ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

ปริญญา

วิศวกรรมไฟฟ้า	วิศวกรรมไฟฟ้า
สาขา	**ภาควิชา**

เรื่อง ระบบรหัสภายในของรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสำหรับช่องสัญญาณเคลื่อนที่

Inner Coding Systems for Concatenated Codes with Vector Symbol Outer Decoder for Mobile Channels

นามผู้วิจัย นายวศิน สุขตลอดชีพ

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

(รองศาสตราจารย์อุศนา ตัณฑุลเวศม์, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม .....

(ผู้ช่วยศาสตราจารย์วิรุณศักดิ์ สันติเพ็ชร์, Ph.D.)

หัวหน้าภาควิชา .....

(รองศาสตราจารย์วิชัย สุระพัฒน์, วศ.ม.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

.....

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... **เดือน** ..... **พ.ศ.** .....

วิทยานิพนธ์

เรื่อง

ระบบรหัสภายในของรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสำหรับช่องสัญญาณเคลื่อนที่

Inner Coding Systems for Concatenated Codes with Vector Symbol Outer Decoder for Mobile Channels

โดย

นายวศิน สุขตลอดชีพ

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2556

วศิน สุขตลอดชีพ 2556: ระบบรหัสภายในของรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสำหรับช่องสัญญาณเคลื่อนที่ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: รองศาสตราจารย์อุศนา ตัณฑุลเวศม์, Ph.D. 92 หน้า

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบระบบรหัสคอนคาทิเนตโดยมีตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสำหรับรหัสคอนโวลูชันที่ใช้สัญลักษณ์ขนาดใหญ่บนช่องสัญญาณไร้สายแบบเคลื่อนที่ โดยเน้นการพัฒนาส่วนของรหัสภายใน ซึ่งประกอบด้วยการพัฒนาตัวถอดรหัสลิสและซอฟท์วีเทอร์บีสำหรับรหัสบล็อก และการใช้รหัสบล็อกแบบนอนไบนารีแทนการใช้รหัสไบนารีสำหรับรหัสภายใน โดยงานวิจัยนี้สามารถจำแนกระบบรหัสคอนคาทิเนตได้ทั้งสิ้น 3 รูปแบบ ได้แก่ (1) รหัสภายในแบบบีซีเอช ถอดรหัสภายในด้วยลิสแบบซอฟท์วีเทอร์บี และถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสัญลักษณ์ละ 26 บิตที่มีข้อมูลตัวเลือก (2) รหัสภายในแบบบีซีเอช ถอดรหัสภายในด้วยซอฟท์วีเทอร์บี และถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสัญลักษณ์ละ 104 บิตแบบไม่มีมีข้อมูลตัวเลือก และ (3) รหัสภายในแบบรีดโซโลมอน ถอดรหัสภายในด้วยวิธีทางพีชคณิตด้วยการตัดสินใจแบบฮาร์ด และถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสัญลักษณ์ละ 102 บิตแบบไม่มีมีข้อมูลตัวเลือก ซึ่งผลจากงานวิจัยแสดงให้เห็นว่าตัวถอดรหัสลิสวีเทอร์บีแบบซอฟท์สำหรับรหัสบล็อกสามารถใช้งานได้สำเร็จ สำหรับผลตัวถอดรหัสเวกเตอร์ซิมโบลที่ไม่มีรหัสภายใน ขนาดสัญลักษณ์ 128 บิตดีกว่าสัญลักษณ์ละ 64 บิต สำหรับช่องสัญญาณแบบไร้สายเคลื่อนที่ นอกจากนี้ระบบที่ใช้รหัสภายในแบบรีดโซโลมอนดีกว่าระบบที่ใช้รหัสภายในแบบบีซีเอชสำหรับช่องสัญญาณไร้สายเคลื่อนที่

_____ _____ ___ / ___ / ___

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Vasin Suktalordcheep 2013: Inner Coding Systems for Concatenated Codes with Vector Symbol Outer Decoder for Mobile Channels. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Department of Electrical Engineering. Thesis Advisor: Associate Professor Usana Tuntoolavest, Ph.D. 92 pages.

The purpose of this research is to design the concatenated coding systems with Vector Symbol Decoder (VSD) for outer convolutional codes with large symbol size in mobile channels. The focus is on the inner coding part, which includes the development of list and soft decision Viterbi Algorithm (VA) for block codes and the use of a nonbinary block code instead of a binary code as an inner code. Three concatenated coding systems have been considered as follows: (1) a BCH inner code with list soft VA inner decoder and VSD with 2 alternative choices and 26-bit outer symbols, (2) a BCH inner code with soft VA and VSD with no alternative choice and 102-bit outer symbols and (3) a Reed-Solomon (RS) inner code with Algebraic hard decision decoding and VSD with no alternative choice and 104-bit outer symbols. The result shows that the soft list VA for block codes can be achieved. For the performance of VSD only with no inner codes, the 128-bit symbol is better than the 64-bit symbol for the selected mobile channel. In addition, the system with RS inner code is better than the one with BCH inner code for the selected mobile channels.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ รศ.ดร.อุศนา ตัณฑุลเวศม์ ประธานกรรมการที่ปรึกษาวิทยานิพนธ์หลัก ผศ.ดร.วิรุณศักดิ์ สันติเพ็ชร์กรรมการที่ปรึกษาสาขาวิชารอง ที่ให้คำปรึกษาในการเรียน การค้นคว้าวิจัย ตลอดจนการตรวจแก้ไขวิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์ อีกทั้งรศ.ดร. ลัญฉกร วุฒิสิทธิกุลกิจ จากภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ที่สำหรับความกรุณาให้คำแนะนำในงานวิจัยของวิทยานิพนธ์ฉบับนี้เพื่อความสมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ให้ความรู้และอบรมสั่งสอน ซึ่งสามารถนำมาใช้ประโยชน์ได้ในการต่อไป รวมทั้งเพื่อนๆทั้งคุณชโณทัย ไชยวรรณ์ และคุณจตุพล ท่นไชยที่ช่วยเหลือในการทำงาน และขอขอบคุณสถาบันวิจัยและพัฒนาแห่งมหาวิทยาลัยเกษตรศาสตร์ที่ให้ทุนสนับสนุนการวิจัยครั้งนี้

ด้วยความดีหรือประโยชน์อันใดจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่บิดามารดาของผู้วิจัยซึ่งได้อบรมสั่งสอนและให้ความสนับสนุนผู้วิจัยมาโดยตลอด

วศิน สุขตลอดชีพ

พฤษภาคม 2556

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

คำอธิบายสัญลักษณ์และคำย่อ

AWGN	=	additive white gaussion noise
BCH	=	Bose Chaudhuri Hocquenghem
BPSK	=	binary phase shift keying
BSC	=	binary symmatric channel
GMD	=	minimum distance decoding
LCM	=	least common multiple
LVA	=	list viterbi algorithm
MLD	=	maximum likelihood decoding
PDF	=	probability density function
PLC	=	power-line channel
RS	=	Reed-Solomon
T-DMB	=	terrestrial-digital multimedia broadcasting
VSD	=	vector symbol decoding

ระบบรหัสภายในของรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบล สำหรับช่องสัญญาณเคลื่อนที่

Inner Coding Systems for Concatenated Codes with Vector Symbol Outer Decoder for Mobile Channels

คำนำ

การเข้ารหัสช่องสัญญาณเป็นกระบวนการตรวจสอบและแก้ไขความผิดพลาดของข้อมูลดังที่ได้กล่าวไว้ข้างต้น โดยมีการเพิ่มบิตตรวจสอบ (Check bit) เข้าไปในข้อมูลเพื่อตรวจสอบความผิดพลาด ถ้าเพิ่มบิตดังกล่าวมากพออาจจะสามารถแก้ไขความผิดพลาดของข้อมูลได้ด้วย ซึ่งการเข้ารหัสช่องสัญญาณก็มีหลายวิธีการแล้วแต่ความเหมาะสมในการใช้งาน แต่สำหรับการสื่อสารแบบไร้สายการเข้ารหัสแบบคอนคาทิเนต (Concatenated codes) เป็นวิธีหนึ่งที่นิยมนำไปใช้งาน

รหัสคอนคาทิเนตสามารถแก้ไขข้อมูลที่มีความผิดพลาดแบบเบริสต์ (Burst error) ซึ่งมักพบในการส่งข้อมูลผ่านช่องสัญญาณแบบไร้สาย โดยการเข้ารหัสช่องสัญญาณแบบคอนคาทิเนตอาจมีการเข้ารหัสสองส่วนหรือมากกว่านั้น แต่ที่นิยมใช้กันคือการเข้ารหัสสองส่วนได้แก่การเข้ารหัสภายนอก (Outer code) และการเข้ารหัสภายใน (Inner code) โดยการเข้ารหัสภายนอกใช้สัญลักษณ์นอนไบนารี (Nonbinary code) และการเข้ารหัสภายในจะใช้สัญลักษณ์ไบนารี (Binary code)

วิทยานิพนธ์นี้เป็นงานวิจัยเพื่อพัฒนารหัสคอนคาทิเนตในส่วนรหัสภายในบนช่องสัญญาณไร้สายเคลื่อนที่ เพื่อเพิ่มประสิทธิภาพของรหัสคอนคาทิเนตให้มากขึ้น โดยพัฒนารหัสภายในให้สามารถถอดรหัสแบบลิสวีเทอร์บีด้วยการตัดสินใจแบบอ่อน (Soft Decision) สำหรับรหัสบล็อก (Block codes) ได้ ซึ่งเป็นองค์ความรู้ใหม่สำหรับการถอดรหัสแบบลิสวีเทอร์บี และช่วยเพิ่มความยืดหยุ่นในการใช้งานรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายในแบบลิสวีเทอร์บีและรหัสภายนอกแบบเวกเตอร์ซิมโบล เพราะฝั่งเข้ารหัสทั้งรหัสภายในและ/หรือรหัสภายนอกสามารถเลือกได้ว่าจะใช้รหัสบล็อกหรือรหัสคอนโวลูชัน (Convolutional codes) และใช้รหัสรีดโซโลมอน (Reed-Solomon) ซึ่งเป็นรหัสนอนไบนารีเป็นรหัสภายใน พร้อมทั้งขยายขนาดสัญลักษณ์ภายนอกและ

ภายในให้ใหญ่ขึ้น ซึ่งช่วยเพิ่มความสามารถในการแก้ไขความผิดพลาดแบบเบริสต์ซึ่งเกิดจากช่องสัญญาณไร้สายเคลื่อนที่ได้ดีขึ้น

งานที่ทำซึ่งมีประโยชน์ต่องานวิจัยเกี่ยวกับรหัสช่องสัญญาณ

1. พัฒนาลิสวีเทอร์บีแบบอ่อนในส่วนการคำนวณและออกแบบแผนภาพแสดงการเปลี่ยนสถานะสำหรับรหัสบล็อก

2. เสนอและวิเคราะห์ประสิทธิภาพของรหัสภายในโดยใช้รหัสแบบนอนไบนารีแทนรหัสไบนารี

3. ปรับปรุงโปรแกรมตัวถอดรหัสเวกเตอร์ซิมโบลด้วยภาษา C++ โดยใช้โปรแกรม Microsoft Visual Studio เพื่อใช้ปรับขนาดสัญลักษณ์ได้ถึง 128 บิต แทนของเดิมที่มีขนาด 32 บิค

4. ตีพิมพ์เผยแพร่ในการประชุมวิชาการระดับนานาชาติ 2 บทความ

ผลงานที่ตีพิมพ์

1. Tuntoolavest, U., V. Suktalordcheep and C. Chaiwan. 2011. List-of-2 soft decision viterbi inner decoder for a generalized concatenated coding system. pp. 264-267. **Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI)**. 17-19 May 2011, Kohnkaen, Thailand.

2. Thonchai, J., V. Suktalordcheep and U. Tuntoolavest. 2013. Lab prototype of list-of-2 soft viterbi decoder for a BCH inner code in a generalized concatenated coding system. **The International Conference on Information and Communication Technology for Embeded System (ICICTES)**. 24-26 January 2013, Samutsongkhram, Thailand.

วัตถุประสงค์

1. พัฒนาส่วนของตัวเข้ารหัสภายในให้สามารถถอดรหัสบล็อกแบบลิส (List)

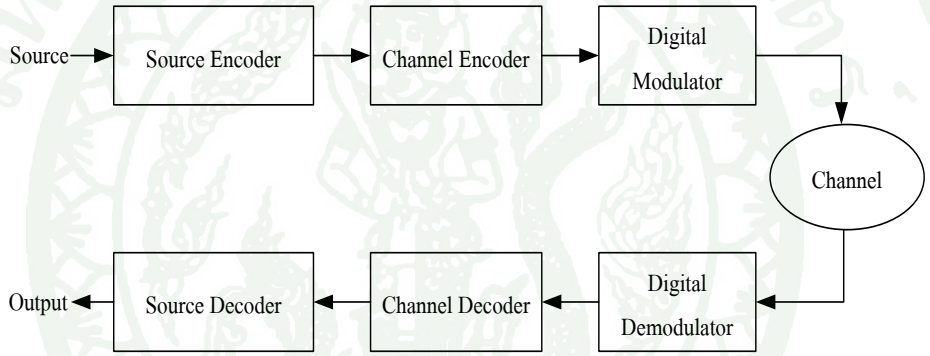
2. พัฒนารหัสภายในให้สามารถเลือกเข้ารหัสเชิงเส้นทั้งรหัสบล็อกหรือรหัสคอนโวลูชันโดยใช้ตัวถอดรหัสภายในตัวเดียวกัน ซึ่งถอดรหัสด้วยการตัดสินใจแบบอ่อน (Soft decision) ที่มีสองตัวเลือก

3. ปรับปรุงโปรแกรมภาษา C++ ที่ใช้ถอดรหัสแบบเวกเตอร์ซิมโบล (Vector symbol decoding) ให้สามารถเปลี่ยนขนาดของสัญลักษณ์

4. เพื่อให้ได้องค์ความรู้ใหม่ เพิ่มประสิทธิภาพของรหัสคอนคาทิเนตและสามารถนำไปพัฒนาชิ้นงานต้นแบบ

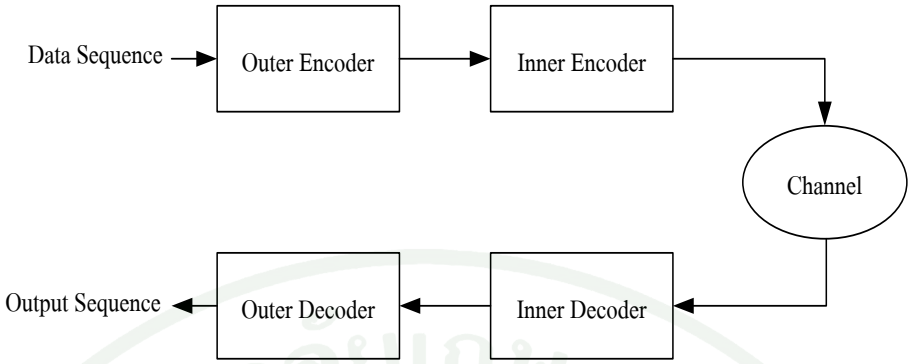
การตรวจเอกสาร

การสื่อสารคือการส่งข่าวสารจากผู้ส่งไปยังผู้รับผ่านตัวกลางหรือสื่อกลาง ซึ่งปัจจุบันระบบสื่อสารแบบดิจิทัลดังภาพที่ 1 ซึ่งประกอบด้วยการเข้าและถอดรหัสแหล่งข้อมูล (Source coding) มีหน้าที่ลดส่วนสำรอง (Redundancy) และเข้ารหัสเพื่อความปลอดภัย การเข้ารหัสและถอดรหัสช่องสัญญาณ (Channel coding) มีหน้าที่เพิ่มความน่าเชื่อถือให้กับข้อมูลโดยการตรวจสอบความผิดพลาดและแก้ไขความผิดพลาดของข้อมูล และการกล้ำสัญญาณและแยกสัญญาณ (modulate and demodulate) แบบดิจิทัล (Digital modulation) มีหน้าที่เปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณคลื่นเพื่อส่งสัญญาณผ่านช่องสัญญาณ



ภาพที่ 1 การสื่อสารแบบดิจิทัล

ในการสื่อสาร การรับ-ส่งข้อมูลอาจเกิดความผิดพลาดได้ สิ่งที่นำมาใช้เพื่อช่วยลดปัญหาดังกล่าวได้แก่รหัสช่องสัญญาณ ต่อมามีการพัฒนาให้มีการใช้งานรหัสช่องสัญญาณในลักษณะต่างๆ และการเข้ารหัสแบบคอนคาทิเนต ซึ่งถูกเสนอแนวคิดโดย (Forney, 1966) เป็นวิธีหนึ่งที่สามารถใช้ได้ดีกับช่องสัญญาณไร้สายซึ่งมักเกิดความผิดพลาดแบบเบริสต์ (Burst error) รหัสคอนคาทิเนตอย่างง่ายประกอบด้วยส่วนของรหัสภายนอกที่นิยมใช้คำรหัสแบบนอนไบนารี (Nonbinary code) และรหัสภายในที่นิยมใช้รหัสแบบไบนารี (Binary code) ดังแสดงในภาพที่ 2



ภาพที่ 2 รหัสช่องสัญญาณแบบคอนคาทิเนตอย่างง่าย

รหัสภายนอก (Outer Codes) ที่นิยมใช้กับรหัสช่องสัญญาณแบบคอนคาทิเนตคือรหัสรีดโซโลมอน (Reed-Solomon: RS) (Reed and Solomon, 1960) เนื่องจากเป็นรหัสที่ถูกสร้างมาให้เหมาะสมกับรหัสนอนไบนารี ส่วนรหัสภายใน (Inner Codes) นิยมใช้รหัสบล็อกแบบบีซีเอช (BCH) (Hocquenghem, 1959; Bose and Ray-Chaudhuri, 1960) หรือรหัสคอนโวลูชันที่เป็นรหัสแบบไบนารี แต่เดิมหากเข้ารหัสแบบใดมักใช้วิธีการถอดรหัสในแบบเดียวกัน เช่นหากเข้ารหัสรีดโซโลมอนหรือบีซีเอชอาจถอดรหัสด้วยวิธี (Berlekamp-Massey,1968) ถ้าเข้ารหัสคอนโวลูชันถอดรหัสด้วยวิธีของวีเทอร์บี (Viterbi,1967) เป็นต้น ภายหลังมีผู้คิดวิธีให้วีเทอร์บีสามารถถอดรหัสบล็อกได้ด้วย เช่นวิธีของ (Reeve and Amarasinghe, 2005) ต่อมามีการพัฒนาการถอดรหัสวีเทอร์บีที่ใช้กับการเข้ารหัสคอนโวลูชันให้มีจำนวนข้อมูลหลังถอดรหัสมากกว่าหนึ่งชุด ซึ่งช่วยเพิ่มประสิทธิภาพในการตรวจสอบความผิดพลาดของข้อมูล การถอดรหัสดังกล่าวถูกเรียกว่าลิสวีเทอร์บีอัลกอริทึม (List Viterbi Algorithm: LVA) (Seshadri and Sundberg, 1994)

จากนั้น Metzner and Tuntoolavest (2002) ใช้รหัสคอนโวลูชันเป็นทั้งรหัสภายนอกและรหัสภายใน และใช้ตัวถอดรหัสภายในที่ถอดรหัสแบบลิสวีเทอร์บีเหมาะที่จะนำมาประยุกต์ใช้ควบคู่กับการถอดรหัสแบบเวกเตอร์ซิมโบล (Vector Symbol Decoding: VSD) เป็นอย่างยิ่ง เพราะตัวถอดรหัสเวกเตอร์ซิมโบลสามารถถอดคำรหัสได้มากกว่าหนึ่งชุดคำรหัสซึ่งช่วยเพิ่มประสิทธิภาพในการแก้ไขคำรหัสที่ผิดพลาด ซึ่งปกติการถอดรหัสคอนโวลูชันแบบนอนไบนารีขนาดใหญ่ทำได้ยาก เนื่องจากมีการเปลี่ยนสถานะที่ซับซ้อน จึงไม่นิยมใช้รหัสคอนโวลูชันเป็นรหัสภายนอก แต่ตัวถอดรหัสเวกเตอร์ซิมโบลสามารถถอดรหัสคอนโวลูชันที่เป็นรหัสนอนไบนารีขนาดใหญ่ได้ง่าย ดังนั้นระบบรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลจึงสามารถเลือกใช้รหัสคอนโวลูชันเป็นรหัสภายนอกได้ อีกทั้งรหัสคอนโวลูชันมีข้อดีคือ สามารถ

เข้ารหัสต่อเนื่องกันโดยไม่จำกัดความยาว ซึ่งต่างจากรหัสบล็อกที่ต้องเข้ารหัสและถอดรหัสทั้งบล็อก ทำให้มีความยืดหยุ่นกว่าในการรับ-ส่งข้อมูล ดังนั้นจุดเด่นของตัวถอดรหัสเวกเตอร์ซิมโบลคือถอดรหัสนอนไบนารีได้ทั้งรหัสแบบบล็อกและรหัสคอนโวลูชัน ทำให้มีความยืดหยุ่นในการใช้งาน โดยตัวถอดรหัสภายนอกยังคงเป็นแบบเวกเตอร์ซิมโบลเหมือนเดิม ซึ่งตัวถอดรหัสเวกเตอร์ซิมโบลนั้นถูกเสนอโดย (Metzner and Kapturowski, 1990)

Tuntoolavest *et al*. (2007, 2010) ยังได้พัฒนารหัสคอนคาทิเนตที่ใช้รหัสภายนอกแบบเวกเตอร์ซิมโบลเป็นชิ้นงานต้นแบบ โดยส่วนรหัสภายในใช้การถอดรหัสแบบลิสวิเทอร์บี และพบว่าการพัฒนาลิสวิเทอร์บีเป็นชิ้นงานต้นแบบ การเพิ่มจำนวนผลลัพธ์มากกว่าสองทางเลือกประสิทธิภาพการถอดรหัสที่ได้เพิ่มขึ้นไม่มาก แต่ความซับซ้อนในการออกแบบเพิ่มขึ้นตามจำนวนทางเลือกที่ใช้ ดังนั้นจึงใช้เพียงสองผลลัพธ์สำหรับการสร้างเป็นชิ้นงานต้นแบบ

ต่อมา Andreadou and Pavlidou (2010) เสนอการใช้รีดโซโลมอนซึ่งเป็นรหัสแบบนอนไบนารีเป็นรหัสภายใน และใช้รหัสแอลดีพีซี (Low Density Parity Check: LDPC) เป็นรหัสภายนอก โดยอ้างว่าให้ประสิทธิภาพที่ดี จึงเป็นสาเหตุให้เกิดแนวคิดในการนำรหัสรีดโซโลมอนมาใช้เป็นรหัสภายในของรหัสคอนคาทิเนตในงานวิจัยนี้

รหัสรีดโซโลมอนเป็นรหัสบล็อกแบบนอนไบนารีที่ได้รับความนิยมมากและถูกใช้อย่างแพร่หลาย เนื่องจากเป็นรหัสที่ดี และมีประสิทธิภาพสูง ทำให้มีหลายงานวิจัยนำรหัสรีดโซโลมอนไปใช้งาน และหลายงานวิจัยพัฒนาการถอดรหัสรีดโซโลมอนให้มีประสิทธิภาพดียิ่งขึ้น โดยแนวทางการพัฒนารหัสรีดโซโลมอนส่วนใหญ่เน้นพัฒนาการถอดรหัสด้วยการตัดสินใจแบบอ่อนเช่นงานวิจัยของ Hu and S Lin (2003); Xia and Cruz (2007) Shayegh and Soleymani (2011) เพื่อเพิ่มประสิทธิภาพการถอดรหัสให้ดีขึ้น หรือลดความซับซ้อนในการถอดรหัสให้น้อยลง

โครงงานวิจัยนี้สนใจการศึกษาและวิจัยในส่วนรหัสภายในของรหัสคอนคาทิเนต โดยใช้การจำลองการทำงานบนเครื่องคอมพิวเตอร์ เพื่อพัฒนาขั้นตอนวิธี (Algorithm) ของรหัสช่องสัญญาณแบบคอนคาทิเนตซึ่งใช้ร่วมกับตัวกลางไร้สายแบบเคลื่อนที่ให้มีประสิทธิภาพและความเหมาะสมในการนำไปใช้งานจริงมากขึ้น โดยพัฒนาการถอดรหัสวีเทอร์บีของรหัสบล็อกให้สามารถถอดรหัสแบบสองผลลัพธ์ได้ และนำไปใช้ถอดรหัสภายนอกเวกเตอร์ซิมโบลแบบมีข้อมูลสำรอง เพื่อง่ายต่อการนำไปใช้งานตามความเหมาะสม เนื่องจากตัวถอดรหัสภายนอกเป็นรหัสเวกเตอร์ซิมโบลซึ่งใช้ได้กับรหัสแบบบล็อกเชิงเส้น หรือรหัสแบบคอนโวลูชันได้ ส่วนตัวถอดรหัสภายในที่เป็นลิสวีเทอร์บีสามารถเลือกใช้ถอดรหัสแบบสองผลลัพธ์ได้ทั้งรหัสแบบบล็อก หรือแบบคอนโวลูชันได้เช่นกัน ทำให้เกิดความหลากหลายในการเลือกใช้งาน และใช้รหัสภายในแบบนอนไบนารีกับรหัสคอนคาทิเนต โดยใช้รหัสรีดโซโลมอนและใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบล รวมถึงขยายขนาดสัญลักษณ์ของรหัสภายนอกให้ใหญ่ขึ้น โดยการปรับปรุงโปรแกรมของตัวถอดรหัสเวกเตอร์ซิมโบลซึ่งเขียนโดยภาษา C++ ให้สามารถปรับขนาดสัญลักษณ์ได้ ทำให้สามารถปรับขนาดสัญลักษณ์ให้หลากหลาย และสามารถเลือกใช้ขนาดสัญลักษณ์ได้สอดคล้องกับขนาดของรหัสภายใน

รหัสช่องสัญญาณเป็นส่วนประกอบหนึ่งของการสื่อสารในระบบดิจิทัลดังภาพที่ 1 โดยมีหน้าที่ตรวจสอบและแก้ไขความผิดพลาดที่เกิดขึ้นในการรับ - ส่งข้อมูลผ่านช่องสัญญาณ สามารถแบ่งได้หลักๆคือรหัสบล็อกเชิงเส้น และรหัสคอนโวลูชัน

1. รหัสบล็อกเชิงเส้น

การเข้ารหัสบล็อกจะแบ่งข้อมูลเพื่อนำไปเข้ารหัสเป็นกลุ่มหรือบล็อกพร้อมกัน ในแต่ละบล็อกประกอบด้วยข้อมูลขนาด k บิตสำหรับรหัสแบบไบนารีหรือ k สัญลักษณ์สำรหัสรหัสแบบนอนไบนารี และได้เป็นคำรหัสจำนวน n บิตหรือ n สัญลักษณ์ สามารถเขียนแทนว่า (n, k) หรือ k/n ก็ได้ดังภาพที่ 3 ซึ่งรหัสบล็อกเชิงเส้นจัดเป็นระบบที่ไม่มีความจำ กล่าวคือคำรหัสที่ได้จะไม่มีความสัมพันธ์กันระหว่างชุดคำรหัสก่อนหน้าหรือหลัง

ภาพที่ 3 คำรหัสแบบบล็อก

คำรหัสสามารถสร้างได้โดย (พิสิฐ วนิชชานันท์ และคณะ, 2552)

$$c = m \cdot G \tag{1}$$

c คือคำรหัส
m คือข้อมูลที่ต้องการเข้ารหัส
G คือเมทริกซ์ตัวกำเนิด

1.1 ชนิดของรหัสบล็อกเชิงเส้น

รหัสบล็อกเชิงเส้นสามารถแบ่งเป็นรหัสที่เป็นระบบ (Systematic code) และรหัสที่ไม่เป็นระบบ (Nonsystematic code) รหัสที่เป็นระบบนั้นทุกคำรหัสที่ได้จะมีข้อมูลต้นฉบับประกอบอยู่ในบล็อก แต่รหัสที่ไม่เป็นระบบคำรหัสจะไม่มีข้อมูลต้นฉบับเป็นส่วนประกอบ ดังตัวอย่างในตารางที่ 1

ตารางที่ 1 คำรหัสของรหัสบล็อกเชิงเส้น (7, 4)

บิตข้อมูล	คำรหัส	
	รหัสที่เป็นระบบ	รหัสที่ไม่เป็นระบบ
0000	0000000	0000000
0001	1110001	1100110
0010	0110010	0011001
0011	1000011	1111111
0100	1010100	1110001
0101	0100101	0010111
0110	1100110	1101000
0111	0010111	0001110
1000	1101000	0110010
1001	0011001	1010100
1010	1011010	0101011
1011	0101011	1001101
1100	0111100	1000011
1101	1001101	0100101
1110	0001110	1011010
1111	1111111	0111100

ที่มา: พิสิฐ และคณะ (2552)

1.2 ระยะแฮมมิงต่ำสุด (Minimum Hamming Distance)

เป็นค่าที่ใช้ในการบ่งบอกถึงความสามารถในการตรวจจับหรือแก้ไขความผิดพลาดของรหัสบล็อกเชิงเส้น (n, k) โดย (พิสิฐ วนิชชานันท์ และคณะ, 2552)

$$d_{\min} \leq n-k+1 \tag{2}$$

ตัวอย่างรหัสบล็อกได้แก่รหัสวนเช่นรหัสแฮมมิง (Hamming code) รหัสบีซีเอช และรหัสรีดโซโลมอนเป็นต้น

1.3 รหัสบีซีเอช

รหัสบีซีเอชถูกคิดค้นโดย (Hocquenghem, 1959) และ (Bose and Chaudhuri, 1960) จึงมีการนำชื่อของทั้งสามคนมารวมกันกลายเป็นรหัสบีซีเอช โดยรหัสบีซีเอชจัดเป็นหนึ่งในรหัสวนแบบไบนารีชนิดหนึ่งที่นิยมใช้งาน การออกแบบรหัสไบนารีบีซีเอชสำหรับ m ที่มีจำนวนเต็มบวกใด $(m \geq 3)$ และ $t(t < 2^{m-1})$ มีพารามิเตอร์ที่สำคัญดังนี้ (พิสิฐ วนิชชานันท์ และคณะ, 2552)

ความยาวคำรหัส $n = 2^m - 1$ (3)

ความยาวพาริตีบิต $n-k \leq mt$ (4)

ระยะแฮมมิงต่ำสุด $d_{\min} \geq 2t+1$ (5)

โดยที่ t คือจำนวนความผิดพลาดที่สามารถแก้ไขได้ และมีพหุนามตัวกำเนิด g(x) ที่ $GF(2^m)$ ได้แก่ $\alpha, \alpha^2, \alpha^3, ..., \alpha^{2t}$ เป็นราก และสามารถสร้างพหุนามตัวกำเนิดได้จาก

$$g(x) = LCM\{\phi_1(x), \phi_2(x), ..., \phi_{2t}(x)\} \tag{6}$$

โดยที่ ϕ_i คือพหุนามต่ำสุดของ α^i สำหรับ $1 \leq i \leq 2t$ เมื่อ i เป็นจำนวนเต็มคู่ และ i' เป็นจำนวนเต็มคี่ และ l เป็นจำนวนเต็มที่มากกว่าหรือเท่ากับหนึ่ง จะได้ว่า

$$i = i'2^l \tag{7}$$

ซึ่งจะได้ว่า $\alpha^i = \alpha^{i'2^l}$ ดังนั้น α^i และ $\alpha^{i'}$ จะมีพหุนามต่ำสุดเท่ากันดังนี้

$$\phi_i(x) = \phi_{i'}(x) \tag{8}$$

1.3 รหัสรีดโซโลมอน

รหัสรีดโซโลมอนถูกคิดค้นโดย (Reed and Solomon, 1960) จัดเป็นรหัสวนแบบนอนไบนารีที่นิยมใช้อย่างแพร่หลายเช่นกัน ยิ่งไปกว่านั้นในหลายปีที่ผ่านมายังมีผู้วิจัยและนำเสนอรหัสรีดโซโลมอนในแนวทางต่างๆ ทั้งการเพิ่มประสิทธิภาพ ตลอดจนถึงการคงประสิทธิภาพไว้แต่ลดความซับซ้อนในการทำงานและการนำไปใช้จริง โดยถูกใช้เป็นหลายมาตรฐานในการสื่อสารแบบดิจิทอลในปัจจุบัน ได้แก่ เทคโนโลยี 3 จี(3 Genertion: 3G) (Ryu *et al.*, 2006), ที-ดีเอ็มบี (Terrestrial-digital multimedia broadcasting: T-DMB) (Nguyen *et al.*, 2007) และพีแอลซี (Power-line channel: PLC) (Chuah, 2009)

