วิศวกรรมสาร ม.ข. ปีที่ 31 ฉบับที่ 4 (395-408) ตุลาคม - ธันวาคม 2547
KKU Engineering Journal Vol. 31 No. 4 (395-408) October –December 2004

การประยุกต์วิธีการถอดรหัสแบบขนานกับอัลกอริทึมการถอดรหัสแบบหลายสแตค

วิระสิทธิ์ อิ่มถวิล มงคล คูพิมาย และวัฒนา กาสังข์
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

บทคัดย่อ

การถอดรหัสแบบหลายสแตค (the multiple stack algorithm, MSA) เป็นการถอดรหัสแบบซีเควนเชียลที่มีประสิทธิภาพและไม่มีอีเรชัว (erasurefree) และใช้ในการถอดรหัสคอนโวลูชันที่มีค่าความยาวคอนสเตรนท์สูงๆ ได้ดีกว่าการถอดรหัสแบบวิเทอบิ เนื่องจากความซับซ้อนในการคำนวณไม่ขึ้นอยู่กับค่าความยาวคอนสเตรนท์ หลักการพื้นฐานของ MSA คือ การใช้ประโยชน์จากสแตคอันดับสูงสำหรับบล็อกข้อมูลที่มีสัญญาณรบกวนมากๆ ที่ต้องการใช้จำนวนรอบของการคำนวณเพื่อหาทางเดินที่ถูกต้องในแผนภูมิต้นไม้แทนรหัส (code tree) มากกว่าปกติ แต่ปัญหาของ MSA คือประสิทธิภาพในอัตราความผิดพลาดบิตมักจะมีค่าที่สูงถ้าการถอดรหัสสิ้นสุดโดยค่าลิมิตของการคำนวณ (computational limit, C_{lim}) ในบทความนี้เรานำเสนอการประยุกต์ใช้วิธีการถอดรหัสแบบขนานกับ MSA เพื่อปรับปรุงประสิทธิภาพในอัตราความผิดพลาดบิตของขบวนการถอดรหัสที่ใช้ MSA เดิม ผลการทดสอบประสิทธิภาพเมื่อจำลองระบบด้วยคอมพิวเตอร์ปรากฏว่า อัตราความผิดบิตใน MSA มีค่าลดลงอย่างมากเมื่อใช้วิธีการถอดรหัสแบบขนานโดยตัวประมวลผลหลายตัว วิธีการประยุกต์ใช้การถอดรหัสแบบขนานกับ MSA เพื่อถอดรหัสคอนโวลูชันเป็นจึงทางเลือกหนึ่งที่น่าสนใจโดยเฉพาะอย่างยิ่งในระบบที่ต้องการค่าอัตราความผิดพลาดบิตที่ต่ำๆ

คำสำคัญ : การถอดรหัสแบบขนาน, การถอดรหัสแบบหลายสแตค, รหัสคอนโวลูชัน

A Parallel Decoding Scheme for the Multiple Stack Algorithm

Virasit Imtawil Mongkol Kupimai and Watana Kasang

Department of Electrical Engineering, Faculty of Engineering, Khon Kaen university

Abstract

The multiple stack algorithm (MSA) is an efficient algorithm for erasurefree sequential decoding of long constraint length convolutional codes. The basic idea of the MSA is to make use of higher order stacks for blocks that require an excessive number of computations to be decoded. Unfortunately, those blocks are frequently decoded with poor error performance if the decoding process is terminated with the computational limit, $C_{\lim}$. We propose in this paper a decoding scheme to improve the error performance of the MSA with the use of parallel decoding. It is found by extensive computer simulations that the bit error rate of the MSA can be further improved with our proposed scheme. Parallel decoding scheme for the MSA appears to be an interesting scheme for decoding convolutional codes when low error probabilities are required.

Key word : Parallel decoding, Multiple Stack Algorithm, Convolutional Codes

บทนำ

รหัสคอนโวลูชัน (convolutional codes) เป็นการเข้ารหัสเพื่อแก้ไขความผิดพลาดซึ่งถูกใช้แพร่หลายในระบบสื่อสารทั้งภาคพื้นดินและผ่านดาวเทียม วิธีการถอดรหัสสัญญาณข้อมูลที่เข้ารหัสแบบคอนโวลูชันมีอยู่ 2 วิธีใหญ่ ๆ คือ การถอดรหัสแบบวิเทอบิ (Viterbi decoding) และการถอดรหัสแบบซีเควนเชียล (sequential decoding) [Forney, 1974] บทความนี้จะเกี่ยวข้องกับการถอดรหัสแบบซีเควนเชียล ซึ่งมีข้อดีเหนือการถอดรหัสแบบวิเทอบิคือสามารถใช้กับรหัสคอนโวลูชันที่มีค่าความยาวคอนสเตรนท์สูง ๆได้ เนื่องจากความซับซ้อนในการคำนวณไม่ขึ้นกับค่าความยาว คอนสเตรนท์ ในขณะที่การถอดรหัสแบบวิเทอบิมีค่าความซับซ้อนในการคำนวณเพิ่มขึ้นแบบเอ็กโพเนนเชียลกับค่าความยาวคอนสเตรนท์ [Lin และ Costello, 1983][Haccoun และ Begin, 1989] ด้วยเหตุนี้จึงทำให้การถอดรหัสแบบซีเควนเชียลสามารถให้อัตราความผิดพลาดบิตที่ต่ำมาก ๆ ได้โดยการเพิ่มค่าความยาวคอนสเตรนท์ของตัวเข้ารหัส อย่างไรก็ตาม ข้อเสียของการถอดรหัสแบบซีเควนเชียลคือ ค่าความแปรปรวนในการคำนวณ (computational variability) กล่าวคือ บล็อกข้อมูลที่มีสัญญาณรบกวนน้อยจะใช้เวลาในการถอดรหัสเร็ว ส่วนบล็อกข้อ

