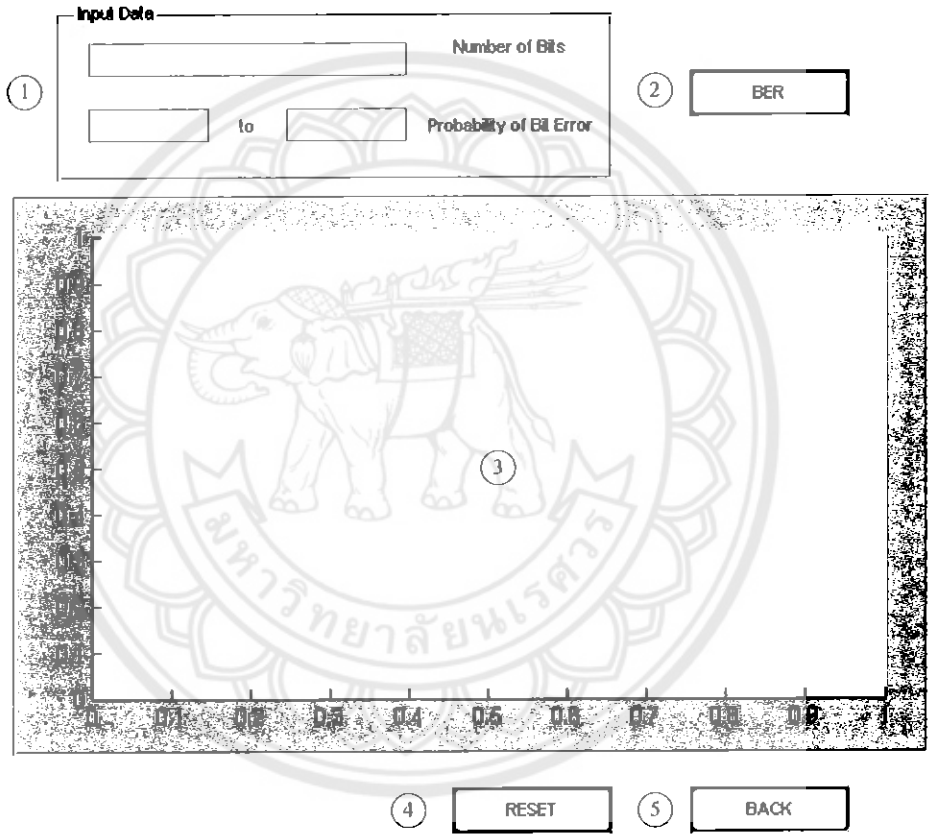
1.7 Convolutional Code 2/7 (อัตราการเข้ารหัสช่องสัญญาณ 2/7)

1.8 Compare Bit Error Rate

มีอัตราการเข้ารหัสช่องสัญญาณให้เลือก 7 ชนิด ดังรูปที่ 4.1

2. เมื่อเลือกอัตราการเข้ารหัสช่องสัญญาณที่ต้องการ โดยทำการคลิกที่ปุ่ม Convolutioanl Code

3. จากนั้นจะมีหน้าต่าง Application ดังรูปที่ 4.2 ปรากฏขึ้นมาให้คลิกที่ปุ่ม BER ซึ่งจะเป็นหน้าต่างการลดระดับอัตราการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8



รูปที่ 4.8 หน้าต่างแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด

จากรูปที่ 4.8

① ช่อง Input Data เป็นส่วนที่ใส่ข้อมูลจำนวนบิตที่ใช้ในการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด

② ปุ่ม BER สำหรับกดรัน เพื่อนำค่าข้อมูลในช่อง Input Data ไปคำนวณ และแสดงค่าอัตราการลดระดับการเกิดข้อมูลผิดพลาด

③ ส่วนแสดงเส้นกราฟการลดระดับการเกิดข้อมูลผิดพลาด

④ ปุ่มสำหรับ Reset ค่าใน Input Data

⑤ ปุ่มสำหรับ Back กลับไปสู่หน้าต่าง Application (รูปที่ 4.2)

4. เมื่อปรากฏหน้าต่างแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด จากนั้นทำการใส่ค่าจำนวนข้อมูลที่ต้องการใช้ในการส่ง (ซึ่งเป็นบิตข้อมูลที่ใช้ในการส่งจริง นั่นคือ มีปริมาณบิตข้อมูลจำนวนมาก) และเลือกความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error)

4.2.2 ตัวอย่างการลดระดับการเกิดข้อมูลผิดพลาด

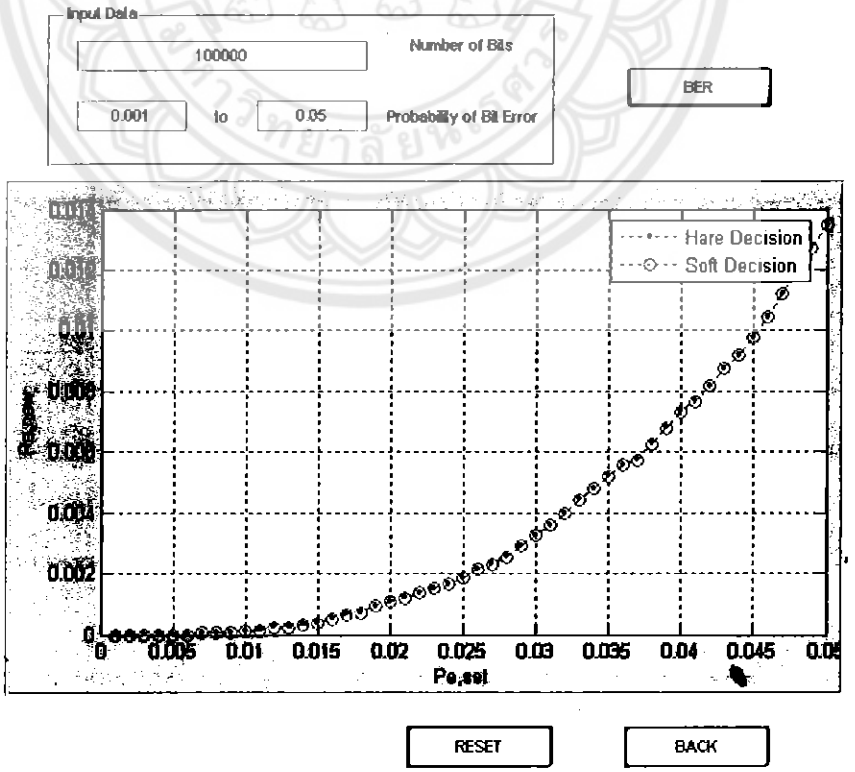
1. อัตราการเข้ารหัสช่องสัญญาณ 1/2 (Covolutioanl Code 1/2)

