

การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักบนรหัสโปรดักส์แอลดีพีซี
สำหรับช่องสัญญาณบันทึกข้อมูลแม่เหล็กสองมิติ

WEIGHTED BIT-FLIPPING DECODING OF PRODUCT LDPC CODES FOR
TWO-DIMENSIONAL MAGNETIC RECORDING (TDMR) CHANNEL



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2559

KMITL-2016-EN-M-010-182

การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักบนรหัสโปรดักส์แอลดีพีซี
สำหรับช่องสัญญาณบันทึกข้อมูลแม่เหล็กสองมิติ

WEIGHTED BIT-FLIPPING DECODING OF PRODUCT LDPC CODES FOR
TWO-DIMENSIONAL MAGNETIC RECORDING (TDMR) CHANNEL



ศิริวิชญ์ กิตติวิชญกุล
SIRAWIT KHITTIWITCHAYAKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2559
KMITL-2016-EN-M-010-182

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WEIGHTED BIT-FLIPPING DECODING OF PRODUCT LDPC CODES FOR
TWO-DIMENSIONAL MAGNETIC RECORDING (TDMR) CHANNEL



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2016

KMITL-2016-EN-M-010-182

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2016

FACULTY OF ENGINEERING

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การถอดรหัสแบบบิดพลิกบิตเชิงชนิดถ่วงน้ำหนักบนรหัสโปรตอกส์แอลดีพีซีสำหรับช่องสัญญาณ
บันทึกข้อมูลแม่เหล็กสองมิติ

Thesis Title Weighted bit-Flipping Decoding of Product LDPC Codes for Two-Dimensional
Magnetic Recording (TDMR) Channel

นักศึกษ นายศิววิษญ์ กิตติวิษญกุล

รหัสประจำตัว 57601155

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ศ.ดร.พรชัย ทรัพย์นินิ

หมายเลขวิทยานิพนธ์ KMITL-2016-EN-M-010-182

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.ณัฐกานต์	พุทธรักษ์	
รศ.ดร.สุวิพล	สิทธิชีวกาศ	
รศ.ดร.สัญญากร	วุฒิสัทธิกุลกิจ	
รศ.ดร.พิสิฐ	บุญศรีเมือง	
ศ.ดร.พรชัย	ทรัพย์นินิ	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 6 ตุลาคม พ.ศ. 2559 เวลา 13.00-15.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
คอมพิวเตอร์ คณะวิศวกรรมศาสตร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
วันที่ 6 ตุลาคม พ.ศ. 2559

หัวข้อวิทยานิพนธ์	การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักบนรหัสโปรดักส์แอลดีพีซี สำหรับช่องสัญญาณบันทึกข้อมูลแม่เหล็กสองมิติ
นักศึกษา	นายศศิวิทย์ กิตติวิญญกุล
รหัสประจำตัว	57601155
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2559
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ศ.ดร.พรชัย ทรัพย์นินิธิ

บทคัดย่อ

วิทยานิพนธ์เล่มนี้ นำเสนอการถอดรหัสบิตฟลิปปีงชนิดถ่วงน้ำหนักสำหรับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ โดยที่ความซับซ้อนในการถอดรหัสของอัลกอริทึมดังกล่าวนี้ จะมีค่าน้อยกว่าการถอดรหัสแอลดีพีซีแบบแพร่กระจายความเชื่อมั่น หรือการถอดรหัสแบบซอฟต์แวร์ เราสามารถแบ่งวิธีการถอดรหัสที่นำเสนอออกเป็น 3 ประเภท ได้แก่ การถอดรหัสแบบเพจ, การถอดรหัสสลับไปมาระหว่างแถวกับหลัก และการถอดรหัสสลับไปมาระหว่างแถวกับหลักแบบตัดแปลง ตามลำดับ โดยที่ปัญหาสมรรถนะไม่ลู่เข้าที่การวนซ้ำรอบสูง และปัญหาด้านความเร็วในการถอดรหัส ที่เกิดขึ้นกับอัลกอริทึมที่นำเสนอ นั้น จะได้รับการปรับแก้ในวิทยานิพนธ์เล่มนี้ด้วยเช่นกัน เราได้ทำการจำลองสมรรถนะของทั้ง 3 อัลกอริทึมที่นำเสนอ บนช่องสัญญาณของระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ ที่มีความหนาแน่นเชิงพื้นที่เท่ากับ 5.62 เทระบิตต่อตารางนิ้ว ผลที่ได้พบว่า ถึงแม้สมรรถนะของอัลกอริทึมที่นำเสนอทั้งสามจะต่ำกว่าการถอดรหัสแบบแพร่กระจายความเชื่อมั่น โดยจะสูญเสียค่าเกนรหัสนีประมาณ 0.5, 1.3 และ 1.5 เดซิเบล ตามลำดับ อย่างไรก็ตาม สามารถลดจำนวนโอเปอเรชันของวงจรรหัสอาร์คัวร์ อันได้แก่ วงจรคูณ และหาร ได้ถึงประมาณ 100 และ 90 เปอร์เซ็นต์ ตามลำดับ

Thesis	Weighted bit-flipping decoding of product LDPC codes for two-dimensional magnetic recording (TDMR) channel
Student	Mr.Sirawit Khittiwitthayakul
Student ID.	57601155
Degree	Master of Engineering
Program	Telecommunications Engineering
Year	2016
Thesis Advisor	Prof.Dr.Pornchai Supnithi

ABSTRACT

In this work, we propose a weighted bit-flipping (WBF) decoding for product LDPC codes of which decoding complexity is lower than belief-propagation (BP) decoding. Three distinct types based on page computation, row/column computation and modified row/column computation are proposed. In addition, the decoding issues of hard decision algorithm for product LDPC codes such as a performance degradation at high iteration number and decoding speed are improved in this work. We will evaluate the performance of three proposed algorithms on the two dimensional magnetic recording (TDMR) channel of which areal density (AD) approaching 5.62 Tb/in^2 . Although the performances of three proposed decoding are worse than that of the belief propagation (BP) algorithm. The coding gains are about 0.5, 1.3 and 1.5 dB worse than the belief propagation (BP) algorithm on two dimensional magnetic recording (TDMR) channel. However, it greatly reduces the number of operations in hardware circuit, such as the multiplication and division operation by 100 and 90 percent, respectively.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากความสนับสนุนจากบุคลากรจากทุกภาคส่วน ลำดับแรกขอกราบขอบพระคุณ คุณพ่อคุณแม่ สำหรับกำลังใจและการสนับสนุนในทุกๆ ด้าน

ขอขอบคุณ ศ.ดร. พรชัย ทริพย์นิธิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ช่วยให้คำปรึกษาตลอดงานวิจัย พร้อมทั้งขอขอบคุณ ดร. เวธิต ภาคย์พิสุทธิ์ ที่คอยให้ข้อเสนอแนะต่างๆ แก่งานวิจัยในวิทยานิพนธ์ฉบับนี้ รวมถึงคณาจารย์ทั้งหลายที่คอยชี้แนะและประสิทธิ์ประสาทวิชาแก่ข้าพเจ้า

ขอขอบคุณกองทุนสนับสนุนงานวิจัยแห่งชาติ (พวอ.) และ บริษัทเวสเทิร์นดิจิทัล (ประเทศไทย) จำกัด ที่ช่วยสนับสนุนค่าใช้จ่าย และองค์ความรู้ต่างๆ ในระยะเวลาการทำงานวิจัยนี้

ศิริวิชญ์ กิตติวิชญกุล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 จุดมุ่งหมายและจุดประสงค์ของการศึกษา.....	3
1.3 ขอบเขตการวิจัย.....	4
บทที่ 2 ระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ	
2.1 โครงสร้างภายในฮาร์ดดิสก์ไดรฟ์.....	5
2.2 เทคโนโลยีในการบันทึกข้อมูลเชิงแม่เหล็กบนฮาร์ดดิสก์ไดรฟ์.....	8
2.2.1 แบบจำลองสื่อบันทึก.....	9
2.2.2 กระบวนการเขียน.....	10
2.2.3 กระบวนการอ่าน.....	12
2.3 แบบจำลองช่องสัญญาณในการเขียนอ่านข้อมูลบนฮาร์ดดิสก์ไดรฟ์.....	16
2.3.1 แบบจำลองช่องสัญญาณเสมือนจริง.....	16
2.3.2 แบบจำลองช่องสัญญาณอุดมคติ.....	18
บทที่ 3 รหัสพาริตีเช็คความหนาแน่นต่ำ	
3.1 รหัสบล็อกเชิงเส้น.....	20
3.2 รหัสแอลดีพีซี.....	24
3.2.1 กราฟแทนเนอร์.....	25
3.2.2 ประเภทของรหัสแอลดีพีซี.....	27
3.2.2.1 รหัสแอลดีพีซีแบบคงที่.....	27
3.2.2.2 รหัสแอลดีพีซีแบบไม่คงที่.....	27
3.2.3 การออกแบบรหัสแอลดีพีซี.....	28
3.3 อัลกอริทึมการถอดรหัสแอลดีพีซี.....	31
3.3.1 การถอดรหัสแอลดีพีซีชนิดอาศัยค่าซอฟต์แวร์.....	31
3.3.1.1 อัลกอริทึมซัมโปรดัคส์.....	31
3.3.1.2 อัลกอริทึมแบบบล็อกซัมโปรดัคส์.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.3 อัลกอริทึมแบบมินซิม	37
3.3.2 การถอดรหัสแอลดีพีซีแบบไม่อาศัยค่าซอฟต์แวร์	37
3.3.2.1 อัลกอริทึมบิตฟลิปปีง	38
3.3.2.2 อัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนัก	40
3.3.2.3 อัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบดัดแปลง	44
3.3.2.4 อัลกอริทึมปรับปรุงบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบดัดแปลง	44
3.3.2.5 อัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนัก แบบอัตราส่วนความน่าเชื่อถือ	46
บทที่ 4 รหัสแอลดีพีซีความซับซ้อนต่ำ	
4.1 รหัสแอลดีพีซีแบบโปรดักส์	48
4.1.1 ความซับซ้อนของรหัสแอลดีพีซีแบบโปรดักส์	51
4.1.1.1 ความซับซ้อนในการเข้ารหัสแอลดีพีซีแบบโปรดักส์	51
4.1.1.2 ความซับซ้อนในการถอดรหัสแอลดีพีซีแบบโปรดักส์	52
4.2 การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบเพจ	53
4.3 การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบแถวสลับหลัก	57
4.4 การถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบดัดแปลงแถวสลับหลัก	61
4.5 ปัญหาสมรรถนะไม่ลู่อู่เข้าที่การวนซ้ำรอบสูง	64
4.6 ความซับซ้อนในการถอดรหัส	69
4.7 การฟลิปหลายบิตเพื่อเพิ่มความเร็วของอัลกอริทึม P-WBF	70
4.7.1 การเลือกหลายบิตแบบคงที่สำหรับอัลกอริทึมการถอดรหัสบิตฟลิปปีง ชนิดถ่วงน้ำหนักแบบเพจ	71
4.7.2 การเลือกหลายบิตแบบเทอร์สโพลด์สำหรับอัลกอริทึมการถอดรหัส บิตฟลิปปีงชนิดถ่วงน้ำหนักแบบเพจ	71
บทที่ 5 การทดสอบสมรรถนะของระบบ	
5.1 ผลการปรับแก้สมรรถนะไม่ลู่อู่เข้า (V-shape) ที่การวนซ้ำรอบสูง	73
5.2 การจำลองระบบที่ช่องสัญญาณรบกวนเกาส์สีขาวแบบบวก	74
5.2.1 สมรรถนะของรหัสแอลดีพีซีที่อัตรารหัสปานกลาง	76
5.2.2 สมรรถนะของรหัสแอลดีพีซีที่อัตรารหัสสูง	78
5.3 สมรรถนะรหัสแอลดีพีซีบนระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ	79
5.4 การฟลิปหลายบิตสำหรับอัลกอริทึมการถอดรหัสแบบเพจ	83
5.4.1 การหาเทอร์สโพลด์สำหรับการเลือกหลายบิตสำหรับอัลกอริทึม PT-WBF	84
5.4.2 การฟลิปหลายบิตสำหรับอัลกอริทึม P-WBF บนช่องสัญญาณรบกวน เกาส์สีขาวแบบบวก	85
5.4.3 การฟลิปหลายบิตสำหรับอัลกอริทึม P-WBF บนช่องสัญญาณบันทึกเชิง แม่เหล็กสองมิติ	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ	89
บรรณานุกรม.....	91
ภาคผนวก	
ผลงานที่ได้รับการตีพิมพ์.....	95
ประวัติผู้เขียน	96



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างทาร์เก็ตแบบพ็อดที่ใช้ในช่องสัญญาณบันทึกข้อมูลเชิงแม่เหล็กแนวตั้ง และแนวนอน.....	19
4.1 ค่าเทรสโพลด์ที่เหมาะสมสำหรับ RC-WBF และ MRC-WBF บนช่องสัญญาณ AWGN.....	67
4.2 ค่าเทรสโพลด์ที่เหมาะสมสำหรับ RC-WBF และ MRC-WBF บนช่องสัญญาณ TDMA 68	68
4.3 ความซับซ้อนของการถอดรหัส สำหรับการวนซ้ำ 1 รอบ ของอัลกอริทึม PCB, PWBF, RC-WBF และ MRC-WBF.....	69
4.4 ค่าเทรสโพลด์ที่เหมาะสมสำหรับการคัดเลือกหลายบิดมาพลิกด้วยอัลกอริทึม PT-WBF สำหรับอัตรารหัส 0.5 และ 0.7.....	71
4.5 ความซับซ้อนของการถอดรหัส สำหรับการวนซ้ำ 1 รอบ ของอัลกอริทึม PF-WBF และ PT-WBF	72
5.1 พารามิเตอร์ในการสร้างรหัสสำหรับอัตรารหัสปานกลาง.....	77
5.2 พารามิเตอร์ในการสร้างรหัสสำหรับอัตรารหัสสูง	78
5.3 พารามิเตอร์ในการออกแบบช่องสัญญาณฮาร์ดดิสก์ไครฟ์แบบ 2 มิติ ตาม S. Kobayashi....	81
5.4 พารามิเตอร์ในการสร้างรหัสสำหรับอัตรารหัสสูงสำหรับช่องสัญญาณฮาร์ดดิสก์ไครฟ์แบบ 2 มิติ.....	81

สารบัญรูป

รูปที่	หน้า
1.1	วงจรถ่างๆ ที่ใช้สำหรับการประมวลผลสัญญาณฮาร์ดดิสก์ไดรฟ์2
2.1	โครงสร้างภายในและอุปกรณ์ต่างๆในฮาร์ดดิสก์ไดรฟ์.....6
2.2	วัสดุแกรนูลาร์ (granular) ที่ถูกนำมาสร้างเป็นแผ่นแพลิตเตอร์ไดรฟ์9
2.3	เกรนขนาดต่างๆ ที่ปรากฏบนวัสดุแกรนูลาร์.....9
2.4	ตัวอย่างการจัดวางเกรนขนาด 1×1 (ก), 1×2 (ข), 2×1 (ค) และ 2×2 (ง) ลงบนแผ่นมีเดีย .. 10
2.5	พิกเซลขนาด 4×3 ที่ถูกใช้สำหรับบันทึกข้อมูล 1 บิต 11
2.6	การบันทึกบิตข้อมูล [1 0; 0 1] ลงไปยังแผ่นแพลิตเตอร์ไดรฟ์ขนาด 48 พิกเซล..... 11
2.7	สภาพแม่เหล็กสำหรับบันทึกข้อมูล [1 0 ; 0 1]..... 12
2.8	โครงสร้างของหัวอ่าน (ก) และแผ่นบันทึกหรือแผ่นแพลิตเตอร์ไดรฟ์ (ข) 13
2.9	ผลตอบสนองในรูปแบบสัญญาณพัลส์สามมิติของค่าความไวในการอ่าน 15
2.10	สัญญาณอ่านกลับ จากช่องสัญญาณ TDMR โดยใช้พารามิเตอร์ของ S. Kobayashi..... 15
2.11	แบบจำลองช่องสัญญาณเสมือนจริง..... 17
2.12	แบบจำลองช่องสัญญาณอุดมคติ..... 18
3.1	การเข้ารหัสบล็อกเชิงเส้น 21
3.2	รูปแบบของคำรหัสเชิงระบบ..... 22
3.3	วงจรรหัสบล็อกเชิงเส้น..... 24
3.4	กราฟแทนเนอร์ของเมทริกซ์พาริตีเช็ค จากสมการที่ 3.10..... 26
3.5	การแผ่กิ่งก้านสาขาของโนดบิต v_i 29
3.6	การส่งผ่านค่าซอฟต์แวร์จากโนดเช็คที่ 1 มายังโนดบิตที่ 8..... 32
3.7	การส่งผ่านค่าซอฟต์แวร์จากโนดบิตที่ 3 ขึ้นไปยังโนดเช็คที่ 1 33
3.8	การถอดรหัสแอลดีพีซีด้วยอัลกอริทึมบิตฟลิปบั้งที่บิตที่ 1..... 38
3.9	การถอดรหัสแอลดีพีซีด้วยอัลกอริทึมบิตฟลิปบั้งที่บิตที่ 7 39
3.10	การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 1 ในกระบวนการประมวลผลโนดเช็ค..... 41
3.11	การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 1 ในกระบวนการประมวลผลโนดบิต..... 42
3.12	การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 7 ในกระบวนการประมวลผลโนดเช็ค..... 45
3.13	การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 7 ในกระบวนการประมวลผลโนดเช็ค..... 46
4.1	โครงสร้างของรหัสแอลดีพีซีแบบโปรดักส์ 49
4.2	แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึมแอลดีพีซีแบบโปรดักส์ 50
4.3	แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม P-WBF 54
4.4	การเชื่อมต่อของกราฟแทนเนอร์กับบิตข้อมูลแบบ 2 มิติ สำหรับ P-WBF 55
4.5	แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม RC-WBF 58
4.6	การเชื่อมต่อของกราฟแทนเนอร์กับบิตข้อมูลสำหรับถอดรหัสใน (ก) แถว, (ข) หลัก 59
4.7	แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม MRC-WBF..... 61
4.8	ความสามารถในการแก้ไขข้อผิดพลาดของอัลกอริทึม RC-WBF 64

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากผู้อนุญาต

4.9	ปัญหาสมรรถนะไม่ลู่เข้า ที่การวนซ้ำภายในและนอกรอบสูงของ RC-WBF.....	65
4.10	การแพร่กระจายของค่า E_{ij} ที่การวนซ้ำภายในรอบที่ (ก) 0, (ข) 18 และ (ค)	66
5.1	แบบจำลองช่องสัญญาณอุดมคติ.....	74
5.2	สมรรถนะของ RC-WBF หลังจากถูกแก้ไขด้วยค่าเทรชโฮลด์ (Threshold).....	74
5.3	สมรรถนะของ MRC-WBF หลังจากถูกแก้ไขด้วยค่าเทรชโฮลด์ (Threshold)	74
5.4	แบบจำลองระบบที่มีการรบกวนจากสัญญาณรบกวนเกาส์สีขาว.....	75
5.5	สมรรถนะของ P-WBF, RC-WBF, MRC-WBF และ PCB สำหรับ R^{2D} เท่ากับ 0.5	77
5.6	สมรรถนะของ P-WBF, RC-WBF, MRC-WBF และ PCB สำหรับ R^{2D} เท่ากับ 0.7	79
5.7	แบบจำลองภาคส่ง และภาครับของช่องสัญญาณระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ	80
5.8	สมรรถนะของอัลกอริทึม P-WBF, RC-WBF, MRC-WBF และ PCB บนช่อง สัญญาณการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR).....	82
5.9	สมรรถนะของอัลกอริทึม P-WBF ที่การวนซ้ำรอบต่างๆ	83
5.10	ค่าเทรชโฮลด์ที่เหมาะสมในการเลือกฟิลิปหลายบิตสำหรับ PT-WBF บน AWGN.....	84
5.11	ค่าเทรชโฮลด์ที่เหมาะสมในการเลือกฟิลิปหลายบิตสำหรับ PT-WBF บน TDMR.....	84
5.12	สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบต่ำ (AWGN)	85
5.13	สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบสูง (AWGN).....	86
5.14	สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบต่ำ (TDMR)	87
5.15	สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบสูง (TDMR)	87

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

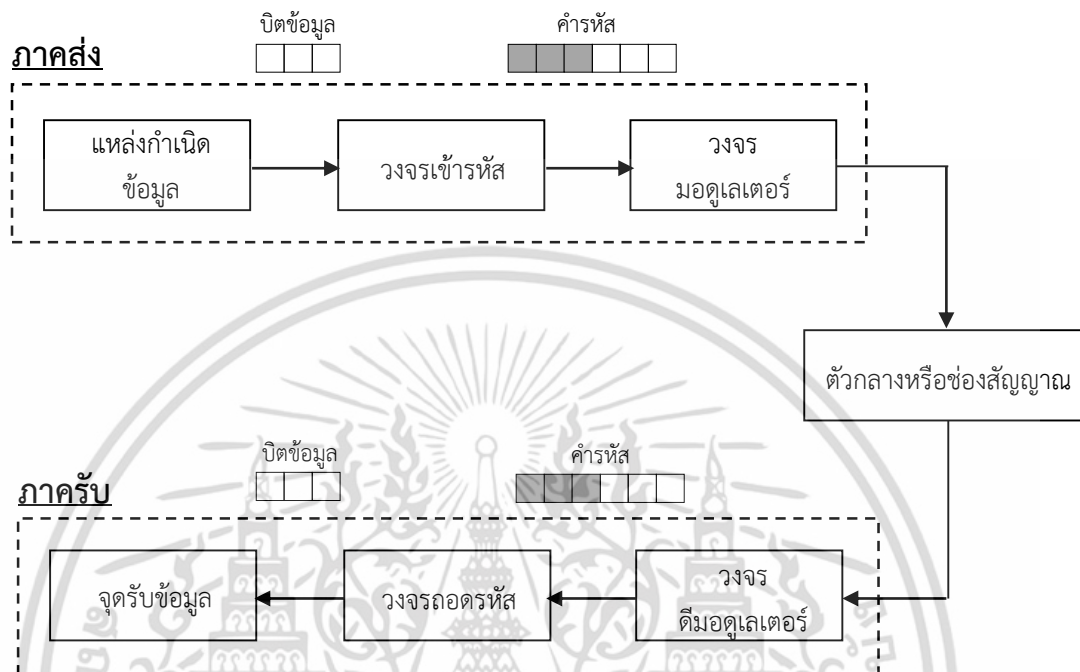
อุปกรณ์ในการบันทึกข้อมูลดิจิทัล ยกตัวอย่างเช่น แฟลชไดรฟ์ (Flash drive) , ฮาร์ดดิสก์ไดรฟ์ (Hard disk drive: HDD) หรือ โซลิดสเตตไดรฟ์ (Solid state drive: SSD) ล้วนถูกใช้กันอย่างแพร่หลายในปัจจุบัน ส่งผลให้ในภาคอุตสาหกรรมมีการแข่งขันกันอย่างดุเดือด ทั้งในด้านราคา และความจุ เพื่อดึงดูดการตัดสินใจเลือกซื้อของผู้บริโภค อย่างไรก็ตาม นอกเหนือจากปัจจัยทั้งสองข้างต้นแล้ว ประสิทธิภาพในการจัดเก็บข้อมูลก็เป็นอีกหนึ่งปัจจัย ที่สำคัญต่อการตัดสินใจเลือกซื้อของผู้บริโภคด้วยเช่นกัน ในระบบบันทึกข้อมูลดิจิทัลนั้น จะแตกต่างจากระบบสื่อสารดิจิทัลทั่วไปคือ เมื่อภาครับมีการรับข้อมูลที่มีความผิดพลาด ภาครับจะไม่สามารถร้องขอให้ภาคส่งทำการส่งข้อมูลมาอีกครั้งได้ ดังนั้นในภาครับจึงต้องมีวิธีการหรืออัลกอริทึมต่างๆ ที่ใช้สำหรับแก้ไขความผิดพลาดด้วยตัวของมันเอง ดังเช่นในวิทยานิพนธ์เล่มนี้ ผู้วิจัยนำเสนอถึงวิธีการแก้ไขความผิดพลาดของข้อมูลที่ภาครับสำหรับระบบบันทึกข้อมูลเชิงแม่เหล็ก หรือฮาร์ดดิสก์ไดรฟ์ ซึ่งฮาร์ดดิสก์ไดรฟ์ เป็นอุปกรณ์ในการจัดเก็บข้อมูลดิจิทัล ที่มีการใช้งานจำนวนมากที่สุดในปัจจุบัน อีกทั้งในประเทศไทยยังเป็นการผลิต และส่งออกฮาร์ดดิสก์ไดรฟ์รายใหญ่ที่สุดในโลก โดยมีมูลค่าในการส่งออกถึง 6 เปอร์เซ็นต์ [34] ของมูลค่าการส่งออกทั้งหมดต่อปี พร้อมทั้งมีการจ้างงานในภาคอุตสาหกรรมถึง 300,000 อัตรา ด้วยเหตุผลดังกล่าวนี้ จึงดึงดูดให้ผู้วิจัยทำการค้นคว้าและวิจัย ทั้งในส่วนของ การเพิ่มปริมาณความจุ และประสิทธิภาพในการแก้ไขความผิดพลาดในฮาร์ดดิสก์ไดรฟ์ควบคู่ไปด้วยกัน เพื่อเป็นแรงสนับสนุนองค์ความรู้ใหม่ๆ แก่ภาคอุตสาหกรรมทางด้านฮาร์ดดิสก์ไดรฟ์ในประเทศไทย

ในวิทยานิพนธ์เล่มนี้ จะมีการกล่าวถึงกระบวนการประมวลผลสัญญาณ (Signal processing) ที่ภาครับของฮาร์ดดิสก์ไดรฟ์ ซึ่งเป็นส่วนที่สำคัญอย่างยิ่ง ในการกู้คืนข้อมูลที่ถูกรบกวน และแก้ไขความผิดพลาดของข้อมูลที่เกิดจากความผิดพลาดต่างๆภายในฮาร์ดดิสก์ไดรฟ์ ยกตัวอย่างเช่น ความร้อนจากวงจรอิเล็กทรอนิกส์ ซึ่งนับเป็นสาเหตุหลักที่ก่อให้เกิดความผิดพลาดของข้อมูล กระบวนการประมวลผลสัญญาณที่ภาครับของฮาร์ดดิสก์ไดรฟ์นั้น สามารถพิจารณาได้ว่าเป็นระบบสื่อสารดิจิทัลประเภทหนึ่ง ที่มีวงจรในภาคส่งและภาครับหลักๆ ดังแสดงในรูปที่ 1.1 เมื่อเราต้องการบันทึกข้อมูลจำนวนหนึ่งให้ปราศจากความผิดพลาดหรือเกิดขึ้นน้อยที่สุดนั้น จำเป็นต้องอาศัยวงจรเข้ารหัส (Encoder) และวงจรถอดรหัส (Decoder) มาช่วยในกระบวนการประมวลผลสัญญาณ โดยที่ภาคส่งนั้น เมื่อนำข้อมูลที่ต้องการบันทึกมาทำการเข้ารหัส (Encoding) เราจะได้ข้อมูลชุดใหม่ที่มีความยาวมากกว่าเดิม โดยเราจะเรียกข้อมูลชุดใหม่นั้นว่า คำรหัส (Codeword) โดยบิตที่ถูกเพิ่มเข้ามานั้น จะทำหน้าที่ในการตรวจจับ และแก้ไขความผิดพลาดของข้อมูลชุดนั้น โดยเราจะเรียกบิตที่ถูกเพิ่มเข้ามาว่า บิตพาริตี (Parity) จากนั้นคำรหัสจะถูกนำไปป้อนเข้าสู่วงจรมอดูเลเตอร์ (Modulator) ซึ่งมีหน้าที่ปรับสภาพคำรหัสให้เหมาะสมกับตัวกลาง (Medium) หรือช่องสัญญาณ (Channel) อย่างไรก็ตาม หากเป็นช่องสัญญาณของฮาร์ดดิสก์ไดรฟ์ คำรหัสจะถูกปรับสภาพให้อยู่ในรูปแบบกระแสไฟฟ้า

เขียน (Write current waveform) ก่อนจะถูกป้อนไปยังหัวเขียน เพื่อทำการเขียนข้อมูลลงไปยังแผ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีเดีย (Media drive) ต่อไป ในส่วนภาครับนั้น วงจรดีมอดูเลเตอร์ (Demodulator) จะทำหน้าที่ตรงกันข้ามกับวงจรมอดูเลเตอร์ โดยจะทำการแยกข้อมูลรหัสออกจากสัญญาณที่ได้รับ จากนั้นข้อมูลรหัสนี้จะถูกนำไปป้อนเข้าสู่วงจรถอดรหัส (Decoder) เพื่อทำการตรวจสอบและแก้ไขข้อผิดพลาดด้วยเทคนิคต่อไปนี้



รูปที่ 1.1 แผนภาพบล็อกระบบการประมวลผลสัญญาณฮาร์ดดิสก์ไทรฟ์

นอกเหนือจากปัญหาความร้อนจากวงจรอิเล็กทรอนิกส์แล้วนั้น ยังมีอีกหนึ่งสาเหตุที่ทำให้ข้อมูลที่บันทึกเกิดความผิดพลาดได้ นั่นคือ ปัญหาจากความไม่แน่นอนของสภาพแม่เหล็กในการบันทึก ยกตัวอย่างเช่น หากเราทำการอ่านบิตข้อมูลที่บันทึกด้วยความเร็วสูงจนเกินไป หรือทำการบีบอัดข้อมูลให้มีความหนาแน่นจนเกินไป ซึ่งถึงแม้จะทำให้ฮาร์ดดิสก์ไทรฟ์มีความเร็ว และปริมาณความจุที่เพิ่มมากขึ้น แต่ต้องแลกกับโอกาสในการเกิดการแทรกสอดจากสภาพแม่เหล็กจากบิตข้างเคียง (Inter-symbol interference: ISI) หรือจากแทร็คข้างเคียง (Inter-track interference: ITI) ซึ่งปัญหาดังกล่าวมักจะเกิดกับฮาร์ดดิสก์ไทรฟ์ที่มีความจุสูง (ฮาร์ดดิสก์ไทรฟ์ส่วนใหญ่ในปัจจุบัน) ดังนั้นในทางปฏิบัติ จำเป็นจะต้องเพิ่มวงจรอีควอลไลเซอร์ (Equalizer) และวงจรถอดรหัส (Detector) ที่ภาครับ เพื่อแก้ไขความผิดพลาดจากปัญหา ISI และ ITI ตามลำดับ

เป็นที่ทราบกันดีแล้วว่า ความผิดพลาดของข้อมูลที่ถูกบันทึกบนฮาร์ดดิสก์ไทรฟ์นั้น จะถูกแก้ไขด้วยวงจรเข้ารหัส และถอดรหัสตามลำดับ อย่างไรก็ตามวิธีการในการออกแบบหรือการเลือกใช้วงจรเข้ารหัสและถอดรหัสให้มีความเหมาะสมนั้น ก็เป็นอีกหนึ่งตัวแปรที่ส่งผลต่อประสิทธิภาพในการแก้ไขความผิดพลาดของข้อมูลด้วยเช่นกัน รหัสช่องสัญญาณ (Channel coding) หรือรหัสแก้ไขความผิดพลาด (Error correction) เป็นเทคนิควิธีการเข้ารหัสและถอดรหัส ที่ถูกใช้กันอย่างแพร่หลายในปัจจุบัน ซึ่งสามารถจำแนกออกเป็นสองประเภทหลักๆ ได้แก่ รหัสคอนโวลูชัน (Convolution code) และรหัสบล็อก (Block code) ความแตกต่างอย่างเห็นได้ชัดของรหัสทั้งสองประเภทนี้คือ รหัสคอนโวลูชันจะ

ทำการเข้ารหัสข้อมูลทีละบิต ซึ่งบิตที่ได้จากการเข้ารหัสคอนโวลูชันจะมีความสัมพันธ์กับบิตในเวลา
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาและวิจัยเท่านั้น ไม่สามารถนำออกจำหน่าย
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นกับบิตก่อนหน้าเสมอ ตัวอย่างของการประยุกต์ใช้รหัสรหัสคอนวอลูชันคือ รหัสเทอร์โบ (Turbo code) [1] ซึ่งเป็นรหัสคอนวอลูชันชนิดหนึ่งที่มีชื่อเสียง และรู้จักกันในหมู่นักวิจัยทั่วโลก ตรงกันข้ามกับรหัสบล็อก ที่จะทำการเข้ารหัสข้อมูลที่ละบล็อก โดยบิตรหัสที่ได้นั้น จะไม่ขึ้นกับบิตข้อมูลจากบล็อกอื่น (รายละเอียดเพิ่มเติมจะกล่าวถึงในบทที่ 3) ตัวอย่างของการประยุกต์ใช้รหัสรหัสบล็อกคือ รหัสพาริตีเช็คความหนาแน่นต่ำ (Low density parity check codes: LDPC) [2] หรือรหัสแอลดีพีซี อย่างไรก็ตาม ถึงแม้รหัสแอลดีพีซีจะถือกำเนิดมาก่อนรหัสเทอร์โบ แต่ในสมัยก่อนนั้นยังไม่ได้ได้รับความนิยมเท่าที่ควร เนื่องจากข้อจำกัดทางฮาร์ดแวร์ และซอฟต์แวร์ ที่ไม่สามารถพิจารณาได้ถึงความสามารถหรือขีดจำกัดในการแก้ไขความผิดพลาดของรหัสแอลดีพีซีได้ รวมถึงรหัสเทอร์โบได้รับการพิสูจน์ว่ามีความสามารถในการแก้ไขความผิดพลาดเข้าใกล้ขีดจำกัดของแชนนอน (Shannon's limit) [3] จนกระทั่งในปี 1996 [4] รหัสแอลดีพีซีจึงได้รับการพิสูจน์ว่ามีความสามารถในการแก้ไขความผิดพลาดของข้อมูลที่ดีกว่ารหัสเทอร์โบ หากบล็อกข้อมูลที่ทำการเข้ารหัสมีความยาวมาก ฉะนั้นในปัจจุบัน รหัสแอลดีพีซีจึงได้รับความนิยมในหมู่นักวิจัยทั่วโลก (80% ของงานวิจัยที่เกี่ยวกับรหัสช่องสัญญาณ ล้วนมีความเกี่ยวข้องกับรหัสแอลดีพีซีทั้งสิ้น) ฉะนั้น รหัสแอลดีพีซีจึงถูกนำมาใช้กับวงจรรหัส สำหรับฮาร์ดดิสก์ไดรฟ์ในปัจจุบัน [5 - 8]

1.2 จุดมุ่งหมายและจุดประสงค์ของการศึกษา

ในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจะทำการศึกษารหัสเข้ารหัส และถอดรหัสแอลดีพีซีสำหรับบล็อกข้อมูลที่มีความยาวมาก ซึ่งการใช้รหัสแอลดีพีซีสำหรับบล็อกข้อมูลความยาวมากนั้น จะมีความสามารถในการแก้ไขความผิดพลาดที่ดีกว่าการใช้รหัสเทอร์โบ อย่างไรก็ตาม การเข้ารหัสและถอดรหัสแอลดีพีซีสำหรับบล็อกข้อมูลความยาวมาก จะเป็นการเพิ่มความซับซ้อนของวงจรรหัสฮาร์ดแวร์ภายในฮาร์ดดิสก์ไดรฟ์ ซึ่งจะส่งผลต่อต้นทุนการผลิต และทำให้ฮาร์ดดิสก์ไดรฟ์มีราคาที่สูงตามไปด้วย ดังนั้นในวิทยานิพนธ์เล่มนี้ จึงมุ่งเน้นไปที่การพัฒนาประสิทธิภาพในการแก้ไขความผิดพลาดของข้อมูล และลดความซับซ้อนของวงจรรหัสและถอดรหัสภายในฮาร์ดดิสก์ไดรฟ์ไปด้วยกัน เมื่อไม่นานมานี้ มีงานวิจัยที่ได้ค้นพบวิธีการลดความซับซ้อนวงจรรหัสและถอดรหัสแอลดีพีซีคือ การจัดเรียงบล็อกข้อมูลความยาวมากให้อยู่ในรูปแบบสองมิติ (Two-dimensional: 2D) [9] ก่อนกระทำการเข้ารหัส รวมทั้งงานวิจัย [10 - 15] ได้คิดค้นวิธีการถอดรหัสแอลดีพีซีแบบหยาบ (Hard decision) ซึ่งผลของทั้งสองวิธีการดังกล่าวข้างต้นนั้น จะสามารถลดความซับซ้อนวงจรรหัส และถอดรหัสแอลดีพีซีได้ (ทั้งสองวิธีการนั้นจะถูกอธิบายอย่างละเอียดในบทที่ 4) ดังนั้น ผู้วิจัยจึงได้นำสองวิธีการดังกล่าวข้างต้นมาประยุกต์ใช้ร่วมกัน เพื่อที่จะสามารถถอดรหัสแอลดีพีซีสำหรับบล็อกข้อมูลความยาวมาก และสามารถลดความซับซ้อนในการเข้ารหัสและถอดรหัสแอลดีพีซีไปพร้อมๆกัน วิทยานิพนธ์เล่มนี้นำเสนอ 3 วิธีการด้วยกันคือ วิธีการถอดรหัสแบบบิตพลิกบิตบ่งชี้ถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping: P-WBF), วิธีการถอดรหัสแบบบิตพลิกบิตบ่งชี้ถ่วงน้ำหนักแบบแถวสลับหลัก (Row-column weighted bit-flipping: RC-WBF) และวิธีการถอดรหัสแบบบิตพลิกบิตบ่งชี้ถ่วงน้ำหนักแบบดัดแปลงแถวสลับหลัก (Modified row-column weighted bit-flipping: MRC-WBF) ตามลำดับ อย่างไรก็ตามทั้ง 3 วิธีการที่นำเสนอในวิทยานิพนธ์เล่มนี้ จะมีปัญหาบางประการเกิดขึ้นในวงจรรหัส เช่น ปัญหาสมรรถนะไม่สูงเท่าที่การวนซ้ำรอบสูง และปัญหาในด้านความเร็วสู่สมรรถนะสูง (ทั้งสองปัญหานี้

จะถูกอธิบายอย่างละเอียดในบทที่ 4) โดยทั้งสองปัญหาที่เกิดขึ้น จะได้รับการแก้ไขและปรับปรุงในวิทยานิพนธ์เล่มนี้ด้วยเช่นกัน

1.3 ขอบเขตการวิจัย

ในวิทยานิพนธ์เล่มนี้ จะทำการศึกษาถึงสมรรถนะในการแก้ไขบิตผิดพลาด และความซับซ้อนวงจรเข้าและถอดรหัสแอลดีพีซีของวิธีการทั้ง 3 ที่นำเสนอ ภายใต้ช่องสัญญาณรบกวนแบบเกาส์สีขาว (ปราศจาก ISI และ ITI) และภายใต้ช่องสัญญาณการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ ที่มีความหนาแน่นเชิงพื้นที่เท่ากับ 5.62 เทระบิตต่อตารางนิ้ว ซึ่งมีการแทรกสอดของสภาพแม่เหล็กจากบิตข้างเคียง (ISI) และจากแทร็กข้างเคียง (ITI) ตามลำดับ ผู้วิจัยจะทำการกำหนดให้บล็อกข้อมูลมีความยาวมาก โดยมีความยาวมากกว่า 100,000 บิต โดยจะทำการจำลองผ่านโปรแกรม Matlab และ Visual studio บนคอมพิวเตอร์พีซี

วิทยานิพนธ์เล่มนี้ จะประกอบไปด้วยส่วนต่างๆ 6 บท โดยในบทแรก จะกล่าวถึงความสำคัญ และวัตถุประสงค์ของวิทยานิพนธ์เล่มนี้ จากนั้นในบทที่ 2 จะกล่าวถึงทฤษฎี และแบบจำลองของระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ ที่ถูกนำมาใช้สำหรับการจำลองสมรรถนะของรหัสแอลดีพีซี ในบทที่ 3 จะกล่าวถึงทฤษฎีการเข้ารหัสช่องสัญญาณชนิดรหัสบล็อกเชิงเส้น พร้อมทั้งกล่าวถึงวิธีการเข้าและถอดรหัสแอลดีพีซีตามลำดับ ส่วนวิธีการลดความซับซ้อนในการเข้าและถอดรหัสแอลดีพีซี รวมถึงอัลกอริทึมทั้งสามที่นำเสนอ นั้น จะถูกกล่าวถึงในบทที่ 4 ผลการจำลอง และการวิเคราะห์สมรรถนะของการถอดรหัสแอลดีพีซีที่นำเสนอทั้งสามชนิด ภายใต้ช่องสัญญาณรบกวนเกาส์สีขาว และช่องสัญญาณการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ จะถูกอธิบายในบทที่ 5 ในส่วนสุดท้าย บทที่ 6 จะเป็นการสรุปผลงานวิจัย และข้อเสนอแนะตามลำดับ

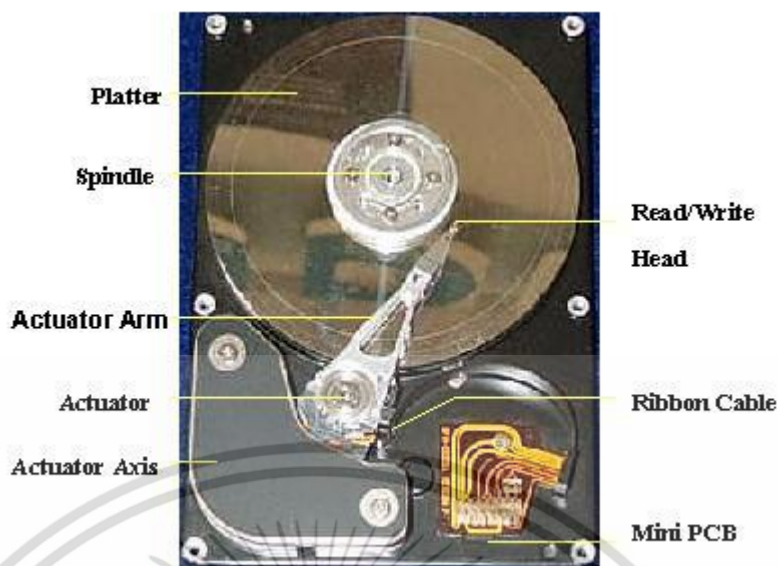
บทที่ 2

ระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR)

การบันทึกข้อมูล หมายถึง การจัดเก็บข้อมูลที่ต้องการลงบนสื่อบันทึก (Media) ชนิดต่างๆ โดยที่เมื่อเวลาผ่านไปแล้ว ยังสามารถเรียกข้อมูลเดิมกลับคืนมา สำหรับขั้นตอนในการบันทึกข้อมูลลงไปยังสื่อบันทึกใดๆ นั้น จะต้องทำการแปลงบิตข้อมูลให้มีสภาพที่เหมาะสมกับสื่อบันทึกชนิดนั้นเสียก่อน โดยที่ในวิทยานิพนธ์เล่มนี้ จะกล่าวถึงเฉพาะระบบบันทึกข้อมูลเชิงแม่เหล็ก หรืออุปกรณ์จำพวกฮาร์ดดิสก์ไดรฟ์ (Hard disk drive) ซึ่งมีสื่อบันทึกอยู่ในรูปแบบจานอะลูมิเนียมที่เคลือบด้วยสารแม่เหล็ก ดังนั้น บิตข้อมูลที่จะทำการบันทึกลงไปยังฮาร์ดดิสก์ไดรฟ์นั้น จะต้องถูกแปลงให้อยู่ในสภาพความเป็นแม่เหล็ก (Magnetization) เพื่อที่จะสามารถบันทึกลงไปยังสื่อบันทึกของฮาร์ดดิสก์ไดรฟ์ได้อย่างไรก็ตาม การบันทึกข้อมูลลงไปยังสื่อบันทึกที่มีสภาพความเป็นแม่เหล็กดังเช่นฮาร์ดดิสก์ไดรฟ์นี้เหมาะแก่การบันทึกข้อมูลดิจิทัลสองสถานะเช่น บิต 0 และ 1 หรือที่เรียกกันว่าข้อมูลแบบไบนารี (Binary) เนื่องจากสื่อบันทึกของฮาร์ดดิสก์ไดรฟ์นั้น จะอาศัยคุณสมบัติความเป็นแม่เหล็ก ที่เมื่อถึงจุดอิ่มตัวแล้วนั้น จะมีสภาพความเป็นแม่เหล็กมีทิศทางกันข้ามกับทิศทางตั้งต้น โดยในบทที่สองนี้ ลำดับแรกจะกล่าวถึงโครงสร้าง และอุปกรณ์ปลั๊กย่อยภายในตัวฮาร์ดดิสก์ไดรฟ์ จากนั้นจะกล่าวถึงวิธีการเขียนหรือการบันทึกข้อมูลไบนารี ลงไปยังสื่อบันทึกของฮาร์ดดิสก์ไดรฟ์และการอ่านกลับ ตามลำดับ โดยในที่สุดท้ายจะกล่าวถึงแบบจำลองที่ใช้ในการทดสอบสมรรถนะของระบบที่ทำการออกแบบในวิทยานิพนธ์เล่มนี้

2.1 โครงสร้างภายในฮาร์ดดิสก์ไดรฟ์

ภายในฮาร์ดดิสก์ไดรฟ์นั้น ประกอบไปด้วยอุปกรณ์ต่างๆ ที่มีหน้าที่การทำงานที่แตกต่างกัน โดยอุปกรณ์ต่างๆ เหล่านั้น จะถูกปกปิดด้วยกล่องบรรจุที่ปิดผนึกแน่นเป็นสุญญากาศ เพื่อป้องกันฝุ่นละอองหรือสิ่งรบกวนใดๆ จากภายนอก ทำให้ไม่สามารถหลุดลอยเข้าไปได้ อุปกรณ์ต่างๆ ภายในฮาร์ดดิสก์ไดรฟ์ ดังแสดงในรูป 2.1



รูปที่ 2.1 โครงสร้างภายในและอุปกรณ์ต่างๆในฮาร์ดดิสก์ไดรฟ์ [33]

1) สื่อบันทึก (Media/Platter drive)

สื่อบันทึกมีลักษณะเป็นแผ่นกลมๆ ทำด้วยโลหะผสม (Metal alloy disc) ที่เคลือบไปด้วยสารแม่เหล็กจำพวก Aluminum alloy หรือ Glass substrate ลักษณะของสื่อบันทึกนั้น มีความกลมและแบน ฉะนั้นจึงถูกเรียกอีกชื่อหนึ่งว่า แผ่นแพลิตเตอร์ (Platter) แผ่นดังกล่าวมีหน้าที่สำคัญคือ ทำการจัดเก็บข้อมูลบิต 0 และ 1 ซึ่งโดยทั่วไปแล้วนั้น แผ่นแพลิตเตอร์จะมีขนาดเส้นผ่าศูนย์กลางประมาณ 3.5 และ 5.25 นิ้ว เป็นต้น ซึ่งในฮาร์ดดิสก์ไดรฟ์แต่ละตัว จะมีจำนวนแผ่นแพลิตเตอร์ไม่เท่ากัน โดยจะขึ้นอยู่กับปริมาณความจุของฮาร์ดดิสก์ไดรฟ์ตัวนั้น

2) มอเตอร์สปินเดิล (Spindle motor)

สื่อบันทึกหรือแผ่นแพลิตเตอร์แต่ละแผ่น จะถูกวางซ้อนๆ กันบนมอเตอร์ตัวหนึ่ง ที่ถูกเรียกว่า มอเตอร์สปินเดิล โดยมอเตอร์ตัวนี้มีหน้าที่หมุนแผ่นแพลิตเตอร์ด้วยความเร็วรอบคงที่ โดยมาตรฐานในปัจจุบันนั้น จะอยู่ที่ความเร็วระดับ 7200 รอบต่อนาที (Rotations-per-minute : RPM) สำหรับฮาร์ดดิสก์ไดรฟ์บางรุ่นอาจมีความเร็วถึง 10000 - 15000 รอบต่อนาที อย่างไรก็ตาม ยิ่งความเร็วสูงมากขึ้นเพียงใด การที่หัวเขียนหรือหัวอ่านจะค้นพบเซ็กเตอร์เป้าหมายในการเขียนหรืออ่านข้อมูล ยิ่งมีความเร็วมากขึ้นตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Actuator

อุปกรณ์ตัวนี้ มีหน้าที่ในการเคลื่อนแขนของตัวเอง (Actuator arms) โดยที่ปลายของ Actuator จะมีหัวเขียน และหัวอ่านติดตั้งอยู่ ฉะนั้น หน้าที่ของ Actuator คือ ทำการเคลื่อนย้ายหัวเขียน และหัวอ่านไปยังแทร็ก (Track) หรือเซ็กเตอร์ (Sector) ที่ต้องการ ดังนั้น Actuator จะถือเป็นอุปกรณ์ที่ใช้บ่งบอกถึงความเร็วในการเขียนและอ่านข้อมูลของฮาร์ดดิสก์ไดรฟ์แต่ละตัว อย่างไรก็ตาม Actuator ต้องมีความแม่นยำในการทำงาน 100% เสมอ มิฉะนั้นการเขียนและอ่านข้อมูลบนฮาร์ดดิสก์ไดรฟ์จะมีความผิดพลาดเกิดขึ้น ซึ่งฮาร์ดดิสก์ไดรฟ์ตัวนั้นจะไม่สามารถนำมาใช้ได้

4) หัวเขียนและหัวอ่าน (Read/Write head)