มูลที่มีสัญญาณรบกวนมากจะใช้เวลาในการถอดรหัสช้าและบ่อยครั้งที่การถอดรหัสไม่สามารถสิ้นสุดภายในเวลาที่กำหนด ทำให้บล็อกข้อมูลถูกลบซึ่งเรียกว่า อีเรชัว (Erasure) เพื่อที่จะแก้ปัญหาของอีเรชัว Chevillat และ Costello (1976) ได้เสนอ อัลกอริทึมสำหรับการถอดรหัสแบบซีเควนเชียลที่ไม่มีอีเรชัวคือ อัลกอริทึมการถอดรหัสแบบหลายสแตค (MSA) ขึ้น ซึ่งเป็นอัลกอริทึมที่พัฒนามาจากการถอดรหัสแบบสแตคเดียว SSA [Jelinek, 1969] โดยแบ่งสแตคออกเป็นหลายสแตค ลักษณะเด่นของ MSA คือ สแตคหลักจะมีขนาดใหญ่ ส่วนสแตครองจะมีขนาดเล็กมากๆ และมีจำนวนหลาย สแตคในขนาดที่เท่ากัน การแบ่งสแตคลักษณะนี้มีข้อดีคือ ทำให้กระบวนการถอดรหัสสามารถไปถึงโนดสุดท้ายของทางเดินในแผนภูมิต้นไม้แทนรหัสได้เร็ว แต่ข้อด้อยคือประสิทธิภาพในอัตราความผิดพลาดบิดมักจะมีค่าสูง ต่อมา Y. Lin และ S.H. Tu (1997) ได้พัฒนาอัลกอริทึม MSA ให้มีประสิทธิภาพในอัตราความผิดพลาดบิดดีขึ้น โดยใช้วิธีการจัดสแตครองในรูปแบบวงแหวนเสมือน (ring-like structure) ซึ่งการจัดหน่วยความจำลักษณะนี้ทำให้สามารถกำหนดขนาดของสแตครองและจำนวนโนดที่ย้ายระหว่างสแตคให้มีค่าสูงได้แต่ปัญหาที่ตามมาของงานวิจัยลักษณะนี้คือการได้ทางเดินของการถอดรหัส (Decoded path) อาจจะใช้เวลานานและส่งผลถึงการเกิดโอเวอร์โฟล์ (overflow) ที่บัฟเฟอร์ด้านอินพุทได้

ในงานวิจัยนี้เรานำเสนอวิธีการถอดรหัสแบบขนานกับ MSA เพื่อปรับปรุงประสิทธิภาพในอัตราความผิดพลาดบิดของขบวนการถอดรหัสที่ใช้ MSA ผลการทดสอบประสิทธิภาพเมื่อจำลองระบบด้วยคอมพิวเตอร์ปรากฏว่า อัตราความผิดบิดใน MSA มีค่าลดลงอย่างมากเมื่อใช้วิธีการถอดรหัสแบบขนาน แนวคิดวิธีการประยุกต์ใช้การถอดรหัสแบบขนานกับ MSA สำหรับถอดรหัสคอนโวลูชันเป็นจึงทางเลือกหนึ่งที่น่าสนใจโดยเฉพาะอย่างยิ่งในระบบที่ต้องการค่าอัตราความผิดพลาดบิดที่ต่ำๆ

การถอดรหัสแบบขนานสำหรับ MSA โดยใช้ตัวประมวลผลหลายตัว

การถอดรหัสแบบหลายสแตค

การถอดรหัสแบบ MSA เมื่อสแตคหลักเต็มจำนวนโนดข้อมูลที่มีค่าเมตริกซ์สูงสุดจำนวนหนึ่งจะถูกย้ายไปยังสแตครอง จากนั้นกระบวนการถอดรหัสจะดำเนินต่อไปในสแตครองโดยใช้จำนวนโนดที่ย้ายมา ถ้าการถอดรหัสถึงโนดสุดท้ายของแผนภูมิต้นไม้แทนรหัสก่อนที่สแตครองเต็ม เส้นทางเดินจากโนดเริ่มต้นจนถึงโนดสุดท้ายจะถูกเก็บไว้ในรีจิสเตอร์พิเศษ (special register) และเรียกทางเดินที่เก็บไว้ว่า *การตัดสินใจเบื้องต้น*