1.1 คลิกเลือก Convolutional Code 1/2 (อัตราการเข้ารหัส 1/2) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

1.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

1.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

1.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.9



รูปที่ 4.9 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/2

จากรูปที่ 4.9 จะเห็นได้ว่า เมื่อทำการส่งข้อมูลจำนวน 100000 บิต และกำหนดค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 แล้ว จากกราฟจะเห็นได้ว่าในช่วงค่าเริ่มต้นของ Pe,set ซึ่งก็คือ ค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่กำหนดไปตั้งแต่ตอนต้นนั้น จะทำให้ได้ค่า Pe,new ซึ่งคือ ค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่ผ่านโปรแกรมการเข้ารหัสเพื่อแก้ไขข้อมูลผิดพลาดแล้ว มีค่าลดลงจนเหลือศูนย์ ซึ่งก็หมายความว่าสามารถที่จะแก้ไขข้อมูลที่ผิดพลาดได้ ไม่เกิดการ error นั่นเอง แต่อย่างไรก็ตาม เมื่อค่า Pe,set มีค่ามากขึ้น ก็จะทำให้ค่า Pe,new เพิ่มขึ้นตามด้วย นั่นก็หมายความว่าถ้ามีความน่าจะเป็นในการเกิดข้อมูลผิดพลาดมาก การเข้ารหัสสัญญาณก็ไม่สามารถแก้ไขข้อมูลผิดพลาดได้ทั้งหมด

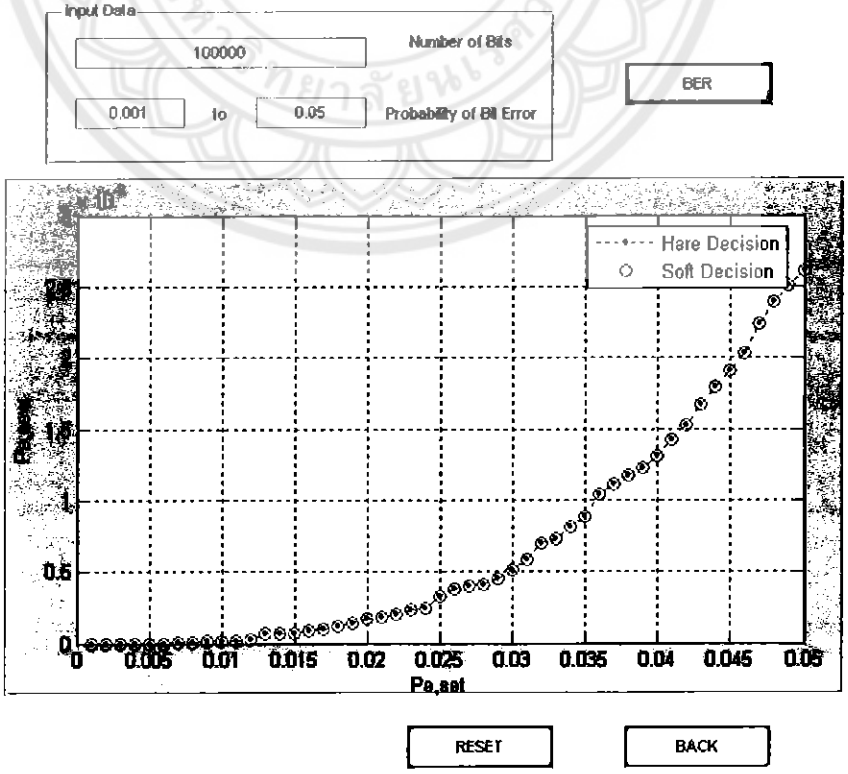
2. อัตราการเข้ารหัสช่องสัญญาณ 1/3 (Covolutioanl Code 1/3)

2.1 คลิกเลือก Convolutional Code 1/3 (อัตราการเข้ารหัส 1/3) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

2.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

2.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

2.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.10



รูปที่ 4.10 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/3

จากรูปที่ 4.10 เมื่อนำมาเปรียบเทียบกับรูปที่ 4.9 แล้ว จะเห็นว่าอัตราการเข้ารหัสช่องสัญญาณ 1/3 นั้นจะมีประสิทธิภาพของการลดอัตราการเกิดข้อมูลผิดพลาดนั้นดีกว่าอัตราการเข้ารหัสช่องสัญญาณ 1/2 ยกตัวอย่างเช่นในค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาด Pe,set ค่าสุดท้าย เท่ากับ 0.05 จะทำให้ได้ความน่าจะเป็นในการเกิดข้อมูลผิดพลาดที่ผ่านการเข้ารหัสแล้วเท่ากับ 0.003 ซึ่งเมื่อสังเกตรูปที่ 4.9 ที่เป็นของอัตราการเข้ารหัสช่องสัญญาณ 1/2 แล้วจะเห็นว่ามีค่าประมาณ 0.0139 มีค่ามากกว่าในกรณีของอัตราการเข้ารหัสช่องสัญญาณ 1/3