หน้าที่ของหัวเขียนและหัวอ่าน มีหน้าที่รับผิดชอบในการเขียนและอ่านข้อมูลบนแผ่นแพลิตเตอร์ (Platter) มันจะถูกติดตั้งอยู่ตรงส่วนปลายของ Actuator ซึ่งหัวอ่านและเขียนนี้ จะถูกติดตั้งไว้ทั้งด้านบนและล่างของแผ่นแพลิตเตอร์แต่ละแผ่น เมื่อฮาร์ดดิสก์ไดรฟ์เริ่มทำงาน มอเตอร์สปินเดิลจะทำหน้าที่ในการหมุนแผ่นแพลิตเตอร์ด้วยความเร็วที่หนืด ซึ่งการหมุนนี้จะทำให้เกิดการไหลเวียนของอากาศ (Air flow) ซึ่งจะช่วยยก Actuator arms ให้สูงขึ้นจากแผ่นแพลิตเตอร์เล็กน้อย (ประมาณ 2 ไมโครเมตร) ประโยชน์ของการยกตัวของ Actuator arms นั้น จำเป็นมากต่อการทำงานของฮาร์ดดิสก์ไดรฟ์ เนื่องจากแผ่นแพลิตเตอร์จะหมุนด้วยความเร็วสูงมาก หากหัวเขียนและอ่านแตะโดนแผ่นแพลิตเตอร์ จะทำให้เกิดรอยขีดข่วนถาวร ซึ่งจะทำให้เซ็กเตอร์นั้น ไม่สามารถจัดเก็บข้อมูลได้อีกต่อไป ในปัจจุบันนี้ ขนาดของหัวเขียนและหัวอ่าน ได้ถูกออกแบบให้มีขนาดเล็กที่สุดเท่าที่จะเป็นไปได้แล้วเพื่อทำให้ประสิทธิภาพในการเขียนและอ่านมีค่าที่ดี

5) แผ่นวงจร (PCB)

ฮาร์ดดิสก์ไดรฟ์แต่ละตัวจะมีแผ่นวงจรที่ถูกยึดติดไว้ภายนอกของตัวฮาร์ดดิสก์ไดรฟ์ ซึ่งแผ่นวงจรนี้จะประกอบไปด้วยชิพไอซีที่เรียกว่าดีเอสพี (Digital signal processors: DSP) ทำหน้าที่ควบคุมการส่งผ่านข้อมูลหรือเก็บข้อมูลในส่วนที่ถูกเรียกใช้บ่อยๆ รวมถึงควบคุมการทำงานของอุปกรณ์ต่างๆ ในฮาร์ดดิสก์ไดรฟ์ เช่น มอเตอร์สปินเดิล, Actuator, หน่วยความจำ, อุปกรณ์ในการประมวลผลการอ่านและเขียน และแหล่งจ่ายไฟ เป็นต้น

6) ส่วนเชื่อมต่อ (Interface)

สำหรับฮาร์ดดิสก์ไดรฟ์ในปัจจุบันนั้น จะใช้การเชื่อมต่อแบบมาตรฐานหรือที่เรียกว่าไอดีอี (Integrated drive electronics: IDE) หรือบางรุ่นอาจจะเป็นเอสซีเอสไอ (Small computer) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

system interfaces: SCSI) โดยการเชื่อมต่อทั้งสองแบบที่กล่าวมานั้น ล้วนเป็นตัวกำหนดถึงแบนด์วิดท์ (Band-width : BW) ในการส่งผ่านข้อมูลไปยังซีพียู เนื่องจากส่วนเชื่อมต่อนี้มีหน้าที่ในการรับส่งข้อมูลไปมาของฮาร์ดดิสก์ไดรฟ์

7) ตัวกรอง (Filter)

อุปกรณ์ตัวกรองหรือฟิลเตอร์นี้ทำหน้าที่ในการกรองฝุ่นละอองไม่ให้เข้าไปยังฮาร์ดดิสก์ไดรฟ์ อีกทั้งยังช่วยในการลดอุณหภูมิภายในตัวของฮาร์ดดิสก์ไดรฟ์ได้อีกด้วย

2.2 เทคโนโลยีในการบันทึกข้อมูลเชิงแม่เหล็กบนฮาร์ดดิสก์ไดรฟ์

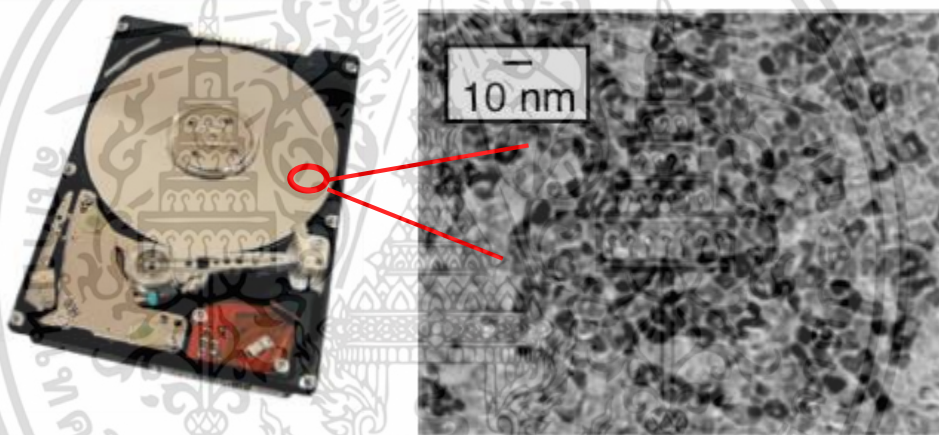
เมื่อไม่นานมานี้ เทคโนโลยีในการบันทึกข้อมูลลงฮาร์ดดิสก์ไดรฟ์นั้น ได้ถูกนำเสนอมาในหลายๆ งานวิจัย เพื่อที่จะทำให้ค่าความหนาแน่นเชิงพื้นที่ (Areal density: AD) มีค่ามากที่สุดเท่าที่จะเป็นไปได้ ยกตัวอย่างเช่น เทคโนโลยีบิตแพทเทิร์นมีเดีย (Bit pattern media: BPM) [16], เทคโนโลยีการบันทึกข้อมูลเชิงแม่เหล็กโดยใช้คลื่นไมโครเวฟ (Microwave assisted magnetic recording: MAMR) [17], เทคโนโลยีการบันทึกข้อมูลเชิงแม่เหล็กโดยการใช้ความร้อน (Heat assisted magnetic recording: HAMR) [18] ทั้งสามเทคโนโลยีข้างต้นนั้น ล้วนเป็นเทคโนโลยีที่สามารถให้ประสิทธิภาพในการบันทึกข้อมูลที่ดี และสามารถเพิ่มความหนาแน่นเชิงพื้นที่ ให้มีค่าในระดับเทระบิตต่อตารางนิ้ว อย่างไรก็ตาม สื่อบันทึก และอุปกรณ์ต่างๆ ที่ใช้สำหรับประมวลผลสัญญาณภายในฮาร์ดดิสก์ไดรฟ์ของสามเทคโนโลยีดังกล่าว จำเป็นจะต้องมีการออกแบบใหม่ทั้งหมด จากเทคโนโลยีที่ใช้ในปัจจุบัน เมื่อไม่นานมานี้ยังมีอีกเทคโนโลยีหนึ่งที่มีความสนใจในปัจจุบันคือ เทคโนโลยีการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (Two-dimensional magnetic recording: TDMR) [19] ที่ถูกนำเสนอโดย R. Wood ในปี 2009 เทคโนโลยีชนิดนี้พยายามลดขนาดพื้นที่ของแผ่นมีเดีย สำหรับการบันทึกข้อมูล 1 บิต ให้มีขนาดเล็กลงมากที่สุดเท่าที่จะเป็นไปได้ เพื่อที่จะทำให้ได้ความหนาแน่นเชิงพื้นที่ หรือความจุของฮาร์ดดิสก์ไดรฟ์มากที่สุดเท่าที่จะเป็นไปได้ เทคโนโลยีชนิดนี้สามารถเพิ่มความหนาแน่นเชิงพื้นที่ได้มากกว่า 10 Tb/in² หากมีกระบวนการประมวลผลสัญญาณ และระบบแก้ไขความผิดพลาดที่ดี ทั้งนี้เทคโนโลยีการบันทึกข้อมูลแบบ TDMR ไม่จำเป็นต้องทำการเปลี่ยนแปลงสื่อบันทึก และอุปกรณ์ภายในฮาร์ดดิสก์ไดรฟ์จากอุปกรณ์ดั้งเดิม

ในวิทยานิพนธ์ฉบับนี้ จะกล่าวถึงเฉพาะเทคโนโลยีการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR) เท่านั้น โดยในส่วนแรกจะกล่าวถึงวิธีในออกแบบแผ่นแพลลิตเตอร์ไดรฟ์ สำหรับการบันทึกข้อมูล หลังจากนั้น จะกล่าวถึงกระบวนการในการเขียนข้อมูลบิต 0 และ 1 ลงไปยังแผ่นแพลลิตเตอร์ไดรฟ์ ส่วนสุดท้าย จะกล่าวถึงขั้นตอนในการอ่านกลับข้อมูลจากแพลลิตเตอร์ไดรฟ์

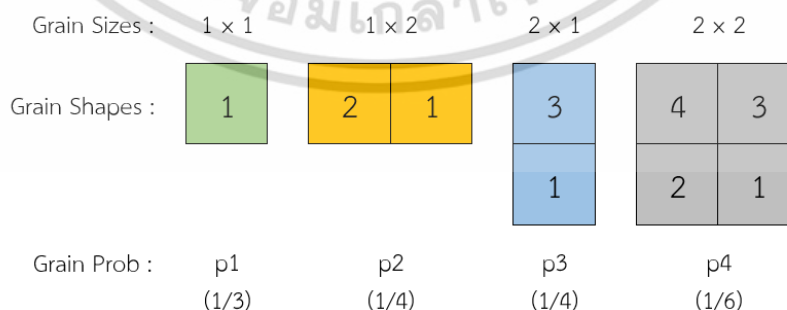
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 แบบจำลองสี่บันทึก (4-Grain media model)

เป็นที่ทราบกันดีว่า บิตข้อมูลที่ต้องการบันทึกลงจะถูกเขียนลงไปยังแผ่นแพลลิตเตอร์ไตร์ฟ โดยแผ่นแพลลิตเตอร์ไตร์ฟที่ใช้กันโดยทั่วไปในภาคอุตสาหกรรมนั้น จะถูกสร้างมาจากวัสดุแบบแกรนูล่า (Granular) ซึ่งถ้าหากมองวัสดุชนิดนี้ในระดับนาโน จะพบว่าประกอบไปด้วยเกรนขนาดเล็กๆ ที่วางเรียงตัวอย่างกระจัดกระจาย โดยมีขนาดของแต่ละเกรนที่ไม่คงที่ ดังแสดงในรูปที่ 2.2 ในงานวิจัย [20] นักวิจัยชื่อ K. San มีเป้าหมายที่จะทำการจำลองวัสดุแกรนูล่าให้อยู่ในรูปแบบอย่างง่าย เพื่อให้ง่ายต่อการนำมาวิเคราะห์กระบวนการเขียน และอ่านของฮาร์ดดิสก์ไตร์ฟ รวมถึงเพื่อที่จะสามารถประเมินสมรรถนะเบื้องต้นของเทคโนโลยี TDMR เมื่อถูกนำมาใช้กับกระบวนการประมวลผลสัญญาณ เขาได้ทำการจำแนกขนาดของเกรนที่ปรากฏบนแผ่นแพลลิตเตอร์ไตร์ฟออกเป็น 4 ขนาด ได้แก่ ขนาด 1×1 , 1×2 , 2×1 , และ 2×2 ตามลำดับ ซึ่งกำหนดให้เกรนแต่ละขนาดนั้น มีความน่าจะเป็นที่จะเกิดขึ้นเท่ากับ $1/3$, $1/4$, $1/4$ และ $1/6$ ตามลำดับ ดังรูปที่ 2.3



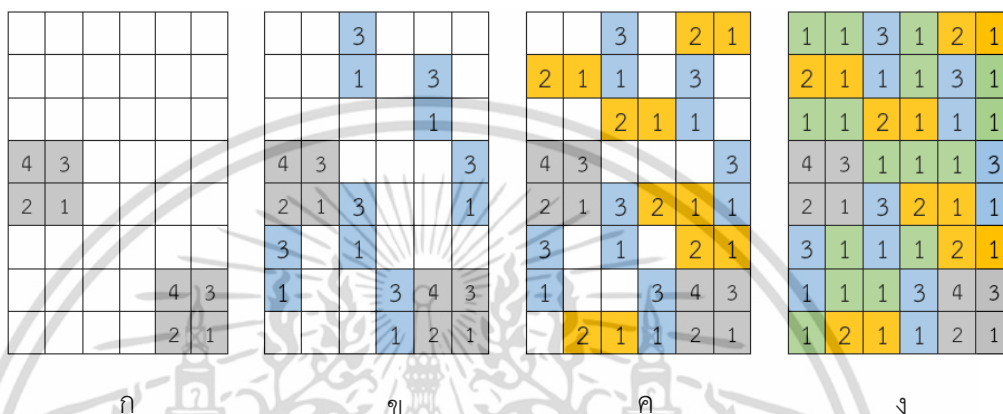
รูปที่ 2.2 วัสดุแกรนูล่า (granular) ที่ถูกนำมาสร้างเป็นแผ่นแพลลิตเตอร์ไตร์ฟ



รูปที่ 2.3 เกรนขนาดต่างๆ ที่ปรากฏบนวัสดุแกรนูล่า (Granular)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีในการสร้างแบบจำลองของแผ่นแพลิตเตอร์ไตร์ฟที่มีโครงสร้างแบบกรนูล่าตามงานวิจัย [20] ในขั้นตอนแรก จะทำการกำหนดตำแหน่งต่างๆ ของเกรนแต่ละขนาด ด้วยตัวเลข 1 ถึง 4 ดังแสดงในรูปที่ 2.3 โดยการกำหนดในลักษณะเช่นนี้ จะเป็นประโยชน์สำหรับนำไปวิเคราะห์ในกระบวนการเขียน ยกตัวอย่างเช่น หากทำการบันทึกข้อมูลบิต 0 หรือ 1 ลงไปยังเกรนขนาดใดๆก็ตาม สภาพความเป็นแม่เหล็กของเกรนนั้น จะมีทิศทางขึ้นอยู่กับตำแหน่งเลข 1 เสมอ สำหรับตัวอย่างในการจัดวางเกรนแต่ละขนาดลงไปยังแผ่นแพลิตเตอร์ไตร์ฟนั้น จะถูกแสดงในรูปที่ 2.4



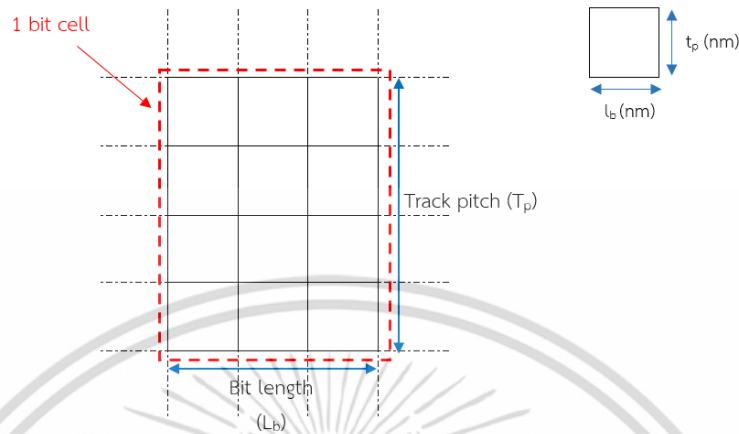
รูปที่ 2.4 ตัวอย่างการวางเกรนขนาด 1x1(ก), 1x2(ข), 2x1(ค), และ 2x2(ง) ลงบนแผ่นมีเดีย

รูปที่ 2.4 แสดงขั้นตอนในการจัดวางเกรนขนาดต่างๆ ลงบนแผ่นแพลิตเตอร์ไตร์ฟจำนวน 48 พิกเซลล์ (กำหนดให้ 1 พิกเซลล์มีขนาดเท่ากับเกรนขนาด 1x1) ลำดับแรก จะทำการจัดวางเกรนขนาดใหญ่ที่สุดหรือขนาด 2x2 ลงไปก่อน เป็นที่ทราบกันดีแล้วว่า เกรนขนาด 2x2 จะมีโอกาสในการเกิดขึ้นบนแผ่นแพลิตเตอร์ไตร์ฟเท่ากับ 1/6 ฉะนั้น เราสามารถที่จะวางเกรนขนาด 2x2 ลงไปได้เพียง 2 เกรน (8 พิกเซลล์จาก 48 พิกเซลล์) หลังจากนั้นจะทำการจัดวางเกรนขนาด 1x2, 2x1 และ 2x2 ลงไปที่แผ่นแพลิตเตอร์ไตร์ฟจำนวน 6, 6 และ 16 เกรน ตามลำดับ โดยการจัดวางเกรนขนาดต่างๆ ที่กล่าวมาข้างต้นนั้นจะเป็นการจัดวางแบบสุ่ม (Random) ทั้งสิ้น

2.2.2 กระบวนการเขียน

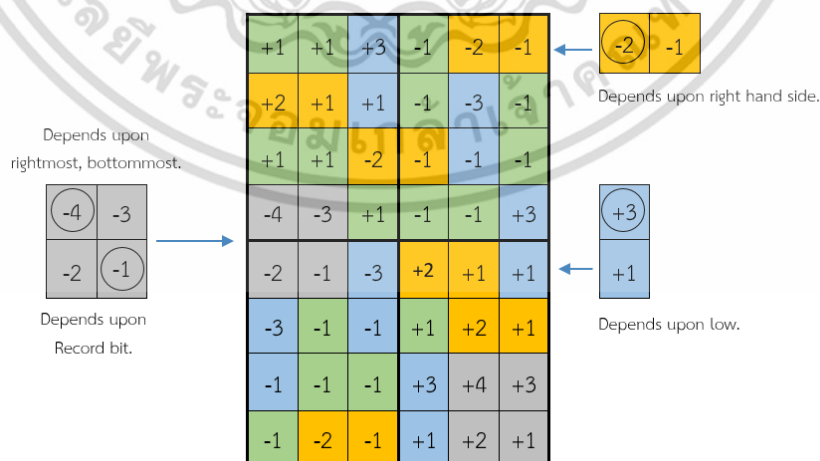
เมื่อเกรนแต่ละขนาดถูกจัดวางลงไปยังแผ่นแพลิตเตอร์ไตร์ฟแล้วนั้น หัวเขียนจะทำการป้อนสภาพแม่เหล็กที่มีทิศทางสัมพันธ์กับบิตข้อมูลไบนารี ลงไปยังแต่ละเกรนบนแผ่นแพลิตเตอร์ไตร์ฟ สำหรับกระบวนการเขียนข้อมูลนั้น บิตข้อมูลจะถูกแปลงให้อยู่ในรูปกระแสไฟฟ้าสี่เหลี่ยม (Rectangular current waveform) ที่เรียกว่ากระแสเขียน จากนั้นกระแสเขียนจะถูกป้อนไปยังขดลวดเหนี่ยวนำทำให้เกิดการเหนี่ยวนำเกิดเป็นสนามเขียนแม่เหล็ก (Magnetic write field) บริเวณช่องว่างระหว่าง

แผ่นแพลิตเตอร์ไคร์ฟกับหัวเขียน โดยทั่วไปนั้น สนามแม่เหล็กเขียนต้องมีความเข้มมากกว่าสภาพลบ
 ล้างแม่เหล็กของแต่ละเกรน เพื่อที่จะสามารถเปลี่ยนสภาพแม่เหล็กของแต่ละเกรนให้มีทิศทางเป็นไป
 ตามบิตข้อมูลไบนารี ที่ต้องการจะบันทึกได้



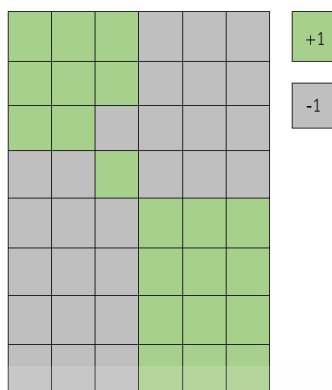
รูปที่ 2.5 พิกเซลขนาด 4x3 ที่ถูกใช้สำหรับบันทึกข้อมูล 1 บิต

สำหรับกระบวนการเขียนนั้น วิทยานิพนธ์เล่มนี้จะอ้างอิงจากงานวิจัย [22] ที่ใช้พิกเซลขนาด
 4x3 ในการบันทึกข้อมูล 1 บิต ดังแสดงในรูปที่ 2.5 โดยที่ความยาวของบิตจะถูกเรียกว่า ความยาว
 ตามแทร็ค (L_b) และความกว้างบิตจะถูกเรียกว่า ความยาวตั้งฉากแทร็ค (T_p) ตามลำดับ เพราะฉะนั้น
 เมื่อเราต้องการบันทึกข้อมูล [1 0 ; 0 1] ลงไปยังแผ่นแพลิตเตอร์ไคร์ฟขนาด 48 พิกเซล ที่ถูก
 ออกแบบในรูปที่ 2.4 ทิศทางของสภาพแม่เหล็กของแต่ละเกรนจะแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 การบันทึกบิตข้อมูล [1 0 ; 0 1] ลงไปยังแผ่นแพลิตเตอร์ไคร์ฟขนาด 48 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 สภาพแม่เหล็กสำหรับบันทึกข้อมูล [1 0 ; 0 1]

รูปที่ 2.6 แสดงถึงการบันทึกบิตข้อมูล [1 0; 0 1] ลงไปยังแผ่นแผ่นฟลิตเตอร์ไดรฟ์ขนาด 48 พิกเซลล์ ที่ถูกออกแบบในรูปที่ 2.4 (การใช้พิกเซลจำนวนเท่ากับ 48 จะทำให้สามารถเก็บบันทึกข้อมูลได้ขนาดเท่ากับ 4 บิต [22]) จะพบว่าเมื่อเราใช้สนามเขียนแม่เหล็กทำการเปลี่ยนแปลงสภาพแม่เหล็กของแต่ละเกรน ให้มีทิศทางที่เป็นไปตามบิตข้อมูลไบนารีที่ต้องการบันทึก (โดยจะกำหนดให้บิต 1 มีสภาพแม่เหล็กทิศบวก และบิต 0 ให้มีสภาพแม่เหล็กทิศลบ) จะสังเกตว่าเกรนแต่ละขนาดจะมีทิศทางของแม่เหล็ก เป็นไปตามตำแหน่งเลข 1 เสมอ เพราะฉะนั้น ทิศทางของสภาพแม่เหล็กสำหรับการบันทึกข้อมูล [1 0; 0 1] ลงบนแผ่นฟลิตเตอร์ไดรฟ์ขนาด 48 พิกเซลล์ จะแสดงได้ดังรูปที่ 2.7 โดยที่สีเขียวแทนการบันทึกบิต 1 โดยจะมีทิศทางแม่เหล็กเป็นบวก และสีเทาแทนการบันทึกบิต 0 โดยจะมีทิศทางแม่เหล็กเป็นลบ ตามลำดับ

2.2.3 กระบวนการอ่าน

ในระหว่างการอ่านข้อมูลจากแผ่นฟลิตเตอร์ไดรฟ์ หัวอ่านจะทำการตรวจจับการเปลี่ยนแปลงของฟลักซ์แม่เหล็กของแต่ละเกรนในแนวตามแตรีก ๓ ตำแหน่งที่มีสภาพความเป็นแม่เหล็กเท่านั้น ซึ่งจะทำให้เกิดสัญญาณพัลส์แรงดันไฟฟ้าเหนี่ยวนำในขดลวด ตามกฎของฟาราเดย์ (Faraday's law) สำหรับบริเวณที่มีการเปลี่ยนสถานะเอกเทศ (Isolated transition) หัวอ่านจะให้สัญญาณพัลส์แรงดันไฟฟ้า หรือที่เรียกว่า สัญญาณพัลส์เปลี่ยนสถานะ (Transition pulse) โดยที่สัญญาณอ่านกลับ (Read back) ที่อ่านได้ คือการรวมกันของสัญญาณพัลส์เปลี่ยนสถานะ ที่มีความสอดคล้องกับข้อมูลไบนารีที่บันทึก อย่างไรก็ตาม กระบวนการในการอ่านกลับ เป็นกระบวนการที่นำดึงดูดสำหรับงานวิจัยใหม่ๆ ในปัจจุบัน ที่ต้องหาวิธีในการอ่านให้มีประสิทธิภาพสูงสุด เพื่อที่จะทำการกักเก็บข้อมูลที่ถูกบันทึกออกมาให้ปราศจากความผิดพลาด หรือมีความผิดพลาดเกิดขึ้นน้อยกว่ามาตรฐานที่กำหนดไว้ ทั้งนี้สำหรับช่องสัญญาณการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR) สัญญาณอ่านกลับจะถูก

อ่านออกมาในรูปแบบสองมิติ กล่าวคือ สัญญาณอ่านกลับที่ถูกอ่านออกมานั้น จะได้รับผลกระทบจากเอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อคุณได้เห็นเอกสารฉบับนี้แล้ว กรุณาอย่าเผยแพร่เอกสารนี้ไปยังผู้อื่นโดยไม่ได้รับอนุญาต การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาต ถือว่าผิดกฎหมาย และจะมีความผิดตามกฎหมายที่เกี่ยวข้อง

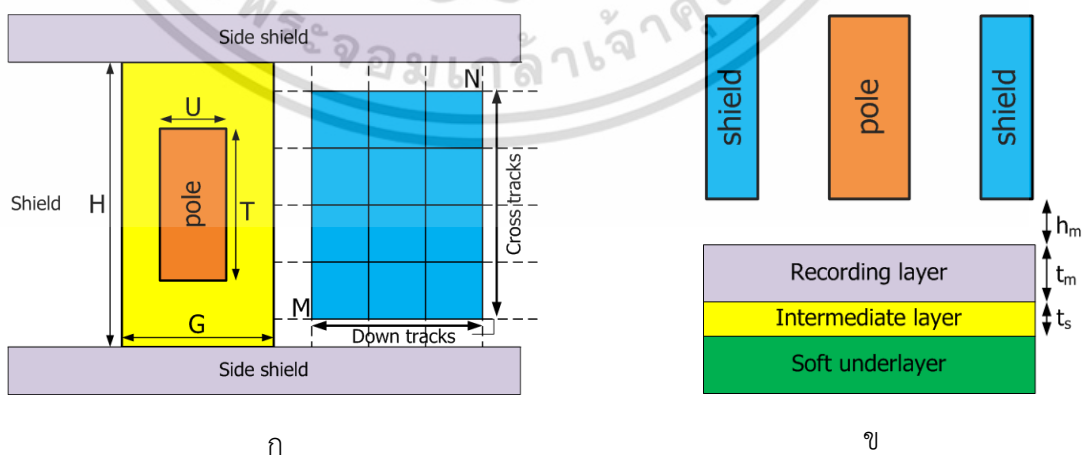
การแทรกสอดของสภาพแม่เหล็กจากบิตข้างเคียง (ISI) และจากแทร็กข้างเคียง (ITI) อันเนื่องมาจากข้อจำกัดของขนาดหัวอ่านที่มีขนาดใหญ่กว่าบิตข้อมูล และความไม่แน่นอนของสภาพแม่เหล็ก ไม่นานมานี้ ในงานวิจัย [23] นักวิจัยชื่อ M. Yamashita ได้ทำการศึกษาถึงผลกระทบที่เกิดจากโครงสร้างของหัวอ่าน และแผ่นบันทึก ที่ส่งผลต่อสัญญาณอ่านกลับที่ถูกอ่านออกมา ซึ่งสามารถเขียนเป็นสมการของความไวในการอ่าน (Sensitivity function) ดังสมการที่ 2.1

$$\psi(x, y) = \alpha_0 \{ \tanh(\alpha_1 x + \alpha_2) - \tanh(\alpha_1 x - \alpha_2) \} \times \dots \{ \tanh(\alpha_3 x + \alpha_4) - \tanh(\alpha_3 x - \alpha_4) \} \tag{2.1}$$

พิกัด (x, y) แทนตำแหน่งพิกัดใดๆ ของการอ่านข้อมูล 1 บิต (หน่วยนาโนเมตร) ซึ่งหาก (x, y) มีค่าเท่ากับ $(0, 0)$ หมายถึง ตำแหน่งศูนย์กลางของบิตเซลล์ที่กำลังอ่าน ส่วนพารามิเตอร์ α_k โดย $k \in \{1, 2, 3, 4\}$ เป็นผลกระทบที่เกิดจากโครงสร้างของหัวอ่านและแผ่นบันทึก ซึ่งสามารถคำนวณได้จากสมการที่ 2.2

$$\alpha_k = |w_{k0}H + w_{k1}G + w_{k2}T + w_{k3}U + w_{k4}h_m + w_{k5}t_m + w_{k6}t_s + w_{k7}| \tag{2.2}$$

โดย $H, G, T, U, h_m, t_m, t_s$ เป็นค่าพารามิเตอร์ที่บอกถึงโครงสร้างของหัวอ่าน และแผ่นบันทึก ในส่วนของ $w_{ki}, i \in \{1, 2, \dots, 7\}$ คือ ค่าที่ถูกนำมาใช้ในการถ่วงน้ำหนักพารามิเตอร์แต่ละตัวข้างต้น ซึ่งหาได้จากการทดลองในห้องปฏิบัติการ [24]



รูปที่ 2.8 โครงสร้างของหัวอ่าน (ก) และแผ่นบันทึก หรือแผ่นแพลิตเตอร์ไดรฟ์ (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.8 ค่าพารามิเตอร์แต่ละตัว แสดงได้ดังนี้

H คือ ความกว้างของ side shields gap [nm]

G คือ ความกว้างของ side gap [nm]

T คือ ความยาวของ pole ในแนวตั้งฉากกับแตรีก [nm]

U คือ ความยาวของ pole ในแนวตามแตรีก [nm]

h_m คือ ความหนาของชั้น magnetic spacing [nm]

t_m คือ ความหนาของชั้น recording layer [nm]

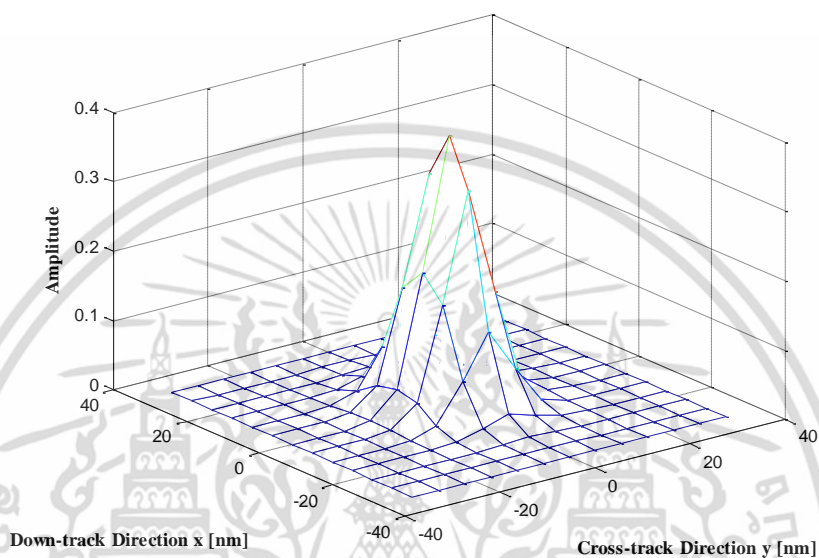
t_s คือ ความหนาของชั้น intermediate layer [nm]

ในสมัยก่อนนั้น M. Yamashita ทราบเพียงแค่ว่า พารามิเตอร์จากโครงสร้างของหัวอ่านและแผ่นบันทึกในรูปที่ 2.8 ส่งผลกระทบต่อค่าความไวในการอ่าน แต่ไม่ทราบว่าค่าพารามิเตอร์แต่ละตัวส่งผลต่อความไวในการอ่านเป็นสัดส่วนมากน้อยเพียงใด ต่อการอ่านข้อมูล 1 บิต อย่างไรก็ตาม ไม่นานมานี้ งานวิจัยของ S. Kobayashi ได้ทำการนำตัวแปรบางอย่าง มาถ่วงน้ำหนักของค่าพารามิเตอร์ข้างต้น นั่นคือเมทริกซ์ถ่วงน้ำหนัก $\mathbf{W} = [\mathbf{w}_0 \ \mathbf{w}_1 \ \mathbf{w}_2 \ \mathbf{w}_3 \ \mathbf{w}_4]$ ขนาด 8×5 โดยค่าใน \mathbf{W} เป็นค่าที่ได้จากการทดลอง และศึกษาในห้องปฏิบัติการเฉพาะทาง โดยในวิทยานิพนธ์เล่มนี้ จะนำค่าที่ได้จากงานวิจัยของ S. Kobayashi [24] มาใช้ โดยที่ค่าของเมทริกซ์ถ่วงน้ำหนัก \mathbf{W} ที่อ้างอิงตาม [24] นั้น แสดงได้ดังสมการด้านล่าง

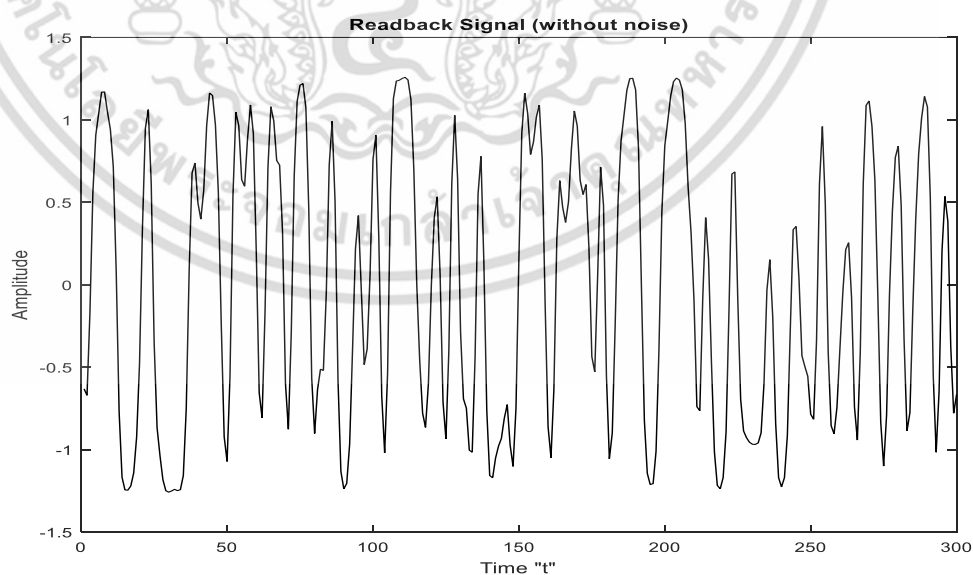
$$\mathbf{W} = \begin{bmatrix} -1.822E-3 & -6.966E-4 & 5.261E-3 & 2.964E-3 & -2.354E-3 \\ 6.417E-3 & -3.361E-3 & -3.753E-2 & 5.768E-4 & -1.458E-3 \\ 1.770E-3 & 1.569E-4 & -3.555E-3 & -3.255E-3 & -9.442E-2 \\ -4.107E-3 & -6.815E-5 & -2.620E-2 & -1.754E-3 & -3.503E-2 \\ 2.761E-2 & -2.060E-2 & 4.717E-2 & 1.634E-2 & 1.1224E-1 \\ -3.064E-3 & -1.626E-3 & -8.730E-4 & 1.420E-3 & 8.322E-3 \\ 6.046E-3 & 1.522E-3 & 9.6652E-3 & -4.806E-3 & -3.61E-2 \\ -3.570E-1 & 3.398E-1 & -2.267E-1 & -2.956E-1 & -7.039E-1 \end{bmatrix} \quad (2.3)$$

ภายหลังจากที่ S. Kobayashi ได้ทำการทดลองในห้องปฏิบัติการ จนกระทั่งสามารถหาค่าในเมทริกซ์ถ่วงน้ำหนัก \mathbf{W} เป็นที่เรียบร้อยแล้ว ต่อมา M. Yamashita ได้นำค่าเหล่านั้น มาพล็อตกราฟผลตอบแทนในรูปแบบพัลส์สามมิติ โดยใช้สมการที่ 2.1 เพื่อดูความสัมพันธ์ระหว่างค่าพารามิเตอร์ของหัวอ่าน และแผ่นบันทึก (ภายหลังจากการใช้ \mathbf{W} มาถ่วงน้ำหนัก) โดยที่กำหนด H, G, T, U , เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h_m , t_m และ t_s มีค่าเท่ากับ 26, 15, 13, 2, 4, 10 และ 1 นาโนเมตร ตามลำดับ ทั้งนี้เขาได้ทำการกำหนดให้การบันทึกข้อมูล 1 บิต ใช้พิกเซลจำนวนเท่ากับ 4×3 พิกเซลล์ ตามงานวิจัย [22] โดยมีค่า L_b และ T_p เท่ากับ 16.5 และ 22 นาโนเมตร ตามลำดับ ซึ่งจะส่งผลให้พื้นที่ของ 1 บิตเซลล์ มีขนาดเท่ากับ 3.63 ไมโครเมตร โดยจะมีค่าความหนาแน่นเชิงพื้นที่เท่ากับ 5.62 เทระบิตต่อตารางนิ้ว กราฟแสดงผลตอบสนองในรูปแบบสัญญาณพัลส์สามมิติของค่าความไวในการอ่าน แสดงได้ดังรูปที่ 2.10



รูปที่ 2.9 ผลตอบสนองในรูปแบบสัญญาณพัลส์สามมิติของค่าความไวในการอ่าน



รูปที่ 2.10 สัญญาณอ่านกลับ จากช่องสัญญาณ TDMR โดยใช้พารามิเตอร์ของ S. Kobayashi [24]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.9 แสดงผลตอบสนองในรูปแบบสัญญาณพัลส์สามมิติของค่าความไวในการอ่านข้อมูล 1 บิต โดยใช้ค่าพารามิเตอร์ของโครงสร้างหัวอ่านและเขียน รวมทั้งเมทริกซ์ถ่วงน้ำหนัก \mathbf{W} จากงานวิจัย [24] ทั้งนี้ จะสังเกตว่าที่จุดศูนย์กลางของบิตเซลล์ $(x, y) = (0, 0)$ จะให้ค่าความไวในการอ่านสูงที่สุด กล่าวคือ เมื่อหัวอ่านทำการอ่านพลักซ์แม่เหล็กจากจุดกึ่งกลางออกมา จะให้ค่าความเข้มสนามแม่เหล็กที่มีค่ามากที่สุด อย่างไรก็ตาม ค่าความเข้มสนามแม่เหล็กที่อ่านได้จะลดลงเรื่อยๆ เมื่อพิกัด (x, y) ใดๆ ห่างจากจุดศูนย์กลางมากขึ้น สังเกตว่า ปัญหาการแทรกสอดระหว่างบิตข้างเคียง (ISI) และแทรกข้างเคียง (ITI) จะเกิดขึ้น เนื่องจากสัญญาณพัลส์ของค่าความไวในการอ่านข้อมูลครอบคลุมพื้นที่เกิน 3.63 ไมโครเมตร (1 บิตเซลล์) โดยในรูปที่ 2.10 แสดงสัญญาณอ่านกลับของข้อมูล 100 บิต จากช่องสัญญาณ TDMR ที่มีผลตอบสนองในรูปแบบสัญญาณพัลส์สามมิติของค่าความไวในการอ่านในการอ่าน ตามรูปที่ 2.9

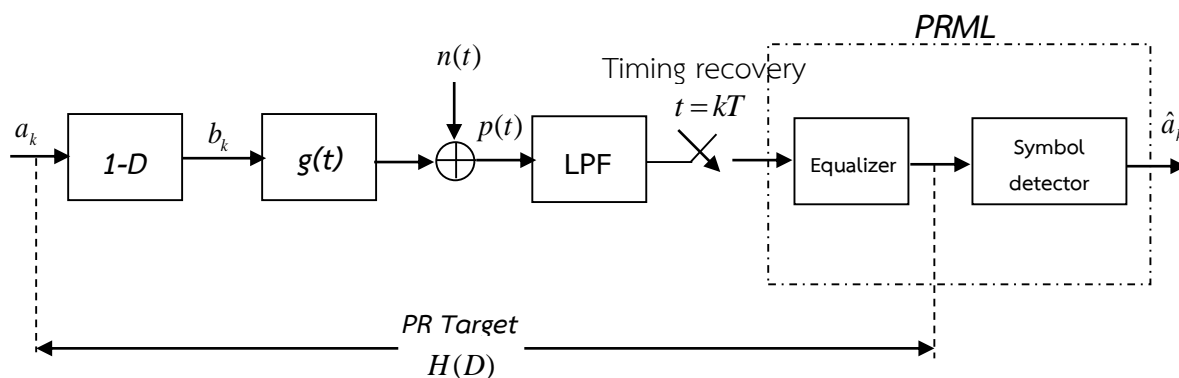
2.3 แบบจำลองช่องสัญญาณในการเขียนและอ่านข้อมูลบนฮาร์ดดิสก์ไดรฟ์

เนื้อหาในส่วนนี้จะดัดแปลงและเรียบเรียงจากหนังสือ [35] ซึ่งผู้เขียนได้กล่าวไว้ว่า แบบจำลองหรือบล็อกไดอะแกรมของกระบวนการประมวลผลสัญญาณ สำหรับระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ไดรฟ์นั้น สามารถแบ่งออกเป็นสองประเภทหลักๆ คือ อย่างแรก แบบจำลองช่องสัญญาณเสมือนจริง (Realistic channel model) ที่มีลักษณะการทำงาน และวงจรรภายในคล้ายกันกับระบบจริง ที่มีการใช้จริงในอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ในปัจจุบัน อย่างที่สองคือแบบจำลองช่องสัญญาณอุดมคติ (Ideal channel model) เป็นแบบจำลองหรือบล็อกไดอะแกรมอย่างง่าย ที่มีักจะถูกนำมาใช้ในการวิเคราะห์สมรรถนะ และประสิทธิภาพของระบบประมวลผลสัญญาณเบื้องต้น สำหรับระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ไดรฟ์ เนื่องจากจะมีการสมมติเกณฑ์บางอย่าง เพื่อที่จะสามารถตัดวงจรในส่วนที่มีความซับซ้อนสูงทิ้งไป เพื่อให้ง่ายแก่การพิจารณาและเข้าใจหลักการการทำงานเบื้องต้น ในงานวิจัยที่เกี่ยวข้องกับกระบวนการประมวลผลสัญญาณฮาร์ดดิสก์ไดรฟ์ในปัจจุบันนั้น จะมีการใช้ทั้งสองแบบจำลอง ซึ่งขึ้นอยู่กับขอบเขตของแต่ละงานวิจัย

2.3.1 แบบจำลองช่องสัญญาณเสมือนจริง

ในรูปที่ 2.11 แสดงแบบจำลองของช่องสัญญาณเสมือนจริง สำหรับระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ไดรฟ์ โดยที่ $a_k \in \{0, 1\}$ แสดงเป็นลำดับข้อมูลอินพุตไบนารีที่มีคาบเวลา T ซึ่งจะถูกส่งผ่านวงจรตรวจหาอนุพันธ์อุดมคติ (Ideal differentiator) $1-D$ โดยที่ D คือตัวดำเนินการหน่วงเวลาไป T หน่วย ซึ่งเมื่อข้อมูลอินพุตไบนารี a_k ถูกส่งผ่าน จะทำให้เกิดการเปลี่ยนแปลงของแอมพลิจูดเป็น 3 ระดับ โดยแทนเป็นลำดับข้อมูล $b_k \in \{-1, 0, 1\}$ เมื่อ $b_k = +1, -1$ หมายถึง การเปลี่ยนสถานะแบบบวก (Positive transition) และแบบลบ (Negative transition) ตามลำดับ อย่างไรก็ตาม $b_k = 0$ หมายถึง ไม่มีการเปลี่ยนสถานะเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



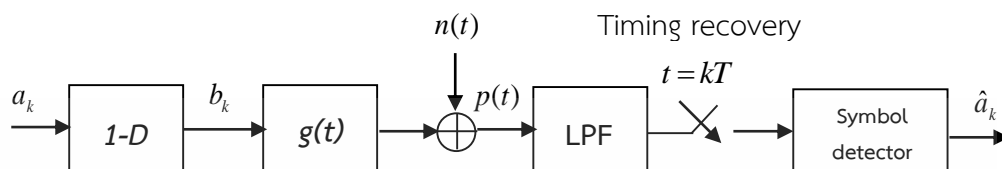
รูปที่ 2.11 แบบจำลองช่องสัญญาณเสมือนจริง

ภายหลังจากนั้น ลำดับข้อมูลที่มีการเปลี่ยนแปลงสถานะ b_k จะถูกส่งผ่านช่องสัญญาณที่มีผลตอบสนองอิมพัลส์เท่ากับผลตอบสนองการเปลี่ยนแปลงสถานะ $g(t)$ โดยที่เอาท์พุทที่ได้จะถูกรบกวนด้วยสัญญาณรบกวนภายนอก $n(t)$ เช่น สัญญาณรบกวนขาวแบบบวก (Additive white Gaussian noise: AWGN) เป็นต้น โดยกำหนดให้สัญญาณอ่านกลับ $p(t)$ ได้รับมาจาก $p(t) = b(t) + n(t)$ ตามลำดับ วงจรกรองต่ำ (Low pass filter : LPF) มีหน้าที่ในการกรององค์ประกอบความถี่ที่ไม่ต้องการ (ความถี่สูง) ที่หายไป หลังจากนั้น สัญญาณอ่านกลับจะถูกนำมาซีกตัวอย่าง ที่ซึ่งถูกควบคุมด้วยระบบไทมมิงรีคัฟเวอรี (Timing recovery) เพื่อทำการเปลี่ยนสัญญาณที่มีความต่อเนื่องทางเวลา (Continuous signal) เป็นสัญญาณที่ไม่ต่อเนื่องทางเวลาหรือสัญญาณดิจิทัล (Discrete/Digital signal) ในส่วนวงจรอีควอลไลเซอร์ (Equalizer) และวงจรตรวจหาสัญลักษณ์ (Symbol detector) จะมีการทำงานร่วมกัน ในการหาลำดับข้อมูลอินพุทที่เป็นไปได้มากที่สุด นั่นคือทำการประมาณค่า a_k ที่ควรจะเป็นมากที่สุดนั่นเอง

ในปัจจุบันนี้ วงจรตรวจหาสัญลักษณ์ที่นิยมนำมาใช้กับกระบวนการประมวลผลสัญญาณฮาร์ดดิสก์ไดรฟ์คือ วงจรตรวจหาวิเทอร์บี (Viterbi detector) [32] ซึ่งเป็นวงจรตรวจหาที่สามารถประมาณค่า a_k ได้ใกล้เคียงที่สุด ซึ่งแสดงให้เห็นว่าวงจรตรวจหาวิเทอร์บีเป็นวงจรตรวจหาลำดับที่ควรจะเป็นมากที่สุด (Maximum-likelihood sequence detector : MLSD) อย่างไรก็ตาม ความซับซ้อนของวงจรตรวจหาวิเทอร์บี จะมีค่าที่เพิ่มมากขึ้นแบบชี้กำลัง ตามจำนวนหน่วยความจำของช่องสัญญาณ ดังนั้น จึงต้องมีการนำมาใช้คู่กับวงจรอีควอลไลเซอร์ ที่มีหน้าที่ในการปรับผลตอบสนองรวมของระบบ ให้เป็นผลตอบสนองที่ต้องการ ที่เรียกว่า ผลตอบสนองทาร์เก็ต (Partial response target : PR) $H(D)$ ซึ่งจะทำให้ความซับซ้อนของวงจรตรวจหาวิเทอร์บีมีค่าลดลง เทคนิคในการนำวงจรอีควอลไลเซอร์มาใช้ร่วมกับวงจรตรวจหาวิเทอร์บีจะถูกเรียกว่า เทคนิคผลตอบสนองบางส่วนความจะเป็นสูงสุด (Partial-response maximum-likelihood : PRML) ซึ่งเป็นเทคนิควิธีการที่ใช้กันอย่างแพร่หลาย ทั้งในสถานการณ์วิจัย และอุตสาหกรรมการผลิต ของกระบวนการประมวลผลสัญญาณฮาร์ดดิสก์ไดรฟ์ในปัจจุบัน

2.3.2 แบบจำลองช่องสัญญาณอุดมคติ

อย่างไรก็ตาม หากทำการสมมติให้ กระบวนการทำงานของวงจรรีควอลไอเซอร์เป็นแบบสมบูรณ์ (Perfect equalization) ซึ่งจะสามารถลดรูปแบบจำลองช่องสัญญาณเสมือนจริง ให้อยู่ในรูปแบบอย่างง่าย หรือที่เรียกว่าแบบจำลองช่องสัญญาณอุดมคติ ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 แบบจำลองช่องสัญญาณอุดมคติ