(tentative decision, TD) หลังจากนั้นตัวถอดรหัสจะลบโนดที่เหลือในสแตครองและกลับมาเริ่มกระบวนการถอดรหัสอีกครั้งในสแตคหลักโดยเริ่มจากโนดที่อยู่บนสุด (ดีที่สุด) ในตอนนี้สแตคหลักจะมีเนื้อที่ว่างเท่ากับจำนวนโนดที่ถูกย้ายในตอนแรก ถ้าการถอดรหัสดำเนินมาถึงโนดสุดท้ายของแผนภูมิต้นไม้แทนรหัสก่อนที่สแตคหลักจะเต็มอีกครั้งตัวถอดรหัสจะเปรียบเทียบค่าเมตริกซ์ของเส้นทางเดินใหม่นี้กับเส้นทางเดิน TD และเลือกทางเดินที่มีค่าเมตริกซ์ที่ดีกว่าเป็นเส้นทางการถอดรหัส ซึ่งวิธีการในลักษณะนี้สามารถให้ผลของการถอดรหัสดีเสมือนกับการถอดรหัสแบบ SSA เมื่อกำหนดขนาดสแตคให้มีค่าโดมากๆ อย่างไรก็ตาม ถ้าสแตคหลักเต็มอีกครั้งสแตครองก็จะถูกสร้างขึ้นมาใหม่และมีการย้ายโนดข้อมูลในลักษณะเดิม การสร้างสแตครองก็จะเกิดขึ้นในลักษณะเช่นนี้ต่อไปเรื่อยๆ จนกระทั่งตัวถอดรหัสถึงโนดสุดท้ายของเส้นทางเดินในแผนภูมิต้นไม้แทนรหัส และทุกครั้งที่มีการถึงโนดสุดท้าย ตัวถอดรหัสก็จะเปรียบเทียบค่าเมตริกซ์ของเส้นทางใหม่กับค่า TD ที่เก็บไว้และเลือกเส้นทางเดินที่มีค่าเมตริกซ์ที่ดีที่สุดเสมอ กระบวนการถอดรหัสจะดำเนินไปในลักษณะเช่นนี้จนกระทั่งจำนวนรอบของการคำนวณถึงค่าลิมิตของการคำนวณ ($C_{\lim}$) ในกรณีเช่นนี้ TD ก็จะถูกเลือกให้เป็นเส้นทางการถอดรหัสสุดท้าย (final decoded path) เพื่อให้การถอดรหัสแบบ MSA เป็นการถอดรหัสที่ไม่มีอีเรชัว สำหรับกรณีที่กำหนดจำนวนโนดที่ย้ายระหว่าง สแตคเท่ากับหนึ่ง ($T = 1$) และอัตราการเข้ารหัสคอนโวลูชันเป็น $1/w$ เมื่อ w คือ จำนวนเอาท์พุตของตัวเข้ารหัส และกำหนดค่าความยาวคอนสเตรนท์เป็น ν ค่าลิมิตของการคำนวณ $C_{\lim}$ จะต้องมีค่ามากกว่าค่าขอบเขตวิกฤตต่ำสุดของการคำนวน (C_{crit}) ซึ่งกำหนดโดย [Chevillat และ Costello, 1976]

$$C_{crit} = \sum_{i=1}^{k-1} (Z_i - 1) + 2(\nu - 1) \qquad (1)$$

โดยที่ $Z_i, i = 1, 2, \cdots, k-1$ เป็นขนาดของสแตคที่ i และ k คือ จำนวนบิตของสัญญาณข้อมูลต่อบล็อก การถอดรหัสที่สิ้นสุดด้วยค่า $C_{\lim}$ ใน MSA ส่วนใหญ่จะเกิดขึ้นกับบล็อกที่มีสัญญาณรบกวนมากๆ ซึ่งกรณีนี้ค่าอัตราความผิดพลาดบิตมักจะมีค่าสูง เนื่องจากบ่อยครั้งที่ TD เป็นเส้นทางเดินที่ไม่ถูกต้อง ถึงแม้ว่าประสิทธิภาพของการถอดรหัสแบบ MSA จะสามารถทำให้ดีขึ้นได้โดยการเพิ่มค่าของ $C_{\lim}$ แต่ก็จะทำให้การถอดรหัสดำเนินไปได้ช้าและใช้หน่วยความจำในการสร้างตัวถอดรหัสมากขึ้นตามไปด้วย

อัลกอริทึมการถอดรหัสแบบขนานใน MSA

ในงานวิจัยนี้มีเป้าหมายหลักคือการปรับปรุงประสิทธิภาพของการถอดรหัสแบบ MSA โดยใช้วิธีการถอดรหัสแบบขนาน ซึ่งได้ออกแบบการใช้ตัวประมวลผลที่มีคุณลักษณะเหมือนกันทุกประการจำนวน n ตัว ทำการถอดรหัสโดยใช้อัลกอริทึม MSA ไปพร้อมๆ กัน ด้วยแนวความคิดของวิธีการถอดรหัสแบบขนานนี้ จะทำให้ค่า TD มีค่าดีขึ้น เนื่องจากการถอดรหัสที่ใช้ตัวประมวลผลหลายตัวสามารถถึงโนดสุดท้ายได้จำนวนครั้งมากขึ้นในเวลาที่กำหนด ดังนั้นโอกาสที่จะได้ทางเดินที่ถูกต้องก็จะมากขึ้นตามไปด้วย ส่งผลให้ค่าอัตราความผิดพลาดบิดมีค่าลดลงเมื่อเปรียบเทียบทั้งสองระบบที่กำหนดให้ค่า $C_{\lim}$ และขนาดของหน่วยความจำรวมเท่ากัน

รูปที่ 1 แสดงโครงสร้างของระบบการถอดรหัสแบบขนานโดยใช้ MSA ซึ่งสามารถอธิบายได้ดังนี้ การถอดรหัสแบบขนานจะมีขนาดหน่วยความจำรวมของสแตคทั้งหมดในระบบสำหรับใช้ในการถอดรหัสเท่ากันกับการถอดรหัสแบบ MSA เมื่อใช้ตัวประมวลผลตัวเดียว และมีขนาดของสแตคทุกสแตคเท่ากัน ข้อแตกต่างของทั้งสองระบบคือ จำนวนสแตครองในระบบ MSA เดิมจะมีจำนวนมากกว่าจำนวนสแตครองของระบบแบบขนาน ถ้ากำหนดให้จำนวนของสแตครองของการถอดรหัสแบบ MSA เท่ากับ n_s และ จำนวนของสแตครองของการถอดรหัสแบบขนานเท่ากับ n_{sp} และ Γ คือ ขนาดของหน่วยความจำรวมของการถอดรหัสแบบ MSA เดิมนั่นคือ $\Gamma = Z_1 + n_s Z$ เมื่อ Z_1 คือ ขนาดของสแตคหลัก และ Z คือ ขนาดของสแตครอง หน่วยความจำของตัวประมวลผลแต่ละตัวของระบบการถอดรหัสแบบขนานสามารถเขียนเป็นสมการได้ดังนี้ คือ

$$\Gamma' = Z_1 + n_{sp} Z, \qquad n_{sp} = \frac{n_s - (n-1)(Z_1 / Z)}{n} \tag{2}$$