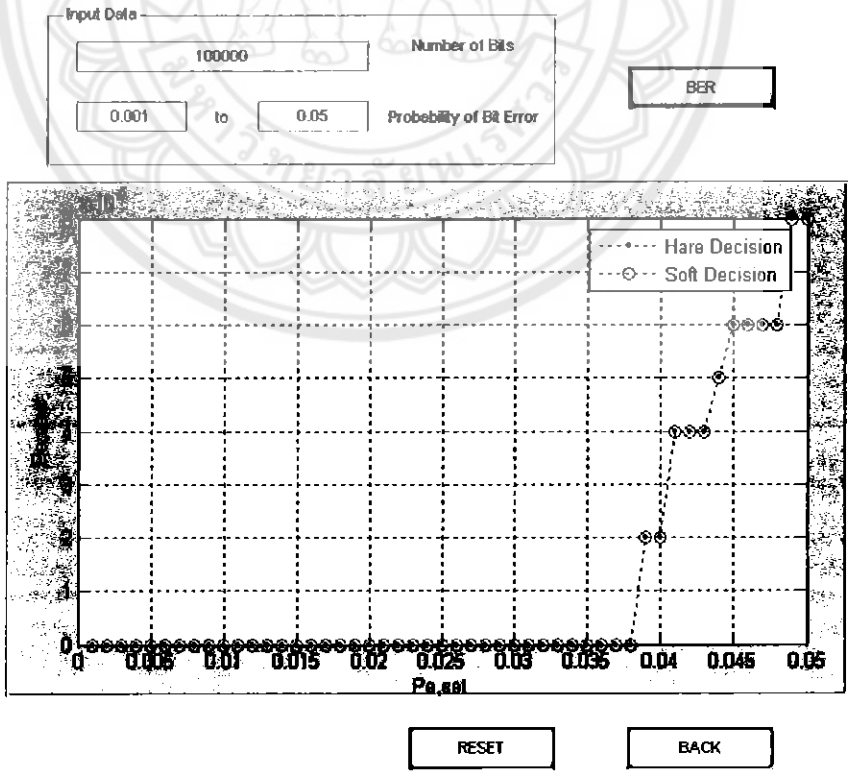
3. อัตราการเข้ารหัสช่องสัญญาณ 1/5 (Covolutioanl Code 1/5)

3.1 คลิกเลือก Convolutional Code 1/5 (อัตราการเข้ารหัส 1/5) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

3.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

3.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

3.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.11



รูปที่ 4.11 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/5

จากรูปที่ 4.11 เมื่อนำไปเปรียบเทียบกับรูปที่ 4.9 และ 4.10 แล้วพบว่า อัตราการเข้ารหัสช่องสัญญาณ 1/5 นั้นมีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดได้ดีกว่าอัตราการเข้ารหัสช่องสัญญาณ 1/2 และ 1/3

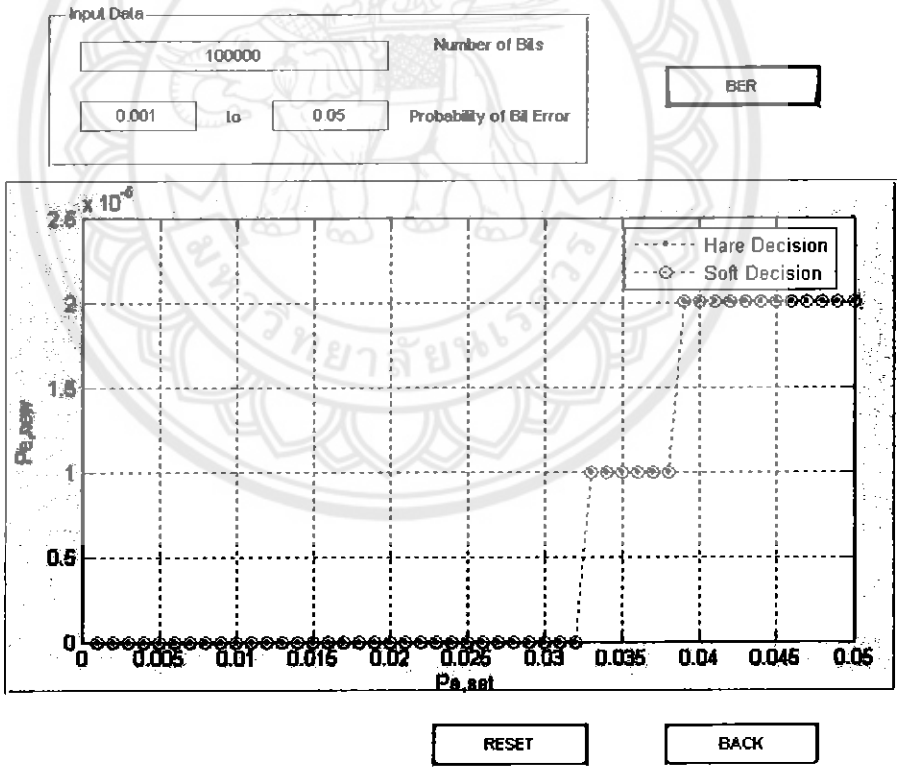
4. อัตราการเข้ารหัสช่องสัญญาณ 1/7 (Covolutioanl Code 1/7)

4.1 คลิกเลือก Convolutional Code 1/7 (อัตราการเข้ารหัส 1/7) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

4.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

4.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

4.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.12



รูปที่ 4.12 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 1/7

จากรูปที่ 4.12 จะเห็นได้ว่าจากที่ผ่านมาทั้งหมดอัตราการเข้ารหัสช่องสัญญาณ 1/7 นั้นสามารถแก้ไขข้อมูลผิดพลาดได้ดีที่สุด

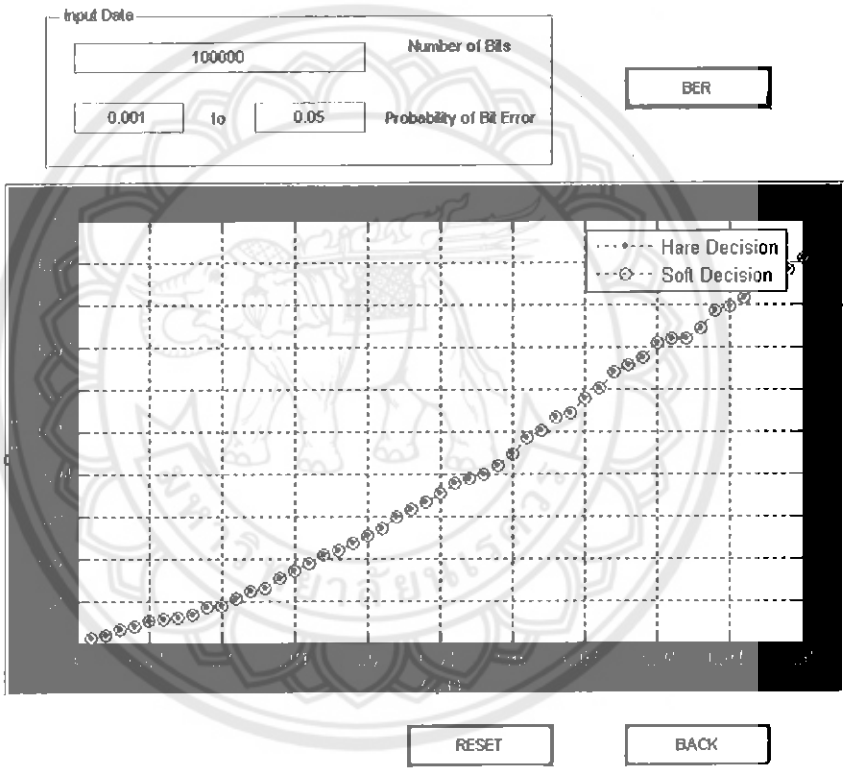
5. อัตราการเข้ารหัสช่องสัญญาณ 2/3 (Covolutioanl Code 2/3)

