

การลดความผิดพลาดของข้อมูลข่าวสารโดยใช้การวางสลับบิต
ของการเข้ารหัสแบบเทอร์โบ

BER INFORMATION REDUCTION BASED ON INTERLEAVING
ON TURBO-CODE



จอมใจ ศรีอำไพ
JOMJAI SRIAMPAI

ณพ.
๗197ก
๗54๘

เลขหมู่.....
เลขทะเบียน..... 60441
วัน,เดือน,ปี 29 ส.ย. 2549

b. 11585869
i.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2548
ISBN 974-15-1403-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**BER, INFORMATION REDUCTION BASED ON INTERLEAVING
ON TURBO-CODE**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATION ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2005

ISBN 974-15-1403-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2005

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	BER Information Reduction Based on Interleaving on Turbo-Code
Student	Miss Jomjai Sriampai
Student ID.	45061083
Degree	Master of Engineering
Programme	Telecommunication Engineering
Year	2005
Thesis Advisor	Assoc.Prof.Dr. Kobchai Dejhan

ABSTRACT

This paper presents a work which is carried out the information via channel. In actually, the information at receiver is uncompleted, also carries out of the solution to reduce (BER) bit error rate on system. This paper presents channel coding based on Turbo code on AWGN channel and decoding with by LOG-MAP based on iterative decoding which is algorithm to decode with energy spreading in all available Turbo codes. On basically, Turbo codes depend on several factors to increase the bit error rate in system, such as interleaving design, frame size, code vector with application to fading channel.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้ที่สามารถสำเร็จลุล่วงได้ดีเพราะได้รับความอนุเคราะห์จาก รศ.ดร.กอบชัย เดชหาญ ที่คอยให้ความช่วยเหลือและคำแนะนำ ในด้านการศึกษาของผู้วิจัย ตลอดจน และขอขอบพระคุณพ่อกับแม่ตลอดจนบุคคลในครอบครัว ที่ซึ่งเป็นบุคคลที่คอยห่วงใย และเป็นกำลังใจที่ดีตลอด ขอขอบคุณพี่เก่ง ปุ้ย เงิน กุ้ย รวมถึงพี่และเพื่อน ๆ ที่คอยให้ความช่วยเหลือด้วยดี

และสุดท้ายขอขอบคุณหัวหน้าและเพื่อนร่วมงานที่บริษัทโมโตโรล่า ประเทศไทย (จำกัด) ที่ให้การช่วยเหลือและสนับสนุนด้านเครื่องมือในการทดสอบเป็นอย่างสูง

จอมใจ ศรีอำไพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สิ่งที่น่าสนใจใหม่ในวิทยานิพนธ์.....	2
1.4 สมมติฐานของการศึกษา.....	2
1.5 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	3
1.6 ขอบเขตการวิจัย.....	3
1.7 ขั้นตอนของการศึกษา.....	3
บทที่ 2 ทฤษฎีและหลักการเข้ารหัสเทอร์โบ.....	5
2.1 หลักการทำงานของระบบสื่อสารแบบไร้สาย.....	5
2.1.1 แบบจำลองทางคณิตศาสตร์สำหรับจำลองลักษณะช่องสัญญาณ.....	6
2.1.2 การลดความผิดพลาดของการรับส่งข้อมูล.....	7
2.2 หลักการทำงานของวิธีการเข้ารหัสแบบต่าง ๆ.....	8
2.2.1 ข้อมูลกระบวนการตัดสินใจแบบหยาบ (Hard Decision) และละเอียด (Soft Decision).....	10
2.2.2 รหัสบล็อก (Block Codes).....	11
2.2.3 รหัสคอนโวลูชัน (Convolution Codes).....	12
2.2.4 รหัสทีซีเอ็ม (Trellis Coded Modulation).....	14
2.3 รหัสเทอร์โบ (Turbo Codes).....	15
2.3.1 ตัวเข้ารหัส RSC.....	15
2.3.2 การวางสลับบิต (Interleaving).....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.3 การสิ้นสุดเทรลลิส (Trellis Termination).....	19
2.3.4 การ Puncturing.....	20
2.4 การถอดรหัสเทอร์โบ.....	21
2.4.1 อัลกอริทึมแบบ BCJR.....	21
2.4.2 การถอดรหัสวนซ้ำ (Iteration Decoding) ของรหัสเทอร์โบ.....	22
2.4.2.1 ฟังก์ชันความน่าจะเป็นจริงแบบล็อก (Log-Likelihood Ratio).....	22
2.4.2.2 เอนด์พุดแบบซอฟต์แวร์ของช่องสัญญาณ (SISO).....	23
2.4.2.3 หลักการถอดรหัสแบบวนซ้ำ.....	25
2.4.3 อัลกอริทึมที่ใช้ในการถอดรหัส.....	27
2.4.3.1 อัลกอริทึมแบบ (Map).....	27
2.4.3.2 อัลกอริทึมแบบ (Log-Map).....	32
บทที่ 3 โครงสร้างการวางสลับบิต.....	36
3.1 คุณลักษณะการวางสลับบิต.....	36
3.2 โครงสร้างการทำงาน (Block Interleaver).....	38
3.3 โครงสร้างการทำงาน (Block Helical-Simile Interleaver).....	39
3.4 โครงสร้างการทำงาน (Random).....	39
3.5 โครงสร้างการทำงาน (Reverse Block Interleaver).....	40
3.6 โครงสร้างการทำงาน (Reverse Block Helical-Simile Interleaver).....	40
บทที่ 4 ผลการจำลองแบบการทำงานของระบบ.....	46
4.1 การจำลองสมรรถนะรหัสเทอร์โบบนช่องสัญญาณ AWGN.....	46
4.2 ผลของปัจจัยที่มีผลต่อสมรรถนะการเข้ารหัสเทอร์โบบนช่องสัญญาณ AWGN.....	47
4.2.1 การวนซ้ำ (Iteration Decoding).....	48
4.2.2 ขนาดเฟรมข้อมูล (Block size)	49
4.2.3 โครงสร้างการเข้ารหัส (Generator Matrix).....	49
4.2.4 อัตราการเข้ารหัส (Code Rate).....	50

สารบัญ (ต่อ)

	หน้า
4.3 ผลการเปรียบเทียบสมรรถนะรหัสเทอร์โบโดยการวางสลับบิต.....	55
4.3.1 ผลการเปรียบเทียบสมรรถนะBlock และ Reverse Block Interleaver ขนาด 100 บิต.....	56
4.3.2 ผลการเปรียบเทียบสมรรถนะ Block และ Reverse Block Interleaver ขนาด 99 บิต.....	57
4.3.3 ผลการเปรียบเทียบสมรรถนะ Block และ Reverse Block Interleaver ขนาด 1024 บิต.....	57
4.3.4 ผลการเปรียบเทียบสมรรถนะBlock Helical-Simile และ Reverse Block Helical-Simile Interleaver ขนาด 99 บิต.....	58
4.3.5 ผลการเปรียบเทียบสมรรถนะสมรรถนะBlock Helical-Simile และ Reverse Block Helical-Simile Interleaver ขนาด 624 บิต.....	58
4.3.6 ผลการเปรียบเทียบสมรรถนะ Block Helical-Simile และ Reverse Block Helical-Simile Interleaver ขนาด 1023 บิต.....	59
4.3.7 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 100 บิต.....	60
4.3.8 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 625 บิต.....	60
4.3.9 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 1024 บิต	61
บทที่ 5 สรุปผลการวิจัยและแนวทางในการพัฒนาต่อไป.....	62
5.1 สรุปผลการวิจัย.....	62
5.2 ข้อเสนอแนะแนวทางในการพัฒนาต่อไป.....	63
เอกสารอ้างอิง.....	64
ภาคผนวก บทความที่ได้รับการตีพิมพ์	66
ประวัติผู้เขียน.....	67

สารบัญรูป

รูปที่	หน้า
2.1 การส่งข้อมูลผ่านช่องสัญญาณไร้สาย.....	6
2.2 แบบจำลองทางคณิตศาสตร์ของช่องสัญญาณไร้สาย.....	6
2.3 การจำลองระบบสื่อสารที่มีการนำวิธีการเข้ารหัสมาใช้งาน.....	8
2.4 ผลลัพธ์ที่ได้จากการใช้วิธีการเข้ารหัสในระบบสื่อสาร.....	9
2.5 การทำงานการตรวจจับข้อมูลบนการตัดสินใจแบบหยาบและละเอียด.....	10
2.6 วงจรเข้ารหัสคอนโวลูชัน $G=[7_8, 5_8]$, $K=3$	12
2.7 ตัวอย่างของ Trellis Diagram.....	13
2.8 ตัวเข้ารหัสเทอร์โบ.....	15
2.9 ตัวเข้ารหัส RSC มีอัตราการเข้ารหัสเท่ากับ $1/2$ และหน่วยความจำ $\nu = 2$	16
2.10 แผนภาพการเปลี่ยนสถานะรหัส RSC มีอัตราการเข้ารหัสเท่ากับ $1/2$ และ $\nu = 2$	17
2.11 แผนภาพ Trellis ของรหัส RSC มีอัตราการเข้ารหัสเท่ากับ $1/2$ และ $\nu = 2$	18
2.12 แผนภาพ Trellis Termination.....	19
2.13 ตัวถอดรหัสแบบ SISO.....	25
2.14 ตัวถอดรหัสเทอร์โบ.....	26
2.15 แผนภาพแสดงการคำนวณแบบ (Forward Recursion).....	29
2.16 แผนภาพแสดงการคำนวณแบบ (Backward Recursion).....	30
3.1 โครงสร้างการเข้ารหัสเทอร์โบ.....	36
3.2 ตัวอย่างของคำรหัสที่ได้จากชุดเข้ารหัสเทอร์โบ.....	37
3.3 โครงสร้างการทำงาน (Block Interleaver).....	38
3.4 โครงสร้างการทำงาน (Block Helical-Simile Interleaver).....	39
3.5 โครงสร้างการทำงาน (Random Interleaver).....	39
3.6 โครงสร้างการทำงาน (Reverse Block Interleaver).....	40
3.7 โครงสร้างการทำงาน (Reverse Block Helical-Simile Interleaver).....	40
3.8 แผนภาพการทำงาน (Block Interleaver).....	41
3.9 แผนภาพการทำงาน (Block Helical-Simile Interleaver).....	42
3.10 แผนภาพการทำงาน (Random Interleaver).....	43
3.11 แผนภาพการทำงาน (Reverse Block Interleaver).....	44
3.12 แผนภาพการทำงาน (Reverse Block Helical-Simile Interleaver).....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 การจำลองสมรรถนะของรหัสเทอร์โบบนช่องสัญญาณ AWGN.....	47
4.2 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Iteration ต่าง ๆ	48
4.3 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Block Size	49
4.4 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Generator Matrix.....	49
4.5 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Code Rate.....	50
4.6 BER ของการวางสลับบิตแบบ Block ขนาด 100 บิต.....	50
4.7 BER ของการวางสลับบิตแบบ Block ขนาด 625 บิต.....	51
4.8 BER ของการวางสลับบิตแบบ Block ขนาด 1024 บิต.....	51
4.9 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 99 บิต.....	52
4.10 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 624 บิต.....	52
4.11 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 1023 บิต.....	53
4.12 BER ของการวางสลับบิตแบบ Random ขนาด 100 บิต.....	53
4.13 BER ของการวางสลับบิตแบบ Random ขนาด 100 บิต.....	54
4.14 BER ของการวางสลับบิตแบบ Random ขนาด 100 บิต.....	54
4.15 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 100 บิต....	56
4.16 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 625 บิต ...	57
4.17 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 1024 บิต...57	
4.18 การเปรียบเทียบ BER ของการวางสลับบิต Block Helical-Simile และ Reverse Block Helical-Simile ขนาด 99 บิต	58
4.19 การเปรียบเทียบ BER ของการวางสลับบิต Block Helical-Simile และ Reverse Block Helical-Simile ขนาด 624 บิต.....	58
4.20 การเปรียบเทียบ BER ของการวางสลับบิต Block Helical-Simile และ Reverse Block Helical-Simile ขนาด 1023 บิต	59
4.21 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 100 บิต.....	60
4.22 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 625 บิต.....	60
4.23 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 1024 บิต.....	61

สารบัญตาราง

ตารางที่

หน้า

3.1 ชุดลำดับข้อมูลทางด้านอินพุตและเอาต์พุตของชุดเข้ารหัส.....37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบการสื่อสารในปัจจุบันเป็นสิ่งมีความจำเป็นต่อชีวิตมนุษย์เป็นอย่างมากไม่ว่าจะเป็น การสื่อสารแบบใช้สายหรือแบบไร้สายโดยที่ การสื่อสารแบบไร้สายยุคที่สาม (3rd Generation of Wireless Communication) เป็นระบบสื่อสารที่สามารถส่งข้อมูลภาพพร้อมเสียง และข้อมูล อื่นๆ ได้อย่างมีประสิทธิภาพสูง จึงมีการศึกษาค้นคว้าเทคนิคที่สามารถทำให้สมรรถนะระบบดีขึ้น เช่น มีขนาดของแบนวิดท์ (Bandwidth) ที่กว้างขึ้นเพื่อให้สามารถจุข้อมูลได้เพิ่มมากขึ้นโดยให้ค่าอัตรา ความผิดพลาดการส่งข้อมูลต่ำ เป็นต้น การพัฒนาเทคโนโลยีสำหรับป้องกันความผิดพลาดของข้อมูล ในระบบสื่อสารไร้สายเป็นสิ่งหนึ่ง ที่มีความสำคัญและมีการพัฒนาอย่างต่อเนื่อง เพื่อให้ข้อมูล ดิจิตอลที่ซึ่งถูกส่งผ่าน ระบบสื่อสารมีค่าอัตราความผิดพลาดบิต (BER) ลดลง โดยสาเหตุหลักส่ง ผลทำให้ข้อมูล เกิดความผิดพลาดขึ้นเกิดการถูกรบกวนจากสัญญาณรบกวน (Noise) โดยเกิดจาก ลักษณะการส่งข้อมูลที่ซึ่งสัญญาณของข้อมูลสามารถแพร่ไปสู่ภาครับได้ จากหลายทิศทางด้วยกัน (Multipath Propagation) เนื่องจากในแต่ละเส้นทางอาจมีคุณสมบัติที่แตกต่างกันจึงส่งผลทำให้ สัญญาณข้อมูลที่ถูกตรวจจับได้ที่ภาครับเกิดความผิดพลาด ดังนั้นเพื่อปรับปรุงคุณภาพข้อมูลจึง พัฒนาวิธีการลดความผิดพลาดข้อมูล วิทยานิพนธ์นี้จึงได้นำเทคนิคการเข้ารหัสช่องสัญญาณแบบ เทอร์โบ (Turbo Code) มีหลายปัจจัยในการปรับปรุงสมรรถนะระบบ ได้แก่ รูปแบบวางสลับบิตข้อมูล ขาวสาร (Interleaver Design) ขนาดเฟรมข้อมูล (Frame Size) โครงสร้างข้อมูล (Generator Matrix) และหน่วยความจำ (Constraint Length) โดยการเปลี่ยนแปลงพารามิเตอร์ ต่าง ๆ เพื่อใช้ วิเคราะห์อัตราความผิดพลาดบิตข้อมูลที่เกิดขึ้น ในระบบแสดงให้เห็นจากการจำลองแบบเข้ารหัส ช่องสัญญาณ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

การส่งผ่านสัญญาณดิจิตอลในระบบสื่อสารทั่วไปมักเกิดปัญหาการผิดเพี้ยนของรูปแบบ สัญญาณเนื่องจากคุณสมบัติที่ไม่เป็นอุดมคติของช่องสัญญาณ หรือเป็นผลกระทบของสัญญาณ ภายนอกกรบกวนในรูปแบบต่าง ๆ ปัญหาเหล่านี้มีส่งผลให้ข้อมูลดิจิตอลที่ภาครับมีความผิดพลาด เกิดขึ้น ซึ่งเหตุนี้ระบบสื่อสารในปัจจุบันต้องการความถูกต้องและแน่นอนในการส่งข้อมูลสูงจึงได้มีการ นำข้อมูลดิจิตอลไปผ่านกระบวนการเข้ารหัสช่องสัญญาณ(Channel Coding)ก่อนจะส่งข้อมูล ออกเพื่อให้การรับส่งข้อมูลมีความผิดพลาดลดลง การเข้ารหัสช่องสัญญาณต้องมีการเพิ่มจำนวน ของบิตที่ส่งออกโดยมีการบิตพิเศษที่เพิ่มมา ที่ช่วยให้ภาครับสามารถจะตรวจจับความผิดพลาดได้

(Error Detection) โดยเลือกการเข้ารหัสของสัญญาณแบบเทอร์โบ ซึ่งมีผลให้อัตราการผิดพลาดของข้อมูลลดลง

1.ศึกษาหลักการทํางานรหัสเทอร์โบ การเข้ารหัส การถอดรหัสวนซ้ำ บนอัลกอริทึมแบบ (Log-Map Algorithm)

2.ศึกษาคุณลักษณะโครงสร้างการวางสลับบิตบนการเข้ารหัสเทอร์โบบนช่องสัญญาณ AWGN เพื่อตรวจสอบอัตราความผิดพลาดของระบบ (BER)

3.ศึกษาคุณลักษณะโครงสร้างการวางสลับบิตที่มีการดัดแปลงบนช่องสัญญาณ AWGN เพื่อตรวจสอบอัตราความผิดพลาดระบบ (BER) และเปรียบเทียบกับกรวางสลับบิตแบบเดิม

วิทยานิพนธ์นี้มีการค้นคว้าและวิจัยซึ่งมีหลายปัจจัยที่ช่วยปรับปรุงสมรรถนะของระบบซึ่งมีการวางสลับบิตสามารถช่วยลดอัตราความผิดพลาดของข้อมูลเพื่อให้ระบบมีประสิทธิภาพมากขึ้น

1.3 สิ่งที่น่าสนใจใหม่ในวิทยานิพนธ์

วิทยานิพนธ์เสนอการรูปแบบใหม่ในการวางสลับตำแหน่งบิตข้อมูล (Interleaver Design) ภายในโครงสร้างของการเข้ารหัสเทอร์โบ โดยได้ใช้วิธีการเลียนแบบโครงสร้างการวางสลับบิตแบบเดิมที่ประกอบด้วยกรวางสลับบิตแบบทิศทางตรงข้ามบล็อก (Reverse-Block Interleaver) และแบบทิศทางตรงข้ามแบบบล็อกเฮลิคอลลิมาย (Reverse-Block Helical-Simile Interleaver) มีพื้นฐานของการพัฒนาที่รูปแบบการวางสลับบิตบล็อก (Block Interleaver) และการวางสลับบิตบล็อกเฮลิคอลลิมาย (Block Helical-Simile Interleaver) เพื่อลดอัตราความผิดพลาดของข้อมูลข่าวสาร หรือเป็นทางเลือกในการปรับปรุงสมรรถนะของระบบ

1.4 สมมติฐานของการศึกษา

รูปแบบในการวางสลับบิตข้อมูลโดยทั่วไปนั้นมีหลายวิธีได้มีการนำเสนอในงานวิจัยต่าง ๆ บ้างว่า สามารถนำมาใช้เป็นส่วนหนึ่งในโครงสร้างการเข้ารหัสเทอร์โบซึ่งสามารถลดอัตราความผิดพลาดของข้อมูล โดยที่วัตถุประสงค์การวางสลับบิตข้อมูลข่าวสารคือการลดขนาดของกลุ่มความผิดพลาดข้อมูล (Burst Error) ระดับหนึ่งและหลังจากนั้นอาจใช้วิธีการอื่น ๆ ช่วยตรวจสอบความผิดพลาดซ้ำซึ่งสามารถลดความซับซ้อนในการตรวจซ้ำ ดังนั้นการพัฒนากรวางสลับตำแหน่งข้อมูลให้เหมาะสมกับการเข้ารหัสเทอร์โบ จึงเป็นแนวทางหนึ่งที่น่าจะเป็นการเพิ่มความสามารถของประสิทธิภาพโดยรวมของการสื่อสารของข้อมูลในยุคปัจจุบันได้ และมีเงื่อนไขหรือขีดจำกัดความสามารถในการกระจายกลุ่มของความผิดพลาดที่เมื่อทําการทํางานร่วมกับวิธีการถอดรหัสแบบเทอร์โบ

1.5 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

ระบบการสื่อสารยุคปัจจุบันต้องการความถูกต้องและความแม่นยำในการส่งข้อมูลสูง โดยที่มีการนำข้อมูลดิจิทัลไปผ่านกระบวนการเข้ารหัสของสัญญาณ (Channel Coding) ก่อนที่ส่งไปยังภาครับ สำหรับวิทยานิพนธ์นี้จะมีการใช้เทคนิคการเข้ารหัสของสัญญาณ แบบเทอร์โบ ซึ่งมีข้อดีจากจำนวนรอบในการวนซ้ำข้อมูลของการถอดรหัส สำหรับการเข้ารหัสแบบเทอร์โบ มีหลายปัจจัยที่ช่วยให้มีการเปลี่ยนแปลงสมรรถนะระบบ ได้แก่ รูปแบบการวางสลับบิต (Interleaver Design) โครงสร้างของการเข้ารหัส (Generator Matrix) ขนาดเฟรมข้อมูล (Frame Size) และหน่วยความจำ (Constraint Length) ซึ่งสำหรับวิทยานิพนธ์นี้เลือกการเปลี่ยนแปลงรูปแบบการวางสลับตำแหน่งข้อมูลที่ทำงานร่วมกับผลดีของการวนซ้ำในการถอดรหัส (Iteration Decoding) แบบเทอร์โบส่งผลให้อัตราความผิดพลาดของข้อมูลลดลง และเพื่อเป็นทางเลือกในการเปลี่ยนแปลงโครงสร้างการเข้ารหัสที่ลดความซับซ้อนของโครงสร้างการเข้ารหัสแบบเทอร์โบ

1.6 ขอบเขตของการวิจัย

การวิจัยของวิทยานิพนธ์นี้ได้ศึกษาและวิจัยเกี่ยวกับการเปลี่ยนแปลงรูปแบบโครงสร้างการวางสลับตำแหน่งบิตข้อมูลภายในโครงสร้างการเข้ารหัสเทอร์โบรวมไปถึงศึกษาปัจจัยที่มีผลในการปรับปรุงสมรรถนะของระบบสำหรับการเข้ารหัสของสัญญาณเทอร์โบ ได้แก่ การวางสลับตำแหน่งบิตข้อมูล (Interleaver Design) ได้แก่ การวางสลับบิตแบบสุ่ม บล็อก และบล็อกเฮลิคอลลิมายซึ่งจำนวนรอบของการวนซ้ำ การถอดรหัส (Iteration Decoding) ขนาดของเฟรมข้อมูล (Frame Size) มีอัตราการส่งข้อมูล (Code Rate) และโครงสร้างการเข้ารหัส (Generator Matrix) หน่วยความจำ (Constraint Length) ซึ่งการเปรียบเทียบการเข้ารหัส แบบเทอร์โบพื้นฐานมีการเปลี่ยนตัววางสลับบิตโดยเขียนโปรแกรมจำลองลงบนโปรแกรม (MATLAB) และคำนวณหาค่าอัตราความผิดพลาด (BER) ที่อัตราส่วน E_b/N_0 ที่อัตรา ส่วนต่าง ๆ ด้วยวิธีมอนติคาร์โล (Monte Carlo Method) ใช้วิธีการถอดรหัสแบบ (Log Map)

1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์เล่มนี้ได้จัดแบ่งเนื้อหาออกเป็น 5 บท ซึ่งภายในแต่ละบทจะประกอบด้วยรายละเอียดต่าง ๆ ได้ดังนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษาสิ่งนำเสนอใหม่ในวิทยานิพนธ์ สมมติฐานของการศึกษาทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย ขอบเขตของการวิจัยและขั้นตอนของการศึกษา

บทที่ 2 อธิบายหลักการทำงานระบบสื่อสารไร้สาย พื้นฐานวิธีการเข้ารหัสแบบคอนโวลูชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีการต่อแบบขนานและการทำงานของเข้ารหัสแบบเทอร์โบ และการถอดรหัสแบบเทอร์โบโดยใช้ อัลกอริทึมแบบ (Map) และ (Log Map)

บทที่ 3 อธิบายถึงโครงสร้างการวางสลับบิตทั้ง 5 ชนิดที่ได้มีการนำไปใช้ในการจำลอง เลียนแบบ ได้แก่ Block, Reverse Block, Block Helical-Simile, Reverse Block Helical-Simile และ Random

บทที่ 4 อธิบายถึงวิธีการจำลองและผลการจำลองแบบบนช่องสัญญาณ AWGN ซึ่งแสดง สมรรถนะของรหัสเทอร์โบโดยที่มีการเปลี่ยนแปลงรูปแบบการวางสลับบิตชนิดต่าง ๆ

บทที่ 5 เป็นการสรุปผลการวิจัยและแนวทางที่จะสามารถนำผลงานวิจัยนี้ไปพัฒนาต่อไป หรือสามารถปรับใช้ในด้านอื่น ๆ และระบบการสื่อสารอื่น ๆ ต่อไป

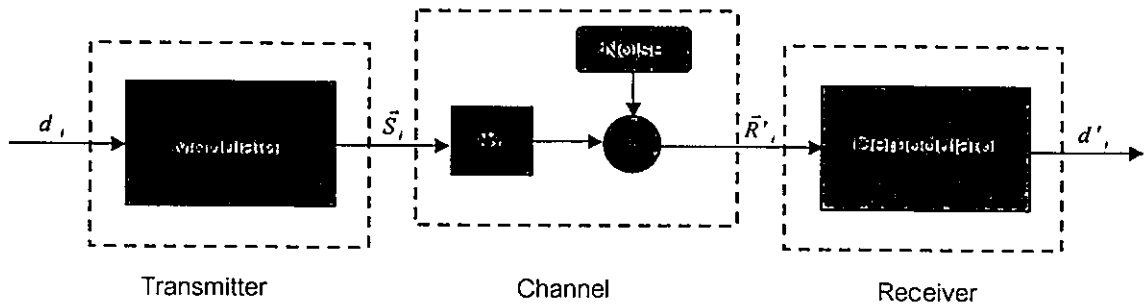


ทฤษฎีและหลักการเข้ารหัสเทอร์โบ

ระบบการสื่อสารไร้สายเป็นรูปแบบหนึ่งของการสื่อสารที่ใช้สำหรับส่งข้อมูลสู่ภาครับผ่านช่องสัญญาณแบบไร้สาย ตัวอย่างเช่น การส่งข้อมูลโดยใช้คลื่นแม่เหล็กไฟฟ้า (Microwave) ระบบสื่อสารผ่านดาวเทียม และระบบโทรศัพท์เคลื่อนที่ เป็นต้น เนื่องจากที่คุณลักษณะของสัญญาณไร้สายสามารถเปลี่ยนแปลงไปตามองค์ประกอบต่าง ๆ ได้ เช่น สิ่งแวดล้อมทางกายภาพ ความเร็วในการเคลื่อนที่ของอุปกรณ์ภาครับหรือส่ง และ อุณหภูมิ เป็นต้น ส่งผลให้สัญญาณข้อมูลที่ถูกตรวจจับได้ ณ ภาครับอาจมีลักษณะที่ผิดเพี้ยน ดังนั้นจึงได้มีการพัฒนาเทคโนโลยีและวิธีการต่าง ๆ ขึ้นมาเพื่อรองรับปัญหาดังที่กล่าวมาแล้ว สำหรับบทนี้แสดงแบบจำลองทางคณิตศาสตร์เพื่อจำลองลักษณะของสัญญาณเพื่อให้เข้าใจถึงปัญหาหลักที่เกิดขึ้นในการส่งข้อมูลผ่านช่องสัญญาณไร้สายและความจำเป็นในการนำหลักการเข้ารหัสมาใช้งาน

