ประสิทธิภาพของการเข้ารหัสแบบคอนโวลูชันโค้ดโดยมีการใช้อินเทอร์ลีฟ

PERFORMANCE OF CONVOLUTIONAL-ENCODED SIGNALS WITH INTERLEAVING

นายการันตร์ พันเลิศพาณิชย์ รหัส 50360531

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	ประสิทธิภาพของการเข้ารหัสแบบคอนโวลูชันโค้ดโดยมีการใช้อินเทอร์-ลีฟ
ผู้ดำเนินโครงงาน	นายการันตร์ พันเลิศพาณิชย์ รหัสนิสิต 50360531
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ

(ดร. ชัยรัตน์ พินทอง)

.....กรรมการ

(ดร. อัครพันธ์ วงค์กังแห)

ชื่อหัวข้อโครงงาน	ประสิทธิภาพของการเข้ารหัสแบบคอนโวลูชันโค้ดโดยมีการใช้อินเทอร์-ลีฟ
ผู้ดำเนินโครงงาน	นายการันตร์ พันเลิศพาณิชย์ รหัส 50360531
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

บทคัดย่อ

การสื่อสารในปัจจุบันมีอยู่หลายรูปแบบและแต่ละแบบนั้นมีการพัฒนาเพื่อให้การสื่อสารนั้นมีประสิทธิภาพมากขึ้น การสื่อสารในระบบคลื่นแม่เหล็กไฟฟ้าเป็นระบบที่นิยมใช้กันอย่างแพร่หลายเนื่องจากระบบนี้ใช้ตัวกลางเป็นอากาศในการส่งสัญญาณซึ่งสะดวกกว่าระบบที่ใช้สาย แต่สิ่งสำคัญที่จะขาดไม่ได้ก็คือ ปลายทางต้องรับสัญญาณได้ถูกต้องฉะนั้นจึงมีกระบวนการต่างๆ เพื่อเข้ามาช่วยป้องกันความผิดพลาดจากสัญญาณรบกวน

ระบบที่ทางผู้จัดทำได้ให้ความสนใจในการป้องกันข้อมูลผิดพลาดไว้ล่วงหน้าของระบบการสื่อสารแบบคลื่นแม่เหล็กไฟฟ้าคือ การเข้ารหัสแบบ Convolutional code และมีการใช้ Interleave ที่จะให้ระบบสามารถแก้ไขบิตผิดพลาดได้มากขึ้น เพื่อที่จะศึกษาประสิทธิภาพของระบบ Convolutional code ในงานวิจัยเล่มนี้ได้มีการจำลองระบบด้วยโปรแกรม MATLAB โดยการทดลองนั้นจะมีการเปรียบเทียบ ค่าความน่าจะเป็นในการตัดสินบิตผิดพลาด(Bit error rate) ทั้งในกรณีที่ใช้ Interleave และไม่ได้ใช้ Interleave ว่าทั้งสองมีประสิทธิภาพแตกต่างกันอย่างไร

Project title	Performance of Convolutional-Encoded Signal with Interleaving
Name	Mr. Karan Phunlertpanich ID.50360531
Project advisor	Assistant Professor Surachet Kanprachar, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2010

Abstract

In the present, there are several forms of communication and each form was developed in order to make it more effective. Communication by electromagnetic waves is widely used because it uses air as a medium of transmission which is more convenient than wired systems. But the important thing is to be dispensable destination to receive the signal correctly. Therefore, there are a few processes preventing the errors from noise.

In this project, the system of interest in which errors are prevented in advance is encoded by means of Convolutional code. This approach has been widely used in electromagnetic communication systems. In addition, the interleaving technique is included in order to reduce the Bit error rate. Hereby, the system simulation is implemented via MATLAB. Cases with and without the interleaving technique are simulated in such a way that the corresponding Bit error rate are compared.

กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความอนุเคราะห์จากบุคคลหลายท่าน ซึ่งไม่อาจนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่ผู้ศึกษาใคร่ขอกราบพระคุณคือ

ขอขอบคุณ ผศ.ดร.สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาโครงงาน ในการให้ความรู้ สละเวลาให้คำแนะเกี่ยวกับการเขียนโปรแกรมและปัญหาที่เกิดขึ้นขณะที่ดำเนินโครงงาน ทางผู้จัดทำขอบกราบขอบพระคุณอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ประสิทธิ์ประสาทวิชาความรู้มาช่วยเป็นพื้นฐานในการทำโครงงานนี้

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิด ที่เป็นผู้ให้กำลังและทำให้ผู้จัดทำมีวันนี้

ประโยชน์ที่ได้จากโครงงานนี้ทางผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

นายการันตร์ พันเลิศพาณิชย์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

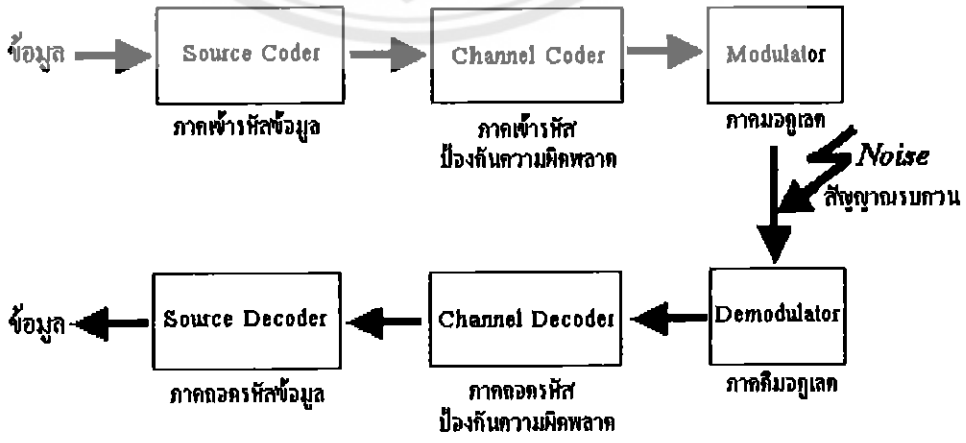
สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของโครงงาน

ในปัจจุบันนี้การสื่อสารและการส่งข้อมูลในระยะทางไกลๆมีความสำคัญมากและการสื่อสารมีอยู่หลายแบบและใช้กันอย่างแพร่หลาย เช่น ระบบส่งสัญญาณวิทยุระบบดิจิตอล โทรศัพท์มือถือ สัญญาณดาวเทียม Bluetooth เป็นต้น ซึ่งการส่งข้อมูลในรูปแบบต่างๆนั้นจำเป็นที่ต้องการความถูกต้องด้วยกันทั้งนั้น และได้มีการนำกระบวนการเช่น Block Codes Convolutional Codes เป็นต้น เพื่อที่จะทำให้ข้อมูลที่จะส่งไปนั้นปลายทางสามารถรับข้อมูลได้อย่างถูกต้อง

หลังจากที่ได้กล่าวมาในข้างต้น โครงงานนี้ได้มีการเสนอเรื่อง การเข้ารหัสแบบ Convolutional code และ การถอดรหัส ซึ่งเป็นวิธีการป้องกันการผิดพลาดล่วงหน้าและเป็นเทคนิครูปแบบหนึ่งที่ใช้แก้ไขการผิดพลาดของข้อมูลที่ได้รับจากสัญญาณรบกวนให้ถูกต้องที่มีประสิทธิภาพ สำหรับการเข้ารหัสข้อมูลในระบบสื่อสาร โดยสำหรับวิธีในการทำงานนั้น จะเป็นการนำข้อมูลที่เป็นแบบข้อมูลดิจิตอลที่จะทำการส่งข้อมูลผ่านระบบสื่อสารมาทำการเปลี่ยนแปลงรูปแบบของข้อมูล ให้อยู่ในอีกลักษณะที่เรียกว่า Codeword ซึ่งจะมีลักษณะพิเศษที่สามารถทำการแก้ไขหรือตรวจจับข้อมูลที่เกิดการผิดพลาดขึ้นได้ ระหว่างการส่งข้อมูลผ่านระบบสื่อสาร ให้กลับมาเป็นข้อมูลที่ถูกต้องได้ เมื่อข้อมูลนั้นถูกส่งมาถึงปลายทาง แสดงดังรูปที่ 1.1ซึ่งลักษณะของข้อมูลที่ได้หลังจากกระบวนการเข้ารหัสข้อมูลที่จะถูกส่งผ่านระบบสื่อสารนั้น จะมีขนาดของข้อมูลที่มากกว่าข้อมูลที่ไม่มีการเข้ารหัสข้อมูล



รูปที่ 1.1 แผนผังการเข้ารหัสข้อมูลของระบบสื่อสาร [1]

โครงงานนี้ยังได้นำเสนอการตรวจสอบและแก้ไขข้อมูลที่ผิดพลาดที่ปลายทางโดย Convolutional Codes และการทำ Modulation แบบ BPSK(Binary Phase Shift Keying) และมีการนำเอา Interleave มาช่วยในการแก้ไขข้อผิดพลาดที่เกิดขึ้นโดยมีการนำเอาโปรแกรม MATLAB เข้ามาสร้าง Code เพื่อที่จะจำลองการทำ Covolutional Code ให้เข้าใจมากขึ้น

1.2. วัตถุประสงค์โครงงาน

1. เพื่อศึกษาการทำ Convolutional Code ทั้งการเข้ารหัสและการถอดรหัส

2. เพื่อศึกษาผลของสัญญาณเมื่อผ่านช่องส่งสัญญาณไปแล้ว

3. เพื่อศึกษาความแตกต่างระหว่างการมี Interleave และไม่มี Interleave

1.3. ขอบเขตของโครงงาน

1. ศึกษาทฤษฎีการทำ Convolutional code และการทำ Interleave ในแบบต่างๆ

2. ศึกษาการเขียนโปรแกรม MATLAB

3. ศึกษาการทำ Convolutional code เมื่อนำมารวมกับการทำ Interleave

1.4. ขั้นตอนการดำเนินงานและแผนการดำเนินงานตลอดโครงการวิจัย

แผนการดำเนินโครงการ	ปี 2553							ปี 2554		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.รวบรวมข้อมูล	■									
2.ศึกษาการทำ Convolutional code ในส่วน Encode		■								
3.ศึกษาการทำ Convolutional code ในส่วนการทำ Decode			■							
4.ศึกษาการทำ Interleave			▒	▒						
5.เขียนโปรแกรมการทำ Convolutional code โดยใช้โปรแกรม Matlab				▒	▒	▒				
6.เขียนโปรแกรมการทำ Convolutional+Interleave โดยใช้โปรแกรม MATLAB						▒	▒	■		
7.ทดสอบโปรแกรม MATLAB									■	
8.สรุปผลและเขียนปริญญานิพนธ์										■

1.5. ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1. เข้าใจผลของความแตกต่างระหว่างกรณีมี Interleave และไม่มี Interleave

2. ทำให้เราได้พัฒนาการใช้โปรแกรม MATLAB

3. ทำให้เข้าใจในการทำ Convolutional code ทั้งการเข้ารหัสและถอดรหัสว่ามีการทำงานอย่างไร

1.6. งบประมาณ

1.ค่าหนังสือ	400	บาท
2.ค่าจัดทำเอกสาร	400	บาท
3.ค่าพิมพ์เอกสาร	200	บาท
รวมทั้งสิ้น(หนึ่งพันบาทถ้วน)	1000	บาท

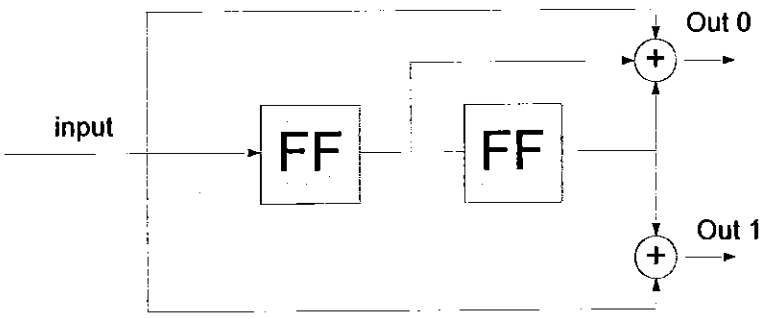
หมายเหตุ:ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่ใช้งาน

สำหรับการเข้ารหัสแบบ Convolutional Codes นั้น จะเป็นรูปแบบของการเข้ารหัสข้อมูลที่มีความแตกต่างจาก Block Codes โดยจะเป็นการเข้ารหัสที่มีการนำข้อมูลในอดีตจำนวนหนึ่งที่ถูกป้อนเข้ามา ภายในวงจรเข้ารหัสมาทำการประมวลผลร่วมกับข้อมูลที่ถูกป้อนเข้ามา ณ เวลานั้นๆ ในการคำนวณหาค่าCodeword ของข้อมูลชุดนั้นโดยสำหรับวิธีการเข้ารหัสข้อมูลทั้งสองรูปแบบนั้นจะมีลักษณะการทำงานที่แตกต่างกันคือ ในส่วนของ Block Code จะเป็นการนำข้อมูลที่จะทำการเข้ารหัสมาทำการแบ่งข้อมูลออกเป็นส่วนๆ หรือ Block ที่มีขนาดที่เท่าๆ กันแล้วจะมีการแทนขนาดของข้อมูลในแต่ละ Block ด้วยตัวแปร k ซึ่งในการทำงานของภาคเข้ารหัสนั้น จะเป็นการดึงข้อมูลที่จะทำการเข้ารหัสมาครั้งละ k บิต เพื่อนำมาทำการประมวลผลเพื่อหาค่าของCodeword ที่มีขนาดเท่ากับ n บิต ซึ่งจะถูกส่งในระบบสื่อสารเพื่อแสดงถึงข้อมูลของ Block นั้น โดยที่จะมีการนิยามตัวแปรที่มีชื่อว่า redundant bit (r) ซึ่งจะเป็นตัวแปรที่ใช้แสดงถึงจำนวนของข้อมูลที่เพิ่มขึ้นในการเข้ารหัสข้อมูลในแต่ละครั้ง ซึ่งจะสามารถคำนวณได้จาก

$$r = n - k \tag{2.1}$$

และในกรณีการเข้ารหัสข้อมูลแบบ Convolutional code เป็นการความสัมพันธ์ระหว่างข้อมูลอินพุตเรียงลำดับอย่างต่อเนื่อง โดยข้อมูลเข้ามาผ่านตัว shift register(flip-flop) และ Modulo-2 adder (exclusive or) การหาเอาท์พุทของภาคเข้ารหัสจะทำโดยนำข้อมูลที่อยู่ใน shift register บวกแบบ Modulo-2 adder ซึ่งจะมีลักษณะของวงจรแสดดังรูปที่ 1 จากวงจรเข้ารหัสจะมีอัตราการเข้ารหัส (Rate) เท่ากับ ½ และค่าConstraint Length (K) เท่ากับ 3 โดยจะใช้ Generator polynomial เพื่อแสดงตำแหน่งของ shift register



รูปที่ 2.1 การเข้ารหัสแบบ Convolutional code ที่ K=3 [2]

จากรูปที่ 2.1 จะเป็นตัวอย่างของวงจรเข้ารหัสข้อมูลแบบ Convolutional code โดยในการทำงานนั้น จะมีการดึงข้อมูลที่จะทำการเข้ารหัสมาครั้งละ 1 บิต (k =1) เข้ามาภายในวงจร ซึ่งจะทำให้ข้อมูลที่อยู่ในตำแหน่งต่างๆ ของ shift-register จากนั้นถูกเลื่อนไปอยู่ในตำแหน่งถัดไป ต่อจากนั้นจะมีการนำข้อมูลทั้งหมดที่เก็บไว้ใน shift-register มาทำการคำนวณเพื่อหาเอาท์พุทจำนวน 2 บิต (n = 2) ซึ่งอัตราการเข้า รหัสเท่ากับ 1/2 (Rate=k/n=1/2) ที่จะเป็นผลลัพธ์ส่งออกไปจากภาคเข้ารหัสข้อมูล ส่วนค่าจำนวนของ shift register และอินพุทข้อมูลที่นำมาบวกแบบ Modulo-2 เรียกว่าค่า Constraint Length (K) โดยจะใช้ Generator polynomial เพื่อแสดงตำแหน่งของใน shift register ที่จะนำมาหาค่าเอาท์พุทโดยการบวกแบบ Modulo-2 ในการแสดงลักษณะของวงจรเข้ารหัส

2.1. การวิเคราะห์การทำงานของวงจรเข้ารหัส [1]

จากวงจรเข้ารหัสแบบ Convolutional codes ที่มีการรับข้อมูลที่จะทำการเข้ารหัสเข้ามาภายในวงจรครั้งละ 1 บิต จากนั้นจึงทำการคำนวณหาค่าของ Codeword ที่จะถูกส่งออกไปเป็นผลลัพธ์ของภาคเข้ารหัสจำนวน 2 บิต โดยที่ในการคำนวณหาค่า Codeword ในแต่ละครั้งนั้นจะมีการนำข้อมูลที่อยู่ใน shift register จำนวน 3 บิต (K) มาใช้ในการคำนวณ ซึ่งในการวิเคราะห์การทำงานของวงจรเข้ารหัสนั้น จะมีการแทนการทำงานต่างๆ ของวงจรด้วยตัวแปรที่เรียกว่า Generator polynomial ซึ่งจะเป็นตัวแปรที่ใช้แสดงถึงลักษณะของการคำนวณหาผลลัพธ์ในการเข้ารหัสของ O/P แต่ละตัว โดยในกรณีของวงจรเข้ารหัสตัวอย่างในรูปที่ 1 นั้น จะสามารถแสดงการทำงานต่างๆของวงจรได้ด้วย Generator polynomial ต่อไปนี้

$$g_1 = [1 \quad 1 \quad 1] \tag{2.2}$$

$$g_2 = [1 \quad 0 \quad 1] \tag{2.3}$$

ซึ่งในการคำนวณหาค่าของข้อมูลที่ได้หลังจากการเข้ารหัสนั้น จะสามารถนำข้อมูลที่ได้จากค่าของ Generator polynomial มาทำการคำนวณหาค่าของผลลัพธ์ที่ได้จากการเข้ารหัส โดยการนำข้อมูลที่จะทำการเข้ารหัสมาทำการ Convolution กับค่าของ Generator polynomial ดังสมการ

$$\frac{o}{p_1} = [input] \times g_i \tag{2.4}$$

$$O/P1 = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \times [1 \quad 1 \quad 1] = 110100$$

$$O/P2 = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1] \times [1 \quad 0 \quad 1] = 100001$$

