

บทที่ 3

รหัสแก้ไขความผิดพลาดของข้อมูล

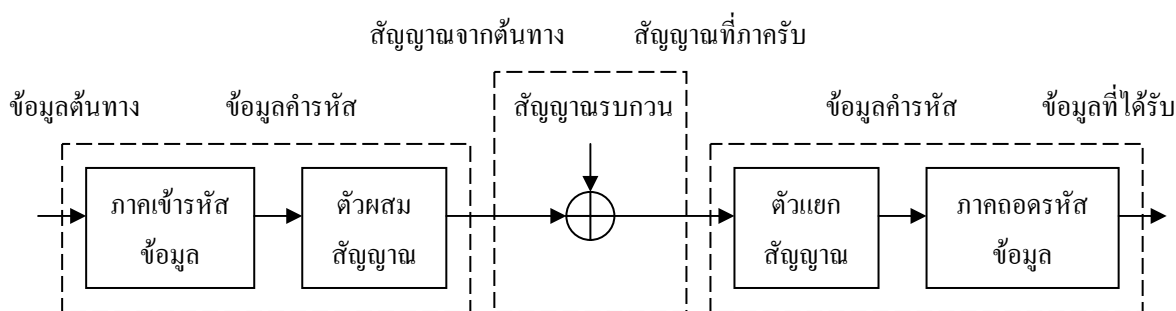
รหัสแก้ไขความผิดพลาด (Error Control Coding) เป็นเทคนิคที่ใช้ในการแก้ไขความผิดพลาดจากการรับส่งข้อมูลในระบบสื่อสารดิจิทัล ซึ่งความผิดพลาดที่เกิดขึ้นนั้นเกิดจากสาเหตุหลายประการ และสาเหตุเหล่านี้ไม่สามารถคาดคะเนเวลาการเกิดได้ โดยเฉพาะสาเหตุที่สำคัญคือการรบกวนจากสัญญาณรบกวน (Noise) ดังนั้นจึงจำเป็นต้องใช้รหัสดังกล่าวเพื่อป้องกันหรือแก้ไขความผิดพลาดจากสาเหตุเหล่านี้

ในบทนี้จะกล่าวถึงรหัสแก้ไขความผิดพลาดแบบคอนวอลูชัน และการถอดรหัสแบบวิทีวเทอร์บีซึ่งเป็นวิธีการแก้ไขข้อผิดพลาดของข้อมูลที่รับได้จากสัญญาณรบกวนให้ถูกต้องที่มีประสิทธิภาพส่งผลให้ข้อมูลที่ถูส่งไปในระบบสื่อสารนั้นมีความน่าเชื่อถือ

3.1 หลักการทำงานพื้นฐานของวิธีการเข้ารหัส

วิธีการเข้ารหัสถูกพัฒนาขึ้นเพื่อลดอัตราการเกิดความผิดพลาดของข้อมูลที่ถูส่งผ่านระบบสื่อสาร สามารถทำให้อัตราการเกิดความผิดพลาดของข้อมูลลดลงโดยปราศจากการเพิ่มกำลังของเครื่องส่งให้สูงขึ้นได้ ซึ่งในการทำงานนั้นข้อมูลดิจิทัลที่ถูส่งผ่านระบบสื่อสารจะถูกนำมาผ่านกระบวนการเข้ารหัส (Encoding) เพื่อแปลงลักษณะของข้อมูลให้กลายเป็นข้อมูลคำรหัส (Codeword) โดยเป็นรูปแบบของข้อมูลดิจิทัลที่ถูส่งผ่านระบบสื่อสาร สำหรับภาครับจะมีการนำข้อมูลที่แสดงถึงข้อมูลคำรหัสที่ถูตรวจจับได้มาผ่านกระบวนการถอดรหัส (Decoding) เพื่อแปลงข้อมูลให้กลายเป็นข้อมูลดิจิทัลเพื่อนำไปใช้งานต่อไป พร้อมทั้งทำการแก้ไขข้อมูลให้มีความถูกต้องมากยิ่งขึ้น

รูปแบบของรหัสที่ถูใช้งานมีหลายรูปแบบด้วยกัน โดยทั่วไปแล้วมีการแบ่งออกเป็น 2 รูปแบบ ได้แก่ รหัสแบบบล็อกและรหัสแบบคอนวอลูชัน สำหรับรหัสแบบบล็อกเป็นรูปแบบแรกของรหัสที่ถูพัฒนาขึ้นถูกนำไปประยุกต์ใช้ในงานหลายประเภท ตัวอย่างเช่น รหัสแฮมมิง (Hamming Code) และรหัสรีด-โซโลมอน (Reed Solomon Code) ที่ถูกนำมาใช้ในอุปกรณ์สำหรับบันทึกข้อมูลเพื่อป้องกันความผิดพลาด เป็นต้น สำหรับรหัสแบบคอนวอลูชันถูกพัฒนาขึ้นเพื่อรองรับการใช้งานสำหรับระบบสื่อสารที่มีการส่งข้อมูลด้วยอัตราเร็วสูง เช่น ระบบสื่อสารผ่านดาวเทียม และ ระบบโทรศัพท์เคลื่อนที่ เป็นต้น โดยมีการพัฒนารูปแบบในการทำงานของภาคถอดรหัสเพื่อรองรับการทำงานของรหัสแบบคอนวอลูชันหลายรูปแบบ เช่น วิธีการถอดรหัสแบบ Sequential Decoding วิธีการ Majority-Logic หรือ Threshold Decoding และ วิธีการ Maximum a Posteriori (MAP)