รหัสรีดโซโลมอนมักถูกกำหนดให้อยู่ในรูปแบบของกาลัว (Galois field) โดยมีความยาวของคำรหัสทั้งหมด $n = 2^m - 1$ สัญลักษณ์ และมีข้อมูลทั้งหมด k สัญลักษณ์ โดยในแต่ละสัญลักษณ์ประกอบด้วยข้อมูล m บิต รหัสรีดโซโลมอนมีคุณสมบัติพิเศษคือมีระยะแฮมมิงต่ำสุด ($d_{\min}$) สูงสุดเท่าที่จะเป็นไปได้ ทำให้มีประสิทธิภาพในการแก้ไขความผิดพลาดสูง มีความสามารถในการจัดการตั้งแต่ความผิดพลาดแบบต่อเนื่องจนไปถึงความผิดพลาดแบบสุ่มโดยสามารถคำนวณความสามารถในการแก้ไขความผิดพลาดของรหัสรีดโซโลมอนได้จากสมการ (9)

$$t = \frac{n-k}{2} \tag{9}$$

การคำนวณหาพหุนามตัวกำเนิดของรหัสรีดโซโลมอนสามารถทำได้โดย

$$g(x)=(x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{2t}) \quad (10)$$

เมื่อได้พหุนามตัวกำเนิดแล้วสามารถนำไปสร้างเป็นคำรหัสได้

ด้านตัวถอดรหัสรีดโซโลมอนถือเป็นสิ่งสำคัญสำหรับรหัสรีดโซโลมอน เนื่องจากมีผู้นำเสนอและพัฒนามากมายตลอดช่วงหลายปีที่ผ่านมา เพื่อเพิ่มประสิทธิภาพในการถอดรหัส ลดความซับซ้อนของการถอดรหัส และเพิ่มความเร็วในการถอดรหัส รวมถึงคำนึงถึงการนำไปพัฒนาเป็นชิ้นงานต้นแบบอีกด้วย โดยมีวิธีการถอดรหัสอยู่หลายวิธี ทั้งการถอดรหัสโดยตัดสินใจแบบหยาบ (Hard Decision) และแบบการตัดสินใจแบบอ่อน แน่นอนว่าการถอดรหัสด้วยการตัดสินใจแบบอ่อนมีประสิทธิภาพที่ดีกว่า โดยการถอดรหัสด้วยการตัดสินใจแบบอ่อนของรหัสรีดโซโลมอนที่ให้ประสิทธิภาพที่ดีที่สุดคือการใช้ผลลัพธ์ที่ควรจะเป็นมากที่สุด (Maximum Likelihood Decoding: MLD) อย่างไรก็ตาม ความซับซ้อนของการถอดรหัสด้วยการใช้ผลลัพธ์ที่ควรจะเป็นมากที่สุด สำหรับรหัสรีดโซโลมอนมีการเพิ่มขึ้นแบบเอ็กโพแนนเชียล (Exponential) กับขนาดของสัญลักษณ์ กล่าวคือถ้าสัญลักษณ์มีขนาดใหญ่ การถอดรหัสก็จะยิ่งซับซ้อนมากขึ้น ดังนั้นการใช้ผลลัพธ์ที่ควรจะเป็นมากที่สุด กับรหัสรีดโซโลมอนที่มีสัญลักษณ์ขนาดใหญ่ในทางปฏิบัติจึงเป็นไปไม่ได้ ดังนั้นจึงต้องใช้การถอดรหัสแบบอื่นที่มีความซับซ้อนไม่มากนักและประสิทธิภาพใกล้เคียงกับการถอดรหัสด้วยการใช้ผลลัพธ์ที่ควรจะเป็นมากที่สุด สองตัวถอดรหัสที่ตัดสินใจแบบอ่อนที่ได้รับความนิยมมากที่สุดคือ Chase Decoding and Generalised และ Minimum Distance Decoding (GMD) และยังมีการถอดรหัสแบบ Sphere Decoding (Shayegh and Soleymani, 2011) และ Adaptive BP Algorithm (Xia and Cruz, 2007) ที่เพิ่มความเร็วพร้อมทั้งลดความซับซ้อนในการถอดรหัส

2. รหัสคอนโวลูชัน

รหัสคอนโวลูชันถูกพัฒนาขึ้นโดย (Elias, 1955) เป็นรหัสที่มีความจำอยู่ m ตัว โดยสามารถเขียนได้เป็น (n, k, m) หรือเขียนเป็นอัตรา k/n เหมือนรหัสบล็อกก็ได้ โดย k เป็นความยาวของข้อมูลและ n เป็นความยาวของคำรหัส ความแตกต่างของรหัสคอนโวลูชันและรหัสบล็อกคือรหัสบล็อกจะแบ่งข้อมูลออกเป็นหลายๆบล็อกที่มีจำนวนเท่ากัน และเข้ารหัสทั้งบล็อก แต่รหัสคอนโวลูชันสามารถเข้ารหัสข้อมูลโดยไม่ต้องแบ่งหรือจำกัดความยาวข้อมูล รหัสคอนโวลูชันสามารถสร้างได้ทั้งรหัสแบบไบนารีและรหัสแบบนอนไบนารีดังนี้

2.1 รหัสแบบไบนารี

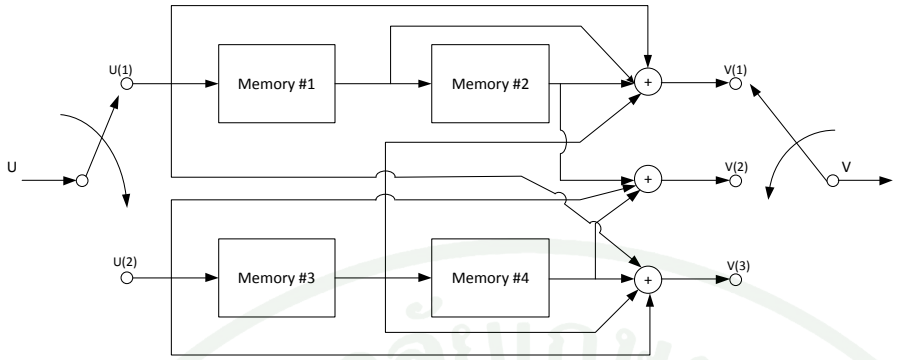
ในการอธิบายขอใช้ตัวอย่างรหัสคอนโวลูชัน (3, 2, 2) ซึ่งมีเมตริกซ์ตัวกำเนิด G(D) ดังนี้

$$\boldsymbol{G}(D)=\begin{bmatrix} 1+D+D^2 & D^2 & 1 \\ D & 1+D^2 & 1+D+D^2 \end{bmatrix} \tag{11}$$

และเขียนให้อยู่ในรูปของโดเมนแปลงได้ดังนี้ (พิสิฐ วนิชชานันท์ และคณะ, 2552)

$$\boldsymbol{G}(D)=\begin{bmatrix} g_0^{(0)}(D) & g_0^{(1)}(D) & \cdots & g_0^{(n-1)}(D) \\ g_1^{(0)}(D) & g_1^{(1)}(D) & \cdots & g_1^{(n-1)}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1}^{(0)}(D) & g_{k-1}^{(1)}(D) & \cdots & g_{k-1}^{(n-1)}(D) \end{bmatrix} \tag{12}$$

ตัวเข้ารหัสมีจำนวนขาเข้า k=2 และคำรหัส n=3 หน่วยความจำ m=2 และสามารถเขียนเป็นลักษณะของไดอะแกรมได้ดังภาพที่ 4



ภาพที่ 4 ไดอะแกรมตัวเข้ารหัสแบบคอนโวลูชัน (3, 2, 2)

กำหนดให้ลำดับข้อมูลขาเข้าเป็นเวกเตอร์ $\boldsymbol{u}$ เขียนให้อยู่ในรูปของสมการได้ดังนี้

$$\boldsymbol{u}=(u_0^{(1)}u_0^{(2)},u_1^{(1)}u_1^{(2)},u_2^{(1)}u_2^{(2)},u_3^{(1)}u_3^{(2)},...) \tag{13}$$

หรือ

$$\boldsymbol{u}^{(1)}=(u_0^{(1)},u_1^{(1)},u_2^{(1)},...) \tag{14}$$

$$\boldsymbol{u}^{(2)}=(u_0^{(2)},u_1^{(2)},u_2^{(2)},...) \tag{15}$$

สามารถเขียนให้อยู่ในรูปเมตริกซ์ได้เป็น

$$\boldsymbol{U}(D)=\left[\boldsymbol{u}^{(1)}(D)\quad \boldsymbol{u}^{(2)}(D)\right] \tag{16}$$

สำหรับคำรหัสเป็นเวกเตอร์ $\boldsymbol{v}$ สามารถเขียนให้อยู่ในรูปของสมการได้ดังนี้

$$\boldsymbol{v}=(v_0^{(1)}v_0^{(2)}v_0^{(3)},v_1^{(1)}v_1^{(2)}v_1^{(3)},v_2^{(1)}v_2^{(2)}v_2^{(3)},...) \tag{17}$$

หรือ

$$\boldsymbol{v}^{(1)} = \left(v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \ldots \right) \tag{18}$$

$$\boldsymbol{v}^{(2)} = \left(v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \ldots \right) \tag{19}$$

$$\boldsymbol{v}^{(3)} = \left(v_0^{(3)}, v_1^{(3)}, v_2^{(3)}, \ldots \right) \tag{20}$$

โดยการหาเมตริกซ์คำรหัส $\boldsymbol{V}(D)$ สามารถคำนวณหาได้จาก

$$\boldsymbol{V}(D) = \boldsymbol{U}(D) \cdot \boldsymbol{G}(D) \tag{21}$$

เมื่อนำสมการที่ (11) และ (16) มาแทนค่าลงในสมการที่ (21) จะได้

$$\boldsymbol{V}(D) = \left[\boldsymbol{U}^{(1)}(D) \quad \boldsymbol{U}^{(2)}(D) \right] \cdot \begin{bmatrix} 1+D+D^2 & D^2 & 1 \\ D & 1+D^2 & 1+D+D^2 \end{bmatrix} \tag{22}$$

จากสมการที่ (22) จะได้

$$\boldsymbol{v}^{(1)}(D) = \boldsymbol{u}^{(1)}(D) + \boldsymbol{u}^{(1)}(D) \cdot D + \boldsymbol{u}^{(1)}(D) \cdot D^2 + \boldsymbol{u}^{(2)}(D) \cdot D \tag{23}$$

$$\boldsymbol{v}^{(2)} = \boldsymbol{u}^{(1)}(D) \cdot D^2 + \boldsymbol{u}^{(2)}(D) + \boldsymbol{u}^{(2)}(D) \cdot D^2 \tag{24}$$

$$\boldsymbol{v}^{(3)} = \boldsymbol{u}^{(1)}(D) + \boldsymbol{u}^{(2)}(D) + \boldsymbol{u}^{(2)}(D) \cdot D + \boldsymbol{u}^{(2)}(D) \cdot D^2 \tag{25}$$

เนื่องจาก n เป็น 3 จึงทำให้ได้เวกเตอร์คำรหัส

$$\boldsymbol{v}(D) = \boldsymbol{v}^{(1)}(D^3) + D \cdot \boldsymbol{v}^{(2)}(D^3) + D^2 \cdot \boldsymbol{v}^{(3)}(D^3) \tag{26}$$

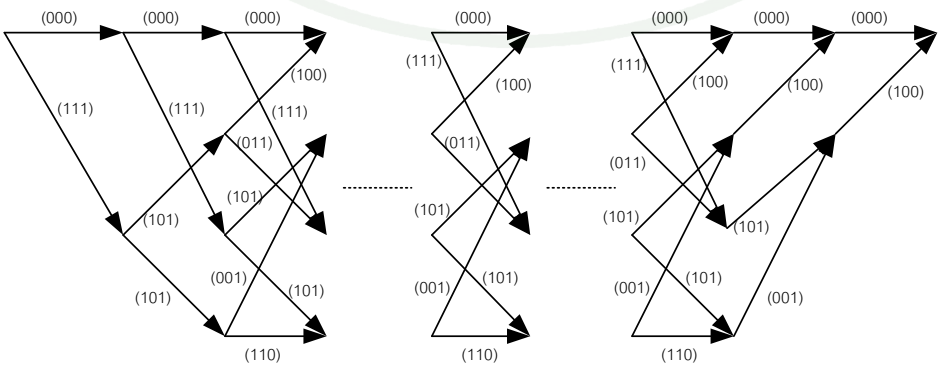
สมการที่ (26) สามารถนำไปคำนวณทางคณิตศาสตร์เพื่อหาคำรหัสได้

2.2 รหัสแบบนอนไบนารี

การเข้ารหัสคอนโวลูชันแบบนอนไบนารีมีความแตกต่างกับการเข้ารหัสคอนโวลูชันแบบไบนารีเล็กน้อยคือส่วนของข้อมูล $\boldsymbol{U}$ ที่ต้องการเข้ารหัสใช้เป็นแบบนอนไบนารี เช่นข้อมูลที่เข้ารหัส $\boldsymbol{U}$ มีขนาด 102 บิตต่อสัญลักษณ์

3. วีเทอร์บี

วีเทอร์บีเป็นกระบวนการถอดรหัสช่องสัญญาณ ถูกคิดค้นโดย (Viterbi, 1967) ถือเป็นวิธีถอดรหัสที่เหมาะสมที่สุดสำหรับรหัสคอนโวลูชันแบบไบนารี โดยใช้หลักการของการถอดรหัสเพื่อให้ได้ผลลัพธ์ที่ควรจะเป็นมากที่สุด ซึ่งการถอดรหัสวีเทอร์บีนั้นสามารถแบ่งได้ 2 ลักษณะด้วยกันคือการถอดรหัสด้วยการตัดสินใจแบบหยาบและการถอดรหัสด้วยการตัดสินใจแบบอ่อน โดยการถอดรหัสด้วยการตัดสินใจแบบหยาบเป็นการถอดรหัสที่พิจารณาข้อมูลที่ได้รับซึ่งมี 2 ระดับคือบิต 0 หรือบิต 1 เท่านั้น แตกต่างกับการถอดรหัสด้วยการตัดสินใจแบบอ่อนที่มีการคำนวณที่ซับซ้อนกว่า กล่าวคือข้อมูลที่ได้รับมีการแบ่งระดับสัญญาณมากกว่า 2 ระดับ ทำให้มีรายละเอียดในการพิจารณามากกว่าการถอดรหัสด้วยการตัดสินใจแบบหยาบ ดังนั้นการถอดรหัสด้วยการตัดสินใจแบบอ่อนจึงมีประสิทธิภาพที่ดีกว่า และในการตัดสินใจทั้งสองแบบยังต้องใช้แผนภาพเทรลลิส (Trillis) ร่วมด้วย



ภาพที่ 5 โครงสร้างแผนภาพเทรลลิสในสถานะในช่วงเริ่มต้น ช่วงสถานะคงตัว และช่วงสถานะสิ้นสุด

ที่มา: พิสิฐ และคณะ (2552)

จากภาพที่ 5 สามารถแบ่งแผนภาพเทรลลิสออกเป็น 3 สถานะได้แก่สถานะเริ่มต้น สถานะคงตัวและสถานะสิ้นสุด

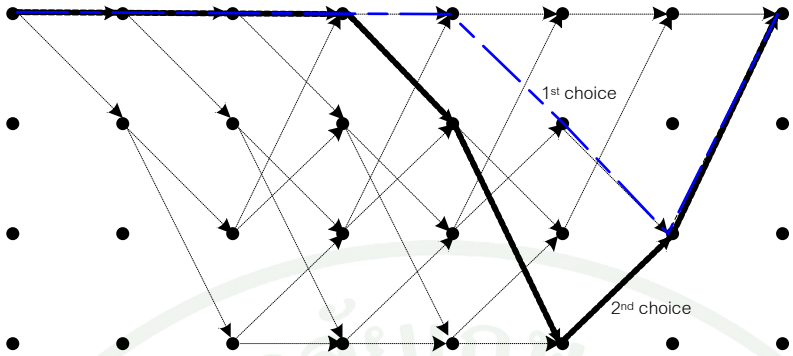
การถอดรหัสคอนโวลูชันไม่ว่าจะใช้การถอดรหัสด้วยการตัดสินใจแบบหยาบหรือการถอดรหัสด้วยการตัดสินใจแบบอ่อน จะนำคำรหัสที่ได้มาใช้แผนภาพเทรลลิสในการพิจารณาหาเส้นทางที่ดีที่สุดหรือชุดคำรหัสที่ดีที่สุด

การถอดรหัสด้วยวีเทอร์บีจะมีรูปแบบความผิดพลาดที่เป็นเอกลักษณ์ กล่าวคือตำแหน่งบิตต้นและบิตปลายของคำรหัสจะมีโอกาสเกิดความผิดพลาดน้อยกว่าตำแหน่งบิตที่อยู่กลางคำรหัสอยู่ประมาณ 30 เปอร์เซ็นต์ตามงานวิจัยของ (Wei and Aulin, 1997)

4. ลิสวีเทอร์บี

ลิสวีเทอร์บีถูกนำเสนอโดย (Seshadri and Sungberg, 1994) เป็นการนำการถอดรหัสแบบวีเทอร์บีมาประยุกต์ใช้โดยนอกจากการเลือกเส้นทางหรือคำรหัสที่ดีที่สุดแล้วยังเลือกพิจารณาเส้นทางหรือคำรหัสที่ดีรองลงมาในการนำไปใช้ในการถอดรหัสเพื่อเพิ่มประสิทธิภาพของการเข้ารหัสช่องสัญญาณ ดังนั้นกล่าวโดยสรุปคือการถอดรหัสแบบลิสวีเทอร์บีสามารถให้ผลลัพธ์ในการถอดรหัสมากกว่าหนึ่งผลลัพธ์นั่นเอง

ตัวถอดรหัสลิสวีเทอร์บีแบ่งได้ 2 แบบได้แก่ลิสวีเทอร์บีแบบขนาน (Parallel LVA) และลิสวีเทอร์บีแบบอนุกรม (Serial LVA) โดยลิสวีเทอร์บีแบบขนานจะทำการคำนวณ L เส้นทางที่ดีที่สุดในแต่ละสเตจ (State) แต่ลิสวีเทอร์บีแบบอนุกรมเริ่มต้นจากการคำนวณหาผลลัพธ์ที่ดีที่สุดจากนั้นจึงคำนวณหาเส้นทางที่ดีรองลงมา



ภาพที่ 6 ผลลัพธ์จากการถอดรหัสแบบลิสวีเทอร์บี

จากภาพที่ 6 เป็นการถอดรหัสลิสวิเทอร์บีแบบสองผลลัพธ์ โดยเส้นประจะเป็นเส้นที่มีผลลัพธ์ที่ดีที่สุด และเส้นทึบเป็นเส้นที่มีผลลัพธ์ที่ดีรองลงมา

5. การถอดรหัสแบบเวกเตอร์ซิมโบล

การถอดรหัสแบบเวกเตอร์ซิมโบลถูกนำเสนอโดย (Metzner and Kapturowski, 1990) ซึ่งเหมาะสำหรับใช้กับรหัสที่เป็นนอนไบนารีขนาดใหญ่ สามารถถอดรหัสได้ทั้งรหัสบล็อกเชิงเส้นและรหัสคอนโวลูชัน ทำให้ง่ายต่อการนำไปใช้งาน โดยจุดเด่นคือสามารถแก้ไขความผิดพลาดที่เกิดขึ้นแบบเบริสต์ได้ดี และสามารถใช้คู่กับตัวถอดรหัสภายในแบบมีหลายทางเลือก เช่นลิสวีเทอร์บี เพื่อเพิ่มประสิทธิภาพในการถอดรหัสช่องสัญญาณ การถอดรหัสจะใช้เมตริกซ์พาริตีเช็ก (Parity Check Matrix) เช่นเดียวกับการถอดรหัสแบบบล็อกเชิงเส้นดังนี้ (พิสิฐ วนิชชานันท์ และคณะ, 2552)

$$\boldsymbol{S} = \boldsymbol{r} \cdot \boldsymbol{H}^{T} \tag{27}$$

$$\boldsymbol{S} = (\boldsymbol{c} + \boldsymbol{e}) \cdot \boldsymbol{H}^{T} = \boldsymbol{e} \cdot \boldsymbol{H}^{T} \tag{28}$$

$\boldsymbol{S}$ คือค่าของเมตริกซ์ซินโดรม

$\boldsymbol{r}$ คือเวกเตอร์รหัสที่รับมา

$\boldsymbol{H}$ คือเมตริกซ์พาริตีเช็ก

$\boldsymbol{c}$ คือเวกเตอร์คำรหัส

$\boldsymbol{e}$ คือเวกเตอร์ความผิดพลาด

หากคำนวณได้ค่าเมตริกซินโดรม $\boldsymbol{s}$ เป็น 0 แสดงว่าไม่เกิดความผิดพลาดในการส่งข้อมูลผ่านช่องสัญญาณ แต่ถ้าเกิดความผิดพลาดในการส่งข้อมูล ค่าเมตริกซินโดรม $\boldsymbol{s}$ ที่คำนวณได้จะไม่เป็น 0

เมตริกซ์พาริตีเช็กสำหรับตัวเข้ารหัสแบบบล็อกเชิงเส้นสามารถคำนวณด้วยวิธีเดียวกับการถอดรหัสบล็อกดังนี้

$$\boldsymbol{G} = \left[P \mid I_k \right] \tag{29}$$

$$\boldsymbol{H} = \left[I_{n-k} \mid P^T \right] \tag{30}$$

เมตริกซ์พาริตีเช็กจะมีขนาด $k \cdot n$ เนื่องจากรหัสบล็อกเชิงเส้นเข้ารหัสทั้งบล็อกเท่าๆกันครั้งละ k สัญลักษณ์ และออกเป็นคำรหัส n สัญลักษณ์ หากตัวเข้ารหัสเป็นคอนโวลูชันที่ไม่สามารถทราบความยาวของข้อมูลที่จะทำการเข้ารหัสเมตริกซ์พาริตีเช็กจะถูกสร้างเป็นเมตริกซ์กึ่งอนันดังสมการที่ (31)

$$\boldsymbol{H} = \begin{bmatrix} h_0^{(1)} & h_0^{(2)} & & & & & \\ h_1^{(1)} & h_1^{(2)} & h_0^{(1)} & h_0^{(2)} & & & \\ \vdots & \vdots & h_1^{(1)} & h_1^{(2)} & h_0^{(1)} & h_0^{(2)} & \\ h_m^{(1)} & h_m^{(2)} & \vdots & \vdots & h_1^{(1)} & h_1^{(2)} & \\ & & h_m^{(1)} & h_m^{(2)} & \vdots & \vdots & \ddots \\ & & & & h_m^{(1)} & h_m^{(2)} & \\ & & & & & & \ddots \end{bmatrix} \tag{31}$$

ที่มา: Tuntoolavest (2009)

การทำงานของตัวถอดรหัสเวกเตอร์ซิมโบลแบบไม่มีข้อมูลสำรองสามารถลำดับได้ดังนี้

1. หลังจากตัวถอดรหัสได้รับบางสัญลักษณ์ ตัวถอดรหัสจะใช้ n สัญลักษณ์แรกคำนวณค่าซินโดรม

2. ถ้าค่าซินโดรมเป็นศูนย์ ตัวถอดรหัสจะตัดสินใจว่าไม่มีค่าความผิดพลาดในสัญลักษณ์ที่ได้รับมา

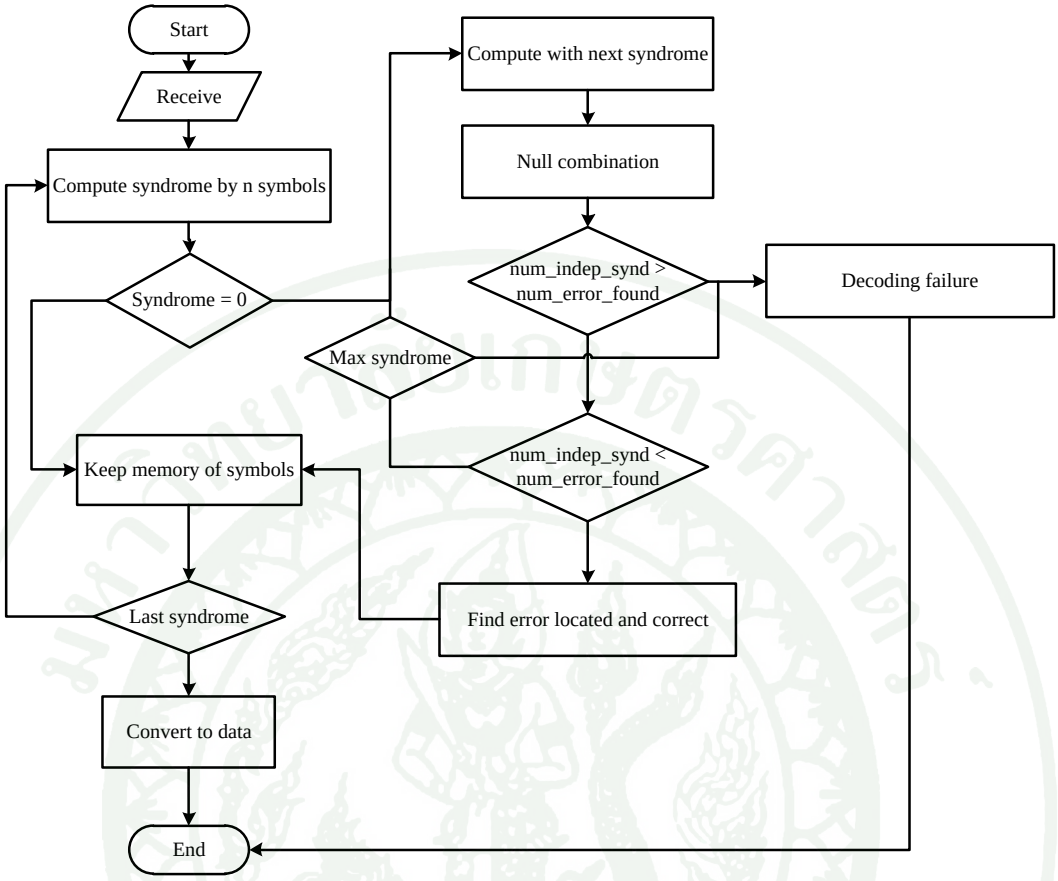
3. ตัวถอดรหัสจะเก็บหน่วยความจำของสัญลักษณ์ที่ถูกถอดรหัสไปแล้ว และใช้ในการช่วยถอดรหัสสัญลักษณ์ที่ได้รับในตัวถัดไป และย้อนกลับไปทำข้อ 1. อีกครั้ง

4. ถ้าค่าซินโดรมไม่เป็นศูนย์ ตัวถอดรหัสจะแก้ไขความผิดพลาดด้วยการใช้ชุดการรวมเป็นศูนย์ (Null Combination) ถ้าแก้ไขความผิดพลาดได้จะย้อนกลับไปทำข้อ 3. อีกครั้ง

5. ถ้าตัวถอดรหัสไม่สามารถแก้ไขความผิดพลาดได้ในข้อ 4. ตัวถอดรหัสจะร้องขอสัญลักษณ์ที่ได้รับมาเพิ่ม และแก้ไขความผิดพลาดด้วยการใช้ชุดการรวมเป็นศูนย์ ในข้อ 4. อีกครั้ง

6. ถ้าตัวถอดรหัสทำซ้ำจนถึงสัญลักษณ์สุดท้ายที่ได้รับมาแต่ยังไม่สามารถแก้ไขค่าความผิดพลาดได้ ตัวถอดรหัสจะแจ้งเตือนว่าการถอดรหัสล้มเหลว

สามารถอธิบายขั้นตอนการถอดรหัสเวกเตอร์ซิมโบลแบบไม่มีชุดข้อมูลสำรองได้ดังภาพที่ 7



ภาพที่ 7 ผังขั้นตอนการถอดรหัสเวกเตอร์ซิมโบลแบบไม่มีชุดข้อมูลสำรอง

การหาค่าซินโดรมในข้อ 1. หาได้จากสมการที่ (32)

$$\boldsymbol{S} = \boldsymbol{r} \cdot \boldsymbol{H}^{T} \tag{32}$$

โดย $\boldsymbol{S}$ คือค่าซินโดรม $\boldsymbol{H}$ คือเมตริกซ์พาริตีเช็กกึ่งอนันต์ และ $\boldsymbol{r}$ คือชุดคำรหัสที่ได้รับ ซึ่งจากสามารที่ (32) พบว่าค่าซินโดรมเป็นศูนย์ เมื่อไม่มีค่าความผิดพลาดในข้อมูลที่ได้รับ

ค่าเมตริกซ์พาริตีเช็กกึ่งอนันต์ของรหัสคอนโวลูชัน (3, 2, 2) คือ

$$\boldsymbol{H}=\begin{bmatrix} 1 & 1 & 1 & & & & \\ 0 & 1 & 1 & 1 & 1 & 1 & \\ 0 & 1 & 0 & 0 & 1 & 1 & \ddots \\ 1 & 0 & 0 & 0 & 1 & 0 & \\ 1 & 1 & 1 & 1 & 0 & 0 & \\ & & & 1 & 1 & 1 & \\ & & & & & & \ddots \end{bmatrix} \tag{33}$$

และข้อมูลที่ได้รับคือ

$$\boldsymbol{r}=(\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{r}_3,\boldsymbol{r}_4\boldsymbol{r}_5\boldsymbol{r}_6,\boldsymbol{r}_7\boldsymbol{r}_8\boldsymbol{r}_9,\ldots,\boldsymbol{r}_{3(k+2)-2}\ \boldsymbol{r}_{3(k+2)-1}\ \boldsymbol{r}_{3(k+2)}) \tag{34}$$

ให้ $\boldsymbol{r}$ แต่ละตัวแทนค่าเวกเตอร์ของสัญลักษณ์ขนาด q บิต ซึ่งงานวิจัยนี้พัฒนาให้สามารถเปลี่ยนค่า q ได้จนถึง 128 บิต และใช้สมการที่ (32) (33) และ (34) ในการคำนวณหาค่าซินโดรม โดยใช้ n สัญลักษณ์แรกที่ได้รับดังสมการที่ (35)

$$\boldsymbol{S}=\left[\boldsymbol{s}_1\right]=\boldsymbol{r}\cdot\boldsymbol{h}_1^{\,T}=\left[r_1\ r_2\ r_3\right]\cdot\left[1\ 1\ 1\right]^T \tag{35}$$

หากข้อมูล n สัญลักษณ์แรกไม่สามารถถอดรหัสได้จะใช้ข้อมูล n สัญลักษณ์ถัดไปเพื่อมาคำนวณหาค่าซินโดรม $\boldsymbol{s}_2$ เพื่อเพิ่มข้อมูลซินโดรมในกระบวนการชุดการรวมเป็นศูนย์ในการตรวจสอบ

$$\boldsymbol{s}_2=\boldsymbol{r}\cdot\boldsymbol{h}_2^{\,T}=\left[r_1\ r_2\ r_3\ r_4\ r_5\ r_6\right]\cdot\left[0\ 1\ 1\ 1\ 1\ 1\right]^T \tag{36}$$

และ

$$\boldsymbol{S} = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T \tag{37}$$

หากยังไม่สามารถถอดรหัสได้ก็จะใช้ข้อมูล n สัญลักษณ์ถัดไปมาช่วยในการถอดรหัส จนกว่าจะถอดรหัสได้หรือหมดชุดคำรหัส โดยในแต่ละชุด n สัญลักษณ์สามารถนำมาคำนวณหาค่าซินโดรมได้ดังตารางที่ 2

ตารางที่ 2 การคำนวณค่าซินโดรมสำหรับรหัสคอนโวลูชัน (3, 2, 2)

พาริตีเวกเตอร์	การคำนวณค่าซินโดรม
$h_1 = [111]$	s_1 = xor ลำดับซิมโบลที่รับได้ 1,2,3
$h_2 = [011\ \ 111]$	s_2 = xor ลำดับซิมโบลที่รับได้ 2,3,4,5,6
$h_3 = [010\ 011\ 111]$	s_3 = xor ลำดับซิมโบลที่รับได้ 2,5,6,7,8,9
$h_4 = [100\ 010\ 011\ 111]$	s_4 = xor ลำดับซิมโบลที่รับได้ 1,5,8,9,10,11,12
$h_5 = [111\ 100\ 010\ 011\ 111]$	s_5 = xor ลำดับซิมโบลที่รับได้ 1,2,3,4,8,11,12,13,14,15
$h_6 = [000\ 111\ 100\ 010\ 011\ 111]$	s_6 = xor ลำดับซิมโบลที่รับได้ 4,5,6,7,11,14,15,16,17,18
$h_7 = [000\ 000\ 111\ 100\ 010\ 011\ 111]$	s_7 = xor ลำดับซิมโบลที่รับได้ 7,8,9,10,14,17,18,19,20,21

ที่มา: Tuntoolavest (2009)

จากตารางที่ 2 จะสังเกตว่า h_5, h_6, h_7 สามารถคำนวณได้ค่าซินโดรมเดียวกันเนื่องจากเข้าสู่ช่วงคงตัว และเมตริกซ์พาริตีเช็กยังใช้หลังการใช้ชุดการรวมเป็นศูนย์อีกด้วย คือสามารถนำมาใช้เป็นเครื่องมือในการหาตำแหน่งบิตที่ผิดพลาด พร้อมทั้งสามารถนำข้อมูลดังกล่าวไปใช้ในการแก้ไขความผิดพลาด

