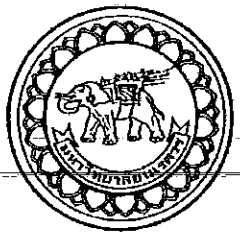


แบบจำลองการเข้ารหัส – ถอดรหัสด้วยรหัสแฮมมิ่งและรหัสบีซีเอช

Simulation of Channel Encoding using Hamming Code and BCH Code

นางสาวชลธิชา ชัยชนะ รหัส 46363149

นางสาวสุภัตรา ปินจันทร์ รหัส 46363438

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	แบบจำลองการเข้ารหัสถอดรหัสด้วยรหัสแฮมมิ่งและบีซีเอช
	Simulation of Channel Encoding using Hamming Code and BCH Code
ผู้ดำเนินโครงงาน	นางสาวชลธิชา ชัยชนะ ID. 46363149
	นางสาวสุภัตรา ปินจันทร์ ID. 46363438
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ

(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ

(ดร.สมยศ เกียรติวนิชวิไล)

หัวข้อโครงงาน	แบบจำลองการเข้ารหัส-ถอดรหัสด้วยรหัสแฮมมิ่งและรหัสบีซีเอช (Simulation of Channel Encoding using Hamming Code and BCH Code)
ผู้ดำเนินโครงงาน	นางสาวชลธิชา ชัยชนะ รหัส 46363149
	นางสาวสุภัตรา ปินจันทร์ รหัส 46363438
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

.....

บทคัดย่อ

ในระบบการสื่อสารข้อมูล การเข้ารหัส-ถอดรหัสเพื่อแก้ไขความผิดพลาดของข้อมูลที่เกิดขึ้นระหว่างการส่งข้อมูลนั้นมีความสำคัญ เนื่องจากในการส่งข้อมูลจากภาคส่งไปยังภาครับนั้น มักจะมีสัญญาณรบกวนเกิดขึ้น ซึ่งส่งผลให้ข้อมูลที่ได้รับมีความผิดพลาด ดังนั้นเพื่อลดการเกิดความผิดพลาดในการส่งข้อมูล จึงมีความจำเป็นต้องทำการเข้ารหัส-ถอดรหัส เพื่อให้สามารถแก้ไขข้อมูลที่ผิดพลาดให้มีความถูกต้องได้

โครงงานนี้เป็นการสร้างแบบจำลองของการเข้ารหัส-ถอดรหัสข้อมูลด้วยโปรแกรม MATLAB โดยใช้การเข้ารหัสแบบ Hamming Code และแบบ BCH Code และทำการวิเคราะห์ประสิทธิภาพในการแก้ไขบิตที่ผิดพลาดของ Hamming Code และ BCH Code

Project Title	Simulation of Channel Encoding using Hamming Code and BCH Code
Name	Miss Chonticha Chaichana ID. 46363149
	Miss Suputtra Pinjun ID. 46363438
Project Advisor	Assistant Professor Surachet Kanprachar , Ph.D.
Major	Electrical Engineering
Departmant	Electrical and Computer Engineering
Academic Year	2006

ABSTRACT

In the information communication system, to be able to correct the errors occurring during the information transfer, channel encoding and decoding are very crucial. Mainly, it is because there is sometimes interference in the processes. Therefore, it is very ideal that the errors correction by channel encoding and decoding are always operated successfully. This brought the idea that there needed to be a study focusing on an ideal errors correction invention; therefore, we tried to study a MATLAB program simulation of the channel encoding and analyzed its proficiency in solving the incorrect bit from the Hamming code and the BCH code.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี ด้วยการช่วยเหลือและสนับสนุนจากหลายๆท่านด้วยกัน ซึ่งผู้เขียนขอขอบพระคุณดังต่อไปนี้

ขอขอบคุณ ผศ.ดร.สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาโครงงาน ในการให้ความรู้ คำปรึกษาเกี่ยวกับการทำโปรแกรมและการค้นหาข้อมูล ตลอดจนสละเวลาให้คำแนะนำทั้งภาคทฤษฎีและภาคปฏิบัติ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์ที่ดีเยี่ยม และขอกราบขอบพระคุณอย่างสูง

ขอขอบคุณ ดร.สมยศ เกียรติวนิชวิไล และ ดร.ชัยรัตน์ พินทอง ที่ให้คำปรึกษาในเรื่องของการเขียนโปรแกรม และการจัดทำรายงาน

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ประสิทธิ์ประสาทวิชาความรู้ และอบรมสั่งสอนให้ผู้จัดทำเป็นคนที่ดีของสังคม

ขอขอบคุณเพื่อนทุกคนที่คอยให้ความช่วยเหลือ ให้กำลังใจ และให้คำปรึกษาในการทำโครงงานนี้จนสำเร็จลุล่วงได้เป็นอย่างดี

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิด และทำให้ผู้จัดทำมีวันนี้

คุณค่า และประโยชน์อันพึงมีจากโครงงานนี้ ทางผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

ชลธิชา ชัยชนะ

สุภัตรา ปินจันทร์

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

หน้า

สารบัญรูปภาพ

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

การติดต่อสื่อสารในปัจจุบันได้กลายเป็นปัจจัยที่สำคัญอย่างหนึ่ง ทำให้สามารถทราบข้อมูลข่าวสารต่างๆได้อย่างไร้พรมแดน รวมทั้งยังให้ความรู้ ความบันเทิง และยังใช้เทคโนโลยีทางด้านการสื่อสารนี้เป็นเครื่องมือในการประกอบธุรกิจทั้งหลายได้อีกด้วยและยังพบว่าปริมาณความต้องการเทคโนโลยีทางด้านการสื่อสารโทรคมนาคมยิ่งมีเพิ่มมากขึ้น เพราะเทคโนโลยีทางด้านการสื่อสารมีการพัฒนาไปอย่างรวดเร็ว ส่งผลให้มีการกระตุ้นให้เกิดการใช้งานในรูปแบบที่มีความหลากหลายมากขึ้น ดังนั้นการติดตามการพัฒนาเทคโนโลยีดังกล่าวจำเป็นที่จะต้องมีความรู้ความเข้าใจเกี่ยวกับระบบสื่อสารเบื้องต้นเป็นอย่างดี โดยจะต้องมีความรู้ความเข้าใจเกี่ยวกับกระบวนการรับ-ส่งข้อมูลในระบบสื่อสาร การศึกษาถึงระบบสื่อสาร จึงมักจะสนใจถึงสมรรถนะและประสิทธิภาพของการรับส่งข่าวสารว่าจะมีมากน้อยเพียงใด

การส่งผ่านสัญญาณดิจิตอลในระบบสื่อสารโดยทั่วไปมักจะเกิดปัญหาการผิดเพี้ยนของรูปสัญญาณเนื่องจากคุณสมบัติที่ไม่เป็นอุดมคติของช่องสัญญาณเองหรือเนื่องจากผลกระทบของสัญญาณรบกวนภายนอกในรูปแบบต่างๆ ปัญหาเหล่านี้อาจส่งผลให้ข้อมูลดิจิตอลที่รับได้ที่ภาครับมีความผิดพลาดเกิดขึ้น ด้วยเหตุนี้ระบบสื่อสารในปัจจุบันจึงต้องการความถูกต้องและความแน่นอนในการส่งข้อมูลสูง มักจะมีการนำข้อมูลดิจิตอลไปผ่านกระบวนการเข้ารหัสช่องสัญญาณ (Channel Coding) ก่อนที่จะส่งออกไป เพื่อให้การรับส่งมีความผิดพลาดน้อยลงและอยู่ในระดับที่ยอมรับได้

ดังนั้นโครงงานนี้จะนำเสนอการศึกษา และแบบจำลองการทำงานการเข้ารหัส และการถอดรหัสของรหัสชนิดต่างๆ ที่ใช้ในระบบสื่อสารไม่ว่าจะเป็นการใช้ Hamming Code และ BCH Code ว่ามีหลักการทำงานอย่างไรและมีคุณสมบัติที่แตกต่างกันอย่างไร เพื่อเป็นแนวทางในการศึกษาและไปสู่การประดิษฐ์คิดค้นเทคโนโลยีใหม่ไปประยุกต์ใช้งานให้เกิดประโยชน์ต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการทำงานในระบบสื่อสารในส่วนของ Channel Coding
2. เพื่อศึกษาการใช้รหัส Hamming Code และ BCH Code
3. เพื่อศึกษาการตรวจสอบบิตที่ผิดพลาดและศึกษาความสามารถในการแก้ไขบิตที่ผิดพลาดของรหัส Hamming Code และ BCH Code
4. เพื่อสร้างแบบจำลองระบบการรับ-ส่งข้อมูล ของรหัส Hamming Code และ BCH Code
5. เพื่อศึกษาการเขียนโปรแกรมด้วย MATLAB

1.3 ขอบข่ายของโครงงาน

1. ศึกษาการใช้รหัส Hamming Code และ BCH Code
2. ศึกษาการเข้ารหัส-ถอดรหัสของแบบรหัส Hamming Code และ BCH Code
3. ศึกษาระบบการส่งข้อมูลในระบบการสื่อสารข้อมูล
4. ศึกษาความสามารถในการแก้ไขบิตที่ผิดพลาดให้กลับมาถูกต้องดังเดิม
5. สร้างแบบจำลองของสื่อสารโดยใช้โปรแกรม MATLAB

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับศึกษาการใช้รหัส Hamming code และ BCH code
2. ศึกษาการทำงานในระบบสื่อสารในส่วนของ Channel Coding
3. ออกแบบระบบการรับ-ส่งข้อมูลในระบบการสื่อสารโดยใช้รหัสต่างๆ
4. เขียนโปรแกรมเพื่อสร้างจำลองการรับ-ส่งข้อมูลของระบบสื่อสาร
5. ทดสอบการทำงาน
6. สรุปการทดลองและจัดทำรูปเล่ม

1.5 การดำเนินงาน

กิจกรรม	ปี 2548			ปี 2549								
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับศึกษาการใช้รหัส Hamming code และ BCH codes	←	—	→									
2. ศึกษาการทำงานของระบบสื่อสารในส่วนของ Channel Coding			←	—	→							
3. ออกแบบระบบการรับ-ส่งข้อมูลในระบบการ					←	—	→					

สื่อสาร												
4. เขียนโปรแกรมเพื่อจำลองการรับ-ส่งข้อมูลของ						←		→				
สื่อสาร												
5.ทดสอบการทำงาน								←		→		
6.สรุปการทดลองและจัดทำรูปเล่ม										←		→

1.6 ผลที่คาดว่าจะได้รับ

1. ได้โปรแกรมคอมพิวเตอร์ที่สามารถแสดงผลของการการเข้ารหัสช่องสัญญาณและการถอดรหัสของ Hamming Code และ BCH Code
2. ได้โปรแกรมคอมพิวเตอร์ที่สามารถแสดงลักษณะของสัญญาณที่เข้ามาและส่งออกไปได้
3. สามารถเปรียบเทียบรหัส Hamming Code และ BCH Code ได้
4. มีความรู้และทักษะในการเขียนโปรแกรมด้วย MATLAB

1.7 งบประมาณที่ต้องใช้

1. ค่าเอกสารประกอบการทำโครงงาน 1100 บาท
2. ค่าจัดทำรูปเล่มโครงงาน 900 บาท

รวมเป็นเงินทั้งสิ้น 2000 บาท

(สองพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2

หลักการและทฤษฎีการเข้ารหัสช่องสัญญาณ

การส่งผ่านสัญญาณดิจิตอลในระบบสื่อสารโดยทั่วไป มักจะเกิดปัญหาการผิดเพี้ยนของรูปสัญญาณเนื่องจากคุณสมบัติที่ไม่เป็นอุดมคติของช่องสัญญาณเองหรือเนื่องจากผลกระทบของสัญญาณรบกวนภายนอกในรูปแบบต่างๆ ปัญหาเหล่านี้อาจส่งผลให้ข้อมูลดิจิตอลที่รับได้ที่ภาครับมีความผิดพลาดเกิดขึ้น ด้วยเหตุนี้ระบบสื่อสารในปัจจุบันจึงต้องการความถูกต้องและความแน่นอน ในการส่งข้อมูลสูงมักจะมีการนำข้อมูลดิจิตอลไปผ่านกระบวนการเข้ารหัสช่องสัญญาณ (Channel Coding) ก่อนที่จะส่งออกไป เพื่อให้การรับส่งมีความผิดพลาดน้อยลงและอยู่ในระดับที่ยอมรับได้ โดยส่วนใหญ่ในปัจจุบันใช้ระบบดิจิตอล (Digital System) ซึ่งเป็นระบบที่มีลักษณะของสัญญาณแบบไม่ต่อเนื่อง (Discrete Signal) ซึ่งมีข้อดีมากกว่าระบบอนาล็อก (Analog System) ที่มีลักษณะของสัญญาณที่ต่อเนื่อง ในการเข้ารหัสช่องสัญญาณนั้นจำเป็นต้องมีการเพิ่มจำนวนบิตที่ส่งออกไป โดยบิตพิเศษที่เพิ่มเข้ามาจะช่วยให้ภาครับสามารถที่จะตรวจจับความผิดพลาดได้ (Error Detection) หรือหากมีการเพิ่มจำนวนบิตเข้าไปเป็นจำนวนที่มากพอ ภาครับก็อาจจะสามารถแก้ไขความผิดพลาด (Error Correction) ของข้อมูลได้ด้วย สังเกตว่าการเข้ารหัสช่องสัญญาณมีผลทำให้อัตราบิตข้อมูลที่ต้องการส่งจริงมีขนาดสูงขึ้น ซึ่งหมายความว่าช่องสัญญาณที่ใช้ส่งจะต้องมีแบนด์วิดที่ใหญ่ขึ้นด้วย หรือถ้าพิจารณาในทางกลับกันหากช่องสัญญาณมีแบนด์วิดที่จำกัดและต้องการให้การรับส่งของข้อมูลมีความถูกต้องมากขึ้น ก็จะต้องลดอัตราการส่งบิตข้อมูลของผู้ใช้ลง

2.1 หลักการพื้นฐานของการเข้ารหัสช่องสัญญาณ [1]

ในการที่จะทำให้การรับส่งข้อมูลมีความถูกต้องโดยอาศัยวิธีการเข้ารหัสช่องสัญญาณสามารถกระทำได้ 2 รูปแบบ คือ

2.1.1. FEC (Forward Error Correction) [1]

วิธีการ FEC นั้นภาครับจะต้องสามารถตรวจจับว่ามีบิตผิดพลาดเกิดขึ้นในระหว่างการส่งสัญญาณหรือไม่ และถ้าหากมีก็จะต้องสามารถระบุได้ว่าบิตที่ผิดพลาดเกิดขึ้นที่ตำแหน่งใด จากนั้นภาครับจึงทำการแก้ไขบิตผิดพลาดดังกล่าวให้ถูกต้อง

2.1.2. ARQ (Automatic Repeat Request) [1]

วิธีการ ARQ ภาครับมีหน้าที่เพียงแต่ตรวจว่ามีบิตผิดพลาดเกิดขึ้นหรือไม่เท่านั้น และหากพบว่ามีก็จะส่งสัญญาณกลับไปที่ภาคส่ง เพื่อขอให้ภาคส่งทำการส่งข้อมูลชุดเดิมกลับมาใหม่

2.2 รหัสช่องสัญญาณ [1]

การเข้ารหัสช่องสัญญาณนั้นสามารถแบ่งออกได้เป็น 2 ประเภท คือ

2.2.1 การเข้ารหัสช่องสัญญาณแบบบล็อก (Block Codes) [1]

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 ที่ภาครับก็จะมีวงจรในการทำหน้าที่ถอดรหัสของสัญญาณเพื่อดึงบิตข้อมูลเดิมออกมาพร้อมกันนั้นก็จะให้ค่าที่เรียกว่าซินโดรม(Syndrome) ออกมาด้วย โดยค่าซินโดรมนั้นมีไว้สำหรับบ่งบอกว่ามีความผิดพลาดเกิดขึ้นในข้อมูลหรือไม่ หรืออาจใช้ในการบ่งบอกถึงตำแหน่งของบิตที่ผิดด้วย

2.2.2 การเข้ารหัสช่องสัญญาณแบบคอนโวลูชัน (Convolutional Codes) [1]

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

2.3 พาริตี้เช็ก (Parity check) [4]

พาริตี้เช็ก (Parity check) เป็นวิธีการเข้ารหัสประเภทหนึ่งที่สามารถตรวจสอบว่าในบิตข้อมูลที่รับได้มีความผิดพลาดหรือไม่ สำหรับขั้นตอนการเข้ารหัสพาริตี้เช็กมีวิธีง่ายๆดังต่อไปนี้

สมมติว่ามีบิตข้อมูลที่จะทำการเข้ารหัสทั้งหมด k บิต ซึ่งประกอบด้วย

$$\mathbf{m} = [m_{k-1}, m_{k-2}, ..., m_2, m_1, m_0] \qquad (2.1)$$

บิตข้อมูลเหล่านี้จะนำมาใช้สำหรับหาค่าพาริตี้ p โดย

$$p = m_{k-1} \oplus m_{k-2} \oplus ... \oplus m_2 \oplus m_1 \oplus m_0 \qquad (2.2)$$

ค่า p ที่คำนวณได้นี้ก็จะนำไปต่อกับบิตข้อมูล ผลที่ได้คือข้อมูลที่ผ่านการเข้ารหัสแล้วซึ่งอยู่ในรูป

$$\mathbf{c} = [\, m_{k-1}, m_{k-2}, ..., m_2, m_1, m_0, p \,] \qquad (2.3)$$

ฉะนั้นชุดรหัสพาริตี้เช็กจึงเขียนได้ในรูป (n,k) โดย $n = k+1$

เมื่อบิตข้อมูลเหล่านี้เดินทางถึงที่ภาครับ ภาครับสามารถตรวจสอบว่ามีการผิดพลาดของบิตข้อมูลในระหว่างการส่งหรือไม่ โดยการหาค่าผลรวมของทุกบิตโดยใช้มอดูโล 2 (Modulo-2) ในลักษณะที่คล้ายกันกับที่ภาคส่ง ถ้าผลรวมที่ได้มีค่าไม่เท่ากับ 0 แสดงว่าบิตข้อมูลเหล่านี้มีความผิดพลาดเกิดขึ้น สังเกตว่าวิธีนี้ไม่สามารถตรวจสอบความผิดพลาดของบิตข้อมูลในกรณีที่จำนวนบิตที่ผิดเป็นจำนวนคู่ แต่จะสามารถจับความผิดพลาดของบิตข้อมูลจำนวนคี่ได้ทั้งหมด หมายเหตุการณ์เข้ารหัสประเภทนี้สามารถกระทำได้ 2 รูปแบบคือ พาริตี้แบบคู่ (Even parity) หรือพาริตี้แบบคี่ (Odd parity) สำหรับวิธีที่ผ่านมาเป็นพาริตี้แบบคู่ ในกรณีที่ต้องการให้เป็นวิธีพาริตี้แบบคี่ ก็เพียงแต่กลับค่าของ p ให้เป็นค่าที่กลับกันคือจาก 0 เป็น 1 และจาก 1 เป็น 0 และที่ภาครับก็จะต้องปรับการตัดสินใจให้ตรงและสอดคล้องกับที่ภาคส่งด้วย

