การออกแบบเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี

ว่าที่ร้อยตรีณัฐ ตันติบุตร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

Design of a Parity-Check Matrix for Non-Binary LDPC Codes

Acting Sub Lieutenant Nut Tantibut

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

หัวข้อวิทยานิพนธ์	การออกแบบเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี
โดย	ว่าที่ร้อยตรีณัฐ ตันติบุตร
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รองศาสตราจารย์ ดร.ปิยะ โควินท์ทวีวัฒน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พสุ แก้วปลั่ง)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(รองศาสตราจารย์ ดร.ปิยะ โควินท์ทวีวัฒน์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์สุวิทย์ นาคพีระยุทธ)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร.พิสิฐ วนิชชานันท์)

ณัฐ ตันติบุตร : การออกแบบเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี (Design of a Parity-Check Matrix for Non-Binary LDPC Codes) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: รศ. ดร.ปิยะ โควินท์ทวีวัฒน์, 53 หน้า.

การส่งผ่านข่าวสารดิจิทัลในระบบสื่อสารไร้สายมักได้รับผลกระทบจากสัญญาณรบกวนและสัญญาณแทรกสอดทำให้เกิดความผิดพลาดบิตขึ้น รหัสแก้ไขความผิดพลาดเป็นเทคนิคสำคัญที่ใช้แก้ปัญหาความผิดพลาดบิตดังกล่าวได้อย่างมีประสิทธิภาพ ในปัจจุบันรหัสรหัสพาริตีเช็กความหนาแน่นต่ำหรือที่เรียกโดยย่อว่ารหัสแอลดีพีซีได้รับความสนใจอย่างมากในการประยุกต์ใช้งานจริงในทางปฏิบัติ วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อปรับปรุงรหัสแอลดีพีซีให้มีสมรรถนะในการแก้ไขความผิดพลาดได้ดีขึ้นโดย 2 แนวทางคือ แนวทางแรกเสนอการปรับปรุงสมรรถนะของรหัสแอลดีพีซีแบบไบนารีที่มีอยู่เดิมให้ดีขึ้นด้วยการแปลงรหัสดังกล่าวให้กลายเป็นรหัสนอนไบนารี โดยเน้นรหัสแอลดีพีซีที่มีน้ำหนักคอลัมน์เท่ากับสอง และแนวทางที่สองเป็นการพัฒนาวิธีการถอดรหัสขึ้นใหม่โดยให้มีประสิทธิภาพดีขึ้นโดยพิจารณาทั้งค่าอัตราความผิดพลาดบิตและจำนวนรอบการวนซ้ำให้ลดลงอย่างมีนัยสำคัญ ผลการทดสอบในส่วนแรกพบว่ารหัสนอนไบนารีที่เสนอขึ้นมีอัตราความผิดพลาดบิตที่ต่ำกว่ารหัสโพรโทกราฟขนาดเดียวกันที่องค์กรนาซาเลือกใช้อยู่ประมาณ 0.4 dB ผลการทดสอบในส่วนที่สองพบว่าการถอดรหัสที่เสนอขึ้นใหม่นี้ให้อัตราความผิดพลาดบิตที่ลดต่ำลงกว่าวิธีการแบบดั้งเดิมอย่างชัดเจน เมื่อทำการทดสอบกับรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n ทั้ง 3 ขนาดคือ 648, 1296 และ 1944 บิต อีกทั้งจำนวนรอบการวนซ้ำลดลงมาก ซึ่งมีความหมายว่าการถอดรหัสแบบนี้นอกจากจะมีอัตราความผิดพลาดที่ต่ำลงกว่าเดิมแล้วการประมวลผลก็ลดลงด้วย

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต .....
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
ปีการศึกษา	2559	ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

5670189421 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS: LDPC CODES / CHANNEL CODING / FAST FOURIER TRANSFORM / DIGITAL COMMUNICATION / HADAMARD MATRIC

NUT TANTIBUT: Design of a Parity-Check Matrix for Non-Binary LDPC Codes. ADVISOR: ASSOC. PROF.LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., CO-ADVISOR: ASSOC. PROF.PIYA KOVINTAVEWAT, Ph.D., 53 pp.

Digital data transmission over wireless communication systems suffers from noise and interference causing bit errors. Error correcting codes are very important technique for handling this problem of errors very efficiently. At present, low density parity check codes or LDPC codes have attracted much attention and been applied in practice. This thesis aims to improve performance of the LDPC codes in two approaches. The first approach proposes to convert existing binary LDPC codes to become non binary LDPC codes with emphasis on column weighted two LDPC codes. The second approach develops a new decoding technique that improve both bit error rates as well as the number of iterations. Simulation results of the first part show that the proposed non binary codes have a lower bit error rate than the protograph codes adopted by NASA by 0.4 dB. Simulation results in the second part show that the proposed decoding technique is superior to the conventional decoding technique for all 3 sizes of LDPC codes in the IEEE 802.11n standard, i.e., 648, 1286 and 1944 bits. In addition, the number of iterations required for convergence is significantly reduced. This means that not only the proposed decoding technique exhibits lower bit error rate but the processing time is also decreased.

Department:	Electrical Engineering	Student's Signature	.....
Field of Study:	Electrical Engineering	Advisor's Signature	.....
Academic Year:	2016	Co-Advisor's Signature	.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากผู้วิจัยได้รับความช่วยเหลือ ดูแลเอาใจใส่เป็นอย่างดีจากหลายๆฝ่าย โดยเฉพาะอาจารย์ที่ปรึกษาสองท่านคือ รองศาสตราจารย์ ดร. ลัญฉกร วุฒิสิทธิกุลกิจ และรองศาสตราจารย์ ดร.ปิยะ โควินท์ทวีวัฒน์ ในการแนะนำ ตรวจแก้ไข ให้ข้อเสนอแนะ และติดตามความก้าวหน้าในการทำวิทยานิพนธ์ นอกจากนี้ผู้วิจัยยังได้รับการช่วยเหลือและกำลังใจจาก คุณพ่อ คุณแม่ พี่สาว และเพื่อนๆ โดยเฉพาะอย่างยิ่ง สพญ.ดร.รักษวรรณ ดีโนนโพธิ์ ผู้ที่คอยอยู่เคียงข้างเป็นกำลังใจให้เสมอมา และคอยช่วยเหลือในด้านต่างๆ ตลอดช่วงระยะเวลาของการทำวิจัยนี้ ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาของทุกๆท่านเป็นอย่างยิ่ง และขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

สารบัญ

หน้า

หน้า

สารบัญรูปภาพ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในอนาคตองค์กรภาคธุรกิจ องค์กรภาครัฐบาล และทางทหาร จะมีความต้องการส่งข้อมูลและระบบจัดเก็บข้อมูลดิจิทัลที่มีประสิทธิภาพและมีความน่าเชื่อถือที่สูงมากยิ่งขึ้น ซึ่งความต้องการที่สูงขึ้นนี้มาจากการควบรวมของโครงข่ายข้อมูลความเร็วสูงขนาดใหญ่ที่ใช้สำหรับการแลกเปลี่ยน การประมวลผล และการจัดเก็บของข้อมูลข่าวสารดิจิทัล จึงมีความต้องการที่จะต้องออกแบบระบบเพื่อรวมการสื่อสารและเทคโนโลยีคอมพิวเตอร์ไว้ด้วยกัน ปัญหาหลักของการออกแบบระบบคือการควบคุมความผิดพลาดของข้อมูลเพื่อให้การแก้ไขข้อมูลที่ผิดพลาดมีความน่าเชื่อถือ

ในปี ค.ศ.1948 คล็อด แชนนอน [1] เสนอว่าการเข้ารหัสข้อมูลข่าวสารที่เหมาะสมนั้นจะช่วยลดความผิดพลาดของข้อมูลข่าวสารที่เกิดจากสัญญาณรบกวนของช่องสัญญาณหรือจากสื่อกลางการจัดเก็บข้อมูลได้โดยไม่ทำสูญเสียอัตราการส่งข้อมูลข่าวสารหรือการเก็บข้อมูลตราบเท่าที่อัตราของข้อมูลข่าวสารน้อยกว่าค่าความจุของช่องสัญญาณ ในสภาพแวดล้อมที่เต็มไปด้วยสัญญาณรบกวน งานของแชนนอนได้พิสูจน์ว่าการใช้รหัสแก้ไขความผิดพลาดจะช่วยเพิ่มประสิทธิภาพของการส่งข่าวสารข้อมูลได้ด้วยการเข้า – ถอดรหัสช่องสัญญาณให้มีประสิทธิภาพเพื่อให้สามารถควบคุมความผิดพลาดของข้อมูลข่าวสาร ในปัจจุบันจึงมีการพัฒนาระบบให้สามารถใช้กับระบบให้สามารถใช้กับระบบดิจิทัลความเร็วสูงได้และการใช้รหัสเพื่อแก้ไขความผิดพลาดกลายเป็นส่วนหนึ่งในการออกแบบระบบการสื่อสารและระบบการเก็บข้อมูลดิจิทัลสมัยใหม่

รหัสพาริตีเช็กความหนาแน่นต่ำหรือรหัสแอลดีพีซีเป็นหนึ่งในรูปแบบของการเข้า – ถอดรหัสที่มีความสามารถในการแก้ไขความผิดพลาดที่เกิดจากสัญญาณรบกวนในระบบการสื่อสาร-โทรคมนาคม ถือกำเนิดขึ้นครั้งแรกในปี ค.ศ.1962 โดย โรเบิร์ด กัลลาเกอร์ [2] ขณะที่กำลังศึกษาในระดับปริญญาเอก ณ สถาบันเทคโนโลยีเมสซาชูเซ็ท ประเทศสหรัฐอเมริกา ซึ่งในขณะนั้นรหัสแอลดีพีซีไม่เป็นที่นิยมมากนักเนื่องจากมีกระบวนการในการเข้า – ถอดรหัสที่ซับซ้อนเมื่อเทียบกับรหัสอื่นๆ เช่น รหัสคอนโวลูชัน รหัสรีด – โซโลมอน นอกจากนั้นเทคโนโลยีทางด้านวิทยาการคอมพิวเตอร์ในสมัยนั้นยังไม่มีความสามารถทางด้านความเร็วมากพอที่จะประมวลผลรหัสแอลดีพีซีให้มีประสิทธิภาพ จึงทำให้รหัสแอลดีพีซีไม่ถูกนำมาเป็นตัวเลือกมากนักในการใช้งานทางด้านระบบสื่อสาร – โทรคมนาคมในสมัยนั้น

อย่างไรก็ตามเมื่อเทคโนโลยีทางด้านวิทยาการคอมพิวเตอร์เริ่มมีการพัฒนาทางด้านความเร็วมากขึ้น รหัสแอลดีพีซีได้ถูกกลับมานำเสนออีกครั้งในปี ค.ศ.1995 โดย เดวิด แม็คเคย์ [3] ซึ่งกล่าวว่ารหัสแอลดีพีซีเป็นรหัสที่มีประสิทธิภาพใกล้เคียงกับขีดจำกัดความจุช่องสัญญาณของแชนนอน ทำให้รหัสแอลดีพีซีเริ่มกลับมาเป็นที่รู้จักอีกครั้งและได้มีนักวิจัยรุ่นหลังพัฒนารหัสแอลดีพีซีในมุมมองที่หลากหลายมากยิ่งขึ้น [4, 5] เช่น การออกแบบรหัสแอลดีพีซีโดยการปรับปรุงการกระจายเมทริกพาริตีเช็กของสมาชิกที่ไม่เป็นศูนย์ การปรับปรุงเส้นทางที่สั้นที่สุด (minimum distance) ของเมทริกพาริตีเช็กในแนวแถวและในแนวคอลัมน์ หรือการออกแบบเส้นทางวนรอบที่สั้นที่สุดของพาริตีเช็กเมทริกเป็นต้น [6]

1.2 วัตถุประสงค์

1.2.1 ศึกษาการทำงานของรหัสแอลดีพีซีแบบนอนไบนารี

1.2.2 นำเสนอเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี

1.2.3 พัฒนาวิธีถอดการถอดรหัสที่มีประสิทธิภาพ สามารถลดการวนซ้ำลงและให้อัตราบิตผิดพลาดที่ลดลง

1.3 ขอบเขตของงานวิจัย

1.3.1 ศึกษาการออกเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี

1.3.2 พัฒนาเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารีเพื่อให้ได้พาริตีเมทริกซ์ที่เหมาะสมสำหรับรหัสแอลดีพีซีแบบนอนไบนารี

1.3.3 ทดสอบสมรรถนะของรหัสแอลดีพีซีแบบนอนไบนารีโดยใช้เมทริกซ์พาริตีเช็กที่ได้พัฒนา ด้วยโปรแกรม MATLAB โดยดูอัตราบิตผิดพลาดต่อพลังงานที่ใช้

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1.4.1 ศึกษางานวิจัยที่เกี่ยวข้องกับการออกแบบเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี

1.4.2 ศึกษาเทคนิคและจำลองสมรรถนะของเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารี และวิเคราะห์ข้อดีข้อเสีย

1.4.3 นำเสนอแนวคิดสำหรับการออกแบบเมทริกซ์พาริตีสำหรับรหัสแอลดีพีซีแบบนอนไบนารีเพื่อให้ได้เมทริกซ์ที่มีความเหมาะสม

1.4.4 จำลองสมรรถนะของเมทริกซ์พาริตีเช็กที่ได้นำเสนอแนวคิดและเปรียบเทียบผลการทดสอบจากการปรับพารามิเตอร์ต่างๆ

1.4.5 พิจารณาเทคนิคการออกแบบเมทริกซ์พาริตีสำหรับรหัสแอลดีพีซีแบบนอนไบนารีชนิดต่างๆ ที่มีอยู่ ในด้านอัตราบิตผิดพลาดต่อพลังงานที่ใช้

1.4.6 ปรับปรุงเทคนิคการออกแบบเมทริกซ์พาริตีเช็กสำหรับรหัสแอลดีพีซีแบบนอนไบนารีเพื่อเพิ่มสมรรถนะ

1.4.7 เขียนบทความทางวิชาการและนำเสนอผลงาน

1.4.8 สรุปผลการวิจัยและจัดทำวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถนำความรู้ที่ได้ไปประยุกต์ใช้งานกับการสื่อสารไร้สายที่ใช้การเข้ารหัสและถอดรหัสแอลดีพีซีแบบนอนไบนารี

1.5.2 ศึกษาเทคนิคการถอดรหัสแอลดีพีซีแบบไบนารีและนอนไบนารี

1.5.3 ศึกษาเทคนิคการแปลงฟูเรียร์แบบเร็วเพื่อช่วยให้การถอดรหัสแอลดีพีซีแบบนอนไบนารีมีความรวดเร็วมากยิ่งขึ้น

1.5.5 ดัดแปลงวิธีการถอดรหัสแอลดีพีซีตามหลักแพร่กระจายความเชื่อและทำการทดสอบสมรรถนะโดยใช้โปรแกรม MATLAB

1.5.6 เปรียบเทียบผลการทดสอบสมรรถนะของเมทริกซ์พาริตีเช็กที่ได้ออกแบบกับเมทริกซ์พาริตีเช็กของงานวิจัยที่ผ่านมาพร้อมสรุปผลการทดสอบ

บทที่ 2
ทฤษฎีพื้นฐาน

2.1 รหัสแก้ไขความผิดพลาดหนาแน่นต่ำหรือรหัสแอลดีพีซีแบบนอนไบนารี

2.1.1 การสร้างรหัสแอลดีพีซีแบบไบนารีวิธีแบบสุ่มและวิธีโครงสร้าง

รหัสแก้ไขความผิดพลาดความหนาแน่นต่ำหรือรหัสแอลดีพีซีมีคุณสมบัติคือมีความหนาแน่นของเมทริกซ์พาริตีเช็กที่ต่ำซึ่งหมายความว่าในเมทริกซ์ประกอบด้วยจำนวนศูนย์เป็นส่วนมากและมีจำนวนที่ไม่เป็นศูนย์เพียงเล็กน้อยเท่านั้นโดยมีพารามิเตอร์ที่สำคัญด้วยกัน 3 อย่างคือ ความยาวคำรหัส n ขนาดของเมทริกซ์พาริตีเช็ก k และจำนวนบิตพาริตี $m = n \times k$ โดยจำนวนสมาชิกที่ไม่เป็นศูนย์ในแต่ละแถวของเมทริกซ์พาริตีเช็กเรียกว่าน้ำหนักแถวหรือแทนด้วย r และจำนวนสมาชิกที่ไม่เป็นศูนย์ในแต่ละคอลัมน์ของเมทริกซ์พาริตีเช็กเรียกว่าน้ำหนักคอลัมน์หรือแทนด้วย g ซึ่งรหัสแอลดีพีซีโดยทั่วไปมีสองแบบดังนี้

- ถ้าน้ำหนักแถวและน้ำหนักคอลัมน์คงที่สำหรับแต่ละแถวและคอลัมน์ของพาริตีเมทริกซ์จะเรียกรหัสแอลดีพีซีแบบสม่ำเสมอ $(regular)$
- ถ้าน้ำหนักแถวและน้ำหนักคอลัมน์ไม่คงที่สำหรับแต่ละแถวและคอลัมน์ของเมทริกซ์พาริตีเช็กจะเรียกรหัสแอลดีพีซีแบบไม่สม่ำเสมอ $(irregular)$

โดยทั่วไปรหัสแอลดีพีซีแบบไม่สม่ำเสมอจะมีประสิทธิภาพดีว่ารหัสแอลดีพีซีแบบสม่ำเสมอ ตัวอย่างของเมทริกซ์พาริตีเช็ก $\mathbf{H}$ ของรหัสแอลดีพีซีแบบไบนารีขนาดเล็กดังรูปที่ 2.1

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

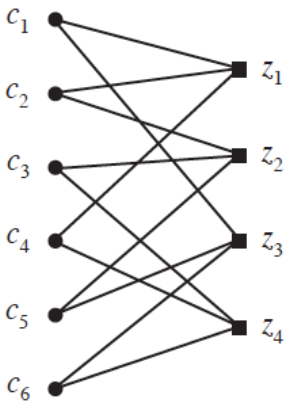
รูปที่ 2.1 เมทริกซ์พาริตีตรวจสอบ H ของรัหสแอลดีพีซีแบบไบนารีขนาดเล็ก

จะเห็นได้ว่าเมทริกซ์ $\mathbf{H}$ มีค่าคงที่น้ำหนักแถว $\rho = 3$ และมีค่าคงที่น้ำหนักคอลัมน์ $\gamma = 3$ หรือกล่าวได้ว่าเป็นรหัสแอลดีพีซีแบบสม่ำเสมอ

2.1.2 กราฟแทนเนอร์

เมทริกซ์พาริตีเช็กสามารถแสดงในรูปแบบของกราฟสองทางหรือเรียกว่าเทนเนอร์กราฟ โดยแต่ละแถวในพาริตีเมทริกซ์จะแทนด้วยสมการพาริตี $z_i, 1 \le i \le m$ และแต่ละคอลัมน์แทนโค้ดบิต

ข้อมูล $c_j, 1 \le j \le n$ จากรูปที่ 2.1 แถวแรกของเมทริกซ์ $\mathbf{H}$ เราจะได้สมการพาริตี $z_1 = c_1 \oplus c_2 \oplus c_4$ ซึ่งหมายถึงสมการพาริตีของแถวแรกเชื่อมต่อกับโค้ดบิตตัวที่ 1 ตัวที่ 2 และตัวที่ 4 ซึ่งถ้าเปรียบเทียบกับเมทริกซ์ในรูปที่ 2.2 บิตที่ถูกเข้ารหัสที่เชื่อมต่อกับสมการพาริตีของแถวที่แรกก็คือโค้ดที่มีค่าเป็น 1 นั่นคือมีการเชื่อมต่อระหว่างโค้ดบิตกับโหนดตรวจสอบตัวนั้นๆ กราฟแทนเนอร์สำหรับเมทริกซ์ $\mathbf{H}$ ในรูปที่ 2.1 สามารถแสดงแบบแทนเนอร์กราฟในรูปที่ 2.2



รูปที่ 2.2 แทนเนอร์กราฟและเมทริกซ์พาริตีเช็กของรูปที่ 2.1

2.1.3 การสร้างเมทริกซ์ตัวกำเนิดจากรหัสแอลดีพีซีแบบไบนารี

การเข้ารหัสข้อความไบนารีเพื่อสร้างคำรัหสของรหัสแอลดีพีซีสามารถทำได้โดยทำการคูณข้อความไบนารีแบบเวกเตอร์กับเมทริกซ์ตัวกำเนิด $\mathbf{G}$ ซึ่งได้จากการทำเกาส์-จอร์แดน อิลิมิเนชัน ของเมทริกซ์พาริตีเช็กเพื่อให้อยู่ในรูปแบบสมการคือ $\mathbf{H} = [\mathbf{I_m} \mid \mathbf{P}]$ เมื่อ $\mathbf{I_m}$ คือเมทริกซ์เอกลักษณ์ขนาด $m \times m$ และ $\mathbf{P}$ คือเมทริกซ์พาริตีเช็ก ดังนั้นเมทริกซ์เจเนอเรเตอร์คือ $\mathbf{G} = [\mathbf{P^T} \mid \mathbf{I_k}]$ เมื่อ $\mathbf{P^T}$ คือทรานสโพสของเมทริกซ์ P และ $\mathbf{I_k}$ คือเมทริกซ์เอกลักษณ์ขนาด $k \times k$

$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \mathrm{H}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

เราจะเห็นได้ว่าแถวที่ 4 ในเมทริกซ์ $\mathbf{H}$ เป็นการทำเกาส์เซียนอิลิมิเนชั่นซึ่งรวมแบบมอดูโล 2 ของแถวที่ 1 แถวที่ 2 และแถวที่ 3 ทำให้ได้แถวที่ 4 มีค่าเป็นศูนย์ทั้งหมดในเมทริกซ์ $\mathbf{H'}$ อย่างไรก็ตามเราสามารถหาเมทริกซ์เจเนอเรเตอร์โดยการตัดแถวสุดท้ายที่เป็นศูนย์ทั้งหมดออกได้จะได้ว่า

$$\mathrm{H}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \mathrm{H}'' = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 1 & 1 \\ 0 & 1 & 0 & : & 1 & 1 & 1 \\ 0 & 0 & 1 & : & 1 & 0 & 1 \end{bmatrix}$$

เมทริกซ์พาริตีเช็ก $\mathbf{H''}$ สามารถหาได้โดยการทำแถวที่ 3 ของเมทริกซ์ $\mathbf{H'}$ รวมกับแถวที่ 2 ของเมทริกซ์ $\mathbf{H'}$ จากนั้นแทนผลลัพธ์ที่ได้ในแถวที่ 2 ดังนั้นเราจึงได้เมทริกซ์เจเนอเรเตอร์คือ

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

จะเห็นได้ว่าเกาส์-จอร์แดน อิลิมิเนชันช่วยเพิ่มอัตรารหัสให้กับรหัสแอลดีพีซีด้วยเนื่องจากเราสามารถตัดแถวที่มีค่าเป็นศูนย์ทั้งหมดออกไปได้ ทำให้รหัสแอลดีพีซีมีโค้ดเรทเท่ากับ 1/3 ซึ่งหลังจากทำเกาส์-จอร์แดน อิลิมิเนชั่น ทำให้ได้รหัสแอลดีพีซีมีอัตรารหัสเท่ากับ 1/2

2.1.4 การถอดรหัสแอลดีพีซีแบบไบนารีโดยใช้อัลกอริทึมการแพร่กระจายความเชื่อ

การถอดรหัสแอลดีพีซีด้วยอัลกอริทึมรวมผลคูณความน่าจะเป็นนำไปสู่การหาคำรหัสความยาวขนาด n เมื่อคำรหัส c_n แต่ละบิตที่ได้เข้ารหัสมีความน่าจะเป็นที่มากที่สุดจะได้ว่า [7, 8]

$$P(c_n \mid \{z_m = 0, m \in M_n\}) \tag{2.1}$$

อัลกอริทึมรวมผลคูณความน่าจะเป็นนำไปสู่การคำนวณความน่าจะเป็นสองค่า ค่าแรกคือความน่าจะเป็น q_{mn} ซึ่งก็คือความน่าจะเป็นของคำรหัสที่ n ซึ่งถูกเปลี่ยนแปลงค่าขึ้นอยู่กับพาริตีที่เชื่อมต่อยกเว้นพาริตีตัวที่ m สามารถเขียนเป็นเงื่อนไขทางคณิตศาสตร์ได้ดังนี้

$$q_{mn}(x) = P(c_n = x \mid \{z_m = 0, m' \in M_n / m\}) \tag{2.2}$$

ค่าความน่าจะเป็นตัวที่สองคือ r_{mn} ซึ่งคือค่าความน่าจะเป็นของเมทริกซ์พาริตีเช็กตัวที่ m ซึ่งเชื่อมต่อกับบิตที่ถูกเข้ารหัส c ทุกค่าที่เป็นไปได้สามารถเขียนเป็นเงื่อนไขทางคณิตศาสตร์ได้ดังนี้

