ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

ปริญญา

วิศวกรรมไฟฟ้า วิศวกรรมไฟฟ้า

สาขา **ภาควิชา**

เรื่อง วิธีการแบ่งนับรหัสลายเซ็นที่ใช้ความซับซ้อนในการค้นหาน้อยสำหรับระบบซีดีเอ็มเอ

Low Search Complexity Signature Quantization Schemes for CDMA

นามผู้วิจัย นายกฤษฎา มามาตร

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

(.....)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม .....

(.....)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม .....

(.....)

หัวหน้าภาควิชา .....

(.....)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

.....

(.....)

คณบดีบัณฑิตวิทยาลัย

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

วิทยานิพนธ์

เรื่อง

วิธีการแบ่งนับรหัสลายเซ็นที่ใช้ความซับซ้อนในการค้นหาน้อยสำหรับระบบซีดีเอ็มเอ

Low Search Complexity Signature Quantization Schemes for CDMA

โดย

นายกฤษฎา มามาตร

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2553

กฤษฎา มามาตร 2553: วิธีการแบ่งนับรหัสลายเซ็นที่ใช้ความซับซ้อนในการค้นหาน้อย สำหรับระบบซีดีเอ็มเอ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์วิรุณศักดิ์ สันติเพ็ชร์, Ph.D. 75 หน้า

ซีดีเอ็มเอเป็นระบบสื่อสารไร้สายในยุคที่ 3 งานวิจัยนี้สนใจการสื่อสารระหว่างโทรศัพท์เคลื่อนที่เป็นฝั่งส่งและสถานีฐานเป็นฝั่งรับ โดยโทรศัพท์เคลื่อนที่หรือผู้ใช้แต่ละรายใช้รหัสลายเซ็นที่แตกต่างกันสำหรับติดต่อกับสถานีฐาน การเพิ่มสมรรถนะของผู้ใช้แต่ละรายหรือระบบโดยรวมผู้ใช้แต่ละรายจำเป็นต้องปรับรหัสลายเซ็นให้เหมาะสมกับช่องสัญญาณที่เปลี่ยนแปลงตามเวลา กำหนดให้สถานีฐานสามารถประมาณช่องสัญญาณและคำนวณรหัสลายเซ็นที่เหมาะสมได้ การส่งข้อมูลของรหัสลายเซ็นจากสถานีฐานกลับไปยังโทรศัพท์เคลื่อนที่นั้นจำเป็นต้องมีการแบ่งนับเนื่องจากช่องสัญญาณป้อนกลับมีอัตราบิตที่จำกัด

วิทยานิพนธ์นี้นำเสนอวิธีการแบ่งนับรหัสลายเซ็นที่มีความซับซ้อนในการค้นหาน้อยสามวิธีได้แก่การแบ่งนับแบบโครงสร้างต้นไม้ การแบ่งนับพีเอเอ็มและการแบ่งนับแบบต้นไม้เคดีโดยเปรียบเทียบวิธีทั้งสามกับวิธีการแบ่งนับเวกเตอร์สุ่มจากงานวิจัยที่ผ่านมาซึ่งใช้ความซับซ้อนในการค้นหามากและเพิ่มขึ้นอย่างเอกซ์โพเนนเชียลกับบิตป้อนกลับ เริ่มต้นจากการนำวิธีการแบ่งนับเวกเตอร์ชนิดต่างๆมาใช้ในระบบซีดีเอ็มเอสำหรับผู้ใช้รายเดียวหลังจากนั้นจึงขยายผลไปยังระบบซีดีเอ็มเอสำหรับผู้ใช้แบบกลุ่ม ผลการจำลองระบบแสดงให้เห็นว่าการแบ่งนับเวกเตอร์แต่ละชนิดใช้ความซับซ้อนในการค้นหาที่เพิ่มขึ้นอย่างเชิงเส้นกับบิตป้อนกลับซึ่งมีแนวโน้มไปในทางเดียวกันทั้งระบบสำหรับผู้ใช้รายเดียวและผู้ใช้แบบกลุ่ม จากผลการจำลองระบบพบว่าการแบ่งนับแบบต้นไม้เคดีมีความเหมาะสมที่สุดสำหรับการแบ่งนับรหัสลายเซ็นเมื่อพิจารณาทั้งสมรรถนะและจำนวนการคำนวณที่ต้องการ โดยที่อัตราส่วนของบิตป้อนกลับต่ออัตราขยายประมวลผลเท่ากับ 1 การแบ่งนับแบบต้นไม้เคดีให้สมรรถนะที่ใกล้เคียงกับการแบ่งนับเวกเตอร์สุ่มอย่างมากและในบางตัวอย่างจำนวนของผลคูณภายในของต้นไม้เคดีลดลงจาก 10^5 เป็น 10^2 เมื่อเปรียบเทียบกับการแบ่งนับเวกเตอร์สุ่ม

_____ _____ ___ / ___ / ___

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Kritsada Mamat 2010: Low Search Complexity Signature Quantization Schemes for CDMA. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Departmant of Electrical Engineering. Thesis Advisor: Assistant Professor Wiroonsak Santipach, Ph.D. 75 pages.

This work considers a reverse-link code division multiple access (CDMA) where mobile is a transmitter and base station is a receiver. In CDMA, mobiles or user use different signature code to communicate with base station. To increase system or user performance, user need to adapt its signature to a time-varying channel. With channel information, a receiver can compute the optimal signature for a desired user. To relay the signature code from base station to the user, the signature need to be quantized due to limited-rate feedback channel.

We propose three signature quantization schemes with low complexity, namely, a tree-structured random vector quantization, PAM (Pulse Amplitude Modulation) and k-dimensional (kd) tree. The proposed schemes are compared with a RVQ scheme, which use exhaustive search to locate optimal signature. We first consider single-user signature quantization and then extend to multi-user one. From numerical results shown, kd-tree among the proposed schemes give the best performance with less complexity. The search complexity of all 3 schemes increases linearly with number of feedback bits for both single-user and multi-user cases. At 1 feedback bit per processing gain, kd-tree perform almost the same as RVQ does. In the example shown number of inner product computations used in kd-tree scheme is 3 order of magnitude less than that used in RVQ.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ผศ.ดร.วิรุณศักดิ์ สันติเพ็ชร์ ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ อ.ดร.พูนลาภ ลามศรีจันทร์ และ ผศ.ดร.อุศนา ตัณฑุลเวศม์ กรรมที่ปรึกษาร่วม อ.ดร.ชัยยศ พิรักษ์ อาจารย์จากบัณฑิตวิทยาลัยวิศวกรรมศาสตร์นานาชาติสิรินธร ไทย-เยอรมัน สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ให้คำปรึกษาในการเรียนการค้นคว้าวิจัยตลอดจนการตรวจแก้ไขวิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์

ขอกราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ได้อบรมสั่งสอนและมอบความรู้อันเป็นประโยชน์อย่างยิ่งในการนำไปใช้ประโยชน์ต่อไป ขอขอบคุณสถาบันวิจัยและพัฒนาอุตสาหกรรมโทรคมนาคม (TRIDI) ที่ได้ให้ทุนสนับสนุนงานวิจัยในครั้งนี้

ด้วยความดีหรือประโยชน์ใดอันเนื่องจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่คุณพ่อ คุณแม่ ที่ได้อบรมและให้กำลังใจผู้วิจัยมาตลอดในทุกเรื่อง และหากวิทยานิพนธ์เล่มนี้มีข้อบกพร่องประการใด ข้าพเจ้ายินดีรับข้อเสนอแนะและขออภัยมา ณ ที่นี้ด้วย

กฤษฎา มามาตร

มกราคม 2553

สารบัญ

หน้า

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

คำอธิบายสัญลักษณ์และคำย่อ

CDMA	=	Code Division Multiple Access
TDMA	=	Time Division Multiple Access
FDMA	=	Frequency Division Multiple Access
AWGN	=	Additive White Gaussian Noise
SINR	=	Signal-to-Interference plus Noise Ratio
RVQ	=	Random Vector Quantization
TS-RVQ	=	Tree-Structured Random Vector Quantization
PAM	=	Pulse Amplitude Modulation
GLRT	=	Generalized Likelihood Ratio Test
SVD	=	Singular Value Decomposition

วิธีการแบ่งนับรหัสลายเซ็นที่ใช้ความซับซ้อนในการค้นหาน้อย สำหรับระบบซีดีเอ็มเอ

Low Search Complexity Signature Quantization Schemes for CDMA

คำนำ

มาตรฐานโทรศัพท์เคลื่อนที่ยุคที่ 3 (Third Generation Mobile Network หรือ 3G) เป็นระบบที่รองรับอัตราการส่งได้มากกว่ามาตรฐานโทรศัพท์เคลื่อนที่ยุคที่ 2 หรือ 2G ที่ใช้อยู่ในปัจจุบัน ซึ่งใน 3G นั้นใช้เทคโนโลยีที่เรียกว่าซีดีเอ็มเอ (Code Division Multiple Access หรือ CDMA) ในการเข้าใช้งานของผู้ใช้แต่ละราย ในระบบ CDMA นั้นใช้แถบความถี่กว้างขึ้นและผู้ใช้แต่ละรายถูกแบ่งแยกโดยรหัสลายเซ็นซึ่งทำให้ระบบรองรับผู้ใช้ได้จำนวนมากและมีความปลอดภัยมากขึ้น

การที่สถานีฐานหรือโทรศัพท์เคลื่อนที่สามารถปรับรหัสลายเซ็นให้เหมาะสมและหลีกเลี่ยงการแทรกสอดจากผู้ใช้รายอื่นทำให้ระบบมีสมรรถนะที่ดีขึ้นและรองรับจำนวนผู้ใช้ได้มากขึ้น โดยได้มีการเปรียบเทียบสมรรถนะของระบบที่มีการปรับและไม่ปรับรหัสลายเซ็นในงานวิจัยที่ผ่านมาพบว่าการปรับรหัสลายเซ็นช่วยเพิ่มสมรรถนะของระบบได้จริง ในการปรับรหัสลายเซ็นนั้นทำได้โดยที่ฝั่งรับซึ่งเมื่อได้รับสัญญาณจากฝั่งส่งแล้วสามารถประมาณช่องสัญญาณได้ หลังจากนั้นฝั่งรับก็จะคำนวณรหัสลายเซ็นที่เหมาะสมและส่งข้อมูลของรหัสลายเซ็นไปยังฝั่งส่งผ่านทางช่องสัญญาณป้อนกลับ ในทางปฏิบัติช่องสัญญาณป้อนกลับนี้มีอัตราการส่งที่จำกัด ดังนั้นรหัสลายเซ็นต้องมีการแบ่งนับก่อนส่งกลับไปยังฝั่งส่ง ซึ่งการแบ่งนับสามารถทำได้โดยที่ฝั่งรับเลือกเวกเตอร์ในกลุ่มของเวกเตอร์หรือโค้ดบุ๊คที่ทำให้อัตราส่วนของสัญญาณต่อสัญญาณแทรกสอดและสัญญาณรบกวน (Signal-to-Interference plus Noise Ratio หรือ SINR) มีค่ามากที่สุดซึ่ง SINR นี้เป็นดรรชนีชี้วัดสมรรถนะของระบบ การแบ่งนับรหัสลายเซ็นส่วนมากในงานวิจัยที่ผ่านมานั้นต้องการความซับซ้อนในการค้นหารหัสลายเซ็นที่เหมาะสมเป็นอย่างมากซึ่งจะเกิดปัญหาเมื่อรหัสลายเซ็นในโค้ดบุ๊คมีจำนวนมาก

งานวิจัยนี้นำเสนอวิธีการแบ่งนับรหัสลายเซ็นที่มีความซับซ้อนในการค้นหาน้อยซึ่งพิจารณาระบบซีดีเอ็มเอที่มีการปรับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวและระบบที่ปรับลายเซ็นสำหรับผู้ใช้ทั้งกลุ่ม โดยแสดงให้เห็นถึงข้อดีข้อด้อยของแต่ละวิธีโดยการนำเสนอในรูปแบบของสมรรถนะและความซับซ้อนในการคำนวณของแต่ละวิธี

วัตถุประสงค์

1. เพื่อนำเสนอวิธีการแบ่งนับรหัสลายเซ็นที่ใช้ความซับซ้อนในการค้นหาน้อย

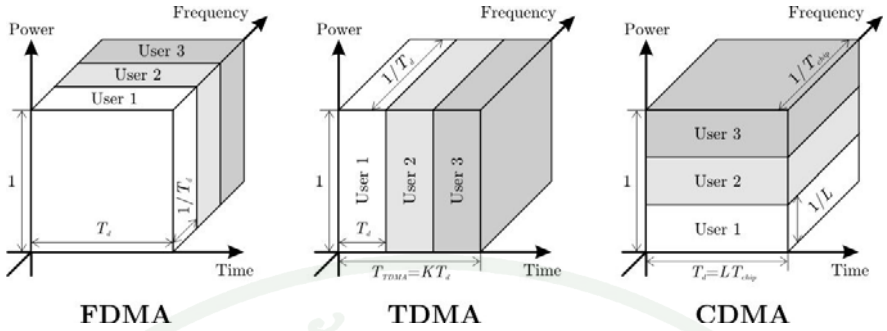
2. เพื่อเปรียบเทียบสมรรถนะของวิธีการแบ่งนับรหัสลายเซ็นและการสร้างโค้ดบุ๊คแบบต่างๆที่ใช้ความซับซ้อนในการค้นหาน้อย

การตรวจเอกสาร

1.เทคโนโลยีซีดีเอ็มเอ

1.1 แนะนำเทคโนโลยีซีดีเอ็มเอ

CDMA (Code Division Multiple Access) เป็นเทคโนโลยีพื้นฐานสำหรับการพัฒนาเทคโนโลยีระบบโทรศัพท์เคลื่อนที่ยุคที่ 3 เพื่อรองรับความต้องการใช้งานระบบสื่อสารไร้สายที่เพิ่มขึ้นอย่างมากในปัจจุบันเช่นการสื่อสารทางเสียงที่มีจำนวนผู้ใช้ในระบบเพิ่มมากขึ้น, การส่งข้อมูลคอมพิวเตอร์ปริมาณมากในอัตราบิตสูง ความต้องการดังกล่าวนี้ระบบโทรศัพท์เคลื่อนที่เซลลูลาร์ยุคที่ 2 ที่ใช้อยู่ในปัจจุบันยังมีข้อจำกัดในการให้บริการ CDMA เป็นเทคนิคการแบ่งเข้าใช้ช่องสัญญาณหลายทางชนิดหนึ่งที่พัฒนาต่อเนื่องมาจาก FDMA (Frequency Division Multiple Access) เทคนิคการเข้าใช้ช่องสัญญาณหลายทางแบบแบ่งความถี่ในระบบโทรศัพท์เคลื่อนที่ยุคที่ 1 ซึ่งวิธีนี้แต่ละช่องสัญญาณจะผลัดเปลี่ยนการใช้งานที่ความถี่ไม่ตรงกันเพื่อรับส่งสัญญาณแบบแอนะล็อกและ TDMA (Time Division Multiple Access) เทคนิคการเข้าใช้ช่องสัญญาณหลายทางแบบแบ่งเวลาในระบบโทรศัพท์เคลื่อนที่ยุคที่ 2 วิธีนี้แต่ละช่องสัญญาณจะผลัดเปลี่ยนกันใช้ความกว้างแถบที่มีช่วงคาบเวลาไม่ตรงกันในการรับส่งสัญญาณแบบดิจิตอล การเข้าใช้ช่องสัญญาณหลายทางแบบแบ่งรหัส CDMA จะอนุญาตให้ผู้ใช้หลายรายเข้าใช้สเปกตรัมความถี่ของระบบพร้อมกันและในเวลาเดียวกันได้โดยผู้ใช้จะใช้รหัสลายเซ็น (signature code) หรือรหัสแผ่ (spreading code) ที่แตกต่างกัน ทำให้สามารถรับสัญญาณและใช้คุณลักษณะของรหัสที่แตกต่างกันของผู้ใช้ในการถอดรหัสเพื่อให้ได้ข้อมูลเดิมของผู้ใช้รายที่ต้องการกลับมาโดยความกว้างแถบความถี่ของสัญญาณรหัสดังกล่าวมีค่ามากกว่าความกว้างแถบความถี่ของสัญญาณข้อมูลมากจึงมีลักษณะทำให้เกิดการแผ่ของสเปกตรัมของสัญญาณข้อมูลด้วยรหัสดังกล่าว ทั้งนี้เทคนิคที่ใช้ในการเข้าใช้ช่องสัญญาณนี้มีชื่อเรียกว่าการมอดูเลตแบบสเปรดสเปกตรัม (spread spectrum)



ภาพที่ 1 เทคนิคการเข้าใช้ช่องสัญญาณด้วยวิธีการร่วมใช้ช่องสัญญาณแบบแบ่งความถี่ (FDMA) การร่วมใช้ช่องสัญญาณแบบแบ่งเวลา (TDMA) และการร่วมใช้ช่องสัญญาณแบบเข้ารหัส (CDMA)

ที่มา: Hass (2004)

เทคโนโลยี CDMA มีข้อดีสามารถสรุปได้ดังนี้

- ความสามารถในการรองรับปริมาณผู้ใช้งานที่มากกว่า (flexible capacity)
- การส่งผ่านสัญญาณที่ราบรื่นลดปัญหาสายหลุด (soft hand-off)
- ความคมชัด และคุณภาพของเสียงในการติดต่อสื่อสาร (rake receiver)
- ลดความสิ้นเปลืองพลังงานจากแบตเตอรี่(power control)
- ความปลอดภัยของสัญญาณออกอากาศ

1.2 การมอดูเลตแบบสเปรดสเปกตรัมและรหัสลายเซ็น

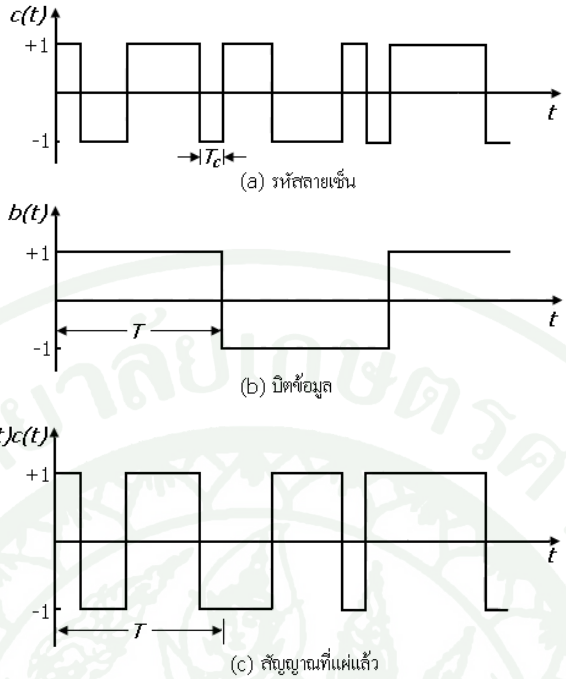
เทคนิคการมอดูเลตแบบสเปรดสเปกตรัมได้พัฒนาขึ้นครั้งแรกเพื่อใช้งานทางด้านการทหารเนื่องจากเทคนิคการมอดูเลตดังกล่าวสามารถป้องกันการส่งสัญญาณเพื่อรบกวนระบบการสื่อสาร (jamming) ได้ดีและเกิดโอกาสน้อยในการถูกดักจับสัญญาณด้วยเหตุผลดังกล่าวจึงได้รับการพัฒนาและถูกนำมาใช้งานในเชิงพาณิชย์ สำหรับระบบโทรศัพท์เคลื่อนที่วิธีการมอดูเลตแบบสเปรดสเปกตรัมจะทำการเข้ารหัสสัญญาณข่าวสารด้วยสัญญาณรหัสแผ่ที่มีคุณสมบัติมีความเป็นอิสระกับรหัสแผ่ของผู้ใช้รายอื่นและมีความกว้างของสเปกตรัมมากกว่าสัญญาณข้อมูลมาก ส่งผลทำให้กำลังของสัญญาณข้อมูลมีการแผ่ออกตลอดช่วงของความกว้างแถบของสัญญาณรหัสแผ่นั้นและทำให้สัญญาณสเปกตรัมแผ่มีความหนาแน่นกำลังงาน (Power Spectral Density หรือ PSD) ลดลงซึ่งเป็นสาเหตุให้การรบกวนสัญญาณและการดักจับสัญญาณในการทหารเพื่อทราบ

ข้อมูลการสื่อสารของฝ่ายตรงข้ามเป็นไปได้ยาก การร่วมใช้ช่องสัญญาณแบบเข้ารหัสแบ่งตามวิธีการมอดูเลตได้เป็นสองประเภทหลักคือ วิธีไดเร็กซีเควนซ์ (DS-CDMA) และวิธีฟรีเควนซ์ซีฮอปปิง (FH-CDMA)

การมอดูเลตแบบสเปรดสเปกตรัมแบบไดเรกซีเควนซ์สามารถกระทำได้โดยนำรหัสลายเซ็นไปคูณกับบิตข้อมูลที่มีความยาวเท่ากับ T วินาทีทีละบิตจากภาพที่ 2 จะเห็นว่ารหัสลายเซ็นมีความกว้างของชิป (chip) เท่ากับ T_c วินาทีซึ่งมีขนาดเล็กกว่าความกว้างของบิตข้อมูลมาก ผลที่ได้หลังจากการทำสเปรดสเปกตรัมเป็นลำดับสัญญาณที่อัตราการเปลี่ยนแปลงที่สูงมากขึ้นและโดยทั่วไปอัตราบิตหลังจากการทำสเปรดสเปกตรัมแล้วจะเรียกว่าอัตราชิป (chip rate) ในภาพที่ 2 นี้อัตราชิปมีขนาดเพิ่มขึ้น 7 เท่าตัวเมื่อเทียบกับอัตราบิตของข้อมูลตั้งต้น ดังนั้นการส่งสัญญาณดังกล่าวจึงต้องการแบนด์วิดท์ที่เพิ่มมากขึ้นกว่าเดิมด้วยตัวประกอบการคูณ N (ลัญฉกร และ คณะ, 2548)

$$N = \frac{T}{T_c} \qquad (1)$$

ทั้งนี้ N มีชื่อเรียกว่าตัวประกอบการแผ่ (spreading factor) หรืออัตราขยายประมวลผล (processing gain)



ภาพที่ 2 ตัวอย่างการทำสเปรดสเปกตรัมที่มีตัวประกอบการแผ่ $N = 7$

ที่มา: Proakis (2001)

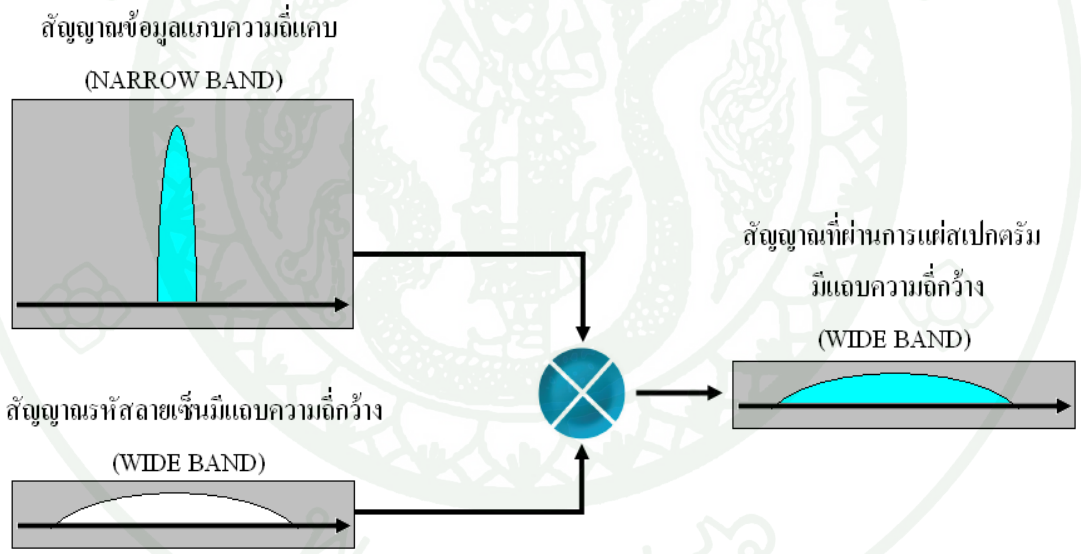
ในระบบไดเร็กซีเควนซ์ซีดีเอ็มเอรหัสลายเซ็นจะถูกนำมาใช้เพื่อแบ่งแยกผู้ใช้แต่ละรายออกจากกันรหัสลายเซ็นที่ดีจะมีคุณสมบัติตั้งฉากกัน (orthogonal) หรือมีค่าสหสัมพันธ์ข้าม (cross correlation) ระหว่างรหัสเป็นศูนย์ อย่างไรก็ตามรหัสลายเซ็นที่ดีนั้นมักออกแบบได้ยากอีกทั้งมีจำนวนชุดรหัสให้ใช้ได้จำกัดและอาจตั้งฉากกันอย่างไม่สมบูรณ์เนื่องจากสภาพแวดล้อมเช่นมีค่าประวิงเวลาที่ไม่เท่ากันผลของช่องสัญญาณทำให้สูญเสียความตั้งฉากตัวอย่างรหัสแผ่ที่นิยมใช้มีสองชนิดคือ

1. รหัสเชิงตั้งฉาก (Orthogonal Code) เป็นรหัสที่มีค่าสหสัมพันธ์ข้ามระหว่างรหัสเป็นศูนย์ ตัวอย่างรหัสชนิดนี้ ได้แก่ รหัสวอลท์ฮาดามาด (Hadamard Walsh Code) เป็นต้น

2. พีเอ็นซีเควนซ์ (Pseudorandom Noise-sequence หรือ PN-sequence) เป็นรหัสที่มีคุณสมบัติคล้ายกับสัญญาณรบกวนซึ่งเป็นกระบวนการแบบสุ่มเป็นรหัสที่ตั้งฉากกันอย่างไม่สมบูรณ์ ตัวอย่างรหัสชนิดนี้ได้แก่ เอ็มซีเควนซ์ (Maximal Length-Sequence หรือ M-Sequence) รหัสโกลด์ (Gold Code) และชุดรหัสคาซามิ (Kasami Sequence) เป็นต้น

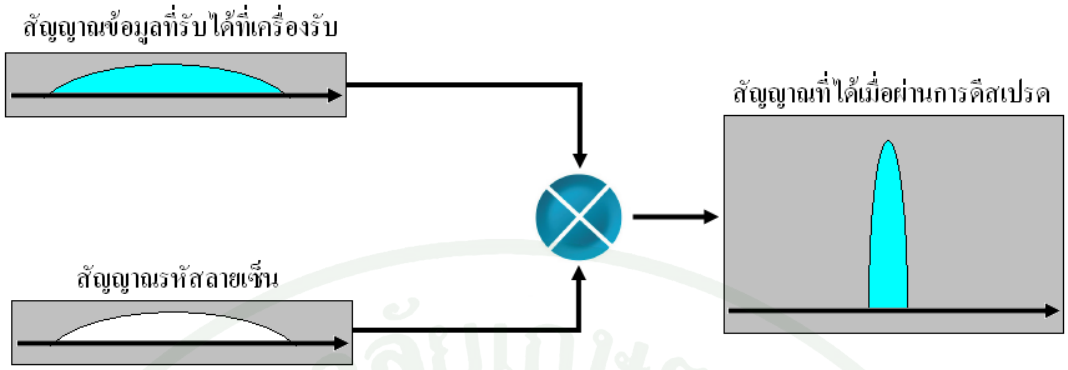
ขั้นตอนที่สำคัญในการสื่อสารข้อมูลในระบบไดเร็กซีเควนซ์ซีดีเอ็มเอคือการแผ่หรือการสเปรด (spreading) และการรวมกลับหรือการดีสเปรด (despreading) การสเปรดเป็นกระบวนการที่ผู้ใช้แต่ละคนใช้ในการส่งข้อมูลเข้าสู่ช่องสัญญาณไปยังผู้รับที่เครื่องรับปลายทาง และการดีสเปรดเป็นกระบวนการที่ใช้ที่เครื่องรับสำหรับประมาณบิตข้อมูลของผู้ใช้ที่สนใจการร่วมใช้ช่องสัญญาณด้วยวิธีการแผ่สเปกตรัมเป็นมาตรฐานของระบบโทรศัพท์เคลื่อนที่ซีดีเอ็มเอซึ่งมีขั้นตอนดังนี้

ขั้นตอนในการสร้างสัญญาณเบสแบนด์ของผู้ใช้แต่ละรายเริ่มต้นด้วยการนำข้อมูลที่ต้องการส่งที่อยู่ในรูปของสัญญาณดิจิตอลซึ่งมีอัตราบิตข้อมูลต่ำนำมาแผ่สเปกตรัมกับรหัสแผ่ซึ่งมีอัตราบิตข้อมูลที่สูงผลที่ได้คือสัญญาณที่ได้จะมีอัตราการส่งสูงกว่าข้อมูลที่ต้องการส่งมากดังภาพที่ 3



ภาพที่ 3 ขั้นตอนการแผ่ (spreading)