*เมื่อเรานำค่าที่ได้ออกมา จากสมการข้างต้นมาจับคู่สามารถสร้างตารางที่ 1 ได้ดังนี้

ตารางที่ 2.1 แสดงการเข้ารหัสมี่ค่าอินพุทเท่ากับ "101011"

INPUT DIGIT	OUTPUT CODEWORD
1	11
0	10
1	00
0	10
1	00
1	01

ดังนั้นค่าของ Codeword ที่ได้จากการเข้ารหัสข้อมูลนั้นจะเกิดจากการข้อมูลที่คำนวณได้จากO/P1 และตามด้วย O/P2 ซึ่งจะมีค่าเท่ากับ 11 10 00 10 00 01 ซึ่งจากการที่การคำนวณต่างๆในการหาผลลัพธ์ของวงจรเข้ารหัสนั้น สามารถที่จะคำนวณได้จากการหาค่า Convolution ดังนั้นจึงมีการเรียกรูปแบบในการเข้ารหัสข้อมูลในลักษณะนี้ว่าเป็นการเข้ารหัสแบบ Convolutional Codes และเมื่อนำมาเขียนรวมกันจะได้เป็น Generator polynomial ของวงจรเข้ารหัสจะมีลักษณะดังนี้

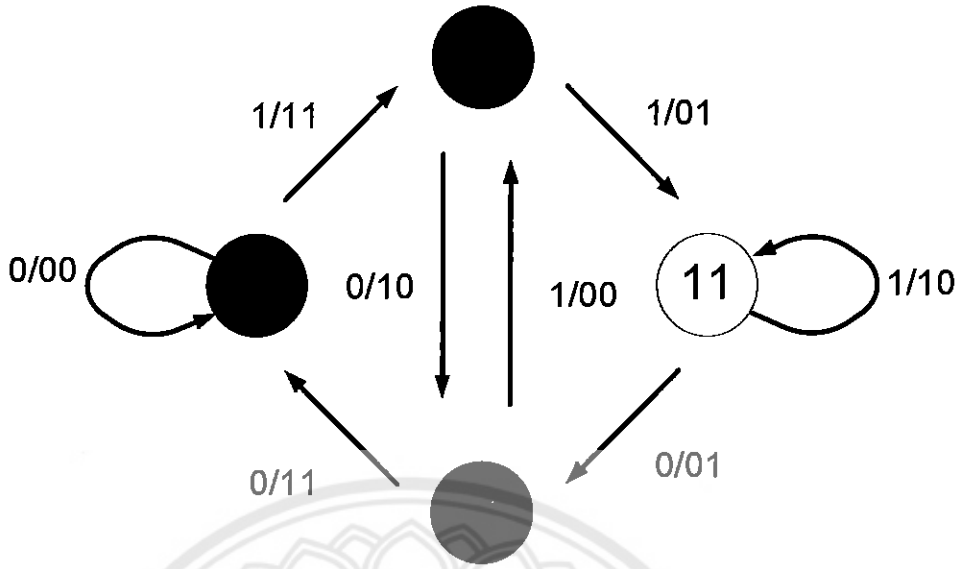
$$G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2.2. State Diagram[2]

State Diagram จะแสดงค่าของข้อมูลใน shift register และเอาท์พุทของตัวเข้ารหัส ซึ่งแสดงดังรูปที่ 2.2 ตัวเลขที่อยู่ในวงกลมแต่ละวงนั้น จะหมายถึงสภาวะต่างๆ ของข้อมูลที่ถูกเก็บไว้ใน Shift registerซึ่งในกรณีของวงจรเข้ารหัสที่ใช้เป็นตัวอย่างนั้น จำนวนสถานะ(State)ทั้งหมดเท่ากับ 4 สถานะ และสำหรับลูกศรที่ถูกแสดงไว้ในรูปนั้นจะแสดงถึงลักษณะของการเปลี่ยนแปลงการทำงานจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง ซึ่งจะขึ้นอยู่กับข้อมูลที่ป้อนเข้ามา ณ เวลานั้นๆ เช่นกรณีที่ข้อมูลที่เก็บไว้มีค่าเป็น 00 เมื่อมีข้อมูล 1 ป้อนเข้ามาจะมีผลทำให้ข้อมูลที่เก็บไว้ถูกเปลี่ยนไปเป็น 10 และจะมีการส่งค่า 11ออกไปจากวงจร (แทนด้วยสัญลักษณ์ 1 / 11 ดังรูปที่ 2.2) แต่ถ้าหากว่ามีการป้อนข้อมูล 0 เข้ามา วงจรก็จะยังคงมีสภาวะเป็น 00 เหมือนเดิมและจะมีการส่งค่า 00 ออกไปจากวงจร(แทนด้วยสัญลักษณ์ 0 / 00 ดังรูปที่ 2.2)

ตารางที่ 2.2 แสดงข้อมูลที่นำไปสร้าง Stage Diagram

INPUT DIGIT	INTIAL STATE	FINAL STATE	OUTPUT CODEWORD
0	11	01	01
1	11	11	10

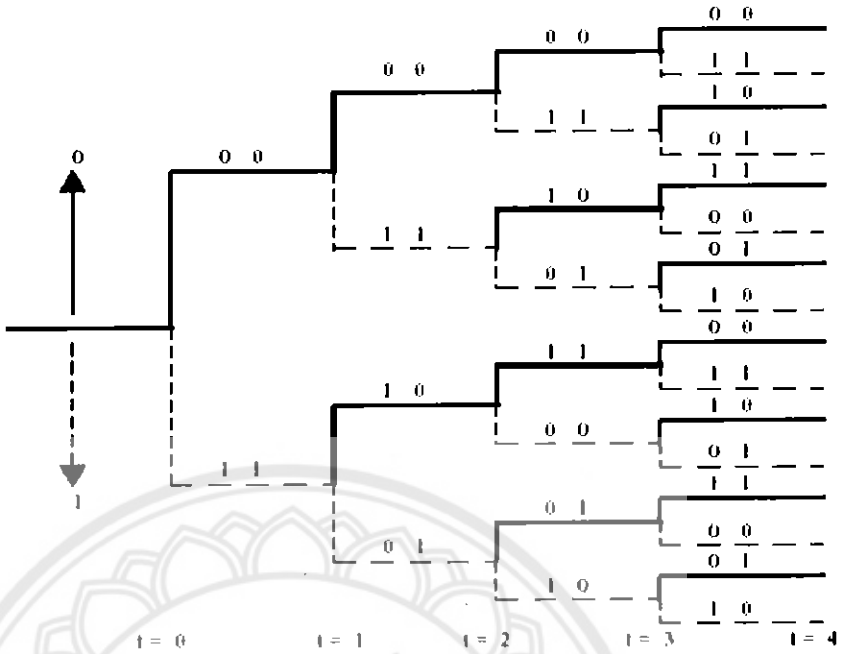


รูปที่ 2.2 State Diagram [3]

จากตารางและรูป Stage Diagram มีความสัมพันธ์กันดังนี้ เช่น ในกรณีสีน้ำเงินก็จะมี 2 กรณี อินพุตเข้า 0 และ 1 เอาต์พุตที่เราได้นั้นก็จะมี 2 กรณี ใน stage diagram เราสามารถเขียนแทนทั้ง 2 กรณีเป็น X/YY หมายความว่า X คือ อินพุตที่เข้ามาในระบบ YY คือ เอาต์พุตที่ได้หลังจากการผ่านการเข้ารหัส

2.3. Tree Diagram[3]

สำหรับ Tree Diagram นั้น จะเป็นการพิจารณาถึงลักษณะของการทำงานของวงจรเข้ารหัสข้อมูลโดยที่จะมีการพิจารณาถึงค่าของผลลัพธ์ที่ได้หลังจากการป้อนข้อมูลต่างๆเข้าไปในวงจรเข้ารหัสเป็นหลัก ซึ่งจะมีลักษณะของ Tree Diagram ดังรูปที่ 2.3



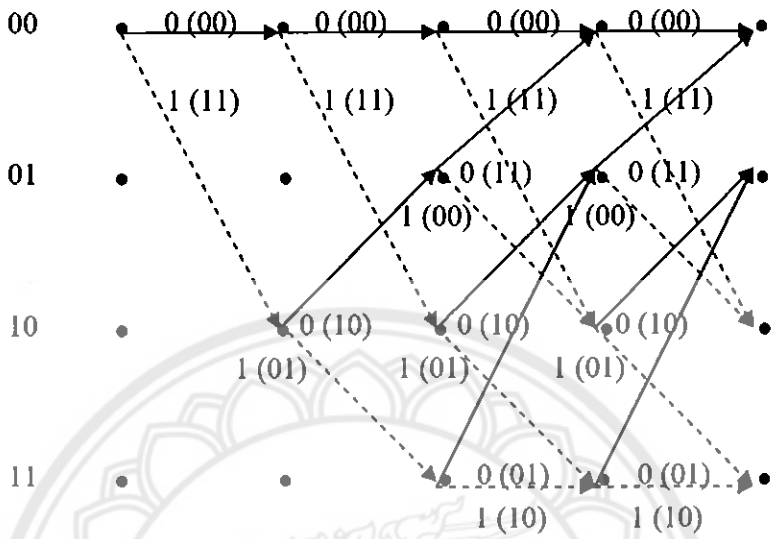
รูปที่ 2.3 Tree Diagram [4]

สำหรับการแสดงการทำงานของวงจรเข้ารหัสแบบ Convolution Codes โดยใช้ Tree Diagram นั้น จะเป็นการพิจารณาการทำงานโดยการคำนึงถึงข้อมูลที่ป้อนเข้ามาและที่จะถูกส่งออกไปจากภาคเข้ารหัสเป็นหลัก ซึ่งในการพิจารณานั้น จะเริ่มต้น ณ ตำแหน่งรากของ Tree Diagram ซึ่งในกรณีของรูปที่ 2.3 นั้น จะอยู่ในตำแหน่งซ้ายมือสุดของรูป ซึ่งจะมีการนำข้อมูลที่ถูกป้อนเข้ามาภายในวงจรเข้ารหัสเป็นตัวกำหนดทิศทางการเดินทางของข้อมูลใน Tree Diagram โดยในกรณีของ Tree Diagram ตัวอย่างนั้น จะกำหนดให้มีการเลื่อนตำแหน่งไปทางข้างบนเมื่อมีการรับข้อมูล 0 เข้ามา และจะเลื่อนตำแหน่งลงล่างเมื่อมีการรับข้อมูล 1 เข้ามา ซึ่งหลังจากที่มีการเลื่อนตำแหน่งที่ใช้พิจารณาแล้ว จะมีการพิจารณาถึงข้อมูลที่จะถูกส่งออกไปจากภาคเข้ารหัส ณ เวลานั้นๆ จากข้อมูลที่อยู่เหนือเส้นทางในตำแหน่งที่มีการพิจารณา

2.4. Trellis Diagram [4]

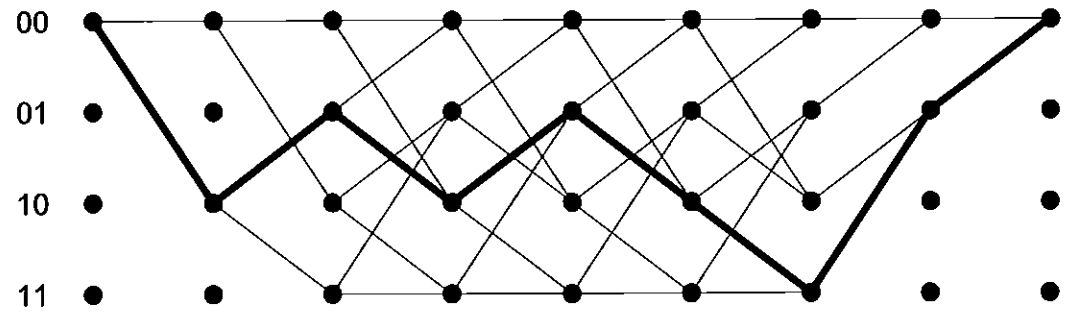
การแสดงการทำงานของวงจรเข้ารหัสโดยใช้ Trellis Diagram นั้น จะเป็นการนำการแสดงการทำงานของวงจรเข้ารหัสโดยใช้ State Diagram มาทำการเปลี่ยนแปลงรูปแบบให้อยู่ในอีกลักษณะหนึ่ง ที่แสดงถึงการเปลี่ยนแปลงของข้อมูลต่างๆภายในวงจรเข้ารหัส ข้อมูลที่ป้อนเข้ามา

และ Codeword ที่จะถูกส่งออกไป ณ เวลาต่างๆ โดยที่จะมีลักษณะของ Trellis Diagram ดังรูปต่อไปนี้



รูปที่ 2.4 Trellis Diagram ในกรณีที่ Rate เท่ากับ 1/2 [5]

จากรูปที่ 2.4 จะเป็นการแสดงการทำงานของวงจรเข้ารหัสข้อมูลแบบ Convolutional Code ในรูปที่ 2.4 ที่มีการนำข้อมูลในอดีตจำนวน 2 บิต มาทำการประมวลผลร่วมกับข้อมูล ณ เวลานั้น (จำนวน state ทั้งหมดใน trellis diagram จะมีค่าเท่ากับ $2^2 = 4$ state) และจะมีข้อมูลป้อนเข้ามาภายในวงจรครั้งละ 1 บิต ซึ่งในกรณีนี้ จะมีข้อมูลที่จะเข้ารหัสทั้งหมด 6 บิต ซึ่งเส้นทางต่างๆที่อยู่ใน Trellis Diagram นั้น จะแสดงถึงลักษณะการเปลี่ยนแปลง สภาวะของวงจรและตัวเลข x/yที่อยู่เหนือทางเดินในแต่ละเส้นทางนั้น จะแสดงถึงข้อมูลที่ป้อนเข้ามา และ Codeword ที่จะถูกส่งออกไปเมื่อมีการป้อนข้อมูลนั้นเข้ามา ซึ่งเมื่อพิจารณาลักษณะของ Trellis Diagram แล้ว จะพบว่ารูปแบบของTrellis Diagram ในแต่ละ State การทำงานนั้น จะมีลักษณะที่คล้ายกันแต่จะมีความแตกต่างกันเฉพาะส่วนหัวและท้าย ซึ่งเป็นผลมาจากข้อมูลที่เก็บอยู่ในวงจรนั้นจะมีค่าที่เริ่มต้นจากสภาวะที่มีข้อมูลเป็น 0 ทั้งหมด และจะจบลงที่สภาวะข้อมูลเป็น 0 เช่นกัน ดังนั้นเส้นทางอื่นๆ ที่ไม่ผ่านจุดที่มีข้อมูลเป็น 0 ทั้งหมด ณ จุดเริ่มต้นและจุดสุดท้ายนั้นจะไม่ถูกนำมาพิจารณา ดังนั้นขนาดของความยาวใน Trellis Diagram นั้นจึงขึ้นอยู่กับข้อมูลที่ถูกนำมาเข้ารหัส ตัวอย่างในการใช้งาน Trellis Diagram เช่น กรณีที่มีการป้อนข้อมูล 101011เข้ามาภายในวงจร จะสามารถใช้ Trellis Diagram ในการหาลักษณะของ Codeword ได้ดังรูปที่ 2.5



รูปที่ 2.5 Trellis Diagram ที่มีการเข้ารหัสโดย codeword 101011 [5]

ดังนั้นข้อมูลที่ได้หลังจากการเข้ารหัสจะมีค่าเท่ากับ 11 10 00 10 00 01 01 11

2.5. ระยะฟรี (free distance) [5]

พารามิเตอร์หนึ่งที่มีผลต่อประสิทธิผลของรหัสคอนโวลูชันคือ ระยะฟรี (free distance) ซึ่งเป็นระยะห่างที่น้อยที่สุดระหว่างคำรหัสสองคำในรหัสคอนโวลูชันหนึ่งๆ ระยะฟรีนี้จะมีผลคล้ายกับระยะใกล้ที่สุด (minimum distance) ของรหัสแบบบล็อกโดยถ้าระยะนี้มีค่ามากจะทำให้รหัสมีโอกาสที่จะแก้ไขความผิดพลาดได้มากขึ้นเนื่องจากเมื่อคำรหัสมีโอกาสที่จะแก้ไขความผิดพลาดได้มากขึ้นเนื่องจากเมื่อคำรหัสแต่ละคำมีความแตกต่างกันมากก็จะทำให้ตัวถอดรหัสตัดสินใจเลือกคำรหัสที่ถูกต้องได้ง่ายขึ้นในรูปสมการอาจเขียนระยะฟรีได้ดังนี้

$$d_{free} = \min\left\{d\left(v',v''\right) : u' \neq u''\right\} \tag{2.5}$$

เมื่อ v' และ v'' คือคำรหัสของข้อมูล u' และ u'' ตามลำดับ โดย $u' \neq u''$ และ $d\left(v',v''\right)$ คือ ระยะแฮมมิ่ง (Hamming distance) ระหว่างคำรหัสทั้งสอง