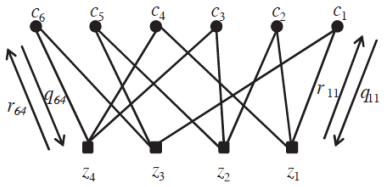
$$r_{mn}(x) = P(z_m = 0 \mid c) \tag{2.3}$$

รูปที่ 2.3 แสดงความเชื่อมโยงของการแลกเปลี่ยนความน่าจะเป็นภายในแทนเนอร์กราฟระหว่าง $c_1, z_1,$ และ z_4

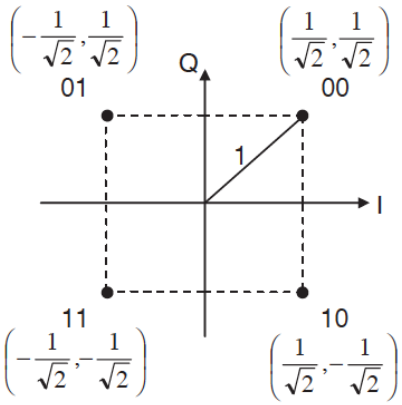
2.1.5 อัลกอริทึมการแพร่กระจายความน่าเชื่อถือของความน่าจะเป็น

ความน่าจะเป็น $q_{mn}(x)$ เป็นค่าเริ่มต้นไปสู่ค่าความน่าจะเป็น $f_n^{(x)}$ ของบิตภาครับตัวที่ n เท่ากับ x นั้นคือ $P(c_n = x)$ ดังนั้นสำหรับสัญญาณรบกวนเกาส์สีขาวแบบบวก (AWGN) ความน่าจะเป็นคือ [7, 8]

$$g_n^{(0)} \alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-(r_n-1)^2/2\sigma^2} \quad \text{และ} \quad g_n^{(1)} \alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-(r_n+1)^2/2\sigma^2} \tag{2.4}$$



รูปที่ 2.3 แทนเนอร์กราฟแสดงการแลกเปลี่ยนข้อมูลระหว่างโค้ดบิตและพาริตีตรวจสอบที่เชื่อมต่อกัน



รูปที่ 2.4 กลุ่มสัญญาณ QPSK

โดยปกติเมื่อรูปแบบการกล้ำสัญญาณมีลักษณะเฟสตรงแบบควอดเดรเจอร์จะสามารถหาความน่าจะเป็นโดยการคำนวณระยะทางยูคลิกสำหรับแต่ละจุดกลุ่มสัญญาณที่เฟสตรงกันดังรูปที่ 2.4 ดังนั้นความน่าจะเป็น $g_n^{(00)}, g_n^{(01)}, g_n^{(11)}$, และ $g_n^{(10)}$ ของกลุ่มสัญญาณกล้ำแบบนี้คือ QPSK

$$
\begin{aligned}
g_n^{(00)} &\alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(r_n^{(I)}-\frac{1}{\sqrt{2}}\right)^2+\left(r_n^{(Q)}-\frac{1}{\sqrt{2}}\right)^2\right]/2\sigma^2}, \\
g_n^{(01)} &\alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(r_n^{(I)}-\frac{1}{\sqrt{2}}\right)^2+\left(r_n^{(Q)}+\frac{1}{\sqrt{2}}\right)^2\right]/2\sigma^2}, \\
g_n^{(11)} &\alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(r_n^{(I)}+\frac{1}{\sqrt{2}}\right)^2+\left(r_n^{(Q)}+\frac{1}{\sqrt{2}}\right)^2\right]/2\sigma^2}, \\
g_n^{(10)} &\alpha \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\left(r_n^{(I)}+\frac{1}{\sqrt{2}}\right)^2+\left(r_n^{(Q)}-\frac{1}{\sqrt{2}}\right)^2\right]/2\sigma^2}
\end{aligned}
\qquad (2.5)
$$

เมื่อ $r_n^{(I)}$ และ $r_n^{(Q)}$ เป็นสัญญาณภาครับแบบควอดเดรเจอร์และมีเฟสตรงกัน ความน่าจะเป็นของบิตภาครับแทนในเวกเตอร์ $\mathbf{f}$ จะได้ว่า

$$
\mathbf{f} = \begin{bmatrix} g_1^{(0)} & g_2^{(0)} & \cdots & g_{n-1}^{(0)} & g_n^{(0)} \\ g_1^{(0)} & g_2^{(0)} & \cdots & g_{n-1}^{(0)} & g_n^{(0)} \end{bmatrix} \qquad (2.6)
$$

เริ่มต้นการคำนวณอัลกอริทึมการรวมผลคูณของความน่าจะเป็น ค่าความเป็นไปได้เริ่มต้น $\mathbf{f}$ ถูกใช้เป็นค่าเริ่มต้นสำหรับเมทริกซ์ Q โดยจะได้ว่า

$$\mathrm{Q}=\begin{bmatrix} g_1^{(0)} & g_2^{(0)} & g_3^{(0)} & g_4^{(0)} & g_1^{(1)} & g_1^{(1)} \\ g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & g_4^{(1)} & g_1^{(1)} & g_1^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ g_1^{(0)} & g_2^{(0)} & g_3^{(0)} & g_4^{(0)} & g_1^{(1)} & g_1^{(1)} \\ g_1^{(1)} & g_2^{(1)} & g_3^{(1)} & g_1^{(1)} & g_1^{(1)} & g_1^{(1)} \end{bmatrix} \tag{2.7}$$

เมทริกซ์ $\mathbf{Q}$ ใส่ค่าความเป็นไปได้ของบิตภาครับที่ i คือ $0, g_i^{(0)}$ หรือ $1, g_i^{(1)}$ เมื่อเมทริกซ์ Q ได้รับค่าไลค์ลิฮูดจาก f แล้วขั้นตอนต่อไปคือการหาค่าความน่าจะเป็น r_{mn} โดยใช้ขั้นตอนที่เรียกว่า Horizontal step ซึ่งนิยามดังนี้ [7, 8]

Horizontal Step :

$$r_{mn(x)}=\sum_{c:c_n=x} P(z_m=0\,|\,\mathrm{c})P(\mathrm{c}\,|\,c_n=x) \tag{2.8}$$

ให้โค้ดบิตที่ n แทนด้วย $c_n=x,\ \mathbf{c}$ เป็นเวกเตอร์ไบนารีของบิตรหัสที่เหลือที่มีขนาด $r-1$ ซึ่งประกอบไปด้วยการรวามกันที่เป็นไปได้ของลำดับบิต ดังนั้นมีจะมีลำดับบิตเท่ากับ 2^{r-1} รวมกัน อย่างไรก็ตามไม่ใช่ทุกลำดับบิตจะเป็นไปตามเงื่อนไขที่ว่า $z_m=0$ เราจะต้องพิจารณาลำดับที่เป็นไปตามเงื่อนไขของเมทริกซ์พาริตีเช็กดังนั้นความน่าจะเป็นของลำดับบิตซึ่งประกอบไปด้วยจำนวนคี่ของ 1 ที่เป็นไปตามเงื่อนไข $z_m=0$ คือศูนย์ ทำให้ $r_{mn}(x)$ คือผลรวมของทุกความน่าจะเป็นของลำดับไบนารีขนาด $r-1$ ซึ่งเท่ากับ x จะได้ว่า

$$r_{mn(x)}=\sum_{c:c_n=x} P(z_m=0\,|\,\mathrm{c})\prod_{n'\in N_m\backslash n} q_{mn'}(x) \tag{2.9}$$

เมื่อ $P(z_m=0\,|\,\mathbf{c})$ เป็นค่าได้ทั้ง 0 หรือ 1 ความน่าจะเป็น $r_{mn}(x)$ ถูกแทนในเมทริกซ์ $\mathbf{R}$ ดังนี้

$$\mathrm{R}=\begin{bmatrix} r_{11}(0) & r_{12}(0) & r_{13}(0) & \cdots & r_{1(n-1)}(0) & r_{1n}(0) \\ r_{11}(1) & r_{12}(1) & r_{13}(1) & \cdots & r_{1(n-1)}(1) & r_{1n}(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{m1}(0) & r_{m2}(0) & r_{m3}(0) & \cdots & r_{m(n-1)}(0) & r_{mn}(0) \\ r_{m1}(1) & r_{m2}(1) & r_{m3}(1) & \cdots & r_{m(n-1)}(1) & r_{mn}(1) \end{bmatrix} \tag{2.10}$$

เมื่อแต่ละสมาชิกในเมทริกซ์พาริตีเช็กมีขนาด 1×2 คอลัมน์เว็กเตอร์ซึ่งมีความน่าจะเป็นเท่ากับ $r_{mn}(0)$ และ $r_{mn}(1)$ [7, 8]

Vertical step : ทำหน้าอัพเดทค่าความน่าจะเป็น q_{mn} สามารถแสดงโดยกฎของเบย์ดังนี้

$$q_{mn}(x) = P(c_n = x \mid \{z_m = 0, m' \in M_n / m\}$$
$$= \frac{P(c_n = x)P(\{z_m = 0, m' \in M_n / m\} \mid c_n = x)}{P(\{z_m = 0, m' \in M_n / m\})} \quad (2.11)$$

ใช้สมการ (2.3) และให้ $f_n^x = P(c_n = x)$ สมการ (2.7) สามารถเขียนอีกอย่างได้คือ [7]

$$q_{mn}(x) = \beta_{mn} f_n^x \prod_{m' \in M_n / m} r_{m'n}(x) \quad (2.12)$$

เมื่อ b_{mn} เป็นค่าคงที่นอร์มอลไลซ์เพื่อทำให้ $\sum q_{mn}(x) = 1$

$$\beta_{mn} = \frac{1}{\sum_x f_n^x \prod_{m' \in M_n \setminus m} r_{m'n}(x)} \quad (2.13)$$

แทนค่า $q_{mn}(x)$ ในสมการ (2.14)

$$\mathrm{Q} = \begin{bmatrix} q_{11}(0) & q_{12}(0) & q_{13}(0) & \cdots & q_{1(n-1)}(0) & q_{1n}(0) \\ q_{11}(1) & q_{12}(1) & q_{13}(1) & \cdots & q_{1(n-1)}(1) & q_{1n}(1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{m1}(0) & q_{m2}(0) & q_{m3}(0) & \cdots & q_{m(n-1)}(0) & q_{mn}(0) \\ q_{m1}(1) & q_{m2}(1) & q_{m3}(1) & \cdots & q_{m(n-1)}(1) & q_{mn}(1) \end{bmatrix} \quad (2.14)$$

แต่ละอิลิเม้นท์ในพาริตีเมทริกซ์ประกอบไปด้วย 1×2 คอลัมน์เว็กเตอร์ซึ่งเก็บค่าความน่าจะเป็น $q_{mn}(0)$ และ $q_{mn}(1)$ (หรือเท่ากับ $1 \times q$ คอลัมน์เว็กเตอร์สำหรับรหัสแอลดีพีซีแบบนอนไบนารีบนสนามกาลัวส์ $\mathrm{GF}(q)$)

ในขั้นตอนนี้ pseudo posterior probabilities หาได้จาก [7, 8]

$$q_{mn}(x) = \beta_{mn} f_n^x \prod_{m' \in M_n} r_{mn}(x) \quad (2.15)$$

เมื่อ b_n เป็นค่าคงที่นอร์มอลไลซ์เพื่อทำให้ $\sum q_{mn}(x) = 1$ เมื่อแทนค่า pseudo posterior probabilities ในเมทริกซ์ Q

$$\mathrm{Q}' = \begin{bmatrix} q_1(0) & q_2(0) & q_3(0) & \cdots & q_{n-1}(0) & q_n(0) \\ q_1(1) & q_2(1) & q_3(1) & \cdots & q_{n-1}(1) & q_n(1) \end{bmatrix} \quad (2.16)$$

จากค่า pseudo posterior probabilities สามารถหาการประมาณค่าความน่าจะเป็นของคำที่ถูกส่งได้จาก

$$c_n = \arg \max_x \beta_n f_n^x \prod_{m \in N_m} r_{mn}(x) \quad (2.17)$$

2.2 การลดความซับซ้อนการถอดรหัสแอลดีพีซีโดยใช้การแปลงฟูเรียร์แบบเร็ว

ถ้าให้สมการพาริตีตรวจสอบ $z_1 = c_1 \oplus c_2 \oplus c_3 = 0$ ดังนั้นเพื่อที่จะหา $r_{11}(x)$ อย่างแรกที่เราต้องหาคำตอบของ $c_2 \oplus c_3 = c_1 = x$ สำหรับ $x = 0$ จะได้ว่า $c_2 = 0, c_3 = 0$ และ $c_2 = 1, c_3 = 1$ ดังนั้น

$$r_{11}(0) = q_{12}(0)q_{13}(0) + q_{12}(1)q_{13}(1)$$

สำหรับ $x = 1$ จะได้ว่า $c_2 = 0, c_3 = 1$ และ $c_2 = 1, c_3 = 0$ ดังนั้น

$$r_{11}(1) = q_{12}(0)q_{13}(1) + q_{12}(1)q_{13}(0)$$

โดยทั่วไปแล้วเราสามารถเขียนสมการด้านบนแบบคอนโวลูชันได้คือ

$$r_{11}(x) = \sum_{v=0}^{1} q_{12}(v)q_{13}(x-v) \quad \text{เมื่อ} \quad v \in \mathrm{GF}(2)$$

กล่าวเป็นอีกนัยได้ว่าเราสามารถหาผลลัพธ์ที่เหมือนกันกับของการทำคอนโวลูชันด้วยการแปลงฟูเรียร์ ดังนั้นเพื่อหา r_{mn} เราจะต้องคำนวณผลลัพธ์ของ**การแปลงฟูเรียร์**จาก $q_{mn'}$ ตัวอื่นๆ จากนั้นจึงใช้การแปลงฟูเรียร์ผกผัน

$$r_{mn}(x) = F^{-1}\left(\prod_{n' \in N_m / n} F(q_{mn'}(x)) \right) \tag{2.18}$$

เมื่อ $F(\,)$ คือการแปลงฟูเรียร์และ F^{-1} คือการแปลงฟูเรียร์ผกผันเนื่องจากสมาชิกทุกตัวอยู่ในกลุ่มบวกของ W_2 ทำให้การแปลงฟูเรียร์ลดรูปเป็นการแปลงฮาดามาร์ด [9]

$$\mathbf{W}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.19}$$

คุณสมบัติของเมทริกซ์ฮาดามาร์ดคือค่าผกผันของมันยังคงเป็นเมทริกซ์ฮาดามาร์ดตัวเดิม

$$\mathbf{W}_2 \mathbf{W}_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \mathbf{I}$$

เมื่อ $\mathbf{I}$ คือเมทริกซ์เอกลักษณ์ทำให้สามารถหาการแปลงฟูเรียร์ผกผันจากการคูณโดยใช้เมทริกซ์ฮาดามาร์ด

2.3 การสร้างรหัสแอลดีพีซีแบบนอนไบนารีซึ่งนิยามอยู่บนสนามจำกัด

รหัสแอลดีพีซีแบบนอนไบนารีคือรหัสแอลดีพีซีที่มีเมทริกซ์พาริตีเช็กประกอบไปด้วยสมาชิกที่นิยามเป็นกลุ่ม วงแหวน หรือบนสนาม ซึ่งสิ่งที่นำมาใช้คือรหัสแอลดีพีซีที่นิยามบนสนามจำกัด $\mathrm{GF}(2^i)$ ที่ i จำนวนเต็มบวกมากกว่า 1 ในปี ค.ศ.1998 แมคเคย์ได้นำเสนอรหัสแอลดีพีซีบนสนาม

จำกัด [10] ที่ช่วยเพิ่มสมรรถนะเหนือกว่ารหัสแอลดีพีซีแบบไบนารีด้วยการเพิ่มขนาดของสนามจำกัด นอกจากนี้แมคเคย์ยังได้แสดงวิธีการถอดรหัสแอลดีพีซีแบบนอนไบนารีด้วยอัลกอริทึมการรวมผลคูณความน่าจะเป็นแต่จะทำให้ความซับซ้อนโดยรวมเพิ่มสูงขึ้น

2.3.1 การสร้างรหัสแอลดีพีซีแบบนอนไบนารีจากรหัสรีด-โซโลมอน

ในบริบทของรหัสแอลดีพีซีแบบนอนไบนารีนิยามบนสนามจำกัด $\mathrm{GF}(q)$ [11] การกระจายอาเรย์คือการกระทำประยุกต์ของสมาชิกแต่ละตัวที่ไม่เป็นศูนย์ในเมทริกซ์โดยแต่ละสมาชิกถูกเปลี่ยนแปลงตำแหน่งเว็กเตอร์ความยาวเท่ากับ $q-1$ สำหรับสมาชิก $\alpha^i \in \mathrm{GF}(q), 0 \le i \le q-2$ จะถูกวางแทนไว้ตำแหน่งที่ i ของตำแหน่งเว็กเตอร์ ตัวอย่างเช่น สมาชิก α^5 ใน $\mathrm{GF}(8)$ จะถูกแทนค่าในตำแหน่งที่ 5ของตำแหน่งเว็กเตอร์ สมาชิกในตำแหน่งเว็กเตอร์นี้ถูกใช้เพื่อสร้างอาเรย์กับแต่ละแถวซึ่งถูกนิยามโดยการเลื่อนแถวไปทางขวาก่อนหน้านี้และคูณด้วยสมาชิกพริมิทีฟ $\mathrm{GF}(q)$ ในอาเรย์ขนาด $(q-1)\times(q-1)$ การทำการกระจายอาเรย์บนเมทริกซ์ขนาด $a\times b$ จะทำให้เกิดเมทริกซ์ขนาดใหญ่ขึ้นซึ่งมีขนาดเท่ากับ $a(q-1)\times b(q-1)$ ดังนั้นหลังจากกระจายอาเรย์สมาชิก α^5 จะได้ว่าที่แถวบนสุดคือตำแหน่งเว็กเตอร์เริ่มแรก สำหรับสมาชิกที่เป็นศูนย์การกระจายอาเรย์จะให้ผลลัพธ์เป็น $(q-1)\times(q-1)$ ของเมทริกซ์ศูนย์

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \alpha^5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha^6 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^4 & 0 & 0 \end{pmatrix}$$

รหัสแอลดีพีซีแบบนอนไบนารีที่สร้างจากรหัสรีด-โซโลมอนซึ่งมีขนาดความยาวของข้อความ $k=2$ จะมีระยะทางแฮมมิงต่ำสุด $d=n-k+1=n-1$ เนื่องจากรหัสรีด-โซโลมอนนิยามบน $\mathrm{GF}(q)$ มีความยาวบล็อค $n-q-1$ รหัสรีด-โซโลมอนจะเป็นรหัสรีด-โซโลมอน $(q-1,2,q-2)$ เนื่องจาก $d=q-2$ หมายถึงน้ำหนักต่ำสุดของคำรหัสเท่ากับ $q-2$ ด้วย หมายความว่าคำรหัสจะมีสมาชิกที่ไม่เป็นศูนย์เท่ากับ $q-2$ และสมาชิกที่เป็นศูนย์เท่ากับ 1

2.3.2 การถอดรหัสแอลดีพีซีแบบนอนไบนารีด้วยอัลกอริทึมรวมผลคูณความน่าจะเป็น

อัลกอริทึมรวมผลคูณความน่าจะเป็นสำหรับรหัสแอลดีพีซีแบบไบนารีสามารถขยายขอบเขตการถอดรหัสสำหรับรหัสแอลดีพีซีแบบนอนไบนารีได้แต่ความซับซ้อนในการถอดรหัสจะเพิ่มขึ้นประการ

แรกสำหรับรหัสแอลดีพีซีแบบนอนไบนารีนิยามบน $\mathrm{GF}(q)$ แต่ละสัญลักษณ์ที่ได้รับสามารถเป็นหนึ่งของ q สมาชิกที่ต่างกันใน $\mathrm{GF}(q)$ ประการที่สอง **horizontal step** จะมีความซับซ้อนโดยจะมีลำดับเลขนอนไบนารีจำนวนมากเพื่อให้เป็นไปตามเงื่อนไขของสมการเมทริกซ์พาริตีเช็ก เมทริกซ์ $\mathbf{Q}$ และ $\mathbf{R}$ ถูกใช้ใน **horizontal step** และ **vertical step** ของอัลกอริทึมรวมผลคูณความน่าจะเป็นซึ่งถูกนิยามในสมการ (2.20) และ (2.21) ตามลำดับ

$$\mathrm{Q} = \begin{bmatrix} q_{11}(0) & q_{12}(0) & q_{13}(0) & \cdots & q_{1,n-1}(0) & q_{1n}(0) \\ q_{11}(1) & q_{12}(1) & q_{13}(1) & \cdots & q_{1,n-1}(1) & q_{1n}(1) \\ q_{11}(\alpha) & q_{12}(\alpha) & q_{13}(\alpha) & \cdots & q_{1,n-1}(\alpha) & q_{1n}(\alpha) \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ q_{11}(\alpha^{q-2}) & q_{12}(\alpha^{q-2}) & q_{13}(\alpha^{q-2}) & \cdots & q_{1,n-1}(\alpha^{q-2}) & q_{1n}(\alpha^{q-2}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{m1}(0) & q_{m2}(0) & q_{m3}(0) & \cdots & q_{m,n-1}(0) & q_{m,n}(0) \\ q_{m1}(1) & q_{m2}(1) & q_{m3}(1) & \cdots & q_{m,n-1}(1) & q_{m,n}(1) \\ q_{m1}(\alpha) & q_{m2}(\alpha) & q_{m3}(\alpha) & \cdots & q_{m,n-1}(\alpha) & q_{m,n}(\alpha) \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ q_{m1}(\alpha^{q-2}) & q_{m2}(\alpha^{q-2}) & q_{m3}(\alpha^{q-2}) & \cdots & q_{m,n-1}(\alpha^{q-2}) & q_{m,n}(\alpha^{q-2}) \end{bmatrix} \quad (2.20)$$

$$\mathrm{R} = \begin{bmatrix} r_{11}(0) & r_{12}(0) & r_{13}(0) & \cdots & r_{1,n-1}(0) & r_{1n}(0) \\ r_{11}(1) & r_{12}(1) & r_{13}(1) & \cdots & r_{1,n-1}(1) & r_{1n}(1) \\ r_{11}(\alpha) & r_{12}(\alpha) & r_{13}(\alpha) & \cdots & r_{1,n-1}(\alpha) & r_{1n}(\alpha) \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ r_{11}(\alpha^{q-2}) & r_{12}(\alpha^{q-2}) & r_{13}(\alpha^{q-2}) & \cdots & r_{1,n-1}(\alpha^{q-2}) & r_{1n}(\alpha^{q-2}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{m1}(0) & r_{m2}(0) & r_{m3}(0) & \cdots & r_{m,n-1}(0) & r_{m,n}(0) \\ r_{m1}(1) & r_{m2}(1) & r_{m3}(1) & \cdots & r_{m,n-1}(1) & r_{m,n}(1) \\ r_{m1}(\alpha) & r_{m2}(\alpha) & r_{m3}(\alpha) & \cdots & r_{m,n-1}(\alpha) & r_{m,n}(\alpha) \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ r_{m1}(\alpha^{q-2}) & r_{m2}(\alpha^{q-2}) & r_{m3}(\alpha^{q-2}) & \cdots & r_{m,n-1}(\alpha^{q-2}) & r_{m,n}(\alpha^{q-2}) \end{bmatrix} \quad (2.21)$$

สังเกตุว่าแต่ละสมาชิกที่ไม่เป็นศูนย์ถูกนิยามบน $\mathrm{GF}(q)$ในเมทริกซ์พาริตีเช็ก $\mathbf{H}$ มี q ค่าสำหรับความน่าจะเป็นสมาชิกแต่ละตัวด้วยสังเกตุว่าไม่ได้มีค่าความน่าจะเป็นสองค่าเหมือนในกรณีของรหัสแอลดีพีซีแบบไบนารี

2.3.3 การหาค่าความน่าจะเป็นของบิตสัญลักษณ์ที่ภาครับ

ในสมการ (2.5) ได้แสดงการหาค่าความน่าจะเป็นของสัญลักษณ์สำหรับช่องสื่อสารที่สัญญาณรบกวนเกาส์เซียนแถบขาวแบบบวก สมมติรหัสแอลดีพีซีแบบนอนไบนารีนิยามบน $\mathrm{GF}(q)$ และเลือกการกล้ำสัญญาณแบบ M-PSK แล้วให้ $q = M$ ค่าความน่าจะเป็นของสัญลักษณ์ที่ถอดการกล้ำสัญญาณจะเป็นค่าความน่าจะเป็นของสัญญาณที่รับได้ด้วย อย่างไรก็ตามสำหรับกรณีที่ $q > M$ และ M หาร q เราจะต้องรวมค่าความน่าจะเป็นของสัญลักษณ์ที่ได้จากการถอดการกล้ำสัญญาณ