ขั้นตอนการแผ่กลับหรือดีสเปรด (despreading) เป็นขั้นตอนในเครื่องรับขั้นตอนนี้เริ่มต้นจากการรับสัญญาณข้อมูลในหนึ่งคาบเวลาแล้วนำมาคำนวณหาค่าสหสัมพันธ์ (correlation) ระหว่างสัญญาณดังกล่าวกับรหัสแผ่ของผู้ใช้ที่ต้องการจากนั้นหาค่าเฉลี่ยของสัญญาณในหนึ่งคาบเวลาของบิตข้อมูลที่ต้องการดังภาพที่ 4

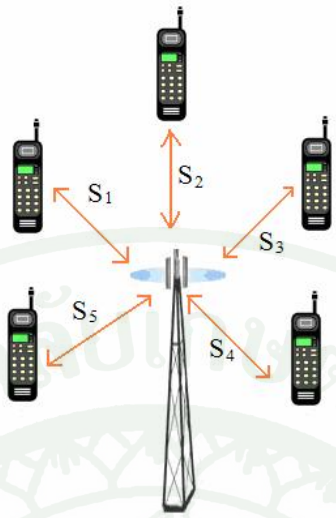


ภาพที่ 4 ขั้นตอนการแผ่กลับหรือดีสเปรด (despreading)

ไดเร็กซีเควนซ์ซีดีเอ็มเอมีลักษณะเด่นคือการสร้างสัญญาณที่ถูกเข้ารหัสทำได้ง่ายโดยใช้กระบวนการคูณธรรมดา วงจรสังเคราะห์ความถี่เป็นวงจรที่ไม่ซับซ้อนเนื่องจากใช้ความถี่พาห์เพียงความถี่เดียวสามารถดีมอดูเลตสัญญาณสเปกตรัมแผ่โดยใช้การดีมอดูเลตแบบร่วมนัยได้และไม่จำเป็นต้องมีการซิงโครไนซ์ระหว่างผู้ใช้ สำหรับข้อด้อยคือการเริ่มต้นและรักษาการซิงโครไนซ์ ระหว่างรหัสของสัญญาณที่รับได้กับสัญญาณรหัสที่สร้างขึ้นทางด้านรับทำได้ยากซึ่งกระบวนการซิงโครไนซ์นี้จะต้องเกิดขึ้นภายในช่วงเวลาของชิป การรับสัญญาณได้อย่างถูกต้องขบวนรหัสที่สร้างขึ้นจะต้องซิงโครไนซ์กับขบวนรหัสที่รับได้ภายในช่วงเวลาของชิปและกำลังงานที่รับได้จากผู้ใช้งานที่อยู่ใกล้สถานีฐานจะมีค่ามากกว่าที่รับได้จากผู้ใช้งานที่อยู่ไกลจากสถานีฐานส่งผลให้ผู้ใช้งานที่อยู่ใกล้เกิดการแทรกสอดสัญญาณของผู้ใช้ที่อยู่ไกลออกไปหรือที่เรียกว่าผลจากปรากฏการณ์ใกล้-ไกล (near-far effect)

2. แบบจำลองทางคณิตศาสตร์ของระบบซีดีเอ็มเอ

การศึกษาแบบจำลองทางคณิตศาสตร์ของระบบซีดีเอ็มเอเริ่มจากการพิจารณาระบบซีดีเอ็มเอดังภาพที่ 5



ภาพที่ 5 ระบบซีดีเอ็มเอที่โทรศัพท์เคลื่อนที่เข้าถึงสถานีฐานโดยการใช้รหัสลายเซ็นที่แตกต่างกัน

จากภาพที่ 5 กำหนดให้โทรศัพท์เคลื่อนที่แต่ละเครื่องติดต่อกับสถานีฐานเดียวกันโดยใช้รหัสลายเซ็นที่ต่างกัน ในกรณีทั่วไปกำหนดให้โทรศัพท์เคลื่อนที่หรือผู้ใช้มีจำนวนเท่ากับ K และระบบมีอัตราขยายประมวลผลเท่ากับ N สัญญาณที่รับได้จากฝั่งรับเป็นเป็นเวกเตอร์ขนาด $N \times 1$ ซึ่งเป็นผลรวมของสัญญาณจากผู้ใช้แต่ละรายและสัญญาณรบกวนสีขาว (Additive White Gaussian Noise หรือ AWGN) ซึ่งสามารถอธิบายได้ดังสมการที่ (2)

$$\mathbf{r} = \sum_{k=1}^{K} \sqrt{A_k}\,\mathbf{H}_k b_k \mathbf{s}_k + \mathbf{n} \qquad (2)$$

โดยตัวแปรต่างๆในสมการที่ 2 สามารถอธิบายได้ดังนี้ $\mathbf{r}$ เป็นเวกเตอร์ของสัญญาณรับ, $\sqrt{A_k}$ เป็นแอมพลิจูดของสัญญาณสำหรับผู้ใช้รายที่ k, $\mathbf{H}_k$ เป็นเมทริกซ์ช่องสัญญาณขนาด $N \times N$, $\mathbf{s}_k$ เป็นรหัสลายเซ็น, b_k เป็นบิตข้อมูลและ $\mathbf{n}$ เป็นสัญญาณรบกวนสีขาวที่มีค่าเฉลี่ย (mean) เท่ากับศูนย์และมีเมทริกซ์แปรผันร่วม (covariance matrix) เท่ากับ $\sigma_n^2\mathbf{I}$ เมื่อ $\mathbf{I}$ เป็นเมทริกซ์เอกลักษณ์ (identity matrix) ในกรณีที่เป็นช่องสัญญาณในอุดมคติ $\sqrt{A_k}$ มีค่าเท่ากับ 1 และ $\mathbf{H}_k$ เป็นเมทริกซ์เอกลักษณ์ $\mathbf{I}_N$ สำหรับช่องสัญญาณจางหายแบบเลือกความถี่ (frequency selective fading channel) ซึ่งเป็นช่องสัญญาณแบบเรย์ลี (Rayleigh) สมมติให้สัญญาณของผู้ใช้แต่ละรายมีเส้นทาง L เส้นทางซึ่งแต่ละเส้นทางมีกำลังงานเท่าๆกัน ช่องสัญญาณ $\mathbf{H}_k$ สามารถเขียนอธิบายได้โดยสมการที่ 3 ดังนี้

$$\mathbf{H}_k = \begin{bmatrix} h_{k,1} & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & h_{k,1} & & \vdots & 0 & & \vdots \\ h_{k,L} & \vdots & \ddots & 0 & \vdots & & 0 \\ 0 & h_{k,L} & & h_{k,1} & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots & h_{k,1} & & 0 \\ 0 & \vdots & & h_{k,L} & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & h_{k,L} & \dots & h_{k,1} \end{bmatrix} \quad (3)$$

จากสมการที่ (3) อัตราการจางหายของผู้ใช้รายที่ k หรือ $h_{k,1}, \dots, h_{k,L}$ เป็นเป็นตัวแปรสุ่ม gaussian แบบเชิงซ้อนที่เป็นอิสระต่อกัน มีค่าเฉลี่ยเป็น 0 และค่าความแปรปรวนเป็น $E\left|h_{k,1}\right|^2, \dots, E\left|h_{k,l}\right|^2$ ตามลำดับ นอกจากช่องสัญญาณจางหายแบบเลือกความถี่แล้ว สัญญาณถูกลดทอนจากระยะทางสามารถอธิบายได้ด้วยสมการที่ 4

$$A_{k,dB} = A_{k,dB}(d_0) + 10\alpha \log\left(\frac{d}{d_0}\right) + X \quad (4)$$

จากสมการที่ 2 กำหนดให้ผู้ใช้รายที่ 1 ($k = 1$) ต้องการปรับรหัสลายเซ็นของตัวเองโดยที่ไม่สนใจผู้ใช้รายอื่นซึ่งในกรณีนี้ผู้ใช้รายอื่นจะประพฤติตัวเป็นสัญญาณแทรกสอด (interference) เข้ามาในระบบของผู้ใช้รายที่ 1 กำหนดให้ผู้ใช้รายที่ 1 ใช้เครื่องรับชนิดแมตช์ฟิลล์เตอร์ (match filter) ที่มีเบซิสฟังก์ชัน (basis function) เป็นเวกเตอร์ตัวเดียวกับรหัสลายเซ็นของผู้ใช้รายที่ 1 อธิบายได้ตามสมการที่ 5

$$\mathbf{c}_1 = \frac{\mathbf{H}_1\mathbf{s}_1}{\left|\mathbf{H}_1\mathbf{s}_1\right|} \quad (5)$$

เมื่อใช้เบซิสฟังก์ชันดังสมการที่ 5 เราสามารถคำนวณอัตราส่วนของสัญญาณต่อสัญญาณแทรกสอดและสัญญาณรบกวน (Signal-to-Interference plus Noise Ratio หรือ SINR) ได้ดังสมการที่ 6

$$SINR(\mathbf{s}_1) = \frac{A_1(\mathbf{s}_1^\dagger\mathbf{H}_1^\dagger\mathbf{H}_1\mathbf{s}_1)^2}{\sum_{j=2}^{K} A_j(\mathbf{s}_1^\dagger\mathbf{H}_1^\dagger\mathbf{H}_j\mathbf{s}_j)^2 + \sigma_n^2(\mathbf{s}_1^\dagger\mathbf{H}_1^\dagger\mathbf{H}_1\mathbf{s}_1)} \quad (6)$$

โดยโอเปอร์เรชัน $[\cdot]^{\dagger}$ หมายถึงการทำเฮอมิเชียนทรานสโพส (hermitian transpose) คือการสลับดำดับของแถวกับหลักและการทำคอนจูเกตของสมาชิกในเมทริกซ์ พิจารณาสมการที่ 6 เทอม $A_1(\mathbf{s}_1^{\dagger}\mathbf{H}_1^{\dagger}\mathbf{H}_1\mathbf{s}_1)^2$ คือกำลังของสัญญาณของผู้ใช้รายที่ 1 จากเอาท์พุตของแมตช์ฟิลล์เตอร์และเทอม $\sum_{j=2}^{K} A_j(\mathbf{s}_1^{\dagger}\mathbf{H}_1^{\dagger}\mathbf{H}_j\mathbf{s}_j)^2$ คือกำลังของสัญญาณแทรกสอดจากเอาท์พุตของแมตช์ฟิลล์เตอร์ ในกรณีที่เป็นช่องสัญญาณในอุดมคติสมการที่ 6 สามารถเขียนได้เป็น

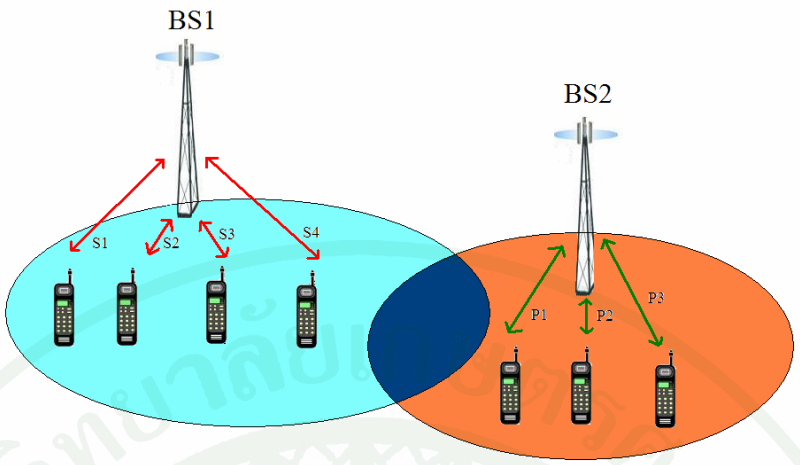
$$SINR(\mathbf{s}_1) = \frac{1}{\sigma_n^2 + \mathbf{s}_1^{\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{s}_1} \tag{7}$$

โดย $\mathbf{S}$ ในสมการที่ 7 เป็นเมทริกซ์ขนาด $N \times (K-1)$ เป็นตัวแทนของรหัสลายเซ็นของผู้ใช้รายอื่นในระบบซีดีเอ็มเอดังสมการที่ 8

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_2 & \mathbf{s}_3 & \dots & \mathbf{s}_k \end{bmatrix} \tag{8}$$

พิจาณาสมการที่ 6 พบว่า SINR เป็นฟังก์ชั่นของรหัสลายเซ็นของผู้ใช้รายที่ 1 ซึ่งรหัสลายเซ็นดังกล่าวสามารถปรับได้เพื่อเพิ่ม SINR ให้สูงขึ้น การปรับรหัสลายเซ็นนั้นสามารถปรับได้ทั้งที่ฝั่งส่งและฝั่งรับ

ระบบซีดีเอ็มเอและสมการคำนวณ SINR ข้างต้นเป็นระบบสำหรับผู้ใช้รายเดียวหรือกล่าวได้ว่ามีผู้ใช้ในระบบเพียงรายเดียวเท่านั้นที่ต้องการปรับรหัสลายเซ็น สำหรับระบบซีดีเอ็มเอที่ผู้ใช้ในระบบที่ต้องการปรับรหัสลายเซ็นมีมากกว่า 1 รายนั้นเราแยกผู้ใช้ออกเป็น 2 กลุ่มโดยกลุ่มแรกเป็นกลุ่มของผู้ใช้ที่ต้องการปรับรหัสลายเซ็นและกลุ่มที่สองเป็นกลุ่มของผู้ใช้ที่ไม่ปรับรหัสลายเซ็นโดยกลุ่มนี้อาจมาองได้ว่าเป็นผู้ใช้แทรกสอดจากเซลล์ข้างเคียง ระบบซีดีเอ็มเอที่ต้องการปรับรหัสสลายเซ็นของผู้ใช้ทั้งกลุ่มแสดงได้ดังภาพที่ 6



ภาพที่ 6 แสดงระบบซีดีเอ็มเอที่ต้องการปรับรหัสสลายเซ็นของผู้ใช้ทั้งกลุ่ม

จากภาพที่ 6 กำหนดให้ผู้ใช้ที่ติดต่อกับสถานีฐาน 1 ต้องการปรับรหัสสลายเซ็นภายในกลุ่ม และมีผู้ใช้ที่ติดต่อกับสถานีฐาน 2 เป็นสัญญาณแทรกสอดเข้ามาโดยกำหนดให้ผู้ใช้ที่ต้องการปรับรหัสสลายเซ็นมีจำนวนเท่ากับ K_a และผู้ใช้ที่แทรกสอดเข้ามามีจำนวนเท่ากับ K_b กำหนดให้ $\{\mathbf{s}_k, 1 \le k \le K_a\}$ เป็นเซตของรหัสลายเซ็นของผู้ใช้ที่ต้องการปรับรหัสลายเซ็นและ $\{\mathbf{p}_i, 1 \le i \le K_b\}$ เป็นเซ็นของรหัสลายเซ็นของผู้ใช้ที่ไม่ปรับรหัสลายเซ็นและประพฤติตัวเป็นสัญญาณแทรกสอด (interference) เข้ามาในระบบ อ้างอิงจากสมการสัญญาณรับสำหรับผู้ใช้รายเดียวเราสามารถอธิบายสัญญาณรับของระบบ CDMAสำหรับผู้ใช้แบบกลุ่มได้ดังสมการที่ 9

$$\mathbf{r} = \sum_{k=1}^{K_a} \sqrt{A_k}\mathbf{H}_k\mathbf{s}_k b_k + \sum_{k=1}^{K_b} \sqrt{A_i}\mathbf{H}_i\mathbf{p}_i b_i + \mathbf{n} \qquad (9)$$

จากสมการเราสามารถคำนวณ SINR เฉลี่ยของระบบได้จากสมการที่ 10

$$SINR(\mathbf{S}) = \frac{\sum_{k=1}^{K_a} A_k \left(\mathbf{s}_k^{\dagger}\mathbf{H}_k^{\dagger}\mathbf{H}_k\mathbf{s}_k\right)^2}{\sum_{k=1}^{K_a}\sum_{l=1,l\neq k}^{K_a} A_l \left(\mathbf{s}_k^{\dagger}\mathbf{H}_k^{\dagger}\mathbf{H}_l\mathbf{s}_l\right)^2 + \sum_{k=1}^{K_a}\sum_{i=1}^{K_b} A_i \left(\mathbf{s}_k^{\dagger}\mathbf{H}_k^{\dagger}\mathbf{H}_i\mathbf{p}_i\right)^2 + \sigma_n^2 \sum_{k=1}^{K_a}\left(\mathbf{s}_k^{\dagger}\mathbf{H}_k^{\dagger}\mathbf{H}_k\mathbf{s}_k\right)} \qquad (10)$$

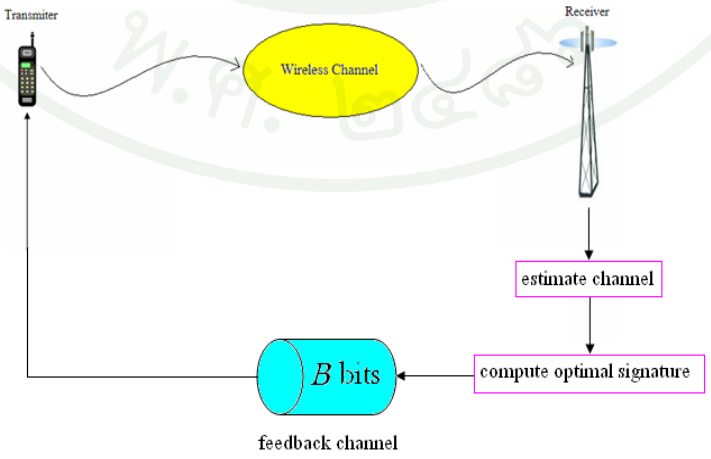
ในกรณีที่เป็นช่องสัญญาณในอุดมคติสมการที่ 10 สามารถเขียนได้เป็น

$$SINR(\mathbf{S}) = \frac{1}{\sigma_n^2 + \frac{1}{K_a} tr\{\mathbf{S}^\dagger \mathbf{S}\mathbf{S}^\dagger \mathbf{S}\} - 1 + \frac{1}{K_a} tr\{\mathbf{P}^\dagger \mathbf{S}\mathbf{S}^\dagger \mathbf{P}\}} \qquad (11)$$

โดยที่ $\mathbf{S} = [\mathbf{s_1} \quad \mathbf{s}_2 \quad \dots \quad \mathbf{s}_{K_a}]$ และ $\mathbf{P} = [\mathbf{p_1} \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_{K_b}]$ โดยในงานวิจัยนี้เราจะใช้สมการคำนวณ SINR เป็นดรรชนีชี้วัดสมรรถนะของการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างๆ

3. การปรับรหัสลายเซ็นโดยช่องสัญญาณป้อนกลับ

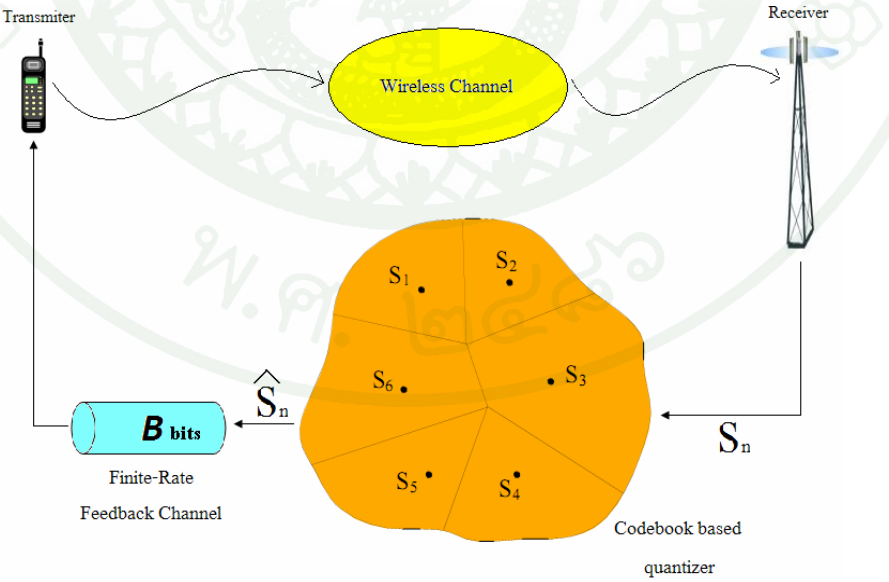
เมื่อพิจารณาสมการคำนวณ SINR แต่ละแบบข้างต้นจะเห็นว่า SINR เป็นฟังก์ชันของรหัสลายเซ็นโดยเราสามารถปรับรหัสลายเซ็นได้เพื่อเพิ่ม SINR ซึ่งเป็นดรรชนีชี้วัดสมรรถนะของระบบ งานวิจัย (Rapajic and Vucetic 1994) สนใจการปรับรหัสลายเซ็นระหว่างการส่งสัญญาณ งานวิจัย (Viswanath *et al.* 1998) สนใจทั้งการปรับรหัสลายเซ็นและกำลังส่งของสัญญาณ งานวิจัย (Rajjappan and Honig 2002) สนใจการปรับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มโดยปรับทั้งฝั่งรับและฝั่งส่งโดยติดต่อผ่านทางช่องสัญญาณป้อนกลับ โดยงานวิจัยที่กล่าวมาข้างต้นนั้นพิจารณาการปรับรหัสลายเซ็นในกรณีที่ช่องสัญญาณป้อนกลับมีอัตราบิตที่ไม่จำกัดซึ่งในทางปฏิบัตินั้นช่องสัญญาณป้อนกลับมีอัตราบิตที่จำกัด การปรับรหัสลายเซ็นผ่านทางช่องสัญญาณป้อนกลับสามารถอธิบายได้ดังภาพที่ 7



ภาพที่ 7 การปรับรหัสลายเซ็นในระบบซีดีอ็มเอโดยผ่านทางช่องสัญญาณป้อนกลับ

จากภาพที่ 7 กำหนดให้โทรศัพท์เคลื่อนที่เป็นผู้ส่งและสถานีฐานเป็นผู้รับ สัญญาส่งถูกส่งผ่านทางช่องสัญญาณไร้สายไปยังผู้รับโดยที่ฝั่งรับสามารถประมาณช่องสัญญาณ, สัญญาณแทรกสอดและสัญญาณรบกวนได้ หลังจากนั้นฝั่งรับจะคำนวณรหัสลายเซ็นที่เหมาะสมของผู้ใช้และส่งข้อมูลของรหัสลายเซ็นไปยังฝั่งส่งผ่านทางช่องสัญญาณ การใช้ประโยชน์จากช่องสัญญาณป้อนกลับในระบบสื่อสารไร้สายนี้สามารถประยุกต์ใช้ในระบบสื่อสารไร้สายแบบหลายสายอากาศ (multiple-antenna) ด้วยซึ่งอยู่ในงานวิจัย (Love, *et al*. 2003), (Santipach and Honig 2009) และ (Santipach *et al*. 2003)

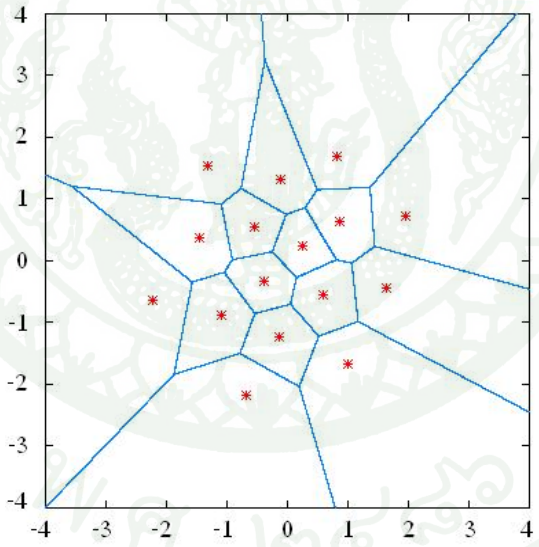
พิจารณาภาพที่ 7 ในกรณีที่ช่องสัญญาณป้อนกลับมีอัตราบิตที่ไม่จำกัดฝั่งรับสามารถส่งข้อมูลของรหัสลายเซ็นที่เหมาะสมไปยังฝั่งส่งได้อย่างสมบูรณ์ แต่ในทางปฏิบัติช่องสัญญาณป้อนกลับนี้มีอัตราบิตที่จำกัด ดังนั้นรหัสลายเซ็นต้องมีการแบ่งนับ (quantize) ก่อนส่งกลับไปยังฝั่งส่ง การแบ่งนับสามารถทำได้โดยที่ฝั่งรับเลือกเวกเตอร์ในกลุ่มของเวกเตอร์หรือโค้ดบุ๊ค (codebook) ที่ใกล้เคียงกับรหัสลายเซ็นที่เหมาะสมหลังจากนั้นฝั่งรับจะส่งดรรชนี (index) ของเวกเตอร์ตัวที่ถูกเลือกจำนวน B บิตไปยังฝั่งส่งผ่านทางช่องสัญญาณป้อนกลับโดยที่ฝั่งส่งเมื่อได้รับข้อมูลแล้วก็จะนำดรรชนีที่ได้ไปจับคู่กับเวกเตอร์ในโค้ดบุ๊คเดียวกันกับฝั่งส่งเพื่อใช้เป็นรหัสลายเซ็นสำหรับการส่งสัญญาณในครั้งต่อไปโดยสามารถอธิบายได้ดังภาพที่ 8



ภาพที่ 8 การแบ่งนับรหัสลายเซ็นสำหรับช่องสัญญาณป้อนกลับที่มีอัตราบิตจำกัด

4. การแบ่งนับรหัสลายเซ็น

จุดประสงค์ของการแบ่งนับรหัสลายเซ็นนั้นเพื่อสามารถใช้จำนวนบิตจำกัดในการบ่งชี้รหัสลายเซ็นที่ต้องการส่งไปยังฝั่งส่ง การแบ่งนับรหัสลายเซ็นนั้นมีพื้นฐานมาจากการแบ่งนับเวกเตอร์ซึ่งการแบ่งนับเวกเตอร์เป็นกระบวนการบีบอัดข้อมูลประเภทหนึ่ง การแบ่งนับเวกเตอร์นั้นมีจุดประสงค์คือการสร้างกลุ่มของเวกเตอร์และวิธีการหาตัวแทนที่ทำให้ความผิดพลาดระหว่างเวกเตอร์ตัวดั้งเดิมและเวกเตอร์ตัวที่ถูกแบ่งนับโดยเฉลี่ยมีค่าน้อยที่สุด โดยทั่วไปจะใช้ระยะทางยูคลิด (euclidean distance) สำหรับการเปรียบเทียบ วิธีการแบ่งนับเวกเตอร์ที่เหมาะสมได้แก่วิธี Generalized Lloyd Algorithm หรือ GLA โดยวิธีนี้จะสร้างตัวแทนของกลุ่มย่อยขึ้นมาเท่ากับจำนวนกลุ่มที่ต้องการหลังจากนั้นจึงเปรียบเทียบระยะทางระหว่างเวกเตอร์ทั้งหมดและตัวแทนของแต่ละกลุ่มย่อย เวกเตอร์ที่อยู่ใกล้ตัวแทนของกลุ่มย่อมใดมากที่สุดก็จัดให้อยู่ในกลุ่มย่อยนั้น ภาพที่ 8 แสดงการแบ่งนับเวกเตอร์ใน 2 มิติ



ภาพที่ 9 การแบ่งนับเวกเตอร์ใน 2 มิติ

ที่มา: Gersho and Gray (1991)

จากภาพที่ 9 เวกเตอร์แต่ละตัวถูกจัดลงในกลุ่มทั้ง 16 กลุ่มซึ่งแต่ละกลุ่มสามารถแทนได้ด้วย 4 บิตข้อมูล จากภาพที่ 9 เราเรียกกลุ่มของเวกเตอร์ทุกกลุ่มรวมกันว่าโค้ดบุ๊ค (codebook) (Gersho and Gray 1991) สำหรับการแบ่งนับรหัสลายเซ็นในระบบซีดีเอ็มเอในงานวิจัยนี้

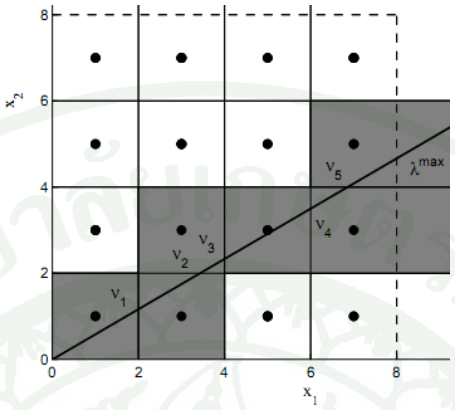
กำหนดให้รหัสลายเซ็นแต่ละตัวเป็นเวกเตอร์ที่มีมิติเท่ากับ N งานวิจัยนี้ศึกษาการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างๆกันดังนี้