จากการที่รหัสคอนโวลูชันเป็นรหัสเชิงเส้น (liner code) เนื่องจากระบบการเข้ารหัสเป็นระบบเชิงเส้นดังที่ได้อธิบายไว้แล้วดังนั้นระยะฟรีจึงสามารถหาได้จากน้ำหนักของคำรหัสที่ไม่เป็นศูนย์ (nonzero codeword) ใดๆ โดยไม่ขึ้นกับความยาวเพราะน้ำหนักของคำรหัสไบนารีคือจำนวนบิตที่มีค่าเป็นหนึ่งซึ่งจำนวนบิตที่เป็นหนึ่งนี้ก็เท่ากับระยะห่างจากคำรหัสศูนย์นั่นเอง

$$d_{free} = \min\left\{w(v) : v \neq 0\right\} \tag{2.6}$$

เนื่องจากคำรหัสคอนโวลูชันมีความยาวได้ไม่จำกัดแต่ในการใช้งานจริงที่ข้อมูลมีค่าสิ้นสุดนั้นจะมีการป้อนบิตศูนย์เข้าไปเพื่อให้ข้อมูลในหน่วยความจำถูกเข้ารหัสออกมาเป็นบิตหางและปรับให้หน่วยความจำมีค่าเป็นศูนย์หมดดังนั้นสามารถพิจารณาได้ว่าคำรหัสเริ่มจากสถานะศูนย์และสิ้นสุดที่สถานะศูนย์เสมอ ซึ่งสามารถใช้แผนภาพเทรลลิสช่วยในการหาคำรหัสเหล่านี้ได้ซึ่งเป็นวิธีที่เห็นภาพชัดเจน นอกจากนั้นยังสามารถใช้หลักการทับซ้อน (superposition principle) ของระบบเชิงเส้นซึ่งเป็นวิธีที่เร็วกว่าแต่อาจเข้าใจยากกว่าเล็กน้อย รวมทั้งยังมีวิธีที่ใช้กราฟการไหลของสัญญาณ (signal flow graph) ร่วมกับสูตรการหาค่าอัตราขยายของ Mason ซึ่งจะให้ข้อมูลครบถ้วนมากกว่าระยะฟรีโดยจะสามารถหาแจกแจงน้ำหนัก (weight distribution) ของรหัสคอนโวลูชันได้ทำให้ทราบได้ว่ามีรหัสกี่คำในแต่ละค่าน้ำหนัก

ตัวอย่างค่า d_{free} ที่สัมพันธ์กับค่า K โดยมีการทดลองโดยได้ค่า Generator sequence อยู่ในลักษณะเลขฐาน 8 ซึ่งค่าที่อยู่ในตาตารางนั้นได้มีการทดลองว่าสามารถแก้ไขบิตผิดพลาดได้จริง ดังแสดงตัวอย่างทั้งหมด 3 Rate มี Rate 1/2 , Rate 1/3 , Rate 1/4 ในตาราง 3 4 และ 5 ตามลำดับ แสดงดังนี้

ตารางที่ 2.3 Rate 1/2 ค่าระยะฟรีสูงสุด [7]

Constraint Length K	Generators in octal		d_{free}	Upper bound On d_{free}
3	5	7	5	5
4	15	17	6	6
5	23	35	7	8
6	53	75	8	8
7	133	171	10	10
8	247	371	10	11
9	561	753	12	12
10	1167	1545	12	13
11	2335	3661	14	14
12	4335	5723	15	15
13	10533	17661	16	16
14	21675	27123	16	17

ตารางที่ 2.4 Rate 1/3 ค่าระยะฟรีสูงสุด [7]

Constraint Length K	Generators in octal			d_{free}	Upper bound On d_{free}
3	5	7	7	8	8
4	13	15	17	10	10
5	25	33	37	12	12
6	47	53	75	13	13
7	133	145	175	15	15
8	225	331	367	16	16
9	557	663	711	18	18
10	1117	1365	1633	20	20
11	2353	2671	3175	22	22
12	4767	5723	6265	24	24
13	10533	10675	17661	24	24
14	21675	35661	37133	26	26

ตารางที่ 2.5 Rate 1/4 ค่าระยะฟรีสูงสุด [7]

Constraint Length K	Generators in octal				d_{free}	Upper bound On d_{free}
3	5	7	7	7	10	10
4	13	15	15	17	13	15
5	25	27	33	37	16	16
6	53	67	71	75	18	18
7	135	135	147	163	20	20
8	235	275	313	357	22	22
9	463	535	733	745	24	24
10	1117	1365	1633	1653	27	27
11	2387	2353	2671	3175	29	29
12	4767	5723	6265	7455	32	32
13	11145	12477	15537	16727	33	33

การหาระยะฟรีโดยพิจารณาจากแผนภาพเทรลลิส

หลักการพิจารณาคำรหัสหรือเส้นทางในแผนภาพเทรลลิสในการหาค่าระยะฟรีมีดังนี้

1. พิจารณาเฉพาะเส้นทางที่เริ่มจากสถานะศูนย์ที่เวลาเริ่มต้นเพราะเส้นทางที่เริ่มจากสถานะศูนย์ที่เวลาอื่นจะเป็นคำรหัสที่เริ่มช้ากว่าเท่านั้น

2. พิจารณาเฉพาะเส้นทางที่จบที่สถานะศูนย์เพราะเป็นเส้นทางที่บิตข้อมูลมีการสิ้นสุดและแทนคำรหัสมีน้ำหนักจำกัด

3. ไม่จำเป็นต้องพิจารณาคำรหัสที่เส้นทางของมันมีการแวะที่สถานะศูนย์แล้วแยกออกมาอีกเพราะคำรหัสแบบนี้จะไม่มีโอกาสเป็นคำรหัสที่มีน้ำหนักน้อยที่สุดเนื่องจากจะมีคำรหัสที่จบที่สถานะศูนย์ซึ่งมีน้ำหนักน้อยกว่าเสมอ

2.6. การถอดรหัสแบบ Viterbi Decoder [6]

การถอดรหัสแบบ Viterbi จะมีลักษณะการทำงานเป็นแบบ Maximum-likelihood decoding algorithm ซึ่งผลลัพธ์ที่ได้จากการทำงานนั้น จะได้เส้นทางเพียงหนึ่งเส้นทางจากเส้นทางทั้งหมดในTrellis Diagram ที่มีลักษณะที่เหมือนกับข้อมูลที่รับได้มากที่สุด

รูปแบบสำหรับการถอดรหัสแบบ Viterbi Decoder

สำหรับการถอดรหัสแบบ Viterbi Decoding นั้น จะมีรูปแบบสำหรับการถอดรหัสที่ใช้งานอยู่ 2 ลักษณะด้วยกัน ได้แก่ แบบ Hard-Decision และ Soft-Decision

1. **Hard Decision**

   สำหรับการทำงานของวงจรถอดรหัสที่ใช้กระบวนการตัดสินใจแบบ Hard Decision นั้น จะเป็นการพิจารณาข้อมูลที่รับเข้ามา โดยการพิจารณาว่าข้อมูลที่รับเข้ามาในแต่บิตนั้น มีค่าของข้อมูลเป็น 0 หรือ 1 เท่านั้น

2. **Soft Decision**

   การถอดรหัสที่มีการใช้กระบวนการตัดสินใจแบบ Soft Decision จะเป็นการพิจารณาถึงข้อมูลที่รับเข้ามาได้โดยการทำการตัดสินใจระดับของข้อมูลที่รับเข้ามาได้โดยการแบ่งระดับของสัญญาณที่ใช้ในการ คำนวณหาค่า metric ที่มากกว่า 2 ระดับ ซึ่งผลลัพธ์ที่ได้นั้น จะได้ข้อมูลรายละเอียดของข้อมูลที่ส่งมาที่มากกว่ากรณีของ Hard-Decision ซึ่งข้อมูลที่ได้จากการตัดสินใจ (soft-output) นั้น จะถูกนำมาใช้ในการคำนวณหาค่า metrics เพื่อเปรียบเทียบข้อมูลที่รับเข้ามา ณ เวลานั้นๆ กับข้อมูลที่อยู่ในเส้นทางต่างๆ ณ เวลานั้น ซึ่งจะมีรูปแบบที่ใช้ในการคำนวณที่แตกต่างกันไป

Viterbi Algorithm [7]

โดยในการทำงานต่างๆนั้นจะต้องมีการคำนวณหาความแตกต่างระหว่างข้อมูลที่รับเข้ามาและค่าที่อยู่ในเส้นทางต่างๆ เพื่อใช้ในกระบวนการตัดสินใจ โดยกระบวนการที่ใช้ในการทำการหาเส้นทางที่ดีที่สุดนั้น จะใช้วิธีการทำงานที่มีชื่อว่า Viterbi Algorithm ซึ่งจะเป็นกระบวนการที่ใช้ในการค้นหาเส้นทางที่อยู่ใน Trellis Diagram ที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับได้มากที่สุด เพื่อที่จะนำข้อมูลในเส้นทางนั้นมาคำนวณหาค่าของข้อมูลที่ถูกส่งมา โดยที่ในกระบวนการค้นหาเส้นทางที่เหมาะสมที่สุดโดยใช้ Viterbi Algorithm นั้น จะมีขั้นตอนในการทำงานดังต่อไปนี้

1. พิจารณาแบ่งข้อมูลที่รับเข้ามาออกเป็นข้อมูลย่อยๆจำนวน m ช่วง ซึ่งแต่ละช่วงนั้นมีขนาดของข้อมูลเท่ากับ n บิต
2. ทำการวาด Trellis diagram ที่มีจำนวน state ในการทำงานเท่ากับ m state โดยจะมีการพิจารณาเฉพาะเส้นทางที่มีความเป็นไปได้ว่าจะถูกส่งมาเท่านั้น โดยสำหรับที่ state ของ Trellis diagramตั้งแต่ L-1 ขึ้นไปนั้น(L คือค่า Constraint Length) ให้วาดเฉพาะเส้นทางที่จะพุ่งเข้าหาสภาวะของวงจรที่มีข้อมูลเป็น 0 ทั้งหมด
3. กำหนดค่าตัวแปร l = 1 และทำการกำหนดค่าเริ่มต้นของตัวแปร metric ในสภาวะเริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมด ให้มีค่าของ metric เท่ากับ 0
4. ทำการคำนวณหาค่าความแตกต่างของข้อมูล (distance) ระหว่างข้อมูลที่รับได้ชุดที่ l กับข้อมูลในเส้นทางในการเปลี่ยนแปลงสถานะใน Trellis diagram จาก state ที่ l ไปเป็น l+1
5. นำค่าที่คำนวณได้นั้นไปบวกกับค่า metric สะสมของstate l เพื่อคำนวณหาค่าของ metric สะสมใน state ที่ l+1 เพื่อใช้ในการตัดสินใจเลือกเส้นทางที่เหมาะสมที่สุด ในการเปลี่ยนแปลงข้อมูลไปยัง state นั้นๆ โดยในแต่ละ stateนั้น จะมีจำนวนเส้นทางทั้งหมดจำนวน 2^k เส้นทางที่จะพุ่งเข้าstate เดียวกัน
6. พิจารณา ณ ตำแหน่งใน state ที่ l+1 ในแต่ละ stateนั้น ทำการเลือกเส้นทางที่มีค่า metric สะสมที่มีค่าน้อยที่สุดที่พุ่งเข้าหาในแต่ละ state โดยทีเส้นทางที่ถูกเลือกนั้น จะถูกเรียกว่า "Survivor"ซึ่งจะเป็นเส้นทางที่ถูกเก็บไว้ทำการคำนวณใน state ต่อไป และสำหรับเส้นทางอื่นๆที่ไม่ได้ถูกเลือกนั้น จะถูกเรียกว่า "Forgetting" โดยจะถูกลบทิ้งออกไปจากระบบการตัดสินใจ
7. ถ้าหากว่า l นั้นมีค่าเท่ากับ m แล้วให้ทำงานในขั้นตอนต่อไปได้ แต่ถ้ายังมีค่าน้อยกว่า จะมีต้องมีการเพิ่มค่า l ขึ้นอีก 1 จากนั้นจึงกลับไปทำงานที่ขั้นตอนที่ 4 ใหม่
8. เริ่มต้นพิจารณา ณ state ที่ m+1 ที่มีสภาวะของข้อมูลสถานะเป็น 0 ทั้งหมด ทำการเลือกเส้นทางที่เป็น "Survival" ซึ่งเป็นเส้นทางที่ถูกเลือกที่เหลืออยู่ย้อนกลับไปจนกระทั่งถึงสภาวะเริ่มต้นของการทำงานที่มีสภาวะในการทำงานเป็น 0 ทั้งหมด ซึ่งเส้นทางที่ได้นั้น จะเป็นเส้นทางที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับเข้ามามากที่สุด ซึ่งจะถูกนำไปใช้ในการ

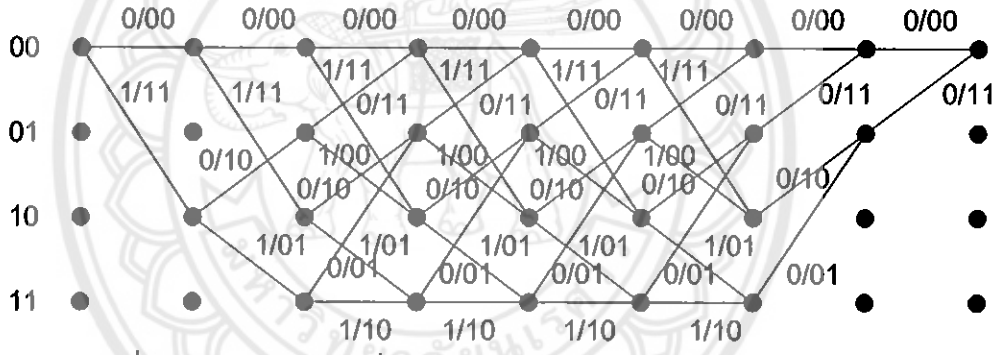
คำนวณหาข้อมูลข่าวสารที่ถูกส่งมา โดยข้อมูลข่าวสารที่จะถูกส่งออกไปจากภาคถอดรหัสนั้น จะเป็นการส่งข้อมูลทั้งหมดที่อยู่ในเส้นทางส่งออกไปยกเว้นยกเว้นข้อมูล 0 จำนวน $k_0(L-1)$ บิต ที่อยู่ท้ายสุดนั้น จะถูกตัดทิ้งไปตัว

ตัวอย่างการถอดรหัสแบบ Hard decision

ข้อมูลที่เราต้องการส่งไปคือ 1 0 1 0 1 1 เมื่อผ่านการเข้ารหัสของConvolutional code เราจะได้โค้ดออกมาดังนี้ 11 10 00 10 00 01 01 11

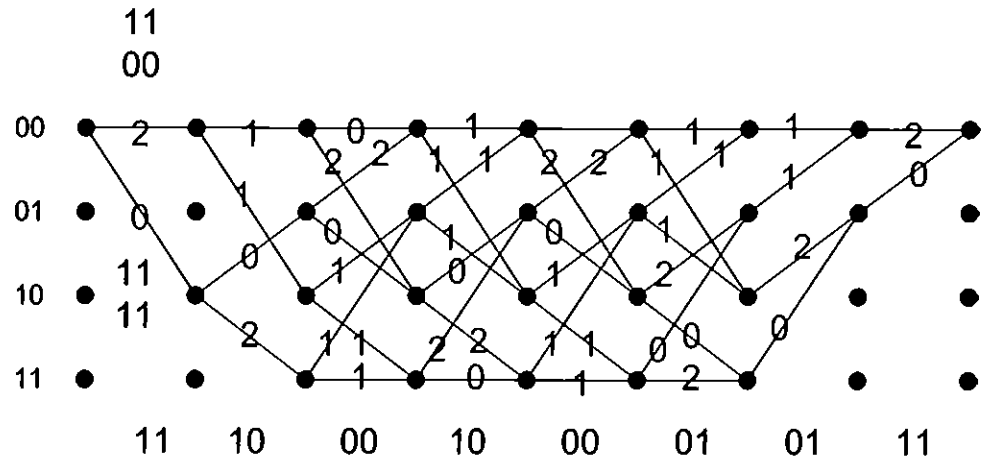
ขั้นตอนการถอดรหัสมีดังนี้

1. แบ่ง Codeword ออกเป็นชุดละ 2 บิต คือ 11 10 00 10 00 01 01 11และใช้แผนภาพเทรริสในการการเลือกเส้นทางในการถอดรหัสตามรูปที่ 2.6



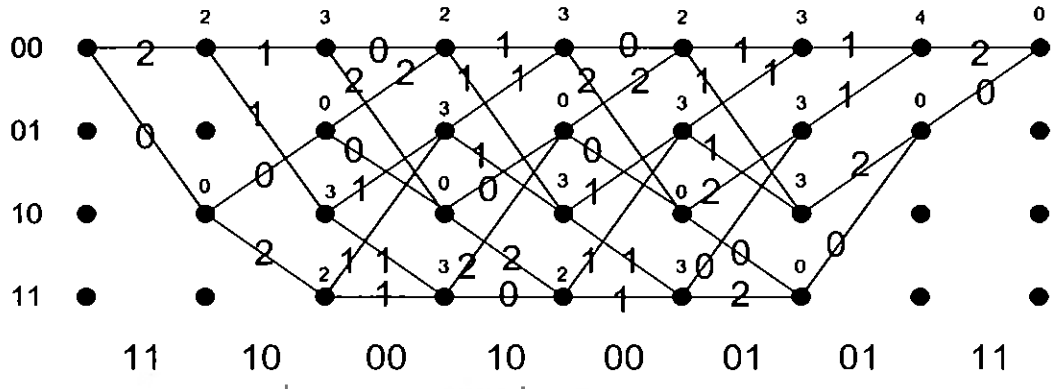
รูปที่ 2.6 แสดงการนำค่าที่ได้จาก State diagram มาสร้างแผนภาพเทรลลิส

2. นำบิตมาเปรียบเทียบทีละ 2 บิต ตามรูปด้านล่างตัวเลขสีแดงแทนข้อมูลที่นำมาถอดรหัส ตัวเลขสีเขียวแสดงความต่างของแต่ละบิต ทำไปเรื่อยๆจนครบทุกเส้นทาง