2.3.4 การสับเปลี่ยนตำแหน่งของค่าความน่าจะเป็น

โดยทั่วไปสมการพาริตีเช็กจะอยู่ในรูปสมการ

$$z_i = \sum_{j=1}^{n} h_{ijc_j} \tag{2.22}$$

ที่ h_{ij} และ $c_j \in \mathrm{GF}(q)$ สมการพารตีตรวจสอบจะเป็นไปตามเงื่อนไขเมื่อ

$$h_{11}c_1 + h_{12}c_2 + \ldots + h_{1n}c_n = 0$$

สำหรับ horizontal step เราคำนวณความน่าจะเป็น $r_{ij}(x)$ จากสมการ (2.7) แทนค่านอนไบนารีทุกค่าที่เป็นไปได้ในรหัสสัญลักษณ์ที่เป็นไปตามเงื่อนไขของสมการพาริตีเช็กเมื่อ $c_j = x$ นั่นคือ

$$h_{i1}c_1 + h_{i2}c_2 + \ldots + h_{in}c_n = h_{ij}c_j$$

จากนั้นหาความน่าจะเป็นของลำดับโดยแต่ละสัญลักษณ์โค้ดมี q ค่าความน่าจะเป็น เมื่อคูณรหัสสัญลักษณ์กับพาริตีเช็กแบบนอนไบนารีเราสามารถทำการชดเชยโดยทำการเลื่อนเว็กเตอร์คอลัมน์ของค่าความน่าจะเป็นลงไปโดยไม่รวมค่าความน่าจะเป็นของแถวแรกด้วยความน่าจะเป็นที่รหัสสัญลักษณ์จะเป็นศูนย์จำนวนของการเลื่อนจะเท่ากับกำลังของจำนวนเฉพาะซึ่งคูณกับโค้ดสัญลักษณ์ซึ่งอธิบายได้ดังรูปที่ 2.4

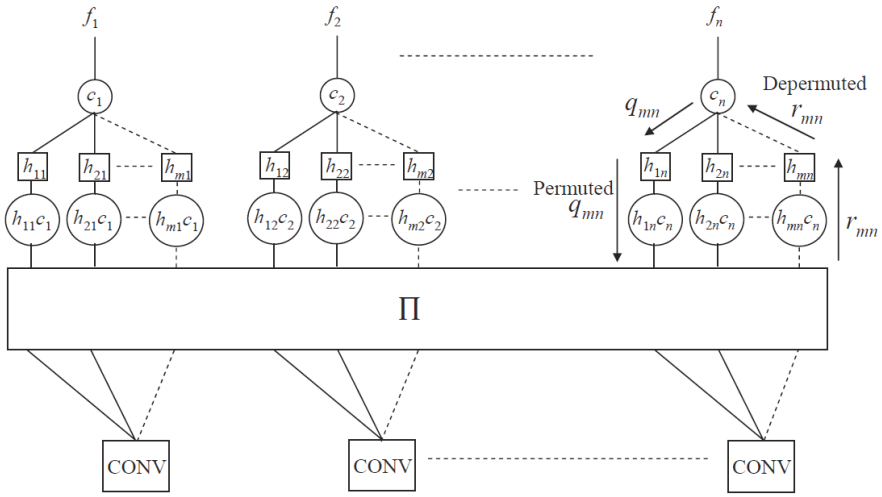
$$c_j \rightarrow \begin{bmatrix} q_{ij}(0) \\ q_{ij}(1) \\ \vdots \\ q_{ij}(\alpha^{q-2}) \end{bmatrix} \quad \alpha c_j \rightarrow \begin{bmatrix} q_{ij}(0) \\ q_{ij}(\alpha^{q-2}) \\ q_{ij}(1) \\ \vdots \end{bmatrix} \quad \alpha^2 c_j \rightarrow \begin{bmatrix} q_{ij}(0) \\ \vdots \\ q_{ij}(\alpha^{q-2}) \\ q_{ij}(1) \end{bmatrix}$$

รูปที่ 2.5 การเลื่อนวนรอบของค่าความน่าจะเป็น

การเลื่อนวนรอบของความน่าจะเป็นเรียกว่า “การสับเปลี่ยนลำดับ” [9] และแปลงสมการพาริตีตรวจสอบจากสมการ (2.22) เป็น

$$c_1 + c_2 + \ldots + c_n = c_j$$

ซึ่งคล้ายกับสมการพาริตีตรวจสอบของไบนารีเป็นอย่างมาก ค่าการสับเปลี่ยนลำดับผกผันคือที่ค่าความน่าจะเป็นที่ถูกเลื่อนวนรอบขึ้นโดยไม่รวมค่าในแถวแรกของค่าความน่าจะเป็น



รูปที่ 2.6 กราฟส่วนประกอบทั่วไปของรหัสแอลดีพีซีแบบนอนไบนารี [8]

2.3.5 กราฟส่วนประกอบของรหัสแอลดีพีซีแบบนอนไบนารี

กราฟส่วนประกอบของรัหสแอลดีพีซีแบบนอนไบนารีคือขั้นตอนการถอดรหัสโดยใช้อัลกอริทึมการรวมผลคูณของความน่าจะเป็นโดยแสดงแบบกราฟดังรูปที่ 2.5

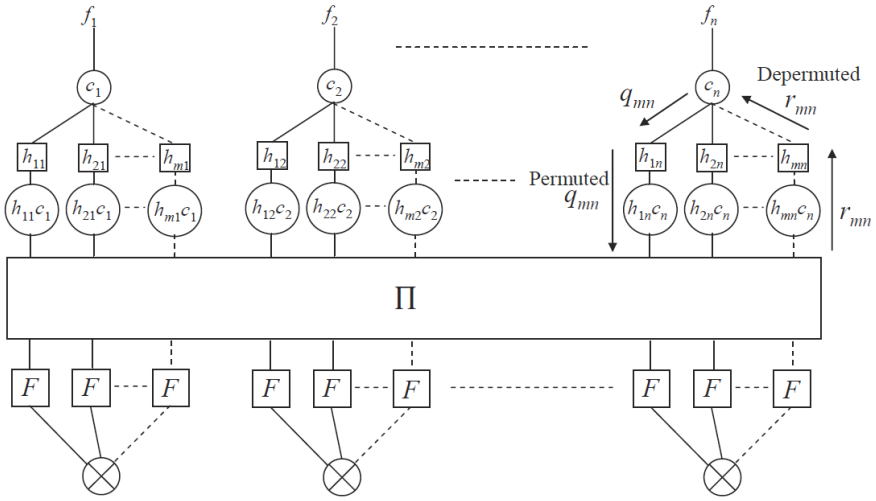
กราฟส่วนประกอบทั่วไปสำหรับรหัสแอลดีพีซีแบบนอนไบนารีจะคล้ายกับกราฟของรหัสแอลดีพีซีแบบไบนารีแต่เราจะต้องใช้จำนวนที่เป็นนอนไบนารีสำหรับเมทริกซ์พาริตีเช็กด้วยซึ่งถูกนิยามเป็น $h_{ij}, i = 1,2,\ldots,m$ และ $j = 1,2,\ldots,n$ ในรูปที่ 2.5 จำนวนสมาชิกของเมทริกซ์พาริตีเช็กที่เชื่อมต่อกับรหัสสัญลักษณ์ c_j คือน้ำหนักคอลัมน์ของรหัสและจำนวนของการเชื่อมต่อไปแต่ละพาริตีเช็ก z_i คือน้ำหนักแถวของรหัส ค่าความน่าจะเป็นของแต่ละรหัสสัญลักษณ์ f_j เป็นเวกเตอร์คอลัมน์ที่มี q ค่าความน่าจะเป็นของรหัสสัญลักษณ์ที่เป็นสมาชิกใน GF(q) บล็อก Π เชื่อมต่อกับสมาชิกนอนไบนารีในแต่ละแถวกับพาริตีเช็ก ในรูปที่ 2.6 คือกราฟส่วนประกอบที่แทนกล่องการทำคอนโวลูชันด้วยกล่องการแปลงฟูเรียร์แบบเร็ว (FFT) เพื่อลดความซับซ้อนในการประมวลผล

กราฟส่วนประกอบของเมทริกซ์พาริตีเช็กในสมการ (2.23) แสดงในรูปที่ 2.6

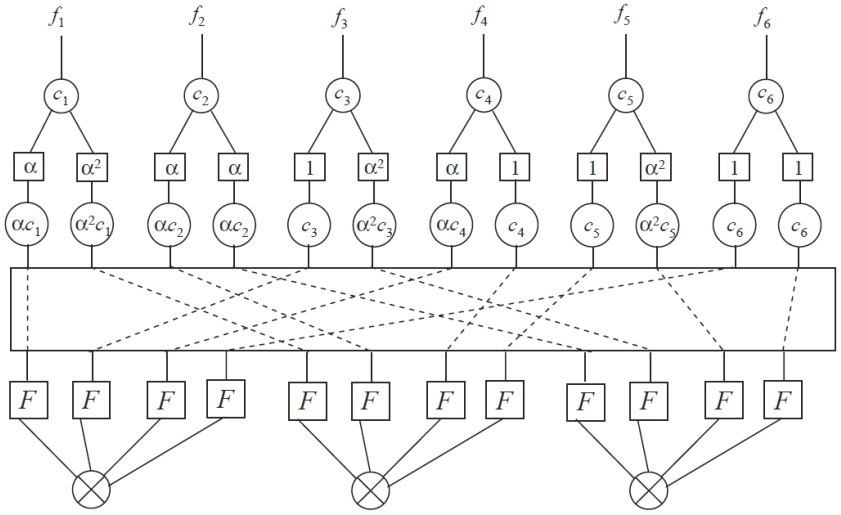
$$\mathbf{H} = \begin{bmatrix} \alpha & 0 & 1 & \alpha & 0 & 1 \\ \alpha^2 & \alpha & 0 & 1 & 1 & 0 \\ 0 & \alpha & \alpha^2 & 0 & \alpha^2 & 1 \end{bmatrix} \tag{2.23}$$

2.3.6 การแปลงฟูเรียร์แบบเร็วสำหรับการถอดรหัสแอลดีพีซีแบบนอนไบนารี

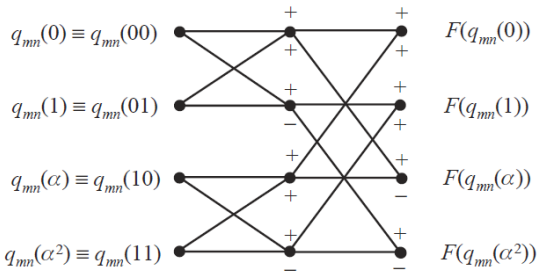
รหัสแอลดีพีซีแบบนอนไบนารีบนสนามกาลัวส์ GF(q) ขนาดของ $\mathbf{Q}$ เท่ากับ $1\times q$ คอลัมน์เว็คเตอร์และค่า Fourier transform ได้จากการคูณจาก tensor product ของ Hadamard matrices



รูปที่ 2.7 กราฟส่วนประกอบทั่วไปของรหัสแอลดีพีซีแบบนอนไบนารีซึ่งแทนบล็อคคอลโวลูชันด้วยการแปลงฟูเรียร์แบบเร็ว [8]



รูปที่ 2.8 กราฟส่วนประกอบสำหรับเมทริกซ์พาริตีเช็กของสมการ (2.23) [8]



รูปที่ 2.9 แผนภาพผีเสื้อแบบ Radix-2 สำหรับ GF(4) [8]

เนื่องจากการถอดรหัสแอลดีพีซีแบบนอนไบนารีมีความซับซ้อนในการถอดรหัสมากขึ้นตามขนาดของสนามกาลัวส์ที่เพิ่มขึ้น จึงมีการใช้เมทริกซ์ฮาดามาร์ดเข้ามาช่วยลดความซับซ้อนในการถอดรหัส

นิยามให้เมทริกซ์ฮาดามาร์ดเริ่มต้น

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

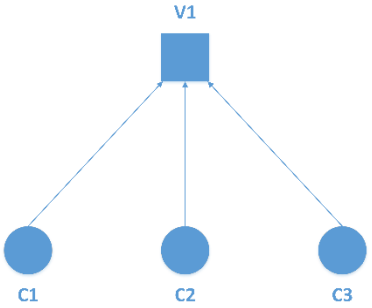
ดังนั้นเราสามารถสร้างเมทริกซ์ฮาดามาร์ด $\mathbf{W}_4$ ซึ่งมาจากการแปลงฮาดามาร์ด $\mathbf{W}_2$

$$\mathbf{H}_4 = \mathbf{H}_2 \otimes \mathbf{H}_2 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

สำหรับรหัสแอลดีพีซีบนสนามกาลัวส์ GF(4) การแปลงฟูเรียร์ของเว็กเตอร์ขนาด 1×4 คอลัมน์ของความน่าจะเป็น $q_{mn}(0)$, $q_{mn}(1)$, $q_{mn}(\alpha)$ และ $q_{mn}(\alpha^2)$ คือ

$$\begin{bmatrix} F(q_{mn}(0)) \\ F(q_{mn}(1)) \\ F(q_{mn}(\alpha)) \\ F(q_{mn}(\alpha^2)) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} q_{mn}(0) \\ q_{mn}(1) \\ q_{mn}(\alpha) \\ q_{mn}(\alpha^2) \end{bmatrix}$$

ก่อนหน้านี้การแปลงฟูเรียร์ผกผันแบบเร็วได้มาจากการคูณเมทริกซ์ฮาดามาร์ด $\mathbf{W_4W_4 = I_4}$ การแปลงฟูเรียร์ผกผันสามารถแสดงในแบบแผนภาพผีเสื้อแบบ Radix-2 ดังรูปที่ 2.8 เห็นได้ว่าลำดับของความน่าจะเป็น $q_{mn}(x)$ มีความสำคัญอย่างมากในการหาค่าการแปลงฟูเรียร์ผกผันในแต่ละคู่ของ $q_{mn}(x)$ ซึ่งนำไปสู่การสร้างตารางการบวกสำหรับรหัสแอลดีพีซีบนสนามกาลัวส์



รูปที่ 2.10 กราฟแทนเนอร์โหนดข้อมูล C_1, C_2 และ C_3 ส่งข้อมูลให้โหนดตรวจสอบ V_1

เพื่อให้การเห็นภาพการนำเมทริกซ์ฮาดามาร์ดมาช่วยเพิ่มความเร็วในการประมวลผลการถอดรหัสแอลดีพีซีแบบนอนไบนารี สมมติให้กราฟแทนเนอร์ในรูปที่ 2.9 แสดงการถอดรหัสแอลดีพีซีแบบนอนไบนารีซึ่งนิยามบนสนามกาลัวส์ $GF(4)$ โหนดบิต C_1, C_2, และ C_3 จะส่งข้อมูลเพื่อให้โหนดตรวจสอบ V_1 ตรวจสอบซึ่งต้องเป็นไปตามเงื่อนไขของสมการ (2.22) โดยที่โหนดตรวจสอบแต่ละโหนดมีค่าความน่าจะเป็นเท่ากับ $P(0)$, $P(1)$, $P(\alpha)$ และ $P(\alpha^2)$ เราสามารถหาค่าการแปลงฟูเรียร์และแปลงฟูเรียร์ผกผันเพื่อหาความน่าจะเป็นของแต่ละโหนดบิตได้ดังนี้

$$F(C_1)=\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}P(C_1=0)\\ P(C_1=1)\\ P(C_1=\alpha)\\ P(C_1=\alpha^2)\end{bmatrix}=\frac{1}{2}\begin{bmatrix}P(C_1=0)+P(C_1=1)+P(C_1=\alpha)+P(C_1=\alpha^2)\\ P(C_1=0)-P(C_1=1)+P(C_1=\alpha)-P(C_1=\alpha^2)\\ P(C_1=0)+P(C_1=1)-P(C_1=\alpha)-P(C_1=\alpha^2)\\ P(C_1=0)-P(C_1=1)-P(C_1=\alpha)+P(C_1=\alpha^2)\end{bmatrix}$$

จาก $C_1 = C_2+C_3$ จะได้ว่า

$$C_1=\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}
\begin{matrix}P(C_2=0)P(C_3=0)+P(C_2=0)P(C_3=1)+P(C_2=0)P(C_3=\alpha)+P(C_2=0)P(C_3=\alpha^2)\\ +P(C_2=1)P(C_3=0)+P(C_2=1)P(C_3=1)+P(C_2=1)P(C_3=\alpha)+P(C_2=1)P(C_3=\alpha^2)\\ +P(C_2=\alpha)P(C_3=0)+P(C_2=\alpha)P(C_3=1)+P(C_2=\alpha)P(C_3=\alpha)+P(C_2=\alpha)P(C_3=\alpha^2)\\ +P(C_2=\alpha^2)P(C_3=0)+P(C_2=\alpha^2)P(C_3=1)+P(C_2=\alpha^2)P(C_3=\alpha)+P(C_2=\alpha^2)P(C_3=\alpha^2)\end{matrix}\\
\begin{matrix}P(C_2=0)P(C_3=0)-P(C_2=0)P(C_3=1)+P(C_2=0)P(C_3=\alpha)-P(C_2=0)P(C_3=\alpha^2)\\ -P(C_2=1)P(C_3=0)+P(C_2=1)P(C_3=1)-P(C_2=1)P(C_3=\alpha)+P(C_2=1)P(C_3=\alpha^2)\\ +P(C_2=\alpha)P(C_3=0)-P(C_2=\alpha)P(C_3=1)+P(C_2=\alpha)P(C_3=\alpha)-P(C_2=\alpha)P(C_3=\alpha^2)\\ -P(C_2=\alpha^2)P(C_3=0)+P(C_2=\alpha^2)P(C_3=1)-P(C_2=\alpha^2)P(C_3=\alpha)+P(C_2=\alpha^2)P(C_3=\alpha^2)\end{matrix}\\
\begin{matrix}P(C_2=0)P(C_3=0)+P(C_2=0)P(C_3=1)-P(C_2=0)P(C_3=\alpha)-P(C_2=0)P(C_3=\alpha^2)\\ +P(C_2=1)P(C_3=0)+P(C_2=1)P(C_3=1)-P(C_2=1)P(C_3=\alpha)-P(C_2=1)P(C_3=\alpha^2)\\ -P(C_2=\alpha)P(C_3=0)-P(C_2=\alpha)P(C_3=1)+P(C_2=\alpha)P(C_3=\alpha)+P(C_2=\alpha)P(C_3=\alpha^2)\\ -P(C_2=\alpha^2)P(C_3=0)-P(C_2=\alpha^2)P(C_3=1)+P(C_2=\alpha^2)P(C_3=\alpha)+P(C_2=\alpha^2)P(C_3=\alpha^2)\end{matrix}\\
\begin{matrix}P(C_2=0)P(C_3=0)-P(C_2=0)P(C_3=1)-P(C_2=0)P(C_3=\alpha)+P(C_2=0)P(C_3=\alpha^2)\\ -P(C_2=1)P(C_3=0)+P(C_2=1)P(C_3=1)+P(C_2=1)P(C_3=\alpha)-P(C_2=1)P(C_3=\alpha^2)\\ -P(C_2=\alpha)P(C_3=0)+P(C_2=\alpha)P(C_3=1)+P(C_2=\alpha)P(C_3=\alpha)-P(C_2=\alpha)P(C_3=\alpha^2)\\ +P(C_2=\alpha^2)P(C_3=0)-P(C_2=\alpha^2)P(C_3=1)-P(C_2=\alpha^2)P(C_3=\alpha)+P(C_2=\alpha^2)P(C_3=\alpha^2)\end{matrix}
\end{bmatrix}$$

$$C_1=\begin{bmatrix}P(C_1=0)\\ P(C_1=1)\\ P(C_1=\alpha)\\ P(C_1=\alpha^2)\end{bmatrix}=\begin{bmatrix}P(C_2=0)P(C_3=0)+P(C_2=1)P(C_3=1)+P(C_2=\alpha)P(C_3=\alpha)+P(C_2=\alpha^2)P(C_3=\alpha^2)\\ P(C_2=0)P(C_3=1)+P(C_2=1)P(C_3=0)+P(C_2=\alpha)P(C_3=\alpha^2)+P(C_2=\alpha^2)P(C_3=\alpha)\\ P(C_2=0)P(C_3=\alpha)+P(C_2=\alpha)P(C_3=0)+P(C_2=1)P(C_3=\alpha^2)+P(C_2=\alpha^2)P(C_3=1)\\ P(C_2=0)P(C_3=\alpha^2)+P(C_2=\alpha^2)P(C_3=0)+P(C_2=1)P(C_3=\alpha)+P(C_2=\alpha)P(C_3=1)\end{bmatrix}$$

$$F(C_2)=\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}P(C_2=0)\\ P(C_2=1)\\ P(C_2=\alpha)\\ P(C_2=\alpha^2)\end{bmatrix}=\frac{1}{2}\begin{bmatrix}P(C_2=0)+P(C_2=1)+P(C_2=\alpha)+P(C_2=\alpha^2)\\ P(C_2=0)-P(C_2=1)+P(C_2=\alpha)-P(C_2=\alpha^2)\\ P(C_2=0)+P(C_2=1)-P(C_2=\alpha)-P(C_2=\alpha^2)\\ P(C_2=0)-P(C_2=1)-P(C_2=\alpha)+P(C_2=\alpha^2)\end{bmatrix}$$

ในทำนองเดียวกัน C_2 หาได้จาก $C_2 = C_1+C_3$

$$C_2=\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}P(C_1=0)P(C_3=0)+P(C_1=0)P(C_3=1)+P(C_1=0)P(C_3=\alpha)+P(C_1=0)P(C_3=\alpha^2)\\ +P(C_1=1)P(C_3=0)+P(C_1=1)P(C_3=1)+P(C_1=1)P(C_3=\alpha)+P(C_1=1)P(C_3=\alpha^2)\\ +P(C_1=\alpha)P(C_3=0)+P(C_1=\alpha)P(C_3=1)+P(C_1=\alpha)P(C_3=\alpha)+P(C_1=\alpha)P(C_3=\alpha^2)\\ +P(C_1=\alpha^2)P(C_3=0)+P(C_1=\alpha^2)P(C_3=1)+P(C_1=\alpha^2)P(C_3=\alpha)+P(C_1=\alpha^2)P(C_3=\alpha^2)\\ \\ P(C_1=0)P(C_3=0)-P(C_1=0)P(C_3=1)+P(C_1=0)P(C_3=\alpha)-P(C_1=0)P(C_3=\alpha^2)\\ -P(C_1=1)P(C_3=0)+P(C_1=1)P(C_3=1)-P(C_1=1)P(C_3=\alpha)+P(C_1=1)P(C_3=\alpha^2)\\ +P(C_1=\alpha)P(C_3=0)-P(C_1=\alpha)P(C_3=1)+P(C_1=\alpha)P(C_3=\alpha)-P(C_1=\alpha)P(C_3=\alpha^2)\\ -P(C_1=\alpha^2)P(C_3=0)+P(C_1=\alpha^2)P(C_3=1)-P(C_1=\alpha^2)P(C_3=\alpha)+P(C_1=\alpha^2)P(C_3=\alpha^2)\\ \\ P(C_1=0)P(C_3=0)+P(C_1=0)P(C_3=1)-P(C_1=0)P(C_3=\alpha)-P(C_1=0)P(C_3=\alpha^2)\\ +P(C_1=1)P(C_3=0)+P(C_1=1)P(C_3=1)-P(C_1=1)P(C_3=\alpha)-P(C_1=1)P(C_3=\alpha^2)\\ -P(C_1=\alpha)P(C_3=0)-P(C_1=\alpha)P(C_3=1)+P(C_1=\alpha)P(C_3=\alpha)+P(C_1=\alpha)P(C_3=\alpha^2)\\ -P(C_1=\alpha^2)P(C_3=0)-P(C_1=\alpha^2)P(C_3=1)+P(C_1=\alpha^2)P(C_3=\alpha)+P(C_1=\alpha^2)P(C_3=\alpha^2)\\ \\ P(C_1=0)P(C_3=0)-P(C_1=0)P(C_3=1)-P(C_1=0)P(C_3=\alpha)+P(C_1=0)P(C_3=\alpha^2)\\ -P(C_1=1)P(C_3=0)+P(C_1=1)P(C_3=1)+P(C_1=1)P(C_3=\alpha)-P(C_1=1)P(C_3=\alpha^2)\\ -P(C_1=\alpha)P(C_3=0)+P(C_1=\alpha)P(C_3=1)+P(C_1=\alpha)P(C_3=\alpha)-P(C_1=\alpha)P(C_3=\alpha^2)\\ +P(C_1=\alpha^2)P(C_3=0)-P(C_1=\alpha^2)P(C_3=1)-P(C_1=\alpha^2)P(C_3=\alpha)+P(C_1=\alpha^2)P(C_3=\alpha^2)\end{bmatrix}$$

$$C_2=\begin{bmatrix}P(C_2=0)\\ P(C_2=1)\\ P(C_2=\alpha)\\ P(C_2=\alpha^2)\end{bmatrix}=\begin{bmatrix}P(C_1=0)P(C_3=0)+P(C_1=1)P(C_3=1)+P(C_1=\alpha)P(C_3=\alpha)+P(C_1=\alpha^2)P(C_3=\alpha^2)\\ P(C_1=0)P(C_3=1)+P(C_1=1)P(C_3=0)+P(C_1=\alpha)P(C_3=\alpha^2)+P(C_1=\alpha^2)P(C_3=\alpha)\\ P(C_1=0)P(C_3=\alpha)+P(C_1=\alpha)P(C_3=0)+P(C_1=1)P(C_3=\alpha^2)+P(C_1=\alpha^2)P(C_3=1)\\ P(C_1=0)P(C_3=\alpha^2)+P(C_1=\alpha^2)P(C_3=0)+P(C_1=1)P(C_3=\alpha)+P(C_1=\alpha)P(C_3=1)\end{bmatrix}$$

$$F(C_3)=\frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}P(C_3=0)\\ P(C_3=1)\\ P(C_3=\alpha)\\ P(C_3=\alpha^2)\end{bmatrix}=\frac{1}{2}\begin{bmatrix}P(C_3=0)+P(C_3=1)+P(C_3=\alpha)+P(C_3=\alpha^2)\\ P(C_3=0)-P(C_3=1)+P(C_3=\alpha)-P(C_3=\alpha^2)\\ P(C_3=0)+P(C_3=1)-P(C_3=\alpha)-P(C_3=\alpha^2)\\ P(C_3=0)-P(C_3=1)-P(C_3=\alpha)+P(C_3=\alpha^2)\end{bmatrix}$$

