



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

ปริญญา

วิศวกรรมไฟฟ้า	วิศวกรรมไฟฟ้า
สาขา	ภาควิชา
เรื่อง	การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับการถอดรหัสแบบแมป สำหรับระบบสื่อสารข้อมูลภาพ MPEG-4 บนช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง Design of Reconfigurable Random Interleavers with MAP Decoding for MPEG-4 Image Transmission System over Rician Block-Fading Channels
นามผู้วิจัย	นายอนุกุล พิมเสน
ได้พิจารณาเห็นชอบโดย	
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	(ผู้ช่วยศาสตราจารย์ศรีจิตรา เจริญลาภนพรัตน์, Ph.D.)
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	(ผู้ช่วยศาสตราจารย์พูนลาภ ลามศรีจันทร์, Ph.D.)
หัวหน้าภาควิชา	(รองศาสตราจารย์วิชัย สุระพัฒน์, วศ.ม.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์กัญญา วีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ เดือน พ.ศ.

วิทยานิพนธ์

เรื่อง

การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับการถอดรหัสแบบแมป
สำหรับระบบสื่อสารข้อมูลภาพ MPEG-4 บนช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

Design of Reconfigurable Random Interleavers with MAP Decoding for MPEG-4 Image
Transmission System over Rician Block-Fading Channels

โดย

นายบุญกุล พิมเสน

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2554

นุกูล พิมเสน 2554: การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับ การถอดรหัสแบบแมป สำหรับระบบสื่อสารข้อมูลภาพ MPEG-4 บนช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง ปรินซ์พาลวิทยาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) สาขา วิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์ศรีจิตรา เจริญลาภนพรัตน์, Ph.D. 86 หน้า

การสื่อสารไร้สายเป็นเทคโนโลยีที่มีพัฒนาการอย่างต่อเนื่อง โดยเฉพาะอย่างยิ่งเราสนใจ การใช้งานทางด้านสื่อประสมในหลายๆด้าน งานวิจัยนี้เป็นส่วนหนึ่งของการพัฒนาระบบรับส่ง รูปภาพ MPEG-4 บนช่องสัญญาณ ไร้สายภายในอาคาร ที่จำลองช่องสัญญาณแบบ Rician บล็อก เฟดดิ้ง โดยงานนี้จะมุ่งเน้นในการบรรเทาความเสียหายต่อสัญลักษณ์เป็นกลุ่ม ที่เป็นผลกระทบ จากการเฟดดิ้ง โดยก่อนหน้านี้ใช้การสลับลำดับข้อมูลแบบสุ่มเพื่อแก้ปัญหา นี้ อย่างไรก็ตามพบว่า ระบบมีความซับซ้อนสูงและทำให้ระบบช้าลง ดังนั้นเราจึงพยายามปรับปรุงการสลับลำดับข้อมูล แบบสุ่ม โดยออกแบบการสลับลำดับข้อมูลแบบสุ่มที่ปรับปรุงแล้ว เรียกว่าการสลับลำดับข้อมูล แบบสุ่มที่กำหนดขนาดได้

เราใช้อัลกอริทึม MPEG-4 EZW แยกข้อมูลภาพออกมาเป็นแพ็กเก็ต LFS จำนวน 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต ที่มีขนาดแตกต่างกัน โดยแพ็กเก็ตที่ยาวที่สุดมีขนาด 878 สัญลักษณ์และสั้นที่สุดมีขนาด 237 สัญลักษณ์ แพ็กเก็ตเหล่านี้จะถูกส่งไปเข้ารหัสช่องสัญญาณ แล้วจะถูกนำไปสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยแทนที่จะสลับลำดับทั้งแพ็กเก็ต เราจะกำหนดจำนวนของสัญลักษณ์ที่จะสลับลำดับแต่ละครั้ง ผลที่ได้แสดงให้เห็นว่าค่า PSNR เฉลี่ยดีขึ้นประมาณ 14.5 dB เมื่อเทียบกับผลที่ได้โดยไม่ผ่านการสลับลำดับข้อมูลแบบสุ่มที่ กำหนดขนาดได้ สำหรับแพ็กเก็ตที่ยาวที่สุด เวลาที่ใช้ในการสลับลำดับของการสลับลำดับข้อมูล แบบสุ่มที่กำหนดขนาดได้จะเร็วกว่าการสลับลำดับข้อมูลแบบสุ่มเกินกว่า 3 เท่า ดังนั้นการสลับ ลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้นอกจากลดผลกระทบจากช่องสัญญาณแบบ Rician บล็อก เฟดดิ้งแล้วยังลดความซับซ้อนของระบบลงอีกด้วย

Nukoon Phimsen 2011: Design of Reconfigurable Random Interleavers with MAP Decoding for MPEG-4 Image Transmission System over Rician Block-Fading Channels. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Department of Electrical Engineering. Thesis Advisor: Assistant Professor Srijidtra Charoenlarnopparut, Ph.D. 86 pages.

Wireless communication is a technology in which, have been developed continuously. In particular, we are interested in the technology on the multimedia application in various aspect. This research is a part of MPEG-4 image transmission system over an indoor wireless channel, which is modeled as a Rician block fading channel. This work emphasizes on alleviating symbol burst error, as a result of the impact from fading. The problem was previously solved with random interleavers. However we found out that the system shows high complexity, and slowly. Therefore, we aim to improve random interleavers by designing a modified random interleaver, called as reconfigurable random interleaver.

We apply MPEG-4 EZW algorithm to decompose a still image into a LFS packet and the variable length 101 HFS packets. The longest packet length is 878 symbols, and the shortest packet one is 237 symbols. These packets are sent to the channel coding process and then are passed to the reconfigurable random interleaver. Instead of interleaving the whole packet, we assign a specific number of symbols to be interleaved. The results show the improvement of average PSNR around 14.5 dB compared to the results without reconfigurable random interleavers. For the longest packet, the interleaving time of the reconfigurable random interleaver is faster than that of the random interleaver more than 3 times. Consequently, the reconfiguration random interleavers not only reduce the effect of Rician block fading but also the system complexity.

Student's signature

Thesis Advisor's signature

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ศรีจิตรา เจริญลาภนพรัตน์ ประธาน
กรรมการที่ปรึกษา ที่ได้ช่วยเหลือวางแผนงานวิจัย ให้คำปรึกษาแนะนำและตรวจสอบแก้ไข
ข้อบกพร่อง และขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พูนลาภ ลามศรีจันทร์ ที่ช่วยแนะนำ
ให้สามารถทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี และกราบขอบพระคุณคณาจารย์ประจำภาควิชา
วิศวกรรมไฟฟ้าทุกท่านที่ สอนวิชาความรู้

ขอขอบพระคุณมหาวิทยาลัยธุรกิจบัณฑิตที่กรุณาให้ทุนการศึกษา และกราบขอบพระคุณ
คุณพ่อ คุณแม่ที่สนับสนุนให้ศึกษาต่อ

หากวิทยานิพนธ์ฉบับนี้มีข้อบกพร่องประการใด ข้าพเจ้ายินดีรับข้อเสนอแนะ และขออภัย
มา ณ ที่นี้ด้วย

นุฎล พิมเสน
พฤษภาคม 2554

สารบัญ

หน้า

สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(6)
คำอธิบายสัญลักษณ์และอักษรย่อ	(9)
คำนำ	1
วัตถุประสงค์	2
การตรวจเอกสาร	3
อุปกรณ์และวิธีการ	27
อุปกรณ์	27
วิธีการ	27
ผลและวิจารณ์	37
ผล	37
วิจารณ์	74
สรุปและข้อเสนอแนะ	76
สรุป	76
ข้อเสนอแนะ	77
เอกสารและสิ่งอ้างอิง	78
ภาคผนวก	80
ประวัติการศึกษาและการทำงาน	86

สารบัญตาราง

ตารางที่	หน้า
1 ตัวเข้ารหัสคอนโวลูชันนอลแบบฐานสองที่มีอัตรา $1/2$ ที่ดีที่สุดและ d_{free}	16
2 ผลจากการทดสอบโดยไม่ผ่านการสลับลำดับข้อมูล โดยการใช้การถอดรหัสแบบ MAP และ ML	39
3 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	40
4 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	41
5 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	42
6 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	43
7 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	44
8 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	45
9 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	46

สารบัญตาราง (ต่อ)

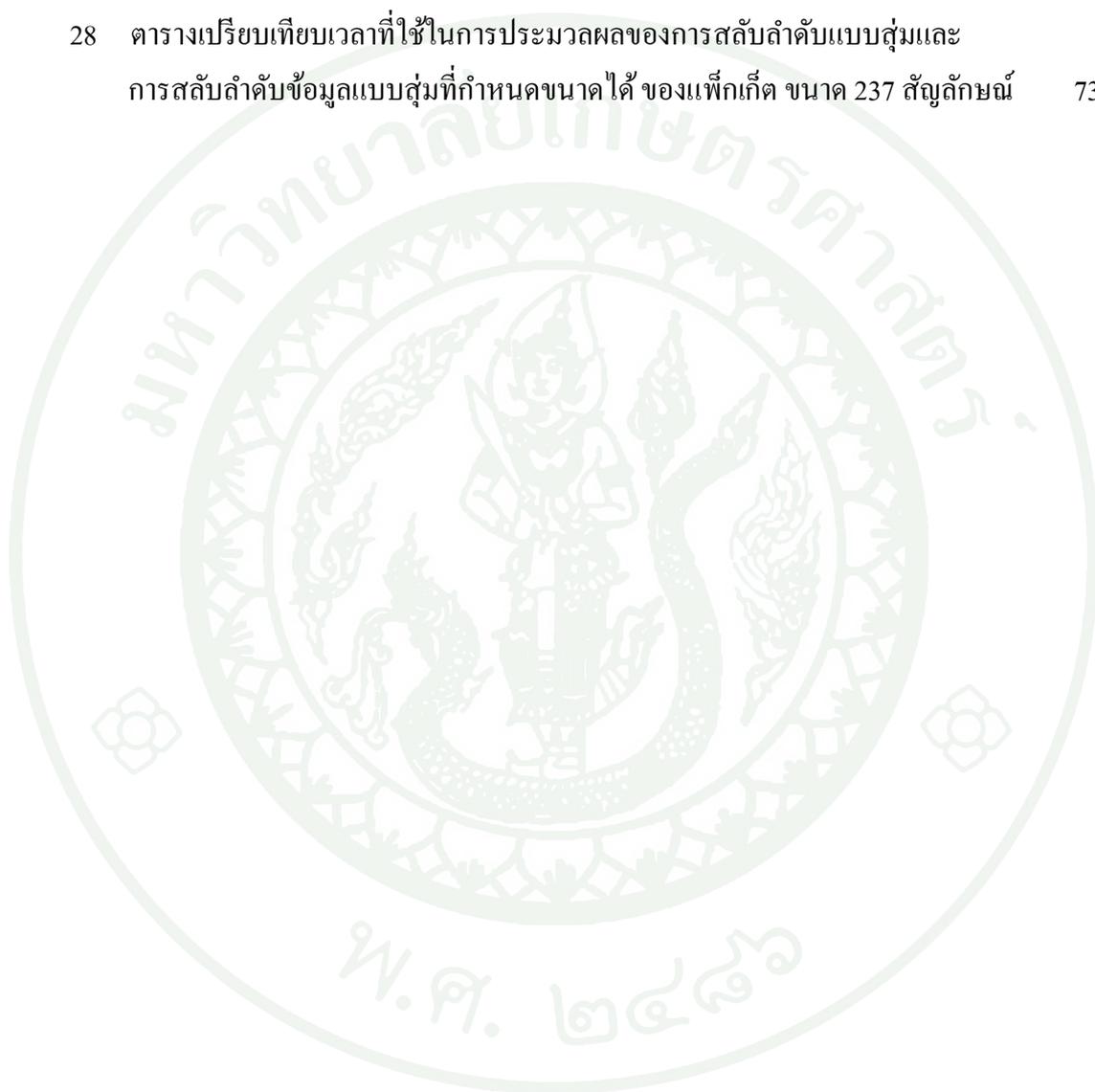
ตารางที่	หน้า
10 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	47
11 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	48
12 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	49
13 ผลจากการทดสอบโดยไม่ผ่านการสลับลำดับข้อมูลกับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP	56
14 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	57
15 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	58
16 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	60
17 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	61
18 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	62

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
19	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 128$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	63
20	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 143$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	64
21	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 143$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	66
22	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม(ทั้งแพ็กเก็ต) กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS	67
23	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม(ทั้งแพ็กเก็ต) กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS	68
24	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 20 dB	70
25	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 21 dB	70
26	ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 22 dB	71
27	ตารางเปรียบเทียบเวลาที่ใช้ในการประมวลผลของการสลับลำดับแบบสุ่มและการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ของแพ็กเก็ต ขนาด 878 สัญลักษณ์	73

สารบัญตาราง (ต่อ)

ตารางที่	หน้า	
28	ตารางเปรียบเทียบเวลาที่ใช้ในการประมวลผลของการสลับลำดับแบบสุ่มและ การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ของแพ็กเก็ต ขนาด 237 สัญลักษณ์	73



สารบัญภาพ

ภาพที่		หน้า
1	โครงสร้างระบบส่งข้อมูลรูปภาพ MPEG-4 ไร้สาย	3
2	กระบวนการแยกข้อมูลภาพโดยเทคนิคแยกกันกระทำ	6
3	แผนผังขั้นตอนการเข้ารหัสของ MPEG-4 EZW	7
4	ตำแหน่งของสัมประสิทธิ์ของการควอนไทซ์ X และสัมประสิทธิ์ข้างเคียง	8
5	แผนผังขั้นตอนการถอดรหัสของ MPEG-4 EZW	9
6	โครงสร้างพื้นฐานของแพ็กเก็ต HFS	10
7	แผนผังของการเข้ารหัสคอนโวลูชันนอล	11
8	แผนผังเทลลิสของตัวอย่างการเข้ารหัสคอนโวลูชันนอล	13
9	ตัวอย่างการหาเส้นทางที่สั้นที่สุดหลังรอบที่ 5 ($5T$)	15
10	แผนผังการทำงานของ TCM	19
11	แบบจำลองของ Gilbert-Elliott สำหรับช่องสัญญาณ Burst-Noise	20
12	เมทริกซ์ P และ P^T สำหรับ ลำดับสุ่ม 1,8,5,2,9,6,3,0,7,4	23
13	ตัวอย่างเวกเตอร์ d_{INT} ที่ได้จากการสลับลำดับตาม ลำดับสุ่ม 1,8,5,2,9,6,3,0,7,4	23
14	ตัวอย่างเวกเตอร์ d ที่ได้จากการสลับลำดับข้อมูลกลับ	24
15	เมทริกซ์ P และ P^T สำหรับ $N_R = 3$ และมีลำดับสุ่มเป็น 1,2,0	25
16	ตัวอย่างการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่มี $N_R = 3$	25
17	ตัวอย่างการสลับลำดับข้อมูลกลับ	26
18	ตัวอย่างการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้	30
19	ผังงานการทำงานของ การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้	31
20	ตัวอย่างการสลับลำดับข้อมูลกลับ	32
21	ผังงานการทำงานของ การสลับลำดับข้อมูลกลับแบบสุ่มที่กำหนดขนาดได้	33
22	รูปภาพ Lena ต้นแบบ	35
23	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB	50
24	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB	50
25	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB	51
26	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB	52

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
27	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB	52
28	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB	53
29	รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP	
	ก) $\bar{\gamma}_b = 6.25$ dB โดยไม่ผ่านการสลับลำดับข้อมูล ได้ PSNR = 13.172 dB, WER = 0.00967	
	ข) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 27.666 dB, WER = 0.00067	53
30	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย	54
31	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย	55
32	กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย	55
33	รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP	
	ก) $\bar{\gamma}_b = 6.25$ dB และ $N_S = 8$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 22.885 dB, WER = 0.0026	
	ข) $\bar{\gamma}_b = 6.25$ dB และ $N_S = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 26.662 dB, WER = 0.00099	69

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
34	
รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP	
ก) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 23$ ได้ PSNR = 20.054 dB, WER = 0.0024	
ข) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 26$ ได้ PSNR = 21.551 dB, WER = 0.0022	
ค) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 31$ ได้ PSNR = 22.568 dB, WER = 0.00137	72
35	
รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP	
ก) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มทั้งเพ็ทเกิดได้ PSNR = 30.450 dB, WER = 0.00012	
ข) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 26.662 dB, WER = 0.00099	74
ภาพผนวกที่	
1 หน้าต่างโปรแกรม Netbeans IDE จากการเปิดโปรแกรมครั้งแรก	81
2 หน้าต่างการสร้างโปรเจกใหม่	82
3 หน้าต่างให้ตั้งชื่อโปรเจกและกำหนดตำแหน่งเก็บโปรแกรที่สร้างขึ้น	83
4 หน้าต่างแสดงคลาสที่สร้างขึ้นมาใหม่	84
5 แสดงคำสั่งภาษา Java ที่พิมพ์เพิ่มลงไป	84
6 แสดงผลลัพธ์จากการ Build โปรเจก	85
7 แสดงผลลัพธ์จากการรันโปรแกรม	85

คำอธิบายสัญลักษณ์และอักษรย่อ

EZW	=	Embedded Zerotree Wavelet
DWT	=	Discrete Wavelet Transform
ML	=	Maximum Likelihood
MAP	=	Maximum <i>a Posteriori</i>
HFS	=	Higher Frequency Subband
LFS	=	Lowest Frequency Subband
DPCM	=	Differential Pulse Code Modulation
ZTR	=	Zerotree Root
IZ	=	Isolated Zero
VZTR	=	Valued Zerotree Root
VAL	=	Value
CRC	=	Cyclic Redundancy Check
SOT	=	Starting Spatial Orientation Tree
MLSE	=	Maximum Likelihood Sequence Estimator
CPM	=	Continuous Phase Modulation
TCM	=	Trellis-coded Modulation
AWGN	=	Additive White Gaussian Noise
PSNR	=	Peak Signal to Noise Ratio
WER	=	Word Error Rate

การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับการถอดรหัสแบบแมป
สำหรับระบบสื่อสารข้อมูลภาพ MPEG-4 บนช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

**Design of Reconfigurable Random Interleavers with MAP Decoding for MPEG-4
Image Transmission System over Rician Block-Fading Channels**

คำนำ

ทุกวันนี้การเปลี่ยนแปลงเทคโนโลยีทางด้านโทรคมนาคมเกิดขึ้นอย่างมากมาย โดยเฉพาะอย่างยิ่งเทคโนโลยีทางการสื่อสารไร้สาย การวิจัยทางการรับส่งภาพผ่านช่องทางการสื่อสารไร้สาย ยังมีเรื่องที่ทำนายต่อนักวิจัยอยู่หลายประเด็น เป็นที่ทราบกันว่าช่องทางการสื่อสารไร้สายเป็นช่องทางที่เกิดสัญญาณรบกวนได้ง่าย ดังนั้นการรับส่งภาพผ่านช่องทางการสื่อสารไร้สายจึงจำเป็นต้องใช้เทคนิคขั้นสูงในขั้นตอนต่างๆ อันได้แก่การเข้ารหัสภาพที่สามารถปรับเปลี่ยนไปตามความผิดพลาดที่เกิดขึ้น

งานวิจัยนี้ได้นำเอาระบบของ Mahapakulchai (2007) มาปรับปรุงเพิ่มเติมโดยมีการปรับปรุงการสลับลำดับแบบสุ่มให้มีความซับซ้อนน้อยลง โดยแทรกการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้เข้าไปในระบบตามภาพที่ 1 เพื่อทดสอบประสิทธิภาพที่เกิดขึ้นหลังจากส่งข้อมูลภาพผ่านช่องสัญญาณไร้สายแบบ Rician บล็อกเฟดดิ้ง

วัตถุประสงค์

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาการสลับลำดับข้อมูลก่อนส่งออกไปยังช่องสัญญาณไร้สายในอาคาร โดยลดความซับซ้อนของการสลับลำดับข้อมูลแบบสุ่มให้น้อยลง โดยมีการปรับจำนวนของสัญลักษณ์ที่จะสลับลำดับ ให้เป็นตามที่ต้องการ จากนั้นสามารถที่จะปรับขนาดให้สอดคล้องกับคุณภาพของช่องสัญญาณไร้สายได้

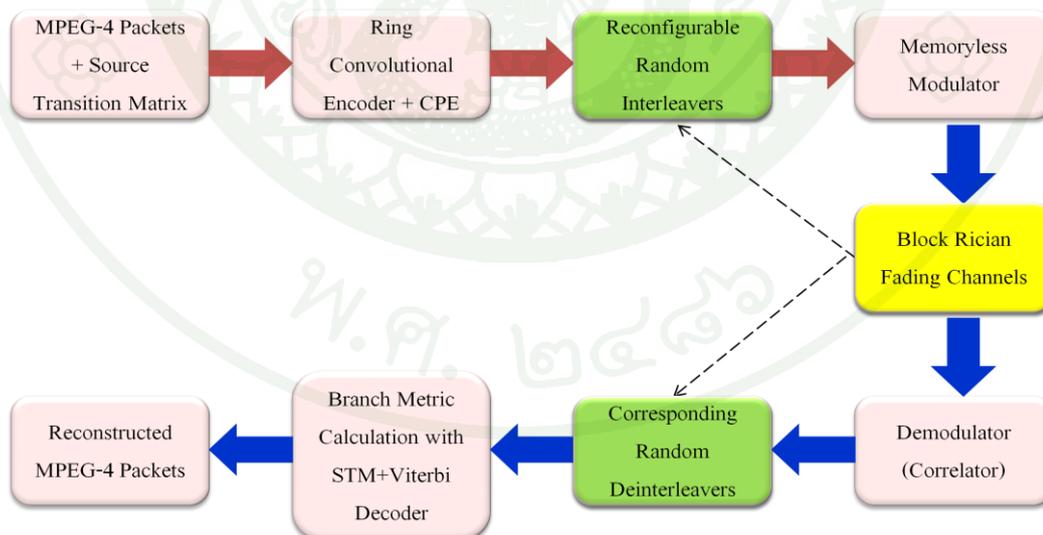
หลังจากลดความซับซ้อนของกระบวนการสลับลำดับข้อมูลแบบสุ่ม จะส่งผลให้เวลาที่ใช้ในกระบวนการทั้งหมดลดลงด้วยเนื่องจากความยาวของข้อมูลที่นำมาสลับลำดับข้อมูลนั้นมีผลอย่างมากต่อเวลาที่ใช้ในการสลับลำดับข้อมูล ดังนั้นในการปรับขนาดของข้อมูลที่จะสลับลำดับข้อมูลได้ เราสามารถควบคุมเวลาที่ใช้ในการสลับลำดับข้อมูลได้ตามมา

หนึ่งรูปภาพที่สร้างกลับมาหลังจากผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้แล้ว ยังคงมีประสิทธิภาพที่ดีอยู่ โดยดูจากค่า PSNR และ WER เป็นหลัก

การตรวจเอกสาร

1. บทนำ

ระบบสื่อสารข้อมูลภาพต้นแบบประกอบด้วย 3 ส่วนหลักๆ ได้แก่ ส่วนแรกจากการจำลองการเข้าและถอดรหัสข้อมูลภาพ ส่วนที่สองจำลองการเข้าและถอดรหัสช่องสัญญาณ และส่วนสุดท้าย การจำลองช่องสัญญาณไร้สาย วิธีการเข้าและถอดรหัสภาพ EZW เป็นวิธีการที่ให้ประสิทธิภาพสูง เริ่มต้นสร้างโดย Shapiro (1993) เป็นการบีบอัดข้อมูลแบบ MPEG-4 การเข้ารหัสช่องสัญญาณ ได้นำงานวิจัยของ Ungerbeck (1982) โดยศึกษาและออกแบบรวมกับการเข้ารหัสคอนโวลูชันกับการมอดูเลต CPFSK จากนั้นเริ่มออกแบบระบบส่งภาพโดย Mahapakulchai and Van Dyck (2000) เป็นการนำการถอดรหัสแบบ MAP เข้ามาใช้กับระบบส่งข้อมูลภาพ พบว่าประสิทธิภาพโดยเฉลี่ยดีขึ้น จากนั้น Mahapakulchai and Van Dyck (2004) มีการเปลี่ยนแปลงรูปภาพที่ใช้ในการทดสอบ ให้มีความหลากหลายมากขึ้น นอกจากนี้ Mahapakulchai (2007) ออกแบบการเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชันนอล ใช้ร่วมกับการถอดรหัสแบบ MAP เพื่อการถอดรหัสที่ดีที่สุด



ภาพที่ 1 โครงสร้างระบบส่งข้อมูลรูปภาพ MPEG-4 ไร้สาย

ที่มา: Mahapakulchai (2007)

ระบบการส่งรูปภาพบนช่องสัญญาณไร้สาย จากภาพที่ 1 แפקเก็ตของ MPEG-4 จะประกอบไปด้วยย่านความถี่ต่ำ หรือ LFS จำนวนหนึ่งแพ็กเก็ตและย่านความถี่สูง หรือ HFS อีกหลายแพ็กเก็ต โดยวัตถุประสงค์ของการแยกออกมาเป็นหลายแพ็กเก็ตเพื่อเพิ่มความเชื่อถือได้ของการส่งข้อมูล โดยที่ แพ็กเก็ตแบบ HFS จะมีขนาดที่ไม่เท่ากันและมีความยาวโดยเฉลี่ยประมาณ 325 สัญลักษณ์ เราได้รับลำดับของสัญลักษณ์ Zerotree จากกระบวนการเข้ารหัสแหล่งข้อมูล MPEG-4 ซึ่งสามารถโมเดลเป็นแหล่งข้อมูลแบบ Markov ลำดับแรกได้ ซึ่งจะนำมาสร้างเป็นเมทริกซ์ของความน่าจะเป็นในการเปลี่ยนแปลงของแหล่งข้อมูลหรือ STM เพื่อที่จะนำมาใช้กับการถอดรหัสแบบ MAP

จากการออกแบบระบบการส่งรูปภาพ MPEG-4 เราได้ใช้ระบบการเข้ารหัสช่องสัญญาณ ร่วมกับการมอดูเลต คือการเข้ารหัสแบบ โพลีโนเมียลคอนโวลูชันนอล อัตราเท่ากับ $1/2$ บนริงของ Z_4 หรือ RCE และการมอดูเลตแบบ CPFSK ที่มีค่า h เท่ากับ $1/4$ ตามลำดับ โดยที่ CPFSK ถูกแบ่งแยกออกไปเป็นการเข้ารหัสเฟสแบบต่อเนื่อง หรือ CPE กับ มอดูเลเตอร์แบบไม่มีความจำ หรือ MM ดังนั้นการเข้ารหัสโดยรวม คือการรวมกันระหว่าง RCE และ CPE เมื่อสัญลักษณ์ที่ถูกเข้ารหัสแล้วจากการเข้ารหัสโดยรวม จะถูกส่งไปยังตัวสลับลำดับข้อมูลแบบสุ่มที่สามารถกำหนดขนาดการสลับลข้อมูลได้ที่มีขนาดแตกต่างกันไป

สัญลักษณ์ที่ผ่านการสลับลำดับมาแล้วจะถูกแมปไปเป็นส่วนประกอบ In phase และ Quadrature phase ของความถี่ฐานของสัญญาณ CPFSK ด้วย MM จากนั้นส่วนประกอบของสัญญาณทั้งสองส่วนจะถูกส่งไปยังช่องสัญญาณ แล้วเกิดความเสียหายจากสัญญาณรบกวนและผิดเพี้ยนจากช่องสัญญาณไร้สายแบบ Rician บล็อกเฟดดิ้ง ในส่วนของฝั่งรับ เราใช้การถอดรหัสแบบ Maximum a Posteriori หรือ MAP ซึ่งประกอบด้วยกระบวนการการตีมอดูเลต การสลับลำดับกลับของสัญลักษณ์ การคำนวณเส้นทางบนทrellis และการถอดรหัสวิเทอร์บี สตริมของ 4-ary ที่ถูกถอดรหัสมาแล้ว ถูกถอดออกมาจากแพ็กเก็ต แล้วส่งต่อไปยังตัวถอดรหัสข้อมูลรูปภาพ MPEG-4 ได้เป็นรูปภาพที่ถูกสร้างกลับมา