รูปที่ 3.1 แบบจำลองระบบสื่อสารที่มีการนำวิธีการเข้ารหัสมาใช้งาน

สำหรับวิธีการถอดรหัสที่นิยมใช้งานได้แก่วิธี Viterbi decoding เนื่องจากประสิทธิภาพในการแก้ไขข้อผิดพลาดมีความเหมาะสมและวงจรที่ใช้งานไม่ซับซ้อน เนื่องจากความต้องการในการใช้งานระบบสื่อสารที่มีการส่งข้อมูลด้วยอัตราเร็วที่สูงขึ้นภายใต้ขนาดแบนด์วิดท์ของช่องสัญญาณที่จำกัด ดังนั้นจึงได้มีการพัฒนารูปแบบของรหัสให้สามารถใช้งานร่วมกับระบบสื่อสารที่มีการมอดูเลตแบบหลายระดับได้ ตัวอย่างเช่นรหัส Trellis-Coded Modulation (TCM) ซึ่งเป็นวิธีการที่ถูกพัฒนาขึ้นมาจากรหัสคอนวอลูชัน [24]

3.2 รหัสแบบคอนวอลูชัน

การเข้ารหัสแบบคอนวอลูชันและการถอดรหัสแบบวิธีไวเทอร์บี (Viterbi Algorithm) ซึ่งเป็นวิธีการป้องกันการผิดพลาดล่วงหน้า (Forward Error Correction :FEC) เป็นเทคนิครูปแบบหนึ่งที่ใช้แก้ไขการผิดพลาดของข้อมูลที่รับได้จากสัญญาณรบกวนให้ถูกต้องที่มีประสิทธิภาพ ทำให้ข้อมูลที่ถูกส่งไปในระบบสื่อสารนั้นมีความน่าเชื่อถือมากขึ้น สำหรับการถอดรหัสคอนวอลูชันนั้นถูกค้นพบโดย Elias ในปี 1955 ซึ่งวิธีการถอดรหัส คอนวอลูชัน มีอยู่หลายแบบด้วย แต่วิธีการถอดรหัสแบบวิธีไวเทอร์บีซึ่งเป็นวิธีที่มีการนิยมใช้งานมากที่สุด การถอดรหัสแบบวิธีไวเทอร์บีค้นพบโดยไวเทอร์บีในปี 1967 ซึ่งจะมีลักษณะการทำงานของไวเทอร์บี เป็นแบบ Maximum-likelihood decoding algorithm ซึ่งหมายถึงในการทำงานของการถอดรหัสนั้น จะทำการค้นหาเส้นทางของข้อมูลในแผนภาพทรีลิส (Trellis Diagram) ที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่ส่งมามากที่สุด [25]

3.2.1 หลักการเข้ารหัส

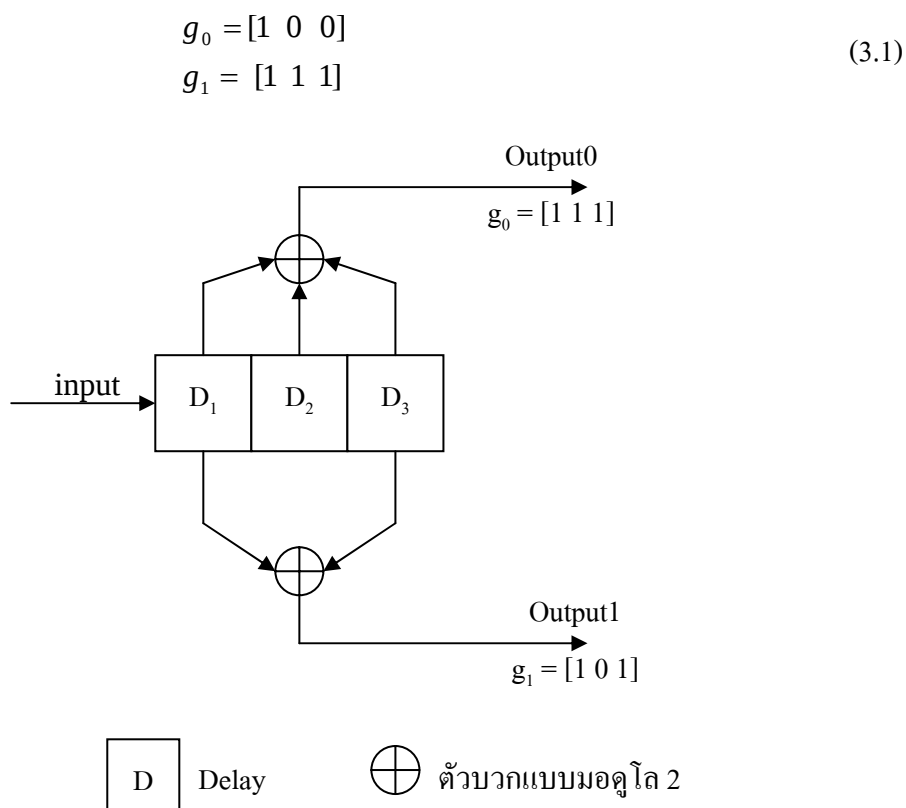
การทำงานของวงจรเข้ารหัสคอนวอลูชันนั้น ข้อมูลอินพุต (k) จะถูกส่งเข้าสู่วงจรเข้ารหัสคอนวอลูชันอย่างต่อเนื่องตามลำดับเวลาการทำงานของ ชิฟริจิสเตอร์ ข้อมูลจากการเข้ารหัส (n) เกิดจากการนำข้อมูลอินพุต (k) และข้อมูลภายในชิฟริจิสเตอร์มาบวกแบบมอดูโล 2 (Modulo 2) ตาม