2.4 ทฤษฎีพื้นฐานของพีชคณิต [1]

พีชคณิตเป็นคณิตศาสตร์ที่ได้รับการค้นพบและพัฒนามานาน สามารถนำมาประยุกต์ใช้ให้เกิดประโยชน์ในสาขาวิชาด้านต่างๆได้มากมาย ในส่วนของการพัฒนารหัสช่องสัญญาณก็เช่นกัน ได้นำคุณสมบัติของคณิตศาสตร์ทางด้านพีชคณิตมาใช้ในการวิเคราะห์และอธิบายถึงคุณสมบัติของรหัสช่องสัญญาณ เช่น รหัสแฮมมิ่ง (Hamming Code) รหัสบีซีเอช (BCH Code) รหัสบล็อกเชิงเส้น (Linear block code) เป็นต้น

2.4.1 กรุ๊ป (Group)

เซต (Set) กลุ่มของอีลิเมนต์ (Element) ที่ไม่ได้มีการกำหนดโอเปอร์เรชัน (Operation) ระหว่างอีลิเมนต์ เซตสามารถแบ่งออกได้เป็น 2 ประเภทคือ เซตจำกัด (Finite set) และเซตไม่จำกัด (Infinite set) เซตจำกัดหมายถึงเซตที่มีจำนวนอีลิเมนต์ในเซตเป็นจำนวนจำกัด เช่น เซตของมหาวิทยาลัยในประเทศไทย และเซตของตัวอักษรในภาษาไทย เป็นต้น ส่วนเซตไม่จำกัด (Infinite set) คือเซตที่มีจำนวนอีลิเมนต์ไม่จำกัด เช่น เซตของจำนวนเต็ม (Integer) และเซตของจำนวนจริง (Real number) เป็นต้น

กรุ๊ป (Group) คือเซตของอีลิเมนต์ G ที่มีการกำหนดโอเปอร์เรชัน " * " ระหว่างอีลิเมนต์ภายในเซต โดยโอเปอร์เรชันที่กำหนดขึ้นต้องมีคุณสมบัติดังต่อไปนี้ จึงจัดว่าเป็นกรุ๊ป

1. คุณสมบัติปิด (Closure) : $\forall a, b \in G : a * b \in G$

2. คุณสมบัติการจัดหมู่ (Associativity) : $\forall a,b,c \in G : a*(b*c)=(a*b)*c$
3. คุณสมบัติการมีเอกลักษณ์ (Identity) : มี $\exists e \in G$: ที่ทำให้ $a*e=e*a=a$ สำหรับ $\forall e \in G$
4. คุณสมบัติการมีอินเวอร์ส (Inverse) : $\forall a \in G : \exists b=a^{-1} \in G : a*b=e$
5. คุณสมบัติการสลับที่ (Commutativity) : $\forall a,b \in G : a*b=b*a$

2.4.2 ซับกรุ๊ป (Subgroup)

ภายในกรุ๊ป G ถ้าพิจารณาเฉพาะอีลิเมนต์บางส่วนของ G ที่มีจำนวนสมาชิกน้อยกว่ากรุ๊ป Gแล้ว พบว่าการประกอบกันของอีลิเมนต์เหล่านี้ มีคุณสมบัติครบตามเงื่อนไขของความเป็นกรุ๊ปแล้ว จะเรียกเซตของอีลิเมนต์ย่อยนี้ว่า ซับกรุ๊ป (Subgroup) ของ G

2.4.2 ริง (Ring)

ริง (Ring) คือกลุ่มของอีลิเมนต์ R ที่มีการกำหนดโอเปอร์เรชันระหว่างอีลิเมนต์ 2 แบบคือ การบวกและการคูณ โดยมีคุณสมบัติดังต่อไปนี้

1. R เป็นกรุ๊ปที่มีคุณสมบัติการสลับที่(CommulativeหรือAbelian group)ภายใต้โอเปอร์เรชันการบวก + โดยอีลิเมนต์ 0 เป็นเอกลักษณ์การบวก (Additive identity element)
2. คุณสมบัติปิด (Closure) ภายใต้โอเปอร์เรชันการคูณ : $\forall a,b \in R : a*b \in R$
3. คุณสมบัติการจัดหมู่(Associativity)ภายใต้โอเปอร์เรชันการคูณ: $\forall a,b,c = R : a*(b*c)=(a*b)*c$
4. คุณสมบัติการแจกแจง (Distributivity) : $\forall a,b,c \in R : a*(b+c)=a*b+a*c$
5. คุณสมบัติการสลับที่ภายใต้โอเปอร์เรชันการคูณ ($a*b=b*a$)
6. คุณสมบัติการมีเอกลักษณ์การคูณ และมีอีลิเมนต์ 1 เป็นเอกลักษณ์

2.4.3 ฟิลด์

ฟิลด์ (Field) คือ กลุ่มของอีลิเมนต์ F ที่มีการกำหนดโอเปอร์เรชันระหว่างอีลิเมนต์ 2 แบบ คือ การบวกและการคูณ โดยมีคุณสมบัติดังต่อไปนี้

1. F เป็นกรุ๊ปที่มีคุณสมบัติการสลับที่ (Abelian หรือ Commutative) ภายใต้การโอเปอร์เรชันการบวก + โดยมีอีลิเมนต์ 0 เป็นเอกลักษณ์การบวก (Additive identity element)
2. $F-\{0\}$ เป็นกรุ๊ปที่มีคุณสมบัติการสลับที่(AbelianหรือCommutative)ภายใต้โอเปอร์เรชันการคูณ
3. คุณสมบัติการแจกแจง (Distributivity) : $\forall a,b,c \in F : a*(b+c)=a*b+a*c$

2.5 Galois field [4]

Galois field คือ เซตที่มีจำนวนสมาชิกจำกัดและมีคุณสมบัติของความเป็นฟิลด์ โดยทั่วไปจะใช้สัญลักษณ์ $GF(q)$ แทนฟิลด์ที่มีอันดับเท่ากับ q

2.5.1 ทฤษฎีของ Galois field

เซตที่ประกอบด้วยจำนวนเต็ม $\{0,1,2,...,q-1\}$ โดย q เป็นจำนวนเฉพาะ จัดเป็นฟิลด์ $GF(q)$ ภายใต้การโอเปอร์เรชันการบวกและการคูณแบบมอดูโล q

2.5.2 คณิตศาสตร์ของตัวเลขไบนารี

ในระบบไบนารีจะมีตัวเลขสำหรับการใช้งานอยู่เพียง 2 แบบเท่านั้นคือ 0 และ 1 การบวกลบคูณหารที่กระทำกับตัวเลขไบนารีนั้นเป็นไปในลักษณะเดียวกับคณิตศาสตร์ที่คุ้นเคยตัวอย่างเช่น 1+0 = 1 แต่หากต้องบวกเลข 1+1 แล้ว ถ้าพิจารณาตามคณิตศาสตร์ที่ใช้งานทั่วไปจะได้ผลลัพธ์เป็น 2 แต่สำหรับคณิตศาสตร์แบบไบนารี จะใช้แทนตัวเลข 2 ด้วย 0 นั่นคือ 1+1 = 2 = 0 จากตัวอย่างนี้ถ้าพิจารณาต่อจะพบว่า ในเมื่อ 1+1 = 0 แล้วแสดงว่า 1 = -1 ด้วย ซึ่งแสดงให้เห็นว่าคณิตศาสตร์ของตัวเลขไบนารีการบวกหรือการลบตัวเลขนั้นไม่ได้มีความแตกต่างกันเลยคือให้ผลเหมือนกันทุกประการ

ในระบบตัวเลขแบบไบนารีก็สามารถเขียนรูปของสมการหลายตัวแปรได้เช่นเดียวกับคณิตศาสตร์ที่ใช้กันอยู่ทั่วไป ยกตัวอย่างเช่น ถ้ามีตัวแปรทั้งหมด 3 ตัว x, y และ z และมีสมการความสัมพันธ์ระหว่าง x, y และ z อยู่ 3 สมการดังต่อไปนี้

$$x + y = 0 \tag{2.4}$$

$$x + z = 1 \tag{2.5}$$

$$x + y + z = 1 \tag{2.6}$$

จากสมการความสัมพันธ์สามารถแก้สมการเพื่อหาค่า x, y และ z ได้ไม่ยาก จากสมการแรกระบุว่า $x + y = 0$ ฉะนั้นเมื่อแทนค่านี้ลงในสมการที่สามจะได้ $z = 1$ และเมื่อแทนค่า $z = 1$ ลงในสมการที่สองจะได้ $x = 0$ ในท้ายสุดแทน $x = 0$ ลงในสมการแรกจะได้ว่า $y = 0$

เนื่องจากสมการทั้งสามนั้นมีผลเฉลยอยู่จริง ดังนั้นสมการเหล่านี้ย่อมจะต้องมีความเป็นอิสระกันแบบเชิงเส้น (Linearly independent) และค่าดีเทอร์มิเนนท์ของสัมประสิทธิ์ที่อยู่ทางด้านซ้ายของสมการย่อมจะต้องมีค่าไม่เป็นศูนย์อย่างแน่นอน เมื่อค่าดีเทอร์มิเนนท์ไม่เท่ากับศูนย์ ก็แน่นอนว่าย่อมมีค่าเท่ากับ 1 หากพิจารณาค่าดีเทอร์มิเนนท์ของตัวอย่างข้างต้น

$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix} = 1 \bullet \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} - 1 \bullet \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} + 0 \bullet \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$$

$$= 1 \bullet 1 - 1 \bullet 0 + 0 \bullet 1 = 1 \tag{2.7}$$

จะเห็นว่าดีเทอร์มิเนนท์มีค่าเป็น 1 จริงตามที่อธิบายไว้ และสามารถใช้หลักเกณฑ์ของ Cramer ในการแก้สมการหา x, y และ z ได้ดังนี้

$$x=\frac{\begin{vmatrix}010\\101\\111\end{vmatrix}}{\begin{vmatrix}110\\101\\111\end{vmatrix}}=0,\quad y=\frac{\begin{vmatrix}100\\111\\111\end{vmatrix}}{\begin{vmatrix}110\\101\\111\end{vmatrix}}=0,\quad \text{และ } z=\frac{\begin{vmatrix}010\\101\\111\end{vmatrix}}{\begin{vmatrix}111\\101\\111\end{vmatrix}}=1 \qquad (2.8)$$

2.5.3 พหุนามที่มีสัมประสิทธิ์เป็นตัวเลขไบนารี GF (2)

พิจารณาพหุนาม $f(x)$ ที่มีสัมประสิทธิ์ของพหุนามที่มีค่าอยู่ในฟิลด์ไบนารี $GF(2)$ ซึ่งสามารถอธิบายในรูปต่อไปนี้คือ

$$f(x)=f_0+f_1x+f_2x^2+\ldots+f_nx^n \qquad (2.9)$$

โดย $f_0, f_1, f_2, \ldots, f_n$ เป็นสัมประสิทธิ์ที่มีค่าได้เพียง 2 รูปแบบคือ 0 หรือ 1 เท่านั้น กำหนดให้ดีกรีของพหุนามคือ ค่ายกกำลังสูงสุดของ x เช่น ในกรณีของ $f(x)$ จะมีดีกรีเท่ากับ n ตัวอย่างของพหุนามที่มีดีกรีค่าต่างๆเป็นดังนี้

พหุนามที่มีดีกรีเท่ากับ 0 ได้แก่ $f(x)=1$

พหุนามที่มีดีกรีเท่ากับ 1 ได้แก่ $f(x)=x$ และ $(x+1)$

พหุนามที่มีดีกรีเท่ากับ 2ได้แก่ $f(x)=x^2,(1+x^2)$ และ $(1+x+x^2)$

2.5.4 การบวกและการคูณพหุนาม

ในการบวกพหุนาม 2 ชุดเข้าด้วยกันมิได้แตกต่างจากกรรมวิธีการบวกธรรมดาทั่วไป นั่นคือให้นำสัมประสิทธิ์ของพหุนามที่มีค่ายกกำลังเท่ากันมาบวกกัน ตัวอย่างเช่น

$$\begin{aligned}a(x)+b(x)&=(1+x+x^3)+(1+x+x^2+x^4+x^5)\\&=(1+1)+(1+1)x+x^2+x^3+x^4+x^5\\&=x^2+x^3+x^4+x^5\end{aligned} \qquad (2.10)$$

ในการคูณพหุนามก็เช่นกัน สามารถใช้กรรมวิธีการคูณที่คุ้นเคยมาใช้เช่นกัน ตัวอย่างเช่น

$$a(x)*b(x)=(1+x+x^3)*(1+x^3+x^5)$$

$$= (1 + x^3 + x^5) + x * (1 + x^3 + x^5) + x^3 * (1 + x^3 + x^5)$$
$$= (1 + x^3 + x^5) + (x + x^4 + x^6) + (x^3 + x^6 + x^8)$$
$$= 1 + x + x^4 + x^5 + x^8 \qquad (2.11)$$

2.5.5 การหารและการแยกตัวประกอบพหุนาม

การหารพหุนามที่สัมประสิทธิ์ได้มาจาก $GF(2)$ พิจารณาพหุนาม $f(x)$ เมื่อนำมาหารด้วยพหุนามอีกตัวหนึ่งคือ $g(x)$ จะได้ความสัมพันธ์ดังต่อไปนี้

$$f(x) = q(x)g(x) + r(x) \qquad (2.12)$$

โดย $q(x)$ คือผลที่ได้จากการหาร (Quotient) และ $r(x)$ คือเศษที่เหลือจากการหาร (Remainder) และ $r(x)$ จะมีดีกรีที่ต่ำกว่า $g(x)$ เสมอ

2.5.6 พหุนามพริมีทีฟ (Primitive Polynomial) [4]

พหุนามใดๆ จะจัดเป็นพหุนามพริมิทีฟ ได้จะต้องไม่สามารถแยกตัวประกอบได้อีก (Irreducible) และต้องหาร $x^n + 1$ ไม่ลงตัว สำหรับ n ที่มีค่าอยู่ตั้งแต่ 1 ถึง $2^m - 2$ ในการพิจารณาว่าพหุนามหนึ่งมีคุณสมบัติพริมีทีฟหรือไม่ ยังไม่มีวิธีลัดในการตรวจสอบได้อย่างรวดเร็ว แต่อย่างไรก็ตามสามารถที่จะแจกแจงพหุนามที่มีคุณสมบัติพริมีทีฟสำหรับพหุนามดีกรีต่างๆได้ โดยจะมีตาราง สำหรับดูค่าพหุนามพริมิทีฟที่สามารถนำมาใช้งานได้ทันที

2.6 รหัสบล็อกเชิงเส้น (Linear block code) [1]

ในระบบรหัสบล็อกเชิงเส้น ข้อมูลที่จะทำการเข้ารหัสจะถูกแบ่งออกเป็นบล็อกข้อมูลขนาดเท่ากันจำนวน k บิตซึ่งเขียนแทนด้วย $m_0, m_1, m_2, ..., m_{k-1}$ ในการเข้ารหัสจะนำบล็อกข้อมูลทั้ง k บิตไปใช้ในการสร้างพาริตี้บิตจำนวน $n-k$ บิตซึ่งเขียนแทนด้วย $b_0, b_1, b_2, ..., b_{n-k-1}$ และเมื่อนำบิตข้อมูลและพาริตี้มาประกอบกันจะได้เป็นคำรหัส $c_0, c_1, c_2, ..., c_n$ ซึ่งถ้าแสดงในรูปของสมการจะเห็นความสัมพันธ์ดังนี้

$$c_i = \begin{cases} b_i & i = 0,1,...,n-k-1 \\ m_{i+k-n} & i = n-k, n-k-1, ..., n-1 \end{cases} \qquad (2.13)$$

กระบวนการเข้ารหัสบล็อกเชิงเส้นจึงเหมือนการแปลงบิตข้อมูลจำนวน k บิต ให้ได้เป็นคำรหัสที่มีขนาดเพิ่มขึ้นเป็น n บิต นั่นเอง ซึ่งหากพิจารณาในเบื้องต้นดูเหมือนว่าเป็นกระบวนการที่ไม่ซับซ้อนหรือยุ่งยากเท่าใดนัก แต่ถ้าพิจารณาในดีจะพบว่าการเข้ารหัสจะต้องมีการพิจารณาบิตครั้งละ

k บิต ซึ่งมีรูปแบบที่เป็นไปได้ทั้งหมดมากถึง 2^k รูปแบบ ฉะนั้นถ้าต้องบรรจุรูปแบบทั้งหมดไว้ในหน่วยความจำเพื่อแปลงให้ได้เป็นคำรหัสที่เหมาะสมที่มีขนาดความยาว n บิต จะต้องอาศัยวงจรที่ซับซ้อนและหน่วยความจำที่มีขนาดใหญ่มากโดยเฉพาะอย่างยิ่งถ้า k มีขนาดใหญ่ขึ้น ความซับซ้อนของวงจรสร้างรหัสนี้เองที่เป็นประเด็นปัญหาหลักในการพัฒนาและก็เป็นเหตุผลสำคัญที่ทำให้การพัฒนาแบบบล็อกแทบทั้งหมดจึงมุ่งเน้นไปในกลุ่มของรหัสบล็อกที่มีคุณสมบัติพิเศษที่เรียกว่าคุณสมบัติเชิงเส้น (linear property) เป็นหลัก เพราะคุณสมบัติเชิงเส้นนั้นสามารถช่วยลดความซับซ้อนของวงจรสร้างรหัสได้อย่างมาก และจะเรียกรหัสที่ได้นี้ว่า รหัสบล็อกเชิงเส้นจากที่กล่าวมาจะเห็นว่าหัวใจของการเข้ารหัสอยู่ที่การคำนวณค่าบิตพาริตีในกรณีของรหัสบล็อกเชิงเส้นค่าของบิตพาริตีจะคำนวณจากบิตข้อมูลในรูปของการบวกเชิงเส้นในรูปแบบดังต่อไปนี้ [1]

$$b_i = p_{i0}m_0 + pi_2m_2 + p_{i,k-1}m_{k-1} \qquad ;\ i = 0,1,2,...,n-k-1 \qquad (2.14)$$

โดยสัมประสิทธิ์ p_{ij} จะมีค่าได้ 2 แบบเท่านั้น คือ 0 หรือ 1 ทั้งนี้ค่าของ p_{ij} จะกำหนดให้สอดคล้องกับความต้องการที่จะให้บิตพาริตี b_i มีความเกี่ยวพันกับบิตข้อมูลที่ m_i หรือไม่ นั่นคือถ้าไม่ต้องการให้มีความสัมพันธ์หรือขึ้นแก่กันก็กำหนด $p_{ij} = 0$ เพราะฉะนั้นจุดสำคัญของการเข้ารหัสจึงอยู่ที่การกำหนด p_{ij} ที่เหมาะสมเพื่อให้ได้คุณสมบัติตามที่ต้องการ