โดยการแก้ไขด้วยชุดการรวมเป็ยศูนย์นั้นเป็นวิธีที่มีความซับซ้อนแต่มีประสิทธิภาพสูง สามารถแก้ไขชุดของสัญลักษณ์ที่มีรูปแบบความผิดพลาดที่ซับซ้อนบางรูปแบบได้ โดยไม่ต้องอาศัยข้อมูลสำรองมาช่วย การถอดรหัสด้วยชุดการรวมเป็นศูนย์สามารถแบ่งได้ 3 ขั้นตอนดังนี้

1. การหาตัวบ่งชี้เป็นศูนย์และผลรวมทางพีชคณิตได้ศูนย์

นำ $\boldsymbol{S}$ มาทำ Gauss-Jordan column operation จากนั้นนำแต่ละแถวของ $\boldsymbol{S}$ มาจับคู่เพื่อทำเอ็กซ์คลูซีฟออร์

ตัวอย่าง **ที่มา**: พิสิฐ วนิชชานันท์ และคณะ (2552)

$$\boldsymbol{S} = \begin{bmatrix} 11100 \\ 11010 \\ 01101 \\ 01101 \\ 00000 \\ 01101 \\ 01101 \end{bmatrix} \Rightarrow \begin{bmatrix} 10100 \\ 10010 \\ 01101 \\ 01101 \\ 00000 \\ 01101 \\ 01101 \end{bmatrix} \Rightarrow \begin{bmatrix} 10000 \\ 11010 \\ 01101 \\ 01101 \\ 00000 \\ 01101 \\ 01101 \end{bmatrix} \Rightarrow \begin{bmatrix} 10000 \\ 01010 \\ 01101 \\ 01101 \\ 00000 \\ 01101 \\ 01101 \end{bmatrix} \Rightarrow \begin{bmatrix} 10000 \\ 01000 \\ 01101 \\ 01101 \\ 00000 \\ 01101 \\ 01101 \end{bmatrix} \Rightarrow \begin{bmatrix} 10000 \\ 01000 \\ 00100 \\ 00100 \\ 00000 \\ 00100 \\ 00100 \end{bmatrix}$$

จากตัวอย่างมีสามแถวที่เป็นอิสระเชิงเส้นต่อกัน และเวกเตอร์ซินโดรมแถวที่สาม-สี่ สาม-หก และสาม-เจ็ด เอ็กซ์คลูซีฟออร์ได้ศูนย์ รวมถึงแถวที่ห้าที่ตัวมันเองเป็นศูนย์ด้วย

2. การหาเวกเตอร์ระบุตำแหน่งที่ผิดพลาด

หากมีคู่ใดทำแล้วได้ศูนย์(ตัวบ่งชี้) ให้ใช้แถวนั้นๆในเมตริกซ์พาริตีเช็กมาเอ็กซ์คลูซีฟออร์กัน จากนั้นนำชุดเวกเตอร์ทั้งหมดมาออร์(OR) กัน จะได้เวกเตอร์แสดงตำแหน่งสัญลักษณ์ที่ผิดพลาด

ตัวอย่าง **ที่มา**: พิสิฐ วนิชชานันท์ และคณะ (2552)

$\boldsymbol{h}_3 + \boldsymbol{h}_4 = [0000011000]$
$\boldsymbol{h}_3 + \boldsymbol{h}_6 = [0110010010]$
$\boldsymbol{h}_3 + \boldsymbol{h}_7 = [0110010001]$
$\boldsymbol{h}_5 = [0110000100]$

ผลลัพธ์การออร์เวกเตอร์ทั้งสี่ คือ [0110011111] ซึ่งบิต “0” แสดงตำแหน่งสัญลักษณ์ที่ผิด คือตำแหน่งที่ 1 4 และ 5

3. หาค่าความผิดพลาดจากสมการที่ (38)

$$\boldsymbol{E} = \boldsymbol{H}^{-1} \cdot \boldsymbol{S} \tag{38}$$

แต่เราไม่สามารถคำนวณจากสมการที่ (38) ได้ตรงๆ เนื่องจากเมตริกซ์พาริตีเช็กมีขนาด $(n-k)\cdot n$ ซึ่งไม่เป็นเมตริกซ์จัตุรัส และในทางปฏิบัติมีวิธีที่ง่ายกว่าที่สามารถหาค่าความผิดพลาดโดยการลดขนาดของ $\boldsymbol{H}$ และ $\boldsymbol{S}$ ให้เล็กลง โดยดึงส่วนที่ต้องใช้ในการหาค่าความผิดพลาดมาใช้เท่านั้น ซึ่งจะได้เมตริกซ์ย่อย (Submatrix) ที่เป็นเมตริกซ์จัตุรัส แล้วจึงทำการหาเมตริกซ์ผกผัน เมื่อใช้เมตริกซ์ย่อยของพาริตีเช็กแล้วก็ต้องใช้เมตริกซ์ย่อยของซินโดรมด้วย จะได้เป็นสมการที่ (39)

$$\boldsymbol{E}_{sub} = \boldsymbol{H}_{sub}^{-1} \cdot \boldsymbol{S}_{sub} \tag{39}$$

และจากตัวอย่างสามารถหาค่าความผิดพลาดได้โดยตำแหน่งบิต “0” ในเวกเตอร์แสดงตำแหน่งสัญลักษณ์ที่ผิดจะกำหนดหลักของเมตริกซ์พาริตีเช็กที่จะอยู่ในเมตริกซ์พาริตีเช็กย่อย และตำแหน่งของแถวของเมตริกซ์ซินโดรมหลังทำ Gauss-Jordan column operation ที่เป็นอิสระต่อกันจะถูกเลือกให้อยู่ในเมตริกซ์ซินโดรมย่อย และจะเป็นตัวกำหนดแถวของเมตริกซ์พาริตีเช็กที่จะอยู่ในเมตริกซ์พาริตีเช็กย่อย

ดังนั้นเมตริกซ์พาริตีเช็กย่อยจะประกอบไปด้วย

$$\boldsymbol{H}_{sub} = \begin{bmatrix} 010 \\ 101 \\ 100 \end{bmatrix} \tag{40}$$

สามารถหาเมตริกซ์ผกผันได้เป็น

$$\boldsymbol{H}_{sub}^{-1} = \begin{bmatrix} 001 \\ 100 \\ 011 \end{bmatrix} \tag{41}$$

และ

$$\boldsymbol{S}_{sub} = \begin{bmatrix} 11100 \\ 11010 \\ 01001 \end{bmatrix} \tag{42}$$

ดังนั้นนำสมการที่ (40) และ (42) แทนค่าในสมการที่ (39) จะได้ว่า

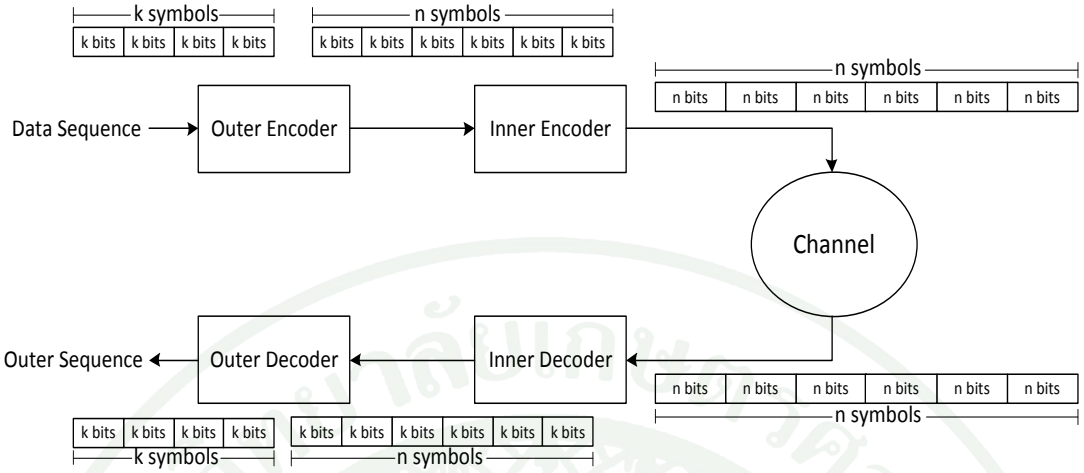
$$\boldsymbol{E}_{sub} = \begin{bmatrix} 011 \\ 100 \\ 011 \end{bmatrix} \begin{bmatrix} 11100 \\ 11010 \\ 01001 \end{bmatrix} = \begin{bmatrix} 01001 \\ 11100 \\ 10011 \end{bmatrix} \quad (43)$$

จากผลลัพธ์ที่ได้ทำให้ทราบว่าความผิดพลาดของสัญลักษณ์ที่หนึ่ง สัญลักษณ์ที่สี่และสัญลักษณ์ที่ห้ามีค่า [01001] [11100] และ [10011] ตามลำดับ ดังนั้นเมื่อทราบตำแหน่งและค่าความผิดพลาดแล้ว ก็สามารถแก้ไขให้ถูกต้องได้

6. รหัสคอนคาทิเนต

รหัสคอนคาทิเนตหรือรหัสต่อร่วม เป็นการนำรหัสช่องสัญญาณมากกว่าหนึ่งรหัสมาต่อร่วมกัน อาจต่อแบบอนุกรมหรือต่อแบบขนาน รหัสคอนคาทิเนตอย่างง่ายประกอบด้วยสองส่วนต่ออนุกรมกัน ประกอบด้วยรหัสภายนอก และรหัสภายใน รหัสภายนอกมักใช้รหัสนอนไบนารีโดยทั่วไปนิยมใช้เป็นรหัสรีดโซโลมอน ส่วนรหัสภายในมักใช้รหัสไบนารี อาจใช้รหัสบีซีเอชหรือรหัสคอนโวลูชัน

ลักษณะการทำงานของรหัสคอนคาทิเนตคือรหัสภายในมีหน้าที่แก้ไขความผิดพลาดแบบสุ่ม กล่าวคือจะช่วยแก้ไขความผิดพลาดในสัญลักษณ์ กรณีที่ความผิดพลาดมีการกระจายตัว ดังนั้นหากสัญลักษณ์ใดมีความผิดพลาดน้อย อาจแก้ไขให้ถูกหมดได้ แต่ถ้าสัญลักษณ์ใดมีความผิดพลาดติดต่อกันมากจะเป็นหน้าที่ในส่วนของรหัสภายนอกที่จะทำการแก้ไขความผิดพลาด เพราะรหัสภายนอกสามารถแก้ไขความผิดพลาดแบบเบริสต์ ดังนั้นรหัสคอนคาทิเนตจึงมักถูกนำไปใช้งานกับการสื่อสารบนช่องสัญญาณแบบไร้สาย เนื่องจากมักพบปัญหาการผิดพลาดแบบเบริสต์



ภาพที่ 8 แผนภาพการทำงานของรหัสคอนคาทิเนต

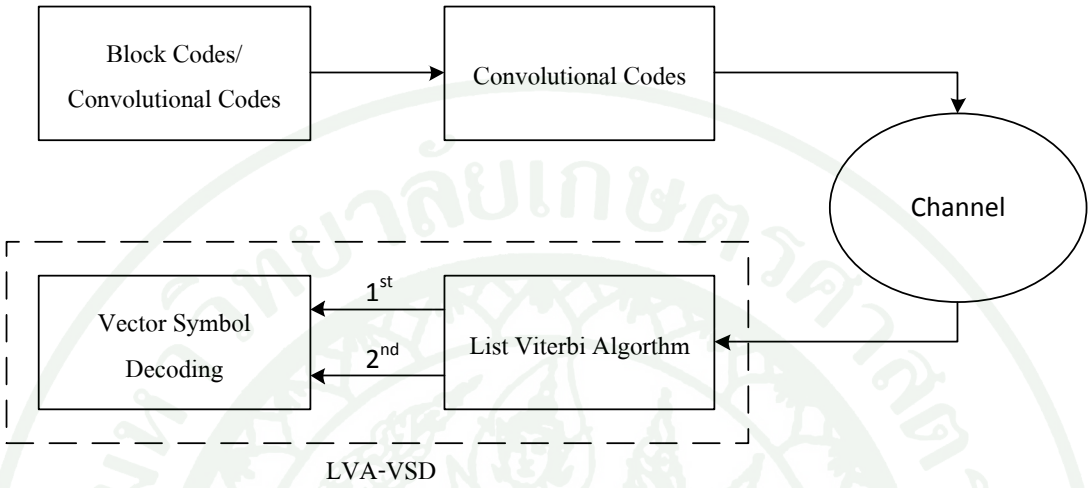
ที่มา: พิสิฐ และคณะ (2552)

ภาพที่ 8 แสดงการจัดเรียงรหัสของรหัสคอนคาทิเนต โดยมีข้อมูลเข้ารหัสภายนอกโดยจัดเป็นรหัสยาว k สัญลักษณ์ สัญลักษณ์ละ k บิต และได้เป็นคำรหัส n สัญลักษณ์ สัญลักษณ์ละ k บิต จากนั้นแบ่งรหัสยาวเป็นรหัสสั้นและเข้ารหัสภายใน จะได้คำรหัส n สัญลักษณ์ สัญลักษณ์ละ n บิต จากนั้นนำคำรหัสผ่านช่องสัญญาณ จัดเรียงรหัสและถอดรหัสเช่นเดียวกับฝั่งเข้ารหัส

7. ตัวถอดรหัส List Viterbi Algorithm - Vector Symbol Decoder (LVA-VSD)

ตัวถอดรหัส LVA-VSD ถูกนำไปประยุกต์ใช้ร่วมกับรหัสคอนคาทิเนต (Tuntoolavest and Seubnaung, 2007) โดยใช้การเข้ารหัสทั้งรหัสภายนอกและรหัสภายในเป็นรหัสคอนโวลูชัน จากนั้นตัวถอดรหัสภายในเป็นแบบลิสวิเทอร์บีแบบสองตัวเลือก ถึงแม้ลิสวิเทอร์บีจะสามารถให้ผลลัพธ์ได้มากกว่าหนึ่ง แต่จากงานวิจัย (Tuntoolavest and Seubnaung, 2007) พบว่าการนำตัวถอดรหัสลิสวิเทอร์บีไปประยุกต์ใช้งานจริง การเพิ่มจำนวนผลลัพธ์ให้มากกว่าสองจะทำให้ส่วนอุปกรณ์ (Hardware) ทำงานซับซ้อนมากยิ่งขึ้น เมื่อพิจารณาแล้วจึงสรุปว่าการนำตัวถอดรหัสแบบลิสวิเทอร์บีมาประยุกต์ใช้งานจริงเลือกใช้เพียงสองผลลัพธ์ก็เพียงพอแล้ว และตัวถอดรหัสภายนอกซึ่งใช้ตัวถอดรหัสเวกเตอร์ซิมโบลสามารถทำงานร่วมกับตัวถอดรหัสภายในที่เป็นแบบลิสวิเทอร์บี

ได้ดี เนื่องจากสองผลลัพธ์จากตัวถอดรหัสแบบลิสวิเทอร์บีช่วยเพิ่มประสิทธิภาพในการถอดรหัสของตัวถอดรหัสเวกเตอร์ซิมโบล



ภาพที่ 9 รหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสแบบ LVA-VSD

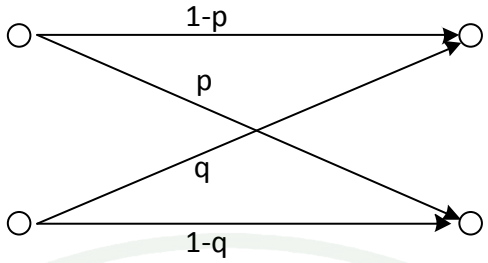
จากภาพที่ 9 แสดงให้เห็นถึงรหัสคอนคาทิเนตที่มีตัวถอดรหัสแบบ LVA-VSD โดยตัวถอดรหัสภายในสามารถถอดรหัสโดยให้สองผลลัพธ์ และรหัสภายนอกฝั่งเข้ารหัสสามารถเลือกใช้ได้ทั้งรหัสบล็อกหรือรหัสคอนโวลูชัน โดยใช้ตัวถอดรหัสเวกเตอร์ซิมโบลถอดรหัสได้ทั้งสอง

8. ช่องสัญญาณ

การจำลองช่องสัญญาณสามารถจำลองความผิดพลาดของช่องสัญญาณได้หลายแบบดังนี้

8.1 ช่องสัญญาณบีเอสซี (Binary Symmetric Channel: BSC)

เป็นช่องสัญญาณพื้นฐานในการรับส่งข้อมูล ข้อมูลที่ส่งผ่านช่องสัญญาณเป็นแบบไบนารีคือมีแค่บิต 0 และบิต 1 เท่านั้น มีความน่าจะเป็นที่จะส่งบิต 0 ผิดเท่ากับ p และมีความน่าจะเป็นที่ส่งบิต 1 ผิดเป็น q ดังภาพที่10



ภาพที่ 10 รูปแบบช่องสัญญาณแบบบีเอสซี

8.2 ช่องสัญญาณเกาท์เซียนขาวแบบบวก (AWGN)

เป็นช่องสัญญาณพื้นฐานที่นิยมใช้ในการจำลองช่องสัญญาณในงานวิจัย เกิดจากการบวกเชิงเส้นของสัญญาณรบกวนขาว (White Noise) ที่มีความหนาแน่นของสเปกทรัล (Spectral) คงที่ กับการแจงแจงแบบเกาท์เซียน (Gaussian Distribution) ของแอมพลิจูด (Amplitude) ซึ่งสัญญาณรบกวนเกาท์เซียนขาวแบบบวกสามารถสร้างจากฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability Density Function: PDF) ที่มีค่าเฉลี่ย (Mean) เป็นศูนย์ ได้ดังนี้

$$p(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}} \qquad (44)$$

โดยที่ σ^2 คือค่าความแปลปรวน (Variance), σ คือค่าเบี่ยงเบนมาตราฐาน และ x คือค่าตัวแปรสุ่ม (Random Variable)

8.3 ช่องสัญญาณการจางหาย (Fading Channel)

สามารถแบ่งได้เป็น

- การจางหายแบบช้า (slow fading)/การจางหายแบบเร็ว (fast fading)
- การจางหายแบบราบ (flat fading)/ การจายหายแบบเลือกความถี่ (frequency-selective fading)
- การจางหายมัลติพาท (multipath fading หรือ small-scale fading)/ ชาโดว์อิง (shadowing หรือ large-scale fading)

แต่ชาโดว์อิงมักไม่ใช้พิจราณาเกี่ยวกับบิต แต่จะนำไปใช้เกี่ยวกับขนาดเซลล์หรือกำลัง เนื่องจากมีขนาดใหญ่

โดยรูปแบบการจำลองช่องสัญญาณการจางหายที่นิยมคือการจางหายแบบเรย์ลี (Rayleigh Fading) และการจางหายแบบไรเซียน (Rician Fading)

8.4 ดอปเพลอร์ (Doppler)

ปรากฏการณ์ดอปเพลอร์เกิดจากการเคลื่อนที่ของตัวส่งสัญญาณหรือตัวรับสัญญาณ หรือทั้งสอง ทำให้มุมของสัญญาณที่มาถึงเปลี่ยน และทำให้คลื่นสัญญาณที่มีถึงมีความถี่เปลี่ยนไป โดยค่าความถี่ดอปเพลอร์สามารถหาได้จาก

$$f_d = \frac{v}{c_0} f_0 \cos\alpha \qquad (45)$$

เมื่อ f_d คือความถี่ดอปเพลอร์

v คือความเร็วสัมพัทธ์ระหว่างเครื่องรับกับเครื่องส่ง (เมตรต่อวินาที)

c_0 คือความเร็วแสง มีค่าเท่ากับ 3×10^8 (เมตรต่อวินาที)

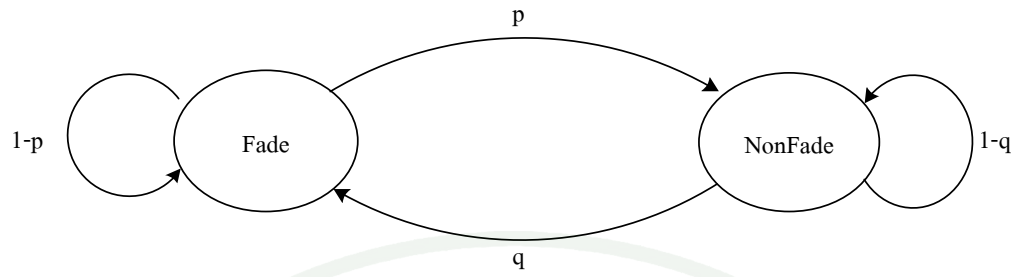
f_0 คือความถี่คลื่นพาห์

α คือมุมของสัญญาณที่มาถึง

ค่าความถี่ดอปเพลอร์จะมีค่ามากที่สุดเมื่อ α มีค่าเท่ากับศูนย์

8.4 ช่องสัญญาณแบบ 2 สเตจ (2 State Model)

ใช้แบบจำลอง 2 สเตจในการจำลองช่องสัญญาณไร้สายแบบไม่มีความจำ ที่มีความผิดพลาดติดต่อกัน หรือมีช่วงที่ช่องสัญญาณเกิดการเฟด (Fading) ด้วยการกำหนดความน่าจะเป็นในการเปลี่ยนสถานะ p และ q



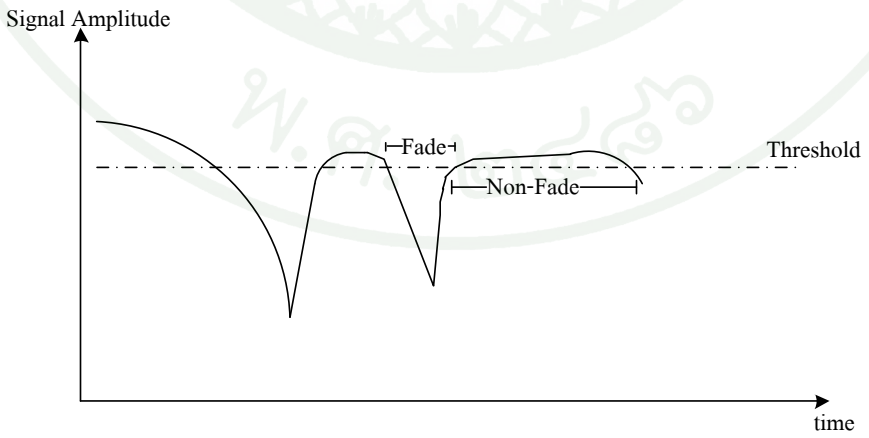
ภาพที่ 11 แบบจำลอง 2 สเตจ

ที่มา: Linnartz (1993)

จากภาพที่ 11 สามารถเขียนในรูปเมตริกซ์ได้

$$T = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix} \tag{46}$$

ดังนั้นสามารถจำลองช่องสัญญาณที่มีช่วงที่ช่องสัญญาณมีประสิทธิภาพดี(Non-fade) และช่วงที่ช่องสัญญาณมีประสิทธิภาพไม่ดีได้(Fade) เสมือนเป็นช่องสัญญาณไร้สาย



ภาพที่ 12 ตัวอย่างสัญญาณเมื่อช่องสัญญาณมีประสิทธิภาพดีและประสิทธิภาพไม่ดีในเวลาใดๆ

จากภาพที่ 12 จำลองสัญญาณเมื่อเวลาใดๆ พบว่าเมื่อสัญญาณมีค่ามากกว่าขีดแบ่ง (Threshold) ช่องสัญญาณมีประสิทธิภาพดี แต่เมื่อสัญญาณมีค่าน้อยกว่ากว่าขีดแบ่ง ช่องสัญญาณจะมีประสิทธิภาพไม่ดี ซึ่งสามารถคำนวณได้ดังนี้

ช่วงเวลาที่สัญญาณมีค่ามากกว่าขีดแบ่งหน่วยเป็นวินาที

$$= \frac{\sqrt{\eta}}{\sqrt{2\pi} \cdot f_d} \tag{47}$$

ช่วงเวลาที่สัญญาณมีค่าน้อยกว่ากว่าขีดแบ่งหน่วยเป็นวินาที

$$= \frac{\sqrt{\eta}}{\sqrt{2\pi} \cdot f_d} \cdot \left[e^{(\frac{1}{\eta})} - 1 \right] \tag{48}$$

และอัตราการเปลี่ยนต่อวินาที

$$= \frac{\sqrt{2\pi} \cdot f_d}{\sqrt{\eta}} \cdot e^{-\frac{1}{\eta}} \tag{49}$$

โดย η คือค่าเฟดมาจิน(Fade margin) หรืออัตราระหว่างค่ากำลังของสัญญาณต่อกำลังเส้นขีดแบ่ง

f_d คือค่าความถี่ดอปเพลอร์สามารถคำนวณจากสมการที่ (45)

จากนั้นสามารถนำค่าทั้งสามมาใช้เพื่อจำลองช่องสัญญาณได้ ซึ่งทั่วไปการจำลองช่องสัญญาณจะใช้ค่ากำลังของสัญญาณต่อกำลังของสัญญาณรบกวน แต่การจำลองช่องสัญญาณแบบ 2 สเตจนั้น ต้องการจำลองช่องสัญญาณไร้สายที่มีความผิดพลาดแบบเบริสต์เป็นช่วงๆ ซึ่งจะพิจารณาช่วงเวลาที่สัญญาณมีคุณภาพสูงหรือต่ำ โดยตัดสินจากค่าเฟดมาจินซึ่งเป็นอัตราระหว่างค่ากำลังของสัญญาณต่อกำลังเส้นขีดแบ่ง กล่าวคือถ้าค่าเฟดมาจินน้อยกว่าศูนย์แสดงว่ากำลังของ

สัญญาณฝั่งรับน้อยกว่าเมื่อเทียบกับกำลังของเส้นขีดแบ่ง หรือก็คือช่องสัญญาณมีคุณภาพต่ำทำให้กำลังของสัญญาณฝั่งรับต่ำกว่าเกณฑ์ที่กำหนด ซึ่งจะทำให้เกิดความผิดพลาดมาก แต่ถ้าค่าเฟดมาจินมากคือกำลังของสัญญาณฝั่งรับมากกว่ากำลังของเส้นขีดแบ่ง หรือช่องสัญญาณมีคุณภาพสูงทำให้กำลังของสัญญาณฝั่งรับสูงกว่าเกณฑ์ที่กำหนด ดังนั้นเมื่อกำลังของสัญญาณมากทำให้เกิดค่าความผิดพลาดน้อย

อุปกรณ์และวิธีการณ์

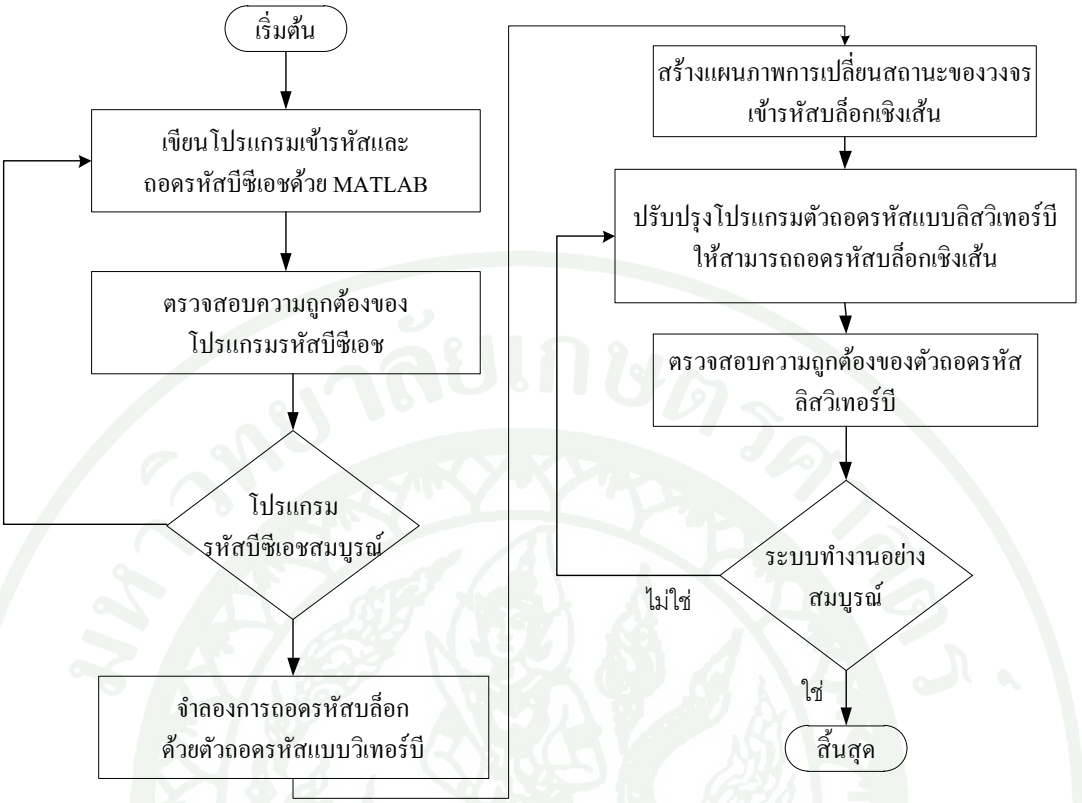
อุปกรณ์

1. เครื่องคอมพิวเตอร์ (Personal Computer)
2. โปรแกรม MATLAB
3. โปรแกรม Microsoft Visual Studio โดยใช้ภาษา C++

วิธีการ

1. พัฒนาตัวถอดรหัสภายในแบบบีซีเอชให้สามารถถอดรหัสด้วยลิสวิเทอร์บี

รหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสเป็น LVA-VSD แต่เดิมต้องใช้ตัวเข้ารหัสภายในแบบคอนโวลูชัน เพื่อให้สามารถใช้งานกับตัวถอดรหัสแบบลิสวิเทอร์บีได้ เพื่อให้เกิดความหลากหลายในการใช้งาน จึงทำการพัฒนาตัวถอดรหัสภายในแบบลิสวีเทอร์บี ให้สามารถถอดรหัสบล็อกเชิงเส้นได้ด้วย ทำให้สามารถใช้รหัสบล็อกเชิงเส้นเป็นรหัสภายในโดยสามารถถอดรหัสแบบลิสได้



ภาพที่ 13 ผังการดำเนินงานพัฒนาตัวถอดรหัสแบบลิสวิเทอร์บีสำหรับรหัสบล็อกเชิงเส้น

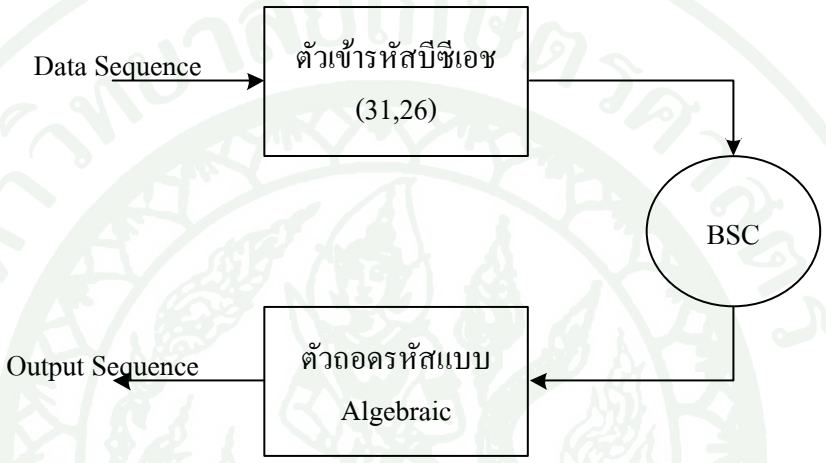
จากภาพที่ 13 สามารถอธิบายการดำเนินงานการดำเนินงานอย่างละเดียดได้ดังนี้

1.1 เข้ารหัสและถอดรหัสแบบบีซีเอช ด้วย MATLAB

เริ่มจากการจำลองรหัสช่องสัญญาณบีซีเอชทั้งฝั่งเข้ารหัสและถอดรหัสด้วยฟังก์ชัน (Function) จากโปรแกรม MATLAB โดยโครงงานนี้ใช้รหัสบีซีเอช (31, 26) สามารถแก้ไขความผิดพลาดได้หนึ่งบิต มีพหุนามตัวกำเนิดเป็น $\boldsymbol{G}(D)=[1+D^3+D^5]=100101_2$ และกำหนดข้อมูลขาเข้า (input) เป็นศูนย์ 26 บิต เพื่อง่ายต่อการตรวจสอบความผิดพลาด เมื่อเข้ารหัสแล้วจะได้ชุดคำรหัส 31 บิต จากนั้นใส่ค่าความผิดพลาดลงในชุดคำรหัสและนำไปถอดรหัส ถ้าใส่ค่าความผิดพลาดหนึ่งบิตแล้วสามารถแก้ไขให้ถูกได้แสดงว่าสามารถจำลองรหัสบีซีเอชได้สมบูรณ์ แต่ถ้าไม่สามารถแก้ไขความผิดพลาดได้แสดงว่าการจำลองรหัสบีซีเอชไม่ถูกต้อง หรือถ้าใส่ค่าความผิดพลาดมากกว่าหนึ่งบิต รหัสบีซีเอชต้องไม่สามารถแก้ไขค่าความผิดพลาดได้ดังทฤษฎี