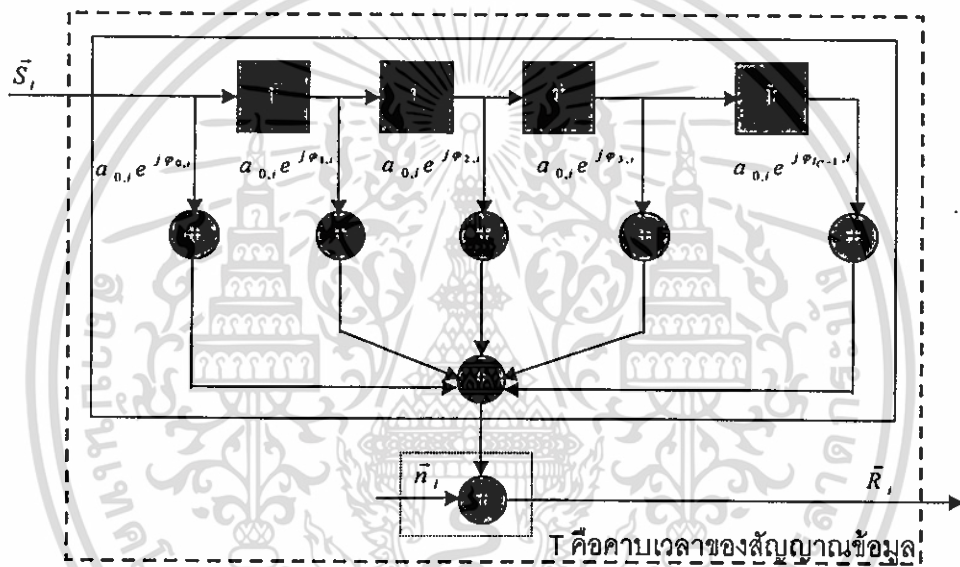
2.1 หลักการทำงานของระบบสื่อสารแบบไร้สาย

สำหรับการออกแบบใช้งานระบบสื่อสารไร้สายเพื่อส่งข้อมูลดิจิทัลนั้นมีความจำเป็นในการพิจารณาองค์ประกอบต่าง ๆ ระบบสื่อสาร เช่น ขนาดแบนด์วิดท์ (Bandwidth) ช่องสัญญาณที่ใช้งานอัตราเร็วในการส่งข้อมูล และกำลังส่ง เป็นต้น โดยจะต้องมีการออกแบบให้เหมาะสมเพื่อให้ใช้งานการระบบสื่อสารได้อย่างมีประสิทธิภาพและไม่รบกวนช่องสัญญาณอื่น ๆ ซึ่งดังนั้น ข้อมูลดิจิทัล d , ต้องมีการผ่านกระบวนการมอดูเลต (Modulated) ก่อนจะถูกส่งผ่านระบบสื่อสารเพื่อแปลงข้อมูลให้กลายเป็น สัญญาณข้อมูล S , ที่มีคุณสมบัติเหมาะสมต่อการส่งผ่านช่องสัญญาณและเมื่อสัญญาณ R , ข้อมูลโดยถูกตรวจจับได้ที่ภาครับ จะถูกนำมาผ่านกระบวนการดีมอดูเลต (Demodulated) เพื่อแปลงสัญญาณข้อมูลให้กลายเป็นข้อมูลดิจิทัล d' , เพื่อนำไปใช้งานโดยที่จะพบว่าข้อมูลดิจิทัลที่ได้จากทำงานของภาครับอาจมีลักษณะที่ผิดเพี้ยนเมื่อเทียบกับข้อมูลที่ถูกส่งออกไปจากภาคส่งซึ่งเกิดมาจากผลขององค์ประกอบต่าง ๆ ในระบบสื่อสารองค์ประกอบหลักที่ส่งผลให้ข้อมูลที่ส่งผ่านระบบสื่อสารเกิดความผิดพลาดขึ้นนั้น โดยเกิดมาจากการถูกรบกวนจากสัญญาณรบกวนและเกิดจากลักษณะในการส่งข้อมูลที่สัญญาณข้อมูล S , สามารถเดินทางไปถึงภาครับได้หลายทิศทางด้วยกันโดยที่ในแต่ละเส้นทางอาจมีคุณสมบัติที่แตกต่าง จึงส่งผลทำให้เกิดความผิดเพี้ยนของสัญญาณในลักษณะของการทับซ้อนระหว่างสัญลักษณ์ (ISI) เกิดขึ้น โดยที่ผลดังกล่าวเป็นตัวแปรสำคัญทำให้ระบบสื่อสารไร้สายเกิดความผิดพลาด ดังนั้นจึงมีวิธีการเข้ารหัสและวิธีการปรับแต่งสัญญาณเพื่อเป็นการลดผลกระทบดังกล่าว



รูปที่ 2.1 การส่งข้อมูลผ่านช่องสัญญาณไร้สาย

2.1.1 แบบจำลองทางคณิตศาสตร์สำหรับจำลองลักษณะของช่องสัญญาณ



รูปที่ 2.2 แบบจำลองทางคณิตศาสตร์ของช่องสัญญาณไร้สาย

การแสดงผลลักษณะช่องสัญญาณแบบไร้สายซึ่งแบบจำลองทางคณิตศาสตร์ถูกนำมาใช้สำหรับแสดงผลกระทบของสัญญาณรบกวนและการทับซ้อนระหว่างสัญลักษณ์ที่เกิดขึ้นในช่องสัญญาณซึ่งกรณีที่มีการพิจารณาสัญญาณข้อมูลทุก ๆ คาบเวลา T พบว่าลักษณะของสัญญาณข้อมูล \bar{R}_i ที่ถูกตรวจจับได้ที่ภาครับ ณ เวลาใด ๆ มีลักษณะความสัมพันธ์กับสัญญาณข้อมูล \bar{S}_i ที่ถูกส่งออกไปจาก ต้นทางตามสมการที่ (2.1)

$$\begin{aligned}\bar{R}_i &= (a_{0,i} e^{j\phi_{0,i}} \cdot \bar{S}_i + a_{1,i} e^{j\phi_{1,i}} \cdot \bar{S}_{i-1} + \dots + a_{L_c-1,i} e^{j\phi_{L_c-1,i}} \cdot \bar{S}_{i-L_c+1}) + \bar{n}_i \\ &= \sum_{k=0}^{L_c-1} (a_{k,i} e^{j\phi_{k,i}} \cdot \bar{S}_k) + \bar{n}_i\end{aligned}\quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวแปร $a_{k,i}$ และ $\varphi_{k,i}$ เป็นขนาดและเฟสของการจางหายของสัญญาณในเส้นทางใด ๆ L_c เป็นจำนวนของสัญญาณข้อมูลทั้งหมดถูกนำมาคำนวณและ \bar{n} เป็นตัวแปรสุ่มสำหรับแสดงลักษณะของสัญญาณรบกวนเป็นตัวแปรที่มีการแจกแจงข้อมูลแบบเกาส์ (Gaussian Distribution) ซึ่งมีลักษณะดังสมการที่ (2.2)

$$p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(n-\bar{n})^2}{2\sigma^2}} \quad (2.2)$$

สำหรับ σ^2 และ \bar{n} เป็นตัวแปรถูกนำมาใช้สำหรับกำหนดค่าความแปรปรวน (Variance) และค่าเฉลี่ย (Mean) ของข้อมูลที่ได้จากการสุ่ม ดังนั้นจากสมการที่ (2.1) สามารถนำไปใช้ในการเขียนแบบจำลองสำหรับช่องสัญญาณแบบไร้สายได้ดังรูปที่ 2.2

2.1.2 การลดความผิดพลาดของการรับและส่งข้อมูล

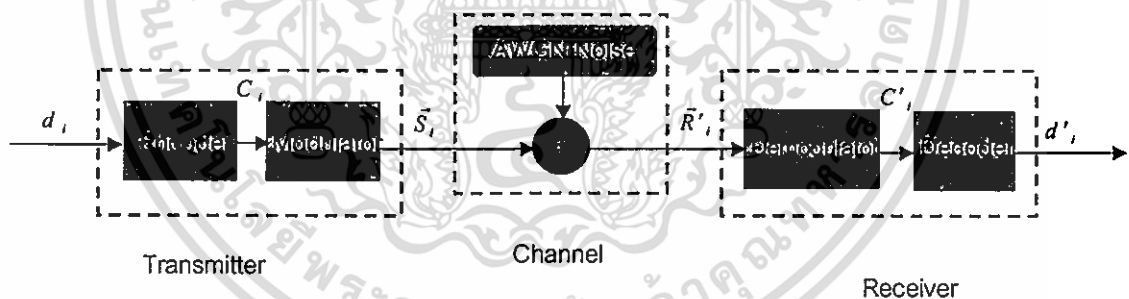
จากการศึกษาคุณสมบัติของสัญญาณแบบไร้สาย พบว่าลักษณะของสัญญาณข้อมูล R' ที่ถูกตรวจจับได้ ณ ภาครับ อาจมีลักษณะที่ผิดเพี้ยนไปได้ ดังนั้นจึงได้มีการพัฒนารูปแบบในการทำงานของระบบสื่อสารขึ้นมาเพื่อลดความผิดพลาดของข้อมูลที่ถูกส่งผ่านระบบสื่อสารสามารถทำได้หลายวิธีการด้วยกัน เช่น การเพิ่มกำลังของเครื่องส่งให้สูงขึ้น เป็นต้น สำหรับแนวทางที่มีความสำคัญและถูกนำมาใช้งานในระบบสื่อสารหลายประเภทด้วยกัน ได้แก่ การนำหลักการของวิธีการเข้ารหัสสัญญาณเข้ามาใช้งาน เพื่อลดผลกระทบของสัญญาณรบกวนและการทับซ้อนระหว่างสัญลักษณ์ซึ่งเป็นวิธีการที่นิยมใช้งานและถูกพัฒนาอย่างต่อเนื่อง

วิธีการเข้ารหัสข้อมูล เป็นวิธีการใช้สำหรับลดความผิดพลาดในการส่งข้อมูลผ่านระบบสื่อสาร ผลลัพธ์ที่ได้จากการทำงานสามารถทำให้ข้อมูลที่ถูกส่งผ่านระบบสื่อสารมีความถูกต้องมากยิ่งขึ้นโดยไม่ต้องมีการเพิ่มกำลังของเครื่องส่งสัญญาณต้นทาง แต่จะต้องส่งข้อมูลผ่านระบบสื่อสารด้วยปริมาณ ที่มากขึ้นถูกนำมาใช้งานหลายประเภท เช่น ระบบโทรศัพท์เคลื่อนที่ ระบบสื่อสารผ่านดาวเทียมและอุปกรณ์สำหรับบันทึกข้อมูลดิจิทัล เป็นต้น ซึ่งมีวิธีการเข้ารหัสที่ใช้ในการทำงานหลายประเภท โดยที่ในแต่ละรูปแบบนั้นมีความเหมาะสมกับการใช้งานในลักษณะที่แตกต่างกันออกไป

2.2 หลักการทำงานพื้นฐานการเข้ารหัสแบบต่าง ๆ

วิธีการเข้ารหัสมีอยู่หลายรูปแบบที่มีการใช้งานกันอย่างแพร่หลาย ได้แก่ รหัสแบบบล็อก รหัสคอนโวลูชัน และรหัสเทอร์โบ เนื่องจากเป็นชนิดของรหัสที่มีประสิทธิภาพและถูกนำมาประยุกต์ใช้งาน ในระบบสื่อสารและอุปกรณ์หลายประเภท

วิธีการเข้ารหัสถูกพัฒนาขึ้น เพื่อลดอัตราการเกิดความผิดพลาดของข้อมูลที่ถูกส่งผ่านระบบสื่อสารสามารถทำให้อัตราการเกิดความผิดพลาดของข้อมูลลดลง โดยที่ปราศจากการเพิ่มกำลังเครื่องส่งให้สูงขึ้นได้ ซึ่งในการทำงานนั้นข้อมูลดิจิทัล d , ถูกส่งผ่านระบบสื่อสารจะถูกนำมาผ่านกระบวนการเข้ารหัส (Encoding) เพื่อแปลงลักษณะของข้อมูลให้เป็นข้อมูลคำรหัส (Codeword) C , โดยเป็นรูปแบบของข้อมูลดิจิทัลที่ถูกส่งผ่านระบบสื่อสาร สำหรับภาครับมีการนำข้อมูลที่แสดงถึงข้อมูลของคำรหัสที่ถูกตรวจจับได้ผ่านกระบวนการถอดรหัส (Decoding) เพื่อแปลงข้อมูลให้กลายเป็นข้อมูลดิจิทัล d' , เพื่อนำไปใช้งานต่อไป พร้อมทั้งแก้ไขข้อมูลให้มีความถูกต้องมากขึ้นซึ่งรูปแบบของรหัสที่ถูกใช้งานหลายรูปแบบด้วยกัน โดยทั่วไปแล้วมีการแบ่งออกเป็น 2 รูปแบบได้แก่ รหัสบล็อก และรหัสแบบคอนโวลูชัน สำหรับรหัสแบบบล็อกเป็นรูปแบบแรกของรหัสที่ถูกพัฒนาขึ้นถูกนำไปใช้ประยุกต์ในงานหลายประเภท เช่น รหัส Hamming และรหัส Reed Solomon ที่ถูกนำมาใช้ในอุปกรณ์สำหรับการบันทึกข้อมูลเพื่อป้องกันความผิดพลาด เป็นต้น สำหรับรหัสแบบคอนโวลูชันที่ถูกพัฒนาขึ้น เพื่อรองรับการใช้งานสำหรับระบบสื่อสารที่มีการส่งข้อมูลด้วยอัตราเร็วสูง ดังเช่น ระบบสื่อสารผ่านดาวเทียมและระบบโทรศัพท์เคลื่อนที่ เป็นต้น โดยมีการพัฒนารูปแบบในการทำงานของถอดรหัส เพื่อรองรับการทำงานรหัสคอนโวลูชันมีหลายแบบ เช่น วิธีการถอดรหัสแบบ Sequential Decoding วิธีการ Threshold Decoding [4] และวิธีการ Maximum a Posteriori (MAP) แต่วิธีการถอดรหัสที่นิยมใช้งาน ได้แก่ วิธีการ Viterbi Decoding เนื่องจากประสิทธิภาพที่ได้รับมีความเหมาะสมและวงจรที่ใช้งานไม่ซับซ้อน

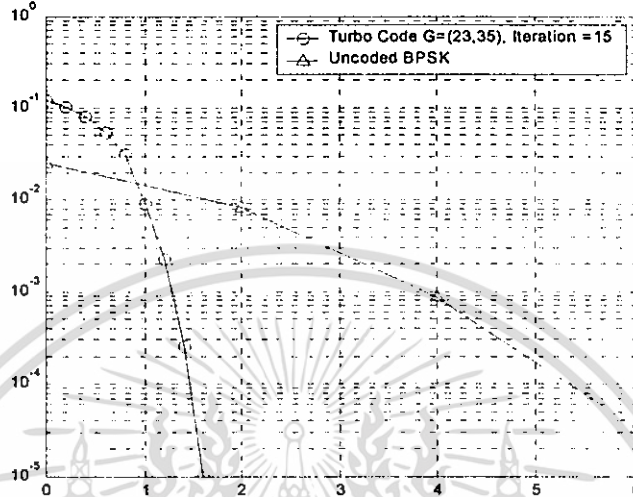


รูปที่ 2.3 การจำลองระบบสื่อสารที่มีการนำวิธีการเข้ารหัสมาใช้งาน

เนื่องจากความต้องการในการใช้งานระบบสื่อสารที่มีการส่งข้อมูลด้วยอัตราเร็วที่สูงขึ้นภายใต้ขนาดแบนด์วิดท์ของสัญญาณจำกัด ดังนั้นจึงได้มีการพัฒนารูปแบบรหัสให้สามารถใช้งานร่วมกันกับระบบสื่อสารที่มีการมอดูเลตแบบหลายระดับได้ เช่น รหัส Trellis Coded Modulation (TCM) ซึ่งเป็นวิธีการที่ถูกพัฒนาขึ้นมาจากรหัสคอนโวลูชันสำหรับรหัสเทอร์โบที่ถูกพัฒนาขึ้นมาในปี พ.ศ.2536 โดย Alan Glavieux, Claude Berrou และPunya Thitmajshima [3] เป็นรูปแบบของรหัสที่มีประสิทธิภาพสูงและนำไปใช้ในการสื่อสาร เช่น ระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 ระบบ Digital Video Broadcast (DVB) และ Deep Space Communications เป็นต้น นอกจากนี้ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้มีการนำหลักการรหัสเทอร์โบมาดัดแปลง และประยุกต์ใช้ร่วมกับรหัสรูปแบบอื่น ๆ เช่น รหัส Turbo-Trellis Coded Modulation (T-TCM) ซึ่งถูกนำไปใช้งานในระบบสื่อสารผ่านดาวเทียม และ รหัส Turbo Product Codes (TPC) หรือ Block Turbo Codes ซึ่งเป็นการนำรหัสแบบบล็อกมาประยุกต์ใช้งานเพื่อให้การทำงานมีความซับซ้อนที่ลดลง



รูปที่ 2.4 ผลลัพธ์ที่ได้จากการใช้วิธีการเข้ารหัสในระบบสื่อสาร

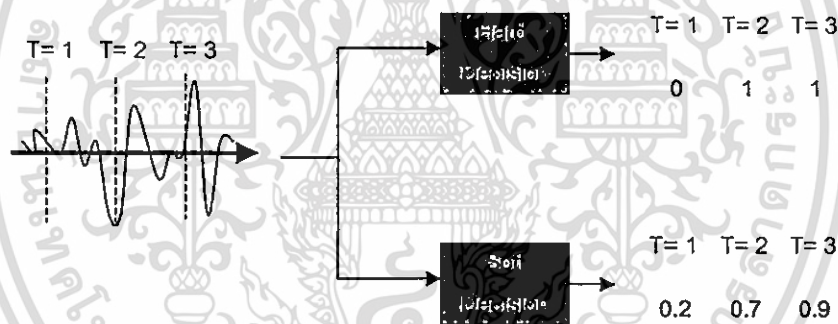
สำหรับอัตราการเกิดความผิดพลาดข้อมูลที่ถูกส่งผ่านระบบสื่อสารโดยทั่วไปเป็นค่าที่มีความสัมพันธ์กับอัตราส่วนระหว่างกำลังงานสัญญาณที่ส่งและสัญญาณรบกวน (Signal-to-Noise Ratio : SNR) โดยจะมีลักษณะความสัมพันธ์ดังรูปที่ 2.4 ซึ่งมีการเปรียบเทียบค่าอัตราความผิดพลาดของบิต (Bit Error Rate : BER) สำหรับระบบสื่อสารที่ไม่เข้ารหัสกับระบบสื่อสารที่มีการใช้วิธีการเข้ารหัสแบบเทอร์โบมีค่า Generator Matrix คือ $[23_g, 35_g]$ ขนาด 1024 บิต ด้วยอัตราการเข้ารหัสเท่ากับ $1/2$ และ ใช้วิธีการถอดรหัสแบบ Log-MAP มีการวนรอบในการถอดรหัสจำนวน 15 รอบ ซึ่งจากรูปที่ 2.4 พบว่านำวิธีการเข้ารหัสมาใช้งานสามารถลดความต้องการค่า SNR สำหรับส่งข้อมูลลงได้ โดยที่ยังคงทำให้ค่าอัตราการเกิดความผิดพลาดข้อมูลมีค่าเท่าเดิม ที่ได้มีการนิยามค่าอัตราขยายการเข้ารหัส (Coding Gain) ขึ้นมาเพื่อแสดงถึงความสามารถลดความต้องการค่า SNR ที่ใช้งานของวิธีการเข้ารหัสรูปแบบต่าง ๆ ได้เนื่องจากข้อมูลคำรหัส C , ที่ได้จากระบวนการทำงานของเข้ารหัสมีขนาดที่เพิ่มขึ้น เมื่อมีการเปรียบเทียบกับข้อมูลดิบ d , ดังนั้นข้อมูลที่ถูกส่งผ่านระบบสื่อสารจะมีขนาดที่ใหญ่มากขึ้นได้มีการนิยามตัวแปรอัตราการเข้ารหัส (Code Rate :R) ขึ้นเพื่อแสดงอัตราส่วนระหว่างขนาดของข้อมูลดิบทั้งหมดที่ป้อนเข้าสู่วงจรเข้ารหัสกับขนาดของข้อมูลคำรหัสทั้งหมดที่ได้จากการทำงานของภาคเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ข้อมูลกระบวนการตัดสินใจแบบหยาบ (Hard-Decision) และละเอียด (Soft-Decision)

จากการศึกษาหลักการของวิธีการเข้ารหัสพบว่าลักษณะของข้อมูลที่ถูกใช้งานที่ภาคถอดรหัส เป็นตัวแปรหนึ่งที่มีความสำคัญและส่งผลต่อประสิทธิภาพที่ได้รับ โดยเฉพาะรหัสเทอร์โบที่รูปแบบของข้อมูล ที่ถูกใช้งานกับภาคถอดรหัสมีผลกระทบต่อประสิทธิภาพที่ได้รับ เมื่อเทียบกับวิธีการอื่น ๆ เนื่อง จากหลักการดังกล่าวมาใช้สำหรับการพัฒนารูปแบบการตรวจจับข้อมูล และ ดังนั้นจึงได้อธิบายถึงรูปแบบของข้อมูลแบบหยาบและละเอียดใช้งานสำหรับการถอดรหัสโดย สำหรับการดำเนินงานของภาคตรวจจับซึ่งถูกใช้สำหรับแปลงสัญญาณข้อมูลที่ภาครับมีลักษณะการตัดสินใจ 2 ลักษณะด้วยกัน ได้แก่ การตัดสินใจแบบหยาบและละเอียด กรณีที่พิจารณา ระบบสื่อสารมีการส่งสัญญาณแบบ On-Off Keying (OOK) มีลักษณะการส่งสัญญาณข้อมูลเพื่อแสดงถึงข้อมูลดิจิทัล "0" หรือ "1" ดังสมการที่ (2.3)

$$r(t) = \begin{cases} 0, & \text{if } = 0 \text{ (is not transmitted)} \\ s(t), & \text{if } = 1 \text{ (is transmitted)} \end{cases} \quad (2.3)$$



รูปที่ 2.5 การทำงานการตรวจจับข้อมูลบนการตัดสินใจแบบหยาบและละเอียด

เมื่อ $s(t)$ เป็นรูปแบบสัญญาณที่ใช้แสดงถึงข้อมูลดิจิทัล "1" ที่มีการส่งข้อมูลด้วยพลังงาน E ดังนั้น จะพบว่าลักษณะของสัญญาณข้อมูลที่ถูกตรวจจับได้ที่ภาครับมีลักษณะดังสมการที่ (2.4)

$$r(t) = \begin{cases} n(t), & \text{if } = 0 \text{ (is not transmitted)} \\ s(t)+n(t), & \text{if } = 1 \text{ (is transmitted)} \end{cases} \quad (2.4)$$

โดย $n(t)$ แสดงถึงผลของสัญญาณที่รบกวนเกิดขึ้นภายในช่องสัญญาณสำหรับกรณี ที่ช่องสัญญาณ มีสัญญาณรบกวนแบบ Additive White Gaussian Noise (AWGN) ลักษณะกระจายข้อมูลเป็นแบบเกาส์ พบว่าลักษณะของสัญญาณข้อมูลที่ถูกตรวจจับได้ที่ภาครับ มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการกระจายข้อมูลดังรูปที่ 2.5 สำหรับการทำงานภาครับเพื่อตรวจจับข้อมูลที่ถูกส่งมา สามารถทำได้โดยนำคุณลักษณะดังกล่าว มาใช้งานในกรณีที่มีการตรวจจับข้อมูลโดยใช้วิธีการตัดสินใจแบบหยาบเป็นการนำสัญญาณข้อมูลที่ถูกรวบรวมได้ ณ เวลาต่าง ๆ มาทำการเปรียบเทียบเพื่อหารูปแบบสัญญาณที่มีความเป็นไปได้มากที่สุดที่จะถูกส่งมาเพื่อส่งลักษณะของสัญญาณที่ถูกเลือกออกเป็นผลลัพธ์ของภาคตรวจจับข้อมูล แต่ในกรณีที่ภาคตรวจจับข้อมูลมีการใช้วิธีการตัดสินใจแบบละเอียดแล้ว จะมีการนำข้อมูลแสดงถึงพลังงานหรือลักษณะของสัญญาณที่ถูกรวบรวมได้ส่งออกเป็นผลลัพธ์ภาคตรวจจับซึ่งเป็นรูปแบบของข้อมูลที่มีความละเอียดมากกว่าวิธีการแรกแต่การทำงานของภาคตรวจจับมีความซับซ้อนมากขึ้นตามไปด้วย และจากการศึกษาหลักการการทำงานของวิธีการเข้ารหัสพบว่า การนำข้อมูลแบบละเอียดเข้ามาใช้ในการประมวลผลของการเข้ารหัสเป็นอีกสิ่งหนึ่งที่มีการพิจารณาโดยเฉพาะระบบสื่อสารที่มีการใช้วิธีการเข้ารหัสแบบเทอร์โบเข้ามาใช้งานซึ่งจากการรวบรวมข้อมูลพบว่า ข้อมูลที่ถูกป้อนเข้าสู่ภาคถอดรหัสแบบเทอร์โบควรเป็นข้อมูลที่ได้จากกระบวนการตัดสินใจแบบละเอียด เนื่องจากเป็นรูปแบบที่เหมาะสมกับการทำงานของภาคถอดรหัสแบบเทอร์โบ

รูปแบบรหัสถูกนำมาใช้งานมีหลายรูปแบบ ซึ่งมีการทำงานและมีประสิทธิภาพที่แตกต่างกัน ออกไป กล่าวถึงรูปแบบของรหัสที่สำคัญ ได้แก่ รหัสแบบบล็อก และรหัสคอนโวลูชันซึ่งเป็นรูปแบบพื้นฐานของรหัส Trellis-Coded Modulation ถูกพัฒนาขึ้น โดยที่ระบบสื่อสารที่มีการมอดูเลตแบบหลายระดับ และรหัสเทอร์โบเป็นรูปแบบของรหัสที่มีประสิทธิภาพสูงโดยมีรายละเอียดดังต่อไปนี้

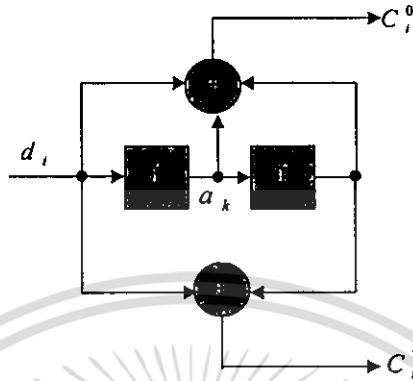
2.2.2 รหัสบล็อก (Block Codes)

รหัสบล็อกเป็นรูปแบบรหัสที่ถูกพัฒนาขึ้นโดยใช้หลักการทางคณิตศาสตร์ขั้นสูงสำหรับการทำงานของภาคเข้ารหัสแบบบล็อก ข้อมูลดิบ d , จะถูกนำมาคำนวณครั้งละ k บิต เพื่อหารูปแบบของข้อมูลคำรหัส C , ขนาดเท่ากับ n บิตที่แสดงถึงข้อมูลข่าวสาร โดยมีการแทนลักษณะการทำงานด้วยค่าสัญลักษณ์ (n, k) สำหรับการทำงานของภาคถอดรหัสมีการนำข้อมูลคำรหัสที่ถูกรวบรวมได้ที่ภาครับมาประมวลผลครั้งละ n บิต เพื่อตรวจจับหรือแก้ไขข้อมูลที่เกิดความผิดพลาดรูปแบบของรหัสบล็อกที่นิยมใช้งานมีหลายแบบ เช่น Cyclic Codes, Hamming Codes และ Reed-Solomon Codes

2.2.3 รหัสคอนโวลูชัน (Convolution Codes)

รหัสคอนโวลูชัน [6] ถูกพัฒนาในปี พ.ศ. 2497 โดย P. Elias เพื่อที่รองรับการทำงานระบบการสื่อสารที่ส่งข้อมูลด้วยอัตราเร็วสูง สำหรับการทำงานของภาคเข้ารหัสคอนโวลูชันเป็นการนำของข้อมูล ดิจิตอลมาประมวลผลครั้งละ k บิต เพื่อคำนวณค่าของข้อมูลคำรหัสจำนวน n บิต

โดยการนำข้อมูลที่ถูส่งเข้ามาในวงจรในก่อนหน้าจำนวนหนึ่งใช้คำนวณร่วมกับข้อมูลที่ถู
ป้อนเข้ามาในวงจรขณะนั้น วงจรเข้ารหัสที่ใช้งาน นำอุปกรณ์หน่วยความจำหรือชิพรีจิสเตอร์
สำหรับเก็บข้อมูล และวงจรวกแบบมอดูโล-2 มาใช้เป็นอุปกรณ์หลักในการทำงาน โดยมีลักษณะ
ดังตัวอย่างการเข้ารหัสรูปที่ 2.6