โดยทั่วไปในการศึกษาโครงสร้างวิธีการเข้าและถอดรหัสบล็อกเชิงเส้น มักจะแสดงค่าต่างๆที่กล่าวมาข้างต้นในรูปของเมตริกซ์ ได้ดังนี้

$$\mathbf{m} = [m_0, m_1, m_2, ..., m_{k-1}] \qquad (2.15)$$

$$\mathbf{b} = [b_0, b_1, b_2, \ldots, b_{n-k}] \qquad (2.16)$$

$$\mathbf{c} = [c_0, c_1, c_2, ..., c_{n-1}] \qquad (2.17)$$

โดย

$$\mathbf{P} = \begin{bmatrix} p_{00} & p_{10} & \cdots & p_{n-k-1,0} \\ p_{01} & p_{11} & \cdots & p_{n-k-1,1} \\ \vdots & \vdots & \cdots & \vdots \\ p_{0,k-1} & p_{1,k-1} & \cdots & p_{n-k,k-1} \end{bmatrix} \qquad (2.18)$$

สำหรับเมตริกซ์ $\mathbf{c}$ สามารถแสดงในรูปของ $\mathbf{b}$ และ $\mathbf{m}$ ได้ดังต่อไปนี้

$$\mathbf{c} = [\mathbf{b}\ \mathbf{m}] \qquad (2.19)$$

อาศัยความสัมพันธ์ตามสมการ จะได้ว่า

$$\mathbf{c} = [\mathbf{mP} \quad \mathbf{m}] = \mathbf{m}[\mathbf{P} \; I_k] \tag{2.20}$$

โดย I_k คือ เมตริกซ์เอกลักษณ์ขนาด $k \times k$

$$I_k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{2.21}$$

ถ้ากำหนดให้

$$\mathbf{G} = [\mathbf{P} \; I_k] \tag{2.22}$$

ซึ่งเป็นเมตริกซ์ ขนาด $k \times n$ ความสัมพันธ์ในสมการ ก็สามารถเขียนได้เป็น

$$\mathbf{c} = \mathbf{mG} \tag{2.23}$$

จากสมการจะเห็นว่าคำรหัส **c** สามารถคำนวณได้จากการคูณชุดข้อมูล **m** โดยตรงกับเมตริกซ์ **G** ดังนั้นจึงเรียกเมตริกซ์ **G** ว่า เมตริกซ์ตัวกำหนด (Generator matrix)

รหัสบล็อกเชิงเส้นมีคุณสมบัติที่น่าสนใจคือ คุณสมบัติปิด(Closure) นั่นคือถ้านำคำรหัสสองคำนวณมารวมกันจะได้เป็นคำรหัสใหม่ที่เป็นสมาชิกของรหัสนั้น ๆ ด้วย สามารถพิสูจน์คุณสมบัติข้อนี้ได้โดยง่าย สมมติให้คำรหัส c_1 และ c_2 ได้จากการเข้ารหัสข้อมูล m_1 และ m_2 ตามลำดับ จากสมการจะได้ว่า

$$c_1 + c_2 = m_1\mathbf{G} + m_2\mathbf{G} = (m_1 + m_2)\mathbf{G} \tag{2.24}$$

เนื่องจาก $m_1 + m_2$ จะได้เป็นข้อมูลชุดใหม่ ซึ่งเมื่อนำคูณกับเมตริกซ์ ตัวกำหนด **G** ก็ย่อมจะได้คำรหัสที่เป็นสมาชิกหนึ่งของรหัสด้วย

สำหรับส่วนต่อไปนี้จะอธิบายถึงการสร้างความสัมพันธ์ที่เป็นประโยชน์กับกระบวนการถอดรหัส นิยามเมตริกซ์ **H** ที่มีขนาด $(n-k) \times n$ ขึ้นดังนี้

$$\mathbf{H} = [I_{n-k} \mid \mathbf{P}^{\mathrm{T}}] \tag{2.25}$$

โดย I_{n-k} คือ เมตริกซ์เอกลักษณ์ขนาด $(n-k)\times(n-k)$ และ $\mathbf{P}^{\mathrm{T}}$ คือ เมตริกซ์ทรานส์โพสของ $\mathbf{P}$ ซึ่งมีขนาดเท่ากับ $(n-k)\times k$ ถ้านำเมตริกซ์ทรานส์โพสของ $\mathbf{G}$ มาคูณทั้งสองด้านจะได้

$$\mathbf{H}\mathbf{G}^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}} + \mathbf{P}^{\mathrm{T}} \tag{2.26}$$

จากคุณสมบัติของการบวกกันแบบมอดูโล 2 จะได้ว่า $\mathbf{P}^{\mathrm{T}} + \mathbf{P}^{\mathrm{T}} = 0$ โดยเมตริกซ์ที่คำนวณได้มีขนาดเท่ากับ $(n-k)\times k$ และมีสมาชิกเป็นศูนย์ทั้งหมดนั่นคือ

$$\mathbf{H}\mathbf{G}^{\mathrm{T}} = 0 \tag{2.27}$$

หรือหากพิจารณาในอีกลักษณะหนึ่งจะได้ว่า $\mathbf{H}\mathbf{G}^{\mathrm{T}} = 0$ ด้วยเช่นกัน สามารถใช้ประโยชน์จากความสัมพันธ์นี้ได้โดยนำทรานส์โพสของเมตริกซ์ $\mathbf{H}$ ไปคูณกับสมการข้างต้นทั้งสองด้านดังจะได้ผลดังนี้

$$\mathbf{c}\,\mathbf{H}^{\mathrm{T}} = \mathbf{m}\mathbf{G}\,\mathbf{H}^{\mathrm{T}} = 0 \tag{2.28}$$

สมการนี้มีประโยชน์กับกระบวนการถอดรหัส ซึ่งจะได้อธิบายถึงวิธีการนำไปใช้งานในส่วนต่อไป สำหรับเมตริกซ์ $\mathbf{H}$ มีชื่อเรียกว่า เมตริกซ์พาริตีเช็ก (Parity-check matrix)

2.6.1 Hamming Code [5]

ในปี ค.ศ. 1950 Hamming ได้ค้นพบว่าสามารถสร้างวิธีการเข้ารหัสแบบบล็อกเชิงเส้นที่สามารถตรวจจับและแก้ไขบิตผิดพลาดได้ 1 บิต โดยปกติการเข้ารหัสข้อมูลจำนวน $n+1$ บิตให้ได้คำรหัสที่ยาว $c = n-k$ บิต ถ้าต้องการให้สามารถแก้ไขได้ 1 บิต จะต้องมีซินโดรมที่ต่างกันอย่างน้อย $2^c \geq n+1$ ค่า โดยจะใช้ซินโดรมจำนวน n ค่าในการระบุถึงตำแหน่งที่ผิด และใช้ซินโดรมค่าสุดท้ายสำหรับบ่งบอกว่าไม่มีบิตที่ผิดพลาดเกิดขึ้นเลย ดังนั้นจำนวนบิตเช็กที่ต้องใช้ $c = n-k$ บิต จะต้องมากพอที่จะทำให้เงื่อนไข $2^c \geq n+1$ เป็นจริงสำหรับค่า c ค่าหนึ่งที่เลือกใช้ สามารถหาค่า n และ k มากที่สุดได้ดังนี้

$$n = 2^c - 1 \tag{2.29}$$

$$k = n - c = 2^c - c - 1 \tag{2.30}$$

จะได้

$$(n,k) = (2^m - 1, 2^m - 1 - m) \tag{2.31}$$

เมื่อ m คือจำนวนเต็มบวก

การส่งข้อมูล

ในการส่งข้อมูลจะใช้หลักทางคณิตศาสตร์ที่เกี่ยวกับการคำนวณในระบบรหัสบล็อกเชิงเส้น (Linear block code) ข้อมูลที่ทำการเข้ารหัสจะถูกแบ่งออกเป็นบล็อกขนาดเท่ากันจำนวน k บิต ซึ่งเขียนแทนด้วย $m_0, m_1, m_2, ..., m_{k-1}$

$$\mathbf{m} = [m_0 \;\; m_1 \;\; m_2 \;\; .... \;\; m_{k-1}] \tag{2.32}$$

ในการเข้ารหัสจะนำบล็อกข้อมูลทั้ง k บิตไปใช้ในการสร้างพาริตี้บิตจำนวน $n-k$ บิตซึ่งเขียนแทนด้วย $b_0, b_1, b_2, \ldots, b_{n-k-1}$

$$\mathbf{b} = [b_0 \;\; b_1 \;\; b_2 \;\; ... \;\; b_{n-k-1}] \tag{2.33}$$

$$X_i = \underbrace{\boxed{b_0, b_1, b_2, \ldots, b_{n-k-1}}}_{parity\ \ bits} \underbrace{\boxed{m_0, m_1, m_2, \ldots, m_{k-1}}}_{message\ \ bits}$$

รูปที่ 2.1 แสดงคำรหัส (Codeword) [5]

จากรูปที่ 2.1 เมื่อนำบิตข้อมูลและพาริตี้มาประกอบกันจะได้เป็นคำรหัส (Codeword) ซึ่งเป็นระบบ Systematic ถ้าแสดงในรูปของสมการจะเห็นความสัมพันธ์ดังนี้

เมื่อ

$$X_i = \begin{cases} b_i & i = 0,1,\cdots,n-k-1 \\ m_{i+k-n} & i = n-k, n-k-1, \ldots n-1 \end{cases} \tag{2.34}$$

จะพบว่ากระบวนการเข้ารหัสบล็อกเชิงเส้นเป็นการแปลงบิตข้อมูลจำนวน k บิตให้เป็นคำรหัสที่มีขนาดเพิ่มขึ้นเป็น n บิตจากการพิจารณาจะพบว่าการเข้ารหัสจะต้องมีการพิจารณาบิตข้อมูลครั้งละ k บิต ซึ่งมีรูปแบบความเป็นไปได้ทั้งหมด 2^k รูปแบบ

การคำนวณค่าพาริตี้จะคำนวณจากบิตข้อมูลในรูปผลบวกเชิงเส้น ดังแสดงในสมการต่อไปนี้

$$b_i = p_{i,0}m_0 + p_{i,1}m_1 + p_{i,2}m_2 + \ldots + p_{i,k-1}m_{k-1} \quad (2.35)$$

โดยสัมประสิทธิ์ $P_{i,j}$ จะมีค่าได้สองแบบเท่านั้นคือ 0 หรือ 1 โดยที่พิจารณาค่าของ b_i กับ m_i ว่ามีความเกี่ยวพันกับบิตข้อมูลหรือไม่ ถ้าไม่สัมพันธ์กันก็กำหนดให้ $P_{i,j} = 0$

$$P_{i,j} = \begin{cases} 1 & if \quad bi \quad depends \quad on \quad m \\ 0 & \text{Otherwise} \end{cases} \quad (2.36)$$

เขียนในรูปเมตริกซ์ได้ดังนี้

$$\mathbf{P} = \begin{bmatrix} P_{0,0} & P_{1,0} & \cdots & P_{n-k-1,0} \\ P_{1,0} & P_{1,1} & \cdots & P_{n-k-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k-1,0} & P_{k-1,1} & \cdots & P_{n-k-1,k-1} \end{bmatrix} \quad (2.37)$$

กำหนดให้ Generator matrix $(G_{k\times n})$ มีค่าเป็น $\mathbf{G} = [\mathbf{P} : I_k]$
เมื่อ

$I_k : k \times k$ Identity matrix (2.38)

$\mathbf{P} : k \times (n-k)$ Binary matrix (2.39)

จะทำให้ได้ข้อมูลที่ส่งไปเป็นระบบ Systematic linear (n,k) block code คือ $\mathbf{x = mG}$

การถอดรหัสข้อมูล

กำหนดให้เมตริกซ์ $\mathbf{H} = [I_k \vdots \mathbf{P}^T]$ (2.41)

กำหนดให้ $[r]$ คือข้อมูลที่รับมา (Received codeword)

$$[r] = [\hat{v}_1\ \hat{v}_2\ \hat{v}_3 \ldots \hat{v}_n] \quad (2.42)$$

คำนวณหาค่า Syndrome

$$Syndrome = [r][H^T] \quad (2.43)$$

2.6.2 รหัส BCH [4]

รหัสแบบบอส-โชดูรี-ฮอคเคนเกม (Bose-Chudhuri-Hocquenghem Codes: BCH)

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

$$\text{Block length}: \quad n = 2^m - 1 \tag{2.44}$$

$$\text{Message Size}: \quad k \geq n - mt \tag{2.45}$$

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

การเข้ารหัสแบบ BCH

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

$$g(X) = \text{LCM}\,(m_1(X), m_2(X), .., m_{2t}(X)) \tag{2.46}$$

โดยที่ $\alpha, \alpha^2, \alpha^3, .., \alpha^{2t}$ เป็นรากของ $g(x)$ นั้นคือ $g(\alpha^i) = 0$ สำหรับ $i = 1,2,3,...,2t$ ดังนั้นกำลังคู่และกำลังคี่ของ α จะมีโพลิโนเมียลต่ำสุดเหมือนกัน ทำให้ผลลัพธ์เจนเนอเรเตอร์โพลิโนเมียลของรหัสลดลงเหลือ

$$g(X) = \text{LCM}\,(m_1(X), m_3(X), .., m_{2t-1}(X)) \tag{2.47}$$

บางครั้งเรียกสมการข้างต้นว่า Narrow sense BCH code เนื่องจากลำดับของแต่ละโพลิโนเมียลต่ำสุดจะเท่ากับหรือน้อยกว่า m อันดับของ $g(X)$ มากที่สุดคือ mt หรืออีกนัยคือ บิตของพาริตีที่เช็ค $n-k$ จะไม่เกิน mt

การเข้ารหัสข้อมูล $u(X)$ โดยการใช้รหัส BCH กับเจอเนอเรเตอร์โพลิโนเมียล $g(X)$ ข้างต้น สามารถหาสัญลักษณ์พาริตี้ตามสมการ

$$b(X) = X^{n-k}u(X) \bmod g(X) \tag{2.48}$$

ดังนั้น รหัสคำข้อมูลที่ผ่านการเข้ารหัส BCH ได้เป็น

$$v(X) = X^{n-k}u(X) + b(X) \tag{2.49}$$

การถอดรหัสแบบ BCH

ให้ $v(X) = v_0 + v_1X + v_2X^2 + ... + v_{n-1}X^{n-1}$ เป็นเวกเตอร์ที่ผ่านการเข้ารหัส BCH และให้ $r(X) = r_0 + r_1X + r_2X^2 + ... + r_{n-1}X^{n-1}$ เป็นเวกเตอร์ที่รับได้ ดังนั้นรูปแบบของรหัสที่ผิดที่เกิดในช่องส่งที่มีสัญญาณรบกวน คือ

$$e(X) = r(X) + v(X) \tag{2.50}$$

การถอดรหัสจะแบ่งออกเป็น 3 ขั้นตอนใหญ่ ๆ คือ

ขั้นตอนที่ 1: คำนวณหาซินโดรม $S = (S_1, S_2, ..., S_{2t})$ จากรหัสเวกเตอร์ที่รับได้ $r(X)$

ขั้นตอนที่ 2: หาโพลิโนเมียล $\sigma(X)$ ของตำแหน่งที่ผิดจากซินโดรม $S = (S_1, S_2, ..., S_{2t})$ มีเทคนิคแบบอินเทอเรทีฟ (Interative) Berlakamp algorithm ช่วยในการหาโพลิโนเมียลตำแหน่งที่ผิด

ขั้นตอนที่ 3: หาตำแหน่งของรหัสที่ผิดและการแก้รหัสที่ผิด ตำแหน่งของรหัสที่ผิดนั้นจะกลับ (Invers) กันกับรากของ $\sigma(X)$ มีวิธีการหาช่วยโดย Chien search

การถอดรหัสจะเสร็จสิ้นสมบูรณ์เมื่อบวกเวกเตอร์ของ $e(X)$ กับ $r(X)$

ตัวอย่างการเข้ารหัสและถอดรหัสแบบ BCH

สำหรับตัวอย่างกำหนดการเข้ารหัส BCH(15, 5) ที่สามารถแก้ไขความผิดพลาดได้ เท่ากับ 3 และเจนเนอเรเตอร์โพลิโนเมียลที่ใช้ คือ

$$g(X) = X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1 \tag{2.51}$$

ข้อมูลที่ต้องการเข้ารหัสเป็น

$$u(X) = X^3 + X \tag{2.52}$$

สัญลักษณ์พาริตีจะได้เป็น

$$b(X) = X^8 + X^7 + X^5 + X^4 + X^3 \tag{2.53}$$

รหัสคำ (Codeword) ที่ได้

$$v(X) = X^{13} + X^{11} + X^8 + X^7 + X^5 + X^4 + X^3 \quad (2.54)$$

สำหรับตัวอย่างนี้สมมุติว่ามีความผิดพลาดของข้อมูลและให้รูปแบบของความผิดพลาดเป็น

$$e(X) = X^{11} + X^8 + X^7 \quad (2.55)$$

เวกเตอร์ที่รับได้จากรูปแบบความผิดพลาด คือ

$$r(X) = X^{13} + X^5 + X^4 + X^3 \quad (2.56)$$

ดังนั้นขั้นตอนแรกในการถอดรหัสจากรูปแบบความผิดพลาด คือ การคำนวณหาส่วนประกอบซีนโดรม S_1 ถึง S_6

$$S_1 = 0 \quad (2.57)$$

$$S_2 = 0 \quad (2.58)$$

$$S_3 = \alpha^{11} \quad (2.59)$$

$$S_4 = 0 \quad (2.60)$$

$$S_5 = \alpha^5 \quad (2.61)$$

$$S_6 = \alpha^7 \quad (2.62)$$

เริ่มต้นกับคุณสมบัติของอัลกอริทึม Berlekamp-Massey

$$\mu = 0 \quad (2.63)$$

$$\sigma(X) = 1 \quad (2.64)$$

$$l = 0 \quad (2.65)$$

$$\beta(X) = 0 \quad (2.66)$$

เข้าไปทำวนซ้ำตามวิธีการ

$\mu = 1$:

$$d_1 = S_1 \quad (2.67)$$

$$d_1 = 0 \quad (2.68)$$

d_1 เป็น ศูนย์ :

$$\beta(X) = X\beta(X) \quad (2.69)$$

$$\beta(X) = X[0] \quad (2.70)$$

$$\beta(X) = 0 \quad (2.71)$$

$\mu = 2$:

$$d_2 = S_2 \quad (2.72)$$

$$d_2 = 0 \quad (2.73)$$

d_2 เป็น ศูนย์ :

$$\beta(\mathrm{X}) = \mathrm{X}\beta(\mathrm{X}) \quad (2.74)$$

$$\beta(\mathrm{X}) = \mathrm{X}[0] \quad (2.75)$$

$$\beta(\mathrm{X}) = 0 \quad (2.76)$$

$\mu = 3$:

$$d_3 = S_3 \quad (2.77)$$

$$d_3 = \alpha^{11} \quad (2.78)$$

d_3 ไม่เป็น ศูนย์ ต้องทำการปรับ $\sigma(\mathrm{X})$:

$$\sigma'(x) = \sigma(X) - dX\beta(X) \quad (2.79)$$

$$\sigma'(\mathrm{X}) = [1] - \alpha^{11}\mathrm{X}[0] \quad (2.80)$$

$$\sigma'(\mathrm{X}) = [1] + [0] \quad (2.81)$$

$$\sigma'(X) = 1 \quad (2.82)$$

$2l = 0$ น้อยกว่า $\mu = 3$ ดังนั้นจะได้ผล คือ

$$\beta(X) = d^{-1}\sigma(X) \quad (2.83)$$

$$\beta(X) = \alpha^4[1] \quad (2.84)$$

$$\beta(X) = \alpha^4 \quad (2.85)$$

$$l = \mu - l = 3 - 0 = 3 \quad (2.86)$$

$$\sigma(X) = \sigma'(X) = 1 \quad (2.87)$$

$\mu = 4$:

$$d_4 = S_4 + \sigma_1 S_3 + \sigma_2 S_2 + \sigma_3 S_1 \quad (2.88)$$

$$d_4 = 0 + 0\alpha^{11} + 00 + 00 \quad (2.89)$$

$$d_4 = 0 \quad (2.90)$$

d_4 เป็น ศูนย์ :

$$\beta(X) = X\beta(X) \quad (2.91)$$

$$\beta(X) = X[\alpha^4] \quad (2.92)$$

$$\beta(X) = \alpha^4 X \quad (2.93)$$

$\mu = 5$:

$$d_5 = S_5 + \sigma_1 S_4 + \sigma_2 S_3 + \sigma_3 S_2 \quad (2.94)$$

$$d_5 = \alpha^5 + 00 + 0\alpha^{11} + 00 \quad (2.95)$$

$$d_5 = \alpha^5 \quad (2.96)$$

d_5 ไม่เป็น ศูนย์ ต้องทำการปรับ $\sigma(X)$:

$$\sigma'(X) = \sigma(X) - d(X)\beta(X) \quad (2.97)$$

$$\sigma'(X) = [1] - \alpha^5 X[\alpha^4 X] \quad (2.98)$$

$$\sigma'(X) = [1] + [\alpha^9 X^2] \quad (2.99)$$

$$\sigma'(X) = \alpha^9 X^2 + 1 \quad (2.100)$$

$2l = 6$ ไม่น้อยกว่า $\mu = 5$ ดังนั้นทำตาม

$$\beta(X) = X\beta(X) \quad (2.101)$$

$$\beta(X) = X[\alpha^4 X] \quad (2.102)$$

$$\beta(X) = \alpha^4 X^2 \quad (2.103)$$

$$\sigma(X) = \sigma'(X) = \alpha^9 X^2 + 1 \quad (2.104)$$

$\mu = 6$:

$$d_6 = S_6 + \sigma_1 S_5 + \sigma_2 S_4 + \sigma_3 S_3 \quad (2.105)$$

$$d_6 = \alpha^7 + 0\alpha^5 + 0\alpha^9 + 0\alpha^{11} \quad (2.106)$$

$$d_6 = \alpha^7 \quad (2.107)$$

d_6 ไม่เป็น ศูนย์ ต้องทำการปรับ $\sigma(X)$:

$$\sigma'(X) = \sigma(X) - dX\beta(X) \quad (2.108)$$

$2l = 6$ ไม่น้อยกว่า $\mu = 6$ ดังนั้นทำให้ได้

$$\beta(X) = X\beta(X) \quad (2.109)$$

$$\beta(X) = X[\alpha^4 X^2] \quad (2.110)$$

$$\beta(X) = \alpha^4 X^3 \quad (2.111)$$

$$\sigma(X) = \sigma'(X) = \alpha^{11} X^3 + \alpha^9 X^2 + 1 \quad (2.112)$$

ดังนั้นโพลิโนเมียลตำแหน่งที่ผิด คือ

$$\sigma(X) = \alpha^{11} X^3 + \alpha^9 X^2 + 1 \quad (2.113)$$

หารากของ $\sigma(X)$ เพื่อทราบตำแหน่งข้อผิดพลาด ตำแหน่งข้อผิดพลาด คือ ส่วนกลับของราก $\sigma(X)$

$$(\alpha^4)^{-1} = \alpha^{11} \quad (2.114)$$

$$(\alpha^7)^{-1} = \alpha^8 \quad (2.115)$$

$$(\alpha^8)^{-1} = \alpha^7 \quad (2.116)$$

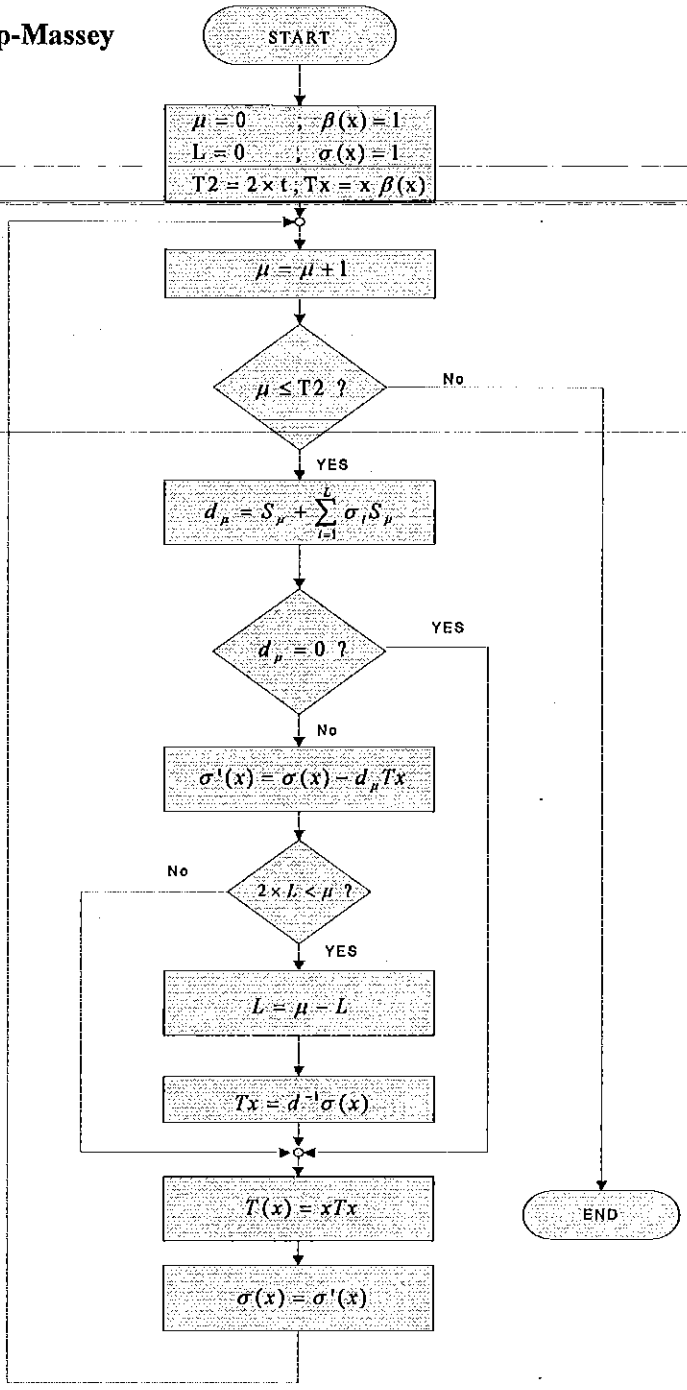
ดังนั้นโพลิโนเมียลของ ความผิดพลาดที่แก้ไข คือ

$$\hat{e}(X) = X^{11} + X^8 + X^7 \quad (2.117)$$

ถอดรหัสขอมูลจะทำการแก้ไขเวกเตอร์ที่รับได้เป็น

$$\hat{v}(X) = r(X) + \hat{e}(X) = X^{13} + X^{11} + X^8 + X^7 + X^5 + X^4 + X^3 \quad (2.118)$$

อัลกอริทึม Berlekamp-Massey



รูปที่ 2.2 Flowchart การถอดรหัสสำหรับ BCH Code

การเข้ารหัสจะมีการคำนวณบิตตรวจสอบหรือพาริตี้เพื่อใช้ตรวจสอบและแก้ไขบิตที่ผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลโดยใช้การเข้ารหัส-ถอดรหัส 2 แบบนั่นคือรหัสแฮมมิ่งและรหัสบีซีเอชซึ่งรหัสเหล่านี้เมื่อนำมาใช้ในการรับ-ส่งข้อมูลจะทำให้ประสิทธิภาพในการรับส่งข้อมูลดียิ่งขึ้น ดังจะกล่าวต่อไปถึงโครงสร้างการทำงาน และการออกแบบการเขียนโปรแกรมจำลองการเข้ารหัสและการถอดรหัสสัญญาณแบบ Hamming Code และแบบ BCH Code ในบทถัดไป

บทที่ 3

การออกแบบโครงงาน และวิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงโครงสร้างการทำงาน และการออกแบบการเขียนโปรแกรมจำลองการเข้ารหัสและการถอดรหัสแบบ Hamming Code และแบบ BCH Code โดยใช้โปรแกรม MATLAB และได้อธิบายวิธีการสร้าง Graphic User Interfaces เพื่อความสะดวกในการใช้งานเพื่อแสดงค่าต่าง ๆ โดยจะแบ่งออกเป็นขั้นตอนดังนี้ คือ

3.1 ขั้นตอนการออกแบบโปรแกรม

3.1.1 การสร้างสัญญาณ

ขั้นตอนแรก ในระบบสื่อสาร การส่งข่าวสารจากแหล่งกำเนิดสัญญาณไปยังอุปกรณ์ภาครับจะต้องทำการสร้างสัญญาณขึ้นมา เพื่อนำสัญญาณที่สร้างขึ้นมาเข้ารหัสแบบ Hamming Code และ BCH Code ก่อนที่จะทำการส่งไปยังปลายทาง ซึ่งการสร้างสัญญาณสามารถทำได้โดยใช้วิธีการสุ่มรหัสของสัญญาณขึ้นมา

3.1.2 การเข้ารหัสช่องสัญญาณ

ขั้นที่สอง ในระบบสื่อสารการส่งข่าวสารจากแหล่งกำเนิดสัญญาณไปยังอุปกรณ์ภาครับนั้น ระบบต้องการส่งสัญญาณ ไปในยังภาครับให้มีปริมาณที่มาก มีความรวดเร็ว และให้ได้ข้อมูลที่มีความถูกต้องที่สุด ดังนั้นจึงต้องนำสัญญาณที่ได้จาการสุ่มมาทำการเข้ารหัสแบบ Hamming Code หรือ BCH Code ในกระบวนการนี้จะมีการเพิ่มพาริตี้หรือบิตตรวจสอบเข้าไปรวมกับสัญญาณที่สร้างขึ้นมา ก่อนที่จะทำการส่งไปยังปลายทางหรืออุปกรณ์ภาครับขั้นตอนนี้ เรียกว่าการเข้ารหัส (Encode)

3.1.3 การสร้างสัญญาณรบกวน

ขั้นที่สาม ในการส่งสัญญาณโดยผ่านช่องสัญญาณสื่อสารมักจะมีความผิดพลาดเกิดขึ้นเนื่องจากสัญญาณรบกวน ซึ่งจะส่งผลกระทบต่อระบบทำให้สัญญาณที่รับได้ที่ปลายทางเกิดการผิดเพี้ยนไปจากเดิม การสร้างสัญญาณรบกวนสามารถทำได้โดยใช้วิธีการสุ่ม เช่นเดียวกันกับการสร้างสัญญาณที่ใช้ในการส่ง การเพิ่มสัญญาณรบกวนเข้าไปในระบบทำเพื่อใช้เป็นข้อพิสูจน์ในการแก้ไขบิตที่ผิดพลาดที่ภาครับ ว่ามีความสามารถในการแก้ไขหรือไม่เมื่อเกิดความผิดพลาดขึ้น

3.1.4 การรวมสัญญาณที่เข้ารหัสกับสัญญาณรบกวน

ขั้นที่สี่ ในการส่งสัญญาณโดยผ่านช่องสัญญาณสื่อสาร จะนำข้อมูลที่ผ่านการเข้ารหัส Hamming Code หรือ BCH Code แล้วมารวมกับสัญญาณรบกวน ซึ่งในขั้นตอนนี้จะเป็นขั้นตอนที่สมมุติว่าในกระบวนการส่งข้อมูลของระบบสื่อสารมีความผิดพลาดขึ้น เพื่อให้ส่วนของการถอดรหัสที่ภาครับ ทำการแก้ไขข้อมูลที่ผิดพลาดให้มีความถูกต้อง

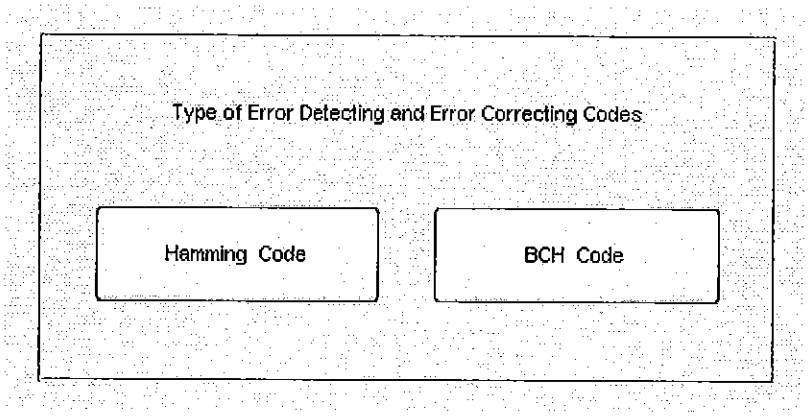
3.1.5 การแก้ไขข้อมูลที่ผิดพลาดตรงปลายทาง

ในขั้นสุดท้าย ในส่วนของภาครับจะรับสัญญาณที่มีทั้งสัญญาณจริง บิตตรวจสอบ และสัญญาณรบกวน ดังนั้นจึงทำการเขียนโปรแกรมเพื่อใช้ในการแก้ไขบิตผิดพลาด โดยใช้วิธีการของ Hamming Decode หรือ BCH Decode

3.2 การออกแบบ Graphic User Interfaces และขั้นตอนการดำเนินงาน

3.2.1 ขั้นตอนการดำเนินงาน

1. ทำการสร้างปุ่ม Push Button เพื่อใช้ในการเลือกการเข้ารหัสแบบต่างๆ ได้แก่ Hamming Code และ BCH Code ดังแสดงไว้ในรูปที่ 3.1



รูปที่ 3.1 Graphic User Interfaces ในการเลือกการเข้ารหัสและถอดรหัสสำหรับ Hamming Codeและ BCH Code

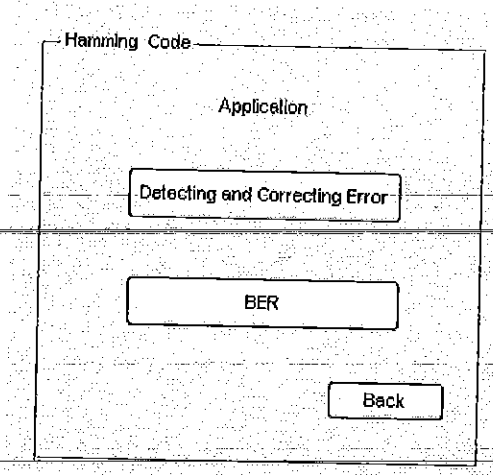
จากรูปที่ 3.1 เป็นรูปแบบของ Graphic User Interfaces ในการเลือกการเข้ารหัสและถอดรหัสสำหรับ Hamming Codeและ BCH Code ที่พร้อมสำหรับการใช้งาน เมื่อทำการคลิกที่ปุ่ม Hamming Code หรือ BCH Code จะปรากฏหน้าต่างของ Application ของรหัสที่ทำการเลือก ดังแสดงในหัวข้อที่ 3.2.2

3.2.2 Application

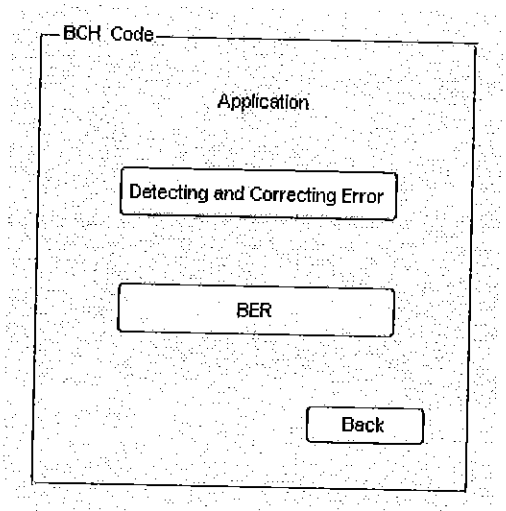
ในขั้นตอนนี้จะทำการสร้างหน้าต่างในการแดงตัวอย่างการแก้ไขบิตที่ผิดพลาด และการแสดงกราฟการลดระดับอัตราการเกิดข้อมูลผิดพลาด มีขั้นตอนดังนี้

1. ทำการสร้างปุ่ม Push Button เพื่อแสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาด
2. ทำการสร้างปุ่ม Push Button เพื่อแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด
3. ทำการสร้างปุ่ม Push Button สำหรับย้อนกลับไปยังหน้าต่างเลือกการเข้ารหัสและถอดรหัส

ขั้นตอนดังกล่าวแสดงไว้ดังรูปที่ 3.2



(ก) Application สำหรับ Hamming Code



(ข) Application สำหรับ BCH Code

รูปที่ 3.2 Graphic User Interfaces ในการเลือก Application สำหรับ Hamming Code และ BCH Code

จากรูปที่ 3.2 เป็นรูปแบบของ Graphic User Interfaces ในการเลือก Application สำหรับ Hamming Code และ BCH Code เมื่อคลิกที่ปุ่ม Detecting and Correcting Error ของรหัสทั้งสองแบบ จะปรากฏหน้าต่างของการตรวจสอบการแก้ไขข้อมูลผิดพลาดของรหัสที่เลือกขึ้นมา ซึ่งแสดงในหัวข้อที่ 3.2.3 และเมื่อคลิกที่ปุ่ม BER จะปรากฏหน้าต่างเพื่อดูกราฟการลดอัตราการเกิดข้อมูลผิดพลาด (BER) ซึ่งแสดงในหัวข้อ 3.2.4

3.2.3 Detecting and Correct Error

เป็นการแสดงตัวอย่างการเข้ารหัสและถอดรหัส ที่กำหนดให้มีจำนวนของบิตข้อมูลที่เป็นสัญญาณในการส่งไม่มากนัก เพื่อใช้ตรวจสอบการถอดรหัสสัญญาณที่ปลายทางว่ามีความสามารถใน