กำหนดให้ลำดับข้อมูลอินพุตไบนารี $a_k \in \{0,1\}$ มีคาบเวลาเท่ากับ T ซึ่งมันจะถูกกล้ำสัญญาณ (Modulate) ด้วยสัญญาณพัลส์ไนควิสต์อุดมคติ (Ideal Nyquist pulse) ที่มีฟังก์ชันแสดงเป็น $q(t) = \sin(\pi t / T) / (\pi t / T)$ หลังจากนั้นเอาท์พุทที่ได้จะถูกกรบกวนจากสัญญาณรบกวน $n(t)$ เช่นเดียวกับกับระบบช่องสัญญาณเสมือนจริง ที่ภาครับนั้นองค์ประกอบนอกความถี่ของสัญญาณอ่านกลับ $p(t)$ จะถูกกรองออกด้วยวงจรรองต่ำ (LPF) และจะถูกทำการนำไปซีกตัวอย่าง เพื่อทำการเปลี่ยนสัญญาณที่มีความต่อเนื่องทางเวลาเป็นสัญญาณที่ไม่ต่อเนื่องทางเวลา หรือสัญญาณดิจิทัล ณ เวลาที่ถูกควบคุมด้วยระบบไทม์มิงรีคฟเวอรี่ต่อไป จากนั้นลำดับข้อมูลดิจิทัลที่ได้ จะถูกส่งไปยังวงจรตรวจหาสัญลักษณ์ เพื่อทำการตรวจหาลำดับข้อมูลอินพุต a_k ที่เป็นไปได้มากที่สุด

ผลตอบสนองทาร์เก็ตบางส่วนหรือที่เรียกว่า ทาร์เก็ตแบบพ็ออาร์ (Partial response: PR) ที่เป็นที่ยอมรับสำหรับระบบการบันทึกแนวตั้งจะอยู่ในรูป

$$H(D) = (1 + D)^n \quad (2.4)$$

โดยที่ n คือเลขจำนวนเต็มบวก และ D คือ ตัวดำเนินการหน่วงเวลาตามลำดับ สำหรับตารางที่ 2.1 จะแสดงทาร์เก็ตพ็ออาร์ (ค่าใน H เป็นจำนวนเต็ม) ที่เป็นที่ยอมรับกันโดยทั่วไป ในระบบบันทึกข้อมูลเชิงแม่เหล็กแนวตั้งและแนวนอน อย่างไรก็ตาม ในวิทยานิพนธ์เล่มนี้เน้นไปที่ระบบบันทึกข้อมูลเชิงแม่เหล็ก 2 มิติ (TDMR) ซึ่งเป็นระบบบันทึกข้อมูลสมัยใหม่ ที่ต้องการกระบวนการประมวลผลสัญญาณที่มีประสิทธิภาพสูง ดังนั้น ทาร์เก็ตแบบจีพ็ออาร์ GPR (ค่าใน H ตัวแรกเป็น 1 ที่เหลือเป็นจำนวนจริง) จะถูกนำมาใช้แทนทาร์เก็ตแบบพ็ออาร์ ส่วนวิธีการออกแบบทาร์เก็ตแบบจีพ็ออาร์ สามารถ

ดูได้จากงานวิจัย [31]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตัวอย่างทาร์เก็ตแบบพ็อดาร์ที่ใช้ในช่องสัญญาณบันทึกข้อมูลเชิงแม่เหล็กแนวตั้ง และแนวนอน

ระบบบันทึกข้อมูลเชิงแม่เหล็ก	ทาร์เก็ตแบบพ็อดาร์ (PR)		
	$n=1$	$n=2$	$n=3$
แนวตั้ง (ปัจจุบัน)	PR1 [1 1] $1+D$	PR2 [1 2 1] $1+2D+D^2$	EPR2 [1 3 3 1] $1+3D+3D^2+D^3$
แนวนอน (อดีต)	PR4 [1 0 -1] $1-D^2$	PR2 [1 1 -1 -1] $1+D-D^2-D^3$	EPR2 [1 2 0 -2 -1] $1+2D-D^3-D^4$

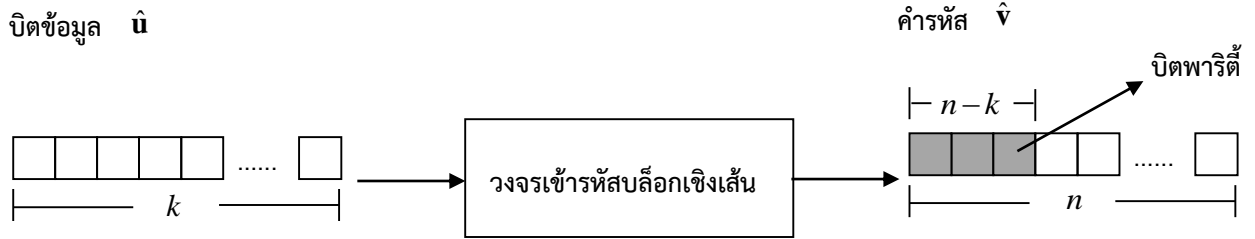
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสพาริตีเช็คความหนาแน่นต่ำ (LDPC)

ในระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ไดรฟ์นั้น นอกจากจะต้องการความจุในการบันทึกข้อมูลในปริมาณที่มากแล้ว ยังต้องการความสามารถในการประมวลผลสัญญาณที่มีประสิทธิภาพสูง เพื่อให้ข้อมูลที่ภาครับปราศจากความผิดพลาด หรือมีความผิดพลาดน้อยที่สุดเท่าที่จะเป็นไปได้ เป็นเหตุให้ในปี 1964 รหัสพาริตีเช็คความหนาแน่นต่ำ (Low-density parity-check codes: LDPC) หรือรหัสแอลดีพีซีจึงถือกำเนิดขึ้น [3] เพื่อแก้ไขความผิดพลาดของข้อมูล จากความผิดพลาดของช่องสัญญาณสื่อสาร รหัสดังกล่าวเป็นรหัสบล็อกเชิงเส้น (Linear block code) ที่ถูกนำมาประยุกต์ใช้กันอย่างแพร่หลายในระบบการบันทึกข้อมูลเชิงแม่เหล็กในปัจจุบัน เนื่องจากเป็นรหัสที่สามารถแก้ไขความผิดพลาดล่วงหน้าได้ (Forward error correction: FEC) พร้อมทั้งยังสามารถแก้ไขความผิดพลาดจากสัญญาณรบกวนชนิดต่างๆ ได้ดี ดังนั้นในบทที่ 3 นี้ ในส่วนแรกจะกล่าวถึงทฤษฎีของรหัสบล็อกเชิงเส้น ซึ่งเป็นรหัสประเภทหนึ่งที่ถูกใช้สำหรับแก้ไขความผิดพลาดของข้อมูล จากนั้นจะกล่าวถึงความหมายของรหัสแอลดีพีซี ซึ่งเป็นรหัสบล็อกเชิงเส้นประเภทหนึ่ง ว่าเหตุใดจึงเป็นที่สนใจในหมู่นักวิจัย จนถูกนำมาประยุกต์ใช้กันอย่างแพร่หลายในปัจจุบัน ส่วนต่อมา จะกล่าวถึงโครงสร้างของรหัสแอลดีพีซี โดยผู้วิจัยจะกล่าวถึงเฉพาะโครงสร้างที่เป็นที่นิยมใช้กันในปัจจุบันเท่านั้น และในส่วนสุดท้ายจะกล่าวถึงการถอดรหัสแอลดีพีซี ด้วยอัลกอริทึมต่างๆ ตามลำดับ

3.1 รหัสบล็อกเชิงเส้น

โดยปกติแล้วรหัสที่สามารถแก้ไขความผิดพลาดของข้อมูล (Error correction codes) ที่เกิดจากความผิดพลาดของช่องสัญญาณ จะถูกเรียกว่ารหัสช่องสัญญาณ (Channel coding) ซึ่งรหัสช่องสัญญาณสามารถแบ่งออกเป็นสองประเภทหลักๆ คือ รหัสคอนโวลูชัน (Convolution codes) และรหัสบล็อกเชิงเส้น (block codes) โดยในวิทยานิพนธ์เล่มนี้ จะกล่าวถึงเฉพาะรหัสบล็อกเชิงเส้นไปนารีเท่านั้น โดยคุณสมบัติที่สำคัญของรหัสบล็อกเชิงเส้นคือ บิตข้อมูลที่ต้องการส่ง จะถูกแบ่งออกเป็นบล็อกย่อยๆ ที่มีความยาวเท่ากับ k บิต เท่าๆกัน ซึ่งบิตข้อมูล (Message) ในแต่ละบล็อกนั้น จะอยู่ในรูปเวกเตอร์ $\mathbf{u} = [u_0, u_1, u_2, \dots, u_{k-1}]$ และเมื่อนำเวกเตอร์บิตข้อมูลแต่ละบล็อกป้อนเข้าสู่วงจรเข้ารหัส จะเกิดเป็นบล็อกข้อมูลชุดใหม่ที่มีความยาวเพิ่มขึ้น และจะถูกเรียกว่า คำรหัส (Codeword) นิยามเป็นเวกเตอร์ $\mathbf{v} = [v_0, v_1, v_2, \dots, v_{n-1}]$ ซึ่งมีความยาวเท่ากับ n จำนวนบิตที่ถูกเพิ่มเข้ามาจะถูกเรียกเรียกว่า บิตพาริตี (Parity) ซึ่งมีความยาวเท่ากับ $n-k$ อัตรารหัสของคำรหัสจะนิยามเป็น $R = k/n$ ในระบบสื่อสารต่างๆไป R จะมีค่ามากกว่าหรือเท่ากับ 0.5



รูปที่ 3.1 การเข้ารหัสบล็อกเชิงเส้น

ในหัวข้อนี้ จะกล่าวถึงเฉพาะการเข้ารหัสบล็อกเชิงเส้นเท่านั้น เนื่องจากการเข้ารหัสบล็อกเชิงเส้น จะถูกนำไปประยุกต์ใช้กับการเข้ารหัสและถอดรหัสแอสซิงโครนัส ซึ่งจะกล่าวถึงในหัวข้อถัดไป พิจารณารูปที่ 3.1 สำหรับรหัสบล็อกเชิงเส้นหนึ่งๆ เราสามารถใช้สัญลักษณ์ $C(n, k)$ เป็นตัวแทนบล็อกของคำรหัสที่เกิดจากการเข้ารหัสบิตข้อมูลความยาว k ให้เกิดเป็นคำรหัสความยาว n ซึ่งภายในวงจรเข้ารหัสบล็อกเชิงเส้นนั้น จะประกอบไปด้วยเมทริกซ์กำเนิด \mathbf{G} ที่มีไว้สำหรับสร้างคำรหัส จากบิตข้อมูลที่ถูกป้อนเข้ามา โดยเมทริกซ์กำเนิดรูปแบบหนึ่งสามารถแสดงได้ดังนี้

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (3.1)$$

โดยที่ p_{ij} ที่ซึ่ง $i \in \{0, 1, 2, \dots, k\}$ และ $j \in \{0, 1, 2, \dots, n-k\}$ คือเมทริกซ์พาริตีย่อย ซึ่งสามารถเขียนให้อยู่ในรูปแบบอย่างย่อเป็น

$$\mathbf{G} = [\mathbf{P} \mathbf{I}_k] \quad (3.2)$$

เมทริกซ์ \mathbf{P} คือเมทริกซ์พาริตีย่อยที่มีขนาดเท่ากับ $k \times (n-k)$ และเมทริกซ์ \mathbf{I}_k คือเมทริกซ์อัตลักษณ์ที่มีขนาดเท่ากับ $k \times k$ โดยกระบวนการที่ทำให้เกิดคำรหัสเวกเตอร์ \mathbf{v} นั้น สามารถกระทำได้โดยการคูณโปรดัคต์ระหว่างเวกเตอร์ข้อมูล \mathbf{u} กับเมทริกซ์กำเนิด \mathbf{G} ดังสมการ

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G} \quad (3.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีเข้ารหัสบล็อกเชิงเส้นโดยการนำเวกเตอร์บิตข้อมูลมาคูณกับเมทริกซ์กำเนิด \mathbf{G} ที่มีโครงสร้างดังสมการที่ 3.1 จะทำให้ได้เวกเตอร์คำรหัสที่เป็นรหัสเชิงระบบ (Systematic format) กล่าวคือ บิตข้อมูล และบิตพาริตีจะถูกแยกออกจากกันอย่างชัดเจนดังแสดงในรูปที่ 3.2 ซึ่งจะมีความสะดวกในการแยกบิตข้อมูลออกจากคำรหัสในส่วนของภาคถอดรหัส

บิตพาริตี	บิตข้อมูล
$n - k$	k

รูปที่ 3.2 รูปแบบของคำรหัสเชิงระบบ

อย่างไรก็ตาม รหัสบล็อกเชิงเส้นจะประกอบไปด้วยเมทริกซ์อีกตัวหนึ่ง ที่ใช้อธิบายความสัมพันธ์ของแต่ละบิตภายในเวกเตอร์คำรหัส \mathbf{v} ซึ่งเมทริกซ์ดังกล่าว ถูกสร้างและดัดแปลงมาจากเมทริกซ์กำเนิด \mathbf{G} โดยจะเรียกเมทริกซ์ตัวนี้ว่า “เมทริกซ์พาริตีเช็ค \mathbf{H} ” โครงสร้างของเมทริกซ์พาริตีเช็คสามารถพิจารณาได้จากสมการที่ 3.4

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} = [\mathbf{I}_{n-k} \mathbf{P}^T] \quad (3.4)$$

เมทริกซ์ \mathbf{P}^T คือ ทรานสโพสของเมทริกซ์พาริตี้อยู่พิจารณาในสมการที่ 3.4 จะพบว่า เมื่อเราทำการคูณกันแบบโพรดักต์ระหว่างแถว \mathbf{g}_i ของเมทริกซ์กำเนิด และแถวที่ \mathbf{h}_i ของเมทริกซ์พาริตีเช็ค จะมีค่าเท่ากับ 0 เนื่องมาจากเวกเตอร์ \mathbf{g}_i และ \mathbf{h}_i ตั้งฉากซึ่งกันและกันดังนั้น

$$\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0} \quad (3.5)$$

และ

$$\mathbf{v} \cdot \mathbf{H}^T = \mathbf{u} \cdot \mathbf{G} \cdot \mathbf{H}^T = \mathbf{0} \quad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปแล้ว การออกแบบรหัสบล็อกเชิงเส้นหรือรหัสแอลดีพีซี จะเริ่มต้นด้วยการออกแบบและสร้างเมทริกซ์พาริตีเช็คด้วยวิธีการต่างๆ และจากนั้น จะทำการแปลงเมทริกซ์พาริตีเช็คให้อยู่ในรูปเมทริกซ์กำเนิด เพื่อนำมาเข้ารหัสด้วยสมการที่ 3.3 อย่างไรก็ตาม จะสังเกตว่าความสัมพันธ์ในสมการที่ 3.6 นั้นสามารถเขียนใหม่ได้เป็น

$$\mathbf{H} \cdot \mathbf{v}^T = [\mathbf{I}_{n-k} \ \mathbf{P}^T] \cdot \begin{bmatrix} \mathbf{b} \\ \mathbf{u} \end{bmatrix} = \mathbf{0} \quad (3.7)$$

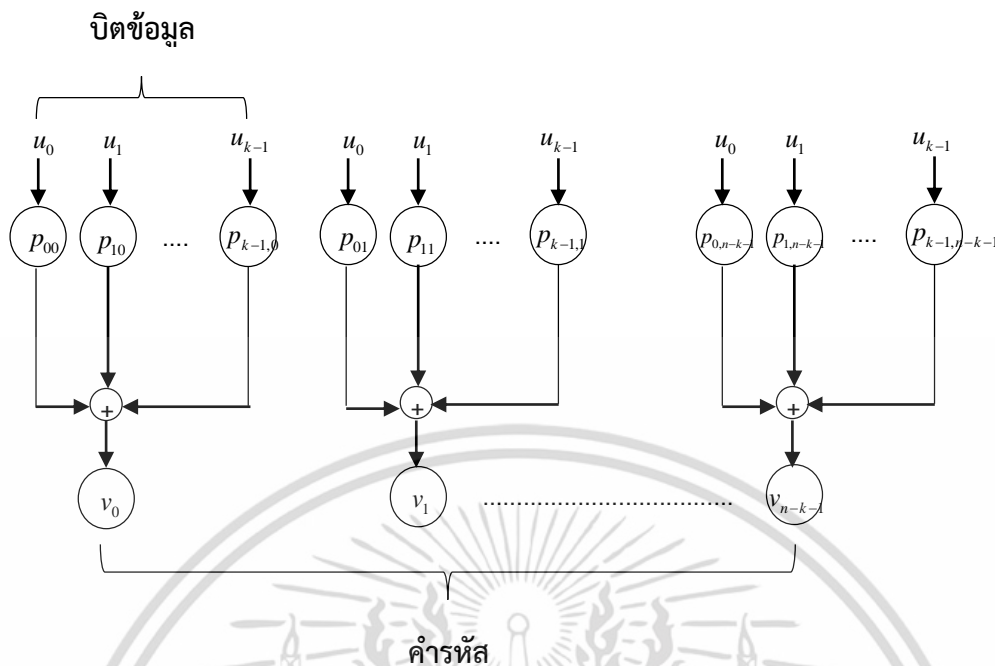
โดย \mathbf{b} คือเวกเตอร์พิตพาริตีเช็คที่จะทำการค้นหา ซึ่งมีความยาวเท่ากับ $n-k$ และ \mathbf{u} คือ เวกเตอร์ข้อมูล ฉะนั้น เมื่อทำการคูณกันระหว่างเมทริกซ์ทั้งสองตามสมการที่ 3.7 จะได้

$$\begin{bmatrix} b_0 & 0 & \cdots & 0 & u_0 p_{00} & u_1 p_{10} & \cdots & u_{k-1} p_{k-1,0} \\ 0 & b_1 & \cdots & 0 & u_0 p_{01} & u_1 p_{11} & \cdots & u_{k-1} p_{k-1,1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & b_{n-k-1} & u_0 p_{0,n-k-1} & u_1 p_{1,n-k-1} & \cdots & u_{k-1} p_{k-1,n-k-1} \end{bmatrix} = \mathbf{0} \quad (3.8)$$

จากสมการที่ 3.8 จะพบว่า การเข้ารหัสบล็อกเชิงเส้น สามารถกระทำผ่านเมทริกซ์พาริตีเช็ค \mathbf{H} ได้โดยตรง ไม่จำเป็นต้องแปลงเมทริกซ์พาริตีเช็คให้อยู่ในรูปเมทริกซ์กำเนิด \mathbf{G} ฉะนั้น จากสมการที่ 3.8 เราสามารถหาพิตพาริตีเช็คได้จากสมการ

$$b_i = u_0 p_{0i} + u_1 p_{1i} + \cdots + u_{k-1} p_{k-1,i} \quad (3.9)$$

โดย $i=0,1,\dots,n-k-1$ อย่างไรก็ตาม อัลกอริทึมที่ถูกกล่าวถึงทั้งหมดในวิทยานิพนธ์เล่มนี้ จะทำการเข้ารหัสแอลดีพีซีโดยใช้ความสัมพันธ์จากสมการที่ 3.7 – 3.9 ผ่านเมทริกซ์พาริตีเช็ค \mathbf{H} ทั้งสิ้น จากสมการที่ 3.9 พบว่าวงจรเข้ารหัสบล็อกเชิงเส้นสามารถแสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 วงจรเข้ารหัสบล็อกเชิงเส้น

3.2 รหัสพาริตีเช็คความหนาแน่นต่ำ (LDPC)

รหัสพาริตีเช็คความหนาแน่นต่ำ (low density parity-check codes: LDPC) หรือรหัสแอลดีพีซี [2] ถูกคิดค้นโดย R. Gallager ในปี 1964 รหัสแอลดีพีซีเป็นรหัสบล็อกเชิงเส้น ที่เกิดจากการเข้ารหัสบิตข้อมูลด้วยเมทริกซ์พาริตีเช็คความหนาแน่นต่ำ (หากเทียบกับขนาดของเมทริกซ์) เพื่อให้มีระยะห่างต่ำสุด d_{\min} ของรหัสมีค่าสูง ในอดีตนั้น รหัสแอลดีพีซีไม่ค่อยได้รับความนิยมเท่าที่ควร เนื่องจากเทคโนโลยีในสมัยนั้น (ฮาร์ดแวร์ และซอฟต์แวร์) ยังไม่เอื้อต่อการจำลอง และพิจารณาถึงสมรรถนะของมัน ประกอบกับในช่วงเวลาดังกล่าว มีการคิดค้นรหัสเทอร์โบ [1] ซึ่งเป็นรหัสคอนวูลูชันชนิดหนึ่ง ที่ถูกใช้งานกันอย่างแพร่หลาย เนื่องจากสมรรถนะของรหัสเทอร์โบเข้าใกล้ขีดจำกัดของแชนนอน อย่างไรก็ตาม จนกระทั่งในปี 1981 นักคณิตศาสตร์ชื่อ R. M. Tanner ได้นำเสนอวิธีการอธิบายความสัมพันธ์ของรหัสแอลดีพีซี ให้อยู่ในรูปแบบกราฟสองส่วน (Bipartite graph) โดยเรียกกราฟนี้ว่า “กราฟแทนเนอร์” [5] การคิดค้นกราฟแทนเนอร์นี้ ส่งผลให้เกิดอัลกอริทึมสำหรับถอดรหัสแอลดีพีซี เช่น อัลกอริทึมซัมโปรดักต์ (Sum-product) [4] หรืออัลกอริทึมแพร่กระจายค่าความเชื่อมั่น (Belief propagation) ซึ่งอัลกอริทึมนี้ อาศัยกราฟแทนเนอร์เป็นตัวอธิบายถึงวิธีการถอดรหัสแอลดีพีซีให้ง่ายขึ้น จากนั้น ในปี 1990 นักวิจัย D. Mackay ได้ทำการพิสูจน์ว่าการถอดรหัสแอลดีพีซีด้วยอัลกอริทึมแพร่กระจายค่าความเชื่อมั่น มีผลทำให้สมรรถนะของรหัสแอลดีพีซีเข้าใกล้ขีดจำกัดของแชนนอน [4] เช่นเดียวกับกับรหัสเทอร์โบ แต่มีความซับซ้อนในการเข้ารหัส และถอดรหัสที่ต่ำกว่ารหัสเทอร์โบมาก ต่อมาในงานวิจัยของ T. J. Richardson [25] ได้แสดงให้เห็นว่า รหัสแอลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

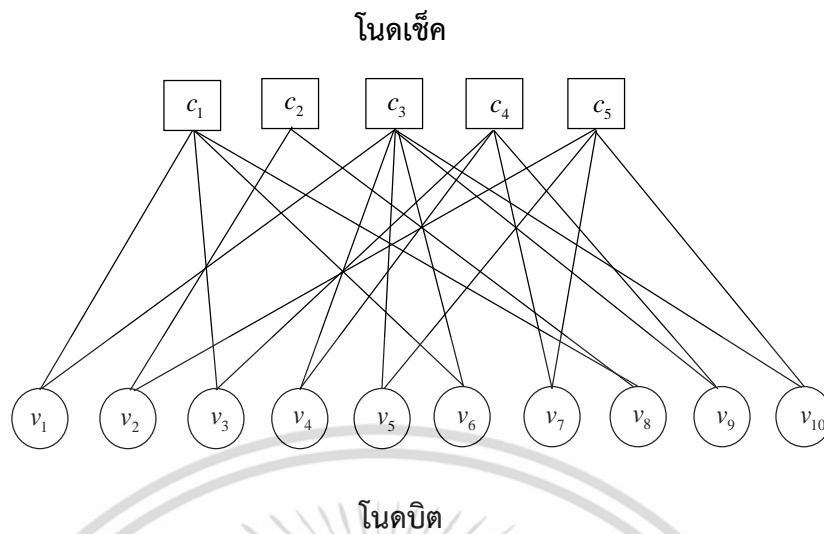
พีซีที่มีบล็อกรหัสข้อมูลขนาดใหญ่มีสมรรถนะในการแก้ไขความผิดพลาดที่ดีกว่ารหัสเทอร์โบ ดังนั้นสำหรับหลายๆ แอปพลิเคชันในปัจจุบัน ที่ต้องการส่งข้อมูลขนาดใหญ่ที่สุดเท่าที่จะเป็นไปได้ เพื่อให้ได้อัตราข้อมูล (Data rate) สูง อีกทั้งยังต้องการความซับซ้อนของวงจรฮาร์ดแวร์ต่ำ รหัสแอลดีพีซีจึงเป็นที่นิยมแก่การนำมาใช้งานมากกว่ารหัสเทอร์โบ รวมถึงในวิทยานิพนธ์เล่มนี้ รหัสแอลดีพีซีก็ถูกนำมาใช้สำหรับระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ไดรฟ์ด้วยเช่นกัน

3.2.1 กราฟแทนเนอร์ (Tanner graph)

ในปี 1981 นักคณิตศาสตร์ชื่อ R. M. Tanner ได้คิดค้นกราฟสองส่วน เพื่ออธิบายความสัมพันธ์ของแต่ละบิตในเวกเตอร์คำรหัส ที่เกิดจากการนำเวกเตอร์บิตข้อมูล u ไปคูณกับเมทริกซ์พาริตีเช็ค H ดังแสดงในสมการที่ 3.7 ซึ่งกราฟนี้ถูกเรียกว่า “กราฟแทนเนอร์” กราฟแทนเนอร์จะประกอบไปด้วยกลุ่มโนดสองกลุ่มได้แก่ กลุ่มโนดบิต (Bit node) ที่เป็นตัวแทนของบิตรหัส และกลุ่มโนดเช็ค (Check node) ที่เป็นตัวแทนของบิตพาริตีเช็ค เพื่อความง่ายแก่การพิจารณา ผู้วิจัยจะสมมติให้ เมทริกซ์พาริตีเช็ค H ที่ใช้สำหรับเข้ารหัสบิตข้อมูล u จำนวน 5 บิต ให้เป็นคำรหัส v ความยาวเท่ากับ 10 บิต แสดงดังสมการที่ 3.10

$$H_{5 \times 10} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

พิจารณาเมทริกซ์พาริตีเช็ค H ในสมการที่ 3.10 เมื่อกำหนดให้ c_1, c_2, \dots, c_5 และ v_1, v_2, \dots, v_{10} คือเวกเตอร์ของแต่ละแถว และแต่ละหลักของเมทริกซ์พาริตีเช็ค ตามลำดับ โดยกลุ่มโนดเช็คของเมทริกซ์พาริตีเช็คจะประกอบไปด้วย 5 โนด ซึ่งสังเกตว่า จะมีความสอดคล้องกับจำนวนแถวของเมทริกซ์พาริตีเช็คเช่นเดียวกันกลุ่มของโนดบิต จะประกอบไปด้วย 10 โนด ซึ่งจะมีความสอดคล้องกับจำนวนหลักเช่นเดียวกัน ฉะนั้น หากพิจารณาเมทริกซ์พาริตีเช็คในสมการที่ 3.10 จะพบว่า ในแถวที่ c_i ประกอบไปด้วยเลข 1 ในหลักที่ 1, 3, 6 และ 8 ตามลำดับ ดังนั้น โนดเช็คที่ 1 หรือ c_1 จะเชื่อมต่อกับโนดบิตที่ 1, 3, 6 และ 8 ตามลำดับ อย่างไรก็ตาม สำหรับโนดเช็คที่ 2, 3, 4 และ 5 นั้น จะมีตำแหน่งของโนดบิตที่เชื่อมต่อกับตำแหน่งเลข 1 ในแต่ละหลักของมันเอง ตามลำดับ จะเห็นได้ว่าความสัมพันธ์หรือการเชื่อมต่อกันระหว่างโนดบิต และโนดเช็คสามารถพิจารณาได้จากตำแหน่งเลข 1 ในเมทริกซ์พาริตีเช็ค ยกตัวอย่างเช่น ตำแหน่งเลข 1 ที่ปรากฏในตำแหน่งแถวที่ 3 หลักที่ 4 ของเมทริกซ์พาริตีเช็คจะหมายถึง การเชื่อมต่อของโนดเช็คตัวที่ 3 กับโนดบิตตัวที่ 4 ดังนั้น การเชื่อมต่อระหว่างโนดเช็คแต่ละโนดกับโนดบิตแต่ละโนด สำหรับเมทริกซ์พาริตีเช็ค H ในสมการที่ 3.10 สามารถเขียนให้อยู่ในรูปแบบของกราฟสองส่วนหรือกราฟแทนเนอร์ได้ดังรูปที่ 3.4



รูปที่ 3.4 กราฟแทนเนอร์ของเมทริกซ์พาริตีเซ็คจากสมการที่ 3.10

จะเห็นได้ว่าเมื่อนำเมทริกซ์พาริตีเซ็ค \mathbf{H} มาวาดให้อยู่ในรูปแบบของกราฟสองส่วน หรือกราฟแทนเนอร์ดังแสดงในรูปที่ 3.4 จะสามารถพิจารณาได้ถึงความสัมพันธ์ระหว่างบิตแต่ละบิตของคำรหัสว่ามีความสัมพันธ์กันเช่นไร ยกตัวอย่างเช่น เมื่อนำเวกเตอร์บิตข้อมูล $\mathbf{u} = [u_1, u_2, u_3, \dots, u_5]$ มาเข้ารหัสกับเมทริกซ์พาริตีเซ็ค \mathbf{H} ในสมการที่ 3.10 ด้วยความสัมพันธ์จากสมการที่ 3.7 ก่อให้เกิดเป็นเวกเตอร์คำรหัส $\mathbf{v} = [v_1, v_2, v_3, \dots, v_{10}]$ ซึ่งเมื่อพิจารณาจากกราฟแทนเนอร์ในรูปที่ 4.3 จะพบว่า บิตรหัส v_1, v_3, v_6 และ v_8 จะมีความสัมพันธ์กันผ่านทางโนดเซ็คที่ 1 กล่าวคือ เมื่อนำบิตดังกล่าวมาบวกกันแบบมอดุโลสอง จะมีค่าเท่ากับ 0 เนื่องจากมันอาศัยความสัมพันธ์จากสมการที่ 3.7 มาเข้ารหัสนั่นเอง จะเห็นได้ว่า หน้าทีของแต่ละโนดเซ็คนั้น เป็นตัวที่ใช้เชื่อมความสัมพันธ์ระหว่างโนดบิตแต่ละบิต ซึ่งเมื่อเรานำบิตรหัสที่เชื่อมต่อกันภายใต้โนดเซ็คเดียวกัน มาบวกกันแบบมอดุโลสองจะมีค่าเท่ากับ 0 เสมอ ซึ่งในภาคถอดรหัสจะนำความสัมพันธ์ดังกล่าวมาพิจารณา และตัดสินใจว่าคำรหัสที่ถูกส่งผ่านช่องสัญญาณ ควรมีค่าเป็นอย่างไร อย่างไรก็ตาม อัลกอริทึมที่ใช้สำหรับพิจารณา และตัดสินใจบิตรหัสที่ถูกส่งผ่านช่องสัญญาณ มีด้วยกันหลายวิธี ซึ่งแต่ละวิธีก็จะมีข้อดีและเสียแตกต่างกันไป ซึ่งจะกล่าวถึงในหัวข้อถัดๆ ไปตามลำดับ อนึ่ง ค่าพารามิเตอร์สำคัญ ที่ส่งผลกระทบต่อสมรรถนะในการถอดรหัสแอลดีพีซี ทีซึ่งสามารถสังเกตได้จากกราฟแทนเนอร์นั้นคือ “ความยาวไซเคิล (Cycle)” โดยนิยามของความยาวไซเคิลหมายถึง จำนวนเส้นเชื่อมที่ใช้สำหรับเดินทางจากโนดบิตหนึ่งเพื่อกลับมาถึงโนดบิตเดิม โดยทั่วๆ ไปแล้ว รหัสแอลดีพีซีจะถูกออกแบบให้ปราศจากไซเคิลที่มีความยาวเท่ากับ 4 เนื่องจากมันจะส่งผลร้ายแรงต่อกระบวนการถอดรหัสแอลดีพีซี [26] อย่างไรก็ตาม ในปัจจุบันนี้ มีงานวิจัยหลายๆงาน ที่ได้แสดงถึงวิธีการจัดวางเส้นเชื่อมบนกราฟแทนเนอร์ (การจัดวางเลข 1 บนเมทริกซ์พาริตีเซ็ค) ให้มีตำแหน่งที่เหมาะสมที่สุด ตามเกณฑ์ที่กำหนด เพื่อที่จะทำให้สมรรถนะในการถอดรหัสแอลดีพีซีมีค่าดีที่สุดเท่าที่จะเป็นไปได้ ดังจะกล่าวถึงในหัวข้อถัดๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ประเภทของรหัสแอลดีพีซี

จากหัวข้อที่ 3.2.1 จะเห็นว่าองค์ประกอบที่เป็นตัวกำหนดความสัมพันธ์ของรหัสแอลดีพีซีคือ ตำแหน่งเลข 1 บนเมทริกซ์พาริตีเช็ค \mathbf{H} ซึ่งถูกแสดงเป็นเส้นเชื่อมที่เชื่อมต่อระหว่างโนดบิตและโนดเช็คบนกราฟแทนเนอร์นั่นเอง ดังนั้น ในหัวข้อนี้จะทำการจำแนกรหัสแอลดีพีซีออกเป็น 2 ประเภทหลักๆ ตามลักษณะการกระจายตัวของเลข 1 บนเมทริกซ์พาริตีเช็ค \mathbf{H} ดังนี้

3.2.2.1 รหัสแอลดีพีซีแบบคงที่ (Regular LDPC codes)

รหัสแอลดีพีซีแบบคงที่ เกิดจากการเข้ารหัสบิตข้อมูลกับเมทริกซ์พาริตีเช็ค \mathbf{H} ที่มีจำนวนเลข 1 ในแต่ละหลักของเมทริกซ์เท่ากันทุกหลัก และมีจำนวนเลข 1 เท่ากันในทุกๆ แถว ด้วยเช่นกัน กำหนดให้ d_v และ d_c คือจำนวนเลข 1 ในหลัก และแถวของเมทริกซ์พาริตีเช็ค \mathbf{H} ตามลำดับ หากพิจารณาในรูปแบบของกราฟแทนเนอร์จะพบว่า แต่ละโนดเช็คจะเชื่อมต่อกับโนดบิตจำนวน d_c บิต และแต่ละโนดบิตจะเชื่อมต่อกับโนดเช็คจำนวน d_v โหนด ตามลำดับ อย่างไรก็ตาม ในงานวิจัยของ T. J. Richardson [25] ได้อธิบายว่า เมื่อใช้เมทริกซ์พาริตีเช็ค \mathbf{H} ขนาด $m \times n$ มาเข้ารหัสแอลดีพีซีที่มีอัตรารหัสเท่ากับ $R = (n - m) / n$ ที่ซึ่งอัตรารหัส R จะมีความสัมพันธ์กับ d_v และ d_c ดังนี้

$$R = 1 - \frac{d_v}{d_c} \quad (3.11)$$

3.2.2.1 รหัสแอลดีพีซีแบบไม่คงที่ (Irregular LDPC codes)

ในทางตรงกันข้าม รหัสแอลดีพีซีแบบไม่คงที่คือ รหัสแอลดีพีซีที่เกิดจากการเข้ารหัสบิตข้อมูลกับเมทริกซ์พาริตีเช็ค \mathbf{H} ที่มีจำนวนเลข 1 ในแต่ละแถว และแต่ละหลักไม่คงที่ การออกแบบหรือจัดวางเลข 1 ในแต่ละแถว และหลักของเมทริกซ์พาริตีเช็คนั้น สามารถกำหนดได้จากสมการที่ 3.12 และ 3.13 ตามลำดับ โดยกำหนดให้ $\rho(X)$ และ $\lambda(X)$ คือ จำนวนเลข 1 ที่อยู่ในแถว และหลักตามลำดับ

$$\rho(X) = \sum_{i=1}^{d_c} \rho_i X^{i-1} \quad (3.12)$$

$$\lambda(X) = \sum_{i=1}^{d_v} \lambda_i X^{i-1} \quad (3.13)$$

โดยที่ d_v และ d_c แทนดีกรีสูงสุดของโนดบิต และโนดเช็ค ตามลำดับ

ρ_i แทนสัดส่วนระหว่างผลรวมของจำนวนสมาชิกที่ไม่เป็น 0 ของทุกแถวที่มีน้ำหนัก i ของเมทริกซ์พาริตีเช็ค กับผลรวมของจำนวนสมาชิกทุกตัวที่ไม่เป็น 0 ของเมทริกซ์พาริตีเช็คเมื่อ

$$\sum_{i=2}^{d_i} \rho_i = 1$$

λ_i แทนสัดส่วนระหว่างผลรวมของจำนวนสมาชิกที่ไม่เป็น 0 ของทุกหลักที่มีน้ำหนัก i ของเมทริกซ์พาริตีเช็ค กับผลรวมของจำนวนสมาชิกทุกตัวที่ไม่เป็น 0 ของเมทริกซ์พาริตีเช็คเมื่อ

$$\sum_{i=2}^{d_v} \lambda_i = 1$$

ฉะนั้น อัตรารหัส R สำหรับรหัสแอลดีพีซีแบบไม่สม่ำเสมอ สามารถคำนวณได้จากสมการที่ 3.14

$$R = 1 - \frac{\sum_{i=2}^{d_c} \rho_i / i}{\sum_{i=2}^{d_v} \lambda_i / i} \quad (3.14)$$

จะเห็นว่ากรรมวิธีในการออกแบบรหัสแอลดีพีซีแบบไม่คงที่ตามสมการที่ 3.12 – 3.13 มีรูปแบบในการคำนวณตำแหน่งในการจัดวางเลข 1 ที่มีความซับซ้อนสูงมาก เป็นผลให้รหัสแอลดีพีซีแบบไม่สม่ำเสมอ จะมีความซับซ้อนในการถอดรหัสที่สูงกว่ารหัสแอลดีพีซีแบบคงที่ ถึงแม้ T. J. Richardson [25] จะแสดงให้เห็นว่าสมรรถนะของรหัสแอลดีพีซีแบบไม่คงที่ จะให้สมรรถนะในการถอดรหัสที่เข้าใกล้ขีดจำกัดของแชนนอนถึง 0.0045 dB ที่ความยาวคำรหัสเท่ากับ 10^5 บิต อย่างไรก็ตาม รหัสแอลดีพีซีแบบไม่สม่ำเสมอ จะให้สมรรถนะที่ดีได้ก็ต่อเมื่อ อัตรารหัส R มีค่าน้อยกว่าหรือเท่ากับ $3/4$ เท่านั้น อีกทั้งความยาวของคำรหัสต้องมีค่ามากกว่า 5,000 บิต ด้วยเช่นกัน ฉะนั้น จึงเป็นการยากที่จะนำมาประยุกต์ใช้งาน เนื่องจากในแอปพลิเคชันส่วนใหญ่ในปัจจุบัน ต้องการอัตรารหัสที่สูงที่สุดเท่าที่จะเป็นไปได้ เพื่อที่จะได้อัตราเร็วในการส่งข้อมูลที่สูง ดังนั้น ในวิทยานิพนธ์เล่มนี้ จึงเน้นไปที่การประยุกต์ใช้รหัสแอลดีพีซีแบบสม่ำเสมอเท่านั้น

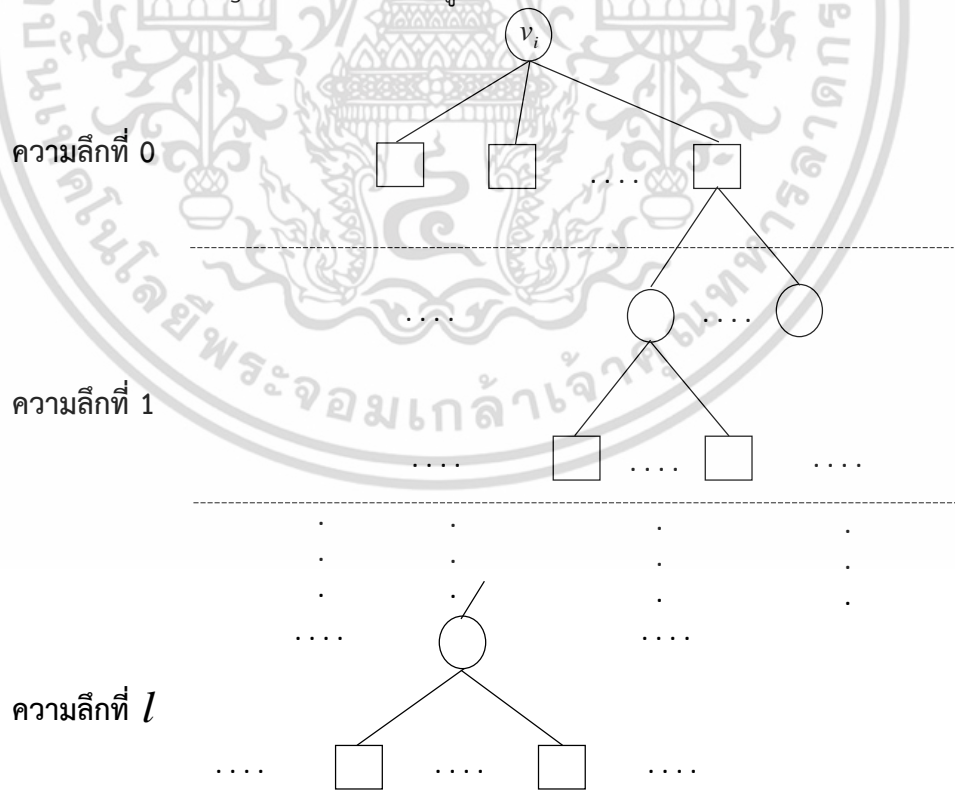
3.2.3 การออกแบบรหัสแอลดีพีซี

จากหัวข้อที่ 3.2.1-3.2.2 แสดงให้เห็นแล้วว่า การเชื่อมต่อกันระหว่างโนดบิต และโนดเช็คบนกราฟแทนเนอร์ หรือตำแหน่งการจัดวาง และการกระจายตัวของเลข 1 บนเมทริกซ์พาริตีเช็ค \mathbf{H} นั้น มีผลต่อสมรรถนะในการถอดรหัสแอลดีพีซี ดังนั้น กระบวนการออกแบบรหัสแอลดีพีซี จึงเป็นการออกแบบ และจัดวางเส้นเชื่อมระหว่างโนดบิต และโนดเช็คบนกราฟแทนเนอร์ โดยมีวัตถุประสงค์เพื่อเพิ่มสมรรถนะในการถอดรหัสหรือเพื่อลดความซับซ้อนในการถอดรหัส เป็นต้น อย่างไรก็ตาม การออกแบบรหัสแอลดีพีซีสามารถกระทำผ่านเมทริกซ์พาริตีเช็ค \mathbf{H} ได้ โดยการจัดวางเลข 1 ในเมทริกซ์พาริตีเช็คให้มีตำแหน่งที่เหมาะสมที่สุดในยุคแรกๆ ของรหัสแอลดีพีซี เมทริกซ์พาริตีเช็ค \mathbf{H} จะมีการจัดวางตำแหน่งเลข 1 แบบสุ่ม อย่างไรก็ตาม ในช่วงหลังๆ ได้มีการพิสูจน์ว่าการจัดวางตำแหน่งเลข 1 แบบสุ่ม ไม่เหมาะแก่การนำมาประยุกต์ใช้งาน ฉะนั้น จึงจำเป็นที่จะต้องมีการจัดวางตำแหน่งเลข 1 ให้มีแบบแผน ในปัจจุบัน การออกแบบรหัสแอลดีพีซีด้วยวิธีการจัดวางตำแหน่งเลข 1 บนเมทริกซ์พา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีเช็ค H ให้มีแบบแผน มีหลากหลายวิธีการด้วยกัน ยกตัวอย่างเช่น การจัดวางแบบอาร์เรย์, การจัดวางแบบมอดติไฟอาร์เรย์, การจัดวางแบบควอไซไซคลิก และการจัดวางแบบพีอีจี เป็นต้น ซึ่งในทุกๆ วิธีการ เป็นการออกแบบรหัสแอสตีพีซีที่ปราศจากไซเคิลความยาวเท่ากับ 4 ทั้งสิ้น อย่างไรก็ตาม ในวิทยานิพนธ์เล่มนี้ จะกล่าวถึงเฉพาะการออกแบบรหัสแอสตีพีซีด้วยวิธีพีอีจีเท่านั้น เนื่องจาก เป็นวิธีการที่จะนำมาประยุกต์ใช้กับอัลกอริทึมที่จะถูกนำเสนอในบทถัดๆ ไป

เป็นที่ทราบกันดีแล้วว่า ความยาวไซเคิลบนกราฟแทนเนอร์ จะส่งผลต่อสมรรถนะในการถอดรหัสรหัสแอสตีพีซี ฉะนั้น ในปี 2005 นักวิจัย X. Y. Hu ได้นำเสนอวิธีการออกแบบกราฟแทนเนอร์ ให้มีความยาวของไซเคิลใหญ่ที่สุดเท่าที่จะเป็นไปได้ โดยให้ชื่อวิธีการนี้ว่า “การออกแบบรหัสแอสตีพีซีด้วยการขยายเส้นเชื่อมแบบก้าวหน้า (Progressive edge growth: PEG)” [26] โดยค่าพารามิเตอร์ที่จำเป็นสำหรับการออกแบบรหัสแอสตีพีซีด้วยวิธีการพีอีจี นั้น ประกอบไปด้วย 1. โลคอลเกิร์ธ (Local girth) นิยามเป็น ความยาวของไซเคิลที่สั้นที่สุด ที่เดินทางจากแต่ละโนดบิตกลับมาหาโนดบิตเดิม และ 2. เกิร์ธ (Girth) นิยามเป็นไซเคิลที่สั้นที่สุดที่ปรากฏอยู่ในกราฟแทนเนอร์ โดยเป้าหมายของอัลกอริทึมพีอีจีคือ การทำให้โลคอลเกิร์ธของแต่ละโนดบิตมีขนาดใหญ่ที่สุดเท่าที่จะเป็นไปได้ ซึ่งจะส่งผลให้เกิร์ธมีขนาดใหญ่ขึ้นตามไปด้วย การสร้างโลคอลเกิร์ทให้มีขนาดใหญ่ที่สุดเท่าที่จะเป็นไปได้ นั้น สามารถกระทำได้โดยการแผ่กิ่งก้านสาขาของโนดบิต v_i ให้ลงไปถึงระดับความลึก (depth) ที่ l หรือความลึกสูงสุดเท่าที่จะเป็นไปได้ โดยโนดเช็คที่ปรากฏอยู่ที่ความลึกที่ l นั้น หากนำมาเชื่อมต่อกับโนดบิต v_i จะเป็นการเชื่อมต่อที่ทำให้ความยาวไซเคิลที่ใหญ่ที่สุดเท่าที่จะเป็นไปได้ โดยจะให้ความยาวไซเคิลที่มีค่าเท่ากับ $2(l+2)$ การเขียนลำดับของการแผ่กิ่งก้านสาขาของโนดบิต v_i จะถูกเรียกว่า “แผนภาพต้นไม้ (Tree diagram)” ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 การแผ่กิ่งก้านสาขาของโนดบิต v_i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบเมทริกซ์พาริตีเช็ค **H** ด้วยอัลกอริทึมพีอีจี (PEG)

```

for i มีค่าตั้งแต่ 1 จนถึง n-1
  for k มีค่าตั้งแต่ 0 จนถึง  $d_v-1$ 
    if k มีค่าเท่ากับ 1
      เลือกวางเลข 1 ในแถวหรือโนตเช็คที่มีจำนวนเลข 1 น้อยที่สุด หาก
      จำนวนมีเท่ากันหลายโนตเช็คให้ทำการสุ่มวางลงไปยังโนตใดโนตหนึ่ง
      ซึ่งจะไม่รับประกันว่าจะให้สมรรถนะสุดท้ายที่ดีที่สุด
    else
      ทำการขยายกิ่งก้านสาขาของโนตบิต  $v$ , ไปยังความลึกที่  $l$  ซึ่งหมายความว่า
      มีโนตเช็ค  $k$  โนตที่ไม่ซ้ำกัน ที่โนตบิต  $v$ , แผ่กระจายกิ่งก้านสาขาไป
      ถึง จากนั้นทำการวางเลข 1 ในตำแหน่งโนตเช็ค ที่ปรากฏอยู่ที่ความลึกที่
       $l$  ซึ่งจะให้เกิดโกลอกรีตมีขนาด  $2(l+2)$  ถ้าหากมีหลายโนตเช็คให้ทำการ
      วางแบบสุ่มลงไป (ไม่รับประกันว่าจะให้สมรรถนะสุดท้ายที่ดีที่สุด)
      หมายเหตุ ในช่วงแรกๆ การแพร่กระจายของโนตบิต  $v$ , อาจไม่สามารถ
      แพร่กระจายไปยังความลึกที่  $l$  ได้ ซึ่งในกรณีนี้ ให้ทำการวางเลข 1 ไปที่
      โนตเช็ค ที่โนตบิต  $v$ , แผ่กิ่งก้านสาขาไปไม่ถึง ซึ่งในกรณีนี้จะไม่ทำให้
      เกิดโกลอกรีต อย่างไรก็ตาม หากมีหลายโนตเช็คให้ทำการวางแบบสุ่ม
      ซึ่งไม่รับประกันว่าจะให้สมรรถนะสุดท้ายที่ดีที่สุด
    end
  end
end