เมื่อ Γ' คือ หน่วยความจำรวมของตัวประมวลผลแต่ละตัวของระบบการถอดรหัสแบบขนาน โดยที่

$$\Gamma = n\Gamma' \tag{3}$$

ขั้นตอนการถอดรหัสแบบขนานใน MSA

ขั้นตอนที่ 1 ตั้งค่า $j = 1$ โหลดค่าโนดเริ่มต้นในสแตคหลักของตัวประมวลผลตัวแรก (DEC #1)

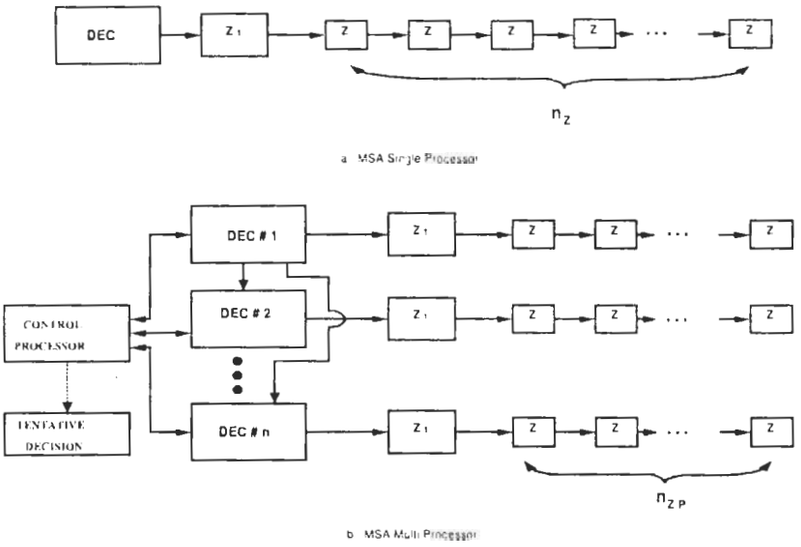
ขั้นตอนที่ 2 DEC #1 เริ่มทำการถอดรหัสแบบ MSA ในสแตคหลักเท่านั้น

ขั้นตอนที่ 3 ตรวจสอบว่าสแตคหลักของ DEC #1 เต็มหรือไม่ ถ้าใช่ให้ตั้งค่า $j = j + 1$ จากนั้นย้ายโนดข้อมูลจำนวน T' โนดในสแตคหลักของ DEC

#1 มายังสแตคหลักของตัวประมวลผลตัวที่ j (DEC # j)

ขั้นตอนที่ 4 ตรวจสอบว่า $j = n$ หรือไม่, ถ้าใช่ให้ไปขั้นตอนที่ 5 ถ้าไม่ใช่ให้กลับไปขั้นที่ 2 ในขณะเดียวกัน DEC #2 ถึง DEC# j จะเริ่มถอดรหัสสัญญาณแบบ MSA อย่างอิสระพร้อมๆ กันในขั้นตอนนี้

ขั้นตอนที่ 5 DEC #1 ถึง DEC # j ดำเนินการถอดรหัสโดยวิธีการถอดรหัสแบบ MSA ไปพร้อมๆ กัน



รูปที่ 1 โครงสร้างการถอดรหัสแบบขนานสำหรับ MSA
(a) MSA ตัวประมวลผลเดียว (b) MSA หลายตัวประมวล

กฎเกณฑ์การสิ้นสุดของขบวนการถอดรหัส

กระบวนการถอดรหัสจะสิ้นสุดเมื่อตัวประมวลผลควบคุม (control processor, CP) ตรวจสอบพบเงื่อนไขในข้อใดข้อหนึ่งต่อไปนี้ คือ

1) การถอดรหัสดำเนินถึงโนดสุดท้ายของเส้นทางเดินในแผนภูมิต้นไม้แทนรหัสในสแตคหลัก (แรก) ของ DEC #1 ซึ่งกรณีนี้ค่าเมตริกซ์ของเส้นทางเดินจากโนดเริ่มต้นถึงโนดสุดท้ายจะถูกนำมาเปรียบเทียบกับค่าเมตริกซ์ของทางเดิน TD จากนั้นเลือกทางเดินที่ดีกว่า (ค่าเมตริกซ์ที่สูงกว่า)

เป็นเส้นทางเดินของการถอดรหัสสุดท้าย

2) จำนวนรอบของการถอดรหัสมีค่าเท่ากับ C_{lim} ในกรณีนี้ทางเดิน TD ที่ถูกบันทึกไว้ในรีจิสเตอร์พิเศษจะเป็นเส้นทางการถอดรหัสสุดท้าย

หน้าที่ของตัวประมวลผลควบคุม CP

1) เปรียบเทียบและบันทึกค่าทางเดินที่ดีที่สุด TD ในรีจิสเตอร์พิเศษเมื่อตัวประมวลผลแต่ละตัวถึงโนดสุดท้าย (ในกรณีที่การถอดรหัสถึงโนดสุดท้ายในสแตคอื่นที่ไม่ใช่สแตคหลักของ DEC #1)

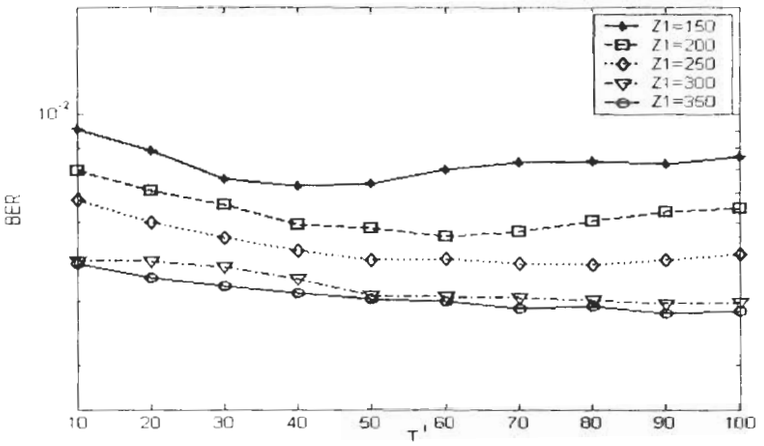
2) สั่งให้ตัวประมวลผลทุกตัวหยุดทำงานเมื่อกระบวนการถอดรหัสดำเนินมาถึงข้อใดข้อหนึ่งของกฎเกณฑ์การสิ้นสุดของขบวนการถอดรหัสข้างต้น