รูปที่ 2.6 วงจรเข้ารหัสคอนโวลูชัน $G=[7,5]_8$, $K=3$

รูปที่ 2.6 เป็นวงจรเข้ารหัสคอนโวลูชันที่มีการนำข้อมูลข่าวสารมาคำนวณครั้งละ 1 บิต ซึ่ง
หลังจากรับข้อมูลเข้ามา ข้อมูลข่าวสารที่ผ่านการประมวลผลไปแล้วถูกเก็บไว้ในหน่วยความจำจะ
ถูกเลื่อนตำแหน่งไปเก็บไว้ในหน่วยความจำชุดถัดไปและจะมีการนำข้อมูลทั้งหมดที่ถูกเก็บไว้ใน
หน่วยความจำ ใช้เพื่อคำนวณค่าของข้อมูลเข้ารหัสทั้ง 2 บิต โดยใช้วงจรวกเลขแบบมอดูโล-2 ทุก
ๆ ครั้งที่ได้มีการป้อนข้อมูลดิจิตอลเข้ามาในวงจรมีวงจรวกเลขแบบมอดูโล-2 ในรูปที่ 2.6 มีลักษณะการคำนวณ
ดังสมการ

$$C_i^0 = d_i \oplus d_{i-1} \oplus d_{i-2} \quad (2.5)$$

$$C_i^1 = d_i \oplus d_{i-2} \quad (2.6)$$

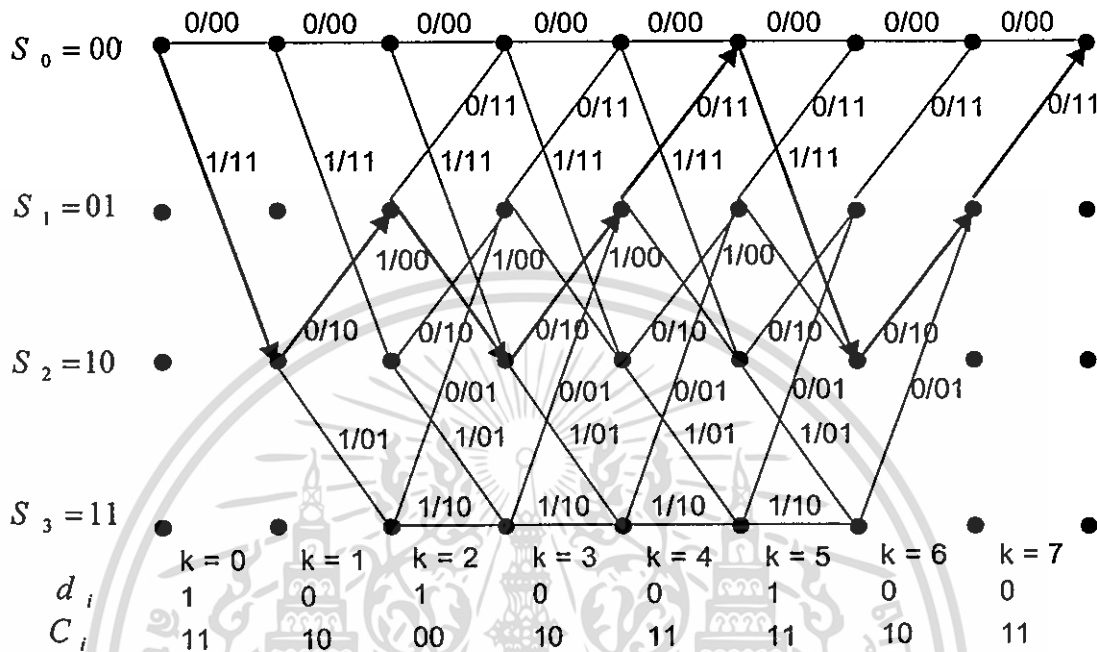
โดยทั่วไปแล้วมีการใช้ค่า Generator Matrix (G) สำหรับอธิบายลักษณะการคำนวณข้อมูลเข้ารหัส
เป็น ชุดข้อมูลตัวเลขฐาน 8 แสดงถึงตำแหน่งข้อมูลที่อยู่ในหน่วยความจำที่ถูกนำมาใช้คำนวณค่า
ข้อมูลเข้ารหัส ในบิตต่าง ๆ เช่น วงจรในรูปที่ 2.6 ค่า Generator Matrix ของวงจรรหัสมีค่าเป็น

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [7,5]_8 \quad (2.7)$$

นอกจากนั้นมีการนิยามตัวแปร Constraint Length (K) เพื่อแสดงถึงจำนวนของข้อมูลทั้งหมดถูก
ใช้ในการคำนวณกรณีวงจรรหัสในรูปที่ 2.6 นั้น มีค่า K เป็น 3 สำหรับการวิเคราะห์การทำงานของวงจรรหัส
เข้ารหัสคอนโวลูชันสามารถทำได้หลายวิธี โดยการนำรูปภาพ Trellis Diagram มาอธิบายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานของวงจรเข้ารหัสจะเป็นแนวทางหนึ่งที่ยอมรับใช้งานเป็นรูปภาพที่แสดงถึงความสัมพันธ์ระหว่างข้อมูลที่ถูกเก็บไว้ในหน่วยความจำข้อมูลที่ถูกป้อนเข้ามาในวงจรและข้อมูลที่ถูกส่งออกไปจากวงจรจะมีลักษณะดังตัวอย่างรูปที่ 2.7 ซึ่งเป็นตัวอย่างเส้นทางของ Trellis Diagram ของวงจรเข้ารหัสในรูปที่ 2.6



รูปที่ 2.7 ตัวอย่าง Trellis Diagram

เป็น Trellis Diagram สำหรับวิเคราะห์การทำงานของวงจรเข้ารหัสกรณีที่มีการข้อมูลจำนวน 6 บิตที่ป้อนเข้าสู่วงจรเข้ารหัสและภายหลังจากเข้ารหัสข้อมูล จะมีการทำให้ข้อมูลที่อยู่ในวงจรเข้ารหัสกลับสู่สภาวะเริ่มต้นมีข้อมูลทั้งหมดเป็นศูนย์อีกครั้ง โดยป้อนข้อมูลดิจิทัล "0" สำหรับจุดใด ๆ อยู่ในรูปของ Trellis Diagram ถูกใช้เพื่อแสดงถึงข้อมูลดิจิทัลที่เป็นไปได้ทั้งหมดที่ถูกเก็บไว้ในหน่วยความจำภายในวงจร ณ เวลาต่าง ๆ เส้นที่ถูกลากระหว่างจุดในรูปมีเครื่องหมาย (x_i, x_{01}, x_{02}) จะแสดงถึงลักษณะในการเปลี่ยนแปลงสถานะของข้อมูลที่อยู่ในหน่วยความจำที่มีค่าตรงกับจุดที่อยู่ต้นทางไปเป็นสถานะตามจุดที่ปลายทางเมื่อมีการรับข้อมูล x_i เข้ามาภายในวงจรและข้อมูลค่ารหัส ซึ่งได้จากการประมวลผลมีค่า เป็น x_{01}, x_{02} สำหรับการถอดรหัสข้อมูลค่ารหัสที่ถูกตรวจจับได้ในกรณีที่ใช้วิธีการเข้ารหัสคอนโวลูชัน มีรูปแบบใช้ทำงานอยู่หลายวิธี เช่น วิธีการ Sequential Decoding วิธีการ Threshold Decoding และ วิธีการ Viterbi Decoding เป็นต้น

การถอดรหัสแบบ Viterbi เป็นวิธีการที่ยอมรับใช้งานเนื่องจากมีประสิทธิภาพและความซับซ้อน ของวงจรที่ใช้งานไม่ซับซ้อนเป็นการถอดรหัสโดยนำรูปภาพ Trellis Diagram เข้ามาใช้ในการประมวลผลร่วมกับข้อมูลค่ารหัสที่ถูกตรวจจับได้ เพื่อคำนวณหาค่าของข้อมูลที่จะถูกส่งออก

ไปเป็นผลลัพธ์ของภาคถอดรหัส โดยที่นำข้อมูลคำรหัสทั้งหมดที่ถูกตรวจจับได้ที่ภาครับ มาทำการเปรียบเทียบกับข้อมูลในเส้นทางทั้งหมดที่อยู่ใน Trellis Diagram เพื่อคำนวณหาเส้นทางใด ๆ มีลักษณะตรงกับสัญญาณที่ถูกตรวจจับได้มากที่สุด เพื่อนำข้อมูลที่อยู่ในเส้นทางมาใช้สำหรับการกำเนิดข้อมูลดิบเพื่อส่งออกจากภาคถอดรหัส โดยใช้ Viterbi Algorithm เป็นวิธีการหลักสำหรับค้นหาเส้นทางที่มีความเหมาะสมที่สุด

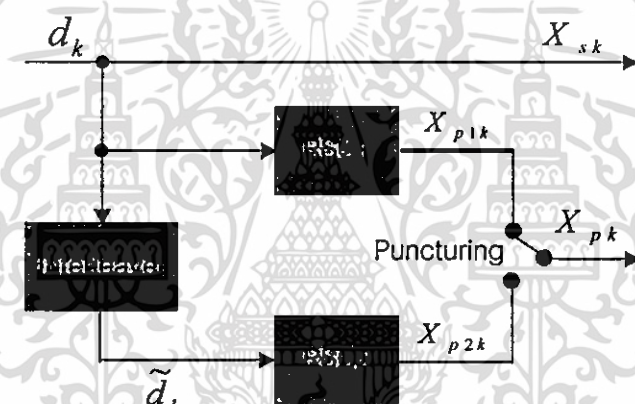
2.2.4 รหัสทีซีเอ็ม (Trellis Coded Modulation)

รหัส Trellis Coded Modulation หรือ TCM ซึ่งเป็นรหัสที่ถูกพัฒนาขึ้นในปี พ.ศ.2519 โดย Gottfried Ungerboeck เพื่อรองรับการใช้งานร่วมกับระบบสื่อสารที่มีการมอดูเลตแบบหลายระดับ หลักการทำงานของภาคเข้ารหัสและถอดรหัสแบบ TCM เป็นการนำวงจรเข้ารหัสคอนโวลูชันมาพัฒนา โดยที่ใช้งานร่วมกับภาคมอดูเลตสัญญาณข้อมูลที่ถูกตรวจจับได้คำนวณโดยตรงเพื่อจะลดผลกระทบของความผิดพลาดที่เกิดจากภาคตรวจจับข้อมูลซึ่งถูกใช้สำหรับแปลงสัญญาณที่ถูกตรวจจับได้ให้อยู่ในรูปของข้อมูลดิจิทัล เป็นการออกแบบลักษณะการทำงานของวงจรเข้ารหัสคอนโวลูชัน และมีรูปแบบ Mapping สัญญาณให้มีลักษณะเหมาะสม เนื่องจากข้อมูลต่าง ๆ ได้จากกระบวนการทำงานของภาคเข้ารหัส TCM อยู่ในรูปสัญญาณข้อมูล ดังนั้นในการออกแบบวงจรเข้ารหัสจึงต้องมีการใช้งานตัวแปร ต่าง ๆ ในรูปแบบสัญญาณข้อมูลด้วย เพื่อให้มีประสิทธิภาพในการป้องกันความผิดพลาดของข้อมูลที่เหมาะสม ได้มีการออกแบบลักษณะของสัญญาณที่ได้ จากกระบวนการทำงานของภาคเข้ารหัสแบบ TCM ในแต่ละกรณีให้มีลักษณะของสัญญาณที่แตกต่างกันมากที่สุดเท่าที่เป็นไปได้ โดยจะมีการใช้ค่า Minimum Euclidean Distance สำหรับกระบวนการเปรียบเทียบเป็นค่าที่แสดงถึงความแตกต่างที่มีน้อยที่สุดของ Signal Sequence ใด ๆ ในทุกกรณีที่สามารถถูกส่งออกมาจากภาคเข้ารหัส TCM โดยการออกแบบวงจรให้มีคุณสมบัติที่เหมาะสม โดยที่นำหลักการสำหรับแบ่งกลุ่มของสัญญาณที่เรียกว่า Mapping by Set Partitioning มาใช้ในการออกแบบ

การทำงานของภาคถอดรหัส TCM มีการนำหลักการของวิธีการถอดรหัสแบบ Viterbi มาประยุกต์ใช้งาน โดยจะมีการนำสัญญาณข้อมูลที่ถูกตรวจจับได้มาเปรียบเทียบกับ Trellis Diagram ที่แสดงถึงคุณสมบัติของวงจรเข้ารหัสที่ใช้งานเพื่อหาเส้นทางใด ๆ ใน Trellis Diagram ที่มีค่าเรียกว่า Euclidean Distance โดยที่เส้นทางเมื่อมีการเปรียบเทียบกับสัญญาณที่ถูกตรวจจับได้น้อยที่สุดโดยใช้วิธีการแบบ Viterbi Algorithm ในการคำนวณและส่งข้อมูลที่แสดงถึงเส้นทางที่ถูกเลือกให้นำไปเป็นผลลัพธ์ภาคถอดรหัส TCM

2.3 รหัสเทอร์โบ (Turbo Codes)

รหัสเทอร์โบได้ถูกค้นพบในปี พ.ศ.2536 โดย Alian Glavieux Claude Berrou และ ปัญญา รัฐิมัชฌิมา เป็นรหัสที่มีการนำวงจรเข้ารหัสแบบ Recursive Systematic Convolution Codes (RSC) จากรูปที่ 2.8 ตัวเข้ารหัสเทอร์โบที่ประกอบด้วยตัวเข้ารหัสย่อยที่เหมือนกันสองตัว ถูกนำมาต่อขนานกัน (Parallel Concatenation) ผ่านตัววางสลับบิต (Interleaver) โดยตัวเข้ารหัสย่อยแต่ละตัวจะเป็นตัวเข้ารหัสซิสเต็มมาติกคอนโวลูชันแบบรีเคอร์ซีฟ (Recursive Systematic Convolution Encoder :RSC) ดังนั้นรหัสเทอร์โบมีชื่อเรียกว่า รหัสคอนโวลูชันที่ต่อแบบขนาน (Parallel Concatenated Convolution Codes :PCCC) ในการเข้ารหัสซึ่งลำดับข่าวสารชุดเดียวกันจะถูกเข้ารหัสสองครั้งด้วยลำดับที่ต่างกันเนื่องจากตัววางสลับบิตโดยที่เวลา k บิตข่าวสาร d_k จะถูกเข้ารหัสด้วย RSC1 ได้บิตพาริตี X_{p1k} ส่วนบิตข่าวสาร \tilde{d}_k (d_k ที่ถูกวางสลับตำแหน่ง) จะถูกเข้ารหัสด้วย RSC2 ได้บิตพาริตี X_{p2k} ดังนั้นที่เวลา k คำรหัสที่ได้เท่ากับ (d_k, X_{p1k}, X_{p2k})



รูปที่ 2.8 ตัวเข้ารหัสเทอร์โบ

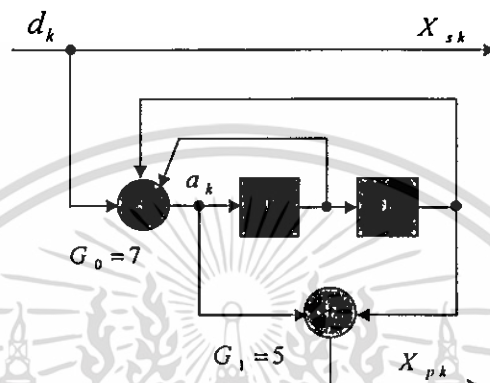
รหัสเทอร์โบนี้มีอัตราการเข้ารหัสเท่ากับ $1/3$ และเมื่อต้องการเพิ่มอัตราการเข้ารหัสสามารถทำได้ด้วยการ (Puncturing) โดยลบบิตพาริตีบางบิตออกก่อนส่งดังนั้นคือที่เวลา k คำรหัส (Codeword) ที่ได้คือ (d_k, X_{p1k}) หรือ (d_k, X_{p2k}) สลับกันไปและรหัสเทอร์โบมีอัตราการเข้ารหัสเพิ่มขึ้นเป็น $1/2$

2.3.1 ตัวเข้ารหัส (Recursive Systematic Convolution Encoder :RSC)

ตัวเข้ารหัส RSC คือเข้ารหัสซิสเต็มมาติกคอนโวลูชันที่มีการป้อนกลับ ในรหัสซิสเต็มมาติก ลำดับข่าวสารเป็นส่วนหนึ่งของคำรหัสสำหรับรหัส RSC อัตราการเข้ารหัสเท่ากับ $1/2$ และเมตริกซ์ตัวกำเนิด (Generator Matrix) มีค่าดังนี้

$$G(D) = \begin{bmatrix} 1 & \frac{g_1(D)}{g_0(D)} \end{bmatrix} \quad (2.8)$$

เมื่อ $g_0(D)$ และ $g_1(D)$ เป็นพหุนามตัวกำเนิดดีกรี ν ที่แสดงถึงการป้อนกลับ (Feedback) และค่าการป้อนไปหน้า (Feedforward) ตามลำดับ โดย ν แสดงถึงหน่วยความจำ (Memory) ของตัวเข้ารหัสโครงสร้างตัวเข้ารหัส RSC มีอัตราการเข้ารหัสเท่ากับ $1/2$ และ $\nu = 2$ สามารถแสดงดังรูปที่ 2.9



รูปที่ 2.9 ตัวเข้ารหัส RSC มีอัตราการเข้ารหัสเท่ากับ $1/2$ และหน่วยความจำ $\nu = 2$

จากรูปที่ได้พหุนามตัวกำเนิดดังนี้

$$g_0(D) = 1 + D + D^2 \quad (2.9)$$

$$g_1(D) = 1 + D^2 \quad (2.10)$$

$$G(D) = \begin{bmatrix} 1 & \frac{1 + D^2}{1 + D + D^2} \end{bmatrix} \quad (2.11)$$

โดยทั่วไปสามารถอธิบายตัวเข้ารหัส RSC ด้วยตัวกำเนิด G_0 และ G_1 ซึ่งแสดงถึงการเลื่อนระหว่างรีจิสเตอร์ (Shift Register) กับตัวบวกมอดดูโลสอง (Modulo 2 Adder) แทนการอธิบายด้วยพหุนามตัวกำเนิด $g_0(D)$ และ $g_1(D)$ ตามลำดับ โดยตัวกำเนิดเขียนในรูปเลขฐานสองและฐานแปด

$$G_0 = \{g_{00} \ g_{01} \ g_{02}\} = \{1 \ 1 \ 1\} = 7 \quad (2.12)$$

$$G_1 = \{g_{10} \ g_{11} \ g_{12}\} = \{1 \ 0 \ 1\} = 5 \quad (2.13)$$

จากตัวเข้ารหัส RSC สร้างคำรหัส (X_{sk}, X_{pk}) ที่เวลา k โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_k = d_k \oplus \sum_{i=1}^v g_{oi} a_{k-i} \quad (2.14)$$

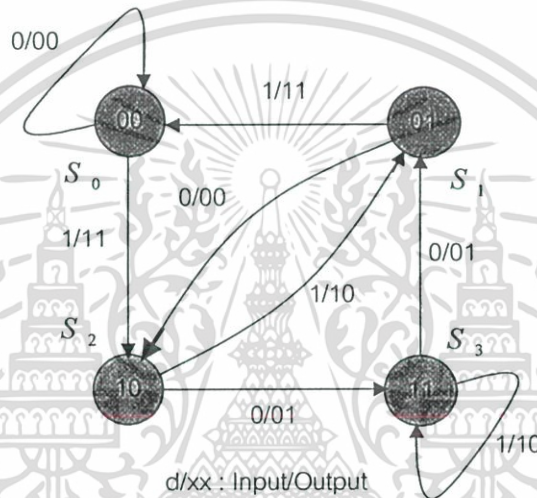
$$X_{sk} = d_k = \sum_{i=0}^v g_{oi} a_{k-i} \quad (2.15)$$

$$X_{pk} = \sum_{i=0}^v g_{1i} a_{k-i} \quad (2.16)$$

เมื่อ d_k คือ บิตข้อมูลข่าวสาร

X_{sk} คือ บิต(Systematic)

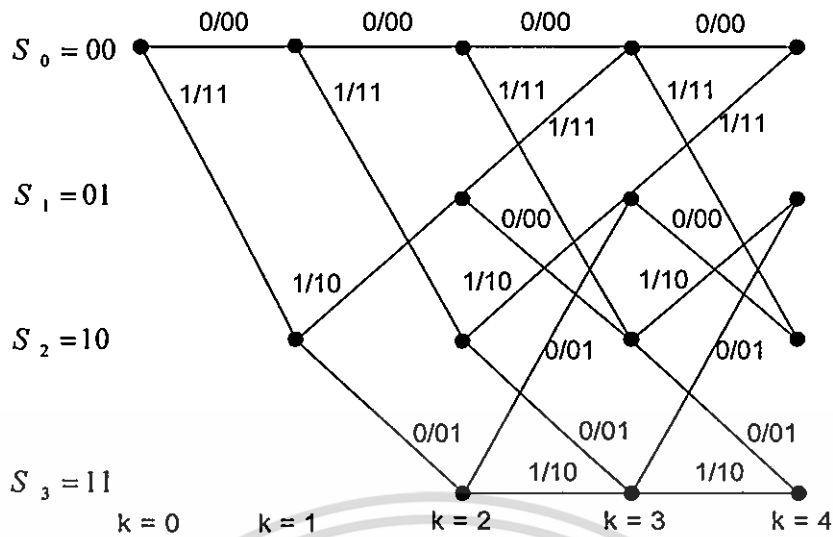
X_{pk} คือ บิต(Parity)



รูปที่ 2.10 แผนภาพการเปลี่ยนสถานะรหัส RSC มีอัตราการใช้รหัสเท่ากับ 1/2 และ $v = 2$

เนื่องจากกระบวนการเข้ารหัสนั้นเป็นกระบวนการมาร์คอฟสถานะจำกัดที่ไม่ต่อเนื่องทางเวลา (Discrete-Time Finite-State Markov Process) ดังนั้นสามารถอธิบายได้โดยแผนภาพการเปลี่ยนสถานะ (State Transition Diagram) และแผนภาพเทรลลิส (Trellis Diagram) ที่มีจำนวนสถานะเท่ากับ 2^v สำหรับตัวเข้ารหัส RSC ในรูปที่ 2.10 มีแผนภาพการเปลี่ยนสถานะแผนภาพเทรลลิสดังรูปที่ 2.11 ตามลำดับ

เมื่อมีการออกแบบตัวเข้ารหัส RSC ที่มีความเหมาะสมรหัสเทอร์โบซึ่งมีระยะฟรีประสิทธิภาพ (Free Effective Distance) ให้มีค่ามากที่สุดมีผลทำให้ได้สมรรถนะที่ดี ดังนั้นตัวเข้ารหัสย่อยของเทอร์โบจึงต้องเป็นรหัส RSC และนอกจากนี้โดยที่มีข้อดีของรหัส RSC เมื่อเทียบกับรหัส NSC คือ มีการเข้ารหัสที่ง่ายกว่า โดยมีการใช้ฮาร์ดแวร์น้อยกว่า และไม่มีความจำเป็นต้องใช้วงจรย้อนกลับเพื่อกู้ข่าวสารคืนจากคำรหัส



รูปที่ 2.11 แผนภาพ Trellis ของรหัส RSC มีอัตราเข้ารหัสเท่ากับ $1/2$ และ $v = 2$

2.3.2 การวางสลับบิต (Interleaving)

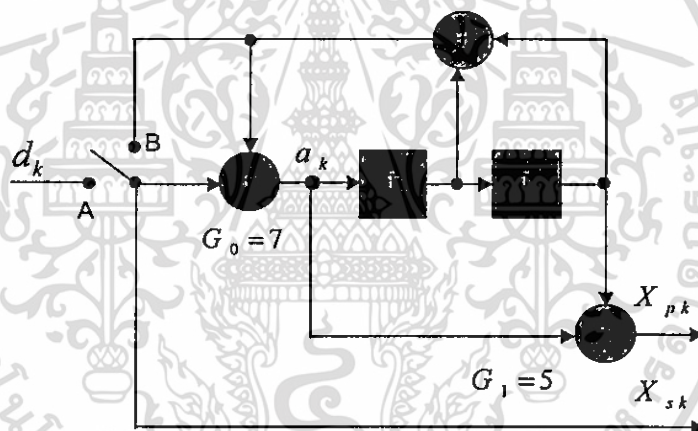
ในการเข้ารหัสเทอร์โบ การวางสลับบิตเป็นกระบวนการจัดการข่าวสารใหม่ก่อนส่งให้กับตัวเข้ารหัส RSC2 กระบวนการย้อนกลับของการวางสลับบิตเรียกว่า "ดีอินเตอร์ลีฟ" (Deinterleave) เป็นการจัดลำดับข้อมูลข่าวสารให้กลับคืนดังเดิม ในทางปฏิบัติการวางสลับบิตเป็นเทคนิคที่ช่วยความสามารถในการแก้ไขความผิดพลาดในการเข้ารหัส มีบทบาทสำคัญในการทำให้การเข้ารหัสเทอร์โบมีสมรรถนะที่ดี การสร้างรหัสบล็อกขนาดยาว ๆ จากรหัสคอนโวลูชันที่มีหน่วยความจำน้อย ๆ โดยใช้การวางสลับตำแหน่งทำให้ได้คุณสมบัติระยะห่าง (Distance) ที่ดีซึ่งสามารถถอดรหัสส่วนซ้ำได้อย่างมีประสิทธิภาพ การวางสลับบิตจะกำจัดลำดับอินพุตมีน้ำหนัก (Weight) ต่ำ ๆ ให้หมดไปทำให้ระยะห่างของรหัสเพิ่มขึ้นหรือทำให้ระยะห่างของลำดับข่าวสารลดลง กล่าวคือ การวางสลับบิตซึ่งจะกระจายความผิดพลาดแบบเบิสต์ (Burst Error) ออกโดยการส่งข้อมูลถูกสลับตำแหน่งไปตัวเข้ารหัส (RSC2) และที่ตัวถอดรหัส โดยการวางสลับบิตจะทำการลดสหสัมพันธ์ (Correlation) ระหว่างอินพุตของตัวถอดรหัส SISO ทั้งสองเพื่อให้การถอดรหัสส่วนซ้ำมีการแลกเปลี่ยนข่าวสารที่ไม่มีสหสัมพันธ์กันระหว่างตัวถอดรหัส SISO ทั้งสองสามารถทำงานได้ กล่าวคือ หลังจากที่ตัวถอดรหัส SISO1 ทำการแก้ไขความผิดพลาดบางส่วนแล้วความผิดพลาดที่เหลือจะถูกกระจายออกโดยตัววางสลับบิตเพื่อให้ SISO2 สามารถทำการแก้ไขได้ เมื่อจำนวนรอบในการวนซ้ำ การถอดรหัสเพิ่มขึ้น อัตราความผิดพลาดจะเข้าใกล้กรณีของตัวถอดรหัสแบบความน่าจะเป็นจริงสูงสุด (Maximum Likelihood Decoder) โดยปกติแล้วเมื่อขนาดตัววางสลับบิตเพิ่มขึ้นจะเป็นผลดีต่อคุณสมบัติของรหัสและสมรรถนะของการถอดรหัสส่วนซ้ำ ทำให้สมรรถนะรหัสเทอร์โบดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัววางสลับบิตแบ่งตามโครงสร้างได้ 4 ชนิด ได้แก่ ตัววางสลับบิตบล็อก (Block Interleaver) ตัววางสลับบิตคอนโวลูชัน (Convolution Interleaver) ตัววางสลับบิตสุ่ม (Random Interleaver) และตัววางสลับบิตแบบเหมาะสมรหัส (Code Match Interleaver) แต่ละชนิดสามารถแยกย่อยได้หลายชนิด สามารถศึกษาได้ [1]

2.3.3 การสิ้นสุดเทรลลิส (Trellis Termination)

ตัวเข้ารหัสเทอร์โบที่มีอินพุตเป็นกลุ่มบิตข่าวสารสามารถ พิจารณาได้ว่าเป็นตัวเข้ารหัสบล็อก การสิ้นสุดเทรลลิส (Trellis Termination) คือการทำให้ตัวเข้ารหัสมีสถานะเป็นศูนย์ทั้งหมดเมื่อสิ้นสุดเข้ารหัสแต่ละบล็อกข่าวสารเพื่อให้แน่ใจว่าสถานะเริ่มต้นสำหรับบล็อกต่อไปเป็นศูนย์ทั้งหมดเนื่องจากตัวเข้ารหัสเทอร์โบมีตัวเข้ารหัสย่อยเป็นแบบรีเคอร์ซีฟ (Recursive) ไม่สามารถทำการสิ้นสุดเทรลลิสโดยการส่งบิตหาง (Tail Bits) ที่เป็นศูนย์จำนวน ν บิต เช่นเดียวกับกับกรณีของตัวเข้ารหัสแบบไม่เป็นรีเคอร์ซีฟได้ ดังนั้นสำหรับตัวเข้ารหัสแบบรีเคอร์ซีฟ บิตหางจึงไม่เป็นศูนย์เสมอไป แต่ขึ้นอยู่กับสถานะตัวเข้ารหัสหลังจากทำการเข้ารหัสบิตข่าวสารจำนวน N บิต



รูปที่ 2.12 แผนภาพ Trellis Termination

วิธีการสิ้นสุดเทรลลิสอย่างง่ายสำหรับตัวเข้ารหัสแบบรีเคอร์ซีฟ [11] แสดงได้ดังรูปที่ 2.11 เมื่อทำการส่งข่าวสาร N บิต สวิตช์ที่ตำแหน่ง A และเมื่อทำการส่งบิตหาง ν บิต สวิตช์จะอยู่ที่ตำแหน่ง B ทำให้ตัวเข้ารหัสมีสถานะเป็นศูนย์ทั้งหมด คือการสิ้นสุดเทรลลิสต้องทำให้อินพุต a_k ที่เข้าสู่รีจิสเตอร์เลื่อนเป็นศูนย์ เพื่อให้รีจิสเตอร์ทั้งหมดเป็นศูนย์หลังการเลื่อน ν ครั้ง

เมื่อรหัสเทอร์โบใช้การวางสลับบิตแบบสุ่มและบล็อก วิธีการดังกล่าวข้างต้นไม่สามารถทำการสิ้นสุดเทรลลิสของตัวเข้ารหัสย่อยทั้งสองพร้อมกันได้เมื่อส่งบิตหางเพียง ν บิต เนื่องจากตัวเข้ารหัสย่อยทั้งสองทำงานแบบรีเคอร์ซีฟ ซึ่งโดยทั่วไปแล้วมีเพียงตัวรหัสย่อยตัวแรกเท่านั้นที่มีสถานะเป็นศูนย์ทั้งหมด การไม่ทราบสถานะสุดท้ายของตัวเข้ารหัสย่อยตัวที่สองทำให้สมรรถนะ