1.2 จำลองระบบรหัสช่องสัญญาณและบันทึกผล

เมื่อสามารถสร้างตัวเข้ารหัสและตัวถอดรหัสช่องสัญญาณแบบบีซีเอชแล้ว จึงนำมาจำลองระบบรหัสช่องสัญญาณ โดยเพิ่มช่องสัญญาณแบบบีเอสซีเพื่อให้เกิดเป็นระบบที่มีการรับ - ส่งข้อมูลผ่านช่องสัญญาณที่มีความผิดพลาดดังภาพที่ 14



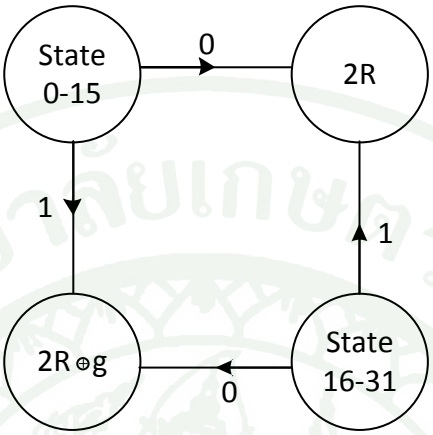
ภาพที่ 14 การจำลองการทำงานด้วยโปรแกรม MATLAB ของรหัสบีซีเอช โดยใช้การถอดรหัสแบบ Algebraic ด้วยการตัดสินใจแบบหยาบ

โดยตัวถอดรหัสช่องสัญญาณแบบบีซีเอชของ MATLAB นั้นเป็นการถอดรหัสช่องสัญญาณแบบ Algebraic ด้วยการตัดสินใจแบบหยาบ และเลือกใช้ช่องสัญญาณเป็นช่องสัญญาณบีเอสซีเพราะเกิดความผิดพลาดแบบสุ่ม รหัสบีซีเอช (31, 26) ซึ่งสามารถแก้ไขความผิดพลาดได้หนึ่งบิตมีโอกาสแก้ไขความผิดพลาดที่เกิดขึ้นได้ สุดท้ายบันทึกผลความน่าจะเป็นที่เกิดความผิดพลาด

1.3 ศึกษาการถอดรหัสบล็อกเชิงเส้นด้วยวิเทอร์บี

ผู้วิจัยศึกษาการถอดรหัสบล็อกเชิงเส้นโดยใช้วิเทอร์บีจากบทความหรือผลงานวิชาการต่างๆที่ได้ถูกนำเสนอหรือตีพิมพ์ และพบว่ามีผู้เคยเสนอการถอดรหัสบล็อกเชิงเส้นโดยใช้วิเทอร์บี (Reeve and Amarasinghe, 2004) โดยเป็นการนำเสนอวิธีการคำนวณเปลี่ยนสถานะของรหัสบล็อก

เมื่อสามารถทราบการเปลี่ยนสถานะก็สามารถนำมาถอดรหัสแบบวิเทอร์บีได้ โดยนำการเปลี่ยนสถานะไปเขียนแผนภาพเทรลลิสและคำนวณหาเส้นทาง



ภาพที่ 15 วิธีหาการเปลี่ยนสถานะของรหัสบล็อกเชิงเส้น

ที่มา: Reeve and Amarasinghe (2004)

จากภาพที่ 15 เป็นการใช้สถานะปัจจุบันคำนวณหาสถานะถัดไป โดยถ้าสถานะปัจจุบันคือสเตจ (state) ที่ 0-15 ให้เริ่มจากวงกลมบนซ้าย หากข้อมูลขาเข้าคือศูนย์ให้ใช้วงกลมบนขวาคือ สองคูณสเตจปัจจุบัน ในการคำนวณสเตจถัดไป หากข้อมูลขาเข้าเป็นหนึ่ง ให้ใช้วงกลมล่างซ้าย คือ สองคูณสเตจปัจจุบันเอ็กซ์คลูซีฟออร์(XOR) กับพารามิเตอร์ g ในการคำนวณสเตจถัดไป ถ้าสถานะปัจจุบันคือสเตจที่ 16-31ให้เริ่มจากวงกลมล่างขวา หากข้อมูลขาเข้าคือศูนย์ให้ใช้วงกลมล่างซ้าย คือ สองคูณสเตจปัจจุบันเอ็กซ์คลูซีฟออร์กับพารามิเตอร์ g ในการคำนวณสเตจถัดไป หากข้อมูลขาเข้าเป็นหนึ่ง ให้ใช้วงกลมบนขวาคือ สองคูณสเตจปัจจุบัน ในการคำนวณสเตจถัดไป

โดยการคำนวณการเปลี่ยนสถานะเริ่มจากคำนวณพหุนามตัวกำเนิน $G(D)$

$$G = 100101_2 \tag{50}$$

พารามิเตอร์ “Q”

$$Q = 2^{n-k-1} = 2^{31-26-1} = 2^4 = 16 = 1000_2 \tag{51}$$

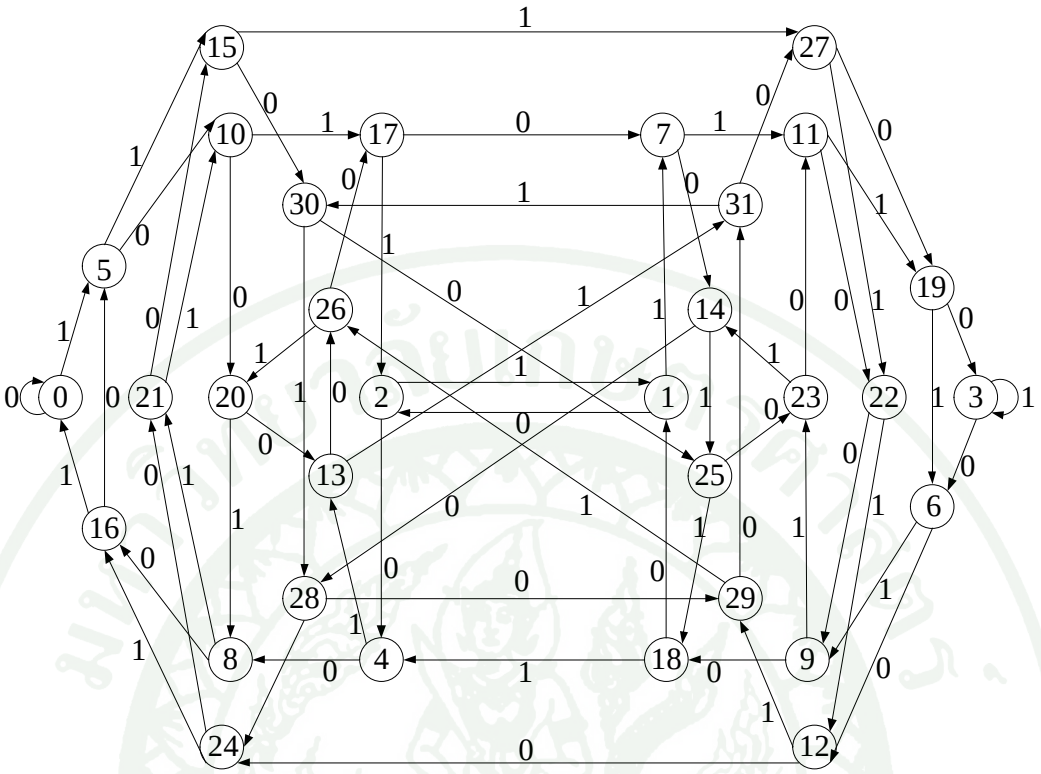
พารามิเตอร์ g คำนวณจากการเอ็กซ์คลูซีฟออร์ระหว่างพารามิเตอร์ G และ Q

$$g = G \oplus Q \tag{52}$$

แทนสมการที่ (50) และ (51) ในสมการที่ (52)

$$g = 100101_2 \oplus 1000_2 = 101_2 = 5 \tag{53}$$

จากนั้นนำพารามิเตอร์ g จากสมการที่ (53) ไปคำนวณหาการเปลี่ยนสถานะได้จากภาพที่ 15 และได้แผนภาพจากการคำนวณ โดยสามารถแสดงแผนภาพการเปลี่ยนสถานะของรหัสบีซีเอช (31, 26) ดังภาพที่ 16



ภาพที่ 16 แผนภาพแสดงการเปลี่ยนสถานะของรหัสบีซีเอช (31,26) $\boldsymbol{G}(D)=[1+D^3+D^5]$

ที่มา: Tuntoolavest *et al*. (2011)

เมื่อนำพารามิเตอร์ G และพารามิเตอร์ Q มาคำนวณหาพารามิเตอร์ g และใช้การคำนวณคู่กับภาพที่ 16 สามารถเขียนแผนภาพการเปลี่ยนสถานะของรหัสบีซีเอช (31,26) ดังภาพที่ 16 และสามารถเขียนการเปลี่ยนสถานะได้ดังตารางที่ 3 และตารางที่ 4

ตารางที่ 3 การเปลี่ยนสถานะของรหัสบีซีเอช (31,26) ที่มี $\boldsymbol{G}(D)=[1+D^3+D^5]$ ตั้งแต่สเตจที่ 0 ถึง 15

	Current State		Next State	
input	State in Binary	State Number	State in Binary	State Number
0	0 0 0 0 0	0	0 0 0 0 0	0
1	0 0 0 0 0	0	0 0 1 0 1	5
0	0 0 0 0 1	1	0 0 0 1 0	2
1	0 0 0 0 1	1	0 0 1 1 1	7
0	0 0 0 1 0	2	0 0 1 0 0	4
1	0 0 0 1 0	2	0 0 0 0 1	1
0	0 0 0 1 1	3	0 0 1 1 0	6
1	0 0 0 1 1	3	0 0 0 1 1	3
0	0 0 1 0 0	4	0 1 0 0 0	8
1	0 0 1 0 0	4	0 1 1 0 1	13
0	0 0 1 0 1	5	0 1 0 1 0	10
1	0 0 1 0 1	5	0 1 1 1 1	15
0	0 0 1 1 0	6	0 1 1 0 0	12
1	0 0 1 1 0	6	0 1 0 0 1	9
0	0 0 1 1 1	7	0 1 1 1 0	14
1	0 0 1 1 1	7	0 1 0 1 1	11
0	0 1 0 0 0	8	1 0 0 0 0	16
1	0 1 0 0 0	8	1 0 1 0 1	21
0	0 1 0 0 1	9	1 0 0 1 0	18
1	0 1 0 0 1	9	1 0 1 1 1	23
0	0 1 0 1 0	10	1 0 1 0 0	20
1	0 1 0 1 0	10	1 0 0 0 1	17
0	0 1 0 1 1	11	1 0 1 1 0	22
1	0 1 0 1 1	11	1 0 0 1 1	19
0	0 1 1 0 0	12	1 1 0 0 0	24
1	0 1 1 0 0	12	1 1 1 0 1	29
0	0 1 1 0 1	13	1 1 0 1 0	26
1	0 1 1 0 1	13	1 1 1 1 1	31
0	0 1 1 1 0	14	1 1 1 0 0	28
1	0 1 1 1 0	14	1 1 0 0 1	25
0	0 1 1 1 1	15	1 1 1 1 0	30
1	0 1 1 1 1	15	1 1 0 1 1	27

ที่มา: Tuntoolavest *et al.* (2011)

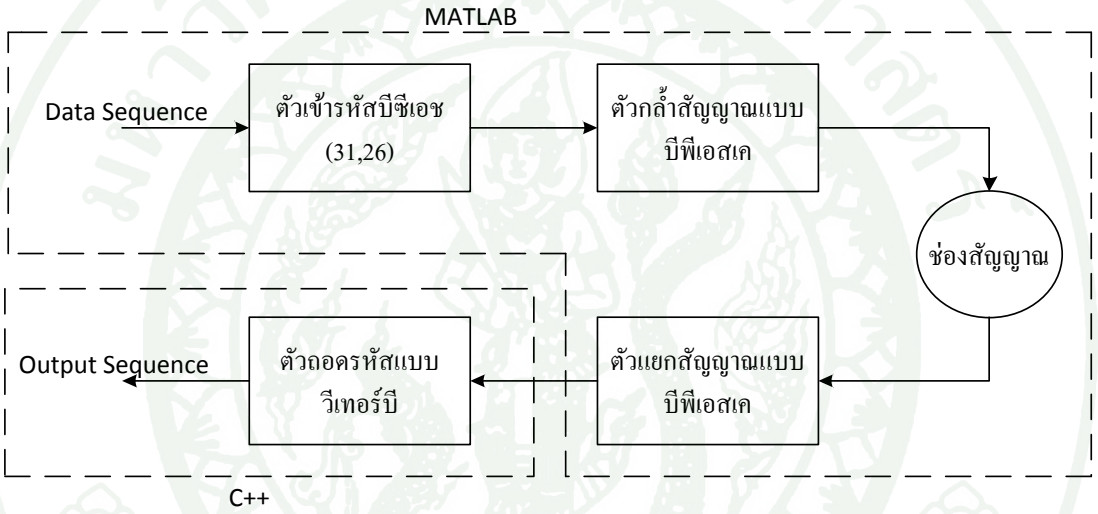
ตารางที่ 4 การเปลี่ยนสถานะของรหัสบีซีเอช (31,26) ที่มี $\boldsymbol{G}(D)=[1+D^3+D^5]$ ตั้งแต่สเตจที่ 16 ถึง 31

	Current State		Next State	
input	State in Binary	State Number	State in Binary	State Number
0	1 0 0 0 0	16	0 0 1 0 1	5
1	1 0 0 0 0	16	0 0 0 0 0	0
0	1 0 0 0 1	17	0 0 1 1 1	7
1	1 0 0 0 1	17	0 0 0 1 0	2
0	1 0 0 1 0	18	0 0 0 0 1	1
1	1 0 0 1 0	18	0 0 1 0 0	4
0	1 0 0 1 1	19	0 0 0 1 1	3
1	1 0 0 1 1	19	0 0 1 1 0	6
0	1 0 1 0 0	20	0 1 1 0 1	13
1	1 0 1 0 0	20	0 1 0 0 0	8
0	1 0 1 0 1	21	0 1 1 1 1	15
1	1 0 1 0 1	21	0 1 0 1 0	10
0	1 0 1 1 0	22	0 1 0 0 1	9
1	1 0 1 1 0	22	0 1 1 0 0	12
0	1 0 1 1 1	23	0 1 0 1 1	11
1	1 0 1 1 1	23	0 1 1 1 0	14
0	1 1 0 0 0	24	1 0 1 0 1	21
1	1 1 0 0 0	24	1 0 0 0 0	16
0	1 1 0 0 1	25	1 0 1 1 1	23
1	1 1 0 0 1	25	1 0 0 1 0	18
0	1 1 0 1 0	26	1 0 0 0 1	17
1	1 1 0 1 0	26	1 0 1 0 0	20
0	1 1 0 1 1	27	1 0 0 1 1	19
1	1 1 0 1 1	27	1 0 1 1 0	22
0	1 1 1 0 0	28	1 1 1 0 1	29
1	1 1 1 0 0	28	1 1 0 0 0	24
0	1 1 1 0 1	29	1 1 1 1 1	31
1	1 1 1 0 1	29	1 1 0 1 0	26
0	1 1 1 1 0	30	1 1 0 0 1	25
1	1 1 1 1 0	30	1 1 1 0 0	28
0	1 1 1 1 1	31	1 1 0 1 1	27
1	1 1 1 1 1	31	1 1 1 1 0	30

ที่มา: Tuntoolavest *et al.* (2011)

1.4 จำลองการถอดรหัสบล็อกเชิงเส้นด้วยวิเทอร์บีและบันทึกผล

เมื่อทราบการเปลี่ยนสถานะของรหัสบีซีเอชแล้ว นำข้อมูลดังกล่าวไปประยุกต์ใช้กับโปรแกรมถอดรหัสลิสวิเทอร์บีที่มีสองผลลัพธ์โดยใช้การตัดสินใจแบบหยาบ ด้วยโปรแกรม Microsoft Visual Studio โดยใช้ภาษา C++ (Tuntoolavest) ซึ่งโปรแกรมจะถอดรหัสให้ผลลัพธ์ที่ดีที่สุดและรองลงมา จากนั้นเปรียบเทียบความถูกต้องและบันทึกผลความน่าจะเป็นที่เกิดความผิดพลาด



ภาพที่ 17 การจำลองการทำงานฝั่งเข้ารหัสด้วยโปรแกรม MATLAB และใช้ตัวถอดรหัสวีเทอร์บีด้วยการตัดสินใจแบบหยาบโดยโปรแกรม Microsoft Visual Studio ด้วยภาษา C++

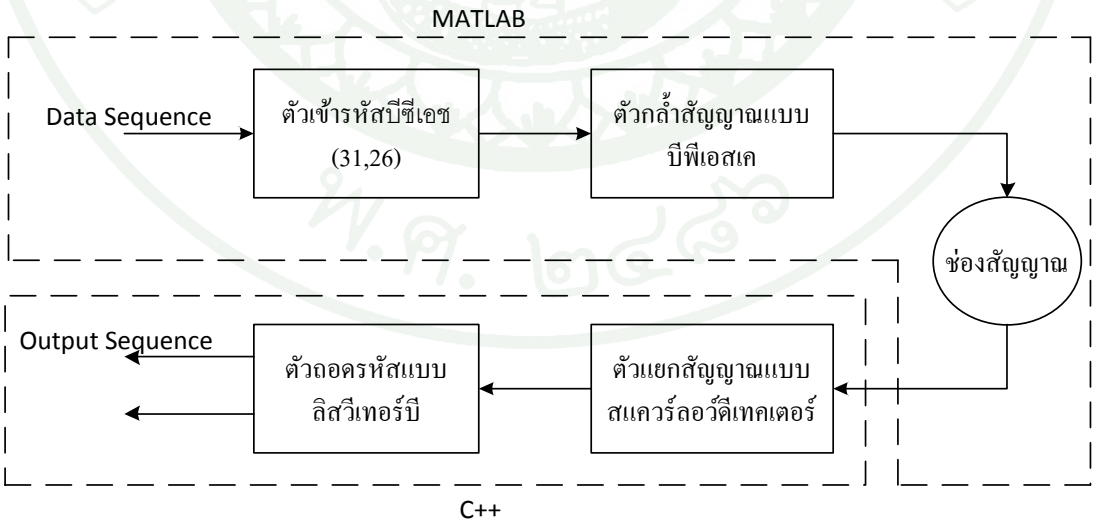
ภาพที่ 17 แสดงการจำลองการทำงานในแต่ละส่วน ทั้งส่วนของ MATLAB ที่ทำงานตั้งแต่เข้ารหัสบีซีเอช กล้ำสัญญาณ ผ่านช่องสัญญาณและแยกสัญญาณแบบบีพีเอสเค (Binary Phase Shift Keying: BPSK) และส่วนของ Microsoft Visual Studio ที่ใช้ภาษา C++ ทำงานเป็นตัวถอดรหัสแบบวิเทอร์บี

1.5 เปรียบเทียบผลการถอดรหัสระหว่างการถอดรหัสโดย Algebraic ด้วยการตัดสินใจแบบหยาบ และการถอดรหัสโดยวิเทอร์บีด้วยการตัดสินใจแบบหยาบ

ตรวจสอบความถูกต้องของการถอดรหัสบล็อกเชิงเส้นด้วยวิเทอร์บีโดยการเปรียบเทียบผลลัพธ์ที่ดีที่สุดของการถอดรหัสวิเทอร์บีด้วยการตัดสินใจแบบหยาบดังแผนภาพที่ 17 และการถอดรหัสโดย Algebraic ด้วยการตัดสินใจแบบหยาบ โดยใช้ฟังก์ชันการถอดรหัสบีซีเอชจากโปรแกรม MALTAB ซึ่งผลของทั้งสองวิธีต้องเท่ากันจึงจะถูกต้อง

1.6 จำลองการถอดรหัสลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อนและบันทึกผล

เพิ่มประสิทธิภาพของระบบโดยการทำลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อน โดยใช้โปรแกรม Microsoft Visual Studio (Tuntoolavest and Seubnaung, 2007) ดังภาพที่ 18 ซึ่งจะให้ประสิทธิภาพที่ดีกว่าการถอดรหัสด้วยการตัดสินใจแบบหยาบ เนื่องจากข้อมูลที่จะถอดรหัสด้วยการตัดสินใจแบบอ่อนมีหลายระดับ ทำให้สามารถใช้ข้อมูลดังกล่าวคำนวณได้ค่าที่สามารถนำไปถอดรหัสได้แม่นยำกว่า จากนั้นบันทึกผลและเทียบประสิทธิภาพหลังถอดรหัสกับการถอดรหัสโดยใช้การตัดสินใจแบบหยาบ โดยการถอดรหัสโดยใช้การตัดสินใจแบบอ่อนควรมีประสิทธิภาพดีกว่าตามทฤษฎี



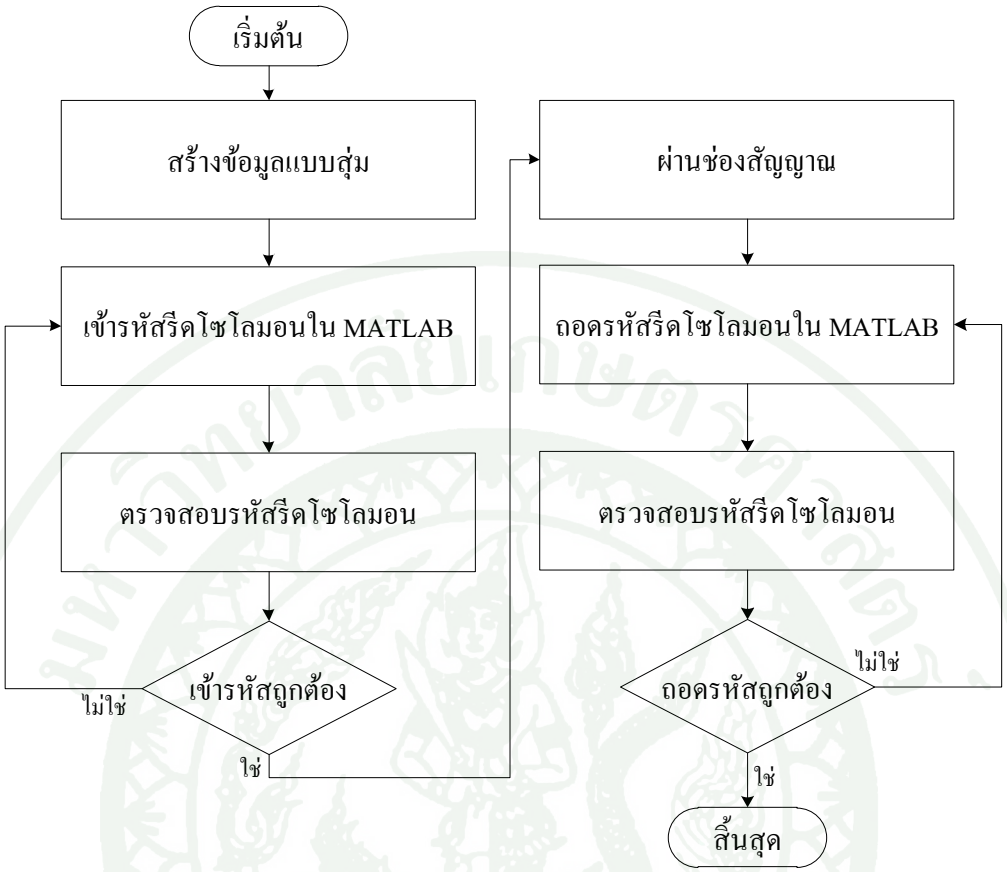
ภาพที่ 18 ภาพจำลองระบบรหัสบล็อกเชิงเส้นที่ถอดรหัสด้วยลิสวิเทอร์บีแบบอ่อน

1.7 ทดลองระบบในแบบจำลองช่องสัญญาณอื่น

ทดลองระบบกับแบบจำลองช่องสัญญาณอื่นได้แก่ไวส์เกาท์เซียนแบบบวก รวมทั้งการจางหายแบบเรย์ลีและการจางหายแบบไรเซียน ทั้งที่มีดอปเพลอร์ และไม่มีดอปเพลอร์ และบันทึกผล

2. ใช้รหัสภายในเป็นรหัสรีดโซโลมอน

จากการศึกษางานวิจัยพบว่ามีแนวคิดในการนำรหัสนอนไบนารีมาใช้เป็นรหัสภายใน (Andreadou and Pavlidou, 2010) และให้ประสิทธิภาพที่ดี ประกอบกับปัจจุบันการสื่อสารมีความเร็วในการรับ - ส่งข้อมูลที่มากขึ้น ดังนั้นในช่วงเวลาเท่าเดิมแม้จะสามารถรับ - ข้อมูลได้มากขึ้น แต่ก็มีโอกาสที่จะเกิดความผิดพลาดมากขึ้นเช่นกัน จึงเป็นแนวคิดที่นำรหัสรีดโซโลมอนซึ่งเป็นรหัสวนแบบนอนไบนารีมาใช้เป็นรหัสภายใน เนื่องจากรหัสรีดโซโลมอนเป็นรหัสที่ดีเพราะมีค่าระยะแฮมมิงต่ำสุด ($d_{\min}$) สูงสุด ทำให้มีประสิทธิภาพในการแก้ไขข้อมูลสูง อีกทั้งยังเป็นรหัสที่นิยมใช้อย่างแพร่หลาย ทำให้มีการพัฒนาในหลากหลายรูปแบบ จึงง่ายต่อการพัฒนาระบบและเพิ่มประสิทธิภาพ ซึ่งช่วยเพิ่มประสิทธิภาพในการถอดรหัส ที่มีตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบล โดยขั้นตอนการดำเนินงานการใช้รหัสภายในให้เป็นรหัสรีดโซโลมอนสามารถเขียนเป็นแผนผังได้ดังภาพที่ 19



ภาพที่ 19 ผังการดำเนินงานการใช้รหัสภายในให้เป็นรหัสรีดโซโลมอน

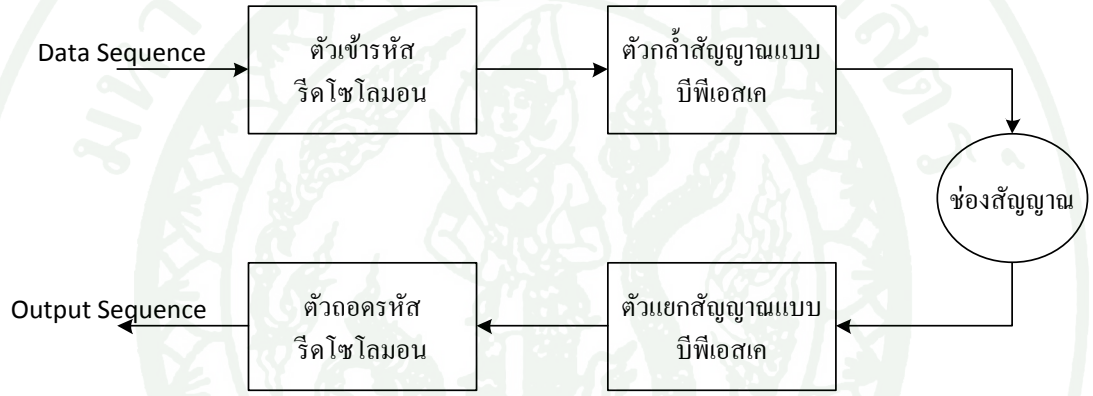
โดยขั้นตอนการนำรหัสรหัสรีดโซโลมอนมาใช้เป็นรหัสภายในมีดังนี้

2.1 เข้ารหัสและถอดรหัสแบบรีดโซโลมอนด้วย MATLAB

ทดลองเขียนรหัสรีดโซโลมอนทั้งฝั่งเข้ารหัสและถอดรหัสในโปรแกรม MATLAB ด้วยการจำลองใส่ข้อมูลขาเข้าด้วยบิตศูนย์ทั้งหมด k สัญลักษณ์เช่นเดียวกับการทดลองเขียนรหัสบีซีเอช จะได้ชุดคำรหัส n สัญลักษณ์ จากนั้นใส่ค่าความผิดพลาดลงในชุดคำรหัสและนำไปถอดรหัส ถ้าค่าความผิดพลาดไม่เกิน $(n-k)/2$ สัญลักษณ์ จะสามารถแก้ไขความผิดพลาดได้

2.2 จำลองระบบและบันทึกผล

จำลองระบบด้วยโปรแกรม MATLAB โดยเพิ่มการกล้ำสัญญาณ การแยกสัญญาณและช่องสัญญาณ แม้จะเป็นรหัสนอนไบนารี แต่โครงงานนี้หลังจากเข้ารหัสแล้ว จะแปลงคำรหัสจากนอนไบนารีให้เป็นไบนารีและใช้ตัวกล้ำสัญญาณรวมถึงตัวแยกสัญญาณแบบบีพีเอสเค เพราะเป็นตัวแปรควบคุม เนื่องจากต้องการให้ระบบรหัสคอนคาทิเนตดังกล่าวมีองค์ประกอบใกล้เคียงกับการใช้รหัสภายในแบบบีซีเอชให้มากที่สุด และสามารถเทียบประสิทธิภาพได้ โดยสามารถเขียนแผนภาพการจำลองการทำงานได้ดังภาพที่ 20

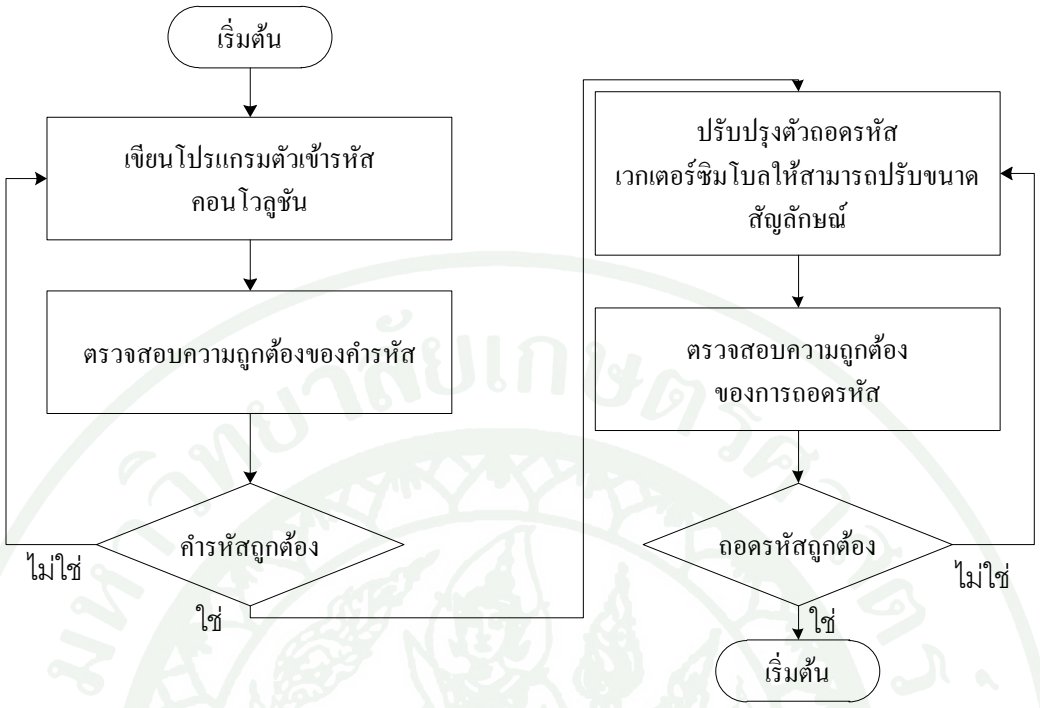


ภาพที่ 20 การจำลองการทำงานด้วยโปรแกรม MATLAB ของรหัสรีดโซโลมอน

โครงงานนี้ใช้รหัสรีดโซโลมอน (63, 51) ด้วย $GF(2^6)$ หรือสัญลักษณ์ละ 6 บิต โดยเข้ารหัสครั้งละ 51 สัญลักษณ์ และได้เป็นคำรหัส 63 สัญลักษณ์ ซึ่งจะมีอัตรารหัสใกล้เคียงกับรหัสบีซีเอช (31, 26) และจากภาพที่ 20 สามารถนำประสิทธิภาพการถอดรหัสของทั้งสองรหัสมาเปรียบเทียบค่าความผิดพลาดต่อบิตได้

3. พัฒนาตัวถอดรหัสเวกเตอร์ซิมโบลให้สามารถเปลี่ยนขนาดสัญลักษณ์ได้

การที่ความเร็วในการรับ-ส่งข้อมูลเพิ่มขึ้นอาจทำให้สามารถรับ-ส่งข้อมูลมากขึ้นในเวลาเท่าเดิม แต่จำนวนข้อมูลที่ผิดพลาดก็เพิ่มขึ้นเช่นกัน ทำให้เกิดแนวคิดในการขยายขนาดสัญลักษณ์ของรหัสภายนอก เพื่อรองรับค่าความผิดพลาดที่มากขึ้น การดำเนินงานการพัฒนาตัวถอดรหัสเวกเตอร์ซิมโบลให้สามารถเปลี่ยนขนาดสัญลักษณ์แสดงในภาพที่ 21