หลักการเบื้องต้นของการถอดรหัสแบบขนานใน MSA คือ การใช้ประโยชน์จากสแตครองให้มากที่สุดโดยการช่วยกันค้นหาทางเดินที่ดีที่สุดในแผนภูมิต้นไม้แทนรหัส โดยตัวประมวลผลหลายตัว

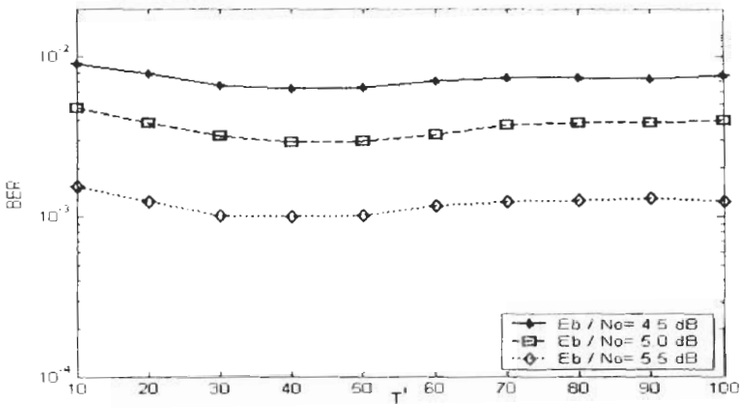
ผลการทดสอบประสิทธิภาพ

เพื่อทดสอบประสิทธิภาพของหลักการที่ได้นำเสนอในงานวิจัยนี้ ผู้วิจัยได้จำลองระบบโดยการเขียนโปรแกรมคอมพิวเตอร์ภาษาซี บนระบบปฏิบัติการลินุกซ์ พารามิเตอร์ที่ใช้ในการจำลองที่สำคัญมีดังต่อไปนี้คือ ช่องสื่อสารเป็นแบบ AWGN (additive white Gaussian noise) อัตราการเข้ารหัสคอนโวลูชันเป็น 1/2 ค่าความยาวคอนสเตรนท์ของตัวเข้ารหัสมี 2 ค่า คือ $m = 5$ โดยมีเมตริกซ์ของตัวกำเนิด (generator matrices) $G_1 = 110101$ $G_2 = 101111$ และ $m = 7$ ซึ่งใช้ generator matrices $G_1 = 10111101$ และ $G_2 = 11000011$ จำนวนบิตต่อเฟรม K เป็น 60 บิต 120 บิตและ 180 บิตซึ่งมีจำนวนบิตทดสอบรวมเป็น 1,200,000 บิตเท่ากันในทุกกรณี $C_{\text{lim}} = 2,500$ จำนวนโนดที่ย้ายระหว่างสแตคหลักและสแตครองภายในแต่ละตัวประมวลผล T เป็น 2 ขนาดของสแตครอง Z เป็น 12 จำนวนสแตครองของระบบตัวประมวลเดียว n_Z เป็น 300 และของตัวประมวลผลหลายตัว n_{ZP} เป็นไปตามสมการที่ (2) รูปที่ 2 แสดงค่าอัตราความผิดพลาดบิต (bit error rate, BER) เทียบกับจำนวนโนดที่ย้ายจาก สแตคหลักของ DEC #1 ไปยังสแตคหลักของ DEC # j ($(j = 2,\ldots,n)$) เมื่อกำหนดค่า $E_b / N_o = 4.5$ dB จะเห็นว่าค่า T' ที่ให้ค่า BER ต่ำที่สุดจะมากขึ้นตามค่าของ Z_1 (สังเกตจากกราฟแต่ละเส้นที่ให้ค่า BER ต่ำสุด) และเมื่อกำหนดให้ขนาดของสแตคหลักคงที่ที่ค่า $Z_1 = 150$ โดยปรับค่า E_b / N_o จาก 4.5-5.5 dB ให้ค่า BER เมื่อใช้ตัวประมวลผล 2 ตัวดังแสดงใน

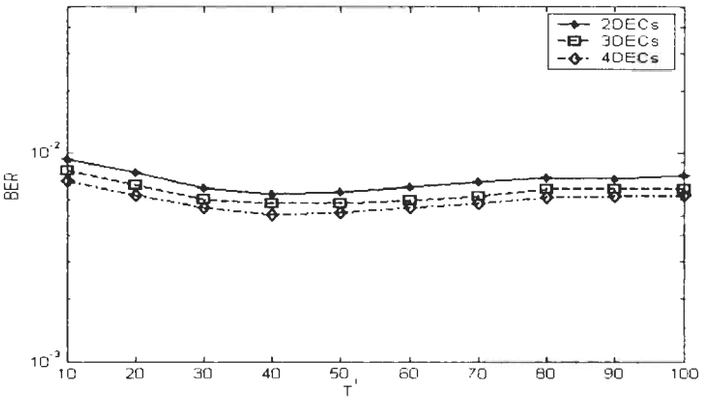
รูปที่ 3 สำหรับรูปที่ 4 เป็นผลการจำลองเพื่อดูค่า BER เมื่อใช้ตัวประมวลผล 2 ตัว 3 ตัว และ 4 ตัว โดยกำหนดให้ $Z_1 = 150$ คงที่เช่นเดิม รูปที่ 5 แสดงผลอัตราความผิดพลาดบิตที่ค่า m และ K ต่างๆ กันจะเห็นว่าเมื่อใช้จำนวน m และ K มากขึ้นค่า T' ที่ทำให้ BER ต่ำที่สุดมีค่าลดลง จากผลการทดสอบข้างต้นจะเห็นว่าที่ค่า $Z_1 = 150$ ค่า T' ที่เหมาะสมคือค่าตั้งแต่ 40 ถึง 50 ดังนั้น จึงได้เลือกค่า $Z_1 = 150$ สำหรับ $m = 5$ และ $T' = 50$ เพื่อใช้ทดสอบประสิทธิภาพในบทความนี้