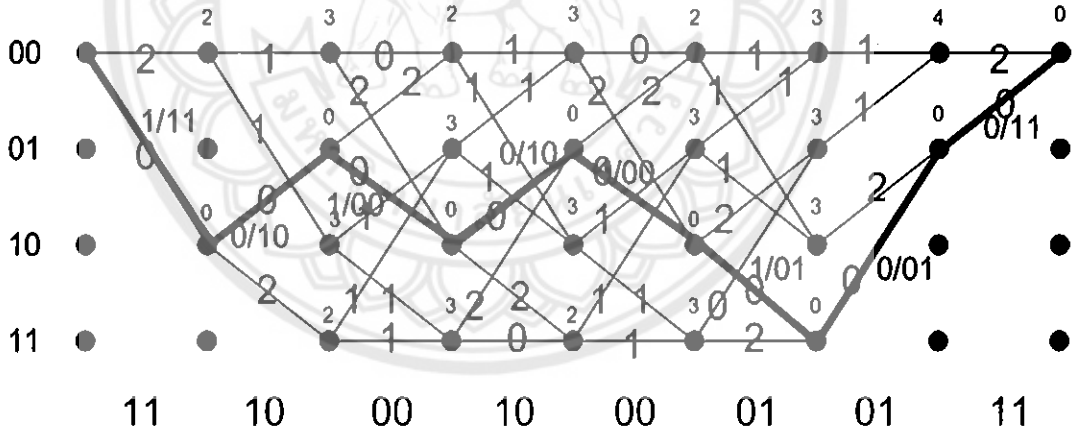
รูปที่ 2.7 แสดงการนำค่าของข้อมูลที่ละ 2 บิตมาเปรียบเทียบกัน

3. นำผลรวมความต่างของแต่ละเส้นทางมารวมกันที่โหนด



รูปที่ 2.8 แสดงการนำค่าที่ได้จากแต่ละเส้นทางมารวมกัน

4. หาเส้นทางที่มีผลรวมน้อยที่สุด และไล่ดูตามเส้นทางที่เป็น x/yy จะพบว่าข้อมูลที่เราไล่ตามเส้นทางดูตามค่า x จะได้ข้อมูลที่ถอดรหัสออกมา เป็น 1 0 1 0 1 1 0 0 ในกรณีนี้ 2 บิตสุดท้ายไม่คิด จึงได้ข้อมูลออกมาเป็น "1 0 1 0 1 1"

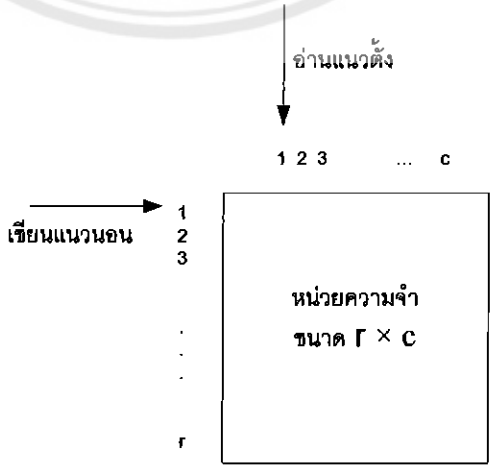


รูปที่ 2.9 แสดงการนำค่าของแต่ละโหนดมารวมกันเพื่อหาเส้นทางที่ผลรวมน้อยที่สุด

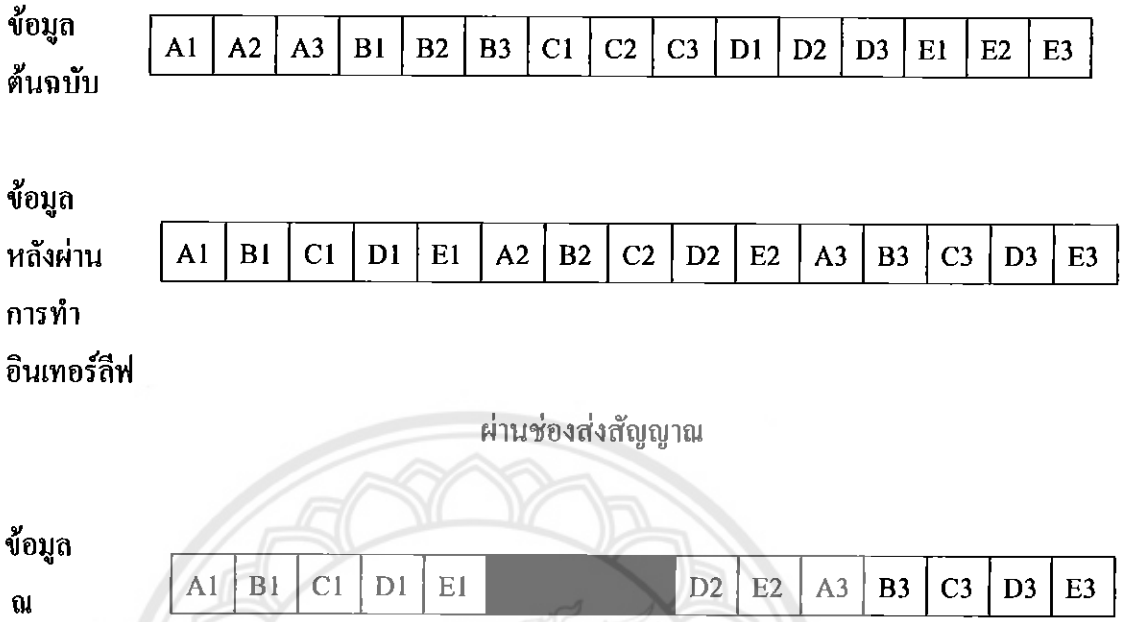
2.7. Interleaving [8]

อินเทอร์ลีฟวิง เป็นวิธีการที่สำคัญอีกวิธีหนึ่งที่มีการนำมาใช้งานในระบบสื่อสารที่มีความผิดพลาดแบบเบิรสต์ เนื่องจากความผิดพลาดแบบเบิรสต์เป็นรูปแบบของการผิดพลาดที่เกิดขึ้นบ่อยครั้งในการส่งสัญญาณไร้สาย เช่น ในระบบโทรศัพท์เครื่องที่ซึ่งเกิดจากการที่สัญญาณถูกบดบังด้วยวัตถุที่อยู่รอบข่ายไปชั่วขณะ ทำให้บิตข้อมูลที่รับได้มีการผิดพลาดอย่างต่อเนื่อง ดังนั้นการทำอินเทอร์ลีฟจึงมีประโยชน์อย่างมากในทางปฏิบัติ

หลักการของการทำอินเทอร์ลีฟก็คือจะแบ่งบิตข้อมูลออกเป็นบล็อกๆ ขนาด $r \times c$ บิต โดย r คือจำนวนแถว และ c คือจำนวนคอลัมน์จากนั้นก็เป็นการสลับตำแหน่งของแต่ละบิตภายในบล็อกเดียวกันก่อนการส่งออก โดยอาศัยวิธีการง่ายๆ คือนำข้อมูลที่จะส่งมาเขียนลงในหน่วยความจำที่ละแถวตามลำดับของข้อมูลจนครบหนึ่งบล็อก จากนั้นทำการอ่านข้อมูลเหล่านี้ในแนวตั้งเพื่อส่งออกที่ละคอลัมน์จนหมด ดังที่ส่งอยู่ในรูปที่ 2.10 และเมื่อบิตข้อมูลเหล่านี้ถึงที่ภาครับแล้วก็จะทำการสลับตำแหน่งของบิตให้กลับเป็นปกติโดยอาศัยกระบวนการที่กลับกันกับที่ภาคส่ง ดูตัวอย่างประโยชน์ที่ได้จากการทำอินเทอร์ลีฟในรูปที่ 2.11 ข้อมูลที่จะถูกส่งแบ่งออกเป็นบล็อกขนาด 5×3 บิต หลังจากที่ได้ทำการอินเทอร์ลีฟ ลำดับการส่งบิตข้อมูลจะแตกต่างไปจากข้อมูลของผู้ใช้เดิม สังเกตว่าถึงแม้ว่าบิตข้อมูลเหล่านี้จะได้รับผลกระทบจากความผิดพลาดแบบเบิรสต์เนื่องจากความไม่เป็นอุดมคติของช่องสัญญาณที่ส่งผ่าน เมื่อถึงที่ภาครับซึ่งจะมีการทำดีอินเทอร์ลีฟเพื่อให้บิตข้อมูลมีลำดับที่ถูกต้องเหมือนเดิมนั้น ความผิดพลาดแบบเบิรสต์ก็จะถูกกระจายลงในบิตข้อมูลในรูปแบบที่ต่างไปจากเดิม ซึ่งโดยทั่วไปความผิดพลาดจะมีลักษณะการกระจายอย่างสุ่มมากขึ้น ซึ่งประโยชน์ต่อการออกแบบรหัสแก้ไขความผิดพลาดที่มีประสิทธิภาพ กล่าวคือ ระบบไม่ต้องอาศัยวิธีการเข้ารหัสที่มีขีดความสามารถในการแก้ไขความผิดพลาดแบบเบิรสต์ที่มีความยาวมาก



รูปที่ 2.10 ตัวอย่างวิธีการทำอินเทอร์ลีฟ [6]



รูปที่ 2.11 ตัวอย่างแสดงประโยชน์และการใช้งานของอินเทอร์ลีฟ **[6]**

ในบทที่ 2 นี้เป็นการแสดงทฤษฎีต่างที่ใช้ในงานวิจัยนี้และได้มาแสดง**ตัวอย่างการทำงาน เพื่อที่จะแสดงให้ผู้สนใจในการศึกษางานวิจัยฉบับนี้ทำความเข้าใจง่ายขึ้น** ซึ่ง**ข้อมูลในบทนี้จะเป็นพื้นฐานในการสร้างโปรแกรมที่ใช้ในการทดลองในบทที่ 3 บทที่ 4 และบทที่ 5 ฉะนั้นทฤษฎีและวิธีการต่างที่ใช้ได้แสดงไว้ในบทที่ 2**

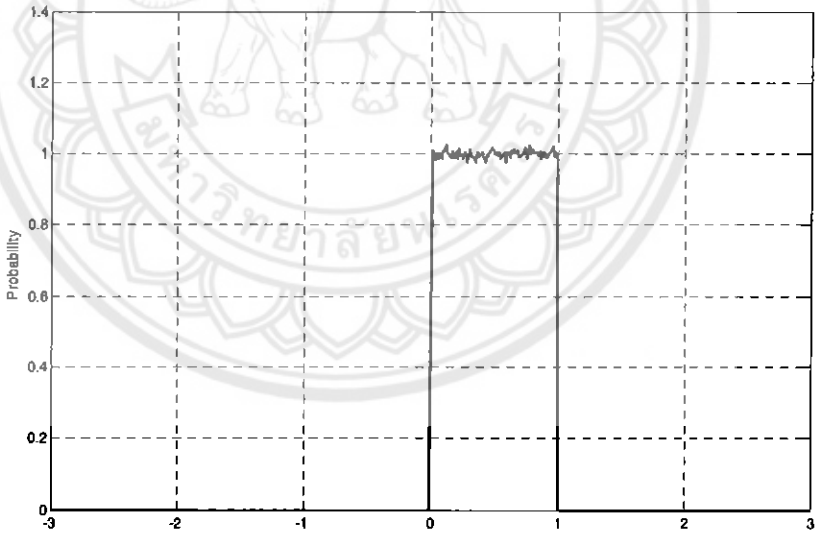
บทที่ 3
การออกแบบโครงงานและวิธีดำเนินงาน

ในบทนี้กล่าวถึงโครงสร้างการทำงานและการออกแบบการเขียนโปรแกรมจำลองการเข้ารหัสแบบ Convolutional Code และการถอดรหัสแบบ Viterbi ทั้งในกรณีที่มีการทำ Interleave และไม่มี Interleave โดยแบ่งออกเป็นขั้นตอนดังนี้

3.1. ขั้นตอนการออกแบบโปรแกรม

3.1.1. การสร้างสัญญาณ

ขั้นตอนที่ 1 เราจะทำการสุ่มบิตข้อมูลขึ้นมาโดยอาศัยหลักของความน่าจะเป็นโดยความน่าจะเป็นที่จะเป็นบิต 1 และ 0 เท่าๆกัน เพื่อที่จะนำสัญญาณนี้มาเข้ารหัส Convolutional Code และทำการส่งเข้าไปในระบบการสื่อสาร



รูปที่ 3.1 แสดงความน่าจะเป็นที่ใช้ในการสุ่มบิตข้อมูล

จากรูปที่ 3.1 จะพบว่าข้อมูลจะมีค่าอยู่ระหว่าง 0 ไปจนถึง 1 ซึ่งรูปดังกล่าวจะอยู่ในรูปแบบความน่าจะเป็นแบบยูนิฟอร์ม ค่าที่ได้ออกมาจากกราฟนั้นจะพบว่ายังไม่ได้อยู่ในรูปบิต 0 และ 1

จะต้องอาศัยคำสั่งในโปรแกรม MATLAB มาปรับให้อยู่ในรูปบิต 0 และ 1 ตามความน่าจะเป็นที่เกิดขึ้น

3.1.2. การเข้ารหัส

ขั้นตอนที่ 2 เราจะนำสัญญาณที่เราสร้างมาในขั้นตอนแรกนั้นมาทำการเข้ารหัสแบบ Convolutional Code โดยมีอัตราการเข้ารหัสที่แตกต่างกันออกไป ได้แก่ อัตราการเข้ารหัส 1/2, อัตราการเข้ารหัส 1/3,อัตราการเข้ารหัส 1/4, ,อัตราการเข้ารหัส2/3,อัตราการเข้ารหัส 2/4 โดยวิธีการเข้ารหัสนั้นได้แสดงวิธีการไว้ในบทที่ 2 จะพบว่าอัตราการเข้ารหัสทั้งหมดจะทำให้ข้อมูลมีจำนวนบิตที่มากขึ้นซึ่งในการเข้ารหัสนี้จะทำให้พลังงานต่อบิตลดลงซึ่งมีผลต่อการสร้าง สัญญาณรบกวนในขั้นตอนต่อไป

3.1.3. การทำ Interleave

ขั้นตอนที่ 3 เป็นการนำสัญญาณที่ผ่านการเข้ารหัสจากขั้นตอนที่ 2 มาทำการแบ่งบิตข้อมูลออกเป็นบล็อก ขนาด $r \times c$ บิต โดย r คือจำนวนแถว และ c คือจำนวนคอลัมน์ จะพบว่าข้อมูลที่เข้ามานั้นยังไม่สามารถที่นำมาทำการแบ่งเป็นบล็อกเท่ากันได้จะมีเศษเหลืออยู่ที่ปลายข้อมูลเกิดขึ้น ในงานวิจัยเรื่องนี้เราจะตัดและเก็บไว้ก่อน และนำข้อมูลที่เป็นบล็อกมาเขียนลงในหน่วยความจำที่ละแถวตามลำดับของข้อมูลจนครบหนึ่งบล็อก จากนั้นทำการอ่านข้อมูลเหล่านี้ในแนวตั้งเพื่อส่งออกที่ละคอลัมน์จนหมด แล้วจึงนำข้อมูลที่เราเก็บไว้มาต่อในช่วงสุดท้ายของบิตข้อมูล วิธีในการสร้างข้อมูลที่ผ่านการทำ Interleave ตัวอย่างแสดงไว้ด้านล่างนี้

<u>ตัวอย่าง</u>

ข้อมูลทั้งหมด $n = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]$

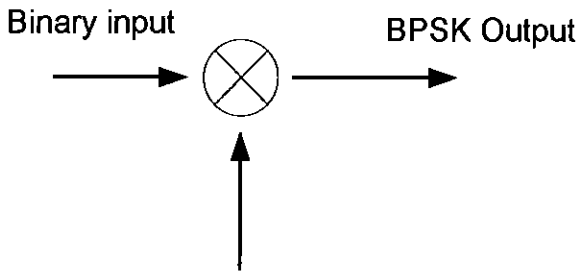
แบ่งข้อมูลออกเป็น 2 ชุด $n_1 = [1 \quad 1 \quad 0 \quad 0 \quad 1]$, $n_2 = [1 \quad 1 \quad 0 \quad 0 \quad 0]$

นำข้อมูลมาเรียงเป็นเมทริกซ์ 2×5 จะได้เป็น $m = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$

ต่อมาดึงข้อมูลออกมาทีละหลักมาเรียงเป็นแถวจะได้ข้อมูลตัวใหม่ออกมาเป็น n_{new}

$$n_{new} = [1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]$$

3.1.4. การ Modulate แบบ BPSK



รูปที่ 3.2 แสดงการ Modulate แบบ BPSK

ขั้นตอนที่ 4 เป็นการทำงานดังรูป 3.2 การนำสัญญาณที่ผ่านการทำ Interleave จากขั้นตอนที่ 2 มาทำการ Modulate จะพบว่าสัญญาณที่ได้ออกมานั้นอยู่ในรูปกราฟ sine wave การทำงานในส่วนของขั้นตอนนี้คือ เมื่อสัญญาณอินพุตที่เป็นสัญญาณดิจิตอลมีการเปลี่ยนสถานะ (จาก "0" เป็น "1" หรือจาก "1" เป็น "0") ทำให้เอาท์พุตเปลี่ยนเฟสไป 180 องศา ซึ่งทำให้แทนลักษณะการ modulate สัญญาณดิจิตอลด้วยเทคนิค BPSK ดังนี้