การแบ่งนับเวกเตอร์สุ่ม (Random Vector Quantization หรือ RVQ) การแบ่งนับรหัสลายเซ็นแบบ RVQ นี้ถูกนำเสนอครั้งแรกในงานวิจัย (Santipach and Honig 2005) โดยแนวคิดของ RVQ นั้นเกิดจากเมื่อระบบซีดีเอ็มเอมีช่องสัญญาณป้อนกลับที่มีอัตราบิตที่ไม่จำกัดแล้วรหัสลายเซ็นที่เหมาะสมที่สุดที่ลดกำลังงานของสัญญาณแทรกสอดจะเป็นเวกเตอร์ลักษณะเฉพาะ (eigenvector อ่านว่า “ไอเกนเวกเตอร์”) ของเมทริกซ์แปรผันร่วมของสัญญาณแทรกสอด (interference covariance matrix) โดยไอเกนเวกเตอร์ดังกล่าวเป็นเวกเตอร์ตัวที่ให้ค่าลักษณะเฉพาะ (eigenvalue) ต่ำที่สุดและสามารถหาได้จากกระบวนการ singular value decomposition หรือ SVD ซึ่งไอเกนเวกเตอร์ที่ได้จากกระบวนการ SVD นั้นมีแต่ละตัวเป็นอิสระต่อกัน มีคุณสมบัติที่เหมือนกันและมีการกระจายในทุกทิศทางเท่าๆกัน (independent and isotropically distributed หรือ i.i.d.) ดังนั้นงานวิจัย (Santipach and Honig 2005) จึงได้สร้างโค้ดบุ๊คที่มีลักษณะ i.i.d. ขึ้นโดยให้ชื่อว่าโค้ดบุ๊คแบบ RVQ โดยดรรชนีหรือจำนวนเวกเตอร์ในโค้ดบุ๊คถูกแทนด้วยไบนารี่บิตดังนั้นจำนวนเวกเตอร์ทั้งหมดในโค้ดบุ๊คจึงเท่ากับ 2^B สำหรับการค้นหาเวกเตอร์ตัวที่เหมาะสมนั้นการแบ่งนับเวกเตอร์แบบ RVQ จะคำนวณ SINR จากเวกเตอร์ทุกตัวแล้วเลือกเวกเตอร์ตัวที่ให้ SINR สูงที่สุดเป็นรหัสลายเซ็นสำหรับผู้ใช้ที่ต้องการปรับรหัสลายเซ็น การแบ่งนับเวกเตอร์ด้วยวิธี RVQ นี้เป็นการแบ่งนับรหัสลายเซ็นที่เหมาะสมที่สุดในระบบจำกัดที่มีขนาดใหญ่กล่าวคือระบบที่จำนวนผู้ใช้ (K) อัตราขยายประมวลผล (N) และจำนวนบิตป้อนกลับ (B) ลู่เข้าสู่อนันต์โดยที่กำหนดให้อัตราส่วนของ K/N และ B/N มีค่าคงที่ งานวิจัย (Santipach and Honig 2005) ได้วิเคราะห์สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบ RVQ และแสดงให้เห็นว่าผลการวิเคราะห์สามารถใช้ทำนายแนวโน้มของสมรรถนะได้อย่างแม่นยำ อย่างไรก็ตามถึงแม้ว่าการแบ่งนับเวกเตอร์แบบ RVQ จะเหมาะสมที่สุดในระบบที่มีขนาดใหญ่และมีสมรรถนะใกล้เคียงกับโค้ดบุ๊คที่เหมาะสม แต่เนื่องจากการแบ่งนับเวกเตอร์แบบ RVQ ต้องทดสอบเวกเตอร์ทุกตัวในโค้ดบุ๊คเพื่อการได้มาของเวกเตอร์ตัวที่เหมาะสม ดังนั้นการแบ่งนับเวกเตอร์ด้วยวิธี RVQ จึงต้องใช้การคำนวณที่ซับซ้อนและใช้เวลาเป็นอย่างมากเมื่อจำนวนบิตป้อนกลับเพิ่มขึ้นเนื่องจากจำนวนเวกเตอร์ในโค้ดบุ๊คเป็นฟังก์ชันของบิตป้อนกลับหรือเท่ากับ 2^B ตัวหรือกล่าวได้ว่าจำนวนการคำนวณเพื่อเข้าถึงเวกเตอร์ตัวที่เหมาะสมของวิธี RVQ เพิ่มขึ้นอย่างเอกซ์โพเนนเชียลกับบิตป้อนกลับ B ซึ่งเป็นปัญหาใหญ่เมื่อบิตป้อนกลับมีจำนวนมาก

จากปัญหาของการแบ่งนับเวกเตอร์แบบ RVQ ที่ต้องทำการทดสอบเวกเตอร์ทุกตัวในโค้ดบุ๊คนั้นได้มีนักวิจัยต้องการลดความซับซ้อนในการแบ่งนับเวกเตอร์สำหรับช่องสัญญาณป้อนกลับและปัญหาอื่นที่ใกล้เคียงโดยคิดวิธีการแบ่งนับและการสร้างโค้ดบุ๊คที่โครงสร้างมีลักษณะแตกต่างกันออกไป การแบ่งนับเวกเตอร์ที่งานวิจัยนี้สนใจเป็นอันดับแรกได้แก่การแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้ (Tree-Structured Vector Quantization หรือ TS-VQ) โดยโครงสร้างต้นไม้เป็นการจัดเรียงข้อมูลอย่างย่อประเภทหนึ่งโดยมีจุดประสงค์เพื่อลดความซับซ้อนในการเข้าถึงข้อมูล การสร้างโค้ดบุ๊คแบบโครงสร้างต้นไม้นั้นใช้วิธีที่เรียกว่า Generalized Lloyd Algorithm หรือ GLA สำหรับแบ่งเวกเตอร์ออกเป็นสองกลุ่มย่อย งานวิจัย (Katsavounidis *et al.* 1996) ใช้การแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้สำหรับการหาเวกเตอร์ใกล้เคียง (nearest neighbor) กับเวกเตอร์อินพุต งานวิจัย (Santipach, 2008) .ใช้การแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้สำหรับแบ่งนับเวกเตอร์กำหนดทิศทาง (beamforming vector) สำหรับช่องสัญญาณป้อนกลับในระบบหลายสายอากาศ โดยใช้โค้ดบุ๊คชนิด RVQ ตั้งต้นสำหรับสร้างโค้ดบุ๊คแบบโครงสร้างต้นไม้และให้ชื่อว่า Tree-Structured Random Vector Quantization หรือ TS-RVQ โดยการค้นหาเวกเตอร์ตัวที่เหมาะสมในโค้ดบุ๊ค TS-RVQ นั้นทำได้โดยการแวะผ่าน (transverse) เข้าไปในปมรากหรือโหนด (node) ของโครงสร้างต้นไม้โดยการเปรียบเทียบฟังก์ชันเป้าหมาย (objective function) ผลที่ได้จากงานวิจัยนี้ปรากฏว่าการแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้ต้องการจำนวนการคำนวณที่น้อยกว่าการแบ่งนับเวกเตอร์แบบสุ่มเมื่อเปรียบเทียบที่สมรรถนะที่เท่ากัน

PAM (Pulse Amplitude Modulation) เป็นเทคนิคการมอดูเลตสัญญาณประเภทหนึ่งซึ่งวิธีการดีมอดูเลตสัญญาณพีเอเอ็มนั้นใช้วิธีที่เรียกว่า noncoherent detection โดยวิธีการดังกล่าวเป็นการหาเวกเตอร์ใกล้เคียงกับสัญญาณที่รับได้ ซึ่งวิธีการดังกล่าวแต่เดิมต้องทดสอบเวกเตอร์ทุกตัวในโค้ดบุ๊คชนิดพีเอเอ็ม สำหรับ M-ary PAM นั้นจำนวนเวกเตอร์ในโค้ดบุ๊คชนิดพีเอเอ็มจะเพิ่มขึ้นตามค่า M เมื่อ M มีค่ามากก็จะต้องใช้จำนวนการคำนวณที่มากครั้งเนื่องจากเวกเตอร์ในโค้ดบุ๊คพีเอเอ็มมีจำนวนเท่ากับ M^N ซึ่งเป็นปัญหาดังกล่าวคล้ายกับปัญหาของการแบ่งนับเวกเตอร์แบบสุ่ม งานวิจัย (Ryan, *et al.* 2007) ได้เสนอวิธีลดจำนวนการคำนวณสำหรับการหาเวกเตอร์ใกล้เคียงกับสัญญาณพีเอเอ็มที่รับได้จากแนวคิดที่ว่าเวกเตอร์ที่เหมาะสมจะอยู่ในทิศทางเดียวกันกับสัญญาณที่รับได้ ดังนั้นจำนวนเวกเตอร์ที่ต้องทำการสดสอบจึงลดลงลงโดยอ้างอิงจากสัญญาณที่รับได้ซึ่งกลุ่มของเวกเตอร์ที่ต้องนำมาทดสอบนั้นเป็นเวกเตอร์ที่ใกล้เคียงในเชิงมุม (closest in angle) กับเส้นตรงที่เกิดจากการนำสัญญาณที่รับได้มาคูณกับค่าคงที่และตั้งชื่อกลุ่มของเวกเตอร์ที่ลดลงนั้นว่า

“nearest neighbor set” และใช้วิธี Generalized Likelihood Ratio Test หรือ GLRT ทดสอบเวกเตอร์ทุกตัวใน nearest neighbor set ภาพที่ 10 แสดง nearest neighbor set ของ *8*-ary PAM ที่ $N = 2$



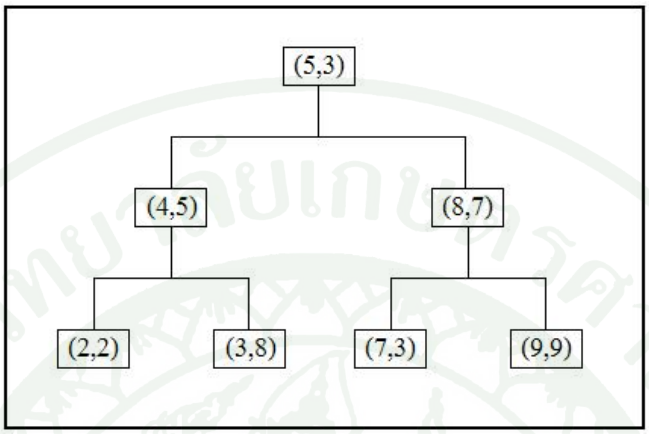
ภาพที่ 10 แสดง nearest neighbor set ของ*8*-ary PAM ที่ $N = 2$

ที่มา: Ryan *et al*. (2007)

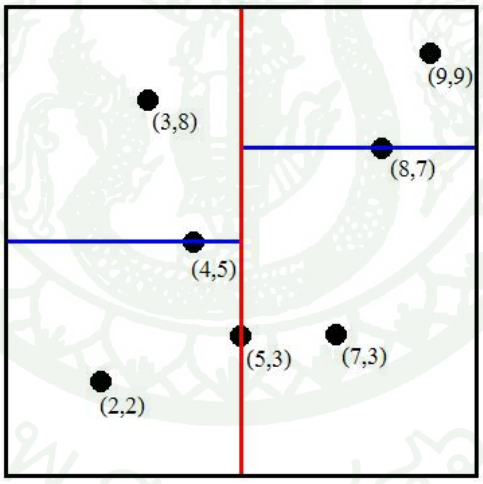
จากภาพที่ 10 กำหนดให้ส่วนที่แรเงาในภาพคือ nearest neighbor set ซึ่งจากภาพจะเห็นว่าวิธีดังกล่าวสามารถลดจำนวนการคำนวณลงได้ซึ่งงานวิจัยนี้อธิบายไว้ว่าจำนวนการคำนวณเท่ากับจำนวนของเวกเตอร์ใน nearest neighbor set งานวิจัย (Ryan *et al*. 2007) ใช้การแบ่งนับเวกเตอร์แบบ PAM ในกรณีที่เป็นจำนวนเชิงซ้อนหรือ QAM สำหรับการเลือกเวกเตอร์กำหนดทิศทางในระบบสื่อสารไร้สายแบบหลายสายอากาศโดยการกำหนดให้สัญญาณที่รับได้เป็นไอเกนเวกเตอร์ของเมทริกซ์แปรผันร่วมของช่องสัญญาณ (channel covariance matrix) ซึ่งการแบ่งนับเวกเตอร์ดังกล่าวต้องการการคำนวณที่น้อยกว่า RVQ เมื่อกำหนดให้สมรรถนะเท่ากัน

ต้นไม้เคดี (*k*d-tree) เป็นโครงสร้างต้นไม้แบบทวิภาคอีกประเภทหนึ่งที่มีวิธีการสร้างที่แตกต่างจากโครงสร้างต้นไม้โดย *k* นั้นหมายถึงจำนวนมิติของเวกเตอร์นั้นๆ แนวคิดของต้นไม้เคดีนั้นเกิดจากความต้องการสร้างโค้ดบุ๊คแบบโครงสร้างต้นไม้ที่ทำได้ง่ายกว่าวิธี GLA เนื่องจากวิธี GLA นั้นต้องใช้การคำนวณที่ซับซ้อนและจะทำงานได้ไม่ดีนักเมื่อเวกเตอร์มีจำนวนที่เพิ่มมากขึ้น การแบ่งเวกเตอร์ออกเป็นสองกลุ่มย่อยสำหรับวิธีการสร้างต้นไม้เคดีนั้นจะใช้ค่าประจำมิติของเวกเตอร์ ภาพที่ 11 และ 12 แสดงตัวอย่างของเวกเตอร์ใน 2 มิติ 7 ตัวได้แก่ (5,3), (4,5), (8,7), (2,2), (3,8), (7,3) และ (9,9) มิติแรกที่ใช้แบ่งเวกเตอร์คือแกน x โดยให้เวกเตอร์ทุกตัวที่มีค่า x น้อยกว่า 5

อยู่ในกลุ่มซ้ายและค่า x มากกว่า 5 อยู่ในกลุ่มขวาหลังจากนั้นจึงใช้แกน y แบ่งเวกเตอร์ในกลุ่มย่อยต่อไป



ภาพที่ 11 การจัดเวกเตอร์ใน 2 มิติให้อยู่ในโครงสร้างเคดี



ภาพที่ 12 การจัดเวกเตอร์ใน 2 มิติให้อยู่ในโครงสร้างเคดีในระนาบ xy

งานวิจัย (Bently, 1980) ใช้การแบ่งนับเวกเตอร์แบบ *k*d-tree ในการหาเวกเตอร์ใกล้เคียงโดยใช้การเปรียบเทียบค่าประจำมิตินั้นๆและงานวิจัย (Moore, 1991) แสดงให้เห็นว่าการแบ่งนับเวกเตอร์แบบต้นไม้เคดีใช้การคำนวณที่น้อยมากเมื่อเทียบกับจำนวนเวกเตอร์ทั้งหมดในโค้ดบุ๊ค

อุปกรณ์และวิธีการ

อุปกรณ์

1. คอมพิวเตอร์ส่วนบุคคล Intel Core 2 Quad CPU Q9400, 2.66GHz, 4GB DDR2-RAM, 500 GB HDD

2. โปรแกรม Matlab

3. โปรแกรม Visual Studio

วิธีการ

1. การแบ่งนับรหัสลายเซ็นด้วยวิธีเวกเตอร์สุ่ม

การแบ่งนับรหัสลายเซ็นด้วยวิธีเวกเตอร์สุ่มนั้นจะใช้โค้ดบุ๊คที่มีสมาชิกเป็นเวกเตอร์สุ่ม (random vector) มีการกระจายเท่ากันทุกทิศทาง (isotropic distribution) มีนอร์มเป็นหนึ่ง (unit norm) และเป็นอิสระต่อกัน (independent) เวกเตอร์ดังกล่าวสามารถสร้างโดยใช้ฟังก์ชัน randn ในโปรแกรม MATLAB และทำการนอร์มัลไลซ์ด้วยนอร์มของตัวมัน ในกรณีที่แบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวโค้ดบุ๊คดังกล่าวมีจำนวนสมาชิกเท่ากับ 2^B เมื่อ B แทนจำนวนบิตที่ใช้ป้อนกลับโดยสามารถเขียนสมการอธิบายได้ดังนี้

$$V = \left\{ \mathbf{v}_j ; 1 \le j \le 2^B \right\} \tag{12}$$

สำหรับการค้นหาเวกเตอร์ตัวที่เหมาะสมนั้น การแบ่งนับเวกเตอร์แบบสุ่มจะคำนวณ SINR จากเวกเตอร์ทุกตัว แล้วเลือกเวกเตอร์ตัวที่ให้ SINR สูงที่สุดเป็นรหัสลายเซ็นสำหรับผู้ใช้ที่ต้องการปรับรหัสลายเซ็นดังสมการที่ 13

$$\mathbf{s}_1 = \arg\max_{\mathbf{v}_j \in V} SINR(\mathbf{v}_j) \tag{13}$$

หรือสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

RVQ Encoding Algorithm (Santipach and Honig 2005)

1: SINR($\mathbf{s}_1$) = 0

2: **for** $j = 1 : 2^B$ **do**

3: Compute SINR($\mathbf{v}_j$).

4: **if** SINR($\mathbf{v}_j$) $\geq$ SINR($\mathbf{s}_1$) **then**

5: $\mathbf{s}_1 = \mathbf{v}_j$

6: SINR($\mathbf{s}_1$) = SINR($\mathbf{v}_j$)

7: **end if**

8: **end for**

9: **return** the optimal vector $\mathbf{v}_1$.

สำหรับการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์กลุ่มสำหรับผู้ใช้แบบกลุ่มนั้นทำได้โดยการเพิ่มขนาดของโค้ดบุ๊คให้เหมาะสมกับจำนวนผู้ใช้ในระบบที่ต้องการปรับรหัสลายเซ็น โดยจำนวนของสมาชิกในโค้ดบุ๊คดังกล่าวมีเท่ากับ $2^B \times K_a$ ซึ่งสามารถเขียนอธิบายเป็นสมการได้ดังนี้

$$V = \left\{ \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{n \times K_a} \right\} \tag{14}$$

หลังจากนั้นจึงจับกลุ่มของเวกเตอร์ในโค้ดบุ๊คที่อยู่ติดกันให้เป็นชุดของรหัสลายเซ็นสำหรับการแบ่งนับโดยที่เวกเตอร์ในแต่ละชุดมีจำนวนเท่ากับ K_a ดังนั้นเราจะได้โค้ดบุ๊คใหม่ที่มีสมาชิกเป็นชุดของเวกเตอร์จำนวน 2^B ชุดดังนี้

$$V = \{ \mathbf{V}_j, 1 \leq j \leq 2^B \} \tag{15}$$

ชุดของเวกเตอร์แต่ละชุดในสมการที่ 15 สามารถอธิบายได้ดังสมการที่ 16

$$\mathbf{V}_j = [\mathbf{s}_1 \quad \mathbf{s}_2 \quad ... \quad \mathbf{s}_{K_a}] \tag{16}$$

การแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มสำหรับผู้ใช้แบบกลุ่มทำได้โดยการนำชุดของเวกเตอร์แต่ละชุดมาทดสอบ SINR เฉลี่ยของระบบตามสมการที่ 10 และ 11 หลังจากนั้นจึงเลือกชุดของเวกเตอร์ที่ให้ SINR สูงที่สุดเป็นชุดของรหัสลายเซ็นสำหรับผู้ใช้ซึ่งสามารถอธิบายได้โดยสมการที่ 17

$$\mathbf{S} = \arg\max_{\mathbf{V}_j \in V} SINR(\mathbf{V}_j) \qquad (17)$$

โดยวิธีการแบ่งนับเวกเตอร์สุ่มสำหรับผู้ใช้แบบกลุ่มสามารถอธิบายเป็นซูโดโค้ดได้ดังนี้

RVQ Encoding Algorithm (Santipach and Honig 2005)

1: $SINR(\mathbf{S}) = 0$
2: **for** $j = 1 : n$ **do**
3: Compute $SINR(\mathbf{V}_j)$
4: **if** $SINR(\mathbf{V}_j) \geq SINR(\mathbf{S})$ **then**
5: $\mathbf{S} = \mathbf{V}_j$
6 $SINR(\mathbf{S}) = SINR(\mathbf{V}_j)$
7: **end if**
8: **end for**
9: **return** the optimal set of signature **S**.

สำหรับวิธีการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มสำหรับผู้ใช้แบบกลุ่มนั้นเมื่อพิจารณาโค้ดบุ๊คในสมการที่ 14 แล้วจะพบว่าเราสามารถนำเวกเตอร์แต่ละตัวในโค้ดบุ๊คมาเรียงสับเปลี่ยน (permutation) กันโดยกำหนดให้จำนวนที่จะเรียงสับเปลี่ยนเท่ากับ K_a หลังจากนั้นนำชุดของเวกเตอร์เวกเตอร์ที่ได้จากการเรียงสับเปลี่ยนแต่ละชุดมาทดสอบ SINR และเลือกชุดของเวกเตอร์ที่ให้ SINR สูงที่สุดเป็นชุดของรหัสลายเซ็นสำหรับผู้ใช้ โดยวิธีการแบ่งนับเวกเตอร์สุ่มแบบเรียงสับเปลี่ยนสามารถเขียนอธิบายเป็นซูโดโค๊ดได้ดังนี้

RVQ with Permutation Encoding Algorithm

1: $SINR(\mathbf{S}) = 0$

2: **for** $j = 1 : \frac{(n \times K_a)!}{((n \times K_a) - K_a)!}$

3: $\mathbf{V}_j = [\mathbf{s}_a \quad \mathbf{s}_b \quad ... \quad \mathbf{s}_c]$; $a \neq b \neq ... \neq c, c = K_a$

4: Compute $SINR(\mathbf{V}_j)$

5: **if** $SINR(\mathbf{V}_j) \geq SINR(\mathbf{S})$ **then**

6: $\mathbf{S} = \mathbf{V}_j$

7: $SINR(\mathbf{S}) = SINR(\mathbf{V}_j)$

8: **end if**

9: **end for**

10. **return** the optimal set of signature **S**.

การแบ่งนับกลุ่มรหัสลายเซ็นด้วยวิธีเรียงสับเปลี่ยนนี้ให้สมรรถนะดีที่สุดสำหรับการแบ่งนับรหัสลายเซ็นในกรณีของผู้ใช้แบบกลุ่มเนื่องจากวิธีนี้ทดสอบทุกชุดของเวกเตอร์ที่เป็นไปได้ซึ่งมีจำนวนเท่ากับเท่ากับ $\frac{(n \times K_a)!}{((n \times K_a) - K_a)!}$ ชุด ดังนั้นวิธีนี้จึงต้องใช้วิธีการคำนวณที่ซับซ้อนเป็นอย่างมากในการทดสอบ

2. การแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้

การแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้นั้นเริ่มจากการนำโค้ดบุ๊คแบบ RVQ มาจัดเรียงโครงสร้างใหม่โดยใช้โค้ดบุ๊คจากสมการที่ 12 สำหรับผู้ใช้รายเดียวและสมการที่ 14 สำหรับผู้ใช้แบบกลุ่ม หลังจากนั้นใช้วิธีที่มีชื่อว่า Generalized Lloyd Algorithm หรือ GLA ในการจัดเรียงโครงสร้างแบบต้นไม้ โดยวิธี GLA นี้จะแบ่งเวกเตอร์ทั้งหมดออกเป็นสองกลุ่มย่อยโดยเรียกแต่ละกลุ่มว่าปมรากหรือโหนด (node) ซึ่งแต่ละโหนดจะมีตัวแปรซึ่งเก็บตัวชี้ (pointer) ไปยังโหนดอื่นๆ ได้ การแบ่งเวกเตอร์ด้วยวิธี GLA เริ่มจากการสร้างตัวแทนของกลุ่มเวกเตอร์ (centroid) $\mathbf{c}_1$ และ $\mathbf{c}_2$ ที่ทำให้ระยะทางยูคลิด (euclidean distance) รวมระหว่างเวกเตอร์ในกลุ่มและตัวแทนกลุ่มมีค่าน้อยที่สุดซึ่งสามารถอธิบายได้ตามสมการที่ 18

$$d = \sum_{\mathbf{v}_j \in V_1} \left\| \mathbf{c}_1 - \mathbf{v}_j \right\| + \sum_{\mathbf{v}_k \in V_2} \left\| \mathbf{c}_2 - \mathbf{v}_k \right\| \qquad (18)$$

โดยที่ V_1 และ V_2 ในสมการที่ 18 แทนแต่ละกลุ่มย่อยของเวกเตอร์ตามสมการที่ (19) และ (20)

$$V_1 = \left\{ \mathbf{v}_j \in V : \left\| \mathbf{c}_1 - \mathbf{v}_j \right\| \le \left\| \mathbf{c}_2 - \mathbf{v}_j \right\| \right\} \qquad (19)$$

และ

$$V_2 = V \setminus V_1 \qquad (20)$$

จากสมการที่ 19 และ 20 V_1 คือเซตของเวกเตอร์ที่ระยะทางระหว่างสมาชิกทุกตัวในเซตกับ $\mathbf{c}_1$ น้อยกว่าระยะทางระหว่างสมาชิกทุกตัวในเซตกับ $\mathbf{c}_2$ และ V_2 คือเซตที่เหลือจาก V_1 สำหรับขั้นตอนที่ 1 ในวิธีวิธี GLA นั้นเราสามารถทำซ้ำกระบวนการหาระยะทางยูคลิด d และกลุ่มย่อยได้และนำมาเปรียบเทียบกับระยะทางยูคลิด d จากรอบก่อนหน้าจนกระทั่งระยะ d ไม่มีการเปลี่ยนแปลง วิธีการดังกล่าวสามารถเขียนบรรยายเป็นซูโดโค้ดได้ดังนี้

Generalized Lloyd Algorithm (Santipach, 2008)

1: Start with initial centroids $\mathbf{c}_1^{(0)}$, $\mathbf{c}_2^{(0)}$, RVQ codebook $V = \{\mathbf{v}_1, ..., \mathbf{v}_{2^B}\}$, and the distance metric $d^0 = 0$. Set $k = 0$.

2: **repeat**

3: Find

$$V_1^{(k)} = \{\mathbf{v}_i \in V \mid \left\| \mathbf{c}_1^{(k)} - \mathbf{v}_i \right\| \le \left\| \mathbf{c}_2^{(k)} - \mathbf{v}_i \right\|\}$$
$$V_2^{(k)} = V \setminus V_1^{(k)}.$$

4: Compute

$$d^{(k)} = \sum_{\mathbf{v}_j \in V_1^{(k)}} \left\| \mathbf{c}_1^{(k)} - \mathbf{v}_j \right\| + \sum_{\mathbf{v}_k \in V_2^{(k)}} \left\| \mathbf{c}_2^{(k)} - \mathbf{v}_k \right\|.$$

5: Compute new centroids

$$\mathbf{c}_1^{(k+1)} = \sum_{\mathbf{v}_i \in V_1^{(k)}} \mathbf{v}_i \Big/ \left| V_1^{(k)} \right|$$
$$\mathbf{c}_2^{(k+1)} = \sum_{\mathbf{v}_j \in V_2^{(k)}} \mathbf{v}_j \Big/ \left| V_2^{(k)} \right|$$

where $\left|V_i^{(k)}\right|$ denotes number of vectors in the set.

6: $k = k + 1$

7: **until** $\frac{d^{(k)} - d^{(k-1)}}{d^{(k)}} < \varepsilon$

เมื่อกระบวนการ GLA เสร็จสิ้นจะได้เวกเตอร์สองโหนดคือโหนดทางซ้ายและโหนดทางขวาซึ่งประกอบด้วยเวกเตอร์ตัวแทนกลุ่มและสมาชิกในเซต หลังจากนั้นใช้วิธี GLA กับเวกเตอร์ในแต่ละโหนดอีกครั้ง ทำซ้ำกระบวนการนี้ไปเรื่อยๆจนกระทั่งแต่ละโหนดมีเวกเตอร์อยู่เพียงตัวเดียวโดยโหนดดังกล่าวมีชื่อว่าโหนดใบ (leaf node) ขั้นตอนดังกล่าวสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

Tree Building Algorithm (Santipach, 2008)

1: Generate RVQ codebook with 2^B entries.

2: Store the entrie codebook at the root of binary tree.

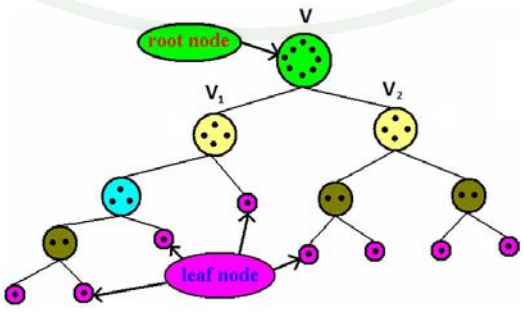
3: **repeat**

4: Use the GLA to obtain the two centroids $\mathbf{c}_1$ and $\mathbf{c}_2$ and the corresponding two clusters.

5: Create two child nodes and store each subset in each child node.