```

3.3 อัลกอริทึมการถอดรหัสแอสดีพีซี

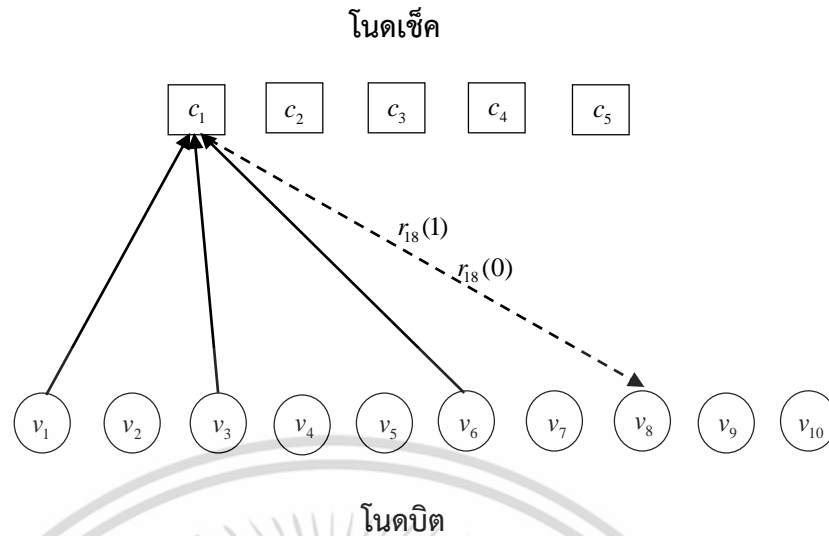
ภายหลังจากการเข้ารหัสแอสดีพีซีที่ภาคส่ง คำรหัส \mathbf{v} ที่ได้จะถูกมอดูเลตด้วยสัญญาณบีพีเอสเค (BPSK) เกิดเป็นสัญญาณเตรียมส่ง \mathbf{x} โดยที่ $\mathbf{x} = 2\mathbf{v} - 1$ จากนั้น มันจะถูกส่งผ่านตัวกลางหรือช่องสัญญาณสื่อสาร ที่ประกอบไปด้วยสัญญาณรบกวนชนิดต่างๆ ซึ่งภาครับจะมีวงจรถอดรหัส ที่ทำหน้าที่ในการถอดรหัสสัญญาณที่ได้รับ \mathbf{y} โดยที่ $\mathbf{y} = \mathbf{x} + \mathbf{n}$ เมื่อ \mathbf{n} คือสัญญาณรบกวน เพื่อกู้ข้อมูล \mathbf{v} จากภาคส่งกลับคืนมา ในหัวข้อนี้ จะกล่าวถึงการถอดรหัสแอสดีพีซีที่ภาครับด้วยวิธีการแบบต่างๆ ที่มีจุดเด่น และจุดด้อยของแต่ละวิธีการแตกต่างกันไป อย่างไรก็ตาม การถอดรหัสแอสดีพีซี จะอาศัยความสัมพันธ์ และการเชื่อมต่อกันระหว่างโนดบิต และโนดเซ็คบนกราฟแทนเนอร์ เป็นตัวกำหนดวิธีการ และขั้นตอนในการถอดรหัสแอสดีพีซี กรรมวิธีในการถอดรหัสแอสดีพีซีสามารถแบ่งออกเป็น 2 ประเภทหลักๆ ได้แก่

3.3.1 การถอดรหัสแอสดีพีซีชนิดอาศัยค่าซอฟต์แวร์ (Soft decision)

การถอดรหัสแอสดีพีซีชนิดอาศัยค่าซอฟต์แวร์หรือเรียกสั้นๆ ว่า “การถอดรหัสแอสดีพีแบบซอฟต์แวร์” โดยความหมายของค่าซอฟต์แวร์ในที่นี้คือ ค่าที่ใช้ในการบ่งบอกว่าแต่ละบิตที่ได้รับมาจากช่องสัญญาณ ควรจะเป็นบิต 0 หรือ 1 ด้วยความน่าจะเป็นเท่าไร การถอดรหัสแบบซอฟต์แวร์จะอาศัยค่าซอฟต์แวร์ของข้อมูลแต่ละบิต ส่งผ่านเส้นเชื่อมระหว่างโนดบิต และโนดเซ็ค ตามเส้นทางที่แสดงบนกราฟแทนเนอร์ เพื่อทำการตรวจสอบว่า แต่ละบิตนั้น มีค่าความน่าจะเป็นในท้ายที่สุดเป็นอย่างไร โดยวิธีการส่งผ่านข้อมูลซอฟต์แวร์ผ่านทางโนดบิต และโนดเซ็คจะถูกเรียกว่า “กระบวนการแพร่กระจายค่าความเชื่อมั่น (Belief propagation: BP)” ซึ่งในหัวข้อนี้ ผู้วิจัยจะขอยกตัวอย่างอัลกอริทึมในการถอดรหัสแอสดีพีด้วยการอาศัยกระบวนการแพร่กระจายค่าความเชื่อมั่น 3 อัลกอริทึมหลักๆ ซึ่งเนื้อหาของแต่ละอัลกอริทึมในหัวข้อ 3.3.1 นี้ ผู้วิจัยได้ดัดแปลงและเรียบเรียงเนื้อหาใหม่จากหนังสือ [36] ใจความดังนี้

3.3.1.1 อัลกอริทึมซัมโปรดักต์ (The sum-product algorithm: SPA)

อัลกอริทึมซัมโปรดักต์เป็นอัลกอริทึมพื้นฐานในการส่งผ่านค่าซอฟต์แวร์ระหว่างโนดบิต และโนดเซ็ค โดยมีเส้นทางเป็นไปตามเส้นทางของกราฟแทนเนอร์หรือตำแหน่งเลข 1 บนเมทริกซ์พาริตีเซ็ค \mathbf{H} อย่างไรก็ตาม การถอดรหัสด้วยวิธีอาศัยกระบวนการแพร่กระจายค่าความเชื่อมั่นนั้น กราฟแทนเนอร์ต้องปราศจากความยาวไซเคิลเท่ากับ 4 เนื่องจากจะมีผลร้ายแรงกับสมรรถนะของการถอดรหัสแอสดีพีซี ดังกล่าวไว้ใน [26] พิจารณากราฟแทนเนอร์ในรูปที่ 3.3 สามารถเขียนเส้นทางในการส่งผ่านข้อมูลแบบซอฟต์แวร์ระหว่างโนดเซ็ค ไปยังโนดบิต (แสดงเป็นเส้นประ) ได้ดังรูปที่ 3.6



รูปที่ 3.6 การส่งผ่านค่าซอฟต์แวร์จากโนตเช็คที่ 1 มายังโนตบิตที่ 8

จากกราฟแทนเนอร์ที่แสดงในรูปที่ 3.6 จะพบว่า โนตบิตแต่ละโนตที่เชื่อมต่อกันภายใต้โนตเช็คที่ 1 มีความสัมพันธ์กันคือ $v_1 + v_3 + v_6 + v_8 = 0$ (ตามสมการที่ 3.7) ฉะนั้น ข้อมูลแบบซอฟต์แวร์ที่ถูกส่งผ่านจากโนตเช็คที่ 1 มายังโนตบิตที่ 8 สามารถพิจารณาได้จากข้อมูลแบบซอฟต์แวร์ จากโนตบิตที่ 1, 3 และ 6 ที่ถูกส่งขึ้นไปยังโนตเช็คที่ 2 ซึ่งสามารถคำนวณได้จาก

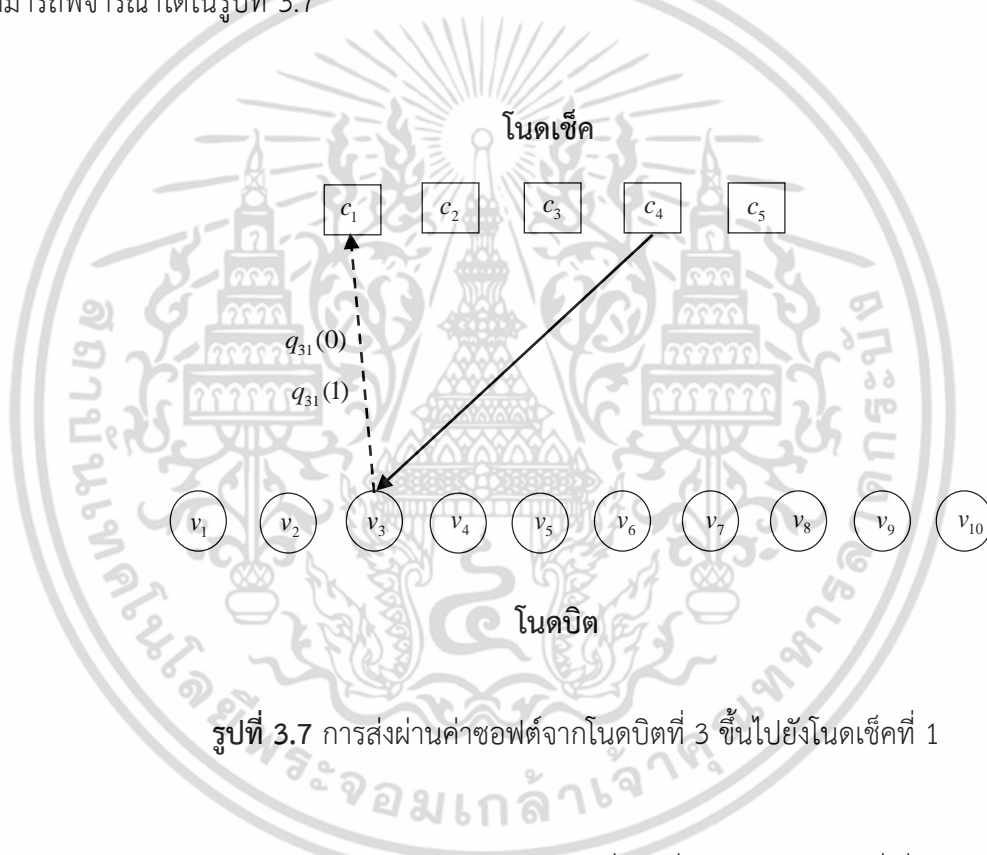
$$\begin{aligned}
 r_{18}(1) &= P(v_8 = 1, C_1 | y_i) \\
 &= P(v_8 = 1, v_1 + v_3 + v_6 + v_8 = 0 | y_i) \\
 &= P(v_1 + v_3 + v_6 = 1 | y_i) \\
 &= [p(v_1 = 1)p(v_3 = 0)p(v_6 = 0)] + [p(v_1 = 0)p(v_3 = 1)p(v_6 = 0)] + \dots \\
 &\quad [p(v_1 = 0)p(v_3 = 0)p(v_6 = 1)] + [p(v_1 = 1)p(v_3 = 1)p(v_6 = 1)]
 \end{aligned} \tag{3.15}$$

โดย C_1 แสดงถึง ความสัมพันธ์ของแต่ละโนตบิตภายใต้โนตเช็คที่ 1 จากสัญญาณที่ได้รับ y_i และ $P(v_3 = 0)$ หมายถึง ค่าความน่าจะเป็นจากโนตบิตที่ 3 ที่ถูกส่งขึ้นไปบอกโนตเช็คที่ 1 ซึ่งสามารถเขียนให้อยู่ในรูป $q_{31}(0)$ หมายความว่า โนตบิตที่ 3 ทำการส่งข้อมูลซอฟต์แวร์มาบอกโนตเช็คที่ 1 ว่าโนตบิตที่ 3 มีค่าความน่าจะเป็นที่จะเป็นบิต 0 เท่ากับเท่าไร ในทางตรงกันข้าม $q_{31}(1)$ ย่อมหมายถึง การที่โนตบิตที่ 3 ส่งข้อมูลซอฟต์แวร์มาบอกโนตเช็คที่ 1 ว่ามีค่าความน่าจะเป็นที่จะเป็นบิต 1 เท่ากับเท่าไร อย่างไรก็ตาม เมื่อนำสมการที่ 3.15 มาเขียนให้อยู่ในรูปสมการแบบกระชับ เพื่อที่จะสามารถนำมาใช้คำนวณค่าซอฟต์แวร์ที่ถูกส่งผ่านจากโนตเช็คมายังโนตบิตแต่ละโนต จะสามารถเขียนได้ดังสมการที่ 3.16 และ 3.17 ตามลำดับ

$$r_{ji}(0) = \frac{1}{2} - \frac{1}{2} \prod_{i' \in C_j \setminus i} (1 - 2q_{ij'}(1)) \quad (3.16)$$

$$r_{ji}(1) = 1 - r_{ji}(0) \quad (3.17)$$

โดยที่ $C_j \setminus i$ หมายความว่า ทำการพิจารณาเส้นเชื่อมจากโนดบิตที่ i' ทุกๆ โนด ที่เชื่อมต่อกับ โนดเซ็คที่ j โดยที่ไม่รวมเส้นเชื่อมจากโนดบิตที่ i (เส้นเชื่อมที่กำลังคำนวณ) ในส่วนกราฟแทนเนอร์ ที่แสดงถึง การส่งผ่านค่าซอฟต์แวร์หรือค่าความน่าจะเป็นจากโนดบิตใดๆ ไปยังโนดเซ็คที่เชื่อมต่อกัน สามารถพิจารณาได้ในรูปที่ 3.7



รูปที่ 3.7 การส่งผ่านค่าซอฟต์แวร์จากโนดบิตที่ 3 ขึ้นไปยังโนดเซ็คที่ 1

การส่งข้อมูลซอฟต์แวร์หรือค่าความน่าจะเป็นจากโนดบิตที่ i เพื่อไปบอกโนดเซ็คที่เชื่อมต่อกัน อยู่ในแต่ละเส้นทางนั้น สามารถคำนวณได้จากค่า r_{ji} จากแต่ละโนดเซ็คที่เชื่อมต่อกับโนดบิตที่ i ซึ่งสามารถเขียนเป็นสมการในการคำนวณได้ดังนี้

$$q_{ij}(0) = K_{ij}(1 - P_i) \prod_{j' \in V_i \setminus j} r_{ji'}(0) \quad (3.18)$$

$$q_{ij}(1) = K_{ij}P_i \prod_{j' \in V_i \setminus j} r_{ji'}(1) \quad (3.19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $V_i \setminus j$ หมายความว่า ทำการพิจารณาเส้นเชื่อมจากโนดเซตที่ j' ทุกๆ โนด ที่เชื่อมต่อกับ โนดบิตที่ i โดยที่ไม่รวมเส้นเชื่อมจากโนดเซตที่ j (เส้นเชื่อมที่กำลังคำนวณ) ซึ่ง $K_{i,j}$ คือค่าคงที่ ที่ทำให้ $q_{ij}(0) + q_{ij}(1) = 1$ และ P_i คือความน่าจะเป็นของโนดบิตที่ i จะมีค่าเท่ากับ 1 ซึ่งได้รับมาจาก ช่องสัญญาณสื่อสาร ในทางตรงกันข้าม $1 - P_i$ ย่อมหมายถึง ความน่าจะเป็นของโนดบิตที่ i จะมีค่าเท่ากับ 0 อย่างไรก็ตาม ในกรณีของช่องสัญญาณเกาส์สีขาว ค่า P_i สามารถหาได้จาก

$$P_i = \frac{1}{1 + e^{\frac{-2y_i}{\sigma^2}}} \quad (3.20)$$

ผู้วิจัยสามารถสรุปวิธีการถอดรหัสแอลดีพีซีด้วยอัลกอริทึมซิมโพรดักส์ได้ดังนี้

ขั้นเตรียมการ : กำหนดให้ตัวแปร X มีค่าเท่ากับ 0 จากนั้น ทำการกำหนดการวนซ้ำสูงสุดที่ ต้องการให้มีค่าเท่ากับ k

ขั้นตอนที่ 1 : คำนวณค่า $q_{ij}(0)$ และ $q_{ij}(1)$ ของทุกเส้นทางที่พุ่งออกมาจากแต่ละโนดบิตด้วยสมการ 3.18 – 3.19 หาก k มีค่าเท่ากับ 1 จะกำหนดให้ $q_{ij}(0)$ และ $q_{ij}(1)$ มีค่าเท่ากับ $1 - P_i$ และ P_i ตามลำดับ (คำนวณจากสมการที่ 3.20)

ขั้นตอนที่ 2 : คำนวณค่า $r_{ji}(0)$ และ $r_{ji}(1)$ ของทุกเส้นทางที่พุ่งเข้าแต่ละโนดบิต จากสมการ 3.16 - 3.17

ขั้นตอนที่ 3 : คำนวณ $X = X + 1$ จากนั้นทำการวนซ้ำ ขั้นตอนที่ 1 – 3 จนกว่า X จะมีค่าเท่ากับ k

ขั้นตอนที่ 4 : คำนวณค่าซอฟต์แวร์หรือค่าความน่าจะเป็นท้ายสุดของแต่ละโนดบิตด้วยสมการ

$$Q_i(0) = K_i (1 - P_i) \prod_{j \in V_i} r_{ji}(0) \quad (3.21)$$

$$Q_i(1) = K_i P_i \prod_{j \in V_i} r_{ji}(1) \quad (3.22)$$

โดย K_i คือค่าที่ทำให้ $Q_i(0) + Q_i(1) = 1$

ขั้นตอนที่ 5 : ทำการตัดสินใจด้วยความสัมพันธ์

$$c_i = \begin{cases} 1, & Q_i(1) \geq Q_i(0) \\ 0, & Q_i(0) > Q_i(1) \end{cases} \quad (3.23)$$

หมายเหตุ โดยปกติแล้ว เมื่อจำนวนรอบการวนซ้ำ k มีค่าประมาณ 10~20 สมรรถนะในการ ถอดรหัสแอลดีพีซีจะมีการลู่เข้า (Convergent) [26]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 อัลกอริทึมแบบล็อกซัมโปรดักส์ (Logarithmic sum-product algorithm)

พิจารณาอัลกอริทึมการถอดรหัสแวลดีฟี่ซี ด้วยอัลกอริทึมซัมโปรดักส์ ดังที่กล่าวถึงในหัวข้อก่อนหน้า จะพบว่ากระบวนการในการคำนวณค่า q และ r จะใช้โอเปอร์เรชันในการคูณเป็นจำนวนมาก ซึ่งในทางปฏิบัติ จำนวนวงจรมีผลต่อความซับซ้อนในการถอดรหัสอย่างมาก [13] ดังนั้น ในงานวิจัย [28] ได้มีการออกแบบอัลกอริทึมล็อกซัมโปรดักส์ ที่สามารถลดจำนวนวงจรมูลงจากอัลกอริทึมซัมโปรดักส์ แต่ยังคงให้สมรรถนะที่มีค่าเท่าเดิม โดยวิธีการถอดรหัสของอัลกอริทึมนี้ จะประยุกต์ใช้ค่าลอการิทึมธรรมชาติ สำหรับใช้ในการคำนวณค่าซอฟต์แวร์ q และ r ตามลำดับ อย่างไรก็ตาม สามารถนิยามอัตราส่วนความน่าจะเป็นแบบล็อก (Log Likelihood Ratio: LLR) ของค่าซอฟต์แวร์ต่างๆ ได้ดังนี้

$$L(c_i) = \log \left(\frac{\Pr(c_i = 0 | y_i)}{\Pr(c_i = 1 | y_i)} \right) \quad (3.24)$$

$$L(r_{ji}) = \log \left(\frac{r_{ji}(0)}{r_{ji}(1)} \right) \quad (3.25)$$

$$L(q_{ij}) = \log \left(\frac{q_{ij}(0)}{q_{ij}(1)} \right) \quad (3.26)$$

$$L(Q_i) = \log \left(\frac{Q_i(0)}{Q_i(1)} \right) \quad (3.27)$$

จากสมการที่ 3.17 ทำการแทน $r_{ji}(0)$ ด้วย $1 - r_{ji}(1)$ จะได้

$$1 - 2r_{ji}(1) = \sum_{i \in C_j \setminus i} (1 - 2q_{ij}(1)) \quad (3.28)$$

จากความสัมพันธ์ $\tanh \left(\frac{1}{2} \log \left(\frac{p_0}{p_1} \right) \right) = p_0 - p_1 = 1 - 2p_1$ ดังนั้น

$$\tanh \left(\frac{1}{2} L(r_{ji}) \right) = \prod_{i \in C_j \setminus i} \tanh \left(\frac{1}{2} L(q_{ij}) \right) \quad (3.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$L(r_{ji}) = 2 \tanh^{-1} \left(\prod_{i' \in C_j \setminus i} \tanh \left(\frac{1}{2} L(q_{i'j}) \right) \right) \quad (3.30)$$

กำหนดให้ $L(q_{ij}) = \alpha_{ij} \beta_{ij}$ โดย $\alpha_{ij} = \text{sign}[L(q_{ij})]$ และ $\beta_{ij} = |L(q_{ij})|$

$$\begin{aligned} L(q_{ij}) &= \left(\prod_{i' \in C_j \setminus i} \alpha_{i'j} \right) 2 \tanh^{-1} \left(\prod_{i' \in C_j \setminus i} \left(\frac{1}{2} \beta_{i'j} \right) \right) \\ &= \left(\prod_{i' \in C_j \setminus i} \alpha_{i'j} \right) 2 \tanh^{-1} \log^{-1} \log \left(\prod_{i' \in C_j \setminus i} \left(\frac{1}{2} \beta_{i'j} \right) \right) \\ &= \left(\prod_{i' \in C_j \setminus i} \alpha_{i'j} \right) 2 \tanh^{-1} \log^{-1} \sum_{i' \in C_j \setminus i} \log \left(\tanh \left(\frac{1}{2} \beta_{i'j} \right) \right) \end{aligned} \quad (3.31)$$

จากนั้นกำหนดให้ $\phi(x) = -\log[\tanh(x/2)] = \log((e^x + 1)/(e^x - 1))$ และ $\phi(x) = \phi^{-1}(x)$ ที่ $x > 0$ ดังนั้นสัดส่วนความน่าจะเป็นแบบล็อก ที่ถูกส่งจากโนดเข้าไปถึงแต่ละโนดบิตสามารถคำนวณได้จาก

$$L(r_{ji}) = \left(\prod_{i' \in C_j \setminus i} \alpha_{i'j} \right) \left(\phi \left(\sum_{i' \in C_j \setminus i} \phi(\beta_{i'j}) \right) \right) \quad (3.32)$$

จากนั้นสามารถคำนวณสัดส่วนความน่าจะเป็นแบบล็อก ที่ส่งจากโนดบิตไปยังโนดเข้าด้วยสมการด้านล่าง

$$L(q_{ij}) = L(c_i) + \sum_{j' \in V_i \setminus j} L(r_{ij'}) \quad (3.33)$$

และทำการเปลี่ยนค่าซอฟต์แวร์หรือสัดส่วนความน่าจะเป็นของแต่ละโนดบิตให้อยู่ในรูป

$$L(Q_i) = L(c_i) + \sum_{j' \in V_i} L(r_{ij'}) \quad (3.34)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการตัดสินค่าของแต่ละโนดบิตด้วยความสัมพันธ์

$$c_i \begin{cases} 1, L(Q) < 0 \\ 0, L(Q) \geq 0 \end{cases} \quad (3.35)$$

ขั้นตอนในการถอดรหัสแวลติฟิซี ด้วยอัลกอริทึมแบบล็อกซิมโพรดักส์ จะเหมือนกันกับวิธีการของอัลกอริทึมซิมโพรดักส์ โดยสามารถคำนวณค่าซอฟต์แวร์หรือค่าสัดส่วนค่าความน่าจะเป็น แบบล็อกจากโนดบิตไปโนดซีคได้จากสมการ 3.33 (หาก $k = 1$ ใช้สมการที่ 3.24) และสามารถคำนวณค่าสัดส่วนค่าความน่าจะเป็นแบบล็อกจากโนดซีคไปโนดบิตจากสมการ 3.32 ตามลำดับ จากนั้นเมื่อการวนซ้ำรอบสูงที่สุดมาถึง ให้คำนวณค่าความน่าจะเป็นแบบล็อกในท้ายที่สุด พร้อมทำการตัดสินใจค่าของโนดบิตแต่ละโนด ได้จากสมการที่ 3.34 และ 3.35 ตามลำดับ

3.3.1.3 อัลกอริทึมแบบมินซัม (The min-sum algorithm)

พิจารณาสมการในการคำนวณค่าสัดส่วนความน่าจะเป็นที่ถูกส่งผ่านจากโนดซีคมาโนดบิตในสมการที่ 3.32 จะพบว่า เป็นสมการที่มีการคำนวณที่มีความซับซ้อนค่อนข้างมาก เนื่องจากมีการซิมเมชันค่าซอฟต์แวร์จากหลายๆ โนดบิต ดังนั้น อัลกอริทึมมินซัมจึงถูกออกแบบมาเพื่อประมาณค่าสัดส่วนความน่าจะเป็นที่ถูกส่งผ่านจากโนดซีคมาโนดบิตให้มีการใช้ค่าพารามิเตอร์ที่น้อยลง และยังส่งผลให้ค่าที่ได้นั้นใกล้เคียงเดิม ซึ่งสมการ 3.32 จะถูกเขียนใหม่ดังนี้

$$\begin{aligned} L(r_{ji}) &= \left(\prod_{i' \in C_i \setminus i} \alpha_{i'j} \right) \left(\phi \left(\sum_{i' \in C_i \setminus i} \phi(\beta_{i'j}) \right) \right) \\ &\approx \left(\prod_{i' \in C_i \setminus i} \alpha_{i'j} \right) \left(\phi \left(\phi \left(\min_{i' \in C_i \setminus i} \beta_{i'j} \right) \right) \right) \\ &\approx \left(\prod_{i' \in C_i \setminus i} \alpha_{i'j} \right) \min_{i' \in C_i \setminus i} \beta_{i'j} \end{aligned} \quad (3.36)$$

3.3.2 การถอดรหัสแวลติฟิซีแบบไม่อาศัยค่าซอฟต์แวร์ (Hard decision)

การถอดรหัสแวลติฟิซีชนิดที่สองคือ การถอดรหัสแวลติฟิซีโดยไม่อาศัยค่าซอฟต์แวร์หรือที่เรียกกันว่า “การถอดรหัสแวลติฟิซีแบบหยาบ” การถอดรหัสชนิดนี้อาศัยค่าฮาร์ดของสัญญาณที่ได้รับ นำมาใช้ในการถอดรหัส โดยนิยามของค่าฮาร์ดคือ การตัดสินสัญญาณที่ได้รับด้วยค่าความน่าจะเป็นสูงสุด กล่าวคือ เมื่อเวกเตอร์ของสัญญาณที่ได้รับจากช่องสัญญาณคือ $\mathbf{y} = [y_1, y_2, \dots, y_n]$ จะสามารถหาค่าฮาร์ดของแต่ละบิตได้จาก $z_n = 1$ ก็ต่อเมื่อ $y_n \geq 0$ และ $z_n = 0$ ก็ต่อเมื่อ $y_n < 0$ จากนั้นจะนำเวกเตอร์

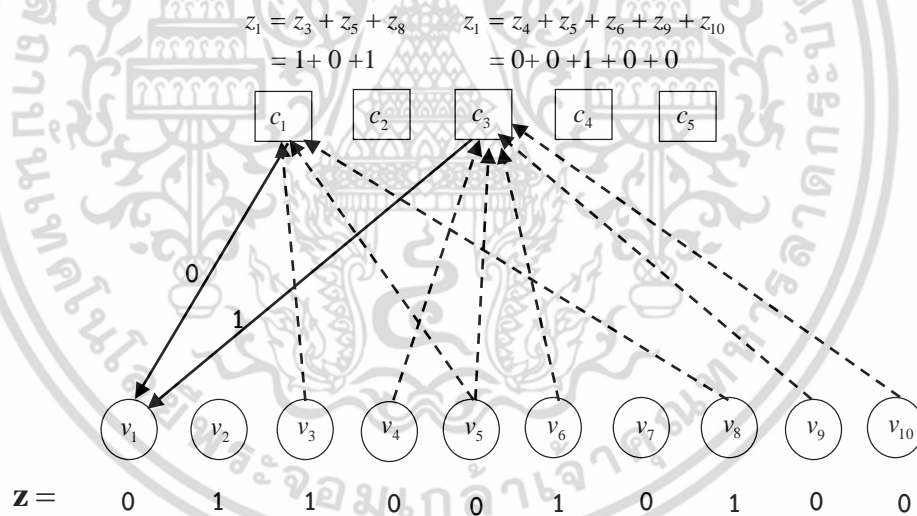
$\mathbf{z} = [z_1, z_2, \dots, z_n]$ มาใช้ในการถอดรหัสแวลติฟิซีแบบหยาบต่อไป ความแตกต่างอย่างเห็นได้ชัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการถอดรหัสแวลดีพีซีแบบอาศัยค่าซอฟต์แวร์ และแบบไม่อาศัยค่าซอฟต์แวร์คือ เมื่อมีการวนรอบที่สูงขึ้น ค่าความน่าจะเป็นหรือค่าซอฟต์แวร์ของการถอดรหัสแบบอาศัยค่าซอฟต์แวร์ จะมีการเพิ่มขึ้นหรือลดลงสะสมกันไปเรื่อยๆ ตามความถูกต้องของบิตๆ นั้น ซึ่งต่างจากการถอดรหัสแวลดีพีซีแบบหยาบซึ่งค่าดังกล่าวจะมีค่าที่คงที่ถูกรอบการวนซ้ำ ในหัวข้อต่อไปนี้จะทำการยกตัวอย่างอัลกอริทึมการถอดรหัสแวลดีพีซีแบบหยาบที่สำคัญดังนี้

3.3.2.1 อัลกอริทึมบิตพลิกปิ้ง (Bit-flipping algorithm: BF) [10]

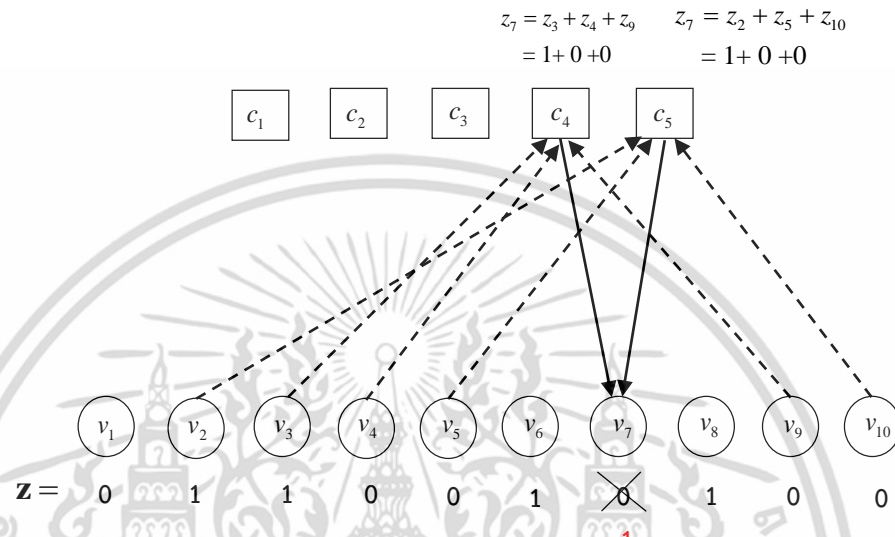
อัลกอริทึมบิตพลิกปิ้งเป็นอัลกอริทึมพื้นฐาน ที่ใช้สำหรับถอดรหัสแวลดีพีซีแบบหยาบ ซึ่งถูกคิดค้นมาพร้อมๆ กับรหัสแวลดีพีซี โดย R. Gallager อัลกอริทึมนี้เป็นอัลกอริทึมสำหรับถอดรหัสแวลดีพีซีแบบหยาบ โดยอาศัยความสัมพันธ์ระหว่างโนดบิต และโนดเซ็คบนกราฟแทนเนอร์ ผสมผสานด้วยวิธีการโหวตด้วยเสียงส่วนใหญ่ (Majority vote) สำหรับตัวอย่างการถอดรหัสแวลดีพีซีด้วยอัลกอริทึมบิตพลิกปิ้ง (BF) ผู้เขียนจะยกตัวอย่างด้วยกราฟแทนเนอร์รูปที่ 3.3 โดยทำการกำหนดให้สัญญาณที่ได้รับจากช่องสัญญาณคือ $\mathbf{y} = [-1.22, 1.56, 3.21, -0.02, -2.05, 2.14, -0.08, 0.86, -0.99, -3.23]$ ฉะนั้น เมื่อทำการคำนวณค่าฮาร์ตของสัญญาณที่รับแต่ละบิตจะได้ $\mathbf{z} = [0, 1, 1, 0, 0, 1, 0, 1, 0, 0]$ วิธีการถอดรหัสแบบบิตพลิกปิ้ง (BF) โดยอาศัยกราฟแทนเนอร์สามารถพิจารณาได้จากรูปที่ 3.8



รูปที่ 3.8 การถอดรหัสแวลดีพีซีด้วยอัลกอริทึมบิตพลิกปิ้งพิจารณาบิตที่ 1

จากกราฟแทนเนอร์ในรูป 3.8 เมื่อพิจารณาบิตที่ 1 จะพบว่าโนดบิตที่ 1 เชื่อมต่อกับโนดเซ็คที่ 1 และ 3 ตามลำดับ ซึ่งจากความสัมพันธ์ในสมการที่ 3.7 จะพบว่าทุกบิตรหัสที่เชื่อมต่อกับโนดเซ็คเดียวกัน จะบวกกันแบบมอดุโลเท่ากับ 0 เสมอ ซึ่งจะเห็นได้ว่าโนดเซ็คที่ 1 เชื่อมอยู่กับโนดบิตที่ 1, 3, 5 และ 8 ตามลำดับ ฉะนั้น ค่าของโนดบิตที่ 1 ที่ส่งมาจากโนดเซ็คที่ 1 จะมีค่าเท่ากับ $z_1 = z_3 + z_5 + z_8 = 1 + 0 + 1$ หรือมีค่าเท่ากับ 0 นั่นเอง อย่างไรก็ตาม โนดบิตที่ 1 จะได้รับค่า 1 มา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโนดเซ็คที่ 3 ด้วยวิธีการเดียวกัน จากนั้น จะทำการถอดรหัสบิตที่ 1 ด้วยวิธีโหวตด้วยเสียงส่วนใหญ่ (Majority vote) จากกราฟแทนเนอร์จะพบว่าบิตที่ 1 มีจำนวนในการโหวตเท่ากับ 3 เสียง อันได้แก่ ค่าที่ส่งมาจากโนดเซ็คที่ 1, ค่าที่ส่งมาจากโนดเซ็คที่ 3 และค่าฮาร์ดที่ได้รับจากช่องสัญญาณ ซึ่งจะเห็นได้ว่าบิตที่ 1 นั้นได้รับการโหวตให้เป็นบิตที่ 0 มากที่สุด ดังนั้น ค่าฮาร์ดของบิตที่ 1 จะยังคงมีค่าเท่ากับ 0 เช่นเดิม



รูปที่ 3.9 การถอดรหัสแอลดีพีซีด้วยอัลกอริทึมบิตพลิกบิตที่ 7

การถอดรหัสแอลดีพีซีแบบบิตพลิกบิต (BF) จะกระทำต่อโนดบิตเรียงจากโนดบิตที่ 1 ถึงบิตที่ n ตามลำดับ หากพิจารณากราฟแทนเนอร์รูปที่ 3.9 ที่แสดงการถอดรหัสบิตที่ 7 จะพบว่าบิตที่ 7 นี้เชื่อมต่ออยู่กับโนดเซ็คที่ 4 และ 5 ตามลำดับ โดยที่โนดเซ็คที่ 4 เชื่อมต่อกับโนดบิตที่ 3, 4, 7 และ 9 ตามลำดับ ซึ่งจากความสัมพันธ์ $z_3 + z_4 + z_7 + z_9 = 0$ ฉะนั้น โนดเซ็คที่ 4 จึงได้ส่งค่า 1 ไปยังโนดบิตที่ 7 ในทำนองเดียวกัน โนดเซ็คที่ 5 ก็ทำการส่งค่า 1 ไปยังโนดบิตที่ 7 ด้วยเช่นกัน หลังจากนั้น เมื่อมีการโหวตด้วยเสียงส่วนใหญ่ จะพบว่าโนดบิตที่ 7 ถูกเชื่อว่าเป็นบิตที่มีค่าเท่ากับ 1 มากที่สุด ดังนั้น อัลกอริทึมการถอดรหัสแบบบิตพลิกบิต (BF) จะทำการพลิกบิตในตำแหน่งที่ 7 จากค่า 0 ให้เป็น 1

อันเนื่องมาจาก การถอดรหัสแอลดีพีซีมีกระบวนการทำงานแบบวนซ้ำ ซึ่งภายหลังจากทำการถอดรหัสในบิตที่ n เรียบร้อยแล้ว จะเรียกว่าเป็นการถอดรหัส 1 รอบ หลังจากนั้น จะกลับมาถอดรหัสในบิตที่ 1 จนถึง n อีกครั้ง จนกระทั่งครบจำนวนการวนซ้ำสูงสุดที่กำหนดไว้ หรือจนกว่าโนดบิตทุกตัวจะบวกกันแบบมอดุโลสองได้เท่ากับ 0 ภายใตโนดเซ็คของมัน อย่างไรก็ตาม บิตที่ถูกพลิกในการวนซ้ำรอบที่ k อาจจะถูกพลิกอีกครั้งในรอบที่ $k+c$, $c \in I^+$ เนื่องจาก ภายหลังจากการพลิกบิตใดๆ ไปนั้น โนดเซ็คที่เชื่อมต่อกับบิตที่ถูกพลิก จะถูกอัปเดตใหม่เสมอ ฉะนั้น หากการโหวตในครั้งใดก็ตาม ที่มีจำนวนเสียงที่เท่ากัน ให้ทำการละเว้นบิตนั้นไปก่อน จากนั้นจึงจะทำการกลับมาถอดอีกครั้งในการวนซ้ำรอบถัดไป เพื่อเป็นการหลีกเลี่ยงปัญหากรณีข้างต้น หากต้องการที่จะถอดรหัส

แอลดีพีซีด้วยอัลกอริทึมนี้ เมทริกซ์พาริตีที่เช็คต้องถูกออกแบบโดยกำหนดให้ d มีค่าเป็นจำนวนคู่ เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น เมื่อคุณผู้เห็นเห็นเอกสารนี้ กรุณาอย่าเผยแพร่หรือแจกจ่ายเอกสารนี้แก่ผู้อื่นโดยไม่ได้รับอนุญาตจากผู้จัดทำเอกสารนี้ หากฝ่าฝืน อาจต้องรับผิดชอบต่อความเสียหายที่เกิดขึ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนในการถอดรหัสแอสติฟิซีแบบบิตพลิกบั้ง (BF)

for k จะมีค่าตั้งแต่ 1 ถึงจำนวนรอบการวนซ้ำสูงสุดที่กำหนดไว้

for i จะมีค่าตั้งแต่ 1 ถึง n

รวบรวมค่าที่ส่งมาจากโน้ตเช็คแต่ละโน้ต ที่เชื่อมต่ออยู่กับโน้ตบิตที่ i
 จากนั้น นำทุกค่ามาพิจารณาร่วมกับค่าฮาร์ตของบิตที่ i ว่าเสียงส่วนใหญ่
 ตัดสินใจให้บิตที่ i มีค่าเป็นเท่าไร ขั้นสุดท้าย ทำการพลิกบิต หากเสียง
 ส่วนใหญ่ มีค่าที่ตรงกันข้ามกับค่าฮาร์ตของบิตที่ i
 หมายเหตุ หากการโหวตนั้นมีจำนวนเสียงที่เท่ากันให้ละเว้นโน้ตบิตที่ i ไป
 ก่อน

end

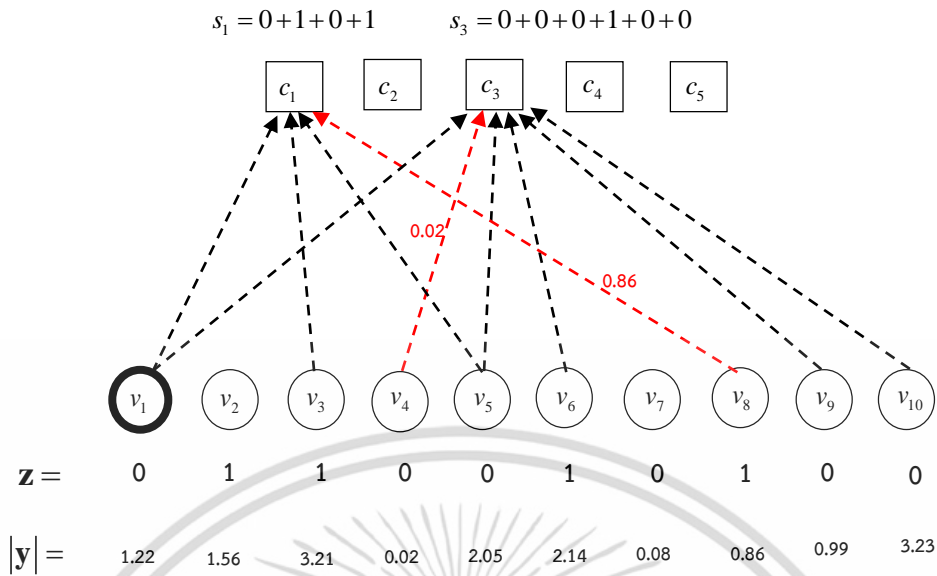
เช็คความถูกต้องของทุกบิตด้วยความสัมพันธ์ในสมการที่ 3.7 หากทุกบิตมีค่าถูกต้องจะหยุด
 การทำงานของอัลกอริทึมทันที

end

3.3.2.2 อัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนัก (Weighted bit-flipping: WBF) [11]

ในหัวข้อนี้ จะอธิบายถึงอัลกอริทึมการถอดรหัสแอสติฟิซีแบบบิตพลิกบั้งชนิดถ่วงน้ำหนัก (WBF) ที่เป็นการถอดรหัสแอสติฟิซีแบบหลายชนิดหนึ่ง ที่นอกจากจะอาศัยค่า \mathbf{z} มาใช้ในการถอดรหัสแล้ว อัลกอริทึมนี้ยังเป็นอัลกอริทึมแรก ที่นำค่าขนาดของสัญญาณที่ได้รับ $|y|$ มาพิจารณาใช้ในการถอดรหัสด้วย เนื่องจากขนาดของสัญญาณที่ได้รับจากช่องสัญญาณเกาส์สีขาว สามารถบ่งชี้ได้ถึงความน่าเชื่อถือของแต่ละโน้ตบิต โดยที่หาก $|y|$ มีค่ามากหมายถึง บิตที่มีความน่าเชื่อถือสูง ในทางตรงกันข้าม หาก $|y|$ มีค่าน้อย ย่อมหมายถึง บิตที่มีความน่าเชื่อถือต่ำ ตามลำดับ อัลกอริทึมนี้ถูกคิดค้นโดย Y. Kou ในปี 1998 โดยนำเสนอเพื่อปรับปรุงสมรรถนะการถอดรหัสของอัลกอริทึมบิตพลิกบั้ง (BF) เพื่อความง่ายในการพิจารณา ผู้วิจัยจะยกตัวอย่างให้สัญญาณที่ได้รับจากช่องสัญญาณสื่อสาร มีค่าเท่ากับ $\mathbf{y} = [-1.22, 1.56, 3.21, -0.02, -2.05, 2.14, -0.08, 0.86, -0.99, -3.23]$ และค่าฮาร์ตของสัญญาณที่ได้รับมีค่าเท่ากับ $\mathbf{z} = [0, 1, 1, 0, 0, 1, 0, 1, 0, 0]$ วิธีการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนัก (WBF) จะทำการคำนวณค่าความไม่น่าเชื่อถือ E_n ของแต่ละโน้ตบิต จากนั้น จะทำการพลิกบิตที่มีค่าความไม่น่าเชื่อถือสูงสุด (มีค่า E_n สูงที่สุดในแต่ละรอบการวนซ้ำ) ขั้นตอนในการได้มาของค่า E_n ของแต่ละบิต สามารถแบ่งเป็น 2 กระบวนการหลักๆคือ กระบวนการประมวลผลโน้ตเช็ค และ กระบวนการประมวลผลโน้ตบิต ตามลำดับ พิจารณากราฟแทนเนอร์ในรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 1 ในกระบวนการประมวลผลโนตเช็ค

กราฟแทนเนอร์ในรูปที่ 3.10 แสดงขั้นตอนในการคำนวณค่า E_n ของบิตที่ 1 ในกระบวนการประมวลผลโนตเช็ค จากรูปพบว่า โนตบิตที่ 1 เชื่อมต่อกับโนตเช็คที่ 1 และ 3 ตามลำดับ จากนั้น โนตเช็คที่ 1 และ 3 จะถูกนำมาคำนวณค่าซินโดรม s_m โดยที่ m มีค่าเท่ากับ 1 และ 3 นิยามของค่าซินโดรมคือ ค่าที่บ่งบอกถึงบิตรหัสทั้งหมดที่เชื่อมต่อภายใต้โนตเช็คหนึ่งๆ ว่ามีความน่าเชื่อถือมากน้อยเพียงใด ค่าซินโดรม s_m สามารถคำนวณได้จากการบวกกันแบบมอดุโลสองของค่าฮาร์ดจากโนตบิตที่เชื่อมต่อภายใต้โนตเช็คเดียวกัน เพราะฉะนั้น หาก $s_m = 0$ ย่อมหมายถึงบิตที่เชื่อมต่อกับโนตเช็คที่ m มีความน่าเชื่อถือสูงหรือมีโอกาสที่จะเป็นบิตที่ถูกตัดอยู่แล้ว เนื่องจากมีความสัมพันธ์เป็นไปตามสมการที่ 3.7 ในตรงกันข้าม หาก $s_m = 1$ ย่อมหมายถึง บิตที่เชื่อมต่อกับโนตเช็คที่ m มีความน่าเชื่อถือต่ำหรือมีโอกาสที่จะเป็นบิตที่ผิดพลาด วิธีในการคำนวณค่า s_m ของแต่ละโนตเช็คสามารถจัดให้อยู่ได้รูปของสมการได้ดังนี้

$$s_m = \sum_{n=1}^N z_n H_{mn} \tag{3.37}$$

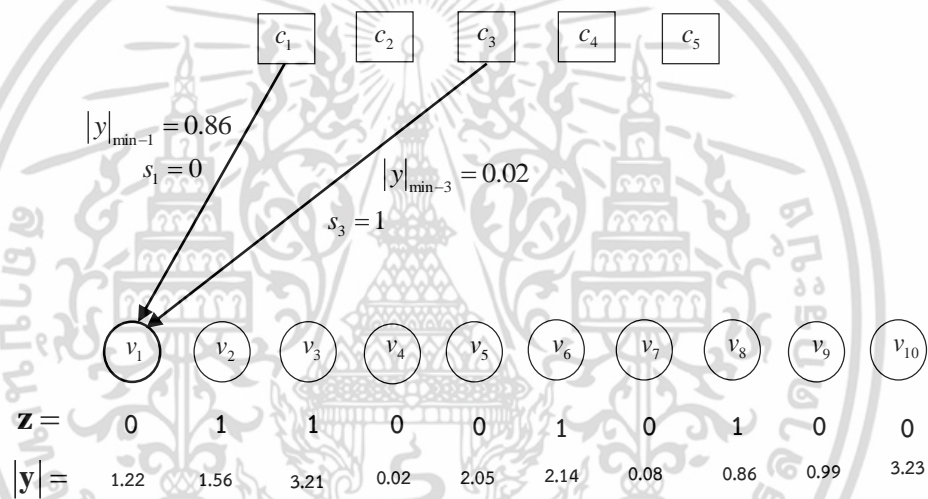
โดยที่ N คือ จำนวนของโนตบิตหรือความยาวคำรหัส ส่วน H_{mn} คือ ค่าในแถวที่ m และหลักที่ n ของเมทริกซ์พาริตีเช็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายหลังจากการคำนวณค่าซินโดรม s_m ของโนดเชคที่ 1 และ 3 เสร็จเรียบร้อยแล้ว โนดเชคที่ 1 และ 3 จะทำการคัดเลือกค่าสัญญาณที่ได้รับที่มีขนาดต่ำสุด $|y|_{\min-m}$ (โนดบิตที่มีความน่าเชื่อถือต่ำสุดของแต่ละโนดเชค) จากโนดบิตทั้งหมดภายใต้โนดเชคที่ 1 และ 3 ซึ่งจากกราฟแทนเนอร์ในรูปที่ 3.10 จะเห็นว่า โนดเชคที่ 1 และ 3 จะเลือกค่า 0.86 และ 0.02 ตามลำดับ โดยวิธีการคำนวณค่า $|y|_{\min-m}$ ของแต่ละโนดเชคนั้น สามารถจัดให้อยู่ได้รูปของสมการได้ดังนี้

$$|y|_{\min-m} \triangleq \min_{n \in N(m)} |y_n| \quad (3.38)$$

โดย $N(m)$ คือ เซตของตำแหน่งเลข 1 ในแถวที่ m ของเมทริกซ์พาริตีเชค และ $|y_n|$ คือค่าขนาดของสัญญาณที่ได้รับ



รูปที่ 3.11 การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 1 ในกระบวนการประมวลผลโนดบิต

พิจารณาในรูปที่ 3.11 ซึ่งแสดงกระบวนการประมวลผลโนดบิตของบิตที่ 1 เพื่อหาค่าความไม่น่าเชื่อถือของบิตที่ 1 หรือ E_1 ซึ่งจะพบว่าบิตที่ 1 จะทำการรับค่า s_m และ $|y|_{\min-m}$ จากโนดเชคที่ 1 และ 3 ตามลำดับ หลังจากนั้น จะนำค่าดังกล่าวมาคำนวณหาค่าความไม่น่าเชื่อถือ E_n ของโนดบิตที่ 1 วิธีการในการคำนวณค่า E_n ของแต่ละโนดบิตสามารถคำนวณได้จาก

$$E_n = \sum_{m \in M(n)} (2s_m - 1) |y|_{\min-m} \quad (3.39)$$

โดยที่ $M(n)$ คือเซตของตำแหน่งเลข 1 ในหลักที่ n ของเมทริกซ์พาริตีเชค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 3.39 จะเห็นได้ว่า หากบิตใดที่ E_n มีค่าเป็นบวก บิตนั้นย่อมมีค่าความน่าเชื่อถือต่ำ เนื่องจากเครื่องหมายของค่า E_n สำหรับบิตใดๆนั้น ถูกกำหนดจากค่าซินโดรม s_m อย่างไรก็ตาม หากพิจารณาเฉพาะค่า E_n ที่มีค่าเป็นบวก จะพบว่า สำหรับค่า E_n ที่มีแอมพลิจูดสูง จะมีค่าความน่าเชื่อถือที่ต่ำลงไปอีก เนื่องจากแอมพลิจูดของค่า E_n จะถูกกำหนดจากค่า $|y|_{\min-m}$ ฉะนั้น อัลกอริทึมบิตฟลิปบิงชนิดถ่วงน้ำหนัก (WBF) จะทำการฟลิปบิตในตำแหน่งที่มีค่า E_n สูงสุดในแต่ละรอบการวนซ้ำ