ภาพที่ 21 ผังการดำเนินงานการพัฒนาตัวถอดรหัสเวกเตอร์ซิมโบลให้สามารถเปลี่ยนขนาดสัญลักษณ์

ตัวเข้ารหัสภายนอกที่ใช้คือคอนโวลูชัน (3, 2, 2) ซึ่งมีเมตริกซ์ตัวกำเนิด (Generator Matrix)

$$\boldsymbol{G}(D)=\begin{bmatrix} 1+D+D^2 & D^2 & 1 \\ D & 1+D^2 & 1+D+D^2 \end{bmatrix} \tag{54}$$

สามารถนำสมการที่ (54) มาเขียนเป็นแผนภาพวงจรได้ดังภาพที่ 4

ตัวอย่างการเข้ารหัสคอนโวลูชัน (3, 2, 2) ถ้าข้อมูลที่เข้ารหัสคือ

$$U=(11,00,11,00,00) \tag{55}$$

หรือ

$$U^{(1)} = (1,0,1,0,0) \tag{56}$$

$$U^{(2)} = (1,0,1,0,0) \tag{57}$$

จะได้คำรหัส

$$V = (110,001,011,001,101,000,000) \tag{58}$$

หรือ

$$V^{(1)} = (1,0,0,0,1,0,0) \tag{59}$$

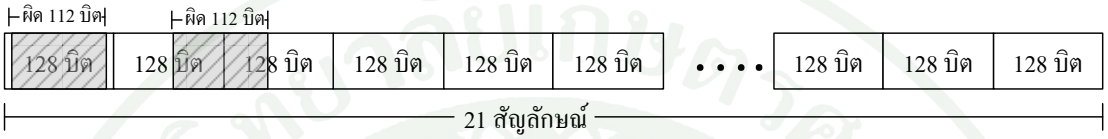
$$V^{(2)} = (1,0,1,0,0,0,0) \tag{60}$$

$$V^{(3)} = (0,1,1,1,1,0,0) \tag{61}$$

โดยมีข้อมูลเข้ารหัสทั้งหมด 10 สัญลักษณ์ และออกเป็นคำรหัส 21 สัญลักษณ์ ซึ่งความยาวคำรหัสเท่ากับ $n(m+L)$ โดยที่ L คือความยาวข้อมูลในแต่ละชุด m คือจำนวนหน่วยความจำ และ n คือจำนวนชุดคำรหัส

เมื่อสามารถเข้ารหัสคอนโวลูชันแล้ว จึงพัฒนาตัวถอดรหัสเวกเตอร์ซิมโบลโดยพัฒนาโปรแกรมให้รหัสมีความยืดหยุ่นมากขึ้น สามารถเพิ่มขนาดสัญลักษณ์ให้ใหญ่ขึ้นได้ถึง 4 เท่าจากของเดิม

ภาพที่ 22 ขนาดสัญลักษณ์ก่อนพัฒนาโปรแกรมถอดรหัสเวกเตอร์ซิมโบล



ภาพที่ 23 ขนาดสัญลักษณ์หลังพัฒนาโปรแกรมถอดรหัสเวกเตอร์ซิมโบล

จากภาพที่ 22 และภาพที่ 23 เปรียบเทียบลักษณะชุดคำรหัสแบบเดิมและแบบขยายขนาดสัญลักษณ์ จะเห็นว่าจากชุดรหัสแบบเดิมสมมุติให้มีความผิดพลาด 112 บิตติดต่อกันทำให้ชุดคำรหัสอาจมีสัญลักษณ์ที่ผิด 4 หรือ 5 สัญลักษณ์ แต่เมื่อขยายขนาดสัญลักษณ์มีโอกาสที่จะทำให้ความผิดพลาดทั้งหมดอยู่ในสัญลักษณ์เดียวกันหรืออยู่คร่อมกันระหว่างสองสัญลักษณ์เท่านั้น ดังนั้นการใช้สัญลักษณ์ขนาดใหญ่ขึ้นจึงเพิ่มประสิทธิภาพในการถอดรหัสได้

โดยขั้นตอนปรับขนาดสัญลักษณ์ของรหัสเวกเตอร์ซิมโบลมีดังนี้

3.1 แก้ไขโปรแกรมตัวถอดรหัสเวกเตอร์ซิมโบลในส่วนของขนาดสัญลักษณ์

เพื่อให้ง่ายต่อการสังเกตในแต่ละขั้นตอนการทำงานจึงกำหนดข้อมูลขาเข้าเป็นศูนย์ทั้งหมด และกำหนดบิตที่ผิดพลาดเป็นหนึ่ง จากนั้นตรวจสอบทีละขั้นตอนเทียบกับทฤษฎี โดยเริ่มจากการ

- คำนวณค่าซินโดรมแรกว่าถูกต้องหรือไม่ ถ้าไม่ถูกให้รับซินโดรมตัวถัดไปมาช่วยในการถอดรหัส
- ทำ Gauss-Jordan column operation
- คำนวณชุดการรวมเป็นศูนย์ เพื่อหาเวกเตอร์ระบุตำแหน่งที่ผิดพลาด

- ตรวจสอบว่าสามารถแก้ไขความผิดพลาดได้หรือไม่ ถ้าแก้ไขไม่ได้ให้รับค่าซินโดรมถัดไป เพื่อใช้ในการคำนวณหาชุดการรวมเป็นศูนย์อีกครั้ง

- ตรวจสอบว่าโปรแกรมเวกเตอร์ซิมโบลสามารถแก้ไขความผิดพลาดได้ถูกต้องหรือไม่

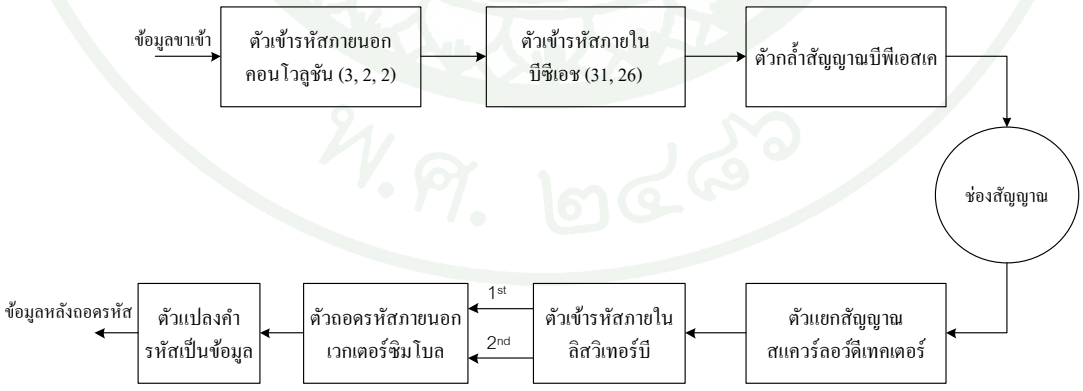
3.3 ทดลองและบันทึกผล

เมื่อแก้ไขข้อมูลจนสามารถทำงานได้ถูกต้องทุกขั้นตอนแล้วจึงเปลี่ยนข้อมูลขาเข้าเป็นแบบสุ่มและตรวจสอบผลลัพธ์ว่าระบบโดยรวมถูกต้อง สุดท้ายบันทึกผลลัพธ์ที่ได้

4. ทดลองระบบคอนคาทิเนตโดยใช้รหัสภายนอกและรหัสภายในที่พัฒนาขึ้น

โครงงานนี้พัฒนารหัสภายนอกแบบบีซีเอชให้สามารถถอดรหัสแบบลิสวิเทอร์บี และปรับใช้รหัสภายในแบบรีดโซโลมอน ขณะที่พัฒนารหัสภายนอกแบบเวกเตอร์ซิมโบลให้สามารถขยายขนาดสัญลักษณ์ได้ และสามารถจับคู่การทำงานรหัสภายนอกและรหัสภายในดังนี้

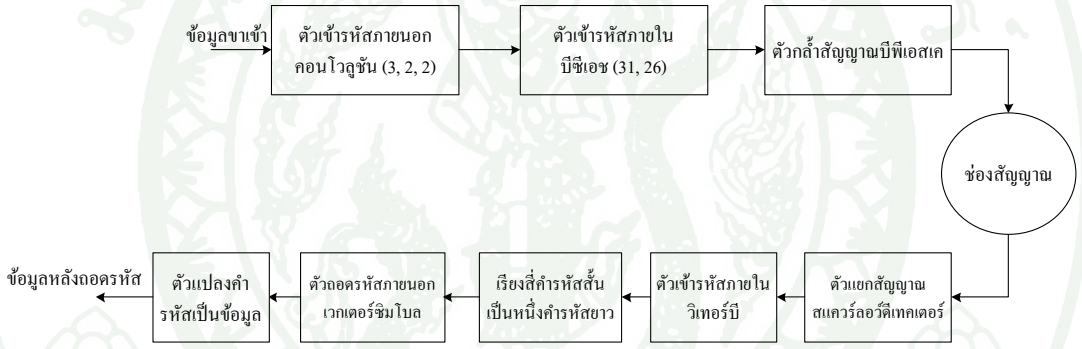
4.1 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบบีซีเอช และถอดรหัสโดยลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อน ร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบล



ภาพที่ 24 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบบีซีเอช และถอดรหัสโดยลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อน ร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบล

จากการพัฒนาตัวถอดรหัสลิสวิเทอร์บีที่ใช้การตัดสินใจแบบอ่อนสำหรับรหัสบล็อก สามารถนำมาใช้เป็นรหัสภายในของรหัสคอนคาทิเนตที่มีรหัสภายนอกแบบเวกเตอร์ซิมโบลได้ดังภาพที่ 24 ซึ่งตัวเข้ารหัสภายนอกเป็นรหัสคอนโวลูชัน (3, 2, 2) ที่ใช้ขนาดสัญลักษณ์ละ 26 บิต และเข้ารหัสภายในแบบบีซีเอช (31, 26) จากนั้นกล้ำสัญญาณด้วยบีพีเอสเค และผ่านแบบจำลองช่องสัญญาณต่างๆ แยกสัญญาณด้วยบีพีเอสเค และถอดรหัสภายในด้วยลิสวีเทอร์บีแบบอ่อน ซึ่งจะได้สองผลลัพธ์ นำสองผลลัพธ์ที่ได้จากรหัสภายในมาใช้ในการคำนวณการถอดรหัสภายนอกแบบเวกเตอร์ซิมโบล เพื่อช่วยเพิ่มประสิทธิภาพของการถอดรหัสของระบบรหัสคอนคาทิเนต

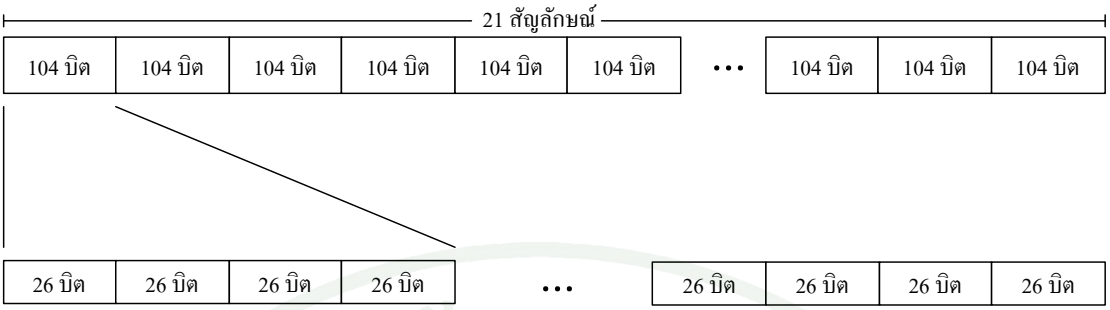
4.2 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบบีซีเอช และถอดรหัสโดยวิเทอร์บีด้วยการตัดสินใจแบบอ่อน ร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ขยายขนาดสัญลักษณ์



ภาพที่ 25 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบบีซีเอชและถอดรหัสโดยลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อนร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ขยายขนาดสัญลักษณ์

จากการพัฒนาการขยายขนาดสัญลักษณ์ของรหัสเวกเตอร์ซิมโบลจนแล้วเสร็จ นำรหัสเวกเตอร์ซิมโบลมาประยุกต์ใช้กับระบบรหัสคอนคาทิเนตที่ใช้รหัสภายในเป็นรหัสบีซีเอชที่ถอดรหัสวีเทอร์บีด้วยการตัดสินใจแบบอ่อนได้ดังภาพที่ 25

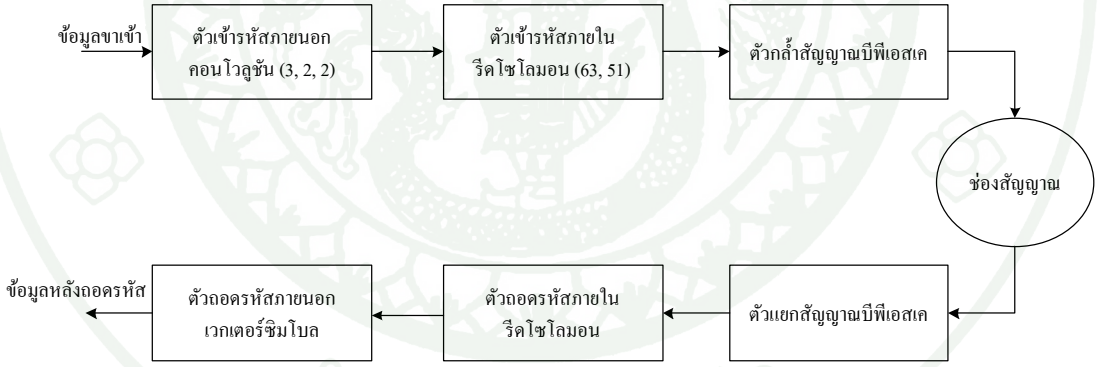
ตัวเข้ารหัสภายนอกใช้รหัสคอนโวลูชัน (3, 2, 2) ขนาดสัญลักษณ์ละ 104 บิต และรหัสภายในใช้รหัสบีซีเอช (31, 26) โดยหนึ่งสัญลักษณ์ของรหัสภายนอกสามารถแบ่งเป็นรหัสภายในได้ 4 สัญลักษณ์ดังภาพที่ 26



ภาพที่ 26 แสดงการแบ่งรหัสยาวเป็นรหัสสั้น

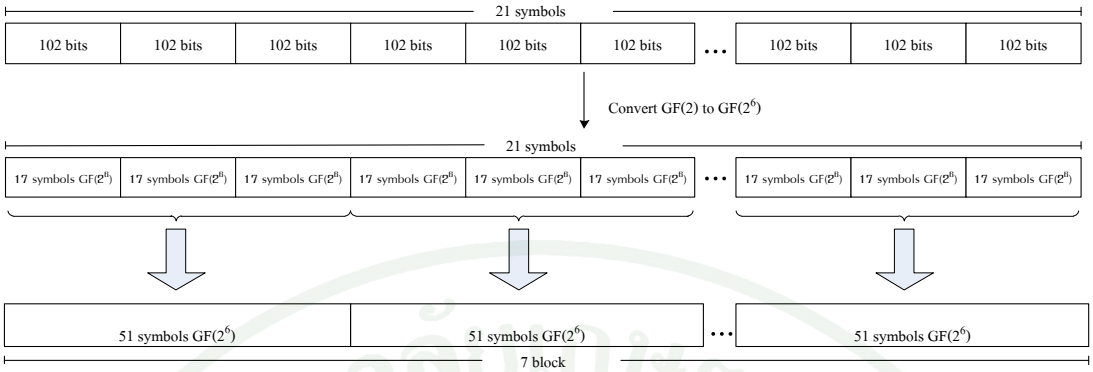
จากนั้นถอดรหัสด้วยวิเทอร์บี และนำผลลัพธ์มาเรียงกลับเป็นคำรหัสยาวสัญลักษณ์ละ 104 บิต จากนั้นถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบลแบบไม่มีข้อมูลสำรอง

4.3 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบรีดโซโลมอน ร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ขยายขนาดสัญลักษณ์



ภาพที่ 27 ระบบคอนคาทิเนตที่ใช้รหัสภายในแบบรีดโซโลมอน ร่วมกับรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ขยายขนาดสัญลักษณ์

รหัสคอนคาทิเนตที่ใช้รหัสภายในแบบรีดโซโลมอนสามารถเขียนแผนภาพได้ดังภาพที่ 27 โดยตัวถอดรหัสภายนอกใช้รหัสคอนโวลูชัน (3, 2, 2) ขนาดสัญลักษณ์ละ 102 บิต และรหัสภายในใช้รีดโซโลมอน (63,51) ซึ่งเป็น GF(2^6)โดยสามสัญลักษณ์ของรหัสภายนอกสามารถรวมเป็นรหัสภายในได้ 1 บล็อกดังภาพที่ 28



ภาพที่ 28 แสดงการแปลงคำรหัสจากไบนารีเป็นนอนไบนารีแบบ GF(2^6)

หลังจากเปลี่ยนคำรหัสให้อยู่ในรูปของ GF(2^6) จากนั้นเข้ารหัสรีดโซโลมอน (63, 51) และส่งข้อมูลผ่านแบบจำลองช่องสัญญาณต่างๆ ใช้การถอดรหัสรีดโซโลมอนโดยวิธี Algebraic ด้วยการตัดสินใจแบบหยาบ และแปลงคำรหัสจากนอนไบนารี GF(2^6) เป็นไบนารีและจัดเรียงคำรหัสเป็นสัญลักษณ์ละ 102 บิต จากนั้นนำคำรหัสไปถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบลแบบไม่มีข้อมูลสำรอง

5. จำลองช่องสัญญาณด้วย 2 สเตจ

จำลองความผิดพลาดแบบเบริสต์ด้วย 2 สเตจ โดยกำหนดความน่าจะเป็นในการเปลี่ยนจากช่วงที่ช่องสัญญาณมีประสิทธิภาพดีเป็นช่วงที่ช่องสัญญาณมีประสิทธิภาพไม่ดี หรือพารามิเตอร์ p และช่วงที่ช่องสัญญาณมีประสิทธิภาพไม่ดีเป็นช่วงที่ช่องสัญญาณมีประสิทธิภาพดี หรือพารามิเตอร์ q โดยใช้ช่วงเวลาที่สัญญาณมีค่ามากกว่ากว่าขีดแบ่งต่อวินาที และช่วงเวลาที่สัญญาณมีค่าน้อยกว่ากว่าขีดแบ่งต่อวินาที และอัตราการรับ-ส่งข้อมูลมาคำนวณหาความน่าจะเป็นในการเปลี่ยนสเตจ

ตัวอย่าง ความถี่ 2100 MHz ความเร็ว 120 กิโลเมตรต่อชั่วโมง ค่าเฟดมาจิน 25 อัตราการรับ-ส่ง 2 Mbps สามารถคำนวณค่าความถี่ดอปเพลอร์จากสมการที่ (33) ได้ 233.31 Hz ช่วงเวลาที่สัญญาณมีค่ามากกว่าขีดแบ่งต่อวินาทีคำนวณจากสมการที่ (35) ได้ 0.008935 ช่วงเวลาที่สัญญาณมีค่าน้อยกว่ากว่าขีดแบ่งต่อวินาทีคำนวณจากสมการที่ (36) ได้ 0.000322 และอัตราการเปลี่ยนต่อวินาทีคือ 108.0264 ครั้งต่อวินาที ดังนั้นจะพบว่าเมื่ออัตราการรับ-ส่งมีค่า 2 Mbps

ช่วงเวลาที่สัญญาณมีค่ามากกว่าขีดแบ่งแต่ละครั้งจะเกิดขึ้น 17,870 บิต และช่วงเวลาที่สัญญาณมีค่าน้อยกว่ากว่าขีดแบ่งแต่ละครั้งจะเกิดขึ้น 644 บิต ดังนั้นค่าพารามิเตอร์ p คือ $1/17{,}870$ เนื่องจากทุกๆ 17,870 บิตถึงจะเปลี่ยนสถานะ และค่าพารามิเตอร์ q คือ $1/644$ เนื่องจากทุกๆ 644 บิตถึงจะเปลี่ยนสถานะดังภาพที่ 29

ภาพที่ 29 การเกิดความผิดพลาดแบบเบริสต์ที่เกิดจากการจำลองช่องสัญญาณแบบ 2 สเตจ

ตารางที่ 5 ช่วงที่สัญญาณคุณภาพดี ช่วงที่สัญญาณคุณภาพไม่ดีและอัตราการเปลี่ยนต่อวินาทีในความเร็วต่างๆ

Speed(km/h)	Average Fade Duration(s)	Average Non-fade Duration(s)	Crossing Rate
30	0.001396	0.034217	28.079
40	0.001047	0.025655	37.450
50	0.000837	0.020521	46.821
60	0.000763	0.017098	55.988
70	0.000598	0.014662	65.531
80	0.000524	0.012828	74.895
90	0.000465	0.011401	84.274
100	0.000419	0.010260	93.642
110	0.000381	0.009327	103.008
120	0.000349	0.008935	107.712
130	0.000322	0.007893	121.729
140	0000299	0.007329	131.096
150	0.000279	0.006840	140.462

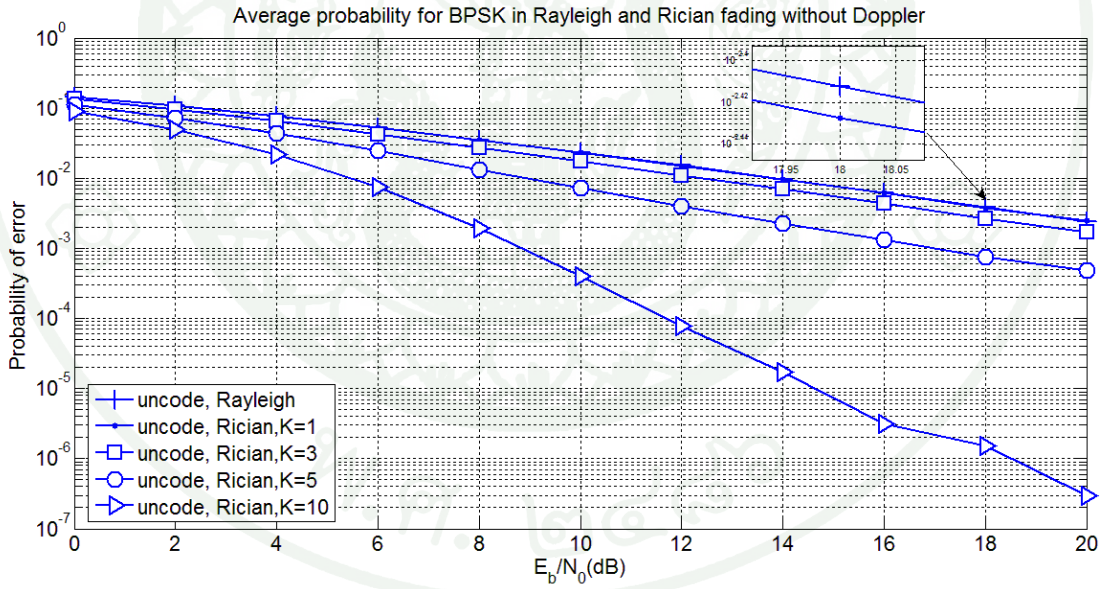
ผลและวิจารณ์

ผล

1. ผลการจำลองช่องสัญญาณ

ในการจำลองการทำงานช่องสัญญาณของรหัสคอนคาทิเนต นอกจากรหัสคอนคาทิเนตแล้ว ช่องสัญญาณก็เป็นหนึ่งในส่วนที่ต้องให้ความสนใจ เนื่องจากทำให้การจำลองระบบสมบูรณ์แบบจำลองช่องสัญญาณมีหลายแบบด้วยกัน แต่ละแบบก็มีลักษณะที่แตกต่างกันไป ต่อไปนี้คือแบบจำลองช่องสัญญาณที่ใช้ในโครงงานนี้

1.1 จำลองการจางหายแบบเรย์ลี และการจางหายแบบไรเซียนแบบไม่มีดอปเพลอร์



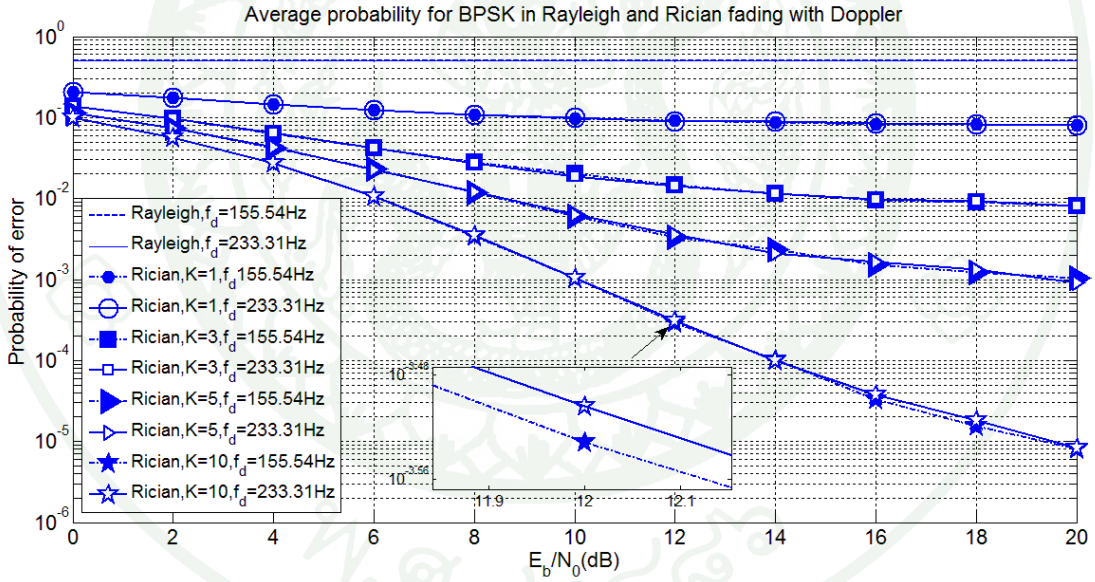
ภาพที่ 30 แสดงค่าเฉลี่ยความน่าจะเป็นความผิดพลาดของช่องสัญญาณการจางหายแบบเรย์ลีและไรเซียนที่ไม่มีดอปเพลอร์ โดยกล้ำสัญญาณและแยกสัญญาณแบบบีพีเอสเค

จากภาพที่ 30 เป็นการแสดงค่าความผิดพลาดในช่องสัญญาณการจางหายโดยไม่มีการเข้ารหัสช่องสัญญาณ จะเห็นว่าช่องสัญญาณการจางหายแบบเรย์ลี มีประสิทธิภาพที่แย่ที่สุด

เนื่องจากไม่มีสัญญาณที่เป็นเส้นทางตรง (Direct Path) เลย มีเพียงสัญญาณที่เป็นเส้นทางกระจาย (Scattering Path) เท่านั้น แต่ช่องสัญญาณการจางหายแบบไรเซียน มีสัญญาณที่เป็นเส้นทางตรงแปลผันตามพารามิเตอร์ K โดยพารามิเตอร์ K เป็นอัตราส่วนของสัญญาณที่เป็นเส้นทางตรงต่อสัญญาณที่เป็นเส้นทางกระจาย ดังนั้นเมื่อค่า K เพิ่มขึ้นทำให้สัญญาณที่เป็นเส้นทางตรงมีอัตราส่วนที่มาก จึงเป็นเหตุให้มีประสิทธิภาพช่องสัญญาณดีกว่า

1.2 จำลองการจางหายแบบเรย์ลีและไรเซียนแบบมีดอปเพลอร์

ดอปเพลอร์เกิดเมื่อภาคส่งหรือภาครับหรือทั้งสองมีการเคลื่อนที่ด้วยความเร็วใดๆ ทำให้เกิดการเฟดของช่องสัญญาณ หากเคลื่อนที่เร็วจะเกิดดอปเพลอร์มาก ทำให้เกิดความผิดพลาดมากกว่าการเคลื่อนที่ช้า



ภาพที่ 31 แสดงค่าเฉลี่ยความน่าจะเป็นความผิดพลาดของช่องสัญญาณการจางหายแบบเรย์ลี และไรเซียนที่มีดอปเพลอร์เป็น 155.54Hz และ 233.31Hz โดยกล้ำสัญญาณและแยกสัญญาณแบบบีพีเอสเค

จากภาพที่ 31 เป็นการจำลองช่องสัญญาณจำลองการจางหายแบบเรย์ลีและไรเซียนแบบมีดอปเพลอร์ ที่ความเร็ว 80 กิโลเมตรต่อชั่วโมง ($f_d = 155.54 Hz$) และที่ความเร็ว 120 กิโลเมตรต่อชั่วโมง ($f_d = 233.31 Hz$) โดยสามารถคำนวณหาค่าความถี่ดอปเพลอร์ได้จากสมการที่ (45)

โดยใช้ $f_c = 2100 MHz$

$c_0 = 3 \times 10^8$ เมตรต่อวินาที

v คือความเร็ว 80 และ 120 กิโลเมตรต่อชั่วโมง

$\alpha = 0$ คือมุมที่สัญญาณทำกับเสาตัวส่ง-ตัวรับแล้วทำให้ความถี่ดอปเพลอร์สูงสุด

คลื่นพาห์ความถี่ 2100 MHz คือย่านที่เป็นมาตรฐานโทรศัพท์เคลื่อนที่ยุคที่ 3 (3G) และประเทศไทยอนุญาติให้ใช้งานบนย่านความถี่ดังกล่าว

ช่องสัญญาณแบบมีค่าดอปเพลอร์ (เคลื่อนที่) มีค่าความผิดพลาดมากกว่าช่องสัญญาณแบบไม่มีค่าดอปเพลอร์ (ไม่เคลื่อนที่) และช่องสัญญาณที่มีค่าดอปเพลอร์มาก (เคลื่อนที่เร็ว) จะมีค่าความผิดพลาดมากกว่าช่องสัญญาณที่มีค่าดอปเพลอร์น้อย (เคลื่อนที่ช้า) ดังช่องเล็กในภาพที่ 31

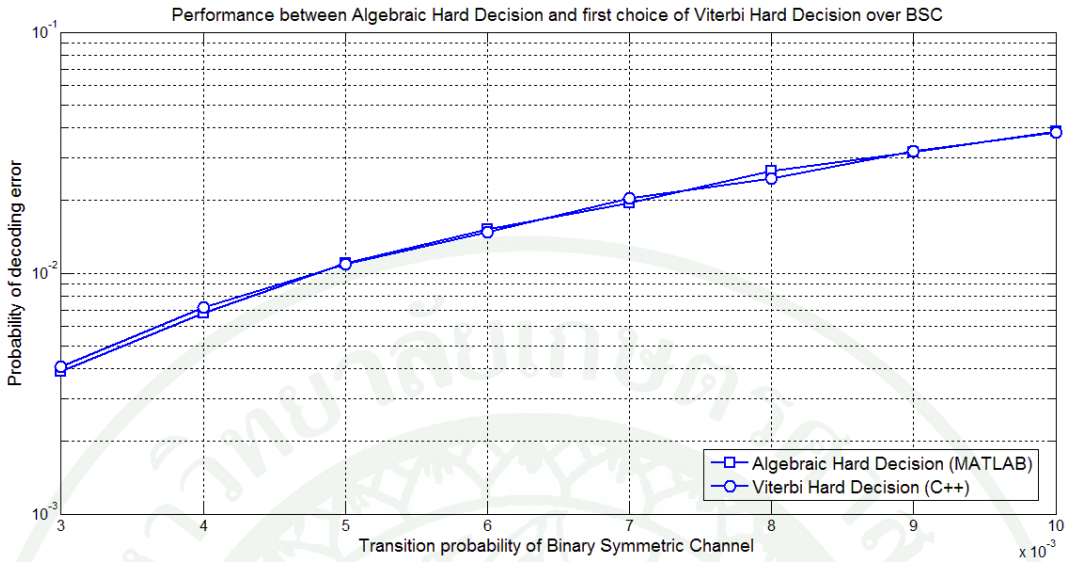
2. ผลการทดลองการปรับปรุงตัวถอดรหัสภายในแบบลิสวิเทอร์บี (List Viterbi Algorithm) สำหรับรหัสบล็อกเชิงเส้น

ทำการทดลองโดยใช้รหัสบีซีเอช(31, 26) ที่มี $G(D) = 1 + D^3 + D^5$ เปรียบเทียบผลการถอดรหัสโดย Algebraic ด้วยการตัดสินใจแบบหยาบซึ่งใช้โปรแกรม MATLAB และวิธีวีเทอร์บีด้วยการตัดสินใจแบบหยาบเช่นกัน ซึ่งใช้ C++ ด้วยโปรแกรม Microsoft Visual Studio โดยการถอดรหัสทั้งสองผ่านช่องสัญญาณแบบบีเอสซี

ตารางที่ 6 เปรียบเทียบผลการถอดรหัสแบบ Algebraic และวีเทอร์บีด้วยการตัดสินใจแบบหยาบบนช่องสัญญาณบีเอสซี

Transition Probability of BSC	Algebraic Hard Decision	1st choice of Viterbi Hard Decision
0.003	0.0039	0.0041
0.004	0.0068	0.0072
0.005	0.0110	0.0109
0.006	0.0152	0.0148
0.007	0.0195	0.0205
0.008	0.0265	0.0246
0.009	0.0318	0.0321
0.010	0.0388	0.0385