6: **until** Each node contain only one vector

ด้วยวิธีการดังกล่าวโค้ดบุ๊คแบบโครงสร้างต้นไม้ที่ได้อาจจะไม่สมดุล (balance) ภาพที่ 13 แสดงตัวอย่างของโครงสร้างต้นไม่ที่ไม่สมดุล ในกรณีที่โค้ดบุ๊คแบบโครงสร้างต้นไม้มีความสมดุลนั้นความลึก (depth) ของโครงสร้างต้นไม้ที่ได้จะเท่ากับ B



ภาพที่ 13 ตัวอย่างโครงสร้างต้นไม้ที่ไม่สมดุล (unbalanced tree)

สำหรับการค้นหาเวกเตอร์ตัวที่เหมาะสมจากโครงสร้างต้นไม้นั้นทำได้โดยการแวะผ่าน (transverse) เข้าไปในโครงสร้างต้นไม้โดยการเปรียบเทียบ SINR($\mathbf{c}_1$) และ SINR($\mathbf{c}_2$) เมื่อ $\mathbf{c}_1$ และ $\mathbf{c}_2$ คือเวกเตอร์ตัวแทนของโหนดซ้ายและโหนดขวาที่อยู่ถัดไปตามลำดับ หลังจากนั้นจึงย้ายไปยังโหนดที่ให้ SINR มากกว่าและทำเช่นนี้ไปเรื่อยๆจนถึงโหนดใบ วิธีการดังกล่าวสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

Tree-Structured Encoding Algorithm

1: Start at the root node with interfering signature **S**.
2: **while** The vector set for the node has more than one vector **do**.
3: **if** $SINR(\mathbf{c}_1) \geq SINR(\mathbf{c}_2)$ **then**
4: Move to the left child node.
5: **else**
6: Move to the right child node.
7: **end if**
8: **end while**
9: **return** The vector in the set.

สำหรับการแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้นั้นการเข้าถึงเวกเตอร์ตัวที่เหมาะสมในแต่ละครั้งนั้นจะมีการคำนวณก็ต่อเมื่อต้องการแวะผ่านแต่ละโหนดเท่านั้นซึ่งดังนั้นจำนวนการคำนวณที่ต้องการสำหรับการเข้าถึงเวกเตอร์ตัวที่เหมาะสมจึงขึ้นอยู่กับความลึก (depth) ของโครงสร้างต้นไม้ หรืออาจกล่าวได้ว่าจำนวนของการคำนวณที่ต้องการเพิ่มขึ้นอย่างเส้นตรง (linearly) กับบิตป้อนกลับ B

สำหรับการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้สำหรับผู้ใช้แบบกลุ่มนั้นเราใช้วิธีการดังต่อไปนี้

Tree-Structured Encoding for Multiuser Algorithm

1: $\mathbf{S} = [\ ]$
2: **for** $i = 1 : K_a$ **do**
3: Start at root node of a TS-RVQ codebook with $\mathbf{S}$.

4: **while** Not a leaf node. **do**

5: $\mathbf{V}_L = [\mathbf{S} \quad \mathbf{c}_1]$

6: $\mathbf{V}_R = [\mathbf{S} \quad \mathbf{c}_2]$

7: **if** $SINR(\mathbf{V}_L) \geq SINR(\mathbf{V}_R)$ **then**

8: Move to the left child node.

9: $\mathbf{T} = \mathbf{V}_L$

10: **else**

11: Move to the right child node.

12: $\mathbf{T} = \mathbf{V}_R$

13: **end if**

14: **end while**

15: $\mathbf{S} = \mathbf{T}$

16: **end for**

17: **return** $\mathbf{S}$

การค้นหารหัสลายเซ็นของผู้ใช้รายที่ 1 ทำได้โดยเริ่มต้นที่โหนดรากของโครงสร้างต้นไม้และแวะผ่านเข้าไปในโครงสร้างต้นไม้จนถึงโหนดใบก็จะได้รหัสลายเซ็นสำหรับผู้ใช้รายที่ 1 สำหรับผู้ใช้รายถัดไปนั้นจะกลับมาเริ่มต้นที่โหนดรากและแวะผ่านไปในโครงสร้างต้นไม้อีกครั้งโดยแต่ละครั้งจะนำเอารหัสลายเซ็นที่ได้จากครั้งก่อนหน้าทุกตัวมาคำนวณ SINR ด้วย ทำซ้ำกระบวนการนี้ไปเรื่อยๆจนกระทั้งได้รหัสลายเซ็นครบสำหรับผู้ใช้ทุกราย การแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้สำหรับผู้ใช้แบบกลุ่มนี้ในการคำนวณ SINR แต่ละครั้งเราจะคำนวณเพียงเวกเตอร์ในหลักสุดท้ายของเมทริกซ์ **S** เท่านั้นเนื่องจากเวกเตอร์ในหลักอื่นๆของเมทริกซ์ **S** นั้นเราได้คำนวณมาก่อนแล้วในการแวะผ่านไปในโครงสร้างต้นไม้ก่อนหน้านี้ซึ่งได้แสดงไว้ในภาพที่ 14

$$\frac{1}{\sigma_n^2+\frac{1}{K_a}tr\{\mathbf{c}_1^{\dagger}\mathbf{c}_1\mathbf{c}_1^{\dagger}\mathbf{c}_1\}+\frac{1}{K_a}tr\{\mathbf{P}^{\dagger}\mathbf{c}_1\mathbf{c}_1^{\dagger}\mathbf{P}\}}$$

คำนวณ SINR สำหรับผู้ใช้รายที่ 1

$$\frac{1}{\sigma_n^2+\frac{1}{K_a}tr\{[\mathbf{s}_1 \quad \underline{\mathbf{c}}_1]^{\dagger}[\mathbf{s}_1 \quad \underline{\mathbf{c}}_1][\mathbf{s}_1 \quad \underline{\mathbf{c}}_1]^{\dagger}[\mathbf{s}_1 \quad \underline{\mathbf{c}}_1]\}+\frac{1}{K_a}tr\{\mathbf{P}^{\dagger}[\mathbf{s}_1 \quad \underline{\mathbf{c}}_1][\mathbf{s}_1 \quad \underline{\mathbf{c}}_1]^{\dagger}\mathbf{P}\}}$$

คำนวณ SINR สำหรับผู้ใช้รายที่ 2

$$\frac{1}{\sigma_n^2+\frac{1}{K_a}tr\{[\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1]^{\dagger}[\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1][\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1]^{\dagger}[\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1]\}+\frac{1}{K_a}tr\{\mathbf{P}^{\dagger}[\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1][\mathbf{s}_1 \quad \mathbf{s}_2 \quad \underline{\mathbf{c}}_1]^{\dagger}\mathbf{P}\}}$$

คำนวณ SINR สำหรับผู้ใช้รายที่ 3

ภาพที่ 14 ตัวอย่างการนับจำนวนการคำนวณของการแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้สำหรับผู้ใช้แบบกลุ่มในกรณีที่ $K_a = 3$

จากภาพที่ 14 การคำนวณ SINR สำหรับผู้ใช้รายที่ 1 นั้นกำหนดให้ $\mathbf{c}_1$ เป็นตัวแทนจากโหนดต่างๆ ของโค้ดบุ๊คแบบโครงสร้างต้นไม้ซึ่งเมื่อทำการแวะผ่านเข้าไปในโครงสร้างต้นไม้จนได้รหัสลายเซ็นสำหรับผู้ใช้รายที่ 1 แล้วเรากำหนดให้ $\mathbf{s}_1$ แทนรหัสลายเซ็นสำหรับผู้ใช้รายที่ 1 ซึ่งเราจะใช้ $\mathbf{s}_1$ นี้ช่วยในการคำนวณ SINR สำหรับผู้ใช้รายที่ 2 สำหรับการคำนวณ SINR สำหรับผู้ใช้รายที่ 2 นั้นเราจะคำนวณเพียงแค่ $\mathbf{c}_1$ เนื่องจาก $\mathbf{s}_1$ นั้นเราได้คำนวณไว้แล้วจากขั้นตอนก่อนหน้า เหตุผลดังกล่าวสามารถใช้อธิบายการคำนวณ SINR สำหรับผู้ใช้รายที่ 3 ได้เช่นกัน ดังนั้นการนับจำนวนการคำนวณของการแบ่งนับวิธีนี้จะเท่ากับ $\frac{1}{K_a}$ ของการคำนวณ SINR แบบปกติ

3. การแบ่งนับรหัสลายเซ็นแบบพีเอเอ็ม

การแบ่งนับกลุ่มรหัสลายเซ็นวิธีนี้มาจากงานวิจัย (Ryan *et al*. 2007) ซึ่งกล่าวไว้ว่าเมื่อช่องสัญญาณป้อนกลับมีอัตราบิตป้อนกลับที่ไม่จำกัดแล้วรหัสลายเซ็นที่เหมาะสมที่สุดจะเป็นเวกเตอร์ลักษณะเฉพาะหรือไอเกนเวกเตอร์ (eigenvector) ที่ให้ค่าลักษณะเฉพาะ (eigenvalue) ต่ำที่สุดของเมทริกซ์แปรผันร่วมของสัญญาณแทรกสอด (interference covariance matrix) งานวิจัยนี้ใช้โค้ดบุ๊คชนิดพีเอเอ็มในการหาเวกเตอร์ใกล้เคียงกับไอเกนเวกเตอร์ดังกล่าว โดยโค้ดบุ๊คแบบพีเอเอ็มนั้นมีลักษณะแตกต่างจากโค้ดบุ๊คแบบเวกเตอร์สุ่มกล่าวคือโค้ดบุ๊คดังกล่าวเป็นสัญญลักษณ์ของสัญญาณพีเอเอ็มสามารถเขียนอธิบายได้ดังนี้ $\mathbf{x} = [x_1, ..., x_N]^{\dagger}$ โดยที่

$x_n \in \{\pm 1, \pm 3, ..., \pm(M-1)\}$ สำหรับ M-ary PAM สำหรับการหาเวกเตอร์ใกล้เคียงนั้นจะอ้างอิงจากวิธีที่เรียกว่า generalized likelihood ratio test (GLRT) แต่มีการลดจำนวนเวกเตอร์ที่ต้องการทดสอบโดยอ้างอิงจากสัญญาณที่รับได้เพื่อต้องการลดจำนวนการคำนวณ โดยเวกเตอร์ที่ต้องนำมาทดสอบนั้นเป็นเวกเตอร์ที่ใกล้เคียงในเชิงมุม (closest in angle) กับเส้นตรงที่เกิดจากการนำเวกเตอร์อินพุตมาคูณกับค่าคงที่ $\lambda_1, \lambda_2, ...$ โดยที่ $0 < \lambda_i \le \lambda^{\max}$ ซึ่งค่าคงที่ $\lambda^{\max}$ สามารถหาได้จากสมการที่ 21

$$\lambda^{\max} = (M + 2N - 2) / \max_n \{y_n\} \qquad (21)$$

โดยกลุ่มของเวกเตอร์ที่ต้องการทดสอบนี้มีชื่อว่า “nearest neighbor set” การหา nearest neighbor set และการหาเวกเตอร์ใกล้เคียงในกรณีทั่วไปสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

<u>M-ary Real-PAM Encoding Algorithm</u> (Ryan *et al*. 2007)

1: **begin**
2: $\mathbf{s} = \operatorname{sgn} \mathbf{y}$ // Store sign for each y_n, where **y** is input vector.
3: $\mathbf{y} = \mathbf{s} \otimes \mathbf{y}$ // Make each y_n positive.
4: $B^{\max} = M + T - 2$
5: $m = \arg\max_t \{y_t\}$
6: $\lambda^{\max} = (M + 2T - 2) / |y_m|$ // Search region: $0 < \lambda < \lambda^{\max}$
7: $V_0 = [\ ]$ // Calculate and store $P(\underline{x})$ boundary points.
8: **for** $n = 1 : N$ **do**
9: **for all** $b \in \{2, 4, ..., M-2\}$ **do**
10: $v = b / y_n$
11: **if** $v < \lambda^{\max}$
12: $V_0 = \{V_0(v, n)\}$
13: **else break**
14: **end for all**
15: **end for**
16: $V = \text{sort}\ (V_0)$ // Sort V_0 in ascending order of v.
17: $V = \{V, (\lambda^{\max}, 0)\}$

18: $\hat{\mathbf{x}} = [11...1]^{\dagger}$ // Initialize data estimate.

19: $\alpha = \hat{\mathbf{x}}^{\dagger}\mathbf{y}$ // Initialize likelihood terms.

20: $\beta = \|\hat{\mathbf{x}}\|^2$

21: $L = \alpha^2 / \beta$

22: $\lambda = V(1,1)/2$

23: **for** k=1 : $|V_0| - 1$ **do**

24: $n = V(k,2)$

25: $\alpha = \alpha + 2y_n$ // Update likelihood terms.

26: $\beta = \beta + 4\hat{x}_n + 4$

27: $\hat{x}_n = \hat{x}_n + 2$ // Update **x**

28: **if** $\alpha^2 / \beta > L$ // If better **x** found

29: $L = \alpha^2 / \beta$ // Update likelihood

30: $\lambda = (V(k,1) + V(k+1,1))/2$ // Store point in $P(\underline{x})$

31: **end if**

32: **end for**

33: **return** $\hat{\mathbf{x}}^{opt} = \mathbf{s} \otimes NN(\lambda \mathbf{y})$

ฟังก์ชัน $NN(\cdot)$ ในบรรทัดที่ 33 คือการปัดตัวเลขให้เป็นจำนวนเต็มที่เป็นสัญลักษณ์ของ PAM

สำหรับการแบ่งนับรหัสลายเซ็นในระบบซีดีเอ็มเอนั้นจะใช้วิธีการหาเวกเตอร์ใกล้เคียงกับไอเกนเวกเตอร์ดังที่กล่าวมาข้างต้น ในกรณีของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวนั้นสามารถเขียนอธิบายเป็นซูโค้ดได้ดังนี้

PAM Encoding for Singleuser Algorithm

1: Start with $\mathbf{S}$ matrix of interfering signatures

2: $\mathbf{R} = \mathbf{S}^{\dagger}\mathbf{S}$ \\ interference covariance matrix

3: $\mathbf{s}^{opt}$ = eigenvector with minimum eigenvalue of $\mathbf{R}$.

4: $\mathbf{y} = \mathbf{s}^{opt}$

5: *M*-ary Real-PAM Encoding Algorithm

6: $\mathbf{s}_1 = \hat{\mathbf{x}}^{opt}$

7: $\mathbf{s}_1 = \frac{\mathbf{s}_1}{\|\mathbf{s}_1\|}$

8: **return** $\mathbf{s}_1$ the optimal signature for adapting user

M-ary Real-PAM Encoding Algorithm ในขั้นตอนที่ 5 คือการเรียกใช้วิธีการหาเวกเตอร์ที่ใกล้เคียงกับเวกเตอร์อินพุตโดยกำหนดให้เวกเตอร์อินพุตคือไอเกนเวกเตอร์ที่ให้ค่าลักษณะเฉพาะต่ำที่สุดของเมทริกซ์แปรผันร่วมของสัญญาณแทรกสอด $\mathbf{S}$ ในขั้นตอนที่ 1 สำหรับวิธีการแบ่งนับแบบพีเอเอ็มนี้โค้ดบุ๊คจะมีลักษณะแตกต่างจากโค้ดบุ๊คสำหรับการแบ่งนับวิธีอื่นๆ โดยเวกเตอร์ในโค้ดบุ๊คเป็นเวกเตอร์ที่ค่าประมิติเป็นสัญลักษณ์พีเอเอ็มไม่ได้เป็นเวกเตอร์สุ่มโดยจำนวนสัญลักษณ์ M เท่ากับ $2^{B/N}$ สำหรับการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มโดยใช้โค้ดบุ๊คชนิดพีเอเอ็มนั้น เราใช้วิธีหารหัสลายเซ็นทีละตัวเหมือนการแบ่งนับด้วยวิธีโครงสร้างต้นไม้แตกต่างกันตรงที่ในแต่ละรอบเราจะนำรหัสลายเซ็นทุกตัวที่ได้จากรอบก่อนหน้านี้เข้ามารวมเป็นสัญญาณแทรกสอดในเมทริกซ์ $\mathbf{P}$ เพื่อใช้กระบวนการ SVD สำหรับการหาไอเกนเวกเตอร์สำหรับรอบต่อไป การแบ่งนับกลุ่มรหัสลายเซ็นดังกล่าวเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

PAM Encoding for Multiuser Algorithm

1: $\mathbf{S} = [\ ]$

2: **for** $i = 1 : K_a$ **do**

3: Start with $\mathbf{P}$ matrix of interfering signatures.

4: $\mathbf{R} = \mathbf{P}^{\dagger}\mathbf{P}$ \\ interference covariance matrix

5: $\mathbf{s}^{opt}$ = eigenvector with minimum eigenvalue of $\mathbf{R}$.

6: $\mathbf{y} = \mathbf{s}^{opt}$

7: *M*-ary Real-PAM Encoding Algorithm

8: $\mathbf{s}_1 = \hat{\mathbf{x}}^{opt}$

9: $\mathbf{s}_1 = \frac{\mathbf{s}_1}{\|\mathbf{s}_1\|}$

10: $\mathbf{P} = [\mathbf{P} \quad \mathbf{s}_1]$

11: $\mathbf{S} = [\mathbf{S} \quad \mathbf{s}_1]$

12: **end for**

13: **return** the optimal set of signature **S**.

สำหรับโค้ดบุ๊คชนิดพีเอเอ็มงานวิจัย (Ryan, *et al*. 2007) อธิบายไว้ว่าใช้การคำนวณเท่ากับกับจำนวนของเวกเตอร์ใน nearest neighbor set สำหรับผู้ใช้รายเดียวในกรณีที่ผู้ใช้หลายรายเราจะนำจำนวนเวกเตอร์ใน nearest neighbor set ที่ได้ในแต่ละรอบมารวมกันเพื่อคิดเป็นจำนวนของการคำนวณที่ต้องใช้ทั้งหมด ซึ่งจากวิธีการนับจำนวนการคำนวณข้างต้นจะพบว่าการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มต้องการจำนวนการคำนวณที่เพิ่มขึ้นอย่างเชิงเส้นกับบิตป้อนกลับ B

4. การแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดี

การสร้างโค้ดบุ๊คที่มีลักษณะเป็นโครงสร้างต้นไม้เคดีนั้นเราจะใช้โค้ดบุ๊คแบบเวกเตอร์สุ่มในสมการที่ 12 สำหรับผู้ใช้รายเดียวและ 14 สำหรับผู้ใช้แบบกลุ่ม การสร้างโค้ดบุ๊คต้นไม้เคดีเริ่มจากการเรียงค่าในมิติที่ 1 ของเวกเตอร์ทุกตัวในเซตหลังจากนั้นเลือกค่าที่ใกล้เคียงกับค่ากลาง (median) มากที่สุดเป็นจุดกึ่งกลางเรียกว่า pivot หลังจากนั้นใช้ค่า pivot แบ่งเวกเตอร์ออกเป็นสองโหนดโดยให้ค่าที่มากกว่า pivot เป็นโหนดหนึ่งและให้ค่าที่น้อยกว่า pivot เป็นอีกโหนดหนึ่ง เมื่อได้เวกเตอร์สองโหนดแล้วก็แบ่งเวกเตอร์ในกลุ่มย่อยนั้นอีกครั้งโดยใช้ค่าในมิติถัดไป ทำซ้ำกระบวนการนี้ไปเรื่อยๆจนกว่าแต่ละโหนดมีเวกเตอร์เพียงตัวเดียวโดยเรียกโหนดเหล่านั้นว่าโหนดใบ การสร้างโค้ดบุ๊คดังกล่าวสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

Kd-Tree Construction Algorithm (Bently, 1980)

1: Start with set of *k*-dimensional vectors.

2: Store all entries at the root of a binary tree.

3: Set $n = 1$ where n is the index of element in an N-dimensional vector.

4: *k*d-tree = build_tree(root node, *n*)

5: **return** *k*d-tree

Function : *k*d-tree = build_tree(node, *n*)

1: **if** Node contains only one vector. **Then**

2: **return**

3: **else**

4: **if** $n > N$ **then**

5: $n = 1$

6: **end if**

7: Sort point list by the *n*th dimension.

8: Select the median as pivot element.

9: Use pivot to divide vectors in two groups.

10: $n = n + 1$

11: build_tree(left child node, *n*)

12: build_tree(right child node, *n*)

13: **end if**

สำหรับการค้นหาเวกเตอร์ตัวที่เหมาะสมในโค้ดบุ๊คแบบต้นไม้เคดีนั้นจะใช้วิธีการแวะผ่านเช่นเดียวกับโค้ดบุ๊คแบบโครงสร้างต้นไม้ในหัวข้อที่ 3 แตกต่างกันตรงที่การแบ่งนับแบบต้นไม้เคดีนั้นใช้การแทรกสอด (interference) เป็นฟังก์ชันเป้าหมาย สำหรับการแวะผ่านแต่ละครั้งนั้นจะคำนวณการแทรกสอดจากมิติที่ใช้สร้างต้นไม้เคดีเท่านั้น วิธีดังกล่าวสามารถเขียนอธิบายเป็นซูโดโค้ดได้ดังนี้

Kd-Tree Encoding for Singleuser Algorithm

1: Start at the root node of a Kd-tree codebook with interfering signature matrix $\mathbf{S}$.

2: Set $n = 1$ where n is the index of element in an *N*-dimensional vector.

3: **while** not a leaf node. **do**

4: **if** $n > N$ **then**

5: $n = 1$

6: **end if**

7: $\mathbf{p}_L^{(n)} = [0 \quad 0 \quad \ldots \quad p_{L,n+1} \quad \ldots \quad 0 \quad 0]^{\dagger}$ where $\mathbf{p}_L$ is the vector of the left child node.

8: $\mathbf{p}_R^{(n)} = [0 \quad 0 \quad \ldots \quad p_{R,n+1} \quad \ldots \quad 0 \quad 0]^{\dagger}$ where $\mathbf{p}_R$ is the vector of the right child node.

9: **if** $\mathbf{p}_L^{(n)\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{p}_L^{(n)} < \mathbf{p}_R^{(n)\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{p}_R^{(n)}$ **then**

10: Move to the left child node.

11: **else**

12: Move to right child node.

13: **end if**

14: $n = n + 1$

15: **end while**

16: **s** = vector from leaf node

17: $I = \mathbf{s}^{\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{s}$

18: $n = 1$

19: **s** = check_interference(root node, n, **S**, I, **s**)

20: **return s**

โดยฟังก์ชัน check_interference ในขั้นตอน 19 เป็นการตรวจสอบการแทรกสอดที่คำนวณได้จากโหนดใบโดยการเปรียบเทียบกับการแทรกสอดจาก pivot อื่นๆ โดยสามารถเขียนซูโดโค้ดอธิบายได้ดังนี้

Function : check_interference(node, n, **S**, I, **s**)

1: **if** leaf node **then**

2: $\mathbf{s}_c$ = vector of a leaf node

3: $I_c = \mathbf{s}_c^{\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{s}_c$

4: **if** $I < I_c$ **then**

5: $I = I_c$

6: $\mathbf{s} = \mathbf{s}_c$

7: **return s**

8: **end if**

9: **end if**

10: $\mathbf{p}_c^{(n)} = [0 \quad 0 \quad \ldots \quad p_{c,n} \quad \ldots \quad 0 \quad 0]$ where $\mathbf{p}_c$ is the vector of the current node.

11: $I_p = \mathbf{p}_c^{(n)\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{p}_c^{(n)}$

12: **if** $I_p < I$ **then**

13: **if** $n > N$ **then**

14: $n = 1$

15: **end if**

16: **if** $\mathbf{p}_L^{(n)\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{p}_L^{(n)} < \mathbf{p}_R^{(n)\dagger}\mathbf{S}\mathbf{S}^{\dagger}\mathbf{p}_R^{(n)}$ **then**

17: $n = n + 1$

18: check_interference(left-child node, n, **S**, I, **s**)

19: **else**

20: $n = n + 1$

21: check_interference(right-child node, n, **S**, I, **s**)

22: **end if**

23: **else**

24: $n = n + 1$

25: check_interference(left-child node, n, **S**, I, **s**)

26: check_interference(right-child node, n, **S**, I, **s**)

27: **end if**

สำหรับการแบ่งนับกลุ่มรหัสลายเซ็นสำหรับผู้ใช้หลายรายโดยโค้ดบุ๊คแบบต้นไม้เคดีนั้นมีหลักการเหมือนกับการแบ่งนับแบบพีเอเอ็มคือคิดว่ารหัสลายเซ็นทุกตัวที่ได้จากรอบที่ได้จากรอบก่อนหน้านี้เป็นสัญญาณแทรกสอดสำหรับการหารหัสลายเซ็นในรอบต่อไปดังซูโดโค้ดต่อไปนี้

Kd-Tree Encoding for Multiuser Algorithm

1: $\mathbf{S} = [\ ]$

2: **for** $i = 1 : K_a$ **do**

3: Start at the root node of a Kd-tree codebook with interfering signature matrix $\mathbf{P}$.

4: Kd-Tree Encoding for Singleuser Algorithm

5: $\mathbf{P} = [\mathbf{P} \quad \mathbf{s}]$

6: $\mathbf{S} = [\mathbf{S} \quad \mathbf{s}]$

7: **end for**

8: **return** the optimal set of signature **S**.

การแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีนี้ใช้วิธีการแวะผ่านโหนดสำหรับการค้นหาเวกเตอร์ตัวที่เหมาะสมเช่นเดียวกับการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้ต่างกันตรงที่การแบ่งนับแบบต้นไม้เคดีนั้นการแวะผ่านโหนดแต่ละครั้งนั้นการคำนวณจะใช้เพียงค่าประจำมิติค่าเดียวในการคำนวณซึ่งจะเห็นได้จากขั้นตอนที่ 7-9 ของอัลกอริทึม Kd-Tree Encoding for Singleuser Algorithm ซึ่งช่วยในการลดจำนวนการคำนวณได้เป็นอย่างดี

ผลและวิจารณ์

ผล

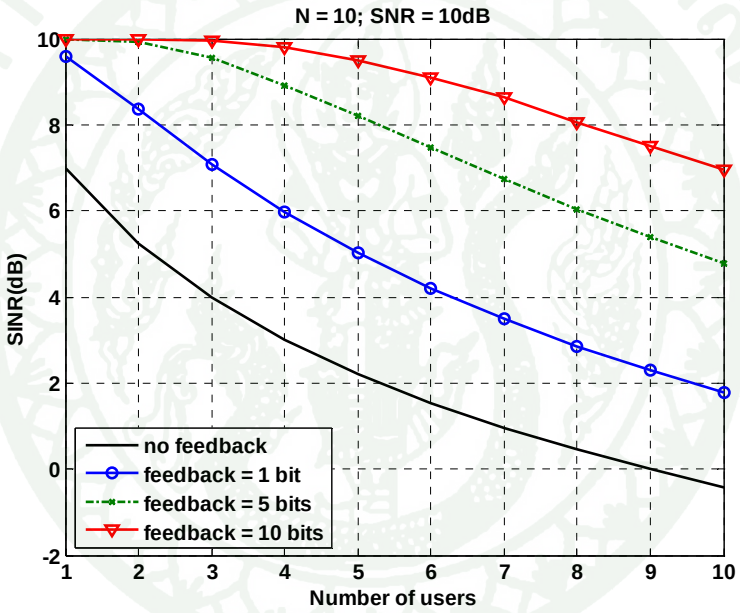
การทดสอบสมรรถนะ (performance) ของการแบ่งนับรหัสลายเซ็นแต่ละวิธีนั้นงานวิจัยนี้ใช้โปรแกรม MATLAB ในการจำลองระบบ โดยใช้วิธีที่เรียกว่า monte carlo simulation ซึ่งวิธีการ monte carlo simulation นี้เป็นการเปลี่ยนการคำนวณทางคณิตศาสตร์ที่ยุ่งยากซับซ้อนให้ง่ายขึ้นโดยการนับจำนวนครั้งแล้วหาค่าเฉลี่ย เนื่องจากการสร้างรหัสลายเซ็นโดยใช้โปรแกรม MATLAB นั้นรหัสลายเซ็นที่ได้เป็นเวกเตอร์ของจำนวนจริงที่มี N มิติโดยที่ค่าในแต่ละมิติเป็นตัวแปรสุ่ม (random variable) ที่มีการกระจายแบบปกติ (normal distribution) โดยมีค่าเฉลี่ย (mean) เป็น 0 และมีค่าความแปรปรวน (variance) เท่ากับ $\frac{1}{N}$ นอกจากรหัสลายเซ็นที่กล่าวมาข้างต้น ช่องสัญญาณ บิตข้อมูลและสัญญาณรบกวนต่างก็เป็นตัวแปรสุ่มทั้งสิ้น ดังนั้นหากต้องการหาความคาดหวัง (expectation) ของตัวแปรสุ่มดังกล่าวจึงต้องมีการหาค่าเฉลี่ยด้วยวิธีข้างต้น

สำหรับการทดลองในงานวิจัยนี้เริ่มจากการเปรียบเทียบระบบซีดีเอ็มเอที่ใช้ช่องสัญญาณป้อนกลับในการปรับรหัสลายเซ็นกับระบบซีดีเอ็มเอที่ไม่ใช้ช่องสัญญาณป้อนกลับในการปรับรหัสลายเซ็นพร้อมทั้งศึกษาผลกระทบของจำนวนผู้ใช้ที่มีต่อสมรรถนะของระบบ ขั้นตอนต่อไปจึงทดลองปรับรหัสลายเซ็นของผู้ใช้รายเดียวด้วยวิธีการแบ่งนับแบบต่างๆในช่องสัญญาณที่เป็นอุดมคติ (ideal channel) หลังจากนั้นจึงนำเสนอการปรับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มโดยทำการทดลองที่พารามิเตอร์ที่แตกต่างกันเพื่อตรวจสอบแนวโน้มของสมรรถนะของการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างทั้งในช่องสัญญาณที่เป็นอุดมคติและช่องสัญญาณจางหายแบบเลือกความถี่ (frequency selective fading channel)

1. การเปรียบเทียบระบบซีดีเอ็มเอที่มีและไม่มีช่องสัญญาณป้อนกลับ

การเปรียบเทียบระบบซีดีเอ็มเอที่มีและไม่มีช่องสัญญาณป้อนกลับนั้นใช้ผลการวิเคราะห์สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มจากงานวิจัย (Santipach and Honig 2005) โดยเปรียบเทียบระบบซีดีเอ็มเอที่ไม่มีการปรับรหัสลายเซ็นกับระบบซีดีเอ็มเอที่ใช้บิตป้อนกลับ

จำนวน 1, 5 และ 10 บิตในการปรับรหัสลายเซ็นซึ่งได้ผลดังภาพที่ 15 ซึ่งเป็นการจำลองระบบซีดีเอ็มเอที่มีอัตราขยายประมวลผล (processing gain หรือ N) เท่ากับ 10 จากภาพที่ 15 จะเห็นว่าระบบซีดีเอ็มเอที่ไม่มีช่องสัญญาณป้อนกลับนั้นมีสมรรถนะที่แย่กว่าระบบซีดีเอ็มเอที่มีช่องสัญญาณป้อนกลับและยังพบว่าเมื่อจำนวนบิตป้อนกลับเพิ่มขึ้นสมรรถนะของระบบก็จะเพิ่มขึ้นตามไปด้วยซึ่งหมายความว่าสมรรถนะของระบบนั้นขึ้นกับจำนวนบิตป้อนกลับด้วย นอกจากนั้นเมื่อพิจารณาจากภาพที่ 15 ยังพบว่าจำนวนของผู้ใช้ในระบบมีผลต่อสมรรถนะกล่าวคือเมื่อจำนวนผู้ใช้ในระบบเพิ่มมากขึ้นสมรรถนะของระบบจะลดลงเนื่องจากเมื่อจำนวนผู้ใช้ในระบบเพิ่มมากขึ้นกำลังงานของการแทรกสอดก็จะเพิ่มขึ้นตามไปด้วยนั่นเอง



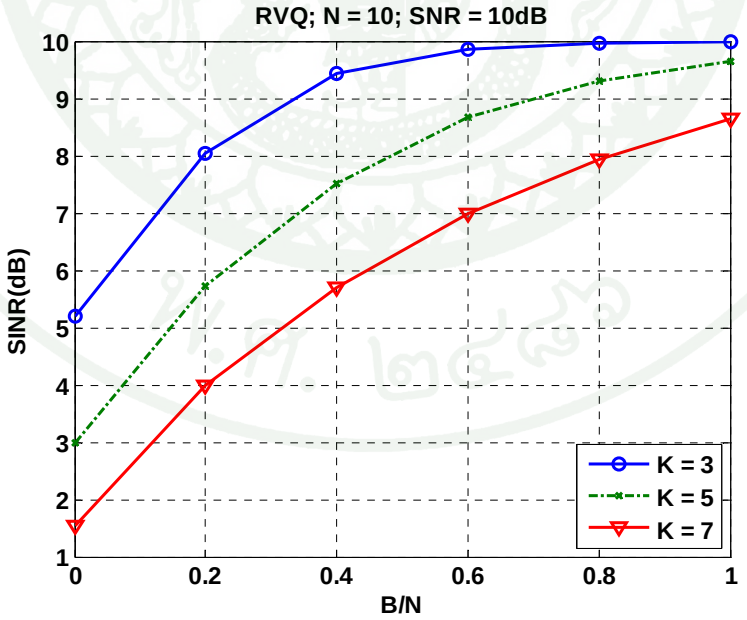
ภาพที่ 15 แสดงการเปรียบเทียบระบบซีดีเอ็มเอที่มีและไม่มีช่องสัญญาณป้อนกลับ

2. การแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียว

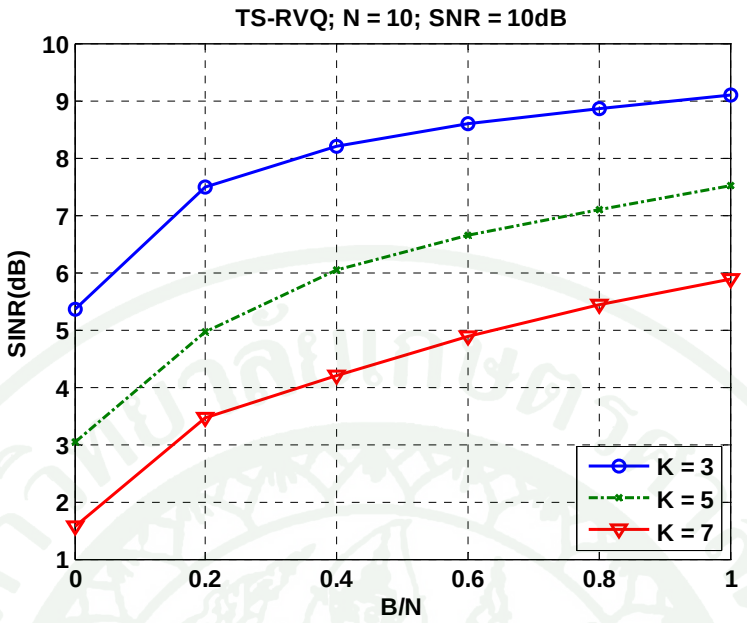
การทดลองแบ่งนับรหัสลายเซ็นสำหรับผู้รายเดียวนั้นผู้วิจัยได้ทดลองใช้การแบ่งนับเวกเตอร์สี่วิธีได้แก่การแบ่งนับเวกเตอร์แบบสุ่ม, การแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้, การแบ่งนับเวกเตอร์แบบพีเอเอ็มและการแบ่งนับเวกเตอร์แบบต้นไม้เคดี โดยได้ทดลองที่ระบบซีดีเอ็มเอที่มี N เท่ากับ 10, อัตราส่วนของสัญญาณต่อสัญญาณรบกวนภูมิหลัง (background SNR) เท่ากับ 10 โดยกำหนดให้ระบบมีผู้ใช้เท่ากับ 3, 5 และ 7 สำหรับทุกวิธีการแบ่งนับรหัสลายเซ็น ซึ่ง

ในระบบที่ใช้งานจริงนั้นค่าพารามิเตอร์ต่างๆจะมีค่ามากกว่าค่าที่ใช้ในการทดลองตัวอย่างเช่น ระบบมี N เท่ากับ 128 หรือ 256 แต่เนื่องจากข้อจำกัดทางด้านทรัพยากรคอมพิวเตอร์ทำให้ไม่สามารถทำการทดลองที่พารามิเตอร์ดังกล่าวได้ อย่างไรก็ตามการทดลองที่ระบบขนาดเล็กสามารถใช้ทำนายสมรรถนะของระบบที่มีขนาดใหญ่ได้เป็นอย่างดี (Santipach and Honig 2005)

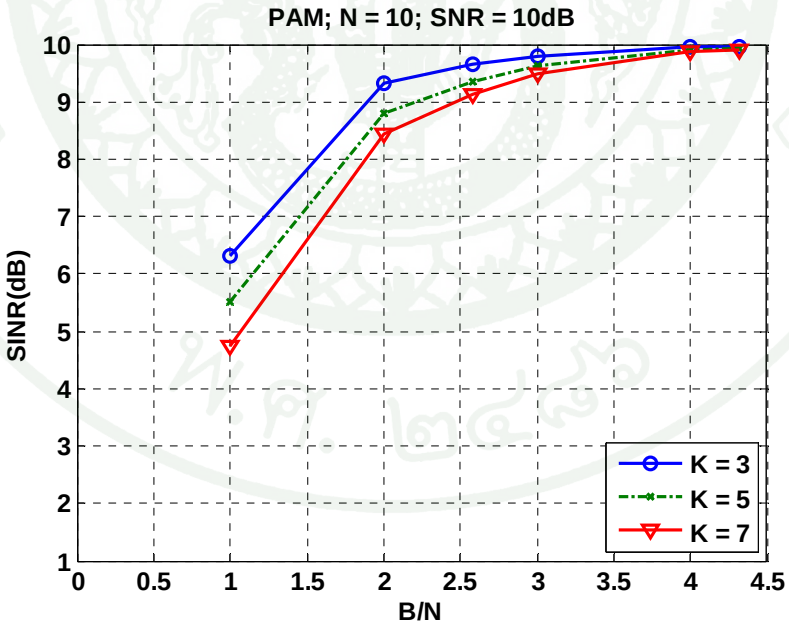
ภาพที่ 16, 17, 18 และ 19 แสดงสมรรถนะของการแบ่งนับเวกเตอร์แบบสุ่ม การแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้ การแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มและการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีตามลำดับ โดยแต่ละภาพที่แสดงนั้นเป็นกราฟระหว่างอัตราส่วนของบิตป้อนกลับต่ออัตราขยายประมวลผล (normalized feedback bit หรือ B/N) และ SINR ในหน่วยเดซิเบล จากภาพทั้งสี่จะเห็นว่าเมื่อจำนวนบิตป้อนกลับเพิ่มขึ้นสมรรถนะของผู้ใช้รายที่ต้องการปรับรหัสลายเซ็นก็จะเพิ่มขึ้นตามไปด้วยซึ่งสอดคล้องกับผลการทดลองในหัวข้อที่ 1 นอกจากนั้นจำนวนผู้ใช้ในระบบมีผลต่อสมรรถนะของผู้ใช้ที่ต้องการปรับรหัสลายเซ็นกล่าวคือเมื่อจำนวนผู้ใช้ในระบบเพิ่มมากขึ้นสมรรถนะของผู้ใช้ดังกล่าวจะลดลงซึ่งซึ่งตรงตามสมมติฐานที่ตั้งไว้ ทั้งนี้เนื่องจากผู้ใช้รายอื่นประพฤติตัวเป็นสัญญาณแทรกสอดเข้ามาหรือกล่าวได้ว่าผู้ใช้รายอื่นเป็นภาระ (load) ของระบบนั่นเอง



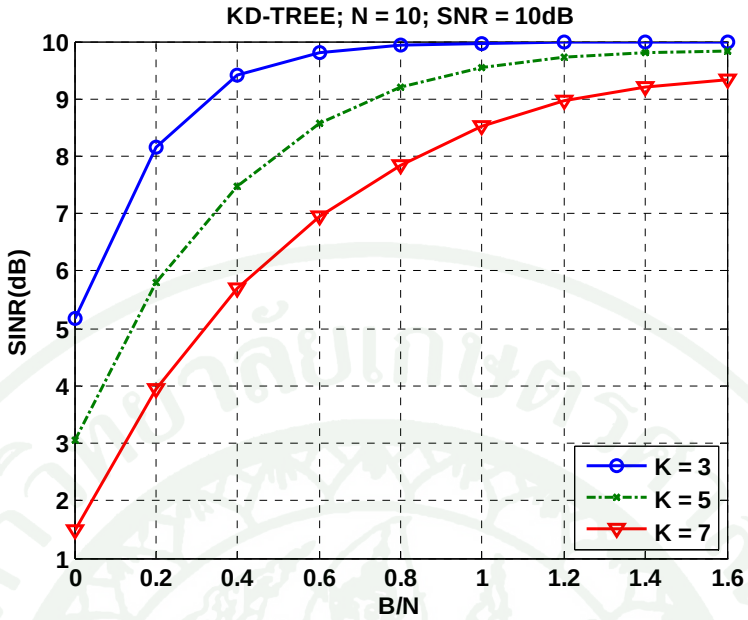
ภาพที่ 16 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มในระบบซีดีเอ็มเอสำหรับผู้ใช้รายเดียว



ภาพที่ 17 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้ในระบบซีดีเอ็มเอสำหรับผู้ใช้รายเดียว

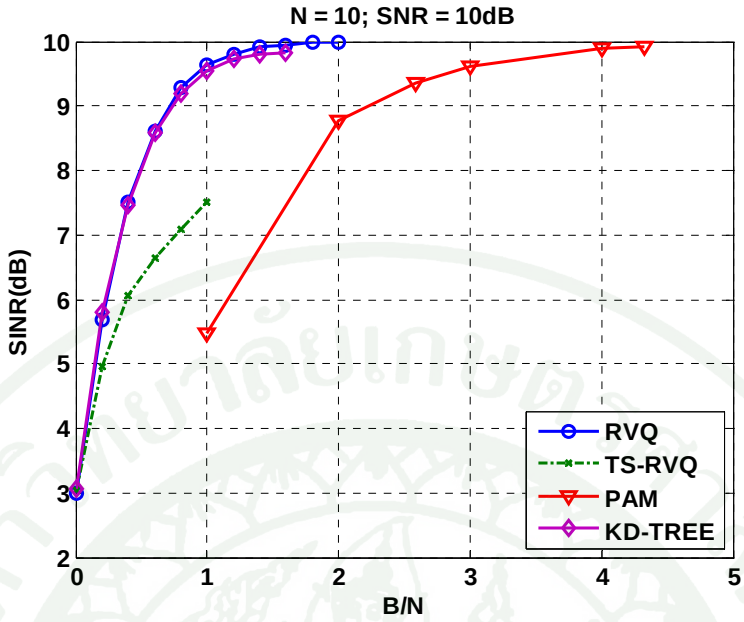


ภาพที่ 18 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มในระบบซีดีเอ็มเอสำหรับผู้ใช้รายเดียว



ภาพที่ 19 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีในระบบซีดีเอ็มเอสำหรับผู้ใช้รายเดียว

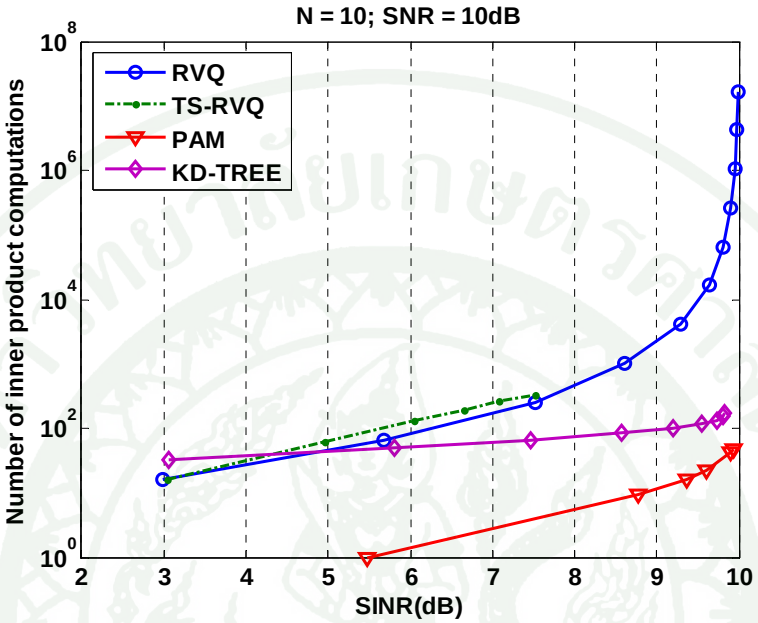
ภาพที่ 20 และ 21 แสดงการเปรียบเทียบการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างๆสำหรับผู้ใช้รายเดียวจากภาพที่ 20 จะเห็นว่าทุกวิธีการแบ่งนับให้สมรรถนะเพิ่มขึ้นเมื่อจำนวนบิตป้อนกลับเพิ่มขึ้น การแบ่งนับเวกเตอร์แบบสุ่มให้สมรรถนะดีที่สุดเมื่อพิจารณาที่ B/N มีค่าเท่ากัน เมื่อพิจารณาที่ B/N เท่ากับ 1 พบว่าการแบ่งนับแบบต้นไม้เคดีนั้นให้สมรรถนะที่ใกล้เคียงกับการแบ่งนับเวกเตอร์แบบสุ่มอย่างมากส่วนการแบ่งนับเวกเตอร์แบบพีเอเอ็มนั้นไม่สามารถแสดงสมรรถนะได้ที่ B/N น้อยกว่า 1 เนื่องจากที่ค่าดังกล่าวนั้นไม่สามารถหาสัญลักษณ์สำหรับพีเอเอ็มได้ สำหรับการแบ่งนับแบบโครงสร้างต้นไม้นั้นให้สมรรถนะที่แย่กว่าการแบ่งนับเวกเตอร์แบบสุ่มและแบบต้นไม้เคดีแต่ให้สมรรถนะที่ดีกว่าพีเอเอ็มที่ B/N เท่ากับ 1 อย่างไรก็ตามเราไม่สามารถหาสมรรถนะของการแบ่งนับแบบโครงสร้างต้นไม้ที่ B/N มากกว่า 1 ได้เนื่องจากการแบ่งนับวิธีนี้ใช้วิธี GLA ในการจัดโครงสร้างต้นไม้ซึ่งวิธี GLA นี้ไม่สามารถทำงานได้ดีเมื่อเวกเตอร์ในโค้ดบุ๊คมีจำนวนเพิ่มมากขึ้น เนื่องจากเมื่อเวกเตอร์ในโค้ดบุ๊คมีจำนวนมากขั้นโปรแกรมคอมพิวเตอร์ไม่สามารถที่จะทำให้ระยะทางยูคลิดในขั้นตอนที่ 7 ของวิธี GLA มีค่าคงที่ได้



ภาพที่ 20 เปรียบเทียบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวด้วยวิธีต่างๆ

ภาพที่ 21 แสดงการเปรียบเทียบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวด้วยวิธีต่างๆในเชิงของจำนวนการคำนวณโดยงานวิจัยนี้ใช้จำนวนครั้งของการกระทำ inner product ของเวกเตอร์ในโค้ดบุ๊คและเวกเตอร์แต่ละตัวในเมทริกซ์ของสัญญาณแทรกสอด ในการเปรียบเทียบโดยภาพที่ 21 เป็นกราฟระหว่าง SINR และจำนวนของการกระทำ inner product จากภาพที่ 21 เมื่อพิจารณาที่ SINR เท่ากันจะเห็นว่าการแบ่งนับแบบพีเอเอ็มนั้นใช้จำนวนการคำนวณน้อยที่สุดแต่เมื่อย้อนกลับไปพิจารณาภาพที่ 20 จะเห็นว่าการแบ่งนับแบบพีเอเอ็มนั้นต้องการจำนวนบิตป้อนกลับที่สูงมากเพื่อที่จะได้ SINR ที่ต้องการ พิจารณาการแบ่งนับเวกเตอร์แบบสุ่มการแบ่งนับเวกเตอร์ชนิดนี้ต้องการจำนวนการคำนวณที่สูงมากเนื่องจากการแบ่งนับเวกเตอร์ด้วยวิธีนี้ต้องทดสอบเวกเตอร์ทุกตัวในโค้ดบุ๊คเพื่อค้นหาเวกเตอร์ตัวที่เหมาะสมดังนั้นจำนวนการคำนวณของการแบ่งนับเวกเตอร์ด้วยวิธีนี้จึงเพิ่มขึ้นแบบเอกโพเนนเชียลกับจำนวนบิตป้อนกลับ สำหรับการแบ่งนับแบบโครงสร้างต้นไม้นั้นวิธีนี้ใช้สมรรถนะที่ด้อยกว่าการแบ่งนับเวกเตอร์แบบสุ่มและแบบต้นไม้เคดีและใช้จำนวนการคำนวณที่มากกว่าการแบ่งนับอีกสามวิธีที่เหลือเมื่อพิจารณาที่ SINR ที่เท่ากัน ดังนั้นการแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้จึงไม่เหมาะสมสำหรับการปรับรหัสลายเซ็นสำหรับผู้ใช้รายเดียว สำหรับการแบ่งนับแบบต้นไม้เคดีนั้นให้ SINR ที่สูงเมื่อใช้จำนวนบิตป้อนกลับน้อยและใช้จำนวนการคำนวณที่น้อยกว่าการแบ่งนับเวกเตอร์แบบสุ่มมากเมื่อพิจารณาที่ SINR ที่เท่ากันดังนั้นการแบ่งนับแบบต้นไม้เคดีจึงเป็นการแบ่งนับที่มีประสิทธิภาพสำหรับการ

ปรับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวเนื่องจากใช้จำนวนบิตป้อนกลับที่น้อยและใช้จำนวนการคำนวณที่ไม่สูงมาก



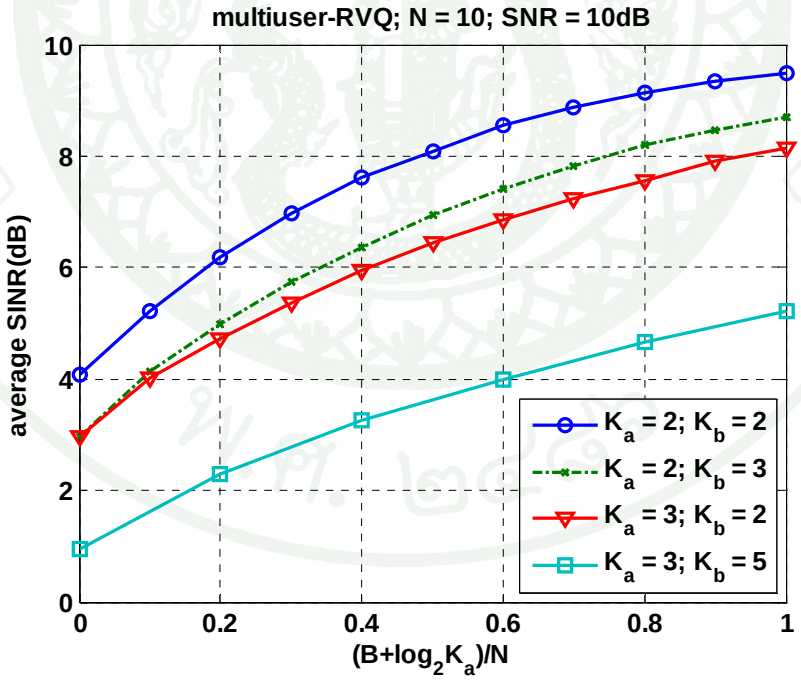
ภาพที่ 21 เปรียบเทียบจำนวนการคำนวณของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวด้วยวิธีต่างๆ

3. การแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่ม

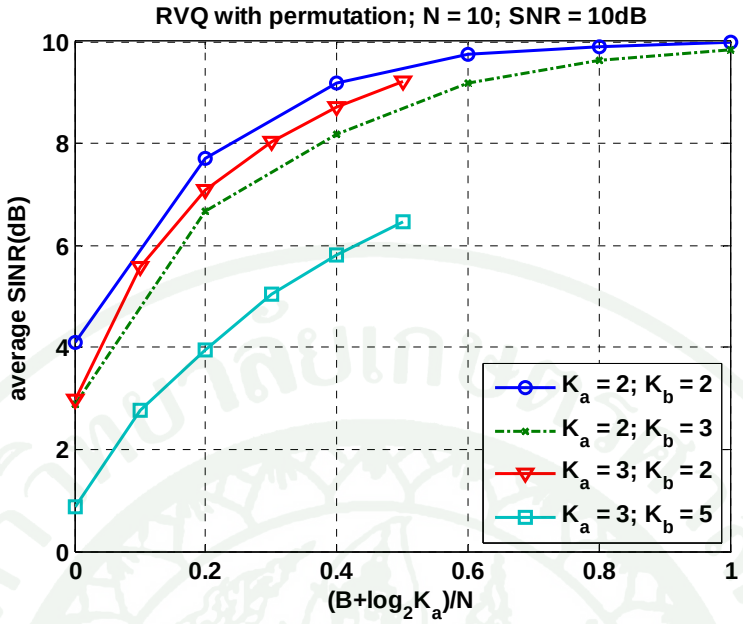
การแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มนั้นงานวิจัยนี้ได้ทดลองใช้การแบ่งนับเวกเตอร์ 5 วิธีได้แก่การแบ่งนับเวกเตอร์สุ่ม, การแบ่งนับเวกเตอร์สุ่มแบบเรียงสับเปลี่ยน, การแบ่งนับแบบโครงสร้างต้นไม้, การแบ่งนับแบบพีเอเอ็มและการแบ่งนับแบบต้นไม้เคดี สำหรับการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มนั้นผู้วิจัยได้ทำการทดลองที่จำนวนผู้ใช้ที่ต้องการปรับรหัสลายเซ็นและจำนวนผู้ใช้ที่ประพฤติตัวเป็นสัญญาณแทรกสอดที่ค่าต่างกันดังนี้ $K_a = 2$; $K_b = 2$, $K_a = 2$; $K_b = 3$, $K_a = 3$; $K_b = 2$ และ $K_a = 3$; $K_b = 5$ ตามลำดับทั้งนี้เพื่อต้องการศึกษาแนวโน้มของสมรรถนะเมื่อจำนวนของผู้ใช้ในระบบเปลี่ยนไป

ภาพที่ 22, 23, 24, 25 และ 26 เป็นผลการทดลองแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆโดยแต่ละภาพเป็นการเปรียบเทียบระหว่างอัตราส่วนของบิตป้อนกลับต่อผู้ใช้ต่อ

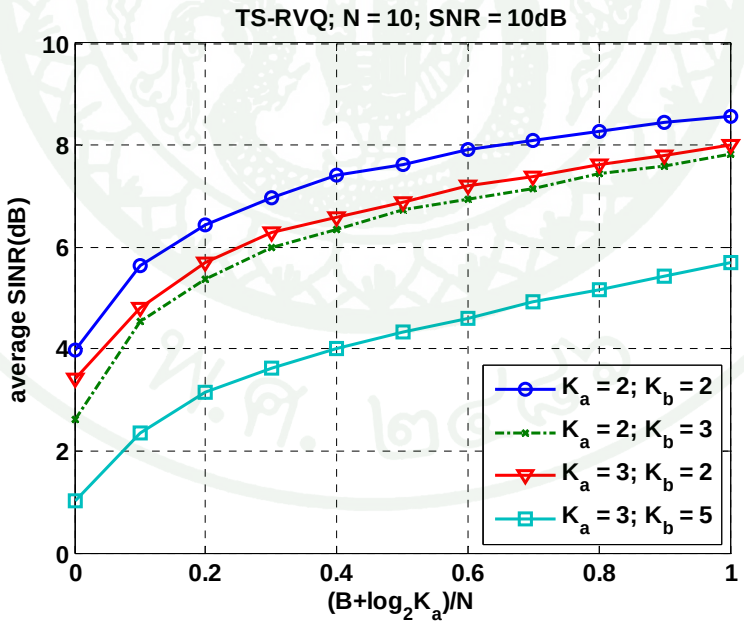
อัตราขยายประมวลผลหรือ $(B+\log_2 K_a)/N$ และ SINR เฉลี่ยของผู้ใช้ที่ต้องการปรับรหัสลายเซ็นทั้งหมด พิจารณาแต่ละภาพจะเห็นว่าเมื่อจำนวนผู้ใช้รวมในระบบเพิ่มขึ้นสมรรถนะของระบบจะลดลง พิจารณาภาพที่ 23, 24 และ 26 ที่จำนวนผู้ใช้รวมของระบบเท่ากับ 5 สมรรถนะของระบบที่ $K_a = 3; K_b = 2$ ดีกว่า สมรรถนะของระบบที่ $K_a = 2; K_b = 3$ อย่างเห็นได้ชัดและภาพที่ 25 สมรรถนะของระบบที่ที่ $K_a = 3$ ดีกว่าสมรรถนะของระบบที่ $K_a = 2$ เล็กน้อยเนื่องจากระบบที่ $K_a = 3$ นั้นผู้ใช้ที่ต้องการปรับรหัสลายเซ็นในระบบมีมากกว่า ทั้งนี้เนื่องจากการปรับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มนั้นรหัสลายเซ็นของผู้ใช้ละรายจะต้องรบกวนกันน้อยที่สุดดังนั้นเมื่อจำนวนผู้ใช้ที่ต้องการปรับรหัสลายเซ็นมากกว่าก็จะสามารถปรับรหัสลายเซ็นให้หลีกเลี่ยงการรบกวนกันได้มากกว่าระบบที่มีผู้ใช้จำนวนเท่ากันแต่มีผู้ใช้ที่ต้องการปรับรหัสลายเซ็นน้อยกว่า ภาพที่ 22 สมรรถนะของระบบที่ที่ $K_a = 3$ แย่กว่าสมรรถนะของระบบที่ $K_a = 2$ เนื่องจากการแบ่งนับเวกเตอร์สุ่มสำหรับผู้ใช้หลายรายนั้นใช้เซตของรหัสลายเซ็นในการทดสอบ SINR ดังนั้นจึงมีความอิสระในการเลือกรหัสลายเซ็นจึงมีน้อยกว่าการแบ่งนับด้วยวิธีอื่นซึ่งอาจเป็นสาเหตุที่ทำให้สมรรถนะของระบบที่มี K_a มากกว่าแย่กว่าสมรรถนะของระบบที่มี K_a น้อยกว่า