ขั้นตอนในการถอดรหัสแอลดีพีซีแบบบิตฟลิปบิงชนิดถ่วงน้ำหนัก (WBF)

for k จะมีค่าตั้งแต่ 1 ถึงการวนซ้ำรอบสูงสุดที่กำหนดไว้

กระบวนการประมวลผลโนตเช็ค

If k มีค่าเท่ากับ 1

ทำการคำนวณค่า s_m ของแต่ละโนตเช็คจากสมการที่ 3.37 จากนั้นทำการคำนวณค่า $|y|_{\min-m}$ ของแต่ละโนตเช็คจากสมการที่ 3.38

else

ทำการคำนวณค่า s_m และ $|y|_{\min-m}$ เฉพาะโนตเช็คที่เชื่อมต่อกับบิตที่ถูกฟลิปไปในรอบที่ k-1

end

กระบวนการประมวลผลโนตบิต

If k มีค่าเท่ากับ 1

ทำการคำนวณค่า E_n ของแต่ละโนตบิตจากสมการที่ 3.39

else

ทำการคำนวณค่า E_n จากสมการที่ 3.39 เฉพาะบิตที่เชื่อมต่อกับโนตเช็คที่ทำการคำนวณในกระบวนการประมวลผลโนตเช็ค

end

จัดเรียงค่า E_n จากน้อยไปมาก จากนั้นฟลิปบิตในตำแหน่งที่มีค่า E_n สูงที่สุด พร้อมทั้งตรวจสอบความถูกต้องของทุกบิตจากความสัมพันธ์ในสมการที่ 3.7 หากทุกบิตถูกต้องจะหยุดอัลกอริทึมทันที

end

3.3.2.3 อัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (Modified weighted bit-flipping: M-WBF) [12]

อัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (M-WBF) ถูกคิดค้นโดย J. Zhang ในปี 2004 อัลกอริทึมนี้สามารถปรับปรุงสมรรถนะในการถอดรหัส เมื่อเทียบกับอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักพื้นฐาน (WBF) อันเนื่องมาจาก ขนาดของสัญญาณที่ได้รับจากช่องสัญญาณเกาส์สีขาว สามารถบ่งชี้ได้ถึงความน่าเชื่อถือของแต่ละโนดบิต ดังนั้น อัลกอริทึม M-WBF จึงนำค่า $|y_n|$ ของตัวมันเอง มาทำการนอร์มัลไลซ์ด้วยการลบ ในขั้นตอนคำนวณค่า E_n เพื่อเพิ่มความแม่นยำในการคัดเลือกบิตมา พลิก โดยที่หาก $|y_n|$ มีค่ามาก (บิตน่าเชื่อถือสูง) จะส่งผลให้ค่า E_n มีค่าต่ำลงมาก ในตรงกันข้ามหาก $|y_n|$ มีค่าน้อย (บิตน่าเชื่อถือต่ำ) จะส่งผลให้ค่า E_n ลดลงไม่มาก ขั้นตอนการถอดรหัสของอัลกอริทึม M-WBF จะเหมือนกันกับอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักพื้นฐาน (WBF) เพียงแต่ในกระบวนการประมวลผลโนดบิตนั้น สมการในการคำนวณค่าความไม่น่าเชื่อถือ E_n จะถูกแก้ไขใหม่ดังสมการที่ 3.40

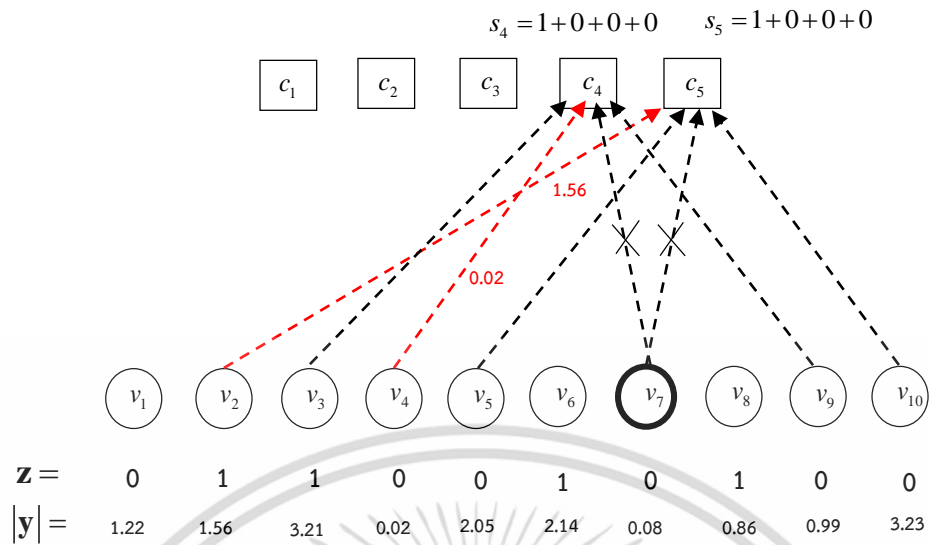
$$E_n = \left[\sum_{m=N(m)} (2s_m - 1) |y_{\min-m}| \right] - \alpha |y_n| \quad (3.40)$$

โดย α คือค่า weight factor ที่ได้จากการจำลอง ซึ่งจะต้องทำการหาค่าที่เหมาะสมที่สุดในแต่ละ E_b/N_0 และ d_v อย่างไรก็ตาม ค่า α นี้ สามารถประมาณการได้จาก [13]

3.3.2.4 อัลกอริทึมปรับปรุงบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง

(Improved modified weighted bit-flipping: IM-WBF) [13]

อัลกอริทึมปรับปรุงบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (IM-WBF) ถูกคิดค้นโดย M. Shan ในปี 2005 อัลกอริทึมนี้สามารถปรับปรุงสมรรถนะในการถอดรหัส เมื่อเทียบกับอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักพื้นฐาน (WBF) และอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (M-WBF) อัลกอริทึมนี้แก้ไขขั้นตอนในการถอดรหัสของอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักพื้นฐาน ทั้งในส่วนของการประมวลผลโนดซีค และกระบวนการประมวลผลโนดบิต พิจารณากราฟแทนเนอร์ในรูปที่ 3.12



รูปที่ 3.12 การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 7 ในกระบวนการประมวลผลโนตเช็ค

พิจารณารูปแทนเนอร์รูปที่ 3.12 ที่แสดงเป็นกระบวนการคำนวณค่าความไม่น่าเชื่อถือ E_n ของบิตที่ 7 ในส่วนของกระบวนการประมวลผลโนตเช็ค กระบวนการทำงานของอัลกอริทึมปรับปรุงบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบตัดแปลง (IM-WBF) ในส่วนของการคำนวณค่า s_m จะยังคงเหมือนกับการคำนวณของอัลกอริทึมบิตฟลิปบิงชนิดถ่วงน้ำหนักพื้นฐาน (WBF) ที่เป็นการบวกกันแบบมอดุโลสองของค่าฮาร์ตของแต่ละโนตบิตภายใต้โนตเช็คที่เชื่อมต่อกับโนตบิตที่ 7 แต่ความแตกต่างของสองอัลกอริทึมคือ กระบวนการคัดเลือกค่า $|y|_{\min-m}$ โดยที่อัลกอริทึมปรับปรุงบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบตัดแปลง (IM-WBF) จะไม่นำบิตที่ 7 มาพิจารณาด้วย โดยจะทำการตัดเส้นเชื่อมที่เชื่อมมายังโนตบิตที่ 7 ทิ้งไป กล่าวคือ อัลกอริทึมนี้ จะไม่พิจารณาเส้นเชื่อมจากบิตของตัวเอง ฉะนั้น สามารถเขียนสมการในการคำนวณค่า $|y|_{\min-m}$ ใหม่ได้ดังนี้

$$|y|_{\min-m} \triangleq \min_{n' \in N(m)_n} |y_{n'}| \tag{3.41}$$

โดย $|y|_{\min-m}$ คือ สัญญาณที่ได้รับที่มีขนาดต่ำสุดภายใต้โนตเช็คที่ m โดยที่ไม่นำขนาดของสัญญาณที่ได้รับในบิตที่ n มาพิจารณาด้วย ฉะนั้น ในส่วนการคำนวณค่าความไม่น่าเชื่อถือ E_n ของกระบวนการประมวลผลโนตบิต สำหรับอัลกอริทึมปรับปรุงบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบตัดแปลง (IM-WBF) จะถูกแก้ไขใหม่ดังนี้

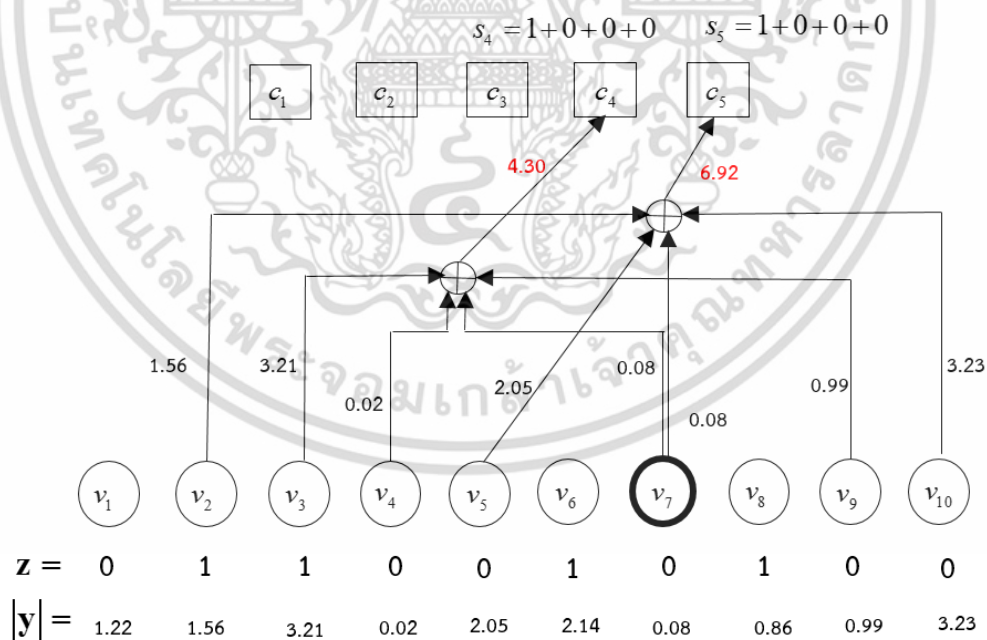
$$E_n = \left[\sum_{m=N(m)} (2s_m - 1) |y|_{\min-m} \right] - \alpha |y_n| \tag{3.42}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่า α คือค่า weight factor ที่ได้จากการจำลอง ซึ่งจะต้องทำการหาค่าที่เหมาะสมที่สุดในแต่ละ E_b/N_0 และ d_v ส่วนขั้นตอนการถอดรหัสของอัลกอริทึมนี้ จะเหมือนกันกับอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักพื้นฐาน (WBF) เพียงแต่เปลี่ยนสมการในการคำนวณค่า $|y|_{\min-m}$ และ E_n ตามลำดับ

3.3.2.5 อัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (reliability ratio weighted bit-flipping: RR-WBF) [14-15]

เป็นที่ทราบกันดีว่าอัลกอริทึมปรับปรุงบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (IM-WBF) จะให้สมรรถนะในการถอดรหัสที่ดีที่สุดสำหรับการถอดรหัสแบบหายาในปัจจุบัน โดยมีสมรรถนะขึ้นอยู่กับค่า weight-factor α ที่ต้องทำการค้นหาค่าที่เหมาะสมที่สุดในแต่ละ E_b/N_0 และ d_v ต่างๆ ซึ่งจะได้มาจากการจำลอง หรือสามารถประมาณการได้จากสมการใน [13] ที่มีสมการในการคำนวณค่อนข้างยาก และซับซ้อน อย่างไรก็ตาม สมรรถนะของอัลกอริทึมดังกล่าว จะถูกลดทอนอย่างมากหากค่า α มีความคลาดเคลื่อน จากค่าที่เหมาะสมที่สุด [13] ดังนั้นในปี 2005 นักวิจัยชื่อ F. Guo ได้นำเสนออัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) ที่สามารถหลีกเลี่ยงผลกระทบจากค่า α โดยจะยังให้สมรรถนะในการถอดรหัสเทียบเคียงกับอัลกอริทึมปรับปรุงบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบดัดแปลง (IM-WBF) โดยกระบวนการประมวลผลโนดเช็ค และโนดบิตจะถูกแก้ไขใหม่ดังนี้



รูปที่ 3.13 การคำนวณค่าความไม่น่าเชื่อถือของบิตที่ 7 ในกระบวนการประมวลผลโนดเช็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกระบวนการประมวลผลโนดเช็คเพื่อคำนวณค่าความไม่น่าเชื่อถือ E_n ของโนดบิตที่ 7 สำหรับ อัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) นั้น ค่าซินโดรม s_m ยังคงคำนวณด้วยวิธีการบวกกันแบบมอดุโลของค่าฮาร์ดแต่ละบิตภายใต้โนดเช็คที่เชื่อมต่อกับโนดบิตที่ 7 เช่นเดิม อย่างไรก็ตาม จุดเด่นของอัลกอริทึมนี้คือ ไม่มีการคำนวณค่า $|y|_{\min-m}$ แต่ค่าที่ถูกส่งขึ้นไปยังโนดเช็คพร้อมกับค่า s_m จะเป็นค่าที่เกิดจากการบวกกันของขนาดของสัญญาณที่ได้รับจากแต่ละโนดบิตภายใต้โนดเช็คที่เชื่อมต่อกับโนดบิตที่ 7 ดังรูปที่ 3.13 ฉะนั้น กระบวนการประมวลผลโนดบิตสำหรับคำนวณค่า E_n ของแต่ละโนดบิตสำหรับอัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) สามารถคำนวณได้จาก

$$E_n = \frac{1}{|y_n|} \sum_{m \in M(n)} (2s_m - 1) \left(\sum_{n \in N(m)} |y_n| \right) \quad (3.43)$$

โดย $N(m)$ คือเซตของตำแหน่งเลข 1 ในแถวที่ m ของเมทริกซ์พาริตีเช็ค เช่นเดียวกับ $M(n)$ คือเซตของตำแหน่งเลข 1 ในแถวที่ n ของเมทริกซ์พาริตีเช็ค จะสังเกตว่า การคำนวณค่าความไม่น่าเชื่อถือ E_n ในสมการที่ 3.43 จะไม่ขึ้นกับค่า α อย่างไรก็ตาม กระบวนการคำนวณค่า E_n ของอัลกอริทึมนี้จะคล้ายคลึงกับอัลกอริทึม M-WBF และ IM-WBF ที่มีการนำค่า $|y_n|$ ของตัวมันเองมาทำการนอร์มัลไลซ์ แต่อัลกอริทึมนี้จะทำการนอร์มัลไลซ์ด้วยการหาร เพื่อเพิ่มความแม่นยำในการคัดเลือกบิตมาฟลิป โดยที่หาก $|y_n|$ มีค่ามาก (บิตน่าเชื่อถือสูง) จะส่งผลให้ค่า E_n มีค่าต่ำลงมาก ในตรงกันข้ามหาก $|y_n|$ มีค่าน้อย (บิตน่าเชื่อถือต่ำ) จะส่งผลให้ค่า E_n ลดลงไม่มาก นอกจากนี้ การคำนวณค่า E_n ของอัลกอริทึมนี้ เกิดจากการรวมกันของขนาดของแต่ละโนดบิตภายใต้โนดเช็คที่เชื่อมต่อกับบิตที่ต้องการหาค่า ดังนั้น รหัสแอลดีพีซีที่ทำการออกแบบนั้น ต้องเป็นรหัสแอลดีพีซีแบบคงที่เท่านั้น กล่าวคือ จำนวนโนดบิตที่เชื่อมต่อกับโนดเช็คต้องมีค่าเท่ากันเสมอหรือมีค่า d_c เท่ากันนั่นเอง ส่วนขั้นตอนการถอดรหัสของอัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) จะเหมือนกันกับอัลกอริทึมบิตฟลิปปีงชนิดถ่วงน้ำหนักพื้นฐาน (WBF) เพียงแต่ไม่มีการคำนวณค่า $|y|_{\min-m}$ และแก้ไขสมการในการคำนวณค่า E_n ตามสมการ 3.43 ตามลำดับ

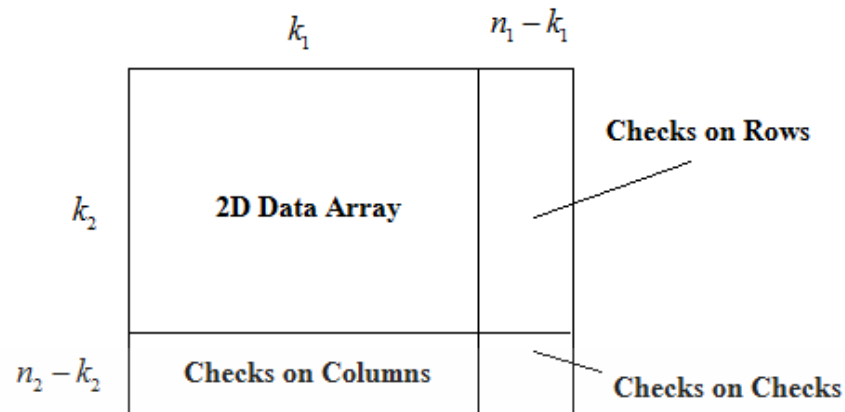
บทที่ 4

รหัสแอลดีพีซีความซับซ้อนต่ำ

เป็นที่ทราบกันแล้วว่า สมรรถนะของรหัสพาริตีเช็คความหนาแน่นต่ำ (Low-density parity-check codes) หรือรหัสแอลดีพีซี ที่ถูกกล่าวถึงในบทที่ 3 นั้น จะมีสมรรถนะหรือความสามารถในการแก้ไขความผิดพลาดที่ดีขึ้น เมื่อขีดข้อมูลหรือคำรหัสมีขนาดใหญ่มากขึ้น อย่างไรก็ตาม การที่ขีดข้อมูลหรือคำรหัสมีขนาดใหญ่มากขึ้นนั้น เป็นอุปสรรคต่อการเข้ารหัส และถอดรหัสในทางปฏิบัติ เนื่องจาก เป็นการเพิ่มความซับซ้อนของวงจรเข้ารหัส และถอดรหัสตามลำดับ ดังแสดงในงานวิจัย [14-15] ซึ่งในหัวข้อที่ 4 ของวิทยานิพนธ์เล่มนี้ จะกล่าวถึงเทคนิค และวิธีการต่างๆ ที่จะถูกนำมาใช้เพื่อลดความซับซ้อนของวงจรเข้ารหัส และถอดรหัสแอลดีพีซี รวมถึงวิธีการถอดรหัสบิดพลิปปิงชนิดถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping : P-WBF), ชนิดแถวสลับหลัก (Row-column weighted bit-flipping : RC-WBF) และชนิดดัดแปลงแถวสลับหลัก (Modified row-column weighted bit-flipping : MRC-WBF) ซึ่งเป็นวิธีการที่ผู้วิจัยนำเสนอในวิทยานิพนธ์เล่มนี้

4.1 รหัสแอลดีพีซีแบบโปรดักต์ (Product LDPC Codes: PCB) [9]

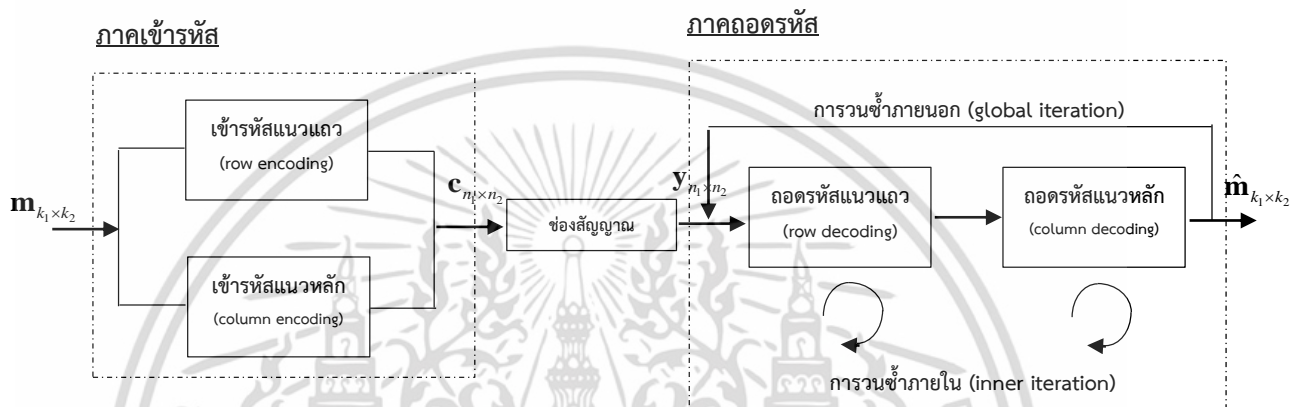
อัลกอริทึมรหัสแอลดีพีซีแบบโปรดักต์ [9] เป็นอัลกอริทึมที่ถูกนำเสนอในปี 2004 โดยงานวิจัยของ Z. Qi และ C. Sum อัลกอริทึมดังกล่าว ถูกนำเสนอเพื่อลดความซับซ้อนในการเข้ารหัสแอลดีพีซีสำหรับขีดข้อมูลที่มีความยาวมาก อัลกอริทึมนี้จะทำการแบ่งขีดข้อมูลก่อนการเข้ารหัส ออกเป็นบล็อกย่อยหลายๆ บล็อก ซึ่งแต่ละบล็อกมีความยาวเท่ากัน โดยจะสามารถพิจารณาขีดข้อมูลความยาวมาก (ที่ถูกแบ่งออกเป็นบล็อกย่อยๆ) ให้อยู่ในรูปแบบของข้อมูลสองมิติ (Two-dimensional : 2D) จากนั้น ขีดข้อมูลสองมิติจะถูกเข้ารหัสแอลดีพีซีในแนวแถว และหลักแยกจากกัน คำรหัสสองมิติที่ได้ จะถูกเรียกว่า “รหัสแอลดีพีซีแบบโปรดักต์” ในส่วนภาครับนั้น คำรหัสที่ได้รับจากช่องสัญญาณสื่อสาร จะถูกถอดรหัสแอลดีพีซีทีละบล็อกในแนวแถว และหลักสลับกันไปมา ตามจำนวนรอบการวนซ้ำที่กำหนด โครงสร้างรหัสแอลดีพีซีแบบโปรดักต์ สามารถพิจารณาได้จากรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของรหัสแอลดีพีซีแบบโปรดักส์

กำหนดให้ บล็อกข้อมูลหนึ่งมิตีก่อนเข้ารหัสมีความยาวมาก ซึ่งมีจำนวนบิตทั้งหมดเท่ากับ k บิต ซึ่งหากนำบล็อกข้อมูลดังกล่าว มาทำการแปลงให้อยู่ในรูปแบบข้อมูลสองมิติหรือทำให้อยู่ในโครงสร้างของรหัสแอลดีพีซีแบบโปรดักส์ ดังรูปที่ 4.1 ชุดบิตข้อมูลหนึ่งมิตมีความยาวมากจำนวน k บิตนั้น จะถูกแบ่งออกเป็นบล็อกย่อยๆ k_2 บล็อก โดยในแต่ละบล็อกนั้น จะมีความยาวเท่ากับ k_1 บิต เท่าๆกัน ตามลำดับ ซึ่งเมื่อจัดเรียงบล็อกข้อมูลใหม่ให้อยู่ในรูปแบบของข้อมูลสองมิติ (2D) ข้อมูลใหม่จะมีขนาดเท่ากับ $k_1 \times k_2$ บิต หลังจากนั้น บิตข้อมูลแต่ละบล็อกในแนวแถว และหลักจะถูกเข้ารหัสแอลดีพีซีด้วยเมทริกซ์พาริตีเช็ค H_1 และ H_2 เกิดเป็นคำรหัสที่มีความยาวเท่ากับ n_1 และ n_2 ตามลำดับ ดังนั้น สำหรับรหัสแอลดีพีซีแบบโปรดักส์ บิตข้อมูลสองมิติจะถูกปกป้องและแก้ไขความผิดพลาดด้วยพิตพาริตีในแนวแถว (Checks on Rows), พิตพาริตีในแนวหลัก (Checks on Columns) และพิตพาริตีของพิตพาริตี (Checks on Checks) สำหรับรหัสแอลดีพีซีแบบโปรดักส์นั้น สามารถคำนวณหาขนาดของบิตข้อมูล, คำรหัส และอัตรารหัสโดยรวมของรหัสแอลดีพีซีแบบโปรดักส์ จาก $k^{2D} = k_1 \times k_2$, $n^{2D} = n_1 \times n_2$ และ $R^{2D} = k/n = (k_1 \times k_2)/(n_1 \times n_2)$ ตามลำดับ

กำหนดให้ \mathbf{c}_i โดย $i = \{1, 2, 3, \dots, n_2\}$ คือเวกเตอร์ค่ารหัสแต่ละบล็อกในแนวแถว ซึ่งแต่ละบล็อกในแนวแถวจะมีความสัมพันธ์กับเมทริกซ์พาริตีเช็ค $\mathbf{c}_i \mathbf{H}_1^T = \mathbf{0}$ เช่นเดียวกับกับเวกเตอร์ค่ารหัส \mathbf{c}_j โดย $j = \{1, 2, 3, \dots, n_1\}$ ซึ่งเป็นตัวแทนของแต่ละบล็อกในแนวหลัก ก็จะมีสัมพันธ์กับเมทริกซ์พาริตีเช็ค $\mathbf{c}_j \mathbf{H}_2^T = \mathbf{0}$ ด้วยเช่นกัน สำหรับขั้นตอนการถอดรหัสของรหัสแอลดีพีซีแบบโปรดักส์ จะอาศัยสองคุณสมบัติดังกล่าวข้างต้น ในการถอดรหัสสลับกันไปมาระหว่างแนวแถว และหลักตามจำนวนรอบการวนซ้ำที่กำหนด ซึ่งขั้นตอนการถอดรหัสแอลดีพีซีแบบโปรดักส์สามารถพิจารณาได้จากแบบจำลองในรูปที่ 4.2



รูปที่ 4.2 แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึมแอลดีพีซีแบบโปรดักส์

พิจารณาแบบจำลองการเข้ารหัส และถอดรหัสของอัลกอริทึมแอลดีพีซีแบบโปรดักส์ในรูปที่ 4.2 บิตข้อมูลที่ต้องการส่งจะถูกจัดให้อยู่ในรูปแบบสองมิติ $\mathbf{m}_{k_1 \times k_2}$ จากนั้น บิตข้อมูลในแต่ละแถว และหลักจะถูกเข้ารหัสแอลดีพีซีด้วยเมทริกซ์พาริตีเช็ค \mathbf{H}_1 และ \mathbf{H}_2 ตามลำดับ เกิดเป็นค่ารหัสสองมิติ $\mathbf{c}_{n_1 \times n_2}$ ก่อนที่ค่ารหัสจะถูกส่ง ค่ารหัสจะต้องถูกมอดูเลชันให้มีสภาพที่เหมาะสมกับตัวกลางหรือช่องสัญญาณสื่อสารใดๆ จากนั้นจึงจะถูกส่งผ่านช่องสัญญาณสื่อสาร ที่ประกอบไปด้วยสัญญาณรบกวนชนิดต่างๆ ในส่วนภาครับนั้น สัญญาณที่ได้รับ $\mathbf{y}_{n_1 \times n_2}$ จะถูกนำไปถอดรหัสแอลดีพีซี ในแต่ละแถวด้วยอัลกอริทึมแพร่กระจายความเชื่อมั่นหรือซมโปรดักส์ (SPA) [27] จากนั้นค่าซอฟต์แวร์ที่ได้ จะถูกนำไปถอดรหัสซ้ำอีกครั้งในแนวหลักด้วยอัลกอริทึมเดียวกัน วิธีการถอดรหัสแอลดีพีซี ด้วยอัลกอริทึมซมโปรดักส์ (SPA) สามารถพิจารณาได้อย่างละเอียดในหัวข้อที่ 3.3.1 อย่างไรก็ตาม เป็นที่ทราบกันดีว่ารหัสแอลดีพีซีจะมีสมรรถนะที่ดีได้ เมื่อมีการประมวลผลแบบวนซ้ำหรือมีถอดรหัสแบบวนซ้ำ ซึ่งการวนซ้ำของรหัสแอลดีพีซีแบบโปรดักส์ จะมีความพิเศษที่แตกต่างจากรหัสแอลดีพีซีแบบหนึ่งมิติทั่วไปคือ จะประกอบด้วยการวนซ้ำ 2 ชนิด อันได้แก่

- 1) การวนซ้ำภายนอก (global iteration) คือ จำนวนครั้งในการป้อนกลับค่าซอฟต์แวร์
ภายหลังจากถอดรหัสในแนวหลัก กลับมายังแนวแถวอีกครั้ง
- 2) การวนซ้ำภายใน (inner iteration) คือ การวนซ้ำของรหัสแอลดีพีซีทั่วไปในแต่ละบล็อก
ของแนวแถวและหลัก

4.1.1 ความซับซ้อนของรหัสแอลดีพีซีแบบโปรดักต์

ในงานวิจัยที่ [9] ได้เปรียบเทียบความซับซ้อนในการเข้า และถอดรหัสของอัลกอริทึมแอลดีพีซีแบบโปรดักต์ เทียบกับอัลกอริทึมถอดรหัสแอลดีพีซีหนึ่งมิติในความยาวที่เท่ากัน ซึ่งโดยทั่วไปแล้ว ความซับซ้อนของรหัสแอลดีพีซี จะถูกประเมินอย่างง่ายจากจำนวนของวงจรวกหรือวงจรรูณตามลำดับ อย่างไรก็ตาม ในหัวข้อนี้ผู้วิจัยจะกล่าวถึง การคำนวณความซับซ้อนด้วยวิธีการนับจำนวนวงจรวก และวงจรรูณที่ถูกใช้ในการเข้ารหัส และถอดรหัสของทั้ง 2 อัลกอริทึมตามลำดับ

4.1.1.1 ความซับซ้อนในการเข้ารหัสแอลดีพีซีแบบโปรดักต์

สำหรับการคำนวณความซับซ้อนในการเข้ารหัสแอลดีพีซีหนึ่งมิติทุกๆ ไปนั้น เป็นที่ทราบกันดีว่า บิตพาริตี b_i ของรหัสแอลดีพีซี จะถูกคำนวณจากการคูณกันระหว่างองค์ประกอบในแต่ละแถวของเมทริกซ์พาริตีเช็ค \mathbf{H} กับเวกเตอร์ของบิตข้อมูล $\mathbf{u} = [u_0, u_1, \dots, u_{k-1}]$ จากนั้น จะถูกบวกกับแบบมอดูโลสอง ดังสมการ

$$b_i = u_0 h_{i1} + u_1 h_{i2} + \dots + u_{k-1} h_{ik-1} \quad (4.1)$$

โดย u_i และ h_i คือ บิตข้อมูลที่จะทำการเข้ารหัส และองค์ประกอบในแต่ละแถวของเมทริกซ์พาริตีเช็ค \mathbf{H} ตามลำดับ ซึ่งทั้งสององค์ประกอบมีจำนวนเท่ากับ k ดังนั้น จำนวนวงจรรูณ และวงจรวกที่ใช้สำหรับการเข้ารหัสเพื่อสร้างบิตพาริตีความยาว $n-k$ จะมีจำนวนทั้งหมดเท่ากับ $(n-k)k$ อย่างไรก็ตาม สำหรับรหัสแอลดีพีซีแบบโปรดักต์ ที่มีการเข้ารหัสในแนวแถว และหลักแยกจากกัน วงจรรูณหรือวงจรวกในแต่ละแถว และหลัก จะมีจำนวนเท่ากับ $(n_1 - k_1)k_1$ และ $(n_2 - k_2)k_2$ ตามลำดับ เป็นที่ทราบดีว่าบิตพาริตีในแนวแถว (Checks on Rows) จะเกิดขึ้นจากการเข้ารหัสในแนวแถวแต่ละแถว และบิตพาริตีในแนวหลัก (Checks on Columns) จะเกิดขึ้นจากการเข้ารหัสในแนวแถวแต่ละหลัก แต่ในส่วนของบิตพาริตีของบิตพาริตี (Checks on Checks) นั้นสามารถที่จะเข้ารหัสจากบิตพาริตีในแนวแถวหรือในแนวหลักได้อย่างหนึ่ง ดังนั้น จำนวนวงจรวกหรือวงจรรูณสำหรับอัลกอริทึมแอลดีพีซีแบบโปรดักต์ สามารถคำนวณได้จาก [9] ดังนี้

$$complexity[2D] = ((n_1 - k_1)k_1)k_2 + ((n_2 - k_2)k_2)n_1 \quad (4.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ

$$\text{complexity}[2D] = ((n_1 - k_1)k_1)n_2 + ((n_2 - k_2)k_2)k_1 \quad (4.4)$$

ดังนั้นอัตราส่วนความซับซ้อนในการเข้ารหัส ระหว่างรหัสแอสติพีซีแบบโปรตักส์กับรหัสแอสติพีซีหนึ่งมิติโดยทั่วไปที่ความยาวเท่ากัน สามารถคำนวณได้จาก

$$\frac{\text{complexity}[2D]}{\text{complexity}[1D]} = \frac{((n_1 - k_1)k_1)k_2 + ((n_2 - k_2)k_2)n_1}{(n_1n_2 - k_1k_2)k_1k_2} \quad (4.5)$$

หรือ

$$\frac{\text{complexity}[2D]}{\text{complexity}[1D]} = \frac{((n_1 - k_1)k_1)n_2 + ((n_2 - k_2)k_2)k_1}{(n_1n_2 - k_1k_2)k_1k_2} \quad (4.6)$$

จากสมการที่ 4.5 และ 4.6 จะเห็นได้ว่า ไม่ว่าจะรหัสแอสติพีซีจะมีความยาวบิตข้อมูลเป็นเท่าใดก็ตาม รหัสแอสติพีซีแบบโปรตักส์ จะสามารถลดความซับซ้อนในการเข้ารหัสหรือลดการใช้วงจรวจรหรือวงจรรวมเมื่อเปรียบเทียบกับรหัสแอสติพีซีหนึ่งมิติโดยทั่วไป ที่ความยาวเข้ารหัสเท่ากันได้เสมอ อย่างไรก็ตาม อัตรารหัสโดยรวมของทั้งสองวิธีการ อาจมีค่าที่ไม่เท่ากัน

4.1.1.2 ความซับซ้อนในการถอดรหัสแอสติพีซีแบบโปรตักส์

โดยปกติแล้วความซับซ้อนในการถอดรหัสรหัสแอสติพีซี จะมีค่าขึ้นอยู่กับค่าเฉลี่ยของจำนวนเลขหนึ่งในแต่ละหลักของเมทริกซ์พาริตีเช็ค \mathbf{H} [14-15] เมื่อจำนวนรอบการวนซ้ำ และความยาวของรหัสถูกกำหนดให้มีค่าคงที่ อย่างไรก็ตาม เนื่องจากอัลกอริทึมแอสติพีซีแบบโปรตักส์ใช้เมทริกซ์พาริตีเช็คถึง 2 เมทริกซ์ สำหรับการเข้ารหัสในแนวแถวและหลัก ฉะนั้น อัตราส่วนความซับซ้อนของวงจรถอดรหัส ระหว่างรหัสแอสติพีซีแบบโปรตักส์กับรหัสแอสติพีซีหนึ่งมิติทั่วไป สามารถที่จะประมาณอย่างคร่าวๆ ได้จาก [9] ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\text{complexity}[2D]}{\text{complexity}[1D]} = \frac{\frac{N_R + N_C}{n_1} + \frac{N_C}{n_2}}{\frac{N_L}{n}} \quad (4.7)$$

โดยที่ N_R และ N_C คือ จำนวนเลขหนึ่งทั้งหมดในเมทริกซ์พาริตีเช็ค ที่ใช้ในการเข้ารหัสแอสติฟิซีแบบโปรตักส์ในแต่ละแถว และหลัก ตามลำดับ ส่วน N_L คือ จำนวนเลขหนึ่งทั้งหมดในเมทริกซ์พาริตีเช็ค ที่ใช้เข้ารหัสสำหรับรหัสแอสติฟิซีหนึ่งมิติ อย่างไรก็ตาม พิจารณาจากรูปที่ 4.1 เมื่อกำหนดตัวแปร n หรือความยาวของรหัสแอสติฟิซีหนึ่งมิติทุกๆ ไป ให้มีความยาวเท่ากับรหัสแอสติฟิซีแบบโปรตักส์หรือมีค่าเท่ากับ $n_1 \times n_2$ ดังนั้นสมการที่ 4.7 จะถูกเขียนใหม่เป็น

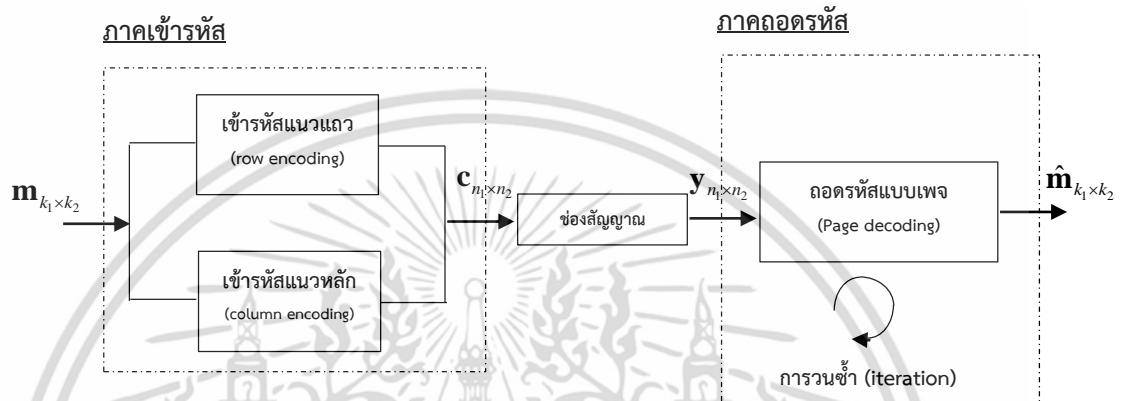
$$\frac{\text{complexity}[2D]}{\text{complexity}[1D]} = \frac{n_2 N_R + n_1 N_C}{N_L} \quad (4.8)$$

พิจารณาจากสมการที่ 4.8 จะสังเกตว่า อัตราส่วนของความซับซ้อนในการถอดรหัสของอัลกอริทึมทั้ง 2 จะมีค่าขึ้นอยู่กับารออกแบบของแต่ละบุคคล ว่าต้องการความซับซ้อนของอัลกอริทึมทั้ง 2 มากน้อยกว่ากันเพียงใด โดยสามารถออกแบบด้วยวิธีการเพิ่มหรือลดจำนวนเลขหนึ่งในแต่ละหลักของเมทริกซ์พาริตีเช็ค อย่างไรก็ตาม การเพิ่มหรือลดจำนวนเลขหนึ่งต้องเป็นไปตามทฤษฎีการออกแบบรหัสแอสติฟิซี ที่ได้กล่าวถึงในหัวข้อ 3.2.3 ซึ่งต้องไม่มีความยาวไซเคิลเท่ากับ 4 และในแต่ละหลักของเมทริกซ์พาริตีเช็คต้องมีเลขหนึ่งอย่างน้อย 1 ตัวตามลำดับ

4.2 การถอดรหัสแบบบิตพลิกปิ้งชนิดถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping decoding: P-WBF)

จากหัวข้อที่ 4.1 ได้กล่าวถึง รหัสแอสติฟิซีแบบโปรตักส์ที่มีการเข้ารหัส และถอดรหัส ในแนวแถวและหลักสลับกันไปมา ตามจำนวนรอบการวนซ้ำทั้ง 2 แบบที่กำหนดไว้ ซึ่งผลที่ได้ จะทำให้ความซับซ้อนในการเข้ารหัสแอสติฟิซีลดลงอย่างมีนัยยะสำคัญ เมื่อเทียบกับรหัสแอสติฟิซีหนึ่งมิติทุกๆ ไป ที่ความยาวคำรหัสเท่ากัน [9] ส่วนความซับซ้อนในการถอดรหัสแอสติฟิซีของทั้งสองอัลกอริทึม จะขึ้นอยู่กับารออกแบบเมทริกซ์พาริตีเช็ค ที่ใช้สำหรับเข้ารหัสข้อมูลตามสมการที่ 4.8 โดยความซับซ้อนในการถอดรหัสของทั้ง 2 อัลกอริทึม จะขึ้นอยู่กับจำนวนเลขหนึ่งโดยเฉลี่ยบนเมทริกซ์พาริตีเช็ค อย่างไรก็ตาม อาจจะเป็นการยาก ที่จะปรับลดความซับซ้อนในการถอดรหัสของรหัสแอสติฟิซีแบบโปรตักส์ ให้มีค่าน้อยกว่ารหัสแอสติฟิซีหนึ่งมิติทุกๆ ไป เนื่องจากการเพิ่มหรือลดจำนวนเลขหนึ่งบนเมทริกซ์พาริตีเช็ค อาจเป็นการทำลายโครงสร้างที่เหมาะสมของรหัสแอสติฟิซี เช่น ขนาดของเกร็ดขนาดวงรอบ หรือค่าระยะห่างต่ำสุด (d_{\min}) เป็นต้น ดังที่ได้กล่าวถึงในหัวข้อที่ 3.2.3

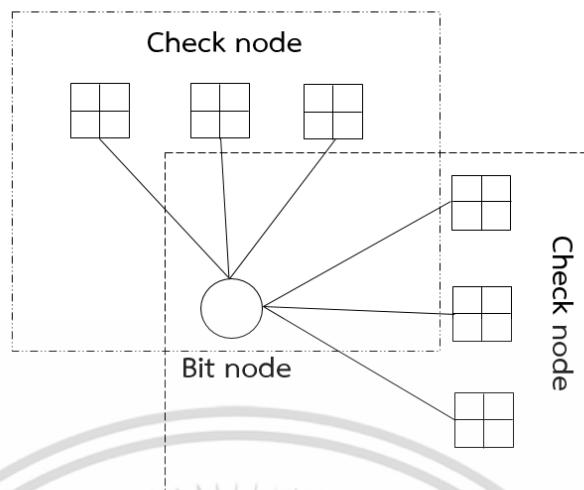
ดังนั้นในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจึงได้นำเสนออัลกอริทึมการถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (P-WBF) ที่เป็นการผสมผสานอัลกอริทึมบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) เข้ากับการเข้ารหัสแอลดีพีซีแบบโปรดักส์ โดยที่สามารถลดความซับซ้อนในการเข้ารหัส พร้อมทั้งยังสามารถลดความซับซ้อนในการถอดรหัสลงจากรหัสแอลดีพีซีหนึ่งมิติต่างๆ ไป โดยไม่จำเป็นต้องทำการปรับลดจำนวนเลขหนึ่งในเมทริกซ์พาริตีเช็ค พิจารณาแบบจำลองการเข้ารหัส และถอดรหัสของอัลกอริทึม P-WBF ในรูปที่ 4.3



รูปที่ 4.3 แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม P-WBF

พิจารณารูปที่ 4.3 ในส่วนภาคเข้ารหัสของอัลกอริทึม P-WBF ที่ซึ่งจะเข้ารหัสในแต่ละแถว และแต่ละหลักแยกจากกัน เหมือนกันกับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ [9] ฉะนั้น โครงสร้าง และพารามิเตอร์ของข้อมูลและคำรหัส จะเหมือนกันกับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ในรูปที่ 4.1 แต่ในส่วนภาคถอดรหัส จะมีความแตกต่างจากอัลกอริทึมแอลดีพีซีแบบโปรดักส์อย่างเห็นได้ชัด 2 ประการคือ ประการแรก อัลกอริทึม P-WBF จะใช้การถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) หรือการถอดรหัสแบบหยาบ แทนที่การถอดรหัสด้วยอัลกอริทึมซัมโปรดักส์ (SPA) [27] หรือการถอดรหัสแบบซอฟต์แวร์ ประการที่สอง อัลกอริทึม P-WBF จะถอดรหัสในแถว และหลักไปพร้อมๆ กัน กล่าวคือ จะทำการคำนวณค่า E_n ในทุกๆ บิต เช่นเดียวกับกับอัลกอริทึม RR-WBF ที่ถูกกล่าวถึงในหัวข้อ 3.3.2 จากนั้น จึงจะทำการพลิกบิตในตำแหน่งที่มีค่า E_n สูงที่สุดในแต่ละรอบการวนซ้ำ ดังนั้น ขั้นตอนการถอดรหัสสำหรับอัลกอริทึม P-WBF จะคล้ายคลึงกันกับอัลกอริทึม RR-WBF เพียงแต่สมการในการคำนวณค่าความไม่น่าเชื่อถือ E_n จะถูกเปลี่ยนแปลงใหม่ เนื่องจากแต่ละบิตข้อมูลของอัลกอริทึม P-WBF จะเชื่อมต่อกันแบบ 2 มิติ กล่าวคือ แต่ละโนดบิต จะถูกเชื่อมต่อกับกราฟแทนเนอร์ทั้งในแนวแถว และแนวหลัก ที่เกิดจากการเข้ารหัสแยกจากกันด้วยเมทริกซ์พาริตีเช็ค \mathbf{H}_1 และ \mathbf{H}_2 ตามลำดับ

กราฟแทนเนอร์ในแนวแถว



กราฟแทนเนอร์ในแนวหลัก

รูปที่ 4.4 การเชื่อมต่อของกราฟแทนเนอร์กับบิตข้อมูลแบบ 2 มิติ สำหรับ P-WBF

พิจารณารูปที่ 4.4 จะเห็นได้ว่าบิตข้อมูลหรือโนดบิตในแต่ละบิต จะถูกเชื่อมต่อด้วยกราฟแทนเนอร์ทั้งในแนวแถวและหลัก ซึ่งจะมีโครงสร้างเป็นไปตามเมทริกซ์พาริตีที่เช็ค ที่ใช้เข้ารหัสในแนวแถวและหลัก ตามลำดับ เพราะฉะนั้น ขั้นตอนการถอดรหัสข้อมูลที่ได้รับจากช่องสัญญาณแบบสองมิติ $\mathbf{y}_{n_1 \times n_2}$ ด้วยอัลกอริทึม P-WBF จะถูกแก้ไขใหม่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ ดัชนี i และ j คือ ตำแหน่งเวกเตอร์คาร์รหัสแถวที่ i และหลักที่ j โดยเซตของ $M_{row}(j)$ และ $M_{col}(i)$ ถูกกำหนดให้เป็น ตำแหน่งเลข 1 ในหลักที่ j ของเมทริกซ์พาริตีเซต \mathbf{H}_1 ที่ใช้เข้ารหัสในแนวแถว และตำแหน่งเลข 1 ในหลักที่ i ของเมทริกซ์พาริตีเซต \mathbf{H}_2 ที่ใช้เข้ารหัสในแนวหลัก ในส่วนของเซต $N_{row}(m)$ และ $N_{col}(m)$ คือตำแหน่งเลข 1 ในแถวที่สัมพันธ์กับสมาชิกในเซต $M_{row}(j)$ และ $M_{col}(i)$ ของเมทริกซ์พาริตีเซต ที่ใช้เข้ารหัสในแนวแถว \mathbf{H}_1 และหลัก \mathbf{H}_2 ตามลำดับ

for k จะมีค่าตั้งแต่ 1 ถึงการวนซ้ำในจำนวนรอบที่ต้องการ

ทำการตัดสินใจแบบหยาบของสัญญาณ y โดย $z_{ij} = 1$ เมื่อ $y_{ij} \geq 0$ นอกนั้น $z_{ij} = 0$

for i มีค่าตั้งแต่ 1 ถึง n_1

for j มีค่าตั้งแต่ 1 ถึง n_2

if k มีค่าเท่ากับ 1

คำนวณค่าซินโดรม (s_m) ของแต่ละเซตโนดบนกราฟแพนเนอร์ในแนวแถว และหลักที่เชื่อมต่อกับบิตในตำแหน่งแถวที่ i และหลักที่ j โดยค่า s_m ของแต่ละโนดเซต จะมีค่าเท่ากับการบวกกันแบบมอดุโลสองของ z_{ij} ทุกบิตที่เชื่อมต่อกับโนดเซตนั้น จากนั้นคำนวณค่าความไม่น่าเชื่อถือ ($E_{i,j}$) ของบิต ในตำแหน่งแถวที่ i และหลักที่ j ด้วยสมการ

$$E_{i,j} = \frac{1}{|y_{i,j}|} \left[\sum_{m \in M_{row}(j)} (2s_m - 1) \left(\sum_{j \in N_{row}(m)} |y_{i,j}| \right) + \sum_{m \in M_{col}(i)} (2s_m - 1) \left(\sum_{i \in N_{col}(m)} |y_{i,j}| \right) \right]$$

else

ทำการคำนวณค่าซินโดรม (s_m) และค่าความไม่น่าเชื่อถือ ($E_{i,j}$) เฉพาะบิตที่ถูกพลิกไปในรอบที่ $k-1$ และบิตที่เชื่อมต่อกับบิตที่ถูกพลิกเท่านั้น

end

end

end

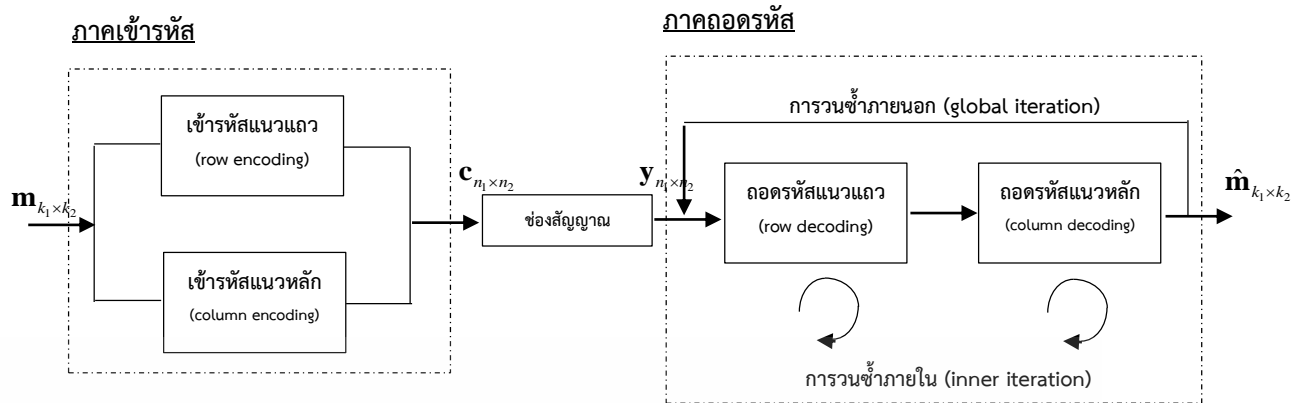
พลิกบิต z_{ij} ในตำแหน่งที่มีค่า $E_{i,j}$ สูงที่สุด หากทุกบล็อกในแนวแถว และหลักถูกต้องทั้งหมด (เช็คความถูกต้อง ด้วยความสัมพันธ์ ($\mathbf{cH}^T = 0$) หรือถ้า k มีค่าเท่ากับจำนวนรอบสูงสุด **อัลกอริทึมนี้จะหยุดการทำงาน**

end

4.3 การถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบแถวสลับหลัก (Row-column weighted bit-flipping decoding: RC-WBF)