การแก้ไขบิตผิดพลาดหรือไม่ เมื่อทำการคลิกที่ปุ่ม Detecting and Correcting Error จะปรากฏหน้าต่าง ดังแสดงในรูปที่ 3.3

ในขั้นตอนนี้จะทำการสร้างหน้าต่างในการแสดงตัวอย่างการแก้ไขข้อมูลผิดพลาดของ Hamming Code และ BCH Code มีขั้นตอนดังต่อไปนี้

1. ทำการสร้างช่อง Edit Text สำหรับใส่บิตข้อมูลที่มีจำนวนไม่มาก เพื่อใช้ในการแสดงการทดสอบการถอดรหัสของภาครับ ว่ามีความสามารถในการแก้ไขบิตผิดพลาดได้หรือไม่

2. ทำการสร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด ซึ่งก็คือ Probability of bit error ซึ่งค่านี้จะนำไปคำนวณเพื่อหาความผิดพลาดที่เกิดขึ้น

3. ทำการสร้างปุ่ม Pop-up Menu เพื่อเลือก Type แบบต่างๆของ Hamming Code หรือ BCH Code

4. ทำการสร้างปุ่ม Push button เพื่อนำข้อมูลที่ต้องการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดไปคำนวณ

5. ทำการสร้างช่อง Edit Text สำหรับแสดงข้อมูลดังนี้

5.1 บิตข้อมูลที่ได้จากการสุ่ม

5.2 บิตข้อมูลที่ผ่านการเข้ารหัส

5.3 บิตข้อมูลผิดพลาดที่ได้จากการสุ่ม

5.4 บิตข้อมูลที่ผ่านการเข้ารหัสรวมกับบิตข้อมูลที่ผิดพลาด

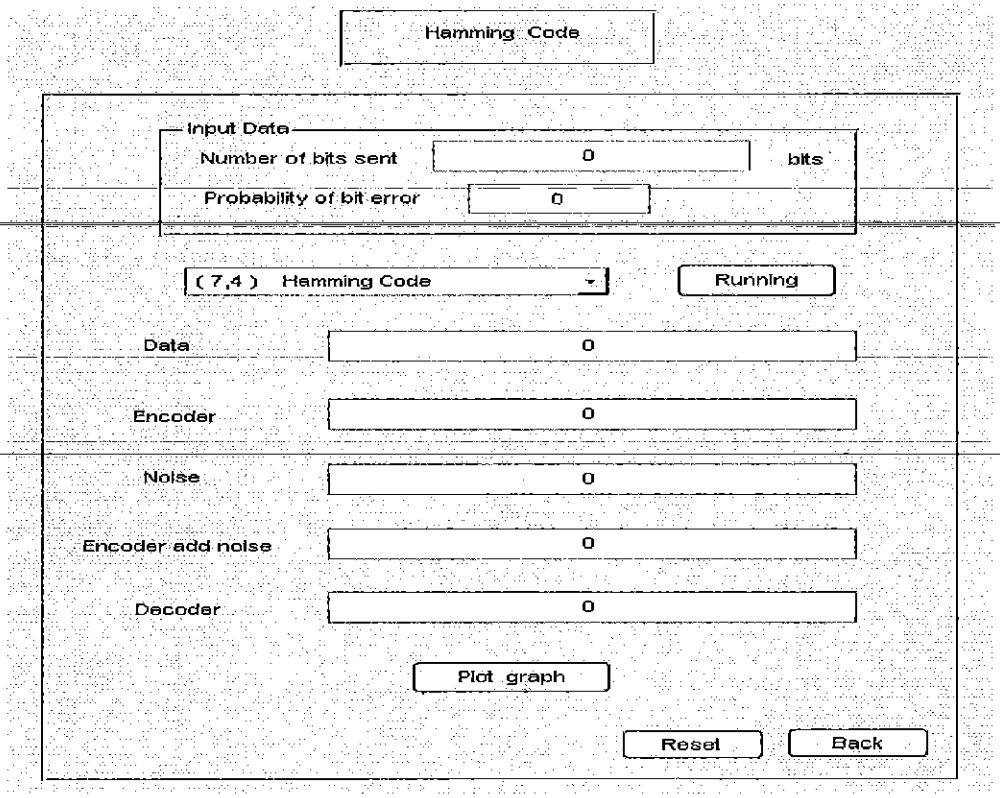
5.5 บิตข้อมูลที่ผ่านการถอดรหัส

6. ทำการสร้างปุ่ม Push Button สำหรับวาดกราฟสัญญาณที่ได้จาการเข้ารหัสและถอดรหัส

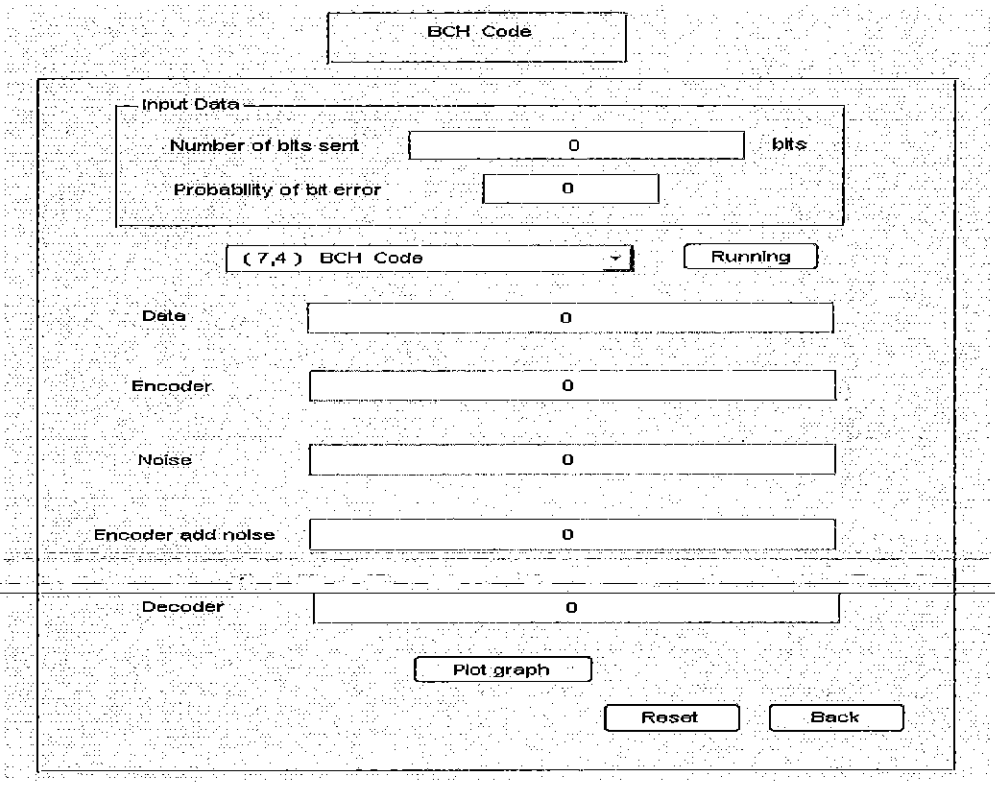
7. ทำการสร้างปุ่ม Push Button สำหรับ Reset ค่าต่างๆที่โปรแกรมคำนวณ เพื่อพร้อมที่จะใช้งานใหม่

8. ทำการสร้างปุ่ม Push Button สำหรับ Back ไปสู่หน้าต่าง Application

ขั้นตอนดังกล่าวข้างต้นแสดงในรูปที่ 3.3



(ก) Hamming Code



(ก) BCH Code

รูปที่3.3 Graphic User Interfaces แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับ Hamming Code และ BCH Code

3.2.4 BER

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

ขั้นตอนการดำเนินงาน

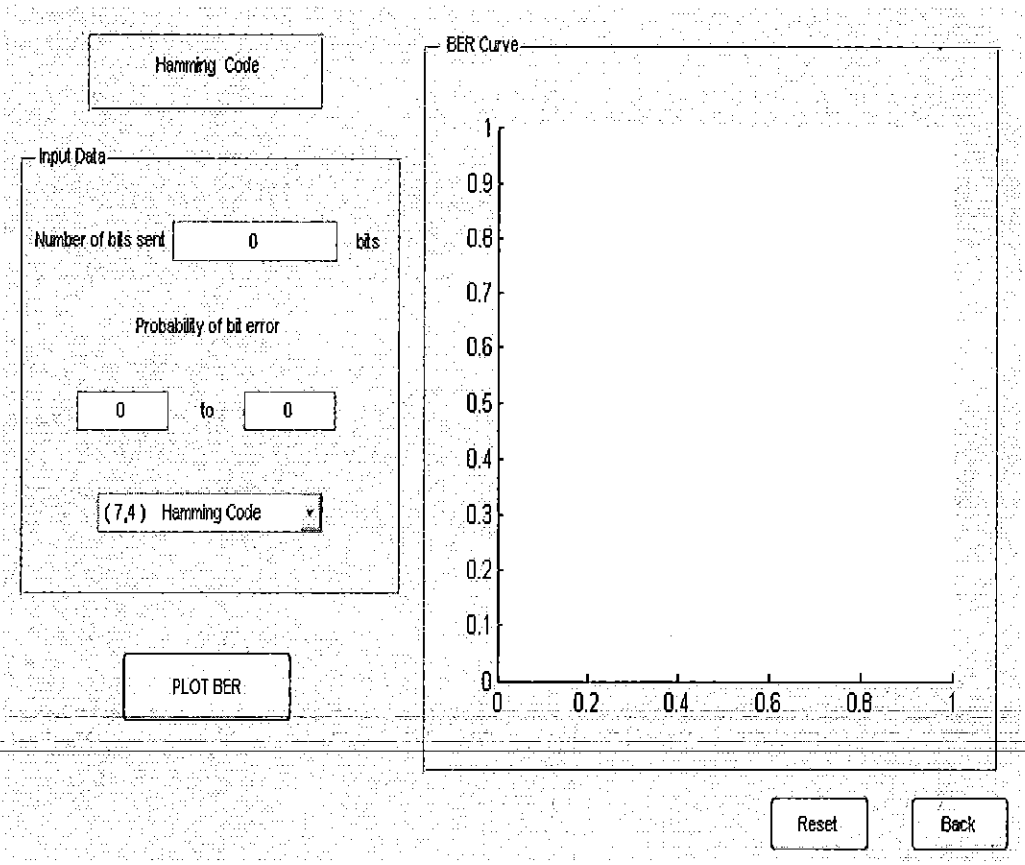
1. ทำการสร้างช่อง Edit Text สำหรับใส่บิตข้อมูล สำหรับส่วนนี้จะเป็นบิตข้อมูลที่ใช้ในการส่งสัญญาณ

2. ทำการสร้างช่อง Edit Text สำหรับใส่ความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of bit error)

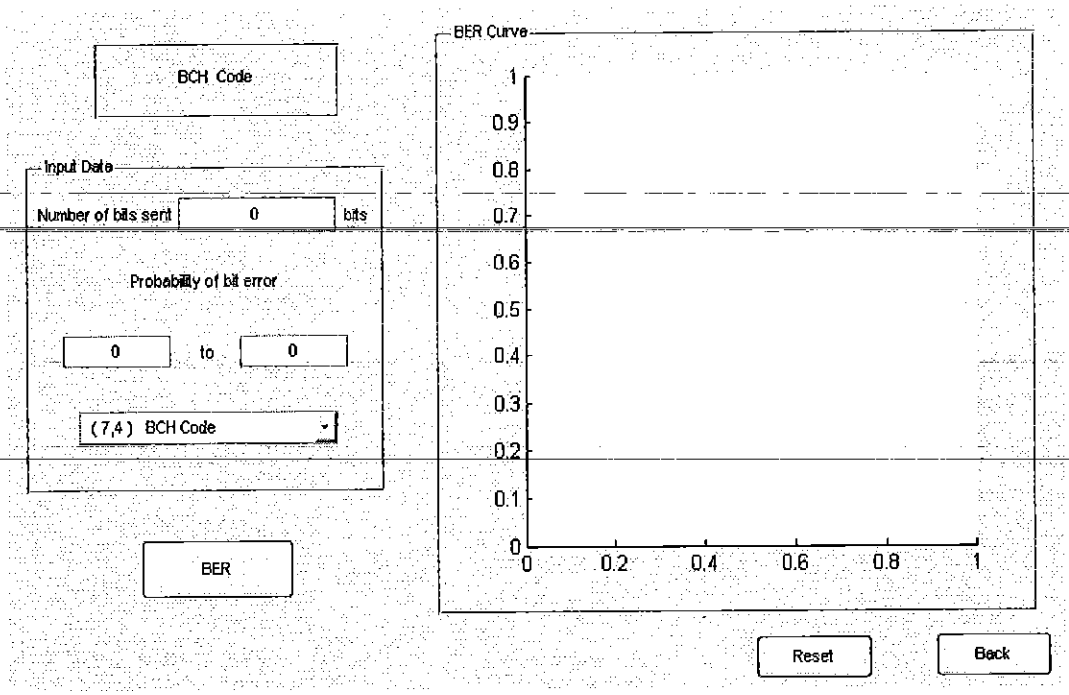
3. ทำการสร้างปุ่ม Push Button เพื่อนำข้อมูลที่ต้องการส่ง และความน่าจะเป็นในการเกิดข้อมูลผิดพลาดไปคำนวณ

4. ทำการสร้าง Axes เพื่อแสดงกราฟการลดระดับการเกิดข้อมูลผิดพลาดของ Hamming Code และ BCH Code

ขั้นตอนดังกล่าวแสดงดังรูปที่ 3.4



(ก) Hamming Code



(ข) BCH Code

รูปที่ 3.4 Graphic User Interface แสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาดสำหรับ Hamming Code และ BCH Code

ในบทที่ 3 ได้แสดงหลักการทำงานของโปรแกรม และได้อธิบายถึงการสร้าง Graphic User Interfaces เพื่อใช้ในการแสดงการทำงานของโปรแกรมของ Hamming Code และ BCH Code ในรูปแบบที่เข้าใจง่าย และสะดวกต่อการใช้งาน ซึ่งสามารถแบ่งได้เป็น

1. หน้าต่างที่ใช้แสดงการเลือกการเข้ารหัสและถอดรหัสของ Hamming Code และ BCH Code
2. หน้าต่างที่ใช้แสดงการแก้ไขข้อมูลที่ผิดพลาด
3. หน้าต่างที่ใช้แสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด

ในบทต่อไปจะแสดงผลการทดลองโปรแกรมการคำนวณ และแสดงค่าต่างๆ ที่ได้ออกแบบไว้ในบทนี้ โดยการทดลองใส่บิตข้อมูลเข้าไปในโปรแกรม เพื่อทดสอบการทำงานของโปรแกรม ทั้งความสามารถในการแก้ไขบิตที่ผิดพลาดและกราฟแสดงการลดระดับอัตราการเกิดบิตผิดพลาดสำหรับ Hamming Code และ BCH Code

บทที่ 4
ผลการดำเนินโครงงาน

ในบทนี้จะทำแสดงผลการทดลองในการใส่บิตข้อมูลเข้าไปในโปรแกรมจำลองการเข้ารหัสถอดรหัสของ Hamming Code และ BCH Code ซึ่งสามารถแสดงการแก้ไขข้อมูลผิดพลาด การลดระดับอัตราการเกิดข้อมูลผิดพลาด และการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาดของรหัสที่ทำการทดลอง โดยแสดงผลการทดลองในรูปแบบของ Graphic User Interfaces

4.1 โปรแกรมแสดงการแก้ไขข้อมูลผิดพลาดของ Hamming Code และ BCH Code

4.1.1 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม

1. ในขั้นตอนแรก เมื่อทำการเปิดหน้าต่าง Detecting and Correcting Error จะปรากฏหน้าต่างขึ้นมาดังรูปที่ 4.1 ซึ่งรายละเอียดในการใส่ค่าต่างๆมีดังนี้

1.1 ช่อง Number of bits sent คือ ช่องใส่ค่าบิตที่ต้องการส่ง

1.2 ช่อง Probability of bit error คือช่องใส่ความน่าจะเป็นในการเกิดบิตผิดพลาด

1.3 ช่องเลือก Type แต่ละแบบของรหัส Hamming Code และ BCH Code

1.4 ปุ่ม Running จะนำค่าบิตที่ต้องการส่งและความน่าจะเป็นในการเกิดบิตที่ผิดพลาดเข้าสู่โปรแกรมการคำนวณ และแสดงผล

1.5 ช่อง Data จะแสดงผลการสุ่มสัญญาณในการส่งแต่ละครั้ง

1.6 ช่อง Encoder จะนำสัญญาณจากการสุ่มมาทำการเข้ารหัส

1.7 ช่อง Noise จะแสดงผลการสุ่มของบิตผิดพลาดที่เกิดขึ้น

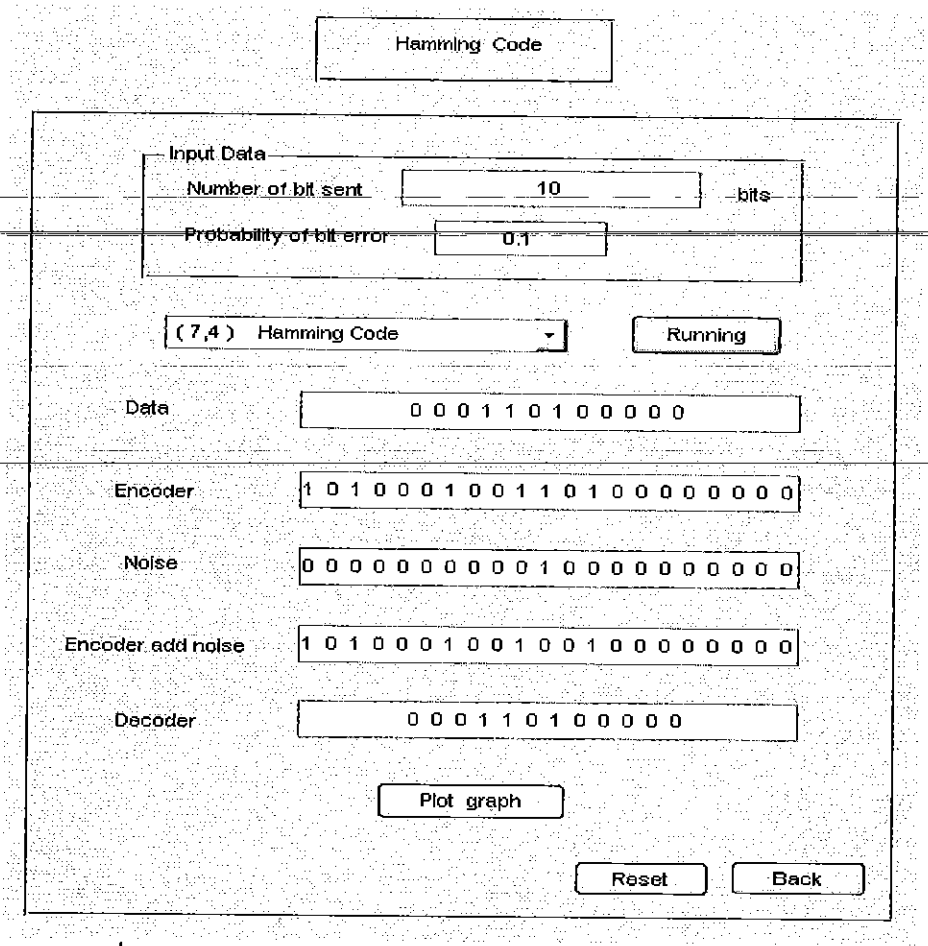
1.8 ช่อง Encoder add noise จะนำค่าสัญญาณที่ได้จากการเข้ารหัสมารวมกับ Noise ที่เกิดขึ้น เพื่อทำให้เกิดการผิดพลาดของข้อมูล