รหัสเทอร์โบลดลงเพียงเล็กน้อยเมื่อการวางสลับบิตมีขนาดใหญ่ขึ้น (N มาก ๆ) สำหรับรหัสเทอร์โบที่มีอัตราเข้ารหัส $1/3$ และทำการสิ้นสุดเทรลลิส สามารถพิจารณาได้ว่ารหัสชิสเตมาติกเชิงเส้น สำหรับวิทยานิพนธ์ฉบับนี้ใช้การวางสลับบิตทั้งหมดมีเพียงตัวเข้ารหัสย่อยตัวแรกเท่านั้นที่มีการสิ้นสุดเทรลลิส สำหรับโครงสร้างการวางสลับบิตแบบพิเศษที่สามารถทำให้ตัวเข้ารหัสย่อยทั้งสองตัวมีสถานะเป็นศูนย์พร้อมกันซึ่งจะมีการศึกษาใน [1]

2.3.4 การ Puncturing

โดยทั่วไปรหัสย่อยทั้งสองและอัตราของรหัสแต่ละตัวไม่จำเป็นต้องเหมือนกัน โดยที่อัตราของรหัสย่อยทั้งสองแทนค่าด้วย R_1 และ R_2 ตามลำดับ ซึ่งอัตราทั้งหมดของรหัสเทอร์โบที่ยังไม่มีการทำ Puncturing : R หาได้จาก

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} - 1 \quad (2.17)$$

ยกตัวอย่างเช่น

$$R_1 = R_2 = 1/2 \quad R = 1/3$$

$$R_1 = 1/3 \quad R_2 = 1/4 \quad R = 1/6$$

$$R_1 = R_2 = 3/4 \quad R = 3/5$$

$$R_1 = R_2 = 4/5 \quad R = 2/3$$

การ Puncturing เป็นกระบวนการกำจัดสัญลักษณ์และตำแหน่งบางตัวออกจากคำรหัส เพื่อให้ความยาวของคำรหัสลดลง และอัตราการเข้ารหัสเพิ่มขึ้น รหัสเทอร์โบ [2] มีการ Puncturing ครั้งหนึ่งของชุดบิตตัวเข้ารหัส RSC แต่ละตัวการ Puncturing ครั้งหนึ่งของชุดบิตตัวเข้ารหัส RSC ที่แต่ละตัวนั้นหมายถึงการส่งบิตตัวเข้ารหัส RSC ในครั้งหนึ่ง ดังนั้นอัตรารหัสทั้งหมดเท่ากับ $1/2$ นอกจากการเลือกใช้เข้ารหัสแบบต่าง ๆ ยังมีผลกระทบต่อ การ Puncturing

เมื่อมีการ Puncture จะมีการลบบางบิตออกจากคำรหัส (d_k, X_{p1k}, X_{p2k}) ตามรูปแบบกำหนดโดยเมตริกซ์ Puncture เช่น รหัสเทอร์โบที่มีอัตราเข้ารหัส $1/2$ จะได้มาจากการ Puncture P รหัสเทอร์โบที่มีอัตราเข้ารหัส $1/3$ สามารถแสดงเป็นรูปแบบเมตริกซ์ดังสมการที่ (2.18) เมื่อคาบเวลาการ Puncture เท่ากับ 2 จากเมตริกซ์ บิตพาริตีจากตัวเข้ารหัสย่อยทั้งสองจะถูกลบสลับกันดังนั้น ที่เวลา k คำรหัสเทอร์โบที่มีการ Puncture ประกอบด้วยข่าวสารหนึ่งบิตตามด้วยพาริตีหนึ่งบิต ซึ่งบิตพาริตีนี้ได้มาจากตัวเข้ารหัสย่อยตัวแรกและจากตัวที่สองสลับกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.18)$$

2.4 การถอดรหัสเทอร์โบ (Turbo Decoder)

ตัวถอดรหัสเทอร์โบประกอบด้วยตัวถอดรหัสย่อยที่เหมือนกันสองตัวถูกนำมาต่ออนุกรมกัน (Serial concatenation) ผ่านตัววางสลับบิตที่เหมือนกันกับตัวเข้ารหัส ตัวถอดรหัสย่อยซึ่งแต่ละตัวจะเป็นตัวถอดรหัสแบบ (SISO : Soft -Input / Soft- Output) 2^o สถานะ ซึ่งเหมาะสมกับการถอดรหัสวนซ้ำ เนื่องจากตัวถอดรหัสนี้รับอินพุตเป็นข่าวสาร (A Priori Information) และให้เอาต์พุตเป็นข้อมูล (A Posteriori Information) ในการถอดรหัสเทอร์โบ จะใช้อัลกอริทึมในการถอดรหัสแบบต่าง ๆ ที่มีพื้นฐานทฤษฎี อัลกอริทึมเหล่านี้เหมาะสำหรับการประมาณค่าลำดับสถานะของกระบวนการมาร์คอฟสถานะจำกัดที่ไม่ต่อเนื่องทางเวลาเมื่อพิจารณาพร้อมกับสัญญาณรบกวน ในการถอดรหัสลำดับรหัสที่ถูกรบกวนจะใช้อัลกอริทึมแบบ (Maximum A Posteriori Probability : MAP) ในการประเมินข่าวสารที่น่าเป็นจริงที่สุดกับลำดับรหัสที่ส่งมา

2.4.1 อัลกอริทึมแบบ BCJR

อัลกอริทึมแบบ BCJR [12] หรือเรียกว่า MAP เป็นส่วนประกอบหลักใช้ในระบบการตรวจจับและการถอดรหัสวนซ้ำหลาย ๆ ระบบอัลกอริทึมแบบ MAP เหมาะสมกับการประมาณค่าสถานะหรือเอาต์พุตของกระบวนการมาร์คอฟเมื่อพิจารณานบนช่องสัญญาณแบบเกาส์เซียน โดยที่อัลกอริทึมแบบนี้ซึ่งจะมีการสร้างลำดับความน่าจะเป็นแบบ (A Posteriori Probability : APP) $P_r(d_k = i | observation)$ ของบิตข้อมูล d_k เมื่อ $i = 0,1$ และ *observation* คือลำดับที่รับได้ทั้งหมด แต่เนื่องจากการแทนค่าตัวเลขความน่าจะเป็นฟังก์ชันที่ไม่เป็นเชิงเส้น และการที่มีทั้งการคูณและการบวกค่าเหล่านี้ อาจทำให้อัลกอริทึมนี้ยุ่งยาก จึงได้มีการปรับปรุงอัลกอริทึมในรูปแบบต่าง ๆ เพื่อให้เหมาะสมกับการใช้งานถอดรหัสเทอร์โบ เช่น อัลกอริทึม (Log Map) ซึ่งได้รับการพัฒนาจากอัลกอริทึม MAP โดยได้มีการเปลี่ยนโครงสร้างทางคณิตศาสตร์เพื่อช่วยให้ง่ายขึ้นในการประมวลผล

2.4.2 การถอดรหัสวนซ้ำของรหัสเทอร์โบ

2.4.2.1 พิสูจน์ของความเป็นจริงแบบล็อก (LLR)

พิชคณิตของความเป็นจริงแบบล็อก $L(u_k)$ อัตราส่วนความเป็นจริงของ (Log-Likelihood Ratio : LLR) ตัวแปรสุ่มฐานสอง u_k นิยามได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$L(u_k) = \ln \frac{P_r(u_k = +1)}{P_r(u_k = -1)} \quad (2.19)$$

เมื่อ $u_k = 2d_k - 1$ คือบิตข่าวสารที่เวลา k
เนื่องจาก

$$P_r(u_k = -1) = 1 - P_r(u_k = +1) \quad (2.20)$$

และ

$$L(u_k) = \ln \frac{P_r(u_k = +1)}{1 - P_r(u_k = +1)} \quad (2.21)$$

แล้ว

$$e^{L(u_k)} = \frac{P_r(u_k = +1)}{1 - P_r(u_k = +1)} \quad (2.22)$$

ดังนั้น

$$P_r(u_k = +1) = \frac{e^{L(u_k)}}{1 + e^{L(u_k)}} = \frac{1}{1 + e^{-L(u_k)}} = \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{L(u_k)/2} \quad (2.23)$$

และ

$$P_r(u_k = -1) = 1 - \frac{1}{1 + e^{-L(u_k)}} = \frac{e^{-L(u_k)}}{1 + e^{-L(u_k)}} = \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{-L(u_k)/2} \quad (2.24)$$

สมการ (2.23) และ (2.24) สามารถเขียนอยู่ในรูปทั่วไป

$$P_r(u_k = \pm 1) = \frac{e^{\pm L(u_k)}}{1 + e^{\pm L(u_k)}} = \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{\pm L(u_k)/2} = A_k \cdot e^{u_k \cdot L(u_k)/2} \quad (2.25)$$

เมื่อ $A_k \cdot e^{u_k \cdot L(u_k)/2}$ คือ แฟกเตอร์ร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตัวแปรสุ่มฐานสอง u_k ถูกกำหนดเงื่อนไขตัวแปรสุ่ม r_k แล้ว LLR แบบมีเงื่อนไข $L(u_k | r_k)$ ที่สามารถหาได้จาก

$$\begin{aligned}
 L(u_k | r_k) &= \ln \frac{P_r(u_k = +1 | r_k)}{P_r(u_k = -1 | r_k)} \\
 &= \ln \frac{p(r_k | u_k = +1) \cdot P_r(u_k = +1)}{p(r_k | u_k = -1) \cdot P_r(u_k = -1)} \\
 &= \ln \frac{p(r_k | u_k = +1)}{p(r_k | u_k = -1)} + \ln \frac{P_r(u_k = +1)}{P_r(u_k = -1)} \quad (2.26) \\
 &= L(r_k | u_k) + L(u_k)
 \end{aligned}$$

2.4.2.2 เอาดัตหลุดแบบซอฟต์แวร์ของช่องสัญญาณ (SISO)

พิจารณาการส่งสัญญาณผ่านช่องสัญญาณ AWGN ที่มีการจางหาย [1] เนื่องจาก

$$p(r_k | u_k, a_k) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{E_s}{N_0} (r_k - u_k \cdot a_k)^2\right) \quad (2.27)$$

ดังนั้น

$$\begin{aligned}
 L(r_k | u_k, a_k) &= \ln \frac{p(r_k | u_k = +1, a_k)}{p(r_k | u_k = -1, a_k)} \\
 &= \ln \frac{\exp\left(-\frac{E_s}{N_0} (r_k - a_k)^2\right)}{\exp\left(-\frac{E_s}{N_0} (r_k + a_k)^2\right)} \\
 &= -\frac{E_s}{N_0} \left((r_k - a_k)^2 - (r_k + a_k)^2 \right) \\
 &= (4 \cdot a_k \cdot \frac{E_s}{N_0}) \cdot r_k = L_C \cdot r_k \quad (2.28)
 \end{aligned}$$

โดยที่

$$\begin{aligned}
 L_C &= 4 \cdot a_k \cdot \frac{E_s}{N_0} \\
 &= 4 \cdot a_k \cdot R \cdot \frac{E_b}{N_0} \quad (2.29)
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ L_c คือ ค่าความเชื่อถือได้ของช่องสัญญาณ

a_k คือ การจางหายที่เวลา k (สำหรับช่องสัญญาณ AWGN ที่ไม่มีการจางหาย $a_k = 1$)

R คือ อัตราการเข้ารหัส

E_b คือ พลังงานที่ได้รับต่อหนึ่งบิตข้อมูลข่าวสาร (J/Bit)

$E_s = R \cdot E_b$ คือ พลังงานที่ได้รับหนึ่งบิตสัญลักษณ์รหัส (J/Symbol)

$N_0 = 2\sigma^2$ คือ ค่าความหนาแน่นเชิงสเปกตรัม (Power Spectral Density : PSD) ของค่าสัญญาณรบกวนบนแกนความถี่ข้างเดียว (W/Hz)

σ^2 คือ ค่าความแปรปรวนของสัญญาณรบกวน

ความสัมพันธ์ระหว่าง σ^2 และ E_b/N_0 เป็นดังนี้

$$\sigma^2 = \frac{E_s}{2 \cdot R \cdot (E_b / N_0)} \quad (2.30)$$

แทนค่า $L(r_k | u_k)$ ในสมการ (2.26) ด้วย $L(r_k | u_k, a_k)$ จากสมการ (2.28) ได้

$$\begin{aligned} L(u_k | r_k) &= L(r_k | u_k, a_k) + L(u_k) \\ &= L_c \cdot r_k + L(u_k) \end{aligned} \quad (2.31)$$

2.4.2.3 หลักการถอดรหัสแบบวนซ้ำ (Iteration Decoding)

พิจารณาตัวถอดรหัสแบบ SISO ดังรูปที่ 2.13



รูปที่ 2.13 ตัวถอดรหัสแบบ SISO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุตของตัวถอดรหัส SISO ที่ใช้อัลกอริทึมแบบแมป (Map) ที่มีการกำหนดให้ LLR แบบ (A Posteriori) ซึ่งเป็นค่าอัลกอริทึมของอัตราส่วนความน่าจะเป็นที่บิตข่าวสาร d_k จะเท่ากับ 1 หรือ 0 เมื่อได้รับค่า *observation* (r)

$$L(\hat{d}_k) = L(d_k | r) = \ln \frac{P_r(d_k = 1 | r)}{P_r(d_k = 0 | r)} \quad (2.32)$$

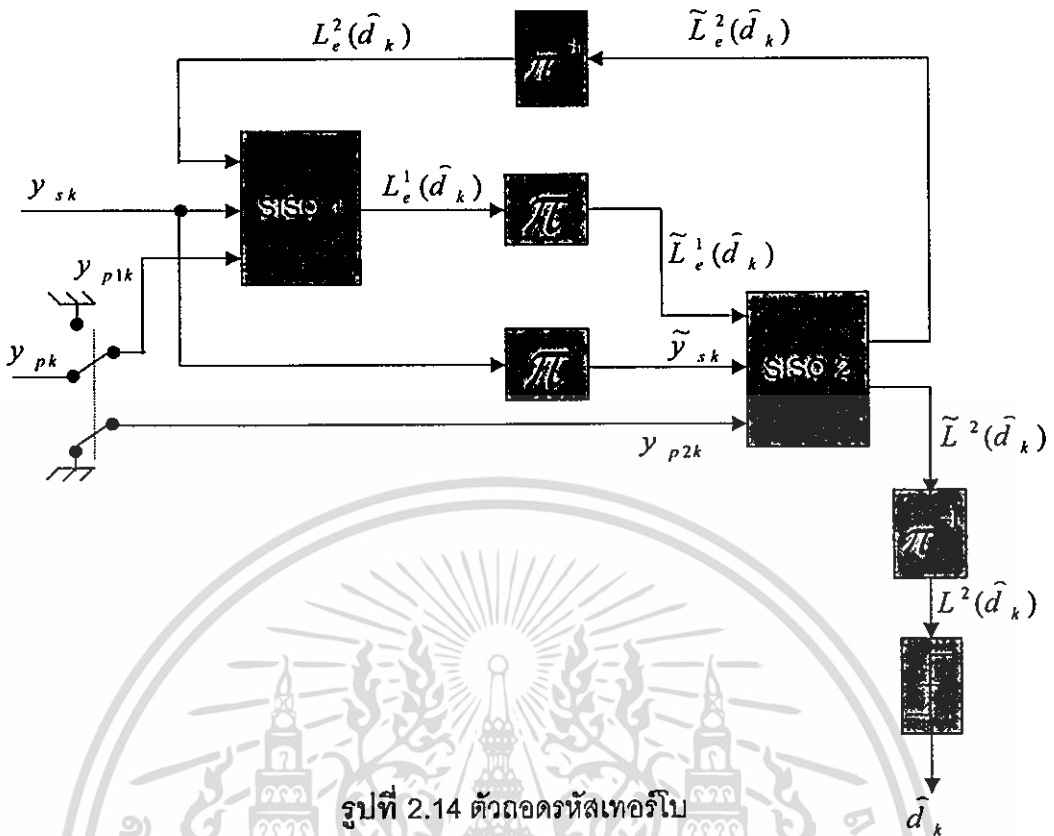
ตัวถอดรหัสนี้จะใช้ค่า (A Priori) $L(d_k)$ และค่าของสัญญาณ $L_C \cdot r_k$ เพื่อเอาต์พุตเป็นแบบซอฟต์แวร์ $L(\hat{d}_k)$ และข่าวสาร (Extrinsic Information) $L_e(\hat{d}_k)$ สำหรับรหัส (Systematic) เอาต์พุตแบบซอฟต์แวร์ $L(\hat{d}_k)$ หาได้จาก

$$L(\hat{d}_k) = L_C \cdot r_k + L(d_k) + L_e(\hat{d}_k) \quad (2.33)$$

เมื่อพิจารณาตัวถอดรหัสเทอร์โบอัตราเข้ารหัส 1/2 ดังรูปที่ 2.14 ซึ่งเป็นกระบวนการถอดรหัสวนซ้ำที่ประกอบด้วยตัวถอดรหัส SISO สองตัว คำรหัสที่ได้ $r_k = (r_{sk}, r_{pk})$ ซึ่งเพื่อความสะดวกในการอธิบายถึงจึงกำหนดให้ $y_k = L_C \cdot r_k = L_C \cdot (r_{sk}, r_{pk}) = (y_{sk}, y_{pk})$ ในการถอดรหัสวนซ้ำในรอบแรก ตัวถอดรหัส SISO1 จะคำนวณข่าวสาร (Extrinsic Information) ได้

$$L_e^1(\hat{d}_k) = L^1(\hat{d}_k) - (y_k + L(d_k)) \quad (2.34)$$

เนื่องจากในตอนเริ่มต้นให้มีบิตข้อมูลข่าวสารอาจเป็น 1 หรือ 0 ด้วยค่าน่าจะเป็นที่เท่ากัน ดังนั้นในการถอดรหัสวนซ้ำรอบแรกจึงกำหนดให้ $L_e^2(\hat{d}_k) = L(d_k) = 0$



รูปที่ 2.14 ตัวถอดรหัสเทอร์โบ

ข่าวสาร (Extrinsic Information) จาก SISO1 $L_e^1(\hat{d}_k)$ จะถูกวางสลับบิตเป็น $\tilde{L}_e^1(\hat{d}_k)$ เพื่อใช้เป็นค่า (A Priori) สำหรับ SISO2 แทนที่ $L(d_k)$ ดังนั้นข่าวสาร (Extrinsic Information) ที่คำนวณได้มาจากตัวถอดรหัส SISO2 มีค่า

$$\tilde{L}_e^2(\hat{d}_k) = \tilde{L}^2(\hat{d}_k) - (\tilde{y}_k + \tilde{L}_e^1(\hat{d}_k)) \quad (2.35)$$

จากนั้นในการถอดรหัสส่วนซ้ำรอบที่สอง ข่าวสาร (Extrinsic Information) จาก SISO2 $\tilde{L}_e^2(\hat{d}_k)$ จะถูกวางสลับบิตเป็น $L_e^2(\hat{d}_k)$ เพื่อใช้เป็นค่า (A Priori) สำหรับ SISO1 แทนที่ $L(d_k)$ เช่นเดียวกันกระบวนการถอดรหัสจะทำการวนซ้ำเช่นนี้ไปเรื่อย ๆ จนครบตามจำนวนรอบที่ได้กำหนดไว้หรือจนกว่าเอาต์พุตแบบซอฟต์ $\tilde{L}^2(\hat{d}_k)$ จะมีเสถียรภาพและมีการเปลี่ยนแปลงเพียงเล็กน้อยเมื่อมีจำนวนรอบเพิ่มขึ้น ในการถอดรหัสส่วนซ้ำรอบสุดท้ายตัวถอดรหัส SISO2 จะคำนวณเอาต์พุตแบบซอฟต์สามารถคำนวณได้

$$\tilde{L}^2(\hat{d}_k) = \tilde{y}_k + \tilde{L}_e^1(\hat{d}_k) + \tilde{L}_e^2(\hat{d}_k) \quad (2.36)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุตแบบซอฟต์ $\tilde{L}^2(\hat{d}_k)$ จะถูกวางสลับบิตเป็น $L^2(\hat{d}_k)$ เพื่อใช้ตัดสินใจ (Hard Decision) บิตข้อมูลข่าวสาร \hat{d}_k ตามเงื่อนไข

$$\hat{d}_k = \begin{cases} 1, & L^2(\hat{d}_k) \geq 0 \\ 0, & L^2(\hat{d}_k) < 0 \end{cases} \quad (2.37)$$

2.4.3 อัลกอริทึมที่ใช้ในการถอดรหัส

อัลกอริทึมแบบความน่าจะเป็นจริงสูงสุดเช่นอัลกอริทึมแบบไวเทอร์บี (Viterbi Algorithm) จะทำการหาลำดับความน่าจะเป็นของข้อมูลข่าวสารที่น่าเป็นจริงมากที่สุด ขณะที่อัลกอริทึมแบบแมปจะทำการหาบิตข่าวสารที่มีความน่าจะเป็นจริงมากที่สุดเมื่อมีการส่งลำดับการเข้ารหัสออกมา โดยที่บิตข้อมูลข่าวสารที่ได้จากอัลกอริทึมแบบแมปจึงไม่มีความจำเป็นต้องอยู่ในเส้นทางของเทรลลิสที่ต่อกัน

อัลกอริทึมแบบแมปเหมาะสมที่สุดสำหรับการประมาณค่าสถานะหรือเอาต์พุตของกระบวนการ การมาร์คอฟ แต่ไม่สามารถนำมาใช้งานได้ เนื่องจากปัญหาการแทนค่าตัวเลขความน่าจะเป็นฟังก์ชันที่ไม่เป็นเชิงเส้น การที่มีทั้งการคูณและการบวกจำนวนมาก จึงได้มีการปรับปรุงอัลกอริทึมนี้ในรูปแบบต่าง ๆ เพื่อให้เหมาะสมที่จะนำมาใช้งานได้ในการถอดรหัสเทอร์โบ ตัวอย่างเช่น อัลกอริทึมแบบ (Log-Map)[9] ซึ่งให้สมรรถนะที่ใกล้เคียงกันกับอัลกอริทึมแบบ (Map) แต่มีความซับซ้อนน้อยกว่า เนื่องจากว่าอัลกอริทึมแบบ (Log-Map) จะมีการพิจารณาในทางลึกลับซึ่งมีผลในการคำนวณค่าทางคณิตศาสตร์ช้ากว่าและสะดวกต่อการใช้งานมาก นอกจากนี้ยังมีอัลกอริทึมที่เหมาะสมรองลงมา (Sub Optimal) ได้แก่ อัลกอริทึมแบบไวเทอร์บีที่มีเอาต์พุตแบบซอฟต์ (Soft-Output Viterbi Algorithm : SOVA) และอัลกอริทึมแบบแมกซ์ล็อกแมป (Max-Log-Map) รายละเอียดของอัลกอริทึมทั้งหมดดังกล่าวสามารถศึกษาเพิ่มจาก [1],[5] ในสำหรับวิทยานิพนธ์นี้ใช้ อัลกอริทึมแบบ (Log-Map) ในการถอดรหัสวนซ้ำ จึงขอกล่าวเฉพาะอัลกอริทึม (Map, Log-Map)

2.4.3.1 อัลกอริทึมแบบ (Map)

กำหนดให้สถานะของตัวเข้ารหัสที่เวลา k คือ S_k ซึ่งมีค่าเป็นจำนวนเต็มตั้งแต่ 0 ถึง $2^v - 1$ และบิตข้อมูลข่าวสาร d_k จะทำให้มีการเปลี่ยนสถานะจาก $S_{k-1} = s'$ ไปเป็น $S_k = s$ โดยที่มีการหาค่าอัตราส่วนความน่าจะเป็นจริงสูงสุด ได้ดังนี้

จากสมการ (2.32) LLR จะมีค่าดังนี้

$$\begin{aligned}
 L(\hat{d}_k) &= \ln \frac{P_r(d_k = 1 | r)}{P_r(d_k = 0 | r)} \\
 &= \ln \frac{\sum_{d_k=1}^{(s',s)} P_r(S_{k-1}=s', S_k=s | r)}{\sum_{d_k=0}^{(s',s)} P_r(S_{k-1}=s', S_k=s | r)} \\
 &= \ln \frac{\sum_{d_k=1}^{(s',s)} P_r(s', s, r)}{\sum_{d_k=0}^{(s',s)} P_r(s', s, r)}
 \end{aligned} \tag{2.38}$$

โดยที่

$$\begin{aligned}
 P_r(s', s, r) &= P_r(s', s, r_{j < k}, r_k, r_{j > k}) \\
 &= P_r(r_{j > k} | s', s, r_{j < k}, r_k) \cdot P_r(s', s, r_{j < k}, r_k) \\
 &= P_r(r_{j > k} | s', s, r_{j < k}, r_k) \cdot P_r(s, r_k | s', r_{j < k}) \cdot P_r(s', r_{j < k}) \\
 &= P_r(r_{j > k} | s) \cdot P_r(s', r_k | s') \cdot P_r(s', r_{j < k}) \\
 &= \beta_k(s) \cdot \gamma_k(s', s) \cdot \alpha_{k-1}(s')
 \end{aligned} \tag{2.39}$$

ในที่นี้ $r_{j < k}$ หมายถึงลำดับของสัญลักษณ์ที่รับได้จากเวลา $j = 1$ จนถึงเวลา $j = k - 1$ และ $r_{j > k}$ หมายถึงลำดับของสัญลักษณ์ที่รับได้จากเวลา $j = k + 1$ จนถึงเวลา $j = N + v = \tau$ แทนสมการ (2.39) ในสมการ (2.38) ได้ว่า

$$L(\hat{d}_k) = \ln \frac{\sum_{d_k=1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)} \tag{2.40}$$

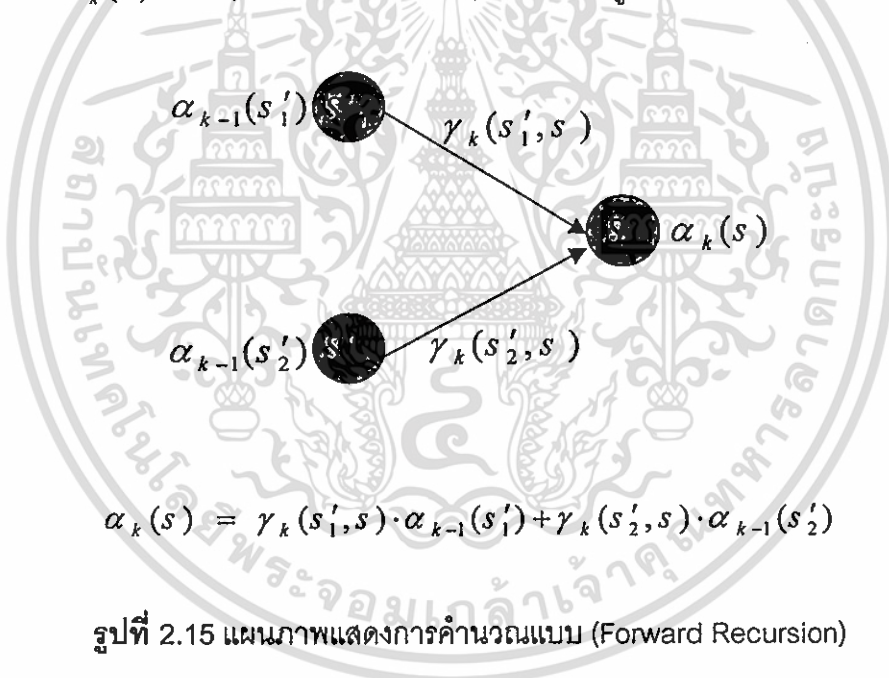
จากสมการ (2.39) ค่าความน่าจะเป็น $\alpha_k(s)$ โดยการคำนวณแบบซ้ำไปหน้า (Forward Recursion) สามารถหาความน่าจะเป็น $\alpha_k(s)$ ได้ดังสมการที่ (2.41)

$$\begin{aligned}
\alpha_k(s) &= P_r(s, r_{j < k}, r_k) \\
&= \sum_{s'} P_r(s', s, r_{j < k}, r_k) \\
&= \sum_{s'} P_r(s, r_k | s', r_{j < k}) \cdot P_r(s', r_{j < k}) \\
&= \sum_{s'} P_r(s, r_k | s') \cdot P_r(s', r_{j < k}) \\
&= \sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s')
\end{aligned} \tag{2.41}$$

โดยกำหนดเงื่อนไขเริ่มต้นของ $\alpha_k(s)$ ที่เวลา $k = 0$

$$\alpha_0(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \tag{2.42}$$

การคำนวณ $\alpha_k(s)$ แบบ (Forward Recursion) แสดงได้ดังรูปที่ 2.15



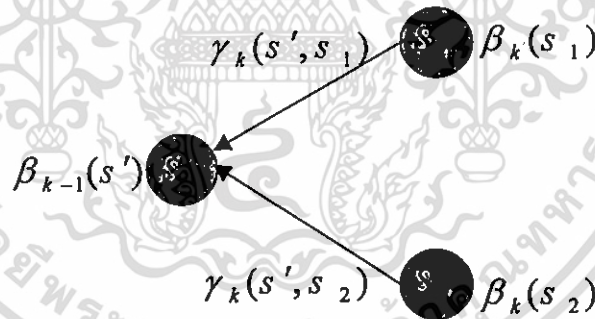
จากสมการ (2.39) ค่าความน่าจะเป็น $\beta_k(s)$ สามารถหาได้จากการการคำนวณแบบซ้ำย้อนกลับ (Backward Recursion) โดยที่จะสามารถหาความน่าจะเป็น $\beta_k(s)$ ได้ดังสมการที่ (2.43)

$$\begin{aligned}
\beta_{k-1}(s') &= P_r(r_k, r_{j>k} | s') \\
&= \sum_s P_r(s, r_k, r_{j>k} | s') \\
&= \sum_s \frac{P_r(s', s, r_k, r_{j>k})}{P_r(s')} \\
&= \sum_s \frac{P_r(r_{j>k} | s', s, r_k) \cdot P_r(s', s, r_k)}{P_r(s')} \\
&= \sum_s P_r(r_{j>k} | s', s, r_k) \cdot P_r(s', r_k | s') \\
&= \sum_s P_r(r_{j>k} | s) \cdot P_r(s, r_k | s') \\
&= \sum_s \beta_k(s) \cdot \alpha_k(s', s)
\end{aligned} \tag{2.43}$$