จากตารางที่ 6 เป็นการเปรียบเทียบผลของค่าความผิดพลาดหลังการถอดรหัสบีซีเอชด้วยวิธีทางพิชคณิตแบบหยาบและวิธีวีเทอร์บีแบบหยาบ ซึ่งจะเห็นว่าตัวถอดรหัสทั้งสองวิธีนั้นสามารถให้ผลลัพธ์หลังถอดรหัสได้เหมือนกัน จึงสามารถสรุปได้ว่าตัวถอดรหัสวิเทอร์บีสำหรับรหัสบล็อกที่พัฒนาขึ้นมาสามารถถอดรหัสได้อย่างถูกต้อง และสามารถเปรียบเทียบผลดังกล่าวได้ดังภาพที่ 32



ภาพที่ 32 เปรียบเทียบผลการถอดรหัสแบบ Algebraic และวีเทอร์บีด้วยการตัดสินใจแบบหยาบบนช่องสัญญาณบีเอสซี

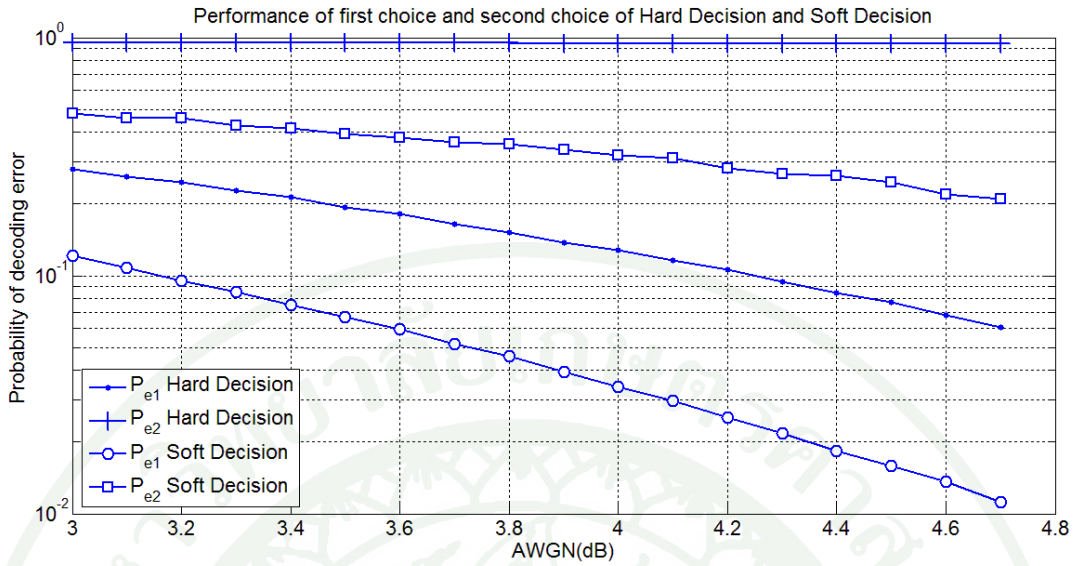
ที่มา: Tuntoolavest *et al.* (2011)

จากภาพที่ 32 จะเห็นว่าตัวถอดรหัสทั้งสองให้ผลลัพธ์ที่ใกล้เคียงกันในทุกๆค่าของความน่าจะเป็นความผิดพลาดของช่องสัญญาณแบบบีเอสซี โดยยิ่งเพิ่มความผิดพลาดของช่องสัญญาณค่าความผิดพลาดในการถอดรหัสช่องสัญญาณก็เพิ่มตามดังทฤษฎี

เมื่อแน่ใจว่าตัวถอดรหัสวีเทอร์บีสำหรับรหัสบล็อกสามารถถอดรหัสได้ถูกต้อง จึงพัฒนาต่อเป็นลิสวีเทอร์บีทั้งการตัดสินใจแบบหยาบและแบบอ่อนที่ให้สองผลลัพธ์โดยใช้การจำลองบนช่องสัญญาณไวส์เกาท์เซียนแบบบวก ซึ่งเป็นช่องสัญญาณพื้นฐานได้ผลดังตารางที่ 7

ตารางที่ 7 เปรียบเทียบผลการถอดรหัสของลิสวิเทอร์บีแบบสองผลลัพธ์ของการตัดสินใจแบบหยาบและแบบอ่อนบนช่องสัญญาณไวส์เกาท์เซียนแบบบวก

AWGN (E_b/N_0)	Hard Decision		Soft Decision	
	P_{e1}	P_{e2}	P_{e1}	P_{e2}
3.0	0.2803	0.9539	0.1209	0.4811
3.1	0.2598	0.9531	0.1083	0.4579
3.2	0.2459	0.9520	0.0952	0.4579
3.3	0.2272	0.9520	0.0855	0.4289
3.4	0.2131	0.9506	0.0754	0.4161
3.5	0.1943	0.9510	0.0672	0.3948
3.6	0.1818	0.9486	0.0596	0.3813
3.7	0.1652	0.9481	0.0515	0.3638
3.8	0.1520	0.9481	0.0458	0.3567
3.9	0.1371	0.9448	0.0393	0.3375
4.0	0.1281	0.9465	0.0340	0.3199
4.1	0.1157	0.9433	0.0298	0.3114
4.2	0.1058	0.9442	0.0254	0.2833
4.3	0.0944	0.9455	0.0217	0.2674
4.4	0.0849	0.9434	0.0184	0.2638
4.5	0.0771	0.9420	0.0159	0.2470
4.6	0.0682	0.9397	0.0136	0.2188
4.7	0.0609	0.9406	0.0112	0.2099



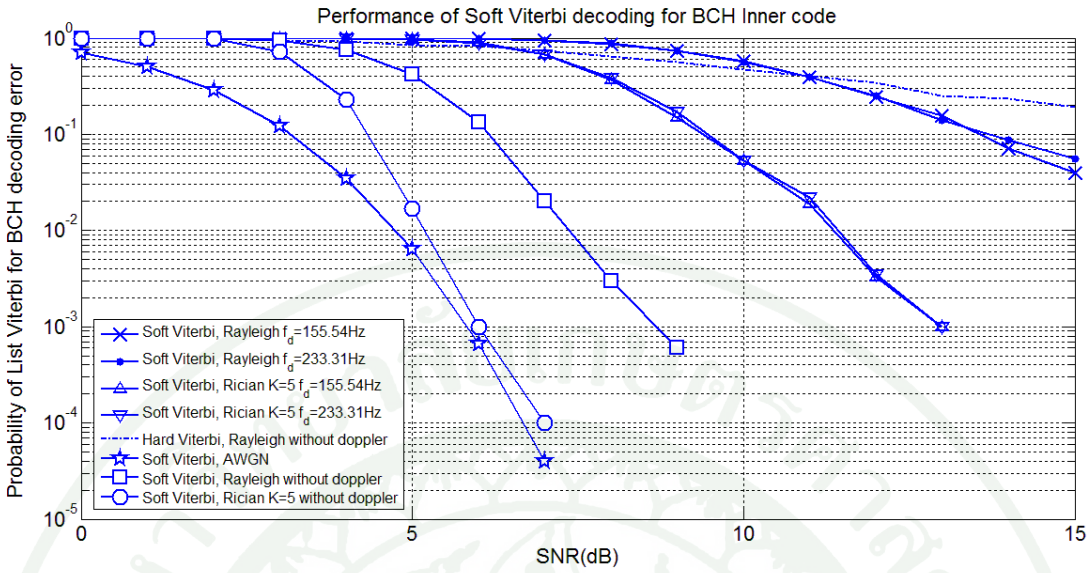
ภาพที่ 33 เปรียบเทียบผลการถอดรหัสของลิสวิเทอร์บีแบบสองผลลัพธ์ของการตัดสินใจแบบหยาบและแบบอ่อนบนช่องสัญญาณไวส์เกาท์เซียนแบบบวก

ที่มา: Tuntoolavest *et al.* (2011)

ภาพที่ 33 เปรียบเทียบผลของตัวถอดรหัสลิสวิเทอร์บีแบบสองผลลัพธ์ทั้งการตัดสินใจแบบหยาบและการตัดสินใจแบบอ่อน โดย P_{e1} คือค่าความผิดพลาดของผลลัพธ์ที่ดีที่สุด และ P_{e2} คือค่าความผิดพลาดของผลลัพธ์ที่ดีรองลงมาเมื่อผลลัพธ์ที่ดีที่สุดผิด เมื่อเทียบผล P_{e1} และ P_{e2} ของทั้งสองการตัดสินใจพบว่าค่า P_{e1} ของแต่ละการตัดสินใจดีกว่า P_{e2} และเมื่อเทียบ P_{e1} ของการตัดสินใจแบบหยาบและการตัดสินใจแบบอ่อน จะเห็นว่า P_{e1} ของการตัดสินใจแบบอ่อนจะมีประสิทธิภาพในการถอดรหัสที่ดีกว่า

ที่เป็นเช่นนี้เนื่องจากการตัดสินใจแบบอ่อนมีการใช้ระดับในการตัดสินใจที่มากกว่า เช่น อาจมี 4 หรือ 8 ระดับในการตัดสินใจ ต่างกับการตัดสินใจแบบหยาบที่มีเพียง 2 ระดับ คือ 0 และ 1 เท่านั้น ดังนั้นด้วยการที่ความซับซ้อนในการทำงานต่างกันประสิทธิภาพของการตัดสินใจทั้งสองจึงต่างกัน

จากนั้นทำการรับ - ส่งข้อมูลในแบบจำลองช่องสัญญาณอื่นได้แก่แบบจำลองการจางหายแบบเรย์ลี และแบบจำลองการจางหายแบบไรเซียนทั้งที่มีดอปเพลอร์และไม่มี



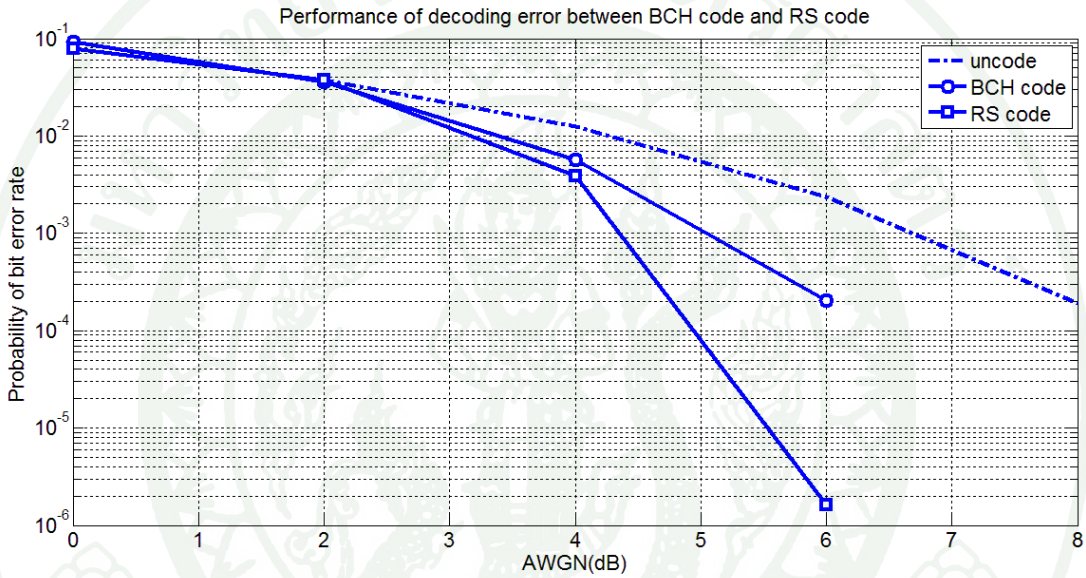
ภาพที่ 34 แสดงประสิทธิภาพการถอดรหัสบีซีเอชโดยวิเทอร์บีด้วยการตัดสินใจแบบหยาบและแบบอ่อน บนช่องสัญญาณไวส์เกาท์เซียนแบบบวก ช่องสัญญาณการจางหายแบบเรย์ลีการจางหายแบบไรเซียนที่มีดอปเพลอร์และไม่มีดอปเพลอร์

ที่มา: Thonchai *et al.* (2011)

จากภาพที่ 34 เป็นการแสดงการเปรียบเทียบการถอดรหัสบีซีเอชด้วยวิเทอร์บีเมื่อเปลี่ยนช่องสัญญาณไวส์เกาท์เซียนแบบบวก การจางหายแบบเรย์ลี การจางหายแบบไรเซียนที่มีดอปเพลอร์และไม่มีดอปเพลอร์ โดยใช้การตัดสินใจแบบหยาบและแบบอ่อนบนช่องสัญญาณเรย์ลีที่ไม่มีดอปเพลอร์ พบว่าการถอดรหัสโดยใช้การตัดสินใจแบบอ่อนมีประสิทธิภาพดีกว่าการถอดรหัสโดยใช้การตัดสินใจแบบหยาบ การถอดรหัสบนช่องสัญญาณไวส์เกาท์เซียนแบบบวกมีประสิทธิภาพดีกว่าการถอดรหัสบนช่องสัญญาณที่มีการจางหาย และการถอดรหัสบนช่องสัญญาณที่มีการจางหายแบบไม่มีดอปเพลอร์มีประสิทธิภาพดีกว่าแบบมีดอปเพลอร์

3. ผลการใช้รหัสภายในแบบรีดโซโลมอน

เลือกใช้รหัสรีดโซโลมอน (63, 51) ซึ่งมีอัตรารหัสใกล้เคียงกับรหัสบีซีเอช (31, 26) จากนั้นถอดรหัสทั้งสองโดยวิธี Algebraic ด้วยการตัดสินใจแบบหยาบและเปรียบเทียบค่าความผิดพลาดต่อบิตเพื่อเปรียบเทียบประสิทธิภาพของการใช้รหัสภายของรหัสบล็อกเชิงเส้นในแบบไบนารีและแบบนอนไบนารี



ภาพที่ 35 เปรียบเทียบผลการถอดรหัสของรหัสบีซีเอชกับรหัสรีดโซโลมอนบนช่องสัญญาณไวส์เกาท์เซียนแบบบวก

จากภาพที่ 35 เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสช่องสัญญาณบีซีเอชและรีดโซโลมอนบนช่องสัญญาณไวส์เกาท์เซียนแบบบวก โดยใช้การถอดรหัสแบบ Algebraic ด้วยการตัดสินใจแบบหยาบ โดยใช้ฟังก์ชัน(Function) จากโปรแกรม MATLAB จะเห็นว่าเมื่อเทียบค่าความผิดพลาดต่อบิต รหัสรีดโซโลมอนมีแนวโน้มในการมีประสิทธิภาพที่ดีกว่ารหัสบีซีเอช

4. ผลการเปลี่ยนขนาดสัญลักษณ์ของรหัสเวกเตอร์ซิมโบลแบบไม่มีข้อมูลลำรอง

จากทฤษฎีไม่ว่าขนาดสัญลักษณ์จะเป็นเท่าใด แต่หากเกิดความผิดพลาดที่สัญลักษณ์เดียวกันจะต้องสามารถแก้ไขข้อมูลได้ หรือไม่ได้เหมือนๆกัน ดังนั้นจึงทดลองสุ่มสัญลักษณ์ที่ผิดเหมือนๆกันของรหัสเวกเตอร์ขนาดสัญลักษณ์ 64 บิต และรหัสเวกเตอร์ขนาดสัญลักษณ์ 128 บิต และเทียบค่าความผิดพลาดได้ดังตารางที่ 8

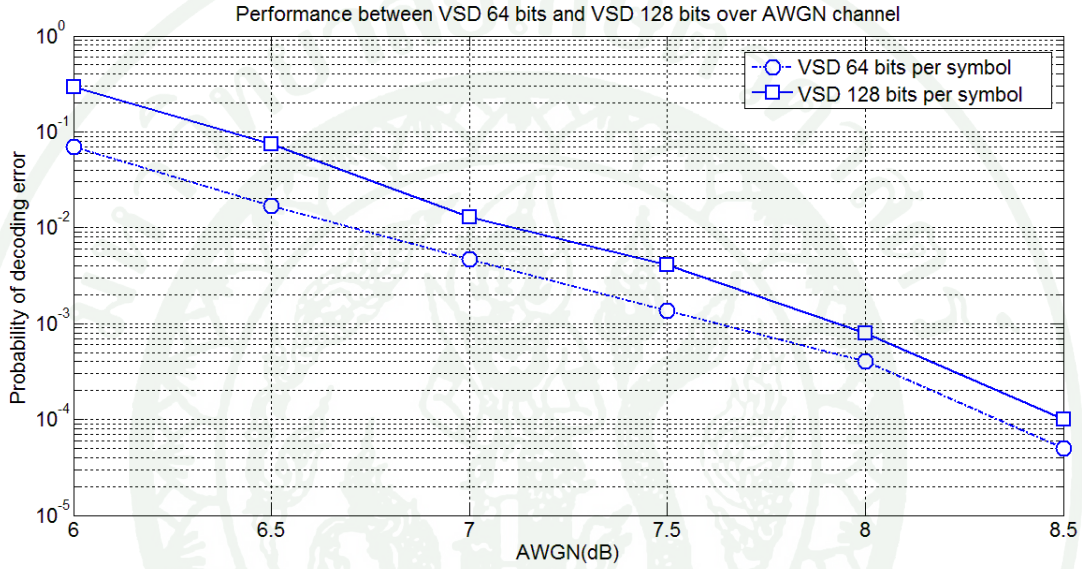
ตารางที่ 8 แสดงค่าความผิดพลาดในการถอดรหัสของรหัสเวกเตอร์ขนาดสัญลักษณ์ 64 บิต และรหัสเวกเตอร์ขนาดสัญลักษณ์ 128 บิต เมื่อสุ่มการเกิดความผิดพลาดของสัญลักษณ์

ความน่าจะเป็นความผิดพลาดของสัญลักษณ์	VSD 64 บิต	VSD 128 บิต
0.001	0	0
0.005	0	0
0.01	0	0
0.02	0	0
0.03	0.0001	0.0001
0.04	0.0004	0.0004
0.05	0.0004	0.0004
0.06	0.0010	0.0010
0.07	0.0018	0.0018
0.08	0.0034	0.0034
0.09	0.0064	0.0064
0.1	0.0066	0.0066

หมายเหตุ ใช้จำนวนครั้งในการทำงาน 10,000 ครั้ง

จากตารางที่ 8 สรุปได้ว่าตัวถอดรหัสเวกเตอร์ซิมโบลสามารถทำงานได้อย่างถูกต้อง เนื่องจากตัวถอดรหัสเวกเตอร์ซิมโบลทั้งสองซึ่งขนาดสัญลักษณ์ต่างกันสามารถถอดรหัสได้เหมือนกันทุกประการ และเมื่อเพิ่มค่าการเกิดความผิดพลาดของสัญลักษณ์ ค่าความผิดพลาดของการถอดรหัสของทั้งสองก็เพิ่มตามด้วย

เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณไวส์เกาท์เซียนแบบบวก ดังภาพที่ 36 ซึ่งมีลักษณะความผิดพลาดแบบสุ่ม พบว่ารหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 128 บิต มีโอกาสเกิดความผิดพลาดมากกว่าเนื่องจากสัญลักษณ์ขนาดใหญ่ ทำให้ในหนึ่งสัญลักษณ์มีโอกาสสุ่มเกิดความผิดพลาดมากกว่า

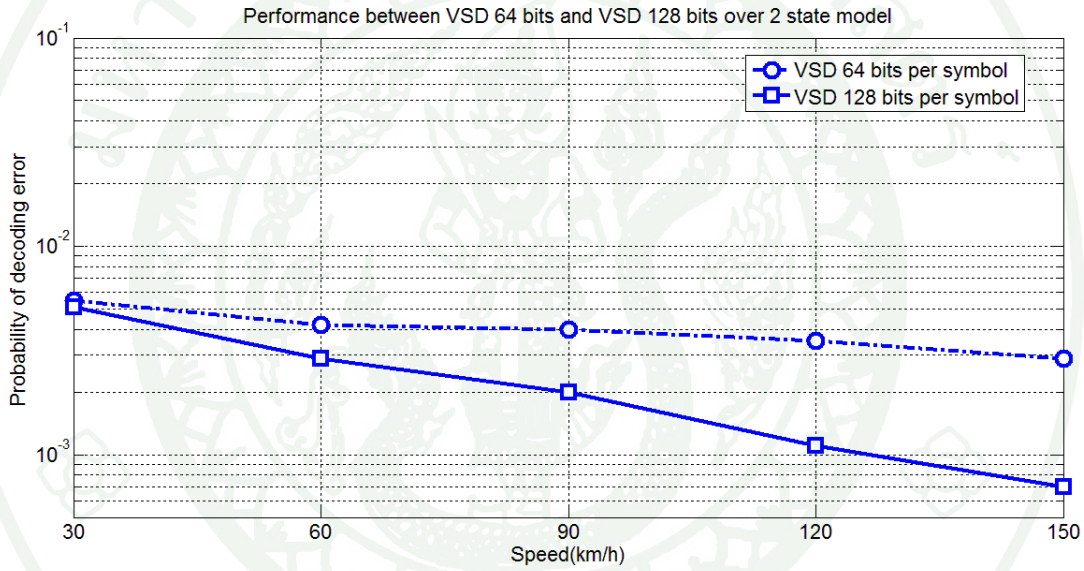


ภาพที่ 36 เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณไวส์เกาท์เซียนแบบบวก

ตารางที่ 9 เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณไวส์เกาท์เซียนแบบบวก

AWGN(E_b/N_0)	VSD 64 bits per symbol	VSD 128 bits per symbol
6	0.06925	0.296
6.5	0.01700	0.0752
7	0.00465	0.013
7.5	0.00135	0.0041
8	0.00040	0.0008
8.5	0.00005	0.0001

เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณที่มีการผิดพลาดแบบเบริสต์ โดยใช้การจำลองช่องสัญญาณแบบ 2 สเตจ (state) ที่ความน่าจะเป็นในการเปลี่ยนสถานะอ้างอิงตามความเร็วที่เปลี่ยนแปลงไป โดยเมื่อความเร็วต่ำความผิดพลาดจะเกิดไม่บ่อย แต่จะเกิดติดต่อกันยาวหลายสัญลักษณ์ ทำให้ไม่สามารถแก้ไขได้ความผิดพลาดได้ แต่เมื่อความเร็วเพิ่มขึ้น ความผิดพลาดจะเกิดขึ้นบ่อย แต่เกิดติดต่อกันแบบสั้นๆ ทำให้ความผิดพลาดกระจายอยู่ในหลายสัญลักษณ์ จึงมีโอกาสแก้ไขความผิดพลาดได้มากกว่า และยิ่งขนาดสัญลักษณ์ใหญ่ก็มีโอกาสมากขึ้นที่ความผิดพลาดจะรวมอยู่ภายในสัญลักษณ์เดียว ดังภาพที่ 37



ภาพที่ 37 เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณที่มีการผิดพลาดแบบเบริสต์ โดยใช้การจำลองช่องสัญญาณแบบ 2 สเตจ

จากภาพที่ 37 จะเห็นว่าที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง ประสิทธิภาพของการถอดรหัสทั้งสองแทบไม่มีความแตกต่างกัน เนื่องจากความผิดพลาดที่เกิดขึ้น เกิดติดต่อกันหลายสัญลักษณ์ ทำให้การใช้สัญลักษณ์ขนาดใหญ่ไม่สามารถช่วยได้ แต่เมื่อความเร็วเพิ่มขึ้น ความผิดพลาดที่เกิดจะเกิดบ่อยขึ้นแต่ความยาวความผิดพลาดสั้นลง ทำให้เริ่มเห็นผลของขนาดสัญลักษณ์ที่แตกต่างกัน

ตารางที่ 10 เปรียบเทียบประสิทธิภาพการถอดรหัสของรหัสเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาด 64 บิต และสัญลักษณ์ขนาด 128 บิต บนช่องสัญญาณที่มีการผิดพลาดแบบเบริสต์ โดยใช้การจำลองช่องสัญญาณแบบ 2 สเตจ

Speed (km/h)	VSD 64 bits per symbol	VSD 128 bits per symbol
30	0.00550	0.0051
40	0.00490	0.0047
50	0.00450	0.0040
60	0.00442	0.0029
70	0.00420	0.0030
80	0.00405	0.0021
90	0.00400	0.0020
100	0.00405	0.0019
110	0.00390	0.0018
120	0.00350	0.0011
130	0.00315	0.0007
140	0.00295	0.0007
150	0.00290	0.0007

5. ผลของรหัสคอนคาทิเนตที่ใช้รหัสภายในเป็นรหัสบีซีเอชและรีดโซโลมอน ตัวเข้ารหัสภายนอกเป็นคอนโวลูชัน และตัวถอดรหัสภายนอกเป็นเวกเตอร์ซิมโบลแบบปรับขนาดสัญลักษณ์ได้

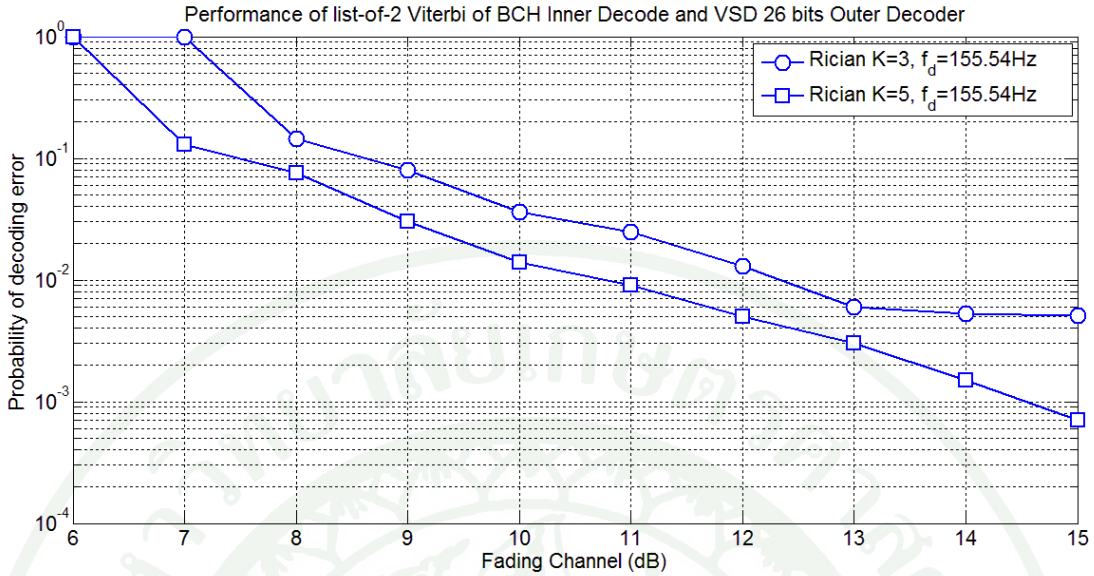
จากการบันทึกผลการทำงานแยกส่วน ทั้งรหัสภายในแบบบีซีเอชและรีดโซโลมอน รหัสภายนอกแบบเวกเตอร์ซิมโบลที่ปรับขนาดสัญลักษณ์ได้ รวมถึงแบบจำลองช่องสัญญาณต่างๆ เมื่อระบบทุกส่วนสามารถทำงานได้อย่างถูกต้องแล้วจึงนำมาใช้ร่วมกันเพื่อหาประสิทธิภาพโดยรวมของระบบรหัสคอนคาทิเนตได้ดังนี้

5.1 รหัสคอนคาทิเนตที่ใช้รหัสภายในแบบบีซีเอช

จากการถอดรหัสบีซีเอชด้วยลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อนบนช่องสัญญาณ สามารถนำผลดังกล่าวมาถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบลแบบมีข้อมูลตัวเลือก โดยใช้ช่องสัญญาณที่มีการจางหาย เนื่องจากช่องสัญญาณดังกล่าวมีความผิดพลาดแบบสุ่มจากไวส์เกาท์เซียนแบบบวก และมีความผิดพลาดแบบเบริสต์จากการเฟดของช่องสัญญาณ ดังนั้นจึงเหมาะจะใช้ในการจำลองระบบคอนคาทิเนต เพื่อให้รหัสภายในแก้ไขความผิดพลาดแบบสุ่มและรหัสภายนอกแก้ไขความผิดพลาดแบบเบริสต์ โดยใช้ตัวเข้ารหัสภายนอกเป็นคอนโวลูชัน (3, 2, 2) สัญลักษณ์ละ 26 บิต จากนั้นเข้ารหัสภายในด้วยบีซีเอช จากนั้นผ่านช่องสัญญาณที่มีการจายหายและถอดรหัสโดยลิสวีเทอร์บีด้วยการตัดสินใจแบบอ่อนแบบสองผลลัพธ์ และถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบลแบบมีข้อมูลทางเลือก ได้ผลดังตารางที่ 11 และภาพที่ 38

ตารางที่ 11 ผลการถอดรหัสคอนคาทิเนตโดยใช้รหัสบีซีเอชภายในและรหัสเวกเตอร์ซิมโบลภายนอก

snr	Rician($K=3, f_d=155.54$Hz) 2^{nd} choice – VSD 26 bits	Rician($K=5, f_d=155.54$Hz) 2^{nd} choice – VSD 26 bits
6	1	1
7	1	0.13
8	0.144	0.075
9	0.0800	0.03
10	0.036	0.014
11	0.0250	0.009
12	0.0130	0.005
13	0.0060	0.003
14	0.0053	0.0015
15	0.0051	0.0007



ภาพที่ 38 ผลการถอดรหัสคอนคาทิเนตโดยใช้รหัสบีซีเอชภายในและรหัสเวกเตอร์ซิมโบลภายนอก

จากภาพที่ 38 แสดงผลรหัสคอนคาทิเนตที่ใช้รหัสภายในเป็นรหัสบีซีเอช (31, 26) และถอดรหัสด้วยการตัดสินใจแบบอ่อนด้วยลิสวีเทอร์บี และใช้สองผลลัพธ์ถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบล โดยผ่านช่องสัญญาณจางหายแบบไรเซียนที่มีค่าพารามิเตอร์ K เป็น 3 และ 5 และค่าดอปเพลอร์ 155.54Hz

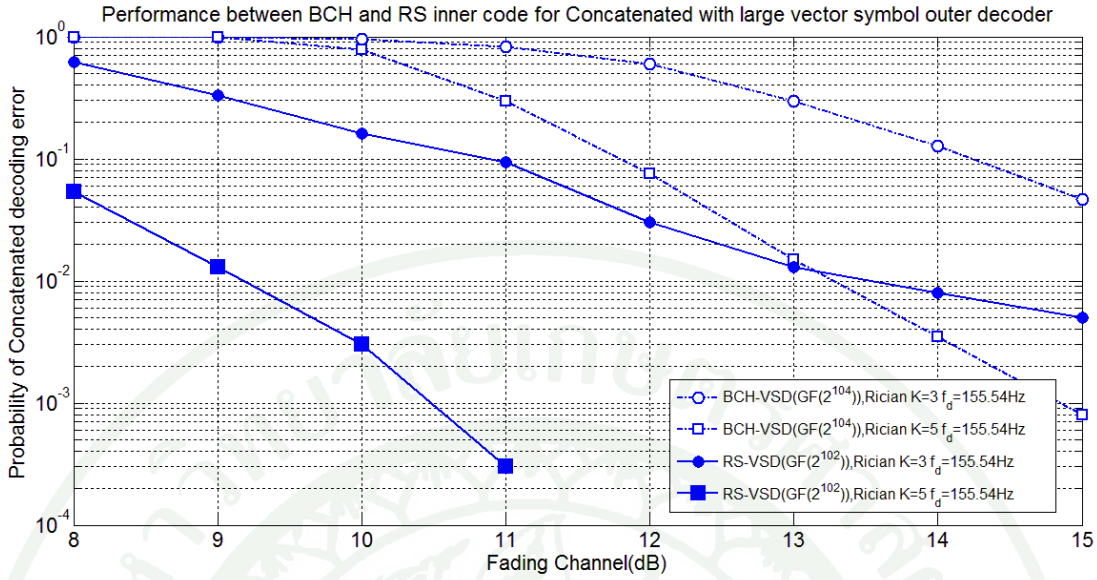
5.2 รหัสคอนคาทิเนตที่ใช้รหัสภายในแบบรีดโซโลมอน

ใช้รหัสภายในเป็นรีดโซโลมอน (63, 51) ที่มีค่า $GF(2^6)$ และถอดรหัสภายในด้วยวิธี Algebraic ด้วยการตัดสินใจแบบหยาบ จากนั้นถอดรหัสภายนอกด้วยเวกเตอร์ซิมโบลแบบไม่มีชุดข้อมูลสำรองโดยใช้สัญลักษณ์($GF(2^{102})$)ขนาดใหญ่ เปรียบเทียบประสิทธิภาพของการใช้รหัสภายในแบบไบนารีและนอนไบนารีโดยเทียบกับการใช้รหัสภายในด้วยบีซีเอช (31, 26) ที่ถอดรหัสด้วยการตัดสินใจแบบอ่อนของวิเทอร์บีโดยใช้ผลลัพธ์ที่ดีที่สุด