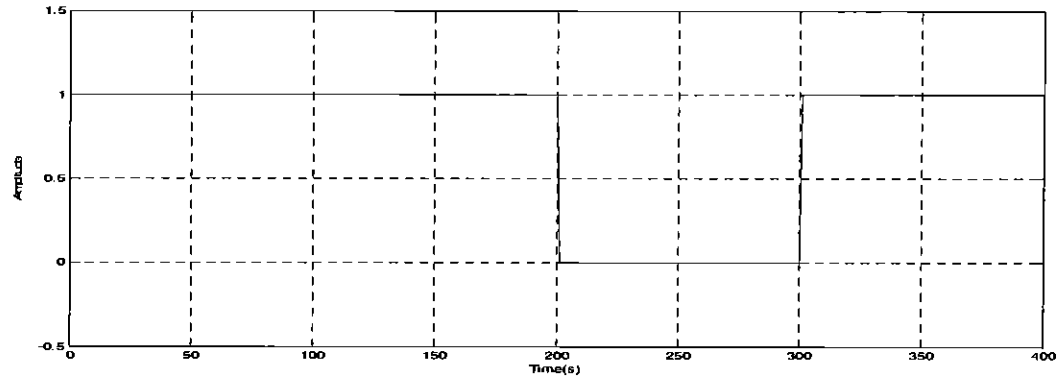
ค่าบิตข้อมูลที่เป็น "0" ให้มุมเลื่อนของสัญญาณพาห์เท่ากับ 0

ค่าบิตข้อมูลที่เป็น "1" ให้มุมเลื่อนของสัญญาณพาห์เท่ากับ π

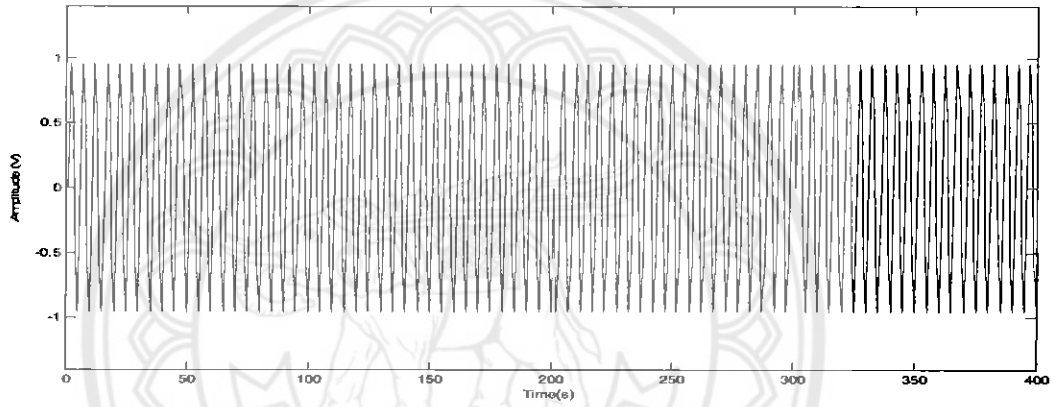
จากที่กล่าวมาสามารถเขียนสมการของการ modulate สัญญาณดิจิตอลด้วยเทคนิค BPSK อีกรูปแบบหนึ่งได้ดังนี้

$$s(t) = \begin{cases} A\cos(2\pi f_c t + \pi); binary 1 \\ A\cos(2\pi f_c t); binary 0 \end{cases} \quad (3.1)$$

ในรูปที่ 3.3 (a) นี้เป็นการแสดงสัญญาณดิจิตอลที่มีสัญญาณข้อมูลตัวอย่างเป็น 1 1 0 1 เมื่อกำหนดการ modulate เป็นแบบ BPSK จะได้สัญญาณเอาต์พุตดังในรูป 3.3 (b)



(ก)



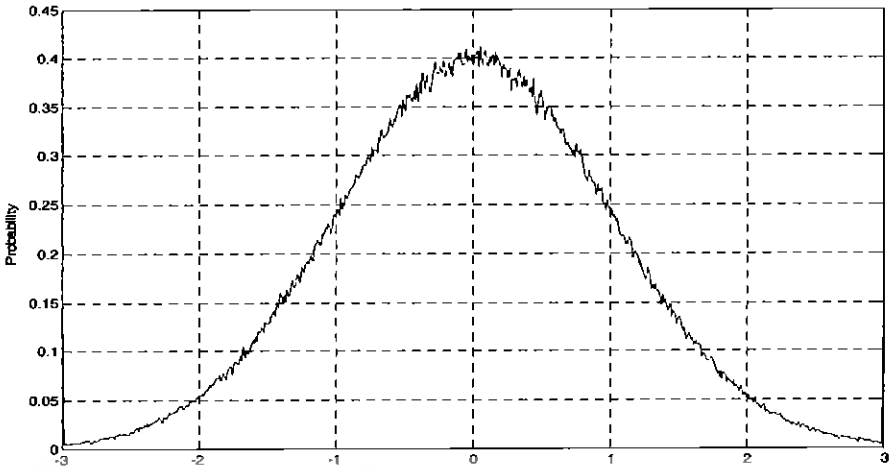
(ข)

รูปที่ 3.3 (ก) เป็นการแสดงสัญญาณดิจิตอลที่มีสัญญาณข้อมูลเป็น 1 1 0 1 เมื่อทำการmodulate แบบ BPSK จะได้สัญญาณเอาต์พุตดังในรูป (ข)

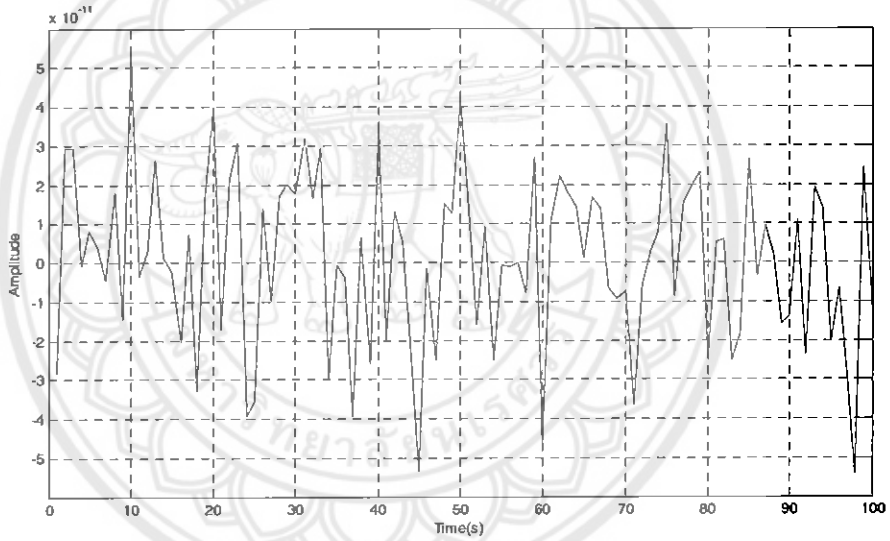
จะพบว่าเมื่อเรานำสัญญาณที่ได้จากการที่ผ่านการเข้ารหัสมาทำการคูณด้วย $\cos(\omega_c t)$ สัญญาณที่ได้นั้นจะอยู่ในรูปสัญญาณดังรูป 3.3 (ก)

3.1.5. การสร้างสัญญาณรบกวน

ขั้นตอนที่ 5 จะเป็นการสร้างสัญญาณรบกวนขึ้นมาโดยวิธีของเกาส์เซียน โดยใช้คำสั่ง randn ในโปรแกรม MATLAB ซึ่งสัญญาณรบกวนที่ได้ออกมานั้นจะเป็นเชิงเส้น สัญญาณที่ขาเข้าจะเป็นสัญญาณบวกกับสัญญาณรบกวนจะเป็นดังสมการที่ 3.2 สัญญาณรบกวนจะความน่าจะเป็นในการเกิดดังรูปที่ 3.4 และสัญญาณรบกวนที่ได้ออกมานั้นจะเป็นดังรูปที่ 3.5



รูปที่ 3.4 ความน่าจะเป็นที่ใช้ในการสุ่มสัญญาณรบกวน



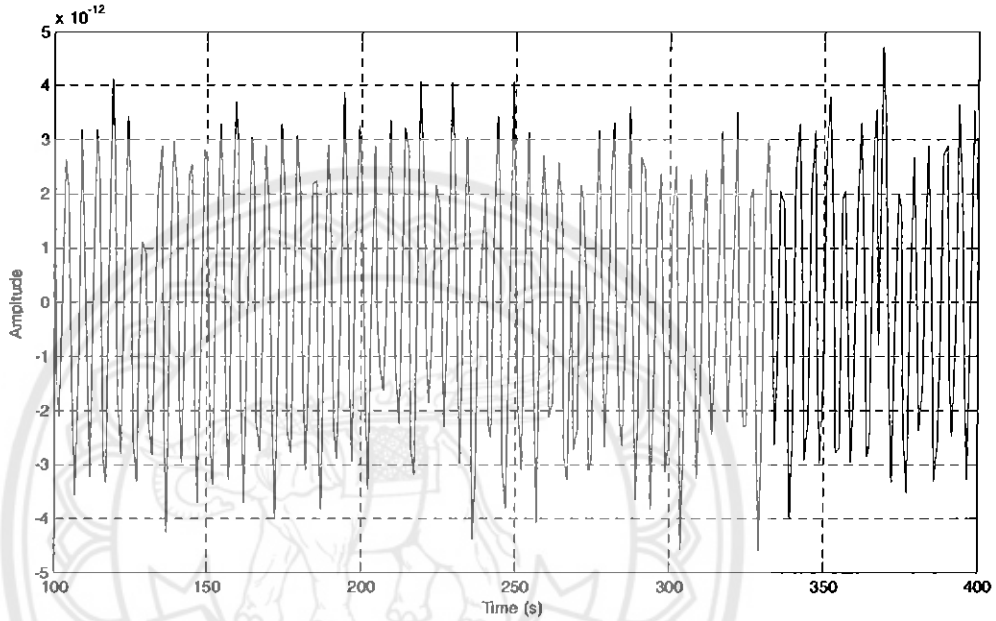
รูปที่ 3.5 แสดงการสุ่มสัญญาณรบกวนในช่วงเวลา 100 วินาที

3.1.6. การรวมสัญญาณ

ขั้นตอนที่ 6 จะเป็นการรวมสัญญาณที่เกิดจากการ modulate เข้ากับสัญญาณรบกวนที่สร้างขึ้นมา ดังสมการที่ 3.2 จะพบว่าการสุ่มบิตข้อมูลแบบเกาส์เซียนสามารถนำมาบวกกันได้ เนื่องจากเป็นเชิงเส้นในขั้นตอนนี้จะเป็นการแสดงให้เห็นว่าเมื่อสัญญาณถูกส่งผ่านระบบสื่อสารนั้นมีข้อผิดพลาดเกิดขึ้น การรวมสัญญาณแสดงดังสมการที่ 3.2

$$r(t) = s(t) + n(t) \tag{3.2}$$

เมื่อ $r(t)$ เป็นสัญญาณที่รวมกับสัญญาณรบกวนแล้ว, $n(t)$ เป็นสัญญาณรบกวนที่เป็นตัวแปรสุ่มแบบเกาส์เซียนที่มีค่าเฉลี่ยเท่ากับศูนย์และ $s(t)$ เป็นข้อมูลที่อยู่ในรูปแบบ sine wave



รูปที่ 3.6 แสดงการรวมสัญญาณที่ช่วงเวลา 100 ถึง 400 วินาที SNR = 30 dB

จากรูปที่ 3.6 จะเห็นได้ว่าเป็นการนำสัญญาณที่ได้ออกมานั้นเป็นสัญญาณที่รวมกับสัญญาณรบกวนแล้ว จะเห็นได้ว่าสัญญาณไม่คงที่ Amplitude ไม่เท่ากันในแต่ละวินาที นั้นเป็นผลมาจากสัญญาณรบกวนซึ่งเป็นผลให้เราอาจตัดสินบิตผิดพลาดได้ในขั้นตอนต่อๆไป

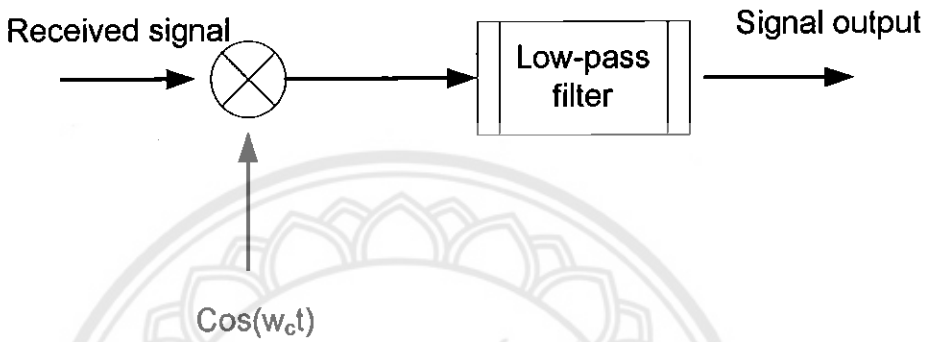
3.1.7. การ Demodulate แบบ BPSK

ขั้นตอนที่ 7 จะเป็นการเปลี่ยนรหัสสัญญาณจากขั้นตอนที่ 5 จากที่สัญญาณที่อยู่ในรูปสัญญาณ sine wave ให้เปลี่ยนมาอยู่ในรูปแบบบิต 1 0 เพื่อสามารถนำเข้าไปถอดรหัสแบบ viterbi ในกรณีของ Hard Decision ได้ ขั้นตอนการทำดังนี้ การ modulate แบบ BPSK สามารถนำสัญญาณคลื่นพาห์คูณกับสัญญาณที่รับเข้ามาได้ซึ่งสามารถเขียนสมการดังนี้

$$r(t) = \left[A * m(t)\cos(2\pi f_c t)\right] * \cos(2\pi f_c t) \quad (3.3)$$

$$= 0.5A * m(t)\cos(4\pi f_c t) + 0.5Am(t) \quad (3.4)$$

เมื่อผ่านวงจรฟิลเตอร์ความถี่ต่ำจะได้สัญญาณ $0.5A * m(t)$ ซึ่งเป็นสัญญาณไบนารีดังรูปที่ 3.7



รูปที่ 3.7 แสดง Block diagram สำหรับการ Demodulate แบบ BPSK

3.1.8. การทำ DeInterleave

ในขั้นตอนนี้จะเป็นการกลับข้อมูลให้กลับมาอยู่ในลักษณะที่เหมือนกับที่ออกมาจากการเข้ารหัสซึ่งเป็นข้อมูลที่แท้จริง ทำโดยนำข้อมูลมาตัดเป็นบล็อกแล้วนำมาเรียงเป็นหลักแล้วเราจะได้ข้อมูลที่เป็นแถวเรียงตามแถวไปเป็นข้อมูลที่แท้จริง

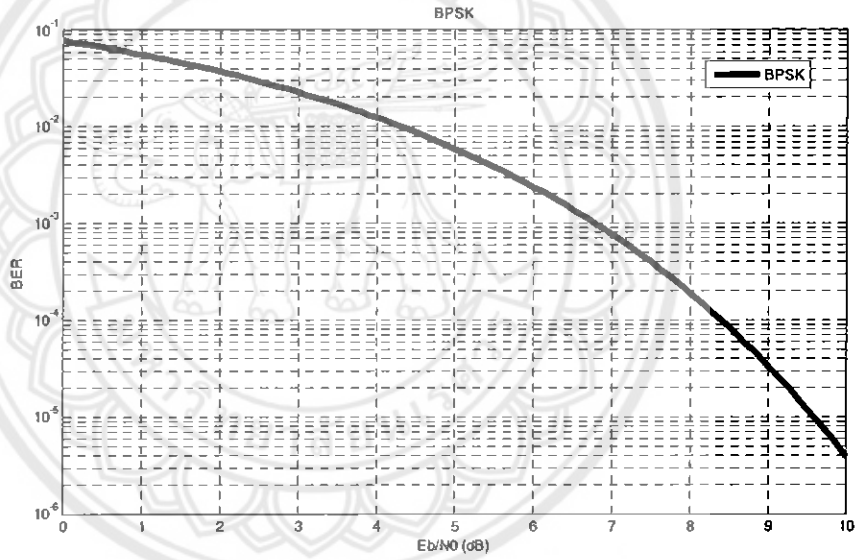
3.1.9. การถอดรหัสแบบ Viterbi

ในขั้นตอนนี้เป็นการนำเอาวิธีของ Viterbi Decoder ในกรณีของ Hard Decision มาทำการแก้ไขบิตผิดพลาด ซึ่งวิธีในการทำงานได้แสดงไว้ในบทที่ 2

3.1.10. การสร้างกราฟ BER

ในขั้นตอนนี้จะเป็นการนำเอาข้อมูลที่เราทำการเก็บข้อมูล SNR ที่เรากำหนด ในที่นี้จะกำหนดข้อมูลตั้งแต่ 0 ไปจนถึง 10 dB ในการวิเคราะห์ BER จากความน่าจะเป็นตาสมการที่ 3.5 ค่า P_e จะเป็นค่าความน่าจะเป็นที่ของค่า BER ซึ่งเมื่อนำค่าที่ได้ออกมานั้นมาสร้างกราฟ BER จะสามารถบอกความสามารถในการแก้ไขบิตผิดพลาด เราจะได้กราฟออกมาจากการเก็บค่าทีละตัว ในรูปที่ 3.8 จะเป็นการแสดงในกรณีที่ไม่มีการเข้ารหัสและถอดรหัสแต่จะมีการ modulate BPSK จึงได้กราฟดังรูป