และ C_3 หาได้จาก $C_3 = C_1+C_2$

$$C_3=\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}P(C_1=0)P(C_2=0)+P(C_1=0)P(C_2=1)+P(C_1=0)P(C_2=\alpha)+P(C_1=0)P(C_2=\alpha^2)\\ +P(C_1=1)P(C_2=0)+P(C_1=1)P(C_2=1)+P(C_1=1)P(C_2=\alpha)+P(C_1=1)P(C_2=\alpha^2)\\ +P(C_1=\alpha)P(C_2=0)+P(C_1=\alpha)P(C_2=1)+P(C_1=\alpha)P(C_2=\alpha)+P(C_1=\alpha)P(C_2=\alpha^2)\\ +P(C_1=\alpha^2)P(C_2=0)+P(C_1=\alpha^2)P(C_2=1)+P(C_1=\alpha^2)P(C_2=\alpha)+P(C_1=\alpha^2)P(C_2=\alpha^2)\\ \\ P(C_1=0)P(C_2=0)-P(C_1=0)P(C_2=1)+P(C_1=0)P(C_2=\alpha)-P(C_1=0)P(C_2=\alpha^2)\\ -P(C_1=1)P(C_2=0)+P(C_1=1)P(C_2=1)-P(C_1=1)P(C_2=\alpha)+P(C_1=1)P(C_2=\alpha^2)\\ +P(C_1=\alpha)P(C_2=0)-P(C_1=\alpha)P(C_2=1)+P(C_1=\alpha)P(C_2=\alpha)-P(C_1=\alpha)P(C_2=\alpha^2)\\ -P(C_1=\alpha^2)P(C_2=0)+P(C_1=\alpha^2)P(C_2=1)-P(C_1=\alpha^2)P(C_2=\alpha)+P(C_1=\alpha^2)P(C_2=\alpha^2)\\ \\ P(C_1=0)P(C_2=0)+P(C_1=0)P(C_2=1)-P(C_1=0)P(C_2=\alpha)-P(C_1=0)P(C_2=\alpha^2)\\ +P(C_1=1)P(C_2=0)+P(C_1=1)P(C_2=1)-P(C_1=1)P(C_2=\alpha)-P(C_1=1)P(C_2=\alpha^2)\\ -P(C_1=\alpha)P(C_2=0)-P(C_1=\alpha)P(C_2=1)+P(C_1=\alpha)P(C_2=\alpha)+P(C_1=\alpha)P(C_2=\alpha^2)\\ -P(C_1=\alpha^2)P(C_2=0)-P(C_1=\alpha^2)P(C_2=1)+P(C_1=\alpha^2)P(C_2=\alpha)+P(C_1=\alpha^2)P(C_2=\alpha^2)\\ \\ P(C_1=0)P(C_2=0)-P(C_1=0)P(C_2=1)-P(C_1=0)P(C_2=\alpha)+P(C_1=0)P(C_2=\alpha^2)\\ -P(C_1=1)P(C_2=0)+P(C_1=1)P(C_2=1)+P(C_1=1)P(C_2=\alpha)-P(C_1=1)P(C_2=\alpha^2)\\ -P(C_1=\alpha)P(C_2=0)+P(C_1=\alpha)P(C_2=1)+P(C_1=\alpha)P(C_2=\alpha)-P(C_1=\alpha)P(C_2=\alpha^2)\\ +P(C_1=\alpha^2)P(C_2=0)-P(C_1=\alpha^2)P(C_2=1)-P(C_1=\alpha^2)P(C_2=\alpha)+P(C_1=\alpha^2)P(C_2=\alpha^2)\end{bmatrix}$$

$$C_3=\begin{bmatrix}P(C_3=0)\\ P(C_3=1)\\ P(C_3=\alpha)\\ P(C_3=\alpha^2)\end{bmatrix}=\begin{bmatrix}P(C_1=0)P(C_2=0)+P(C_1=1)P(C_2=1)+P(C_1=\alpha)P(C_2=\alpha)+P(C_1=\alpha^2)P(C_2=\alpha^2)\\ P(C_1=0)P(C_2=1)+P(C_1=1)P(C_2=0)+P(C_1=\alpha)P(C_2=\alpha^2)+P(C_1=\alpha^2)P(C_2=\alpha)\\ P(C_1=0)P(C_2=\alpha)+P(C_1=\alpha)P(C_2=0)+P(C_1=1)P(C_2=\alpha^2)+P(C_1=\alpha^2)P(C_2=1)\\ P(C_1=0)P(C_2=\alpha^2)+P(C_1=\alpha^2)P(C_2=0)+P(C_1=1)P(C_2=\alpha)+P(C_1=\alpha)P(C_2=1)\end{bmatrix}$$

บทที่ 3
การศึกษารหัสแอลดีพีซีแบบนอนไบนารีด้วยวิธีการสุ่ม

3.1 การออกแบบรหัสแอลดีพีซีด้วยวิธีการสร้างเมทริกซ์พาริตีแบบกำหนดเกิร์ท

ในการสร้างเมทริกซ์พาริตีที่มีเกิร์ท g_t และจำนวนโหนดบิต N_t ที่ต้องการเริ่มจากการสร้างเมทริกซ์ขนาด $\frac{g_t}{2}\times\frac{g_t}{2}$ มีคอลัมน์เวทเท่ากับสอง [12] จากนั้นทำการเพิ่มจำนวนคอลัมน์เท่ากับ N_t คอลัมน์

3.2 การออกแบบรหัสแอลดีพีซีแบบนอนไบนารีแบบซีซีแอสดีแอส (คณะกรรมการที่ปรึกษาสำหรับระบบสื่อสารข้อมูลด้านอวกาศ)

งานวิจัยรหัสแอลดีพีซีโดยส่วนใหญ่ในปัจจุบันยังคงเป็นการศึกษารหัสแอลดีพีซีแบบไบนารีเป็นหลัก [13, 14] ในปัจจุบันงานมีวิจัยเพียงไม่กี่ชิ้นที่ได้ศึกษารหัสแอลดีพีซีแบบนอนไบนารี ซึ่งแสดงให้เห็นว่ารหัสแอลดีพีซีแบบนอนไบนารีขนาดสั้นซึ่งนิยามบนสนามกาลัวส์ GF(256) มีประสิทธิภาพดีกว่ารหัสแอลดีพีซีแบบไบนารี 1.0 ถึง 1.3 dB อย่างไรก็ตามความซับซ้อนในการถอดรหัสจะมีมากกว่าการถอดรหัสแอลดีพีซีแบบไบนารี ซึ่งวิธีการออกแบบรหัสแอลดีพีซีบนสนามกาลัวส์ GF(256) จะกล่าวถึงในหัวข้อถัดไป สำหรับรหัสแอลดีพีซีซึ่งนิยามบนสนามกาลัวส์ขนาดเล็ก GF(16) ก็สามารถให้ประสิทธิภาพใกล้เคียงกับรหัสแอลดีพีซีซึ่งนิยามบนสนามกาลัวส์ GF(256)

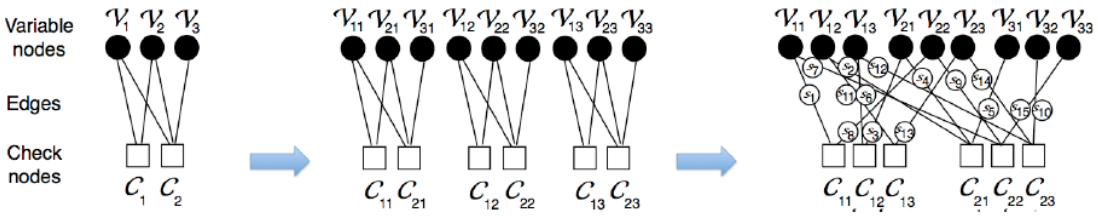
การถอดรหัสแอลดีพีซีแบบนอนไบนารียังเป็นเรื่องที่ต้องมีการศึกษาวิจัยอีกมาก [12] ข่าวสารที่เราส่งผ่านบนกิ่งระหว่างโหนดเป็นเว็กเตอร์ที่บรรจุค่าความน่าจะเป็นของสมาชิกในสนามแต่ละตัว ดังนั้นถ้าใช้สนามกาลัวส์ขนาด q ข่าวสารบนแต่ละกิ่งก็จะประกอบด้วยความน่าจะเป็นเท่ากับ q ค่าหรือมีองศาของความเป็นอิสระเท่ากับ q-1 การคำนวณที่โหนดของกราฟโดยทั่วไปแล้วเป็นการคำนวณโดยใช้การแปลงฟูเรียร์แบบเร็วบนสนามกาลัวส์ ซึ่งมีความซับซ้อนเท่ากับ $q\log q$ ถ้าเป็นอัลกอริทึมที่ถอดรหัสเหมาะสมรองลงมาก็จะช่วยให้ความซับซ้อนลดลงแต่ประสิทธิภาพก็จะถดถอยลงมา โดยทั่วไปความซับซ้อนในการถอดรหัสก็ยังไม่เป็นที่เข้าใจมากนักทำให้ในปัจจุบันทำให้เป็นการยากในการเลือกความเหมาะสมในการถอดรหัสแอลดีพีซีแบบนอนไบนารี

3.3 รหัสแอลดีพีซีบนสนามกาลัวส์ GF(256) โดย ดิฟซาราและโดเลเซ็ค

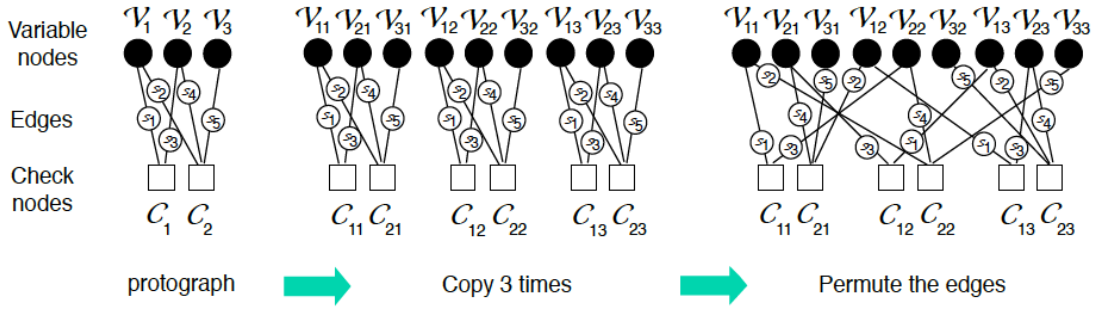
ดิฟซาราและโดเลเซ็คได้ทำการวิจัยเกี่ยวกับการสร้างรหัสแอลดีพีซีแบบนอนไบนารีโดยใช้โปรโตกราฟ [12] ซึ่งจะมีสัมประสิทการคูณหรือสเกลในแต่ละกิ่งของกราฟสัมประสิทสามารถกำหนดได้สองวิธีแต่ในกรณีนี้เริ่มต้นจากการออกแบบโปรโตกราฟโดยยังไม่ได้ออกแบบกิ่ง เหมือนกับรหัสแอลดี

พีซีแบบไบนารี สมรรถนะของโปรโตกราฟในย่านน้ำตกสามารถทำนายได้ด้วยการวิเคราะห์ชาร์ท EXIT แต่ในที่นี้จุดเริ่มต้นการถอดรหัสจะเป็นฟังก์ชันกับขนาดของสนามกาลัวส์ ความจุของช่องสัญญาณของไบนารีอินพุทบนช่องสัญญาณรบกวนเกาส์เซียนแบบขาว เท่ากับ ½ บิตต่อช่องสัญญาณ Eb/N0 = 0.187 dB การวิเคราะห์ชาร์ท EXIT ให้ขนาดบล็อกเท่ากับอนันต์และไม่ทำนายพฤติกรรมการเกิดข้อผิดพลาดแบบพื้น ดังนั้นการเลือกโปรโตกราฟจะขึ้นอยู่กับการทดลองและการจำลองรหัสแอลดีพีซีที่ใส่พารามิเตอร์เข้าไปในการวิเคราะห์

สัมประสิทของสนามจำกัดสามารถใส่ในโปรโตกราฟได้สองวิธีการ วิธีการที่ยืดหยุ่นมากที่สุดคือการขยายโปรโตกราฟไปสู่กราฟแทนเนอร์โดยการใช้การหมุนวนมาตรฐาน จากนั้นใส่สมาชิกสนามในแต่ละกิ่งของกราฟ ดังแสดงในรูปที่ 3.1 เรียกว่า วิธีการสร้างรหัสแบบนอนไบนารีโดยวิธีโปรโตกราฟแบบไม่มีข้อจำกัด “The unconstrained non-binary protograph based (U-NBPB) method of code construction” อีกวิธีการหนึ่งสมาชิกของสนามจำกัดสามารถเพิ่มเข้าไปในกิ่งของโปรโตกราฟจากนั้นทำการคัดลอกด้วยกิ่งที่โปรโตกราฟถูกขยายสู่กราฟแทนเนอร์ดังแสดงในรูปที่ 3.2 วิธีการนี้มีอิสระในการออกแบบน้อยกว่าแต่ช่วยลดความซับซ้อนในกระบวนการเลือกลงได้และช่วยให้การถอดรหัสแอลดีพีซีมีความซับซ้อนลดลง



รูปที่ 3.1 วิธีการสร้างรหัสแบบนอนไบนารีโดยวิธีโพรโตกราฟแบบไม่มีข้อจำกัด [12]



รูปที่ 3.2 วิธีการสร้างรหัสแบบนอนไบนารีโดยวิธีโปรโตกราฟแบบมีข้อจำกัด [14]

โปรโตกราฟแบบ regular (2,4) ได้ถูกนำมาใช้กับสนามกาลัวส์ GF(256) จะได้เมทริกซ์โปรโตกราฟคือ [15, 16]

$$H_p = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

รหัสแบบไบนารี (128,64) เมทริกซ์โปรโตกราฟขยายโดยสัมประสิทธิเท่ากับ 4 ด้วยการหมุนวนเมทริกซ์โปรโตกราฟขนาด 4x4 ถ้าเราให้

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

ดังนั้นการหมุนวนเมทริกซ์สามารถกำหนดได้ด้วยเลขยกกำลังของ σ จะได้ว่าการหมุนของรหัสแบบ (128,64) คือ

$$H_{(128,64)} = \begin{bmatrix} \sigma^0 & \sigma^1 & \sigma^3 & \sigma^2 \\ \sigma^0 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix}$$

ในทำนองเดียวกัน

$$H_{(256,128)} = \begin{bmatrix} \sigma^0 & \sigma^3 & \sigma^6 & \sigma^4 \\ \sigma^1 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix} \quad \text{และ} \quad H_{(128,64)} = \begin{bmatrix} \sigma^0 & \sigma^7 & \sigma^{12} & \sigma^8 \\ \sigma^2 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix}$$

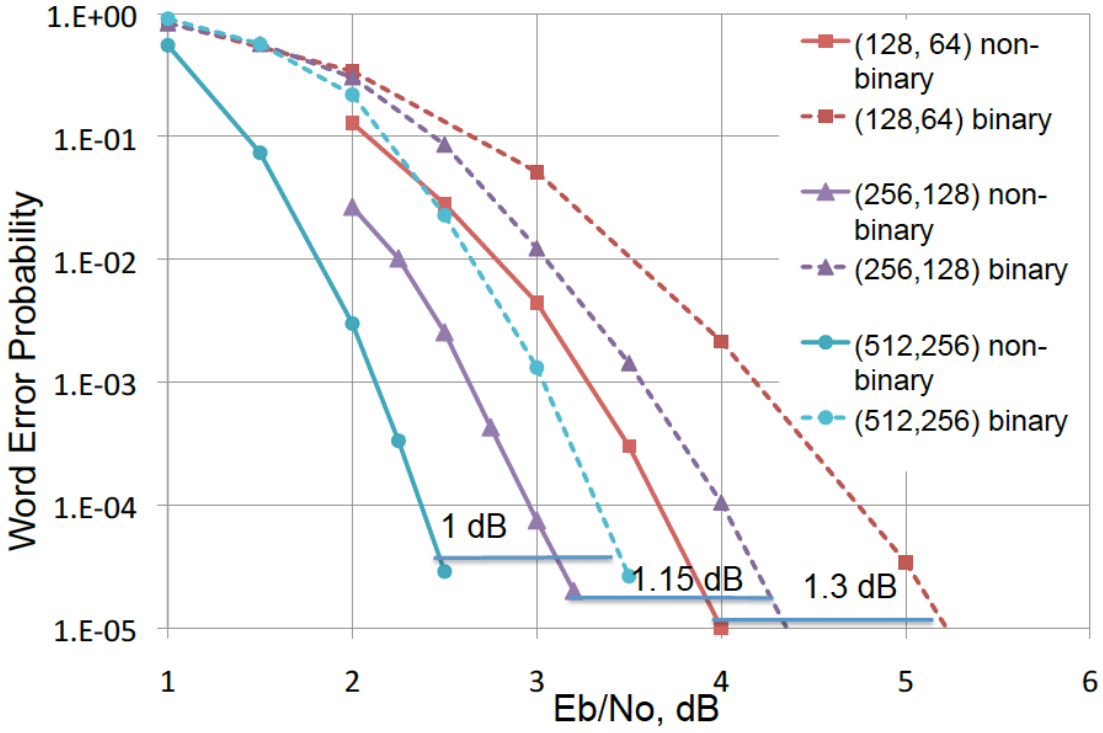
ดังนั้นรหัสแบบนอนไบนารีโดยวิธีโปรโตกราฟแบบไม่มีข้อจำกัด บนสนามกาลัวส์ GF(256) สามารถสร้างได้ดังนี้ [12]

$$H = \begin{bmatrix} \alpha^0 & 0 & 0 & 0 & 0 & \alpha^{89} & 0 & 0 & 0 & 0 & 0 & \alpha^{81} & 0 & 0 & \alpha^9 & 0 \\ 0 & \alpha^8 & 0 & 0 & 0 & 0 & \alpha^0 & 0 & \alpha^{182} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{173} \\ 0 & 0 & \alpha^{173} & 0 & 0 & 0 & 0 & \alpha^8 & 0 & \alpha^0 & 0 & 0 & \alpha^{183} & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^8 & \alpha^0 & 0 & 0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & \alpha^{80} & 0 & 0 \\ \alpha^{183} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^8 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & 0 \\ 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & 0 & \alpha^{80} & 0 & 0 & 0 & \alpha^8 & 0 & 0 \\ 0 & 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{167} & 0 & 0 & 0 & \alpha^{127} & 0 & 0 & 0 & \alpha^{40} & 0 \\ 0 & 0 & 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{182} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^8 \end{bmatrix}$$

โดยที่ α เป็นสมาชิกพริมิทิบของสนามกาลัวส์ GF(256) ดังนั้นรหัสจะถูกขยายออกเป็นรหัสแบบไบนารีที่มีการหมุนวนเท่ากับเลขยกกำลังของ α โดยหารากของสมการพหุนาม $p(x) = 1 + x^2 + x^3 + x^4 + x^8$ ดังนั้น α สามารถแทนด้วยเมทริกซ์ขนาด 8x8 คือ

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

เมื่อทำการแทนค่าเมทริกซ์ α กลับเข้าไปในเมทริก H ก็จะได้พาริตีเมทริกซ์ H ของรหัสแอลดีพีซีแบบไบนารี ประสิทธิภาพของรหัสแอลดีพีซีแบบไบนารีด้วยวิธีการสร้างรหัสแบบนอนไบนารีโดยวิธีโปรโตกราฟแบบไม่มีข้อจำกัดสามารถแสดงประสิทธิภาพได้ตามรูปที่ 3.3 รวมทั้งประสิทธิภาพในการเข้ารหัสบนสนามจำกัดขนาดต่างๆ ก็แสดงได้ตามรูปที่ 3.7 ด้วยเช่นกัน



รูปที่ 3.3 กราฟสมรรถนะของรหัสแบบโพรโตกราฟที่สนามกาลัวส์ GF(256) [12]

3.4 การประยุกต์วิธีการสร้างรหัสแบบนอนไบนารีโดยวิธีโปรโตกราฟ

ในการสร้างเมทริกซ์พาริตีเช็กให้มีเกิร์ทเป้าหมาย g_t และจำนวนโหนดบิต N_t สารมารถทำได้โดยกำหนดเมทริกซ์เริ่มต้นขนาดเท่ากับ $\frac{g_t}{2} \times \frac{g_t}{2}$ ในการขยายเมทริกซ์ไปสู่ขนาด N_t คอลัมน์ จำเป็นจะต้องหาเงื่อนไขที่เหมาะสมในการใส่ตำแหน่งโหนดบิตและโหนดตรวจสอบ ซึ่งจะทำให้ขนาดของเกิร์ทที่ได้น้อยกว่า g_t ดังนั้นโหนดตรวจสอบอันใหม่ควรจะใส่ในตำแหน่งที่เหมาะสมเพื่อรักษาเกิร์ทเป้าหมายให้ได้ตามที่เรากำหนดไว้ ซึ่งสามารถสร้างได้จากอัลกอริทึมตามขั้นตอนที่ 3.1 และ 3.2 ดังนี้ [17]

ขั้นตอนที่ 1:อัลกอริทึมการสร้างเมทริกซ์พาริตีเช็กโดยกำหนดเกิร์ทเป้าหมาย

1: กำหนดเกิร์ทเป้าหมาย g_t

2: กำหนดจำนวนคอลัมน์เป้าหมาย N_t

3: สร้ามเมทริกซ์พาริตีเช็ก $\mathbf{H}$ เริ่มต้นที่มีน้ำหนักคอลัมน์เท่ากับ 2 ขนาด $g_t / 2 \times g_t / 2$

4: N = จำนวนคอลัมน์ของเมทริกซ์พาริตีเช็ก $\mathbf{H}$

5: กำหนด while $N < N_t$ do

6: ทำการเลือกโหนดตรวจสอบที่มีน้ำหนักแถวต่ำที่สุดและให้ เป็น level 1

7: กำหนด $(L_{\max}, c_{L_{\max}}) = \text{findLmax}(\mathbf{H}, c_{\text{init}})$ \\ Algorithm 2

8: กำหนดเงื่อนไข if $L_{\max}$ ³ $g_t / 2$ then

9: เพิ่มโหนดบิตในเมทริกซ์พาริตีเช็ก $\mathbf{H}$ เพื่อเชื่อมต่อ $c_{L_{\max}}$ และ c_{init}.

10: else

11: เพิ่มโหนดบิตจำนวน $g_t / 2$ - $L_{\max}$ +1เข้าไปในเมทริกซ์พาริตีเช็ก $\mathbf{H}$และเพิ่มโหนดตรวจสอบจำนวน $g_t / 2 - L_{\max}$ เพื่อรักษาระดับเกิร์ทเป้าหมายให้เท่ากับ g_t.