1.9 ช่อง Decoder แสดงการถอดรหัส ทำให้ได้สัญญาณเดิมที่ส่งกลับมา

1.10 ปุ่ม Plot graph แสดงกราฟจากการเข้ารหัสและถอดรหัสในรูปของกราฟ

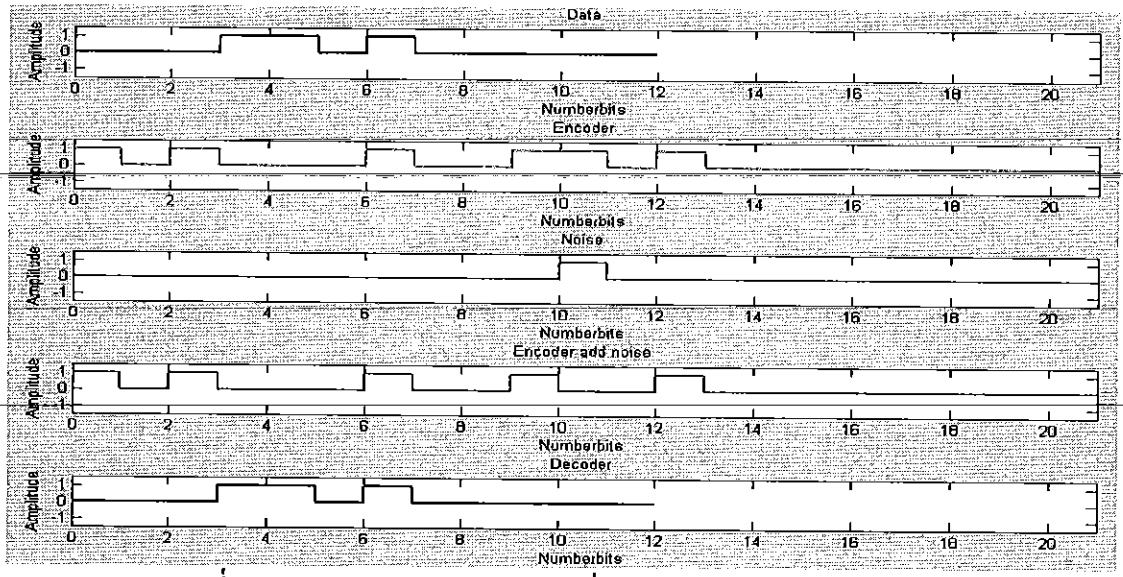
1.11 ปุ่ม Reset เมื่อกดปุ่มนี้จะทำการ Reset ค่าในหน้าต่างแสดงการทำงาน เพื่อพร้อมสำหรับการทำงานต่อไป

1.12 ปุ่ม Back เมื่อทำการกดปุ่ม Back จะกลับไปยังหน้าต่างหลัก



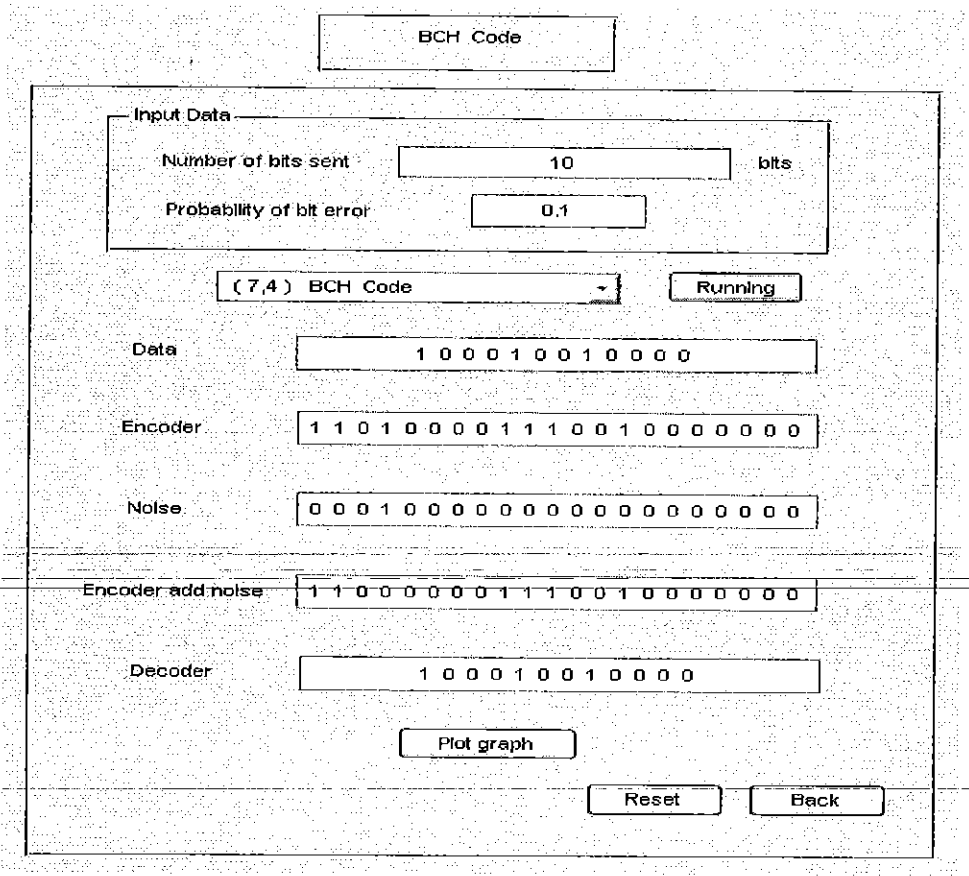
รูปที่ 4.1 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับ Hamming Code

จากรูปที่ 4.1 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับ(7,4)Hamming Code จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการ (Number of bits sent) เท่ากับ 10 บิต และความน่าจะเป็นในการเกิดบิตผิดพลาด (Probability of bit error) เท่ากับ 0.1 เมื่อกดปุ่ม Running ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างในรูปที่ 4.1 เมื่อใช้ (7,4)Hamming Code ส่งข้อมูลจำนวน 10 บิต ข้อมูลที่ได้จากการสุ่มจะมีเท่ากับ 12 บิต เนื่องจากส่งข้อมูลครั้งละ 4 บิตในหนึ่งบล็อก ในกรณีนี้จึงต้องส่งทั้งหมด 3 บล็อก ทำให้ได้บิตที่ส่งทั้งหมด 12 บิต โดยบิตที่เกินมาจากข้อมูลที่ส่งจริงนั้นกำหนดให้มีค่าเท่ากับ 0 บิต ข้อมูลที่ได้จากการสุ่มนี้จะเข้าไปทำการเข้ารหัส โดยจะมีพาริตี้บิตเพิ่มขึ้นมาดังแสดงในช่อง Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดแสดงในช่อง Noise เมื่อมีบิตผิดพลาดเกิดขึ้นโปรแกรมจะทำการรวมข้อมูลที่ผ่านการเข้ารหัสและบิตผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดการผิดพลาดระหว่างการส่งข้อมูล จากนั้นจะเข้าสู่ขั้นตอนการถอดรหัส เพื่อแก้ไขบิตผิดพลาดที่เกิดขึ้นดังแสดงในช่อง Decoder เมื่อเปรียบเทียบข้อมูลที่ส่งมาจากภาคส่งและภาครับ พบว่าโปรแกรมสามารถแก้ไขบิตผิดพลาดให้กลับมาดังเดิมได้ แต่มีขีดจำกัดในเรื่องความสามารถแก้ไขบิตผิดพลาดได้เพียงบิตเดียว เมื่อกดปุ่ม Plot graph จะปรากฏหน้าต่างดังรูปที่ 4.2



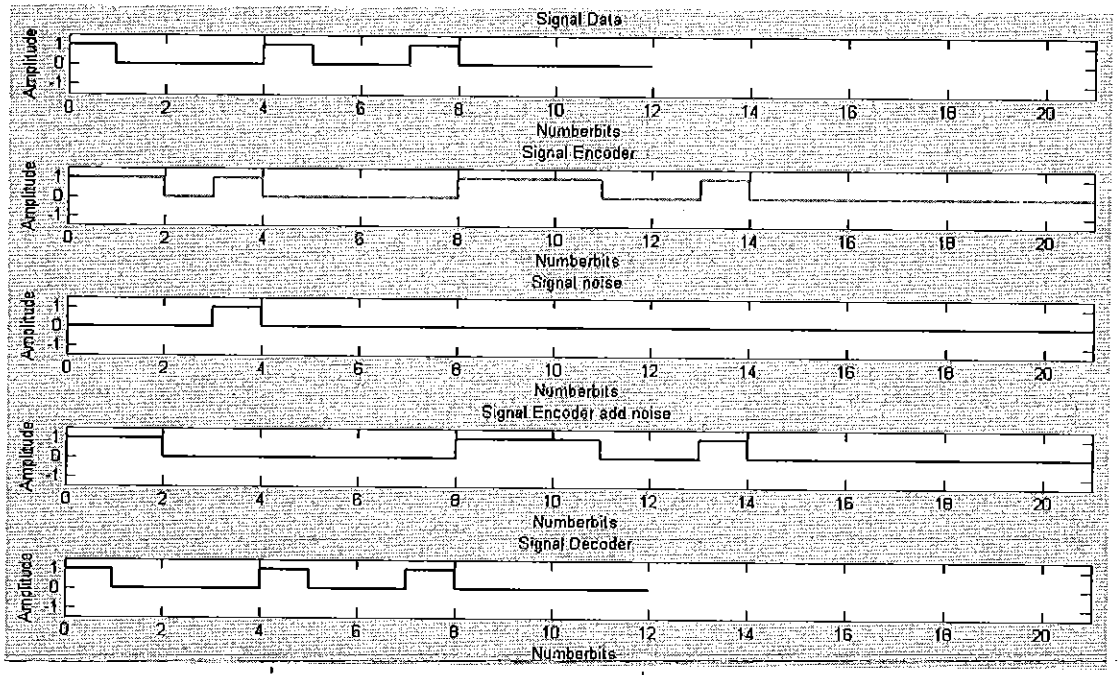
รูปที่ 4.2 กราฟแสดงการแก้ไขข้อมูลที่ผิดพลาดสำหรับ Hamming Code

จากรูปที่ 4.2 เป็นการนำผลลัพธ์จากการทดลองแก้ไขข้อมูลที่ผิดพลาดสำหรับ Hamming Code โดยนำมาแสดงในรูปแบบของกราฟ เพื่อให้เห็นภาพชัดเจนยิ่งขึ้น



รูปที่ 4.3 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับ BHC Code

จากรูปที่ 4.3 แสดงตัวอย่างการแก้ไขข้อมูลที่ผิดพลาดสำหรับ (7,4)BCH Code จากการทดสอบการใช้งานของโปรแกรม ซึ่งได้กำหนดข้อมูลที่ต้องการ (Number of bits sent) เท่ากับ 10 บิต และความน่าจะเป็นในการเกิดบิตผิดพลาด (Probability of bit error) เท่ากับ 0.1 เมื่อกดปุ่ม Running ข้อมูลที่ต้องการส่งเข้าสู่โปรแกรมเพื่อทำการประมวลผล และแสดงผลการทำงานออกมา จากตัวอย่างในรูปที่ 4.3 เมื่อใช้ (7,4)BCH Code ส่งข้อมูลจำนวน 10 บิต ข้อมูลที่ได้จากการสุ่มจะมีเท่ากับ 12 บิต เนื่องจากส่งข้อมูลครั้งละ 4 บิตใน 1 บล็อก ในกรณีนี้จึงต้องส่งทั้งหมด 3 บล็อก ทำให้ได้บิตที่ส่งทั้งหมด 12 บิต โดยบิตที่เกินมาจากข้อมูลที่ส่งจริงนั้นกำหนดให้มีค่าเท่ากับ 0 บิตข้อมูลที่ได้จากการสุ่มนี้จะเข้าไปทำการเข้ารหัส โดยจะมีพาริตี้บิตเพิ่มขึ้นมาดังแสดงในช่อง Encoder จากนั้นโปรแกรมจะทำการสุ่มบิตผิดพลาดแสดงในช่อง Noise เมื่อมีบิตผิดพลาดเกิดขึ้นโปรแกรมจะทำการรวมข้อมูลที่ผ่านการเข้ารหัสและบิตผิดพลาดเข้าด้วยกัน เพื่อทำให้เกิดการผิดพลาดระหว่างการส่งข้อมูล จากนั้นจะเข้าสู่ขั้นตอนการถอดรหัส เพื่อแก้ไขบิตผิดพลาดที่เกิดขึ้นดังแสดงในช่อง Decoder เมื่อเปรียบเทียบข้อมูลที่ส่งมาจากภาคส่งและภาครับ พบว่าโปรแกรมสามารถแก้ไขบิตผิดพลาดให้กลับมาดังเดิมได้ เมื่อกดปุ่ม Plot graph จะปรากฏหน้าต่างดังรูปที่ 4.4



รูปที่ 4.4 กราฟแสดงการแก้ไขข้อมูลที่ผิดพลาดสำหรับ BCH Code

จากรูปที่ 4.4 เป็นการนำผลลัพธ์จากการทดลองแก้ไขข้อมูลที่ผิดพลาดสำหรับ BCH Code โดยนำมาแสดงในรูปแบบของกราฟ เพื่อให้เห็นภาพชัดเจนยิ่งขึ้น

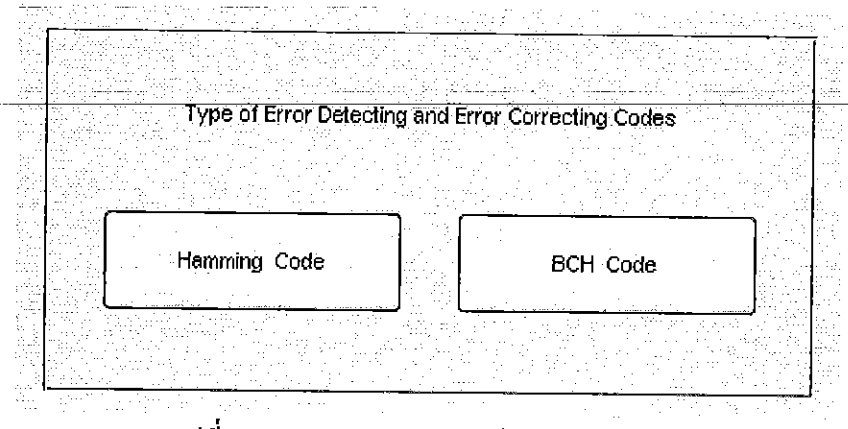
4.2 โปรแกรมแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด

4.2.1 รายละเอียดของโปรแกรมและขั้นตอนการรันโปรแกรม(Hamming Code)

1. ในขั้นตอนแรกต้องทำการรันโปรแกรมการเลือกชนิดของการเข้ารหัสช่องสัญญาณ แบบใด

1.1 Hamming Code

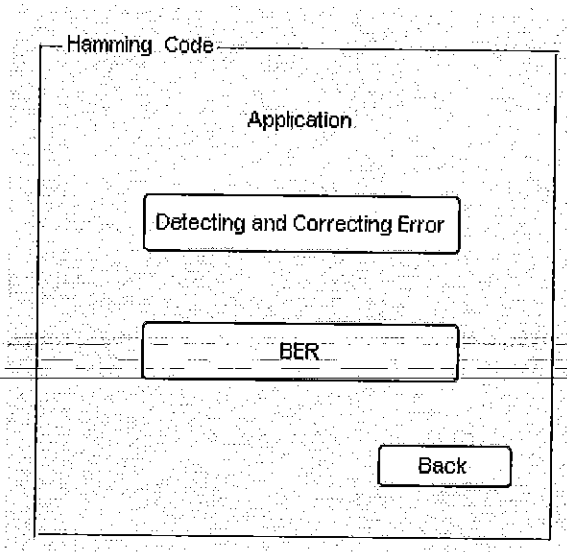
1.2 BCH Code



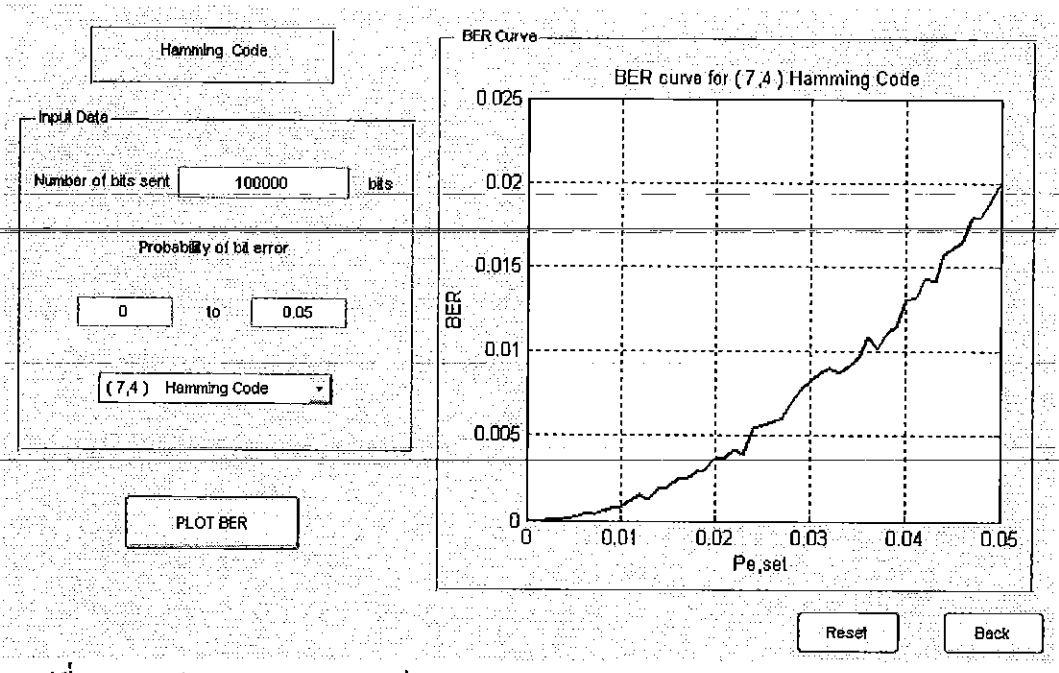
รูปที่ 4.5 แสดงชนิดของการเข้ารหัส-ถอดรหัส