จำนวนของค่า Constraint Length (K) ซึ่งเป็นค่าที่แสดงจำนวนของข้อมูลอินพุต (k) และจำนวนของชิพรีจิสเตอร์ที่จะนำมารวมกันแบบมอดูโล 2 เพื่อที่จะหาค่าเอาต์พุตจากการเข้ารหัส ส่วนค่า Generator Polynomial (g) เป็นค่าที่ใช้แสดงตำแหน่งของข้อมูลภายในชิพรีจิสเตอร์ที่จะนำมารวมกันแบบมอดูโล 2 ค่า Generator Polynomial นี้นิยมเขียนในรูปแบบของเลขฐานแปด เช่น ตำแหน่งข้อมูลที่จะนำมารวมกันแบบมอดูโล 2 เป็น “111” นิยมเขียนเป็น 7_8 ตัวอย่างวงจรเข้ารหัสคอนโวลูชันแสดงดังรูปที่ 3.2

ในส่วนของอัตราส่วนการเข้ารหัส R เป็นอัตราส่วนของจำนวนข้อมูลอินพุต (k) ต่อข้อมูลเอาต์พุต (n) หรือ $\frac{k}{n}$ โดยจำนวนเอาต์พุตของการเข้ารหัสคอนโวลูชันมีจำนวนมากกว่าอินพุต จากรูปที่ 3.2 วงจรมีอัตราส่วนการเข้ารหัสเท่ากับ $\frac{1}{2}$ ค่า Constraint Length เท่ากับ 3 และค่า Generator Polynomials g_0 เท่ากับ 7_8 และ g_1 เท่ากับ 5_8

3.2.2 การวิเคราะห์การทำงานของวงจรเข้ารหัส

การวิเคราะห์การทำงานของวงจรเข้ารหัสคอนโวลูชันนี้จะใช้ค่า Generator Polynomial ในการคำนวณหาผลลัพธ์ที่ได้จากการเข้ารหัส ซึ่งวงจรของการเข้ารหัสนั้นค่า Generator Polynomial จะแสดงในรูปแบบตัวเลขฐานสอง ดังตัวอย่างแสดงในรูปที่ 3.2



รูปที่ 3.2 การเข้ารหัสแบบคอนโวลูชัน ที่ $K=3$

การคำนวณหาค่าเอาต์พุตที่ได้จากการเข้ารหัสนั้นจะนำค่า Generator Polynomial มาทำการคอนโวลูชัน (Convolution) กับข้อมูลอินพุตดังสมการ

$$Output_i = [Input] * g_i \quad (3.2)$$

ตัวอย่างในกรณีวงจรเข้ารหัสที่แสดงดังรูปที่ 3.2 สมมติให้ข้อมูลที่ต้องการเข้ารหัสคือ “1 0 0 1” การคำนวณหาค่าเอาต์พุตที่ได้จากการเข้ารหัสสามารถคำนวณค่าได้ดังนี้

$$Output_0 = [1 0 0 1] * [1 0 0] = 1 0 0 1 0 0 \quad (3.3)$$

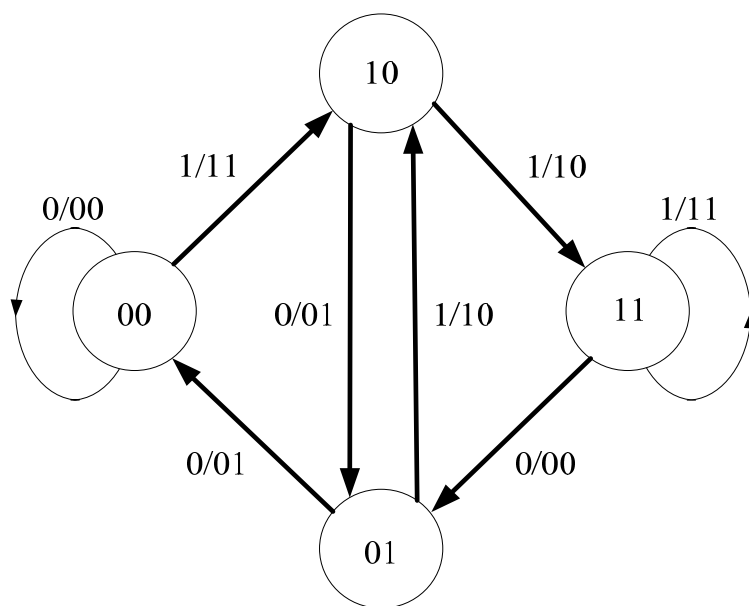
$$Output_1 = [1 0 0 1] * [1 1 1] = 1 1 1 1 1 1 \quad (3.4)$$

ค่าเอาต์พุตที่ได้จากการเข้ารหัสข้อมูลนั้นเกิดจากข้อมูลที่ได้จาก $Output_0$ และ $Output_1$ ซึ่งเมื่อนำมารวมกันจะมีค่าเท่ากับคํารหัส ดังนี้ “11 01 01 11 01 01”