ตารางที่ 12 ผลการเปรียบเทียบประสิทธิภาพรหัสคอนคาทิเนตโดยใช้รหัสรีดโซโลมอนและรหัสบีซีเอชภายในและรหัสเวกเตอร์ซิมโบลภายนอก

snr	Rician(K=3,f_d=155.54Hz)		Rician(K=5,f_d=155.54Hz)	
	BCH-VSD(104)	RS-VSD(102)	BCH-VSD(104)	RS-VSD(102)
8	1	0.625	1	0.054
9	1	0.332	0.986	0.013
10	0.950	0.162	0.780	0.003
11	0.824	0.094	0.295	0.0003
12	0.601	0.030	0.075	0
13	0.295	0.013	0.015	0
14	0.128	0.008	0.0035	0
15	0.0465	0.005	0.0008	0

หมายเหตุ RS-VSD(102) ใช้จำนวนครั้งในการทำงาน 100,000 ครั้ง



ภาพที่ 39 ผลการเปรียบเทียบประสิทธิภาพรหัสคอนคาทิเนตโดยใช้รหัสรีดโซโลมอนและรหัสบีซีเอชภายในและรหัสเวกเตอร์ซิมโบลภายนอก

จากภาพที่ 39 เปรียบเทียบรหัสภายในแบบบีซีเอช (31, 26) ที่ถอดรหัสด้วยการตัดสินใจแบบอ่อนของวิเทอร์บี กับรหัสรีดโซโลมอน (63, 51) ที่มี $GF(2^6)$ ที่ใช้การตัดสินใจแบบหยาบของ Algebraic พบว่าประสิทธิภาพรหัสคอนคาทิเนตที่ใช้ขนาดสัญลักษณ์ใกล้เคียงกัน รหัสภายในแบบรีดโซโลมอนแม้ถอดรหัสด้วยการตัดสินใจแบบหยาบ แต่ให้ประสิทธิภาพที่ดีกว่า รหัสภายในแบบบีซีเอชที่ถอดรหัสด้วยการตัดสินใจแบบอ่อนด้วยวิเทอร์บี

วิจารณ์

จากผลการทดลองสามารถแบ่งเนื้อหาเพื่อวิจารณ์ผลได้ดังนี้

1. รหัสคอนคาทิเนตที่ใช้การถอดรหัสภายในด้วยลิสวิเทอร์บีสำหรับรหัสบีซีเอช (31, 26) และรหัสภายนอกเป็นเวกเตอร์ซิมโบล

จากการพัฒนาตัวถอดรหัสลิสวิเทอร์บีสำหรับรหัสบล็อกเชิงเส้น สามารถถอดรหัสบล็อกโดยใช้วิเทอร์บีได้ถูกต้อง เนื่องจากผลการถอดรหัสด้วยการตัดสินใจแบบหยาบของวิธี Algebraic และวิธีวิเทอร์บีตรงกัน และพัฒนาเป็นตัวถอดรหัสลิสวิเทอร์บีด้วยการตัดสินใจแบบอ่อนได้ ทำให้ตัวถอดรหัสลิสวิเทอร์บีสามารถถอดรหัสทั้งรหัสบล็อกและรหัสคอนโวลูชัน เพียงกำหนดการเปลี่ยนสถานะให้ถูกต้องเพื่อใช้กับแผนภาพเทรลลิส และยังสามารถนำข้อมูลสำรองไปใช้ในการถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลด้วย

การพัฒนาตัวถอดรหัสลิสวิเทอร์บีสำหรับรหัสบล็อกเชิงเส้นทำให้รหัสคอนคาทิเนตที่มีการรวมกันของตัวถอดรหัสเป็นลิสวิเทอร์บีและเวกเตอร์ซิมโบลสามารถเลือกใช้ตัวเข้ารหัสทั้งภายนอกและภายในเป็นรหัสบล็อกเชิงเส้นหรือคอนโวลูชันก็ได้ ทำให้สามารถนำรหัสคอนคาทิเนตไปใช้งานได้ง่ายและเลือกใช้ได้อย่างเหมาะสม เพราะสามารถเลือกได้ว่าจะเข้ารหัสแบบใด ถึงแม้การเลือกใช้รหัสภายในเป็นรหัสบีซีเอช (31, 26) จะสามารถแก้ไขคำผิดพลาดได้เพียงหนึ่งบิต แต่การถอดรหัสด้วยการตัดสินใจแบบอ่อนสามารถช่วยเพิ่มประสิทธิภาพได้ หรืออาจเลือกเปลี่ยนอัตรารหัสเป็นบีซีเอช (31, 21) หรือ (31, 16) ก็ได้ แต่ต้องเพิ่มความซับซ้อนในส่วนของการถอดรหัสด้วยวิเทอร์บีเนื่องจากมีจำนวนสเตจเพิ่มขึ้นจากสูตร $R=2^{n-k}$ ซึ่ง R คือค่าจำนวนสเตจ

2. รหัสคอนคาทิเนตที่ใช้รหัสภายในเป็นรหัสรีดโซโลมอนและรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ขยายขนาดสัญลักษณ์

รหัสรีดโซโลมอนเป็นรหัสที่มีประสิทธิภาพในการถอดรหัสที่ดี และนิยมใช้อย่างแพร่หลาย เมื่อเทียบความผิดพลาดต่อบิตในการถอดรหัสแบบ Algebraic ของรหัสรีดโซโลมอนและรหัสบีซีเอช พบว่ารหัสรีดโซโลมอนมีค่าความผิดพลาดที่น้อยกว่ารหัสบีซีเอช เมื่อใช้อัตรารหัสใกล้เคียงกัน แต่ข้อเสียของการใช้รหัสรีดโซโลมอนเป็นรหัสภายในคือยากต่อการถอดรหัสด้วย

ลิสวิเทอร์บีเนื่องจากเป็นรหัสแบบนอนไบนารี แต่สามารถพัฒนาตัวถอดรหัสให้ตัดสินใจแบบอ่อนตามงานวิจัยต่างๆที่มีผู้นำเสนอ เพื่อเพิ่มประสิทธิภาพในการถอดรหัสให้มากขึ้น

ดังนั้นงานวิจัยนี้จึงพัฒนารหัสคอนคาทิเนตโดยใช้รหัสรีดโซโลมอน (63, 51) ของ $GF(2^6)$ ซึ่งมีอัตรารหัสใกล้เคียงกับรหัสบีซีเอช (31, 26) โดยใช้ตัวถอดรหัสภายนอกเป็นเวกเตอร์ซิมโบลที่สัญลักษณ์ขนาดใหญ่แบบไม่มีข้อมูลตัวเลือก และเปรียบเทียบประสิทธิภาพของระบบเมื่อทั้งคู่เป็นรหัสภายใน โดยการถอดรหัสรีดโซโลมอนด้วยการตัดสินใจแบบหยาบของ Algebraic แต่การถอดรหัสของบีซีเอชใช้การตัดสินใจแบบอ่อนด้วยวีเทอร์บี ขยายขนาดของสัญลักษณ์ภายนอกให้สอดคล้องกับรหัสภายใน และเทียบผลของระบบทั้งสอง โดยพบว่าแม้รหัสรีดโซโลมอนจะถอดรหัสโดยตัดสินใจแบบหยาบ แต่ระบบรหัสคอนคาทิเนตที่ใช้รหัสภายในแบบรีดโซโลมอนมีประสิทธิภาพดีกว่าระบบรหัสคอนคาทิเนตที่ใช้รหัสภายในเป็นบีซีเอชที่ถอดรหัสโดยใช้การตัดสินใจแบบอ่อนด้วยวีเทอร์บี

3. การขยายขนาดสัญลักษณ์ของรหัสภายนอก

จากแนวคิดที่ว่าอัตราการรับ-ส่งข้อมูลเพิ่มขึ้น ความผิดพลาดของข้อมูลก็เพิ่มขึ้นด้วย และเป็นความผิดพลาดแบบเบริสต์ที่ยาวขึ้น ดังนั้นการใช้สัญลักษณ์ขนาดเล็ก ทำให้ความผิดพลาดแบบเบริสต์ครอบคลุมหลายสัญลักษณ์ติดต่อกัน และทำให้ตัวถอดรหัสภายนอกไม่สามารถแก้ไขความถูกต้องได้ ดังนั้นจึงขยายขนาดสัญลักษณ์ให้ใหญ่ขึ้น ทำให้สามารถลดจำนวนสัญลักษณ์ที่ผิดพลาดติดต่อกันดังกล่าวได้ และประสิทธิภาพในการถอดรหัสเพิ่มขึ้น แต่ความยาวความผิดพลาดกับขนาดสัญลักษณ์จำเป็นต้องสอดคล้องกัน กล่าวคือหากความผิดพลาดยาวกว่าขนาดสัญลักษณ์มากๆ การเพิ่มขนาดสัญลักษณ์อาจไม่สามารถช่วยเพิ่มประสิทธิภาพได้ เนื่องจากลักษณ์ยังผิดติดต่อกันเหมือนเดิม หรือความผิดพลาดสั้นกว่าขนาดสัญลักษณ์มากๆ การเพิ่มขนาดสัญลักษณ์อาจไม่เกิดประโยชน์ เพราะใช้ขนาดสัญลักษณ์ที่เล็กกว่าก็สามารถแก้ไขความผิดพลาดได้เหมือนกัน ดังนั้นจึงควรหาความสัมพันธ์ระหว่างขนาดสัญลักษณ์กับความยาวความผิดพลาด เพื่อใช้ในการพัฒนาระบบรหัสรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลต่อไป

สรุปและข้อเสนอแนะ

สรุป

งานวิจัยนี้ออกแบบและจำลองระบบรหัสคอนคาทิเนตที่ใช้ตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลสำหรับรหัสคอนโวลูชัน (3, 2, 2) โดยพัฒนาส่วนของรหัสภายใน โดยการใช้ตัวถอดรหัสภายในแบบลิสวีเทอร์บีสำหรับรหัสบล็อก และการประยุกต์ใช้รหัสภายในแบบนอนไบนารี โดยการจำลองระบบรหัสคอนคาทิเนตบนช่องสัญญาณไร้สายเคลื่อนที่

จากการทดลองสามารถพัฒนาตัวถอดรหัสภายในด้วยลิสวิเทอร์บีสำหรับรหัสบล็อกได้ โดยสามารถถอดรหัสได้ทั้งการตัดสินใจแบบหยาบและการตัดสินใจแบบอ่อน และนำข้อมูลตัวเลือกที่ได้จากรหัสภายในไปใช้ร่วมกับรหัสภายนอกที่เป็นเวกเตอร์ซิมโบลซึ่งใช้เป็นรหัสภายนอกของระบบรหัสคอนคาทิเนตและให้ประสิทธิภาพที่ดี เปลี่ยนรหัสภายในของระบบรหัสคอนคาทิเนตจากเดิมที่ใช้รหัสแบบไบนารีเป็นแบบนอนไบนารี โดยใช้รหัสรีดโซโลมอน (63, 51) ซึ่งมี $GF(2^6)$ สุดท้ายนำรหัสบีซีเอชและรหัสรีดโซโลมอนไปใช้ร่วมกับตัวถอดรหัสภายนอกแบบเวกเตอร์ซิมโบลที่ใช้สัญลักษณ์ขนาดใหญ่

ผลการทดลองสามารถสรุปว่าสามารถพัฒนาตัวถอดรหัสลิสวีเทอร์บีสำหรับรหัสบล็อกได้ อีกทั้งการใช้ตัวถอดรหัสภายในแบบลิสวีเทอร์บีด้วยการตัดสินใจแบบอ่อนเพิ่มประสิทธิภาพในการถอดรหัสบีซีเอชได้ และการเปลี่ยนรหัสภายในเป็นนอนไบนารีควบคู่กับการขยายขนาดสัญลักษณ์ภายนอกเหมาะสำหรับช่องสัญญาณไร้สายแบบเคลื่อนที่มากกว่าการใช้รหัสภายในแบบไบนารี เนื่องจากช่องสัญญาณมีคุณภาพต่ำและเกิดความผิดพลาดแบบเบริสต์ การแก้ไขความผิดพลาดทั้งสัญลักษณ์ของรหัสแบบนอนไบนารีจึงมีโอกาสแก้ไขความผิดพลาดได้มากกว่าการแก้ไขความผิดพลาดเป็นบิตของรหัสไบนารี

ข้อเสนอแนะ

การจำลองการทำงานต่างๆไม่ว่าจะเป็นการเข้ารหัส ถอดรหัส หรือช่องสัญญาณโดยใช้โปรแกรม MATLAB ทำได้สะดวกเนื่องจากบางส่วนสามารถเรียกใช้ฟังก์ชัน (Function) จากโปรแกรมได้เลย แต่ปัญหาที่พบคือต้องใช้เวลานานมากในการดำเนินงานการจำลองระบบ ดังนั้นอาจพิจารณาเขียนโปรแกรมใน Microsoft Visual Studio โดยใช้ภาษา C++ ในบางส่วนเพื่อเพิ่มความเร็วในการดำเนินงานการจำลองระบบ อีกทั้งยังสามารถนำไปใช้กับส่วนของชิ้นงาน Field-Programmable Gate Array (FPGA) ได้

รหัสภายในแบบรีดโซโลมอนซึ่งใช้ตัวถอดรหัสแบบ Algebraic สามารถพัฒนาให้มีประสิทธิภาพดีขึ้นได้ดังที่ได้มีผู้นำเสนอตัวถอดรหัสรีดโซโลมอนในหลายวิธีการ ดังนั้นสามารถนำวิธีการดังกล่าวมาประยุกต์ใช้กับระบบเพื่อเพิ่มประสิทธิภาพโดยรวมของระบบคอนคาทิเนตให้ดีขึ้นได้

โปรแกรมเวกเตอร์ซิมโบลซึ่งเขียนโปรแกรมจำลองใน Microsoft Visual Studio โดยใช้ภาษา C++ มีข้อจำกัดของขนาดตัวแปรที่ประกาศในโปรแกรม โดยขนาดตัวแปรที่ใหญ่ที่สุดคือ “unsigned long long int” ซึ่งมีขนาด 64 บิต ทำให้การขยายขนาดสัญลักษณ์ทำได้ยาก ดังนั้นจึงต้องปรับปรุงโปรแกรมให้สามารถปรับขนาดสัญลักษณ์ได้ง่ายขึ้น

เอกสารและสิ่งอ้างอิง

พิสิฐ วนิชชานันท์, ปิยะ โควินท์ทวีวัฒน์, อุศนา ตัณฑุลเวศม์, เกียรติศักดิ์ ศรีพิมานวัฒน์, กำพล วรดิษฐ์, ปรมินทร์ แสงวงษ์งาม, ดิสพล ฉ่ำเฉียวกุล และ สัญฉกร วุฒิสิทธิกุลกิจ. 2552. **ทฤษฎีรหัสช่องสัญญาณ**. พิมพ์ครั้งที่ 1. สถาบันวิจัยและพัฒนาอุตสาหกรรมโทรคมนาคม.

พงษ์พิสุทธิ์ นรดี. 2553. **ต้นแบบเครื่องถอดรหัสภายในสำหรับตัวถอดรหัสคอนโวลูชันภายนอก.** วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์.

รัชนนท์ อินที่สกุล. 2551. **การออกแบบและสร้างระบบเพื่อใช้สำหรับการเข้ารหัสและถอดรหัสสัญลักษณ์นอนไบนารีด้วยบอร์ด FPGA**. วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์.

อรรถภัทร์ สืบเนื่อง. 2552. **ชิ้นงานต้นแบบของเครื่องถอดรหัสด้วยวิธีเวกเตอร์ซิมโบลดีโคดดิงสำหรับรหัสแบบคอนโวลูชัน (3, 2, 2) ที่มีชุดข้อมูลขาเข้า 2 ตัวเลือก บนบอร์ดทดลอง FPGA.** วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์.

Andreadou, N. and F. N. Pavlidou. 2010. Mitigation of Impulsive Noise Effect on the PLC Channel With QC–LDPC Codes as the Outer Coding Scheme. **IEEE Transactions on Power Delivery**. 25: 1440-1449.

Berlekamp, E. R. 1974. Key papers in the development of coding theory. **IEEE Press**. New York.

Bose, R. C. and D. K. Ray-Chaudhuri. 1960. On a class of error correcting binary group codes. **Information and Control**. 3: 68-79.

Chuah, T. C. 2009. Onreed-solomon coding for data communications over power-line channels. **IEEE Transactions on Power Delivery**. 24: 614-620.

Ellias, P. 1955. Coding for Noisy Channels. **IRE Convention Record**. 3: 37-46.

Forney, G. D., Jr. 1966. **Concatenated Codes**. MIT Press. Cambridge. MA.

Hocquenghem, A. 1959. Codes corecteurs k'erreurs. **Chiffres**. 2: 147-156.

Hu, T. H. and S. Lin. 2003. An efficient hybrid decoding algorithm for reed-solomon codes based on bit reliability. **IEEE Transactions on Communication**. 51: 1073-1081.

Lin, S. and D. J. Costello Jr. 2004. **Error Control Coding**. 2nd edition ed. Pearson Education, Inc. Upper Saddle River, NJ.

Linnartz J. P. 1993. **Narrowband Land-Mobile Radio Networks**. ARTECH HOUSE, INC. 685 Canton Street Norwood, MA 02062.

Metzner, J. J. and E. J. Kapturowski. 1990. A general decoding technique applicable to replicated file disagreement location and concatenated code decoding. **IEEE Transactions on Information Theory**. 36: 911-917.

Nguyen, M. V., K. Ko, W. Lee and H. S. Lee. 2007. A new scheme to predict erasures for reed-solomon decoder in T-DMB receiver. **IEEE Transactions on Broadcasting**. 53: 530-538.

Reed, I. S. and G. Solomon. 1960. Polynomial codes over certain finite fields. **Society for Industrial and Applied Mathematics**. 8: 300-304.

Reeve, J. S. and K. Amarasinghe. 2004. A FPGA implementation of a parallel Viterbi decoder for block cyclic and convolution codes. pp. 2596-2599. **IEEE International Conference on Communications (ICC2004)**. 20-24 June 2004. Paris, France.

Shayegh, F. and M. R. Soleymani. 2011. Efficient soft decoding of reed-solomon codes based on sphere decoding. **IET Communications**. 5: 141-153.

Seshadri, N. and C. E. W. Sundberg. 1994. List Viterbi decoding algorithms with applications. **IEEE Transactions on Communication**. 234: 313-323.

Thonchai, J., V. Suktalordcheep and U. Tuntoolavest. 2013. Lab prototype of list-of-2 soft viterbi decoder for a BCH inner code in a generalized concatenated coding system. **The International Conference on Information and Communication Technology for Embeded System (ICICTES)**. 24-26 January 2013, Samutsongkhram, Thailand.

Tuntoolavest, U. 2009. **Vector Symbol Decoding for Wireless Fading Channels.** VDM publishing.

Tuntoolavest, U. and A. Seubnaung. 2007. Performance investigation of convolutional vector symbol decoding with larger than two choices and with incomplete second choices. **Kasetsart Journal (Natual Science)** 41: 364-370

Tuntoolavest, U. and J.J. Metzner. 2002. Vector symbol decoding with list symbol decisions and outer convolutional codes for reliable communications. **Integrated Computer-Aided Engineering Journal** 9: 101-116.

Tuntoolavest, U. and P. Noradee. 2010. Lab prototype of a list-of-2 viterbi decoder: a diversity inner decoder for the outer vector symbol decoder. pp. 973-977. **Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI)**. 19-21 May 2010.

Tuntoolavest, U., V. Suktalordcheep and C. Chaiwan. 2011. List-of-2 soft decision viterbi inner decoder for a generalized concatenated coding system. pp. 264-267. **Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI)**. 17-19 May 2011, Kohnkaen, Thailand.

Viterbi, A. J. 1967. Error bounds for bonvolutional codes and an asymptotically optimum decoding algorithm. **IEEE Transactions on Information Theory.** 13: 260-269.

Xia, H. and J. R. Cruz. 2007. Performance of reliability-based iterative soft-decision reed-solomon decoding on magnetic recording channels. **IEEE Transactions on magnetics**. 43: 3320-3323.

Wei, L., T. Aulin and H. Qi. 1997. On the effect of truncation length on the exact performance of a convolutional code. **IEEE Transactions on Information Theory**. 43: 1678-1681.

ภาคผนวก

ECTI-CON2011 KHON KAEN UNIVERSITY
Communication Systems
Modulation, Coding and Channel Analysis
Paper ID 1150

List-of-2 Soft Decision Viterbi Inner Decoder for a Generalized Concatenated Coding System

Usana Tuntoolavest*, Vasin Suktalordcheep and Chanothai Chaiwan
Department of Electrical Engineering, Faculty of Engineering
Kasetsart University, Bangkok, 10900, Thailand
Tel: +66-2-9428555 ext 1565, Fax: +66-2-942-8555 ext 1550
E-mail: fengunt@ku.ac.th, g5314501317@ku.ac.th and g5314501201@ku.ac.th

***Abstract*-This paper proposes the use of list-of-2 soft decision Viterbi Algorithm (LSOVA) as a suitable diversity block inner decoding technique for the new generalized concatenated coding system. In this system, the LSOVA is the inner decoder and Vector Symbol Decoding (VSD) is the outer decoder. This new coding system allows the inner code and the outer code to be any combination of a linear block code and a convolutional code. This is possible because the outer decoding technique, VSD, can decode any linear block or convolutional codes that use large nonbinary symbols. In addition, this paper combines the list decoding and the soft decision decoding of block codes together. Simulation results show that only soft decision provides significant benefit of the short list-of-2 for block decoding with list Viterbi algorithm (LVA). Hard decision provides almost no benefit with very short list. The (31, 26, 3) BCH code is the selected inner code since it is a well-known good code and the 26-bit outer symbol size is large enough for VSD.**

I. INTRODUCTION

Conventionally, the decoding of linear block codes is an algebraic hard decision decoding technique. However, the decoding of convolutional codes is commonly done with both the hard and the soft decision decoding. A widely used algorithm to decode convolutional code is Viterbi Algorithm (VA) [1], which is a maximum likelihood decoding technique. For hard decision decoding, the Hamming distance can be used as the metric for VA. For soft decision decoding, the Euclidean distance can be used as the metric for VA [2].

Since the soft decision decoding can provide better performance than the hard decision one, there were interests in soft decision decoding of block codes. This interests started with the paper by Forney in 1988 [3]. He showed that there was simple trellis structure for some block codes such as the Reed-Muller (RM) codes. By representing codes in Trellis form, block codes could be viewed as a finite-state machine. VA could also be applied as the decoding algorithm, and therefore the soft decision decoding for block codes was possible. Lin and Costello explained basic trellis for block codes in [2].

Since BCH codes [4], [5] are much more widely used than RM codes; it led naturally to researches about trellis for these codes. Reeve and Amarasinghe showed in their 2004 paper [6] a way to draw the state diagram for block cyclic codes using BCH codes as examples. They also presented the implementation of a parallel Viterbi decoder for block codes. There were also other approaches for soft decision decoding of specific block codes such as the sphere decoding proposed by Shayegh and Soleymani for Reed-Solomon (RS) codes in 2008 [7]. This was sub-optimal decoding, but it was designed to reduce the decoding complexity for long codes.

To achieve better decoding performance, another way is to employ diversity such as space diversity with several received antennas. In addition, list decoding can also be used to provide several possible decoded sequences. List decoding was first proposed by Elias [8] in 1957 and Wonzencraft [9] in 1958 independently. The list-of-L decoding failed only when the correct data sequence was not in the list of L possible decoded sequences. Since then, there were many papers on list decoding. The early ones were listed in Elias's 1991 paper [10]. Elias also proved in the same paper that with list decoding, a code could correct more number of errors. Seshadri and Sundberg [11] applied list decoding for convolutional codes in 1994. They proposed LVA that can output L possible decoded sequences as the inner decoding technique. The outer code is simply an error detection code. Therefore, they still maintained the idea that the list-of-L decoding failed only when the correct data sequence is not in the list.

Tuntoolavest and Metzner mentioned a way of using list decoding with an error correcting outer code in [12]. The suitable outer decoding technique for this purpose was Vector Symbol Decoding (VSD). VSD principle was first proposed by Metzner and Kapturowski in 1990 [13]. It is a general decoding technique that could be applied for any linear block or convolutional codes that used large nonbinary symbols. The actual algorithms for block codes and for convolutional codes do differ in some aspects. However, the main principles are the same. VSD had an interesting feature that it could easily select the correct codeword from a list of L possible decoded sequences. VSD with list was first explained in [14]. More details on VSD can be found in [15].

The list decoding with an error correcting outer code was extended into a new concatenated coding system called "LVA-VSD" in [16]. This LVA-VSD system employed convolutional codes as both the outer and an inner code and used LVA as the

Supported by Kasetsart University Research and Development Institute

diversity inner decoder and Vector Symbol Decoding (VSD) as the outer decoder. The "diversity" inner decoder refers to the fact that the inner decoder provides more than one choice of inputs (diversity) to the outer decoder.

In this paper, the LVA-VSD coding system is generalized to allow the inner code and the outer code to be any combination of a linear block code and a convolutional code. While in the previous work, the inner and the outer codes must be either both block codes or both convolutional codes. In addition, only soft decision LVA was considered for the convolutional case before. List decoding for block inner codes was not shown either. In this paper, both hard and soft decision LVA are investigated to see their effects when combined with list decoding for a selected good block cyclic code. Specifically, the (31, 26, 3) single-error-correcting BCH code is selected.

Section II describes the block diagram of the proposed concatenated coding system. Section III covers the method. Section IV shows the simulation results that list decoding provides significant benefit for the soft decision case, but almost no benefit for the hard decision case, especially for a very short list. The reason for this is explained in section V. Finally, section VI concludes the ideas and results of the paper.

II. Generalized LVA-VSD Concatenated Coding System

A. Block Diagram of the System

The generalized LVA-VSD concatenated coding system is shown in fig. 1. This system allows the inner code to be either a binary block code or a binary convolutional code. The outer code can also be either a nonbinary block code or a nonbinary convolutional code. The inner encoder is simply a binary block or convolutional encoder. For block cyclic outer codes, the encoder can be implemented using shift register circuits similar to the encoding of RS codes [2]. For convolutional outer code, the encoder can be implemented as shown in [18]. The decoding part is more complicated. The outer decoder is VSD for block [14] or VSD for convolutional [12], [15] as suitable. The inner decoder is a list-of-2 VA. This inner decoder provides two decoded outputs in the order of their likelihood function. The use of list-of-2 VA for convolutional codes was shown in [15]. In this paper, we will focus on the use of list-of-2 VA for block codes as well as investigate the effect of using hard and soft decision with list decoding in block codes.

B. System Setup

For the simulation, the inner code was selected to be the (31, 26, 3) single-error-correcting BCH code. The reason is that it is a well known good block cyclic code and each decoded sequence consists of 26 bits. This is suitable for VSD because VSD has the assumption that the error symbols are linearly independent. This assumption is valid when the symbol size is large enough and the typical size for VSD is 24-bit symbol or larger [17]. Fig. 2 shows the system setup for the comparison of the algebraic hard decision decoding using Matlab function with the hard decision VA in a binary symmetric channel (BSC). Fig. 3 shows the system setup for the comparison between the hard and soft decision list-of-2 VA in an Additive

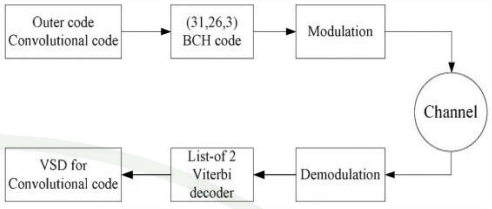


Figure 1. Generalized LVA-VSD concatenated coding system

Figure 2. System setup for hard decision decoding using algebraic decoding in comparison with VA

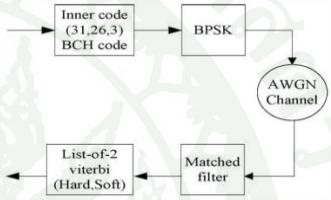


Figure 3. System setup for hard and soft decision list-of-2 VA

White Gaussian Noise (AWGN) channel. The modulation technique was binary phase shift keying (BPSK) and the demodulation was done by a matched filter.

III. Method

To apply list-of-2 VA to a block code, the code should be viewed as a finite-state-machine. Table 1 shows the state table of the (31, 26, 3) BCH code. This table was derived based on the method to generate state transition diagram of BCH codes proposed by Reeve and Amarasinghe [6]. From the state table, the state diagram and the trellis diagram could be obtained in a straightforward manner as shown in fig. 4. Then, we applied the LVA that was written in C++ based on the method by Seshadri and Sundberg [11]. To verify that the C++ program worked properly, the result of the hard decision LVA in C++ is compared to the result of the algebraic hard decision decoding using Matlab function as setup in fig 2. After this was verified, both the hard and the soft decision decoding were done for (31, 26, 3) BCH code using the C++ program as shown in fig 3. The metric for the hard decision decoding was the Hamming distance. The metric for the soft decision decoding was the correlation metric using the matched filter as the optimum receiver. Note that the simulation was done for the inner decoder level only. The results can then be analyzed for the overall system using the property of VSD.

To save space, the trellis for this code is not shown. However, it can be derived directly from the state diagram or the state table. Other block codes that can be represented in trellis form can also be decoded using the described method.

TABLE I
STATE TABLE FOR (31,26,3) BCH CODE

Previous state	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
next state	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
previous state	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
next state	5	7	1	3	13	15	9	11	21	23	17	19	29	31	25	27

input = 0

previous state	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
next state	5	7	1	3	13	15	9	11	21	23	17	19	29	31	25	27
previous state	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
next state	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30

input = 1

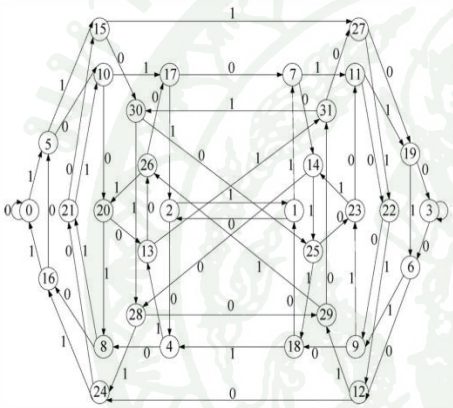


Figure 4. The state transition diagram for the (31, 26, 3) BCH code

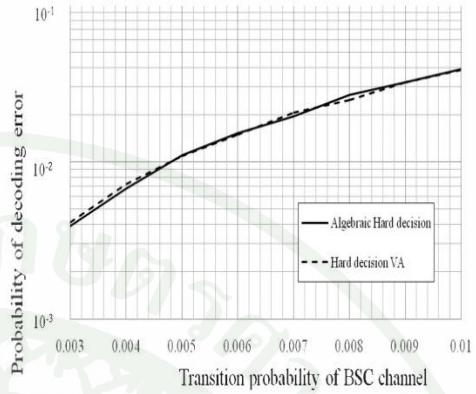


Figure 5. Performance of Algebraic hard decision in comparison with hard decision VA for (31,26,3) BCH code in BSC channel

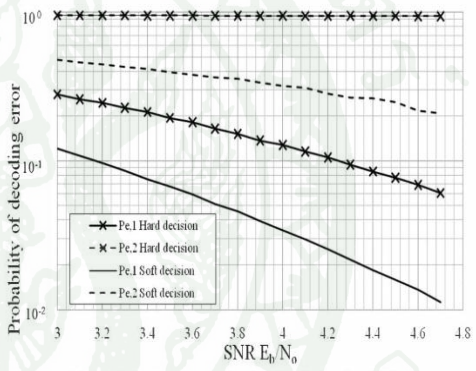


Figure 6. Performance of hard and soft decision LVA for (31, 26, 3) BCH code in Gaussian channel

IV. RESULTS

Fig. 5 shows that the decoding failure probability of a algebraic hard decision BCH decoding using Matlab function is almost the same as the hard decision BCH decoding using the list-of-2 Viterbi algorithm written with C++. This shows that both decoding methods provide the same result for hard decision. It also verifies that the list-of-2 VA works properly.

Fig. 6 shows the performance of list-of-2 VA with both hard and soft decision of the (31, 26, 3) BCH code in Additive White Gaussian Noise (AWGN) channel. The performance of the soft decision decoding of BCH code is better than the hard decision decoding of BCH code for both p_{e1} and p_{e2}. The parameter p_{e1} is the probability that the most likely decoded sequence is wrong. The parameter p_{e2} is the probability that the second most likely decoded sequence is wrong given that the most likely one was wrong. The better p_{e1} is as expected since it is well known that soft decision decoding uses the reliability information of the channel and therefore can usually provide better decoded sequence than the hard decision decoding that does not use this information.

The much better p_{e2} is significant because it shows that the list decoding provides great benefit for soft decision. Comparing the p_{e2} for both decoding, the p_{e2} for soft decision is in the range of approximately 21-48% for the given SNR range. This means that when the most likely decoded sequence is wrong, there is a very high probability (52-79%) that the correct codeword is the second decoded sequence. Since the two inner decoded sequences are the inputs to VSD, this means that VSD will have a much higher probability of receiving at least one correct input sequence with the use of list. This will help improve the performance of VSD and the overall performance of the coding system.