2. เมื่อเลือกอัตราการเข้ารหัสช่องสัญญาณที่ต้องการ โดยการคลิกปุ่ม Hamming Code

3. จะปรากฏหน้าต่าง Application ดังรูปที่ 4.2 ปรากฏขึ้นมาให้คลิกปุ่ม BER ซึ่งเป็นหน้าต่างการลดอัตราการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.6

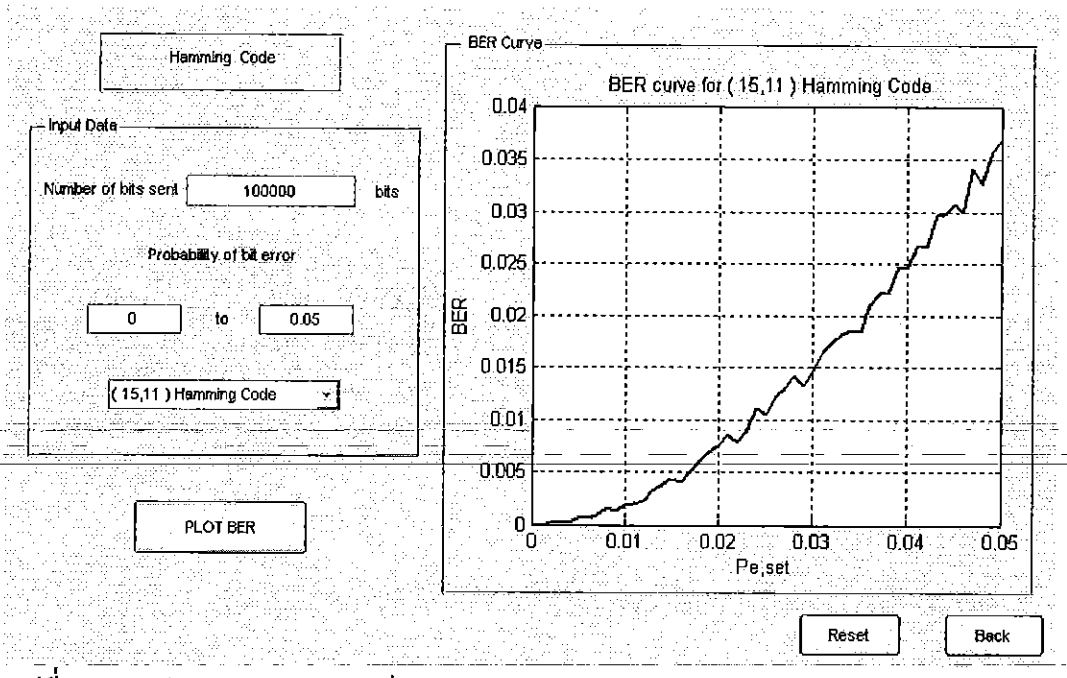


รูปที่ 4.6 แสดงหน้าต่าง Application



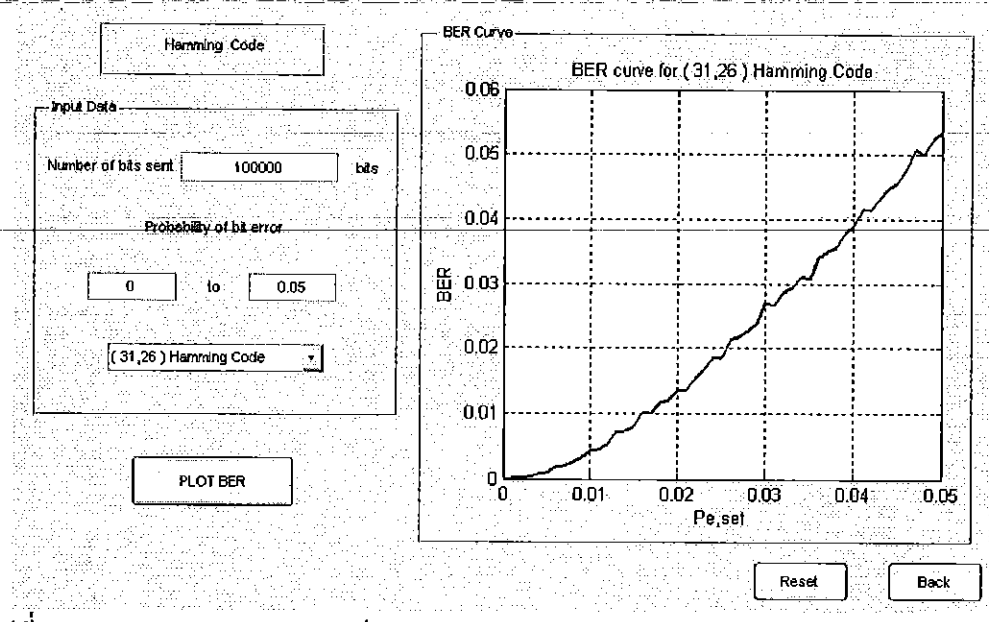
รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่าง Pe,set และ BER สำหรับ (7,4)Hamming Code

จากรูปที่ 4.7 เป็นการทดสอบการส่งข้อมูลจำนวน 100,000 บิต โดยส่งครั้งละ 4 บิต และกำหนดค่าความน่าจะเป็นในการเกิดบิตผิดพลาดมีค่าอยู่ในช่วง 0 ถึง 0.05 เห็นได้ว่าช่วง Pe,set ที่สนใจนี้สามารถแก้ไขบิตที่ผิดพลาดที่เกิดขึ้นได้ พิจารณาที่ Pe,set =0.02 จะได้ค่า BER= 0.0035 Hamming code (100,000 บิต)



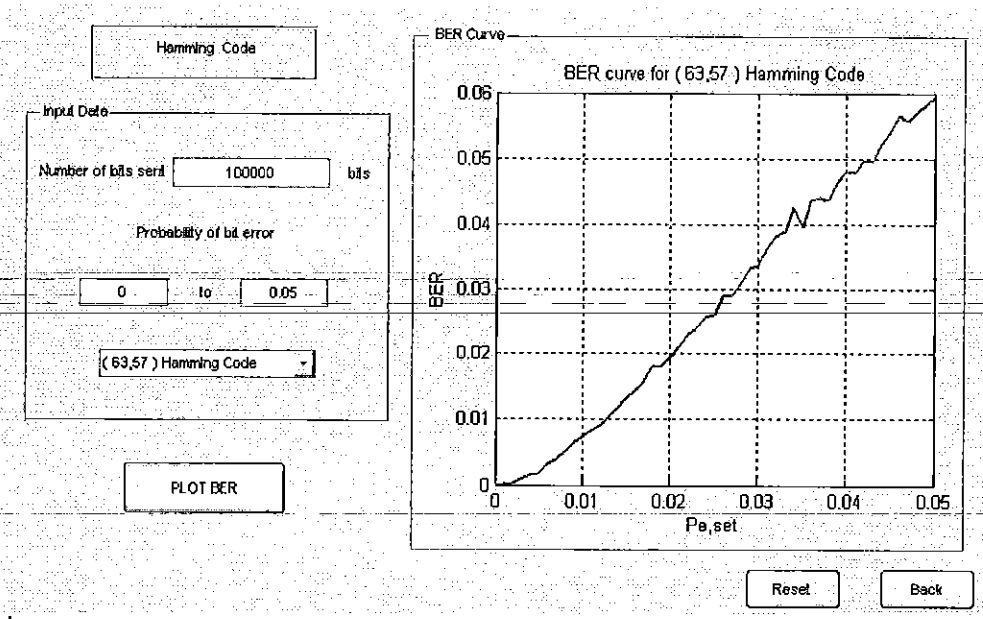
รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่าง Pe,set และ BER สำหรับ (15,11)Hamming Code

จากรูปที่ 4.8 เป็นการทดสอบการส่งข้อมูลจำนวน 100,000 บิต โดยส่งครั้งละ 11 บิต และกำหนดค่าความน่าจะเป็นในการเกิดบิตผิดพลาดมีค่าอยู่ในช่วง 0 ถึง 0.05 พิจารณาที่ Pe,set =0.02 จะได้ค่า BER=0.0075



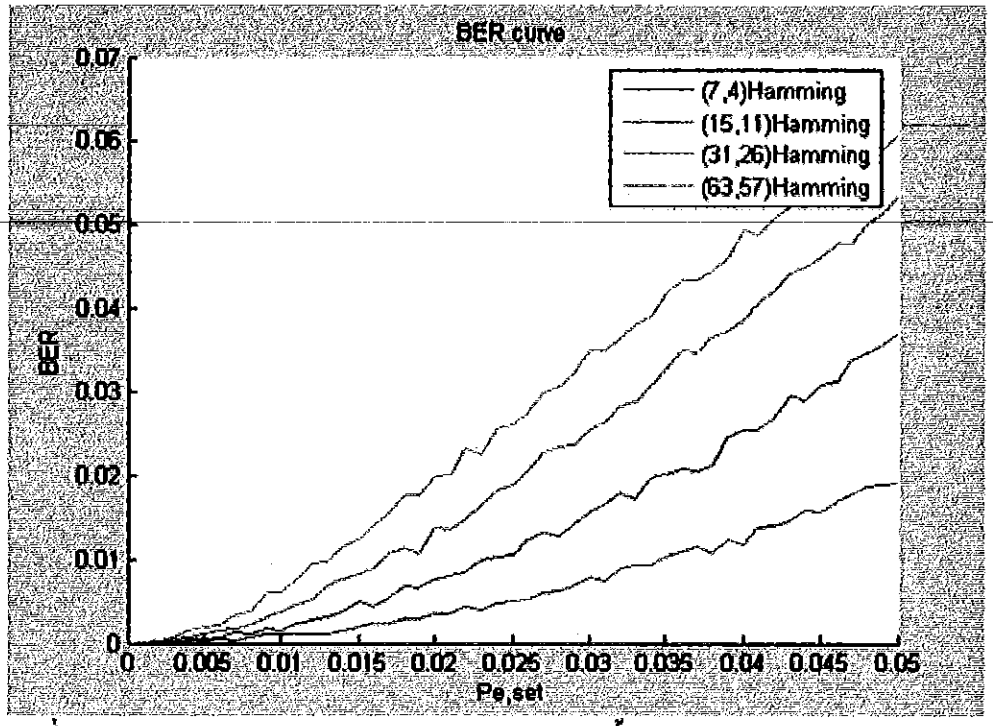
รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่าง Pe,set และ BER สำหรับ (31,26)Hamming Code

จากรูปที่ 4.9 เป็นการทดสอบการส่งข้อมูลจำนวน 100,000 บิต โดยส่งครั้งละ 26 บิต และกำหนดค่าความน่าจะเป็นในการเกิดบิตผิดพลาดมีค่าอยู่ในช่วง 0 ถึง 0.05 นี้สามารถแก้ไขบิตที่พิจารณาที่ Pe,set =0.02 จะได้ค่า BER=0.014



รูปที่ 4.10 กราฟแสดงการเปรียบเทียบระหว่าง Pe,set กับ BER สำหรับ(63,57)Hamming code

จากรูปที่ 4.10 เป็นการทดสอบการส่งข้อมูลจำนวน 100,000 บิต โดยส่งครั้งละ 57 บิต และกำหนดค่าความน่าจะเป็นในการเกิดบิตผิดพลาดมีค่าอยู่ในช่วง 0 ถึง 0.05 นี้สามารถแก้ไขบิตที่พิจารณาที่ Pe,set =0.02 จะได้ค่า BER=0.02



รูปที่ 4.11 กราฟแสดงการเปรียบเทียบค่า BER ของทั้ง 4 กรณีสำหรับ Hamming code (100,000 บิต)

จากรูปที่ 4.11 เห็นได้ว่า ทำการส่งข้อมูลจำนวน 100,000 บิต และกำหนดค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0 ถึง 0.05 แล้ว จากกราฟจะเห็นได้ว่าในช่วงค่าเริ่มต้นของ Pe,set ที่ Pe,set เท่ากับ 0 ถึง 0.0025 คำนวณ BER มีค่าใกล้เคียง 0 ทั้ง 4 กรณีทั้งนี้เพราะ Pe,set มีค่าน้อย โอกาสที่จะเกิดความผิดพลาดในการรับ-ส่งข้อมูลก็น้อยลงไปด้วยทำให้สามารถแก้ไขบิตผิดพลาดได้ ถ้าพิจารณาที่ Pe,set เท่ากับ 0.02 ของกราฟทั้ง 4 กรณีของ Hamming code พบว่าค่าความน่าจะเป็นในการเกิดบิตที่ผิดพลาด

BER สำหรับ (7,4)Hamming Code = 0.0035

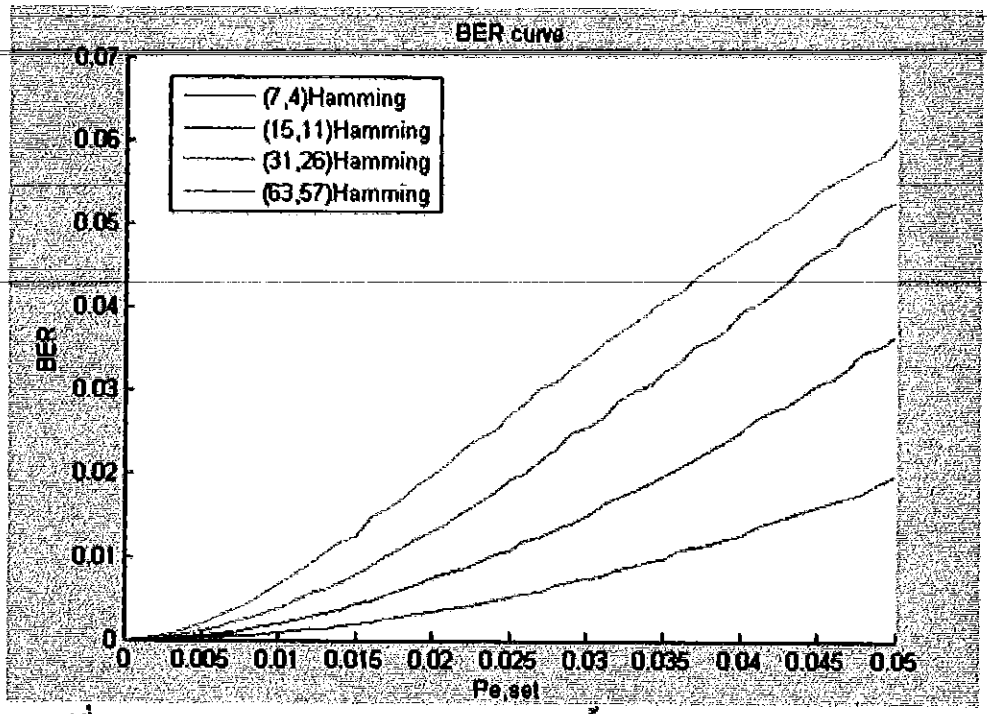
BER สำหรับ (15,11)Hamming Code = 0.0075

BER สำหรับ (31,26)Hamming Code = 0.014

BER สำหรับ (63,57)Hamming Code = 0.02

ค่า BER มีค่าน้อยที่สุดในการทดลองนี้คือBER สำหรับ (7,4)Hamming Code = 0.0035 คือมีความสามารถในการแก้ไขบิตที่ผิดได้ดีที่สุดใน 4 กรณีนี้ เพราะเมื่อพิจารณาการส่งข้อมูลทั้ง 4 กรณี ซึ่งมี

ความสามารถในการแก้ไขบิตผิดพลาดได้เพียงหนึ่งบิตเท่ากันทุกกรณี ดังนั้นกรณี (7,4)Hamming Code จะส่งข้อมูลทีละ 7 บิต โอกาสที่จะเกิดบิตที่ผิดพลาดจึงมีค่าน้อยกว่ากรณีอื่นๆ



รูปที่ 4.12 กราฟแสดงการเปรียบเทียบค่า BER ของทั้ง 4 กรณีสำหรับ Hamming code (1,000,000 บิต)

จากรูปที่ 4.12 จะเห็นได้ว่า เมื่อทำการส่งข้อมูลจำนวน 1,000,000 บิต และกำหนดค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0 ถึง 0.05 แล้ว จากกราฟจะเห็นได้ว่าเส้นกราฟดูเรียบขึ้น (smooth) เนื่องจาก

$$BER = (\text{จำนวนบิตที่ผิด})/(\text{จำนวนบิตข้อมูลทั้งหมด}) \qquad (4.1)$$

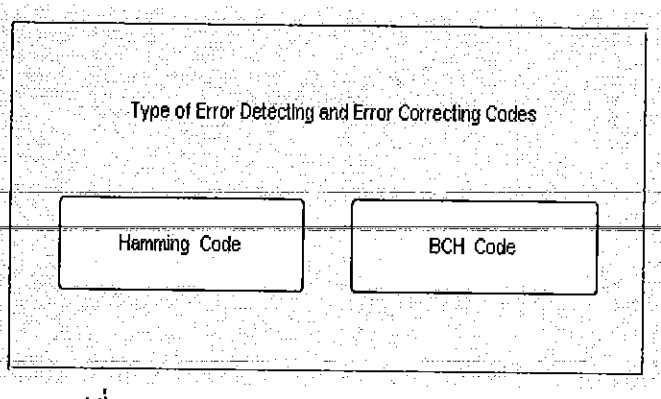
ดังนั้นเมื่อจำนวนบิตข้อมูลมีมาก ส่งผลให้ค่า *BER* มีความละเอียดมากขึ้น ด้วยเหตุนี้เส้นกราฟจึงดูเรียบขึ้นเมื่อเทียบกับกราฟแสดงการเปรียบเทียบค่า *BER* ของทั้ง 4 กรณีของ Hamming code (100,000 บิต) ดังรูปที่ 4.11

4.2.2 รายละเอียดของโปรแกรม และขั้นตอนการรันโปรแกรม (BCH code)