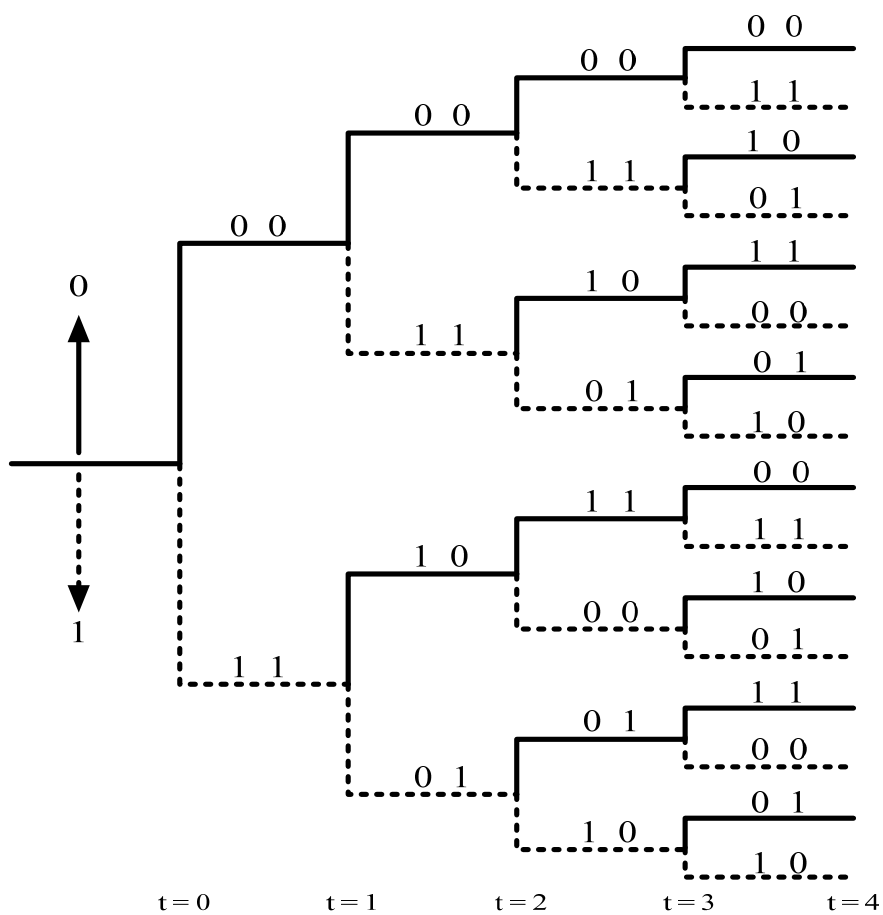
3.2.3 แผนภาพแสดงการทำงานของวงจรเข้ารหัส

ในการวิเคราะห์การทำงานของวงจรเข้ารหัสแบบ คอนโวลูชัน นั้น นอกจากจะมีการใช้สมการของ Generator polynomial ในการแสดงลักษณะของวงจรเข้ารหัสแล้ว จะมีวิธีการแสดงการทำงานต่างๆ ของวงจรเข้ารหัส โดยการใช้รูปภาพแสดงความสัมพันธ์ระหว่างข้อมูลที่ถูกเก็บไว้ในวงจรเข้ารหัสข้อมูลที่ถูกป้อนเข้ามา ณ เวลานั้น และ ค่าของคํารหัสที่จะถูกส่งออกไป เมื่อมีข้อมูลในกรณีต่างๆ ป้อนเข้ามา ซึ่งรูปแบบของภาพที่ใช้แสดงการทำงานของวงจรเข้ารหัสนั้น จะมีอยู่ด้วยกัน 3 รูปแบบด้วยกัน ได้แก่ แผนภาพแสดงสถานะ (State Diagram) แผนภาพแบบต้นไม้ (Tree Diagram) และแผนภาพแบบทรีลิส (Trellis Diagram) ซึ่งจะมีรายละเอียดดังต่อไปนี้

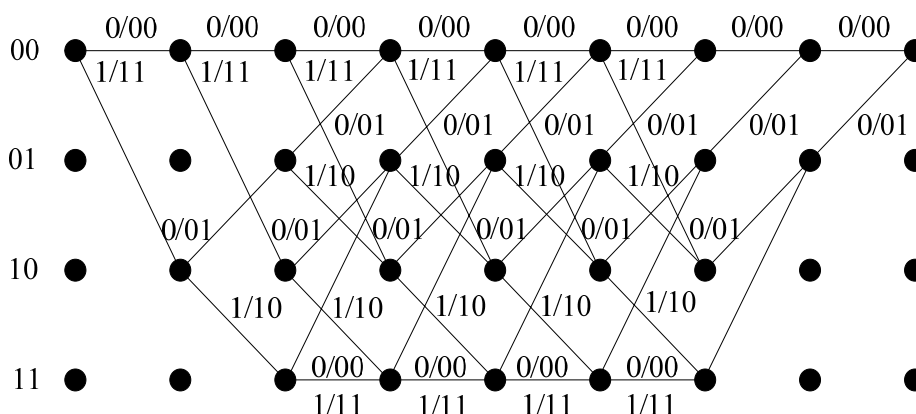
รูปแบบแรกคือแบบแผนภาพแสดงสถานะจะแสดงค่าของข้อมูลใน shift register และ เอาท์พุทของตัวเข้ารหัส ซึ่งแสดงดังรูปที่ 3.3 ตัวเลขที่อยู่ในวงกลมแต่ละวงนั้น จะหมายถึงสถานะต่างๆ ของข้อมูลที่ถูกเก็บไว้ใน Shift register ซึ่งในกรณีของวงจรเข้ารหัสที่ใช้เป็นตัวอย่างนั้น จำนวนสถานะ(State)ทั้งหมดเท่ากับ 4 สถานะ และสำหรับลูกศรที่ถูกแสดงไว้ในรูปนั้นจะแสดงถึงลักษณะของการเปลี่ยนแปลงการทำงานจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง ซึ่งจะขึ้นอยู่กับข้อมูลที่ป้อนเข้ามา ณ เวลานั้นๆ เช่นกรณีที่มีข้อมูลที่เก็บไว้มีค่าเป็น 00 เมื่อมีข้อมูล 1 ป้อนเข้ามาจะมีผลทำให้ข้อมูลที่เก็บไว้ถูกเปลี่ยนไปเป็น 10 และจะมีการส่งค่า 11 ออกไปจากวงจร (แทนด้วยสัญลักษณ์ 1/11 ดังรูปที่ 3.3) แต่ถ้าหากว่ามีการป้อนข้อมูล 0 เข้ามา วงจรก็จะยังคงมีสถานะเป็น 00 เหมือนเดิมและจะมีการส่งค่า 00 ออกไปจากวงจร (แทนด้วยสัญลักษณ์ 0/00 ดังรูปที่ 3.3)



รูปที่ 3.3 แผนภาพแสดงสถานะ (State Diagram)