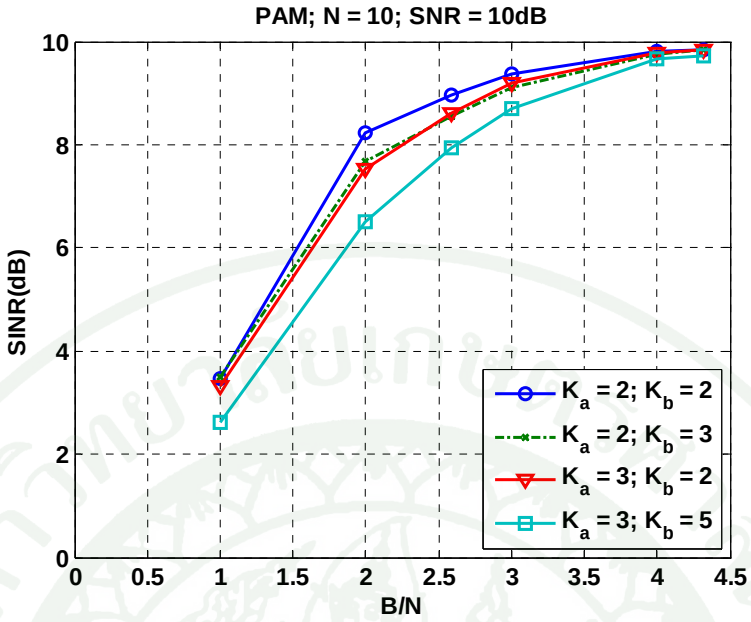
ภาพที่ 22 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มสำหรับผู้ใช้แบบกลุ่ม



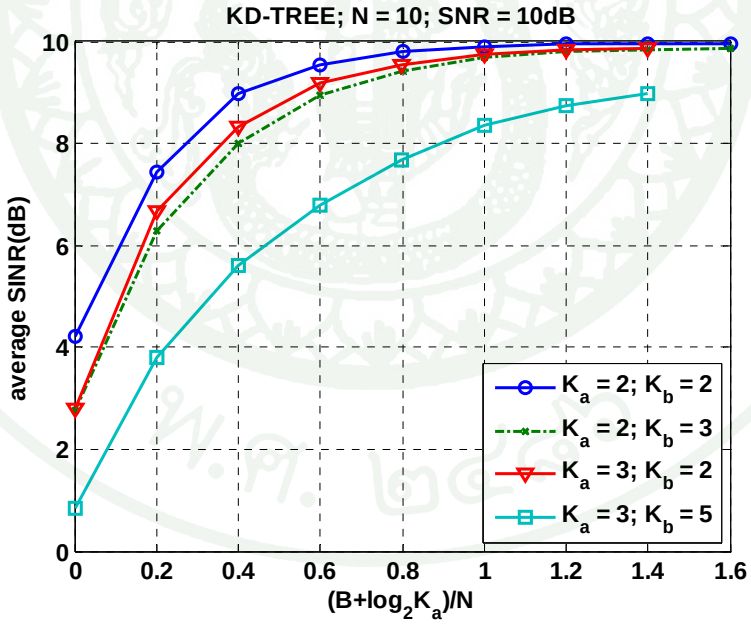
ภาพที่ 23 สมรรถนะของการแบ่งนับรหัสลายเซ็นด้วยด้วยวิธีเวกเตอร์สุ่มแบบเรียงสับเปลี่ยนสำหรับผู้ใช้แบบกลุ่ม



ภาพที่ 24 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้สำหรับผู้ใช้แบบกลุ่ม

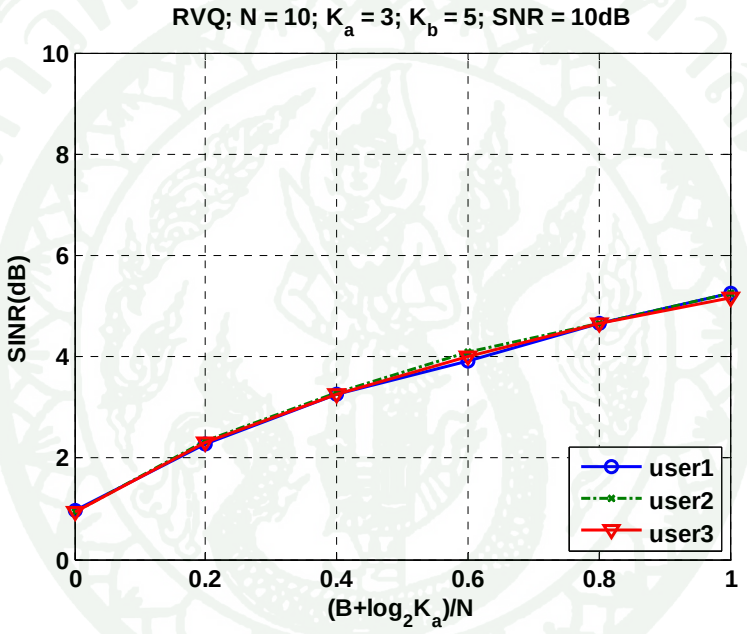


ภาพที่ 25 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มสำหรับผู้ใช้แบบกลุ่ม

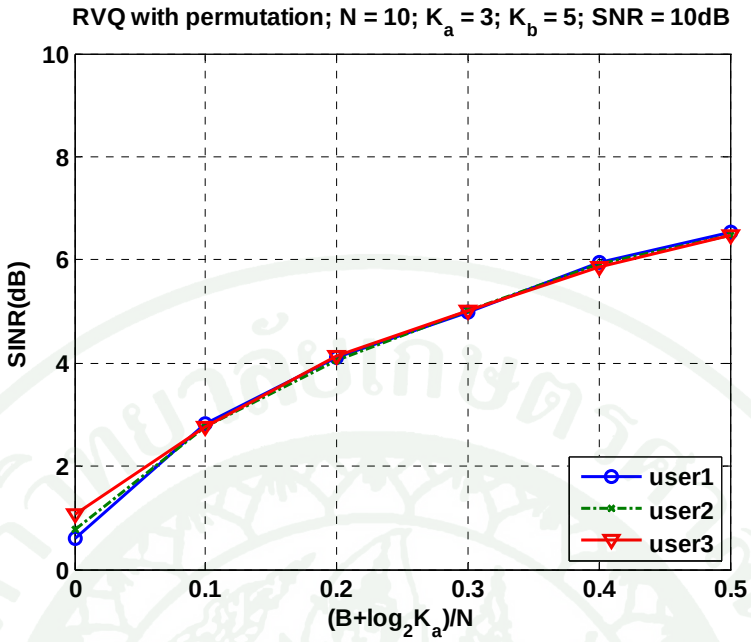


ภาพที่ 26 สมรรถนะของการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีสำหรับผู้ใช้แบบกลุ่ม

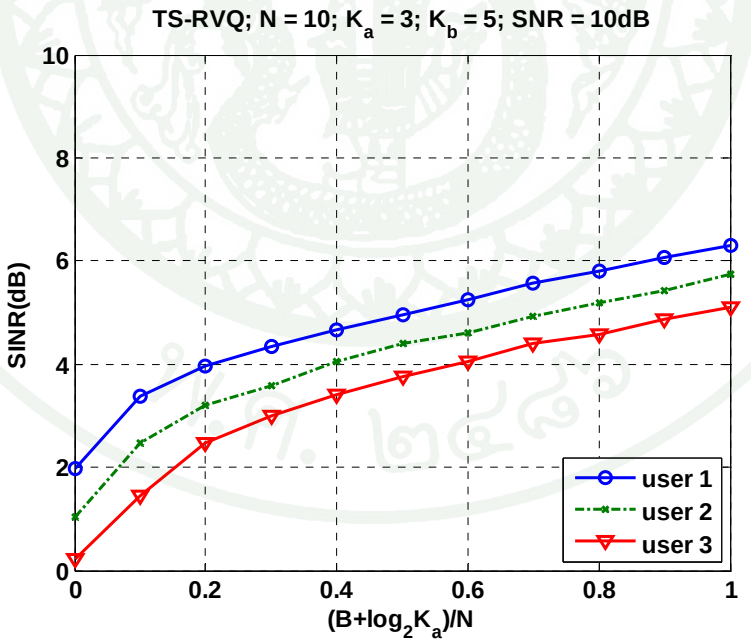
การทดสอบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆนั้น นอกจากจะพิจารณาที่สมรรถนะเฉลี่ยของระบบแล้ว สมรรถนะของผู้ใช้แต่ละรายที่ต้องการปรับรหัสลายเซ็นก็ได้ถูกนำมาพิจารณาเช่นกัน ภาพที่ 27, 28, 29, 30 และ 31 แสดงสมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่ม, การแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มแบบเรียงสับเปลี่ยน, การแบ่งนับรหัสลายเซ็นโครงสร้างต้นไม้, การแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มและการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีตามลำดับ



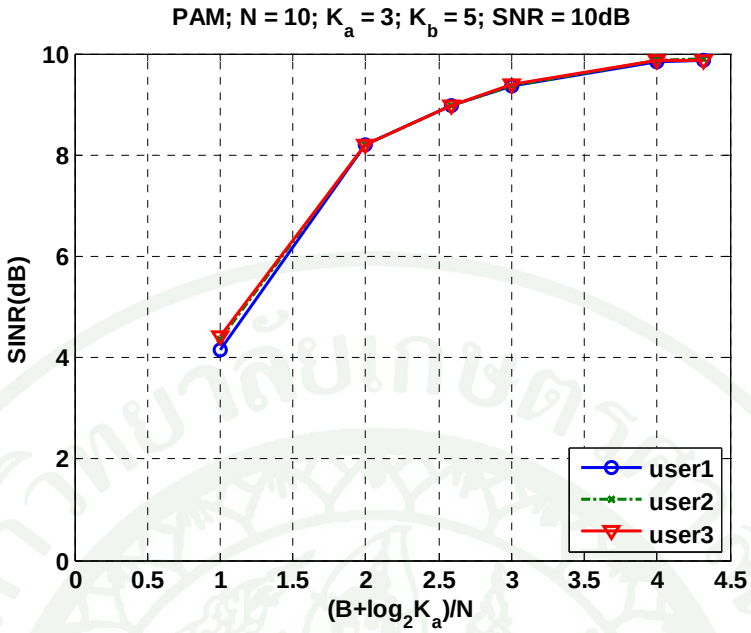
ภาพที่ 27 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่ม



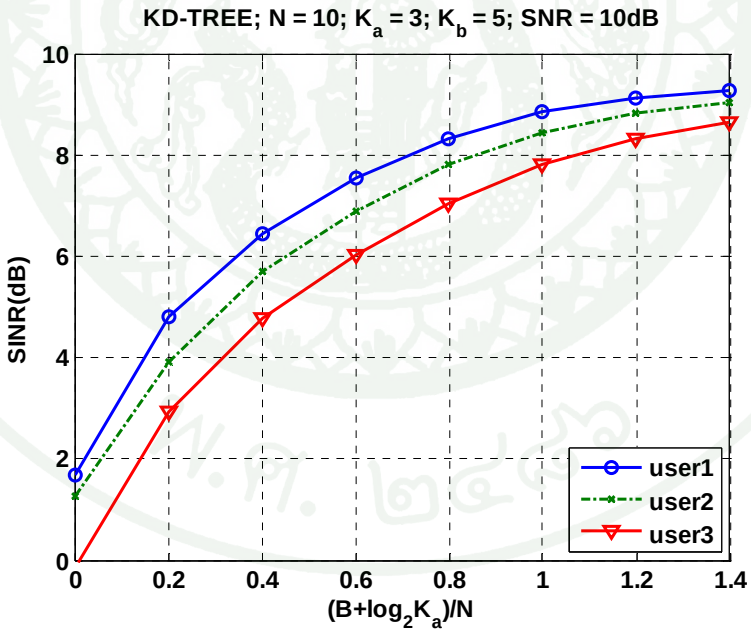
ภาพที่ 28 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มแบบเรียงสับเปลี่ยน



ภาพที่ 29 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นโครงสร้างต้นไม้

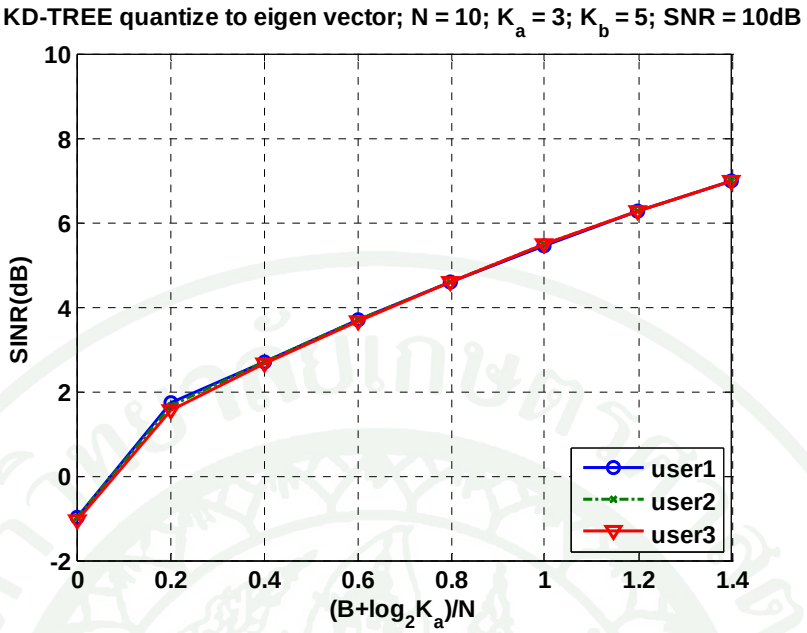


ภาพที่ 30 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็ม



ภาพที่ 31 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดี

พิจารณาภาพที่ 27 และ 28 จะเห็นว่าสมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นด้วยวิธีเวกเตอร์สุ่มและเวกเตอร์สุ่มแบบเรียงสับเปลี่ยนนั้นมีความใกล้เคียงกันเนื่องจากทั้งสองวิธีนี้การจะเลือกรหัสลายเซ็นพร้อมกันทั้งเซต ดังนั้นเซตของรหัสลายเซ็นที่ถูกเลือกหรือเซตรหัสลายเซ็นที่ให้ SINR เฉลี่ยของระบบสูงที่สุดนั้นรหัสลายเซ็นที่เป็นสมาชิกทุกตัวในเซตนี้จึงให้ SINR ที่ใกล้เคียงกัน แต่เมื่อพิจารณาภาพที่ 29 และ 31 นั้นจะพบว่าสมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้และแบบต้นไม้เคดีนั้นแตกต่างกันอย่างเห็นได้ชัด และมีแนวโน้มไปในทางเดียวกันคือสมรรถนะของผู้ใช้รายที่ 1 มีค่ามากที่สุดรองลงมาคือผู้ใช้รายที่ 2 และผู้ใช้รายที่ 3 ตามลำดับ ซึ่งเมื่อพิจารณาวิธีการแบ่งนับทั้งสองวิธีที่จะหารหัสลายเซ็นของผู้ใช้รายที่ 1 เป็นอันดับแรกหลังจากนั้นจึงหารหัสลายเซ็นของผู้ใช้รายที่ 2 และผู้ใช้รายที่ 3 ดังนั้นจึงสามารถอธิบายภาพทั้งสองได้กล่าวคือการแบ่งนับรหัสลายเซ็นทั้งสองวิธีนั้นผู้ใช้รายที่ 1 จะมีอิสระในการหารหัสลายเซ็นมากที่สุดรองลงมาคือผู้ใช้รายที่ 2 ที่ต้องปรับรหัสลายเซ็นให้หลีกเลี่ยงการรบกวนจากผู้ใช้รายที่ 1 จึงมีอิสระในการเลือกเวกเตอร์น้อยกว่า เหตุผลดังกล่าวสามารถใช้อธิบายสมรรถนะของผู้ใช้รายที่ 3 ได้เช่นเดียวกัน สำหรับภาพที่ 30 การแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มนั้นจากเหตุผลที่อธิบายข้างต้นนั้นแนวโน้มของสมรรถนะของผู้ใช้แต่ละรายควรจะเหมือนกับการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้และการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดี ซึ่งจากภาพที่ 30 ปรากฏว่าไม่ได้เป็นไปตามที่กล่าวมา ด้วยเหตุนี้ผู้วิจัยจึงได้ทดลองใช้การแบ่งนับแบบต้นไม้เคดีสำหรับการหาเวกเตอร์ใกล้เคียงกับไอเกนเวกเตอร์ที่ให้ค่าลักษณะเฉพาะต่ำที่สุดของเมทริกซ์แปรผันร่วมของสัญญาณแทรกสอดเช่นเดียวกับการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มซึ่งได้ผลออกมาดังภาพที่ 32

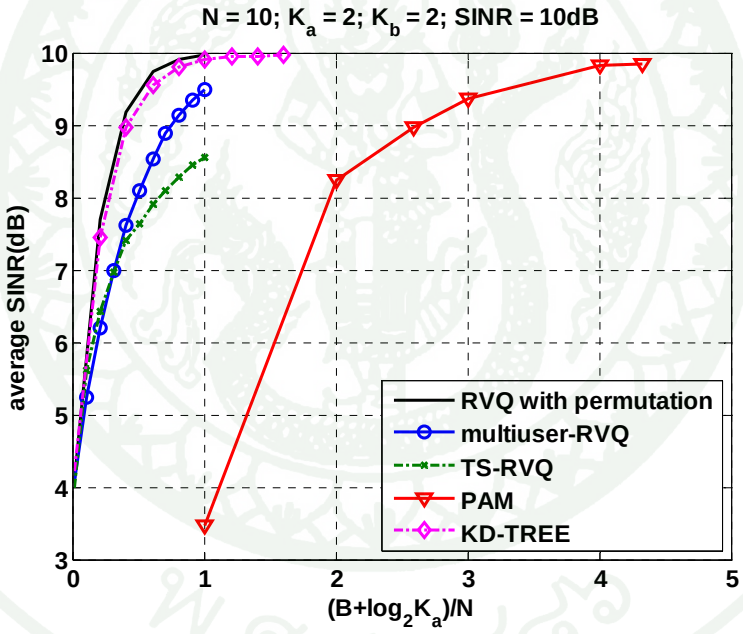


ภาพที่ 32 สมรรถนะของผู้ใช้แต่ละรายสำหรับการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีเมื่อใช้วิธีหาเวกเตอร์ใกล้เคียงกับไอเกนเวกเตอร์ของเมทริกซ์แปรผันร่วมของสัญญาณแทรกสอด

จากภาพที่ 32 จะเห็นว่าแนวโน้มของสมรรถนะของผู้ใช้แต่ละรายเหมือนกับการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็ม ดังนั้นในเบื้องต้นจึงสรุปได้ว่าการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มแบบพีเอเอ็มนั้นแม้จะใช้วิธีการหารหัสลายเซ็นสำหรับผู้ใช้ทีละราย แต่วิธีการหารหัสลายเซ็นนั้นใช้การหาเวกเตอร์ใกล้เคียงกับไอเกนเวกเตอร์ของผู้ใช้แต่ละรายซึ่งไม่ได้หาเซตของเวกเตอร์ที่สมาชิกแต่ละตัวอยู่ห่างกันมากที่สุดดังนั้นสมรรถนะของผู้ใช้แต่ละรายจึงใกล้เคียงกัน

ภาพที่ 33 และ 34 แสดงการเปรียบเทียบการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆในระบบที่ $K_a = 2$; $K_b = 2$ โดยภาพที่ 33 แสดงสมรรถนะและภาพที่ 34 แสดงจำนวนการคำนวณที่ต้องใช้สำหรับวิธีต่างๆ พิจารณาภาพที่ 33 พบว่าการแบ่งนับเวกเตอร์สุ่มแบบเรียงสับเปลี่ยนให้สมรรถนะดีที่สุดเมื่อพิจารณาที่จำนวนบิตป้อนกลับที่เท่ากัน การแบ่งนับแบบต้นไม้เคดีนั้นให้สมรรถนะใกล้เคียงกับการแบ่งนับเวกเตอร์สุ่มแบบเรียงสับเปลี่ยนเป็นอย่างมาก ส่วนการแบ่งนับเวกเตอร์แบบโครงสร้างต้นไม้และการแบ่งนับแบบเวกเตอร์สุ่มนั้นให้สมรรถนะที่แย่กว่าการแบ่งนับเวกเตอร์แบบต้นไม้เคดีโดยที่เมื่อจำนวนบิตป้อนกลับมีค่าน้อยการแบ่งนับแบบโครงสร้างต้นไม้ให้สมรรถนะที่แย่กว่าการแบ่งนับเวกเตอร์แบบสุ่มแต่เมื่อจำนวนบิตป้อนกลับเพิ่มขึ้นการแบ่งนับเวกเตอร์แบบสุ่มให้สมรรถนะที่ดีกว่าการแบ่งนับแบบโครงสร้างต้นไม้

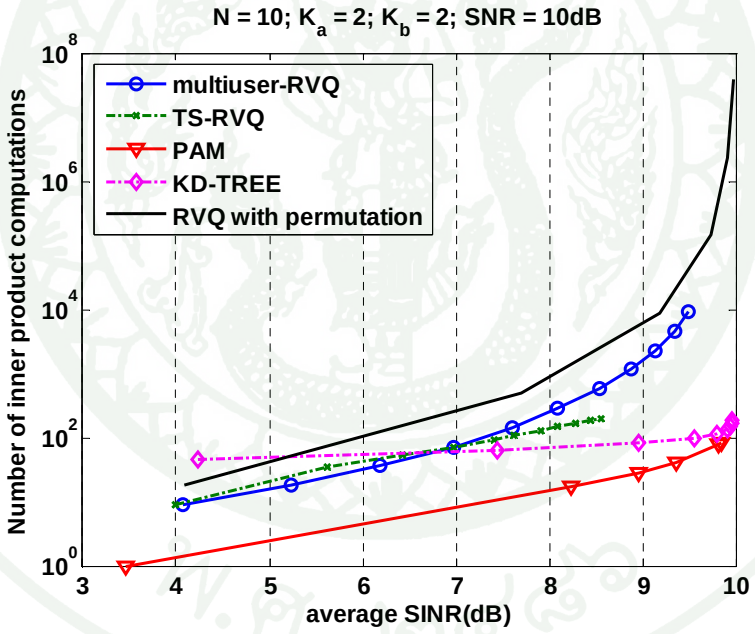
สำหรับการแบ่งนับนับแบบพีเอเอ็มนั้นไม่สามารถหาสมรรถนะได้ที่ $(B+\log_2 K_a)/N$ น้อยกว่า 1 เช่นเดียวกับในกรณีของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียว พิจารณาที่ $(B+\log_2 K_a)/N$ เท่ากับ 1 การแบ่งนับแบบพีเอเอ็มให้สมรรถนะที่ด้อยกว่าการแบ่งนับรหัสลายเซ็นด้วยวิธีอื่นๆซึ่งเมื่อเปรียบเทียบกับสมรรถนะของการแบ่งนับรหัสลายเซ็นด้วยวิธีอื่นๆที่ $(B+\log_2 K_a)/N$ เท่ากับ 0 หรืออีกความหมายหนึ่งคือไม่มีช่องสัญญาณป้อนกลับแล้วสมรรถนะของการแบ่งนับแบบพีเอเอ็มที่ $(B+\log_2 K_a)/N$ เท่ากับ 1 ก็ยังด้อยกว่าเนื่องจากการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มนั้นที่ $(B+\log_2 K_a)/N$ เท่ากับ 1 มีสัญลักษณ์พีเอเอ็มอยู่เพียงสองตัวคือ -1 และ 1 ดังนั้นเมื่อใช้สัญลักษณ์ดังกล่าวเป็นรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มแล้วโอกาสที่รหัสลายเซ็นของผู้ใช้ที่ต้องการปรับรหัสลายเซ็นจะรบกวนกันเองจึงมีมากกว่าการสุ่มเลือกรหัสลายเซ็นโดยไม่ใช้ช่องสัญญาณป้อนกลับเลย



ภาพที่ 33 เปรียบเทียบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆ ในระบบที่ $K_a = 2; K_b = 2$

ภาพที่ 34 แสดงการเปรียบเทียบจำนวนการคำนวณที่การแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มวิธีต่างๆต้องการ พิจารณาภาพที่ 34 พบว่าการแบ่งนับเวกเตอร์แบบสุ่มแบบเรียงสับเปลี่ยนนั้นใช้จำนวนการคำนวณที่มากกว่าการแบ่งนับวิธีอื่นเมื่อเปรียบเทียบที่ SINR มีค่ามากๆเนื่องจากการแบ่งนับรหัสลายเซ็นวิธีนี้ต้องทดสอบเซตของรหัสลายเซ็นทุกเซตที่เป็นไปได้ในโค้ดบุ๊คซึ่งมีจำนวน

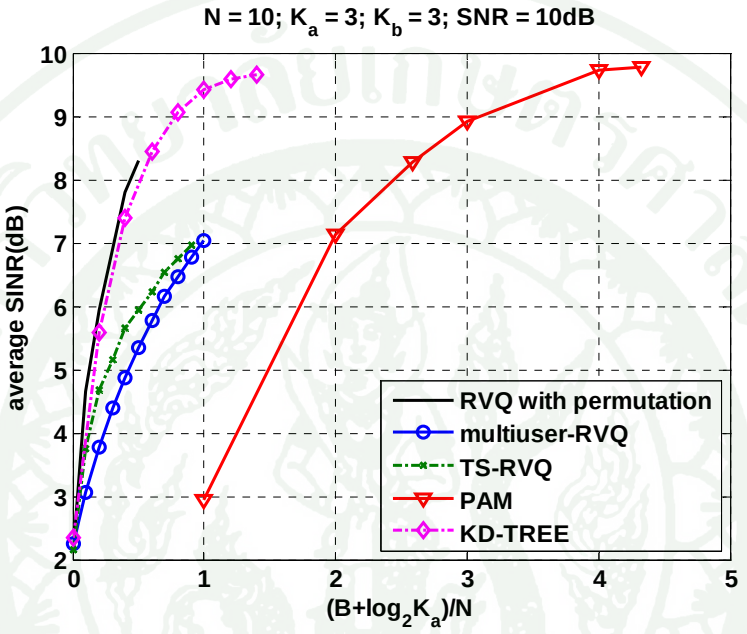
เท่ากับ $\frac{(n\times K_a)!}{((n\times K_a)-K_a)!}$ เมื่อ n คือจำนวนเวกเตอร์ในโค้ดบุ๊คดังนั้นการแบ่งนับรหัสลายเซ็นวิธีนี้จึงให้สมรรถนะดีที่สุด การแบ่งนับแบบต้นไม้เคดีต้องการความซับซ้อนในการคำนวณน้อยกว่าการแบ่งนับเวกเตอร์แบบสุ่มแบบเรียงสับเปลี่ยนมากเมื่อพิจารณาที่ SINR มีค่าเท่ากันและจากภาพที่ 34 จะเห็นว่าสมรรถนะของการแบ่งนับแบบต้นไม้เคดีใกล้เคียงกับการแบ่งนับเวกเตอร์แบบสุ่มแบบเรียงสับเปลี่ยนอย่างมากดังนั้นการแบ่งนับแบบต้นไม้เคดีจึงเหมาะสมสำหรับการปรับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่ม พิจารณาการแบ่งนับแบบเวกเตอร์สุ่มและแบบโครงสร้างต้นไม้พิจารณาที่ SINR เท่ากับ 7.5 เดซิเบลจะเห็นว่าการแบ่งนับแบบโครงสร้างต้นไม้ใช้จำนวนการคำนวณที่น้อยกว่า สำหรับการแบ่งนับแบบพีเอเอ็มนั้นจะเห็นว่าใช้จำนวนการคำนวณที่น้อยกว่าทุกวิธีแต่ต้องใช้จำนวนบิตป้อนกลับมากสำหรับ SINR ที่มีค่าสูงดังนั้นการแบ่งนับแบบพีเอเอ็มจึงเหมาะกับระบบซีดีเอ็มเอที่ช่องสัญญาณป้อนกลับมีอัตราบิตสูงและต้องการการคำนวณที่ไม่มาก