พิจารณาขั้นตอนการถอดรหัสของอัลกอริทึม P-WBF จะพบว่าความซับซ้อนในการถอดรหัสของอัลกอริทึม P-WBF จะมีค่าน้อยกว่าอัลกอริทึมแอลดีพีซีแบบโปรดักส์ เนื่องจากอาศัยการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) หรือการถอดรหัสแบบหยาบสำหรับจุดด้อยของอัลกอริทึม P-WBF คือ ภายหลังจากการคำนวณค่าความไม่น่าเชื่อถือ ($E_{i,j}$) จนครบทุกบิตแล้ว อัลกอริทึมนี้จะทำการจัดเรียงค่า $E_{i,j}$ ของทุกๆบิตจากน้อยไปมาก เพื่อทำการเลือกพลิกบิตในตำแหน่งที่มีค่า $E_{i,j}$ สูงที่สุด ในแต่ละรอบการวนซ้ำ อย่างไรก็ตาม หากข้อมูลสองมิติที่ทำการถอดรหัสนั้นมีขนาดที่ใหญ่มากขึ้น จะมีผลทำให้ความซับซ้อนในการถอดรหัสของอัลกอริทึม P-WBF มีค่าเพิ่มขึ้น ด้วยแนวโน้มแบบฟังก์ชันกำลังสอง ของค่า n

ดังนั้นในหัวข้อที่ 4.3 ผู้วิจัยจึงนำเสนออัลกอริทึมใหม่ ที่สามารถปรับปรุงจุดด้อยของอัลกอริทึม P-WBF โดยจะลดจำนวนบิตที่จะถูกคำนวณค่า $E_{i,j}$ ในแต่ละรอบการวนซ้ำ กล่าวคือ อัลกอริทึมใหม่นี้ สามารถลดความซับซ้อนในการถอดรหัสของอัลกอริทึม P-WBF ได้ อย่างไรก็ตาม ในภาคเข้า และถอดรหัสของอัลกอริทึมใหม่ จะมีความคล้ายคลึงกันกับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ ที่ได้นำเสนอในหัวข้อที่ 4.1 เพียงแต่ภายในบล็อกถอดรหัสในแนวแถว และหลัก จะอาศัยการถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) หรือการถอดรหัสแบบหยาบ มาแทนที่อัลกอริทึมซมโปรดักส์ (SPA) [27] หรือการถอดรหัสแบบซอฟต์แวร์ อันเนื่องมาจาก อัลกอริทึมใหม่มีการเข้า และถอดรหัสในแนวแถวและหลักสลับกันไปมาคล้ายกับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ พร้อมทั้งมีการนำเอาอัลกอริทึมการถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) มาประยุกต์ใช้กับวงจรถอดรหัส ดังนั้น ผู้วิจัยจึงให้ชื่ออัลกอริทึมใหม่นี้ว่า “อัลกอริทึมการถอดรหัสแบบบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบแถวสลับหลัก (RC-WBF)” แบบจำลองของอัลกอริทึม RC-WBF สามารถพิจารณาได้จากรูปที่ 4.5



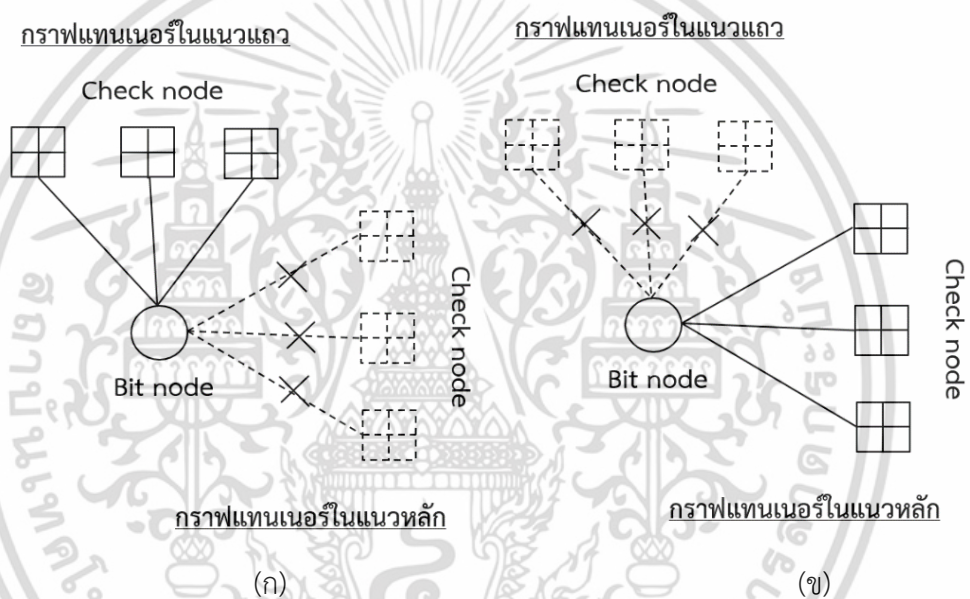
รูปที่ 4.5 แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม RC-WBF

พิจารณาแบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม RC-WBF ในรูปที่ 4.5 หลังจากบิตข้อมูลแบบสองมิติ $\mathbf{m}_{k_1 \times k_2}$ ถูกนำมาเข้ารหัสในแต่ละแถว และหลักแยกจากกัน เกิดเป็นค้ำรหัสสองมิติ $\mathbf{c}_{n_1 \times n_2}$ หลังจากนั้น ค้ำรหัสจะถูกนำมาถอดดูเลต เพื่อให้มีสภาพที่เหมาะสมกับช่องสัญญาณสื่อสารหรือตัวกลาง โดยที่ภาครับนั้น สัญญาณที่ได้รับ $\mathbf{y}_{n_1 \times n_2}$ จะถูกถอดรหัสในแต่ละแถว ด้วยอัลกอริทึมการถอดรหัสแบบบิตพลิกบิตตวงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) หรือการถอดรหัสแบบหยาบ จากนั้น จะเป็นหน้าที่ของการถอดรหัสในแนวหลัก ที่จะถอดรหัสในแต่ละหลักซ้ำอีกครั้งหนึ่ง ซึ่งจะถอดรหัสสลับไปสลับมาเรื่อยๆ ระหว่างแนวแถว และหลักจนครบจำนวนรอบการวนซ้ำที่กำหนดหรือจนกว่าบิตข้อมูลจะมีความถูกต้องทั้งหมด สังเกตว่า การถอดรหัสด้วยอัลกอริทึม RC-WBF นั้น แอมพลิจูดค่าซอฟต์แวร์ของสัญญาณที่ได้รับ จะไม่มีการเปลี่ยนแปลงในแต่ละรอบการวนซ้ำ โดยจะมีการเปลี่ยนแปลงเฉพาะเครื่องหมายเท่านั้น ซึ่งจะมีความแตกต่างจากอัลกอริทึมการถอดรหัสแอลดีพีซีแบบโปรดักส์ที่อาศัยอัลกอริทึมการถอดรหัสแบบซิมโปรดักส์ (SPA) [27] หรือการถอดรหัสแบบซอฟต์แวร์อย่างเห็นได้ชัด ที่ในแต่ละรอบของการวนซ้ำ แอมพลิจูดค่าซอฟต์แวร์ของสัญญาณที่ได้รับ จะมีค่าที่เพิ่มขึ้นหรือลดลงทบกันไปเรื่อยๆ ตามความน่าเชื่อถือของแต่ละบิต อย่างไรก็ตาม การวนซ้ำของอัลกอริทึมการถอดรหัส RC-WBF จะมีนิยามคล้ายกันกับอัลกอริทึมแอลดีพีซีแบบโปรดักส์ ในหัวข้อ 4.1 โดยจะประกอบไปด้วย

- 1.) การวนซ้ำภายนอก (global iteration) คือจำนวนครั้งในการบ้อนกลับค่าฮาร์ด ภายหลังการถอดรหัสในแนวหลักกลับมายังแนวแถวอีกครั้ง
- 2.) การวนซ้ำภายใน (inner iteration) คือการวนซ้ำของรหัสแอลดีพีซีแบบบิตพลิกบิตตวงน้ำหนัก (RR-WBF) ในแต่ละบล็อกของแนวแถวและหลัก หรือคือจำนวนบิตที่ถูกพลิกในแต่ละบล็อกนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถอดรหัสบิตข้อมูลสองมิติด้วยอัลกอริทึม RC-WBF จะถอดรหัสในแนวแถว และหลักสลับไปมาอย่างแยกจากกัน ฉะนั้น ถึงแม้บิตข้อมูลในแต่ละบิตจะถูกเชื่อมต่อด้วยกราฟแทนเนอร์ทั้งในแนวแถว และหลักพร้อมๆกัน ดังรูปที่ 4.4 แต่เมื่อทำการคำนวณค่าซินโดรม (s_m) และค่าความไม่น่าเชื่อถือ ($E_{i,j}$) ของแต่ละบิต เราจะพิจารณาการเชื่อมต่อจากกราฟแทนเนอร์เพียงแนวเดียวเท่านั้น ซึ่งขึ้นอยู่กับว่าในขณะนั้นบิตข้อมูลที่กำลังพิจารณาถูกถอดรหัสในแนวใด พิจารณาในรูปที่ 4.6 (ก) จะแสดงบิตข้อมูลที่กำลังถูกถอดรหัสในแนวแถว ซึ่งค่า s_m และ $E_{i,j}$ จะถูกคำนวณจากกราฟแทนเนอร์ในแนวแถวเท่านั้น โดยจะทำการตัดการเชื่อมต่อจากกราฟแทนเนอร์ในแนวหลักทิ้งไป เช่นเดียวกันในรูปที่ 4.6 (ข) บิตข้อมูลนั้นกำลังถูกถอดรหัสในแนวหลัก ซึ่งการเชื่อมต่อจากกราฟแทนเนอร์ในแนวแถวก็จะถูกตัดทิ้งไปเช่นเดียวกัน



รูปที่ 4.6 การเชื่อมต่อของกราฟแทนเนอร์กับบิตข้อมูลสำหรับถอดรหัสใน (ก) แถว, (ข) หลัก

เนื่องจากอัลกอริทึม RC-WBF จะทำการถอดรหัสบิตข้อมูลในแต่ละบล็อกในแนวแถว และหลักสลับกันไปมา ฉะนั้น การคำนวณค่า ($E_{i,j}$) จะกระทำเพียง n_1 บิต สำหรับแต่ละบล็อกในแนวแถว และ n_2 บิต สำหรับแต่ละบล็อกในแนวหลัก ต่อหนึ่งรอบการวนซ้ำ ดังนั้น ต่อให้คำรหัสแบบสองมิติ มีขนาดใหญ่เพิ่มขึ้นเพียงใด ความซับซ้อนในการถอดรหัสของอัลกอริทึม RC-WBF จะมีการเพิ่มขึ้นแบบเป็นเชิงเส้นเท่านั้น ซึ่งนั่นจะทำให้ความซับซ้อนในการถอดรหัสอัลกอริทึม RC-WBF มีค่าน้อยกว่าอัลกอริทึม P-WBF เสมอ ขั้นตอนการถอดรหัสด้วยอัลกอริทึม RC-WBF สามารถเขียนได้ดังนี้

กำหนดให้ ดัชนี i และ j คือ ตำแหน่งเวกเตอร์คาร์ทีเซียนแถวที่ i และหลักที่ j โดยเซตของ $M_{row}(j)$ และ $M_{col}(i)$ ถูกกำหนดให้เป็น ตำแหน่งเลข 1 ในหลักที่ j ของเมทริกซ์พาริตีเซต \mathbf{H}_1 ที่ใช้เข้ารหัสในแนวแถว และตำแหน่งเลข 1 ในหลักที่ i ของเมทริกซ์พาริตีเซต \mathbf{H}_2 ที่ใช้เข้ารหัสในแนวหลัก ในส่วนของเซต $N_{row}(m)$ และ $N_{col}(m)$ คือตำแหน่งเลข 1 ในแถวที่สัมพันธ์กับสมาชิกในเซต $M_{row}(j)$ และ $M_{col}(i)$ ของเมทริกซ์พาริตีเซต ที่ใช้เข้ารหัสในแนวแถว \mathbf{H}_1 และหลัก \mathbf{H}_2 ตามลำดับ

for k จะมีค่าตั้งแต่ 1 ถึงจำนวนการวนซ้ำภายนอก (global iteration) ที่กำหนด หรือจนกว่าบิตข้อมูลจะถูกต้องทุกบิต

ทำการตัดสินใจแบบหยาบของสัญญาณ y_{ij} โดย $z_{ij} = 1$ เมื่อ $y_{ij} \geq 0$ นอกนั้น $z_{ij} = 0$

for i มีค่าตั้งแต่ 1 ถึง n_2 (#แถวรหัสในแนวแถว)

ทำการคำนวณค่าซินโดรม s_m ของแต่ละบิตในบล็อกข้อมูลแถวที่ i (พิจารณาเฉพาะกราฟแทนเนอร์ในแนวแถว) จากนั้นคำนวณค่าความไม่น่าเชื่อถือของแต่ละบิต ($E_{i,j}$) จาก

$$E_{i,j} = \frac{1}{|y_{i,j}|} \sum_{m \in M_{row}(j)} (2s_m - 1) \left(\sum_{j \in N_{row}(m)} |y_{i,j}| \right)$$

ทำการพลิกบิตในตำแหน่งที่มีค่า $E_{i,j}$ สูงที่สุด ในบล็อกข้อมูลแถวที่ i และทำการคำนวณค่า s_m และ $E_{i,j}$ ซ้ำไปเรื่อยๆจนกว่าบิตข้อมูลในแถวนั้น จะถูกต้องทุกบิตหรือครบจำนวนรอบการวนซ้ำภายใน (inner iteration) ที่กำหนด

end

for j มีค่าตั้งแต่ 1 ถึง n_1 (#แถวรหัสในแนวหลัก)

ทำการคำนวณค่าซินโดรมของแต่ละบิตในบล็อกข้อมูลหลักที่ j (พิจารณาเฉพาะกราฟแทนเนอร์ในแนวหลัก) จากนั้นคำนวณค่าความไม่น่าเชื่อถือของแต่ละบิต ($E_{i,j}$) จาก

$$E_{i,j} = \frac{1}{|y_{i,j}|} \sum_{m \in M_{row}(i)} (2s_m - 1) \left(\sum_{i \in N_{row}(m)} |y_{i,j}| \right)$$

ทำการพลิกบิตในตำแหน่งที่มีค่า $E_{i,j}$ สูงที่สุด ในบล็อกข้อมูลหลักที่ j และทำการคำนวณค่า s_m และ $E_{i,j}$ ซ้ำไปเรื่อยๆจนกว่าบิตข้อมูลในแถวนั้น จะถูกต้องทุกบิตหรือครบจำนวนรอบการวนซ้ำภายใน (inner iteration) ที่กำหนด

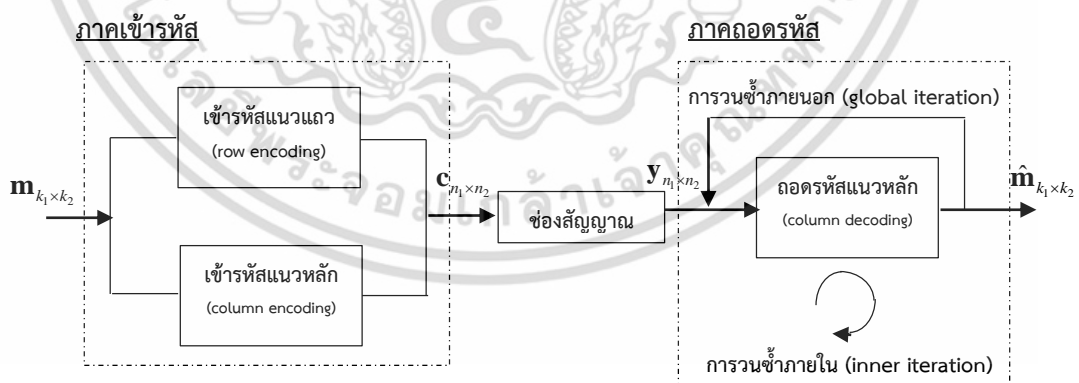
end

end

4.4 การถอดรหัสแบบบิตพลิกปิ้งชนิดถ่วงน้ำหนักแบบตัดแปลงแถวสลับหลัก (Modified row-column weighted bit-flipping decoding: MRC-WBF)

ในหัวข้อที่ 4.3 อัลกอริทึมการถอดรหัส RC-WBF ได้ถูกนำเสนอเป็นอัลกอริทึมที่สอง โดยอัลกอริทึมดังกล่าวเป็นการประยุกต์ใช้อัลกอริทึมบิตพลิกปิ้งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) เพื่อถอดรหัสบล็อกข้อมูลในแนวแถว และหลักสลับกันไปมาตามจำนวนการวนซ้ำที่กำหนด อย่างไรก็ตาม อัลกอริทึมดังกล่าวได้ถูกนำเสนอเพื่อปรับปรุงความซับซ้อนในการถอดรหัสของอัลกอริทึม P-WBF อย่างไรก็ตาม สมรรถนะของอัลกอริทึมดังกล่าว อาจจะแยกลงเมื่อเทียบกับอัลกอริทึม P-WBF อันเนื่องมาจาก อัลกอริทึม RC-WBF มีการถอดรหัสข้อมูลในแต่ละบล็อกภายในแถว และหลักอย่างแยกจากกัน ซึ่งจะพิจารณาผลกระทบที่เกิดจากกราฟแทนเนอร์เพียงแนวเดียวเท่านั้น อาจจะทำให้บิตข้อมูลได้รับข้อมูลเพื่อตัดสินใจในจำนวนที่น้อยกว่าอัลกอริทึม P-WBF

ในหัวข้อนี้ผู้วิจัยจึงได้ทำการนำเสนออัลกอริทึมที่ 3 โดยอัลกอริทึมนี้ สามารถลดช่องว่างของสมรรถนะระหว่างอัลกอริทึม P-WBF และ RC-WBF ได้ โดยใช้ความซับซ้อนในการถอดรหัสเพิ่มขึ้นเพียงเล็กน้อยจากอัลกอริทึม RC-WBF ผู้วิจัยจะให้ชื่ออัลกอริทึมใหม่นี้ว่า “อัลกอริทึมการถอดรหัสบิตพลิกปิ้งชนิดถ่วงน้ำหนักแบบตัดแปลงแถวสลับหลัก (MRC-WBF)” ในส่วนภาคถอดรหัสของอัลกอริทึมนี้ จะเป็นการผสมผสานจุดเด่นของสองอัลกอริทึม P-WBF และ RC-WBF เข้าด้วยกัน กล่าวคือ อัลกอริทึม MRC-WBF จะทำการคำนวณค่าซินโดรม (s_m) และความไม่น่าเชื่อถือ ($E_{i,j}$) ของแต่ละบิต จากกราฟแทนเนอร์ทั้งสองแนวเหมือนกับอัลกอริทึม P-WBF ซึ่งจะทำได้สมรรถนะในการถอดรหัสที่ดี พร้อมทั้งมีกระบวนการถอดรหัสข้อมูลในแต่ละบล็อกย่อยคล้ายกับอัลกอริทึม RC-WBF ซึ่งจะส่งผลให้ความซับซ้อนในการถอดรหัสมีค่าใกล้เคียงกับอัลกอริทึม RC-WBF พิจารณาแบบจำลองในภาคเข้า และถอดรหัสของอัลกอริทึม MRC-WBF ในรูปที่ 4.7



รูปที่ 4.7 แบบจำลองภาคเข้ารหัสและถอดรหัสของอัลกอริทึม MRC-WBF

หมายเหตุ สำหรับในงานวิจัยเล่มนี้ อัลกอริทึม MRC-WBF สามารถที่จะใช้การถอดรหัสในแนวแถว หรือแนวหลักอย่างใดอย่างหนึ่งแทนก็ได้ ซึ่งจะไม่ส่งผลกระทบต่อสมรรถนะของอัลกอริทึมการถอดรหัส เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการวิจัยเท่านั้น เมื่อผู้ใดเห็นประโยชน์ในการนำเอกสารนี้ไปใช้ ไม่ว่าจะโดยวิธีใด ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับแบบจำลองการเข้า และถอดรหัสของอัลกอริทึม MRC-WBF ในรูปที่ 4.7 บิตข้อมูลจะถูกเข้ารหัสในแต่ละแถว และหลักอย่างแยกจากกันคล้ายกับอัลกอริทึม P-WBF และ RC-WBF ส่วนในภาคถอดรหัส จะเป็นการผสมผสานกันระหว่างภาคถอดรหัสของอัลกอริทึม P-WBF และ RC-WBF โดยสัญญาณที่ได้รับ จะถูกนำไปถอดรหัสในแต่ละบล็อกภายในแนวหลักเพียงแนวเดียวเท่านั้น แต่สำหรับการถอดรหัสแต่ละบล็อกในแนวหลักนั้น ค่าซินโดรม (s_m) และความไม่น่าเชื่อถือ ($E_{i,j}$) ของแต่ละบิต จะถูกคำนวณจากกราฟแทนเนอร์ทั้งสองแนวเหมือนกันกับ P-WBF สังเกตว่าภายในภาคถอดรหัสของอัลกอริทึม MRC-WBF จะเป็นการรวมจุดเด่นของ 2 อัลกอริทึมเข้าด้วยกันคือ การพิจารณาค่าซินโดรม (s_m) และความไม่น่าเชื่อถือ ($E_{i,j}$) จากกราฟแทนเนอร์ทั้งสองแนวเหมือนกับอัลกอริทึม P-WBF (ทำให้ได้สมรรถนะที่ดี) และการถอดรหัสเป็นบล็อกย่อยๆ คล้ายกับอัลกอริทึม RC-WBF (ช่วยให้ความซับซ้อนในการถอดรหัสต่ำ) อย่างไรก็ตาม อัลกอริทึม MRC-WBF ก็จะเป็นการรวมจุดเด่นของทั้งสองอัลกอริทึมเข้าด้วยกันอีกด้วย ยกตัวอย่างเช่น ภายหลังจากฟิลิป 1 บิต จะมีการอัปเดตค่าซินโดรมจากโนดเช็คเป็น 2 เท่าคล้ายกับอัลกอริทึม P-WBF (ส่งผลให้มีความซับซ้อนในการถอดรหัสสูงขึ้นเล็กน้อย) และมีการเลือกบิตที่ควรจะถูกฟิลิปในแต่ละรอบของการวนซ้ำ จากคู่แข่งที่มีจำนวนเพียง n_1 บิต คล้ายกับอัลกอริทึม RC-WBF (ส่งผลให้สมรรถนะมีการลดทอน) จากเหตุผลดังกล่าวข้างต้น ทำให้ทั้งสมรรถนะ และความซับซ้อนในการถอดรหัสของอัลกอริทึม MRC-WBF อยู่ตรงกลางระหว่างอัลกอริทึม P-WBF และ RC-WBF สำหรับการวนซ้ำของอัลกอริทึม MRC-WBF จะมีความเหมือนกันกับอัลกอริทึม RC-WBF ที่จะประกอบไปด้วยการวนซ้ำภายใน (inner iteration) และการวนซ้ำภายนอก (global iteration) แต่จะมีการนิยามความหมายที่แตกต่างกันดังนี้

- 1.) การวนซ้ำภายนอก (global iteration) คือ จำนวนครั้งในการป้อนกลับค่าฮาร์ดทาบภายหลังการถอดรหัสในหลักสุดท้ายกลับมายังหลักแรกอีกครั้ง
- 2.) การวนซ้ำภายใน (inner iteration) คือการวนซ้ำของรหัสแอลดีพีซีแบบบิตฟิลิปปีงชนิดถ่วงน้ำหนัก (RR-WBF) ในแต่ละบล็อกของแนวหลัก หรือคือจำนวนบิตที่ถูก ฟิลิปในแต่ละบล็อกนั่นเอง

อันเนื่องมาจาก การวนซ้ำของทั้ง 3 อัลกอริทึมที่นำเสนอ นั้น มีการนิยามความหมายที่ต่างกััน ดังนั้น เพื่อความยุติธรรมในการเปรียบเทียบสมรรถนะในบทถัดไป ขั้นตอนแรก ผู้วิจัยจะทำการกำหนดจำนวนบิตทั้งหมดที่ต้องการแก้ไขให้ถูกต้องมีจำนวนเท่ากับ Q บิต ฉะนั้น หากแต่ละอัลกอริทึมต้องการที่จะฟิลิปบิตจำนวน Q บิต จำต้องใช้การวนซ้ำทั้งสิ้น

สำหรับอัลกอริทึม P-WBF

$$Q = I \quad (4.9)$$

สำหรับอัลกอริทึม RC-WBF

$$Q = (I_1 n_1 + I_2 n_2) I_g \quad (4.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับอัลกอริทึม MRC-WBF

$$Q = 2(I_1 n_1 + I_2 n_2) I_g \quad (4.11)$$

โดยที่ I , I_1 และ I_g ถูกกำหนดให้เป็น การวนซ้ำของอัลกอริทึมการถอดรหัส P-WBF, การวนซ้ำภายใน และการวนซ้ำภายนอก ตามลำดับ ขั้นตอนการถอดรหัสด้วยอัลกอริทึม MRC-WBF สามารถเขียนได้ดังนี้

กำหนดให้ ดัชนี i และ j คือ ตำแหน่งเวกเตอร์คาร์รหัสแถวที่ i และหลักที่ j โดยเซตของ $M_{row}(j)$ และ $M_{col}(i)$ ถูกกำหนดให้เป็น ตำแหน่งเลข 1 ในหลักที่ j ของเมทริกซ์พาริตีเช็ค \mathbf{H}_1 ที่ใช้เข้ารหัสในแนวแถว และตำแหน่งเลข 1 ในหลักที่ i ของเมทริกซ์พาริตีเช็ค \mathbf{H}_2 ที่ใช้เข้ารหัสในแนวหลัก ในส่วนของเซต $N_{row}(m)$ และ $N_{col}(m)$ คือตำแหน่งเลข 1 ในแถวที่สัมพันธ์กับสมาชิกในเซต $M_{row}(j)$ และ $M_{col}(i)$ ของเมทริกซ์พาริตีเช็ค ที่ใช้เข้ารหัสในแนวแถว \mathbf{H}_1 และหลัก \mathbf{H}_2 ตามลำดับ

for k จะมีค่าตั้งแต่ 1 ถึงจำนวนการวนซ้ำภายนอก (global iteration) ที่กำหนด หรือจนกว่าบิตข้อมูลจะถูกต้องทุกบิต

ทำการตัดสินใจแบบหยาบของสัญญาณ y โดย $z_{ij} = 1$ เมื่อ $y_{ij} \geq 0$ นอกนั้น $z_{ij} = 0$

for j มีค่าตั้งแต่ 1 ถึง n_1 (#ถอดรหัสในแนวหลัก)

ทำการคำนวณค่าซินโดรม (s_m) ของแต่ละบิต ในบล็อกข้อมูลหลักที่ j (แต่พิจารณากราฟแทนเนอร์ในแนวแถว และหลักพร้อมๆกัน เหมือนอัลกอริทึม P-WBF) จากนั้นคำนวณค่าความไม่น่าเชื่อถือของแต่ละบิต ($E_{i,j}$) จาก

$$E_{i,j} = \frac{1}{|y_{i,j}|} \left[\sum_{m \in M_{row}(j)} (2s_m - 1) \left(\sum_{j \in N_{row}(m)} |y_{i,j}| \right) + \sum_{m \in M_{col}(i)} (2s_m - 1) \left(\sum_{i \in N_{col}(m)} |y_{i,j}| \right) \right]$$

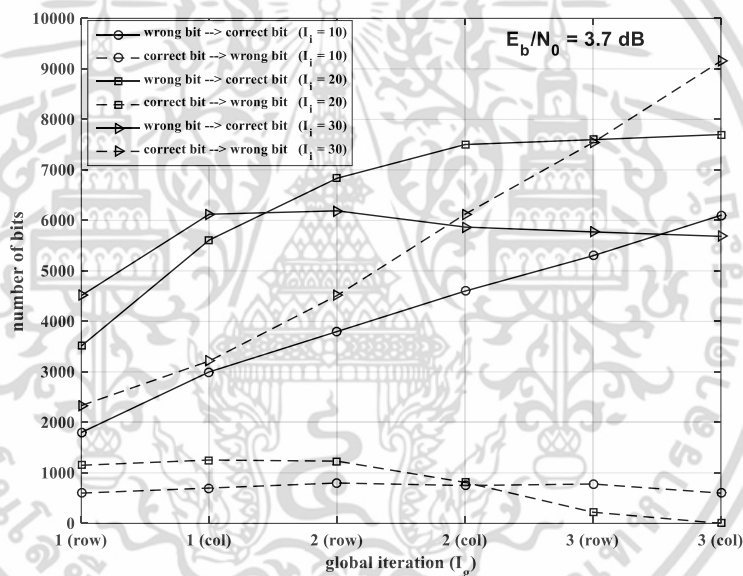
ทำการพลิกบิตในตำแหน่งที่มีค่า $E_{i,j}$ สูงที่สุด ในบล็อกข้อมูลหลักที่ j และทำการคำนวณค่า s_m และ $E_{i,j}$ ซ้ำไปเรื่อยๆจนกว่าบิตข้อมูลในแถวนั้น จะถูกต้องทุกบิตหรือครบจำนวนรอบการวนซ้ำภายใน (inner iteration) ที่กำหนด

end

end

4.5 ปัญหาสมรรถนะไม่ลู่เข้าที่การวนซ้ำรอบสูง (V-shape situation)

เป็นที่ทราบกันดีว่า รหัสแอลดีพีซีมีการถอดรหัสแบบวนซ้ำ ที่ซึ่งความสามารถในการแก้ไขบิตผิดพลาดหรือสมรรถนะของรหัสแอลดีพีซีจะมีค่าที่ดีขึ้น เมื่อจำนวนรอบในการวนซ้ำเพิ่มขึ้น จากนั้นที่จุดหนึ่งๆ ความสามารถในการแก้ไขบิตผิดพลาดหรือสมรรถนะของรหัสแอลดีพีซีจะมีค่าที่คงที่แม้จำนวนการวนซ้ำจะถูกเพิ่มเป็นเท่าไรก็ตาม เช่นเดียวกันกับรหัสแอลดีพีซีแบบสองมิติที่ถูกกล่าวถึงในวิทยานิพนธ์เล่มนี้ อาทิเช่น อัลกอริทึมการถอดรหัสแอลดีพีซีแบบโปรดักส์ที่เป็นการถอดรหัสแบบซอฟต์แวร์ และการถอดรหัสแบบบิตฟลิปปีงชนิดเพจ (P-WBF) ที่เป็นการถอดรหัสแบบหยาบเป็นต้น ในทางตรงกันข้าม เมื่อนำอัลกอริทึมการถอดรหัส RC-WBF และ MRC-WBF มาใช้กับการวนซ้ำภายใน (inner iteration) จากรอบต่ำไปรอบสูง เพื่อวิเคราะห์ความสามารถในการแก้ไขบิตผิดพลาดของรหัสแอลดีพีซีหรือดูจำนวนบิตผิดพลาดที่สามารถแก้ไขได้บนช่องสัญญาณเกาส์สีขาว จะได้ผลดังแสดงในรูป 4.8 การจำลองนี้จะกำหนดให้การวนซ้ำภายนอก (global iteration) มีค่าเท่ากับ 1 - 3 และบิตข้อมูลสองมิติมีขนาดเท่ากับ 500×500 บิต อัตรารหัสคือ 0.5



รูปที่ 4.8 ความสามารถในการแก้ไขบิตผิดพลาดของอัลกอริทึม RC-WBF

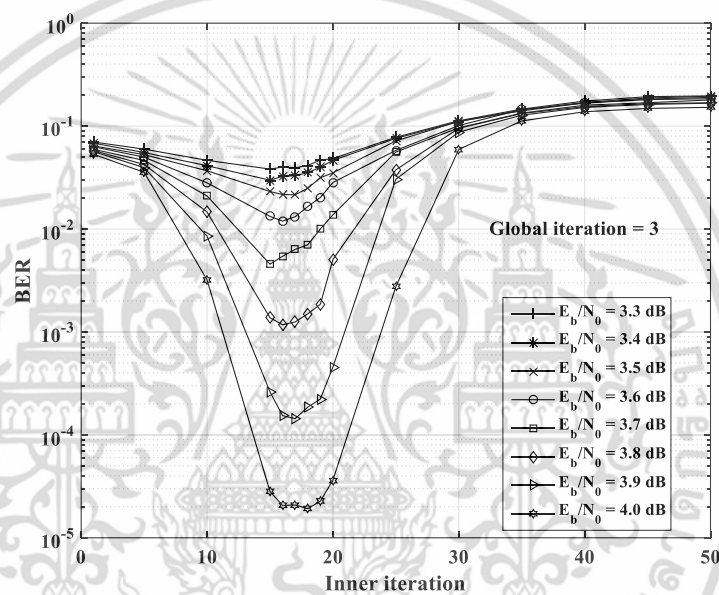
พิจารณารูปที่ 4.8 เป็นการจำลองความสามารถในการแก้ไขบิตที่ผิดพลาดของอัลกอริทึม RC-WBF โดยเส้นที่บ่งชี้จำนวนบิตที่สามารถแก้ไขจากบิตที่ผิดพลาดให้เป็นบิตที่ถูกต้อง เส้นประจะแทนจำนวนบิตที่ถูกต้องอยู่แล้วแต่ถูกฟลิปให้กลายเป็นบิตที่ผิดพลาด จะสังเกตว่า เมื่อกำหนดให้การวนซ้ำภายในมีค่าเท่ากับ 10 เมื่อทำการเพิ่มจำนวนการวนซ้ำภายนอก พบว่าความสามารถในการแก้ไขบิตผิดพลาดจะมีค่าที่มากขึ้น แต่ในส่วนการฟลิปบิตที่ถูกต้องให้กลายเป็นบิตผิดพลาดจะมีจำนวนน้อยอย่างไม่มีนัยยะสำคัญ จากนั้น ทำการเพิ่มการวนซ้ำภายในให้มีค่าเท่ากับ 20 จะพบว่าเมื่อทำการเพิ่มการวนซ้ำภายนอกนั้น ความสามารถในการแก้ไขบิตผิดพลาดจะมีค่าเพิ่มขึ้นด้วยความชันสูง และหลังจากนั้นจะเริ่มคงที่ที่การวนซ้ำภายนอกในรอบที่ 2 ของการถอดรหัสในแนวหลัก โดยในส่วนการฟลิปบิตที่ถูกต้องให้กลายเป็นบิตผิดพลาด จะยังคงมีค่าที่น้อยอย่างไม่มีนัยยะสำคัญ ในกรณี

ดังกล่าวจะเป็นไปตามทฤษฎีของการถอดรหัสแอลดีพีซี ที่ความสามารถในการแก้ไขความผิดพลาดจะ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการเรียนการสอนเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

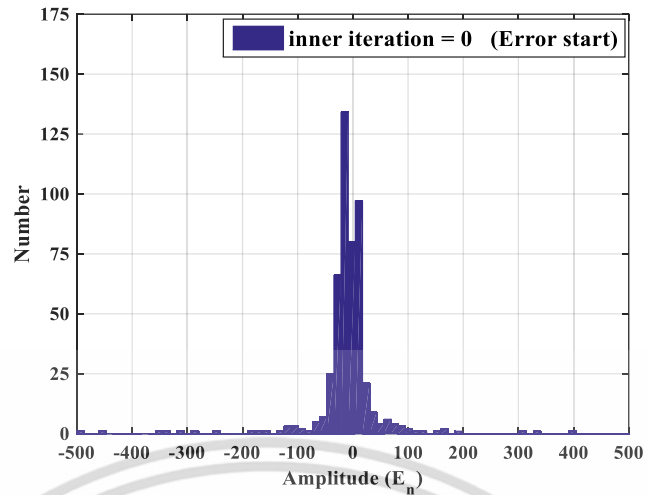
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีค่าที่ต่ำลงเรื่อยๆเมื่อมีการวนซ้ำที่มากขึ้น จากนั้นจึงคงที่ อย่างไรก็ตาม หากพิจารณาที่การวนซ้ำภายในรอบสูงหรือที่ประมาณ 30 รอบ ความสามารถในการแก้ไขความผิดพลาดก็ยังมีที่ต่ำลง เมื่อมีการวนซ้ำภายนอกที่มากขึ้น จากนั้นจึงจะคงที่ แต่จำนวนบิตที่ต้องแต่กลับถูกพลิกให้เป็นบิตผิดพลาดกลับมีจำนวนมากขึ้นตามไปด้วย ซึ่งเมื่อนำมาหักล้างกันแล้วนั้น จะทำให้การถอดรหัสด้วยอัลกอริทึม RC-WBF ที่การวนซ้ำภายใน และภายนอกรอบสูง ไม่สามารถที่จะแก้ไขบิตที่ผิดพลาดให้กลายเป็นบิตที่ถูกต้องได้ หรือมีประสิทธิภาพต่ำในการแก้ไขความผิดพลาด เนื่องจากจำนวนบิตที่ถูกพลิกจากบิตที่ถูกต้องให้เป็นบิตผิดพลาดมีจำนวนมากกว่า เพราะฉะนั้น ต่อไปนี้ผู้วิจัยจะทำการจำลองสมรรถนะของอัลกอริทึมการถอดรหัส RC-WBF ในรูปแบบกราฟ BER (Bit error rate) โดยจะทำการกำหนดให้การวนซ้ำภายนอกมีค่าคงที่เท่ากับ 3 (รอบสูง) เพื่อดูว่าที่การวนซ้ำภายในเท่าใดที่ความสามารถในการแก้ไขบิตผิดพลาดดีที่สุด

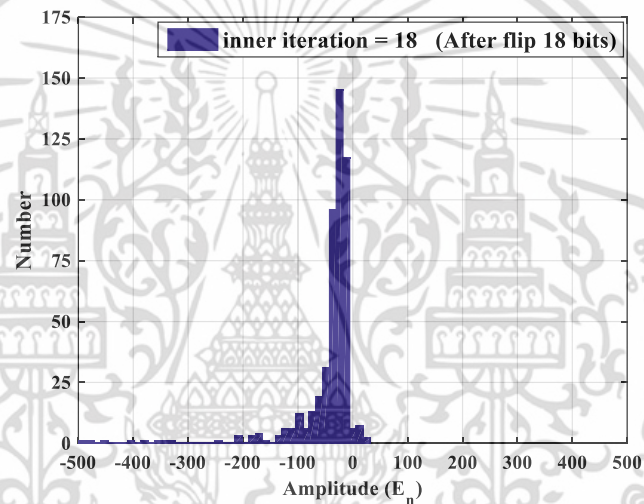


รูปที่ 4.9 ปัญหาสมรรถนะไม่ลู่เข้า ที่การวนซ้ำภายในและนอกกรอบสูงของ RC-WBF

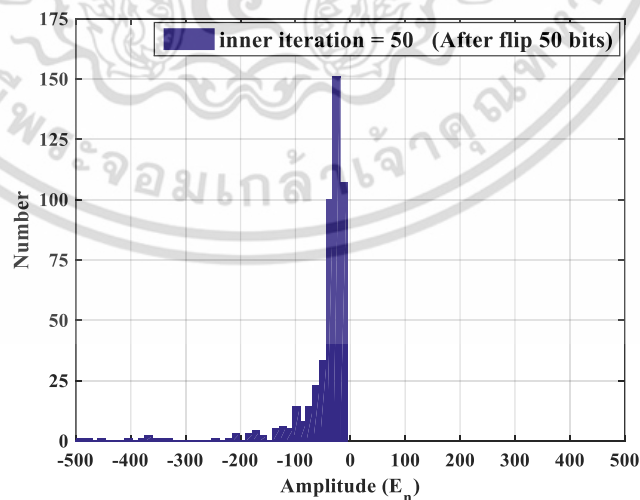
จากกราฟรูปที่ 4.9 ค่า BER จะถูกนิยามเป็นอัตราส่วนระหว่างบิตผิดพลาดหารด้วยจำนวนบิตทั้งหมด โดยที่ จะทำการกำหนดให้การวนซ้ำภายนอกมีค่าเท่ากับ 3 (รอบสูง) และทำการเพิ่มการวนซ้ำภายในไปเรื่อยๆ จากกราฟจะพบว่า เมื่อการวนซ้ำภายในมีค่าเท่ากับ 18 จะให้สมรรถนะที่ดีที่สุด ในทุกช่วง E_b/N_0 กล่าวคือ ผลต่างระหว่างจำนวนบิตที่ถูกพลิกจากบิตผิดพลาดให้เป็นบิตที่ถูกต้อง และจำนวนบิตที่ถูกต้องแต่ถูกพลิกให้เป็นบิตที่ผิดพลาด มีจำนวนมากที่สุด นั่นหมายความว่า เมื่อการวนซ้ำภายในมีค่าเท่ากับ 18 จะมีความสามารถในการแก้ไขความผิดพลาดที่ดีที่สุดนั่นเอง อย่างไรก็ตาม ภายหลังจากการวนซ้ำภายในรอบที่ 18 เป็นต้นไปนั้น กราฟ BER จะมีค่าที่สูงขึ้น (สมรรถนะแย่งลง) สาเหตุมาจากจำนวนบิตที่ถูกต้องแต่กลับถูกพลิกให้เป็นบิตที่ผิดพลาดเริ่มมีค่าที่มากขึ้นเรื่อยๆ ในขณะที่เดียวกันจำนวนบิตที่ผิดพลาดที่ถูกพลิกให้ถูกต้องกลับมีค่าคงที่ ดังนั้น เพื่อแก้ไขปัญหสมรรถนะที่ไม่ลู่เข้าข้างต้นนั้น ผู้วิจัยจะทำการวิเคราะห์การกระจายตัวของค่า $E_{i,j}$ ที่การวนซ้ำภายในรอบต่างๆ ด้วยกราฟฮิสโทแกรมดังนี้



(ก)



(ข)



(ค)

รูปที่ 4.10 การแพร่กระจายของค่า $E_{i,j}$ ที่การวนซ้ำภายในรอบที่ (ก) 0, (ข) 18 และ (ค) 50 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟฮิสโทแกรมในรูปที่ 4.10 แสดงถึง การกระจายตัวของค่า $E_{i,j}$ ที่การวนซ้ำภายในในรอบต่างๆ โดยกำหนดให้การวนซ้ำภายนอกมีค่าเท่ากับ 3 จากกราฟรูปที่ 4.10ก เป็นการแสดงการกระจายตัวของค่า $E_{i,j}$ ก่อนที่บิตแรกจะถูกพลิก โดยค่า $E_{i,j}$ ที่มีแอมพลิจูดสูง จะแสดงถึงบิตที่มีค่าความน่าเชื่อถือต่ำ ซึ่งเป็นบิตที่ควรจะต้องถูกพลิกไป แต่ในส่วนของบิตที่มีค่า $E_{i,j}$ ใกล้เคียง 0 หรือมีค่าน้อยกว่า 0 จะแสดงถึงบิตที่มีความน่าเชื่อถือสูงนั่นเอง เนื่องจากค่า $E_{i,j}$ ถูกกำหนดจากค่าซินโดรมและขนาดของสัญญาณที่ได้รับ สังเกตที่การวนซ้ำภายในในรอบที่ 18 หรือเมื่อบิตข้อมูลจำนวน 18 บิตที่มีความน่าเชื่อถือต่ำถูกพลิกไป ดังแสดงในรูปที่ 4.10ข พบว่าค่า $E_{i,j}$ ที่สูงที่สุด จะมีค่าที่เข้าใกล้ 0 ซึ่งถ้าหากพิจารณาในกราฟรูปที่ 4.9 ควบคู่กันไปด้วยจะพบว่า การวนซ้ำในรอบที่ 18 นี้ จะให้ค่า BER หรือสมรรถนะในการแก้ไขบิตผิดพลาดดีที่สุด อย่างไรก็ตาม ในรูปที่ 4.10ค นั้นแสดงถึงการที่ 50 บิตถูกพลิกไปหรือที่การวนซ้ำภายในในรอบที่ 50 จะพบว่าบิตที่มีค่า $E_{i,j}$ ใกล้เคียง 0 และน้อยกว่า 0 จะถูกพลิกไป ซึ่งในกรณีนี้อาจทำให้บิตข้อมูลที่มีความน่าเชื่อถือสูงหรือบิตข้อมูลที่มีความถูกต้องอยู่แล้วถูกพลิกไปได้ ประกอบกับการที่อัลกอริทึมการถอดรหัส RC-WBF มีการถอดรหัสที่แยกกันระหว่างแนวแถว และหลัก ซึ่งนั่นจะทำให้ความผิดพลาดในการพลิกบิต มีการสะสมเพิ่มมากขึ้น และส่งผลออกมาอย่างเด่นชัด ที่การวนซ้ำภายใน และภายนอกรอบสูง

ดังนั้น ผู้วิจัยจึงทำการปรับปรุงอัลกอริทึมการถอดรหัส RC-WBF เพื่อหลีกเลี่ยงการพลิกบิตที่มีความน่าเชื่อถือสูง วิธีการที่ถูกนำมาใช้ในวิทยานิพนธ์เล่มนี้คือ การกำหนดค่าขอบเขตเทรสโฮอล์ดของค่า $E_{i,j}$ ที่จะเป็นตัวกำหนดว่า ค่า $E_{i,j}$ สูงสุดในขณะนั้นควรมีค่าเท่าใด จึงจะทำการหยุดยั้งอัลกอริทึม ตารางที่ 4.1 และ 4.2 จะแสดงถึงค่าเทรสโฮอล์ดของค่า $E_{i,j}$ ที่เป็นตัวกำหนดขอบเขตในการพลิกบิตสำหรับบิตข้อมูลขนาด 500×500 บิต ซึ่งนักวิจัยได้มาจากการจำลองซ้ำแล้วซ้ำเล่าเพื่อหาค่าเทรสโฮอล์ดที่เหมาะสมที่สุดภายใต้เงื่อนไขต่างๆ

ตารางที่ 4.1 ค่าเทรสโฮอล์ดที่เหมาะสมสำหรับ RC-WBF และ MRC-WBF บนช่องสัญญาณ AWGN

Global iteration	RC-WBF			Global iteration	MRC-WBF		
	$R^{2D} = 0.5$	$R^{2D} = 0.6$	$R^{2D} = 0.7$		$R^{2D} = 0.5$	$R^{2D} = 0.6$	$R^{2D} = 0.7$
1 (แนวแถว)	19.5	26	32	1	24	40	50
1 (แนวหลัก)	16	24	30				
2 (แนวแถว)	10	16	22	2	-	-	-
2 (แนวหลัก)	3	4	5				
3 (แนวแถว)	-	-	-	3	-	-	-
3 (แนวหลัก)	-	-	-				
4 (แนวแถว)	-	-	-	4	-	-	-
4 (แนวหลัก)	-	-	-				
5 (แนวแถว)	-	-	-	5	-	-	-
5 (แนวหลัก)	-	-	-				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าเทรสโฮลด์ที่เหมาะสมสำหรับ RC-WBF และ MRC-WBF บนช่องสัญญาณ TDMR โดยใช้ ค่าพารามิเตอร์ตาม S. Kobayashi [23-24]

Global iteration	RC-WBF	Global iteration	MRC-WBF
	$R^{2D} = 0.7$		$R^{2D} = 0.7$
1 (แนวแถว)	45	1	60
1 (แนวหลัก)	30		
2 (แนวแถว)	20	2	-
2 (แนวหลัก)	8		
3 (แนวแถว)	-	3	-
3 (แนวหลัก)	-		
4 (แนวแถว)	-	4	-
4 (แนวหลัก)	-		
5 (แนวแถว)	-	5	-
5 (แนวหลัก)	-		

H = 26 nm.