12: end if

13: ทำการอัปเดท N = จำนวนคอลัมน์ในเมทริกซ์พาริตีเช็ก $\mathbf{H}$

14: end while

15: ทำการอัปเดทค่าทั้งหมดในเมทริกซ์พาริตีเช็ก $\mathbf{H}$ล่าสุด จะได้ค่าตามที่กำหนดไว้

ขั้นตอนที่ 2: อัลกอริทึมการหา $\mathbf{L_{max}}$

ฟังก์ชัน : $(L_{\max}, c_{L_{\max}}) = \text{findLmax}(\mathbf{H}, c_{\text{init}})$

1: $L = 1;$ \\ตั้งค่าล่าสุด level = 1

2: $\mathbf{c}_L = c_{\text{init}}$ \\ ตั้งค่าโหนดตรวจสอบเริ่มต้นเท่ากับ level = 1

3: กำหนด while forever do

4: $\mathbf{c}_u = [\ ]$ \\ ตั้งค่าการเก็บโหนดตรวจสอบเริ่มต้นทุกโหนด

5: $\mathbf{b}_u = [\ ]$ \\ ต้งค่าการเก็บโหนดบิตเริ่มต้นทุกโหนด

6: กำหนด for all $c_n \in \mathbf{c}_L$ do

7: $\mathbf{b}_n = \text{find}(\mathbf{H}(c_n, :))$ \\ ทำการหาโหนดบิตทุกตัวที่เชื่อมต่อกับโหนดตรวจสอบ

8: $\mathbf{b}_u = [\mathbf{b}_u\ \mathbf{b}_n]$ \\ ทำการเก็บค่าโหนดบิต

9: $\mathbf{H}(c_n, \mathbf{b}_n) = 0$ \\ กำจัดค่าโหนดบิตและโหนดตรวจสอบที่เป็นตัวเอง

10: end for all

11: กำหนด for all $b_n \in \mathbf{b}_u$ do

12: $\mathbf{c}_n = \text{find}(\mathbf{H}(:, b_n))$

13: $\mathbf{c}_u = [\mathbf{c}_u\ \mathbf{c}_n]$

14: $\mathbf{H}(\mathbf{c}_n, b_n) = 0$

15: end for all

16: กำหนด if $\mathbf{c}_u$ เป็นเซ็ตว่าง \\ ให้ทำการตรวจสอบเลเวลที่ต่ำที่สุด

17: return $L_{\max} = L$

18:: return $c_{L_{\max}} = \mathbf{c}_L(1)$

19: break

20: else

21: $L = L + 1$ \\ เลื่อนเลเวลไปที่ระดับต่อไป

22: $\mathbf{c}_L = \mathbf{c}_u$ \\ อัปเดตค่า $\mathbf{c}_L$

23: end if

24: end while

จากวิธีการดังกล่าวเราสามารถออกแบบเมทริกซ์พาริตีเช็กขนาด (128,64) โดยสร้างเมทริกซ์พาราตีเช็ก H_1 ด้วยการใส่ค่า α แบบสุ่ม และเมทริกซ์พาราตีเช็ก H_2 สำหรับรหัสแอลดีพีซีโดยการใส่ค่า α ตามหลักการ *binary images methods* ทำให้ได้เมทริกซ์พาริตีเช็กทั้งสองแบบดังนี้ [18]

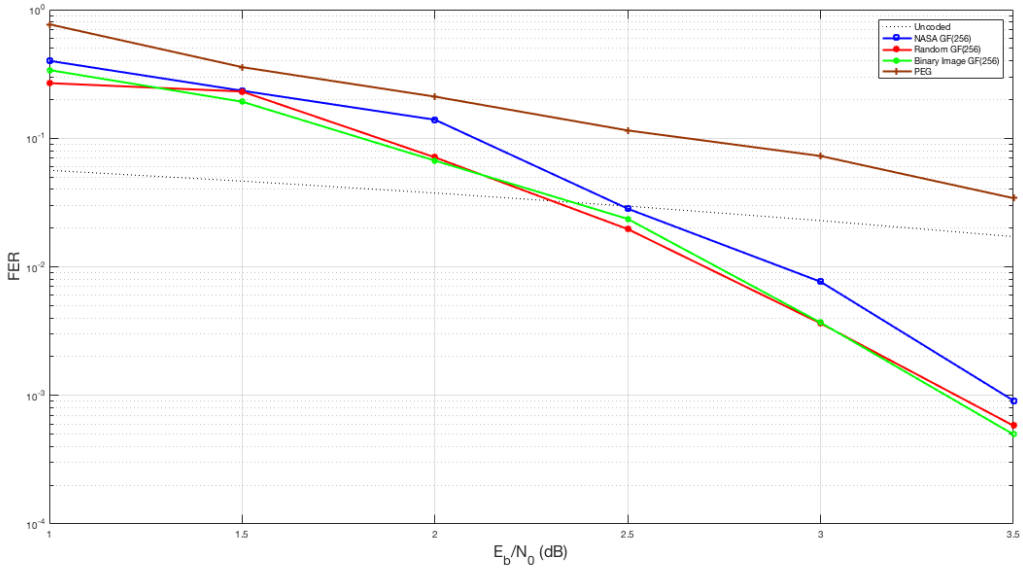
$$\mathbf{H}_1 = \begin{bmatrix}
\alpha^{207} & \alpha^{245} & 0 & 0 & \alpha^{71} & 0 & 0 & 0 & 0 & \alpha^{207} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha^{246} & \alpha^{234} & 0 & 0 & 0 & \alpha^{192} & 0 & 0 & 0 & 0 & 0 & \alpha^{119} & 0 & 0 & 0 \\
0 & 0 & \alpha^{202} & \alpha^{189} & 0 & \alpha^{195} & 0 & 0 & 0 & 0 & 0 & \alpha^{19} & 0 & 0 & 0 & 0 \\
\alpha^{232} & 0 & 0 & \alpha^{100} & 0 & 0 & 0 & 0 & \alpha^{35} & 0 & 0 & 0 & 0 & \alpha^{166} & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha^{177} & \alpha^{47} & 0 & \alpha^{191} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{137} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^{41} & \alpha^{47} & \alpha^{65} & 0 & \alpha^{72} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{157} & \alpha^{193} & \alpha^{238} & 0 & 0 & 0 & \alpha^{21} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{134} & \alpha^{21} & \alpha^{112} & \alpha^{101}
\end{bmatrix}$$

$$\mathbf{H}_2 = \begin{bmatrix}
\alpha^{66} & \alpha^{221} & 0 & 0 & \alpha^{198} & 0 & 0 & 0 & 0 & \alpha^{164} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha^{147} & \alpha^{19} & 0 & 0 & 0 & \alpha^{186} & 0 & 0 & 0 & 0 & 0 & \alpha^{249} & 0 & 0 & 0 \\
0 & 0 & \alpha^{61} & \alpha^{125} & 0 & \alpha^{59} & 0 & 0 & 0 & 0 & 0 & \alpha^{58} & 0 & 0 & 0 & 0 \\
\alpha^{232} & 0 & 0 & \alpha^{124} & 0 & 0 & 0 & 0 & \alpha^{78} & 0 & 0 & 0 & 0 & \alpha^{29} & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha^{24} & \alpha^{209} & 0 & \alpha^{198} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{236} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^{75} & \alpha^{20} & \alpha^{130} & 0 & \alpha^{58} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{223} & \alpha^{215} & \alpha^{109} & 0 & 0 & 0 & \alpha^{124} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{66} & \alpha^{129} & \alpha^{147} & \alpha^{159}
\end{bmatrix}$$

จากผลการทดสอบแสดงให้เห็นว่าที่การวนซ้ำ 20 รอบ เมทริกซ์พาริตีเช็ก H_1 ให้สมรรถนะที่ดีกว่า เมทริกซ์พาริตีเช็ก H_2 จากรูปที่ 3.4 แสดงสมรรถนะในเทอม FER ซึ่งให้สมรรถนะที่ขึ้นประมาณ 0.15 dB [18]

$$\mathbf{H}_{NASA} = \begin{bmatrix}
\alpha^{0} & 0 & 0 & 0 & 0 & \alpha^{89} & 0 & 0 & 0 & 0 & 0 & \alpha^{81} & 0 & 0 & \alpha^{9} & 0 \\
0 & \alpha^{8} & 0 & 0 & 0 & 0 & \alpha^{0} & 0 & \alpha^{182} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{173} \\
0 & 0 & \alpha^{173} & 0 & 0 & 0 & 0 & \alpha^{8} & 0 & \alpha^{0} & 0 & 0 & \alpha^{183} & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha^{8} & \alpha^{0} & 0 & 0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & \alpha^{80} & 0 & 0 \\
\alpha^{183} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^{8} & 0 & 0 & 0 & \alpha^{0} & 0 & 0 & 0 \\
0 & \alpha^{0} & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & 0 & \alpha^{80} & 0 & 0 & 0 & \alpha^{8} & 0 & 0 \\
0 & 0 & \alpha^{0} & 0 & 0 & 0 & \alpha^{167} & 0 & 0 & 0 & \alpha^{127} & 0 & 0 & 0 & \alpha^{40} & 0 \\
0 & 0 & 0 & \alpha^{0} & 0 & 0 & 0 & \alpha^{182} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^{8}
\end{bmatrix}$$

นอกจากนี้เมื่อนำเมทริกซ์พาริตีเช็ก $\mathbf{H}_1$ และ $\mathbf{H}_2$ มากเปรียบเทียบสมรรถนะกับวิธีการออกแบบเมทริกซ์พาริตีเช็กแบบโพรโตกราฟ $\mathbf{H}_{NASA}$ รูปที่ 3.4 แสดงให้เห็นว่าเมทริกซ์พาริตีเช็ก $\mathbf{H}_1$ และ $\mathbf{H}_2$ มีสมรรถนะที่ดีกว่าเมทริกซ์พาริตีเช็กแบบโพรโตกราฟ $\mathbf{H}_{NASA}$



รูปที่ 3.4 กราฟสมรรถนะแบบ FER ของรหัสแอลดีพีซีแบบนอนไบนารีที่การวนซ้ำการถอดรหัส 20 รอบ [18]

บทที่ 4
การดัดแปลงวิธีการถอดรหัสแอลดีพีซีตามหลักการแพร่กระจายความเชื่อ

4.1 รหัสแอลดีพีซีสำหรับมาตรฐาน IEEE 802.11n

ปัจจุบันเทคโนโลยีโครงข่ายท้องถิ่นไร้สายหรือ Wireless Local Area Networks (WLANs) ได้เข้ามามีบทบาทในชีวิตประจำวัน โดยเทคโนโลยี WLANs ทำให้การเชื่อมต่อระหว่างคอมพิวเตอร์ปราศจากสายสัญญาณหรือเป็นการเชื่อมต่อแบบไร้สายนั้นเอง เทคโนโลยีการเชื่อมต่อไร้สายแบ่งเป็นได้หลายมาตรฐาน เช่น IEEE 802.11a, IEEE 802.11b, IEEE 802.11g และ IEEE 802.11n โดย IEEE ย่อมาจาก Institute of Electronics and Electrical Engineers ซึ่งได้ออกมาตรฐานสำหรับเทคโนโลยีการเชื่อมต่อไร้สายเป็นครั้งแรกในปี ค.ศ. 1999 ปัจจุบันมาตรฐานที่ได้รับความนิยมคือ IEEE 802.11n โดยสถาบัน IEEE ได้เผยแพร่มาตรฐานนี้ครั้งแรกในปี ค.ศ. 2009 และได้มีการปรับปรุงมาตรฐานใหม่เพื่อให้การเชื่อมต่อไร้สายมีประสิทธิภาพสูงขึ้น โดยแต่ละมาตรฐานจะใช้คลื่นความถี่วิทยุแตกต่างกันดังตารางที่ 3.1 มีการนำเทคโนโลยี Multiple Input Multiple Output (MIMO) ซึ่งเป็นการรับส่งข้อมูลโดยใช้เสาสัญญาณหลายๆ ต้นพร้อมกัน ทำให้ความเร็วสูงสุดของการรับส่งข้อมูลเพิ่มขึ้นเป็น 600 Mbps

การรับ – ส่งข้อมูลระหว่างอุปกรณ์ที่ใช้การเชื่อมต่อไร้สายท้องถิ่น WLANs ข้อมูลจะถูกรบกวนจากสัญญาณรบกวนผ่านทางช่องสื่อสารไร้สายหรือจากความบกพร่องของสัญญาณเอง ทำให้ภาครับได้รับข้อมูลที่มีความผิดเพี้ยนไป ถึงแม้ว่าที่ภาครับจะมีการร้องขอให้ภาคส่งทำการส่งข้อมูลชุดเดิมมาอีกครั้ง แต่ข้อมูลชุดใหม่ก็จะถูกรบกวนจากสัญญาณรบกวนจากช่องสัญญาณสื่อสารไร้สายหรือเกิดความบกพร่องของสัญญาณได้อยู่ดี ดังนั้นการส่งข้อมูลที่ภาคส่งจึงต้องมีการเข้ารหัสช่องสัญญาณเพื่อลดอัตราความผิดพลาดของข้อมูล ซึ่งรหัสช่องสัญญาณที่ใช้ในมาตรฐาน IEEE 802.11n คือรหัสแอลดีพีซี โดยสถาบัน IEEE ได้กำหนดความยาวคำรหัสที่ใช้เป็น 3 แบบคือ 648 บิต 1296 บิต และ 1944 บิต และกำหนดอัตรารหัสคือ 1/2, 2/3, 3/4, และ 5/6

ตารางที่ 4.1 โพรโทคอลของมาตฐาน WLANs [19]

Protocol	Release Date	Frequency	Speed (Max)	Range (Indoor)	Range (Outdoor)
IEEE 802.11a	1999	5 GHz	54 Mbps	35 m	120 m
IEEE 802.11b	1999	2.4 GHz	11 Mbps	35 m	120 m
IEEE 802.11g	2003	2.4 GHz	54 Mbps	38 m	140 m
IEEE 802.11n	2009	2.4 GHz 5 GHz	600 Mbps	70 m	250 m
IEEE 802.11ac	2013	5 GHz	1.3 Gbps	70 m	250 m

การออกแบบรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n จะดัดแปลงโครงสร้างของรหัสสะสมข้อมูลซ้ำแบบไม่สม่ำเสมอ (Irregular Repeat-Accumulate Code) ให้มีลักษณะเป็นรหัสแบบหมุนวน เมทริกซ์พาริตีเช็กของรหัสแอลดีพีซีที่ใช้ในมาตรฐาน IEEE 802.11n สามารถแบ่งได้เป็นสองส่วนคือ [20]

$$\mathbf{H}=\begin{bmatrix}\mathbf{H}_1 & \mathbf{H}_2\end{bmatrix} \tag{4.1}$$

เมื่อ $\mathbf{H}_1$ คือเมทริกซ์ขนาด $(N-K)\times K$ โดย N คือความยาวของคำรหัส K คือความยาวของบิตข้อมูล ภายในเมทริกซ์ $\mathbf{H}_1$ จะประกอบไปด้วยเมทริกซ์ย่อย $\mathbf{h}_{ij}$ ขนาด $z\times z$ ดังนี้

$$\mathbf{H}_1=\begin{bmatrix}\mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1,K_b}\\ \mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2,K_b}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{h}_{Mb,1} & \mathbf{h}_{Mb,2} & \cdots & \mathbf{h}_{Mb,K_b}\end{bmatrix} \tag{4.2}$$

โดยเมทริกซ์ย่อย $\mathbf{h}_{ij}$ เป็นเมทริกซ์ศูนย์หรือเมทริกซ์เรียงสับเปลี่ยน $\mathbf{P}_l$ โดย $\mathbf{P}_l$ ได้จากการนำเมทริกซ์เอกลักษณ์ $\mathbf{I}$ มาทำการเลื่อนแถวหรือหลักเป็นจำนวน l ครั้ง

ตัวอย่างของเมทริกซ์เรียงสับเปลี่ยน $\mathbf{P}_1$ ขนาด 3×3 สร้างจากการเลื่อนแถวหรือหลักเมทริกซ์เอกลักษณ์ $\mathbf{I}$ เป็นจำนวน 1 ครั้ง และเมทริกซ์เรื่องสับเปลี่ยน $\mathbf{P}_2$ สร้างจากการเลื่อนแถวหรือหลักของเมทริกซ์เอกลักษณ์ $\mathbf{I}$ เป็นจำนวนสองครั้ง สังเกตว่าพอเมทริกซ์เรียงสับเปลี่ยนเป็น $\mathbf{P}_3$ก็จะกลับมาเป็นเมทริกซ์เอกลักษณ์ $\mathbf{I}$ เหมือนเดิมดังสมการ (3.3)

$$\mathbf{I}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\quad \mathbf{P}_1=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\quad \mathbf{P}_2=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\quad \mathbf{P}_3=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{4.3}$$

คุณสมบัติที่สำคัญของเมทริกซ์เรียงสับเปลี่ยนคือเมื่อทำการเลื่อนแถวหรือหลักเป็นจำนวนครั้งเท่ากับขนาดของเมทริกซ์ผลลัพธ์ที่ได้จะเป็นเมทริกซ์ตัวเดิม นอกจากนี้เมื่อนำเมทริกซ์ใดๆ คูณกับเมทริกซ์เรียงสับเปลี่ยนผลลัพธ์ที่ได้จะเป็นการเลื่อนสมาชิกที่อยู่ในเมทริกซ์ดังสมการ (3.4)

$$\begin{aligned}\mathrm{P}_0\mathrm{u}^{\mathrm{T}}&=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} \quad \mathrm{P}_1\mathrm{u}^{\mathrm{T}}=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}\\ \mathrm{P}_2\mathrm{u}^{\mathrm{T}}&=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix} \quad \mathrm{P}_3\mathrm{u}^{\mathrm{T}}=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\end{aligned} \tag{4.4}$$

ในส่วนของเมทริกซ์ $\mathbf{H}_2$ จะมีขนาดเท่ากับ $(N-K)\times(N-K)$ ซึ่งในเมทริกซ์ $\mathbf{H}_2$ ประกอบไปด้วยเมทริกซ์ศูนย์ เมทริกซ์เอกลักษณ์ $\mathbf{I}$ และเมทริกซ์เรียงสับเปลี่ยน $\mathbf{P}_1$ ดังสมการ (3.5)

$$\mathbf{H}_2=\begin{bmatrix}\mathbf{P}_1&\mathbf{I}&0&0&\cdots&0&0&0\\0&\mathbf{I}&\mathbf{I}&0&\cdots&0&0&0\\0&0&\mathbf{I}&\mathbf{I}&\cdots&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\\mathbf{I}&0&0&0&\ddots&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\0&0&0&0&\cdots&\mathbf{I}&\mathbf{I}&0\\0&0&0&0&\cdots&0&\mathbf{I}&\mathbf{I}\\\mathbf{P}_1&0&0&0&\cdots&0&0&\mathbf{I}\end{bmatrix} \tag{4.5}$$

เมื่อนำเมทริกซ์ $\mathbf{H_1}$ เมทริกซ์ $\mathbf{H}_2$ในสมการ (4.2) และ (4.5) แทนค่าในเมทริกซ์ $\mathbf{H}$ ในสมการ (4.1) จะได้เมทริกซ์ตามมาตรฐาน IEEE 802.11n ดังนี้

$$\mathbf{H}=\begin{bmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1,K_b} & \mathbf{P}_1 & \mathbf{I} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2,K_b} & 0 & \mathbf{I} & \mathbf{I} & 0 & \cdots & 0 & 0 & 0 \\ \mathbf{h}_{31} & \mathbf{h}_{32} & \cdots & \mathbf{h}_{3,K_b} & 0 & 0 & \mathbf{I} & \mathbf{I} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{h}_{x1} & \mathbf{h}_{x2} & \ddots & \mathbf{h}_{x,K_b} & \mathbf{I} & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{h}_{M_b-2,1} & \mathbf{h}_{M_b-2,2} & \cdots & \mathbf{h}_{M_b-2,K_b-2} & 0 & 0 & 0 & 0 & \cdots & \mathbf{I} & \mathbf{I} & 0 \\ \mathbf{h}_{M_b-1,1} & \mathbf{h}_{M_b-1,2} & \cdots & \mathbf{h}_{M_b-1,K_b} & 0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{I} & \mathbf{I} \\ \mathbf{h}_{M_b,1} & \mathbf{h}_{M_b,2} & \cdots & \mathbf{h}_{M_b,K_b} & \mathbf{P}_1 & 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{I} \end{bmatrix} \quad (4.6)$$

มาตรฐาน IEEE 802.11n มีการกำหนดเมทริกซ์ย่อยขนาดที่แตกต่างกัน 3 ขนาดคือ $z = 27/54$ และ 81 ซึ่งใช้กับคำรหัสที่มีความยาว $N = 648/1296$ และ 1944 บิต ตามลำดับ รูปที่ 4.1 – 4.3 เป็นรายละเอียดของเมทริกซ์ $\mathbf{H}$ ที่แสดงการเรียงสับเปลี่ยนของเมทริกซ์ ตัวเลขที่แสดงอยู่ในตารางระบุถึงจะนวนครั้งในการหมุนวนของเมทริกซ์เรียงสับเปลี่ยนนั้น โดยตัวเลข -1 แสดงถึงเมทริกซ์เอกลักษณ์ $\mathbf{I}$ และตัวเลข 0 แสดงถึงเมทริกซ์ศูนย์

$$\mathbf{H}=\begin{bmatrix}
0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & -1 & 0 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
22 & 0 & -1 & -1 & 17 & -1 & 0 & 0 & 12 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
6 & -1 & 0 & -1 & 10 & -1 & -1 & -1 & 24 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
2 & -1 & -1 & 0 & 20 & -1 & -1 & -1 & 25 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
23 & -1 & -1 & -1 & 3 & -1 & -1 & -1 & 0 & -1 & 9 & 11 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
24 & -1 & 23 & 1 & 17 & -1 & 3 & -1 & 10 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
25 & -1 & -1 & -1 & 8 & -1 & -1 & -1 & 7 & 18 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
13 & 24 & -1 & -1 & 0 & -1 & 8 & -1 & 6 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
7 & 20 & -1 & 16 & 22 & 10 & -1 & -1 & 23 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
11 & -1 & -1 & -1 & 19 & -1 & -1 & -1 & 13 & -1 & 3 & 17 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
25 & -1 & 8 & -1 & 23 & 18 & -1 & 14 & 9 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
3 & -1 & -1 & -1 & 16 & -1 & -1 & 2 & 25 & 5 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ก)

$$\mathbf{H}=\begin{bmatrix}
25 & 26 & 14 & -1 & 20 & -1 & 2 & -1 & 4 & -1 & -1 & 8 & -1 & 16 & -1 & 18 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
10 & 9 & 15 & 11 & -1 & 0 & -1 & 1 & -1 & -1 & 18 & -1 & 8 & -1 & 10 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
16 & 2 & 20 & 26 & 21 & -1 & 6 & -1 & 1 & 26 & -1 & 7 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
10 & 13 & 5 & 0 & -1 & 3 & -1 & 7 & -1 & -1 & 26 & -1 & -1 & 13 & -1 & 16 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
23 & 14 & 24 & -1 & 12 & -1 & 19 & -1 & 17 & -1 & -1 & -1 & 20 & -1 & 21 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
6 & 22 & 9 & 20 & -1 & 25 & -1 & 17 & -1 & 8 & -1 & 14 & -1 & 18 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
14 & 23 & 21 & 11 & 20 & -1 & 24 & -1 & 18 & -1 & 19 & -1 & -1 & -1 & -1 & 22 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
17 & 11 & 11 & 20 & -1 & 21 & -1 & 26 & -1 & 3 & -1 & -1 & 18 & -1 & 26 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ข)

$$\mathbf{H}=\begin{bmatrix}
16 & 17 & 22 & 24 & 9 & 3 & 14 & -1 & 4 & 2 & 7 & -1 & 26 & -1 & 2 & -1 & 21 & -1 & 1 & 0 & -1 & -1 & -1 & -1 \\
25 & 12 & 12 & 3 & 3 & 26 & 6 & 21 & -1 & 15 & 22 & -1 & 15 & -1 & 4 & -1 & -1 & 16 & -1 & 0 & 0 & -1 & -1 & -1 \\
25 & 18 & 26 & 16 & 22 & 23 & 9 & -1 & 0 & -1 & 4 & -1 & 4 & -1 & 8 & 23 & 11 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
9 & 7 & 0 & 1 & 17 & -1 & -1 & 7 & 3 & -1 & 3 & 23 & -1 & 16 & -1 & -1 & 21 & -1 & 0 & -1 & -1 & 0 & 0 & -1 \\
24 & 5 & 26 & 7 & 1 & -1 & -1 & 15 & 24 & 15 & -1 & 8 & -1 & 13 & -1 & 13 & -1 & 11 & -1 & -1 & -1 & -1 & 0 & 0 \\
2 & 2 & 19 & 14 & 24 & 1 & 15 & 19 & -1 & 21 & -1 & 2 & -1 & 24 & -1 & 3 & -1 & 2 & 1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ค)

$$\mathbf{H}=\begin{bmatrix}
17 & 13 & 8 & 21 & 9 & 3 & 18 & 12 & 10 & 0 & 4 & 15 & 19 & 2 & 5 & 10 & 26 & 19 & 13 & 13 & 1 & 0 & -1 & -1 \\
3 & 12 & 11 & 14 & 11 & 25 & 5 & 18 & 0 & 9 & 2 & 26 & 26 & 10 & 24 & 7 & 14 & 20 & 4 & 2 & -1 & 0 & 0 & -1 \\
22 & 16 & 4 & 3 & 10 & 21 & 12 & 5 & 21 & 14 & 19 & 5 & -1 & 8 & 5 & 18 & 11 & 5 & 5 & 15 & 0 & -1 & 0 & 0 \\
7 & 7 & 14 & 14 & 4 & 16 & 16 & 24 & 24 & 10 & 1 & 7 & 15 & 6 & 10 & 26 & 8 & 18 & 21 & 14 & 1 & -1 & -1 & 0
\end{bmatrix}$$