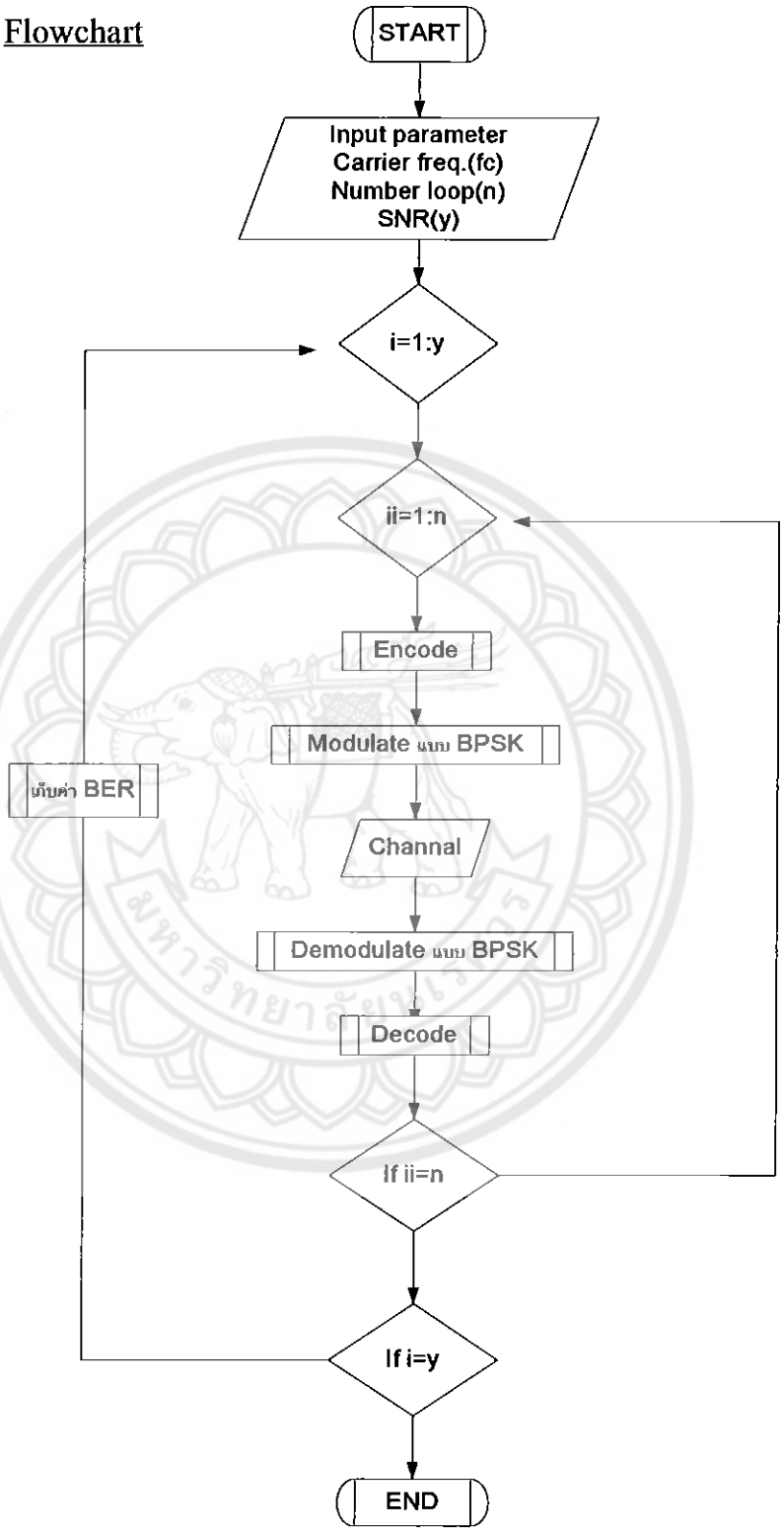
$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{3.5}$$



รูปที่ 3.8 ตัวอย่างกราฟ Bit error rate ในกรณี ที่มีการ modulate แบบ BPSK

การทำงานของระบบทั้งหมดจะมีการทำงานแบ่งออกเป็นส่วนๆ ซึ่งการทำงานในแต่ละขั้นตอนเป็นการจำลองขึ้นจากสถานการณ์จริง โดย Chanel ที่เป็นตัวกลางที่ใช้ในการส่งสัญญาณนั้น มีการสุ่มสัญญาณรบกวนแบบ Gaussian เพื่อทำการทดสอบว่าระบบ Convolutional Code มีความสามารถในการแก้ไขบิตผิดพลาดได้มากน้อยแค่ไหนซึ่งระบบจะแสดงค่าออกมาเป็นในรูปรแบบ BER การทำงานของโปรแกรมได้แสดงไว้ดังรูป 3.9

Flowchart



รูปที่ 3.9 แสดงการทำงานของโปรแกรมทั้งหมดที่ใช้ในการทดลอง

Flowchart เป็นการสรุปว่าโปรแกรมนี้มีการการทำงานอย่างไรบ้าง เริ่มต้นโปรแกรมจะทำการรับค่า SNR จำนวนรอบของ loop และจำนวนบิตข้อมูล Carrier frequency ค่าทั้งหมดนี้จะเป็นผลกับโปรแกรมทั้งหมด โปรแกรมนี้จะมีแบ่งเป็น loop ใหญ่ทั้งหมด 2 loop การทำงานของ loopที่ 1 จะเป็นการทำงานครอบคลุมอีก loop หนึ่งอยู่โดยจำนวนในการวนจะขึ้นอยู่กับจำนวนของ SNR เมื่อครบตามจำนวน SNR แล้วก็จะโปรแกรมจะเสร็จสิ้นการทำงาน ในส่วนของ loop ที่สองนั้นเป็น loop ที่วนตามจำนวนที่เรากำหนดจำนวนรอบตั้งแต่เริ่มต้นโปรแกรม การทำงานจะมีการสุ่มบิตข้อมูลตามจำนวนที่กำหนดทุกๆรอบ ต่อมานำข้อมูลที่ได้มาทำการเข้ารหัส(encode) ในขั้นตอนนี้จะจะเพิ่มขึ้นตาม rate ที่กำหนด นำข้อมูลที่ได้มาผ่านModulate แบบ BPSK ข้อมูลจะอยู่ในรูป sine wave จากนั้นนำสัญญาณมาผ่าน Channel ในส่วนนี้จะมีการเพิ่มสัญญาณรบกวนเข้าไปดังสมการที่ 3.2 เหมือนกับเป็นการจำลองช่องส่งสัญญาณขึ้นมา เมื่อได้สัญญาณออกมาแล้วนั้นจะต้องมาผ่านการ Demodulate เพื่อให้ข้อมูลกลับมาอยู่ในรูปของบิตข้อมูลอีกครั้งและนำข้อมูลตัวนั้นมาทำการถอดรหัส(decode) เมื่อได้ข้อมูลออกมาเราจะนำไปลบกับข้อมูลที่เราสุ่มมาตั้งแต่แรกเพื่อหาค่า BER ออกมาเป็นการเสร็จสิ้นโปรแกรม

3.2. การออกแบบการทดลอง

สำหรับในงานวิจัยเป็นการแสดงถึงประสิทธิภาพในการแก้ไขบิตผิดพลาดในกรณีที่มีการใช้ การเข้ารหัสแบบ Convolutional Code และการถอดรหัสแบบ Viterbi ทั้งในกรณีที่มีการนำเอาหลักการ Interleave เข้ามาช่วยในการแก้ไขบิตผิดพลาด ทั้งนี้ในงานวิจัยยังสามารถแบ่งการทดลองออกเป็นกรณีดังนี้

1. Convolutional Code 1/2 (อัตราการเข้ารหัสของสัญญาณเท่ากับ 1/2)

2. Convolutional Code 1/3 (อัตราการเข้ารหัสของสัญญาณเท่ากับ 1/3)

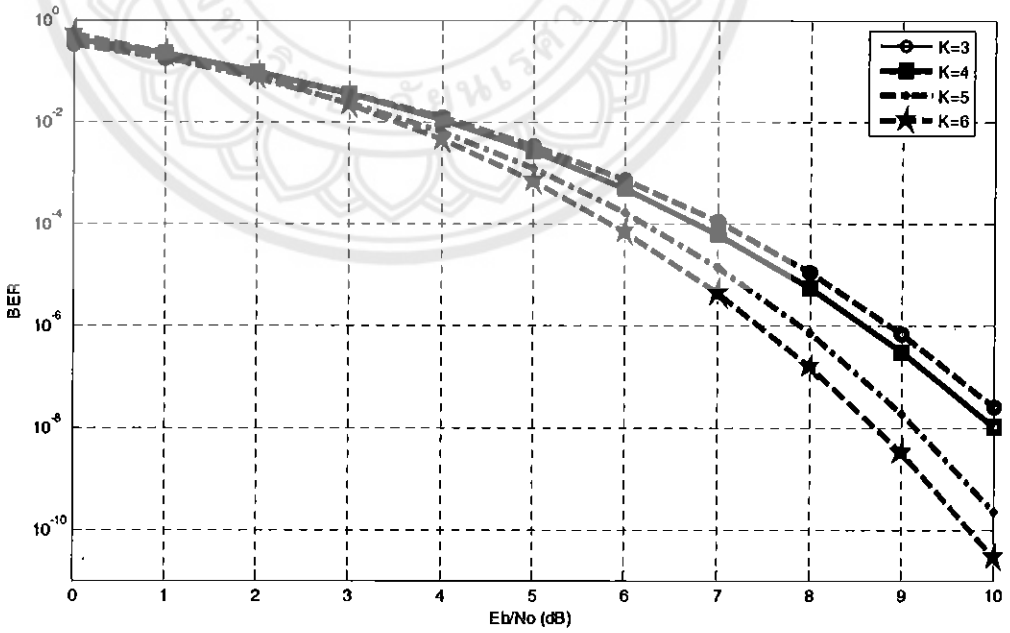
3. Convolutional Code 1/4 (อัตราการเข้ารหัสของสัญญาณเท่ากับ 1/4)

ในกรณีข้างต้นจะพบว่ามีการแบ่งออกเป็นทั้งหมด 3 กรณี ซึ่งในงานวิจัยยังมีการใส่การทำ Interleave ไปในทุกกรณีเพื่อที่จะเปรียบเทียบประสิทธิภาพ ผลที่ได้ออกมานั้นเราจะนำมาสร้างกราฟ Bit error rate สำหรับกราฟที่ได้นำมาวิเคราะห์ว่าในแต่ละกรณีเมื่อมีการนำ Interleave เข้ามาใช้ จะส่งผลต่อประสิทธิภาพของการแก้ไขบิตผิดพลาดว่าเป็นผลอย่างไร

บทที่ 4
ผลการดำเนินโครงงาน

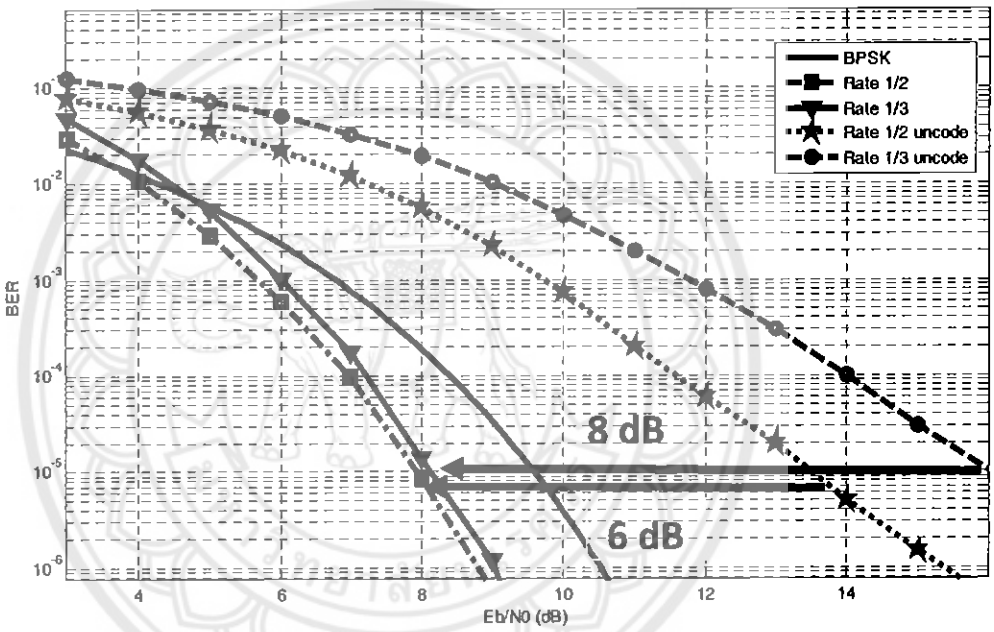
จากบทที่ 3 ได้ศึกษาทฤษฎี วิธีดำเนินงานต่างๆ และวิเคราะห์ BER เมื่อมีการประยุกต์การทำ Interleave มาใช้แล้วในบทนี้จะเป็นการนำเอาผลของการ simulate มาวิเคราะห์ว่าผลที่ได้ออกมานั้นเป็นอย่างไร และทำการเปรียบเทียบกับกรณีต่างๆ ในบทนี้จะแบ่งออกเป็น 3 ส่วน คือ ส่วนที่ 1 วิเคราะห์ผลของค่า K เมื่อมีการเปลี่ยนค่า K ในการทำ Convolutional Code มีผลกับค่า BER อย่างไร ส่วนที่ 2 จะเป็นการวิเคราะห์ว่าจำนวนบิตข้อมูลที่ใช้ในการทดลองมีผลกับค่า BER เป็นอย่างไร ส่วนที่ 3 จะเป็นการวิเคราะห์การเข้ารหัสแบบ Convolutional code ในกรณีที่มีการทำ Interleave และกรณีที่ไม่มี Interleave ว่าผลของ BER ได้ออกมานั้นเป็นอย่างไร ซึ่งทั้งหมดจะได้แสดงดังต่อไปนี้

4.1. ผลของการเข้ารหัส Convolutional code แบบ Hard decision ในกรณีที่มีการเปลี่ยนค่า K



รูปที่ 4.1 กราฟ BER Convolutional Code R=1/2 โดยใช้วิธีแบบ Hard-decision

จากการนำทฤษฎีเข้ามา จะเห็นว่ามีการเปลี่ยนค่า K ไปทั้งหมด 4 กรณีดังรูปที่ 4.1 จะพบว่าค่า K เพิ่มมากขึ้นในอัตรา Rate ที่เท่ากันจะทำให้ค่า BER นั้นดีขึ้น ในการที่ระบบมีหน่วยความจำมากขึ้นก็จะทำให้ระบบนั้นมีประสิทธิภาพมากขึ้นซึ่งเป็นผลมาจาก เมื่อมีการเพิ่มค่า K จะส่งผลให้ค่า d_{free} เพิ่มขึ้นตามไปด้วย และในอีกกรณีหนึ่งคือ กรณีที่ ค่า K เท่ากันแต่ Rate ไม่เท่ากันจะพบว่าอัตราส่วนของ Rate ที่มีค่าของ Output น้อยจะมีค่า BER ดีกว่า Rate ที่มีค่า Output มากกว่า แสดงดังรูปที่ 4.2



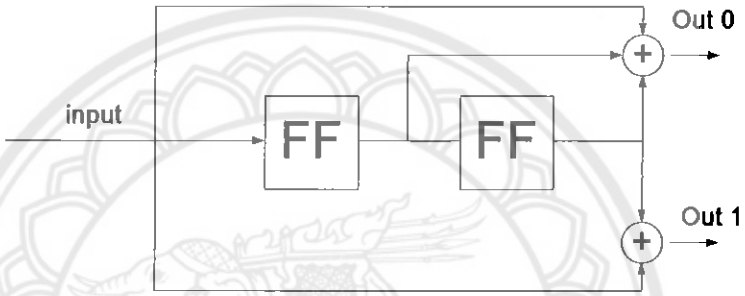
รูปที่ 4.2 กราฟเปรียบเทียบ BER Convolutional Code ของ Rate1/2 และ Rate 1/3ในกรณีที่ค่า K เท่ากัน

จากรูปที่ 4.2 จะเห็นว่าค่า BER ของกรณี Rate 1/2 มีค่าดีกว่า Rate 1/3 เนื่องจากว่าในกรณี Rate 1/2 มีค่า Bandwidth ที่น้อยกว่า ถึงแม้ว่า Rate 1/3 สามารถแก้ไขบิตผิดพลาดได้มากกว่าก็ตาม รูปที่ 4.2 ที่ BER เท่ากับ 10^{-5} จะเห็นได้อย่างชัดเจน เมื่อเปรียบเทียบระหว่างลูกศรสีแดงและลูกศรสีดำ จะเห็นได้ว่าลูกศรสีแดงยาวประมาณ 8 dB และลูกศรสีดำยาวประมาณ 6 dB ความยาวของลูกศรจะแสดงความสามารถในการแก้ไขบิตผิดพลาดจึงทำให้เห็นได้อย่างชัดเจนว่า Rate 1/3 สามารถแก้ไขบิตผิดพลาดได้มากกว่า แต่เมื่อดูจากค่า BER แล้ว Rate 1/2 จะดีกว่า Rate 1/3 เนื่องจากเหตุผลที่ว่า Rate 1/2 ใช้ Bandwidth น้อยกว่า Rate 1/3 ดังนั้นจากกราฟเราจึงวิเคราะห์ได้

ว่า Rate 1//3 สามารถแก้ไขบิตผิดพลาดได้มากกว่า Rate 1/2 แต่ค่า BER ของ Rate 1/2 จะดีกว่าในกรณีที่มีค่า K เท่ากัน

จากกรณีข้างต้นในรูปที่ 4.2 จะวิเคราะห์เพื่อหาข้อเท็จจริงจากข้อมูลในรูปที่ 4.2 ซึ่งเป็นผลจากการ simulate นั้นเป็นจริง ตามที่ได้วิเคราะห์ไปข้างต้นว่า ค่า BER Convolutional code ของ Rate 1/2 ดีกว่า Rate 1/3 ที่ค่า K=3 ทั้ง Rate 1/2 และ Rate 1/3 ซึ่งจะแสดงการคำนวณในส่วนของ Convolutional code แบบ Hard-decision แสดงการคำนวณดังต่อไปนี้