รูปที่ 3.4 แผนภาพแบบต้นไม้ (Tree Diagram)



รูปที่ 3.5 แผนภาพแบบทรีลิส (Trellis Diagram)

สำหรับแผนภาพแบบต้นไม้ นั้นจะเป็นการพิจารณาถึงลักษณะของการทำงานของวงจรเข้ารหัสข้อมูล โดยที่จะมีการพิจารณาถึงค่าของผลลัพธ์ที่ได้หลังจากการป้อนข้อมูลต่างๆ เข้าไปในวงจรเข้ารหัสเป็นหลัก ซึ่งจะมีลักษณะของแผนภาพแบบต้นไม้ ดังรูปที่ 3.4 สำหรับการแสดงการทำงานของวงจรเข้ารหัสแบบคอนโวลูชันนั้นจะเป็นการพิจารณาการทำงานโดยการคำนึงถึงข้อมูลที่ป้อนเข้ามาและที่จะถูกส่งออกไปจากภาคเข้ารหัสเป็นหลัก ซึ่งในการพิจารณานั้นจะเริ่มต้น ณ ตำแหน่งรากของแผนภาพแบบต้นไม้ซึ่งในรูปที่ 3.4 นั้น จะอยู่ในตำแหน่งซ้ายมือสุดของรูป ซึ่งจะมีการนำข้อมูลที่ถูกรับเข้ามาภายในวงจรเข้ารหัสเป็นตัวกำหนดทิศทางการเดินทางของข้อมูลในแผนภาพแบบต้นไม้ โดยในกรณีของแผนภาพแบบต้นไม้ตัวอย่างนั้น จะกำหนดให้มีการเลื่อนตำแหน่งไปทางข้างบนเมื่อมีการรับข้อมูล 0 เข้ามา และจะเลื่อนตำแหน่งลงล่างเมื่อมีการรับข้อมูล 1 เข้ามา ซึ่งหลังจากที่มีการเลื่อนตำแหน่งที่ใช้พิจารณาแล้ว จะมีการพิจารณาถึงข้อมูลที่จะถูกส่งออกไปจากภาคเข้ารหัส ณ เวลานั้นๆ จากข้อมูลที่อยู่เหนือเส้นทางในตำแหน่งที่มีการพิจารณา

การแสดงการทำงานของวงจรเข้ารหัสโดยใช้แผนภาพแบบทรีลิสนั้น จะเป็นการนำการแสดงการทำงานของวงจรเข้ารหัสโดยใช้แผนภาพแสดงสถานะมาทำการเปลี่ยนแปลงรูปแบบให้อยู่ในอีกลักษณะหนึ่ง ที่แสดงถึงการเปลี่ยนแปลงของข้อมูลต่างๆภายในวงจรเข้ารหัส, ข้อมูลที่ป้อนเข้ามาและค่ารหัสที่จะถูกส่งออกไป ณ เวลาต่างๆ โดยที่จะมีลักษณะของแผนภาพแบบทรีลิส ดังรูปที่ 3.5

จากรูปจะเป็นการแสดงการทำงานของวงจรเข้ารหัสข้อมูลแบบคอนโวลูชันในรูปที่ 3.2 ที่มีการนำข้อมูลในอดีจำนวน 2 บิต มาทำการประมวลผลร่วมกับข้อมูล ณ เวลานั้น (จำนวน state ทั้งหมดใน trellis diagram จะมีค่าเท่ากับ $2^2 = 4$ state) และจะมีข้อมูลป้อนเข้ามาภายในวงจรครั้งละ 1 บิต ($k = 1$) ซึ่งในกรณีนี้ จะมีข้อมูลที่จะเข้ารหัสทั้งหมด 6 บิต ซึ่งเส้นทางต่างๆ ที่อยู่ในแผนภาพแบบทรีลิส นั้น จะแสดงถึงลักษณะการเปลี่ยนแปลง สถานะของวงจรและตัวเลข x/y ที่อยู่เหนือทางเดินในแต่ละเส้นทางนั้น จะแสดงถึงข้อมูลที่ป้อนเข้ามา และค่ารหัส (Codeword) ที่จะถูก

ส่งออกไปเมื่อมีการป้อนข้อมูลนั้นเข้ามา ซึ่งเมื่อพิจารณาลักษณะของแผนภาพแบบทรีลิสแล้วจะพบว่ารูปแบบของแผนภาพแบบทรีลิสในแต่ละสถานะการทำงานนั้น จะมีลักษณะที่คล้ายกันแต่จะมีความแตกต่างกันเฉพาะส่วนหัวและท้าย ซึ่งเป็นผลมาจากข้อมูลที่เก็บอยู่ในวงจรนั้นจะมีค่าที่เริ่มต้นจากสถานะที่มีข้อมูลเป็น 0 ทั้งหมด และจะจบลงที่สถานะข้อมูลเป็น 0 เช่นกัน ดังนั้นเส้นทางอื่นๆ ที่ไม่ผ่านจุดที่มีข้อมูลเป็น 0 ทั้งหมด ณ จุดเริ่มต้นและจุดสุดท้ายนั้นจะไม่ถูกนำมาพิจารณา ดังนั้นขนาดของความยาวในแผนภาพแบบทรีลิสนั้นจึงขึ้นอยู่กับข้อมูลที่ถูกลำเลียงเข้ารหัส

3.2.4 การถอดรหัสแบบคอนโวลูชัน

การถอดรหัสคอนโวลูชันนั้นมีขั้นตอนที่ซับซ้อนกว่าวิธีการเข้ารหัสมาก ซึ่งวิธีที่ใช้ในการถอดรหัสคอนโวลูชันนั้นมี 3 วิธี คือ Sequential Decoding, Threshold Decoding และ Viterbi Decoding ซึ่งมีรายละเอียดดังนี้ [25]