5.1 คลิกเลือก Convolutional Code 2/3 (อัตราการเข้ารหัส 2/3) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

5.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

5.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

5.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.13



รูปที่ 4.13 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/3

จากรูปที่ 4.13 พบว่าประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัสช่องสัญญาณ 2/3 นั้นไม่ดีนัก เมื่อเทียบกับอัตราการเข้ารหัสช่องสัญญาณที่ผ่านมาทั้งหมด

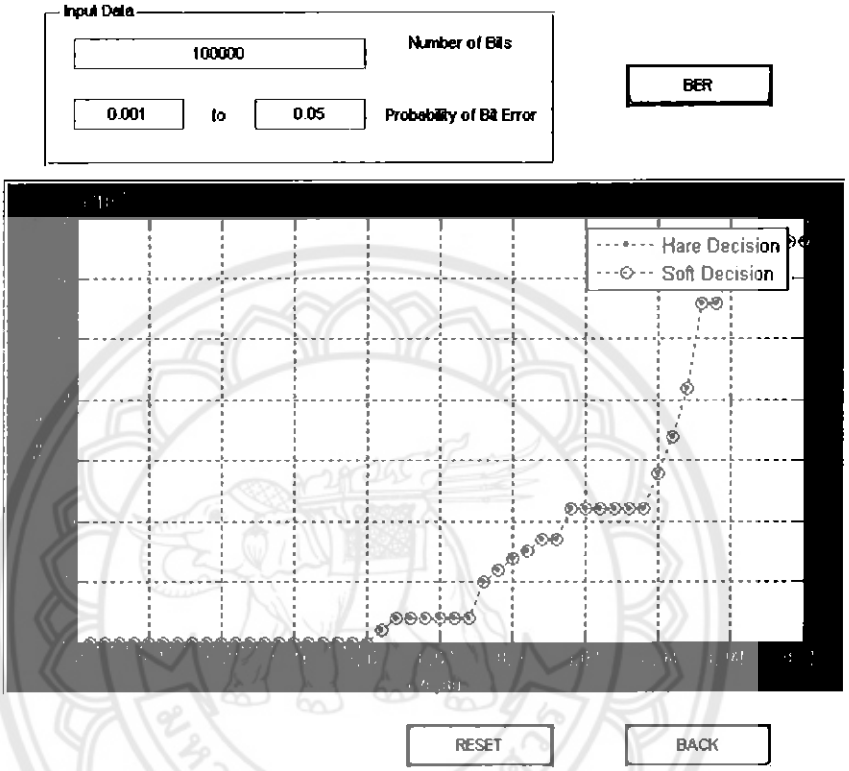
6. อัตราการเข้ารหัสช่องสัญญาณ 2/5 (Covolutioanl Code 2/5)

6.1 คลิกเลือก Convolutional Code 2/5 (อัตราการเข้ารหัส 2/5) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

6.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

6.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

6.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.14



รูปที่ 4.14 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/5

จากรูปที่ 4.14 จะเห็นได้ว่าประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดของอัตราการเข้ารหัสช่องสัญญาณ 2/5 นั้นดีกว่าอัตราการเข้ารหัสช่องสัญญาณ 2/3 ดังรูปที่ 4.13

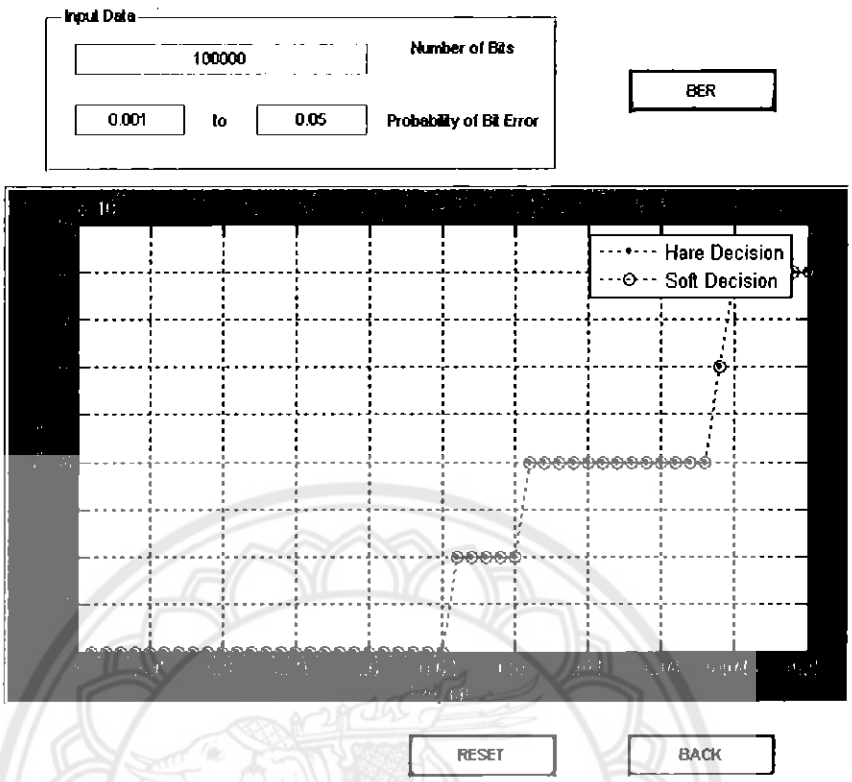
7. อัตราการเข้ารหัสช่องสัญญาณ 2/7 (Covolutioanl Code 2/7)

7.1 คลิกเลือก Convolutional Code 2/7 (อัตราการเข้ารหัส 2/7) โดยคลิกที่หน้าต่างการเลือกอัตราการเข้ารหัส ดังรูปที่ 4.1

7.2 จากนั้นคลิกที่ BER จากหน้าต่าง Application ดังรูปที่ 4.2

7.3 จะปรากฏหน้าต่างแสดงการลดระดับการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.8