การปรับปรุงระบบการส่งภาพ โดยการนำเทคนิคการสลับลำดับข้อมูลมาใช้ เริ่มจาก Mahapakulchai and Thongnumpen (2007) ได้นำกระบวนการสลับลำดับข้อมูลแบบบล็อกมาใช้กับระบบ แต่เนื่องจากความยาวของแพ็กเก็ตไม่คงที่ จึงทำให้วิธีการนี้ มีความยุ่งยากในการคำนวณขนาดของเมทริกซ์ที่นำมาใช้ในการสลับลำดับข้อมูล จากนั้น Mahapakulchai and Sapisirisopon

(2008a) ได้ทำการออกแบบการสลับลำดับข้อมูลแบบสุ่ม ซึ่งได้ประสิทธิภาพที่ใกล้เคียง แต่ลดปัญหาความยุ่งยากในการคำนวณขนาดของเมทริกซ์ที่นำมาใช้ในการสลับลำดับ นอกจากนี้ Mahapakulchai and Sapasirisopon (2008b) นำการถอดรหัสแบบ MAP มาทดสอบกับการสลับลำดับข้อมูลแบบสุ่ม ทำให้ได้ประสิทธิภาพที่ดีขึ้นเล็กน้อย

แต่อย่างไรก็ตาม ทั้ง Mahapakulchai and Sapasirisopon (2008a) และ Mahapakulchai and Sapasirisopon (2008b) ยังมีการสลับลำดับข้อมูลที่เดียวทั้งแพ็กเก็ต ซึ่งทำให้เกิดความซับซ้อนในการทำงานสูงเนื่องจากต้องมีการสร้าง ลำดับสุ่มขึ้นมา ตามขนาดของแพ็กเก็ต โดยที่ขนาดของแพ็กเก็ตมีขนาดที่แตกต่างกันไป จึงทำให้กระบวนการสลับลำดับข้อมูลมีความซับซ้อนสูง และทำให้ใช้ทรัพยากรของระบบสูงตามไปด้วย ซึ่งทำให้ Mahapakulchai and Nukoon (2010) ต้องการจะลดความซับซ้อนที่เกิดจากกระบวนการสลับลำดับข้อมูล จึงได้ปรับกระบวนการสลับลำดับข้อมูลแบบสุ่มเดิม ให้สามารถกำหนดช่วงของข้อมูลที่จะสลับลำดับข้อมูลได้ แทนการสลับลำดับข้อมูลทั้งแพ็กเก็ต ทำให้สามารถลดความซับซ้อนของการสลับลำดับข้อมูลแบบสุ่มลงได้มาก ซึ่งดูได้จากเวลาที่ใช้ในการประมวลผลของการสลับลำดับข้อมูลที่ลดลงอย่างเห็นได้ชัด และยังคงได้คุณภาพใกล้เคียงกับการสลับลำดับข้อมูลทั้งแพ็กเก็ต

2. ระบบการส่งรูปภาพบนช่องสัญญาณไร้สาย

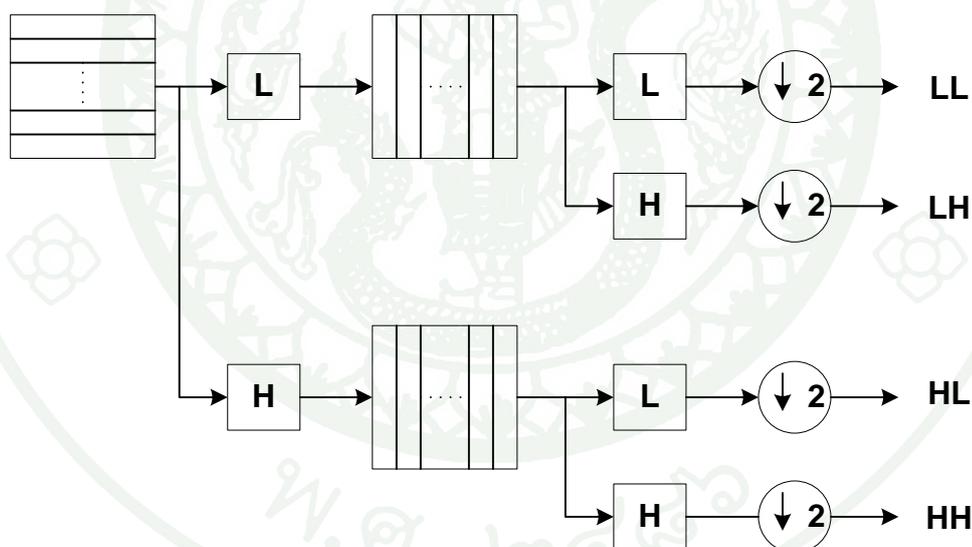
2.1 การบีบอัดและการเข้ารหัสถอดรหัสรูปภาพ

การเข้ารหัสข้อมูลภาพแบบ MPEG-4 เป็นการบีบอัดข้อมูลภาพที่ได้รับความนิยมอย่างแพร่หลาย เนื่องจากมีประสิทธิภาพในการบีบอัดสูง โดยอัลกอริทึมที่เราเลือกใช้คือ EZW โดยจะทำการแยกข้อมูลภาพที่รับเข้ามาออกเป็นสองประเภทคือ ข้อมูล LFS ซึ่งจะเป็นข้อมูลโครงสร้างหลักของรูปภาพ และ กลุ่มของข้อมูล HFS ซึ่งเป็นข้อมูลของรายละเอียดปลีกย่อยของภาพ โดยข้อมูลทั้ง 2 ประเภทนี้จะถูกนำมาเข้ารหัสแยกกัน ซึ่งทำให้เราสามารถป้องกันการบิดเบือนของข้อมูลที่แตกต่างกันได้ง่าย

อัลกอริทึม EZW เป็นการบีบอัดที่ แบ่งข้อมูลภาพออกเป็นหลายๆส่วน โดยจะแทนค่าด้วยสัมประสิทธิ์เวฟเลต ซึ่งมีคุณสมบัติที่ดีหลายอย่าง จึงทำให้ได้รับความนิยมอย่างสูงในงานด้านภาพนิ่งและภาพเคลื่อนไหว สำหรับภาพดิจิทัล เราได้ใช้การแปลงค่าเป็นสัมประสิทธิ์เวฟเลต ที่

เรียกว่า Discrete Wavelet Transform หรือ DWT เราสามารถอธิบาย DWT ด้วย ระบบการเข้ารหัส ความถี่ย่อยแบบลำดับชั้น ชนิดหนึ่งได้ โดยระบบการเข้ารหัสนี้จะทำการแยกสัญญาณออกมา โดยการใช้กลุ่มของตัวกรอง (Filter) ที่เรียกว่า “filter bank” ไปเป็นย่านความถี่ย่อยด้วยเทคนิคของการกรองซ้ำ และ Downsampling โดยที่กระบวนการ Downsampling ใช้แทนคำว่า “สเกล” ในการแยก เวฟเลต โดยที่สเกลใหญ่หมายถึงหยาบ และสเกลเล็กหมายถึงละเอียด

เนื่องจากรูปภาพเป็นสัญญาณ 2 มิติ เราสามารถทำกระบวนการแยกข้อมูลภาพโดย แยกกันกระทำระหว่างแถวและหลักของรูปภาพ โดยใช้ตัวกรองความถี่ต่ำ (Low-pass Filter) และ ตัวกรองความถี่สูง (High-pass Filter) ตามภาพที่ 2 แสดงการสร้างย่านความถี่ย่อย 4 ย่านความถี่ (LL_1, LH_1, HL_1, HH_1) หลังจากการแยกข้อมูลภาพขั้นแรก จากชื่อของย่านความถี่ ตัวอักษรตัวแรก บ่งบอกประเภทของตัวกรองที่ใช้สำหรับแถว ซึ่งเป็นการอ่านข้อมูลภาพตามแนวนอน และตัวอักษร ตัวที่สองบ่งบอกประเภทของตัวกรองที่ใช้สำหรับหลัก ซึ่งเป็นการอ่านข้อมูลภาพตามแนวตั้ง

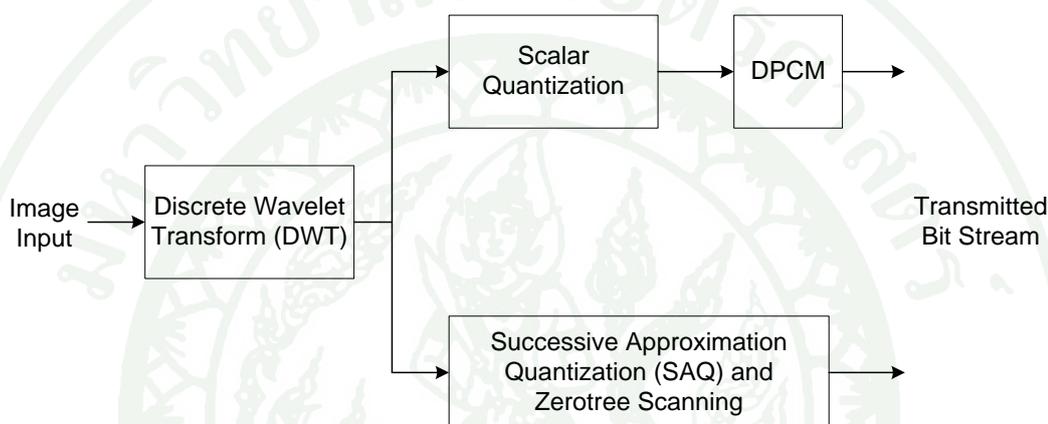


ภาพที่ 2 กระบวนการแยกข้อมูลภาพโดยเทคนิคแยกกันกระทำ

กระบวนการเข้ารหัสของ MPEG-4 EZW เริ่มจากการแปลงรูปภาพด้วย DWT โดยที่ งานวิจัยนี้มีการแยกรูปภาพด้วย DWT แบบ 5 ระดับ โดยการใช้ตัวกรอง Daubechies (9,15) tap biorthogonal ในการแยกสัมประสิทธิ์ของ LFS และ HFS ออกมา ซึ่งถูกนำไปเข้ารหัสแยกกัน

2.1.1 การเข้ารหัสข้อมูลภาพ

การเข้ารหัสข้อมูลภาพ จะทำการเข้ารหัสสัมประสิทธิ์ LFS ที่ถูกแปลงมา โดยใช้ Scalar Quantization แล้วตามด้วย Differential Pulse Code Modulation (DPCM) ส่วนการเข้ารหัสของสัมประสิทธิ์ HFS ถูกกระทำโดยการใช้กระบวนการ Successive Approximation Quantization ร่วมกับ Zerotree Scanning โดยแสดงขั้นตอนของการเข้ารหัส สำหรับ LFS และ HFS จากภาพที่ 3



ภาพที่ 3 แผนผังขั้นตอนการเข้ารหัสของ MPEG-4 EZW

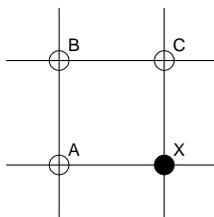
ในส่วนการเข้ารหัสของสัมประสิทธิ์ LFS เริ่มจากสัมประสิทธิ์ของ LFS จะถูกควอนไทซ์ ด้วย Scalar Quantization แล้วตามด้วยกระบวนการ DPCM เพื่อทำนายค่าสัมประสิทธิ์ของการควอนไทซ์ X จากค่าของการควอนไทซ์ใกล้เคียง 3 ค่า จากภาพที่ 4 คือ A, B, C จากนั้น ค่าที่ทำนายได้ \hat{X} จะถูกลบออกจากสัมประสิทธิ์ X ได้ค่าสัมประสิทธิ์ Residual ซึ่งถูกใช้ส่งออกไป โดยสามารถอธิบายอัลกอริทึม DPCM ได้ดังนี้

ถ้า $(|A - B| < |A - C|)$ เป็นจริง

ถ้าเช่นนั้น $\hat{X} = C$

ถ้าไม่เช่นนั้น $\hat{X} = A$.

ค่าสัมประสิทธิ์ Residual = $X - \hat{X}$



ภาพที่ 4 ตำแหน่งของสัมประสิทธิ์ของการควอนไทซ์ X และสัมประสิทธิ์ข้างเคียง

สำหรับมาตรฐานของ MPEG-4 ค่าของสัมประสิทธิ์ Residual ทั้งหมดจะถูกออฟเซตโดยค่าที่น้อยที่สุดของพวกมัน ซึ่งจะทำให้เป็นค่าบวกได้เท่านั้น จากนั้นพวกมันจะถูกเข้ารหัสด้วยการเข้ารหัสแบบ Adaptive Arithmetic แต่สำหรับงานนี้เราทำการเลื่อนสัมประสิทธิ์ Residual ด้วยจำนวนขั้นของการควอนไทซ์ ซึ่งการเปลี่ยนแปลงนี้ทำให้เราไม่ต้องส่งค่าออฟเซตไปด้วย นอกจากนี้เราไม่ได้ทำการเข้ารหัสด้วย Arithmetic สำหรับสัมประสิทธิ์ LFS สำหรับรูปภาพขนาด 512×512 และการแยกข้อมูล 5 ระดับ เราได้ขนาดของ LFS ของค่าสัมประสิทธิ์ 16×16 ทำให้ย่านความถี่ย่อยนี้ต้องการขนาดของคำรหัสที่มีดัชนีเป็น 7 ที่ตายตัวแค่ 1,792 บิตเท่านั้น

สำหรับการเข้ารหัสของสัมประสิทธิ์ HFS แบ่งเป็น 2 ประเภทหลักคือ Single Quantization และ Multi-Quantization สำหรับงานนี้ใช้แบบ Single Quantization ที่เป็นการควอนไทซ์ทุกสัมประสิทธิ์เวฟเลตเพียงครั้งเดียวด้วย Multi-level Quantizer โดยลำดับของบิตที่ได้มาจะไม่มีคุณสมบัติ Embedded Code เนื่องจากไม่ได้ส่งบิตที่เป็นบิตนัยความสำคัญสูงสุดของทุกสัมประสิทธิ์เวฟเลตไปก่อน โครงสร้างของทรีเริ่มต้นด้วยแต่ละค่าสัมประสิทธิ์ Root ในย่านความถี่ย่อย 3 ย่านความถี่ที่สเกลหยาบสุด หรือ Coarsest Scale คือ HL_3 , LH_3 และ HH_3 ทุก SOT เหล่านี้จะถูกสแกนตามลำดับ และสัมประสิทธิ์จะถูกแบ่งออกเป็น 4 กลุ่ม คือ Zerotree Root (ZTR), Isolated Zero (IZ), Valued Zerotree Root (VZTR) และ Value (VAL) โดยถ้าค่าสัมประสิทธิ์มีค่าน้อยกว่าค่าที่ขีดแบ่งไว้ สัมประสิทธิ์นั้นจะอยู่ในกลุ่ม ZTR ถ้าสัมประสิทธิ์ที่ต่อจากมันทั้งหมดมีค่าน้อยกว่าค่าที่ขีดแบ่งไว้ด้วย หรือสัมประสิทธิ์นั้นจะอยู่ในกลุ่ม IZ ถ้ามีสัมประสิทธิ์ที่ต่อจากมันอย่างน้อย 1 สัมประสิทธิ์ มีค่ามากกว่าค่าที่ขีดแบ่งไว้ นอกเหนือจากนี้สัมประสิทธิ์จะจัดอยู่ในกลุ่ม VZTR หรือ VAL ซึ่ง VZTR คือค่าสัมประสิทธิ์ที่มีค่ามากกว่าค่าที่ขีดแบ่งไว้แต่ค่าสัมประสิทธิ์ที่ต่อจากมันทั้งหมดต้องมีค่าน้อยกว่าค่าที่ขีดแบ่งไว้ด้วย แต่ถ้ามีสัมประสิทธิ์ที่ต่อจากมันอย่างน้อย 1 สัมประสิทธิ์ มีค่ามากกว่าค่าที่ขีดแบ่งไว้ มันจะจัดอยู่ในกลุ่ม VAL เมื่อสัมประสิทธิ์เป็นกลุ่ม VAL

หรือ IZ จะทำให้ค่าสัมประสิทธิ์ที่สอดคล้องกัน 4 ค่า สร้างทรีใหม่ 4 ทรี และกระบวนการจะทำซ้ำสำหรับแต่ละทรีใหม่จนกระทั่งทรีเต็มสมบูรณ์ แล้วจะเริ่มทำกระบวนการกับทรีถัดไป

ส่วนแบบ Multi-Quantization จะนำค่าสัมประสิทธิ์เวฟเลตทั้งหมดมาเปรียบเทียบกับลำดับของค่าขีดแบ่งถัดไป ในแต่ละครั้ง แต่ละค่าสัมประสิทธิ์เวฟเลตพร้อมด้วยสัมประสิทธิ์ที่ต่อจากมันทั้งหมดจะถูกเปรียบเทียบกับค่าที่ขีดแบ่งไว้แล้วจะถูกแบ่งออกเป็น 4 กลุ่มตามแบบ Single Quantization ที่ได้อธิบายไปแล้วก่อนหน้านี้ และก่อนการสแกนครั้งถัดไปจะมีการเปลี่ยนแปลงค่าขีดแบ่งใหม่ โดยปกติจะลดลงครึ่งหนึ่งของค่าเดิม

2.1.2 การถอดรหัสข้อมูลภาพ

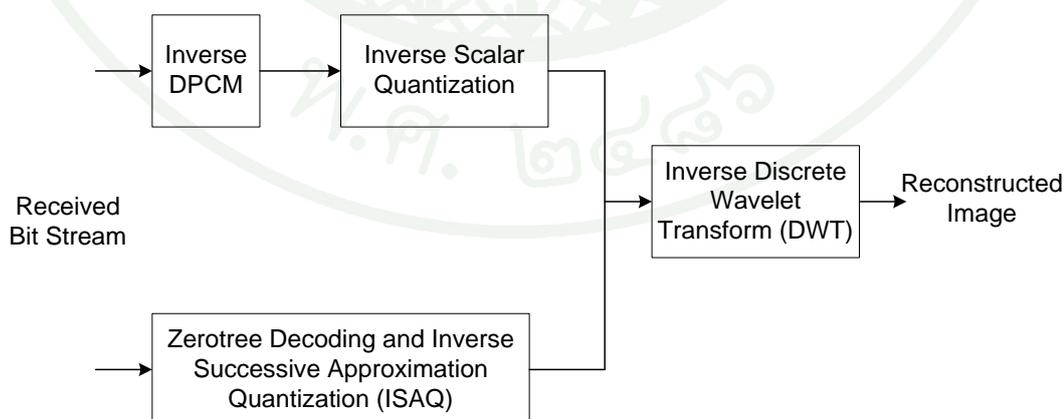
การถอดรหัสสำหรับของสัมประสิทธิ์ LFS ค่าสัมประสิทธิ์ที่ได้รับได้จะถูกเลื่อนกลับด้วยจำนวนขั้นของการควอนไทซ์ จากนั้นส่งผ่านกระบวนการ Inverse DPCM ตามภาพที่ 5 เพื่อหาค่าสัมประสิทธิ์ที่ถูกต้องจากค่าสัมประสิทธิ์ Residual ตามอัลกอริทึมต่อไปนี้

ถ้า $|A - B|$ น้อยกว่า $|A - C|$ เป็นจริง

ถ้าเช่นนั้น $\hat{X} = C$

ถ้าไม่เช่นนั้น $\hat{X} = A$.

$X = \hat{X} +$ ค่าสัมประสิทธิ์ Residual



ภาพที่ 5 แผนผังขั้นตอนการถอดรหัสของ MPEG-4 EZW

จากนั้นจะทำกระบวนการควอนไทซ์กลับ (Inverse Quantization) กับค่าสัมประสิทธิ์ที่ถูกต้องที่ได้มา จึงได้ค่าสัมประสิทธิ์เวฟเลตของ LFS กลับมา

ส่วนการถอดรหัสสำหรับของสัมประสิทธิ์ HFS โดยกำหนดค่าเริ่มต้นให้กับค่าขีดแบ่งแล้ว สัญลักษณ์ซีโรทรี จะถูกแยกออกมาจากลำดับบิตที่รับเข้ามา โดยเริ่มต้น ทุกค่าสัมประสิทธิ์จะเซตให้เป็นศูนย์ ถ้าสัญลักษณ์ที่รับมาเป็น VZTR หรือ VAL จะทำการเพิ่มค่าของสัมประสิทธิ์ที่สอดคล้องกันตามค่าขีดแบ่งนั้น แต่ถ้าสัญลักษณ์ที่รับมาเป็น ZTR หรือ IZ ค่าสัมประสิทธิ์ที่สอดคล้องกันจะเป็นค่าเดิม (สำหรับ VZTR และ ZTR ค่าสัมประสิทธิ์ถัดไปจะไม่ถูกถอดรหัส) ค่าขีดแบ่งจะถูกปรับปรุงตามรูปแบบที่ใช้ตอนเข้ารหัสสำหรับการถอดรหัสครั้งถัดไป ขั้นสุดท้ายใช้ Inverse DWT (IDWT) เพื่อแปลงค่าสัมประสิทธิ์เวฟเลตที่ได้มาเป็นรูปภาพ

Resynchronize Marker	Starting SOT	Number of SOTs	Compressed Image Data Zerotree Symbols and Non-zero Values	CRC
----------------------	--------------	----------------	--	-----

ภาพที่ 6 โครงสร้างพื้นฐานของแพ็คเกจ HFS

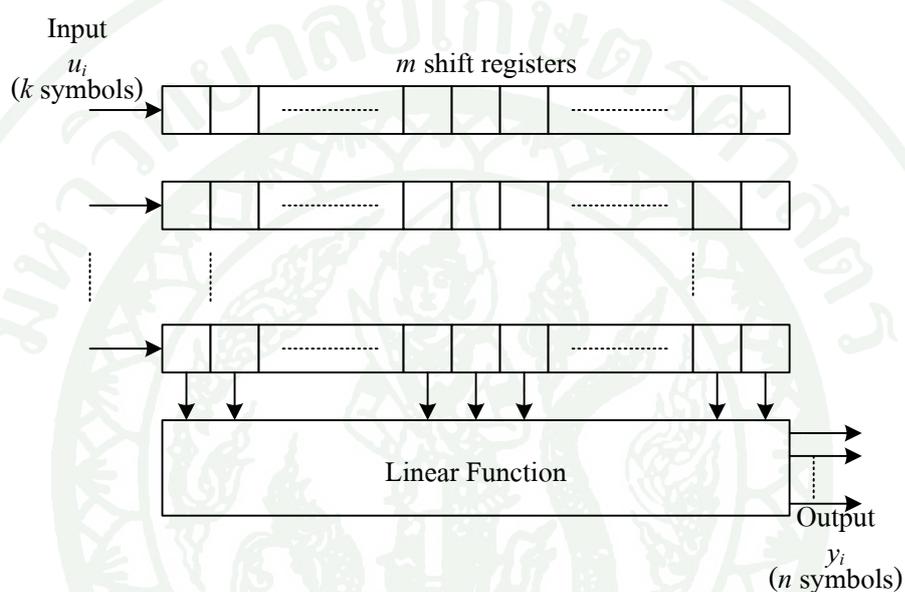
เพื่อให้มีความสามารถในการยืดหยุ่นต่อความผิดพลาด เมื่อใช้การเข้ารหัสที่มีอัตราการบีบอัดสูงสำหรับช่องสัญญาณที่มีสัญญาณรบกวน ในงานนี้จึงได้แบ่งช่วงลำดับบิตที่ได้มาจากการเข้ารหัสออกมาเป็นแพ็คเกจที่มีขนาดแตกต่างกันไปซึ่งนำมาจากลำดับบิตของ HFS โดยมีโครงสร้างพื้นฐานสำหรับการควอนไทซ์แบบ Single Quantization ตามภาพที่ 6 ประกอบด้วย “Resynchronization Marker” จำนวน 20 บิต ตามด้วย “Starting SOT” จำนวน 10 บิต เพื่อบอกตำแหน่งแรกของทรี แล้วตามด้วย “Start of The SOTs” จำนวน 8 บิตเพื่อบอกจำนวนของ SOT แล้วตามด้วย บิตของข้อมูลภาพที่ถูกเข้ารหัสมา และสุดท้ายคือ CRC จำนวน 4 บิต สำหรับตรวจสอบความผิดพลาดของแพ็คเกจเอง

2.2 การเข้ารหัสและถอดรหัสช่องสัญญาณ

การเข้ารหัสช่องสัญญาณ เป็นการเพิ่มบิตพิเศษผนวกเข้าไปกับข้อมูล เพื่อใช้ในการตรวจสอบหรือแก้ไขความผิดพลาดของคำรหัส

2.2.1 การเข้ารหัสช่องสัญญาณ

การเข้ารหัสคอนโวลูชันนอล (n, k, m) บน Finite Field F โดยที่ k คือจำนวนบิตอินพุต n คือจำนวนเอาต์พุต และ m คือจำนวนสถานะของ Shift Register โดยแสดงแผนผังของการเข้ารหัสตามภาพที่ 7



ภาพที่ 7 แผนผังของการเข้ารหัสคอนโวลูชันนอล

เราสามารถแทนรหัสคอนโวลูชันนอลด้วย Generator Matrix โดยมีอัตราการเข้ารหัส $R = k/n$ และความยาวของรหัสคือ $(m + 1)k$ โดยทั่วไปลำดับของอินพุตและเอาต์พุตสามารถกำหนดให้เป็น Semi-Finite และสามารถแสดงไว้ในรูปแบบของ Unit Time Delay (D) ได้เป็นสมการ

$$\mathbf{u}(D) = \mathbf{u}_0 + \mathbf{u}_1 D + \mathbf{u}_2 D^2 + \dots \quad (1)$$

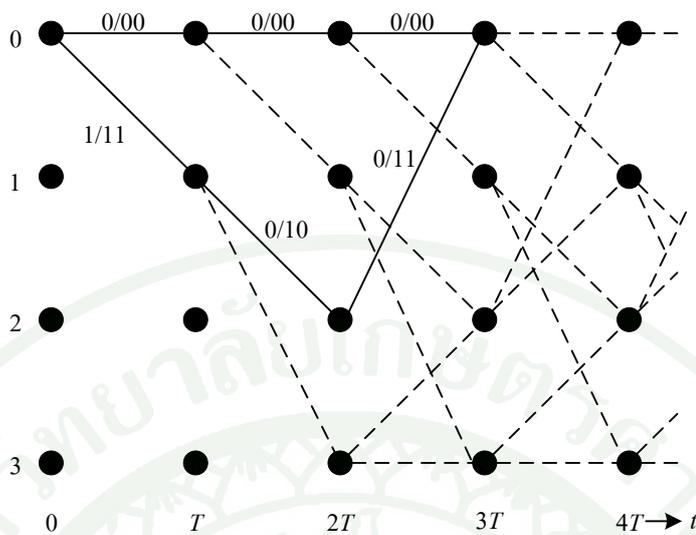
และ

$$\mathbf{y}(D) = \mathbf{y}_0 + \mathbf{y}_1 D + \mathbf{y}_2 D^2 + \dots \quad (2)$$