โดยกำหนดเงื่อนไขเริ่มต้นของ $\beta_k(s)$ ที่เวลา $k = \tau$

$$\beta_\tau(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \tag{2.44}$$

การคำนวณ $\beta_k(s)$ แบบ (Backward Recursion) แสดงได้ดังรูปที่ 2.16



$$\beta_{k-1}(s') = \gamma_k(s', s_1) \cdot \beta_k(s_1) + \gamma_k(s', s_2) \cdot \beta_k(s_2)$$

รูปที่ 2.16 แผนภาพแสดงการคำนวณแบบ (Backward Recursion)

จากสมการ (2.39) ค่าความน่าจะเป็น $\gamma_k(s', s)$ สามารถจัดรูปใหม่

$$\begin{aligned}
\gamma_k(s', s) &= P_r(s, r_k | s') \\
&= \frac{P_r(s', s, r_k)}{P_r(s')} \\
&= \frac{P_r(s', s)}{P_r(s')} \cdot \frac{P_r(s', s, r_k)}{P_r(s', s)} \\
&= P_r(u_k) \cdot p(r_k | u_k)
\end{aligned} \tag{2.45}$$

จากสมการ (2.41) และ สมการ (2.43) สามารถลดรูปเหลือได้

$$\gamma_k(s', s) = \exp\left(\frac{1}{2} \cdot u_k \cdot (L_C \cdot r_{k,1} + L(u_k))\right) \cdot \gamma_k^{(e)}(s', s) \tag{2.46}$$

เมื่อ

$$\gamma_k^{(e)}(s', s) = \exp\left(\frac{1}{2} \sum_{\nu=2}^n (L_C \cdot r_{k,\nu} \cdot x_{k,\nu})\right) \tag{2.47}$$

แทนสมการ (2.46) ในสมการ (2.47) จะได้ LLR

$$\begin{aligned}
L(\hat{d}_k) &= \ln \frac{\sum_{d_k=1}^{(s',s)} \alpha_{k-1}(s') \cdot \left(\exp\left(\frac{1}{2} \cdot (u_k = +1) \cdot (L_C \cdot r_{k,1} + L(u_k))\right)\right) \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s',s)} \alpha_{k-1}(s') \cdot \left(\exp\left(\frac{1}{2} \cdot (u_k = -1) \cdot (L_C \cdot r_{k,1} + L(u_k))\right)\right) \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)} \\
&= \ln \frac{\exp\left(\frac{1}{2} \cdot (L_C \cdot r_{k,1} + L(u_k))\right)}{\exp\left(-\frac{1}{2} \cdot (L_C \cdot r_{k,1} + L(u_k))\right)} + \ln \frac{\sum_{d_k=1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)} \\
&= L_C \cdot r_{k,1} + L(u_k) + \ln \frac{\sum_{d_k=1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)}
\end{aligned} \tag{2.48}$$

เมื่อพิจารณาสมการ (2.48) เปรียบเทียบกับสมการ (2.33) ข่าวดสาร (Extrinsic Information) มีค่า

$$L_e(\hat{d}_k) = \ln \frac{\sum_{d_k=1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k^{(e)}(s', s) \cdot \beta_k(s)} \tag{2.49}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวคือ อัลกอริทึม (Map) จะทำการคำนวณค่าต่าง ๆ ในโดเมนความน่าจะเป็น ซึ่งทำให้มีความยุ่งยากซับซ้อนเกินกว่าที่จะนำมาใช้งาน

2.4.3 2 อัลกอริทึม (Log-Map)

เมื่อพิจารณาอัลกอริทึม (Map) ดังที่ได้กล่าวข้างต้นในโดเมนล็อก (Log) ค่าความน่าจะเป็น $\gamma_k(s', s)$, $\alpha_k(s)$ และ $\beta_k(s)$ จะถูกแปลงให้เป็นค่า $\gamma_k^{LM}(s', s)$, $\alpha_k^{LM}(s)$ และ $\beta_k^{LM}(s)$ ตามลำดับ

ค่า (Branch Metric) $\gamma_k(s', s)$ หาได้จาก

$$\begin{aligned} \gamma_k^{LM}(s', s) &= \ln \gamma_k(s', s) \\ &= \frac{1}{2} \cdot u_k \cdot L(u_k) + \frac{1}{2} \cdot L_c \cdot r_{k,1} \cdot u_k + \frac{1}{2} \sum_{v=2}^n (L_c \cdot r_{k,v} \cdot x_{k,v}) \end{aligned} \quad (2.50)$$

ค่าสถานะแบบไปหน้า (Forward State Metric) $\alpha_k^{LM}(s)$ หาได้จาก

$$\begin{aligned} \alpha_k^{LM}(s) &= \ln \alpha_k(s) \\ &= \ln \left(\sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s') \right) \\ &= \ln \left(\sum_{s'} e^{\gamma_k^{LM}(s', s)} \cdot e^{\alpha_{k-1}^{LM}(s')} \right) \\ &= \ln \left(\sum_{s'} \exp(\gamma_k^{LM}(s', s) + \alpha_{k-1}^{LM}(s')) \right) \end{aligned} \quad (2.51)$$

โดยกำหนดเงื่อนไขเริ่มต้นของ $\alpha_k^{LM}(s)$ ที่เวลา $k = 0$

$$\alpha_0^{LM}(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases} \quad (2.52)$$

ค่าสถานะแบบย้อนกลับ (Backward State Metric) $\beta_k^{LM}(s)$ หาได้จาก

$$\begin{aligned}
\beta_{k-1}^{LM}(s') &= \ln \alpha_{k-1}(s') \\
&= \ln \left(\sum_s \gamma_k(s', s) \cdot \beta_k(s) \right) \\
&= \ln \left(\sum_s e^{\gamma_k^{LM}(s', s)} \cdot e^{\beta_k^{LM}(s)} \right) \\
&= \ln \left(\sum_s \exp(\gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \right)
\end{aligned} \tag{2.53}$$

โดยกำหนดเงื่อนไขเริ่มต้นของ $\beta_k^{LM}(s)$ ที่เวลา $k = \tau$

$$\beta_{\tau}^{LM}(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases} \tag{2.54}$$

ดังนั้นค่า LLR สามารถจัดรูปใหม่

$$\begin{aligned}
L(\hat{d}_k) &= \ln \frac{\sum_{d_k=1}^{(s', s)} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}{\sum_{d_k=0}^{(s', s)} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)} \\
&= \ln \frac{\sum_{d_k=1}^{(s', s)} e^{\alpha_{k-1}^{LM}(s')} \cdot e^{\gamma_k^{LM}(s', s)} \cdot e^{\beta_k^{LM}(s)}}{\sum_{d_k=0}^{(s', s)} e^{\alpha_{k-1}^{LM}(s')} \cdot e^{\gamma_k^{LM}(s', s)} \cdot e^{\beta_k^{LM}(s)}} \\
&= \ln \frac{\sum_{d_k=1}^{(s', s)} \exp(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s))}{\sum_{d_k=0}^{(s', s)} \exp(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s))} \\
&= \ln \left(\sum_{d_k=1}^{(s', s)} \exp(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \right) \\
&\quad - \ln \left(\sum_{d_k=0}^{(s', s)} \exp(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \right)
\end{aligned} \tag{2.55}$$

เพื่อที่จะทำให้ค่า $\alpha_k^{LM}(s)$, $\beta_k^{LM}(s)$ และ $L(\hat{d}_k)$ จัดให้อยู่ในรูปที่ง่ายขึ้น ต้องทำการพิจารณาฟังก์ชัน $E(\cdot)$

กำหนดให้

$$E(x, y) = \ln(e^x + e^y) \tag{2.56}$$

จะได้ว่า

$$\begin{aligned}
 \ln(e^x + e^y) &= \ln e^x + \ln(e^x + e^y) - \ln e^x \\
 &= x + \ln \frac{e^x + e^y}{e^x} \\
 &= x + \ln(1 + e^{y-x})
 \end{aligned}
 \tag{2.57}$$

หรือ

$$\begin{aligned}
 \ln(e^x + e^y) &= \ln e^y + \ln(e^x + e^y) - \ln e^y \\
 &= y + \ln \frac{e^x + e^y}{e^y} \\
 &= y + \ln(1 + e^{x-y})
 \end{aligned}
 \tag{2.58}$$

ดังนั้น

$$E(x, y) = \max(x, y) + \ln(1 + e^{-|x-y|}) \tag{2.59}$$

สมการ (2.59) เรียกว่า ลอการิทึมแบบจาโคเบียน (Jacobian Logarithm)

ในทำนองเดียวกัน กำหนดให้

$$E(x, y, z, w) = \ln(e^x + e^y + e^z + e^w) \tag{2.60}$$

จะได้ว่า

$$\ln(e^x + e^y + e^z + e^w) = x + \ln(1 + e^{y-x} + e^{z-x} + e^{w-x}) \tag{2.61}$$

หรือ

$$\ln(e^x + e^y + e^z + e^w) = y + \ln(1 + e^{x-y} + e^{z-y} + e^{w-y}) \tag{2.62}$$

$$\ln(e^x + e^y + e^z + e^w) = z + \ln(1 + e^{x-z} + e^{y-z} + e^{w-z}) \tag{2.63}$$

$$\ln(e^x + e^y + e^z + e^w) = w + \ln(1 + e^{x-w} + e^{y-w} + e^{z-w}) \tag{2.64}$$

จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$E(x, y, z, w) = \max(x, y, z, w) + \ln(e^{x - \max(x, y, z, w)} + e^{y - \max(x, y, z, w)} + e^{z - \max(x, y, z, w)} + e^{w - \max(x, y, z, w)}) \quad (2.65)$$

ดังนั้น ฟังก์ชัน $E(\cdot)$ สามารถเขียนในรูปทั่วไป

$$\begin{aligned} E(x_i) &= E(x_1, x_2, \dots, x_k) \\ &= \ln\left(\sum_{i=1}^k e^{x_i}\right) \\ &= \max(x_i) + \ln\left(\sum_{i=1}^k e^{x_i - \max(x_i)}\right) \end{aligned} \quad (2.66)$$

เมื่อนำฟังก์ชัน $E(\cdot)$ จากสมการ (2.66) ไปใช้กับสมการ (2.51), (2.53), (2.55) จะได้ว่า

$$\alpha_k^{LM}(s) = E_{s'}(\gamma_k^{LM}(s', s) + \alpha_{k-1}^{LM}(s')) \quad (2.67)$$

$$\beta_{k-1}^{LM}(s') = E_s(\gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \quad (2.68)$$

และ

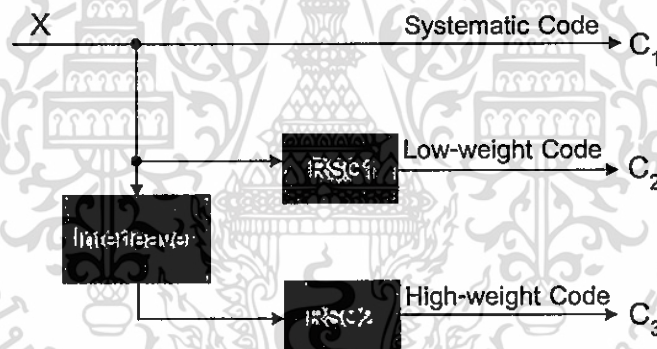
$$\begin{aligned} L(\hat{d}_k) &= E_{(s', s), d_k=1}(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \\ &\quad - E_{(s', s), d_k=0}(\alpha_{k-1}^{LM}(s') + \gamma_k^{LM}(s', s) + \beta_k^{LM}(s)) \end{aligned} \quad (2.69)$$

กล่าวคือการใช้อัลกอริทึมแบบ (Log-Map) จะทำการคำนวณค่าต่าง ๆ ในโดเมน ลอการิทึม (Log) โดยอาศัยฟังก์ชัน $E(\cdot)$ หรือ (Jacobian Logarithm) ทำให้การคำนวณทำได้ง่ายกว่าอัลกอริทึม แบบ (Map) และยังคงมีสมรรถนะที่ใกล้เคียงกัน

โครงสร้างการวางสลับบิต

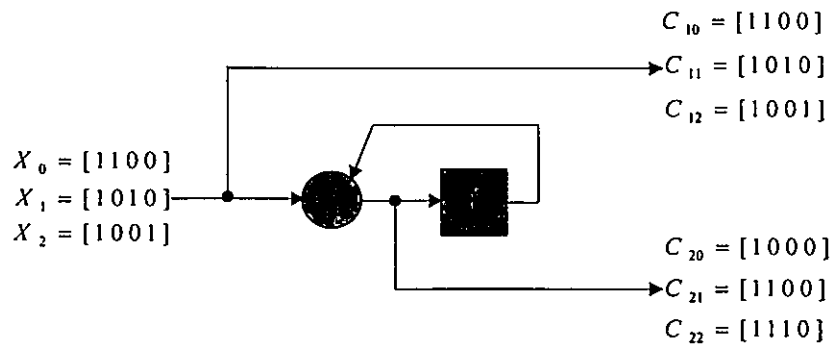
3.1 คุณลักษณะการวางสลับบิต

การวางสลับบิต (Interleaving) เป็นวิธีการสำคัญอีกวิธีหนึ่งในการนำมาใช้งานแพร่หลาย เพื่อใช้แก้ปัญหาของการผิดพลาดบิตข้อมูลแบบเบริสต์ เนื่องจากการผิดพลาดแบบเบริสต์เป็นรูปแบบของการผิดพลาดได้บ่อยครั้ง เนื่องจากความไม่เป็นอุดมคติของช่องสัญญาณที่ส่งผ่านทำให้บิตข้อมูลที่ได้รับมีการผิดพลาดอย่างต่อเนื่อง และผลดีสำหรับระบบที่มีการวางสลับบิตมีผลให้ความผิดพลาดแบบเบริสต์ก็จะถูกกระจายลงไปเป็นบิตข้อมูล ซึ่งการกระจายความผิดพลาดนี้จะช่วยทำให้ระบบไม่ต้องอาศัยวิธีการเข้ารหัสที่มีขีดความสามารถสูงในการแก้บิตที่ผิดพลาด สำหรับการออกแบบการวางสลับบิตบนการเข้ารหัสเทอร์โบ ได้มีการนำตัววางสลับบิตเข้ามาใช้ระหว่างองค์ประกอบภายในชุดตัวเข้ารหัส RSC ทั้ง 2 ชุด โดยที่กระบวนการทำงานการวางสลับบิตนั้นคือการจัดตำแหน่งของชุดบิตข้อมูล ให้มีการจัดเรียงตำแหน่งแตกต่างไปจากชุดเดิม ซึ่งทำให้ค่าน้ำหนักของคำรหัสมีค่าเพิ่มมากขึ้น แสดงดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างการเข้ารหัสเทอร์โบ

จากรูปที่ 3.1 ค่าของลำดับของข้อมูลอินพุตของชุดลำดับข้อมูลในชุดเข้ารหัส และมีค่าน้ำหนักของข้อมูลเป็น Low-Weight ได้ข้อมูลเอาต์พุตคือสำหรับชุดตัวเข้ารหัสที่ 1 (RSC1) และถ้าเรานำเอาข้อมูลทางด้านขาเข้าตัวเดียวกันเข้ามายังชุดตัวเข้ารหัสที่ 2 (RSC2) มีผลให้ค่าน้ำหนักของข้อมูลชุดตัวเข้ารหัสที่ 2 มีค่าเป็น Low-Weight เช่นเดียวกัน แต่เราไม่ต้องการค่า Low-Weight เราจึงหาวิธีการโดยจัดการข้อมูลด้านอินพุตของชุดตัวเข้ารหัสที่ 1 และ 2 ให้มีความแตกต่างกัน โดยการนำเอาการวางสลับบิตเข้ามาจัดตำแหน่งข้อมูลก่อนส่งข้อมูลมาที่ชุดตัวเข้ารหัสที่ 2 เพื่อที่จะได้ High-Weight ออกมาดังนั้นระบบการเข้ารหัสแบบเทอร์โบ ประกอบไปด้วยชุดของคำรหัสที่มีค่า Low, High ทั้งสองที่จะสามารถสรุปได้ว่า Low-Weight มาชุดเข้ารหัสที่ 1 ส่วน High-Weight จะได้มาจากชุดเข้ารหัสชุดที่ 2 แสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 ตัวอย่างของคำรหัสที่ได้จากชุดเข้ารหัสเทอร์โบ

จากรูปที่ 3.2 สัญญาณที่ด้านอินพุต คือ จะได้เอาต์พุต และตามลำดับ ข้อมูลอินพุต และจะมีรูปแบบที่ต่างกัน ซึ่งแสดงได้ดังตารางที่ 3.1 แสดงรายละเอียดผลลัพธ์ คือ คำรหัส (Codeword) และน้ำหนัก (Weight)

ตารางที่ 3.1 ชุดลำดับข้อมูลทางด้านอินพุตและเอาต์พุตของชุดเข้ารหัส

	Input Sequence(Xi)	Output Sequence(C1i)	Output Sequence(C2i)	Code word Weight(i)
i = 0	1100	1100	1000	1
i = 1	1010	1010	1100	2
i = 2	1001	1001	1110	3

จากตารางที่ 3.1 แสดงให้เห็นว่า คำรหัสและน้ำหนักนั้นสามารถมีค่าเปลี่ยนแปลงได้จากการวางสลับบิต ผลกระทบของการวางสลับบิตมีผลต่อสมรรถนะการเข้ารหัสเทอร์โบซึ่งมีผลโดยตรงต่อคุณ สมบัติของระยะการเข้ารหัส (Distance) เพื่อทำการหลีกเลี่ยงไม่ให้เกิดค่า Low-Weight อัตราความผิดพลาดของข้อมูล (BER) ของระบบการเข้ารหัสเทอร์โบเพื่อปรับปรุงสมรรถนะให้ดีขึ้น ดังนั้นจึงได้มีการค้นคว้าวิธีการออกแบบการวางสลับบิตสำหรับวิทยานิพนธ์ฉบับนี้เสนอ 5 ชนิดดังนี้สามารถแบ่งออกเป็น 2 ส่วนคือ

3.1.1. การวางสลับบิตที่นิยมใช้งานทั่วไปได้แก่

3.1.1.1 การวางสลับบิตแบบบล็อก (Block Interleaver)

3.1.1.2 การวางสลับบิตแบบบล็อกเฮลิคอลลิมาย (Block Helical-Simile Interleaver)

3.1.1.3 การวางสลับบิตแบบสุ่ม (Random Interleaver)

3.1.2 การวางสลับบิตแบบใหม่ซึ่งเกิดจากดัดแปลงโครงสร้างเดิม

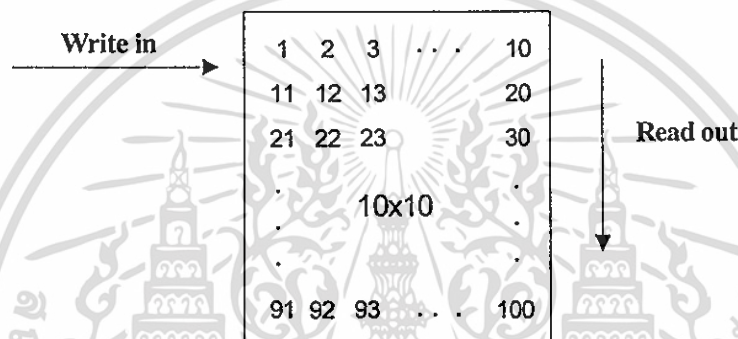
3.1.2.1 การวางสลับบิตแบบทิศทางตรงข้ามบล็อก (Reverse-Block Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การวางสลับบิตแบบทิศทางตรงข้ามบล็อกเฮลิคอลลีนา (Reverse Block Helical– Simile Interleaver)

3.2 การจำลองการทำงาน (Block Interleaver)

การวางสลับบิตแบบบล็อกสามารถอธิบายในรูปแบบของเมตริกซ์ ซึ่งมีวิธีการจัดเรียงตำแหน่งลำดับข้อมูลโดยให้ข้อมูลด้านอินพุตเรียงตามแนวนอน (Row) และข้อมูลด้านเอาต์พุตเรียงตามแนวตั้ง (Column) และขนาดของตัววางสลับบิตมีความสัมพันธ์กับชุดลำดับข้อมูลตัวเข้ารหัส (RSC1, RSC2) และขนาดของตัววางสลับบิต ($Row \times Column$) โดยจากรูปที่ 3.3 หมายถึงการอ่านข้อมูลทีละ 10 บิต จำนวน 10 ชุด โดยที่มีทิศทางการสลับตำแหน่งลำดับข้อมูล



รูปที่ 3.3 โครงสร้างการทำงาน (Block Interleaver)

ตัวอย่างเช่น (Write in) คือ 1 2 3 4 ... 10, 11 12 ... 90, 91 99 100

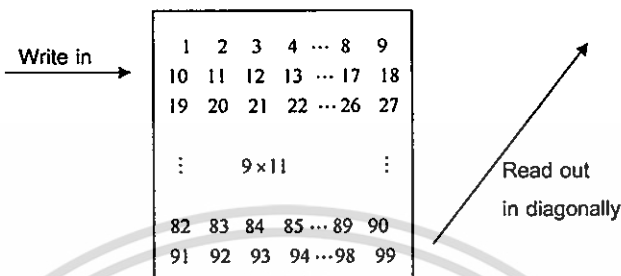
(Read out) คือ 1 11 21 ... 91, 2 22 ... 99, 10 20 100

การทำงานจากแผนภาพจำลองการทำงาน (Block Interleaver) ที่นำเสนอวิทยานิพนธ์มีขนาด 100, 625, 1024 บิต

จากรูปที่ 3.8 แสดงแผนภาพการทำงานของ (Block Interleaver) โดยที่มีโครงสร้างทางคณิตศาสตร์แบบเมตริกซ์มีลักษณะดังรูปที่ 3.3 แต่ในทางการทำงานการส่งข้อมูลจะส่งเป็นแนวยาวต่อเนื่องในลักษณะบิตต่อบิตและมีขั้นตอนการทำงานคือ ลำดับข้อมูลแนวยาวจะถูกอ่านเข้ามาเก็บยังหน่วยความจำ (Array) เพื่อที่ส่งข้อมูลออกไป โดยข้อมูลที่ส่งออกทีละแถว รูปแบบการวางสลับบิตนี้ มีการอ่านข้อมูลทีละเฟรมซึ่งประกอบไปด้วยข้อมูลในแต่ละเฟรม โดยข้อมูลตำแหน่งที่หนึ่งมาจากข้อมูลเฟรมที่หนึ่งและข้อมูลเฟรมที่สองมาจากตำแหน่งที่สองที่อ่านได้ตามลำดับจนกระทั่งถึงข้อมูลชุดสุดท้าย และการวางสลับบิตที่ภาคถอดรหัสจะอ้างอิงการทำงานในรูปแบบการสลับตำแหน่งบิตข้อมูลที่ภาคการเข้ารหัส เพื่อให้เป็นการสอดคล้องกันในการทำงาน

3.3 การจำลองการทำงาน (Block Helical-Simile Interleaver)

การวางสลับบิตแบบบล็อกเฮลิคอลลิมายสามารถอธิบายในรูปของเมตริกซ์ซึ่งมีวิธีการจัดเรียง ตำแหน่งลำดับข้อมูลโดยให้ข้อมูลด้านอินพุตเรียงตามแนวนอน (Row) และข้อมูลด้านเอาต์พุตเรียงกันตามแนวเส้นทแยงมุม (Diagonal) มีเงื่อนไขว่าจำนวนแถวและจำนวนหลักต้องไม่เท่ากันหรือจำนวน หลักต้องน้อยกว่าจำนวนแถวเสมอตามรูปที่ 3.4



รูปที่ 3.4 โครงสร้างการทำงาน (Block Helical-Simile Interleaver)

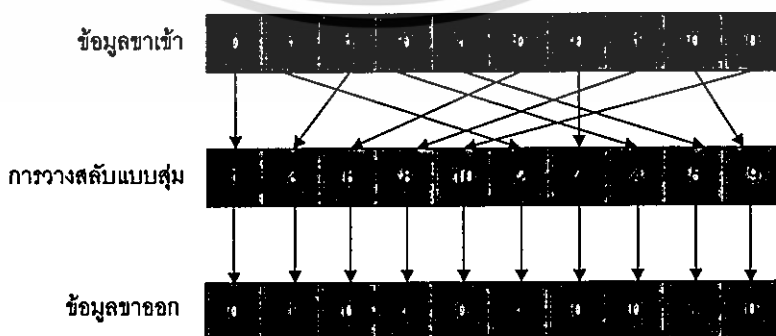
ตัวอย่างเช่น (Write in) คือ 1 2 3 4 ... 9, 10 11 ... 90, 91 92 99

(Read out) คือ 91 83 75 ... 27, 64 56 ... 8 99, 82 74 18 1 36 44...92

สามารถดูรายละเอียดขั้นตอนการทำงานจาก แผนภาพรูปที่ 3.9 ที่นำเสนอในวิทยานิพนธ์ มีขนาด 99, 624, 1023 บิต

3.4 การจำลองการทำงาน (Random Interleaver)

การวางสลับบิตแบบสุ่มมีวิธีการจัดตำแหน่งของข้อมูลโดยที่มีการกำหนดค่าตำแหน่งการวางลำดับข้อมูลแบบสุ่มแล้วเก็บค่าตำแหน่งดังกล่าวไว้ในหน่วยความจำ เพื่อใช้ในขั้นตอนการย้อนกลับเอาข้อมูลปลายทางตำแหน่งที่ได้จากการสุ่มนั้น จะนำมาเป็นตำแหน่งที่ใช้อ้างอิงในการจัดวางตำแหน่งดังรูปที่ 3.5 แสดงความยาวขนาดข้อมูลมีค่าเท่ากับ $N = 10$



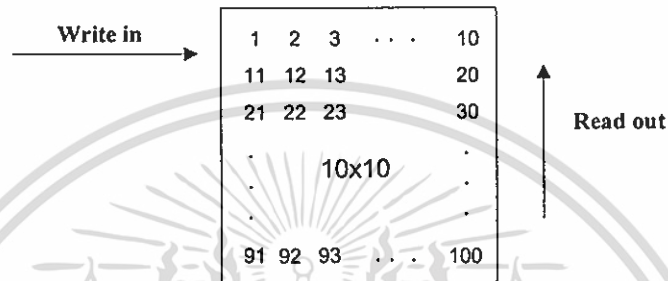
รูปที่ 3.5 โครงสร้างการทำงาน (Random Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถดูรายละเอียดขั้นตอนการทำงานจาก แผนภาพรูปที่ 3.10 นำเสนอในวิทยานิพนธ์ มีขนาด 100, 625, 1024 บิต

3.5 การจำลองการทำงาน (Reverse Block Interleaver)

การวางสลับบิตแบบทิศทางตรงข้ามบล็อก มีคุณสมบัติหลักเหมือนกันกับแบบบล็อกแต่มีการนำเสนอการเปลี่ยนรูปแบบโครงสร้างการสลับตำแหน่งต่างกัน เพื่อศึกษาอัตราการผิดพลาดของข้อมูล (BER) มีโครงสร้างการเปลี่ยนตำแหน่งข้อมูลดังรูปที่ 3.6



รูปที่ 3.6 โครงสร้างการทำงาน (Reverse Block Interleaver)

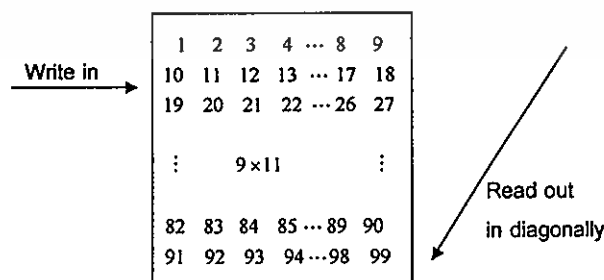
ตัวอย่างเช่น (Write in) คือ 1 2 3 4 ... 10, 11 12 ... 90, 91 99 100

(Read out) คือ 91 81 71 ... 1, 92 82 ... 2, 100 90 10

สามารถดูรายละเอียดขั้นตอนการทำงานจาก แผนภาพรูปที่ 3.11 นำเสนอในวิทยานิพนธ์ มีขนาด 100, 625, 1024 บิต

3.6 การจำลองการทำงาน (Reverse Block Helical-Simile Interleaver)