สำหรับกรณี Rate 1/2

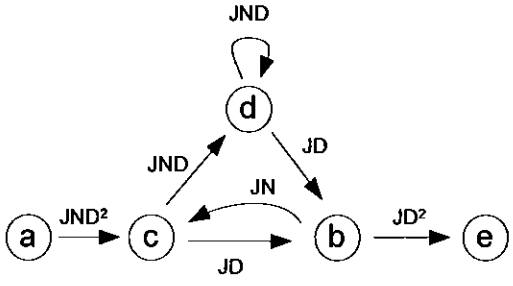


รูปที่ 4.3 การเข้ารหัสแบบ Convolutional code ที่ K=3 Rate 1/2

จากรูปที่ 4.3 มีค่า generator matrix(g) =[5 7] ทำการวิเคราะห์ค่า BER Rate 1/2

วาด State diagram จากรูปที่ 4.3 ได้เป็นดังรูปที่ 2.2 ซึ่งอยู่ในบทที่ 2 จากนั้นนำมาสร้าง State diagram ในรูปแบบที่สามารถนำไปสร้าง Transfer function ได้ดังรูปที่ 4.4 ดังนี้

เมื่อ a,e = 00 , b = 10 , c = 01 , d = 11



รูปที่ 4.4 State diagram Rate 1/2

จาก Transfer function เพื่อหาค่า $\frac{X_e}{X_a}$ จากสมกรที่ได้จาก state diagram

$$X_c = JND^2 X_a + JNX_b \tag{4.1}$$

$$X_b = JDX_c + JDX_d \tag{4.2}$$

$$X_d = JNDX_c + JNDX_d \tag{4.3}$$

$$X_e = JD^2 X_b \tag{4.4}$$

จากสมการที่ (4.1) จะได้

$$X_a = \frac{X_c - JNX_b}{JND^2} \tag{4.5}$$

จากสมการที่ (4.3) จะได้

$$X_c = \left(\frac{1-JND}{JND}\right) X_d \tag{4.6}$$

นำสมการที่ (4.6) แทนใน (4.2) จะได้

$$X_b = \frac{1}{N} X_d \tag{4.7}$$

นำสมการที่ (4.6) และ(4.7) แทนใน (4.5) จะได้

$$X_a = \left(\frac{1-JND-J^2ND}{J^2ND^3}\right) X_b \tag{4.8}$$

นำสมการที่ (4.4) หารด้วยสมการ (4.8) จะได้

$$T(D,N,J) = \left(\frac{J^3ND^5}{1-JND-J^2ND}\right) \tag{4.9}$$

ในกรณีนี้เราจะให้ $J=1$ แทนในสมการที่ (4.9) จะได้

$$T(D,N,1)=\frac{ND^5}{1-2ND}=ND^5+2N^2D^6+4N^3D^7+8N^4D^8+16N^5D^9+... \tag{4.10}$$

ในสมการที่ (4.10) นำมาเขียนอีกรูปแบบหนึ่งจะได้

$$a=[1,2,4,8,16],\beta=[1,4,12,32,80] \tag{4.11}$$

ค่า BER ของ Convolutional code สามารถหาได้จากสมการที่ (4.12)

$$P_e<\sum_{d=d_{free}}^{\infty}a\times P_2(d) \tag{4.12}$$

โดยที่

$$P_2(d)=\begin{cases}\displaystyle\sum_{k=\frac{d+1}{2}}^{d}\binom{d}{k}p^k(1-p)^k, d_{free}=odd\\ \displaystyle\sum_{k=(d/2)+1}^{d}\binom{d}{k}p^k(1-p)^{d-k}+\frac{1}{2}\binom{d}{\frac{d}{2}}p^{d/2}(1-p)^{d/2}, d_{free}=even\end{cases} \tag{4.13}$$

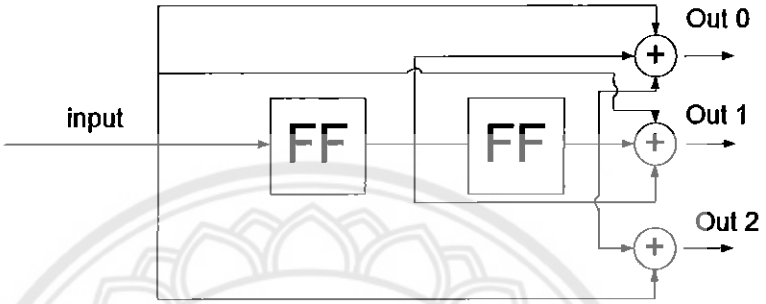
เมื่อแทนค่า a ใน (4.12) จะได้

$$P_e<P_2(5)+2P_2(6)+4P_2(7)+8P_2(8)+16P_2(9)$$

$$\begin{aligned}P_e<&\left[\binom{5}{3}p^3(1-p)^2+\binom{5}{4}p^4(1-p)+\binom{5}{5}p^5\right]+\\ &2\left[\frac{1}{2}\binom{6}{3}p^3(1-p)^3+\binom{6}{4}p^4(1-p)^2+\binom{6}{5}p^5(1-p)+\binom{6}{6}p^6\right]+\\ &4\left[\binom{7}{4}p^4(1-p)^3+\binom{7}{5}p^5(1-p)^2+\binom{7}{6}p^6(1-p)+\binom{7}{7}p^7\right]+\\ &8\left[\frac{1}{2}\binom{8}{4}p^4(1-p)^4+\binom{8}{5}p^5(1-p)^3+\binom{8}{6}p^6(1-p)^2+\binom{8}{7}p^7(1-p)+\binom{8}{8}p^8\right]+\\ &16\left[\binom{9}{5}p^5(1-p)^4+\binom{9}{6}p^6(1-p)^3+\binom{9}{7}p^7(1-p)^2+\binom{9}{8}p^8(1-p)+\binom{9}{9}p^9\right]\end{aligned} \tag{4.14}$$

จากสมการ (4.14) ที่ Eb/N0 เท่ากับ 5 เราจะได้ค่า BER =0.0024 ซึ่งเมื่อนำค่าที่ได้จากการคำนวณมาเปรียบเทียบกับค่าที่ได้จากการ simulate จะพบว่าค่าที่ได้ออกมาจากการคำนวณนั้นมีค่าเท่ากับค่าที่ได้จากการ simulate สังเกตได้จากรูปที่ 4.2

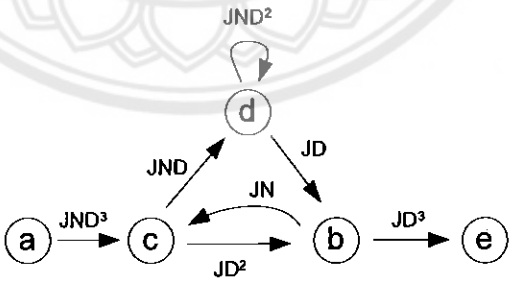
สำหรับกรณี Rate 1/3



รูปที่ 4.5 การเข้ารหัสแบบ Convolutional code ที่ K=3 Rate 1/3

วาด State diagram จากรูปที่ 4.5 นำมาสร้าง State diagram ในรูปแบบที่สามารถนำไปสร้าง Transfer function ได้ดังรูปที่ 4.6 ดังนี้

เมื่อ a,e = 00 , b = 10 , c = 01 , d = 11



รูปที่ 4.6 State diagram Rate 1/3

จาก Transfer function เพื่อหาค่า $\frac{X_e}{X_a}$ จากสมกรที่ได้จาก state diagram

$$X_c = JND^3 X_a + JNX_b \tag{4.15}$$

$$X_b = JD^2 X_c + JDX_d \tag{4.16}$$

$$X_d = JNDX_c + JND^2 X_d \tag{4.17}$$

$$X_e = JD^3 X_b \tag{4.18}$$

จากสมการที่ (4.17) จะได้

$$X_c = \frac{1 - JND^2}{JND} X_d \tag{4.19}$$

นำสมการที่ (4.17) แทนในสมการที่ (4.16) จะได้

$$X_b = \frac{JD^2(1 - JND^2) + JD(JND)}{JND} X_d \tag{4.20}$$

จากสมการที่ (4.15) จะได้

$$X_a = \frac{X_c - JNX_b}{JND^3} \tag{4.21}$$

นำสมการที่ (4.20) แทนใน (4.19) จะได้

$$X_c = \frac{1 - JND^2}{JD^2(1 - JND^2) + JD(JND)} X_b \tag{4.22}$$

นำสมการที่ (4.22) แทนในสมการที่ (4.18) จะได้

$$X_a = \frac{\frac{1 - JND^2}{JD^2(1 - JND^2) + JD(JND)} - JN}{JND^3} X_b \tag{4.23}$$

เมื่อนำสมการ (4.18) หารกับ สมการ (4.23) จะได้

$$T(D,N,J)=\frac{X_e}{X_a}=\frac{J^2ND^6[JD^2(1-JND^2)+JD(JND)]}{1-JND^2-JN[JD^2(1-JND^2)+JD(JND)]} \tag{4.24}$$

เมื่อกำหนดให้ค่า $J=1$ แทนลงในสมการที่ (4.24) จะได้

$$T(D,N,1)=ND^8+N^2D^8+N^2D^{10}+N^3D^{10}+2N^3D^{10}+N^4D^{10}+N^3D^{12}$$

$$+6N^4D^{12}+5N^5D^{12}+... \tag{4.25}$$

ในสมการที่ (4.25) นำมาเขียนอีกรูปแบบหนึ่งจะได้

$$a=[2,0,5,0,13],\beta=[3,0,15,0,36] \tag{4.26}$$

เมื่อแทนค่า a ใน (4.25) จะได้

$$P_e<2P_2(8)+5P_2(10)+13P_2(12)$$

$$P_e<\left[\frac{1}{2}\binom{8}{4}p^4(1-p)^4+\binom{8}{5}p^5(1-p)^3+\binom{8}{6}p^6(1-p)^2+\binom{8}{7}p^7(1-p)^1+\binom{8}{8}p^8\right]+$$

$$5\left[\frac{1}{2}\binom{10}{5}p^5(1-p)^5+\binom{10}{6}p^6(1-p)^4+\binom{10}{7}p^7(1-p)^3+\binom{10}{8}p^8(1-p)^2+\binom{10}{9}p^9(1-p)+\binom{10}{10}p^{10}\right]+ \tag{4.27}$$

$$13\left[\frac{1}{2}\binom{12}{6}p^6(1-p)^6+\binom{12}{7}p^7(1-p)^5+\binom{12}{8}p^8(1-p)^4+\binom{12}{9}p^9(1-p)^3+\binom{12}{10}p^{10}(1-p)^2+\binom{12}{11}p^{11}(1-p)+\binom{12}{12}p^{12}\right]$$

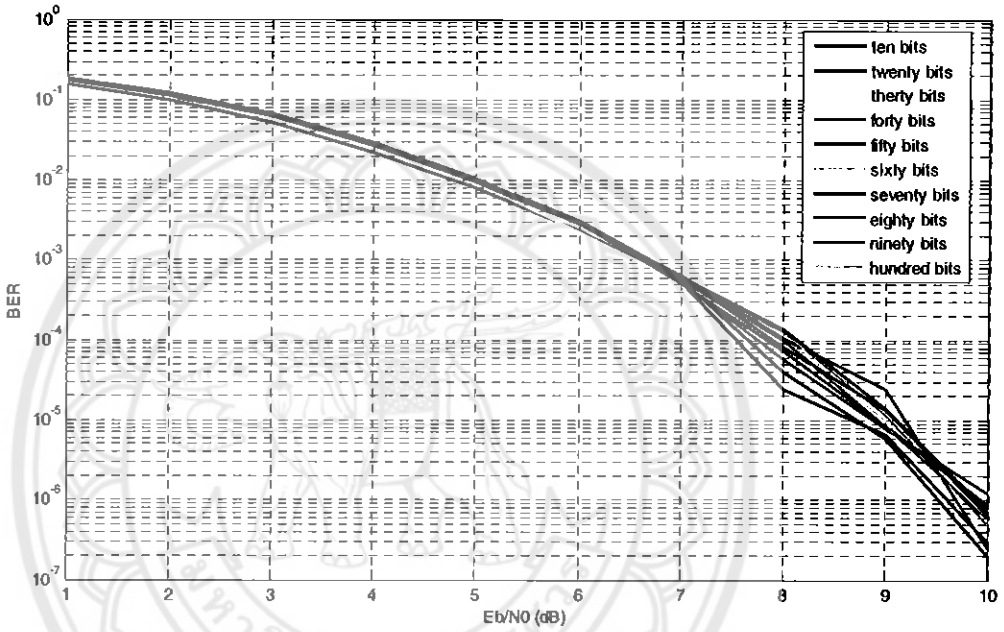
จากสมการ (4.27) ที่ Eb/N0 เท่ากับ 5 เราจะได้ค่า BER =0.0034ซึ่งเมื่อนำค่าที่ได้จากการคำนวณมาเปรียบเทียบกับค่าที่ได้จากการ simulate จะพบว่าค่าที่ได้ออกมาจากการคำนวณนั้นมีค่าเท่ากับค่าที่ได้จากการ simulate สังเกตได้จากรูปที่ 4.2

จากค่า BER ของ Rate 1/2 มีค่าท่ากับ 0.0024 และค่า BER ของ Rate 1/3 มีค่าเท่ากับ 0.0034 จะพิสูจน์ให้เห็นว่าค่า BER ของ Rate 1/2 มีค่าน้อยกว่า ค่า BER ของ Rate 1/3 ซึ่งจะทำให้ทราบว่าข้อมูลในกรณีที่ Rate 1/2 มีค่า BER ที่ดีกว่า จะเห็นได้ดังรูปที่ 4.2

การวิเคราะห์ดังกล่าวสามารถสรุปได้ว่า ค่า BER ของ Rate 1/2 มีค่าดีกว่าของ Rate 1/3 ส่วนในเรื่องของการแก้ไขบิตผิดพลาดนั้น Rate 1/3 สามารถแก้ไขบิตผิดพลาดได้มากกว่า Rate 1/2 จะเห็นได้อย่างชัดเจนว่า Rate 1/3 สามารถแก้ไขบิตผิดพลาดประมาณ 8 dB และ Rate 1/2 สามารถแก้ไขบิตผิดพลาดได้ประมาณ 3 dB แต่ค่า Bandwidth ของ Rate 1/3 จะมีค่ามากกว่า Rate 1/2 ดังนั้นจะเห็นว่าค่า BER ของระบบ Convolutional code ขึ้นอยู่กับค่า D_{free} และค่า Rate ซึ่งค่า Rate นั้นยิ่งน้อยจะทำให้ค่า Bandwidth มีค่ามากขึ้น และค่า D_{free} ค่าที่ออกมานั้นจะสอดคล้องกับค่า K เมื่อค่า K เปลี่ยนแปลงก็จะส่งผลให้กับค่า D_{free} ด้วย

4.2. ผลของการเข้ารหัส Convolutional code แบบ Hard decision ในกรณีที่มีจำนวน Input ไม่เท่ากัน

ในการเข้ารหัสแบบ Convolutional code จากการศึกษาไม่ได้มีการระบุว่าในการใช้จำนวนบิตข้อมูลเท่าไรจึงจะสามารถทำให้ระบบ Convolutional code ได้ประสิทธิภาพดีที่สุด ฉะนั้นจึงมีการทดลองในการเปลี่ยนจำนวนบิตข้อมูลในการเข้ารหัส เพื่อที่ดูว่าค่า BER ของจำนวนบิตเท่าไรถึงจะมีค่า BER ดีที่สุด ซึ่งผลจากการ simulate นั้นได้แสดงไว้ในรูปที่ 4.3

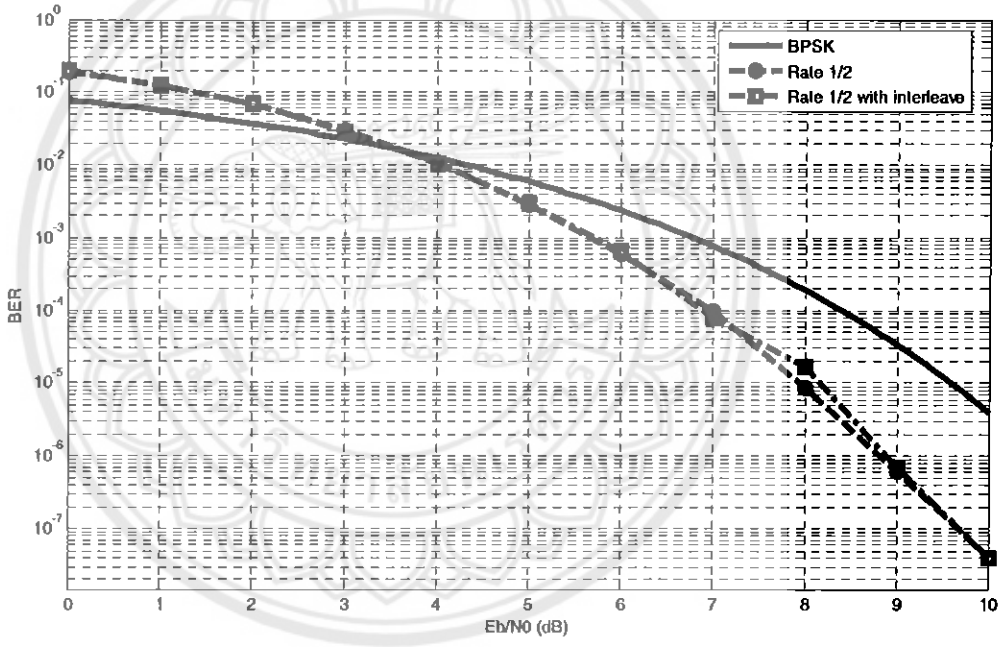


รูปที่ 4.7 แสดงการเปลี่ยนจำนวนข้อมูลเริ่มตั้งแต่ 10 บิต ไปจนถึง 100 บิต

จากรูปที่ 4.7 จะพบว่ามีการเปลี่ยนจำวนบิตข้อมูลที่ใช้ในการทำ Convolutional Code จะเห็นว่าเมื่อมีการเปลี่ยนข้อมูลในการเข้ารหัส ตั้งแต่ 10 บิต ไปจนถึง 100 บิต จะพบว่าค่า BER มีค่าใกล้เคียงกันมาก ที่ Eb/N0 มีค่า ตั้งแต่ 7-10 dB จะแตกต่างกันเพียงเล็กน้อย ซึ่งจะสามารถบอกได้ว่าการเปลี่ยนจำนวนบิตข้อมูลในการเข้ารหัสนั้นมีผลเพียงเล็กน้อย