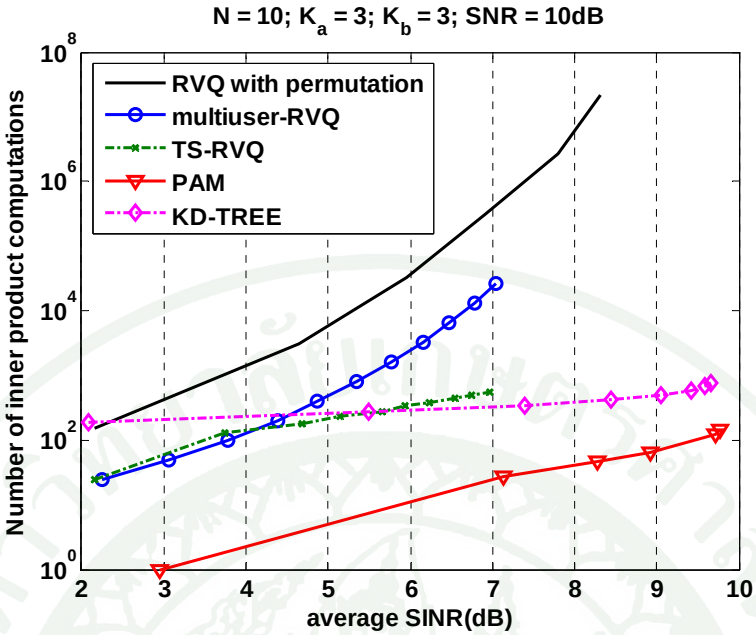
ภาพที่ 34 เปรียบเทียบจำนวนการคำนวณของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆในระบบที่ $K_a = 2$; $K_b = 2$

ภาพที่ 35 และ 36 แสดงการเปรียบเทียบการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆ ในระบบที่ $K_a = 3$; $K_b = 3$ พิจารณาทั้งสองภาพจะเห็นว่าแนวโน้มของสมรรถนะของการแบ่งนับด้วยวิธีต่างๆไม่แตกต่างจากระบบที่ $K_a = 2$; $K_b = 2$ มากนักแต่ที่น่าสนใจก็คือการแบ่งนับแบบโครงสร้างต้นไม้ให้สมรรถนะที่ดีกว่าการแบ่งนับเวกเตอร์สุ่มเมื่อจำนวนผู้ใช้ที่ต้องการปรับรหัส

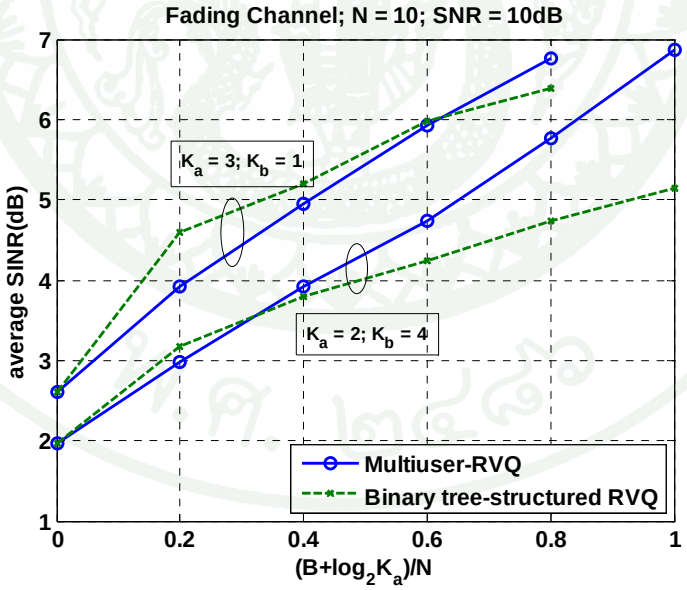
ลายเซ็นเพิ่มขึ้นจาก 2 เป็น 3 และคาดว่าช่องว่างระหว่างการแบ่งนับแบบโครงสร้างต้นไม้และการแบ่งนับเวกเตอร์สุ่มจะเพิ่มขึ้นเมื่อจำนวนผู้ใช้ที่ต้องการปรับรหัสลายเซ็นเพิ่มขึ้น ที่เป็นเช่นนี้เนื่องจากเมื่อจำนวนผู้ใช้ที่ต้องการปรับรหัสลายเซ็นเพิ่มขึ้นการแบ่งนับแบบโครงสร้างต้นไม้จะมีอิสระในการเลือกเวกเตอร์มากกว่าการแบ่งนับแบบเวกเตอร์สุ่มนั่นเอง



ภาพที่ 35 เปรียบเทียบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆ ในระบบที่ $K_a = 3; K_b = 3$

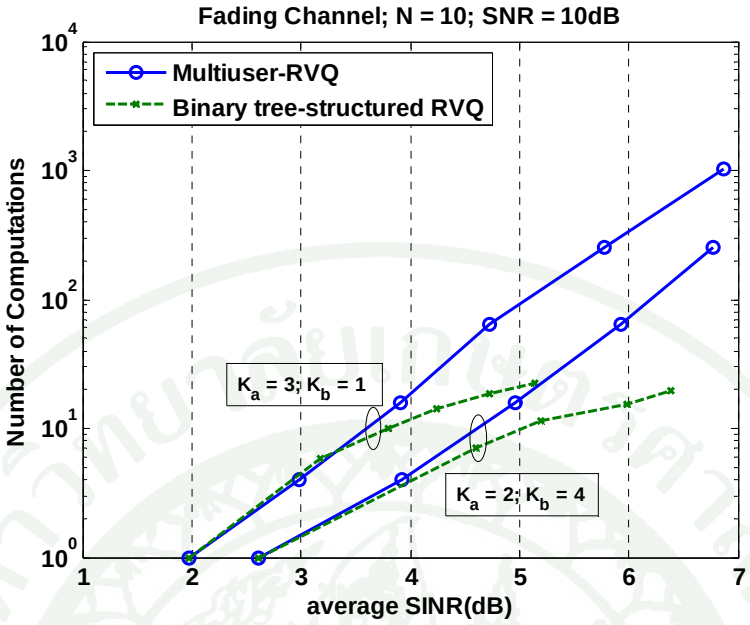


ภาพที่ 36 เปรียบเทียบจำนวนการคำนวณของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีต่างๆในระบบที่ $K_a = 3; K_b = 3$



ภาพที่ 37 เปรียบเทียบสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีการแบ่งนับแบบเวกเตอร์สุ่มและแบบโครงสร้างต้นไม้ในช่องสัญญาณจางหายแบบเลือกความถี่แบบช้าโดยกำหนดให้ $L = 2$ และ $\alpha = 4$ ทำการจำลองระบบที่ $K_a = 3; K_b = 1$ และ $K_a = 2; K_b = 4$

ภาพที่ 37 และ 38 แสดงสมรรถนะของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มของวิธีการแบ่งนับแบบเวกเตอร์สุ่มสำหรับผู้ใช้แบบกลุ่มและการแบ่งนับแบบโครงสร้างต้นไม้ในช่องสัญญาณจางหายแบบเลือกความถี่ โดยงานวิจัยนี้กำหนดให้เป็นการจางหายแบบช้า (slow fading) ซึ่งช่องสัญญาณประเภทนี้จะมีการเปลี่ยนแปลงช้า เนื่องจากงานวิจัยนี้กำหนดให้ที่ฝั่งรับสามารถประมาณช่องสัญญาณและคำนวณรหัสลายเซ็นที่เหมาะสมแล้วจึงทำการแบ่งนับและส่งข้อมูลไปยังฝั่งส่งทางบิตป้อนกลับ ซึ่งถ้าช่องสัญญาณมีการเปลี่ยนแปลงอย่างรวดเร็วการปรับรหัสลายเซ็นโดยใช้บิตป้อนกลับก็จะไม่มีประโยชน์ จากแบบจำลองมาตามสมการที่ 3 และ 4 กำหนดให้ $L = 2$ และ $\alpha = 4$ ในภาพที่ 37 และ 38 นี้ได้ทำการจำลองระบบที่ $K_a = 3$; $K_b = 1$ และ $K_a = 2$; $K_b = 4$ พิจารณาจากทั้งสองภาพจะเห็นว่าแนวโน้มของสมรรถนะของการแบ่งนับรหัสลายเซ็นทั้งสองวิธียังคงเหมือนกับช่องสัญญาณที่เป็นอุดมคติทั้ง SINR และ จำนวนการคำนวณสำหรับการแบ่งนับรหัสลายเซ็นด้วยวิธีอื่นๆในช่องสัญญาณนั้นในงานวิจัยนี้ไม่ได้แสดงไว้ด้วยเหตุผลดังนี้ การแบ่งนับรหัสลายเซ็นแบบสุ่มแบบเรียงสับเปลี่ยนนั้นต้องใช้ความซับซ้อนในการคำนวณเป็นอย่างมากและเมื่อนำมาพิจารณาในช่องสัญญาณที่สมาชิกในเมทริกซ์ของช่องสัญญาณและการลดทอนเป็นค่าสุ่มจึงทำให้ต้องใช้ต้องใช้การนับจำนวนครั้งที่เท่ากับ 100,000 ครั้งสำหรับวิธี monte carlo simulation ซึ่งคอมพิวเตอร์ต้องใช้เวลานานในการทำงาน สำหรับการแบ่งนับรหัสลายเซ็นแบบพีเอเอ็มและแบบต้นไม้เคดีนั้นเมื่อสัญญาณส่งผ่านช่องสัญญาณแล้วเอาท์พุตที่ได้จะเป็นจำนวนเชิงซ้อนซึ่งการแบ่งนับรหัสลายเซ็นทั้งสองวิธียังไม่สามารถใช้กับจำนวนเชิงซ้อนได้ในงานวิจัยนี้



ภาพที่ 38 เปรียบเทียบจำนวนการคำนวณของการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มด้วยวิธีการแบ่งนับแบบเวกเตอร์สุ่มและแบบโครงสร้างต้นไม้ในช่องสัญญาณจางหายแบบเลือกความถี่แบบช้าโดยกำหนดให้ $L = 2$ และ $\alpha = 4$ ทำการจำลองระบบที่ $K_a = 3; K_b = 1$ และ $K_a = 2; K_b = 4$

วิจารณ์

1. ในระบบซีดีเอ็มเอนั้นเมื่อจำนวนผู้ใช้ในระบบเพิ่มมากขึ้นจะทำให้สมรรถนะของระบบลดลงทั้งในส่วนของผู้ใช้รายเดียวและผู้ใช้แบบกลุ่มเนื่องจากเมื่อจำนวนผู้ใช้ในระบบเพิ่มขึ้นการรบกวนกันเองภายในกลุ่มและระหว่างกลุ่มจะเพิ่มขึ้นตามไปด้วย จากการทดลองพบว่าเมื่อจำนวนของผู้ใช้ในระบบไม่เกิน 50 เปอร์เซ็นต์ของอัตราขยายประมวลผลหรือ processing gain สมรรถนะของระบบจะยังอยู่ในเกณฑ์ที่ดี และเมื่อย้อนกลับไปพิจารณาทฤษฎีของระบบซีดีเอ็มเอพบว่าเมื่อจำนวน processing gain เพิ่มมากขึ้นทำให้สัญญาณข้อมูลสามารถขยายแถบความถี่ได้กว้างมากขึ้นซึ่งทำให้ความแตกต่างของสัญญาณส่งมีมากขึ้นโอกาสที่สัญญาณจะรบกวนกันเองมีน้อยลงทำให้สมรรถนะของระบบเพิ่มขึ้น ดังนั้นระบบซีดีเอ็มเอที่ต้องการมีผู้ใช้ในระบบจำนวนมากควรจะมีค่า processing gain ที่สูงเพื่อที่จะได้สมรรถนะตามที่ต้องการ

2. การเลือกใช้วิธีการแบ่งนับรหัสลายเซ็นควรคำนึงถึงทรัพยากรที่มีอยู่เพื่อประกอบการพิจารณาด้วยเช่นอัตราของบิตป้อนกลับ, ความสามารถของตัวแบ่งนับ (quantizer) เนื่องจากการแบ่งนับรหัสลายเซ็นแต่ละวิธีมีข้อดีข้อเสียที่แตกต่างกัน เช่นการแบ่งนับแบบพีเอเอ็มให้สมรรถนะที่ดีเมื่ออัตราบิตป้อนกลับสูงและใช้การคำนวณที่ไม่ซับซ้อนหรือการแบ่งนับแบบเวกเตอร์สุ่มให้สมรรถนะดีที่สุดแต่ใช้การคำนวณที่ซับซ้อนเมื่อจำนวนบิตป้อนกลับเพิ่มขึ้นแต่เมื่อพิจารณาที่บิตป้อนกลับมีจำนวนน้อยแล้วพบว่าการแบ่งนับแบบเวกเตอร์สุ่มต้องการจำนวนการคำนวณที่ไม่แตกต่างจากการแบ่งนับด้วยวิธีอื่นๆมากนัก

3. การสร้างโค้ดบุ๊คแบบโครงสร้างต้นไม้หรือแบบต้นไม้เคดีนั้นเป็นการนำโค้ดบุ๊คแบบเวกเตอร์สุ่มมาจัดเรียงใหม่ซึ่งกระบวนการจัดเรียงดังกล่าวต้องมีการคำนวณด้วยอย่างไรก็ตามการจัดเรียงโครงสร้างของโค้ดบุ๊คทั้งสองแบบจะทำครั้งเดียวและสามารถใช้สำหรับการปรับรหัสลายเซ็นได้ตลอดเวลา สำหรับการแบ่งนับแบบพีเอเอ็มนั้นไม่ต้องมีการเก็บข้อมูลของเวกเตอร์ที่ใช้เป็นรหัสลายเซ็นซึ่งเป็นการประหยัดหน่วยความจำ

4. การแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้และแบบต้นไม้เคดีนั้นใช้วิธีการแวะผ่านเข้าไปในโครงสร้างต้นไม้เช่นเดียวกัน แต่จากผลการทดลองทั้งในกรณีของผู้ใช้รายเดียวและผู้ใช้หลายรายพบว่าการแบ่งนับรหัสสลายลายเซ็นแบบต้นไม้เคดีให้สมรรถนะที่ดีกว่าการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้ ทั้งนี้เนื่องจากการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีนั้นหลังจากที่ได้รหัสลายเซ็นแล้วจะนำรหัสลายเซ็นกลับไปตรวจสอบความถูกต้องโดยการแวะผ่านเข้าไปในต้นไม้เคดีอีกครั้งหนึ่ง ซึ่งวิธีดังกล่าวการแบ่งนับรหัสลายเซ็นแบบโครงสร้างต้นไม้ไม่สามารถทำได้

สรุปและข้อเสนอแนะ

สรุป

การนำเสนอวิธีการแบ่งนับรหัสลายเซ็นที่มีความซับซ้อนในการค้นหาน้อยในงานวิจัยนี้มี 3 วิธีคือการแบ่งนับแบบโครงสร้างต้นไม้, การแบ่งนับแบบพีเอเอ็มและการแบ่งนับแบบต้นไม้เคดี โดยงานวิจัยนี้ได้นำวิธีการแบ่งนับทั้งสามวิธีมาเปรียบเทียบกับวิธีการแบ่งนับเวกเตอร์แบบสุ่ม การทำงานเริ่มจากการศึกษาการแบ่งนับเวกเตอร์แบบต่างๆและทำการทดสอบการแบ่งนับเวกเตอร์ด้วยวิธีต่างๆในระบบซีดีเอ็มเอที่ผู้ใช้รายเดียวต้องการปรับรหัสลายเซ็นโดยได้ทำการทดสอบสมรรถนะของการแบ่งนับรหัสลายเซ็นที่มีจำนวนผู้ใช้ในระบบที่แตกต่างกันคือ 3, 5 และ 7 หลังจากนั้นจึงนำวิธีการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างๆไปทดสอบกับระบบซีดีเอ็มเอสำหรับผู้ใช้แบบกลุ่มและได้ทำการทดสอบที่ $K_a = 2; K_b = 2$, $K_a = 2; K_b = 3$, $K_a = 3; K_b = 2$ และ $K_a = 3; K_b = 5$ ตามลำดับ โดยผลการทดสอบทั้งระบบสำหรับผู้ใช้รายเดียวและระบบสำหรับผู้ใช้แบบกลุ่มสามารถสรุปได้ดังนี้

1. การแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวนั้นการแบ่งนับเวกเตอร์สุ่มให้สมรรถนะสูงที่สุดเมื่อเปรียบเทียบที่จำนวนบิตป้อนกลับเท่ากันแต่เมื่อพิจารณาในส่วนของจำนวนการคำนวณแล้วการแบ่งนับแบบพีเอเอ็มใช้จำนวนการคำนวณที่น้อยที่สุดเมื่อเปรียบเทียบที่ SINR เท่ากันแต่ข้อเสียของการแบ่งนับแบบพีเอเอ็มคือไม่สามารถใช้ SINR เป็นฟังก์ชันเป้าหมายได้โดยตรงดังนั้นการแบ่งนับวิธีนี้จึงต้องใช้จำนวนบิตป้อนกลับมากเพื่อให้ได้ SINR ที่มีค่าสูง

2. สำหรับการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้แบบกลุ่มนั้นแนวโน้มของสมรรถนะไม่ต่างจากการแบ่งนับรหัสลายเซ็นสำหรับผู้ใช้รายเดียวมากนัก แต่ที่น่าสนใจก็คือการแบ่งนับรหัสลายเซ็นแบบเวกเตอร์สุ่มนั้นเมื่อจำนวนผู้ใช้ในกลุ่มที่ต้องการปรับรหัสลายเซ็นเพิ่มขึ้นสมรรถนะของการแบ่งนับรหัสลายเซ็นวิธีนี้จะแย่ลงเมื่อเปรียบเทียบกับการแบ่งนับรหัสลายเซ็นวิธีอื่น ทั้งนี้เนื่องจากการแบ่งนับรหัสลายเซ็นวิธีนี้มีอิสระในการเลือกรหัสลายเซ็นน้อยกว่าวิธีอื่นๆ

3. เมื่อพิจารณาสมรรถนะและจำนวนการคำนวณของการแบ่งนับรหัสลายเซ็นวิธีต่างๆ พบว่าการแบ่งนับแบบต้นไม้เคดีมีความเหมาะสมที่สุดเนื่องจากให้สมรรถนะที่สูงเมื่อใช้บิต

ป้อนกลับไม่มากและใช้จำนวนการคำนวณที่ไม่มากนักเมื่อเทียบกับสมรรถนะที่ได้ ทั้งในกรณีของผู้ใช้รายเดียวละผู้ใช้แบบกลุ่ม

ข้อเสนอแนะ

1. การนำเสนอวิธีการแบ่งนับรหัสลายเซ็นด้วยวิธีต่างๆในงานวิจัยนี้ พิจารณาระบบที่เป็น reverse-link คือสัญญาณถูกส่งจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐานและสถานีฐานเป็นผู้เลือกรหัสลายเซ็นสำหรับผู้ใช้แต่ละรายซึ่งสามารถทำได้เนื่องจากสถานีฐานมีข้อมูลของผู้ใช้ทุกราย ระบบที่เป็น forward-link หรือผู้ใช้แต่ละรายทำหน้าที่เป็นผู้เลือกรหัสลายเซ็นนั้นทำได้ยากเนื่องจากผู้ใช้แต่ละรายไม่มีข้อมูลของผู้ใช้รายอื่นซึ่งถ้าจะมีการติดต่อกันระหว่างผู้ใช้แต่ละรายก็จะเปลืองทรัพยากรในส่วนของช่องสัญญาณที่ใช้ติดต่อถึงกัน

2. เมื่อพิจารณาจากผลการทดลองทั้งในเชิงของสมรรถนะและเชิงของจำนวนการคำนวณนั้น พบว่าการแบ่งนับรหัสลายเซ็นแบบต้นไม้เคดีใช้จำนวนการคำนวณที่น้อยเมื่อพิจารณาที่สมรรถนะสูง แต่เมื่อพิจารณาที่สรรถนะต่ำแล้วพบว่ามีการแบ่งนับรหัสลายเซ็นด้วยวิธีอื่นที่ใช้จำนวนการคำนวณที่น้อยกว่า ดังนั้นหากจะนำไปใช้งานจริงควรจะเลือกการแบ่งนับรหัสลายเซ็นให้เหมาะสมกับสมรรถที่ต้องการ

3. งานวิจัยนี้สนใจเฉพาะการจำลองระบบด้วยโปรแกรมคอมพิวเตอร์และการสร้างรหัสลายเซ็นด้วยการสุ่มเท่านั้น สำหรับการนำไปใช้งานจริงนั้นรหัสลายเซ็นถูกสร้างด้วยชิพรีจิสเตอร์จากวงจรดิจิตอลและได้รหัสลายเซ็นชนิดต่างๆได้แก่ รหัสเชิงตั้งฉาก (Orthogonal Code) รหัสโกลด์ (Gold Code) และชุดรหัสคาซามิ (Kasami Sequence) นอกจากนั้นการออกแบบและการสร้างระบบสำหรับแบ่งนับรหัสลายเซ็นด้วยวงจรดิจิตอลยังไม่มีการสร้างขึ้นจริงซึ่งการออกแบบและการสร้างระบบดังกล่าวจะเป็นงานในอนาคต

เอกสารและสิ่งอ้างอิง

ลัญฉกร วุฒิสิทธิกุลกิจ, เขมธนะ สุวพิชญ์ภูมิ, จุมพฏ ชูสิงห์, วรางรัตน์ วัฒนวรากุล และ อนุชิต มั่นจิรังกุล. 2548. **เทคโนโลยีสื่อสารไร้สาย CDMA**. สำนักพิมพ์จุฬาลงกรณ์ มหาวิทยาลัย ,กรุงเทพฯ.

Bently, J. 1980. Multidimentional divide and conquer. **Commun. of the ACM.** vol. 22, no. 4: pp. 214-229.

Gresho, A. and R. M. Gray. 1991. **Vector Quantization and Signal Compression**. Springer.

Katsvaounidis,I., C.-C. J. Kuo, and Z. Zhang. 1996. Fast tree-structured nerest neighbor encoding for vector quantization. **IEEE Trans. Image Orocess.** vol. 5, no. 2: pp 398-404.

Moore, A. 1991. **An introductory tutorial on kd-trees**. Ph.D. Thesis. University of Cambridge.

Love, D., R. W. Heath, Jr., W. Santipach, and M. L. Honig. 2004. What is the value of limited feedback for MIMO channels ?**. IEEE Commun. Mag.** vol. 42, no. 10: pp. 54-59.

Proakis, J. 2001. **Digital Communications.** McGraw-Hill, Singapore.

Rajappan, G. and M. L. Honig. 2002 Signature Sequence Adaptation for DS-CDMA in multipath. **IEEE J. Sel Areas Commun.** Vol 20, no.2, pp. 384-395.

Rapajic, P. and B. Vucetic. 1995. Linear adaptive transmitter-receiver structures for asynchronous CDMA system. **European Trans. on Telecommun.** vol. 6, no. 1: pp. 21-28.

Ryan, D., I. B. Collings, and I. V. L. Clarkson. 2007. GLRT-Optimal Noncoherent Lattice Decoding. **IEEE Trans. Sig. Proc.** vol. 55, no. 7: pp. 3773-3785.

Ryan, D., I. V. L. Clarkson, I. B. Collings, D. Gao, and M. L. Honig. 2007. QAM Codebooks for Low-Complexity Limited Feedback MIMO beamforming, pp. 54-59. **IEEE Int. Conf. on Commun. (ICC).** Glasgow Scotland.3

Santipach, W. 2008. Tree-structured random vector quantization for beam forming in a multiantenna channel, pp. 325-328. **ECTI-CON.** Krabi, Thailand.

Santipach, W. and M. L. Honig. 2009. Capacity of a multiple antenna fading channel with a quantized precoding matrix. **IEEE Trans. Inf. Theory.** vol. 55, no. 3: pp. 1218-1234.

Santipach, W. and M. L. Honig. 2005. Signature optimization for CDMA with limited feedback. **IEEE Trans. Inf. Theory.** vol. 51, no. 10: pp. 3475-3492.

Santipach, W., Y. Sun, and M. L. Honig. 2003. Benefits of limited feedback for wireless channels. **Allerton Conf. on Commun., Control, and Computing.** Monticello, Illinois, USA.

Viswanath, P., V. Anantharam, and D. N. C. Tse. 1999. Optimal sequence power control and capacity of spread-spectrum system with multiuser linear receivers. **IEEE Trans. Inf. Theory.** vol. 45, no 6: pp. 1968-1983.

ภาคผนวก

Multiuser Signature Quantization With Tree-Structured Codebook in DS-CDMA

Kritsada Mamat and Wiroonsak Santipach
Department of Electrical Engineering
Faculty of Engineering, Kasetsart University
Bangkok, Thailand 10900
Email: {g5065253, wiroonsak.s}@ku.ac.th

***Abstract*—We consider a signature quantization scheme for a group of users in a reverse-link direct sequence (DS)-code division multiple access (CDMA). Assuming perfect channel knowledge, a receiver selects the set of signatures that maximizes an average signal-to-interference plus noise ratio (SINR) from a random vector quantization (RVQ) codebook, which consists of independent isotropically distributed unit-norm vectors. The quantized signatures are relayed from the receiver to users via noiseless rate-limited feedback channels. Previously, we have proposed to organize entries of RVQ codebook into a tree structure (TS) to speed up a search for the optimal entry. Here we extend the TS scheme for a *multiuser* signature quantization. Numerical results show that for a given performance, a TS-RVQ codebook can be an order of magnitude less complex than an RVQ codebook.**

I. Introduction

We can improve a user performance in a direct sequence (DS)-code division multiple access (CDMA) by optimizing the user's signature sequence [1]. A receiver, which can estimate an interference covariance matrix, can compute the optimal signature for the user. The optimal signature that minimizes an interference power is the eigenvector of the interference-plus-noise covariance matrix corresponding to the smallest eigenvalue. The user then obtains the optimal signature from the receiver via a feedback channel. However, a feedback channel in practice has a very limited rate and thus, the signature computed by the receiver needs to be quantized. Some of our previous work [2]–[4] and references therein have considered signature quantization in CDMA. Solutions to this signature quantization problem can also be applied in a multiantenna system where a spatial signature, which consists of transmit antenna weights, is quantized [4]–[8].

In [2], [9], we have proposed a Random Vector Quantization (RVQ) codebook, which consists of independent isotropically distributed unit-norm vectors and we have showed that RVQ is optimal over all codebooks in a large system limit in which processing gain, number of interfering users, and number of available feedback bits tend to infinity with fixed ratios. The large system limit has been shown to predict the performance of a finite-size system very well [9]. Furthermore, RVQ performs close to the optimal codebook for a finite-size system [5], [8]. However, RVQ and most of the codebooks proposed in literature require an exhaustive search to locate the optimal signature. With B feedback bits available, the codebook contains 2^B entries. Thus, the search complexity increases exponentially with a number of feedback bits and becomes an issue for a large B. Other simpler quantization schemes [7], [10] have been proposed. In [7], RVQ entries were organized into a tree and thus, the search complexity increases linearly with B. A codebook with entries consisting of only QAM symbols was considered in [10] and it was shown that a number of searches required also grows linearly with B. In [2], [9], a number of coefficients to quantize is reduced by projecting the signature vector onto a lower dimensional subspace and the coefficients are then scalar quantized. The complexity of this scheme is much less than the vector quantization, but the performance also suffers greatly.

Here we extend the TS-RVQ scheme proposed in [7] to *multiple users*. Even though there have been much work on signature quantization for a single user, signature quantization for group of users has not been paid much attention. This problem should interest a service operator, who would like to differentiate quality of service for a certain group of users. For practical reason, we consider a frequency-selective channel with path loss. Compared with an RVQ, TS-RVQ may give a worse performance for a given number of feedback bits. However, we show that search complexity of TS-RVQ can be an order of magnitude less than that of RVQ.

II. System Model

We consider a discrete-time reverse-link synchronous DS-CDMA with processing gain N. We assume that K_a users adapt their signature sequences while the rest of users, K_b, do not adapt signature sequences and hence, the *total* number of users $K = K_a + K_b$. The $N \times 1$ received vector is given by

$$\boldsymbol{r} = \sum_{k=1}^{K_a} \sqrt{A_k}\boldsymbol{H}_k \boldsymbol{s}_k b_k + \sum_{i=1}^{K_b} \sqrt{A_i}\boldsymbol{H}_i \boldsymbol{p}_i b_i + \boldsymbol{n} \qquad (1)$$

where $\sqrt{A_k}$ is an amplitude of user k, $\boldsymbol{H}_k$ is an $N \times N$ channel matrix for user k, $\boldsymbol{s}_k$ is an $N \times 1$ signature vector for adapting user k, $\boldsymbol{p}_i$ is an $N \times 1$ signature vector for interfering user i, b_k is a transmitted symbol for user k, $\boldsymbol{n}$ is an additive white Gaussian noise with zero mean and covariance $\sigma_n^2 \boldsymbol{I}$, and $\boldsymbol{I}$ is an identity matrix.

For an ideal non-fading channel, $\boldsymbol{H}_k = \boldsymbol{I}$ for all k. For a frequency-selective fading channel, assuming that each user

K. Mamat was supported by the 2008 Telecommunications Research and Industrial Development Institute (TRIDI) scholarship while W. Santipach was supported by the Commission on Higher Education and Thailand Research Fund under grant MRG5080174.