การวางสลับบิตแบบทิศทางตรงข้ามบล็อกเฮลิคอลลิมาย มีคุณสมบัติหลักเหมือนกันกับแบบบล็อกเฮลิคอลลิมาย แต่มีการนำเสนอการเปลี่ยนรูปแบบโครงสร้างการสลับตำแหน่งต่างกัน เพื่อการหาอัตราการผิดพลาดของข้อมูล (BER) มีโครงสร้างการเปลี่ยนตำแหน่งข้อมูลดังรูปที่ 3.7



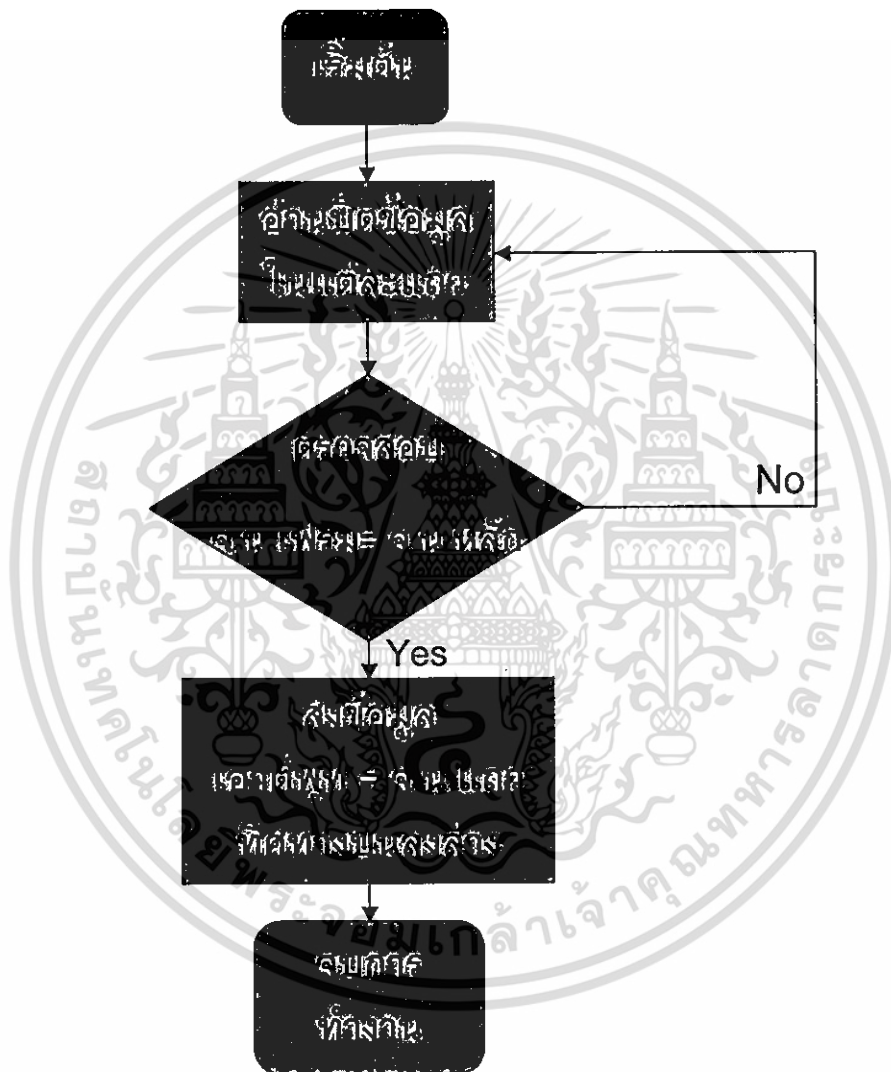
รูปที่ 3.7 โครงสร้างการทำงาน (Reverse Block Helical-Simile Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น (Write in) คือ 1 2 3 4 . . . 10, 11 12 . . . 90, 91 99 100

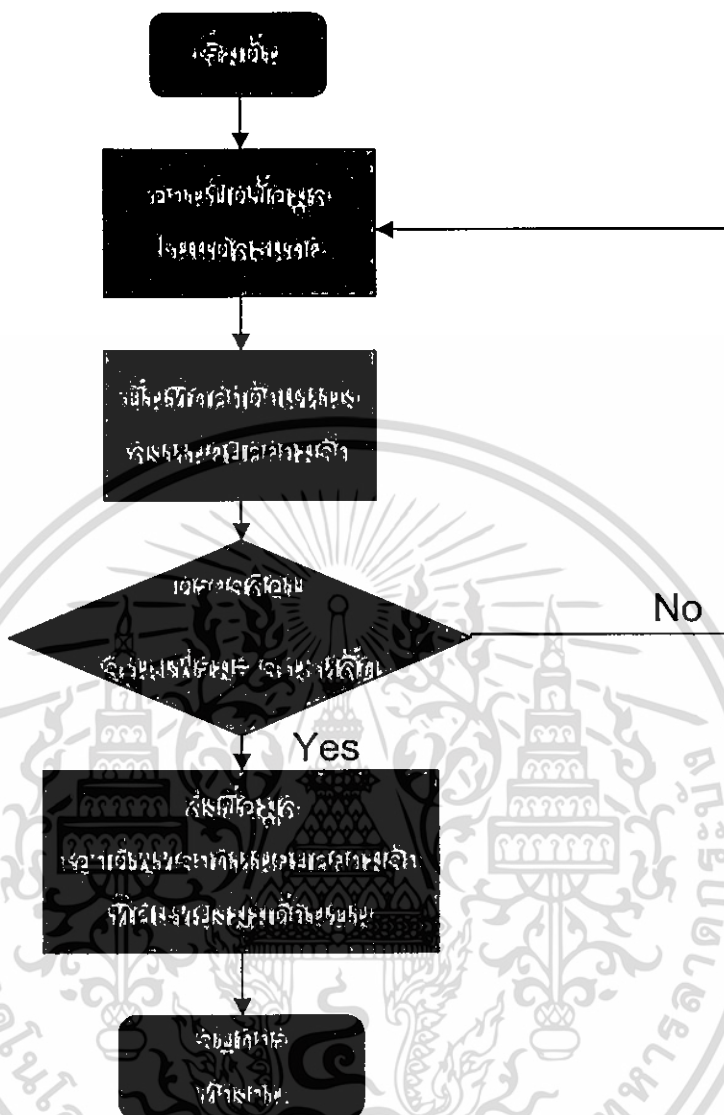
(Read out) คือ 9 17 73 . . . 36, 44 52 . . . 1, 18 26 82, 45 53 2 10, . . .

สามารถดูรายละเอียดขั้นตอนการทำงานจาก แผนภาพรูปที่ 3.12 นำเสนอในวิทยานิพนธ์
มีขนาด 99, 624, 1023 บิต



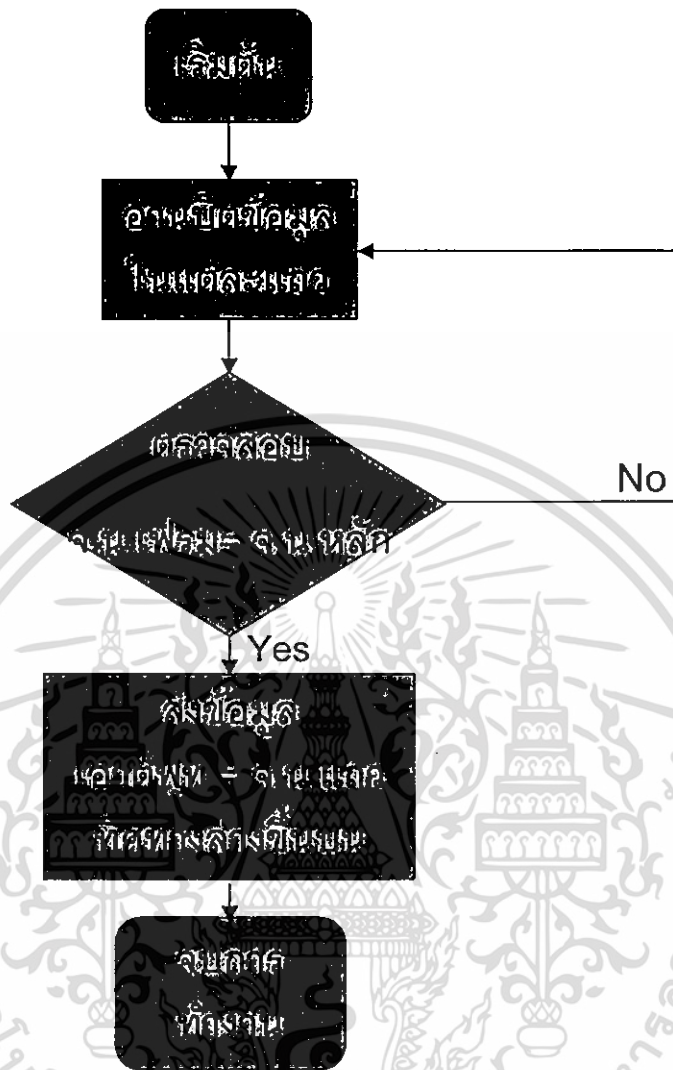
รูปที่ 3.8 แผนภาพการทำงาน (Block Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



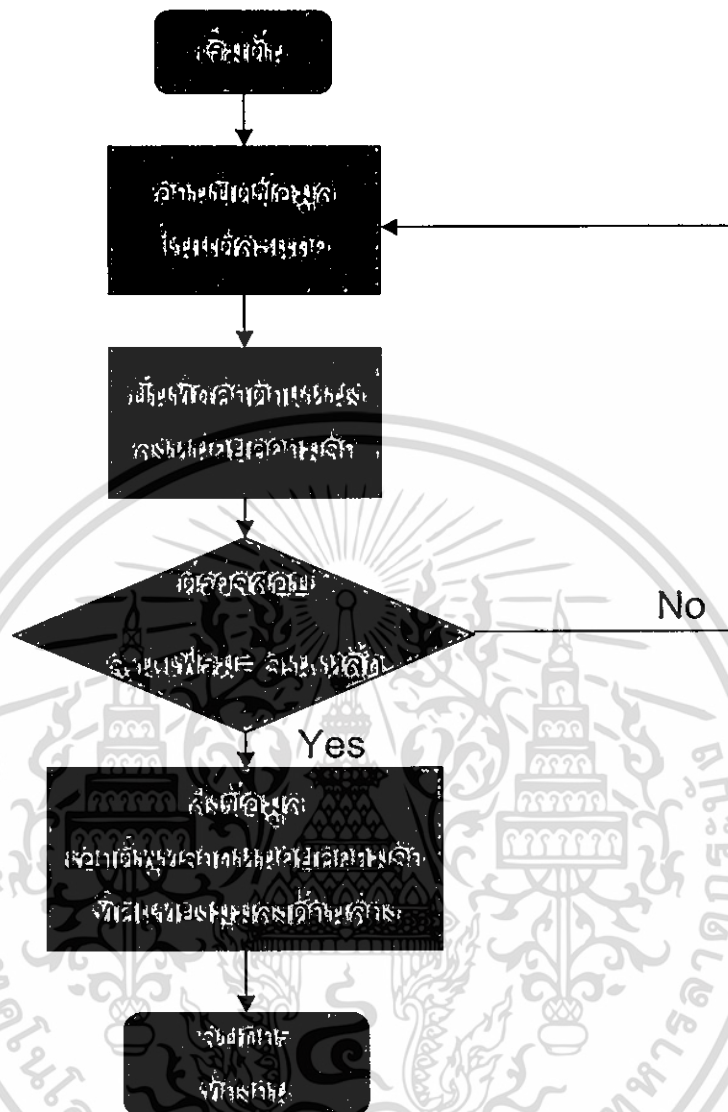
รูปที่ 3.9 แผนภาพการทำงาน (Block Helical-Simile Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แผนภาพการทำงาน (Reverse Block Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แผนภาพการทำงาน (Reverse Block Helical-Simile interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการจำลองแบบการทำงานของระบบ

วิทยานิพนธ์ฉบับนี้ทำการศึกษาสสมรรถนะของรหัสเทอร์โบบนช่องสัญญาณ AWGN โดยการเปลี่ยนแปลงรูปแบบการวางสลับบิต การจำลองแบ่งตามการศึกษาปัจจัยมีผลทำให้สมรรถนะการเข้ารหัสเทอร์โบมีประสิทธิภาพประกอบด้วย 5 ส่วน การวนซ้ำ (Iteration Decoding) ขนาดเฟรมของข้อมูล (Block size) โครงสร้างการเข้ารหัส (Generator Matrix) และอัตราการเข้ารหัส (Code Rate) และรูปแบบโครงสร้างการวางสลับบิต (Interleaver Design) โดยที่ประกอบด้วย 5 ชนิดที่นำมาใช้งาน ได้แก่ (Block , Reverse-Block , Helical-Simile, Reverse Block Helical-Simile and Random) ซึ่งการจำลองทั้งหมดในวิทยานิพนธ์ฉบับนี้ทำโดยการเขียนโปรแกรมและจำลองผลลงบนโปรแกรม (MATLAB) ทำงานบนระบบปฏิบัติการวินโดวส์ (Windows) บนคอมพิวเตอร์ส่วนบุคคล (Personal Computer : PC) มีรายละเอียดต่าง ๆ ของวิธีการจำลองและผลการจำลองมีดังต่อไปนี้

พารามิเตอร์หลักในการเข้ารหัสเทอร์โบ

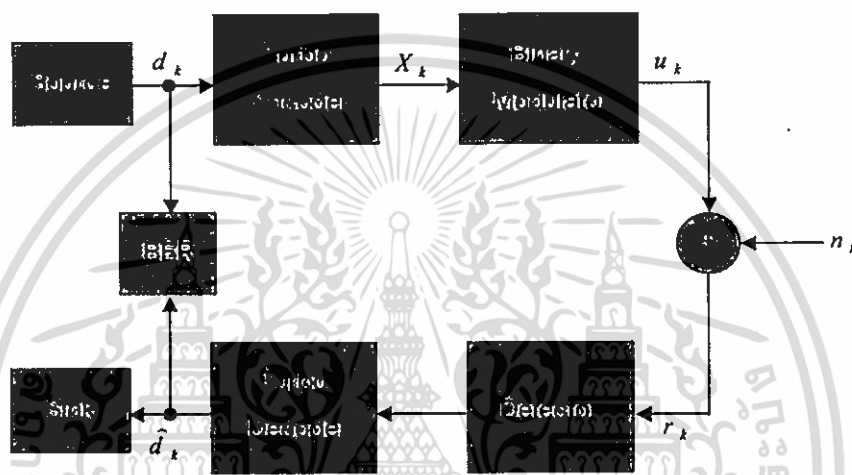
1. อัตราการเข้ารหัส (Code Rate)	1/2, 1/3
2. โครงสร้างการเข้ารหัส (Generator Matrix)	$G = [7, 5]_8$, $G = [37, 21]_8$
3. ชุดหน่วยความจำ (Constraint Length)	$K = 3, 5$
4. ขนาดเฟรมข้อมูล (Frame Size)	100, 625, 1024
5. การวางสลับบิต (Interleaver Design)	Block, Reverse-Block, Block Helical-Simile, Reverse Helical-Simile and Random
6. การวนซ้ำ (Iteration Decoding)	8 รอบ
7. การถอดรหัส (Decode's Algorithm)	Log-Map
8. การมอดูเลตชั้น (Modulation)	BPSK
9. ช่องสัญญาณ (Channel)	AWGN

4.1 การจำลองสมรรถนะรหัสเทอร์โบบนช่องสัญญาณ AWGN

ในการจำลองสมรรถนะของรหัสเทอร์โบบนช่องสัญญาณ AWGN ใช้วิธีมอนติคาร์โล (Monte Carlo Method) โดยที่เวลาใด ๆ เริ่มต้นด้วยการสร้างลำดับฐานสอง (0, 1) ด้วยความน่าจะเป็นเท่ากันและเป็นอิสระทางสถิติต่อกัน โดยสมมติให้บิตข้อมูลตัวเข้ารหัสเทอร์โบทำการเข้ารหัสบิตข้อมูลให้เป็นคำรหัส (Codeword) เนื่องจากช่องสัญญาณเป็นแบบไม่ต่อเนื่อง ตัวมอดูเลตฐานสองจะแปลงคำรหัส ให้เป็นสัญญาณฐานสองมีค่าเป็น (-1, +1) ก่อนที่ส่งผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องสัญญาณ โดยที่สัญญาณที่ส่งผ่านช่องสัญญาณ AWGN มีค่าการจางหาย ($=1$) รวมถึงการรบกวนแบบเกาส์เซียน ที่มีค่าการแปรปรวนตามสมการที่ (2.30) เนื่องจากตัวถอดรหัสเทอร์โบต้องการอินพุตแบบซอฟต์แวร์ ตัวตรวจจับจึงป้อนสัญญาณให้กับตัวถอดรหัสได้เป็นบิตข้อมูล ต่อจากนั้นทำการเปรียบเทียบบิตข้อมูลกับว่าเหมือนหรือแตกต่างกัน กระบวนการจำลองทั้งหมดดำเนินการต่อจนกระทั่งครบตามจำนวนที่ต้องการ สำหรับในวิทยานิพนธ์ฉบับนี้ได้ทำการทดสอบบิตข้อมูลตั้งแต่ ($\approx 10^4 - 10^5$) บิต ที่แต่ละค่าเมื่อทำการส่งบิตข้อมูลที่ละเฟรมจนครบตามจำนวนแล้ว จึงนับจำนวนบิตข้อมูลที่ตัดสินใจผิดพลาดเพื่อนำไปคำนวณหาอัตราความผิดพลาดที่สามารถแสดงได้ดังรูปที่ 4.1



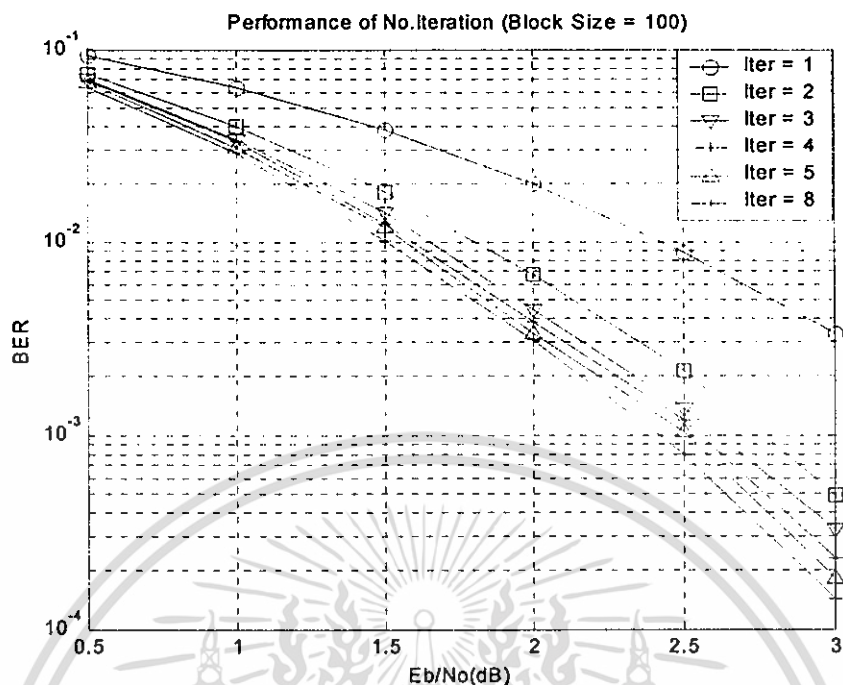
รูปที่ 4.1 การจำลองสมรรถนะของรหัสเทอร์โบบนช่องสัญญาณ AWGN

4.2. ผลจำลองแบบปัจจัยมีผลต่อสมรรถนะการเข้ารหัสเทอร์โบบนช่องสัญญาณ AWGN

- 4.2.1 การวนซ้ำ (Iteration Decoding)
- 4.2.2 ขนาดเฟรมข้อมูล (Block size)
- 4.2.3 โครงสร้างการเข้ารหัส (Generator Matrix)
- 4.2.4 อัตราการเข้ารหัส (Code Rate)
- 4.2.5 โครงสร้างการวางสลับบิต (Interleaver Design) ซึ่งประกอบด้วย 5 ชนิด
 - 4.2.5.1 Block
 - 4.2.5.2 Block Helical-Simile
 - 4.2.5.3 Random
 - 4.2.5.4 Reverse-Block
 - 4.2.5.5 Reverse Block Helical-Simile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลง (Iteration Decoding)



รูปที่ 4.2 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Iteration ต่าง ๆ

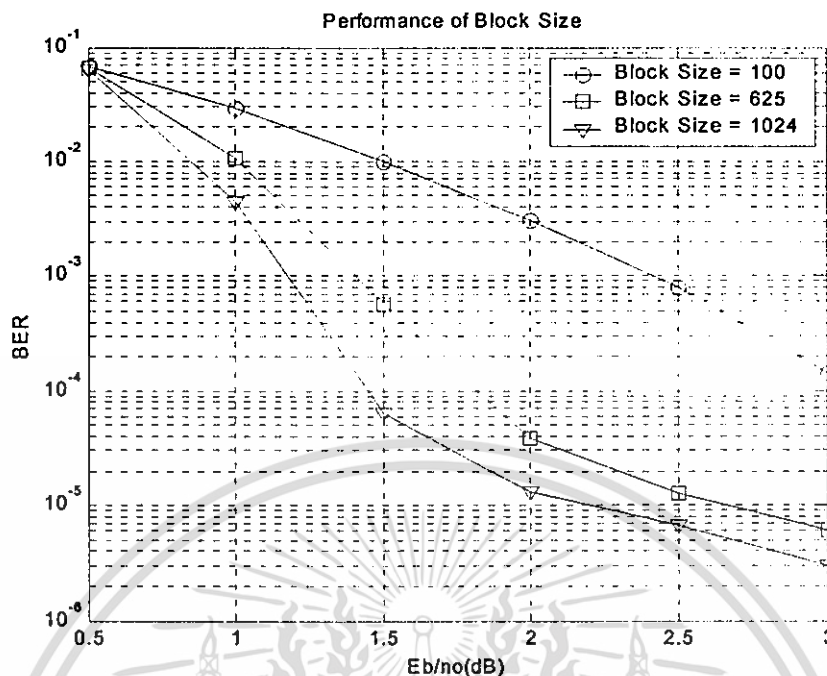
รูปที่ 4.2 แสดงผลการทำงานการวนซ้ำบนช่องสัญญาณ AWGN แสดงการวนซ้ำครั้งที่ 1, 2, 3, 4, 5 และ 8 โดยที่การการวนซ้ำครั้งที่ 6, 7, 8 มีค่าใกล้เคียงกันจะเห็นว่าเมื่อจำนวนรอบในการวนซ้ำมากขึ้น ดังนั้นรหัสเทอร์โบจะมีสมรรถนะที่ดีขึ้น

รูปที่ 4.3 แสดงผลการเปลี่ยนขนาดของเฟรมข้อมูลที่ส่ง 100, 625, 1024 บิต จะเห็นได้ว่าเมื่อมีการส่งข้อมูลขนาดใหญ่ขึ้นมีผลทำให้ค่าความผิดพลาดข้อมูลลดลง ดังนั้นรหัสเทอร์โบจะมีสมรรถนะที่ดีขึ้นสำหรับในวิทยานิพนธ์ฉบับนี้ใช้คำว่าเฟรมการส่งข้อมูล คือ (Block Size หรือขนาดของ Frame Size) มีความหมายเหมือนกัน เนื่องจากขนาดเฟรมข้อมูลก็คือขนาดของการวางสลับบิตเช่นกัน

รูปที่ 4.4 แสดงผลการเปลี่ยนแปลงรูปแบบโครงสร้างรหัสเป็น $G = [7,5]_8$ และ $G = [37,21]_8$ จะเห็นได้ว่าเมื่อมีการโครงสร้างใหญ่ขึ้นมีความซับซ้อนและหน่วยความจำมากขึ้นมีผลทำให้ค่าความผิดพลาดข้อมูลลดลง ดังนั้นรหัสเทอร์โบจะมีสมรรถนะที่ดีขึ้น

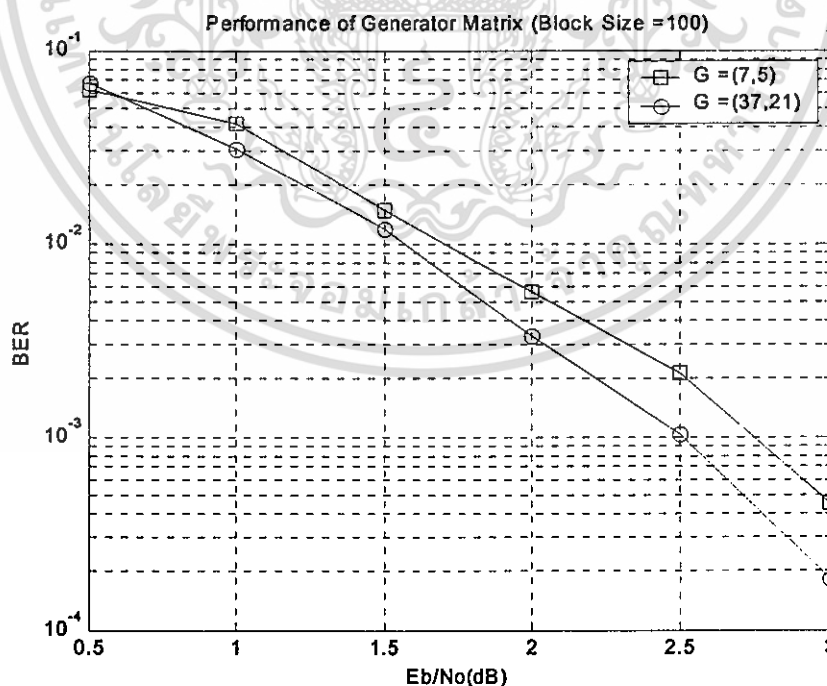
รูปที่ 4.5 แสดงผลการเปลี่ยนแปลงอัตราการใช้รหัสจาก 1/2 และ 1/3 จะเห็นได้ว่าเมื่อมีอัตราการใช้รหัสที่ต่ำ ทำให้มีค่าอัตราความผิดพลาดของข้อมูลลดลงเนื่องจากมีจำนวนบิตพาริตีมากในการตรวจสอบความผิดพลาด ดังนั้นรหัสเทอร์โบจะมีสมรรถนะที่ดีขึ้น

4.2.2 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลง (Block Size)



รูปที่ 4.3 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Block Size

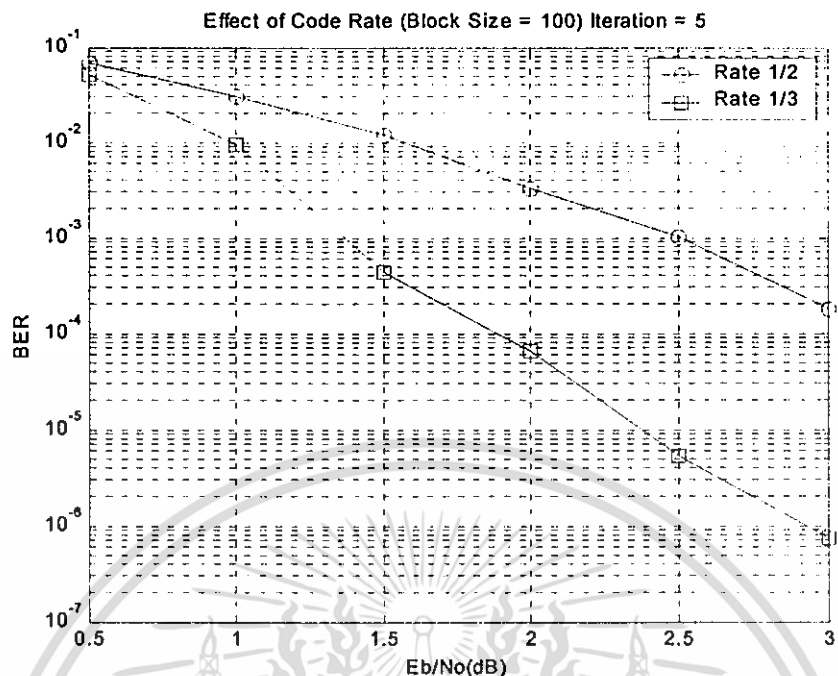
4.2.3 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลง (Generator Matrix)



รูปที่ 4.4 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Generator Matrix

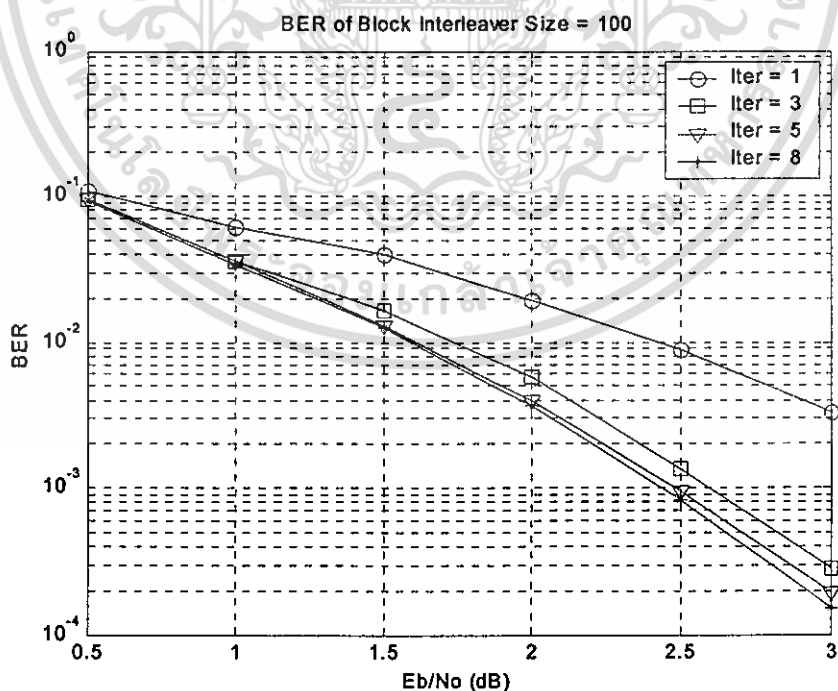
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลง (Code Rate)