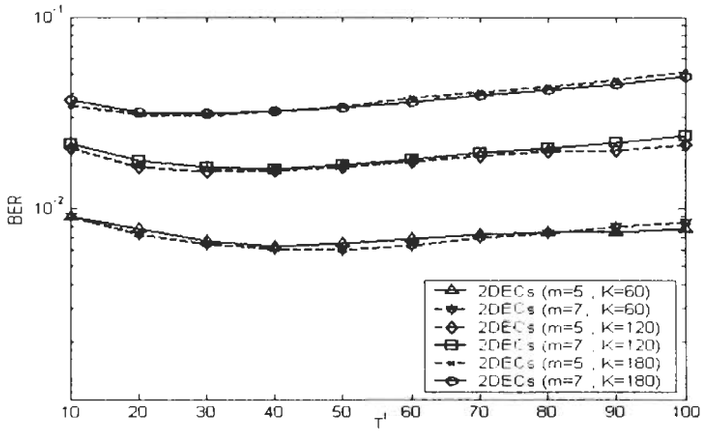
รูปที่ 2 แสดงค่า BER กับ T' เมื่อเปลี่ยนจำนวน Z_1



รูปที่ 3 แสดงค่า BER กับ T' ที่ค่า E_b / N_o ต่าง ๆกัน



รูปที่ 4 แสดงค่า BER กับ T' เมื่อใช้จำนวนตัวประมวลผลต่างกัน



รูปที่ 5 แสดงค่า BER กับ T' เมื่อใช้ ค่าของ m และ K ต่างกัน

ตารางที่ 1 แสดงผลจำนวนบล็อกของการถอดรหัสซึ่งสิ้นสุดภายในสแตคหลักของตัวประมวลผลตัวแรก การเพิ่ม T' จะทำให้ได้จำนวนบล็อกของการถอดรหัสที่สิ้นสุดในสแตคหลักของ DEC#1 เพิ่มมากขึ้นด้วย แต่เส้นทางเดินของการถอดรหัสที่การถอดรหัสที่สิ้นสุดภายในสแตคหลักของ DEC#1 (หลังการย้าย T') นั้นอาจจะไม่ได้เป็นเส้นทางเดินของการถอดรหัสที่มีค่าเมตริกซ์ที่ดีที่สุด

ตารางที่ 1

แสดงจำนวนบล็อกที่ถอดรหัสเสร็จในสแตคหลักของ DEC# 1

จำนวนตัวประมวลผล	จำนวนบล็อกที่ถอดรหัสเสร็จในสแตคหลักของ DEC #1					
	$T' = 20$	$T' = 30$	$T' = 40$	$T' = 50$	$T' = 60$	$T' = 70$
1 DEC	18,507*					
2 DECs	18,548	18,578	18,582	18,639	18,682	18,745
3 DECs	18,569	18,635	18,692	18,763	18,832	18,903
4 DECs	18,594	18,691	18,790	18,895	18,915	19,045

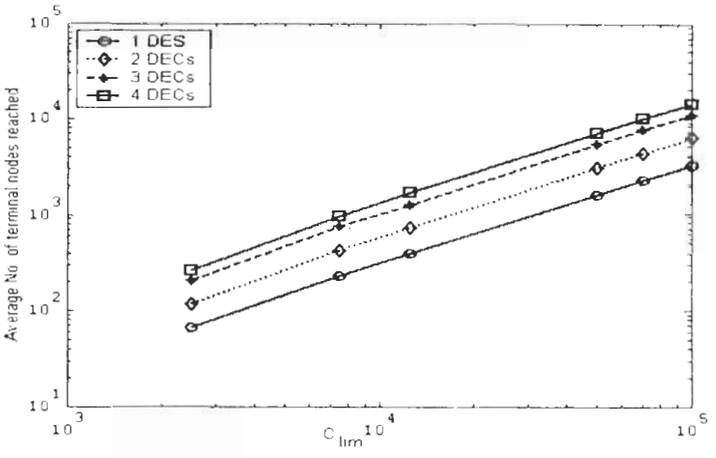
* ตัวประมวลผลตัวเดียวไม่มีการย้ายโนดข้อมูลระหว่างตัวถอดรหัส

รูปที่ 6 แสดงจำนวนครั้งเฉลี่ยที่กระบวนการถอดรหัสถึงโนดสุดท้ายในแผนภูมิต้นไม้แทนรหัสเทียบกับค่า $C_{\lim}$ จะเห็นว่า การประยุกต์ใช้วิธีการถอดรหัสแบบขนานกับ MSA ให้จำนวนครั้งที่ถึงโนดสุดท้ายได้มากกว่า MSA ที่ใช้ตัวประมวลผลเดียว ที่ค่า $C_{\lim}$ เท่ากันและนั่นหมายถึงโอกาสที่จะได้ทางเดินของการถอดรหัสที่ดีกว่าในระบบที่ใช้ตัวประมวลผลหลายตัว อย่างไรก็ตาม จำนวนครั้งของการถึงโนดสุดท้ายในรูปที่ 6 ไม่ได้หมายถึงจำนวนครั้งของการบันทึกค่าใหม่ของ TD เสมอไป รูปที่ 7 ได้แสดงให้เห็นถึงจำนวนครั้งเฉลี่ยที่มีการบันทึกค่าทางเดินใหม่ใน TD ซึ่งผลที่ได้สอดคล้องกับรูปที่ 6 นั่นคือ ระบบการถอดรหัสแบบขนานสำหรับ MSA ให้ค่าเฉลี่ยของจำนวนครั้งในการบันทึกค่า TD ใหม่ที่สูงขึ้นและจะสูงกว่าเดิมเมื่อเพิ่มจำนวนตัวประมวลผลในการถอดรหัสมากขึ้น