G = 15 nm.

T = 13 nm.

U = 2 nm.

$h_m = 4$ nm.

$t_m = 10$ nm.

$t_s = 1$ nm.

$T_p = 22$ nm.

$L_b = 16.5$ nm.

วิธีในการใช้ค่าเทรสโฮลด์ในตารางที่ 4.1 และ 4.2 กับอัลกอริทึม RC-WBF และ MRC-WBF คือ จะต้องทำการตรวจสอบค่า $E_{i,j}$ ที่สูงที่สุด ทุกๆ ครั้งเสมอ ก่อนที่จะทำการพลิกบิตใดๆ ก็ตาม ถ้าหากค่า $E_{i,j}$ ที่สูงที่สุดในขณะนั้น มีค่าน้อยกว่าค่าเทรสโฮลด์เมื่อใด การถอดรหัสในบล็อกๆ นั้น จะหยุดลงทันที ตารางที่ 4.1 แสดงค่าเทรสโฮลด์ของสองอัลกอริทึมอันได้แก่ RC-WBF และ MRC-WBF เมื่อถูกใช้บนช่องสัญญาณเกาส์สีขาว จะพบว่าค่าเทรสโฮลด์ที่เป็นตัวกำหนดขอบเขตในการพลิกบิตนั้น จะขึ้นอยู่กับอัตราหัส และจะไม่ขึ้นอยู่กับค่า E_b/N_0 การใช้ค่าเทรสโฮลด์สำหรับอัลกอริทึม RC-WBF จะถูกใช้เฉพาะที่การวนซ้ำภายนอกที่น้อยกว่าเท่ากับ 2 เท่านั้น โดยในส่วนของ การวนซ้ำรอบสูงกว่านั้น ค่าเทรสโฮลด์จะถูกกำหนดให้มีค่าเท่ากับลบอินฟินิตี้ หมายความว่า ค่าเทรสโฮลด์จะไม่มีผลต่อสมรรถนะหรือความสามารถในการแก้ไขผิดพลาดของอัลกอริทึม เช่นเดียวกับค่าเทรสโฮลด์ของอัลกอริทึมการถอดรหัส MRC-WBF ที่จะถูกใช้เฉพาะที่การวนซ้ำภายนอกเท่ากับ 1 เท่านั้น และหลังจากนั้นค่าเทรสโฮลด์ก็จะถูกกำหนดให้มีค่าเท่ากับลบอินฟินิตี้เช่นเดียวกัน อย่างไรก็ตาม ที่ช่องสัญญาณที่มีการแทรกสอดสัญญาณบนช่องสัญญาณบันทึกเชิงแม่เหล็กสองมิติ (ค่าพารามิเตอร์ตาม S. Kobayashi [23-24]) ค่าเทรสโฮลด์สำหรับทั้งสองอัลกอริทึมถูกแสดงในตารางที่ 4.2 โดยมีแนวโน้ม และการใช้งานเหมือนกันกับที่ช่องสัญญาณเกาส์สีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ความซับซ้อนในการถอดรหัส

เป็นที่ทราบกันดีแล้วว่า ในภาคเข้ารหัสของอัลกอริทึมที่นำเสนอทั้ง 3 อันได้แก่ อัลกอริทึม P-WBF, RC-WBF และ MRC-WBF รวมถึงอัลกอริทึมแอสติฟิซีแบบโปรดักส์ [9] จะมีวิธีการในการเข้ารหัสที่เหมือนกัน ฉะนั้น ความซับซ้อนในการเข้ารหัสของอัลกอริทึมดังกล่าวข้างต้น จึงมีค่าที่เท่ากัน และมีค่านี้น้อยกว่าอัลกอริทึมแอสติฟิซีหนึ่งมิติทั่วๆ ไป (สามารถดูอัตราส่วนความซับซ้อนในการเข้ารหัสของทั้ง 3 อัลกอริทึม จากสมการที่ 4.5 – 4.6) อย่างไรก็ตาม ถึงแม้ภาคเข้ารหัสของอัลกอริทึมทั้งสาม จะมีความซับซ้อนที่เท่ากัน แต่ในภาคถอดรหัสนั้น ความซับซ้อนในการถอดรหัสของทั้ง 3 อัลกอริทึมจะมีความแตกต่างกันอย่างชัดเจน เนื่องจากวิธีการในการถอดรหัสของทั้ง 3 อัลกอริทึมที่นำเสนอ รวมถึงอัลกอริทึมการถอดรหัสแอสติฟิซีแบบโปรดักส์มีความแตกต่างกันตามที่ได้อธิบายไว้ในหัวข้อ 4.1 – 4.4 เพราะฉะนั้นในหัวข้อต่อไปนี้จะทำการเปรียบเทียบความซับซ้อนในการถอดรหัสของทั้ง 4 อัลกอริทึม อันได้แก่ อัลกอริทึมการถอดรหัสแอสติฟิซีแบบโปรดักส์ (PCB), อัลกอริทึมบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบเพจ (P-WBF), อัลกอริทึมบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบแถวสลับหลัก (RC-WBF) และอัลกอริทึมอัลกอริทึมบิตฟลิปบิงชนิดถ่วงน้ำหนักแบบตัดแปลงแถวสลับหลัก (MRC-WBF)

โดยปกติแล้วความซับซ้อนในการถอดรหัสจะถูกประเมินจากจำนวนโอเปอเรชันภายในวงจรฮาร์ดแวร์ไม่ว่าจะเป็น วงจรคุณ, วงจรหาร หรือวงจรบวก โดยใน [30] นักวิจัยชื่อ Z. Liu และ D. A. Pados ผู้ที่ซึ่งได้ทำการออกแบบ และศึกษาวงจรฮาร์ดแวร์สำหรับการถอดรหัสแอสติฟิซีด้วยวิธีการต่างๆ เขาได้ทำการเปรียบเทียบความซับซ้อนในการถอดรหัสแอสติฟิซีด้วยอัลกอริทึมการถอดรหัสแบบซัมโปรดักส์ (SPA) [27] กับอัลกอริทึมการถอดรหัสแอสติฟิซีแบบบิตฟลิปบิงชนิดถ่วงน้ำหนัก (WBF) หลายๆชนิด เป็นที่น่าเสียดาย ที่เขาได้ทำการศึกษา และเปรียบเทียบเฉพาะรหัสแอสติฟิซีแบบหนึ่งมิติเท่านั้น ดังนั้นในวิทยานิพนธ์เล่มนี้ ที่เป็นการศึกษาการถอดรหัสข้อมูลสองมิติ จึงได้ทำการนำความซับซ้อนของอัลกอริทึมการถอดรหัสแอสติฟิซีหลายๆชนิด ที่นำเสนอใน [30] มาประยุกต์ใช้กับอัลกอริทึมการถอดรหัสแอสติฟิซีสองมิติ ที่ได้นำเสนอในวิทยานิพนธ์เล่มนี้ ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ความซับซ้อนของการถอดรหัส สำหรับการวนซ้ำ 1 รอบ ของอัลกอริทึม PCB, P-WBF, RC-WBF และ MRC-WBF

ชนิดอัลกอริทึม	จำนวนวงจรถคูณ	จำนวนวงจรถหาร	จำนวนวงจรถบวก
PCB [9]	$11n_1t_c - 6(2n_1 - k_1)$	$n_1(t_c + 1)$	$n_1^2(3t_c + 1)$
P-WBF (นำเสนอ 1)	0	n_1n_2	$2n_1n_2d_vd_c$
RC-WBF (นำเสนอ 2)	0	n_1	$n_1d_vd_c$
MRC-WBF (นำเสนอ 3)	0	n_1	$2n_1d_vd_c$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ d_v และ d_c แทนจำนวนเลข 1 ในแต่ละแถวและหลักของเมทริกซ์พาริตีที่เช็คตามลำดับ โดยกำหนดให้เป็นรหัสแอลดีพีซีแบบสม่ำเสมอ และใช้เมทริกซ์ตัวเดียวกันสำหรับเข้ารหัสในแนวแถว และหลัก จากตารางที่ 4.3 จะพบว่า อัลกอริทึมที่นำเสนอทั้งสามอันได้แก่ อัลกอริทึม P-WBF, RC-WBF และ MRC-WBF จะมีความซับซ้อนในการถอดรหัสที่ต่ำกว่าอัลกอริทึม PCB เนื่องจากอัลกอริทึมทั้งสามดังกล่าวข้างต้น จะไม่ใช้วงจรคูณสำหรับการถอดรหัสข้อมูล สำหรับการวนซ้ำในแต่ละรอบ

การเปรียบเทียบความซับซ้อนในการถอดรหัสของอัลกอริทึมที่นำเสนอ นั้น จะพิจารณาจากจำนวนวงจรรวมและบวกทั้งหมด ที่ใช้สำหรับคำนวณค่า $E_{i,j}$ ต่อหนึ่งรอบการวนซ้ำหรือต่อการพลิกบิต 1 ครั้ง จากตารางพบว่า อัลกอริทึม P-WBF จะมีความซับซ้อนในการถอดรหัสที่มากที่สุด เนื่องจาก มีการคำนวณค่าความไม่น่าเชื่อถือ $E_{i,j}$ ทั้งหมด $n_1 n_2$ บิต ต่อหนึ่งรอบการวนซ้ำหรือต่อการพลิกบิต 1 ครั้ง ฉะนั้น จะส่งผลทำให้ความซับซ้อนในการถอดรหัสมีการเพิ่มขึ้นแบบฟังก์ชันกำลังสองเมื่อบิตข้อมูลมีขนาดใหญ่มากขึ้น อย่างไรก็ตาม สำหรับอัลกอริทึม RC-WBF และ MRC-WBF จะมีความซับซ้อนที่ใกล้เคียงกัน เนื่องจาก มีการถอดรหัสเป็นบล็อกย่อยๆ ในแนวแถวหรือหลักสลับกันไปมา กล่าวคืออัลกอริทึมทั้งสองนี้ มีการคำนวณค่า $E_{i,j}$ ทั้งหมด n_1 หรือ n_2 บิต ต่อหนึ่งรอบการวนซ้ำหรือต่อการพลิกบิต 1 ครั้งเท่านั้น ทำให้ความซับซ้อนมีการเพิ่มขึ้นแบบเชิงเส้น เมื่อบิตข้อมูลมีขนาดใหญ่มากขึ้น ถ้าหากพิจารณาในเชิงลึก จะพบว่าความซับซ้อนในการถอดรหัสของอัลกอริทึม MRC-WBF จะมีค่าที่มากกว่าอัลกอริทึม RC-WBF เล็กน้อย เนื่องจากมีการคำนวณค่า $E_{i,j}$ แบบสองมิติ (พิจารณารูปแทนเนอร์ในแนวแถว และหลักพร้อมๆ กัน) ดังนั้น จำนวนโนดเช็คและโนดบิตที่ใช้สำหรับคำนวณค่า $E_{i,j}$ จะเป็นสองเท่าของอัลกอริทึม RC-WBF เสมอ จำนวนโนดเช็คและโนดบิตที่ใช้แสดงเป็น d_v และ d_c ตามลำดับ

4.7 การพลิกหลายบิตเพื่อเพิ่มความเร็วของอัลกอริทึม (P-WBF)

ในหัวข้อต่อไปนี้จะเน้นย้ำไปที่อัลกอริทึม P-WBF ที่เป็นการถอดรหัสแอลดีพีซีแบบหลายสองมิติ โดยเป็นการนำเอาอัลกอริทึมการถอดรหัสบิตพลิกบิตซึ่งชนิดถ่วงน้ำหนักแบบอัตราส่วนความน่าเชื่อถือ (RR-WBF) มาถอดรหัสบิตข้อมูลสองมิติ โดยอัลกอริทึมการถอดรหัส P-WBF จะทำการถอดรหัสทั้งหมด $n_1 n_2$ บิต โดยการพลิกบิตที่มีค่าความน่าเชื่อถือต่ำสุดในแต่ละการวนรอบการวนซ้ำ พร้อมทั้งมีการพิจารณาค่า s_m และ $E_{i,j}$ จากกราฟแทนเนอร์ทั้งสองแนวพร้อมๆ กัน ดังที่แสดงในหัวข้อที่ 4.2 ฉะนั้น หากบิตข้อมูลสองมิติที่ถูกถอดรหัสมีขนาดที่ใหญ่มาก อัลกอริทึมนี้จะต้องการจำนวนการวนซ้ำที่มากขึ้น เพื่อให้สมรรถนะของรหัสมีค่าต่ำ ซึ่งจะส่งผลให้ใช้เวลาแฝง (Latency) ในการถอดรหัสมีค่ามากขึ้นตามไปด้วย เพราะฉะนั้น ในหัวข้อต่อไป นี้ ผู้วิจัยจะทำการปรับปรุงอัลกอริทึมการถอดรหัส P-WBF ให้สามารถเลือกพลิกได้หลายบิตในแต่ละรอบการวนซ้ำเพื่อเพิ่มความเร็วในการถอดรหัส โดยจะทำการแบ่งเป็น 2 วิธีการ ดังนี้

4.7.1 การเลือกหลายบิตแบบคงที่สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (PF-WBF)

วิธีการนี้เป็นวิธีการอย่างง่ายสำหรับการเลือกหลายบิตมาพลิก วิธีการนี้จะทำการกำหนดจำนวนบิตที่ต้องการพลิก (N) ให้มีค่าคงที่ในแต่ละรอบการวนซ้ำ โดยที่ $N > 1$ ฉะนั้น นักวิจัยจึงเรียกอัลกอริทึมนี้ว่า “อัลกอริทึมการเลือกพลิกหลายบิตอย่างคงที่สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page fixed weighted bit-flipping: PF-WBF)” วิธีในการใช้งานอัลกอริทึมนี้คือ หลังจากคำนวณค่าความไม่น่าเชื่อถือ $E_{i,j}$ จากทุกๆ บิตแล้ว อัลกอริทึมนี้จะทำการเลือกบิตที่มีค่ามากที่สุด N อันดับ มาพลิกในแต่ละรอบการวนซ้ำ อย่างไรก็ตาม หลังจากการพลิกบิตใดๆ จะต้องทำการตรวจเช็คความถูกต้องของบิตข้อมูลสองมิติก่อนเสมอ

4.7.2 การเลือกหลายบิตแบบเทรชโฮลด์สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (PT-WBF)

วิธีการที่สองเป็นวิธีการในการเลือกบิตจำนวน N มาพลิกในแต่ละรอบการวนซ้ำคล้ายกันกับ PF-WBF แต่ค่า N นั้น จะไม่คงที่ในแต่ละรอบการวนซ้ำ ซึ่งจะมีค่าเปลี่ยนแปลงไปตามค่าเทรชโฮลด์ของค่า $E_{i,j}$ ที่ได้จากการจำลอง ผู้วิจัยจะให้ชื่อเรียกอัลกอริทึมนี้ว่า “อัลกอริทึมการเลือกหลายบิตแบบเทรชโฮลด์สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page threshold weighted bit-flipping: PT-WBF)” ค่าเทรชโฮลด์ที่ใช้สำหรับการเลือกบิตมาพลิกสำหรับอัลกอริทึม PT-WBF สามารถพิจารณาได้จากตารางที่ 4.4 (วิธีการได้มาซึ่งค่าเทรชโฮลด์ดูได้จากบทที่ 5)

ตารางที่ 4.4 ค่าเทรชโฮลด์ที่เหมาะสมสำหรับการคัดเลือกหลายบิตมาพลิกด้วยอัลกอริทึม PT-WBF สำหรับอัตรารหัส 0.5 และ 0.7

Channel	Iter = 1	Iter = 2	Iter = 3	Iter = 4	Iter = 5	Iter = 6
AWGN ($R^{2D} = 0.5$)	85	50	40	30	20	10
AWGN ($R^{2D} = 0.7$)	145	120	80	65	40	20
TDMR ($R^{2D} = 0.5$)	80	40	25	-	-	-
TDMR ($R^{2D} = 0.7$)	160	50	30	-	-	-

หมายเหตุ ค่าเทรชโฮลด์ที่แสดงในตารางที่ 4.4 เป็นค่าเทรชโฮลด์ที่ถูกนำมาใช้คนละจุดประสงค์กับค่าที่แสดงในตารางที่ 4.1-4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีในการใช้งานอัลกอริทึม PT-WBF คือ ภายหลังจากคำนวณค่าความไม่น่าเชื่อถือ $E_{i,j}$ จากทุกๆ บิตแล้ว อัลกอริทึมนี้จะทำการคัดเลือกบิตที่มีค่าความไม่น่าเชื่อถือ $E_{i,j}$ มากกว่าหรือเท่ากับเทรสโฮลด์ มาพลิกทั้งหมด โดยค่าเทรสโฮลด์จะมีค่าที่ขึ้นอยู่กับจำนวนการวนซ้ำ และอัตรารหัส จะสังเกตว่า ค่าเทรสโฮลด์สำหรับช่องสัญญาณ AWGN จะมีค่าที่อยู่ระหว่างการวนซ้ำรอบที่ 1 ถึงรอบที่ 6 เท่านั้น ซึ่งหลังจากนั้นอัลกอริทึม PT-WBF จะทำการพลิกเพียง 1 บิต ที่มีค่าความไม่น่าเชื่อถือสูงสุดเหมือนเดิม เช่นเดียวกับที่ช่องสัญญาณ TDMR (ค่าพารามิเตอร์ตาม S. Kobayashi [23-24]) ที่ค่าเทรสโฮลด์จะมีค่าอยู่ระหว่าง การวนซ้ำรอบที่ 1 ถึง 3 เท่านั้น อย่างไรก็ตาม ภายหลังจากมีการพลิกบิตใดๆ จะต้องทำการตรวจเช็คความถูกต้องของบิตข้อมูลสองมิติก่อนเสมอ

อันเนื่องมาจาก อัลกอริทึมการถอดรหัส PF-WBF และ PT-WBF จะมีการคำนวณค่าความไม่น่าเชื่อถือ $E_{i,j}$ ทั้งหมด $n_1 n_2$ บิต ต่อหนึ่งรอบการวนซ้ำ เหมือนกันกับอัลกอริทึม P-WBF แม้ว่าจะมีการพลิกบิตที่มากกว่าก็ตาม ดังนั้น จำนวนวงจรและบวกของอัลกอริทึม PF-WBF และ PT-WBF จึงยังคงมีค่าเท่ากับอัลกอริทึม P-WBF ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 ความซับซ้อนของการถอดรหัส สำหรับการวนซ้ำ 1 รอบ ของอัลกอริทึม PF-WBF และ PT-WBF

ชนิดอัลกอริทึม	จำนวนวงจรคูณ	จำนวนวงจรหาร	จำนวนวงจรถูก
P-WBF (นำเสนอ1)	0	$n_1 n_2$	$2n_1 n_2 d_v d_c$
PF-WBF (นำเสนอ4)	0	$n_1 n_2$	$2n_1 n_2 d_v d_c$
PT-WBF (นำเสนอ5)	0	$n_1 n_2$	$2n_1 n_2 d_v d_c$

บทที่ 5

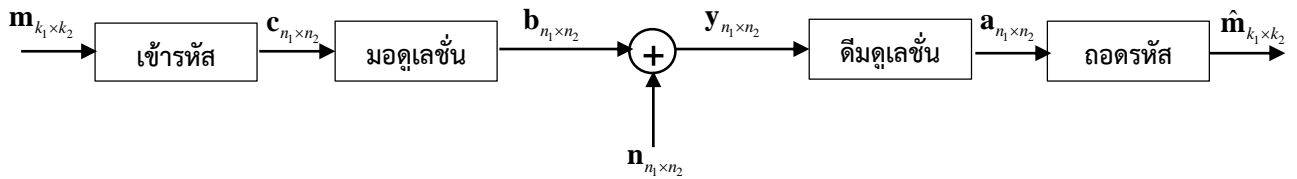
การทดสอบสมรรถนะของระบบ

ในบทที่ 5 นี้จะกล่าวถึง การจำลองสมรรถนะ BER เพื่อเปรียบเทียบสมรรถนะในการถอดรหัสระหว่างอัลกอริทึมที่ได้ทำนำเสนอในบทที่ 4 ได้แก่ อัลกอริทึมการถอดรหัสแบบบิตพลิกบิตต่งขวาง น้ำหนักแบบเพจ (Page-based weighted bit-flipping : P-WBF), แบบแถวสลับหลัก (Row-column weighted bit-flipping : RC-WBF) และแบบดัดแปลงแถวสลับหลัก (Modified row-column weighted bit-flipping : MRC-WBF) อย่างไรก็ตาม ผู้วิจัยจะแสดงสมรรถนะของรหัสแอลดีพีซีแบบโปรดักต์ที่มีการถอดรหัสด้วยอัลกอริทึมซิมโพรดักต์ (SPA) หรือการถอดรหัสแบบซอฟต์แวร์ [9] ร่วมด้วย ผู้วิจัยจะเรียกอัลกอริทึมข้างต้นว่า “อัลกอริทึมแอลดีพีซีแบบโปรดักต์ด้วยการแพร่กระจายความเชื่อมั่น (product LDPC code with belief propagation: PCB)” ทั้งนี้ อัลกอริทึมทั้ง 4 ข้างต้น จะถูกจำลองบนช่องสัญญาณรบกวนเกาส์สีขาว และบนระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR) ที่มีการแทรกสอดระหว่างสัญลักษณ์ (Inter-symbol interference: ISI) และแทรกสอดระหว่างแทร็ค (Inter-track interference: ITI) สำหรับการทดสอบสมรรถนะของทุกๆ อัลกอริทึม นั้น เมทริกซ์พาริตีที่เช็คที่ใช้เข้ารหัสในแนวแถว และหลัก จะถูกออกแบบด้วยอัลกอริทึมขยายเส้นเชื่อมแบบก้าวหน้า (PEG) [26] โดยใช้รหัสแอลดีพีซีแบบคงที่ d_v เท่ากับ 3 ผู้วิจัยสามารถแบ่งส่วนของการจำลองออกเป็น 4 ส่วนหลักๆ ดังนี้

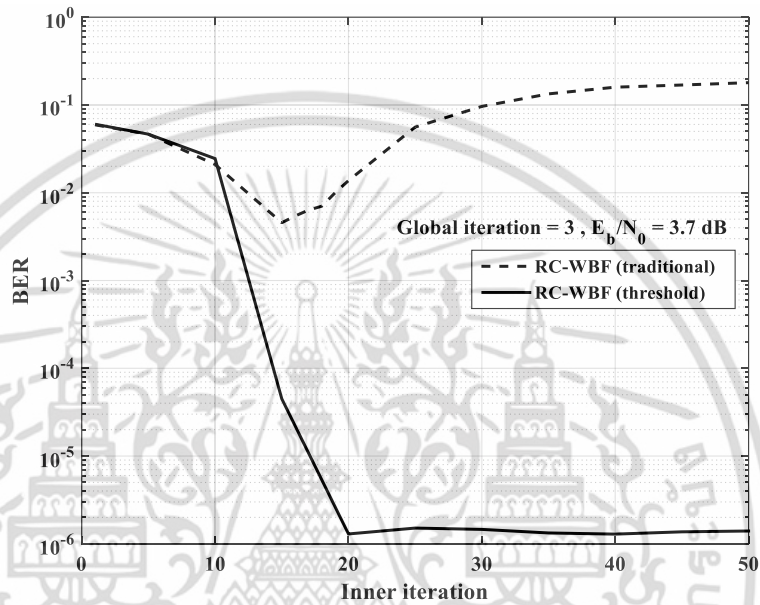
5.1 ผลการปรับแก้สมรรถนะไม่ลู่เข้า (V-shape) ที่การวนซ้ำรอบสูง

เป็นที่ทราบกันดีว่า รหัสแอลดีพีซีที่มีการถอดรหัสแบบวนซ้ำ ซึ่งความผิดพลาดของบิตข้อมูลในแต่ละบล็อก จะถูกแก้ไขให้มีความถูกต้องมากขึ้น เมื่อมีการวนซ้ำในรอบที่สูงขึ้น ประสิทธิภาพในการแก้ไขความผิดพลาดของบิตข้อมูลจะลู่เข้า (Convergent) เมื่อมีการวนซ้ำในรอบที่สูงเพียงพอ ในทางตรงกันข้าม อัลกอริทึม RC-WBF และ MRC-WBF ที่ถูกนำเสนอในบทที่ 4 จะมีความขัดแย้งกับทฤษฎีของการถอดรหัสแอลดีพีซี ที่ซึ่งประสิทธิภาพในการแก้ไขความผิดพลาดของข้อมูล สำหรับสองอัลกอริทึมดังกล่าว จะดีขึ้นในช่วงแรกๆ จนกระทั่งถึงจุดต่ำสุด จากนั้น สมรรถนะของมันจะแย่ลงเมื่อมีการวนซ้ำในรอบที่สูงขึ้น ปัญหาดังกล่าว จะมีผลทำให้สมรรถนะของทั้งสองอัลกอริทึม มีลักษณะไม่ลู่เข้า ที่การวนซ้ำรอบสูง หรือที่เรียกว่าปัญหา V-shape อย่างไรก็ตาม สำหรับปัญหาดังกล่าว ผู้วิจัยได้ทำการแก้ไขวิธีการถอดรหัสของทั้งสองอัลกอริทึม ด้วยวิธีการกำหนดขอบเขตการพลิกบิตด้วยค่าเทรสโฮลด์ที่เหมาะสม ดังแสดงในหัวข้อที่ 4.5 โดยบิตข้อมูล $\mathbf{m}_{k_1 \times k_2}$ และคำรหัส $\mathbf{c}_{n_1 \times n_2}$ จะถูกกำหนดให้มีขนาดเท่ากับ 350×350 บิต และ 500×500 ตามลำดับ ดังนั้น อัตรารหัส R^{2D} จะมีค่าเท่ากับ 0.5 กำหนดการวนซ้ำภายนอก (global iteration) ของอัลกอริทึม RC-WBF และ MRC-WBF ให้มีค่าเท่ากับ 3 และ 6 ตามลำดับ ผู้วิจัยเลือกใช้ช่องสัญญาณในอุดมคติหรือช่องสัญญาณเกาส์สีขาว (สามารถดูอย่างละเอียดได้ในหัวข้อที่ 2.3 หรือ 5.2) ในการจำลองเพื่อดูสมรรถนะภายหลังการแก้ไขปัญหาดังกล่าวของทั้งสองอัลกอริทึม ดังแสดงในรูปที่ 5.1

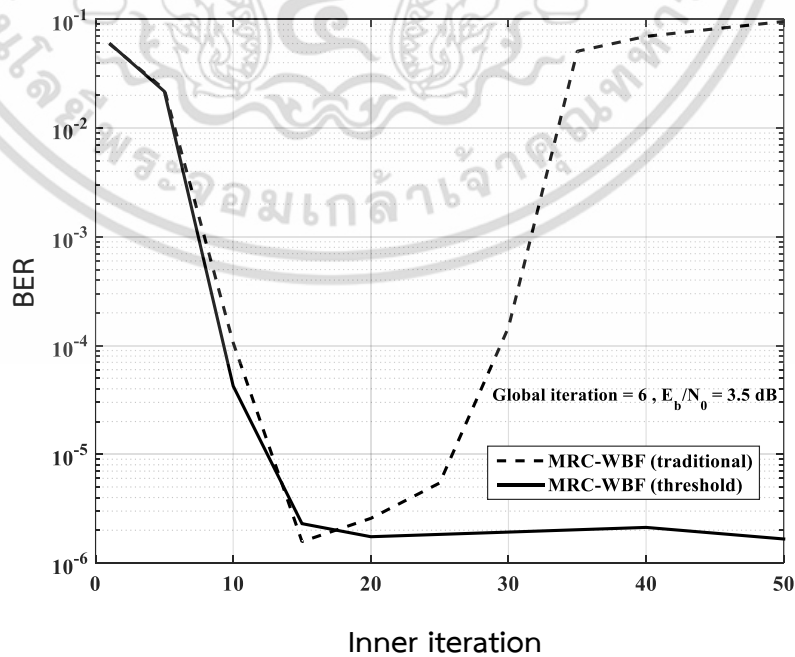
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แบบจำลองช่องสัญญาณอุดมคติ



รูปที่ 5.2 สมรรถนะของ RC-WBF หลังจากถูกแก้ไขด้วยค่าเทรชโฮลด์ (Threshold)

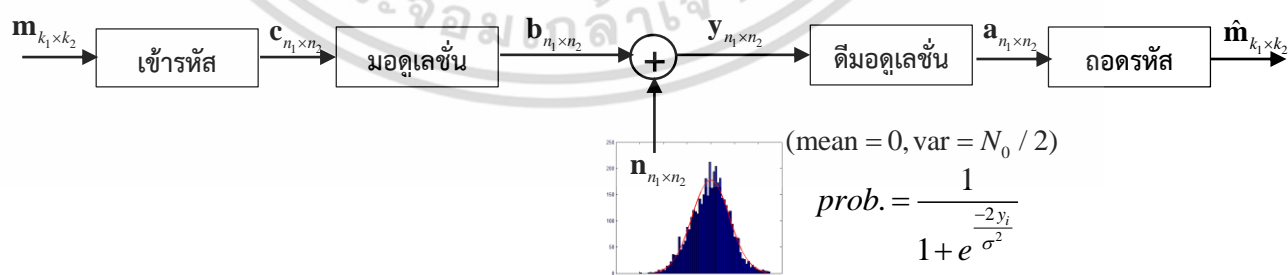


เอกสารนี้เป็นเอกสารที่รูปที่ 5.3 สมรรถนะของ MRC-WBF หลังจากถูกแก้ไขด้วยค่าเทรชโฮลด์(Threshold) ราคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการใช้ค่าเทรสโฮลด์ (ตารางที่ 4.1) เป็นตัวกำหนดขอบเขตในการฟลิปบิต สำหรับอัลกอริทึม RC-WBF ดังแสดงในรูปที่ 5.2 พบว่าสามารถแก้ไขปัญหาสมรรถนะไม่ลู่เข้าที่เกิดขึ้นได้ โดยสมรรถนะ BER ของอัลกอริทึม RC-WBF จะลู่เข้าเมื่อการวนซ้ำภายใน (inner iteration) มีค่ามากกว่าหรือเท่ากับ 20 ยิ่งกว่านั้น การใช้เทรสโฮลด์เป็นตัวกำหนดขอบเขตในการฟลิปบิต สามารถที่จะปรับปรุงสมรรถนะ BER ของอัลกอริทึม RC-WBF (ก่อนการปรับแก้) ให้ดีขึ้นถึง 3 เท่า กล่าวคือ เมื่ออัลกอริทึม RC-WBF แบบดั้งเดิม ถูกใช้ที่การวนซ้ำภายในเท่ากับ 20 รอบ ($E_b/N_0 = 3.7$ dB) จะมีความผิดพลาดเท่ากับ 1 บิตจาก 100 บิตข้อมูล แต่เมื่อนำค่าเทรสโฮลด์มาเป็นตัวกำหนดขอบเขตในการฟลิปบิต ความผิดพลาดจะมีค่าเท่ากับ 1 บิต จาก 1 ล้านบิตข้อมูล เช่นเดียวกันกับอัลกอริทึม MRC-WBF ซึ่งเมื่อมีการแก้ไขด้วยวิธีการกำหนดขอบเขตในการฟลิปบิตด้วยค่าเทรสโฮลด์ จะช่วยให้ปัญหาสมรรถนะไม่ลู่เข้าได้รับการปรับแก้เช่นเดียวกัน จากรูปที่ 5.2 สมรรถนะ BER ของอัลกอริทึม MRC-WBF จะเริ่มลู่เข้าเมื่อการวนซ้ำภายใน (inner iteration) มีค่าเท่ากับ 15 ส่วนสมรรถนะหลังจากใช้ค่าเทรสโฮลด์เป็นตัวกำหนดขอบเขตในการฟลิปบิตนั้น ยังคงมีค่าเทียบเคียงกับกับสมรรถนะดั้งเดิม

5.2 การจำลองระบบที่ช่องสัญญาณรบกวนเกาส์สีขาวแบบบวก

โดยปกติแล้วสัญญาณรบกวนเกาส์สีขาวแบบบวก สามารถพบเห็นได้ทั่วไป ในระบบสื่อสารอนาล็อกและดิจิทัล สัญญาณรบกวนชนิดนี้มีการกระจายตัวแบบเกาส์เซียนหรือแบบสุ่ม ซึ่งไม่สามารถทำนายระดับแอมพลิจูดของสัญญาณรบกวนชนิดนี้ล่วงหน้าได้ ในวิทยานิพนธ์เล่มนี้ จะเน้นไปที่ระบบการบันทึกข้อมูลเชิงแม่เหล็ก ซึ่งสัญญาณรบกวนชนิดนี้ จะเกิดขึ้นจากร้อนของวงจรรอิเล็กทรอนิกส์ ซึ่งจะส่งผลให้บิตข้อมูลที่ถูกอ่านออกมาจากสื่อบันทึกมีความผิดพลาดเกิดขึ้นแบบจำลองสัญญาณรบกวนเกาส์สีขาว ที่ใช้ในการจำลองสมรรถนะของแต่ละอัลกอริทึม สามารถพิจารณาได้จากรูปที่ 5.4 ทั้งนี้ทั้ง 3 อัลกอริทึมที่น่าเสนอ อันได้แก่ P-WBF, RC-WBF และ MRC-WBF รวมถึงอัลกอริทึม PCB ล้วนเป็นรหัสแอลดีพีซีที่ถูกพิจารณาแบบสองมิติทั้งสิ้น ดังนั้น ในส่วนของวงจรรหัส และถอดรหัสของแต่ละอัลกอริทึมนั้น จะมีโครงสร้างอย่างละเอียดตามที่ได้กล่าวถึงในบทที่ 4.1 4.2 4.3 และ 4.4 ตามลำดับ



รูปที่ 5.4 แบบจำลองระบบที่มีการรบกวนจากสัญญาณรบกวนเกาส์สีขาวแบบบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากบิตข้อมูลสองมิติ $\mathbf{m}_{k_1 \times k_2}$ ถูกเข้ารหัสแอสติฟซี จนเกิดเป็นคำรหัส $\mathbf{c}_{n_1 \times n_2}$ จากนั้นคำรหัสที่ได้ จะถูกนำไปมอดูเลตแบบบีฟีสเค (binary phase shift keying: BPSK) เป็นระดับสัญญาณ $\mathbf{b}_{n_1 \times n_2}$ ที่พร้อมสำหรับการส่งออกไปยังเป้าหมาย โดย $\mathbf{b}_{n_1 \times n_2}$ จะถูกส่งผ่านช่องสัญญาณที่มีการรบกวนจากสัญญาณรบกวนแบบเกาส์สีขาว $\mathbf{n}_{n_1 \times n_2}$ ซึ่งที่ภาครับนั้น สัญญาณที่ได้รับ $\mathbf{y}_{n_1 \times n_2}$ จะได้มาจากแอมพลิจูดของ $\mathbf{b}_{n_1 \times n_2}$ รวมกับแอมพลิจูดของสัญญาณรบกวนแบบเกาส์สีขาว $\mathbf{n}_{n_1 \times n_2}$ ที่มีความหนาแน่นสเปกตรัมกำลังงานเท่ากับ $N_0/2$ W/Hz สัญญาณที่ได้รับ $\mathbf{y}_{n_1 \times n_2}$ จะถูกดีมอดูเลตเป็น $\mathbf{a}_{n_1 \times n_2}$ และถูกนำไปถอดรหัสแอสติฟซีเพื่อกู้บิตข้อมูล $\hat{\mathbf{m}}_{k_1 \times k_2}$ กลับคืนมา ในบทที่ 5 นี้ จะนำเอาบิตข้อมูลหลังจากวงจรถอดรหัส $\hat{\mathbf{m}}_{k_1 \times k_2}$ มาคำนวณหาอัตราส่วนความผิดพลาดบิต (BER) โดยที่อัตราส่วนพลังงานของคำรหัสต่อสัญญาณรบกวนแบบเกาส์สีขาวหรือค่าเอสเอนอาร์ในหน่วยเดซิเบล (dB) นั้นสามารถคำนวณได้จากสมการที่ 5.1

$$(E_b / N_0)_{dB} = 10 \log_{10} \left(\frac{E_c}{R^{2D} N_0} \right) \quad (5.1)$$

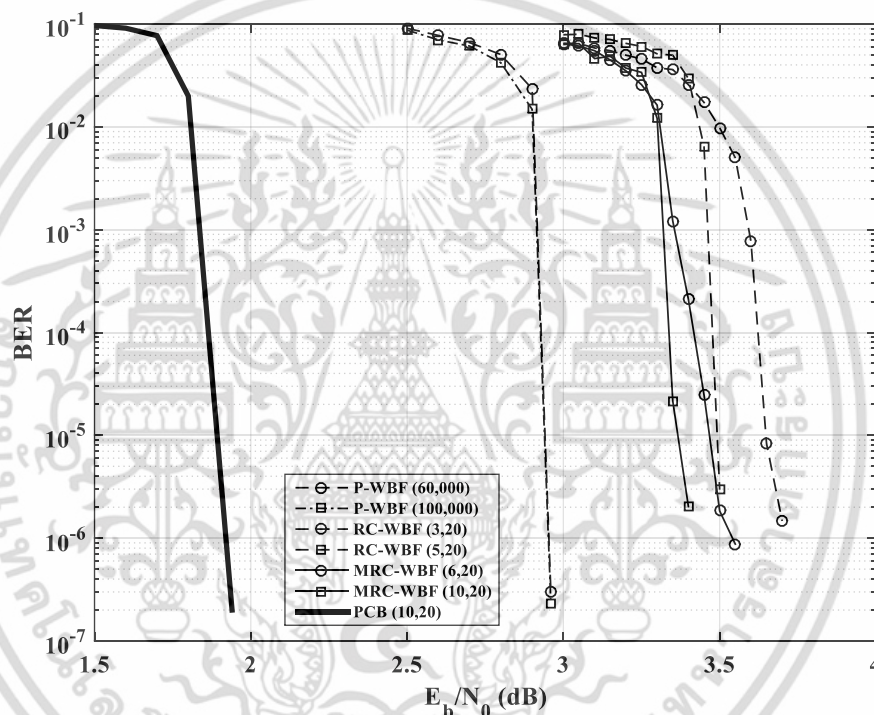
เมื่อ E_c , E_b และ R^{2D} คือ พลังงานเฉลี่ยของคำรหัส, พลังงานเฉลี่ยของบิตข้อมูล และอัตราหัสแบบ 2 มิติตามลำดับ

5.2.1 สมรรถนะของรหัสแอสติฟซีที่อัตราหัสปานกลาง

ในส่วนนี้จะทำการจำลองสมรรถนะของรหัสแอสติฟซี เพื่อเปรียบเทียบกันระหว่างทั้งสามอัลกอริทึมที่นำเสนอ อันได้แก่ อัลกอริทึมการถอดรหัสแบบบิตพลิกบิงซ์ชนิดถ่วงน้ำแบบชนิดเพจ (P-WBF), แบบแถวสลับหลัก (RC-WBF) และแบบตัดแปลงแถวสลับหลัก (MRC-WBF) โดยใช้อัตราหัสแบบสองมิติเท่ากับ 0.5 อย่างไรก็ตาม สมรรถนะของอัลกอริทึมโปรดักส์แอสติฟซี ที่มีการถอดรหัสแบบซอฟต์แวร์ (PCB) จะถูกแสดงเป็นเส้นอ้างอิง (ซึ่งถูกคาดหวังว่าจะมีสมรรถนะที่ดีที่สุดอยู่แล้ว) เพื่อความยุติธรรมแก่การจำลองสมรรถนะ BER ของทั้งสามอัลกอริทึมที่นำเสนอ ฉะนั้น จำนวนบิตที่ถูกพลิกของแต่ละอัลกอริทึม จะถูกกำหนดให้มีค่าที่เท่ากัน ซึ่งสามารถนำจำนวนบิตที่ต้องการพลิกทั้งหมด มาคำนวณหาการวนรอบที่ยุติธรรม ได้จากสมการที่ 4.9 - 4.11 ในทางตรงกันข้าม สำหรับอัลกอริทึม PCB เป็นการยากที่จะปรับให้ยุติธรรมสำหรับทั้งสามอัลกอริทึมข้างต้น เนื่องจากอัลกอริทึม PCB เป็นการถอดรหัสแอสติฟซีแบบซอฟต์แวร์ ที่มีการนิยามจำนวนรอบของการวนซ้ำแตกต่างจากการถอดรหัสแบบหยาบที่นำเสนอ ดังนั้น ผู้วิจัย จะกำหนดให้จำนวนรอบการวนซ้ำของอัลกอริทึม PCB มีค่าที่ทำให้สมรรถนะของมันลู่เข้า การจำลองในหัวข้อ 5.2.1 นี้จะใช้ค่าพารามิเตอร์ในการออกแบบรหัสแต่ละชนิด ดังแสดงในตารางที่ 5.1

ตารางที่ 5.1 พารามิเตอร์ในการสร้างรหัสสำหรับอัตรารหัสปานกลาง

รหัส	แนวแถว				แนวหลัก				R^{2D}	การวนซ้ำแบบ ยูติธรรม	
	N	N-K	R	dv	N	N-K	R	dv		รอบต่ำ	รอบสูง
P-WBF	500	350	0.7	3	500	350	0.7	3	0.5	60,000	100,000
RC-WBF	500	350	0.7	3	500	350	0.7	3	0.5	(3,20)	(5,20)
MRC-WBF	500	350	0.7	3	500	350	0.7	3	0.5	(6,20)	(10,20)
PCB	500	350	0.7	3	500	350	0.7	3	0.5	(5,20) *convergent	

รูปที่ 5.5 สมรรถนะของ P-WBF, RC-WBF, MRC-WBF และ PCB สำหรับ R^{2D} เท่ากับ 0.5

จากรูปที่ 5.5 นั้น เป็นการเปรียบเทียบอัตราผิดพลาด (BER) ของรหัสทั้ง 3 ชนิด ที่นำเสนอในช่วงของอัตราส่วนพลังงานของคาร์รหัสต่อสัญญาณรบกวนเกาส์สีขาวหรือค่าเอสเอ็นอาร์ระหว่าง 1.5– 4 dB ซึ่งพบว่าอัลกอริทึม P-WBF จะให้ค่าเกนรหัส (Coding gain) เท่ากับ 0.74 dB และ 0.59 dB เมื่อเทียบกับอัลกอริทึม RC-WBF และ MRC-WBF ที่การวนซ้ำรอบต่ำ สำหรับอัตราความผิดพลาดอยู่ที่ 1×10^{-6} ตามลำดับ ส่วนการวนซ้ำในรอบสูงพบว่า อัลกอริทึมการถอดรหัสแบบ P-WBF จะยังคงให้สมรรถนะเท่าเดิม ส่วนอัลกอริทึม RC-WBF และ MRC-WBF จะให้สมรรถนะที่ดีขึ้นกว่าเดิมเล็กน้อย เมื่อเทียบกับการวนซ้ำในรอบต่ำ อย่างไรก็ตามอัลกอริทึม P-WBF ที่การวนซ้ำรอบสูง จะยังคงมีสมรรถนะที่ดีกว่า RC-WBF และ MRC-WBF โดยจะให้ค่าเกนรหัสเท่ากับ 0.54 dB และ 0.44 dB เมื่อเทียบกับอัลกอริทึม RC-WBF และ MRC-WBF ที่อัตราความผิดพลาดอยู่ที่ 1×10^{-6}

ตามลำดับ แต่เมื่อหากนำสมรรถนะของอัลกอริทึมที่นำเสนอทั้งสาม มาเปรียบเทียบกับอัลกอริทึม PCB ที่เป็นการถอดรหัสด้วยวิธีการแพร่กระจายความเชื่อมั่น (BP) ด้วยอัลกอริทึมซมโพรดักส์ (SPA) หรือการถอดรหัสแบบซอฟต์แวร์ ที่สมรรถนะสูง ผลจะเป็นตามที่คาดหมายคือ สมรรถนะของอัลกอริทึม PCB ควรจะดีที่สุด แต่เป้าหมายหลักของวิทยานิพนธ์นี้คือ สนใจไปที่อัลกอริทึมแบบหลาย (ดูรายละเอียดเพิ่มเติมในหัวข้อ 4.6)

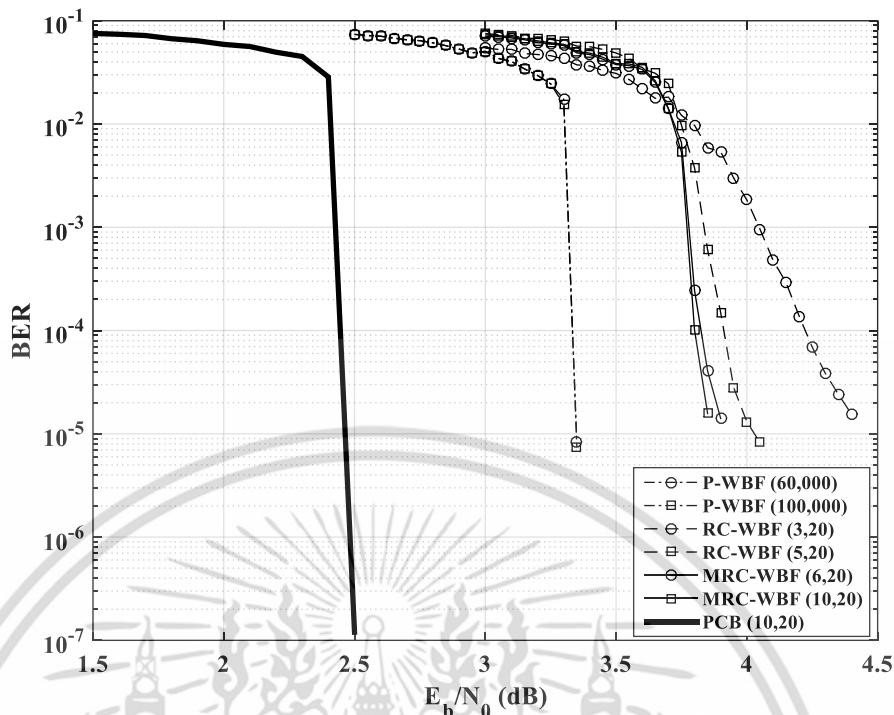
5.2.2 สมรรถนะของรหัสแอลดีพีซีที่อัตราการสูง

ในหัวข้อนี้ จะกล่าวถึงอัลกอริทึมการถอดรหัสทั้ง 3 ชนิด อันได้แก่ อัลกอริทึมการถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบชนิดเพจ (P-WBF), แบบแถวสลับหลัก (RC-WBF) และแบบตัดแปลงแถวสลับหลัก (MRC-WBF) ที่ซึ่งรหัสทั้งหมดนั้น ถูกเข้ารหัสแอลดีพีซีด้วยเมทริกซ์พาริตีที่ซ้ำกัน ที่มีจำนวนแถวน้อยลง (จำนวนโนดเซ็นน้อยลง) ซึ่งจะทำให้เกิดคำรหัสที่มีจำนวนบิตพาริตีลดลงหรือมีอัตราการสูงที่สูงขึ้น ซึ่งอัตราการสูงนี้ จะถูกใช้กันอย่างแพร่หลายในระบบสื่อสารดิจิทัลโดยทั่วไป เนื่องจากจะให้ อัตราเร็วในการส่งผ่านข้อมูล (Throughput) ที่มีค่าสูง อย่างไรก็ตาม สมรรถนะของอัลกอริทึมโพรดักส์แอลดีพีซี ที่มีการถอดรหัสแบบซอฟต์แวร์ (PCB) จะถูกแสดงเป็นเส้นอ้างอิง (ซึ่งถูกคาดหวังว่าจะมีสมรรถนะที่ดีที่สุดอยู่แล้ว) การจำลองในหัวข้อ 5.2.2 จะใช้ค่าพารามิเตอร์ในการออกแบบรหัสแต่ละชนิด ดังแสดงในตารางที่ 5.2

ตารางที่ 5.2 พารามิเตอร์ในการสร้างรหัสสำหรับอัตราการสูง

รหัส	แนวแถว				แนวหลัก				R ^{2D}	การวนซ้ำแบบยุติธรรม	
	N	N-K	R	dv	N	N-K	R	dv		รอบต่ำ	รอบสูง
P-WBF	500	425	0.85	3	500	425	0.85	3	0.7	60,000	100,000
RC-WBF	500	425	0.85	3	500	425	0.85	3	0.7	(3,20)	(5,20)
MRC-WBF	500	425	0.85	3	500	425	0.85	3	0.7	(6,20)	(10,20)
PCB	500	425	0.85	3	500	425	0.85	3	0.7	(5,20) *convergent	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