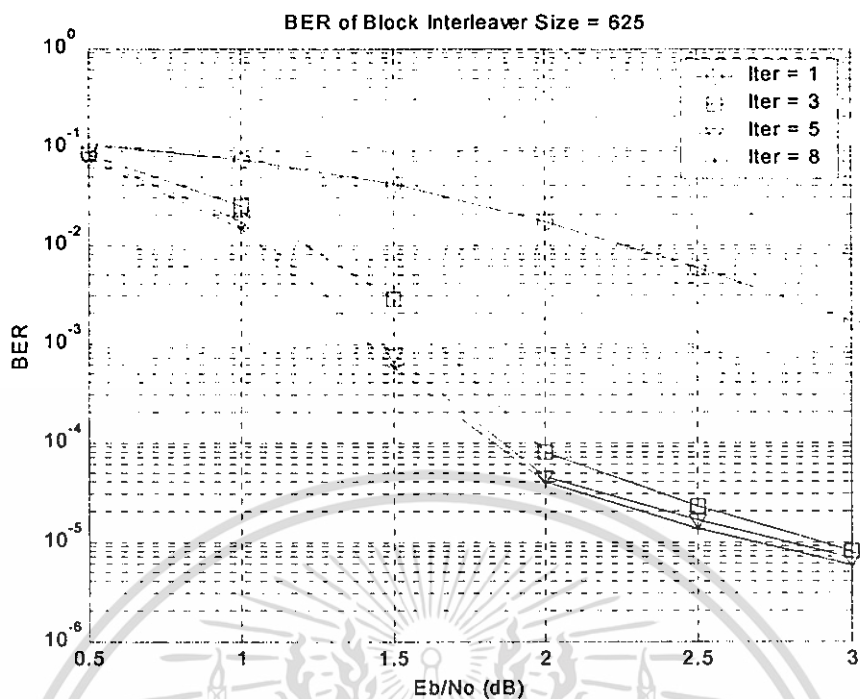
รูปที่ 4.5 BER ของการเข้ารหัสเทอร์โบที่เปลี่ยนแปลงตาม Code Rate

4.2.5.1 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลงโครงสร้างการวางสลับบิตแบบ Block ขนาด 100, 625 และ 1024 บิต

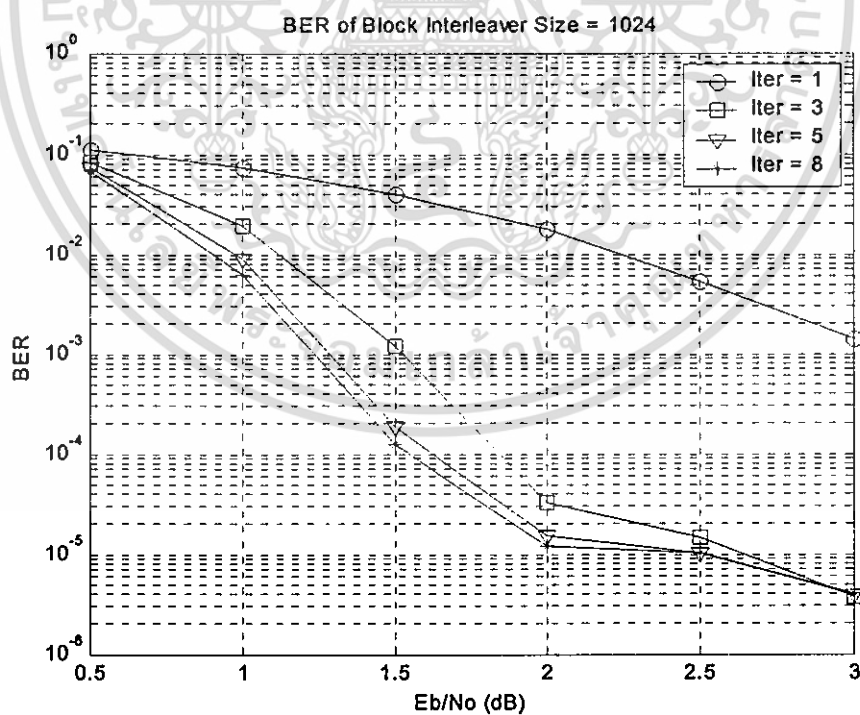


รูปที่ 4.6 BER ของการวางสลับบิตแบบ Block ขนาด 100 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



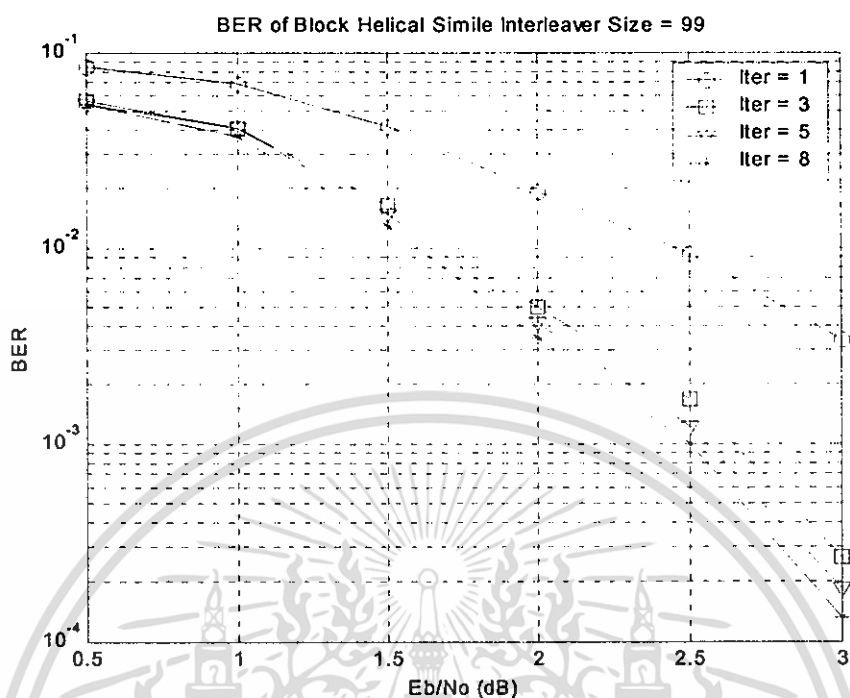
รูปที่ 4.7 BER ของการวางสลับบิตแบบ Block ขนาด 625 บิต



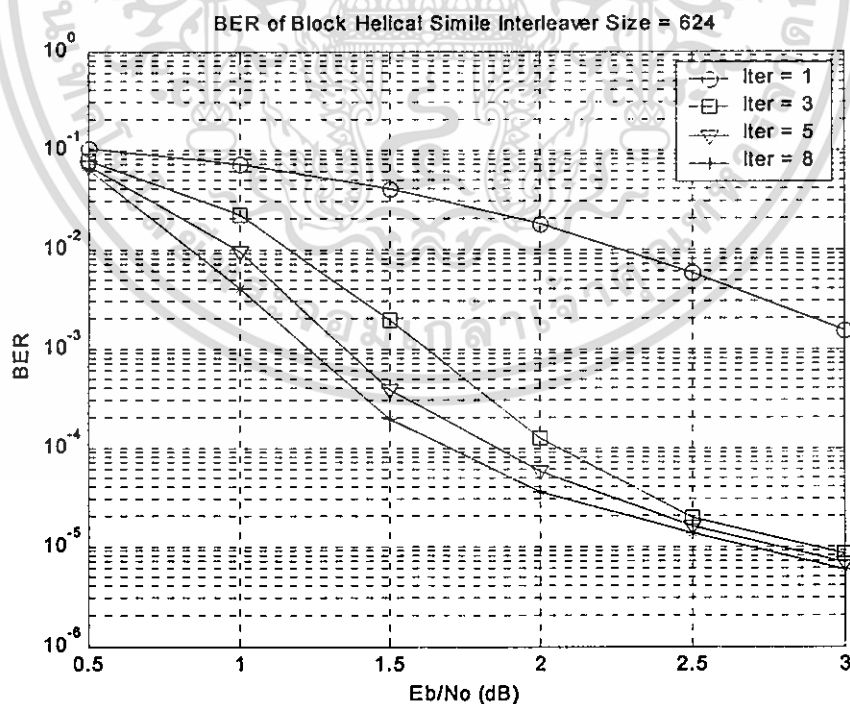
รูปที่ 4.8 BER ของการวางสลับบิตแบบ Block ขนาด 1024 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5.2 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลงโครงสร้างการวางสลับบิตแบบ Block Helical-Simile ขนาด 99, 624 และ 1023 บิต

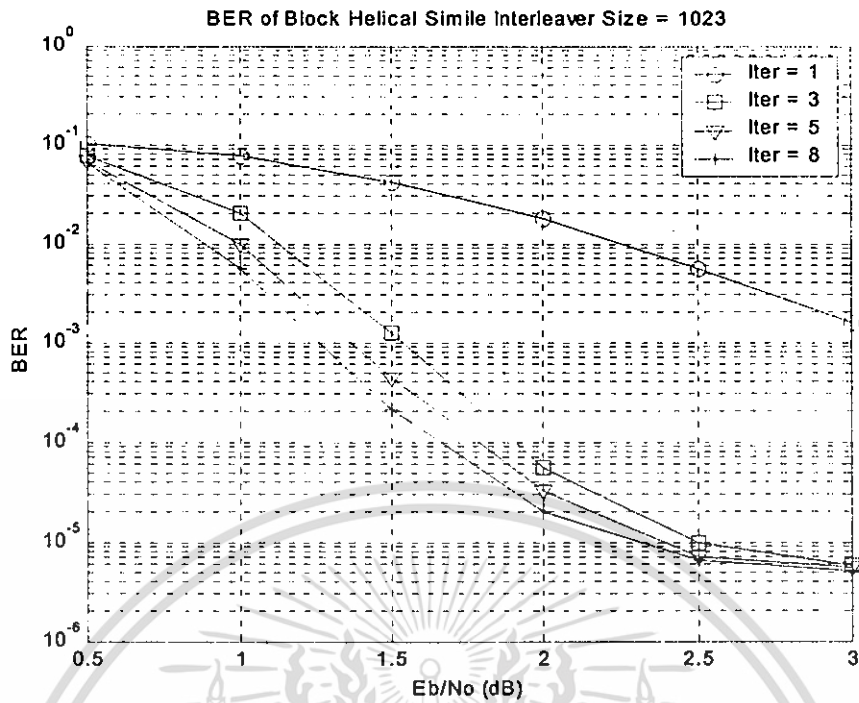


รูปที่ 4.9 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 99 บิต



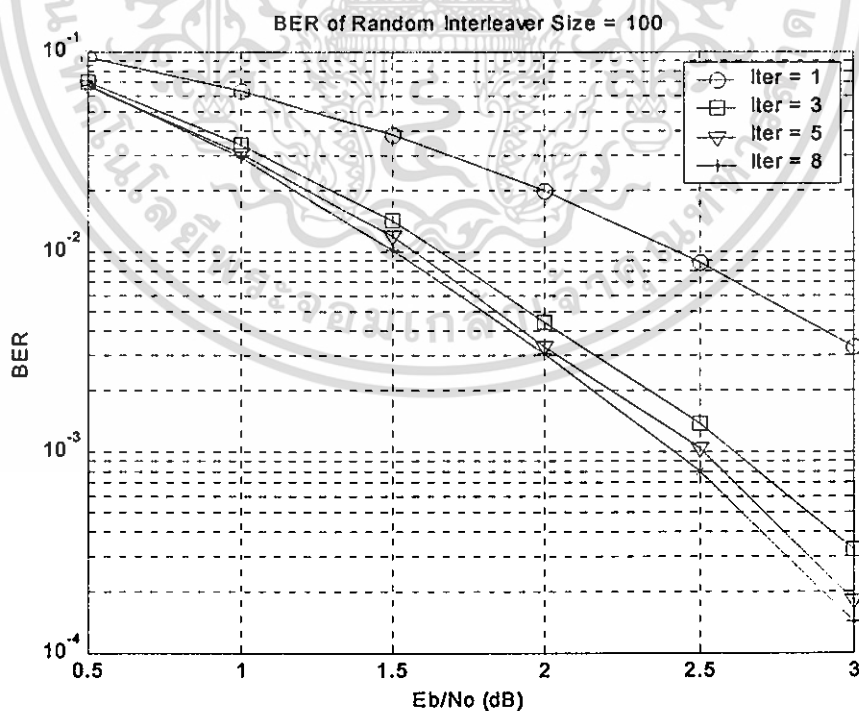
รูปที่ 4.10 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 624 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



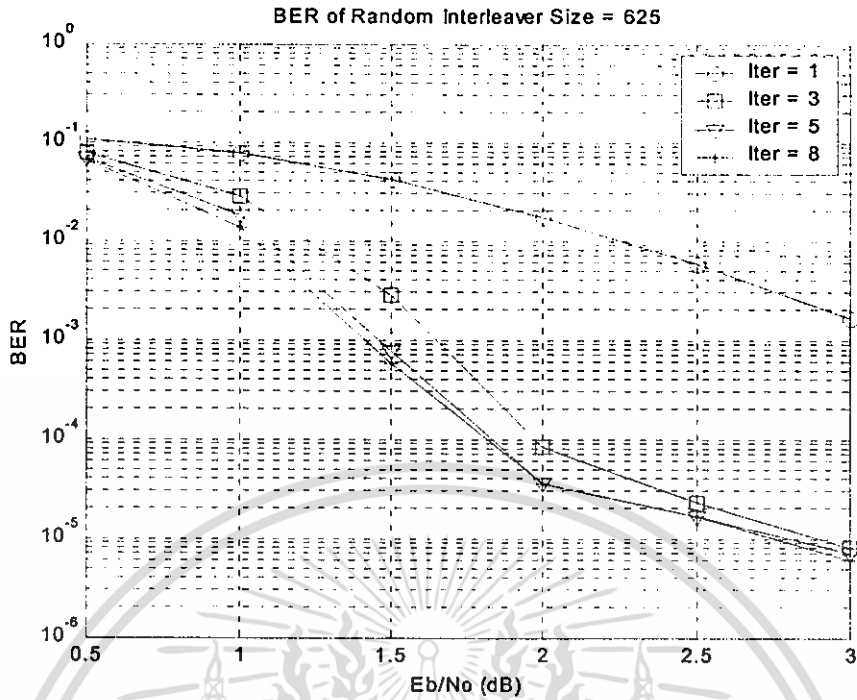
รูปที่ 4.11 BER ของการวางสลับบิตแบบ Block Helical-Simile ขนาด 1023 บิต

4.2.5.3 ผลการจำลองแบบการเข้ารหัสเทอร์โบโดยเปลี่ยนแปลงโครงสร้างการวางสลับบิตแบบ Random ขนาด 100, 625 และ 1024 บิต

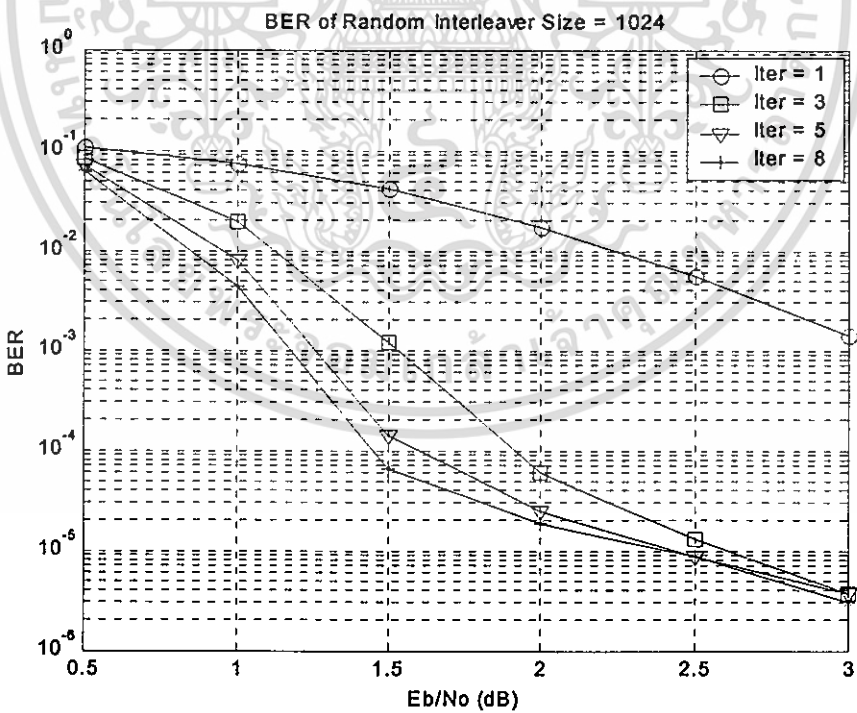


รูปที่ 4.12 BER ของการวางสลับบิตแบบ Random ขนาด 100 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 BER ของการวางสลับบิตแบบ Random ขนาด 625 บิต



รูปที่ 4.14 BER ของการวางสลับบิตแบบ Random ขนาด 1024 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.6, 4.7 และ 4.8 เป็นการแสดงการวางสลับบิตแบบ Block มีขนาดของเฟรมข้อมูลเปลี่ยนแปลงตั้งแต่ 100, 625 และ 1024 บิต เนื่องจากต้องการดูแนวโน้มการตกลงของเส้นกราฟและจากผลดีของการเข้ารหัสเทอร์โบมีการใช้วิธีการวนซ้ำจากการถอดรหัสโดยที่ค่าพารามิเตอร์ $\tilde{L}^2(\bar{d}_k)$ จะมีค่าที่เปลี่ยนแปลงเล็กน้อยเมื่อมีการจำนวนรอบวนซ้ำมากขึ้น และเมื่อมีขนาดเฟรมข้อมูลใหญ่ขึ้นมีผลให้ความผิดพลาดของข้อมูลลดลง เมื่อพิจารณาในแง่การวางสลับบิตโอกาสของการเกิดความน่าจะเป็นระหว่างความสัมพันธ์ของข้อมูล RSC 2 ชุดเนื่องจากคุณสมบัติของการวางสลับบิตคือลดความสัมพันธ์ของข้อมูลการเข้ารหัสทั้งสองชุด ดังนั้นเมื่อข้อมูลมีขนาดใหญ่คุณสมบัติการแยกความสัมพันธ์ระหว่างสองชุดข้อมูลทำงานได้เต็มที่

รูปที่ 4.9, 4.10 และ 4.11 แสดงการวางสลับบิตแบบ Block Helical-Simile มีขนาดของเฟรมข้อมูลเปลี่ยนแปลงตั้งแต่ 99, 624 และ 1023 บิต เนื่องจากต้องการดูแนวโน้มการตกลงของเส้นกราฟจากขนาดเฟรมข้อมูลที่ไม่เท่ากับ Block เนื่องจากรูปโครงสร้างการวางสลับบิตมีรูปแบบลักษณะเมตริกซ์ที่มีมิติขนาดแถวและหลัก โดยที่มีเงื่อนไขของขนาดของมิติหลักต้องน้อยกว่ามิติจำนวนแถวจากสมรรถนะการทำงานการถอดรหัสเทอร์โบเมื่อจำนวนการวนซ้ำมากขึ้นความผิดพลาดข้อมูลลดลงและเมื่อมีขนาดเฟรมข้อมูลใหญ่ขึ้นมีผลให้ความผิดพลาดของข้อมูลลดลง

รูปที่ 4.12, 4.13 และ 4.14 โดยที่แสดงการวางสลับบิตแบบ Random มีขนาดของเฟรมข้อมูลเปลี่ยนแปลงตั้งแต่ 100, 625 และ 1024 บิต เนื่องจากต้องการดูแนวโน้มการตกลงของเส้นกราฟความผิดพลาดของข้อมูล โดยนำผลที่ได้จากการถอดรหัสจะได้ข้อมูลและนำมาหาค่าความผิดพลาดด้วยวิธีนับจำนวนบิตข้อมูลที่ผิดพลาดเมื่อเทียบกับชุดบิตข้อมูลจริงก่อนการเข้ารหัสและหารด้วยจำนวนบิตข้อมูลที่นำมาเปรียบเทียบ ดังนั้นได้ค่าความผิดพลาดของข้อมูล (BER) และจากการวางสลับบิตแบบ Random มีแนวโน้มการตกลงของเส้นกราฟมีลักษณะคล้ายกันกับการวางสลับบิตแบบ Block และ Block Helical-Simile แต่มีค่าความผิดพลาดไม่เท่ากันดังนั้นในหัวข้อที่ 4.3 จะแสดงการเปรียบเทียบค่าความผิดพลาดเมื่อเปลี่ยนแปลงรูปแบบการวางสลับบิต

4.3 ผลการเปรียบเทียบสมรรถนะรหัสเทอร์โบโดยการวางสลับบิต

รูปที่ 4.15 แสดงผลการเปรียบเทียบการวางสลับบิตแบบ Block และ Reverse Block ที่มีขนาดข้อมูล 100 บิต จะเห็นได้ว่าการวางสลับบิตแบบ Reverse Block มีสมรรถนะที่ดีกว่าแบบ Block เนื่องจากวิธีการสลับตำแหน่งบิตข้อมูลแบบทิศทางตรงกันข้ามมีโอกาสเกิดความน่าจะเป็นในการลดความสัมพันธ์ระหว่างสองชุดข้อมูลมากกว่า

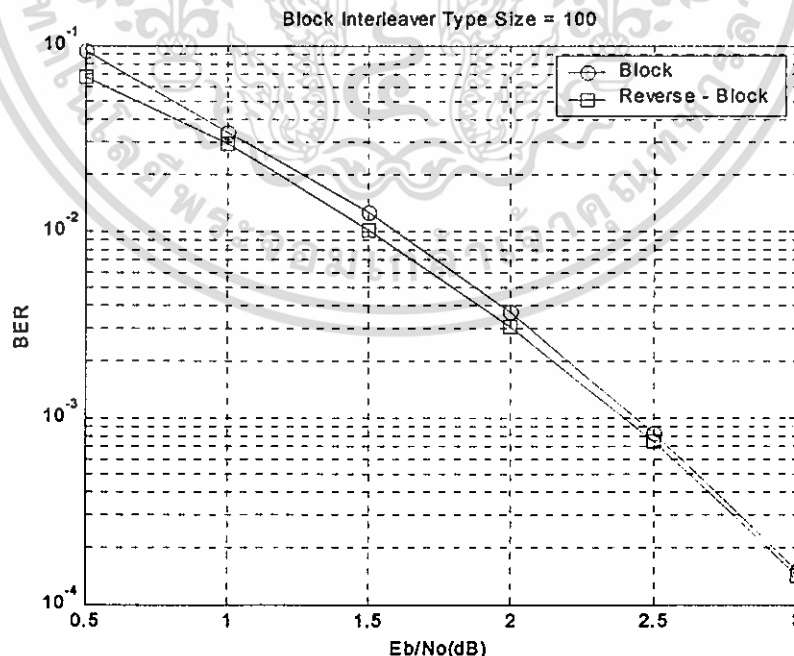
รูปที่ 4.16 ผลเปรียบเทียบการวางสลับบิต Block และ Reverse Block ที่ขนาด 625 บิต เห็นได้ว่าการวางสลับบิต Block และ Reverse Block มีค่าอัตราความผิดพลาดที่ใกล้เคียงกัน

รูปที่ 4.17 แสดงผลการเปรียบเทียบการวางสลับบิตแบบ Block และ Reverse Block ที่ขนาดข้อมูล 1024 บิต จะเห็นได้ว่าการวางสลับบิตแบบ Reverse Block มีสมรรถนะที่ดีกว่าแบบ Block เมื่อมีการพิจารณา E_b/N_o ที่ 3 dB และจากภาพรวมเมื่อขนาดข้อมูลใหญ่ขึ้นมีความผิดพลาดลดลงเนื่องจากมีโอกาสเกิดความน่าจะเป็นในการลดความสัมพันธ์ของข้อมูลสองชุดสูงขึ้น และเมื่อพิจารณากรณีทิศทางสลับตำแหน่งตำแหน่งบิตข้อมูลมีผลต่อความสัมพันธ์ของข้อมูลโดยที่การกระจายกลุ่มของความผิดพลาดได้มากขึ้น

รูปที่ 4.18 เป็นการแสดงผลการเปรียบเทียบการวางสลับบิตแบบ Block Helical-Simile และ Reverse Block Helical-Simile ที่มีขนาดข้อมูล 99 บิต จะเห็นได้ว่าการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะที่ดีกว่าแบบ Block Helical-Simile ที่ E_b/N_o เท่ากับ 3 dB จากภาพรวมมีอัตราความผิดพลาดที่ใกล้เคียงกัน เมื่อพิจารณาที่ SNR สูง ๆ แล้วการทำงานของ Reverse Block Helical-Simile มีสมรรถนะที่ดีกว่า

รูปที่ 4.19 เป็นการแสดงผลการเปรียบเทียบการวางสลับบิตแบบ Block Helical-Simile และ Reverse Block Helical-Simile ที่มีขนาดข้อมูล 624 บิต จะเห็นได้ว่าการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะที่ดีกว่าแบบ Block Helical-Simile ที่ E_b/N_o เท่ากับ 3 dB จากภาพรวมมีอัตราความผิดพลาดที่ใกล้เคียงกัน เมื่อพิจารณาที่ SNR สูง ๆ แล้วการทำงานของ Reverse Block Helical-Simile มีสมรรถนะที่ดีกว่า

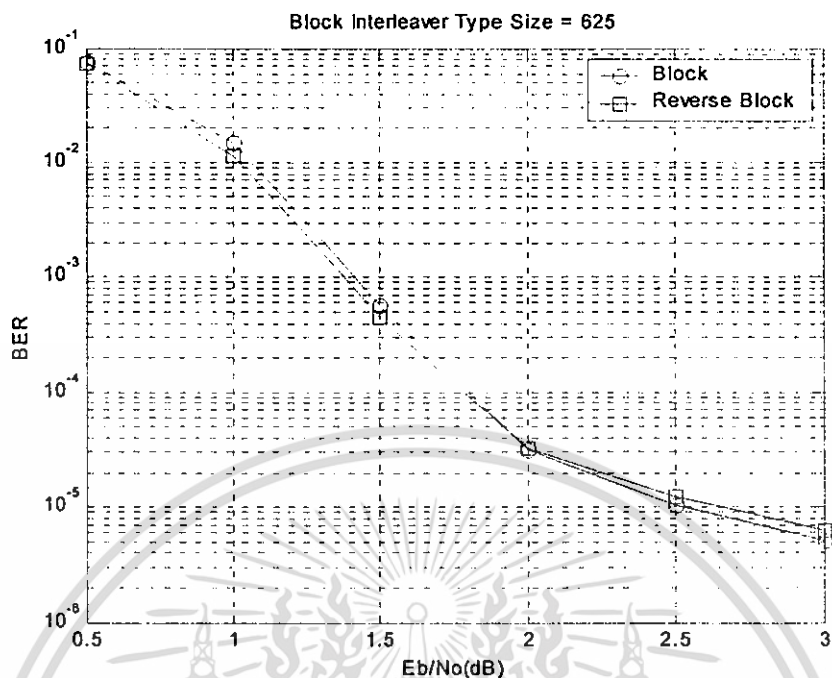
4.3.1 ผลเปรียบเทียบสมรรถนะ Block /Reverse Block Interleaver ขนาด 100บิต



รูปที่ 4.15 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 100 บิต

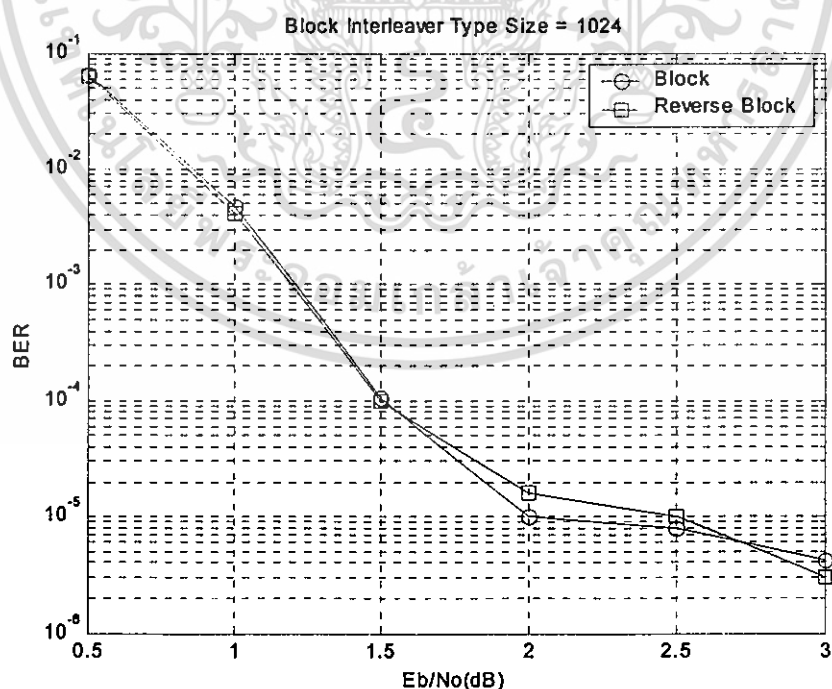
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ผลเปรียบเทียบสมรรถนะ Block / Reverse Block Interleaver ขนาด 625 บิต