For hard decision the, p_{e2} is very high (approximately 95%). This means that the second decoded sequence is almost always wrong when the most likely decoded sequence was wrong.

V. DISCUSSIONS

Simulation results show that for hard decision, the algebraic decoding and VA provide the same result for the most likely decoded sequence. Therefore, if list decoding is not used, the

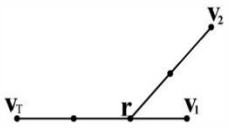


Figure 7. Positions of various vectors in the decision sphere for an example that both $\mathbf{v}_1$ and $\mathbf{v}_2$ are wrong

Figure 8. Positions of various vectors in the decision sphere for an example that $\mathbf{v}_1$ is wrong and $\mathbf{v}_2$ is correct

algebraic decoding with less complexity is more attractive. With list decoding, the simulation results show that the second decoded sequence provides great benefits in the soft decision case. However, it provides almost no benefit in the hard decision case. Therefore, the list should be used with soft decision only for BCH codes.

The reason that the second choice is not very useful for hard decision can be explained with the decision sphere concept. Since the minimum distance (d_{min}) for this single-error-correcting (31, 26, 3) is 3, some codewords differ in only 3 bit positions. When there is more than one error bit, the decoder will be wrong.

Fig. 7 shows the case that there are two error bits in the received sequence. The received sequence $\mathbf{r}$ can be viewed as a vector in the 31-dimension sphere. This $\mathbf{r}$ differs two bits from the transmitted codeword $\mathbf{v}_T$. With hard decision, the Hamming distance is used and the distance value is an integer number. The decoder corrects the received sequence $\mathbf{r}$ by changing it into a codeword that differs only one bit from $\mathbf{r}$. This codeword is the most likely decoded codeword $\mathbf{v}_1$. For list-of-2 decoding, the decoder also attempts to find another codeword that is a little less likely to be a correct codeword. In fig. 7, it chooses $\mathbf{v}_2$ as the second most likely decoded codeword since $\mathbf{v}_2$ differs only two bits from $\mathbf{r}$. The problem is that for hard decision, there are many codewords that differ two bits from $\mathbf{r}$. Therefore, the decoder will randomly pick one as the second choice. The chance that this selected one will be correct is thus very small.

Fig. 8 shows the case that the decoder picks the correct one for the second choice when there are two error bits in the received sequence. In this case, $\mathbf{v}_2$ is exactly the same as $\mathbf{v}_T$. Therefore, the second choice is correct. For the soft decision, the distance can be a Euclidean distance, which is not a real number. Thus, it is not likely that there will be candidates with equal metric. The decoder will then pick the one with the second best metric to be the second choice. This results in a much better selection of the second decoded sequence.

VI. Conclusions

This paper proposes a generalized concatenated coding system. This system allows any combination of block and convolution codes as the inner and the outer code. The outer decoder is VSD and the inner decoder is list-of-2 soft decision VA. The (31, 26, 3) BCH code is selected as the inner code due to its good cyclic structure and the appropriate symbol size for the outer decoder. With hard decision and no list decoding, there is no advantage to use the more complex decoding technique and the algebraic one is preferable. With list decoding, the soft decision is necessary to ensure good performance of the second decoded sequence. Based on the feature of VSD, the high probability that the second choice is correct when the first choice is wrong will significantly improve the decoding failure probability of the overall concatenated coding system. It will also reduce the decoding time of the outer decoder.

References

[1] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, vol. IT-13, pp. 260-269, Apr 1967.

[2] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd edition, Pearson Education, Upper Saddle River, NJ, 2004.

[3] G.D. Forney Jr., "Coset Codes II: Binary Lattices and Related Codes," *IEEE Trans. Inf. Theory*, vol. IT-34, pp. 1152-1187, 1988.

[4] A. Hocquenghem, "Codes corecteurs d'erreurs," *Chiffres*, vol. 2, pp. 147-156, 1959.

[5] R.C. Bose and D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inform. Control*, vol. 3, pp. 68-79, March 1960.

[6] J.S. Reeve and K. Amarasinghe, "A FPGA Implementation of a parallel Viterbi decoder for block cyclic and convolution codes," *Proceedings of IEEE Inter. Conf. on Comm. (ICC2004)*, vol.5, pp. 2596 – 2599, June 2004

[7] F. Shayehg and M.R. Soleymani, "Soft decision decoding of Reed-Solomon codes using sphere decoding," *Proceedings of IEEE Inter. Conf. on Comm. (ICC2008)*, pp. 4489 - 4495, May 2008, Beijing, China

[8] P. Elias, "List decoding for noisy channels," *MIT Res. Lab. Electron.*, Cambridge, MA, Sep. 1957.

[9] J. M. Wozencraft, "List decoding," *MIT Res. Lab. Electron.*, Cambridge, MA, Jan., 1958.

[10] P. Elias, "Error-correcting codes for list decoding," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 5-12, Jan 1991.

[11] N. Seshadri and C. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. on Comm.*, vol.42, pp.313-323, Feb/Mar/Apr 1994.

[12] U. Tuntoolavest and J.J. Metzner, "Vector symbol decoding with list symbol decisions and outer convolutional codes for reliable communications," *Integrated Computer-Aided Engineering Journal*, vol. 9, no. 2, 2002, p. 101-116, IOS Press, ISBN 1069-2509.

[13] J.J. Metzner and E.J. Kapturowski, "A general decoding technique applicable to replicated file disagreement location and concatenated code decoding," *IEEE Trans. Inf. Theory*, vol.36, pp.911-917, July 1990.

[14] J.J. Metzner, "Vector Symbol Decoding with list inner symbol decision," *IEEE Trans. on Comm.*, vol.51, no. 3, pp.371-380, Mar 2003.

[15] U. Tuntoolavest, *Vector Symbol Decoding for Wireless Fading Channels*, VDM publishing, 184 pages, 2009

[16] U. Tuntoolavest and P. Noradee, "Lab Prototype of a List-of-2 Viterbi Decoder: A Diversity Inner Decoder for the Outer Vector Symbol Decoder," *Proceedings of ECTI-CON 2010*, pp. 973-977, May 19-21, 2010, Chiang Mai, Thailand

[17] U. Tuntooolavest, "A simple method to improve the performance of convolutional vector symbol decoding with small symbol size." *Proceeding of IEEE TENCON 2004*, Vol. B, pp.676-679, November, 21-24, 2004, Chiang Mai, Thailand.

[18] U. Tuntoolavest and R. Intharasakul, "Nonbinary Convolutional Encoder for Vector Symbol Decoding on FPGA board." *2006 10th International Conference on Communications Technology Proceedings (ICCT2006)*, vol. 1, pp.716-719, November 27-30, 2006, Guilin, China.

Lab Prototype and Performance Investigation of List-of-2 Soft Viterbi Decoder for a BCH Inner Code

Jatupon Thonchai, Vasin Suktalordcheep and Usana Tuntoolavest*
Department of Electrical Engineering
Kasetsart University, Bangkok, Thailand
Tel: +66-2-7970999 ext 1565, Fax: +66-2-7970999 ext 1550
E-mail: g5314501171@ku.ac.th, g5314501317@ku.ac.th, fengunt@ku.ac.th

***Abstract*—**List-of-2 soft Viterbi algorithm (VA) decoder provides a list of two possible decoded sequences in the order of their likelihood. It can decode convolutional codes and any block codes represented in a trellis diagram. It is an important component of a generalized concatenated coding system that allows the inner and outer codes to be any combination of convolutional and block codes with some modifications. This paper implemented the list-of-2 soft VA and the concatenated encoder in a Nanobaord3000 with Xilinx Spartan 3AN FPGA chip. The results from the C++ simulation and the lab prototype were exactly the same in various fading channel conditions. Thus, the lab prototype worked properly as designed. The results showed that the soft VA was better than the hard VA as expected. The presence of Doppler increased the decoding error probability. The Rician fading channel provided noticeable improvement compared to the Rayleigh channel.

***Keywords*—list-of-2 soft Viterbi, VSD, prototype, inner decoder, generalized concatenated codes, fading channels.**

I. Introduction

Serial concatenated codes were first proposed by Forney in 1966 [1]. It was originally called "concatenated codes". Since turbo codes [2] were proposed and used widely, the original one was sometimes called serial concatenated codes to distinguish them from turbo codes that were sometimes referred to as parallel concatenated codes [3]. The outer code of a serial concatenated code was usually a Reed-Solomon (RS) code [4] as described in CCSDS standards [5]. RS codes are block codes with nonbinary symbols from GF(q) [6]. The inner code is usually a binary block or a binary convolutional code. The inner decoder depends on the inner code selected. The outer decoder is a RS code decoder.

In 2011, the concept of a generalized concatenated decoder for serial concatenated codes was presented [7]. The decoding principle proposed could be applied to any combinations of block and convolutional codes for inner and outer codes. In other words, the inner code can be either of the two types of codes and the outer code can also be either of the two types. This was possible because the principle of Vector Symbol Decoding (VSD), the outer decoding algorithm, can be applied to both types of codes with some modifications in the implementation [8]. Moreover, list-of-2 Viterbi decoding, the inner decoding algorithm, can be used to decode block codes as well as convolutional codes by representing the block codes in the trellis diagram [6]. The list Viterbi algorithm (LVA) was presented by Seshadri and Sundberg in [9]. LVA provided an ordered list of possible decoded sequences for each received sequences based on the likelihood. In 2011, a soft decoding for binary cyclic codes was proposed, but it was a "light soft version of permutation decoding" [10]. It was not a Viterbi decoding and no list decoding was used. The soft Viterbi decoding combined with list decoding was described and analyzed in 2011 [11]. Metzner presented the algorithm of VSD with list inner decisions in [12]. VSD with list for various structures of convolutional codes was analyzed in [13]. LVA was mentioned as a way to provide list inner decisions for VSD in [14]. LVA with only two choices (list-of-2 VA) was selected as the inner decoder for VSD in [15].

In [11], the concept of this coding system was presented with the performance comparison of inner decoder only. Specially, the comparison was done for the algebraic decoding, hard Viterbi and list-of-2 soft VA decoding of a BCH code in the AWGN (Additive white Gaussian noise) channel only. It was concluded from the results that the algebraic decoding and the hard decision VA provided basically the same decoding error probability. However, the soft list-of-2 VA provided much lower decoding error probability. In [16], the lab prototype of list-of-2 VA was implemented for a convolutional inner code, while the one in this paper is for a block inner code. The selected board was also different. The prototype in [16] was programmed with VHDL (VHSIC hardware description language). The current prototype was programmed with C, which makes it much easier to modify.

In this paper, we implemented the hard decision and the list-of-2 soft decision VA in an FPGA (Field Programmable Gate Array) board, which was the Nanoboard 3000. We also implemented other components of the generalized concatenated coding system, which were a convolutional outer encoder and the BCH inner encoder in a board. We demonstrated an example of a convolutional outer code instead of a block code because standard serial concatenated codes usually used RS codes, which were nonbinary block codes as the outer codes. Therefore, if a convolutional outer code was shown, it would

Supported by Kasetsart University Research and Development Institute

emphasize the generality of this coding system. For the block outer code case such as RS codes, the outer encoder will be the nonbinary block encoder such as the shift register circuits and the decoder will be VSD for block codes. We also extend the performance investigation of the inner decoder from AWGN in [11] to fading channels in this paper. These models were more realistic for its wireless applications. The wireless fading channels were Rayleigh and Rician fading channels with AWGN. The Doppler effect was also investigated for both fading models. For the performance investigation, all-zero code words was assumed for simplicity since this assumption is valid with no loss of generality.

In the near future, the practical outer decoder of this concatenated coding system will be added to complete the system. This outer decoder will use the Vector Symbol Decoder (VSD) with list decoding [8,12]. In addition, another function to map the decoded sequence to its corresponding data sequence needs to be included. This conversion is not trivial for a nonsystematic convolutional code. Since the nonsystematic convolutional code provided better performance than the systematic one, it was preferable and the outer decoder would need to include this mapping.

II. SYTEM DESIGN

A. *Block Diagram of the System*

The presented concatenated code consists of an inner block code and the outer convolutional code as shown in fig. 1. The modulation, demodulation and the channel were modeled with Matlab. The encoders and the decoders had been previously simulated with C++. In this paper, the inner and outer encoders as well as the hard Viterbi and the list-of-2 soft Viterbi decoders were implemented in a nanoboard3000 with an FPGA chip.

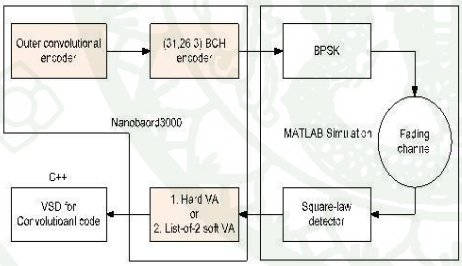


Figure 1. The concatenated coding system

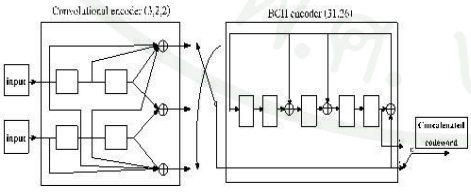


Figure 2. The structure of the selected concatenated encoder

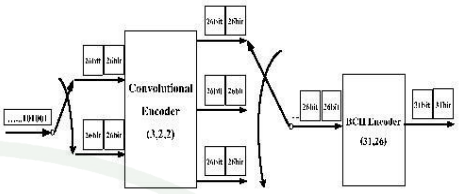


Figure 3. The inputs and outputs of the inner and outer encoders

In the next phase, the outer decoder (VSD) will be implemented in the same board. The standard decoder for a BCH code is the algebraic decoder. In addition, Viterbi can be used to decode block codes that can be represented by a trellis diagram [6]. Therefore, both hard decision and list-of-2 soft decision Viterbi were also used as the inner decoder. The fading channels were Rayleigh and Rician channels with and without Doppler effect.

The inner code was a single error correcting binary BCH code of length 31. The outer code was a (3,2,2) nonbinary convolutional code. The detailed structure of the selected inner and outer encoders is illustrated in fig. 2. Each memory block of the (3,2,2) nonbinary code contains 26 bits, but that of the binary BCH code contains 1 bit. Fig. 3 shows the input, output and the interface inside the concatenated encoders. It can be seen that each set of two 26-bit symbols were encoded by the outer encoder into three 26-bit symbols. These three 26-bit symbols were then multiplexed and input to the inner encoder. Each 26-bit sequence was encoded into a 31-bit inner code word. The encoded sequence was modulated by a binary phase shift keying (BPSK) scheme as described in fig. 1 and transmitted through a fading channel.

B. *Hardware Aspect*

The hardware was designed with Altium Designer. The FPGA board used was the Nanoboard 3000. The FPGA chip was the Xilinx Spartan 3AN device (XC3S1400AN-4FG676C). This chip consisted of 140,000 gates. It contained the TSK-3000A 32-bit RISC (Reduced Instruction Set Computer) processor as shown in fig. 4. The open bus was

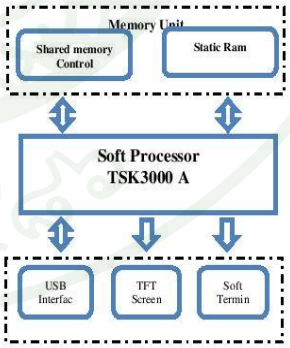


Figure 4. System design in hardware

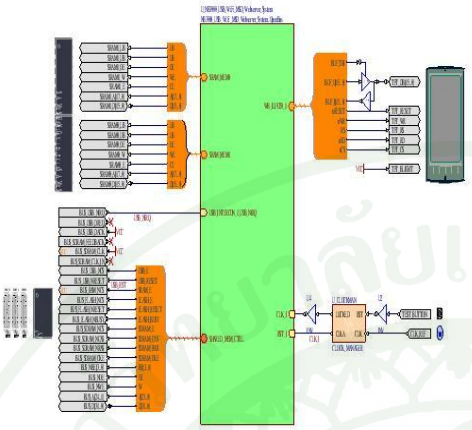


Figure 5. The schematic diagram of the system

designed with "soft" processor. The soft processor was chosen because it was flexible, low cost and faster to implement. This is because it behaved as a CPU (central processing unit). We could write the algorithm in C and downloaded into this soft processor. The authors planned to implement other inner coding system and compare their performance in the next phase. The design in fig. 4 shows that the soft processor was connected to a 1-megabit independent SRAM (Static Random-Access Memory) and a 1-megabit common bus memory.

In addition, the input/output ports of the processor were connected to a USB port, a soft terminal and a colour TFT-LCD (Thin film transistor-liquid crystal display) panel. The soft terminal and the TFT-LCD panel were for the input output display. All selected components were linked together in the schematic diagram illustrated in fig. 5 to allow the flow of signals between them and the processor. The left hand side of the processor was connected with the independent memory and the common bus memory. The right hand side of the processor was connected to the TFT-LCD panel, the clock generator and the test button.

III. METHOD

First, all parts of the coding system were modeled by either Matlab or Visual C++. These helped with the hardware test. Then the hardware was designed as described in section II. The C++ program was modified and simplified into a C program with only basic functions to meet the requirement of the processor in the board. This new program was input into the Nanoboard3000.

A. Simulations

The simulations were done using the combination of Matlab and C++ programming. Specifically, the fading channels were modeled with Matlab. The fading channels in consideration were the Rayleigh and Rician fading channel with or without the Doppler effect. In addition, the BCH inner encoder and the algebraic decoder were modeled with Matlab since these functions were readily available. The C++ programming was used for the outer encoder, the list Viterbi inner decoder and the outer VSD decoder since they were not standard encoder and decoders. The modulation scheme was the BPSK. The demodulation was done with a square-law detector.

The Doppler shift (f_d) can be computed from [16]

$$f_d = f_c \frac{v \cdot \cos(\gamma)}{c_0} \qquad (1)$$

where f_c is the carrier frequency

v is the velocity that the receiver moves away from the transmitter

c_0 is the light speed and equal to $3\text{x}10^8$ m/s

and γ is the angle between the transmitter and the receiver

The maximum Doppler shift occurs when γ is 0. In the simulation, we assumed maximum Doppler shift. The carrier frequency used was 2.1 GHz, which is the standard frequency for the third generation mobile communication system (3G). The velocities considered were 80 km/hr and 120 km/hr. From the computation, the 80 km/hr case caused the Doppler shift of 155.54 Hz. The 120 km/hr case caused the Doppler shift of 233.31 Hz.

B. FPGA Implementations

Fig. 6 shows the setup of the nanoboard in operation. The board is connected to the computer via a USB (Universal Serial Bus) port. For the experiment, the input file was in a flash drive USB that was connected to the board. The flash drive was used to test that the inner decoder worked properly as designed. To test the function of the board, we separated the test of the encoding and the decoding part. For the encoding part, the function was tested by several pilot data sequences. The encoded sequences were compared to the correct encoded sequences from the coding theory.

For the inner decoder part, the same received sequences obtained at the output of the square-law detector were input into the inner decoder with C++ program and the inner decoder inside the board. The results were then compared for hard decision and list-of-2 soft VA in various fading channel conditions.

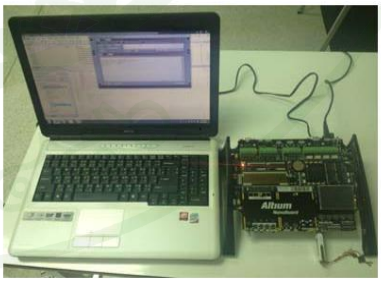


Figure 6. Actual hardware setup during tests.

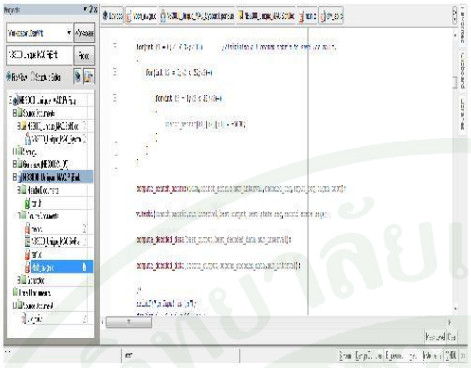


Figure 7. The code explorer window in Altium designer showing the Viterbi functions

In fig. 7, the inner decoder functions are shown in the Altium designer code explorer window. The detailed programming in each function was not demonstrated due to the lack of space. The concept of the list-of-2 soft VA had been explained in detail in [11].

IV. RESULTS

The encoding functions of the system on the board were tested by inputting several pilot data sequences into the board and check whether the encoded sequences were properly encoded. Fig. 8 shows an example of the pilot data sequence and the encoded sequence on the soft terminal of the board. The encoding functions were straightforward and the encoded sequence matched with the expected results from the theory.

To ensure that the Viterbi program written in C++ worked correctly, its decoding error probability was compared with the algebraic decoding function in Matlab. The modulation scheme was BPSK and the demodulation was readily available with Matlab function for the hard decision case. The channel was modeled as a Rayleigh fading channel with AWGN. The result in fig. 9 confirms that they are approximately the same. However, the decoding error probability was high since the inner code can correct only 1 error in a 31-bit received sequence. Since the decoding error probability is in the small range of 0.1-1 in fig. 9, the y-axis was shown in linear scale.

Figure 8. An example of the concatenated encoding result shown in the TFT-LCD display on the Nanoboard 3000 board.

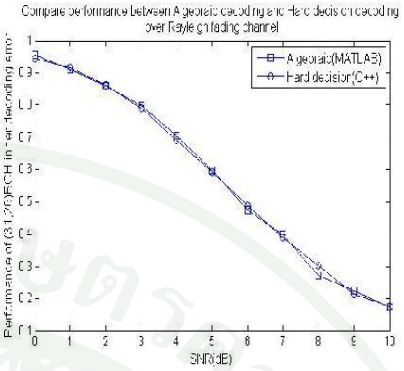


Figure 9. Decoding error probability of the algebraic and the hard VA decoders for the (31,26,3) BCH code in a Rayleigh channel

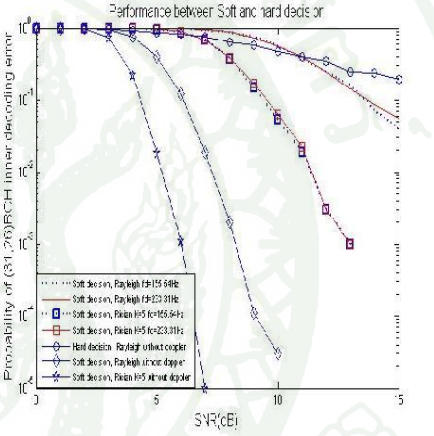


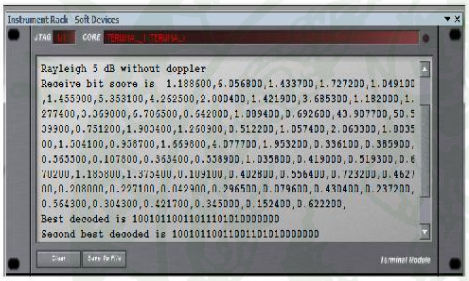
Figure 10. Decoding error probability of the soft and hard VA decoder for the (31, 26, 3) BCH code in fading channels

Recall that list-of-2 VA provides two possible decoded sequences, which can be called "the first choice" and "the second choice". Fig. 10 shows the comparison between the hard and soft decision VA. Only the decoding error probability of the first choice was shown since they were compared in several channel conditions. It is clear that the soft decision VA provided much better performance than the hard decision VA. Consequently, only the soft VA was investigated further in details. Fig. 10 shows the performance of soft VA in Rayleigh and Rician channels with and without Doppler effect. It is seen that the fading channels with no Doppler effect provided significantly better performance than the ones with Doppler effect.

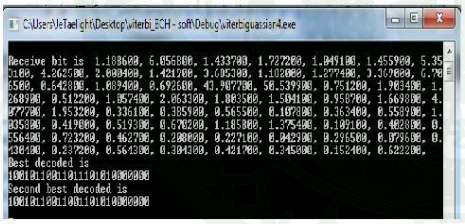
The Rician channel resulted in much lower decoding error than the Rayleigh channel both when there was a Doppler effect and when there was no Doppler effect. This is as expected since the Rayleigh channel is a special case of Rician channel when the Rician factor k = 0, which means that there is no line-of-sight path. For the channel with the Doppler effect and the carrier frequency of 2.1 GHz, the f_d of 155.54 Hz is for

the velocity of 80 km/hr case and the f_d of 233.31 Hz is for the velocity of 120 km/hr case. The performance for both Doppler frequency cases was approximately the same for both Rayleigh and Rician channels.

The results from the lab prototype were exactly the same as the results from the C++ simulation. Thus, fig. 10 represents the results from both hardware and software simulations since their graphs completely overlapped. To demonstrate the results from the lab prototype, fig. 11 and 12 show the soft terminal display in comparison with C++ output for several cases. In details, fig. 11 shows an example of a decoded sequence in two displays for the Rayleigh fading channel with AWGN that had SNR of 5 dB and no Doppler effect. The first one is the soft terminal display of the hardware. The second one is the C++ output display. It is clear that the results from the hardware and the software in fig. 11 were exactly the same. Notice that there were two possible decoded sequences in the display because the decoder is a list-of-2 decoder. Both decoded sequences will be the input to the outer decoder.



a) Soft terminal display of the prototype

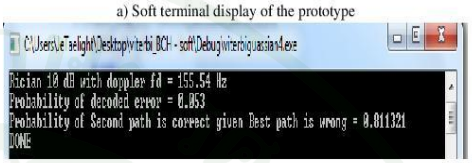


b) The C++ output display

Figure 11. Example of a decoding result in Rayleigh fading channel with SNR = 5 dB and no Doppler.

Fig. 12 shows a decoding error probability in the Rician fading channel with SNR = 10 dB and the Doppler frequency of f_d = 155.54 Hz. The number of trials used was 1,000 trials. Two probabilities were shown in the display. The probability that the first choice is wrong is considered the decoding error probability of the decoder. This is because the first choice is the main decoded sequence. If the inner decoder is used by itself, the decoding will fail when the first choice is wrong. However, when VSD with list inner symbol decision is used as the outer decoder, it can replace many wrong first choices with the correct second choices. This second choice can improve the performance of VSD. Therefore, the probability that the second choice is correct given that the first choice is wrong is of interest.

a) Soft terminal display of the prototype



b) The C++ output display

Figure 12. Example of a decoding error probability in Rician fading channel with SNR = 10 dB and f_d = 155.54 Hz.

Table 1. Resource utilization of the prototype

Logic Utilization	Used	Total	% Usage
Number of Slice Flip Flops	5,951	22,528	26%
Number of 4 input LUTs	9,743	22,528	43%
Logic Distribution			
Number of occupied Slices:	7,072	11,264	62%
Total Number of 4 input LUTs	10,273	22,528	45%
Number used as logic	9,473		
Number used as a route-thru	530		
Number used as 16x1 RAMs	14		
Number used for Dual Port RAMs	256		
Number of bonded IOBs	194	502	62
Number of BUFGMUXs	4	24	16
Number of DCMs	1	8	12
Number of MULT18X18SIOs	10	32	31
Number of RAMB16BWEs	20	32	62

The resource utilization is shown in table 1. This FPGA board operated at 50 MHz. Since there are a lot of logic units left in this board, we plan to implement the outer decoder on the same board.

V. Discussions

The FPGA implementation is a natural step for testing proposed system already tested in software simulation. The main objective is to show that the generalized concatenated coding system can be implemented. In this paper, only the encoders and the inner decoder were implemented. In the near future, other inner coding systems and the outer decoder will also be implemented.

The decoding error probabilities of the hardware and the software for this inner decoder were exactly the same when the same received sequences were used as the input. This confirmed that the lab prototype worked properly as designed. It also emphasized that the list Viterbi is not too complex to be implemented in hardware. The results also showed that the list-of-2 soft VA provided significantly better performance than the hard VA. Therefore, only list-of-2 soft VA will be considered in a generalized concatenated coding system.

The performance investigation was done for various channel conditions. It is clear that the Rician channels provided better performance than the Rayleigh channels. In addition, the Doppler effect increased the probability of decoding errors in both the Rician and Rayleigh channels. For this particular code, different Doppler frequencies resulted in almost the same decoding error probability for both Rayleigh and Rician fading channels. This was because each inner code word had a length of only 31 bits, which is relatively short. It was also a single-error-correcting BCH code, which has low correction capability, even when the soft decision decoding was employed. This inner code word was not powerful enough to correct the burst errors due to the fast fading caused by the Doppler effect. Consequently, both Doppler frequencies caused almost no difference in the inner decoder performance even though the higher Doppler frequency caused shorter fading periods and increased the number of crossover between the fade and non-fade. More powerful and longer inner codes may improve the performance of the inner decoder for fading channels with Doppler effect.

VI. Conclusions

This paper has presented the implementation of a concatenated encoding system and the list-of-2 soft VA in a Nanobaord3000 with the Xilinx Spartan 3AN FPGA chip. (XC3S1400AN-4FG676C). The performance investigation for the list-of-2 soft VA for the inner BCH code was done for fading channels with and without Doppler effects.

This list-of-2 soft VA is an important component of a generalized concatenated coding system that allows the inner and outer codes to be any combination of convolutional code(s) and block code(s). The actual algorithms and implementations depended on the selected codes, but the main principle and algorithm are the same. The test results showed that the prototype worked properly as designed. The soft processor was selected instead of FPGA implementation with hardware language like VHDL (VHSIC hardware description language) because we planned to test different inner coding systems to see which one will be most suitable for the proposed generalized concatenated coding system. This presented prototype will be used in the complete system in a near future.

Acknowledgment

The authors would like to thank Mr. Chanothai Chaiwan for the help with the C++ program of the list-of-2 VA for the selected BCH code.

References

[1] G.D. Forney, Jr., Concatenated Codes, MIT Press, Cambridge Mass., 1966.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in Proc. ICC'93, pp. 1064–1070, May 1993, Geneva, Switzerland.

[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," IEEE Trans on Inf. Theory, vol. 42, issue: 2, pp. 409 – 428, 1996.

[4] I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields," Journal of Society for Industrial and Applied Mathematics, vol. 8, pp. 300-304, 1960

[5] CCSDS 130.1-G-1, TM synchronization and channel coding-summary of concept and rationales, green book, 89 pages, June 2006.

[6] Lin and D. J. Costello Jr., Error Control Coding, 2nd edition, Pearson Education, Upper Saddle River, NJ, 2004.

[7] U. Tuntoolavest and J. Thonchai, "VHDL design of a convolutional concatenated encoding system," in Proc. of ICICTES 2011, pp.140-144, Jan 27-29, 2011, Pattaya, Chonburi, Thailand.

[8] U. Tuntoolavest, Vector Symbol Decoding for Wireless Fading Channels, VDM publishing, 184 pages, 2009.

[9] N. Seshadri and C. W. Sundberg, "List Viterbi decoding algorithms with applications," IEEE Trans. on Comm., vol. 42, pp. 313-323, Feb/Mar/Apr 1994.

[10] I. Chana, H. Allouch and M. Belkasmi, "An efficient new soft-decision decoding algorithm for binary cyclic codes," in Proc. of ICMCS, pp. 823-828, April 2011, Ouarzazate, Morocco.

[11] U. Tuntoolavest, V. Suktalordcheep and C. Chaiwan, "List-of-2 soft decision Viterbi inner decoder for a generalized concatenated coding system," in Proc. of ECTI-CON 2011, pp. 264-267, May 17-19, 2011, Khon Kaen, Thailand.

[12] J.J. Metzner, "Vector Symbol Decoding with list inner symbol decision," IEEE Trans. on Comm., vol. 51, no. 3, pp. 371-380, Mar 2003.

[13] U. Tuntoolavest and C. Chaiwan, "On adjusting vector symbol decoding for many different nonbinary convolutional codes," Kasetsart Journal (Natural Science), vol. 46, no. 2, pp. 305-317, March-April 2012.

[14] U. Tuntoolavest and J.J. Metzner, "Vector symbol decoding with list symbol decisions and outer convolutional codes for reliable communications," Integr comput-aided engineer Journal, vol. 9, no. 2, 2002, pp. 101-116, IOS Press.

[15] U. Tuntoolavest and A. Seubnaung, "Performance investigation of convolutional vector symbol decoding with larger than two choices and with incomplete second choices," Kasetsart Journal (Natural Science), supplement issue, vol. 41, no 5, p 364-370, January-December 2007.

[16] U. Tuntoolavest and P. Noradee, "Lab prototype of a list-of-2 Viterbi decoder: a diversity inner decoder for the outer vector symbol decoder," in Proc. of ECTI-CON 2010, pp. 973-977, May 19-21, 2010, Chiang Mai, Thailand

[17] A.F. Molisch, Wireless communications, John Wiley & sons, 622 pages, 2005.

ประวัติการศึกษาและการทำงาน

ชื่อ-นามสกุล นายวศิน สุขตลอดชีพ

วัน เดือน ปี ที่เกิด วันที่ 8 พฤษภาคม 2531

สถานที่เกิด จังหวัดกรุงเทพมหานคร

ประวัติการศึกษา วศ.บ.(วิศวกรรมไฟฟ้า) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ตำแหน่งหน้าที่การงานปัจจุบัน

สถานที่ทำงานปัจจุบัน

ผลงานดีเด่นและรางวัลทางวิชาการ

ทุนการศึกษาที่ได้รับ