- วิธี Sequential Decoding เป็นวิธีการถอดรหัสคอนโวลูชันแบบแรกที่น่าเสนอโดย Wozencraft ในปี 1961 วิธีการถอดรหัสใช้แผนภาพทรีมาใช้ในการค้นหาเส้นทางของข้อมูล วิธีนี้มีข้อเสียคือใช้เวลาในการถอดรหัสและมีการทำงานที่ซับซ้อน
- วิธี Threshold Decoding เป็นวิธีที่น่าเสนอโดย Massey ในปี 1963 การถอดรหัสด้วยวิธีนี้ใช้หน่วยตัดสินใจที่เรียกว่า Threshold Gate ในการพิจารณาตัดสินใจจากความผิดพลาด ซึ่งวิธีนี้มีความซับซ้อนน้อยแต่มีประสิทธิภาพการถอดรหัสที่ต่ำ
- วิธี Viterbi Decoding เป็นวิธีที่น่าเสนอโดย A. J. Viterbi ในปี 1967 สำหรับการถอดรหัสวิธีนี้จะใช้อัลกอริทึมที่เรียกว่า Maximum Likelihood Decoding และแผนภาพทรีลิสในการถอดรหัส

3.2.5 การถอดรหัสแบบวิธีไวเทอร์บี (Viterbi Algorithm)

การถอดรหัสแบบวิธีไวเทอร์บีจะมีลักษณะการทำงานเป็นแบบ Maximum-likelihood Decoding Algorithm ซึ่งผลลัพธ์ที่ได้จากการทำงานนั้น จะได้เส้นทางเพียงหนึ่งเส้นทางจากเส้นทางทั้งหมดในแผนภาพแบบทรีลิสที่มีลักษณะที่เหมือนกับข้อมูลที่รับได้มากที่สุด ซึ่งการถอดรหัสแบบวิธีไวเทอร์บีนั้น จะมีรูปแบบสำหรับการถอดรหัสที่ใช้งานอยู่ 2 ลักษณะด้วยกัน ได้แก่ แบบหยาบ (Hard-Decision) และแบบละเอียด (Soft-Decision)

สำหรับการทำงานของวงจรถอดรหัสที่ใช้กระบวนการตัดสินใจแบบหยาบนั้น จะเป็นการพิจารณาข้อมูลที่รับเข้ามา โดยการพิจารณาว่าข้อมูลที่รับเข้ามาในแต่ละบิตนั้น มีค่าของข้อมูลเป็น 0 หรือ 1 เท่านั้น ส่วนการถอดรหัสที่มีการใช้กระบวนการตัดสินใจแบบละเอียดจะเป็นการพิจารณาถึงข้อมูลที่รับเข้ามาได้โดยการทำการตัดสินใจระดับของข้อมูลที่รับเข้ามาได้โดยการแบ่งระดับของสัญญาณที่ใช้ในการ คำนวณค่าเมตริกซ์ที่มากกว่า 2 ระดับ ซึ่งผลลัพธ์ที่ได้นั้น จะได้ข้อมูลรายละเอียดของข้อมูลที่ส่งมาที่มากกว่ากรณีของการตัดสินใจแบบหยาบ ซึ่งข้อมูลที่ี้ได้จากการ

ตัดสินใจ (soft-output) นั้น จะถูกนำมาใช้ในการคำนวณหาค่าเมตริกซ์เพื่อเปรียบเทียบข้อมูลที่รับเข้ามา ณ เวลานั้นๆ กับข้อมูลที่อยู่ในเส้นทางต่างๆ ณ เวลานั้น ซึ่งจะมีรูปแบบที่ใช้ในการคำนวณที่แตกต่างกันไป โดยในการทำงานต่างๆ ของวิธีไวเทอร์บีนั้นจะต้องมีการคำนวณหาความแตกต่างระหว่างข้อมูลที่รับเข้ามาและค่าที่อยู่ในเส้นทางต่างๆ เพื่อใช้ในการคำนวณการตัดสินใจ โดยกระบวนการที่ใช้ในการทำการหาเส้นทางที่ดีที่สุดนั้น จะใช้วิธีการทำงานที่มีชื่อว่าไวเทอร์บีอัลกอริทึมซึ่งจะเป็นกระบวนการที่ใช้ในการค้นหาเส้นทางที่อยู่ในแผนภาพทริลิสที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับได้มากที่สุดเพื่อที่จะนำข้อมูลในเส้นทางนั้นมาคำนวณค่าของข้อมูลที่ถูกส่งมา โดยที่ในกระบวนการค้นหาเส้นทางที่เหมาะสมที่สุดโดยใช้วิธีไวเทอร์บีนั้นมีขั้นตอนในการทำงานดังต่อไปนี้