7.4 สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0.001 ถึง 0.05 จะได้ค่าต่าง ๆ ดังรูปที่ 4.15



รูปที่ 4.15 การลดระดับอัตราการเกิดข้อมูลผิดพลาด โดยใช้อัตราการเข้ารหัส 2/7

จากรูปที่ 4.15 เมื่อนำไปเปรียบเทียบกับอัตราการเข้ารหัสช่องสัญญาณที่ผ่านมาแล้วจะมีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดที่แย่กว่า อัตราการเข้ารหัสช่องสัญญาณ 1/7 และอัตราการเข้ารหัสช่องสัญญาณ 1/5 เท่านั้น

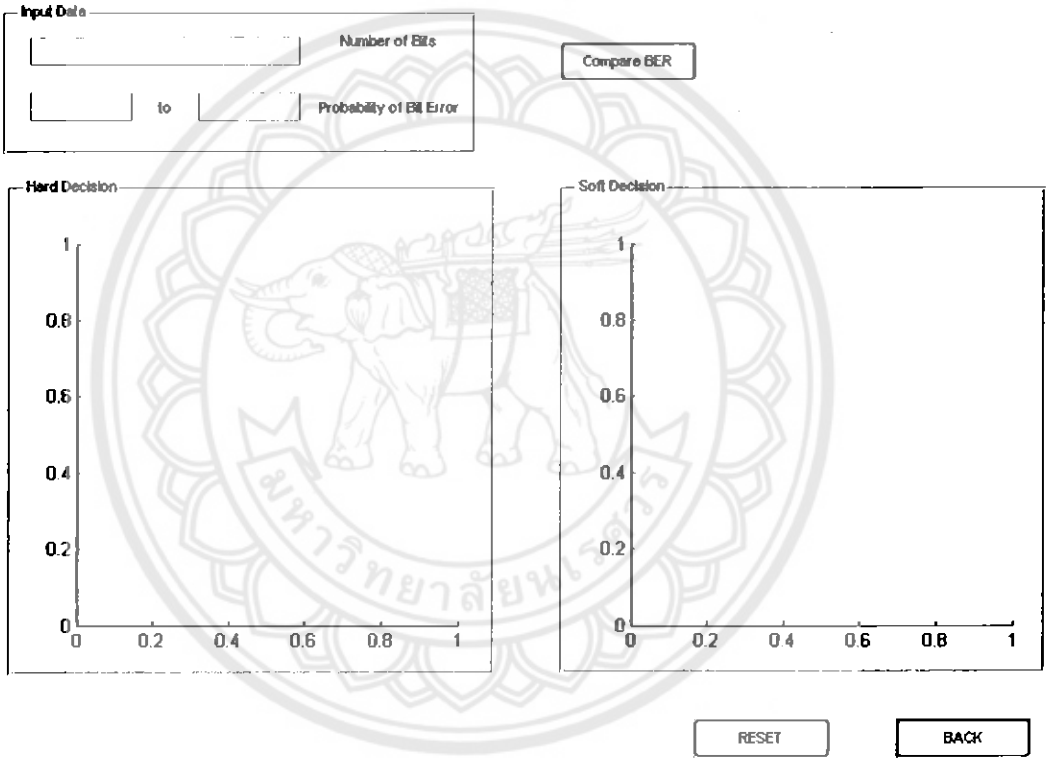
เมื่อพิจารณากราฟในรูปที่ 4.9 – 4.12 จะเห็นได้ว่าความต่อเนื่องของเส้นกราฟนั้นลดลง จนทำให้กราฟในบางช่วงมีลักษณะหักศอก เช่นเดียวกัน ถ้าพิจารณากราฟในรูปที่ 4.13 – 4.15 ก็จะมีลักษณะความต่อเนื่องของเส้นกราฟลดลง ทั้งนี้เพราะว่าจำนวนบิตที่ได้ใส่เข้าไปในโปรแกรมจำลองการเข้ารหัสนั้นมีค่าน้อย และประสิทธิภาพการแก้ไขข้อมูลผิดพลาดในบางอัตราการเข้ารหัสช่องสัญญาณ เช่น อัตราการเข้ารหัส 1/7 นั้น มีประสิทธิภาพที่ดี จึงทำให้เกิดข้อผิดพลาดที่มีค่าน้อยและเท่ากันในบางช่วง ส่งผลให้ได้รูปกราฟออกมาเป็นขั้นบันได แต่ถ้าหากต้องการให้เส้นกราฟที่ได้นั้นมีความต่อเนื่องก็จะต้องเพิ่มจำนวนบิตที่ใช้ในการส่งให้มีจำนวนมากขึ้น

4.3 โปรแกรมแสดงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ

4.3.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1. ในขั้นตอนแรกต้องทำการรันโปรแกรม โดยเลือกชนิดอัตราการเข้ารหัสช่องสัญญาณ (ดังรูปที่ 4.1)

2. เลือก Compare Bit Error Rate จะปรากฏหน้าต่างเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ ดังรูปที่ 4.16