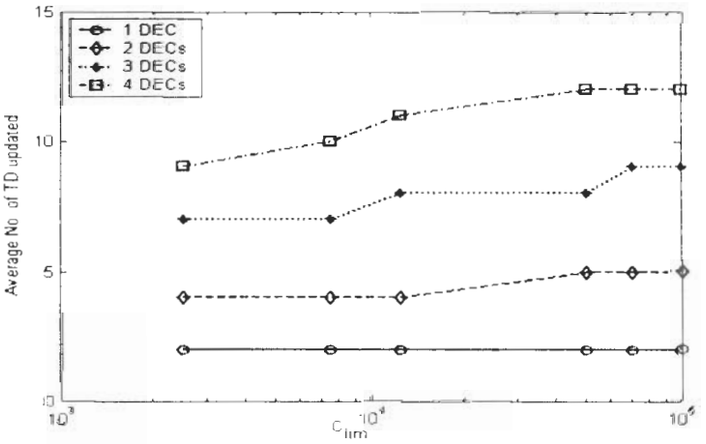
รูปที่ 8 เป็นการทดสอบประสิทธิภาพของ BER เมื่อเปลี่ยนค่า E_b / N_o ในช่วง 4.5-7.0 dB จะเห็นว่าค่า BER ในการถอดรหัสเมื่อใช้วิธีการถอดรหัสแบบขนานสำหรับ MSA มีค่าต่ำกว่า BER ของระบบ MSA เดิมซึ่งใช้ตัวประมวลผลตัวเดียว อย่างไรก็ตาม เมื่อเพิ่มจำนวนตัวประมวลผลเป็น 3 และ 4 ตัว ไม่ช่วยให้ประสิทธิภาพในอัตราความผิดพลาดบิตของการถอดรหัสดีขึ้นจากเดิมมากนักในขณะที่ขนาดของ Z_1 ใหญ่มาก

รูปที่ 9 เป็นการทดสอบประสิทธิภาพของ BER ที่ค่า T' ต่าง ๆ เมื่อทดสอบในขณะที่ขนาดของสแตคหลัก $Z_1 = 75$ และใช้ $m = 7$, $K = 120$ จะเห็นว่าค่า T' ที่เหมาะสมคือ $T' = 20$ ซึ่งให้ค่า BER ต่ำที่สุดในทุกกรณีของการถอดรหัสแบบขนานเมื่อใช้จำนวนตัวประมวลผลต่างกัน และเมื่อทดสอบที่ค่า E_b / N_o ต่างๆ กัน ดังรูปที่ 10 ผลที่ได้แสดงให้เห็นว่าเมื่อเพิ่มค่า m และ K ในขณะที่ลดขนาดของ Z_1 ลงจาก 150 เป็น 75 ตัวประมวลผลที่เพิ่มขึ้นสามารถปรับปรุงประสิทธิภาพของการถอดรหัสในอัตราความผิด

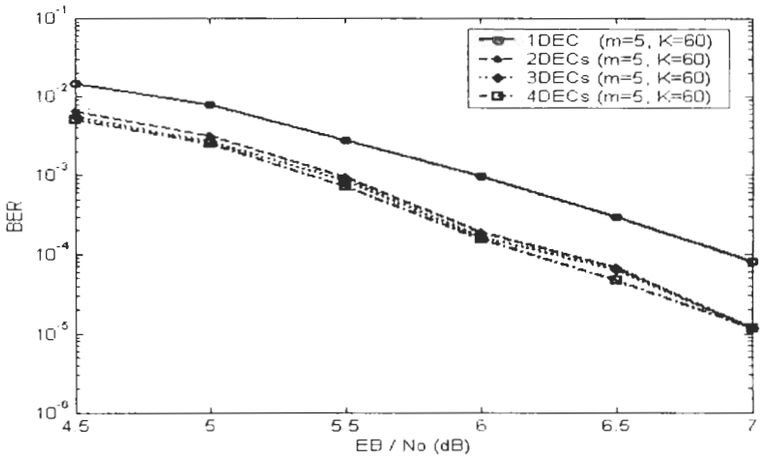
พลาดบิตได้เพิ่มขึ้นเล็กน้อยเมื่อเทียบกับรูปที่ 8



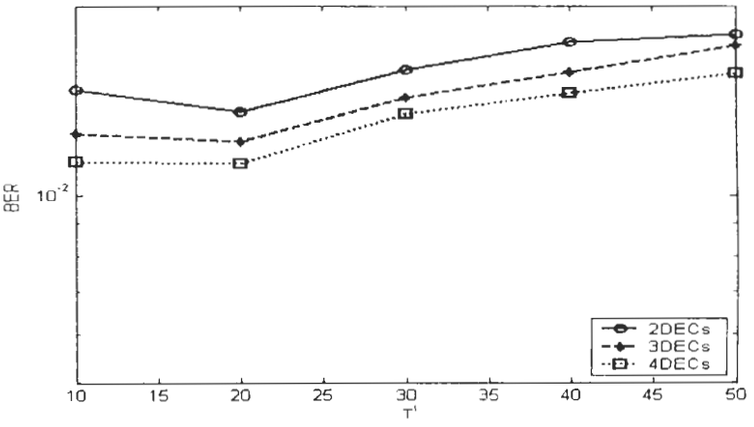
รูปที่ 6 แสดงจำนวนครั้งเฉลี่ยที่การถอดรหัสถึงโนดสุดท้ายในแผนภูมิต้นไม้แทนรหัส