รูปที่ 4.16 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 625 บิต

4.3.3 ผลเปรียบเทียบสมรรถนะ Block /Reverse Block Interleaver ขนาด 1024 บิต

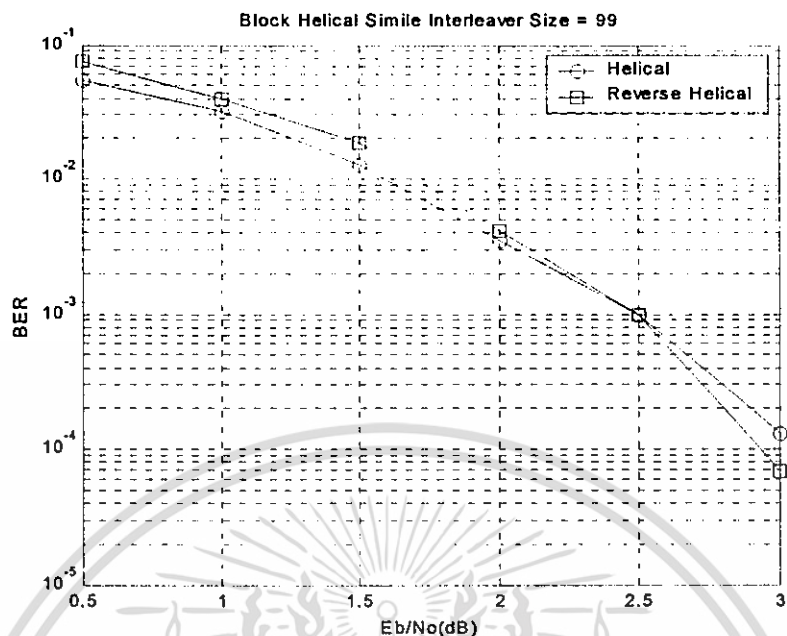


รูปที่ 4.17 การเปรียบเทียบ BER ของการวางสลับบิต Block และ Reverse Block ขนาด 1024 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 ผลเปรียบเทียบสมรรถนะ Block Helical-Simile /Reverse Block Helical-Simile

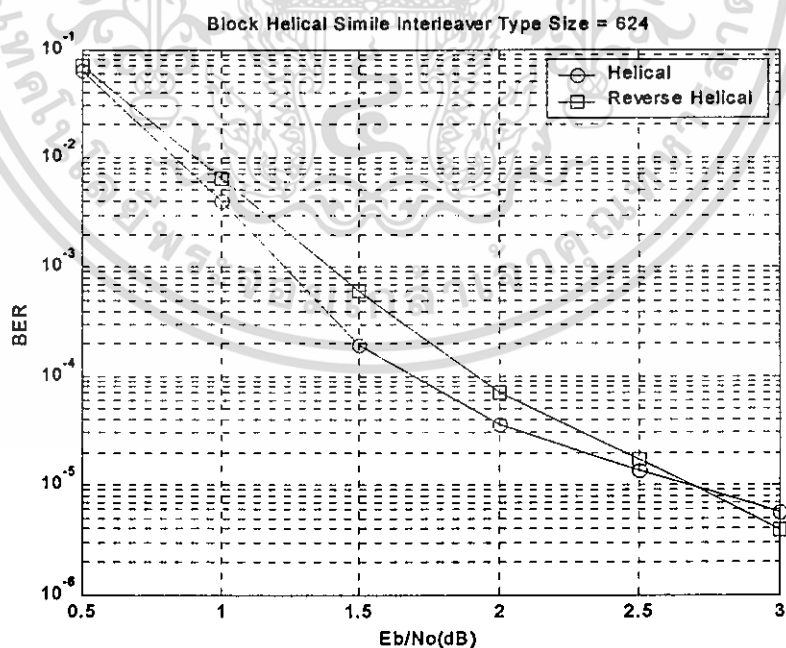
ขนาด 99 บิต



รูปที่ 4.18 การเปรียบเทียบ BER ของการวางลำดับบิต Block Helical-Simile และ Reverse Block Helical-Simile ขนาด 99 บิต

4.3.5 ผลเปรียบเทียบสมรรถนะ Block Helical-Simile /Reverse Block Helical-Simile

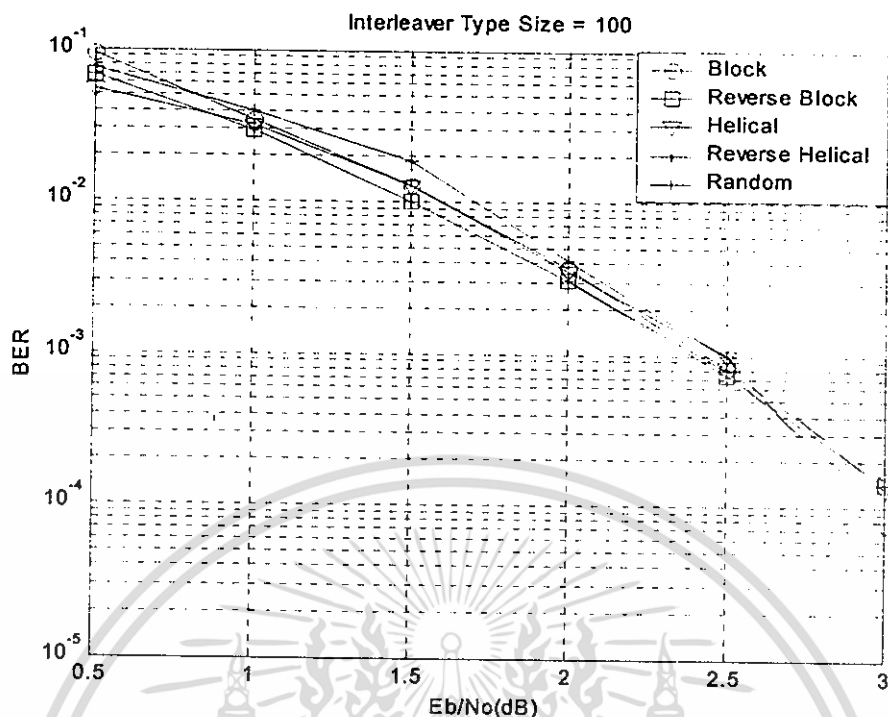
ขนาด 624 บิต



รูปที่ 4.19 การเปรียบเทียบ BER ของการวางลำดับบิต Block Helical-Simile และ Reverse Block Helical-Simile ขนาด 624 บิต

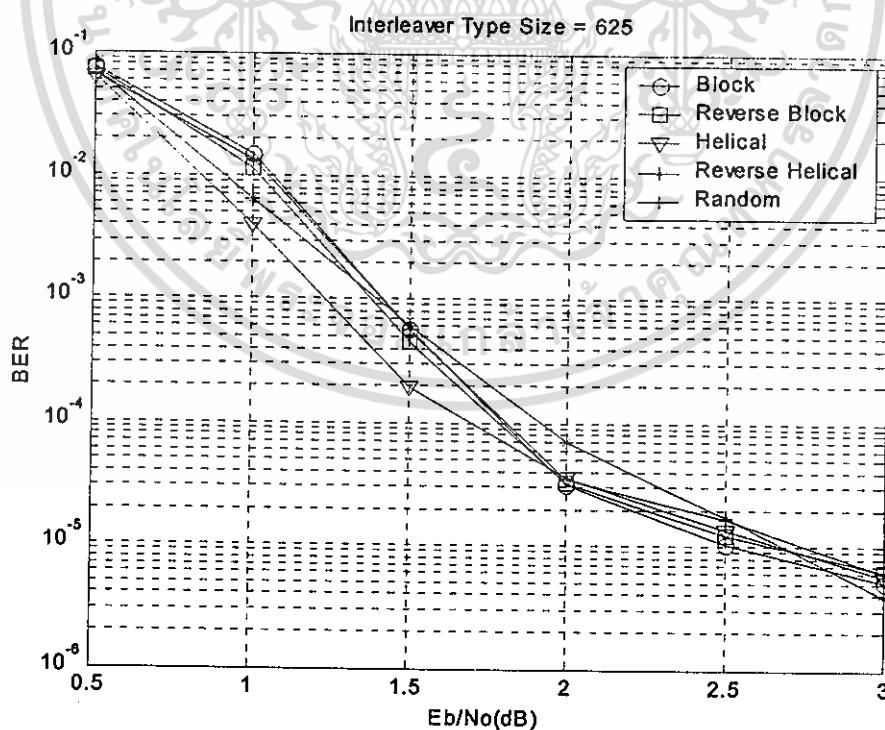
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 100 บิต



รูปที่ 4.21 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 100 บิต

4.3.7 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 625 บิต



รูปที่ 4.22 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 625 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

บทความที่ได้รับการตีพิมพ์

1.จอมใจ ศรีอำไพ, ผานิต ละมุล, กอบชัย เดชหาญ "การวิเคราะห์สมรรถนะการเข้ารหัส Turbo-OFDM มาตรฐาน IEEE 802.11a บนช่องสัญญาณมัลติพาธเฟดดิ้ง" ปีที่ 21 ฉบับที่ 3 เดือนกันยายน 2547



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

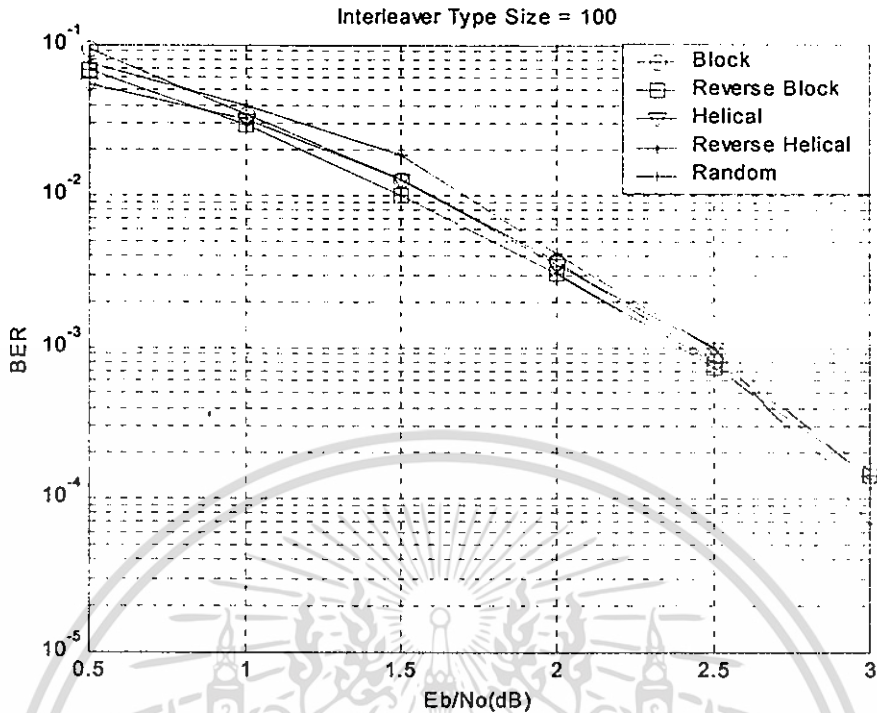
ประวัติผู้เขียน

ชื่อ	นางสาวจอมใจ ศรีอำไพ
เกิด	28 ตุลาคม 2520 จังหวัดเพชรบุรี
ประวัติการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543
ประวัติการทำงาน	ปี 2543 – 2544 ตำแหน่ง RF Engineer (GSM Network) บริษัท NDC Global Service (Thailand) Co., Ltd. ปี 2544 – ปัจจุบัน ตำแหน่ง RF Engineer (CDMA Network) บริษัท Motorola (Thailand) Co., Ltd.



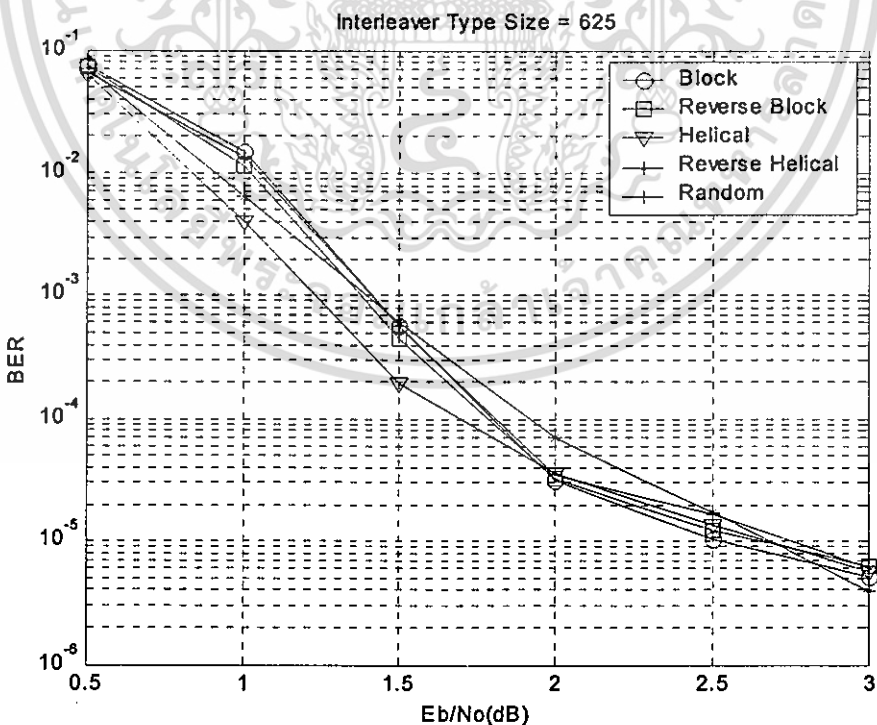
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 100 บิต



รูปที่ 4.21 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 100 บิต

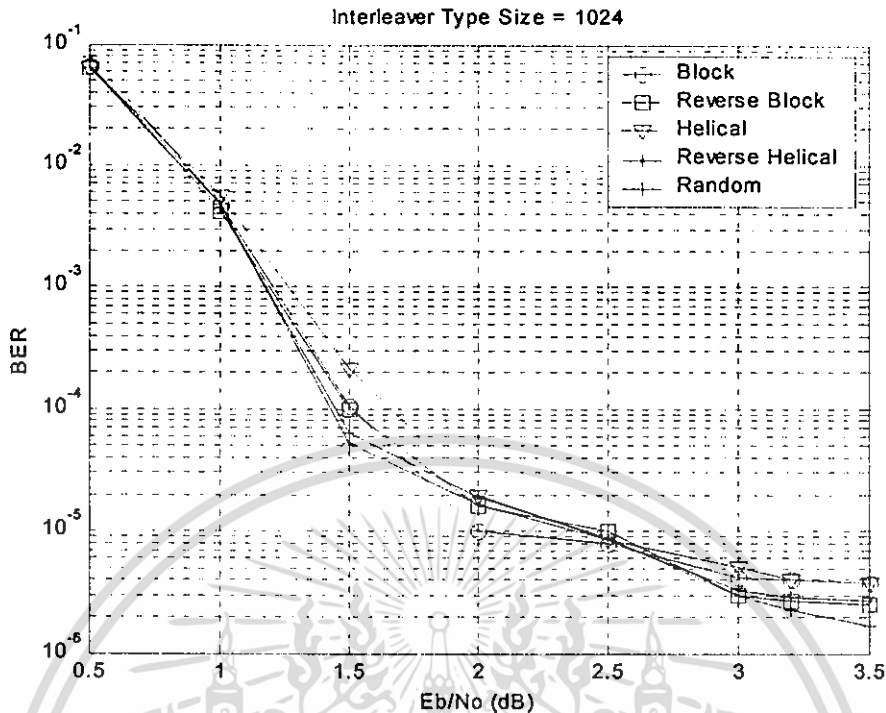
4.3.7 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 625 บิต



รูปที่ 4.22 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 625 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.8 ผลการเปรียบเทียบสมรรถนะการวางสลับบิตทั้งหมด ขนาด 1024 บิต



รูปที่ 4.23 การเปรียบเทียบ BER การวางสลับบิตทั้งหมดขนาด 1024 บิต

รูปที่ 4.21 แสดงผลการเปรียบเทียบการวางสลับบิตทั้งหมดทั้ง 5 ชนิดที่มีขนาดข้อมูล 100 บิต เมื่อพิจารณาที่ E_b/N_o เท่ากับ 3 dB เห็นได้ว่าการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะที่ดีที่สุด โดยที่การวางสลับบิตทั้ง 4 ชนิด จัดอยู่ในชนิดที่มีโครงสร้างจำพวกเดียวกันยกเว้นแบบ Random โครงสร้างการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะในการลดความล้มพันระหว่างข้อมูลสองชุดเข้ารหัส

รูปที่ 4.22 แสดงผลการเปรียบเทียบการวางสลับบิตทั้งหมดทั้ง 5 ชนิดที่มีขนาดข้อมูล 625 บิต เมื่อพิจารณาที่ E_b/N_o เท่ากับ 3 dB เห็นได้ว่าการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะดีที่สุด โครงสร้างการวางสลับบิตแบบ Reverse Block Helical-Simile มีสมรรถนะในการลดความล้มพันระหว่างข้อมูลสองชุดเข้ารหัส

รูปที่ 4.23 แสดงผลการเปรียบเทียบการวางสลับบิตทั้งหมดทั้ง 5 ชนิดที่มีขนาดข้อมูล 1024 บิต เมื่อพิจารณาที่ E_b/N_o เท่ากับ 3.5 dB เห็นได้ว่าการวางสลับบิตแบบ Random มีสมรรถนะดีที่สุดและจากรูปที่ 4.21, 4.22 และ 4.23 พบว่าเมื่อมีการส่งเฟรมข้อมูลขนาดเล็กการวางสลับบิต Reverse Block Helical-Simile มีสมรรถนะที่ดีและเฟรมข้อมูลขนาดกลางการวางสลับบิตแบบสุ่มมีสมรรถนะที่ดีที่สุด และเมื่อพิจารณาความซับซ้อนแล้วแบบสุ่มมีความสะดวกในการใช้งานและสมรรถนะใกล้เคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการลดความผิดพลาดของข้อมูลข่าวสารโดยใช้การวางสลับบิต บนการเข้ารหัสแบบเทอร์โบโดยใช้อัลกอริทึม (Log-Map) ในการถอดรหัสแบบวนซ้ำบนช่องสัญญาณ AWGN โดยที่ได้ทำการทดสอบระบบด้วยวิธีการจำลองแบบ (Simulation) และเปรียบเทียบค่าอัตราความผิดพลาดของบิตข้อมูลที่ใช้งานโดยโปรแกรม (MATLAB) สำหรับวิทยานิพนธ์นี้เสนอการวางสลับบิต ผลที่ได้จากการทำวิจัยสามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

1 ปัจจัยที่มีผลต่อการปรับปรุงสมรรถนะการเข้ารหัสเทอร์โบมีหลายปัจจัยด้วยกันพบว่าการวนซ้ำในการถอดรหัสแบบเทอร์โบซึ่งเป็นคุณสมบัติหลักที่ทำให้ประสิทธิภาพของรหัสเทอร์โบมีเสถียรภาพที่ดีไม่ว่าจะเลือกใช้อัลกอริทึมใด ๆ ในการถอดรหัส (วิทยานิพนธ์นี้เลือกใช้ Log-Map เนื่องจากให้ประสิทธิภาพที่ดีและมีโครงสร้างที่ไม่สลับซับซ้อนมาก)

2 ขนาดของเฟรมข้อมูลก็มีผลต่อสมรรถนะของการเข้ารหัสเทอร์โบซึ่งมีความสัมพันธ์กับขนาดของการวางสลับบิต เมื่อเฟรมข้อมูลขนาดใหญ่ขึ้นจึงมีผลให้ความผิดพลาดของข้อมูลลดลงเนื่องจากมีโอกาสเกิดความน่าจะเป็นสูงที่สามารถลดความสัมพันธ์ระหว่างข้อมูลในการเข้ารหัสได้สูงเช่นกัน

3 โครงสร้างการเข้ารหัสมีผลต่อสมรรถนะของการเข้ารหัสเทอร์โบ ถ้ามีโครงสร้างที่มีความซับซ้อนสูงก็มีอัตราความผิดพลาดต่ำและมีความจำเป็นต้องใช้หน่วยความจำสูงเช่นกัน

4 อัตราการเข้ารหัสมีผลต่อสมรรถนะของการเข้ารหัสเทอร์โบ ถ้าอัตราการเข้ารหัสมีค่าต่ำมีผลให้อัตราความผิดพลาดน้อยเนื่องจากมีโครงสร้างการเข้ารหัสมีการส่งค่าพาริตีตรวจสอบความผิดพลาดจำนวนมากในทางตรงกันข้ามถ้าพิจารณาในแง่แบนวิดท์ของช่องสัญญาณทำให้มีการส่งบิตข้อมูลข่าวสารช้าและสิ้นเปลืองแบนวิดท์สำหรับการสื่อสารข้อมูลที่ต้องการความรวดเร็ว

5 ตัวการวางสลับบิตเป็นส่วนหนึ่งของโครงสร้างการเข้ารหัสเทอร์โบสำหรับวิทยานิพนธ์ได้ทำการจำลอง แบบ 5 ชนิด ได้แก่ Block, Block Helical-Simile, Random, Reverse Block, Reverse Block Helical –Simile ซึ่งการวางสลับบิตแบบต่าง ๆ ที่นำเสนออยู่จำพวกการวางสลับบิตแบบ Block ที่มีการสลับตำแหน่งบิตข้อมูลในทิศทางที่ต่างกัน ยกเว้นการวางสลับบิตแบบ Random โดยที่เลือกใช้การวนซ้ำครั้งที่ 8 ในการเปรียบเทียบอัตราความผิดพลาดของข้อมูล

6 จากผลจำลองการเลียนแบบเปรียบเทียบอัตราความผิดพลาดของข้อมูลระหว่างการวางสลับบิตแบบ Block & Reverse Block เมื่อมีขนาดเฟรมข้อมูลขนาดเล็กและใหญ่ ผลการวางสลับบิตแบบ Reverse Block มีประสิทธิภาพดีกว่าแบบ Block ถ้ามีขนาดเฟรมข้อมูลขนาดใหญ่ผลการวางสลับบิตแบบ Block จะมีค่าอัตราความผิดพลาดคงที่

7 จากผลจำลองการเลียนแบบ เปรียบเทียบอัตราความผิดพลาดของข้อมูลระหว่างการวางสลับบิตแบบ Block Helical Simile & Reverse Block Helical Simile เมื่อมีขนาดเฟรมข้อมูลขนาดใหญ่การวางสลับบิตแบบ Reverse Block Helical Simile มีอัตราความผิดพลาดต่ำกว่าแบบ Block Helical Simile แต่ยังคงมีเสถียรภาพไม่ดิ่งเมื่อเปรียบเทียบกับแบบ Block Helical-Simile

8 จากผลจำลองการเลียนแบบเปรียบเทียบอัตราความผิดพลาดของข้อมูลระหว่างการวางสลับบิตทั้งหมด พบว่า การวางสลับบิตแบบ Random มีเสถียรภาพที่ดีกว่าการสลับบิตที่เหลือเมื่อเปรียบเทียบที่ขนาดบิตข้อมูล 100, 625, 1024 บิต และในแง่ของความซับซ้อนโครงสร้างการเข้ารหัสเทอร์โบการวางสลับบิตแบบ Random มีโครงสร้างไม่ซับซ้อน

สำหรับการจำลองสมรรถนะการวางสลับบิตบนการเข้ารหัสแบบเทอร์โบบนช่องสัญญาณ AWGN โดยการคำนวณหาอัตราความผิดพลาดของบิตข้อมูล (BER) ที่อัตราส่วน E_b/N_0 (dB) ต่าง ๆ ด้วยวิธีมอนติคาร์โล [8] ในวิทยานิพนธ์ฉบับนี้กำหนดจำนวนบิตข้อมูลทั้งหมดที่นำมาทดสอบให้เท่ากันสำหรับ E_b/N_0 (dB) แต่ละค่าซึ่งทำให้การหาค่า E_b/N_0 (dB) ที่ (BER) ต่าง ๆ กันมีความถูกต้องไม่เท่ากัน โดย E_b/N_0 (dB) ที่ (BER) สูง จะมีความถูกต้องมากกว่า ดังนั้นเพื่อให้การหาค่า E_b/N_0 (dB) ที่ (BER) ต่าง ๆ มีความถูกต้องเท่ากัน จึงควรมีการกำหนดจำนวนบิตข้อมูลสอดคล้องกับ (BER) ที่ต้องการ และพบว่าเมื่อตรวจสอบค่า (BER) สูง ใช้บิตข้อมูลจำนวนน้อยและทำให้ใช้เวลาในการจำลองการทำงานน้อยลง และตรวจสอบค่า (BER) ต่ำ จะต้องใช้บิตข้อมูลจำนวนมากและทำให้ใช้เวลาในการจำลองมากขึ้น

5.2 ปัญหาและข้อเสนอแนะในการพัฒนาต่อไป

จากการจำลองแบบการทำงานการเข้ารหัสเทอร์โบโดยการเปลี่ยนรูปแบบการวางสลับบิตเพื่อตรวจสอบค่าอัตราความผิดพลาดของบิตข้อมูลโดยใช้โปรแกรม (MATLAB) ซึ่งง่ายต่อการใช้งาน แต่ก็มีอุปสรรคในการประมวลผลที่ต้องใช้เวลานาน และข้อเสนอแนะในการเลือกใช้คอมไพเลอร์ (Compiler) ตัวอื่น ๆ เช่น C/C++

เอกสารอ้างอิง

- [1] Branka. V. and Jinhong Y., "Turbo Codes : Principles and Applications," Kluwer Academic Publishers, 2001.
- [2] Van. D. H, Kuchenbecker H.P, "Block Interleaving for Soft Decision Viterbi Decoding in OFDM Systems," Proc. VTC Fall 2001, Mobile Technology for Third Millennium; Atlantic City, October 7-11, 2001.
- [3] Claude. B., Alian G. and Punya T., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," Proc. International Conference on Communications, (Geneva, Switzerland), pp.1064-1070, May 1993.
- [4] Robertson P., "Illuminating the Structure of Code and Decode of Parallel Concatenated Recursive Systematic (Turbo) Codes," IEEE, Globecom'94, pp.1298-1303, 1994.
- [5] Bahl L., Cocke J., Jelinek F., and Raviv J., "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. Information Theory, vol. 20, pp.284-287, March 1974.
- [6] Bernard. S. , "Digital Communications Fundamental and Application ,"Printice Hall PTR on Second Edition., 2000.
- [7] Harada H. and Prasad R. , "Simulation and Software Radio for Mobile Communications," Artech House Universal Personal Communication Series, 2002.
- [8] Sadjadpour H. R., Sloane N. J. A., Salehi M.,and Nebe G. , "Interleaver Design for Turbo Code, IEEE Journal on selected Areas in Communications, Vol.19 No.5, May 2001.
- [9] Proakis J. G. , Digital Communications, New York, Mc Graw Hill, 1995.
- [10] Hokfelt J.,Edfors O. and Maseng T., "Interleaver Design for Turbo Codes Based on the Performance of Iterative Decoding ," Proc. IEEE ICC, Vol. 1 Vancouver, BC., Cannada, June, 1999.
- [11] Proakis J. G. and Salehi M. , Contemporary communication systems using Matlab, Boston, PWS Publishing Company, 1998
- [12] Hanzo L., Liew T. H. , Yeap B. L. , Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels, USA. John Wiley & Sons Ltd.

- [13] B.Heand M. Z. Wang , "Interleaver design for Turbo Code," International Conference on Information Communications and signal Processing ICICS , Singapore, September 1997.
- [14] ลัญฉกร วุฒิสัทติกุลกิจ, เทคโนโลยีโทรคมนาคม ทฤษฎีข่าวสารและการเข้ารหัส. กรุงเทพมหานคร : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

บทความที่ได้รับการตีพิมพ์

1.จอมใจ ศรีอำไพ, ผานิต ละมุล, กอบชัย เดชหาญ "การวิเคราะห์สมรรถนะการเข้ารหัส Turbo-OFDM มาตรฐาน IEEE 802.11a บนช่องสัญญาณมัลติพารเฟดดิ้ง" ปีที่ 21 ฉบับที่ 3 เดือนกันยายน 2547



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

- ชื่อ นางสาวจอมใจ ศรีอำไพ
- เกิด 28 ตุลาคม 2520 จังหวัดเพชรบุรี
- ประวัติการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543
- ประวัติการทำงาน ปี 2543 – 2544 ตำแหน่ง RF Engineer (GSM Network)
บริษัท NDC Global Service (Thailand) Co., Ltd.
- ปี 2544 – ปัจจุบัน ตำแหน่ง RF Engineer (CDMA Network)
บริษัท Motorola (Thailand) Co., Ltd.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้