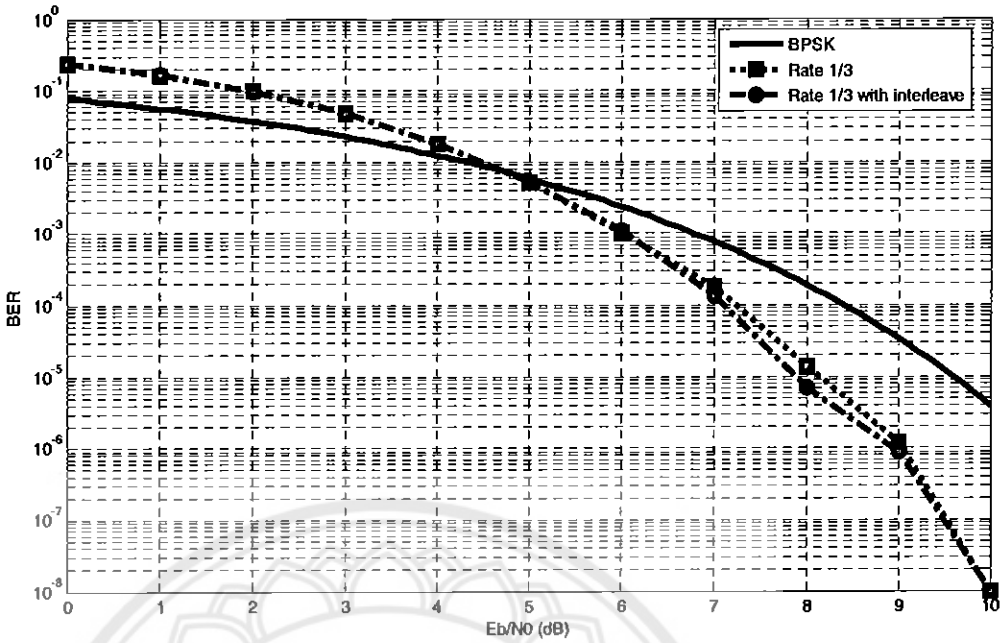
4.3. ผลของการเข้ารหัส Convolutional code แบบ Hard decision ในกรณีที่มี Interleave และไม่มี Interleave

ในการใช้ระบบ Convolutional code เป็นการแก้ไขบิตผิดพลาดไว้ล่วงหน้า เมื่อมีการใช้ระบบ Convolutional code นั้นเป็นผลทำให้ได้ค่า BER ออกมาดีกว่าที่ใช้เพียง BPSK ธรรมดา ในงานวิจัยนี้ยังมีการนำเอาระบบ Interleave เข้ามาใช้ร่วมกับระบบ Convolutional code เพื่อที่จะศึกษาว่าเมื่อมีการนำ Interleave เข้ามาใช้แล้วนั้นมีผลกับค่า BER อย่างไรซึ่งได้แบ่งการทดลองออกเป็น 3 กรณี คือ Rate 1/2 , Rate 1/3 , Rate 1/4 ซึ่งผลจากหัวข้อ 4.2 จึงมีการเลือกใช้จำนวนบิตข้อมูลเท่ากับ 100 บิตและผลการ simulate แสดงดังรูปที่ 4.8 4.9 และ 4.10 ตามลำดับ ต่อไปนี้



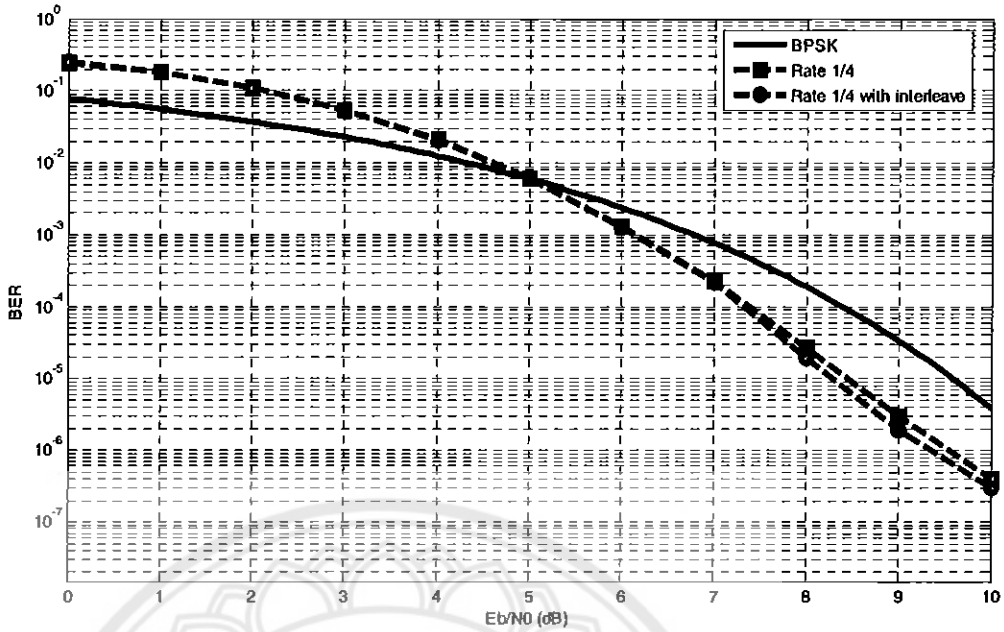
รูปที่ 4.8 แสดงการเปรียบเทียบ BER Convolutional Code (Hard-decision) ระหว่างที่มีการใช้ Interleave และไม่มี Interleave Rate 1/2

จากรูปที่ 4.8 นั้นเป็นกรณีที่ใช้ระบบ Convolutioal code Rate 1/2 ซึ่งเป็นการแสดงค่า BER ที่ใช้Interleave และไม่ได้ใช้ Interleave จะเห็นว่ามีค่าใกล้เคียงกันมาก จะสังเกตเห็นว่า Eb/N0 ในช่วง 1-7 dB ค่าที่ได้ออกมามีค่าเท่ากันพอดี และในช่วง 8-10 dB ค่าที่ได้ออกมาแตกต่างกันเพียงเล็กน้อย จะเห็นว่าโดยรวมแล้วกราฟที่ได้ออกมานั้น Interleave ไม่มีผลกับค่า BER



รูปที่ 4.9 แสดงการเปรียบเทียบ BER Convolutional Code (Hard-decision) ระหว่างที่มีการใช้ Interleave และไม่มี Interleave Rate 1/3

จากรูปที่ 4.9 นั้นเป็นกรณีที่ใช้ระบบ Convolutioal code Rate 1/3 ซึ่งเป็นการแสดงค่า BER ที่ใช้Interleave และไม่ได้ใช้ Interleave จะเห็นว่ามีค่าใกล้เคียงกันมาก จะสังเกตเห็นว่า Eb/N0 ในช่วง 1-6 dB ค่าที่ได้ออกมามีค่าเท่ากันพอดี และในช่วง 7-10 dB ค่าที่ได้ออกมาแตกต่างกันเพียงเล็กน้อย จะเห็นว่าโดยรวมแล้วกราฟที่ได้ออกมานั้น Interleave ไม่มีผลกับค่า BER แต่ค่า BER ที่ได้ออกมานั้นมีค่ามากกว่า Rate 1/2



รูปที่ 4.10 แสดงการเปรียบเทียบ BER Convolutional Code (Hard-decision) ระหว่างที่มีการใช้ Interleave และไม่มี Interleave Rate 1/4

จากรูปที่ 4.10 นั้นเป็นกรณีที่ใช้ระบบ Convolutioal code Rate 1/4 ซึ่งเป็นการแสดงค่า BER ที่ใช้Interleave และไม่ได้ใช้ Interleave จะเห็นว่ามีค่าใกล้เคียงกันมาก จะสังเกตเห็นว่า Eb/N0 ในช่วง 1-7 dB ค่าที่ได้ออกมามีค่าเท่ากันพอดี และในช่วง 8-10 dB ค่าที่ได้ออกมาแตกต่างกันเพียงเล็กน้อย จะเห็นว่าโดยรวมแล้วกราฟที่ได้ออกมานั้น Interleave ไม่มีผลกับค่า BER แต่ค่า BER ที่ได้ออกมานั้นมีค่ามากกว่า Rate 1/2 และ Rate 1/3

ผลจากการ simulate ข้างต้นจะได้ค่า BER ออกมาเมื่อสังเกตจะเห็นว่าค่าที่ได้ออกมานั้นมีค่าใกล้เคียงกันมากในทั้ง 2 กรณี คือ กรณีที่มี Interleave และไม่มี Interleave ประสิทธิภาพมีการเปลี่ยนแปลงไปเล็กน้อยเท่านั้นจากรูปที่ 4.8 จะเห็นได้อย่างชัดเจน ประสิทธิภาพของค่า BER ที่ออกมานั้นมีค่าใกล้เคียงกันมาก ซึ่งในการทดลองทั้ง 3 กรณีนั้นจะสามารถสรุปได้ว่าประสิทธิภาพของระบบ Convolutional code ที่ใช้Interleave ไม่มีผลกระทบกับประสิทธิภาพของกรณีที่ไม่ได้ใช้ Interleave

จากการสรุปข้อมูลข้างต้นที่ว่าการใช้ Interleave นั้นมีผลกระทบกับประสิทธิภาพของระบบ Convolutional code เพียงเล็กน้อย เหตุผลที่เป็นเช่นนั้นเนื่องจาก Interleave เป็นการแก้ไขการผิดพลาดที่ผิดแบบเบิรสต์ คือมีการผิดพลาดหลายๆบิตติดกัน แต่ในงานวิจัยนี้ได้มีการสุ่มสัญญาณรบกวนแบบ White Gaussian noise ซึ่งในการสุ่มแบบนี้จะเกิดการผิดพลาดแบบเบิรสต์น้อยมากหรืออาจจะไม่เกิดขึ้นเลย ฉะนั้นจึงเป็นเหตุผลที่ว่าในการนำเอา Interleave มาใช้ในระบบ Convolutional code จึงไม่ส่งผลกระทบกับประสิทธิภาพของระบบ

ในบทที่ 4 นี้สามารถสรุปได้ว่าปัจจัยที่มีผลกระทบกับระบบ Convolutional code นั้นมีอยู่ 2 ปัจจัยคือค่า Kและค่า Rate ซึ่งค่า K นั้นจะส่งผลกับค่า D_{free} และค่า Rate จะส่งผลกับค่า Bandwidth ในส่วนการเปลี่ยนจำนวนบิตข้อมูลในการเข้ารหัสนั้นจะส่งผลกับค่า BER เพียงเล็กน้อย และเมื่อมีการนำเอา Interleave เข้ามามีส่วนกับระบบนี้ส่งผลกับระบบเพียงเล็กน้อยนั้นเป็นเพราะเหตุผลในการสร้างสัญญาณรบกวนที่เป็นแบบ White Gaussian noise

บทที่ 5
สรุปผลงานวิจัย

ระบบ Convolutional code เป็นเทคนิคในการป้องกันบิตผิดพลาดล่วงหน้าซึ่งเป็นผลกับประสิทธิภาพทางด้าน BER การทำงานในส่วนการเข้ารหัสนั้นจะทำให้ได้ codeword ออกมาซึ่ง codeword นั้นจะมีจำนวนบิตมากกว่า จำนวนข้อมูลที่ส่งเข้าไปในระบบก่อนเข้ารหัสในเวลาที่เท่ากัน ทั้งนี้การที่จะได้ codeword จำนวนมากหรือน้อยขึ้นอยู่กับ Rate ที่ใช้ในระบบ ฉะนั้นเมื่อเราต้องการระบบที่น่าเชื่อถือจะต้องมีระบบที่ต้องใช้ Bandwidth มากขึ้นตามจำนวน Rate ที่ส่งผลกับระบบ ในส่วนของการถอดรหัสของ Convolutional code ในงานวิจัยนี้ศึกษาวิธีการถอดรหัสแบบ Hard-decision จะเป็นการทำงานเมื่อสัญญาณที่ผ่านการ Demodulate แบบ BPSK แล้วต้องทำการตัดสินให้เป็นบิต 0 1 และนำมาเข้า Trellis diagram แล้วจะได้บิตข้อมูลออกมา ในงานวิจัยนี้สามารถสรุปงานวิจัยนี้ออกเป็น 3 ส่วนคือ

- ศึกษาประสิทธิภาพของ Convolutional code
- ศึกษาประสิทธิภาพของจำนวนบิตข้อมูลที่ใช้ในการเข้ารหัส Convolutional code
- ศึกษาประสิทธิภาพของ Convolutional code ทั้งที่ใช้ Interleave และที่ไม่ได้ใช้ Interleave

5.1. ศึกษาประสิทธิภาพของ Convolutional code

จากผลการทดลองในบทที่ 4 สามารถสรุปได้ว่า ค่า BER ของระบบจะมีการเปลี่ยนแปลงตามปัจจัย 2 อย่างนี้ คือ ค่า K และค่า Rate ของระบบจะเห็นว่าเมื่อค่า Rate มีค่าน้อยลงจะทำให้ค่า Bandwidth มีค่าเพิ่มมากขึ้น การที่ Bandwidth เพิ่มขึ้นเป็นผลทำให้ค่า BER มีค่ามากขึ้นซึ่งการเพิ่มขึ้นของ BER นั้นจะทำให้ประสิทธิภาพของระบบมีค่าแย่ลง ในส่วนของค่า K จะมีผลกับค่า D_{free} เมื่อค่า K เพิ่มมากขึ้นจะส่งผลให้ค่า D_{free} มากขึ้นตามและเมื่อ D_{free} เพิ่มมากขึ้นส่งผลทำให้ค่า BER มีค่าน้อยลง การที่ BER มีค่าน้อยลงนั้นส่งผลให้ค่าประสิทธิภาพมีค่ามากขึ้น

5.2. ศึกษาประสิทธิภาพของจำนวนบิตข้อมูลที่ใช้ในการเข้ารหัส Convolutional code

จากผลการทดลองในบทที่ 4 สามารถสรุปผลได้ว่าในการเปลี่ยนจำนวนบิตข้อมูลที่ใช้ในการเข้ารหัสของระบบ Convolutional code จะเห็นว่ามีผลกับค่า BER เพียงเล็กน้อยในงานวิจัยนี้จะทำการเปลี่ยนจำนวนบิตข้อมูลที่ใช้เข้ารหัส ตั้งแต่ 10 บิตไปจนถึง 100 บิตข้อมูล ซึ่งสามารถบอกได้ว่าการเปลี่ยนจำนวนบิตข้อมูลมีผลกับค่า BER เพียงเล็กน้อย ฉะนั้นในการเปลี่ยนจำนวนบิตข้อมูลในการเข้ารหัสนั้นจะส่งผลกับประสิทธิภาพของระบบเพียงเล็กน้อย

5.3. ศึกษาประสิทธิภาพของ Convolutional code ทั้งที่ใช้ Interleave และที่ไม่ได้ใช้ Interleave

จากผลการทดลองในบทที่ 4 นั้นสามารถสรุปได้ว่าในการที่ระบบ Convolutional code นำเอาInterleave เข้ามาใช้ในระบบนั้นส่งผลกับค่า BER น้อยมาก ซึ่งเมื่อพิจารณาค่า Eb/N0 ในช่วง 1-6 dB ของทุกกรณีนั้นจะพบว่าไม่มีผลกระทบกับระบบจะเห็นได้จากค่า BER ที่ออกมานั้นมีค่าเท่ากัน ฉะนั้นจึงสามารถสรุปได้ว่าการใช้ Interleave เข้ามาในระบบ Convolutional code มีผลกับประสิทธิภาพของระบบเพียงเล็กน้อยเท่านั้น เหตุผลที่เป็นเช่นนั้นเนื่องจากในงานวิจัยนี้ได้มีการสุ่มสัญญาณรบกวนแบบ White Gaussian noise ซึ่งในการสุ่มแบบนี้จะเกิดการผิดพลาดแบบเบิรสต์น้อยมากหรืออาจจะไม่เกิดขึ้นเลย ฉะนั้น Interleave เป็นการกระจายบิตผิดพลาดที่อยู่ติดกันและเมื่อไม่เกิดการผิดแบบเบิรสต์ การใช้Interleave จึงไม่ส่งผลกับประสิทธิภาพของระบบ

เอกสารอ้างอิง

[1] http://www.kmitl.ac.th/dslabs สืบค้นเมื่อ 28 สิงหาคม 2553

[2] Bernard SKLAR. "Digital Communications Fundamention and Applications" Prentice-Hill, 1988

[3] http://www.ipsi.fraunhofer.de/mobile/teaching/mobinkom_ws0102/2Funk/faltung.htm สืบค้นเมื่อ 14 สิงหาคม 2553

[4] http://scholar.lib.vt.edu/theses/public/etd-7189715815/materials/chap2.pdf สืบค้นเมื่อ 15 สิงหาคม 2553

[5] http://www.cim.mcgill.ca/~latorres/Viterbi/va_alg.htm สืบค้นเมื่อ 15 สิงหาคม 2553

[6] พิสิฐ วนิชชานันท์ และคณะ. (2552). ทฤษฎีรหัสช่องสัญญาณ. บริษัท แอ็ปป้า พริ้นติ้ง กรุ๊ป จำกัด

[7] Wicker, Stephen B. "Error control coding : fundamentals and applications" Englewood Cliffs, N.J. : Prentice-Hall, Inc., c1995

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายการันตร์ พันเลิศพาณิชย์

ภูมิลำเนา 109/24 หมู่ 7 อ.บางระกำ จ.พิษณุโลก

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนจ่านกร้อง
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail:Karan_P@hotmail.com