(ง)

รูปที่ 4.1 เมทริกซ์พาริตีเช็กสำหรับความยาว $N=648, Z=27$ ที่มีอัตรารหัสต่างๆ [19] (ก) อัตรารหัสเท่ากับ $\frac{1}{2}$, (ข) อัตรารหัสเท่ากับ $\frac{2}{3}$, (ค) อัตรารหัสเท่ากับ $\frac{3}{4}$ และ (ง) อัตรารหัสเท่ากับ $\frac{5}{6}$

$$\mathbf{H}=\begin{bmatrix}
40 & -1 & -1 & -1 & 22 & -1 & 49 & 23 & 43 & -1 & -1 & -1 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
50 & 1 & -1 & -1 & 48 & 35 & -1 & -1 & 13 & -1 & 30 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
39 & 50 & -1 & -1 & 4 & -1 & 2 & -1 & -1 & -1 & -1 & 49 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
33 & -1 & -1 & 38 & 37 & -1 & -1 & 4 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
45 & -1 & -1 & -1 & 0 & 22 & -1 & -1 & 20 & 42 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
51 & -1 & -1 & 48 & 35 & -1 & -1 & -1 & 44 & -1 & 18 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
47 & 11 & -1 & -1 & -1 & 17 & -1 & -1 & 51 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
5 & -1 & 25 & -1 & 6 & -1 & 45 & -1 & 13 & 40 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
33 & -1 & -1 & 34 & 24 & -1 & -1 & -1 & 23 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
1 & -1 & 27 & -1 & 1 & -1 & -1 & -1 & 38 & -1 & 44 & 46 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
-1 & 18 & -1 & -1 & 23 & -1 & -1 & 8 & 0 & 35 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
49 & -1 & 17 & -1 & 30 & -1 & -1 & 2 & 34 & -1 & -1 & 19 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ก)

$$\mathbf{H}=\begin{bmatrix}
39 & 31 & 22 & 43 & -1 & 40 & 4 & -1 & 11 & -1 & -1 & 50 & -1 & -1 & -1 & 6 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
25 & 52 & 41 & 2 & 6 & -1 & 14 & -1 & 34 & -1 & -1 & -1 & 24 & -1 & 37 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
43 & 31 & 29 & 0 & 21 & -1 & 28 & -1 & -1 & 2 & -1 & -1 & 7 & -1 & 17 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
20 & 33 & 48 & -1 & 4 & 13 & -1 & 26 & -1 & -1 & 22 & -1 & -1 & 46 & 42 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
45 & 7 & 18 & 51 & 12 & 25 & -1 & -1 & -1 & 50 & -1 & -1 & 5 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
35 & 40 & 32 & 16 & 5 & -1 & -1 & 18 & -1 & -1 & 43 & 51 & -1 & 32 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
9 & 24 & 13 & 22 & 28 & -1 & -1 & 37 & -1 & -1 & 25 & -1 & -1 & 52 & -1 & 13 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
32 & 22 & 4 & 21 & 16 & -1 & -1 & -1 & 27 & 28 & -1 & 38 & -1 & -1 & -1 & 8 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ข)

$$\mathbf{H}=\begin{bmatrix}
39 & 40 & 51 & 41 & 3 & 29 & 8 & 36 & -1 & 14 & -1 & 6 & -1 & 33 & -1 & 11 & -1 & 4 & 1 & 0 & -1 & -1 & -1 & -1 \\
48 & 21 & 47 & 9 & 48 & 35 & 51 & -1 & 38 & -1 & 28 & -1 & 34 & -1 & 50 & -1 & 50 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
30 & 39 & 28 & 42 & 50 & 39 & 5 & 17 & -1 & 6 & -1 & 18 & -1 & 20 & -1 & 15 & -1 & 40 & -1 & -1 & 0 & 0 & -1 & -1 \\
29 & 0 & 1 & 43 & 36 & 30 & 47 & -1 & 49 & -1 & 47 & -1 & 3 & -1 & 35 & -1 & 34 & -1 & 0 & -1 & -1 & 0 & 0 & -1 \\
1 & 32 & 11 & 23 & 10 & 44 & 12 & 7 & -1 & 48 & -1 & 4 & -1 & 9 & -1 & 17 & -1 & 16 & -1 & -1 & -1 & -1 & 0 & 0 \\
13 & 7 & 15 & 47 & 23 & 16 & 47 & -1 & 43 & -1 & 29 & -1 & 52 & -1 & 2 & -1 & 53 & -1 & 1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ค)

$$\mathbf{H}=\begin{bmatrix}
48 & 29 & 37 & 52 & 2 & 16 & 6 & 14 & 53 & 31 & 34 & 5 & 18 & 42 & 53 & 31 & 45 & -1 & 46 & 52 & 1 & 0 & -1 & -1 \\
17 & 4 & 30 & 7 & 43 & 11 & 24 & 6 & 14 & 21 & 6 & 39 & 17 & 40 & 47 & 7 & 15 & 41 & 19 & -1 & -1 & 0 & 0 & -1 \\
7 & 2 & 51 & 31 & 46 & 23 & 16 & 11 & 53 & 40 & 10 & 7 & 46 & 53 & 33 & 35 & -1 & 25 & 35 & 38 & 0 & -1 & 0 & 0 \\
19 & 48 & 41 & 1 & 10 & 7 & 36 & 47 & 5 & 29 & 52 & 52 & 31 & 10 & 26 & 6 & 3 & 2 & -1 & 51 & 1 & -1 & -1 & 0
\end{bmatrix}$$

(ง)

รูปที่ 4.2 เมทริกซ์พาริตีเช็กสำหรับความยาว $N=1296, Z=54$ ที่มีอัตรารหัสต่างๆ [19] (ก) อัตรารหัสเท่ากับ $\frac{1}{2}$, (ข) อัตรารหัสเท่ากับ $\frac{2}{3}$, (ค) อัตรารหัสเท่ากับ $\frac{3}{4}$ และ (ง) อัตรารหัสเท่ากับ $\frac{5}{6}$

$$\mathbf{H}=\begin{bmatrix}
57 & -1 & -1 & -1 & 50 & -1 & 11 & -1 & 50 & -1 & 79 & -1 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
3 & -1 & 28 & -1 & 0 & -1 & -1 & -1 & 55 & 7 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
30 & -1 & -1 & -1 & 24 & 37 & -1 & -1 & 56 & 14 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
62 & 53 & -1 & -1 & 53 & -1 & -1 & 3 & 35 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
40 & -1 & -1 & 20 & 66 & -1 & -1 & 22 & 28 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
0 & -1 & -1 & -1 & 8 & -1 & 42 & -1 & 50 & -1 & -1 & 8 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
69 & 79 & 79 & -1 & -1 & -1 & 56 & -1 & 52 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
65 & -1 & -1 & -1 & 38 & 57 & -1 & -1 & 72 & -1 & 27 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
64 & -1 & -1 & -1 & 14 & 52 & -1 & -1 & 30 & -1 & -1 & 32 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
-1 & 45 & -1 & 70 & 0 & -1 & -1 & -1 & 77 & 9 & -1 & 46 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
2 & 56 & -1 & 57 & 35 & -1 & -1 & -1 & -1 & -1 & 12 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
24 & -1 & 61 & -1 & 60 & -1 & -1 & 27 & 51 & -1 & -1 & 16 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ก)

$$\mathbf{H}=\begin{bmatrix}
61 & 75 & 4 & 63 & 56 & -1 & -1 & -1 & -1 & -1 & -1 & 8 & -1 & 2 & 17 & 25 & 1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
56 & 74 & 77 & 20 & -1 & -1 & -1 & 64 & 24 & 4 & 67 & -1 & 7 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
28 & 21 & 68 & 10 & 7 & 14 & 65 & -1 & -1 & -1 & 23 & -1 & -1 & -1 & 75 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
48 & 38 & 43 & 78 & 76 & -1 & -1 & -1 & -1 & 5 & 36 & -1 & 15 & 72 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -11 \\
40 & 2 & 53 & 25 & -1 & 52 & 62 & -1 & 20 & -1 & -1 & 44 & -1 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\
69 & 23 & 64 & 10 & 22 & -1 & 21 & -1 & -1 & -1 & -1 & -1 & 68 & 23 & 29 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 \\
12 & 0 & 68 & 20 & 55 & 61 & -1 & 40 & -1 & -1 & -1 & 52 & -1 & -1 & -1 & 44 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
58 & 8 & 34 & 64 & 78 & -1 & -1 & 11 & 78 & 24 & -1 & -1 & -1 & -1 & -1 & 58 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ข)

$$\mathbf{H}=\begin{bmatrix}
48 & 29 & 28 & 39 & 9 & 61 & -1 & -1 & -1 & 63 & 45 & 80 & -1 & -1 & -1 & 37 & 32 & 22 & 1 & 0 & -1 & -1 & -1 & -1 \\
4 & 49 & 42 & 48 & 11 & 30 & -1 & -1 & -1 & 49 & 17 & 41 & 37 & 15 & -1 & 54 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 \\
35 & 76 & 78 & 51 & 37 & 35 & 21 & -1 & 17 & 64 & -1 & -1 & -1 & 59 & 7 & -1 & -1 & 32 & -1 & -1 & 0 & 0 & -1 & -1 \\
9 & 65 & 44 & 9 & 54 & 56 & 73 & 34 & 42 & -1 & -1 & -1 & 35 & -1 & -1 & -1 & 46 & 39 & 0 & -1 & -1 & 0 & 0 & -1 \\
3 & 62 & 7 & 80 & 68 & 26 & -1 & 80 & 55 & -1 & 36 & -1 & 26 & -1 & 9 & -1 & 72 & -1 & -1 & -1 & -1 & -1 & 0 & 0 \\
26 & 75 & 33 & 21 & 69 & 59 & 3 & 38 & -1 & -1 & -1 & 35 & -1 & 62 & 36 & 26 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 0
\end{bmatrix}$$

(ค)

$$\mathbf{H}=\begin{bmatrix}
13 & 48 & 80 & 66 & 4 & 74 & 7 & 30 & 76 & 52 & 37 & 60 & -1 & 49 & 73 & 31 & 74 & 73 & 23 & -1 & 1 & 0 & -1 & -1 \\
69 & 63 & 74 & 56 & 64 & 77 & 57 & 65 & 6 & 16 & 51 & -1 & 64 & -1 & 68 & 9 & 48 & 62 & 54 & 27 & -1 & 0 & 0 & -1 \\
51 & 15 & 0 & 80 & 24 & 25 & 42 & 54 & 44 & 71 & 71 & 9 & 67 & 35 & -1 & 58 & -1 & 29 & -1 & 53 & 0 & -1 & 0 & 0 \\
16 & 29 & 36 & 41 & 44 & 56 & 59 & 37 & 50 & 24 & -1 & 65 & 4 & 65 & 52 & -1 & 4 & -1 & 73 & 52 & 1 & -1 & -1 & 0
\end{bmatrix}$$

(ง)

รูปที่ 4.3 เมทริกซ์พาริตีเช็กสำหรับความยาว $N=1944, Z=81$ ที่มีอัตรารหัสต่างๆ [19] (ก) อัตรารหัสเท่ากับ $\frac{1}{2}$, (ข) อัตรารหัสเท่ากับ $\frac{2}{3}$, (ค) อัตรารหัสเท่ากับ $\frac{3}{4}$ และ (ง) อัตรารหัสเท่ากับ $\frac{5}{6}$

4.2 การเข้ารหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n

รหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n เป็นรหัสเชิงระบบ หมายความว่าคำรหัสจะแบ่งส่วนของบิตข้อมูลและบิตพาริตีอย่างชัดเจนหรือ $\mathbf{c} = \begin{bmatrix}\mathbf{u} & \mathbf{p}\end{bmatrix}$ เมื่อ $\mathbf{u}$ เป็นเวกเตอร์ข้อมูลบิตขนาด $1\times K$ เวกเตอร์ข้อมูล $\mathbf{u}$ แบ่งเป็นเวกเตอร์ข้อมูลย่อยเป็นจำนวน K_b หรือ $\mathbf{u}=\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{K_b}\end{bmatrix}$ โดย $\mathbf{u}_j$ มีขนาด $1\times z$ และ $\mathbf{p}$ เป็นเวกเตอร์พาริตีบิตขนาด $1\times M$ เมื่อ $M = N-K$ แบ่งเวกเตอร์พาริตีย่อยเป็นจำนวน M_b หรือ $\mathbf{p}=\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{M_b}\end{bmatrix}$ โดย $\mathbf{p}_i$ มีขนาด $1\times z$ ดังนั้นสมการเวกเตอร์คำรหัสคือ

$$\mathbf{c} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{K_b} & \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{M_b}\end{bmatrix} \tag{4.7}$$

จากทฤษฎีในบทที่ 2 เราสามารถหาซินโดรม $\mathbf{z} = \mathbf{H}\mathbf{c}^T = 0$ นั่นคือ

$$\begin{bmatrix}
\mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1,K_b} & \mathbf{P}_1 & \mathbf{I} & 0 & 0 & \cdots & 0 & 0 & 0 \\
\mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2,K_b} & 0 & \mathbf{I} & \mathbf{I} & 0 & \cdots & 0 & 0 & 0 \\
\mathbf{h}_{31} & \mathbf{h}_{32} & \cdots & \mathbf{h}_{3,K_b} & 0 & 0 & \mathbf{I} & \mathbf{I} & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{h}_{x1} & \mathbf{h}_{x2} & \ddots & \mathbf{h}_{x,K_b} & \mathbf{I} & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{h}_{M_b-2,1} & \mathbf{h}_{M_b-2,2} & \cdots & \mathbf{h}_{M_b-2,K_b-2} & 0 & 0 & 0 & 0 & \cdots & \mathbf{I} & \mathbf{I} & 0 \\
\mathbf{h}_{M_b-1,1} & \mathbf{h}_{M_b-1,2} & \cdots & \mathbf{h}_{M_b-1,K_b} & 0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{I} & \mathbf{I} \\
\mathbf{h}_{M_b,1} & \mathbf{h}_{M_b,2} & \cdots & \mathbf{h}_{M_b,K_b} & \mathbf{P}_1 & 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{I}
\end{bmatrix}
\begin{bmatrix}
u_1^T \\ u_2^T \\ \vdots \\ \vdots \\ u_{K_b}^T \\ P_1^T \\ P_2^T \\ \vdots \\ P_{M_b}^T
\end{bmatrix} = 0 \tag{4.8}$$

ดังนั้นแต่ล่ะแถวจะมีสมการคือ

$$\sum_{j=1}^{K_b} \mathrm{h}_{1j}\mathrm{u}_j^T + \mathrm{P}_1\mathrm{p}_1^T + \mathrm{P}_2^T = 0 \tag{4.9}$$

$$\sum_{j=1}^{K_b} \mathrm{h}_{ij}\mathrm{u}_j^T + \mathrm{P}_i^T + \mathrm{P}_{i+1}^T = 0, \quad i \neq 1, x, M_b \tag{4.10}$$

$$\sum_{j=1}^{K_b} \mathrm{h}_{xj}\mathrm{u}_j^T + \mathrm{p}_1^T + \mathrm{p}_x^T + \mathrm{p}_{x+1}^T = 0 \tag{4.11}$$

$$\sum_{j=1}^{K_b} \mathrm{h}_{M_b j}\mathrm{u}_j^T + \mathrm{P}_1^T\mathrm{p}_1^T + \mathrm{P}_{M_b}^T = 0 \tag{4.12}$$

นำสมการ (4.9)-(4.12) มาบวกกันแบบโมดูโล-2 จะได้ว่า

$$\sum_{i=1}^{M_b}\sum_{j=1}^{N_b}\mathbf{h}_{ij}\mathbf{u}_j^T+\mathbf{p}_1^T=0 \tag{4.13}$$

กำหนดให้ $\lambda=\sum_{j=1}^{K_b}\mathbf{h}_{ij}\mathbf{u}_j^T$ ดังนั้นพาริตีย่อย $\mathbf{p}_1$ จะมีค่าเท่ากับ

$$\mathbf{p}_1^{\mathrm{T}}=\sum_{i=1}^{M_b}\lambda_i \tag{4.14}$$

นำสมการ (3.14) แทนในสมการ (3.9) จะได้พาริตีย่อย $\mathbf{p}_2$ จะมีค่าเท่ากับ

$$\mathbf{p}_2^T=\lambda_1+\mathbf{P}_1\mathbf{p}_1^{\mathrm{T}} \tag{4.15}$$

นำสมการ (3.10) จะได้ว่า

$$\mathbf{p}_i^{\mathrm{T}}=\lambda_{i-1}+\mathbf{p}_{i-1}^{\mathrm{T}} \tag{4.16}$$

นำสมการ (3.14) แทนในสมการ (3.12) จะได้ว่า

$$\mathbf{P}_{M_b}^{\mathrm{T}}=\lambda_{M_b}+\mathbf{P}_1\mathbf{p}_1^T \tag{4.17}$$

สังเกตว่าการเข้ารหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n จะต้องคำนวณหาพาริตีย่อย $\mathbf{p}_1$ ก่อนจากนั้นจึงคำนวณหาค่า $\mathbf{p}_i$ การคำนวณหาพาริตีย่อยจะใช้กระบวนการคูณและบวกเมทริกซ์ การคูณระหว่างเมทริกซ์เรียงสับเปลี่ยนกับเมทริกซ์ใดๆ จะเป็นการเลื่อนสมาชิกของเมทริกซ์นั้นก็จะสามารถเข้ารหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n ได้

เมื่อทำการเข้ารหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n เรียบร้อยแล้วจึงนำสัญญาณที่เข้ารหัสมาทดสอบสมรรถนะโดยจำลองการส่งข้อมูลข่าวสารผ่านช่องสัญญาณรบกวนเกาส์เซียนสีขาวแบบบวก โดยทำการทดสอบพลังงานบิต (E_b/N_0) ที่ 1-5 dB โดยให้ที่ภาครับเมื่อรับข้อมูลเพื่อทำการถอดรหัสแล้วนำข้อมูลมาทำการถอดรหัสแบบวนซ้ำ (iteration) เป็นจำนวน 50 รอบ จากนั้นเก็บข้อมูลอัตราความผิดพลาดที่ได้จากการจำลองการส่งข้อมูลข่าวสารทำให้ได้สมรรถ นะของรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n ตามรูปที่ 3.4 ได้ยกตัวอย่างของการทดสอบสมรรถนะของรหัสแอลดีพีซีที่ความยาว $N=648$ บิต ที่อัตรารหัสต่างๆ

4.3 วิธีการถอดรหัสแอลดีพีซีแบบเดิมและขั้นตอนการดัดแปลงเพื่อเพิ่มสมรรถนะ

4.3.1 วิธีการถอดรหัสแบบดั้งเดิมตามหลักของการแพร่กระจายความเชื่อ

วิธีการถอดรหัสแบบดั้งเดิมอิงตามหลักของการแพร่กระจายความเชื่อ (Belief Propagation) ซึ่งมีอยู่ 5 ขั้นตอนดังนี้

ขั้นตอนที่ 1 การคำนวณค่าเริ่มต้น

ขั้นตอนนี้คำนวณหาค่าเริ่มต้น $q_{ij}(1)$ และ $q_{ij}(0)$ จากสัญญาณ y_i ที่ได้รับจากช่องสัญญาณ เพื่อใช้เป็นข่าวสารที่จะส่งจากโนดบิต i ไปยังโนดตรวจสอบ j ที่เชื่อมต่อด้วย โดยที่

- $q_{ij}(1) = P_i = P_r(c_i = 1 | y_i)$ คือความน่าจะเป็นที่บิต c_i มีค่าเป็น 1 และ
- $q_{ij}(0) = 1 - P_i = P_r(c_i = 0 | y_i)$ คือความน่าจะเป็นที่บิต c_i มีค่าเป็น 0

ในกรณีของช่องสัญญาณเกาส์สีขาวแบบบวก (AWGN) จะได้ว่า

$$q_{ij}(1) = P_i = \frac{1}{1 + e^{-2y_i/\sigma^2}} \tag{4.18}$$

และ

$$q_{ij}(0) = 1 - P_i = P_r\left(c_i = 0 | y_i\right) = \frac{1}{1 + e^{+2y_i/\sigma^2}} \tag{4.19}$$

โดยที่ y_i เป็นค่าสัญญาณของบิตที่ i ซึ่งได้รับโดยตรงจากช่องสัญญาณ

σ^2 เป็นค่าความแปรปรวนของสัญญาณรบกวนเกาส์สีขาวแบบบวก

ขั้นตอนที่ 2 การอัปเดตโนดตรวจสอบ

เมื่อโหนดตรวจสอบได้รับค่า $q_{ij}(1)$ และ $q_{ij}(0)$ จากโนดบิตแล้ว โนดตรวจสอบแต่ละตัวจะคำนวณค่า $r_{ji}(0)$ และ $r_{ji}(1)$ เพื่อส่งกลับคืนไปให้โนดบิตต่อไป การคำนวณเป็นดังนี้

$$r_{ji}(0) = 0.5 + 0.5 \prod_{i' \in V_j \setminus i} \left[1 - 2q_{i'j}(1)\right] \tag{4.20}$$

$$r_{ji}(1) = 1 - r_{ji}(0) \tag{4.21}$$

โดย $V_j = \{i : h_{ji} = 1\}$ แทนเซตของโนดบิตที่ต่อเชื่อมกับโนดตรวจสอบ f_j

$V_j | i = \{i : h_{ji} = 1\} \setminus \{i\}$ แทนเซตของโนดบิตที่ต่อเชื่อมกับโนดตรวจสอบ f_j ยกเว้นโนดบิต c_i

ขั้นตอนที่ 3 การอัปเดตโนดบิต

เมื่อโนดบิตได้รับค่า $r_{ji}(0)$ และ $r_{ji}(1)$ จากโนดตรวจสอบแล้ว นำข่าวสารที่ได้รับมาคำนวณหาค่า $q_{ij}(1)$ และ $q_{ij}(0)$ เพื่อส่งกลับคืนไปให้โนดตรวจสอบต่อไปแบบวนซ้ำ หรือเพื่อนำไปตัดสินใจบิตในขั้นตอนสุดท้าย

$$q_{ij}(0) = K_{ij}\left(1 - P_i\right) \prod_{j' \in C_i \setminus j} r_{j'i}(0) \tag{4.22}$$

$$q_{ij}(1) = K_{ij} P_i \prod_{j' \in C_i \setminus j} r_{j'i}(1) \tag{4.23}$$

โดยเลือกค่า K_{ij} เพื่อให้ $q_{ij}(0) + q_{ij}(1) = 1$

โดย $C_i = \{ j : h_{ji} = 1 \}$ แทนเซตของโนดตรวจสอบที่ต่อเชื่อมกับโนดบิต c_i

$C_i \mid j = \{ j : h_{ji} = 1 \} \setminus \{ j \}$ แทนเซตของโนดตรวจสอบที่ต่อเชื่อมกับโนดบิต c_i ยกเว้นโนดตรวจสอบ f_j

ขั้นตอนที่ 4 การตัดสินแบบซอฟต์ (Soft Decision)

โอกาสที่จะเป็นไปได้ของค่า q ทุกตัว

$$Q_i(0) = K_{ij} \left(1 - P_i\right) \prod_{j \in C_i} r_{ji}(0) \tag{4.24}$$

$$Q_i(1) = K_{ij} \, P_i \prod_{j \in C_i} r_{ji}(1) \tag{4.25}$$

โดยเลือกค่า K_{ij} เพื่อให้ $Q_i(0) + Q_i(1) = 1$

ขั้นตอนที่ 5 การตัดสินใจบิตขั้นสุดท้าย

$$\hat{\mathbf{c}} = \begin{cases} 1 & ; if \ Q_i(1) > Q_i(0) \\ 0 & ; else \end{cases} \tag{4.26}$$