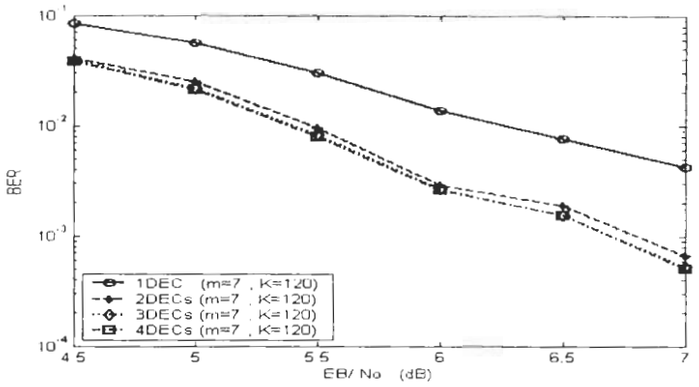
รูปที่ 7 แสดงจำนวนครั้งเฉลี่ยที่มีการบันทึกค่า TD ใหม่



รูปที่ 8 แสดงค่า BER ที่ค่า E_b / N_o **ต่าง ๆ (** $m = 5$, $K = 60$, $Z_1 = 150$, $T' = 50$)



รูปที่ 9 แสดงค่า BER กับ T' **เมื่อใช้ ค่าของ** $m = 7$ **และ** $K = 120$ $Z_1 = 75$



รูปที่ 10 แสดงค่า BER ที่ค่า E_b / N_o **ต่าง ๆ (**$m = 7$, $K = 120$, $Z_1 = 75$, $T = 20$ **)**

สรุปผล

งานวิจัยนี้ได้นำเสนอการประยุกต์ใช้วิธีการถอดรหัสแบบขนานโดยใช้ตัวประมวลผลหลายตัวกับอัลกอริทึมการถอดรหัสแบบ MSA สำหรับรหัสคอนโวลูชัน เพื่อปรับปรุงประสิทธิภาพในอัตราความผิดพลาดบิตของการถอดรหัสแบบ MSA เดิมให้ดีขึ้น ซึ่งปัญหาในการถอดรหัสแบบ MSA เดิมนั้น คือ บล็อกที่มีปริมาณของสัญญาณรบกวนมากไม่สามารถสิ้นสุดกระบวนการถอดรหัสในสแตคหลักได้และเมื่อทำการย้ายโนดบางส่วนในสแตคหลักมาทำการถอดรหัสต่อในสแตครอง โอกาสที่การถอดรหัสจะสามารถกลับมาสิ้นสุดที่แสตคหลักนั้นมีค่าน้อยมาก ดังนั้น การถอดรหัสมักจะสิ้นสุดโดยค่าของ $C_{\lim}$ ซึ่งให้ค่าอัตราความผิดพลาดบิตสูง ในงานวิจัยนี้ได้ทำการประยุกต์การถอดรหัสแบบขนานกับ MSA เพื่อปรับปรุงประสิทธิภาพในอัตราความผิดพลาดบิตของ MSA เดิมให้ดีขึ้น ที่ค่า $C_{\lim}$ และปริมาณหน่วยความจำที่เท่ากัน ผลการจำลองด้วยคอมพิวเตอร์ปรากฏว่าการถอดรหัสแบบขนานโดยใช้ตัวประมวลผล 2 ตัวสามารถทำให้ค่าอัตราความผิดพลาดบิตใน MSA ลดลงได้อย่างมาก อย่างไรก็ตาม เมื่อเพิ่มจำนวนตัวประมวลผลมากกว่า 2 ตัวประมวลผล อัตราความผิดพลาดบิตลดลงไม่มากนัก ดังนั้น โครงสร้างของการถอดรหัสแบบขนานที่นำเสนอในงานวิจัยนี้มีความเหมาะสมสำหรับตัวประมวลผล 2 ตัว มากที่สุด ซึ่งผลงานวิจัยนี้เป็นจุดเริ่มต้นที่น่าสนใจสำหรับการศึกษารูปแบบโครงสร้างของการถอดรหัสแบบขนานโดยใช้ตัวประมวลที่มากกว่า 2 ตัวเพื่อเพิ่มประสิทธิภาพของการถอดรหัสแบบ MSA ให้ดียิ่งขึ้น

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยขอนแก่น ที่ให้การสนับสนุนเงินทุนเพื่อทำการวิจัยนี้

เอกสารอ้างอิง

Chivillat, P.R., and Costello, Jr., D.J.1976. A Multiple Stack Algorithm for Erasurefree Decoding of Convolutional Codes, **IEEE Transactions on Commun.,** COM-25(12), pp. 1460-1470.

Forney, G.D. 1974. Convolutional Codes III: Sequential Decoding. **Inf. Control**, Vol. 25, pp. 267-297.

Haccoun, D., and Begin, G. 1989. High-rate punctured convolutional codes for Viterbi and sequential decoding. **IEEE Trans. Commun**. Vol. 37 No. 11, pp. 1113-1125

Jelinek, F. 1969 A fast sequential decoding using a stack, IBM J. Res. Dev., Vol.13, pp. 675-685.

Lin, S., and Costello, D.J. Jr. 1983. Error Control Coding: Fundamentals and Applications. **Prentice-Hall**

Lin, Y., and Tu, S.H. 1997. Modified multiple stack algorithm for decoding convolutional codes. **IEE Pro.-Commun**, Vol. 44, No. 4, August, pp. 221-228.