โดยที่ $\mathbf{u}_i = (u_{0,i}, u_{1,i}, \dots, u_{k-1,i})$ และ $\mathbf{y}_i = (y_{0,i}, y_{1,i}, \dots, y_{n-1,i})$ คืออินพุตและเอาต์พุตตามลำดับ Generator Matrix $G(D)$ ของรหัสคอนโวลูชันนอลถูกนิยามเป็น rank- k , และเป็นเมทริกซ์พหุนามขนาด $k \times n$ หรือ ฟังก์ชัน Rational ในเทอมของ D ดังนั้นความสัมพันธ์ของอินพุตและเอาต์พุต คือ $\mathbf{y}(D) = \mathbf{u}(D)G(D)$ ลำดับของเอาต์พุตที่ได้เป็นกลุ่มของคำรหัส (Codeword) ที่ถูกสร้างมาจาก $G(D)$ ซึ่งมีค่าสอดคล้องกับลำดับของอินพุต

ระยะทางของรหัสคอนโวลูชันนอลถูกนิยามให้เป็นค่าจากการวัดค่าของระยะห่างระหว่างคู่ของคำรหัส เช่นคำรหัส 00110 และ 11000 มีค่าระยะทางเป็น 4 โดยทั่วไปจะเรียกว่าระยะ Hamming สำหรับรหัสแบบบล็อก ระยะต่ำสุดหรือ d_{min} ถูกนิยามให้เป็นค่าที่ต่ำที่สุดของคำรหัสทั้งหมด สำหรับรหัสเชิงเส้น d_{min} สามารถใช้ในการระบุประสิทธิภาพในการแก้ไขความผิดพลาดของรหัสโดยประมาณ

สำหรับรหัสคอนโวลูชันนอล ประสิทธิภาพในการควบคุมความผิดพลาดขึ้นอยู่กับวิธีการถอดรหัส คือ ถ้าตัวถอดรหัส ถอดรหัสเป็นรหัสที่มีความยาวคงที่ในแต่ละครั้ง ทำให้คล้ายกับการเข้ารหัสแบบบล็อก ซึ่งในกรณีนี้ d_{min} จะถูกใช้เป็นพารามิเตอร์หลัก อย่างไรก็ตามถ้าตัวถอดรหัสใช้เทคนิคที่เรียกว่า Maximum Likelihood Sequence Estimator (MLSE) ระยะทางที่สั้นที่สุดระหว่างคำรหัสจะเรียกว่า d_{free} จะเป็นตัวแปรสำคัญ เนื่องจากจำนวนของคำรหัสทั้งหมดของรหัสคอนโวลูชันนอลขึ้นอยู่กับช่วงเวลาในการสังเกตการณ์ จึงไม่แน่นอนที่จะคำนวณหาระยะห่างระหว่างคำรหัส เป็นข้อดีที่ระยะใกล้สุดไม่ได้ขึ้นอยู่กับเวลาที่สังเกตการณ์ ถ้าช่วงเวลาของการสังเกตการณ์นานพอ เพื่อให้เข้าใจง่ายขึ้น เราจะมาพิจารณาจากแผนผังเทลลิส ภาพที่ 8 ระยะทางที่สั้นที่สุดสำหรับรหัสตัวอย่างนี้จะเกิดขึ้นเมื่อช่วงเวลาที่สังเกตการณ์ $t \geq 3T$ ซึ่งมีค่าเท่ากับ 5 ซึ่งเป็นเส้นทางที่สั้นที่สุดที่เริ่มจากรหัส 00 และย้อนกลับไปรหัส 00 ตามเดิม ระยะใกล้สุดทำให้เราทราบจำนวนบิตต่ำสุดที่จะเปลี่ยนรหัสหนึ่งเป็นรหัสอื่น



ภาพที่ 8 แผนผังเทลลิสของตัวอย่างการเข้ารหัสคอนโวลูชันนอล

เราสามารถที่จะหาตัวเข้ารหัส 2 ตัว ที่สร้างกลุ่มของคำรหัสออกมาเหมือนกัน โดยจะเรียกตัวเข้ารหัสเหล่านั้นว่าเป็นประเภทของตัวเข้ารหัสที่เหมือนกัน ในการเท่ากันของรหัสคอนโวลูชันนอล จะเรียกว่าเท่ากัน ไม่ได้แค่เพียงสามารถสร้างกลุ่มคำรหัสออกมาเหมือนกันเท่านั้น แต่จะต้องมีคู่ของสถานะที่ตรงกันต้องเหมือนกันด้วย จากคุณสมบัติที่เท่ากันนี้ ทำให้สามารถเลือกตัวเข้ารหัสที่ซับซ้อนน้อยลงได้ ในการหาว่าตัวเข้ารหัสนั้นเป็นประเภทของตัวเข้ารหัสที่เท่ากัน โดยกำหนดเมทริกซ์ของการเข้ารหัสคอนโวลูชันนอลที่มีอัตรา k/n จำนวน 2 ชุด คือ $G(D)$ และ $G'(D)$ เป็นประเภทที่เท่ากัน จะมีเมทริกซ์ขนาด $b \times b$ คือ $T(D)$ ที่มีความสัมพันธ์ตามสมการ $G(D) = T(D)G'(D)$

รหัสรีงคอนโวลูชันนอลเป็นการใช้รหัสเชิงเส้นบนริงของเลขจำนวนเต็ม modulo M ซึ่ง $Z_M = (0, 1, 2, \dots, M-1)$ เนื่องจากวิธีการนี้เป็นวิธีการรวมรหัส M -ary Phase Modulation รหัสคอนโวลูชันนอลที่มีอัตราการเข้ารหัส k/n บน Z_M ถูกสร้างขึ้นมาด้วยวิธีเดียวกันกับในกรณีของรหัสคอนโวลูชันนอลบนฟิลด์ของเลขฐานสอง ยกเว้นสัมประสิทธิ์เป็นเลขจำนวนเต็มที่เป็นสมาชิกของ Z_M

2.2.2 การถอดรหัสช่องสัญญาณ

อัลกอริทึมของตัวถอดรหัสคอนโวลูชันแบ่งออกเป็นสองแบบ คือ ตัวถอดรหัสแบบพีชคณิต และตัวถอดรหัสแบบความน่าจะเป็น เราสนใจสำหรับตัวถอดรหัสแบบความน่าจะเป็นชนิดหนึ่งซึ่งรู้จักกันในชื่อ การถอดรหัสวิเทอร์บี อัลกอริทึมนี้ ถูกกล่าวขานว่าได้ผลดีที่สุด เนื่องจากการแก้ปัญหาสำหรับปัญหา การประมาณค่า MAP ของลำดับสถานะของกระบวนการ Finite-state discrete-time Markov โดยที่กระบวนการนี้เรียกว่ากระบวนการ Markov ก็ต่อเมื่อความน่าจะเป็นที่จะเป็นสถานะ x_k ที่เวลา k ที่ทำให้ทุกสถานะ จากเวลาที่ 0 ถึง $K-1$ ขึ้นอยู่กับสถานะของ x_{k-1} ที่เวลา $k-1$ เท่านั้น ตามสมการที่ (3)

$$P(x_k|x_0, x_1, \dots, x_{k-1}) = P(x_k|x_{k-1}) \quad (3)$$

การเข้ารหัสคอนโวลูชันสามารถมองเป็น กระบวนการ Finite-state discrete-time Markov โดยมีแผนผังของสถานะแสดงทุกเส้นทางที่เชื่อมโยงกับสถานะก่อนหน้า x_{k-1} ที่เวลา $k-1$ และทุกสถานะที่เป็นไปได้ที่เวลาปัจจุบัน x_k ที่เวลา k ความน่าจะเป็นในการเปลี่ยนสถานะ $P(u_k|u_{k-1})$ โดย u_k เป็นบิตของข้อมูล ปัญหาการประมาณค่าของลำดับแบบ MAP คือ เพื่อที่จะระบุลำดับของสถานะที่ $P(x|z)$ มีค่าสูงสุด เมื่อ z คือลำดับที่สังเกตการณ์ที่สัมพันธ์กับลำดับสถานะ x โดยที่ปัญหาการประมาณค่านี้เทียบเท่ากับการหาลำดับสถานะสำหรับ $P(x,z) = P(x|z)P(z) = P(z|x)P(x)$ มีค่าสูงสุด ความน่าจะเป็นร่วมกันของลำดับสถานะ x แสดงไว้ในสมการที่ (4)

$$\begin{aligned} P(x) &= P(x_0, x_1, \dots, x_{K-1}) \\ &= P(x_{K-1}|x_{K-2}, \dots, x_0)P(x_{K-2}|x_{K-3}, \dots, x_0) \dots P(x_1|x_0)P(x_0) \end{aligned} \quad (4)$$

เนื่องจากคุณสมบัติของกระบวนการ Markov ในสมการที่ (4) สามารถลดรูปได้เป็นสมการที่ (5)

$$P(x) = P(x_{K-1}|x_{K-2})P(x_{K-2}|x_{K-3}) \dots P(x_1|x_0)P(x_0) \quad (5)$$

โดยใช้กระบวนการ Markov และคุณสมบัติของการปราศจากสัญญาณรบกวน เราได้สมการที่ (6)

$$\arg \max_x P(z|x)P(x) = \arg \max_x \prod_{k=1}^{K-1} P(z_k|x_k, x_{k-1})P(x_k|x_{k-1}) \quad (6)$$

เนื่องจากฟังก์ชัน $\ln(\cdot)$ เป็นฟังก์ชันแบบโมโนโทนิก (Monotonic Function) โดยเราสามารถนำไปใช้กับสมการที่ (6) โดยผลลัพธ์ไม่มีการเปลี่ยนแปลง ทำให้สมการทางขวาเป็นไปตามสมการที่ (7)

$$\arg \max_x \sum_{k=1}^{K-1} \ln P(z_k|x_k, x_{k-1}) + \ln P(x_k|x_{k-1}) \quad (7)$$

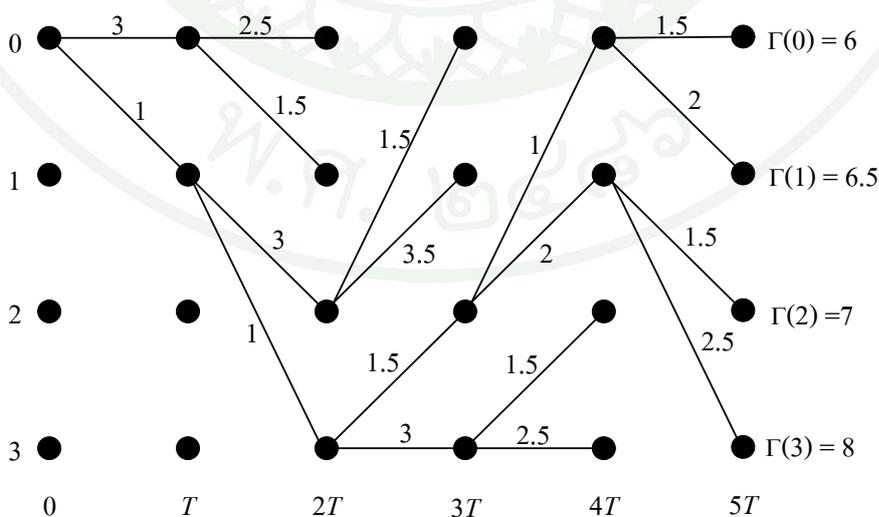
นอกจากนี้ ถ้าเรากำหนดค่าสำหรับแต่ละเส้นทางเป็น

$$\lambda(x_k, x_{k-1}) \triangleq -\ln P(z_k|x_k, x_{k-1}) - \ln P(x_k|x_{k-1}) \quad (8)$$

ทำให้สมการที่ (7) เปลี่ยนเป็นรูปอย่างง่ายได้ตามสมการที่ (9)

$$\arg \min_x \sum_{k=1}^{K-1} \lambda(x_k, x_{k-1}) \quad (9)$$

จากสมการที่ (9) จะเห็นได้ว่าอัลกอริทึมวิเทอร์บีเกี่ยวข้องเนื่องกับการหาระยะทางที่สั้นที่สุดในแผนผังเทลลิส



ภาพที่ 9 ตัวอย่างการหาเส้นทางที่สั้นที่สุดหลังรอบที่ 5 (5T)

ตัวอย่างการใช้อัลกอริทึมหาเส้นทางที่สั้นที่สุดตามภาพที่ 9 โดยแสดงเส้นทางที่เหลือหลังจากการทำรอบที่ 5 ถ้ามีการตัดสินใจ ณ เวลา n เส้นทางที่เลือกจะถูกเลือกจากเส้นทางที่เหลือที่มีค่าสถานะต่ำที่สุด ซึ่งในที่นี้คือที่สถานะศูนย์ ($\Gamma(0) = 6$) ซึ่งสามารถถอดรหัสได้เป็น 1, 1, 0, 0, 0

สำหรับการออกแบบรหัสคอนโวลูชันนอลนั้น เพื่อที่จะระบุถึงรหัสคอนโวลูชันนอลที่ดี อันดับแรกต้องกำหนดเกณฑ์ขึ้นมาก่อน โดยเกณฑ์ที่ดีที่สุดคือสามารถเลือกรหัสที่ทำให้มีความน่าจะเป็นที่จะเกิดความผิดพลาดได้น้อยที่สุด บนกำลังงานของสัญญาณรบกวน เนื่องจากประสิทธิภาพของรหัสคอนโวลูชันนอลที่กำลังงานของสัญญาณรบกวนต่ำๆ ไม่ได้ขึ้นอยู่กับแค่อัลกอริทึมการถอดรหัส แต่ขึ้นอยู่กับ d_{free} ด้วย ดังนั้นรหัสที่ดีที่สุดจะมีค่า d_{free} สูงที่สุด

ตารางที่ 1 ตัวเข้ารหัสคอนโวลูชันนอลแบบฐานสองที่มีอัตราเท่ากับ 1/2 ที่ดีที่สุดและ d_{free}

Number of delays	Best Encoders	d_{free}
2	$[D^2 + D + 1, D^2 + 1]$	5
3	$[D^3 + D^2 + D + 1, D^2 + D + 1]$	6
4	$[D^4 + D^2 + D + 1, D^4 + D^3 + 1]$	7
5	$[D^5 + D^3 + D^2 + D + 1, D^5 + D^4 + D^2 + 1]$	8
6	$[D^6 + D^3 + D^2 + D + 1, D^6 + D^5 + D^3 + D^2 + 1]$	10
7	$[D^7 + D^4 + D^3 + D^2 + D + 1, D^7 + D^6 + D^5 + D^2 + 1]$	10
8	$[D^8 + D^7 + D^5 + D^3 + D^2 + D + 1, D^8 + D^4 + D^3 + D^2 + 1]$	12

ตารางที่ 1 สรุปตัวเข้ารหัสคอนโวลูชันนอลแบบฐานสองที่มีอัตราเท่ากับ 1/2 ที่ดีที่สุด จะสังเกตว่าเมื่อความซับซ้อนของตัวเข้ารหัสเพิ่มขึ้น ประสิทธิภาพของรหัสคอนโวลูชันนอลที่แสดงไว้ด้วย d_{free} จะเพิ่มตามไปด้วย เนื่องจากรหัสคอนโวลูชันนอลได้นำมาใช้เป็นการเข้ารหัสของสัญญาณตามด้วยการมอดูเลตบางอย่าง ดังนั้นการออกแบบของรหัสคอนโวลูชันนอลบางครั้งจะรวมเอารูปคลื่นสัญญาณที่ถูกสร้างจากมอดูเลเตอร์ ข้อดีคือได้รหัสประสิทธิภาพดี โดยไม่กระทบกับประสิทธิภาพของแบนด์วิดท์ นอกจากนี้รูปคลื่นสัญญาณจะ

แทนค่ารหัสโดยตรงและมีความแตกต่างกันด้วย จึงทำให้มีความยืดหยุ่นต่อสัญญาณรบกวน โดยหลักการนี้รู้จักกันในชื่อ Trellis-coded modulation (TCM)

2.3 การมอดูเลตช่องสัญญาณ

การมอดูเลตแบบ Continuous Phase Modulation หรือเรียกย่อๆ ว่า CPM จัดว่าเป็นการมอดูเลตดิจิทัลที่ไม่เป็นเชิงเส้นพร้อมหน่วยความจำ เนื่องจากเฟสของสัญญาณ CPM จะถูกจำกัดให้มีความต่อเนื่องกัน CPM กลายเป็นที่นิยมสำหรับช่องสัญญาณที่มีอยู่จำกัด เนื่องจากการเพิ่มขึ้นของความถี่สำหรับการใช้แบนด์วิดธ์อย่างคุ้มค่าโดยใช้สัญญาณที่มีแอมพลิจูดคงที่ ซึ่งคุณสมบัตินี้ทำให้สามารถต้านทานต่อสัญญาณรบกวนบนช่องสัญญาณที่ไม่เป็นเชิงเส้น ตัวอย่างที่ดีของการใช้งาน CPM คือระบบสื่อสารผ่านดาวเทียม ที่มีลักษณะ

- ในการส่งสัญญาณขาหลัง จะใช้ตัวขยายสัญญาณแบบไม่เป็นเชิงเส้นกำลังสูง ดังนั้นจึงเกิดการลดทอนได้มาก
- ใช้งานย่านความถี่ UHF (300 – 3000 MHz) และ VHF (30 – 300 MHz) ซึ่งโดยปกติจะถูกทำให้เสียหายด้วยการเฟดดิ้ง

ปัญหาดังกล่าวสามารถบรรเทาด้วยการใช้การมอดูเลตที่สร้างสัญญาณ Envelope ที่คงที่ ดังนั้น CPM จึงเป็นทางเลือกที่เหมาะสม ระบบ CPM โดยทั่วไปจะส่งสัญญาณในรูปแบบของ

$$sm(t, \underline{\alpha}) = \sqrt{\frac{2E}{T}} \cos(2\pi f_0 t + \varphi(t, \underline{\alpha}) + \varphi_0) \quad (10)$$

เมื่อ $\underline{\alpha}$ คือลำดับของสัญลักษณ์ข้อมูลแบบ M -ary ที่ไม่มีความสัมพันธ์กัน โดยที่ $\alpha_i \in \{\pm 1, \pm 3, \dots, \pm(M-1)\}$ โดยที่ เฟสเป็นตัวนำพาข้อมูลคือ $\varphi(t, \underline{\alpha})$ จากสมการที่ (11)

$$\varphi(t, \underline{\alpha}) = 2\pi h \sum_{i=-\infty}^{\infty} \alpha_i f(t - iT) \quad -\infty < t < \infty \quad (11)$$

เมื่อ h คือ Modulation Index และ $f(t)$ คือ เฟสตอบสนองของย่านความถี่ฐาน ถ้าถูกกำหนดอยู่ในเทอมของความถี่ $g(t)$ จะได้เป็น $f(t) = \int_{-\infty}^t g(\tau) d\tau, -\infty < t < \infty$ สำหรับวิธีการ

Causal CPM ความถี่ $g(t)$ ต้องสอดคล้องกับ $g(t) = 0, t < 0, t > LT$ และ $g(t) \neq 0, 0 \leq t \leq LT$ ดังนั้น เฟสที่นำพาข้อมูลสามารถเขียนเป็นสมการที่ (12)

$$\varphi(t, \underline{\alpha}) = \pi h \sum_{i=-\infty}^{n-L} \alpha_i + 2\pi h \sum_{i=n-L+1}^n \alpha_i f(t - iT), nT \leq t \leq (n+1)T \quad (12)$$

วิธีการแบบ CPM ถูกกำหนดโดย h, M และ $g(t)$ หรือ $f(t)$ เรามักใช้ $g(t)$ ในรูปแบบของคลื่นสัญญาณสี่เหลี่ยม (L-REC) ตามสมการที่ (13)

$$g(t) = \begin{cases} \frac{1}{2LT}, & 0 \leq t \leq LT \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

หรือฟังก์ชัน Raised Cosine (L-RC) ตามสมการที่ (14)

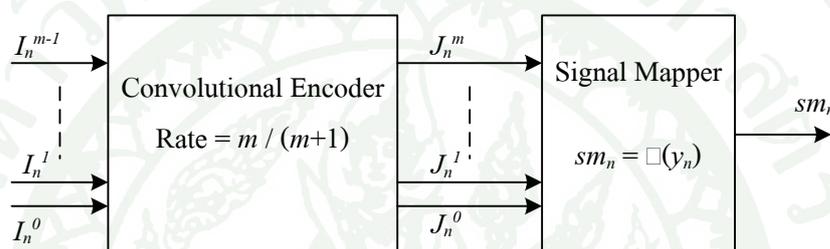
$$g(t) = \begin{cases} \frac{1}{2LT} [1 - \cos(\frac{2\pi t}{LT})], & 0 \leq t \leq LT \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

ถ้า $L = 1$ จะเรียกการมอดูเลตนั้นว่า Full Response CPM แต่ถ้า $L > 1$ จะเรียกว่า Partial Response CPM สำหรับการมอดูเลตที่เรียกว่า Continuous Phase Frequency Shift Keying หรือ CPFSK เป็นกรณีพิเศษของการมอดูเลตแบบ Full Response CPM พร้อมด้วยผลการตอบสนองความถี่ฐานแบบเชิงเส้น หรือ $g(t) = 1-REC$ แบบดัดแปลงที่ถูกใช้โดย สัญญาณ CPM โดยทั่วไป ขึ้นอยู่กับการเลือกค่า $h, g(t)$ และจำนวนของสัญญาณ M โดยถ้าใช้ค่า h สูงจะสลับเปลี่ยนแบนด์วิดท์สูงด้วย และการเลือกใช้ Raised Cosine Frequency จะใช้แบนด์วิดท์ต่ำกว่า Rectangular Frequency และสำหรับการมอดูเลตแบบ Patial Response จะใช้แบนด์วิดท์ต่ำกว่า Full Response

2.3.1 การมอดูเลตรหัสเทลลิสสำหรับ CPM

Trellis-code modulation หรือ TCM เป็นเทคนิคการรวมการเข้ารหัสและการมอดูเลตเข้าไว้ด้วยกัน สำหรับการส่งข้อมูลดิจิทัล เนื่องจากคุณสมบัติที่ดีเหนือกว่าการมอดูเลตหลายระดับของของรหัสที่ไม่ได้เข้ารหัสคอนโวลูชันนอล โดยไม่กระทบกับประสิทธิภาพการใช้แบนด์วิดท์ TCM เป็นที่นิยมใช้กับช่องสัญญาณที่มีแบนด์วิดท์จำกัด วิธีการนี้จะสร้างลำดับของสัญญาณที่ถูกเข้ารหัสแล้วโดยใช้การเข้ารหัสแบบ Finite-state และการเลือกสัญญาณที่มีการมอดูเลตร่วมกัน แนวคิดในการออกแบบที่สำคัญของ TCM คือกระบวนการเข้ารหัสและการ

มอดูเลตถูกออกแบบร่วมกัน โดยการทำให้ Minimum Euclidean Distance มีค่ามากที่สุด หรือที่เรียกว่า “free distance” ระหว่างลำดับของสัญญาณที่เข้ารหัสมาแล้ว เป็นผลทำให้ free distance ของสัญญาณที่ถูกเข้ารหัสแล้วใหญ่กว่า minimum distance ระหว่างสัญญาณที่ไม่ถูกเข้ารหัส ที่อัตราข่าวสารที่เท่ากัน และแบนด์วิดท์โดยประมาณเท่ากัน แผนผังการทำงานของ TCM แสดงไว้ในภาพที่ 10 ลำดับของข่าวสารแสดงไว้เป็น $\underline{I} = (I_0, I_1, \dots, I_n, \dots)$ เมื่อ $I_0 = (I_n^0, I_n^1, \dots, I_n^{m-1})$ การเข้ารหัสคอนโวลูชันนอลที่มีอัตราการเข้ารหัส $\frac{m}{m+1}$ สร้างลำดับของรหัส $\underline{J} = (J_0, J_1, \dots, J_n, \dots)$ เมื่อ $J_0 = (J_n^0, J_n^1, \dots, J_n^m)$ จากนั้นจะนำรหัสที่ได้จับคู่กับลำดับของสัญญาณ $\underline{sm} = (sm_0, sm_1, \dots, sm_n, \dots)$



ภาพที่ 10 แผนผังการทำงานของ TCM

2.3.2 รหัสโพลีโนเมียลริงคอนโวลูชันนอลเทลลิส

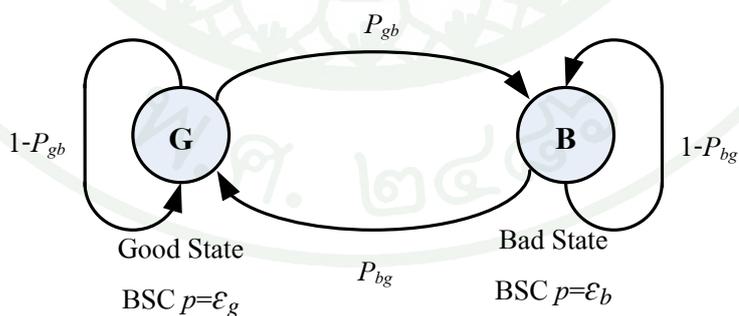
ก่อนทำกระบวนการเข้าและถอดรหัสสัญญาณเทลลิส TCM จะต้องเปลี่ยนข้อมูล 1 บิตเป็นสัญลักษณ์ โดย 1 สัญลักษณ์เท่ากับ 2 บิต เพื่อให้สามารถผ่านการเข้ารหัสแบบริงได้ การเข้ารหัสแบบริงมีข้อดีคือ สามารถออกแบบใช้ร่วมกับการมอดูเลตแบบเฟสของสัญญาณมีความต่อเนื่อง เช่น การมอดูเลตแบบ CPM การเข้ารหัสช่องสัญญาณแบบเทลลิสประกอบด้วย 2 ส่วน คือ การเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชันนอล และการมอดูเลตแบบเฟสของสัญญาณมีความต่อเนื่อง เนื่องจากการมอดูเลตที่เหมาะสมกับช่องสัญญาณไร้สาย ซึ่งมีแถบความถี่แคบ นอกจากนี้การเลือกเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชันนอลสามารถรวมกับการมอดูเลตได้อย่างกลมกลืน การถอดรหัสช่องสัญญาณแบบเทลลิสเลือกใช้การถอดรหัสแบบวีเทอร์บี เป็นเทคนิคที่สามารถตรวจจับผิดพลาดได้ดีที่สุด นอกจากนี้การคำนวณค่าขนาดในแต่ละเส้นทางบนเทลลิสจะใช้ทฤษฎีการตรวจจับสัญญาณแบบ ML และ MAP

สำหรับการนำไปใช้ในช่องสัญญาณไร้สายนั้น ต้องการกระบวนการมอดูเลตที่มีการใช้ช่วงความถี่อย่างมีประสิทธิภาพ ซึ่งจะใช้การมอดูเลตแบบที่เฟสของสัญญาณมีความต่อเนื่อง โดยมีดัชนีการมอดูเลตเท่ากับ $1/4$ ซึ่งสัญญาณที่ได้นั้นมีการใช้ช่วงความถี่เท่ากับกระบวนการ MSK โดยกระบวนการมอดูเลตแบบที่เฟสของสัญญาณมีความต่อเนื่องจะแยกออกเป็น CPE และ MM ซึ่งทั้งหมดจะรวมกันอยู่ภายในกระบวนการเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชันนอลที่อัตราเท่ากับ $1/2$

2.4 ช่องสัญญาณ

2.4.1 ช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

โดยปกติ การจำลองช่องสัญญาณไร้สาย จะใช้สถิติแบบ Rayleigh หรือ Rician โดยที่การจำลองช่องสัญญาณแบบเปิดและไม่มีสัญญาณในเส้นทางตรงเลย จะเหมาะสมกับการจำลองช่องสัญญาณแบบ Rayleigh และสำหรับการจำลองช่องสัญญาณภายในและมีสัญญาณเส้นทางตรงด้วย จะเหมาะสมกับการจำลองช่องสัญญาณแบบ Rician ในงานวิจัยนี้ ไม่ได้สนใจแค่ช่องสัญญาณแบบ Rician เฟดดิ้ง เท่านั้น ยังสนใจในกรณีที่มีสัญญาณรบกวนที่มีความเกี่ยวเนื่องกันสูงด้วย เช่น Burst noise ซึ่งเป็นประเภทของสัญญาณรบกวนที่ลดประสิทธิภาพของระบบการสื่อสารไร้สายอย่างชัดเจน