ถ้า $\hat{\mathbf{c}}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$ หรือจำนวนรอบการถอดรหัสมากกว่าค่าที่กำหนดแล้วให้หยุดทำงานแต่ถ้าไม่ตรงเงื่อนไขให้กลับไปทำงานในขั้นตอนที่ 2 ต่อไป

ตัวอย่างที่ 1:การถอดรหัสแอลดีพีซีด้วยตามหลักของการแพร่กระจายความเชื่อ [8]

ในตัวอย่างนี้เราใช้รหัสแอลดีพีซีแบบไบนารีชนิดสม่ำเสมอ มีน้ำหนักแถว $\rho = 6$ และน้ำหนักคอลัมน์ $\gamma = 3$ และให้เมทริกซ์พาริตีเช็ก H คือ

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

สมมติให้คำรหัสที่เราต้องการส่งคือ $\mathbf{c} = [0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 1]$ และให้คำรหัสที่รับได้คือ $\mathbf{r} = [0 \ \ 0 \ \ 0 \ \ 0 \ \ \mathbf{1} \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 1]$ ซึ่งตำแหน่งที่ 5 เกิดบิตผิดพลาดขึ้น และกำหนดให้แต่ละบิตมีค่าความน่าจะเป็นไลค์ลิฮูดคือ

$$\mathbf{P} = \begin{bmatrix} 0.78 & 0.84 & 0.81 & 0.52 & 0.45 & 0.13 & 0.82 & 0.21 & 0.75 & 0.24 \\ 0.22 & 0.16 & 0.19 & 0.48 & 0.55 & 0.87 & 0.18 & 0.79 & 0.25 & 0.76 \end{bmatrix}$$

เมทริกซ์ $\mathbf{Q}$ เริ่มต้นสามารถสร้างได้โดยค่าไลค์ลิฮูดคือ

$$\mathbf{Q}=\begin{bmatrix}0.78 & 0.84 & 0.81 & 0 & 0 & 0.13 & 0.82 & 0 & 0 & 0.24\\ 0.22 & 0.16 & 0.19 & 0 & 0 & 0.87 & 0.18 & 0 & 0 & 0.76\\ 0.78 & 0 & 0.81 & 0 & 0.45 & 0.13 & 0 & 0.21 & 0.75 & 0.24\\ 0.22 & 0 & 0.19 & 0 & 0.55 & 0.87 & 0 & 0.79 & 0.25 & 0.76\\ 0 & 0 & 0.81 & 0.52 & 0.45 & 0 & 0.82 & 0 & 0.75 & 0.24\\ 0 & 0 & 0.19 & 0.48 & 0.55 & 0 & 0.18 & 0 & 0.25 & 0.76\\ 0 & 0.84 & 0 & 0.52 & 0.45 & 0.13 & 0 & 0.21 & 0 & 0.24\\ 0 & 0.16 & 0 & 0.48 & 0.55 & 0.87 & 0 & 0.79 & 0 & 0.76\\ 0.78 & 0.84 & 0 & 0.52 & 0 & 0 & 0.82 & 0.21 & 0.75 & 0\\ 0.22 & 0.16 & 0 & 0.48 & 0 & 0 & 0.18 & 0.79 & 0.25 & 0\end{bmatrix}$$

เมทริกซ์ $\mathbf{Q}$ ประกอบไปด้วยค่าความน่าจะเป็น $q_{ij}(x)$ เพื่อนำไปใช้ในขั้นตอนการอับเดตโหนดตรวจสอบเพื่อทำการหาค่าเมทริกซ์ $\mathbf{R}$ ซึ่งเป็นค่าความน่าจะเป็น $r_{mn}(x)$ โดยใช้สมการที่ (4.20) และ (4.21)

$$r_{11}(0)=0.5+0.5\prod_{1'\in V_1\setminus 1}\left[1-2q_{1'1}(1)\right]$$
$$=0.5+0.5\left[(1-2(0.84))\times(1-2(0.81))\times(1-2(0.13))\times(1-2(0.82))\times(1-2(0.24))\right]=0.5519$$

$$r_{11}(1)=1-r_{11}(0)=0.4481$$

เมื่อหาค่าทั้งหมดจะได้เมทริกซ์ $\mathbf{R}$ ดังนี้

$$\mathbf{R}=\begin{bmatrix}0.5519 & 0.5428 & 0.5469 & 0 & 0 & 0.4607 & 0.5454 & 0 & 0 & 0.4441\\ 0.4481 & 0.4572 & 0.4531 & 0 & 0 & 0.5393 & 0.4546 & 0 & 0 & 0.5559\\ 0.4933 & 0 & 0.4940 & 0 & 0.5373 & 0.5050 & 0 & 0.5064 & 0.4925 & 0\\ 0.5067 & 0 & 0.5060 & 0 & 0.4627 & 0.4950 & 0 & 0.4936 & 0.5075 & 0\\ 0 & 0 & 0.5003 & 0.5052 & 0.4979 & 0 & 0.5003 & 0 & 0.5004 & 0.4996\\ 0 & 0 & 0.4997 & 0.4948 & 0.5021 & 0 & 0.4997 & 0 & 0.4996 & 0.5004\\ 0 & 0.5004 & 0 & 0.5076 & 0.4970 & 0.4996 & 0 & 0.4995 & 0 & 0.4994\\ 0 & 0.4996 & 0 & 0.4924 & 0.5030 & 0.5004 & 0 & 0.5005 & 0 & 0.5006\\ 0.4975 & 0.4979 & 0 & 0.4647 & 0 & 0 & 0.4978 & 0.5024 & 0.4972 & 0\\ 0.5025 & 0.5021 & 0 & 0.5353 & 0 & 0 & 0.5022 & 0.4976 & 0.5028 & 0\end{bmatrix}$$

ในขั้นตอนถัดไปใช้สมการที่ (4.22) และ (4.23) หาค่าความน่าจะเป็น $q_{11}(x)$

$$q_{11}(0)=K_{11}\left(1-P_1\right)\prod_{1'\in C_1\setminus 1}r_{1'1}(0)$$
$$=K_{11}\times 0.78\times(0.4933\times 0.4975)=0.1914K_{11}$$

$$q_{11}(1)=K_{11}P_1\prod_{1'\in C_1\setminus 1}r_{1'1}(1)$$
$$=K_{11}\times 0.22\times(0.5067\times 0.5025)=0.0560K_{11}$$

เนื่องจาก $0.1914K_{11}+0.0560K_{11}=1$ จะได้ว่า $K_{11}=\dfrac{1}{0.1914+0.0560}=4.0413$ ดังนั้น

$$q_{11}(0) = 4.0413 \times 0.78 \times (0.4933 \times 0.4975) = 0.7736$$

$$q_{11}(1) = 4.0413 \times 0.22 \times (0.5067 \times 0.5025) = 0.2264$$

ค่าความน่าจะเป็น q_{mn} ส่วนที่เหลือจะถูกอับเดตลงในเมทริกซ์ $\mathbf{Q}$ ดังนี้

$$\mathbf{Q} = \begin{bmatrix} 0.7736 & 0.8391 & 0.8065 & 0 & 0 & 0.1321 & 0.8189 & 0 & 0 & 0.2393 \\ 0.2264 & 0.1609 & 0.1935 & 0 & 0 & 0.8679 & 0.1811 & 0 & 0 & 0.7607 \\ 0.8121 & 0 & 0.8375 & 0 & 0.4450 & 0.1130 & 0 & 0.2113 & 0.7482 & 0 \\ 0.1879 & 0 & 0.1625 & 0 & 0.5550 & 0.8870 & 0 & 0.7887 & 0.2518 & 0 \\ 0 & 0 & 0.8340 & 0.4922 & 0.4841 & 0 & 0.8442 & 0 & 0.7422 & 0.2011 \\ 0 & 0 & 0.1660 & 0.5078 & 0.5159 & 0 & 0.1558 & 0 & 0.2578 & 0.7989 \\ 0 & 0.8607 & 0 & 0.4898 & 0.4851 & 0.1152 & 0 & 0.2159 & 0 & 0.2012 \\ 0 & 0.1393 & 0 & 0.5102 & 0.5149 & 0.8848 & 0 & 0.7841 & 0 & 0.7988 \\ 0.8096 & 0.8619 & 0 & 0.5327 & 0 & 0 & 0.8455 & 0.2139 & 0.7447 & 0 \\ 0.1904 & 0.1381 & 0 & 0.4673 & 0 & 0 & 0.1545 & 0.7861 & 0.2553 & 0 \end{bmatrix}$$

ในขั้นตอนสุดท้ายเป็นการตัดสินใจแบบซอฟต์โดยใช้สมการ (4.24) และ (4.25) เพื่อหาค่า $q_1(x)$ จะได้ว่า

$$\begin{aligned} Q_1(0) &= K_{11}(1-P_1)\prod_{1 \in C_1} r_{j1}(0) \\ &= K_{11} \times 0.78 \times (0.5519 \times 0.4933 \times 0.4975) = 0.1057 K_{11} \\ Q_1(1) &= K_{11} P_1 \prod_{1 \in C_1} r_{j1}(1) \\ &= K_{11} \times 0.22 \times (0.4481 \times 0.5067 \times 0.5025) = 0.0251 K_{11} \end{aligned}$$

$$K_{11} = \frac{1}{0.1057 + 0.0251} = 7.6480$$

$$q_1(0) = 7.6480 \times 0.78 \times (0.5519 \times 0.4933 \times 0.4975) = 0.8080$$

$$q_1(1) = 7.6480 \times 0.22 \times (0.4481 \times 0.5067 \times 0.5025) = 0.1920$$

$$\mathbf{Q}' = \begin{bmatrix} 0.8080 & 0.8609 & 0.8342 & 0.4974 & 0.4821 & 0.1151 & 0.8444 & 0.2156 & 0.7425 & 0.2008 \\ 0.1920 & 0.1391 & 0.1658 & 0.5026 & 0.5179 & 0.8849 & 0.1556 & 0.7844 & 0.2575 & 0.7992 \end{bmatrix}$$

ใช้สมการ (4.26) ในการตัดสินใจบิตสุดท้ายจะได้ว่า $\hat{\mathbf{c}} = [0 \quad 0 \quad 0 \quad 1 \quad \mathbf{1} \quad 1 \quad 0 \quad 1 \quad 0 \quad 1]$ ซึ่งยังไม่ตรงกับคำรหัสเดิมที่เราต้องการจะส่ง ดังนั้นการวนซ้ำการถอดรหัสจึงมีความจำเป็นในการถอดรหัสแอลดีพีซีด้วย ในตัวอย่างนี้จะต้องใช้การวนซ้ำการถอดรหัสสามรอบจึงสามารถแก้ไขข้อผิดพลาดที่เกิดขึ้นได้ โดยในรอบสุดท้ายเมทริกซ์ $\mathbf{R}$, $\mathbf{Q}$ และ $\mathbf{Q}'$ มีค่าความน่าจะเป็นดังนี้

$$\mathbf{R} = \begin{bmatrix} 0.5500 & 0.5401 & 0.5444 & 0 & 0 & 0.4631 & 0.5426 & 0 & 0 & 0.4479 \\ 0.4500 & 0.4599 & 0.4556 & 0 & 0 & 0.5369 & 0.4574 & 0 & 0 & 0.5521 \\ 0.4931 & 0 & 0.4937 & 0 & 0.5454 & 0.5055 & 0 & 0.5075 & 0.4913 & 0 \\ 0.5069 & 0 & 0.5063 & 0 & 0.4546 & 0.4945 & 0 & 0.4925 & 0.5087 & 0 \\ 0 & 0 & 0.5000 & 0.5002 & 0.5026 & 0 & 0.5000 & 0 & 0.5000 & 0.5000 \\ 0 & 0 & 0.5000 & 0.4998 & 0.4974 & 0 & 0.5000 & 0 & 0.5000 & 0.5000 \\ 0 & 0.5000 & 0 & 0.5004 & 0.5039 & 0.5000 & 0 & 0.5000 & 0 & 0.5000 \\ 0 & 0.5000 & 0 & 0.4996 & 0.4961 & 0.5000 & 0 & 0.5000 & 0 & 0.5000 \\ 0.4966 & 0.4971 & 0 & 0.4579 & 0 & 0 & 0.4970 & 0.5037 & 0.4957 & 0 \\ 0.5034 & 0.5029 & 0 & 0.5421 & 0 & 0 & 0.5030 & 0.4963 & 0.5043 & 0 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 0.7729 & 0.8384 & 0.8061 & 0 & 0 & 0.1325 & 0.8182 & 0 & 0 & 0.2400 \\ 0.2271 & 0.1616 & 0.1939 & 0 & 0 & 0.8675 & 0.1818 & 0 & 0 & 0.7600 \\ 0.8104 & 0 & 0.8359 & 0 & 0.4565 & 0.1142 & 0 & 0.2125 & 0.7467 & 0 \\ 0.1896 & 0 & 0.1641 & 0 & 0.5435 & 0.8858 & 0 & 0.7875 & 0.2533 & 0 \\ 0 & 0 & 0.8324 & 0.4782 & 0.4993 & 0 & 0.8423 & 0 & 0.7401 & 0.2040 \\ 0 & 0 & 0.1676 & 0.5218 & 0.5007 & 0 & 0.1577 & 0 & 0.2599 & 0.7960 \\ 0 & 0.8590 & 0 & 0.4780 & 0.4980 & 0.1164 & 0 & 0.2175 & 0 & 0.2039 \\ 0 & 0.1410 & 0 & 0.5220 & 0.5020 & 0.8836 & 0 & 0.7825 & 0 & 0.7961 \\ 0.8082 & 0.8604 & 0 & 0.5206 & 0 & 0 & 0.8439 & 0.2150 & 0.7434 & 0 \\ 0.1918 & 0.1396 & 0 & 0.4794 & 0 & 0 & 0.1561 & 0.7850 & 0.2566 & 0 \end{bmatrix}$$

$$\mathbf{Q}' = \begin{bmatrix} 0.8061 & 0.8590 & 0.8324 & 0.4784 & 0.5019 & 0.1164 & 0.8423 & 0.2175 & 0.7401 & 0.2040 \\ 0.1939 & 0.1410 & 0.1676 & 0.5216 & 0.4981 & 0.8836 & 0.1577 & 0.7825 & 0.2599 & 0.7960 \end{bmatrix}$$

$$\hat{\mathbf{c}} = \begin{bmatrix} 0 & 0 & 0 & 1 & \mathbf{0} & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

ดังนั้นในผลการตัดสินใจบิตสุดท้ายจากความน่าจะเป็นที่มีค่ามากที่สุดในแต่ละบิตได้ผลแสดงให้เห็นใน $\hat{\mathbf{c}}$ สังเกตุได้ว่าในการวนซ้ำการถอดรหัสในรอบที่สามสามารถแก้ไข้บิตผิดพลาดที่เกิดขึ้นในตำแหน่งที่ 5 ได้อย่างถูกต้อง

4.3.2 วิธีการที่นำเสนอ

วิธีการดัดแปลงการถอดรหัสตามหลักของการแพร่กระจายความเชื่อ (Belief Propagation) ซึ่งมีอยู่ 4 ขั้นตอนดังนี้

ขั้นตอนที่ 1 การคำนวณค่าเริ่มต้น

ขั้นตอนนี้คำนวณหาค่าเริ่มต้น $q_{ij}(1)$ และ $q_{ij}(0)$ จากสัญญาณ y_i ที่ได้รับจากช่องสัญญาณ เพื่อใช้เป็นข่าวสารที่จะส่งจากโนดบิต i ไปยังโนดตรวจสอบ j ที่เชื่อมต่อด้วย โดยที่

- $q_{ij}(1) = P_i = P_r(c_i = 1 \mid y_i)$ คือความน่าจะเป็นที่บิต c_i มีค่าเป็น 1 และ
- $q_{ij}(0) = 1 - P_i = P_r(c_i = 0 \mid y_i)$ คือความน่าจะเป็นที่บิต c_i มีค่าเป็น 0

ในกรณีของช่องสัญญาณเกาส์สีขาวแบบบวก (AWGN) จะได้ว่า

$$q_{ij}(1) = P_i = \frac{1}{1+e^{-2y_i/\sigma^2}} \tag{4.27}$$

และ

$$q_{ij}(0) = 1 - P_i = P_r\left(c_i = 0 \mid y_i\right) = \frac{1}{1+e^{+2y_i/\sigma^2}} \tag{4.28}$$

โดยที่ y_i เป็นค่าสัญญาณของบิตที่ i ซึ่งได้รับโดยตรงจากช่องสัญญาณ

σ^2 เป็นค่าความแปรปรวนของสัญญาณรบกวนเกาส์สีขาวแบบบวก

ขั้นตอนที่ 2 การอับเดตโหนดตรวจสอบ

เมื่อโนดตรวจสอบได้รับค่า $q_{ij}(1)$ และ $q_{ij}(0)$ จากโนดบิตแล้ว โนดตรวจสอบแต่ละตัวจะคำนวณค่า $r_{ji}(0)$ และ $r_{ji}(1)$ เพื่อส่งกลับคืนไปให้โนดบิตต่อไป การคำนวณเป็นดังนี้

$$r_{ji}(0) = 0.5 + 0.5\prod_{i' \in V_j \setminus i}\left[1 - 2q_{i'j}(1)\right] \tag{4.29}$$

$$r_{ji}(1) = 1 - r_{ji}(0) \tag{4.30}$$

โดย $V_j = \{i : h_{ji} = 1\}$ แทนเซตของโนดบิตที่ต่อเชื่อมกับโนดตรวจสอบ f_j

$V_j \mid i = \{i : h_{ji} = 1\} \setminus \{i\}$ แทนเซตของโนดบิตที่ต่อเชื่อมกับโนดตรวจสอบ f_j ยกเว้นโนดบิต c_i

ขั้นตอนที่ 3 การตัดสินแบบซอฟต์ (Soft Decision)

โอกาสที่จะเป็นไปได้ของค่า q ทุกตัว

$$Q_i(0) = K_{ij}\left(1 - P_i\right)\prod_{j \in C_i} r_{ji}(0) \tag{4.31}$$

$$Q_i(1) = K_{ij}\, P_i \prod_{j \in C_i} r_{ji}(1) \tag{4.32}$$

โดยเลือกค่า K_{ij} เพื่อให้ $Q_i(0) + Q_i(1) = 1$

ขั้นตอนที่ 4 การตัดสินใจบิตขั้นสุดท้าย

$$\hat{\mathbf{c}} = \begin{cases} 1 & ; if\ Q_i(1) > Q_i(0) \\ 0 & ; else \end{cases} \tag{4.33}$$

ถ้า $\hat{\mathbf{c}}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$ หรือจำนวนรอบการถอดรหัสมากกว่าค่าที่กำหนดแล้วให้หยุดทำงานแต่ถ้าไม่ตรงเงื่อนไขให้กลับไปทำงานในขั้นตอนที่ 2 ต่อไป

ตัวอย่างที่ 2 :วิธีการดัดแปลงการถอดรหัสตามหลักของการแพร่กระจายความเชื่อ

ตัวอย่างนี้ใช้เมทริกซ์พาริตีเช็ก H เดียวกับในตัวอย่างที่ 1 เพื่อเปรียบเทียบขั้นตอนและระยะเวลาในการประมวลผลให้เข้าใจมากยิ่งขึ้นดังนั้นคำรหัสและคำรหัสที่รับได้ก็จะใช้ตัวอย่างเดียวกันกับตัวอย่างที่ 1 เราเริ่มต้นด้วยการตั้งค่าไลค์ลิฮูดของแต่ละบิตในเมทริกซ์ **P** ดังนี้

$$\mathbf{P} = \begin{bmatrix} 0.78 & 0.84 & 0.81 & 0.52 & 0.45 & 0.13 & 0.82 & 0.21 & 0.75 & 0.24 \\ 0.22 & 0.16 & 0.19 & 0.48 & 0.55 & 0.87 & 0.18 & 0.79 & 0.25 & 0.76 \end{bmatrix}$$

เมทริกซ์ **Q** เริ่มต้นสามารถสร้างได้โดยค่าไลค์ลิฮูดคือ

$$\mathbf{Q} = \begin{bmatrix} 0.78 & 0.84 & 0.81 & 0 & 0 & 0.13 & 0.82 & 0 & 0 & 0.24 \\ 0.22 & 0.16 & 0.19 & 0 & 0 & 0.87 & 0.18 & 0 & 0 & 0.76 \\ 0.78 & 0 & 0.81 & 0 & 0.45 & 0.13 & 0 & 0.21 & 0.75 & 0.24 \\ 0.22 & 0 & 0.19 & 0 & 0.55 & 0.87 & 0 & 0.79 & 0.25 & 0.76 \\ 0 & 0 & 0.81 & 0.52 & 0.45 & 0 & 0.82 & 0 & 0.75 & 0.24 \\ 0 & 0 & 0.19 & 0.48 & 0.55 & 0 & 0.18 & 0 & 0.25 & 0.76 \\ 0 & 0.84 & 0 & 0.52 & 0.45 & 0.13 & 0 & 0.21 & 0 & 0.24 \\ 0 & 0.16 & 0 & 0.48 & 0.55 & 0.87 & 0 & 0.79 & 0 & 0.76 \\ 0.78 & 0.84 & 0 & 0.52 & 0 & 0 & 0.82 & 0.21 & 0.75 & 0 \\ 0.22 & 0.16 & 0 & 0.48 & 0 & 0 & 0.18 & 0.79 & 0.25 & 0 \end{bmatrix}$$

เมทริกซ์ **Q** ประกอบไปด้วยค่าความน่าจะเป็น $q_{ij}(x)$ เพื่อนำไปใช้ในขั้นตอนการอัปเดตโหนดตรวจสอบเพื่อทำการหาค่าเมทริกซ์ **R** ซึ่งเป็นค่าความน่าจะเป็น $r_{mn}(x)$ โดยใช้สมการที่ (4.20) และ (4.21)

$$r_{11}(0) = 0.5 + 0.5\prod_{1' \in V_1 \setminus 1} [1 - 2q_{1'1}(1)]$$

$$= 0.5 + 0.5[(1-2(0.84)) \times (1-2(0.81)) \times (1-2(0.13)) \times (1-2(0.82)) \times (1-2(0.24))] = 0.55$$

$$r_{11}(1) = 1 - r_{11}(0) = 0.45$$

เมื่อหาค่าทั้งหมดจะได้เมทริกซ์ **R** ดังนี้

$$\mathbf{R}=\begin{bmatrix}0.5519 & 0.5428 & 0.5469 & 0 & 0 & 0.4607 & 0.5454 & 0 & 0 & 0.4441\\ 0.4481 & 0.4572 & 0.4531 & 0 & 0 & 0.5393 & 0.4546 & 0 & 0 & 0.5559\\ 0.4933 & 0 & 0.4940 & 0 & 0.5373 & 0.5050 & 0 & 0.5064 & 0.4925 & 0\\ 0.5067 & 0 & 0.5060 & 0 & 0.4627 & 0.4950 & 0 & 0.4936 & 0.5075 & 0\\ 0 & 0 & 0.5003 & 0.5052 & 0.4979 & 0 & 0.5003 & 0 & 0.5004 & 0.4996\\ 0 & 0 & 0.4997 & 0.4948 & 0.5021 & 0 & 0.4997 & 0 & 0.4996 & 0.5004\\ 0 & 0.5004 & 0 & 0.5076 & 0.4970 & 0.4996 & 0 & 0.4995 & 0 & 0.4994\\ 0 & 0.4996 & 0 & 0.4924 & 0.5030 & 0.5004 & 0 & 0.5005 & 0 & 0.5006\\ 0.4975 & 0.4979 & 0 & 0.4647 & 0 & 0 & 0.4978 & 0.5024 & 0.4972 & 0\\ 0.5025 & 0.5021 & 0 & 0.5353 & 0 & 0 & 0.5022 & 0.4976 & 0.5028 & 0\end{bmatrix}$$