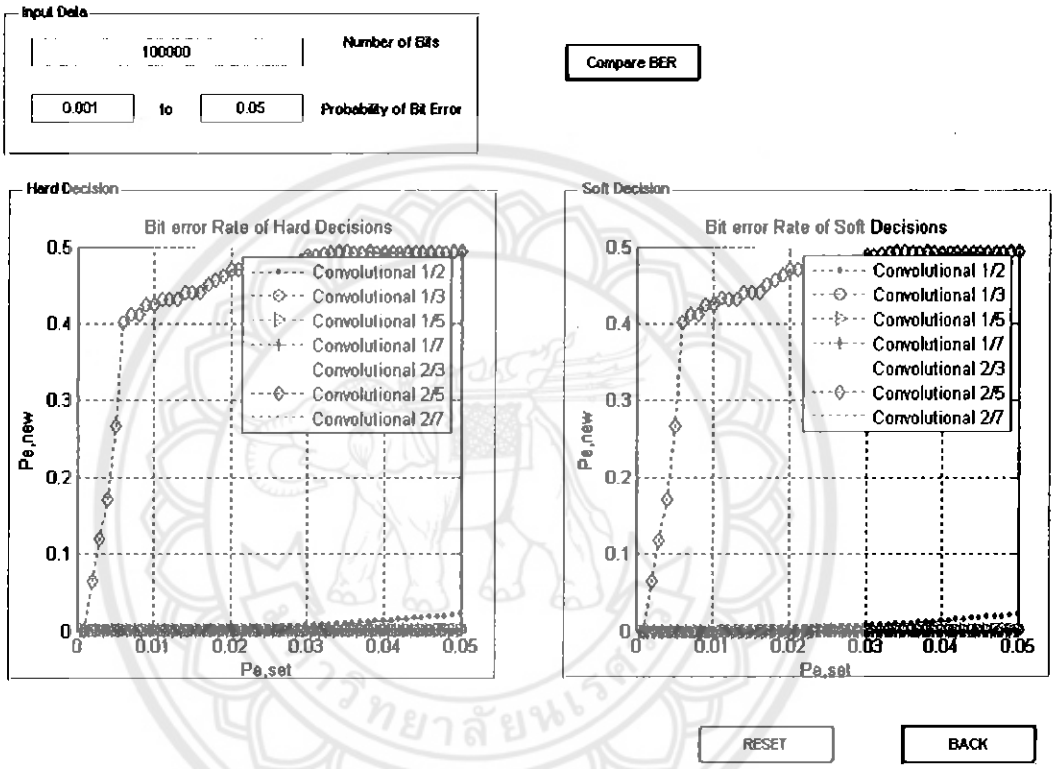


รูปที่ 4.16 หน้าต่างแสดงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 อัตรา

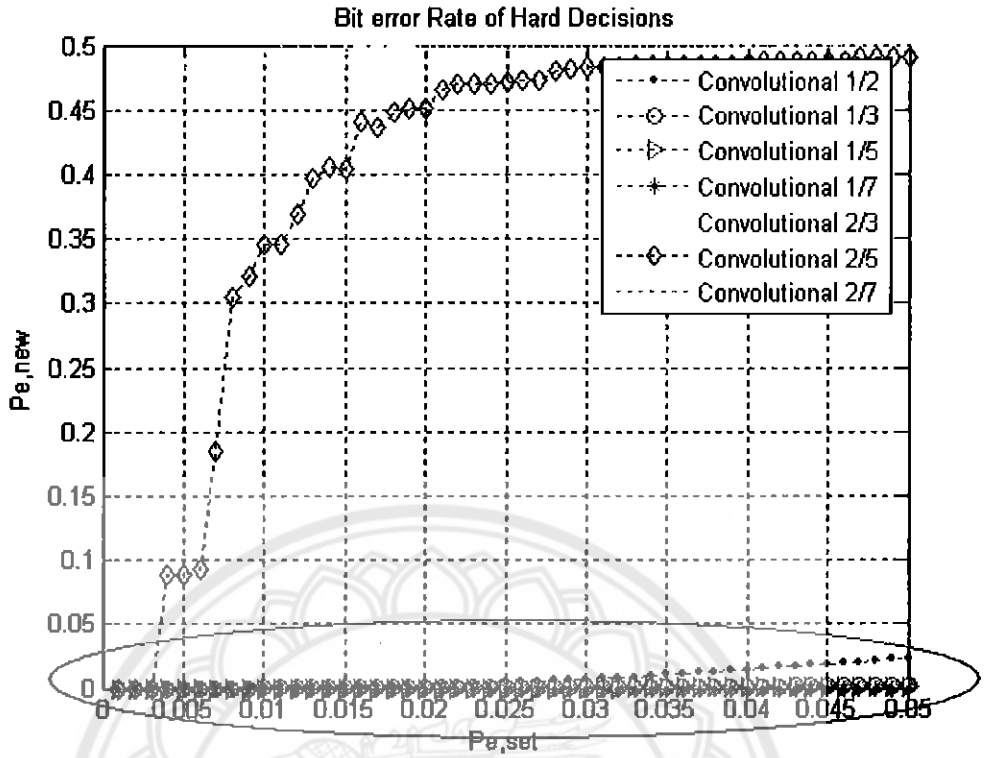
3. เมื่อปรากฏหน้าต่างแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด จากนั้นทำการใส่ค่าจำนวนข้อมูลที่ต้องการใช้ในการส่ง (ซึ่งเป็นบิตข้อมูลที่ใช้ในการส่งจริง นั่นคือ มีปริมาณบิตข้อมูลจำนวนมาก) และเลือกความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error)

4.3.2 ตัวอย่างการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ

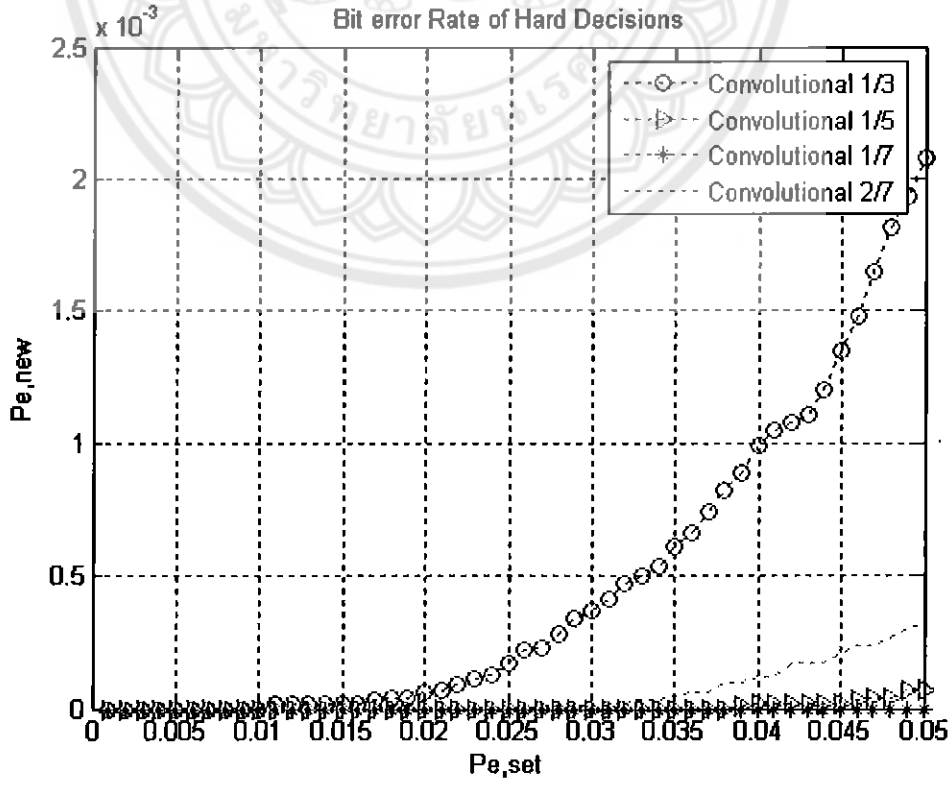
สมมติมีบิตข้อมูลที่ใช้ในการส่ง 100000 บิต และความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) มีค่าตั้งแต่ 0.01 จนถึง 0.05 โดยค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาดนี้จะเพิ่มไปทีละ 0.01 จะได้ค่าต่าง ๆ ดังรูปที่ 4.17