ภาพที่ 11 แบบจำลองของ Gilbert-Elliott สำหรับช่องสัญญาณ Burst-Noise

ที่มา: Elliot (1963)

สำหรับการจำลองแบบแรกที่จะอธิบายพฤติกรรมของช่องสัญญาณ แบบ Burst ได้เสนอ โดย Gilbert (1960) การจำลองของ Gilbert ประกอบด้วย Markov chain สองสถานะ โดยที่ สถานะ “Good” จะไม่มีความผิดพลาดเกิดขึ้น ในขณะที่ สถานะ “Bad” เกิดความผิดพลาดขึ้นด้วยความน่าจะเป็น ϵ_b ความน่าจะเป็นที่จะเกิดความผิดพลาดขึ้นทั้งหมดของโมเดลนี้คือ $\bar{\epsilon} = \frac{P_{gb}\epsilon_b}{P_{gb}+P_{bg}}$ เมื่อ $\frac{P_{gb}}{P_{gb}+P_{bg}}$ คือ ความน่าจะเป็นที่จะเข้าสู่สถานะ Bad จากนั้น Elliott (1963) ได้ขยายความการจำลองของ Gilbert โดยแนะนำถึง ความน่าจะเป็นที่จะเกิดความผิดพลาดขึ้นที่สถานะ Good ด้วยความน่าจะเป็น ϵ_g ดังแสดงไว้ในภาพที่ 11 ความน่าจะเป็นที่จะเกิดความผิดพลาดขึ้นสำหรับแบบจำลองนี้คือ $\bar{\epsilon} = \frac{P_{bg}\epsilon_g+P_{gb}\epsilon_b}{P_{gb}+P_{bg}}$ สำหรับการวิจัยนี้ได้เน้นประสิทธิภาพของระบบสำหรับช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง จากภายใต้สมมติฐานนี้ จะเหมาะสมกับแอปพลิเคชันหลายอย่างที่มีอัตราการรับส่งข้อมูลสูง ซึ่งโดยปกติความสัมพันธ์ทางด้านเวลาของช่องสัญญาณจะนานกว่าช่วงเวลาของสัญลักษณ์หนึ่งหลายๆ ซึ่งจะทำให้เกิดผลกระทบกับการเลือนหายของหลายๆสัญลักษณ์ (Malkamaki, 1997; Ray, 2009)

2.4.2 การถอดรหัสแบบ MAP บนช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

โดยการสมมติให้เป็นการเฟดดิ้งอย่างช้าอย่างเพียงพอ จะทำให้การเลือนของเฟสเป็น 0 จึงได้สัญญาณความถี่ต่ำเทียบเท่าที่รับได้ ลดรูปเป็นสมการที่ (15)

$$r_1(t) = (1 + \beta)s_{ml}(t) + n_{o1}(t) \quad (15)$$

ในการสร้างการถอดรหัสแบบ MAP สำหรับระบบ CPFSK เราจะสังเกต ตัวอย่างข้อมูล $z_{k,i}$ ระหว่างช่วงเวลา $kT \leq t \leq (k+1)T$ โดยเขียนเป็นสมการได้เป็น $z_{k,i} = (1 + \beta)s_{m_{k,i}} + n_{k,i}$ เมื่อ $i = 1, 2, \dots, N$ สำหรับช่องสัญญาณเฟดดิ้งอย่างช้า β คือค่าคงที่ระหว่างแต่ละช่วงของสัญญาณ โดยถ้ามีการประมาณค่า Channel State Information หรือ CSI เราสามารถทราบค่า β และนำไปใช้ร่วมกับการถอดรหัสได้ โดยให้มีข้อมูล CSI เราจะได้เงื่อนไขความน่าจะเป็น ตามสมการที่ (16)

$$P(z_k|x_{2,k+1}x_{2,k}) = P(z_k|sm_k) = \frac{1}{(\sqrt{\pi N_0})^N} \exp \left[-\sum_{i=1}^N \frac{(z_{k,i} - \alpha sm_{k,i})^2}{N_0} \right] \quad (16)$$

โดยที่ $\alpha = (1 + \beta)$ เมื่อนำสมการที่ (16) แทนในสมการที่ (8) จะได้การคำนวณค่าขนาดในแต่ละเส้นทางบนเทลลิส $\lambda(sm_k)$ เป็น

$$\lambda(sm_k) = -\sum_{i=1}^N z_{k,i} sm_{k,i} - \frac{N_0}{2\alpha} \ln P(x_{2,k+1}|x_{2,k}) \quad (17)$$

ถ้าเราแทนที่ลำดับของอินพุตด้วย u ที่มาจากกระบวนการ Markov จะได้สมการการหาเส้นทางเทลลิสเป็น

$$\lambda(sm_k) = -\sum_{i=1}^N z_{k,i} sm_{k,i} - \frac{N_0}{2\alpha} \ln P(u_k|u_{k-1}) \quad (18)$$

จากสมการที่ (18) ถ้าใช้เฉพาะเทอมแรกจะเป็นการถอดรหัสแบบ ML เมื่อมีการใช้ตัวเข้ารหัสภายนอกอยู่ในระบบด้วย เราจะต้องพิจารณาความน่าจะเป็นอย่างมีเงื่อนไข $P(z_k|S_{0,k+1}, S_{0,k})$ แทน $P(z_k|x_{2,k+1}, x_{2,k})$ ซึ่งจะมีผลกระทบบกับแอมพลิจูดของการเฟดดิ้งของความน่าจะเป็นอย่างมีเงื่อนไข $P(z_k|S_{0,k+1}, S_{0,k})$ จากความสัมพันธ์ตามสมการที่ (16) จึงได้สมการคำนวณเส้นทางเทลลิสเป็น

$$\lambda(sm_k) = -\alpha_1 \sum_{i=1}^N z_{k,i} sm_{k,i} - \alpha_2 \sum_{i=1}^{2N} z_{k,i} sm_{k,i} - \frac{N_0}{2} \ln P(S_{0,k+1}|S_{0,k}) \quad (19)$$

โดยที่ α_1 และ α_2 เป็นค่าคงที่ของแอมพลิจูดการเฟดดิ้งระหว่างช่วงของสัญญาณแรกและสัญญาณที่สอง

3. การปรับปรุงระบบการส่งรูปภาพบนช่องสัญญาณไร้สาย

3.1 การสลับลำดับข้อมูลแบบสุ่ม

ก่อนที่จะอธิบายถึงกระบวนการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ จะขออธิบายถึงกระบวนการสลับลำดับข้อมูลแบบสุ่มก่อน โดยฝั่งส่งข้อมูลของการสลับลำดับ (d_{INT}) จะถูกคำนวณจากสมการ $d_{INT} = dP$ โดยที่ d คือเวกเตอร์ข้อมูล และ P คือเมทริกซ์สุ่ม ซึ่งการสร้างเมทริกซ์นี้สามารถทำได้โดยการใช้อัลกอริทึม Linear Congruential และในฝั่งรับ เวกเตอร์ข้อมูลสามารถสลับลำดับกลับมาจากสมการ $d = d_{INT}P^{-1}$ แต่เนื่องจากว่า เมทริกซ์ P เป็นเมทริกซ์แบบ

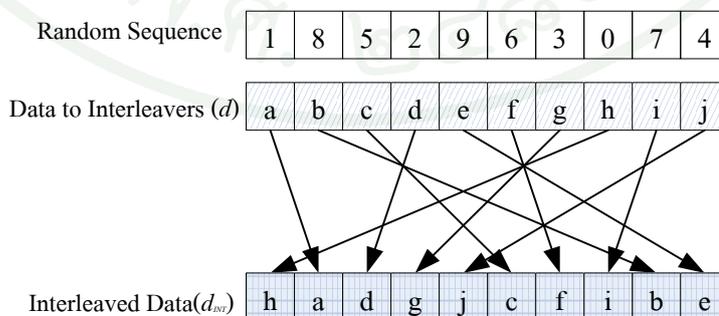
Unitary ซึ่ง $PP^T = P^T P = I$ และ $P^{-1} = P^T$ เมื่อ $(\cdot)^T$ คือการทรานสโพส และ I คือเมทริกซ์เอกลักษณ์ ดังนั้นเราจึงสามารถคำนวณหาเวกเตอร์ข้อมูล d ได้จาก $d = d_{INT} P^T$

จากตัวอย่าง ถ้าเราได้ลำดับสุ่ม คือ 1, 8, 5, 2, 9, 6, 3, 0, 7, 4 เราจะได้เมทริกซ์ P และ P^T ตามภาพที่ 12 โดยเมทริกซ์ P เป็นตัวกำหนดตำแหน่งที่จะสลับลำดับข้อมูล และเมทริกซ์ P^T เป็นตัวกำหนดตำแหน่งที่จะสลับลำดับข้อมูลกลับ

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad P^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

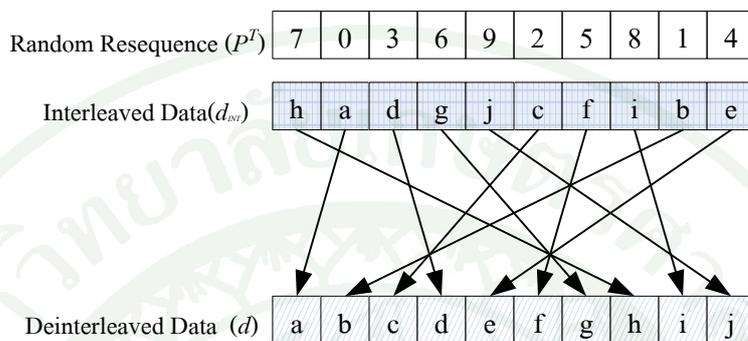
ภาพที่ 12 เมทริกซ์ P และ P^T สำหรับ ลำดับสุ่ม 1, 8, 5, 2, 9, 6, 3, 0, 7, 4

ที่ส่งส่งเมื่อนำเวกเตอร์ของข้อมูล d ที่จะสลับลำดับ คูณกับเมทริกซ์ P ทำให้ตำแหน่งของข้อมูลในเวกเตอร์จะสลับไปตามลำดับที่สุ่มมาได้ เป็นเวกเตอร์ที่ถูกสลับลำดับแบบสุ่ม d_{INT} ตามภาพที่ 13



ภาพที่ 13 ตัวอย่างเวกเตอร์ d_{INT} ที่ได้จากการสลับลำดับตาม ลำดับสุ่ม 1, 8, 5, 2, 9, 6, 3, 0, 7, 4

ในฝั่งรับจะสลับลำดับข้อมูลกลับ โดยใช้ข้อมูลที่ถูกสลับลำดับข้อมูลมา d_{INT} คูณกับเมทริกซ์ P^T ทำให้ตำแหน่งของข้อมูลในเวกเตอร์ d_{INT} จะสลับกลับไปเป็นเวกเตอร์ d ดังเดิมตามภาพที่ 14



ภาพที่ 14 ตัวอย่างเวกเตอร์ d ที่ได้จากการสลับลำดับข้อมูลกลับ

ข้อมูลที่ถูกส่งผ่านช่องสัญญาณที่มีการเลือนหาย จะทำให้ข้อมูลเสียหายเป็นกลุ่มๆ เช่น เกิดความเสียหายต่อเนื่องกันสามสัญลักษณ์เริ่มจากสัญลักษณ์ที่สองถึงสี่ เมื่อนำเอาการสลับลำดับข้อมูลแบบสุ่มมาใช้ก่อนส่งออกไป สัญลักษณ์ที่เสียหายคือ d, g, j เมื่อสลับลำดับข้อมูลกลับมา ผลจะทำให้ข้อมูลที่รับเกิดการเสียหายกระจายออกไปเป็นตำแหน่งที่สี่ เจ็ดและเก้าแทน ซึ่งในฝั่งรับจะเสมือนมีการเสียหายของข้อมูลแบบสุ่ม ซึ่งทำให้สามารถลดความเสียหายเป็นกลุ่ม (Burst error และ block error) ได้ แต่เนื่องจากในความเป็นจริงขนาดของเวกเตอร์ที่ใช้ในการสลับลำดับข้อมูลเป็นไปตามขนาดของแพ็คเกจ ซึ่งมีความถี่แปรเปลี่ยน ซึ่งมีความถี่แตกต่างกันตลอดเวลา และมีขนาดค่อนข้างใหญ่ จึงทำให้การสลับลำดับข้อมูลแบบสุ่มมีความซับซ้อนสูงและใช้ทรัพยากรในการประมวลผลสูง

3.2 การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้

ผลสำเร็จหลัก ของการใช้การสลับลำดับข้อมูลคือ การลดความเสียหายของภาพที่รับได้จากความผิดพลาดเป็นกลุ่ม (Burst error และ block error) ซึ่งเกิดจากการย้ายลำดับที่เกิดจากความผิดพลาดเป็นกลุ่มให้เป็นเสมือนการเกิดความผิดพลาดแบบสุ่มทั่วไปที่ฝั่งรับ จึงทำให้ภาพที่รับได้เสียหายน้อยลงเมื่อเทียบกับการไม่ใช้การสลับลำดับข้อมูล อย่างไรก็ตามการเพิ่มการสลับลำดับข้อมูลเข้าไปในระบบรับส่งข้อมูลภาพ ทำให้ระบบมีความซับซ้อนเพิ่มขึ้นเป็นอย่างมาก ใน

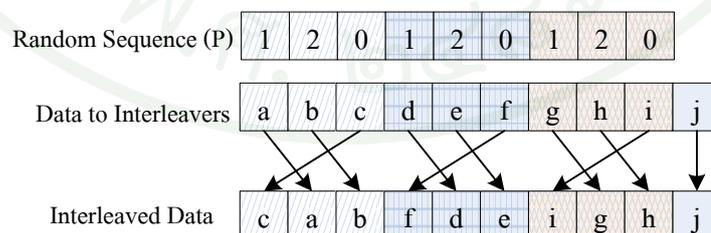
งานนี้เราพยายามที่จะบรรเทาปัญหานี้ โดยการออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ แทนที่จะทำการสลับลำดับข้อมูลทั้งแพ็คเกจ เราจะทำการสลับลำดับข้อมูลตามขนาดที่ต้องการ ซึ่งยังคงมีประสิทธิภาพที่ดีอยู่ แต่ใช้เวลาในการประมวลผลที่เร็วกว่า เราจะแทนขนาดที่จะทำการสลับลำดับด้วย N_R

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad P^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

ภาพที่ 15 เมทริกซ์ P และ P^T สำหรับ $N_R = 3$ และมีลำดับสุ่มเป็น 1, 2, 0

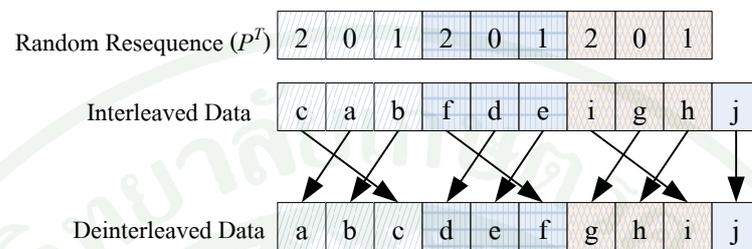
จากตัวอย่าง เรากำหนดให้ N_R มีขนาดเป็น 3 ดังนั้นเราจึงต้องสร้างเมทริกซ์ P และเมทริกซ์ P^T ให้มีขนาดเป็น 3×3 ซึ่งต้องมีการสร้างลำดับสุ่มออกมาสามค่า โดยในที่นี้ สมมติให้ลำดับสุ่มที่สุ่มออกมาได้เป็น 1, 2, 0 โดยจะได้ เมทริกซ์ P และเมทริกซ์ P^T ตามภาพที่ 15

การสลับลำดับข้อมูลตามตัวอย่างภาพที่ 16 เราสมมติให้เวกเตอร์ข้อมูล d ที่มีจำนวน 10 สัญลักษณ์คือ $\{a, b, c, d, e, f, g, h, i, j\}$ และแทนที่จะสลับลำดับของข้อมูลทั้งแพ็คเกจ (ขนาดโดยเฉลี่ยพอประมาณ คือ 325 สัญลักษณ์) เราจะแบ่งเวกเตอร์ข้อมูลออกเป็นขนาดใหม่ให้มีค่าเท่ากับ N_R คือ 3 แล้วทำการสลับลำดับข้อมูลโดยการคูณแต่ละ 3 สัญลักษณ์กับเมทริกซ์ P ซึ่งจะเป็นการสลับลำดับข้อมูลที่ละ 3 สัญลักษณ์ตามลำดับสุ่มที่ได้มา ยกเว้นเศษจากการแบ่งส่วนจะถูกส่งออกไปโดยตรงโดยไม่ถูกสลับลำดับ



ภาพที่ 16 ตัวอย่างการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่มี $N_R = 3$

จากตัวอย่างภาพที่ 17 แสดงการสลับลำดับข้อมูลกลับโดยใช้ข้อมูลที่ถูกสลับลำดับข้อมูลมาแล้ว คูณกับเมทริกซ์ P^T ทำให้ตำแหน่งของข้อมูลที่ถูกสลับลำดับข้อมูลมาแล้ว สลับกลับไปเป็นเวกเตอร์ d ตามเดิม และเศษจากการแบ่งส่วนจะไม่ถูกสลับกลับ



ภาพที่ 17 ตัวอย่างการสลับลำดับข้อมูลกลับ

เนื่องจากขนาดของ N_R มีความยาวน้อยกว่าความยาวของทั้งแพ็คเกจ จึงทำให้ความซับซ้อนลดลงอย่างชัดเจน ดังได้แสดงไว้ตามตัวอย่างภาพที่ 15-17 โดยเทียบกับภาพที่ 12-14 ซึ่งจะเห็นว่าเมทริกซ์ P และเมทริกซ์ P^T ที่ขนาดแตกต่างกันอย่างเห็นได้ชัด จึงทำให้กระบวนการอื่นๆ สามารถทำงานได้รวดเร็วกว่าอย่างชัดเจน แต่มีข้อสังเกตว่าเศษจากการแบ่งด้วยขนาด N_R จะไม่นำมาสลับลำดับข้อมูลด้วย ซึ่งนั่นอาจจะไม่สำคัญเท่าไรหากขนาดของ N_R มีขนาดไม่ยาวมากนักเหมือนในตัวอย่าง แต่อย่างไรก็ตามความยาวของ N_R จะมีผลอย่างมากถ้าความยาวของ N_R มีความยาวเพิ่มมากขึ้น

อุปกรณ์และวิธีการ

อุปกรณ์

1. เครื่องคอมพิวเตอร์ส่วนบุคคล
2. ระบบปฏิบัติการ Windows Vista
3. โปรแกรม Microsoft Office
4. โปรแกรม Microsoft Excel
5. โปรแกรม Java Netbeans IDE
6. โปรแกรม Mathcad

วิธีการ

งานวิจัยนี้ได้นำเอาระบบรับส่งภาพต้นแบบมาจาก Mahapakulchai and Van Dyck (2004) แล้วศึกษาการสลับลำดับข้อมูลแบบสุ่มจาก Mahapakulchai and Sapasirisopon (2008a) จากนั้นได้ปรับปรุงกระบวนการสลับลำดับข้อมูลแบบสุ่มที่กระทำกับข้อมูลทั้งแพ็คเกจให้สามารถปรับเปลี่ยนขนาดการสลับลำดับข้อมูลตามช่วงที่กำหนดไว้ได้ เพื่อลดความซับซ้อนของการสลับลำดับข้อมูล ในขณะที่คุณภาพของภาพที่รับได้ยังคงดีอยู่ โดยเราสามารถทำให้ขนาดที่จะสลับลำดับแปรเปลี่ยนไปตามลักษณะของการเลือนหายของสัญญาณ โดยที่คุณภาพของสัญญาณเป็นไปตามที่เราที่กำหนดไว้ โครงสร้างของระบบการรับส่งภาพแสดงไว้ตามภาพที่ 1 โดยการออกแบบระบบรับส่งภาพต้นแบบ การออกแบบการปรับปรุงระบบเดิมโดยใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ และการจำลองช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

1. การออกแบบระบบการรับส่งข้อมูลภาพของระบบต้นแบบ

การเข้ารหัสข้อมูลรูปภาพแบบ MPEG-4 ของรูปภาพ Lena เริ่มจากการสร้างกระแสบิตของรูปภาพที่ถูกเข้ารหัสแล้ว ซึ่งเป็นค่าของสัมประสิทธิ์ LFS และ HFS เพื่อเพิ่มความเชื่อถือได้ กระแสบิตของ HFS ถูกแบ่งไปเป็นแพ็คเกจที่มีขนาดแตกต่างกัน จากนั้นกระแสบิตเหล่านี้ถูกแปลงไปเป็นสัญลักษณ์ (2 บิตต่อสัญลักษณ์) โดยที่ กลุ่มบิต 00 แทนด้วยสัญลักษณ์ 0 กลุ่มบิต 01 แทนด้วยสัญลักษณ์ 1 กลุ่มบิต 10 แทนด้วยสัญลักษณ์ 2 และ กลุ่มบิต 11 แทนด้วยสัญลักษณ์ 3 จำนวนแพ็คเกจ

เกิดทั้งหมดที่ได้สำหรับรูปภาพ Lena คือ 102 แพ็กเก็ต โดยเป็นแพ็กเก็ตของ LFS จำนวน 1 แพ็กเก็ต และแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต และมีขนาดโดยเฉลี่ยประมาณ 325 สัญลักษณ์ต่อแพ็กเก็ต

การเข้ารหัสช่องสัญญาณแบบเทลลิส ประกอบด้วย 2 ส่วน คือการเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชัน และการมอดูเลตแบบเฟสของสัญญาณต่อเนื่องกัน (CPM) ถ้าเราเข้ารหัสของ CPE ที่มีอัตราเท่ากับ 1/2 จะเข้ารหัสดังสมการที่ (20)

$$G_{CPE} = \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3D & D & 1 & 0 \end{bmatrix} \quad (20)$$

งานวิจัยนี้ได้นำการเข้ารหัสโพลีโนเมียล ริงคอนโวลูชัน 32 ชั้น โดยการทำการวนการ RCE ที่อัตราเท่ากับ 1/2 คู่กับการทำการวนการ CPE ที่อัตราเท่ากับ 2/4 จะได้ Generator Matrix ของการเข้ารหัสแบบริงคอนโวลูชัน ตามสมการที่ (21)

$$\begin{aligned} G_{all}(D) &= \begin{bmatrix} 1 + D + 2D^3 \\ 2 + D \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3D & D & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 + 3D + 3D^2 + 2D^3 \\ 2D + D^2 \\ 1 + 2D^3 \\ 1 + D + 2D^3 \end{bmatrix} \end{aligned} \quad (21)$$

ที่ฝั่งรับมีการคำนวณหาเส้นทางบนเทลลิส โดยใช้ทฤษฎีการตรวจจับสัญญาณแบบ MAP จะใช้กับสัญลักษณ์ซีโรเทรีเท่านั้น ดังนั้นจึงมีการเพิ่มตัวบ่งชี้หรือ Flag เข้าไป เพื่อใช้เป็นบิตตรวจจับเงื่อนไขในการหาเส้นทางบนเทลลิส และใช้ การถอดรหัสวีเทอร์บี แล้วจึงใช้การถอดรหัสกระแสบิตด้วย การถอดรหัสแบบ MPEG-4 ได้ออกมาเป็นรูปภาพ Lena ที่ฝั่งรับ

งานวิจัยนี้มีการจำลองช่องสัญญาณไร้สาย โดยจำลองการเกิดการเลื่อนหายของสัญญาณในเชิงขนาดแบบช้า โดยที่เวลาของ Channel coherence มีความยาวกว่าช่วงเวลาของหนึ่งสัญลักษณ์มาก ทำให้ผลของการเลื่อนหายเดียวกันเกิดขึ้นกับหลายๆสัญลักษณ์

2. การปรับปรุงระบบการรับส่งข้อมูลภาพ

การปรับปรุงระบบการรับส่งข้อมูลภาพเริ่มจากการใช้กระบวนการสลับลำดับข้อมูลแบบบล็อก Mahapakulchai and Thongnumpen (2007) พบว่ากระบวนการมีความยุ่งยากเนื่องจาก ต้องหาเมทริกซ์การสลับลำดับข้อมูลตามความยาวของแต่ละแพ็กเก็ต ซึ่งแพ็กเก็ตมีความยาวไม่คงที่ และการออกแบบเมทริกซ์การสลับลำดับข้อมูลอาจมีตำแหน่งว่างอยู่ ดังนั้นจึงต้องมีบิตพิเศษเข้าไปเติมในพื้นที่ว่างนั้น ส่วนในการสลับลำดับข้อมูลกลับ ต้องมีการแยกบิตพิเศษออกจากข้อมูล ถ้าออกแบบผิดพลาด อาจทำให้บิตพิเศษที่เพิ่มเข้าไปนั้นปะปนเข้าไปอยู่กับบิตข้อมูลได้ ซึ่งจะทำให้เกิดความผิดพลาดของข้อมูลที่รับได้เพิ่มขึ้น

ส่วนการปรับปรุงต่อมา คือ Mahapakulchai and Sapisirisopon (2008a) ได้ทำการออกแบบการสลับลำดับข้อมูลแบบสุ่ม ซึ่งได้ประสิทธิภาพใกล้เคียงกัน แต่ลดปัญหาความยุ่งยากในการคำนวณขนาดของเมทริกซ์ที่นำมาใช้ในการสลับลำดับ โดยเริ่มต้นได้ใช้อัลกอริทึม Linear Congruence ในการสร้างตัวเลขแบบสุ่ม เพื่อนำไปสร้างเป็นเมทริกซ์ P แต่วิธีการนี้ใช้หน่วยความจำสูง และใช้เวลาในการคำนวณ จึงได้ปรับปรุงโดยการใช้ฟังก์ชันสร้างตัวเลขสุ่มของภาษาจาวาแทน โดยกำหนดค่าเริ่มต้นและจำนวนตัวเลขที่ต้องการสุ่มขึ้นมา เพื่อให้ได้จำนวนตัวเลขตามขนาดของแพ็กเก็ต แล้วนำมากำหนดลำดับการสลับลำดับแบบสุ่มให้กับแต่ละแพ็กเก็ต

เนื่องจากว่ากระบวนการสลับลำดับของ Mahapakulchai and Sapisirisopon (2008a) ได้สร้างตัวเลขแบบสุ่มขึ้นมาตามขนาดของแพ็กเก็ตที่จะทำการสลับลำดับ ซึ่งขนาดของแพ็กเก็ตมีขนาดที่แตกต่างกันไป จึงทำให้กระบวนการสร้างตัวเลขแบบสุ่มมีความยุ่งยากในการกำเนิดตัวเลขสุ่มที่มีขนาดแตกต่างกันไป ส่วนกระบวนการสลับลำดับข้อมูลยังมีการสลับลำดับข้อมูลที่เดียวทั้งแพ็กเก็ตอยู่ ซึ่งมีความซับซ้อนในการทำงานสูงเนื่องจากต้องมีการสร้างลำดับสุ่มขึ้นมาตามขนาดของแพ็กเก็ต ในขณะที่ขนาดของแพ็กเก็ตมีขนาดที่แตกต่างกันไป ซึ่งทำให้กระบวนการสลับลำดับข้อมูลมีความซับซ้อนสูง จึงทำให้ใช้ทรัพยากรของระบบสูงตามไปด้วย ส่งผลให้เวลาที่ใช้ในการสลับลำดับข้อมูลของแต่ละแพ็กเก็ตใช้เวลานานเกินไป