ในขั้นตอนถัดไปใช้สมการที่ (4.31) และ (4.32) หาค่าความน่าจะเป็น $q_1(x)$ เพื่อทำการตัดสินใจแบบซอฟต์

$$\begin{aligned}Q_1(0) &= K_{11}(1-P_1)\prod_{1\in C_1} r_{j1}(0)\\ &= K_{11}\times 0.78\times(0.5519\times 0.4933\times 0.4975) = 0.1057K_{11}\\ Q_1(1) &= K_{11}P_1\prod_{1\in C_1} r_{j1}(1)\\ &= K_{11}\times 0.22\times(0.4481\times 0.5067\times 0.5025) = 0.0251K_{11}\end{aligned}$$

$$K_{11}=\frac{1}{0.1057+0.0251}=7.6480$$

$$q_1(0)=7.6480\times 0.78\times(0.5519\times 0.4933\times 0.4975)=0.8080$$

$$q_1(1)=7.6480\times 0.22\times(0.4481\times 0.5067\times 0.5025)=0.1920$$

$$\mathbf{Q}'=\begin{bmatrix}0.8080 & 0.8609 & 0.8342 & 0.4974 & 0.4821 & 0.1151 & 0.8444 & 0.2156 & 0.7425 & 0.2008\\ 0.1920 & 0.1391 & 0.1658 & 0.5026 & 0.5179 & 0.8849 & 0.1556 & 0.7844 & 0.2575 & 0.7992\end{bmatrix}$$

ใช้สมการ (4.33) ในการตัดสินใจบิตสุดท้ายจะได้ว่า $\hat{\mathbf{c}}=[0\ \ 0\ \ 0\ \ 1\ \ \mathbf{1}\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$ ซึ่งยังไม่ตรงกับคำรหัสเดิมที่เราต้องการจะส่ง ดังนั้นเราจึงต้องมีการวนซ้ำการถอดรหัสเช่นเดียวกับในตัวอย่างที่ 1 โดยมีการเปลี่ยนแปลงการอัปเดตค่าความน่าจะเป็นในเมทริกซ์ $\mathbf{Q}$ โดยทำการอัปเดทค่าเมทริกซ์ $\mathbf{P}$ ใหม่ด้วยการแทนค่าความน่าจะเป็นที่ได้จากเมทริกซ์ $\mathbf{Q}'$ ในรอบก่อนหน้าลงไปในเมทริกซ์ $\mathbf{P}$ แล้วกำหนดให้เป็นเมทริกซ์ $\mathbf{P}'$ ดังนั้นในรอบที่สองจะมีค่าความน่าจะเป็นไลค์ลิฮูดใหม่คือ

$$\mathbf{P}'=\begin{bmatrix}0.8080 & 0.8609 & 0.8342 & 0.4974 & 0.4821 & 0.1151 & 0.8444 & 0.2156 & 0.7425 & 0.2008\\ 0.1920 & 0.1391 & 0.1658 & 0.5026 & 0.5179 & 0.8849 & 0.1556 & 0.7844 & 0.2575 & 0.7992\end{bmatrix}$$

เมื่อทำการอัปเดตค่า $\mathbf{P}'$ แล้วให้นำค่าที่ได้จากเมทริกซ์ $\mathbf{P}'$ มาทำการอัปเดทเมทริกซ์ $\mathbf{Q}$ ใหม่ ดังนั้นในการวันซ้ำรอบที่สองค่าความน่าจะเป็นของเมทริกซ์ $\mathbf{Q}$, $\mathbf{R}$ และ $\mathbf{Q}'$ เป็นดังนี้

$$\mathbf{Q} = \begin{bmatrix} 0.8080 & 0.8609 & 0.8342 & 0 & 0 & 0.1151 & 0.8444 & 0 & 0 & 0.2008 \\ 0.1920 & 0.1391 & 0.1658 & 0 & 0 & 0.8849 & 0.1556 & 0 & 0 & 0.7992 \\ 0.8080 & 0 & 0.8342 & 0 & 0.4821 & 0.1151 & 0 & 0.2156 & 0.7425 & 0 \\ 0.1920 & 0 & 0.1658 & 0 & 0.5179 & 0.8849 & 0 & 0.7844 & 0.2575 & 0 \\ 0 & 0 & 0.8342 & 0.4974 & 0.4821 & 0 & 0.8444 & 0 & 0.7425 & 0.2008 \\ 0 & 0 & 0.1658 & 0.5026 & 0.5179 & 0 & 0.1556 & 0 & 0.2575 & 0.7992 \\ 0 & 0.8609 & 0 & 0.4974 & 0.4821 & 0.1151 & 0 & 0.2156 & 0 & 0.2008 \\ 0 & 0.1391 & 0 & 0.5026 & 0.5179 & 0.8849 & 0 & 0.7844 & 0 & 0.7992 \\ 0.8080 & 0.8609 & 0 & 0.4974 & 0 & 0 & 0.8444 & 0.2156 & 0.7425 & 0 \\ 0.1920 & 0.1391 & 0 & 0.5026 & 0 & 0 & 0.1556 & 0.7844 & 0.2575 & 0 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.5765 & 0.5653 & 0.5705 & 0 & 0 & 0.4388 & 0.5685 & 0 & 0 & 0.4212 \\ 0.4235 & 0.4347 & 0.4295 & 0 & 0 & 0.5612 & 0.4315 & 0 & 0 & 0.5788 \\ 0.4975 & 0 & 0.4977 & 0 & 0.5437 & 0.5020 & 0 & 0.5028 & 0.4968 & 0 \\ 0.5025 & 0 & 0.5023 & 0 & 0.4563 & 0.4980 & 0 & 0.4972 & 0.5032 & 0 \\ 0 & 0 & 0.5000 & 0.5024 & 0.5004 & 0 & 0.5000 & 0 & 0.5000 & 0.5000 \\ 0 & 0 & 0.5000 & 0.4976 & 0.4996 & 0 & 0.5000 & 0 & 0.5000 & 0.5000 \\ 0 & 0.5000 & 0 & 0.5034 & 0.5005 & 0.5000 & 0 & 0.5000 & 0 & 0.5000 \\ 0 & 0.5000 & 0 & 0.4966 & 0.4995 & 0.5000 & 0 & 0.5000 & 0 & 0.5000 \\ 0.5004 & 0.5003 & 0 & 0.4577 & 0 & 0 & 0.5003 & 0.4996 & 0.5005 & 0 \\ 0.4996 & 0.4997 & 0 & 0.5423 & 0 & 0 & 0.4997 & 0.5004 & 0.4995 & 0 \end{bmatrix}$$

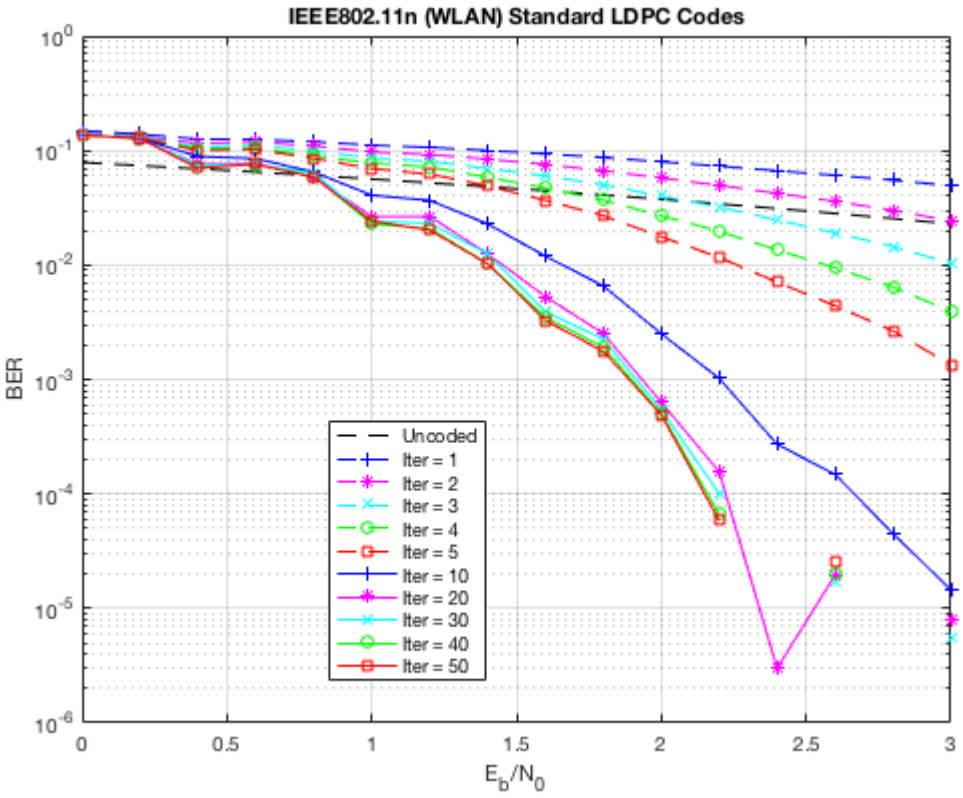
$$\mathbf{Q}' = \begin{bmatrix} 0.8503 & 0.8896 & 0.8688 & 0.4609 & 0.5267 & 0.0930 & 0.8774 & 0.2172 & 0.7404 & 0.1546 \\ 0.1497 & 0.1104 & 0.1312 & 0.5391 & 0.4733 & 0.9070 & 0.1226 & 0.7828 & 0.2596 & 0.8454 \end{bmatrix}$$

$$\hat{\mathbf{c}} = \begin{bmatrix} 0 & 0 & 0 & 1 & \mathbf{0} & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

สังเกตุว่าในวิธีการดัดแปลงการถอดรหัสตามหลักของการแพร่กระจายความเชื่อใช้การถอดรหัสวนซ้ำเพียงสองรอบเท่านั้นก็สามารถแก้ไขบิตผิดพลาดในตำแหน่งที่ 5 ได้ ซึ่งลดลงกว่าอังกอริทึมแพร่กระจายความเชื่อแบบเดิม ดังนั้นการดัดแปลงการถอดรหัสตามหลักการแพร่กระจายความเชื่อนี้นอกจากจะแก้ไขบิตผิดพลาดได้อย่างมีประสิทธิภาพแล้วยังช่วยลดระยะเวลาประมวลผลในการวนซ้ำการถอดรหัสได้อย่างมีประสิทธิภาพ

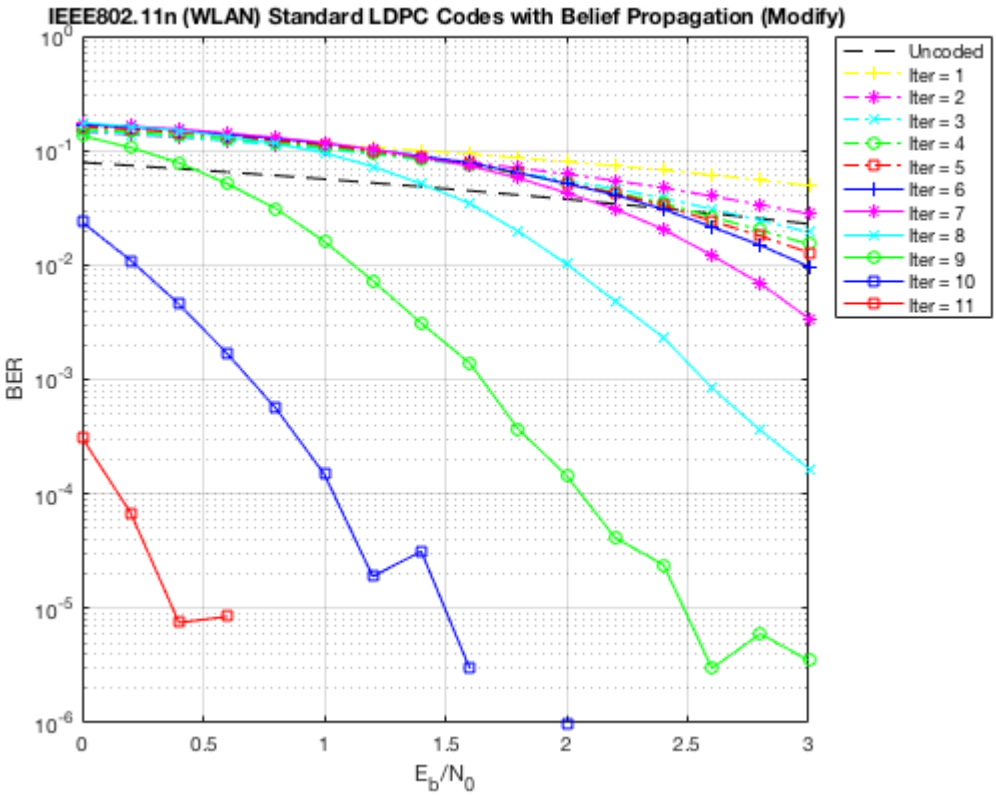
4.4 ผลการทดสอบสมรรถนะตัวถอดรหัสแอลดีพีซีกับมาตรฐาน IEEE 802.11n (WLAN)

ในวิทยานิพนธ์นี้เราได้ทำการทดสอบตัวถอดรหัสแอลดีพีซีตามหลักการแพร่กระจายความเชื่อกับมาตรฐาน IEEE 802.11n (WLAN) เนื่องจากเป็นมาตรฐานที่ได้รับการยอมรับและใช้กันอย่างแพร่หลายในปัจจุบัน ซึ่งตัวรหัสแอลดีพีซีที่ใช้ในมาตรฐาน IEEE 802.11n เองนั้นก็มีการออกแบบที่มีประสิทธิภาพที่ดีเป็นอย่างมาก การทดสอบด้วยมาตรฐานนี้จึงเป็นตัวเปรียบเทียบที่ดีในการทดสอบสมรรถนะระหว่างตัวถอดรหัสแอลดีพีซีตามหลักการแพร่กระจายความเชื่อด้วยวิธีปกติและวิธีที่ดัดแปลงใหม่



รูปที่ 4.4 สมรรถนะของรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n ความยาว $N = 648$ ที่การวนซ้ำการถอดรหัสสูงสุด 50 รอบ

จากรูปที่ 4.5 เป็นกราฟสมรรถนะของรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n (WLAN) ด้วยอัลกอริทึมแพร่กระจายความเชื่อ โดยแสดงสมรรถนะของการถอดรหัสสูงสุด 50 รอบ สังเกตุว่าในแต่ละรอบของการวนซ้ำการถอดรหัส ยิ่งมีการวนซ้ำมากขึ้นสมรรถนะของการลดบิตผิดพลาดก็ยิ่งเพิ่มขึ้นตามไปด้วย โดยในรอบที่ 50 นั้นสามารถลดบิตผิดพลาดได้ต่ำกว่า 3 dB ซึ่งในรูปถัดไปจะเป็นการถอดรหัสด้วยตัวถอดรหัสซึ่งดัดแปลงอัลกอริทึมแพร่กระจายความเชื่อโดยใช้ค่าพารามิเตอร์ต่างๆ เหมือนกับการถอดรหัสตามปกติทั้งหมด



รูปที่ 4.5 สมรรถนะของรหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n ความยาว $N = 648$ ด้วยวิธีการดัดแปลงอัลกอริทึมการแพร่กระจายความเชื่อ

จากผลการทดสอบการถอดรหัสด้วยตัวถอดรหัสวิธีดัดแปลงการแพร่กระจายความเชื่อซึ่งวนซ้ำการถอดรหัส 50 รอบเท่ากันกับวิธีการถอดรหัสแบบก่อนหน้านี้ สังเกตได้ว่าการถอดรหัสด้วยตัวถอดรหัสแบบใหม่ใช้การวนซ้ำการถอดรหัสเพียง 11 รอบเท่านั้นก็สามารถลดอัตราบิตผิดพลาดได้เป็นอย่างมากซึ่งลดลงได้ต่ำกว่า 1 dB ซึ่งเมื่อทำการวนซ้ำการถอดรหัสในรอบที่ 12 เป็นต้นไป โดยการส่งบิตทั้งหมดที่สองล้านบิตก็ยังสามารถแก้ไขบิตผิดพลาดได้ทั้งหมดจึงสามารถแสดงผลกราฟ BER ของการวนซ้ำการถอดรหัสได้ที่ 11 รอบเท่านั้น

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

ในวิทยานิพนธ์นี้ได้ศึกษาแนวทางการออกแบบรหัสแอลดีพีซี โดยได้ศึกษาการออกแบบเมทริกซ์พาริตีเช็กแบบนอนไบนารีโดยมีพื้นฐานการออกแบบเมทริกซ์พาริตีเช็กแบบไบนารีที่มีน้ำหนักคอลัมน์เท่ากับ 2 จากนั้นจึงทำการใส่ค่าตัวเลขแบบสุ่มในเมทริกซ์พาริตีเช็กในจำนวนสมาชิกที่ไม่มีค่าเป็นศูนย์ซึ่งนิยามบนสนามกาลัวส์ โดยได้แบ่งการใส่เลขนอนไบนารีสองแบบคือแบบสุ่มและแบบไบนารีอิมเมจ จากผลการทดสอบพบว่าวิธีการเลือกใส่เลขนอนไบนารีแบบไบนารีอิมเมจให้สมรรถนะที่ดีกว่าแบบสุ่มประมาร 0.15 dB นอกจากนี้ยังได้เปรียบเทียบสมรรถนะกับวิถีการออกแบบเมทริกซ์พาริตีเช็กแบบโพรโตคอลของนาซ่าก็พบว่าให้ประสิทธิภาพที่ดีกว่าอย่างเห็นได้ชัด

ในส่วนที่สองของวิทยานิพนธ์ได้เสนอวิธีการดัดแปลงอัลกอริทึกการถอดรหัสแอลดีพีซีโดยหลักการแพร่กระจายความเชื่อ โดยตัวถอดรหัสตัวใหม่นี้ได้มีการดัดแปลงวิธีการประมวลผลตัดสินใจของโหนดบิตใหม่ทำให้มีขั้นตอนในการประมวลผลที่ลดลงและมีความแม่นยำในประมวลผลมากขึ้นกว่าตัวถอดรหัสแบบปกติ ในการทดสอบประสิทธิภาพได้ใช้รหัสแอลดีพีซีตามมาตรฐาน IEEE 802.11n (WLAN) ซึ่งเป็นมาตรฐานที่ได้รับการยอมรับว่ามีประสิทธิภาพที่ดีและมีการใช้อย่างแพร่หลายในปัจจุบัน จากการทดสอบสมรรถนะตัวถอดรหัสด้วยวิธีใหม่พบว่าให้สมรรถนะที่ดียิ่งขึ้นไปอีก โดยเมื่อทดสอบการวนซ้ำพบว่าเพียงการวนซ้ำการถอดรหัสที่ 11 รอบก็สามารถแก้ไขข้อผิดพลาดได้หมด

การทดสอบตัวถอดรหัสแอลดีพีซีโดยการดัดแปลงอัลกอริทึมการแพร่กระจายความเชื่อยังจะกัดการทดสอบอยู่กับมาตรฐาน IEEE 802.11n ซึ่งยังไม่เคยมีการทดสอบการรหัสแอลดีพีซีที่มีการออกแบบเมทริกซ์พาริตีเช็กแบบอื่นๆ ซึ่งเชื่อว่าตัวถอดรหัสตัวใหม่นี้สามารถใช้ได้กับเมทริกซ์พาริตีเช็กทุกแบบโดยยิ่งมีน้ำหนักคอลัมน์มาก ก็ยิ่งทำให้สมรรถนะเมื่อใช้กับตัวถอดรหัสแบบใหม่ดีขึ้น นอกจากนี้นอกเหนือจากตัวถอดรหัสแอลดีพีซีเองก็เชื่อเป็นว่าอย่างยิ่งว่า การถอดรหัสแบบอื่นๆที่ใช้อัลกอริทึมตามหลักการแพ่กระจายความเชื่อ ถ้าใช้ตัวถอดรหัสแบบใหม่ก็จะช่วยเพิ่มสมรรถนะในการลดอัตราบิตผิดพลาดด้วยเช่นกัน

รายการอ้างอิง

[1] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 5, no. 1, pp. 3-55, 1948.

[2] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory,* vol. 8, no. 1, pp. 21-28, 1962.

[3] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding: 5th IMA Conference Cirencester, UK, December 18–20, 1995 Proceedings*, C. Boyd, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 100-111.

[4] C. Sae-Young, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters,* vol. 5, no. 2, pp. 58-60, 2001.

[5] B. Ammar, B. Honary, K. Yu, X. Jun, and S. Lin, "Construction of low-density parity-check codes based on balanced incomplete block designs," *IEEE Transactions on Information Theory,* vol. 50, no. 6, pp. 1257-1269, 2004.

[6] H. Xiao-Yu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, 2001, vol. 2, pp. 995-1001 vol.2.

[7] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, 2005.

[8] R. A. Carrasco and M. Johnston, *Non-Binary Error Control Coding for Wireless Communication and Data Storage*. Wiley Publishing, 2009, p. 322.

[9] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over GF(2q)," in *Proceedings 2003 IEEE Information Theory Workshop (Cat. No.03EX674)*, 2003, pp. 70-73.

[10] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," in *1998 Information Theory Workshop (Cat. No.98EX131)*, 1998, pp. 70-71.

[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, no. 7, pp. 2711-2736, 2001.

[12] *Short Blocklength LDPC Codes for TC Synchronization and Channel Codding*, 2012.

[13] G. Liva, S. Scalise, E. Paolini, and M. Chiani, "Turbo Codes Based on Time-Variant Memory-1 Convolutional Codes over Fq," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1-6.

[14] D. Divsalar and L. Dolecek, "On the typical minimum distance of protograph-based non-binary LDPC codes," in *2012 Information Theory and Applications Workshop*, 2012, pp. 192-198.

[15] H. Xiao-Yu, E. Eleftheriou, and D. M. Arnold, "Irregular progressive edge-growth (PEG) Tanner graphs," in *Proceedings IEEE International Symposium on Information Theory*, 2002, p. 480.

[16] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," in *IPN Progress Report*, 2003, pp. 42-154.

[17] Nut Tantibut *et al.*, "BER Performance Study of Column Weight Two Non-Binary LDPC Codes with Predetermined Girth," presented at the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2016), Okinawa, Japan, July 11, 2016.

[18] Nut Tantibut, Ambar Bajpai, Gan Srirutchataboon, Lunchakorn Wuttisittikulkij, and P. Kovintavewat, "Performance and Evaluation of Regular (2, ρ) Non-Binary LDPC Codes over GF(q) based on Protograph Design and A New Random Construction Method," presented at the International Conference of "Multidisciplinary Approaches on UN Sustainable Development Goals" (UNSDGs 2016), Bangkok, Thailand, December 28, 2016.

[19] "IEEE Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE P802.11-REVmb/D12, November 2011 (Revision of IEEE Std 802.11-2007, as amended*

by IEEEs 802.11k-2008, 802.11r-2008, 802.11y-2008, 802.11w-2009, 802.11n-2009, 802.11p-2010, 802.11z-2010, 802.11v-2011, 802.11u-2011, and 802.11s-2011), pp. 1-2910, 2011.

[20] Nut Tantibut, Ambar Bajpai, Lunchakorn Wuttisittikulkij, and P. Kovintavewat, "Extended Decoding Performance of LDPC and NB-LDPC Codes with IEEE 802.11n Standard," presented at the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2015), Seoul, Korea, July 1, 2015.

ภาคผนวก

ประวัติผู้เขียนวิทยานิพนธ์

ว่าที่ร้อยตรีณัฐ ตันติบุตร สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปี พ.ศ.2556 ในระหว่างเข้ารับการศึกษา ได้มีโอกาสเข้าร่วมและเป็นผู้นำกิจกรรมในด้านต่างๆ อาทิเช่น

- กิจกรรมรับน้องใหม่ (Staff) องค์การนักศึกษา มหาวิทยาลัยขอนแก่น พ.ศ.2549

- คณะกรรมการ (ฝ่ายจริยธรรม) องค์การนักศึกษา มหาวิทยาลัยขอนแก่น พ.ศ.2549

- ค่ายร่วมอาสาพัฒนาชนบทมหาวิทยาลัยขอนแก่นและมหาวิทยาลัยเชียงใหม่ ครั้งที่ 16 องค์การนักศึกษา มหาวิทยาลัยขอนแก่น พ.ศ.2550

- ชมรมนักศึกษาวิชาทหาร (ประธานรุ่นก่อตั้งชมรมฯ) องค์การนักศึกษา มหาวิทยาลัยขอนแก่น พ.ศ.2551

จากนั้นจึงเข้ารับการศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ.2556 ในระหว่างศึกษาได้มีโอกาสเข้าร่วมและเป็นผู้นำกิจกรรมในด้านต่างๆ อาทิเช่น

- กลุ่มความร่วมมือนิสิตระดับบัณฑิตศึกษา (ประธานชมรมฯ) ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย รุ่นที่ 6 (EEPSACU#6) พ.ศ.2557

- ชมรมนิสิตระดับบัณฑิตศึกษา (รองประธานชมรมฯ คนที่ 2) คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย รุ่นที่ 3 (EGSACU#3) พ.ศ.2557

- กรรมการนิสิตระดับบัณฑิตศึกษา (ฝ่ายสวัสดิการ) จุฬาลงกรณ์มหาวิทยาลัย พ.ศ. 2557

และคาดว่าจะสำเร็จการศึกษาในปี พ.ศ.2560