รูปที่ 4.17 ตัวอย่างการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 อัตรา



รูปที่ 4.18 ภาพขยายของการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสทั้ง 7 อัตรา โดยใช้ตัวถอดรหัสแบบ Hard Decision ในรูปที่ 4.17

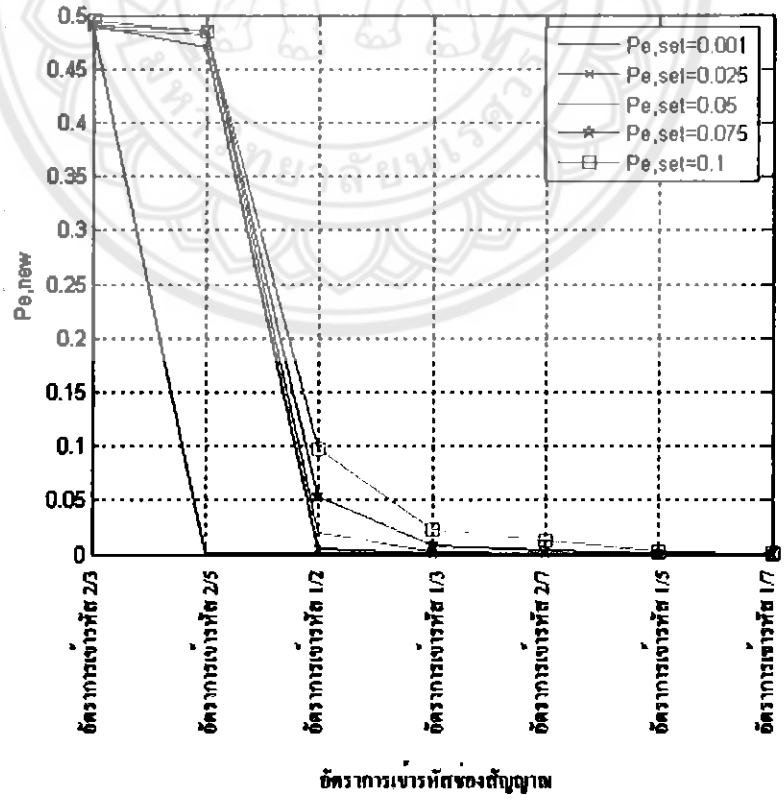


รูปที่ 4.19 ภาพขยายในส่วนที่เป็นวงรี ในรูปที่ 4.18

จากรูปที่ 4.18 และ 4.19 เมื่อนำประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณมาเปรียบเทียบกัน โดยที่พิจารณา ณ Pe,set = 0.05 จะได้ค่า Pe,new ของอัตราการเข้ารหัสช่องสัญญาณแบบต่าง ๆ ทั้ง 7 แบบ ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงการเปรียบเทียบค่า Pe,new ของอัตราการเข้ารหัสทั้ง 7 แบบ โดยที่พิจารณา ณ ค่า Pe,set = 0.05

ประเภท	Pe,new
อัตราการเข้ารหัสช่องสัญญาณ 1/2	13×10^{-3}
อัตราการเข้ารหัสช่องสัญญาณ 1/3	2.1×10^{-3}
อัตราการเข้ารหัสช่องสัญญาณ 1/5	0.1×10^{-3}
อัตราการเข้ารหัสช่องสัญญาณ 1/7	0
อัตราการเข้ารหัสช่องสัญญาณ 2/3	0.5
อัตราการเข้ารหัสช่องสัญญาณ 2/5	0.48
อัตราการเข้ารหัสช่องสัญญาณ 2/7	0.3×10^{-3}



รูปที่ 4.20 การเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของอัตราการเข้ารหัสทั้ง 7 แบบ โดยกำหนดค่า Pe,set

จากรูปที่ 4.20 เป็นการเปรียบเทียบประสิทธิภาพในการแก้ไขข้อมูลผิดพลาด ของอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 แบบ โดยที่จะพิจารณา ณ ค่า Pe,set ต่าง ๆ ซึ่งได้แก่ Pe,set = 0.001, Pe,set = 0.025, Pe,set = 0.05, Pe,set = 0.075 และ Pe,set = 0.1

จะเห็นได้ว่าอัตราการเข้ารหัส 1/7 มีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดมากที่สุด รองลงมาก็คือ อัตราการเข้ารหัส 1/5, อัตราการเข้ารหัส 2/7, อัตราการเข้ารหัส 1/3, อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 2/5 และอัตราการเข้ารหัส 2/3 ตามลำดับ ถึงแม้ว่าอัตราการเข้ารหัส 1/7 จะมีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดได้ดีก็ตาม แต่ก็ต้องเปลือง Bandwidth ที่ใช้ในการส่งสัญญาณ เพราะว่ามี Redundancy bit ที่เพิ่มเข้ามามาก ซึ่งก็คือ จำเป็นจะต้องเพิ่ม Redundancy bit ถึง 6 บิต ต่อจำนวนบิตข้อมูล 1 บิต นั่นเอง

ดังนั้นจากผลการดำเนินโครงงานนี้จึงควรที่จะเลือกอัตราการเข้ารหัสช่องสัญญาณให้เหมาะสมกับการใช้งาน เพราะว่าในงานบางงานนั้นมีการใช้งานของ Bandwidth ที่จำกัด และถ้าหากเลือกอัตราการเข้ารหัสช่องสัญญาณแบบ 1/7 จะทำให้สามารถส่งข้อมูลได้จำนวนน้อยลง เพื่อให้อยู่ในขอบเขตของ Bandwidth ที่จำกัดได้นั่นเอง