- พิจารณาแบ่งข้อมูลที่รับเข้ามาออกเป็นข้อมูลย่อยๆ จำนวน m ช่วง ซึ่งแต่ละช่วงนั้นมีขนาดของข้อมูลเท่ากับ n บิต
- ทำการวาดแผนภาพทริลิสที่มีจำนวนสถานะในการทำงานเท่ากับ m สถานะ โดยจะมีการพิจารณาเฉพาะเส้นทางที่มีความเป็นไปได้ว่าจะถูกส่งมาเท่านั้น โดยสำหรับที่สถานะของแผนภาพทริลิสตั้งแต่ $K-1$ ขึ้นไปนั้น ให้วาดเฉพาะเส้นทางที่จะพุ่งเข้าหาสถานะของวงจรที่มีข้อมูลเป็น 0 ทั้งหมด
- กำหนดค่าตัวแปร 1 เท่ากับ 1 และทำการกำหนดค่าเริ่มต้นของตัวแปรเมตริกซ์ในสถานะเริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมด ให้มีค่าของเมตริกซ์เท่ากับ 0
- ทำการคำนวณหาความแตกต่างของข้อมูล (distance) ระหว่างข้อมูลที่รับได้ชุดที่ 1 กับข้อมูลในเส้นทางในการเปลี่ยนแปลงสถานะในแผนภาพทริลิสจากสถานะที่ 1 ไปเป็น $1+1$
- นำค่าที่คำนวณได้นั้นไปบวกกับค่าเมตริกซ์สะสมของสถานะ 1 เพื่อคำนวณหาค่าของเมตริกซ์สะสมในสถานะที่ $1+1$ เพื่อใช้ในการตัดสินใจเลือกเส้นทางที่เหมาะสมที่สุดในการเปลี่ยนแปลงข้อมูลไปยังสถานะนั้นๆ โดยในแต่ละสถานะนั้น จะมีจำนวนเส้นทางทั้งหมดจำนวน $2k$ เส้นทางที่จะพุ่งเข้าสถานะเดียวกัน
- พิจารณา ณ ตำแหน่งในสถานะที่ $1+1$ ในแต่ละสถานะนั้น ทำการเลือกเส้นทางที่มีค่าเมตริกซ์สะสมที่มีค่าน้อยที่สุดที่พุ่งเข้าหาในแต่ละสถานะโดยที่เส้นทางที่ถูกเลือกนั้นจะถูกเรียกว่า “Survivor” ซึ่งจะเป็นเส้นทางที่ถูกเก็บไว้ทำการคำนวณในสถานะต่อไป และสำหรับเส้นทางอื่นๆ ที่ไม่ได้ถูกเลือกนั้น จะถูกเรียกว่า “Forgetting” โดยจะถูกลบทิ้งออกไปจากระบบการตัดสินใจ
- ถ้าหากว่า 1 นั้นมีค่าเท่ากับ m แล้วให้ทำงานในขั้นตอนต่อไปได้ แต่ถ้ายังมีค่าน้อยกว่า จะต้องมีค่าเพิ่มค่า 1 ขึ้นอีก 1 จากนั้นจึงกลับไปทำงานที่ขั้นตอนที่ 4 ใหม่

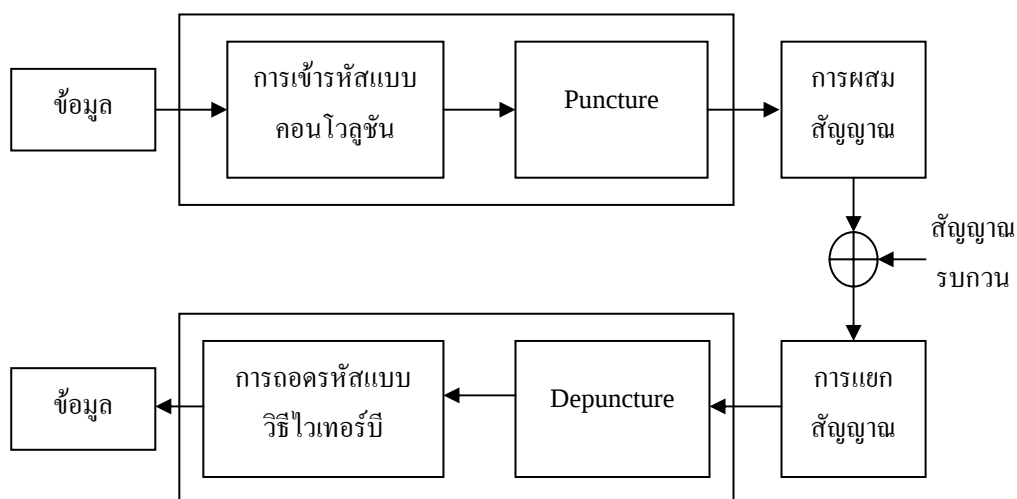
- เริ่มต้นพิจารณา ณ สถานะที่ $m+1$ ที่มีสถานะของข้อมูลสถานะเป็น 0 ทั้งหมด ทำการเลือกเส้นทางที่เป็น “Survival” ซึ่งเป็นเส้นทางที่ถูกเลือกที่เหลืออยู่ย้อนกลับไปจนกระทั่งถึงสถานะเริ่มต้นของการทำงานที่มีสถานะในการทำงานเป็น 0 ทั้งหมด ซึ่งเส้นทางที่ได้นั้น จะเป็นเส้นทางที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับเข้ามามากที่สุด ซึ่งจะถูกนำไปใช้ในการคำนวณหาข้อมูลข่าวสารที่ถูกส่งมา โดยข้อมูลข่าวสารที่จะถูกส่งออกไปจากภาคถอดรหัส นั้น จะเป็นการส่งข้อมูลทั้งหมดที่อยู่ในเส้นทางส่งออกไป ยกเว้นข้อมูล 0 จำนวน $k(K-1)$ บิต ที่อยู่ท้ายสุดนั้น จะถูกตัดทิ้งไป

3.2.6 การลดจำนวนบิตข้อมูล (Punctuation)