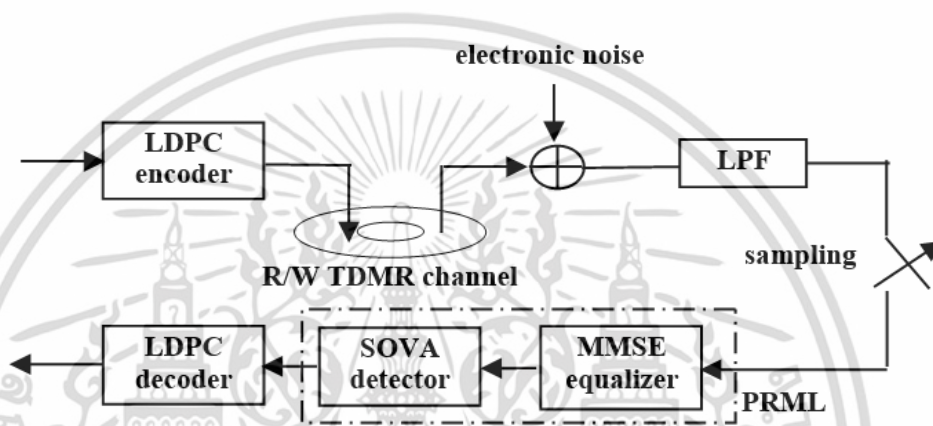
รูปที่ 5.6 สมรรถนะของ P-WBF, RC-WBF, MRC-WBF และ PCB สำหรับ R^{2D} เท่ากับ 0.7

จากรูปที่ 5.6 แนวโน้มสมรรถนะของอัลกอริทึมการถอดรหัสทั้งหมด มีความใกล้เคียงกันกับที่ อัตรารหัสกลาง โดยที่อัลกอริทึม PCB หรือการถอดรหัสแบบซอฟต์แวร์จะยังคงให้สมรรถนะที่ดีที่สุด (เป็นไปตามที่คาดไว้) แต่หากเปรียบเทียบกันเฉพาะอัลกอริทึมที่นำเสนอทั้ง 3 อันได้แก่ P-WBF, RC-WBF และ MRC-WBF อัลกอริทึม P-WBF จะให้ค่าเกณฑ์ส่เท่ากับ 1.1 dB และ 0.5 dB เมื่อเปรียบเทียบกับ RC-WBF และ MRC-WBF ที่การวนซ้ำรอบต่ำ สำหรับอัตราความผิดพลาดอยู่ที่ 1×10^{-5} ตามลำดับ เช่นเดียวกันกับการวนซ้ำรอบสูง อัลกอริทึม P-WBF จะให้ค่าเกณฑ์ส่เท่ากับ 0.6 dB และ 0.45 dB เมื่อเปรียบเทียบกับ RC-WBF และ MRC-WBF ที่อัตราความผิดพลาดอยู่ที่ 1×10^{-5} ตามลำดับ

5.3 สมรรถนะรหัสแอลดีพีซีบนระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR)

เป็นที่ทราบกันดีว่า รหัสแอลดีพีซีเป็นรหัสที่มีประสิทธิภาพสูงในการแก้ไขความผิดพลาดของข้อมูล ดังนั้น รหัสแอลดีพีซีจึงถูกนำมาใช้กันอย่างแพร่หลายในหลายๆ แอปพลิเคชันในระบบดิจิทัล อาทิเช่น ไวไฟ (Wi-Fi), ดิจิตอลทีวี (DVB-T2) และฮาร์ดดิสก์ไดรฟ์ก็เป็นอีกหนึ่งแอปพลิเคชันที่รหัสแอลดีพีซีถูกนำมาประยุกต์ใช้กันอย่างแพร่หลายในปัจจุบัน แบบจำลองในการจำลองสมรรถนะรหัสแอลดีซีบนช่องสัญญาณของฮาร์ดดิสก์ไดรฟ์นั้นถูกแสดงในรูปที่ 5.7 เมื่อสัญญาณอ่านกลับถูกอ่านขึ้นมาจากแผ่นมีเดีย (ถูกออกแบบด้วยเทคโนโลยี TDMR [23-24]) แล้วนั้น แอมพลิจูดของสัญญาณอ่านกลับจะถูกรบกวนด้วยความร้อนจากวงจรอิเล็กทรอนิกส์ (สัญญาณรบกวนเกาส์เซียน) โดยวงจรกรองความถี่ต่ำ (LPF) จะทำหน้าที่ในการกรององค์ประกอบนอกความถี่ที่ต้องการของสัญญาณอ่านเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับทั้งไป หลังจากนั้นสัญญาณอ่านกลับจะถูกนำไปซีกตัวอย่าง (Sampling) เพื่อทำการแปลงสัญญาณอะนาล็อกให้เป็นสัญญาณดิจิทัล โดยสัญญาณดิจิทัลที่ได้นั้นจะยังคงประกอบไปด้วยการลดทอนจากการแทรกสอดระหว่างสัญลักษณ์ (ISI) และการแทรกสอดระหว่างแตรีก (ITI) ข้างเคียงจากการอ่านของหัวอ่าน ซึ่งองค์ประกอบดังกล่าว จะถูกกำจัดด้วยวงจรตอบสนองบางส่วนความจะเป็นสูงสุด (PRML) ซึ่งประกอบไปด้วยวงจรอ็อลกอริทึมที่ถูกรอกแบบด้วยวิธีการ MMSE [31] และวงจรตรวจจับ (Detector) ที่ถูกรอกแบบด้วยอัลกอริทึมซอฟต์แวร์เอาท์พุทเทอร์บีอ็อลกอริทึม (SOVA) [32] หลังจากนั้นสัญญาณเอาท์พุทจากวงจรตรวจจับจะถูกนำมาถอดรหัสด้วยวงจรถอดรหัสแอลดีพีซีเพื่อนำบิตข้อมูลที่ถูกรับที่กลับคืนมา (ในส่วนของความหมายเชิงลึกและรายละเอียดเพิ่มเติมของช่องสัญญาณ TDMR สามารถดูได้จากบทที่ 2)



รูปที่ 5.7 แบบจำลองภาคส่ง และภาครับของช่องสัญญาณระบบบันทึกข้อมูลเชิงแม่เหล็ก
สองมิติ

อย่างไรก็ตามในวิทยานิพนธ์เล่มนี้ ได้นำเสนออัลกอริทึมการถอดรหัสแอลดีพีซี 3 ชนิด อันได้แก่ อัลกอริทึมการถอดรหัสแบบบิตฟลิปปีงชนิดถ่วงน้ำหนักแบบเพจ (P-WBF), แบบแถวสลับหลัก (RC-WBF) และแบบตัดแปลงแถวสลับหลัก (MRC-WBF) รวมถึงรหัสแอลดีพีซีแบบโปรดักส์ [9] ซึ่งถูกแสดงเป็นเส้นอ้างอิง ล้วนแต่เป็นการถอดรหัสแอลดีพีซีในรูปแบบสองมิติทั้งสิ้น ดังนั้น ภายในส่วนของวงจรเข้า และถอดรหัสของแต่ละอัลกอริทึมในแบบจำลองรูป 5.7 จะมีโครงสร้างอย่างละเอียดตามที่ได้กล่าวถึงในหัวข้อที่ 4.1 ถึง 4.4 ตามลำดับ อนึ่ง เทคโนโลยี TDMR ที่ซึ่งสมการความอ่อนไหวในการอ่าน และสัญญาณอ่านกลับ ถูกพิจารณาในรูปแบบสองมิติ จึงเหมาะสมแก่การนำมาใช้เป็นต้นแบบในการจำลองสมรรถนะของอัลกอริทึมทั้ง 4 ข้างต้น อย่างไรก็ตาม ค่าพารามิเตอร์ของหัวอ่าน และโครงสร้างของการบันทึกข้อมูลแต่ละบิตนั้นสำหรับเทคโนโลยี TDMR จะใช้ค่าตามงานวิจัยของ S. Kobayashi [23-24] ซึ่งแสดงในตารางที่ 5.3 รวมถึงค่าพารามิเตอร์สำหรับการสร้างรหัสของทั้งสี่อัลกอริทึม จะถูกแสดงบนตารางที่ 5.4 ด้วยเช่นกัน

ตารางที่ 5.3 พารามิเตอร์ในการออกแบบช่องสัญญาณฮาร์ดดิสก์ไทรฟ์แบบ 2 มิติ ตามแบบจำลองของ S. Kobayashi [23-24]

โครงสร้างบิตข้อมูล (1 บิต)		โครงสร้างหัวอ่าน (Write head)	
ความยาวตามแตรีก (T_p)	22 nm	ความกว้างของ side shields gap (H)	26 nm.
ความยาวตั้งฉากกับแตรีก (L_b)	16.5 nm	ความกว้างของ side gap (G)	15 nm.
ความหนาแน่นเชิงพื้นที่ (AD)	5.62 Tb/in ²	ความยาวของ pole ในแนวตั้งฉากกับแตรีก (T)	13 nm.
		ความยาวของ pole ในแนวตามแตรีก (U)	2 nm.
		ความหนาของชั้น magnetic spacing (h_m)	4 nm.
		ความหนาของชั้น recording layer (t_m)	10 nm.
		ความหนาของชั้น intermediate layer (t_s)	1 nm.

ตารางที่ 5.4 พารามิเตอร์ในการสร้างรหัสสำหรับอัตราหัสสูงสำหรับช่องสัญญาณฮาร์ดดิสก์ไทรฟ์แบบ 2 มิติ

รหัส	แนวแถว				แนวหลัก				R^{2D}	การวนซ้ำแบบยูติธรรม	
	N	N-K	R	dv	N	N-K	R	dv		รอบต่ำ	รอบสูง
P-WBF	500	425	0.85	3	500	425	0.85	3	0.7	60,000	100,000
RC-WBF	500	425	0.85	3	500	425	0.85	3	0.7	(3,20)	(5,20)
MRC-WBF	500	425	0.85	3	500	425	0.85	3	0.7	(6,20)	(10,20)
PCB	500	425	0.85	3	500	425	0.85	3	0.7	(5,20) *convergent	

Using MMSE & SOVA : GPR Target , K = 3, N = 5

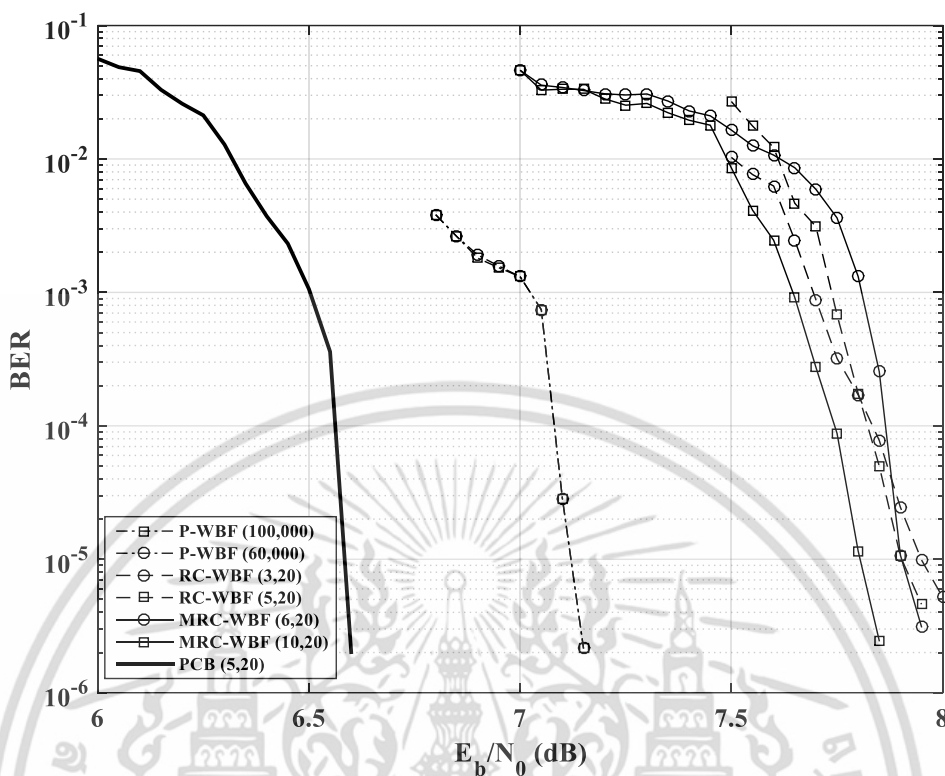
หมายเหตุ การจำลองในส่วนนี้ ใช้สัมประสิทธิ์อิกวอไลเซอร์ความยาวเท่า 11 และใช้ทาร์เก็ตจีพีอาร์สำหรับวงจรตรวจหาอย่างละเอียด (SOVA) ความยาวเท่ากับ 3

อัตราส่วนพลังงานของค้ำรหัสต่อสัญญาณรบกวนหรือค่าเอสเอนอาร์ในหน่วยเดซิเบล (dB) สำหรับระบบที่มีการแทรกสอดสัญลักษณ์บนช่องสัญญาณการบันทึกข้อมูลแม่เหล็กสองมิติ (TDMR) ถูกแสดงในสมการที่ 5.2

$$(E_b / N_0)_{dB} = 10 \log_{10} \left(\frac{\sum_{l=0}^L h_l}{R^{2D} \sigma^2} \right) \quad (5.2)$$

เมื่อ h_l , σ^2 และ R^{2D} คือ สัมประสิทธิ์ของทาร์เก็ตความยาว $L+1$, ความแปรปรวนของสัญญาณรบกวน และอัตราหัสแบบ 2 มิติ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

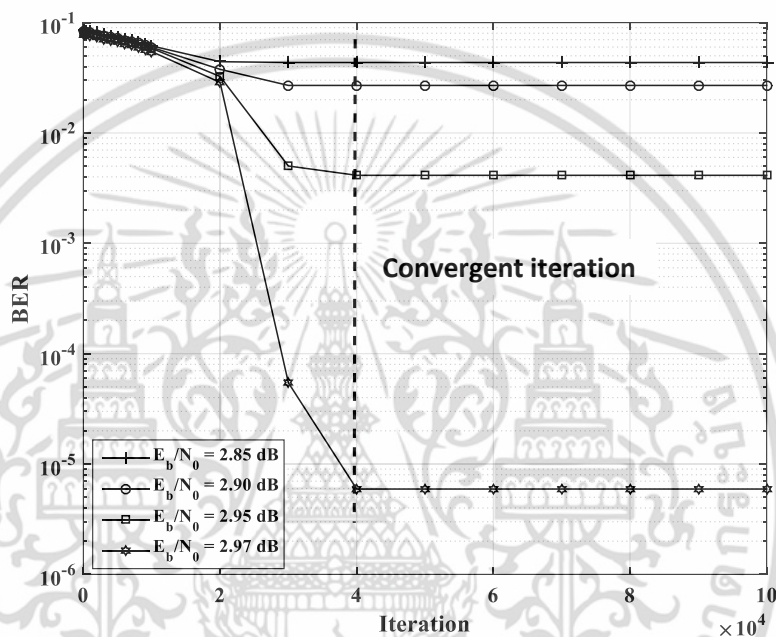


รูปที่ 5.8 สมรรถนะของอัลกอริทึม P-WBF, RC-WBF, MRC-WBF และ PCB บนช่องสัญญาณการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR)

ในหัวข้อต่อไปนี้จะทำการจำลองสมรรถนะของอัลกอริทึมทั้ง 4 บนช่องสัญญาณของการบันทึกแม่เหล็กสองมิติที่มีการแทรกสอดแบบสัญญาณลักษณะ (TDMR) โดยใช้พารามิเตอร์ตาม S. Kobayashi [23-24] ที่อัตราหัสสูง พิจารณาที่การวนซ้ำรอบต่ำพบว่า การถอดรหัสแวลดีฟี่ซีด้วยใช้อัลกอริทึม P-WBF จะให้สมรรถนะที่ดีกว่าอัลกอริทึม RC-WBF และ MRC-WBF ซึ่งจะให้เกินของรหัสเท่ากับ 0.75 dB และ 0.7 dB เมื่อเปรียบเทียบกับอัลกอริทึม RC-WBF และ MRC-WBF ตามลำดับ เช่นเดียวกันกับการวนซ้ำรอบสูง จะพบว่าการถอดรหัสแวลดีฟี่ซีด้วยใช้อัลกอริทึม P-WBF จะยังคงให้สมรรถนะที่ดีกว่า อัลกอริทึม RC-WBF และ MRC-WBF ด้วยเช่นกัน โดยจะให้เกินของรหัสเท่ากับ 1.7 dB และ 0.6 dB ที่อัตราผิดพลาดเท่ากับ 1×10^{-5} ตามลำดับ ดังแสดงในรูปที่ 5.8 อย่างไรก็ตาม อัลกอริทึมที่นำเสนอทั้ง 3 ยังคงมีสมรรถนะที่แยกว่าอัลกอริทึม PCB หรือการถอดรหัสแบบซอฟต์แวร์ ดังที่คาดไว้

5.4 การพลิกหลายบิตสำหรับอัลกอริทึมการถอดรหัสแบบเพจ (multiple bit-flipping)

ในหัวข้อต่อไป จะกล่าวถึงอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (P-WBF) ที่เป็นการถอดรหัสโดยการพลิกบิตเพียงหนึ่งบิต ในแต่ละรอบการวนซ้ำ โดยจะอาศัยค่าสัญญาณที่ได้รับ $\mathbf{y}_{n_1 \times n_2}$ เพื่อคำนวณบิตที่มีความไม่น่าเชื่อถือสูงสุด ดังนั้น เมื่อมีการถอดรหัสในคำรหัสที่มีขนาดใหญ่มาก หากต้องการให้สมรรถนะของมันมีค่าเข้าใกล้จุดคู่เข้า จำเป็นจะต้องใช้จำนวนการวนซ้ำ และเวลาแฝง (Latency) ที่มากขึ้นตามไปด้วย ดังรูป

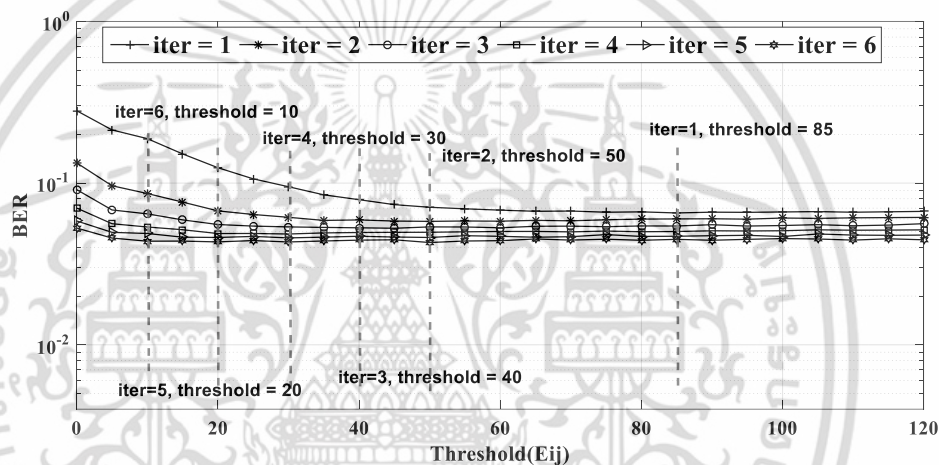


รูปที่ 5.9 สมรรถนะของอัลกอริทึม P-WBF ที่การวนซ้ำรอบต่างๆ

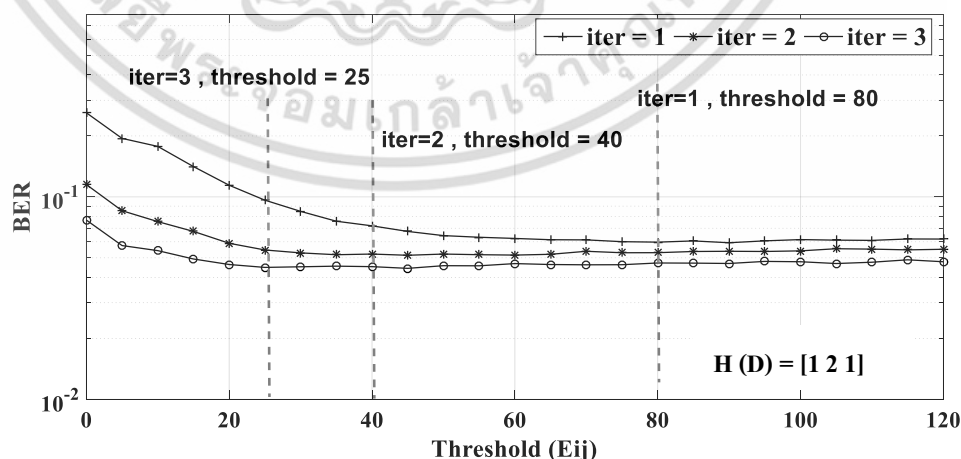
จากรูปที่ 5.9 เป็นการสร้างรหัส P-WBF โดยใช้ค่าพารามิเตอร์ตามตารางที่ 5.1 โดยทำการจำลองสมรรถนะ BER ในแต่ละรอบการวนซ้ำ พบว่าการที่จะเข้าไปถึงจุดที่สมรรถนะคู่เข้านั้น จำเป็นต้องใช้การวนซ้ำถึงประมาณ 40,000 รอบ สำหรับช่วงเอสเอ็นอาร์ 2.85 – 2.97 ฉะนั้น ในวิทยานิพนธ์เล่มนี้ จึงได้ทำการนำเสนออัลกอริทึมในการปรับปรุงความเร็วในการถอดรหัสของอัลกอริทึม P-WBF อันได้แก่ อัลกอริทึมการเลือกพลิกหลายบิตอย่างคงที่สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping: PF-WBF) และอัลกอริทึมการเลือกหลายบิตแบบเทรซโฮลด์สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page threshold weighted bit-flipping: PT-WBF) โดยขั้นตอนของการถอดรหัสทั้งสองวิธีนั้น สามารถดูวิธีการทำงานอย่างละเอียดได้ในหัวข้อที่ 4.7.1 และ 4.7.2 ตามลำดับ

5.4.1 การค้นหาเทรสโพลด์สำหรับการเลือกหลายบิตสำหรับอัลกอริทึม PT-WBF

เป็นที่ทราบกันดีว่าการคัดเลือกบิตจำนวน N โดยที่ $N > 1$ ด้วยอัลกอริทึมการเลือกหลายบิตแบบเทรสโพลด์สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (PT-WBF) จะพิจารณาจำนวนบิตที่พลิกในแต่ละรอบ N จากค่าเทรสโพลด์ที่ได้มาจากการจำลอง และทดลองซ้ำแล้วซ้ำเล่าเพื่อให้ได้ค่าที่เหมาะสมที่สุด โดยค่าเทรสโพลด์ที่เหมาะสมนี้ จะทำให้อัลกอริทึม PT-WBF สามารถเพิ่มความเร็วในการถอดรหัส พร้อมทั้งทำให้ได้ค่าสมรรถนะ BER ที่ดีที่สุดในแต่ละรอบการวนซ้ำ อย่างไรก็ตาม การจำลองเพื่อหาค่าเทรสโพลด์ทั้งในส่วน of ช่องสัญญาณ AWGN และ TDMA ในหัวข้อต่อไปนี้จะใช้ค่าพารามิเตอร์ของอัลกอริทึม P-WBF จากตารางที่ 5.1 และค่าพารามิเตอร์ของหัวอ่าน และโครงสร้างของการบันทึกข้อมูลจะนำมาจากตารางที่ 5.3 (ตามงานวิจัย S. Kobayashi [23-24]) ซึ่งจะได้ผลดังกราฟด้านล่าง



รูปที่ 5.10 ค่าเทรสโพลด์ที่เหมาะสมในการเลือกพลิกหลายบิตสำหรับ PT-WBF บน AWGN



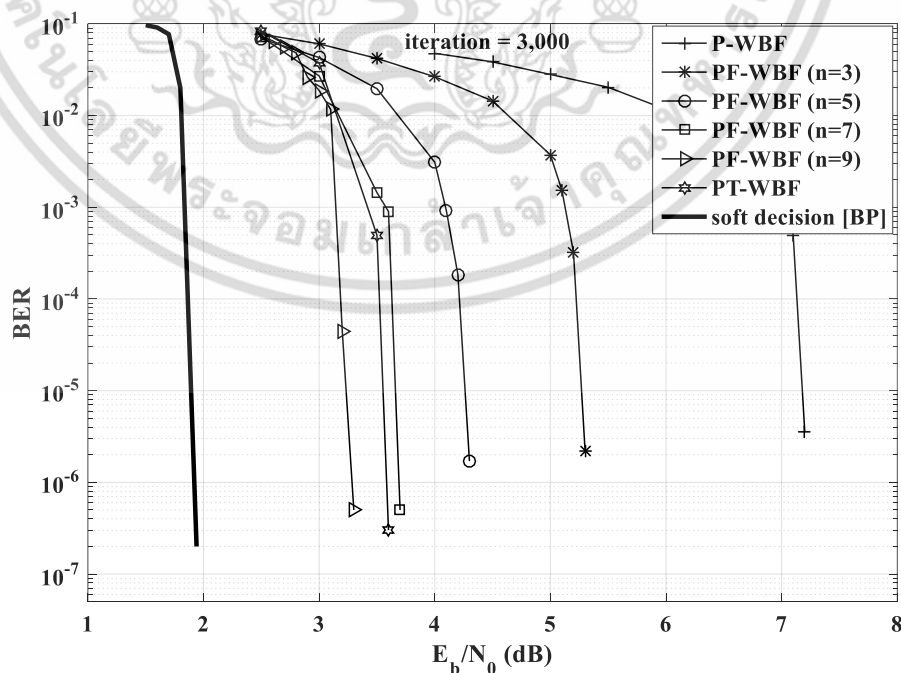
รูปที่ 5.11 ค่าเทรสโพลด์ที่เหมาะสมในการเลือกพลิกหลายบิตสำหรับ PT-WBF บน TDMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

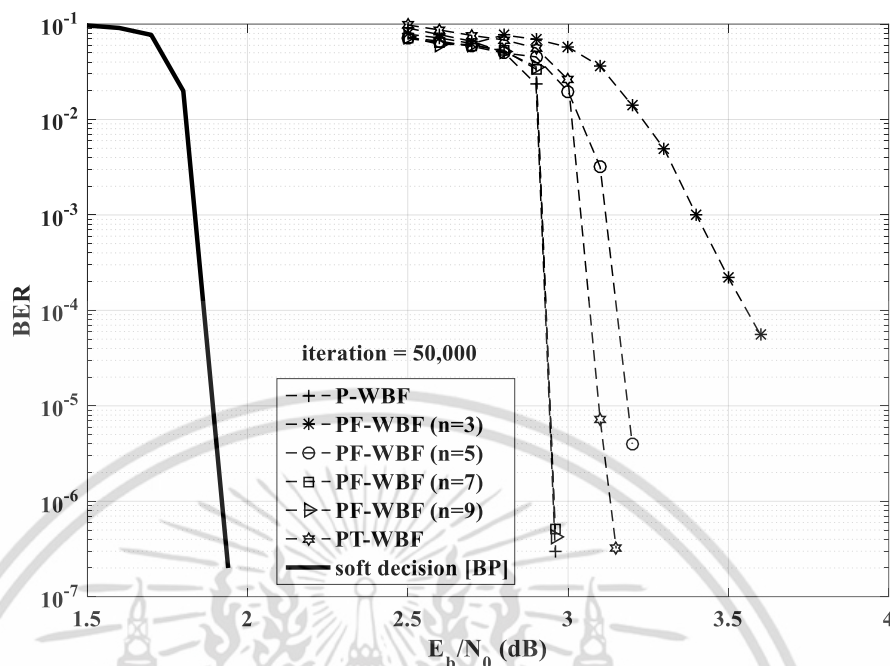
จากรูปที่ 5.10 เป็นการจำลอง และทดลองเพื่อหาค่าเทรสโพลต์ที่เหมาะสมที่สุด บนช่องสัญญาณเกาส์สีขาว สำหรับคัดเลือก N บิต เพื่อทำการพลิกด้วยอัลกอริทึม PT-WBF ซึ่งจะพบว่าที่การวนซ้ำรอบแรกนั้น เมื่อเราทำการพลิกบิตทั้งหมดที่มีค่าความไม่น่าเชื่อถือ $E_{i,j}$ มากกว่า 85 จะทำให้ได้ค่าสมรรถนะ BER ที่ดีที่สุด และยิ่งจะช่วยให้เพิ่มความเร็วในการพลิกบิตด้วยเช่นกัน ดังนั้น ที่การวนซ้ำในรอบแรกค่าเทรสโพลต์ของอัลกอริทึม PT-WBF จึงมีค่ากับ 85 นั้นเอง ส่วนในรอบที่ 2, 3, 4, 5 และ 6 นั้นค่าเทรสโพลต์จะมีค่าเท่ากับ 50, 40, 30, 20 และ 10 ตามลำดับ จะสังเกตว่าเมื่อการวนซ้ำมีจำนวนรอบสูงขึ้นกราฟจะมีลักษณะเป็นเส้นตรง ซึ่งนั่นจะหมายความว่า การเลือกหลายบิตมาพลิกจะไม่ส่งผลกระทบต่อสมรรถนะของอัลกอริทึม ดังนั้น หลังจากการวนซ้ำรอบที่ 6 อัลกอริทึม PT-WBF จะทำการพลิกเพียง 1 บิตเช่นเดิม เช่นเดียวกับกราฟในรูปที่ 5.11 ที่เป็นกราฟแสดงค่าเทรสโพลต์ที่เหมาะสมในการเลือกหลายบิตมาพลิกด้วยอัลกอริทึม PT-WBF บนช่องสัญญาณการบันทึกเชิงแม่เหล็กสองมิติ (TDMR) ที่มีค่าพารามิเตอร์ของหัวอ่าน และโครงสร้างของการบันทึกข้อมูลจากตารางที่ 5.3 โดยจะพบว่าในรอบที่ 1, 2 และ 3 นั้น ค่าเทรสโพลต์จะมีค่าเท่ากับ 80, 40 และ 25 ตามลำดับ (ค่าเทรสโพลต์สำหรับ AWGN และ TDMR ทั้งหมด ถูกแสดงในตารางที่ 4.4)

5.4.2 การพลิกหลายบิตสำหรับอัลกอริทึม P-WBF บนช่องสัญญาณรบกวนเกาส์สีขาวบวก

ในหัวข้อต่อไปนี้จะกล่าวถึงการจำลองสมรรถนะของรหัสระหว่าง 3 อัลกอริทึม อันได้แก่ อัลกอริทึมการถอดรหัสแบบเพจ (P-WBF) และอัลกอริทึมที่ถูกนำเสนอเพื่อเพิ่มความเร็วในการถอดรหัสของอัลกอริทึม P-WBF อันได้แก่ อัลกอริทึมการถอดรหัสบิตพลิกบิตปิ้งชนิดถ่วงน้ำหนักแบบเพจ (PF-WBF) และอัลกอริทึมการเลือกหลายบิตแบบเทรสโพลต์สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบิตปิ้งชนิดถ่วงน้ำหนักแบบเพจ (PT-WBF) ตามลำดับ ค่าพารามิเตอร์สำหรับสร้างรหัสทั้งสามนั้นสามารถพิจารณาได้จากตารางที่ 5.1 (ใช้เฉพาะพารามิเตอร์ของอัลกอริทึม P-WBF)



รูปที่ 5.12 สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบต่ำ (AWGN) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 สมรรถนะของอัลกอริทึม P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบสูง (AWGN)

ในรูปที่ 5.12 เป็นกราฟแสดงสมรรถนะ BER ระหว่าง 3 อัลกอริทึมได้แก่ P-WBF, PF-WBF และ PT-WBF ที่การวนซ้ำรอบต่ำ (ที่ 3,000 รอบ) จะพบว่าอัลกอริทึม PF-WBF ที่ $N = 3, 5, 7$ และ 9 จะให้ค่าเกณฑ์สเท่ากับ 1.9 dB, 2.9 dB, 3.5 dB และ 3.9 dB เมื่อเทียบกับอัลกอริทึม P-WBF ตามลำดับ รวมทั้งอัลกอริทึม PT-WBF ก็ให้ค่าเกณฑ์สเท่ากับ 3.6 dB เมื่อเทียบกับอัลกอริทึม P-WBF ที่อัตราบิดผิดพลาดเท่ากับ 1×10^{-5} อย่างไรก็ตามที่การวนซ้ำรอบสูงหรือที่จุดลู่เข้านั้น อัลกอริทึม PF-WBF ที่ $n = 3$ และ 5 รวมทั้งอัลกอริทึม PT-WBF จะมีสมรรถนะที่แย่กว่าอัลกอริทึม P-WBF ดังแสดงในรูปที่ 5.13

5.4.3 การพลิกหลายบิตสำหรับอัลกอริทึม P-WBF บนช่องสัญญาณบันทึกเชิงแม่เหล็กสองมิติ (TDMR)

ในหัวข้อต่อไปนี้จะกล่าวถึงการจำลองสมรรถนะระหว่าง 3 อัลกอริทึมดังกล่าวข้างต้น บนช่องสัญญาณที่มีการแทรกสอดสัญลักษณ์และแทรกข้างเคียง (มี ISI และ ITI) หรือบนระบบการบันทึกข้อมูลเชิงแม่เหล็กสองมิติ (TDMR) โดยใช้ค่าพารามิเตอร์สำหรับสร้างรหัสจากตารางที่ 5.1 และใช้ค่าพารามิเตอร์ของหัวอ่าน และโครงสร้างของการบันทึกข้อมูลแต่ละบิตนั้นสามารถแสดงในตารางที่ 5.3 (ตามงานวิจัย S. Kobayashi [23-24])

การวนซ้ำรอบต่ำ อัลกอริทึม PF-WBF ที่ $N = 3, 5, 7$ และ 9 จะให้ค่าเกนรหัสนเท่ากับ 4.1 dB, 5.8 dB, 6.7 dB และ 7.1 dB เมื่อเทียบกับอัลกอริทึม P-WBF ตามลำดับ เช่นเดียวกับอัลกอริทึม PT-WBF จะให้ค่าเกนรหัสนเท่ากับ 5.9 dB เมื่อเทียบกับอัลกอริทึม P-WBF ที่อัตราบิดผิดพลาดเท่ากับ 1×10^{-5} จากนั้นสมรรถนะของอัลกอริทึมดังกล่าวข้างต้น จะแย่กว่าอัลกอริทึม P-WBF ดังแสดงในรูปที่ 5.15 เมื่อถูกใช้ที่การวนซ้ำรอบสูง หรือ ณ จุดลู่เข้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการวิจัยและข้อเสนอแนะ

รหัสพรีดีซีเคความหนาแน่นต่ำ (LDPC) เป็นรหัสที่ใช้แก้ไขความผิดพลาดของข้อมูลจากความผิดพลาดของสัญญาณ ซึ่งรหัสดังกล่าวนี้ ใช้กันอย่างแพร่หลายในหลายๆ แอปพลิเคชันในระบบดิจิทัล ปัจจุบัน เนื่องจากสมรรถนะของรหัสแอลดีพีซีได้รับการพิสูจน์จากหลายๆงานวิจัยแล้วว่า มีค่าที่เข้าใกล้ขีดจำกัดของแชนนอนเช่นเดียวกับรหัสเทอร์โบ เมื่อไม่นานมานี้มีการพิสูจน์อีกครั้งว่า รหัสแอลดีพีซีมีความซับซ้อนในการเข้ารหัสและถอดรหัสที่ต่ำกว่ารหัสเทอร์โบ อีกทั้งสมรรถนะของรหัสแอลดีพีซีจะมีค่าที่ดีกว่ารหัสเทอร์โบสำหรับความยาวของคำรหัสสูง อย่างไรก็ตามสำหรับบางแอปพลิเคชัน อาทิ ระบบบันทึกข้อมูลเชิงแม่เหล็กหรือฮาร์ดดิสก์ การเพิ่มความยาวของคำรหัส ส่งผลให้ความซับซ้อนในการเข้ารหัสและถอดรหัสของรหัสมีค่าที่มากขึ้นตามไปด้วย ดังนั้น ในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจึงค้นหาวิธีในการลดความซับซ้อนทั้งในส่วนการเข้ารหัส และถอดรหัส สำหรับความยาวคำรหัสสูง

รหัสแอลดีพีซีแบบโปรดักส์ [9] เป็นอัลกอริทึมที่ถูกนำเสนอเมื่อไม่นานมานี้ โดยถูกนำเสนอเพื่อลดความซับซ้อนในการเข้ารหัสสำหรับความยาวคำรหัสสูง โดยทำการแปลงบิตข้อมูลความยาวสูงให้อยู่ในรูปแบบสองมิติก่อนจะทำการเข้ารหัสและถอดรหัส โดยผลที่ได้แสดงให้เห็นว่ามันสามารถลดความซับซ้อนในการเข้ารหัสได้เป็นอย่างดี แต่ความซับซ้อนในการถอดรหัสยังคงมีค่าที่สูง เนื่องจาก รหัสแอลดีพีซีแบบโปรดักส์ [9] ใช้การถอดรหัสแบบแพร่กระจายความเชื่อมั่น (BP) หรือการถอดรหัสแบบซอฟต์แวร์ ดังนั้น ในงานวิจัยเล่มนี้ จึงทำการนำเสนอสามอัลกอริทึมอันได้แก่ อัลกอริทึมการถอดรหัสแบบบิตพลิกบิตหนักถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping : P-WBF), แบบแถวสลับหลัก (Row-column weighted bit-flipping : RC-WBF) และแบบดัดแปลงแถวสลับหลัก (Modified row-column weighted bit-flipping : MRC-WBF) ที่เป็นการประยุกต์ใช้การถอดรหัสแอลดีพีซีแบบบิตพลิกบิตหนักถ่วงน้ำหนักหรือการถอดรหัสแบบหยาบ ร่วมกับการเข้ารหัสแอลดีพีซีแบบโปรดักส์ ในบทที่ 5 นั้น ผู้วิจัยได้ทำการจำลองสมรรถนะของอัลกอริทึมดังกล่าวบนช่องสัญญาณเกาส์สีขาว และช่องสัญญาณสำหรับระบบบันทึกเชิงแม่เหล็กสองมิติ (TDMR) ที่ความหนาแน่นเชิงพื้นที่เท่ากับ 5.62 Tb/in^2 (ใช้แบบจำลอง S. Kobayashi [23-24]) ซึ่งผลที่ได้พบว่า ถึงแม้สมรรถนะในการแก้ไขความผิดพลาดของทั้งสามอัลกอริทึมที่นำเสนอ จะแยกกว่าอัลกอริทึมแอลดีพีซีแบบโปรดักส์ที่มีการถอดรหัสแบบซอฟต์แวร์เพียงเล็กน้อย แต่ความซับซ้อนในการถอดรหัสจะมีค่าน้อยกว่ามาก และที่สำคัญ ความซับซ้อนในการเข้ารหัสของอัลกอริทึมที่นำเสนอทั้งสาม ยังคงมีค่าที่เทียบเคียงอัลกอริทึมแอลดีพีซีแบบโปรดักส์ที่มีการถอดรหัสแบบซอฟต์แวร์ แต่หากทำการเปรียบเทียบกันเฉพาะอัลกอริทึมที่นำเสนอทั้งสามจะพบว่า อัลกอริทึม P-WBF จะให้สมรรถนะในการแก้ไขความผิดพลาดที่ดีกว่า MRC-WBF และ RC-WBF ตามลำดับ ส่วนความซับซ้อนในการถอดรหัสนั้น อัลกอริทึม RC-WBF จะมีค่าน้อยที่สุด โดยน้อยกว่า MRC-WBF และ P-WBF ตามลำดับ อย่างไรก็ตาม ทั้ง 3 อัลกอริทึมที่นำเสนอนั้น ล้วนมีปัญหาที่เกิดขึ้นในกระบวนการถอดรหัส อาทิ ปัญหาสมรรถนะไม่ลู่เข้าที่การวนซ้ำรอบสูงของอัลกอริทึม RC-WBF และ MRC-WBF ซึ่งจะได้รับการแก้ไขด้วยวิธีการกำหนดขอบเขตในการพลิกบิตด้วยค่าเทรโซลต์ รวมถึงปัญหาใช้รอบการวนซ้ำ และเวลาในการถอดรหัสจำนวนมาก เพื่อเข้าสู่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมรรถนะลู่เข้าของอัลกอริทึม P-WBF ซึ่งปัญหานี้ก็จะได้รับการแก้ไขด้วยอัลกอริทึมที่ประยุกต์มาจากอัลกอริทึม P-WBF อันได้แก่ อัลกอริทึมการเลือกพลิกหลายบิตอย่างคงที่สำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page-based weighted bit-flipping: PF-WBF) และอัลกอริทึมการเลือกหลายบิตแบบเทอร์สโสลสำหรับอัลกอริทึมการถอดรหัสบิตพลิกบั้งชนิดถ่วงน้ำหนักแบบเพจ (Page threshold weighted bit-flipping: PT-WBF) ซึ่งผลที่ได้นั้นแสดงให้เห็นว่าอัลกอริทึม PF-WBF และ PT-WBF สามารถเพิ่มความเร็วในการถอดรหัสของอัลกอริทึม P-WBF ได้ และยังมีสมรรถนะในการแก้ไขความผิดพลาดที่ดีกว่าอัลกอริทึม P-WBF อย่างไรก็ตาม สมรรถนะของทั้ง 2 อัลกอริทึม จะมีการลดทอนที่การวนซ้ำรอบสูง

ในงานวิจัยอนาคต เราสามารถศึกษาถึงความเป็นไปได้ในการเพิ่มขึ้นของค่าความหนาแน่นเชิงพื้นที่ ของระบบบันทึกเชิงแม่เหล็กสองมิติ ที่ซึ่งในปัจจุบันนี้มีค่าที่มากกว่าเท่ากับ 10 Tb/in^2 โดยการลดขนาดของเกรนที่ใช้สำหรับการบันทึก 1 บิต รวมทั้งจะนำแบบจำลองโวโรนอยด์ (Voronoi) มาใช้แทนแบบจำลอง 4-Grain เพื่อความสมจริงในการจำลองยิ่งขึ้น และจะทำการทดสอบอัลกอริทึมการถอดรหัสแอลดีพีซีดั่งที่กล่าวในวิทยานิพนธ์เล่มนี้ ที่ห้องสัญญาณของของระบบบันทึกข้อมูลเชิงแม่เหล็กสองมิติ ที่มีค่าความหนาแน่นเชิงพื้นที่มากกว่าเท่ากับ 10 Tb/in^2 ด้วยแบบจำลองโวโรนอยด์



บรรณานุกรม

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2, pp.1064-1070, 1993.
- [2] Shannon, C.E. **Communication in the presence of noise.** *Proc. Inst. Radio Eng.* vol. 37, pp.10–21, 1949.
- [3] R. Gallager, "Low-density parity-check codes," *IRE Transaction. Information Theory*, vol. 8, pp.21-28, 1962.
- [4] D. J. C. MacKay, and I. N. Neal, "Near Shannon limit performance of low-density parity - check codes," *Electronics Letters*, vol. 33, pp.457-458, 1997.
- [5] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no.5, pp. 533-547, 1981.
- [6] S. Jeon and B. V. K. Vijaya Kumar, "Performance and Complexity of 32 k-bit Binary LDPC Codes for Magnetic Recording Channels," *IEEE Transactions on Magnetics*, vol. 46, pp. 2244-2247, 2010.
- [7] L. Kong, L. He, P. Chen, G. Han and Y. Fang, "Photograph based quasi-cyclic LDPC coding for ultra-high density magnetic recording channels," *IEEE Magnetics Conference (INTERMAG)*, pp. 1-1, 2015.
- [8] P. Chen; L. Kong; Y. Fang and L. Wang, "The design of photograph LDPC codes for two-dimensional magnetic recording channels," *IEEE Magnetics Conference (INTERMAG)*, 2015.
- [9] Z. Qi and N. C. Sum, "LDPC product codes," *Communications Systems, ICCS 2004. The Ninth International Conference on*, pp. 481-483, 2004.
- [10] M. Horstein, "Review of Low-Density Parity-Check Codes (Gallager, R. G.; 1963)," *IEEE Transactions on Information Theory*, vol. 10, 1964.
- [11] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, pp. 2711-2736, 2001.
- [12] J. Zhang, and M. P. C. Fossorier, "A modified weighted bit-flipping decoding of low-density Parity-check codes," *IEEE Communications Letters*, vol. 8, pp. 165-167, 2004.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [13] M. Jiang, C. Zhao, Z. Shi and Y. Chen, "An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes," *IEEE Communications Letters*, vol. 9, pp. 814-826, 2005.
- [14] F. Guo, and L. Hanzo, "Reliability ratio based weighted bit-flipping decoding for low-density parity-check code," *Electronics Letters*, vol. 40, pp. 1356-1358, 2004.
- [15] C. H. Lee, and W. Wolf, "Implementation-efficient reliability ratio based weighted bit-flipping decoding for LDPC codes," *Electronics Letters*, vol. 41, pp. 755-757, 2005.
- [16] H. J. Richter , A. Y. Dobin , O. Heinonen , K. Z. Gao , R. J. M. v. d. Veerdonk , R. T. Lynch , J. Xue , D. Weller , P. Asselen , M. F. Erden and R. M. Brockie, "Recording on bit-patterned media at densities of 1 Tb/in and beyond," *IEEE Transactions on Magnetics.*, vol. 42, pp. 2255-2260, 2006.
- [17] J. G. Zhu, X. Zhu and Y. Tang, "Microwave Assisted Magnetic Recording," *IEEE Transactions on Magnetics.*, vol. 44, pp. 125-131, 2008.
- [18] M. Kryder, E. C. Gage, T. W. McDaniel, W. A. Challener, R. E. Rottmayer, G. Ju, Y.-T. Hsia and M. F. Erden, "Heat assisted magnetic recording," *Proc. IEEE*, vol.96, no.11, pp. 1810-1835, 2008.
- [19] R. Wood, M. Williams, A. Kavcic and J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," *IEEE Transactions on Magnetics*, vol. 45, no. 2, pp. 917-923, 2009.
- [20] K. S. Chan, Jim J. Miles, E. Hwang, B. V. K. VijayaKumar, J. G. Zhu, W. C. Lin and R. Negi, "TDMR Platform Simulations and Experiments," *IEEE Transactions on Magnetics*, vol. 45, no. 10, pp. 3837-3843, 2009.
- [21] Y. Nakamura and I. Tagawa, Possibilities of perpendicular magnetic recording, *IEEE Transactions on Magnetics*, vol. 24, no. 6, pp. 2329-2334, 1988.
- [22] M. Yamashita, H. Osawa; Y. Okamoto, Y. Nakamura, Y. Suzuki, K. Miura and H. Muraoka, "Read/Write channel modeling and two-Dimensional Neural network equalization for two-Dimensional magnetic recording," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 3558-3561, 2011.
- [23] M. Yamashita, H. Osawa; Y. Okamoto, Y. Nakamura, Y. Suzuki, K. Miura and H. Muraoka, "Modeling of Writing Process for Two-Dimensional Magnetic Recording and Performance Evaluation of Two-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Dimensional Neural Network Equalizer," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 4586-4589, 2012.
- [24] S. Kobayashi, "The frontiers of real-coded genetic algorithm," (in japanese) *J. Jpn. Soc. Artif. Intell.*, vol. 24, no. 1, pp. 147-162, 2009.
- [25] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619-637, 2001.
- [26] X. Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386- 398, 2005.
- [27] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Information Theory*, vol. 47, pp. 599-618, 2001.
- [28] D. Oh and Keshab K. Parhi, "Low complexity implementations of sum-product algorithm for decoding low-density parity-check codes," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 262-267, 2006.
- [29] S. Y. Chung, T. J. Richardson and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation", *IEEE Transactions on Information Theory*, vol. 47, pp. 657-670, 2001.
- [30] S. Liu, and D. A. Pados, "Low Complexity Decoding of Finite Geometry LDPC Codes", *IEEE International Conference on Communications (ICC)*, vol. 53, pp.415-421, Mar. 2005.
- [31] J. Moon and W. Zeng, "Equalization for maximum likelihood detector", *IEEE Transactions on Magnetics*, vol.31, pp.1083-1088, March, 2009.
- [32] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", *Proc. GLOBECOM '95*, pp. 1680-1686, 1995.
- [33] <http://www.vcharkarn.com/blog/41002>, 2009.
- [34] <https://www.kasikornresearch.com/TH/K-EconAnalysis>, 2016.
- [35] ดร.ปิยะ โควินท์ทวีวัฒน์, **Signal processing for digital data storage (Receiver design)**, NECTEC, 2007.
- [36] ศ.ดร.พรชัย ทรัพย์นิธิ, **Advanced digital communication systems, 2012**

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลงานที่ได้รับการตีพิมพ์

1. S. Khittiwitayakul, W. Phakphisut and P. Supnithi, “**Weighted bit-flipping decoding for product LDPC codes,**” *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI)*, Chiang Mai, Thailand, June 2016.
2. S. Khittiwitayakul, W. Phakphisut and P. Supnithi, “**Page-based weighted bit-flipping decoding of product LDPC codes for two dimensional magnetic recording,**” *International Technical Conference On Circuits/Systems, Computers and Communications (ITC-CSCC)*, Okinawa, Japan, July 2016.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย ศิริวิชัย กิตติวิชญกุล
วัน เดือน ปีเกิด	13 กุมภาพันธ์ 2535
ที่อยู่	76/2 ม.7 ถ. ชัยภูมิ-บ้านเขว้า ต.โนเมือง อ. เมือง จ.ชัยภูมิ 36000
อีเมลล์	sirawit.khitt@gmail.com, 57601155@kmitl.ac.th
เฟสบุค	https://www.facebook.com/sirawit.khitiwichayakul
ประวัติการศึกษา	พ.ศ.2556 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
งานวิจัยที่สนใจ	รหัสช่องสัญญาณ, ระบบบันทึกข้อมูลเชิงแม่เหล็ก และทฤษฎีการสื่อสาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้