ในบทที่ 4 นี้ ได้แสดง Graphic User Interfaces ที่ใช้แสดงผลของการเข้ารหัสข้อมูลแบบ Convolutional Code และการถอดรหัสแบบ Viterbi Decoder แสดงการแก้ไขข้อมูลผิดพลาด อีกทั้งยังนำมาเปรียบเทียบประสิทธิภาพในการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณชนิดต่าง ๆ เพื่อใช้ในการตัดสินใจเลือกนำไปใช้

ในบทต่อไปจะเป็นการสรุปผลการดำเนินโครงงาน และปัญหาที่พบขณะทำโครงงาน

บทที่ 5
สรุปผลการดำเนินโครงงาน

5.1 ผลการดำเนินโครงงาน

โครงงานนี้เป็นการจำลองระบบสื่อสาร โดยใช้การเข้ารหัสช่องสัญญาณเพื่อแก้ไขข้อมูลที่ผิดพลาด การเข้ารหัสช่องสัญญาณที่ใช้ในการดำเนินโครงงานนั้น คือ วิธีการเข้ารหัสแบบ Convolutioanl Code และการถอดรหัสแบบ Viterbi Decoder ซึ่งใช้โปรแกรม MATLAB ในการดำเนินโครงงาน และแสดงออกมาในรูปของ Graphic User Interfaces โดยกำหนดให้มีอัตราการเข้ารหัสช่องสัญญาณ 7 ชนิดด้วยกัน ได้แก่

1. อัตราการเข้ารหัสช่องสัญญาณ 1/2
2. อัตราการเข้ารหัสช่องสัญญาณ 1/3
3. อัตราการเข้ารหัสช่องสัญญาณ 1/5
4. อัตราการเข้ารหัสช่องสัญญาณ 1/7
5. อัตราการเข้ารหัสช่องสัญญาณ 2/3
6. อัตราการเข้ารหัสช่องสัญญาณ 2/5
7. อัตราการเข้ารหัสช่องสัญญาณ 2/7

โครงงานนี้ได้แสดงการแก้ไขข้อมูลผิดพลาด และยังแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด ในแต่ละอัตราการเข้ารหัสช่องสัญญาณ รวมถึงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาด ของแต่ละอัตราการเข้ารหัสช่องสัญญาณ เพื่อใช้ในการตัดสินใจเลือกได้ว่าอัตราการเข้ารหัสช่องสัญญาณแบบใดที่มีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดได้ดีที่สุด เพื่อที่จะนำไปใช้ในระบบการสื่อสารจริง ส่งผลให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด

ซึ่งจากการดำเนินโครงงานสามารถตัดสินใจได้ว่า จากอัตราการเข้ารหัสช่องสัญญาณทั้ง 7 แบบนั้น อัตราการเข้ารหัสช่องสัญญาณ 1/7 มีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดได้ดีที่สุด แต่จะมีข้อเสียคือ มี Redundancy bit มากส่งผลให้ใช้ Bandwidth มากในการส่งข้อมูล

5.2 ปัญหาที่พบขณะดำเนินโครงงาน

5.2.1 เนื่องจากคำสั่งบางคำสั่งในโปรแกรม MATLAB ผู้ดำเนินโครงงานยังไม่มีความเข้าใจอย่างถ่องแท้ ดังนั้นจึงทำให้เกิดความล่าช้าในขณะดำเนินโครงงาน

5.2.2 เนื่องจากโครงงานนี้ใช้ Graphic User Interfaces ในการแสดงผลของการดำเนินงาน ดังนั้นเครื่องคอมพิวเตอร์ควรที่จะมีโปรแกรม MATLAB เวอร์ชัน 7 ขึ้นไป จึงจะสามารถแสดงผลการดำเนินงานออกมาได้

5.3 ข้อเสนอแนะ

5.3.1 ควรขอคำแนะนำในการดำเนินโครงงานจากอาจารย์ที่ปรึกษาโครงงาน เพื่อให้ได้งานที่มีคุณภาพ และเสร็จทันเวลาที่กำหนด

5.3.2 หากสามารถแก้จุดบกพร่องในโปรแกรมที่เกิดจากการเตือนว่าผิดพลาด (error) ได้ก็จะสามารถกำจัดความไม่สะดวกในการใช้งานโปรแกรม

5.3.3 ถ้านำโครงงานนี้ไปปรับให้มีความเหมาะสมแล้ว สามารถที่จะนำไปใช้เป็นสื่อการเรียนการสอนได้

เอกสารอ้างอิง

[1] ปริย หลิมพานิช และ นิรุธ สุขพูล. **"การศึกษาผลกระทบของรหัสคอนโวลูชันในช่องสัญญาณแบบเกาส์ต่อสัญญาณภาพนิ่ง"**. โครงงานวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยธรรมศาสตร์. 2548. หน้า 4-22.

[2] ลัญฉกร วุฒิสิทธิกุลกิจ. **"เทคโนโลยีโทรคมนาคมทฤษฎีข่าวสาร และการเข้ารหัส"**. พิมพ์ครั้งที่ 1: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย. 2546.

[3] DSL lab. **"ชุดถอดรหัส Viterbi ขนาดเล็กสำหรับระบบโทรศัพท์เคลื่อนที่ยุดที่ 3"**. [Online]. Available: http://www.kmitl.ac.th/dslabs/Viterbi.

[4] Simon Haykin. **"Communication System"**. 4 th edition. 2001. pp. 654-673.

[5] John G. Proakis and Masoud Salehi. **"Communication Systems Engineering"**. second edition. 2002. pp. 623-638.

[6] Bernard Sklar. **"Digital Communication Fundamental and Application"**. Prentice-Hall. 1988.

[8] ลัญฉกร วุฒิสิทธิกุลกิจ. **"MATLAB การประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า"**. พิมพ์ครั้งที่ 1: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย. 2547.

[9] รศ.ดร. มนัส สังวงศิลป์ และ วรรัตน์ ภัทรอมรกุล. **"คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์"**. พิมพ์ครั้งที่ 1: สำนักพิมพ์ อินโฟเพรส. 2543.

ประวัติผู้เขียนโครงงาน

ชื่อ นายพนมกร ทองพัฒนกุล

ภูมิลำเนา 156 หมู่ 21 ต.ตาคลี อ.ตาคลี จ.นครสวรรค์ 60140

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนตาคลีประชาสรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : ballbarbor@hotmail.com