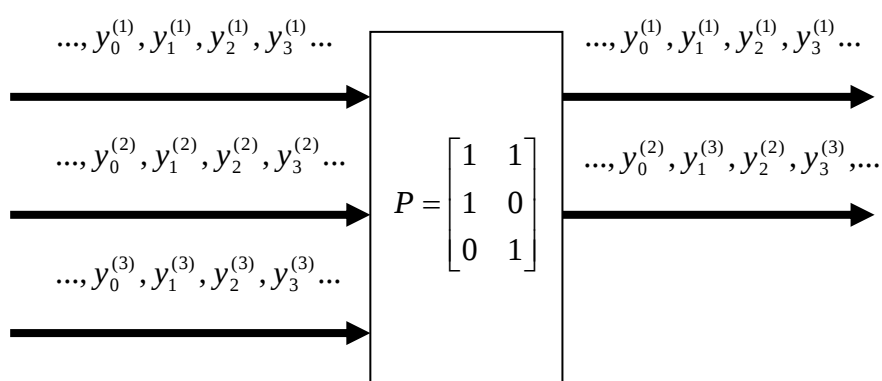
ภาคลดจำนวนบิตข้อมูล (Punctured) ถูกนำมาใช้ในวงจรเข้ารหัสในตำแหน่งหลังจากการเข้ารหัสแบบคอนโวลูชัน ดังรูปที่ 3.6 โดยมีวัตถุประสงค์เพื่อที่จะทำให้จำนวนบิตของข้อมูลที่จะถูกส่งผ่านระบบสื่อสารนั้นมีจำนวนที่ลดลง โดยจะเป็นการการลบข้อมูลที่จะทำการส่งบางส่วนออกไปตามรูปแบบที่มีการกำหนดไว้ โดยในการแสดงถึงรูปแบบในการทำงานนั้น จะมีการแสดงถึงรูปแบบในการลบข้อมูลที่จะทำการส่งโดยใช้ตัวแปรเมตริกซ์ที่จะแสดงถึงรูปแบบในการลบข้อมูลที่ได้จากการเข้ารหัส ดังเช่นเมื่อพิจารณาข้อมูลที่ได้จากการเข้ารหัสจากวงจรเข้ารหัสแบบคอนโวลูชันจะมีลักษณะของข้อมูลที่ได้จากการ puncture เป็นดังรูปที่ 3.7 แสดงถึงการทำงานของภาคลดจำนวนบิตนั้นเป็นการลดจำนวนบิตของข้อมูลในส่วนของข้อมูลที่ได้จากการทำงานของวงจรเข้ารหัสที่มีการเข้ารหัสในอัตรา $1/3$ โดยข้อมูลที่ได้จากการเข้ารหัสเป็น $y_k^{(1)}, y_k^{(2)}$ และ $y_k^{(3)}$ ตามลำดับ โดยจะใช้การสลับกันส่งข้อมูลระหว่างสัญญาณ $y_k^{(2)}$ และ $y_k^{(3)}$ ดังนั้นจะทำให้อัตราการเข้ารหัสข้อมูลของการเข้ารหัสนั้นมีค่าที่เพิ่มขึ้นจาก $1/3$ เป็น $1/2$

ในทางกลับกันเมื่อข้อมูลต่างๆ นั้นถูกส่งมาถึงปลายทาง จะต้องมีการนำข้อมูลมาทำการ Depuncture เพื่อเปลี่ยนแปลงรูปแบบของข้อมูลที่รับได้ให้กลับมาอยู่ในรูปแบบเดิม จากตัวอย่างในรูปที่ 3.8 เป็นกระบวนการ Depuncture ของรูปที่ 3.7 ซึ่งสมมติให้ข้อมูลที่ได้รับมามีค่าเป็น $y_k^{(1)}$ และ $y_k^{(2)}$ สำหรับหลักการทำงานของกระบวนการ Depuncture คือการเพิ่มบิตที่ไม่ทราบค่าซึ่งในที่นี้สมมติให้เป็นค่า C ในตำแหน่งที่ได้ทำการ Puncture ออกไป จากตัวอย่างในรูปที่ 3.8 นั้นจะทำการเพิ่มค่า C ในตำแหน่งที่ 3 เมื่อมีคาบเวลาในการส่งเป็นคู่ และเพิ่มค่า C ในตำแหน่งที่ 2 เมื่อมีคาบเวลาในการส่งเป็นคี่ ซึ่งค่า C นั้นเป็นค่าที่สามารถคำนวณได้จากความน่าจะเป็นของข้อมูลจากภาคส่ง

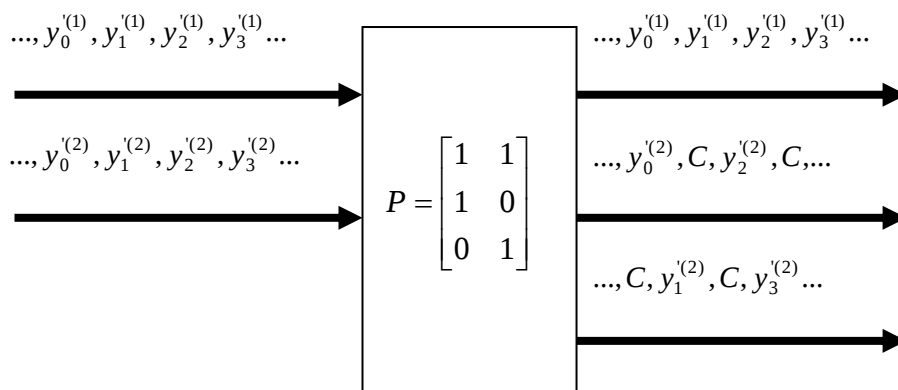
สำหรับในงานวิจัยนี้เป็นการนำรหัสแบบคอนโวลูชันมาใช้งานร่วมกับการผสมสัญญาณเชิงแสงในรูปแบบต่างๆ และการสื่อสารแบบโพดอนหลายระดับเพื่อทดสอบประสิทธิภาพในการรับส่งข้อมูล และมีการประยุกต์ใช้งานของรหัสแบบคอนโวลูชันที่มีการลดจำนวนบิตข้อมูลเพื่อให้ได้รูปแบบของข้อมูลที่เหมาะสมต่อการเข้ารหัสเพื่อแก้ไขข้อผิดพลาดแบบคอนโวลูชัน



รูปที่ 3.6 การทำงานของรหัสแบบคอนโวลูชันที่มีการลดจำนวนบิตข้อมูล



รูปที่ 3.7 ตัวอย่างวิธีการ Punctuation



รูปที่ 3.8 ตัวอย่างวิธีการ Depunctuation