1. ในขั้นตอนแรก ทำการเลือกชนิดของการเข้ารหัส-ถอดรหัส ซึ่งมี 2 ประเภทดังนี้

   1.1 Hamming Code

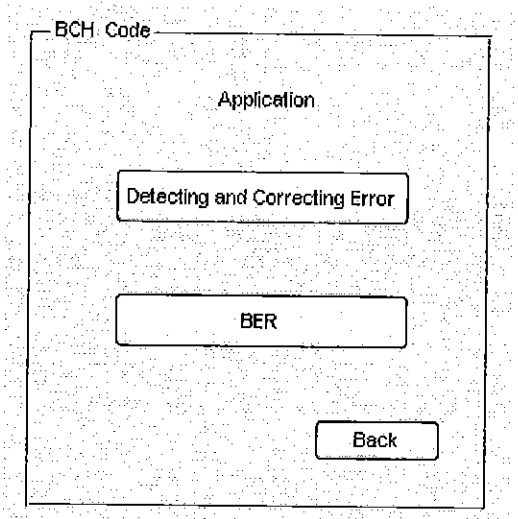
   1.2 BCH Code



รูปที่ 4.13 แสดงชนิดของการเข้ารหัส-ถอดรหัส

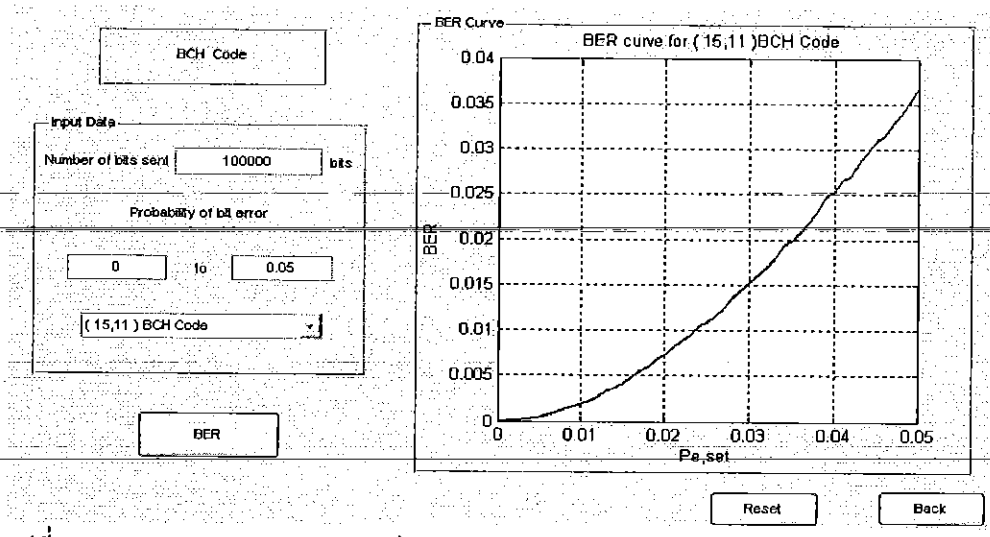
2. เมื่อเลือกชนิดของการเข้า-ถอดรหัสช่องสัญญาณที่ต้องการ ถ้าคลิกที่ปุ่ม BCH Code จะปรากฏหน้าต่างขึ้นมาดังรูปที่ 4.14

3. ทำการเลือก Application ของ BCH Code เมื่อคลิกที่ปุ่ม BER ซึ่งจะเป็นหน้าต่างการลดระดับอัตราการเกิดข้อมูลผิดพลาด ดังรูปที่ 4.15



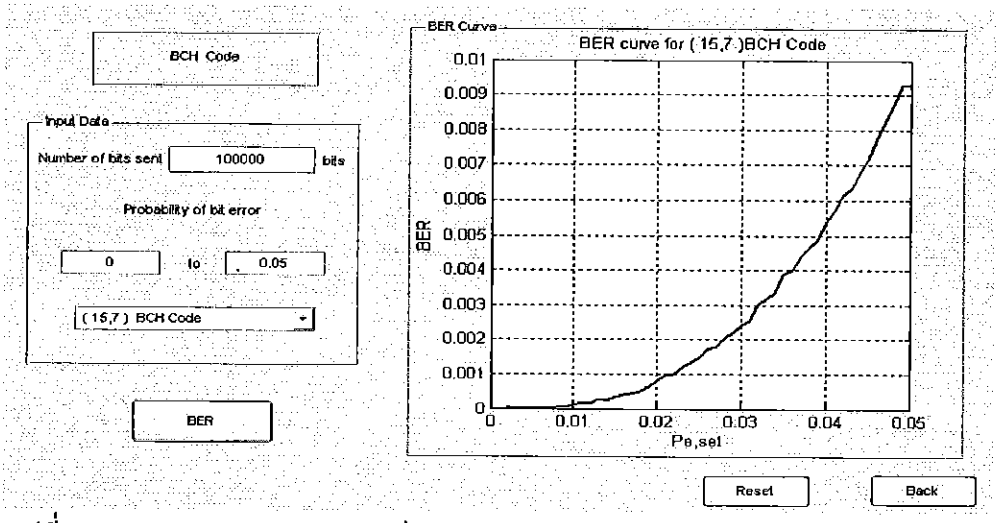
รูปที่ 4.14 แสดงหน้าต่าง Application ของ BCH Code

4. เมื่อทำการกดปุ่ม BER จะปรากฏหน้าต่าง BCH_ber ขึ้นมาดังรูปที่ 4.15 จากนั้นทำการทดสอบโปรแกรมด้วยการใส่ค่าจำนวนบิตที่ต้องการส่งในช่อง Number of bits sent เท่ากับ 100,000 บิต และกำหนดค่า Pe,set ให้มีค่าตั้งแต่ 0 ถึง 0.05 โดยอัตราการเพิ่มขึ้นของค่า Pe,set เพิ่มขึ้นครั้งละ 0.001 และเลือกใช้ (15,11)BCH code จากนั้นกดปุ่ม BER โปรแกรมจะประมวลผลการทำงาน ทำให้ได้กราฟดังรูปที่ 4.15



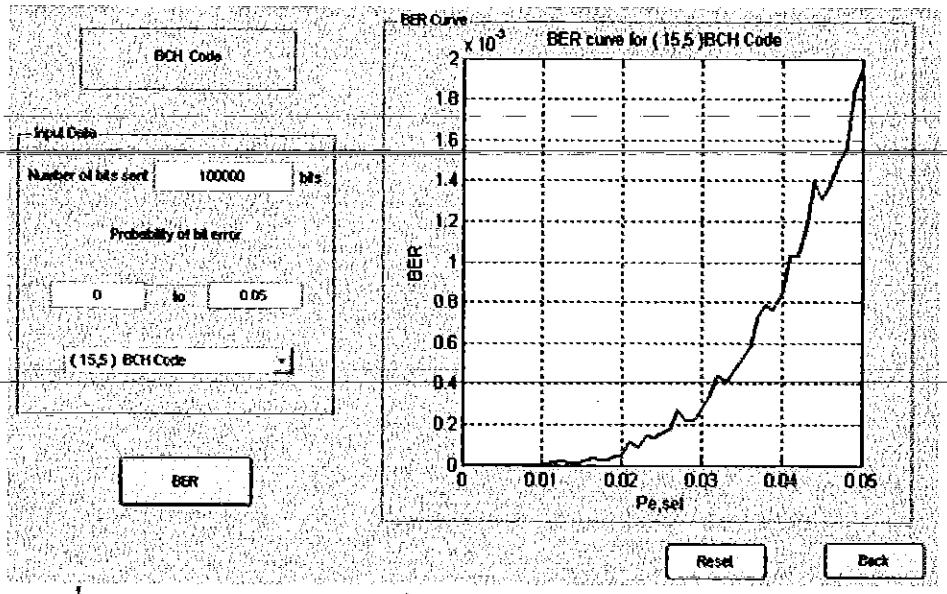
รูปที่ 4.15 แสดงกราฟความสัมพันธ์ระหว่าง BER กับ Pe,set สำหรับ (15,11)BCH Code

จากรูปที่ 4.15 จะเห็นว่า (15,11)BCH Code จากค่าความน่าจะเป็นในการเกิดความผิดพลาดของข้อมูลเมื่อมีการเข้ารหัส-ถอดรหัสแบบ (15, 11) BCH Code กำหนด Pe,set เท่ากับ 0 ถึง 0.05 พิจารณาที่ Pe,set เท่ากับ 0 ถึง 0.0025 ค่า BER มีค่าเป็นศูนย์แสดงว่าสามารถแก้ไขบิตผิดพลาดได้ และที่ Pe,set = 0.01 จะได้ค่า BER เท่ากับ 0.002



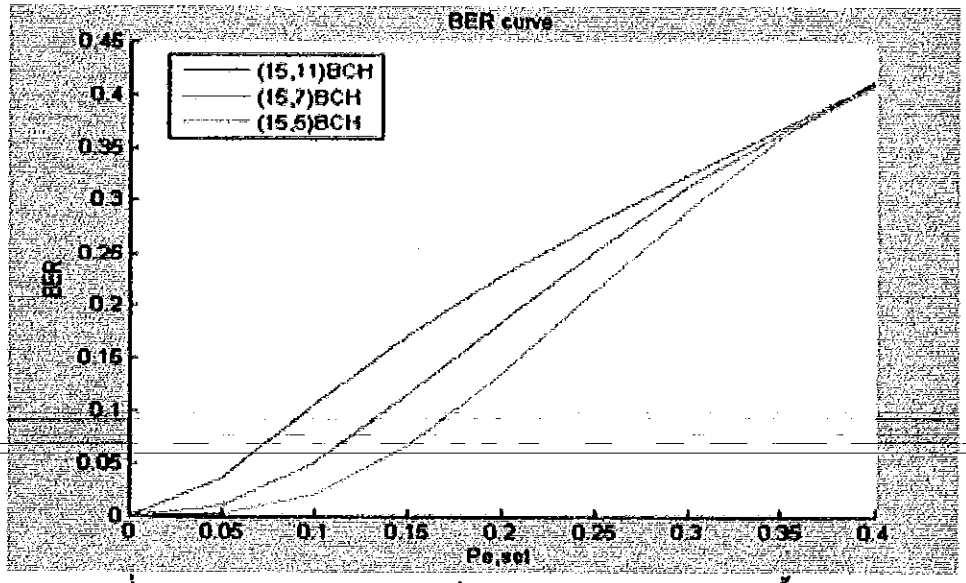
รูปที่ 4.16 แสดงกราฟความสัมพันธ์ระหว่าง BER กับ Pe,set สำหรับ (15,7)BCH Code

จากรูปที่ 4.16 จะเห็นว่า (15,7)BCH Code จากค่าความน่าจะเป็นในการเกิดความผิดพลาดของข้อมูลเมื่อมีการเข้ารหัส-ถอดรหัสแบบ (15,7) BCH Code กำหนด Pe,set เท่ากับ 0 ถึง 0.05 พิจารณาที่ Pe,set เท่ากับ 0 ถึง 0.007 ค่า BER มีค่าเป็นศูนย์แสดงว่าสามารถแก้ไขบิตผิดพลาดได้ดีกว่า (15,11)BCH Code สังเกตจากช่วงของ Pe,set ที่ทำให้ BER มีค่าเป็น 0 มีค่ามากขึ้น และที่ Pe,set =0.01 จะได้ค่า BER เท่ากับ 0.00015



รูปที่ 4.17 แสดงกราฟความสัมพันธ์ระหว่าง BER กับ Pe,set ของ (15,5)BCH Code

จากรูปที่ 4.17 จะเห็นว่า (15,5)BCH Code จากค่าความน่าจะเป็นในการเกิดความผิดพลาดของข้อมูลเมื่อมีการเข้ารหัส-ถอดรหัสแบบ (15,7) BCH Code กำหนด Pe,set เท่ากับ 0 ถึง 0.05 พิจารณาที่ Pe,set เท่ากับ 0 ถึง 0.012 ค่า BER มีค่าเป็นศูนย์แสดงว่าสามารถแก้ไขบิตผิดพลาดได้ดีที่สุด และที่ Pe,set = 0.01 จะได้ค่า BER เท่ากับ 0



รูปที่ 4.18 แสดงกราฟความสัมพันธ์ระหว่าง BER กับ Pe,set ของทั้ง 3 กรณี

จากรูปที่ 4.18 ส่งข้อมูลจำนวน 100,000 บิต และกำหนดค่าความน่าจะเป็นในการเกิดข้อมูลผิดพลาด (Probability of Bit Error) เท่ากับ 0 ถึง 0.05 พิจารณาที่ Pe,set = 0.05 ของกราฟทั้ง 4 กรณีของ BCH Code พบว่าความน่าจะเป็นในการเกิดบิตผิดพลาดมีค่าดังนี้

BER สำหรับ (15,11)BCH Code = 0.002

BER สำหรับ (15,7)BCH Code = 0.00015

BER สำหรับ (15,5)BCH Code = 0

กราฟจะเห็นได้ว่า BER ของ (15,5)BCH มีค่าต่ำกว่ากราฟอื่นๆของทั้ง 3 กรณี ดังแสดงไว้ข้างต้น พบว่า(15,5)BCH Code สามารถแก้ไขบิตที่ผิดได้ดีที่สุดใน 3 กรณีนี้ เพราะการส่งข้อมูลของทั้ง 3 กรณีส่งได้ครั้งละ 15 บิตเท่ากัน แต่มีความสามารถในการแก้ไขบิตที่ผิดพลาดที่ต่างกันนั่นคือ สำหรับ (15,11)BCH Code สามารถแก้ไขบิตผิดพลาดได้ 1 บิต สำหรับ (15,7)BCH Code สามารถแก้ไขบิตผิดพลาดได้ 2 บิตและสำหรับ (15,5)BCH Code สามารถแก้ไขบิตผิดพลาดได้ 3 บิต ด้วยเหตุนี้จึงทำให้ (15,5)BCH Code มีบิตตรวจสอบความผิดพลาดมาก ทำให้โอกาสในการตัดสินบิตผิดพลาดมีค่าน้อยกว่ากรณีอื่นๆ

ในบทที่ 4 นี้ ได้แสดง Graphic User Interfaces ที่ใช้แสดงผลการเข้ารหัส-ถอดรหัสของ Hamming Code และ BCH Code ตลอดจนแสดงกราฟเปรียบเทียบประสิทธิภาพในการแก้ไขข้อมูลที่ผิดพลาด ของรหัสที่ต้องการทดสอบ เพื่อใช้ในการตัดสินใจในการนำไปใช้ต่อไป

ในบทต่อไป จะเป็นการสรุปผลการดำเนินโครงงาน ปัญหาที่พบขณะดำเนินงาน และข้อเสนอแนะ

บทที่ 5
สรุปผลการดำเนินโครงงาน

5.1 ผลการดำเนินโครงงาน

โครงงานนี้เป็นการจำลองการเข้ารหัสและถอดรหัสของ Hamming Code และ BCH Code โดยใช้การเข้ารหัสช่องสัญญาณด้วยการทดลองส่งข้อมูล มีการสร้างสัญญาณรบกวน เพื่อทำให้มีความผิดพลาดเกิดขึ้น การเพิ่มบิตตรวจสอบเพื่อใช้ตรวจสอบความผิดพลาด การแก้ไขบิตที่ผิดพลาด และการถอดรหัสข้อมูล เพื่อตรวจสอบบิตที่เกิดการผิดพลาดแล้วสามารถแก้ไขข้อมูลที่ผิดพลาดให้กลับมาเป็นสัญญาณเดิมให้ถูกต้อง ซึ่งใช้โปรแกรม MATLAB ในการดำเนินโครงงาน และแสดงออกมาในรูปของ Graphic User Interfaces โดยมีการเข้ารหัสช่องสัญญาณ

1. การเข้ารหัส-ถอดรหัสของ(7,4) Hamming Code
2. การเข้ารหัส-ถอดรหัสของ(15,11) Hamming Code
3. การเข้ารหัส-ถอดรหัสของ(31,26) Hamming Code
4. การเข้ารหัส-ถอดรหัสของ(63,57) Hamming Code
5. การเข้ารหัส-ถอดรหัสของ (7,4) BCH Code
6. การเข้ารหัส-ถอดรหัสของ (15,5) BCH Code
7. การเข้ารหัส-ถอดรหัสของ (15,7) BCH Code
8. การเข้ารหัส-ถอดรหัสของ (15,11) BCH Code

โครงงานนี้ได้แสดงตัวอย่างการแก้ไขข้อมูลผิดพลาด และยังแสดงการลดระดับอัตราการเกิดข้อมูลผิดพลาด รวมถึงการเปรียบเทียบประสิทธิภาพการแก้ไขข้อมูลผิดพลาดโดยอาศัยการเข้ารหัส-ถอดรหัส 2 แบบคือ Hamming Code และ BCH Code โดยการส่งข้อมูลแต่ละครั้งผู้ใช้สามารถเลือกรูปแบบการเข้ารหัส-ถอดรหัสได้ว่าจะเลือกใช้แบบไหน เพื่อใช้ในการตัดสินใจเลือกได้ว่าการเข้ารหัสช่องสัญญาณแบบใดที่มีประสิทธิภาพในการแก้ไขข้อมูลผิดพลาดได้ดีที่สุด เพื่อที่จะนำไปใช้ในระบบการสื่อสารจริง ส่งผลให้การรับส่งข้อมูลมีประสิทธิภาพมากที่สุด

5.2 ปัญหาที่พบขณะดำเนินโครงงาน

1. เนื่องจากคำสั่งบางคำสั่งในโปรแกรม MATLAB ผู้ดำเนินโครงงานยังไม่มีความเข้าใจอย่างถ่องแท้ ดังนั้นจึงทำให้เกิดความล่าช้าในขณะดำเนินโครงงาน

2. เนื่องจากโครงงานนี้ใช้ Graphic User Interfaces ในการแสดงผลของการดำเนินงาน ดังนั้นเครื่องคอมพิวเตอร์ควรที่จะมีโปรแกรม MATLAB เวอร์ชัน 7 ขึ้นไป จึงจะสามารถแสดงผลการดำเนินงานออกมาได้

5.3 ข้อเสนอแนะ

1. ควรขอคำแนะนำในการดำเนินโครงงานจากอาจารย์ที่ปรึกษาโครงงาน เพื่อให้ได้งานที่มีคุณภาพ และเสร็จทันเวลาที่กำหนด

2. สามารถใช้ Help ในโปรแกรมแมทแลป (MATLAB Programming) ช่วยในการเขียนโปรแกรมได้โดยจะมีคำอธิบายเกี่ยวกับวิธีการเรียกใช้ฟังก์ชันต่างๆในโปรแกรม

3. โครงงานนี้สามารถนำไปศึกษาเพื่อประกอบการเรียนหรือเป็นสื่อการเรียนการสอนสำหรับผู้ที่สนใจ

เอกสารอ้างอิง

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

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

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

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

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

ประวัติผู้เขียนโครงงาน

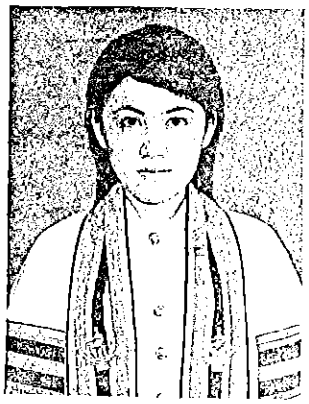
ชื่อ นางสาวชลธิชา ชัยชนะ

ภูมิลำเนา 258 หมู่ 1 ต.วงฆ้อง อ.พรหมพิราม จ.พิษณุโลก

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนเฉลิมขวัญสตรี
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **moo_may_1@hotmail.com**



ชื่อ นางสาวสุภัตรา ปินจันทร์

ภูมิลำเนา 101 หมู่ 6 ต.แม่พูล อ.ลับแล จ.อุตรดิตถ์

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนอุตรดิตถ์ดรุณี
- ปัจจุบัน กำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **r_raiva_tik@hotmail.com**