จากปัญหาต่างๆข้างต้น จะเห็นว่าปัญหาหลักๆ ของกระบวนการสลับลำดับข้อมูลมาจากความซับซ้อนของกระบวนการนั่นเอง เราจึงมีการพัฒนาการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้มาใช้แทน เพื่อลดความซับซ้อนของกระบวนการสลับลำดับข้อมูล ทำให้ระบบมีความ

ยุ่งยากซับซ้อนน้อยลง และทำให้เวลาโดยรวมของกระบวนการสลับลำดับข้อมูลลดลง ในขณะที่ระบบยังมีประสิทธิภาพที่ดีเหมือนเดิม

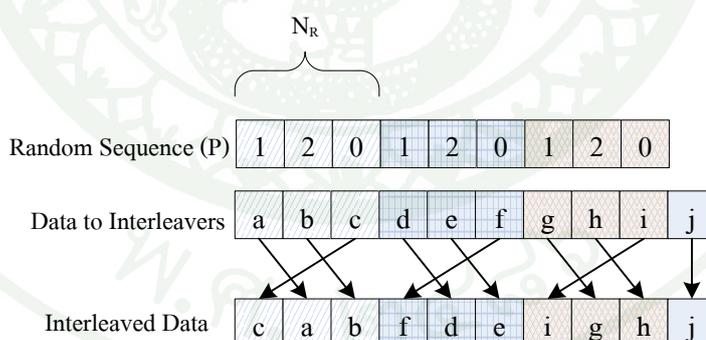
2.1 การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้

2.1.1 การสลับลำดับข้อมูล

การออกแบบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ แทนที่จะทำการสลับลำดับข้อมูลทั้งแพ็คเกจ เราจะทำการสลับลำดับข้อมูลตามขนาดที่ต้องการ โดยเราจะแทนขนาดที่จะทำการสลับลำดับด้วย N_R โดยมีขั้นตอนในการสลับลำดับข้อมูลดังนี้

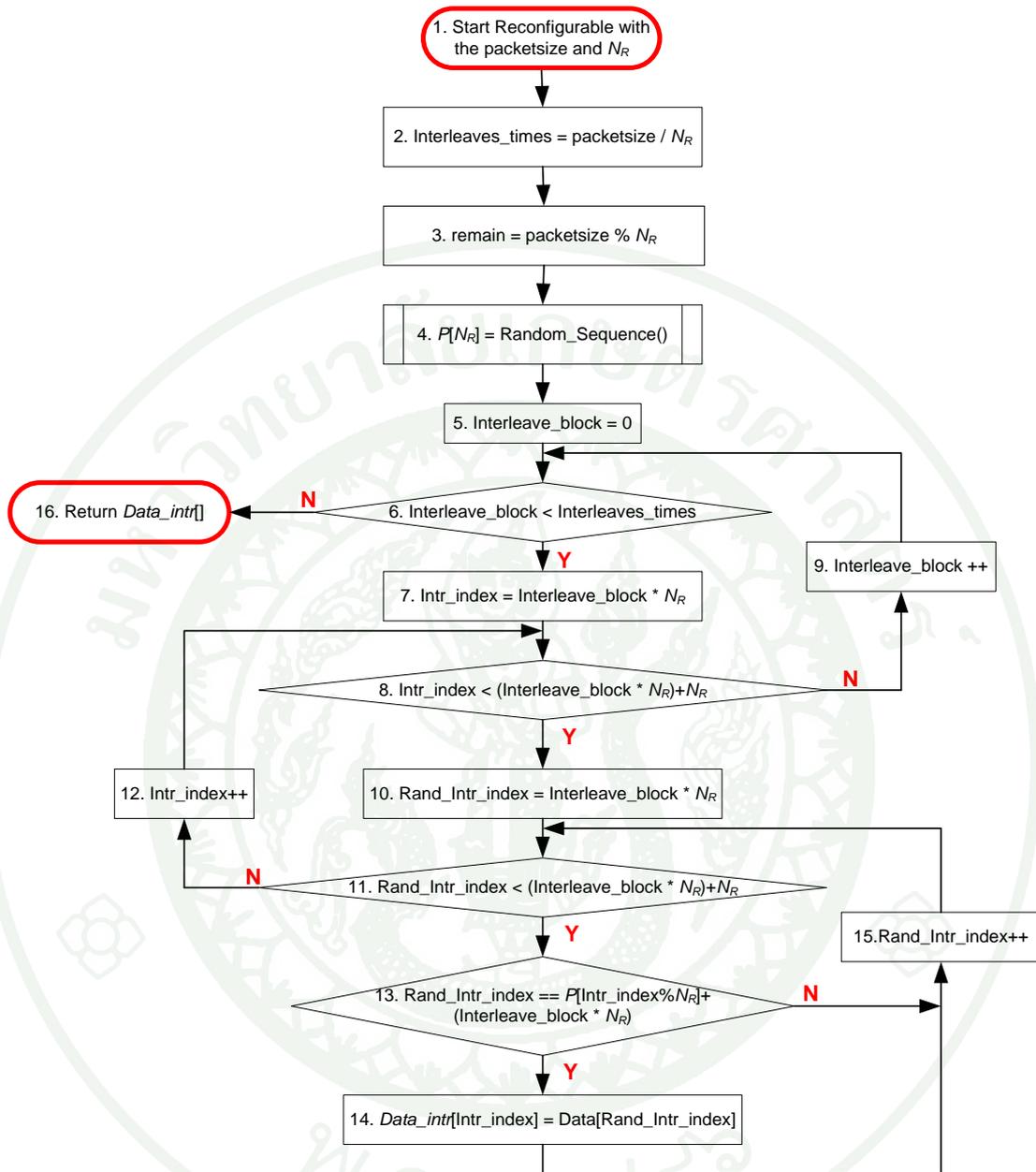
1. กำหนดลำดับแบบสุ่มจำนวน N_R ตัว
2. แบ่งเวกเตอร์ข้อมูล d ให้มีขนาดเท่ากับ N_R ออกเป็นหลายๆชุด
3. นำตัวเลขสุ่มที่กำหนดขึ้นมาเป็นตัวกำหนดลำดับของข้อมูลที่จะสลับ

โดยแสดงการทำงานตามภาพที่ 18



ภาพที่ 18 ตัวอย่างการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้

สามารถอธิบายการทำงาน โดยละเอียดตามผังงานจากภาพที่ 19



ภาพที่ 19 ผังงานการทำงานของกรสลบลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้

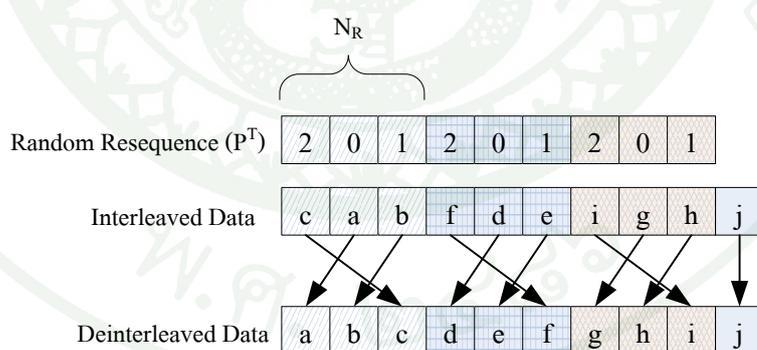
จากผังงาน เริ่มต้นการทำงานที่หมายเลข 1 จะมีการรับค่าขนาดของแพ็กเก็ตและจำนวนข้อมูลที่จะสลบลำดับ (N_R) เข้ามา จากนั้นลำดับที่ 2 จะหาจำนวนครั้งที่จะสลบลำดับข้อมูลของแพ็กเก็ตนั้น โดยคำนวณจากการหารขนาดของแพ็กเก็ตที่รับมาด้วย N_R แบบไม่คิดเศษ จากนั้นลำดับที่ 3 จะหาเศษที่เหลือจากการสลบลำดับข้อมูล โดยการคำนวณจากการหารขนาดของแพ็กเก็ตที่รับมาด้วย N_R แบบคิดแต่เศษ จากนั้นลำดับที่ 4 จะเรียกใช้ฟังก์ชันย่อยในการกำเนิดลำดับแบบสุ่ม

ขึ้นมาจำนวน N_R ตัว เก็บไว้ในอาร์เรย์ P จากนั้นลำดับที่ 5 ถึง 15 จะเริ่มทำการสลับลำดับข้อมูลที่ละชุดจนครบตามจำนวนครั้งที่คำนวณมาได้จากลำดับที่ 2 โดยที่ในการสลับลำดับข้อมูล จะใช้ลำดับที่ถูกสุ่มไว้ในอาร์เรย์ P มาเป็นตัวกำหนดตำแหน่งให้กับข้อมูลที่จะสลับออกไป (ลำดับที่ 13 และ 14) เก็บไว้ในอาร์เรย์ $Data_intr$ พอครบจำนวนครั้งที่กำหนด จะนำเศษที่เหลือที่ไม่ผ่านการสลับลำดับมาต่อไว้โดยตรงโดยไม่มีการสลับลำดับ และคืนค่าของ $Data_intr$ ออกไปให้กับโปรแกรมที่เรียกใช้

2.1.2 การสลับลำดับข้อมูลกลับ

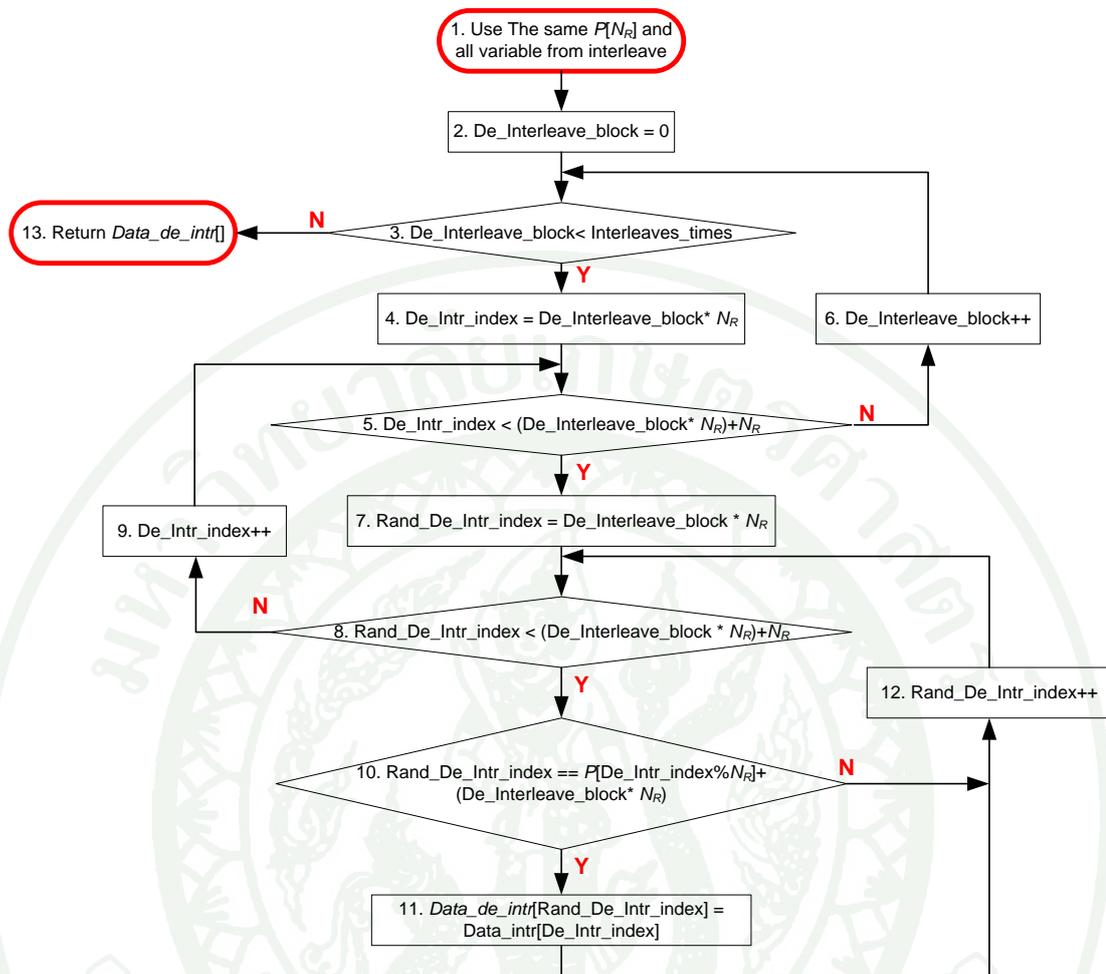
การสลับลำดับข้อมูลกลับที่ฝั่งรับจะกระทำกับข้อมูลที่สลับลำดับข้อมูลมาแล้ว ด้วยขนาดของ N_R ที่เท่ากับฝั่งส่งและใช้ชุดของลำดับสุ่มชุดเดียวกันจากฝั่งส่ง เพื่อนำไปใช้ในการกำหนดลำดับในการสลับข้อมูลกลับ ตามสมการ $d = d_{INT} P^T$ โดยมีขั้นตอนในการสลับลำดับข้อมูลตามนี้ และแสดงการทำงานตามภาพที่ 20

1. รับค่าชุดลำดับสุ่ม ค่า N_R และเวกเตอร์ d_{INT} จากฝั่งส่ง
2. แบ่งเวกเตอร์ข้อมูล d_{INT} ให้มีขนาดเท่ากับ N_R ออกเป็นหลายๆชุด
3. นำตัวเลขสุ่มที่รับมาจากฝั่งส่ง มาตัวกำหนดการสลับลำดับข้อมูลกลับ



ภาพที่ 20 ตัวอย่างการสลับลำดับข้อมูลกลับ

สามารถอธิบายการทำงานโดยละเอียดตามผังงานจากภาพที่ 21



ภาพที่ 21 ฟังก์ชันการทำงานของกระบวนการสลับลำดับข้อมูลกลับแบบสุ่มที่กำหนดขนาดได้

จากฟังก์ชันเริ่มต้นการทำงานที่หมายเลข 1 จะมีการรับค่าอาร์เรย์ของข้อมูลที่ได้รับได้หลังจากผ่านการสลับลำดับข้อมูลมาแล้วจากฟังก์ชัน จำนวนข้อมูลที่จะสลับลำดับ (N_R) และตัวเลขลำดับแบบสุ่ม เข้ามา แล้วหาจำนวนครั้งที่สลับลำดับข้อมูลของแพ็คเกจที่นั้นและเศษที่เหลือที่ไม่ต้องสลับข้อมูลกลับ จากนั้นลำดับที่ 2 เป็นส่วนของการกำหนดค่าเริ่มต้นของการทำซ้ำให้เริ่มการสลับข้อมูลกลับที่ข้อมูลชุดแรกของแพ็คเกจ ลำดับที่ 3 จะเป็นเงื่อนไขในการตรวจสอบว่าได้ทำการสลับลำดับจนครบแล้วหรือยัง ถ้ายังจะไปยังลำดับที่ 4 ซึ่งเป็นส่วนกำหนดค่าเริ่มต้นของการสลับลำดับกลับของข้อมูลแต่ละชุด จากนั้นลำดับที่ 5 จะตรวจสอบว่าทำการสลับลำดับกลับครบตามจำนวนที่กำหนดไว้ตาม N_R แล้ว ในแต่ละรอบของการทำซ้ำจะทำขั้นตอนลำดับที่ 7, 8, 10, 11, 12 เพื่อตรวจสอบกับลำดับแบบสุ่มที่เก็บไว้ในอาร์เรย์ P ว่าตำแหน่งใดถูกสลับมา แล้วสลับกลับไปตามเดิมในลำดับที่ 11 เมื่อครบตามจำนวนครั้งที่สลับกลับแล้วข้อมูลที่ถูกสลับกลับจะถูกเก็บไว้ที่

อาร์เรย์ $Data_de_intr$ แล้วนำเศษมาต่อไว้โดยตรงโดยไม่มีการสลับลำดับ และคืนค่าของ $Data_de_intr$ ออกไปให้กับโปรแกรมที่เรียกใช้

3. การจำลองช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง

การจำลองช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง โดยการกำหนดให้ค่าจาก α ที่ได้มา 1 ค่า มีผลต่อสัญลักษณ์ในช่วงของ N_s สัญลักษณ์ โดยที่ $N_s > 1$ โดยในการทดสอบได้กำหนดให้ $N_s = 5$ และ 8 จึงทำให้ช่องสัญญาณจากเดิมเป็นแบบไม่มีหน่วยความจำ กลายเป็นช่องสัญญาณที่มีหน่วยความจำ สัญญาณที่ได้รับจากช่องสัญญาณนี้ ส่งผลให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการแก้ไขข้อผิดพลาดลดลง โดยที่ $\alpha = 1 + \beta$ คือค่าที่ทำให้สัญญาณที่รับได้เกิดการลดทอนของสัญญาณ

4. การทดสอบระบบการรับส่งข้อมูลภาพ

เพื่อให้เป็นไปตามวัตถุประสงค์ของงานวิจัยข้างต้นที่พยายามจะลดความซับซ้อนของการสลับลำดับข้อมูล ดังที่กล่าวมาแล้ว โดยยังคงรักษาคุณภาพของภาพที่รับได้ เราจึงทดสอบการทำงานการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ให้ใช้ขนาดของ N_R ต่างกันหลายๆค่า และการสลับลำดับข้อมูลแบบสุ่มทั้งแพ็คเกจ เปรียบเทียบกันในเรื่องของประสิทธิภาพ และเวลาที่ใช้ในการประมวลผล ในการทดลองนั้นเราใช้รูปภาพของ Lena จำนวน 200 รูป ซึ่งเป็นรูปที่นิยมใช้กันในการวัดประสิทธิภาพของการรับส่งรูปภาพ โดยรูปนี้เป็นรูปภาพแบบเกรย์สเกลที่มีขนาด 512×512 จุด รูปภาพถูกบีบอัดด้วยอัลกอริทึม EZW โดย EZW สำหรับงานวิจัยนี้จะใช้การแบ่งแยกข้อมูลเป็น 5 ระดับ ซึ่งได้ผลลัพธ์เป็น 16 ย่านความถี่ย่อย โดยเป็นย่านความถี่ย่อยต่ำสุดคือ LFS จำนวนหนึ่งย่านความถี่ย่อย และที่เหลืออีก 15 ย่านความถี่ย่อยเป็นย่านความถี่ย่อยสูงกว่าคือ HFS โดยค่าสัมประสิทธิ์ของ HFS ทั้งหมดจะถูกควอนไทซ์ไปเป็นกระแสบิตแล้วแบ่งออกเป็นแพ็คเกจที่มีขนาดต่างๆกันไป สำหรับรูปภาพที่เราใช้นี้จะได้แพ็คเกจที่เป็น HFS จำนวน 101 แพ็คเกจ ที่มีขนาดโดยเฉลี่ยอยู่ที่ประมาณ 325 สัญลักษณ์ สำหรับช่องสัญญาณแบบไร้สัญญาณรบกวน เราได้ค่า PSNR ของรูปภาพที่สร้างกลับมาเป็น 31.83 dB และอัตราส่วนของการบีบอัดเป็น 31:1 สำหรับการเข้ารหัสช่องสัญญาณของระบบ เราเลือกใช้การเข้ารหัสช่องสัญญาณแบบ โพลีโนเมียลริงคอนโวลูชันนอลแบบ 32 ชั้น พร้อมกับการเข้ารหัสอัตราเท่ากับ 1/2 ที่แสดงไว้ในเมทริกซ์ขนาด 1×2 เมทริกซ์แรกในสมการที่ (21)

ในส่วนของการสลับลำดับข้อมูล เราได้อิมพลิเมนต์ การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยการแทนค่าขนาดที่ใช้ในการสลับลำดับข้อมูลด้วย N_r เราทำการทดลองการทำงานโดยการเปลี่ยนแปลงค่าของ N_r ที่แตกต่างกัน คือ 32, 64, 128 และ 143 สัญลักษณ์ตามลำดับ แล้วทดสอบกับช่องสัญญาณแบบ Rician บล็อกเฟคดิง ที่มีอัตราส่วนของระดับสัญญาณระหว่างเส้นทางตรงและเส้นทางการกระจาย คือ γ ที่มีค่าเท่ากับ -43 dB, -33 dB, และ -23 dB ตามลำดับ แต่ถ้า γ มีค่าเข้าสู่ $-\infty$ dB จะทำให้ช่องสัญญาณเปลี่ยนไปเป็น Rayleigh ในการทดสอบเราใช้รูปภาพ Lena จำนวน 200 รูป และทดสอบในช่วงของอัตราส่วนระหว่างสัญญาณกับสัญญาณรบกวนเฉลี่ย γ_b ภายในช่วงที่สนใจ คือ 3.75, 5.00, และ 6.25 dB



ภาพที่ 22 รูปภาพ Lena ต้นแบบ

5. การวิเคราะห์ประสิทธิภาพของระบบการรับส่งข้อมูลภาพ

5.1 การคำนวณค่า Peak Signal to Noise Ratio (PSNR)

PSNR คือ ค่าสูงสุดของอัตราส่วนระหว่างสัญญาณข้อมูลที่ส่งและสัญญาณรบกวน ซึ่งจะใช้เป็นค่าที่บ่งบอกประสิทธิภาพของระบบการรับส่งข้อมูลภาพ ระบบที่ดีต้องมีค่า PSNR ใกล้เคียงกับภาพต้นแบบมากที่สุด การคำนวณค่า PSNR สามารถคำนวณจากสมการที่ (23)

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (22)$$

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (23)$$

MAX_I คือ ค่าจุดภาพสูงสุดของรูปภาพ คือ $2^B - 1$ ซึ่ง B คือจำนวนของ Bits/Sample และ $\|I(i, j) - K(i, j)\|^2$ คือ ค่าความผิดพลาดของผลต่างระหว่างภาพต้นแบบและภาพที่รับได้

5.2 การคำนวณค่า Word Error Rate (WER)

WER คือ อัตราส่วนระหว่างจำนวนการผิดพลาดของสัญลักษณ์ที่รับได้และจำนวนสัญลักษณ์ทั้งหมดที่ส่ง ระบบที่ดีต้องมีค่า WER ต่ำหรือเข้าใกล้ศูนย์ เราสามารถหาค่า WER ได้จากสมการที่ (24)

$$WER = \text{จำนวนสัญลักษณ์ที่ผิดพลาด} \div \text{จำนวนสัญลักษณ์ทั้งหมดที่ส่ง} \quad (24)$$

6. สถานที่ทำการทดลองและระยะเวลาในการทดลอง

การออกแบบและการทดลองงานวิจัยนี้ได้ทำที่ ห้องปฏิบัติการ Cubic Space ชั้น 5 ห้อง 505/5 อาคาร 60 ปี คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ โดยใช้เวลาในการวิจัยออกแบบ และทดลองผล รวมกันทั้งสิ้น 12 เดือน

ผลและวิจารณ์

ผล

การทดสอบประสิทธิภาพของระบบ ทำโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้เข้าไปในระบบส่งข้อมูลภาพ โดยใช้รูปภาพของ Lena ซึ่งเป็นรูปภาพแบบเกรย์สเกลที่มีขนาด 512×512 จุด จำนวน 200 รูป รูปภาพนี้ถูกบีบอัดด้วยอัลกอริทึม EZW โดยใช้การแบ่งแยกข้อมูลเป็น 5 ระดับ ซึ่งได้ผลลัพธ์เป็น 16 ย่านความถี่ย่อย โดยเป็นย่านความถี่ย่อยต่ำสุดคือ LFS จำนวนหนึ่งย่านความถี่ย่อย และที่เหลืออีก 15 ย่านความถี่ย่อยเป็นย่านความถี่ย่อยสูงกว่าคือ HFS จากนั้นค่าสัมประสิทธิ์ของ HFS ทั้งหมดจะถูกควอนไทซ์ไปเป็นกระแสบิตแล้วแบ่งออกเป็นแพ็คเกจที่มีขนาดต่างๆกันไป สำหรับรูปภาพ Lena นี้จะได้แพ็คเกจที่ได้จาก HFS จำนวน 101 แพ็คเกจ ที่มีขนาดโดยเฉลี่ยอยู่ที่ประมาณ 325 สัญลักษณ์ ใช้การเข้ารหัสช่องสัญญาณแบบโพลีโนเมียลริงคอนโวลูชันนอลแบบ 32 ชั้น พร้อมกับการเข้ารหัสอัตราเท่ากับ $1/2$ ที่แสดงไว้ในเมทริกซ์ขนาด 1×2 เมทริกซ์แรกในสมการที่ (21) จากนั้นทำการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยการแทนค่าขนาดที่ใช้ในการสลับลำดับข้อมูลด้วย N_k เราทำการทดลองการทำงานโดยการเปลี่ยนแปลงค่าของ N_k ที่แตกต่างกัน คือ 32, 64, 128 และ 143 สัญลักษณ์ตามลำดับ ในการทดสอบเราได้ทำ 2 แบบ คือ แบบแรกทำการสลับลำดับเฉพาะแพ็คเกจเกิดของ HFS ส่วนแบบที่สองทำการสลับลำดับทั้ง LFS และ HFS จากนั้นทำการมอดูเลตแล้วส่งผ่านไปยังช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง ทางด้านรับ จะทำการดีมอดูเลต แล้วสลับลำดับข้อมูลกลับ แล้วคำนวณเส้นทางบนเทลลิส และการถอดรหัสวีเทอร์บี ปัจจัยที่ใช้วิเคราะห์ประสิทธิภาพจากผลการทดลองประกอบไปด้วย ค่าเฉลี่ย PSNR, WER, และเวลาที่ใช้ในการประมวลผล ในการทดสอบกับช่องสัญญาณแบบไร้สัญญาณรบกวน เราได้ค่า PSNR ของรูปภาพที่สร้างกลับมาเป็น 31.83 dB และอัตราส่วนของการบีบอัดเป็น 31:1

เราได้แสดงผลจากการทดสอบกับช่องสัญญาณแบบ Rician บล็อกเฟดดิ้ง ที่มีอัตราส่วนของระดับสัญญาณระหว่างเส้นทางตรงและเส้นทางกระจาย คือ γ ที่มีค่าเท่ากับ -43 dB, -33 dB, และ -23 dB ตามลำดับ ช่วงของอัตราส่วนระหว่างสัญญาณกับสัญญาณรบกวนเฉลี่ย $\bar{\gamma}_b$ ภายในช่วงที่สนใจ คือ 3.75, 5.00, และ 6.25 dB เนื่องจากช่วงดังกล่าวเราได้ค่าของ PSNR เฉลี่ยของภาพที่รับได้เมื่อช่องสัญญาณ ถูกกระทบด้วย Rician บล็อกเฟดดิ้ง ออกมาในช่วง 8 dB – 13 dB ซึ่งถือว่าเป็นค่าที่แย่ เพื่อนำไปเปรียบเทียบกับภาพที่รับได้เมื่อผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาด

ได้ โดยใช้การถอดรหัสแบบ MAP ซึ่งสนใจในงานวิจัยนี้เทียบกับ ML โดยใช้ค่า $N_s = 5$ และ มีการทดสอบการถอดรหัสแบบ MAP โดยใช้ $N_s = 5$ เทียบกับ $N_s = 8$ เพื่อทดสอบกับช่องสัญญาณที่มีผลกระทบกับบล็อกเฟดดิ้งที่ต่างกัน จากนั้นเป็นผลจากการทดสอบการคงค่า PSNR ที่เราต้องการ แล้วปรับค่า N_R เพื่อให้ได้ PSNR ที่กำหนดไว้สำหรับสภาพของช่องสัญญาณต่างๆกัน เพื่อเป็นแนวทางในการปรับปรุงระบบต่อไป โดยผลจากการทดลองส่วนนี้จะไม่สลับลำดับเศษที่เหลือจากการแบ่งตามขนาด N_R และเราได้ทดสอบในกรณีที่มีการสลับลำดับเศษที่เหลือด้วยเพื่อเทียบกราฟที่ได้ว่ามีแนวโน้มที่ดีขึ้นอย่างไรบ้าง ส่วนสุดท้ายจะเปรียบเทียบเวลาโดยเฉลี่ยที่ใช้ในการประมวลผลการสลับลำดับข้อมูล และการสลับลำดับข้อมูลกลับสำหรับแพ็กเก็ตยาวที่สุดคือขนาด 878 สัญลักษณ์ และแพ็กเก็ตสั้นที่สุดคือขนาด 237 สัญลักษณ์ เพื่อแสดงให้เห็นว่าขนาดของแพ็กเก็ต มีผลอย่างมากต่อเวลาที่ใช้ในการสลับลำดับข้อมูล และเราได้เปรียบเทียบกับ การสลับลำดับข้อมูลแบบสุ่มที่สลับลำดับข้อมูลทั้งแพ็กเก็ต ซึ่งแสดงให้เห็นว่างานวิจัยนี้สามารถลดเวลาที่ใช้ในการประมวลผลลงได้ประมาณ 3 เท่าในกรณีของแพ็กเก็ตขนาดใหญ่ ในขณะที่คุณภาพของรูปภาพที่สร้างกลับมายังคงคืออยู่คือ ห่างกันประมาณ 3 dB เท่านั้น ซึ่งแทบมองไม่เห็นความแตกต่างจากรูปที่สร้างกลับมา โดยแสดงผลจากการทดลองตามหัวข้อต่อไปนี้

1. การทดสอบโดยไม่ผ่านการสลับลำดับข้อมูล ที่มี $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML

ผลลัพธ์ตารางที่ 2 เป็นการทดสอบเพื่อเป็นข้อมูลอ้างอิงค่าเฉลี่ย PSNR และ WER ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกัน สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน โดยไม่มีการสลับลำดับข้อมูลเลย และแสดงตัวอย่างรูปภาพที่ได้จากการทดสอบไว้ภาพที่ 29 (ก)

ตารางที่ 2 ผลจากการทดสอบโดยไม่ผ่านการสลับลำดับข้อมูลโดยการใช้การถอดรหัสแบบ MAP และ ML

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.746 0.046	9.855 0.031	11.086 0.021	8.795 0.048	9.647 0.032	10.783 0.022
-33	9.053 0.041	10.272 0.027	11.548 0.018	8.999 0.043	10.014 0.028	11.258 0.018
-23	10.097 0.028	11.670 0.017	13.328 0.010	9.958 0.029	11.388 0.018	12.988 0.011

จากตารางจะเห็นว่า ค่า PSNR และ WER เป็นไปตามคุณภาพของช่องสัญญาณ ซึ่งการถอดรหัสแบบ MAP จะมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย แต่ทั้งคู่ยังมีประสิทธิภาพที่ต่ำอยู่เนื่องจากมีความผิดพลาดที่ต่อเนื่องที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวิเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง

2. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ และ $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 3 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_S = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 3 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด

$N_r = 32$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.840 0.020	10.121 0.010	11.662 0.005	8.886 0.021	9.883 0.011	11.272 0.005
-33	9.177 0.017	10.619 0.008	12.230 0.004	9.124 0.018	10.307 0.009	11.896 0.004
-23	10.413 0.010	12.376 0.004	14.546 0.002	10.203 0.010	12.035 0.004	14.195 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 เล็กน้อยเนื่องจากข้อมูลของแพ็กเก็ตเกิด LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ตเกิด LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตามการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

3. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 32$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 4 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 32$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 4 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 32$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	12.355 0.019	15.287 0.009	18.741 0.004	12.456 0.020	15.482 0.010	18.889 0.005
-33	12.976 0.016	16.440 0.008	19.630 0.004	13.132 0.017	16.453 0.008	20.017 0.004
-23	15.694 0.009	19.332 0.004	22.816 0.002	15.768 0.010	19.877 0.004	23.231 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 มากกว่าตารางที่ 3 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอริมีประสิทธิภาพที่ดีขึ้น

4. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 64$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 5 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 64$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 5 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 64$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.906 0.013	10.223 0.006	11.779 0.003	8.951 0.014	9.982 0.007	11.399 0.003
-33	9.257 0.011	10.731 0.005	12.352 0.003	9.198 0.011	10.413 0.005	12.030 0.003
-23	10.517 0.006	12.502 0.003	14.675 0.001	10.318 0.006	12.169 0.003	14.388 0.001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 และตารางที่ 3 เล็กน้อย เนื่องจากข้อมูลของแพ็กเก็ตเกิด LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ตเกิด LFS ที่มีผลมาจาก บล็อกเฟดคิง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตามการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

5. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 64$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 5 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 64$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดคิงที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 6 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็คเกจเกิด LFS และ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	14.248 0.012	18.295 0.006	21.612 0.003	14.062 0.013	17.951 0.006	21.265 0.003
-33	15.013 0.010	19.149 0.005	22.683 0.002	15.031 0.011	18.784 0.005	22.311 0.002
-23	18.092 0.005	22.238 0.002	25.403 0.001	17.989 0.006	21.892 0.002	24.805 0.001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 มากกว่าตารางที่ 3 และ 5 เนื่องจากทั้งข้อมูลของแพ็คเกจเกิด LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็คเกจเกิด LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น และการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

6. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ และ $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลเฉพาะแพ็คเกจเกิดของ HFS

ผลลัพธ์ตารางที่ 7 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดที่ใช้ขนาด $N_R = 128$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_S = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับแพ็คเกจเกิดของ HFS จำนวน 101 แพ็คเกจ

ตารางที่ 7 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด

$N_r = 128$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.919 0.011	10.215 0.006	11.745 0.004	8.963 0.012	9.982 0.006	11.372 0.004
-33	9.266 0.010	10.724 0.005	12.307 0.003	9.209 0.010	10.412 0.005	11.993 0.003
-23	10.516 0.006	12.464 0.003	14.570 0.002	10.316 0.006	12.142 0.003	14.284 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 และตารางที่ 3, 5 เล็กน้อย เนื่องจากข้อมูลของแพ็กเก็ตเกิด LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ตเกิด LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตามการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

7. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 128$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการใช้การสลับลำดับข้อมูลทั้งแพ็กเก็ตเกิด LFS และ HFS

ผลลัพธ์ตารางที่ 8 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 128$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับทั้งแพ็กเก็ตเกิดของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 8 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 128$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็คเกจเกิด LFS และ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	16.560 0.010	19.555 0.005	22.565 0.003	16.839 0.010	20.181 0.005	22.574 0.003
-33	17.399 0.009	20.549 0.004	23.229 0.003	17.617 0.009	20.985 0.004	23.301 0.003
-23	19.580 0.005	23.090 0.003	24.924 0.002	20.231 0.005	23.074 0.003	25.273 0.001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 มากกว่าตารางที่ 3, 5 และ 7 เนื่องจากทั้งข้อมูลของแพ็คเกจเกิด LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็คเกจเกิด LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น

8. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลเฉพาะแพ็คเกจเกิดของ HFS

ผลลัพธ์ตารางที่ 9 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับแพ็คเกจเกิดของ HFS จำนวน 101 แพ็คเกจเกิด

ตารางที่ 9 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด

$N_r = 143$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.931 0.010	10.246 0.005	11.793 0.003	8.976 0.010	10.009 0.005	11.405 0.003
-33	9.289 0.009	10.761 0.005	12.357 0.003	9.225 0.009	10.440 0.005	12.028 0.003
-23	10.548 0.005	12.516 0.003	14.641 0.002	10.339 0.005	12.179 0.003	14.357 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 และตารางที่ 3, 5, 7 เล็กน้อย เนื่องจากข้อมูลของแพ็กเก็ตเกิด LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ตเกิด LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตามการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

9. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 10 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 10 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด

$N_R = 143$ โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต

LFS และ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	16.824 0.009	20.388 0.005	22.649 0.003	16.471 0.009	20.411 0.005	22.910 0.003
-33	17.763 0.007	21.244 0.004	23.295 0.002	17.535 0.008	21.039 0.004	23.663 0.002
-23	20.541 0.004	23.120 0.002	25.254 0.001	20.267 0.005	23.247 0.002	25.634 0.001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 มากกว่าตารางที่ 3, 5, 7 และ 9 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น

10. การทดสอบการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) ที่มี $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 11 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มทั้งแพ็กเก็ต ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 11 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	9.073 0.004	10.434 0.002	12.037 0.001	9.100 0.004	10.176 0.002	11.618 0.001
-33	9.441 0.003	10.963 0.001	12.619 0.001	9.355 0.003	10.618 0.001	12.282 0.001
-23	10.741 0.002	12.819 0.001	15.097 0.0004	10.509 0.002	12.450 0.001	14.783 0.0004

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 และตารางที่ 3, 5, 7, 9 เล็กน้อยเนื่องจากข้อมูลของแพ็กเก็ต LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS ที่มีผลมาจาก บล็อกเฟดคิง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตามการถอดรหัสแบบ MAP ยังคงมีประสิทธิภาพของการถอดรหัสได้ดีกว่า ML เล็กน้อย

11. การทดสอบการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) ที่มี $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และ ML โดยการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 12 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มทั้งแพ็กเก็ต ของการถอดรหัสแบบ MAP และ ML เปรียบเทียบกันสำหรับช่องสัญญาณแบบบล็อกเฟดคิงที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ด้านบน โดยใช้ $N_s = 5$ ซึ่งทำให้สัญลักษณ์ที่ติดกัน 5 สัญลักษณ์กระทบกับค่าจาก α ค่าเดียวกัน และสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 12 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (สลับทั้งแพ็กเก็ต) โดยการใช้การถอดรหัสแบบ MAP และ ML และสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

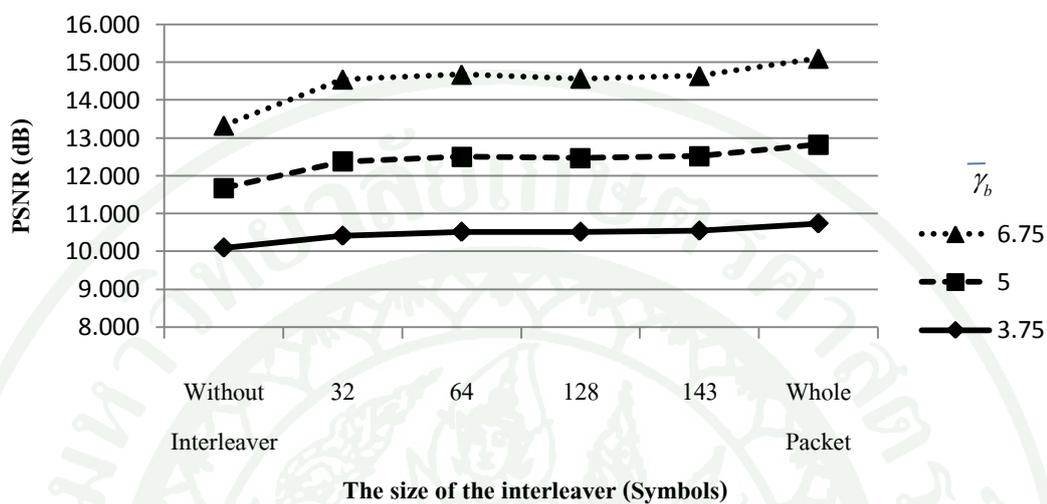
γ (dB)	MAP			ML		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	20.815 0.003	25.231 0.001	28.281 0.0003	20.494 0.003	25.426 0.001	28.621 0.0003
-33	21.721 0.002	26.033 0.001	29.002 0.0002	21.439 0.002	26.338 0.0006	29.136 0.0002
-23	24.684 0.001	28.231 0.0003	30.200 0.0001	24.646 0.001	28.518 0.0003	30.331 0.0001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 2 มากกว่าตารางที่ 3, 5, 7, 9 และ 11 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น

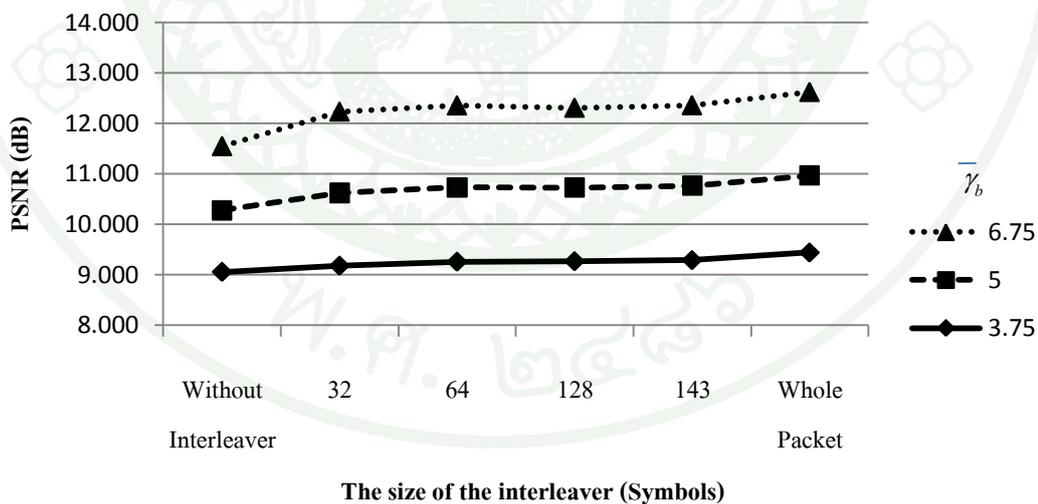
12. กราฟแสดงผลการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่มีค่า N_R ที่แปรผันไปสำหรับช่องสัญญาณเฟดดิ้ง $\gamma = -23$ dB, -33 dB, -43 dB ตามลำดับ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และมีการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

จากตารางด้านบน (ตารางที่ 3, 5, 7, 9, 11) เมื่อเรานำเอาค่า PSNR ของการถอดรหัสแบบ MAP ในกรณีที่มีการสลับลำดับข้อมูลเฉพาะข้อมูลของแพ็กเก็ต HFS มาพล็อตเป็นกราฟตามขนาดของ N_R โดยแยกเป็นกราฟของ γ จาก -23 dB, -33 dB และ -43 dB ตามลำดับ ซึ่งจากทั้ง 3 กราฟจากภาพที่ 23 – 25 แสดงให้เห็นว่า แนวโน้มของค่า PSNR เพิ่มขึ้นไปตามขนาดของ N_R ทั้งหมด แต่อาจจะไม่ได้เพิ่มขึ้นอย่างเห็นได้ชัด เนื่องจากข้อมูลโครงร่างหลักของภาพอยู่ที่แพ็กเก็ต LFS เป็น

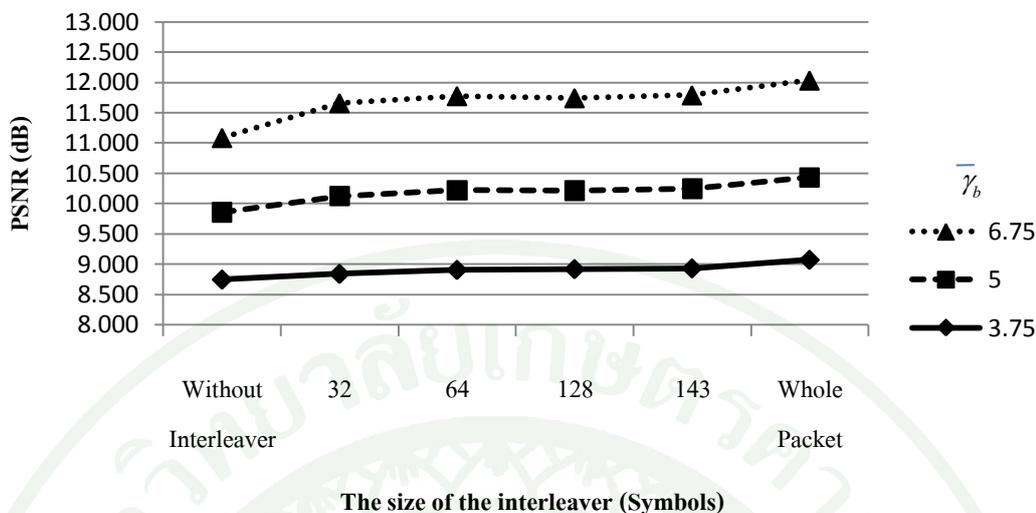
หลัก ซึ่งยังคงถูกผลกระทบจากความผิดพลาด อย่างต่อเนื่องเหมือนเดิมจึงทำให้ ค่า PSNR ดีขึ้น เล็กน้อยเท่านั้น



ภาพที่ 23 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB



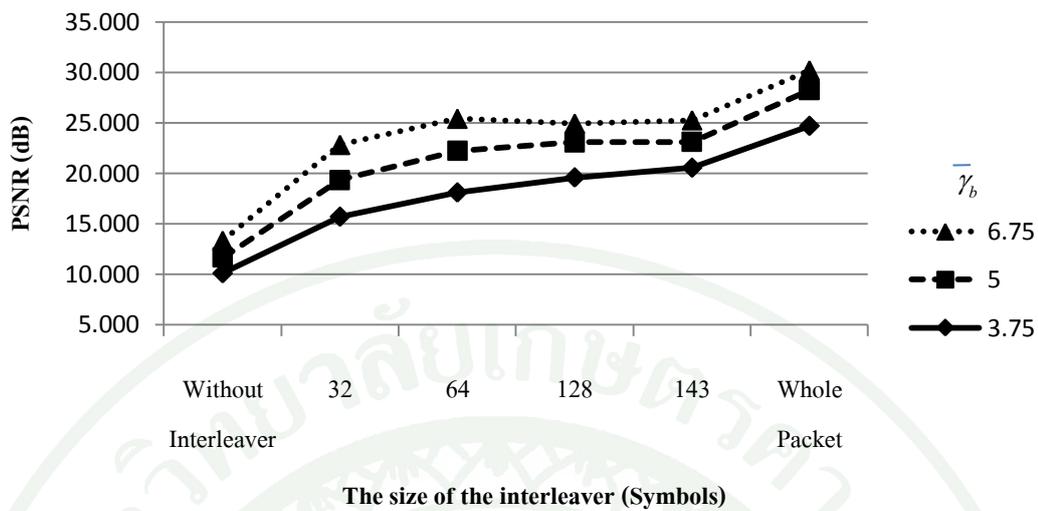
ภาพที่ 24 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB



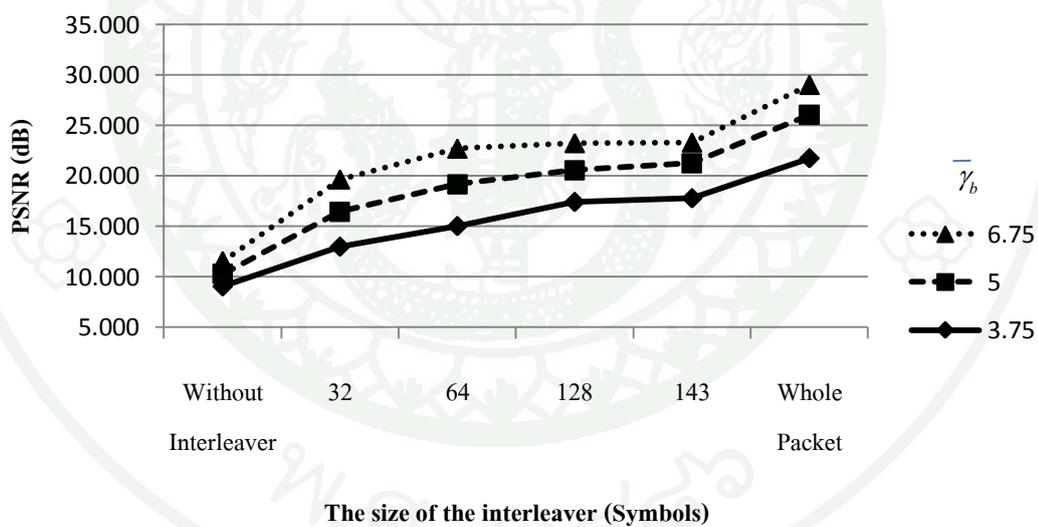
ภาพที่ 25 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB

13. กราฟแสดงผลการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่มีค่า N_R ที่แปรผันไปสำหรับช่องสัญญาณเฟดดิ้ง $\gamma = -23$ dB, -33 dB, -43 dB ตามลำดับ และ $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และมีการสลับลำดับข้อมูลทั้งแพ็คเกจ LFS และ HFS

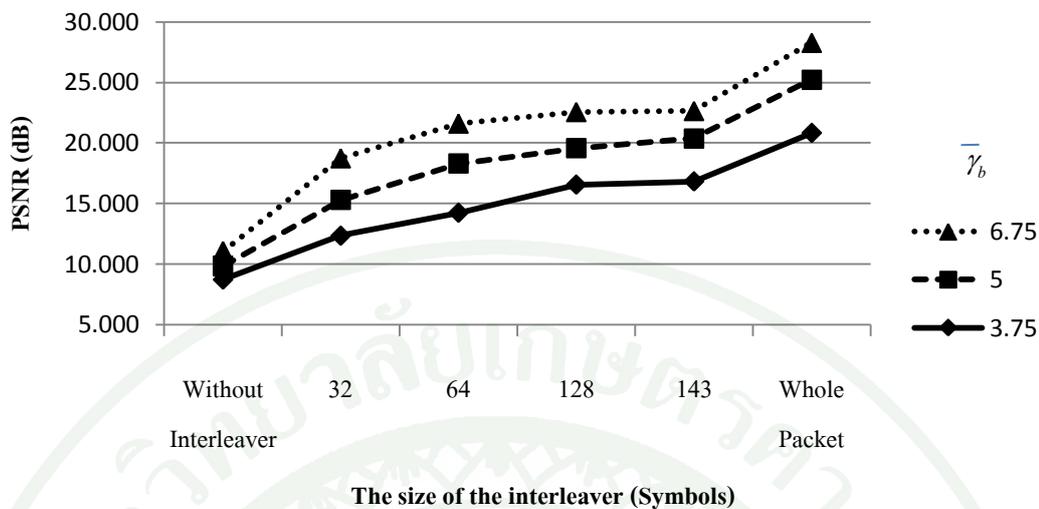
จากตารางด้านบน (ตารางที่ 4, 6, 8, 10, 12) เช่นกัน เมื่อเรานำเอาค่า PSNR ของการถอดรหัสแบบ MAP ในกรณีที่มีการสลับลำดับข้อมูลเฉพาะข้อมูลทั้งแพ็คเกจ LFS และ HFS มาพล็อตเป็นกราฟตามขนาดของ N_R โดยแยกเป็นกราฟของ γ จาก -23 dB, -33 dB และ -43 dB ตามลำดับ ซึ่งจากทั้ง 3 กราฟ จากภาพที่ 26-28 แสดงให้เห็นว่าแนวโน้มของค่า PSNR เพิ่มขึ้นไปตามขนาดของ N_R ทั้งหมดอย่างเห็นได้ชัด เนื่องจากข้อมูลโครงสร้างหลักของภาพที่อยู่ทั้งแพ็คเกจ LFS ถูกสลับลำดับข้อมูลไปด้วย จึงทำให้สามารถลดผลกระทบจากความผิดพลาดอย่างต่อเนื่องได้มาก ส่งผลให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้นอย่างชัดเจน จึงทำให้ค่า PSNR มีแนวโน้มดีขึ้นมาก ดังแสดงให้เห็นจากรูปภาพตัวอย่างที่สร้างกลับมาจากภาพที่ 29 (ข)



ภาพที่ 26 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB



ภาพที่ 27 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB



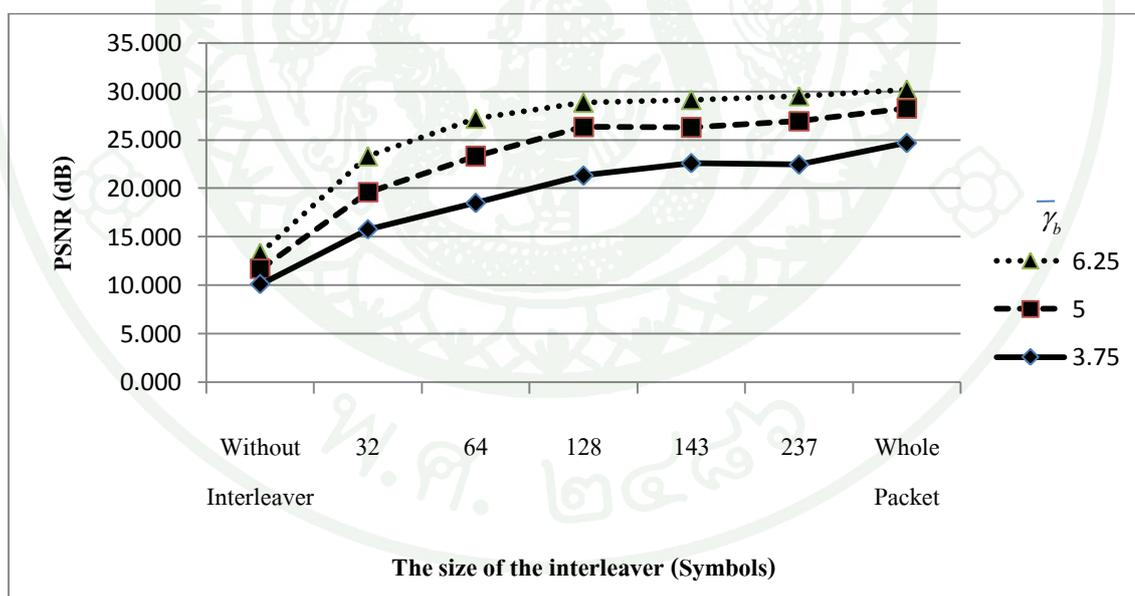
ภาพที่ 28 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB



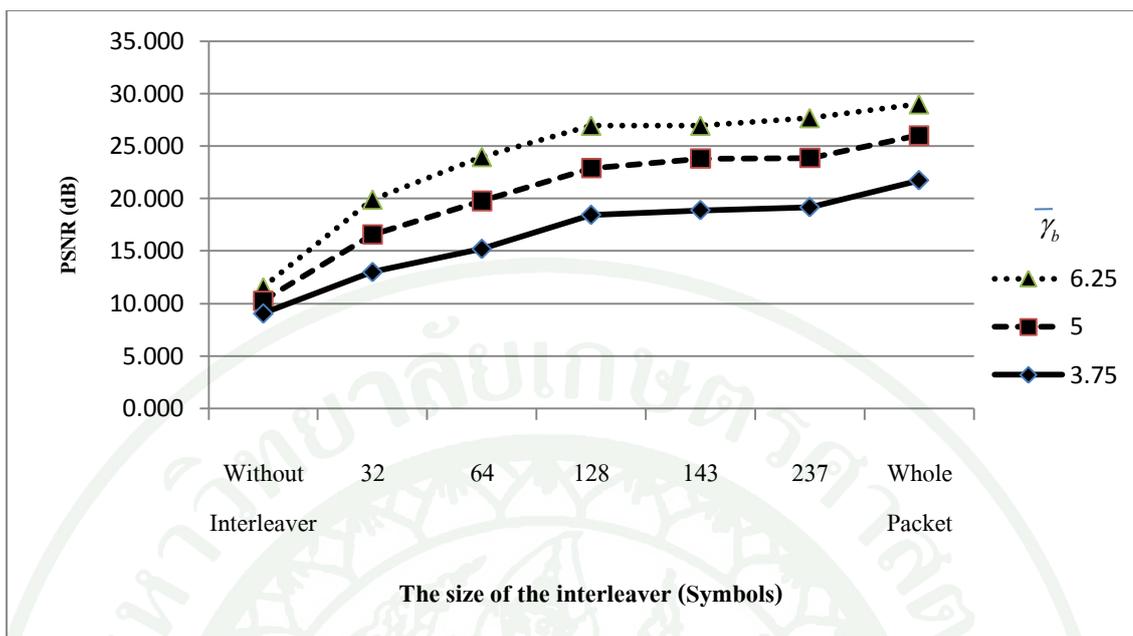
ภาพที่ 29 รูปภาพ Lena ที่สร้างกลับมาโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP ก) $\bar{\gamma}_b = 6.25$ dB โดยไม่ผ่านการสลับลำดับข้อมูล ได้ค่า PSNR = 13.172 dB, WER = 0.00967 ข) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 27.666 dB, WER = 0.00067

14. กราฟแสดงผลการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่มีค่า N_R ที่แปรผันไปสำหรับช่องสัญญาณเฟดดิ้ง $\gamma = -23$ dB, -33 dB, -43 dB ตามลำดับ และ $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และมีการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS โดยสลับลำดับเศษที่เหลือด้วย

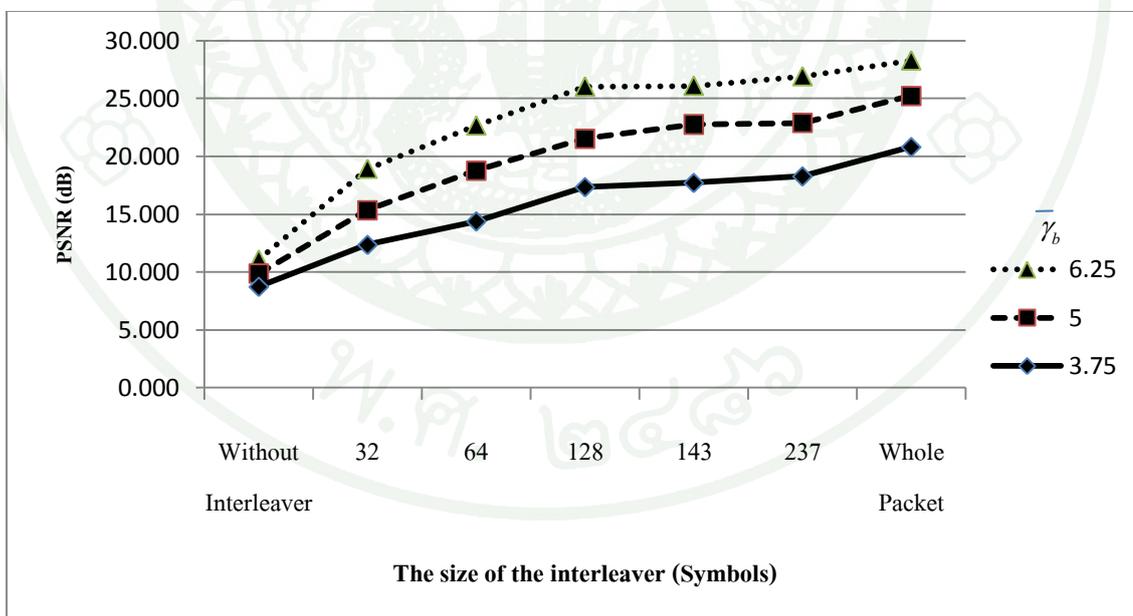
จากภาพที่ 26-28 แสดงให้เห็นว่าแนวโน้มของค่า PSNR เพิ่มขึ้นไปตามขนาดของ NR ทั้งหมดอย่างเห็นได้ชัด เนื่องจากข้อมูลโครงสร้างหลักของภาพที่อยู่แพ็กเก็ต LFS ถูกสลับลำดับข้อมูลไปด้วย จึงทำให้สามารถลดผลกระทบจากความผิดพลาดอย่างต่อเนื่องได้มาก แต่ค่า PSNR ของแต่ละช่วงอาจจะดีขึ้นอย่างต่อเนื่อง เนื่องจากการกำหนดค่า N_R มีผลต่อจำนวนเศษที่เหลือจากการสลับลำดับข้อมูล ถ้ามีเศษเหลือจำนวนมากๆ จะทำให้ค่า PSNR โดยรวมไม่ดีเท่าที่ควร เนื่องจากมีเศษถูกกระทบจากการเฟดดิ้งจำนวนมาก ดังนั้นเราจึงได้ปรับปรุงเพิ่มเติมโดยการสลับลำดับข้อมูลเศษที่เหลือไปด้วย โดยแสดงผลลัพธ์จากกราฟ ตามภาพที่ 30-32 ซึ่งจะเห็นว่าแนวโน้มของค่า PSNR ปรับปรุงดีขึ้นอย่างต่อเนื่อง



ภาพที่ 30 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -23$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย



ภาพที่ 31 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -33$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย



ภาพที่ 32 กราฟแสดงผลการทดสอบสำหรับช่องสัญญาณเฟดดิ้งที่มี $\gamma = -43$ dB โดยสลับลำดับข้อมูลเศษที่เหลือด้วย

15. การทดสอบโดยไม่ผ่านการสลับลำดับข้อมูล กับช่องสัญญาณบล็อกลักษณะเฟดดิ้งที่มี $N_s = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP

ผลลัพธ์ตารางที่ 13 เป็นการทดสอบเพื่อเป็นข้อมูลอ้างอิงค่าเฉลี่ย PSNR และ WER ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกลักษณะเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_s = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน โดยไม่มีการสลับลำดับข้อมูลเลย

ตารางที่ 13 ผลจากการทดสอบโดยไม่ผ่านการสลับลำดับข้อมูลกับช่องสัญญาณบล็อกลักษณะเฟดดิ้งที่มี $N_s = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP

γ (dB)	$N_s = 5$			$N_s = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.746 0.046	9.855 0.031	11.086 0.021	8.752 0.061	9.444 0.044	10.412 0.031
-33	9.053 0.041	10.272 0.027	11.548 0.018	8.981 0.056	9.753 0.039	10.761 0.028
-23	10.097 0.028	11.670 0.017	13.328 0.010	9.662 0.041	10.755 0.027	12.135 0.018

จากตารางจะเห็นว่า ค่า PSNR มีค่าลดลงและ WER มีค่าสูงขึ้น เมื่อสัญลักษณ์ที่ติดกันจำนวนมากว่าถูกกระทบจากค่าจาก α ค่าเดียวกัน ซึ่งทั้งสองกรณียังมีประสิทธิภาพที่ต่ำอยู่เนื่องจากมีความผิดพลาดที่ต่อเนื่องที่มีผลมาจาก บล็อกลักษณะเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง

16. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบลิคเฟดดิ้งที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 14 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบลิคเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน และสลับลำดับแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 14 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบลิคเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	$N_S = 5$ PSNR (dB) WER			$N_S = 8$ PSNR (dB) WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
	-43	8.840 0.020	10.121 0.010	11.662 0.005	8.793 0.038	9.544 0.024
-33	9.177 0.017	10.619 0.008	12.230 0.004	9.033 0.034	9.895 0.020	11.060 0.012
-23	10.413 0.010	12.376 0.004	14.546 0.002	9.791 0.022	11.039 0.012	12.732 0.007

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 เล็กน้อยเนื่องจากข้อมูลของแพ็กเก็ต LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS ที่มีผลมาจาก บลิคเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพ

ในการถอดรหัสลดลง แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแย่งเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

17. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 15 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดการได้ที่ใช้ขนาด $N_R = 32$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกันและสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 15 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 32$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	$N_S = 5$			$N_S = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	12.355 0.019	15.287 0.009	18.741 0.004	10.225 0.038	12.048 0.023	14.252 0.014
-33	12.976 0.016	16.440 0.008	19.630 0.004	10.687 0.033	12.636 0.020	14.787 0.012
-23	15.694 0.009	19.332 0.004	22.816 0.002	12.106 0.021	14.619 0.012	17.876 0.006

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 มากกว่าตารางที่ 14 เนื่องจาก ทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัส วิเทอรับมีประสิทธิภาพที่ดีขึ้น แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแย่งเมื่อสัญลักษณ์ที่ คิดกันจำนวนมากกว่าถูกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

18. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณ บล็อกเฟดดิ้งที่มี $N_s = 5$ และ 8 ตามลำดับ โดยการใช้ออครหัสแบบ MAP และการสลับลำดับ ข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 16 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ ขนาด $N_R = 64$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ γ_b ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_s = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่คิดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน และสลับลำดับแพ็กเก็ตของ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 16 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

γ (dB)	$N_S = 5$			$N_S = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.906 0.013	10.223 0.006	11.779 0.003	8.859 0.022	9.671 0.012	10.852 0.007
-33	9.257 0.011	10.731 0.005	12.352 0.003	9.115 0.019	10.044 0.010	11.289 0.005
-23	10.517 0.006	12.502 0.003	14.675 0.001	9.925 0.012	11.265 0.006	13.074 0.003

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13, 14 เล็กน้อยเนื่องจากข้อมูลของแพ็กเก็ต LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแย่งเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

19. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 17 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 64$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูก

กระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกันและสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 17 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 64$ กับช่องสัญญาณบดล็อกเฟดดิ้งที่มี $N_s = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	$N_s = 5$			$N_s = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	14.248 0.012	18.295 0.006	21.612 0.003	12.816 0.021	15.738 0.011	18.916 0.006
-33	15.013 0.010	19.149 0.005	22.683 0.002	13.356 0.018	16.424 0.009	20.011 0.005
-23	18.092 0.005	22.238 0.002	25.403 0.001	15.727 0.011	19.682 0.005	22.782 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 มากกว่าตารางที่ 14, 16 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บดล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแยกลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

20. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ กับช่องสัญญาณบดล็อกเฟดดิ้งที่มี $N_s = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 18 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 128$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_s = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน และสลับลำดับแพ็คเกจเกิดของ HFS จำนวน 101 แพ็คเกจ

ตารางที่ 18 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_r = 128$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_s = 5$ และ 8 โดยการใช้อัลกอริทึมการถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็คเกจเกิดของ HFS

γ (dB)	$N_s = 5$			$N_s = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.919 0.011	10.215 0.006	11.745 0.004	8.894 0.016	9.720 0.009	10.895 0.006
-33	9.266 0.010	10.724 0.005	12.307 0.003	9.159 0.014	10.099 0.008	11.318 0.005
-23	10.516 0.006	12.464 0.003	14.570 0.002	9.973 0.009	11.300 0.005	13.066 0.003

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13, 14, 15 เล็กน้อยเนื่องจากข้อมูลของแพ็คเกจเกิด LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็คเกจเกิด LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแยกลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

21. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ กับช่องสัญญาณบล็อกลอดจิงที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 19 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 128$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกลอดจิงที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกันและสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 19 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 128$ กับช่องสัญญาณบล็อกลอดจิงที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	$N_S = 5$			$N_S = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	16.560 0.010	19.555 0.005	22.565 0.003	15.439 0.015	18.783 0.008	21.413 0.005
-33	17.399 0.009	20.549 0.004	23.229 0.003	16.518 0.013	19.716 0.007	22.235 0.004
-23	19.580 0.005	23.090 0.003	24.924 0.002	18.529 0.008	21.767 0.004	24.081 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 มากกว่าตารางที่ 14, 15, 18 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกลอดจิงได้เพิ่มขึ้น จึงทำให้การ

ถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแยกลงเมื่อ สัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกระทบจากค่า α ค่าเดียวกันเล็กน้อย

22. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ กับ ช่องสัญญาณบล็อกลอดถึงที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็คเกจของ HFS

ผลลัพธ์ตารางที่ 20 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกลอดถึงที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน และสลับลำดับแพ็คเกจของ HFS จำนวน 101 แพ็คเกจ

ตารางที่ 20 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ กับช่องสัญญาณบล็อกลอดถึงที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็คเกจของ HFS

γ (dB)	$N_S = 5$			$N_S = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	8.931 0.010	10.246 0.005	11.793 0.003	8.915 0.014	9.746 0.008	10.928 0.005
-33	9.289 0.009	10.761 0.005	12.357 0.003	9.181 0.012	10.122 0.007	11.361 0.004
-23	10.548 0.005	12.516 0.003	14.641 0.002	10.001 0.008	11.339 0.004	13.121 0.003

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13, 14, 16, 18 เล็กน้อยเนื่องจากข้อมูลของแพ็กเก็ต LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS ที่มีผลมาจาก บล็อกเฟดดิ้ง จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแย่ลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่า α ค่าเดียวกันเล็กน้อย

23. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 143$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

ผลลัพธ์ตารางที่ 21 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_R = 143$ ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกันและสลับลำดับทั้งแพ็กเก็ตของ LFS 1 แพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 21 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่ใช้ขนาด $N_R = 143$ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	$N_S = 5$			$N_S = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	16.824 0.009	20.388 0.005	22.649 0.003	15.813 0.012	19.007 0.007	21.765 0.004
-33	17.763 0.007	21.244 0.004	23.295 0.002	16.667 0.011	19.777 0.006	22.497 0.003
-23	20.541 0.004	23.120 0.002	25.254 0.001	19.021 0.007	22.144 0.003	24.449 0.002

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 มากกว่าตารางที่ 14, 16, 18, 20 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้ลดการเกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแยกลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่าจาก α ค่าเดียวกันเล็กน้อย

24. การทดสอบการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็กเก็ต) กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_S = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็กเก็ตของ HFS

ผลลัพธ์ตารางที่ 22 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็กเก็ต) ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_S = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของ

รูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกัน และสลับลำดับแพ็คเกจของ HFS จำนวน 101 แพ็คเกจ

ตารางที่ 22 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็คเกจ) กับช่องสัญญาณบล็อกลอดดิงที่มี $N_s = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลเฉพาะแพ็คเกจของ HFS

γ (dB)	$N_s = 5$			$N_s = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	9.073 0.004	10.434 0.002	12.037 0.001	9.044 0.005	9.924 0.002	11.129 0.001
-33	9.441 0.003	10.963 0.001	12.619 0.001	9.323 0.004	10.308 0.002	11.560 0.001
-23	10.741 0.002	12.819 0.001	15.097 0.0004	10.177 0.002	11.544 0.001	13.464 0.001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13, 14, 16, 18, 20 เล็กน้อย เนื่องจากข้อมูลของแพ็คเกจ LFS ซึ่งเป็นรายละเอียดหลักของภาพไม่ถูกสลับลำดับข้อมูลจึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็คเกจ LFS ที่มีผลมาจาก บล็อกลอดดิง จึงทำให้การถอดรหัสวีเทอรัปมีประสิทธิภาพในการถอดรหัสลดลง แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแย่ลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่า α ค่าเดียวกันเล็กน้อย

25. การทดสอบการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็คเกจ) กับช่องสัญญาณบล็อกลอดดิงที่มี $N_s = 5$ และ 8 ตามลำดับ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็คเกจ LFS และ HFS

ผลลัพธ์ตารางที่ 23 เป็นการทดสอบการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็กเก็ต) ของการถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_s = 5$ และ 8 เพื่อแสดงให้เห็นถึงความแตกต่างของรูปภาพที่สร้างกลับมาเมื่อสัญลักษณ์ที่ติดกัน 5 สัญลักษณ์และ 8 สัญลักษณ์ที่ถูกกระทบกับค่าจาก α ค่าเดียวกันมาเปรียบเทียบกันและสลับลำดับทั้งแพ็กเก็ตของ LFS เพียงหนึ่งแพ็กเก็ต และ HFS จำนวน 101 แพ็กเก็ต

ตารางที่ 23 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่ม (ทั้งแพ็กเก็ต) กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_s = 5$ และ 8 โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS

γ (dB)	$N_s = 5$			$N_s = 8$		
	PSNR (dB)			PSNR (dB)		
	WER			WER		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	20.815 0.003	25.231 0.001	28.281 0.0003	20.202 0.003	24.969 0.0009	28.182 0.0003
-33	21.721 0.002	26.033 0.001	29.002 0.0002	21.075 0.002	25.676 0.0007	28.853 0.0002
-23	24.684 0.001	28.231 0.0003	30.200 0.0001	24.329 0.001	28.234 0.0003	30.164 0.0001

จากตารางจะเห็นว่า ค่า PSNR และ WER จะดีกว่าตารางที่ 13 มากกว่าตารางที่ 14, 16, 18, 20, 22 เนื่องจากทั้งข้อมูลของแพ็กเก็ต LFS และ HFS ถูกสลับลำดับข้อมูล จึงทำให้เกิดความผิดพลาดที่ต่อเนื่องกับ แพ็กเก็ต LFS และ HFS ที่มีผลมาจาก บล็อกเฟดดิ้งได้เพิ่มขึ้น จึงทำให้การถอดรหัสวีเทอร์บีมีประสิทธิภาพที่ดีขึ้น แต่อย่างไรก็ตาม ค่า PSNR และ WER จะแยกลงเมื่อสัญลักษณ์ที่ติดกันจำนวนมากกว่าถูกกระทบจากค่า α ค่าเดียวกันเล็กน้อย ตามรูปภาพตัวอย่างที่แสดงไว้จากภาพที่ 33 (ก) ใช้ค่า $N_s = 8$ จะเห็นว่าภาพที่สร้างกลับมาดูผิดเพี้ยนมากกว่าภาพที่ 33 (ข) ที่ใช้ค่า $N_s = 5$



(ก)

(ข)

ภาพที่ 33 รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP ก) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 8$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_r = 143$ ได้ PSNR = 22.885 dB, WER = 0.0026 ข) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_r = 143$ ได้ PSNR = 26.662 dB, WER = 0.00099

26. การทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ กับช่องสัญญาณบล็อกเฟดดิ้งที่มี $N_s = 5$ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็คเกจ LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 20 dB, 21 dB, 22 dB แล้วหาขนาด N_r ที่เหมาะสม

ผลลัพธ์จากตารางที่ 24, 25, 26 สำหรับใช้เป็นแนวทางในการปรับปรุงวิธีการในการพัฒนาระบบเพื่อเปลี่ยนแปลงค่าของ N_r ให้เป็นไปตามสภาพของช่องสัญญาณเพื่อคงค่า PSNR ตามที่เราต้องการโดยการใช้การถอดรหัสแบบ MAP สำหรับช่องสัญญาณแบบบล็อกเฟดดิ้งที่มีค่า γ และ $\bar{\gamma}_b$ ที่เราสนใจในการทดสอบตามที่อธิบายไว้ก่อนหน้านี้แล้ว โดยใช้ $N_s = 5$ และสลับลำดับทั้งแพ็คเกจของ LFS 1 แพ็คเกจ และ HFS จำนวน 101 แพ็คเกจ ตัวอย่างของรูปภาพที่สร้างกลับมาได้แสดงไว้จากภาพที่ 34

ตารางที่ 24 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับช่องสัญญาณบล็อกละเฟรมที่มี $N_s = 5$ โดยการใช้อัลกอริทึมแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็คเกจ LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 20 dB

γ (dB)	N_R		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	-	132	44
-33	-	76	38
-23	132	38	23

จากตารางที่ 24 เราคงค่าของ PSNR = 20 dB และหาขนาดของ N_R ที่เหมาะสมสำหรับแต่ละสภาพช่องสัญญาณที่เราสนใจ โดยจะเห็นว่าไม่มีบางช่องสัญญาณ (แสดงด้วย '-') ที่ไม่อาจหาขนาดของ N_R ที่เหมาะสมได้เนื่องจากระดับของ PSNR คงที่ตั้งแต่ค่าที่น้อยกว่า 20 dB ถึงแม้จะปรับขนาด N_R เพิ่มขึ้นก็ยังไม่ทำให้ PSNR เพิ่มขึ้นถึงค่าที่เรากำหนดให้ได้ แต่สำหรับช่องที่ระบุค่า N_R แสดงให้เห็นว่าขนาดของ N_R แปรผกผันกับค่า γ และ $\bar{\gamma}_b$

ตารางที่ 25 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับช่องสัญญาณบล็อกละเฟรมที่มี $N_s = 5$ โดยการใช้อัลกอริทึมแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็คเกจ LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 21 dB

γ (dB)	N_R		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	-	-	55
-33	-	132	40
-23	-	50	26

จากตารางที่ 25 เราคงค่าของ PSNR = 21 dB และหาขนาดของ N_R ที่เหมาะสมสำหรับแต่ละสภาพช่องสัญญาณที่เราสนใจ โดยจะเห็นว่าไม่มีบางช่องสัญญาณ (แสดงด้วย '-') ที่ไม่อาจหาขนาด

ของ N_R ที่เหมาะสมได้เนื่องจากระดับของ PSNR คงที่ตั้งแต่ค่าที่น้อยกว่า 21 dB ถึงแม้จะปรับขนาด N_R เพิ่มขึ้นก็ยังไม่ทำให้ PSNR เพิ่มขึ้นถึงค่าที่เรากำหนดให้ได้ แต่สำหรับช่องที่ระบุค่า N_R แสดงให้เห็นว่าขนาดของ N_R แปรผกผันกับค่า γ และ $\bar{\gamma}_b$

ตารางที่ 26 ผลจากการทดสอบโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับช่องสัญญาณบิตอ็อกเฟดดิ้งที่มี $N_S = 5$ โดยการใช้การถอดรหัสแบบ MAP และการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS และ HFS โดยรักษาระดับ PSNR อยู่ที่ 22 dB

γ (dB)	N_R		
	$\bar{\gamma}_b = 3.75$ (dB)	$\bar{\gamma}_b = 5.00$ (dB)	$\bar{\gamma}_b = 6.25$ (dB)
-43	-	-	127
-33	-	-	61
-23	-	64	31

จากตารางที่ 26 เราคงค่าของ PSNR = 22 dB และหาขนาดของ N_R ที่เหมาะสมสำหรับแต่ละสภาพช่องสัญญาณที่เราสนใจ โดยจะเห็นว่าไม่มีบางช่องสัญญาณ (แสดงด้วย '-') ที่ไม่อาจหาขนาดของ N_R ที่เหมาะสมได้เนื่องจากระดับของ PSNR คงที่ตั้งแต่ค่าที่น้อยกว่า 22 dB ถึงแม้จะปรับขนาด N_R เพิ่มขึ้นก็ยังไม่ทำให้ PSNR เพิ่มขึ้นถึงค่าที่เรากำหนดให้ได้ แต่สำหรับช่องที่ระบุค่า N_R แสดงให้เห็นว่าขนาดของ N_R แปรผกผันกับค่า γ และ $\bar{\gamma}_b$



(ก)

(ข)

(ค)

ภาพที่ 34 รูปภาพ Lena ที่สร้างกลับมาโดยการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP ก) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_r = 23$ ได้ PSNR = 20.054 dB, WER = 0.0024 ข) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_r = 26$ ได้ PSNR = 21.551 dB, WER = 0.0022 ค) $\bar{\gamma}_b = 6.25$ dB และ $N_s = 5$ โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_r = 31$ ได้ PSNR = 22.568 dB, WER = 0.00137

27. การพิจารณาเวลาที่ใช้ในการประมวลผลของการสลับลำดับแบบสุ่มและการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ของแพ็คเกจขนาดยาวที่สุดและสั้นที่สุด

ผลลัพธ์จากตารางที่ 27, 28 เพื่อเปรียบเทียบให้เห็นว่าขนาดของแพ็คเกจมีผลอย่างมากต่อเวลาที่ใช้ในการสลับข้อมูลและสลับข้อมูลกลับ โดยจากตารางที่ 27 เราได้แสดงเวลาที่ใช้กับแพ็คเกจของ HFS ที่มีขนาดยาวที่สุดคือ 878 สัญลักษณ์ โดยการใช้การสลับลำดับแบบสุ่มที่สลับข้อมูลทั้งแพ็คเกจ และการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ จะเห็นว่าการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ สามารถลดเวลาที่ใช้ในการสลับลำดับข้อมูลและสลับข้อมูลกลับลงได้มาก ซึ่งก็คือการสลับลำดับแบบสุ่มที่สลับข้อมูลทั้งแพ็คเกจใช้เวลาเป็น 3 เท่ากว่าๆ ถ้าเทียบกับ การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ที่ใช้ขนาด $N_r = 143$ โดยที่ภาพที่สร้างกลับมาก็ใกล้เคียงกันคือ PSNR ห่างกันประมาณ 3 dB กว่าๆ ตามที่ได้แสดงไว้จากภาพที่ 35 ส่วนตารางที่ 28 เราได้แสดงเวลาที่ใช้กับแพ็คเกจของ HFS ที่มีขนาดสั้นที่สุดคือ 237 สัญลักษณ์เท่านั้น ซึ่งจะเห็นว่าเวลาที่ใช้อาจห่างกันเล็กน้อยเนื่องจาก ขนาดของข้อมูลที่ทำกรสลับลำดับ ไม่ต่างกันมาก แต่แพ็คเกจขนาดนี้มีเพียงแพ็คเกจเดียวเท่านั้น ซึ่งแพ็คเกจส่วน

ใหญ่มีขนาดใกล้เคียงกับขนาดที่ยาวที่สุดมากกว่า ทำให้โดยรวมแล้วการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ใช้เวลาในการสลับลำดับข้อมูลและสลับลำดับข้อมูลกลับน้อยกว่าการสลับลำดับข้อมูลแบบสุ่มที่สลับข้อมูลทั้งแพ็คเกจได้มาก ในขณะที่คุณภาพของรูปภาพที่สร้างกลับมายังใกล้เคียงกันอยู่

ตารางที่ 27 ตารางเปรียบเทียบเวลาที่ใช้ในการประมวลผลของการสลับลำดับแบบสุ่มและการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ของแพ็คเกจ ขนาด 878 สัญลักษณ์

Packet size = 878	Average Interleaver time	Average Deinterleaver time
Random Interleaver (Whole Packet)	4.846 ms	18.579 ms
Reconfigurable Random Interleaver ($N_R=128$)	1.440 ms	4.784 ms
Difference Time	3.406 ms	13.795 ms
Time Ratio	3.365	3.884

ตารางที่ 28 ตารางเปรียบเทียบเวลาที่ใช้ในการประมวลผลของการสลับลำดับแบบสุ่มและการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ของแพ็คเกจ ขนาด 237 สัญลักษณ์

Packet size = 237	Average Interleaver time	Average Deinterleaver time
Random Interleaver (Whole Packet)	0.381 ms	1.391 ms
Reconfigurable Random Interleaver ($N_R=128$)	0.341 ms	0.844 ms
Difference Time	0.040 ms	0.547 ms
Time Ratio	1.117	1.648



(ก)

(ข)

ภาพที่ 35 รูปภาพ Lena ที่สร้างกลับมาโดยการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้กับการถอดรหัสแบบ MAP ก) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มทั้งแพ็คเกจได้ PSNR = 30.450 dB, WER = 0.00012 ข) $\bar{\gamma}_b = 6.25$ dB โดยผ่านการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ $N_R = 143$ ได้ PSNR = 26.662 dB, WER = 0.00099

วิจารณ์

1. จากการทดสอบการถอดรหัสแบบ MAP และ ML เปรียบเทียบกัน ในทางทฤษฎีแล้ว การถอดรหัสแบบ MAP ควรจะมีประสิทธิภาพมากกว่า เนื่องจากมีการคำนวณความน่าจะเป็นของการเปลี่ยนแปลงข้อมูลมาประกอบกับการถอดรหัส ในขณะที่ ML ไม่ได้ใช้ แต่จากผลลัพธ์ที่ได้ จะเห็นว่าส่วนใหญ่การถอดรหัสแบบ MAP จะมีประสิทธิภาพที่แย่กว่า ML เล็กน้อยในกรณีที่มีการสลับลำดับข้อมูลทุกแพ็คเกจ อาจเนื่องมาจากหลังจากการสลับลำดับข้อมูลกลับ ทำให้ค่าความน่าจะเป็นถูกเปลี่ยนไปจากเดิม จึงทำให้ค่าความน่าจะเป็นที่นำมาใช้กับการถอดรหัสแบบ MAP ไม่ได้ช่วยให้การถอดรหัสแม่นยำขึ้น แต่กลับคิดเพี้ยนไปในบางช่วงของการสลับลำดับข้อมูลกลับ

2. จากการทดสอบการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยการเพิ่มขนาด N_R ไปเรื่อยๆ และมีการสลับลำดับข้อมูลเฉพาะแพ็คเกจที่เป็น HFS จำนวน 101 แพ็คเกจเท่านั้น จะเห็น

ว่าประสิทธิภาพของรูปภาพที่สร้างกลับมาปรับปรุงขึ้นเล็กน้อยเท่านั้น ดังแสดงไว้จากกราฟ ภาพที่ 23, 24 และ 25 เนื่องมาจากข้อมูลหลักซึ่งเป็น โครงร่างหลักของภาพอยู่ที่แพ็กเก็ต LFS เป็นหลัก ซึ่งยังคงถูกผลกระทบจากความผิดพลาดอย่างต่อเนื่องอยู่เหมือนเดิม ซึ่งส่วนที่ทำการสลับลำดับข้อมูลเกิดขึ้นกับแพ็กเก็ตของ HFS เท่านั้น จึงเป็นการลดความผิดพลาดอย่างต่อเนื่องให้กับข้อมูลที่เป็นส่วนประกอบปลีกย่อยของภาพจึงทำให้ ค่า PSNR ดีขึ้นเล็กน้อยเท่านั้น

3. จากการทดสอบจากการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยการเพิ่มขนาด N_R ไปเรื่อยๆ และมีการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS จำนวน 1 แพ็กเก็ตและแพ็กเก็ตเป็น HFS จำนวน 101 แนวโน้มของ PSNR จากการปรับ ขนาด N_R จาก 32 ไปเป็น 64 จะเพิ่มขึ้น ประมาณ 3 dB แต่จาก 64 เป็นต้นไปแนวโน้มของ PSNR มีการเพิ่มขึ้นไม่มากนัก ประมาณ 1-2 dB และค่อยๆ เข้าใกล้ค่าคงที่เรื่อยๆ ซึ่งอาจเกิดจากขนาดของ N_R เมื่อแบ่งขนาดของแพ็กเก็ต LFS เพื่อทำการสลับลำดับข้อมูลแล้ว ในช่วงของ 32 – 64 ทำให้เหลือเศษจากการถูกสลับลำดับน้อยกว่า ขนาดของ N_R ที่มากขึ้น จึงได้ค่า PSNR ที่เพิ่มขึ้นมากกว่า เนื่องจากเศษที่ยังกระทบกับความผิดพลาดอย่างต่อเนื่องมีจำนวนน้อยกว่า และเมื่อเราทำการสลับลำดับเศษที่เหลือด้วยจะได้กราฟที่แสดงแนวโน้มของ PSNR ที่มีแนวโน้มที่ดีขึ้นอย่างเห็นได้ชัด

4. จากการทดสอบจากการสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ โดยการเพิ่มขนาด N_R ไปเรื่อยๆ และมีการสลับลำดับข้อมูลทั้งแพ็กเก็ต LFS จำนวน 1 แพ็กเก็ตและแพ็กเก็ตเป็น HFS จำนวน 101 โดยทดสอบกับช่องสัญญาณบล็อกเฟดดิ้ง ที่มีขนาดของ $N_S = 5$ และ 8 เทียบกันพบว่า ในกรณีที่ N_R อยู่ในช่วง 32 – 64 ค่า PSNR ระหว่าง $N_S = 5$ และ 8 มีความแตกต่างกันค่อนข้างสูง คือ ประมาณ 3 dB แต่ถ้าเพิ่มขนาด N_R มากกว่านี้ ค่า PSNR ระหว่าง $N_S = 5$ และ 8 ห่างกันเล็กน้อยเท่านั้น ซึ่งคาดว่า เศษที่เหลือจากการสลับลำดับข้อมูลที่มากขึ้น ทำให้ดึงค่า PSNR ของทั้งสองกรณีมาใกล้เคียงกัน

5. จากการทดสอบการคงค่าของ PSNR ตามที่กำหนดแล้วหาค่า N_R ที่เหมาะสมอาจได้ขนาดที่ไม่แน่นอนมากนักเนื่องจากไม่ได้ทดสอบจากการปรับค่า N_R ไปตามค่าที่เพิ่มขึ้นเรื่อยๆ แต่สุ่มมาจากช่วงที่เป็นไปได้จากการทดสอบก่อนหน้านี้

สรุปและข้อเสนอแนะ

สรุป

การทดสอบที่ใช้ในงานวิจัยนี้ได้จำลองระบบการรับส่งรูปภาพแบบ MPEG-4 ไว้ทั้งระบบ โดยประกอบด้วยการเข้ารหัสและการถอดรหัสภาพ การเข้ารหัสช่องสัญญาณแบบริงคอนโวลูชันนอลและการคำนวณหาเส้นทางเทลิสและการถอดรหัสวีเทอร์บี การมอดูเลตและดีมอดูเลตสัญญาณ ส่วนที่พัฒนาขึ้นมาเพื่อปรับปรุงคุณภาพของรูปภาพที่สร้างกลับมาโดยประเมินจากค่า PSNR และ WER และยังสามารถลดความซับซ้อนของการสลับลำดับข้อมูลแบบสุ่มลงและลดเวลาในการสลับลำดับในการสลับลำดับข้อมูลลงคือ การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ แล้วทดสอบกับช่องสัญญาณ Rician บล็อกเฟดดิ้ง ที่ทำให้สัญลักษณ์ที่ติดกัน เป็นช่วงๆ ถูกกระทบกับค่าจาก α ค่าเดียวกัน ซึ่งทำให้สัญญาณเสียหายเป็นช่วงๆ เราสามารถสรุปได้จากการทดสอบว่า

1. เราสามารถปรับปรุงประสิทธิภาพของรูปภาพที่สร้างกลับมาได้ถึง ประมาณ 14 dB จากการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ที่มี $N_R = 143$ สัญลักษณ์ กับช่องสัญญาณที่มีค่า $\bar{\gamma}_b = 6.25$ dB และ ค่า $\gamma = -23$ dB ตามตัวอย่างรูปภาพจากภาพที่ 29
2. เราสามารถประเมินความซับซ้อนที่สามารถลดลงได้จากการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ จากเวลาที่ใช้ในการสลับลำดับข้อมูล โดยเวลาที่ลดลงเมื่อประเมินจากแพ็คเกจของ HFS ขนาด 878 สัญลักษณ์แล้ว เราสามารถลดเวลาลงเหลือประมาณ 1 ใน 3 จากการสลับลำดับข้อมูลแบบสุ่มทั้งแพ็คเกจ โดยใช้ขนาด $N_R = 143$ ซึ่งสามารถสร้างภาพกลับมาได้ใกล้เคียงกัน แต่ใช้เวลาต่างกันอย่างชัดเจน
3. ขนาดของ N_R มีผลต่อประสิทธิภาพของภาพที่รับมาได้ ถ้าแบ่งการสลับลำดับข้อมูลตามขนาดของ N_R แล้วได้เศษที่เหลือมากจะทำให้ข้อมูลกระทบกับการเฟดดิ้งเพิ่มขึ้นจึงทำให้ประสิทธิภาพไม่ได้เพิ่มขึ้นเท่าที่ควร แต่จะได้ประสิทธิภาพที่เพิ่มขึ้นอย่างชัดเจนถ้าทำการสลับลำดับข้อมูลเศษที่เหลือด้วย

4. การสลับลำดับข้อมูลของแพ็กเก็ต LFS ซึ่งเป็นข้อมูลเค้าโครงหลักของภาพ สามารถเพิ่มประสิทธิภาพของภาพที่สร้างกลับมาอย่างชัดเจน หลังจากการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้

ข้อเสนอแนะ

1. ค่า N_R ที่นำไปใช้กับการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ควรจะมีสมการในการคำนวณเพื่อให้สอดคล้องกับขนาดของแพ็กเก็ต ซึ่งจะทำได้สามารถปรับปรุงประสิทธิภาพได้มากขึ้น
2. ถ้าสามารถปรับปรุงการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ให้สามารถปรับค่า N_R แบบไดนามิกเพื่อเป็นไปตามสภาพของช่องสัญญาณจะทำให้ระบบมีประสิทธิภาพมากขึ้น
3. ค่า N_R ที่นำไปใช้กับการใช้การสลับลำดับข้อมูลแบบสุ่มที่กำหนดขนาดได้ ไม่จำเป็นต้องเป็นค่าทวิคูณไปเรื่อยๆ เนื่องจากประสิทธิภาพที่เพิ่มขึ้นไม่ได้เป็นไปตามเชิงเส้น ควรจะมีสมการที่หาความสัมพันธ์ระหว่างค่า N_R และ PSNR

เอกสารและสิ่งอ้างอิง

- Elliot, E. O. 1963. Estimates of error rates for codes on burst noise channels. **Bell Syst. Tech. Journal** 42: 1977-1977.
- Gilbert, E. N. 1960. Capacity of a burst-noise channel. **Bell Syst. Tech. Journal** 39: 1253-1265.
- Mahapakulchai, S. 2007. MAP decoding for polynomial ring convolutional trellis codes for MPEG-4 image transmission system over Rician fading channel, pp. 655-658. *In* **Proceeding of the 4th Electrical Engineering/Electronic, Computer, Telecommunications and Information Technology (ECTI) International Conference 2007**. ECTI Association.
- Mahapakulchai, S. and I. Sapisrisophon. 2008a. Design of Random Interleavers for the Variable Length of MPEG-4 Packets in Rician Fading Channels, pp. 409-412. *In* **Proceeding of the 5th Electrical Engineering/Electronic, Computer, Telecommunications and Information Technology (ECTI) International Conference 2008**. ECTI Association.
- Mahapakulchai, S. and I. Sapisrisophon. 2008b. MAP Decoding with Random Interleavers for MPEG-4 Image Indoor Wireless Transmission System, pp. 1-5. *In* **Proceeding of the 14th APCC 2008**.
- Mahapakulchai, S. and N. Phimsen. 2010. Design of reconfigurable random interleavers with MAP decoding for MPEG-4 image transmission system over Rician block-fading channels, pp. 978-982. *In* **Proceeding of the 7th Electrical Engineering/Electronic, Computer, Telecommunications and Information Technology (ECTI) International Conference 2010**. ECTI Association.

- Mahapakulchai,S. and R.E. Van Dyck. 2000. Source-Controlled Channel Decoding of MPEG-4 Imagery using CPFSK and Ring Convolutional Codes. *In Proceeding of IEEE Electro-Information Technology Conference*. June 2000, IEEE, Chicago.
- Mahapakulchai,S. and R.E. Van Dyck. 2004. Design of ring convolutional trellis codes for MAP decoding MPEG-4 images. *IEEE Transaction on Communications* 52(7): 1033-1037.
- Mahapakulchai,S. and W. Thongnumpen. 2007. Design of Block Interleavers in MPEG-4 Image Indoor Wireless Transmission System. *In International Conference on Engineering, Applied Sciences and Technology (ICEAST2007)*. 21-23 November2007, KMITL, Bangkok.
- Malkamaki,E. and H. Leib. 1997. Rate 1/n convolutional codes with interleaving depth of n over a block fading Rician channel, pp. 2002-2006. *In the 47th Vehicular Technology Conference*. 4-7 May 1997, IEEE.
- Ray,F., L. Meritxell, and V. Gregori. 2009. Adaptive Interleaver Based on Rate-Compatible Punctured Convolutional Codes. *IEEE Transaction on Communications* 57(6): 1593-1598.
- Shapiro,J. M. 1993. Embedded Image Coding using Zerotrees of Wavelete Coefficients. *IEEE Transaction on Signal Processing* 41(12): 3445-3462.
- Ungerboeck,G. 1982. Channel Coding with Multilevel/Phase Signals. *IEEE Transaction on Information Theory*. 28(1): 55-67.



1. ข้อเสนอแนะในการติดตั้งและใช้งานเบื้องต้นของโปรแกรม Netbeans IDE

1.1 การติดตั้งโปรแกรม

1.1.1 ดาวน์โหลดโปรแกรม JDK (Java SE Development Kit) จาก java.sun.com

1.1.2 ติดตั้งโปรแกรม JDK

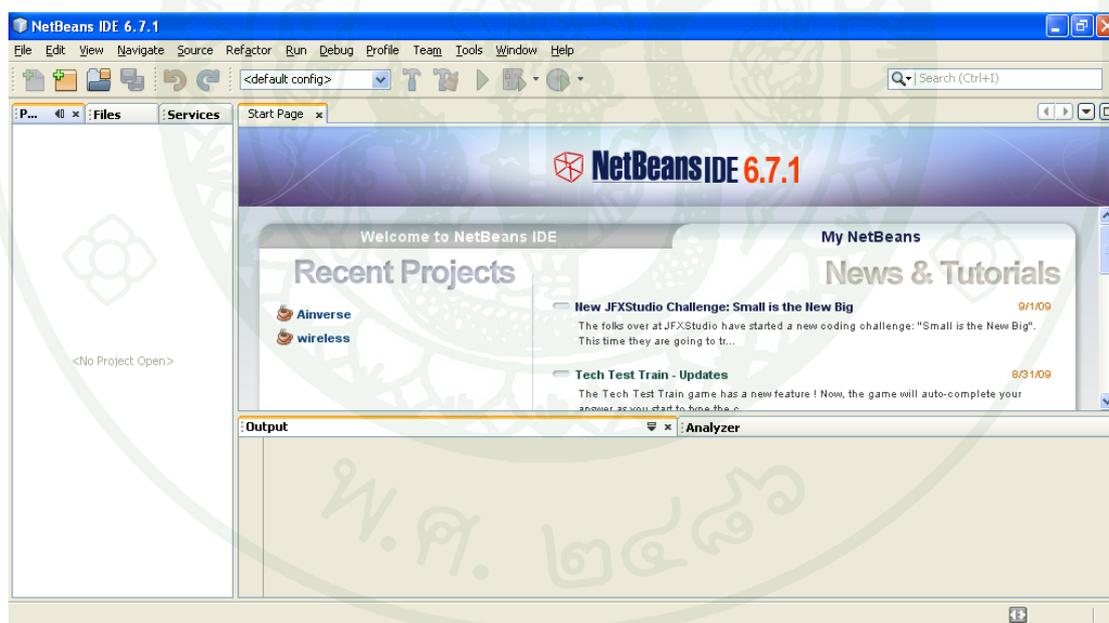
1.1.3 ดาวน์โหลดโปรแกรม Netbeans IDE จาก

www.netbeans.org/downloads/index.html

1.1.4 ติดตั้งโปรแกรม Netbeans IDE

1.2 การพัฒนาโปรแกรมด้วยโปรแกรม Netbeans IDE

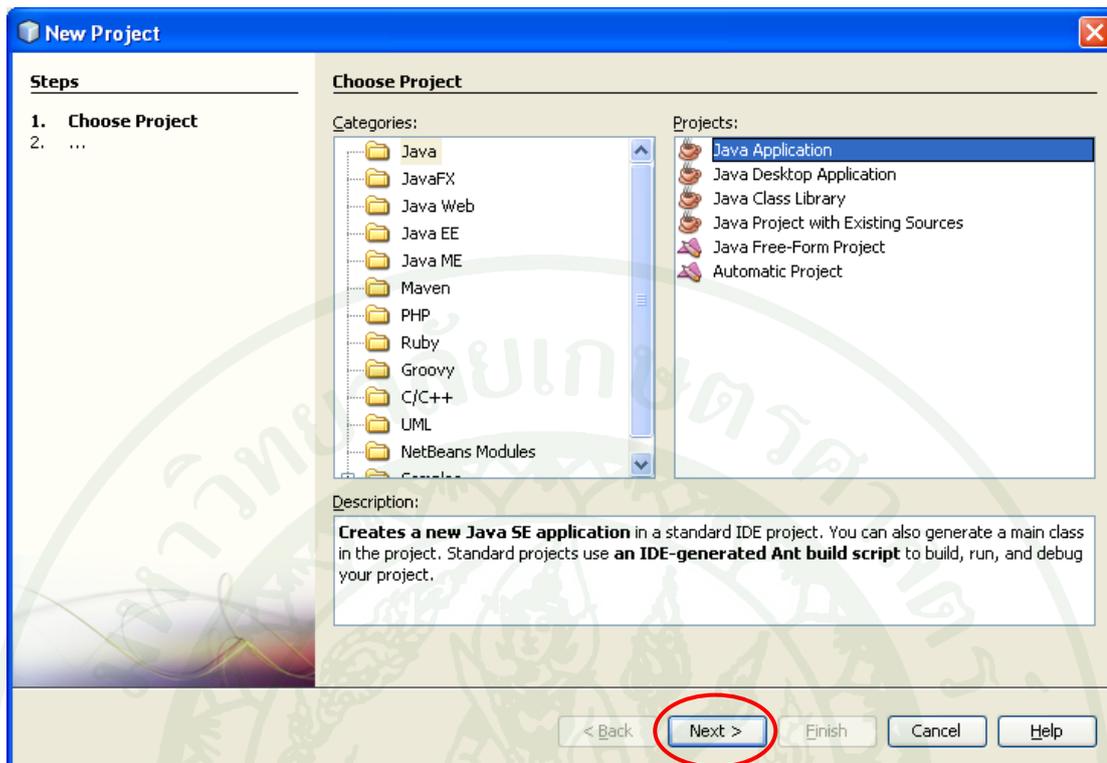
1.2.1 เปิดโปรแกรม Netbeans IDE จาก ปุ่ม Start -> All Program -> NetBeans -> NetBeans IDE 6.X.X จะแสดงหน้าต่างของโปรแกรมดังรูปต่อไปนี้



ภาพผนวกที่ 1 หน้าต่างโปรแกรม Netbeans IDE จากการเปิดโปรแกรมครั้งแรก

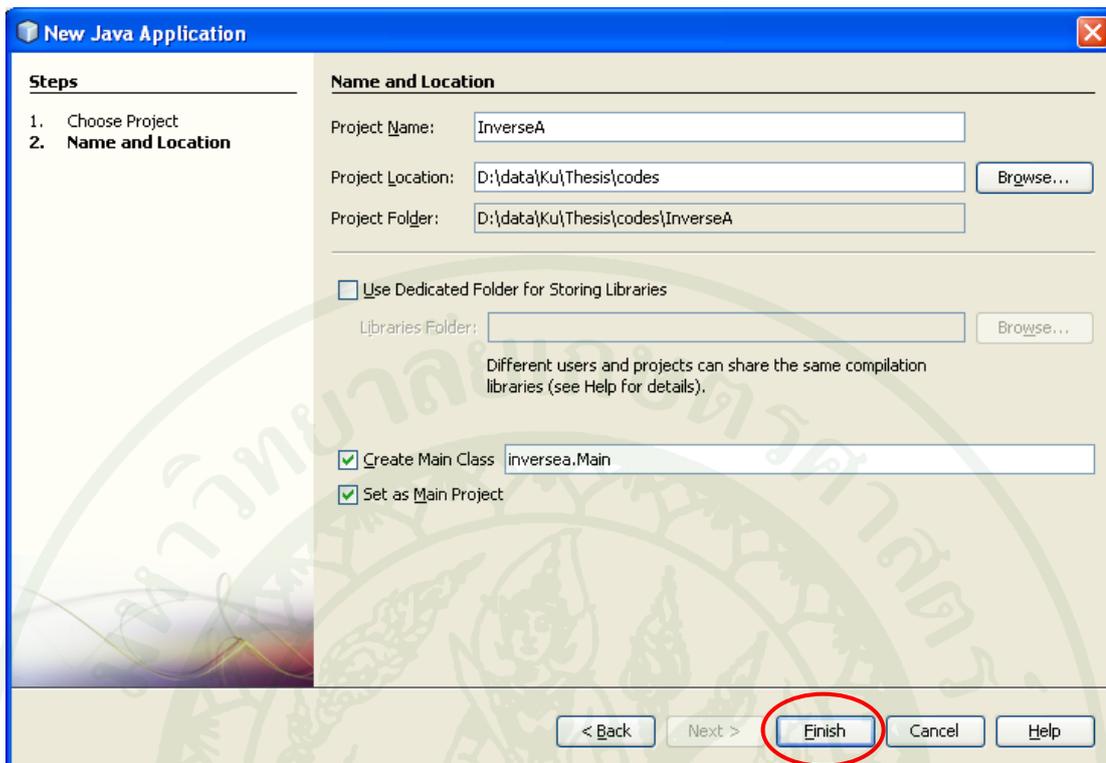
1.2.2 สร้างโปรเจกใหม่ โดยคลิกที่ เมนู File -> New Project จะแสดงหน้าต่างต่อไปนี้

ขึ้นมา



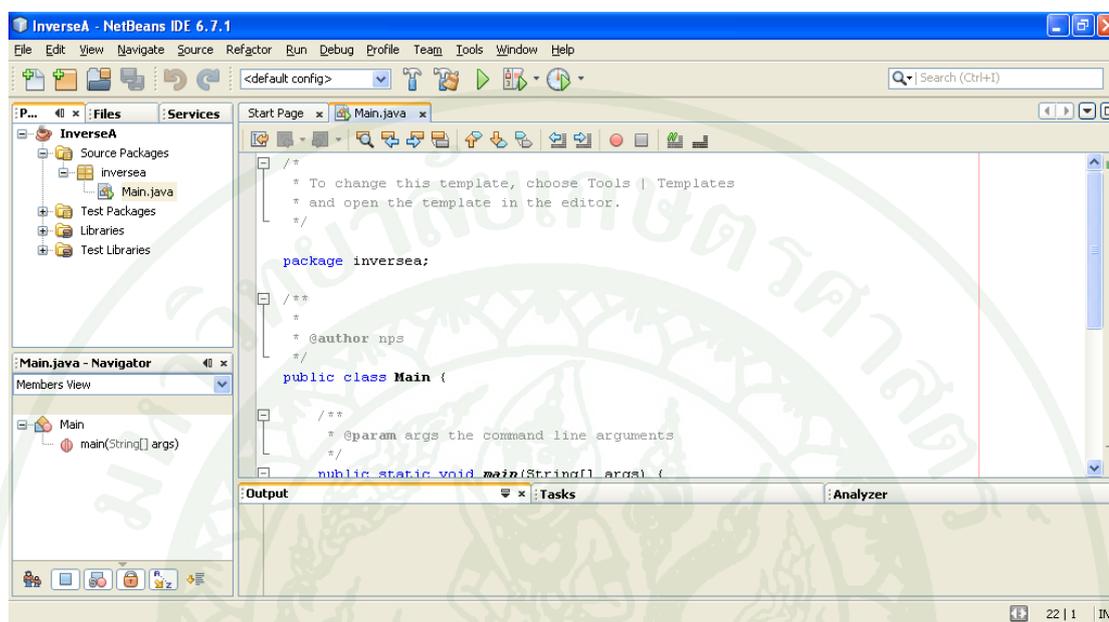
ภาพผนวกที่ 2 หน้าต่างการสร้างโปรเจกใหม่

1.2.3 คลิกที่ Java -> Java Application แล้วคลิก Next จะแสดงหน้าต่างใหม่ให้ใส่ชื่อโปรเจก ตามรูป



ภาพผนวกที่ 3 หน้าต่างให้ตั้งชื่อโปรเจกต์และกำหนดตำแหน่งเก็บโปรแกรมที่สร้างขึ้น

1.2.4 ใส่ชื่อโปรเจกต์แล้วกำหนดตำแหน่งที่จะเก็บโปรแกรมที่สร้างขึ้นจากนั้น คลิกที่ Finish โปรแกรม Netbeans IDE จะสร้างคลาสไว้ให้สอดคล้องตามชื่อที่เราตั้งไว้ ตามรูป



ภาพผนวกที่ 4 หน้าต่างแสดงคลาสที่สร้างขึ้นมาใหม่

1.2.5 จากนั้นเราสามารถพิมพ์คำสั่งภาษา Java ลงไปใน Main.java ตามหลังบรรทัด // *TODO code application logic here within main method* ตามตัวอย่างต่อไปนี้

```

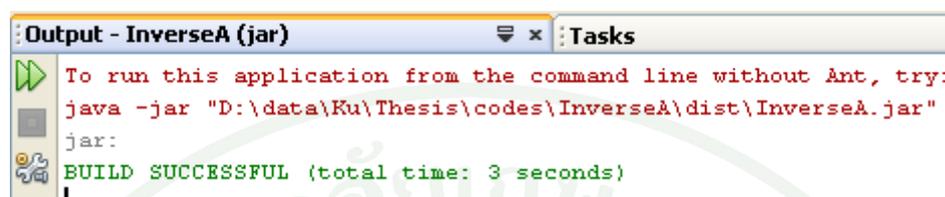
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("\n\tHello NetBeans IDE\n");
    }
}

```

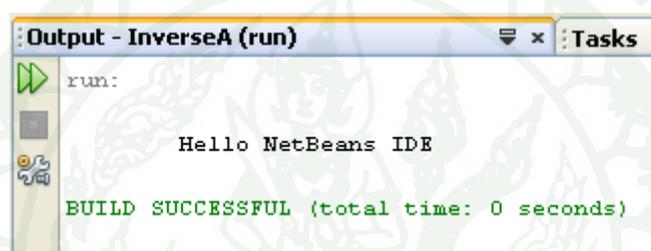
ภาพผนวกที่ 5 แสดงคำสั่งภาษา Java ที่พิมพ์เพิ่มลงไป

1.2.6 จากนั้นเราสามารถสั่งให้ Build โปรเจก แล้วรันโปรแกรม โดยกดคีย์ **F11** เพื่อ Build โปรเจก และกดคีย์ **F6** เพื่อรัน โปรแกรม โดยจะแสดงผลลัพธ์ ออกมาตามรูปต่อไปนี้



```
Output - InverseA (jar) Tasks
To run this application from the command line without Ant, try:
java -jar "D:\data\Ku\Thesis\codes\InverseA\dist\InverseA.jar"
jar:
BUILD SUCCESSFUL (total time: 3 seconds)
```

ภาพผนวกที่ 6 แสดงผลลัพธ์จากการ Build โปรเจก



```
Output - InverseA (run) Tasks
run:
Hello NetBeans IDE
BUILD SUCCESSFUL (total time: 0 seconds)
```

ภาพผนวกที่ 7 แสดงผลลัพธ์จากการรันโปรแกรม

ประวัติการศึกษาและการทำงาน

ชื่อ	นายอนุช พิมเสน
เกิดวันที่	12 เมษายน 2520
สถานที่เกิด	อำเภอบ้านตาก จังหวัดตาก
ประวัติการศึกษา	วศ.บ. (วิศวกรรมคอมพิวเตอร์) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ตำแหน่งปัจจุบัน	อาจารย์ประจำ
สถานที่ทำงานปัจจุบัน	มหาวิทยาลัยธุรกิจบัณฑิตย์
ผลงานดีเด่นและรางวัลทางวิชาการ	ได้รับการตีพิมพ์ผลงานสองเรื่อง คือ <ol style="list-style-type: none"> 1) "Design of reconfigurable random interleavers with MAP decoding for MPEG-4 image transmission system over Rician block-fading channels" ในงานประชุมวิชาการ ECTI-CON 2010 International Conference, pp. 978-982, May 2010. 2) "Study on Realtime Performance of Interleavers for MPEG-4 Image Transmission System" ในงานประชุมวิชาการ IC-ICTES 2011 International Conference, pp. 40-44, Jan., 2011.
ทุนการศึกษาที่ได้รับ	ทุนพัฒนาบุคลากร มหาวิทยาลัยธุรกิจบัณฑิตย์ และได้รับเงินอุดหนุนบางส่วนเพื่อเข้าร่วมประชุมวิชาการจากสถาบันวิจัยและพัฒนา มก (KURDI)