978-1-4244-3388-9/09/$25.00 ©2009 IEEE

traverses L fading paths, we have

$$\boldsymbol{H}_k = \begin{bmatrix} h_{k,1} & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & h_{k,1} & & \vdots & 0 & & \vdots \\ h_{k,L} & \vdots & \ddots & 0 & \vdots & & 0 \\ 0 & h_{k,L} & & h_{k,1} & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots & h_{k,1} & & 0 \\ 0 & \vdots & & h_{k,L} & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & h_{k,L} & \dots & h_{k,1} \end{bmatrix} \quad (2)$$

where fading gains for user k, $h_{k,1}, \dots, h_{k,L}$ are independent complex Gaussian random variables with zero mean and variance $E|h_{k,1}|^2, \dots, E|h_{k,L}|^2$, respectively. For a flat fading channel ($L = 1$), $\boldsymbol{H}_k = h_{k,1}\boldsymbol{I}$. Besides fading, a signal transmitted through medium also undergoes attenuation. It is well known that signal attenuation or path loss can be modeled as a function of distance [11], [12]. Thus, a signal power for user k is given by

$$A_{k,\mathrm{dB}}(d) = A_{k,\mathrm{dB}}(d_o) + 10\alpha \log\left(\frac{d}{d_o}\right) + X \quad (3)$$

where $A_{k,\mathrm{dB}}(d)$ is A_k in dB at distance d from a signal source, d_o is a reference distance, α is a path loss exponent, and X is a zero mean log-normal random variable whose unit is in dB.

We assume that the receiver uses a single-user matched filter given by

$$\boldsymbol{c}_k = \frac{\boldsymbol{H}_k \boldsymbol{s}_k}{|\boldsymbol{H}_k \boldsymbol{s}_k|} \quad (4)$$

to detect user k's transmitted symbol. An average output signal-to-interference plus noise ratio (SINR) over all K_a adapting users is given by

$$\mathrm{SINR}(\boldsymbol{S}) = \frac{\sum_{k=1}^{K_a} A_k (\boldsymbol{s}_k^\dagger \boldsymbol{H}_k^\dagger \boldsymbol{H}_k \boldsymbol{s}_k)^2}{I + \sigma_n^2 \sum_{k=1}^{K_a} (\boldsymbol{s}_k^\dagger \boldsymbol{H}_k^\dagger \boldsymbol{H}_k \boldsymbol{s}_k)} \quad (5)$$

where an interference power

$$I = \sum_{k=1}^{K_a} \sum_{l=1, l\neq k}^{K_a} A_l (\boldsymbol{s}_k^\dagger \boldsymbol{H}_k^\dagger \boldsymbol{H}_l \boldsymbol{s}_l)^2 + \sum_{k=1}^{K_a} \sum_{i=1}^{K_b} A_i (\boldsymbol{s}_k^\dagger \boldsymbol{H}_k^\dagger \boldsymbol{H}_i \boldsymbol{p}_i)^2 \quad (6)$$

and an $N \times K_a$ matrix containing signatures for K_a users

$$\boldsymbol{S} = [\boldsymbol{s}_1 \quad \boldsymbol{s}_2 \quad \cdots \quad \boldsymbol{s}_{K_a}]. \quad (7)$$

We note that SINR depends on a signature set $\{\boldsymbol{s}_k\}$ or $\boldsymbol{S}$ and thus, we would like to maximize this average SINR over a signature set $\{\boldsymbol{s}_k\}$. If the feedback channel between the receiver to users has unlimited rate, the optimized signature set can be fed back to users unquantized. The optimization problem with unlimited feedback has been solved in [1]. With limited number of feedback bits, the signature needs to be quantized. In what follows, we describe the proposed signature quantization scheme and show associated performance.

III. Tree-Structured RVQ

An RVQ codebook was first proposed by [2] to quantize a single signature and the codebook contains independent isotropically distributed vectors. This codebook design was motivated by the fact that the optimal signature is the eigenvector of an interference covariance matrix, which is isotropically distributed. Reference [9] generalizes the codebook design in [2] to quantize multiple signatures. For multiple users, an RVQ codebook consists of matrices whose columns are signature vectors for optimizing users and are independent isotropically distributed, and is denoted by

$$\mathcal{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \dots, \boldsymbol{V}_n\}. \quad (8)$$

The receiver selects the signature matrix from the codebook that maximizes the average SINR in (5)

$$\boldsymbol{S} = \arg\max_{\mathcal{V}} \mathrm{SINR}(\boldsymbol{V}_j) \quad (9)$$

for given channel and set of interfering signatures. We describe the exact steps in Algorithm 1. For RVQ, we must compute

Algorithm 1 Search algorithm for the optimal set of signatures in RVQ

```
SINR(S) = 0
for j = 1 : n do
    Compute SINR(V_j).
    if SINR(V_j) ≥ SINR(S) then
        S = V_j
        SINR(S) = SINR(V_j)
    end if
end for
return the optimal set of signatures S.
```

SINR for every signature matrices in the codebook to locate the optimal signature set. Thus, the search complexity grows linearly with number of entries in the codebook, which is a function of available feedback bits. To reduce the complexity, a tree-structured (TS) RVQ scheme was proposed by [7]. The codebook entries are organized into a tree and thus, the search complexity grows logarithmically with number of entries.

Here we modify the previously proposed TS codebook for a multiuser signature quantization. We start with an RVQ codebook with independent isotropically distributed *vectors*, which is denoted by

$$\mathcal{V}_T = \{\boldsymbol{v}_1, \quad \boldsymbol{v}_2, \ \cdots \ , \boldsymbol{v}_{nK_a}\}. \quad (10)$$

We note that for a subsequent performance comparison, we set a number of vector entries in $\mathcal{V}_T$ equal to a total number of column vectors in RVQ codebook $\mathcal{V}$. To build a tree, we apply a generalized Lloyd Algorithm (GLA) [7], [13]. First, we find two vector centroids $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ that minimize a sum of Euclidean distances between all vector entries to the centroids,

$$d = \sum_{\boldsymbol{v}_j \in \mathcal{V}_{T,1}} \|\boldsymbol{c}_1 - \boldsymbol{v}_j\| + \sum_{\boldsymbol{v}_k \in \mathcal{V}_{T,2}} \|\boldsymbol{c}_2 - \boldsymbol{v}_k\| \quad (11)$$

where the two quantization regions or clusters are given by

$$\mathcal{V}_{T,1} = \{\boldsymbol{v}_j \in \mathcal{V}_T : \|\boldsymbol{c}_1 - \boldsymbol{v}_j\| \leq \|\boldsymbol{c}_2 - \boldsymbol{v}_j\|\} \quad (12)$$

and

$$\mathcal{V}_{T,2} = \mathcal{V}_T \backslash \mathcal{V}_{T,1}. \quad (13)$$

We iterate between computing the sum distance d and finding the clusters until d converges. In the end, we obtain two child nodes, which associate with $\mathcal{V}_{T,1}$ and $\mathcal{V}_{T,2}$, respectively. Then, we move to the child node associated with $\mathcal{V}_{T,1}$ or $\mathcal{V}_{T,2}$ and repeat the process until we obtain all leaf nodes, which contain only one vector entry. With this algorithm, we eventually create a binary tree, which may not be balanced. Building a tree requires an extensive computing power especially for a large codebook. Fortunately, it can be done offline and thus, computation complexity should not be a problem.

With TS codebook, we can search for the optimal set of signatures by using Algorithm 2. For signature of user 1, we start at the root node and transverse the tree by comparing SINR($\boldsymbol{c}_1$) and SINR($\boldsymbol{c}_2$) where $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are centroids of the vector sets of the left and right child nodes, respectively. We then move to the node with the signature that gives larger SINR and repeat these steps until we reach the leaf node. For subsequent users, we start at the root node and follow the same steps, but we also account for signatures we obtain from earlier iterations (see steps 5 and 6). The complexity here relies

Algorithm 2 Search Algorithm for TS-RVQ

1: $\boldsymbol{S} = [\ ]$.
2: **for** $i = 1 : K_a$ **do**
3: Start at the root node of a TS-RVQ codebook with $\boldsymbol{S}$.
4: **while** Not a leaf node. **do**
5: $\boldsymbol{V}_L = [\boldsymbol{S} \quad \boldsymbol{c}_1]$.
6: $\boldsymbol{V}_R = [\boldsymbol{S} \quad \boldsymbol{c}_2]$.
7: **if** SINR($\boldsymbol{V}_L$) $\geq$ SINR($\boldsymbol{V}_R$) **then**
8: Move to the left child node.
9: $\boldsymbol{T} = \boldsymbol{V}_L$.
10: **else**
11: Move to the right child node.
12: $\boldsymbol{T} = \boldsymbol{V}_R$.
13: **end if**
14: **end while**
15: $\boldsymbol{S} = \boldsymbol{T}$.
16: **end for**
17: **return** $\boldsymbol{S}$

largely on SINR computation in step 7. For each iteration in step 2, $\boldsymbol{S}$ remains the same. Hence, part of SINR computation in step 7 that is associated with $\boldsymbol{S}$ may only be determined once. Only SINR computations related to column vector $\boldsymbol{c}_1$ or $\boldsymbol{c}_2$ for $\boldsymbol{V}_L$ and $\boldsymbol{V}_R$, respectively, need to be obtained at every node the algorithm visits. Thus, computing SINR($\boldsymbol{V}_L$) or SINR($\boldsymbol{V}_R$) take only $1/K_a$ of the complexity for computing SINR($\boldsymbol{V}_j$) in (5) for an RVQ codebook. We remark that the total complexity of the TS-RVQ will depend on a number of nodes or the tree's depth.

Next we show the performance of the proposed quantization scheme and compare with that of the original RVQ codebook.

IV. Numerical Results

Here we show some simulation results to illustrate the performance of RVQ and TS-RVQ for both ideal and fading reverse-link channels. Fig. 1 shows the SINR for an ideal channel with normalized feedback bit, which equals a total number of feedback bits per processing gain (B/N) for both RVQ and TS-RVQ. We set $N = 10$, $K_a = 2$, $K_b = 2$, and background signal-to-noise ratio (SNR) $= 1/\sigma_n^2 = 10$ dB. Thus, for $B/N = 1$, $B = 10$ and a number of column vectors in both codebooks equals 2×2^{10}. For larger parameters, a number of codebook entries will be enormous and hence, we will need a very powerful computer to perform simulations. However, the results shown with a relatively small system are still useful for predicting the performance of a larger system.

As expected, both feedback schemes give better performance as amount of feedback increases. For smaller feedback, TS-RVQ performs a bit better while for larger feedback, RVQ is expected to perform better with its exhaustive search. With 1 feedback bit per processing gain, the performance of RVQ is 1.5 dB better than that of TS-RVQ.

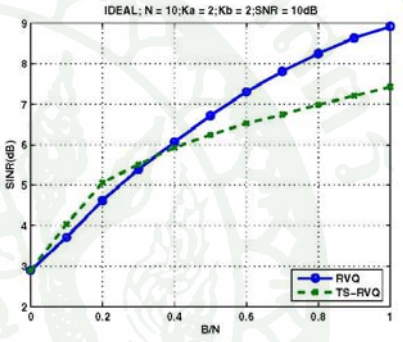


Fig. 1. Shown is an average SINR versus a normalized feedback bit for an ideal non-fading channel. Performance of RVQ and TS-RVQ are compared with $N = 10$, $K_a = 2$, $K_b = 2$, and SNR $= 10$ dB.

In Fig. 2, we show a number of SINR computations required to locate the optimal signature set against SINR for both RVQ and TS-RVQ. The parameters and settings are the same as those in previous figure. For small SINR, a search complexity of both schemes are comparable. As SINR increases (or as a number of feedback bits increases), the search complexity for RVQ increases exponentially while that for TS-RVQ increases linearly. We see that for SINR = 7.5 dB, TS-RVQ needs 20 computations while RVQ needs more than 80 computations. We note that the gap between complexity of both schemes will only widen as SINR increases. Thus, for a given performance, TS-RVQ offers a simpler quantization scheme than RVQ does.

Figs. 3 and 4 show SINR performance of RVQ and TS-RVQ and complexity of both schemes for a frequency-selective fading channel with path loss. We set a number of fading paths $L = 2$ for all users, path loss exponent $\alpha = 3$, $N = 10$, $K_a = 2$, and $K_b = 4$. From the figures, we observe similar performance trends to those for a non-fading channel. With half a feedback bit per processing gain, the performance of the

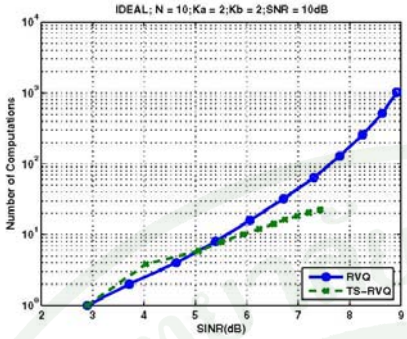


Fig. 2. Shown is a number of SINR computations versus a normalized feedback bit for an ideal non-fading channel with $N = 10$, $K_a = 2$, $K_b = 2$ and SNR $= 10$ dB.

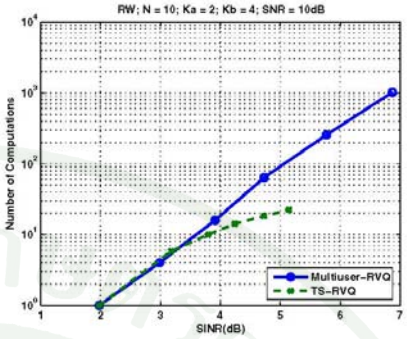


Fig. 4. A number of SINR computation needed for RVQ and TS-RVQ is shown with SINR for a frequency-selective fading channel with path loss. The system parameters are $N = 10$, $K_a = 3$, $K_b = 4$, $L = 3$, $\alpha = 3$ and SNR $= 10$ dB.

two schemes are approximately the same. However, RVQ is an order of magnitude more complex than TS-RVQ for SINR at 5 dB.

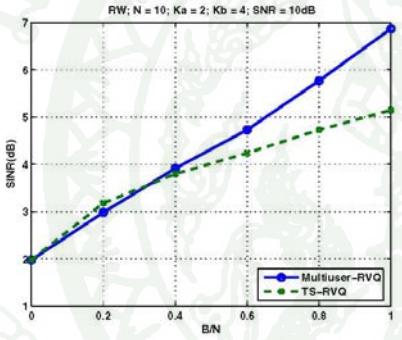


Fig. 3. An average SINR for both RVQ and TS-RVQ with normalized feedback bit is shown for a frequency-selective fading channel with path loss. The system parameters are $N = 10$, $K_a = 2$, $K_b = 4$, $L = 2$, $\alpha = 3$ and SNR $= 10$ dB.

V. Conclusions

We have proposed a tree structured RVQ scheme to quantize signatures for multiple users in DS-CDMA. TS-RVQ gives a lower average SINR for a given amount of feedback than RVQ does. However, complexity of RVQ increases exponentially with number of feedback bits while that of TS-RVQ increases linearly. For a given SINR, complexity of TS-RVQ can be an order of magnitude less than that of RVQ. Although the search for TS-RVQ is less complex, constructing a TS-RVQ codebook is much more computationally intensive than RVQ is . Fortunately, the codebook construction can be done offline. Here we only consider a reverse link where the receiver performs a centralized signature quantization for users. A quantization algorithm for a forward link, on the other hand, need to be distributed and will be our future work.

References

[1] G. S. Rajappan and M. L. Honig, "Signature sequence adaptation for DS-CDMA with multipath," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 384–395, Feb. 2002.

[2] W. Santipach and M. L. Honig, "Signature optimization for DS-CDMA with limited feedback," in *Proc. IEEE Int. Symp. on Spread-Spectrum Tech. and Appl. (ISSSTA)*, Prague, Czech Republic, Sep. 2002, pp. 180–184.

[3] W. Santipach, "Asymptotic performance of DS-CDMA with linear MMSE receiver and limited feedback," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Beijing, China, May 2008.

[4] W. Santipach, Y. Sun, and M. L. Honig, "Benefits of limited feedback for wireless channels," in *Proc. Allerton Conf. on Commun., Control, and Computing*, Monticello, Illinois, USA, Oct. 2003.

[5] D. J. Love, R. W. Heath, Jr., W. Santipach, and M. L. Honig, "What is the value of limited feedback for MIMO channels?" *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 54–59, Oct. 2004.

[6] W. Santipach and M. L. Honig, "Asymptotic capacity of beamforming with limited feedback," in *Proc. IEEE Int. Symp. on Info. Theory (ISIT)*, Chicago, Illinois, USA, Jun. 2004, p. 290.

[7] W. Santipach, "Tree-structured random vector quantization for beamforming in a multiantenna channel," in *Proc. ECTI-CON*, vol. I, Krabi, Thailand, May 2008, pp. 325–328.

[8] W. Santipach and M. L. Honig, "Capacity of a multiple-antenna fading channel with a quantized precoding matrix," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1218–1234, Mar. 2009.

[9] ——, "Signature optimization for CDMA with limited feedback," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3475–3492, Oct. 2005.

[10] D. J. Ryan, I. V. L. Clarkson, I. B. Collings, D. Guo, and M. L. Honig, "QAM and PSK codebooks for limited feedback MIMO beamforming," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Glasgow, Scotland, Jun. 2007.

[11] J. A. Dabin, N. Ni, A. M. Haimovich, E. Niver, and H. Grebel, "The effects of antenna directivity on path loss and multipath propagation in UWB indoor wireless channel," in *Proc. IEEE Conf. on Ultra Wideband Systems and Technologies*, Nov. 2003, pp. 305–309.

[12] T. S. Rappaport, *Wireless Communications*. Upper Saddle River, New Jersey, USA: Prentice Hall, 1996.

[13] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "Fast tree-structured nearest neighbor encoding for vector quantization," *IEEE Trans. Image Process.*, vol. 5, no. 2, pp. 398–404, Feb. 1996.

Kd-Tree Codebook for Limited Feedback CDMA

Kritsada Mamat and Wiroonsak Santipach
Department of Electrical Engineering
Faculty of Engineering, Kasetsart University
Bangkok, Thailand 10900
Email: {g5065253, wiroonsak.s}@ku.ac.th

***Abstract*—We propose a quantization scheme based on a Kd (K dimensional)-tree algorithm for signature sequence in a reverse-link direct sequence (DS)- code division multiple access (CDMA). With a few feedback bits, a receiver quantizes the optimal signature that minimizes interference for a desired user, and relays it to the user via an error-free feedback channel. A user performance depends on the quantization codebook and a number of feedback bits. We show the performance of the proposed Kd-tree codebook with a nearest neighbor criterion and derive the performance approximation. Also we modify the Kd-tree scheme to search for the entry in the codebook, which gives the least interference. Numerical examples show that the Kd-tree codebook performs near the optimal codebook with a only fraction of computational complexity.**

I. Introduction

To increase a user performance in a direct sequence (DS)-code division multiple access (CDMA), a signature sequence for a desired user needs to be optimized for current channel and interference conditions. A receiver, which can estimate channel state information (CSI), can compute the user's optimal signature and sends it to the user via a feedback channel. If the channel is changing slowly, the signature update needs not be frequent and can greatly improve the user performance [1]. A feedback channel is rate-limited and therefore, the user signature optimized at the receiver must be quantized before feedback back to the transmitter, which updates its signature sequence, accordingly. The resulting performance will depend on a quantization scheme used to quantize signature and a feedback rate.

References [1]–[5] have proposed codebook designs, which depend on channel and interference statistics. (In a multi-antenna channel, a signature refers to a set of spatial transmit coefficients.) Assuming that interfering signatures are independent and Gaussian distributed, we proposed a random vector quantization (RVQ) codebook [1], [6] whose entries are independent and isotropically distributed. RVQ is motivated by the fact that the optimal *unquantized* signature is the eigenvector of a interference covariance matrix, corresponding to the minimum eigenvalue. The distribution of the eigenvector is isotropic. In other words, the optimal signature is likely to point in any direction of the signal space. RVQ was shown to be optimal (i.e. minimizing interference) in a large system limit in which processing gain, number of interfering users, and feedback bits all tend to infinity with fixed ratios. Even though optimality is achieved only in a large system regime, RVQ performs close to the optimized codebook for a finite-size system as well [7].

This work was supported by the 2008 Telecommunications Research and Industrial Development Institute (TRIDI) scholarship and the Commission on Higher Education and Thailand Research Fund under grant MRG5080174.

To locate the optimal entry in an RVQ codebook, exhaustive search is required. Since a number of entries in an RVQ codebook grows exponentially with number of feedback bits, search complexity also grows exponentially as well. Reference [8] proposed a low-complexity quantization scheme based on a non-coherent detection technique. Elements for code entries in [8] are constrained to be PAM symbols for real vector entries or QAM symbols for complex ones. A search complexity of this PAM codebook is much less than RVQ and it performs very well for a large feedback. However, performance is lacking for small number of feedback bits. This dues to a nearest neighbor criterion, which PAM scheme employs to locate the selected entry. We note that the entry that is nearest to the optimal unquantized vector does not necessary minimize interference power.

In [9], we proposed to organize entries from RVQ codebook into a binary search tree. The entries are clustered in each step by the generalized Lloyd algorithm [10]–[12]. The associated search complexity for this tree-structured RVQ increases *linearly* with feedback bits. However, the performance of this tree-structured codebook is significantly inferior to the original RVQ codebook with exhaustive search. In this paper, we improve the tree-structured RVQ by applying a K-dimensional (Kd)-tree algorithm proposed by [13], [14]. Kd-tree represents a set of points in a K-dimensional space and supports a nearest neighbor query. First, we apply Kd-tree to locate the code entry in RVQ codebook, which is the nearest neighbor (closest in Euclidean distance) to the optimal unquantized vector. The associated performance is approximately equal to that of PAM codebook. We also derive the performance upper bound and show that it is a good approximation for the performance of Kd-tree. Then, we modify the objective of the search algorithm for Kd-tree from Euclidean distance to the optimal signature vector to an interference power. The Kd-tree search with modified objective performs very close to RVQ with exhaustive search and requires fewer orders of magnitude in computational complexity.

II. System Model

We consider a discrete-time reverse-link DS-CDMA with K users and processing gain N. Assuming an ideal nonfading

channel, the $N \times 1$ received vector is given by

$$\boldsymbol{r} = \sum_{k=1}^{K} \boldsymbol{s}_k b_k + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{s}_k$ is the $N \times 1$ signature vector for user k whose element is independent and Gaussian distributed with zero mean and variance $1/N$, b_k is the transmitted symbol for user k with zero mean and unit variance, and $\boldsymbol{n}$ is the $N \times 1$ additive white Gaussian noise vector with zero mean and covariance $\sigma_n^2 \boldsymbol{I}$ where $\boldsymbol{I}$ is an identity matrix.

We assume that a receiver for user 1 is a linear matched filter given by

$$\boldsymbol{c}_1 = \boldsymbol{s}_1 \tag{2}$$

and the associated output signal-to-interference plus noise ratio (SINR) for user 1 is given by [15]

$$\gamma_1 = \frac{1}{\boldsymbol{s}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_1 + \sigma_n^2} \tag{3}$$

where $\dagger$ denotes Hermitian transpose and $\boldsymbol{S}_1$ is the $N \times (K-1)$ signature matrix whose columns are the interfering signatures $\{\boldsymbol{s}_2, \ldots, \boldsymbol{s}_K\}$. The matched filter was shown to perform worse than other more complex receivers, e.g. linear MMSE receiver [15]. However, its performance can be improved significantly when it is combined with a feedback scheme with only few feedback bits [1].

As we can see from (3), the user performance is a function of signature $\boldsymbol{s}_1$. By adapting $\boldsymbol{s}_1$ to minimize the interference power given by $\boldsymbol{s}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_1$, we can improve the performance. Given the interfering signature matrix $\boldsymbol{S}_1$, the optimal $\boldsymbol{s}_1$ is the eigenvector of the interference covariance matrix $\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger$ corresponding to the smallest eigenvalue. In a typical wireless setting, the receiver can estimate the interference covariance during training and thus, can compute the optimal $\boldsymbol{s}_1$. With B bits, the receiver can quantize $\boldsymbol{s}_1$ and relay the quantized signature to user 1 via an error-free feedback channel. Then, user 1 adjusts its signature accordingly. Here we implicitly assume that channel is changing slowly and hence, a feedback from the receiver is useful.

A singular value decomposition gives

$$\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger = \sum_{i=1}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\dagger \tag{4}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ are the ordered eigenvalues and $\boldsymbol{u}_i$ is the corresponding ith eigenvector. With B feedback bits, the quantization codebook

$$\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{2^B}\} \tag{5}$$

where $\boldsymbol{v}_j$ is the $N \times 1$ codebook entry with $\|\boldsymbol{v}_j\| = 1$ and number of entries is 2^B. The optimal *unquantized* signature that minimizes the interference for user 1 is $\boldsymbol{u}_1$. We can quantize $\boldsymbol{u}_1$ with a nearest neighbor objective and the quantized signature is given by

$$\tilde{\boldsymbol{s}}_1 = \arg\min_{\boldsymbol{v}_j \in \mathcal{V}} \|\boldsymbol{u}_1 - \boldsymbol{v}_j\|^2 = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} \boldsymbol{u}_1^\dagger \boldsymbol{v}_j, \tag{6}$$

which follows from the fact both $\|\boldsymbol{v}_j\| = \|\boldsymbol{u}_1\| = 1$. This is a classical vector quantization problem to which there are many solutions [10] (see references therein). If the objective is changed to minimizing the angle between $\boldsymbol{u}_1$ and $\boldsymbol{v}_j$ denoted by ϕ_j, then the quantized vector is given by

$$\tilde{\boldsymbol{s}}_1 = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} \cos^2 \phi_j = \arg\max_{\boldsymbol{v}_j \in \mathcal{V}} (\boldsymbol{u}_1^\dagger \boldsymbol{v}_j)^2. \tag{7}$$

With statistics of the eigenvector, we can apply the Lloyd-Max iterative algorithm [11], [12] to obtain the optimal $\mathcal{V}$, which minimizes the expected Euclidean distance between $\boldsymbol{u}_1$ and $\tilde{\boldsymbol{s}}_1$ or the angle between $\boldsymbol{u}_1$ and $\tilde{\boldsymbol{s}}_1$. However, either $\tilde{\boldsymbol{s}}_1$ or $\tilde{\boldsymbol{s}}_1$ may not necessarily minimizes the interference power.

To minimize the interference power, the receiver selects

$$\hat{\boldsymbol{s}}_1 = \arg\min_{\boldsymbol{v}_j \in \mathcal{V}} \boldsymbol{v}_j^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{v}_j \tag{8}$$

for given $\boldsymbol{S}_1$. If the interfering signatures are i.i.d., we have shown that RVQ codebook, which contains independent and isotropically distributed vectors is optimal in a large system limit [1], [6]. The large system limit refers to the limit in which N, K, and B goes to infinity with a fixed normalized load $\bar{K} = K/N$ and normalized feedback bits $\bar{B} = B/N$. RVQ codebook is simple to construct. However, RVQ codebook and many codebooks proposed in the literature require exhaustive search to locate the optimal entry. This can pose a serious problem on complexity if B is large. Next we describe our proposed quantization codebook, which demands much less computational complexity, but sacrifices minimal performance.

III. Kd-Tree Codebook

We organize code entries in an RVQ codebook using Kd-tree algorithm proposed by [13], [14]. The algorithm produces an unbalanced binary search tree by clustering the code entries by one dimension at a time in each step. Steps to create Kd-tree is shown in Algorithms 1 and 2. We start at the root node with all 2^B entries from RVQ codebook $\mathcal{V} = \{\boldsymbol{v}_j\}$. We find a median for the *first* elements for all vectors and select the element that is the closest to the median. We call that element the *pivot*. Then vectors are divided into two groups whether the first element is greater or less than the pivot. We store the vector whose first element is the pivot at the root node and create the left- and right-child nodes with two groups of vectors. We move to the left-child node and find the pivot for the *second* dimension elements of the vectors in that node. We divide the group of vectors according to the new pivot. Each time we move down the tree, we operate on the next dimension. We iterate until all nodes contain one code entry and obtain a binary search tree with 2^B nodes at the end. Building the tree is relatively fast since we examine one dimension at a time. Also it can be done offline and thus does not add burden on the transmitter-receiver pair.

A. Finding Nearest Neighbor

Given $\boldsymbol{u}_1$, Kd-tree search shown in Algorithms 3 and 4 [13], [14] will locate the entry that is the nearest neighbor to $\boldsymbol{u}_1$. Algorithm 3 starts at the root and move down to the left- or

Algorithm 1 Kd-TREE construction [13], [14]

1: Start with RVQ codebook with 2^B N-dimensional vectors.
2: Store all entries at the root of a binary tree.
3: Set $n = 1$ where n is the index of element in an N-dimensional vector.
4: kd-tree = build_tree(root node, n)
5: **return** kd-tree

Algorithm 2 function: kd-tree = build_tree(node, n) [13], [14]

1: **if** Node contains only one vector. **then**
2: **return**
3: **else**
4: **if** $n > N$ **then**
5: $n = 1$
6: **end if**
7: Sort codebook entries by the nth dimension.
8: Select the median as pivot element.
9: Use pivot to divide vectors in two groups.
10: $n = n + 1$
11: build_tree(left-child node, n)
12: build_tree(right-child node, n)
13: **end if**

right-child nodes by comparing the element in $\boldsymbol{u}_1$ in specified dimension with the pivot of the present node. Once we reach the leaf node, we have the candidate. Algorithm 4 makes certain that the candidate is the nearest neighbor by comparing the distance of the candidate and $\boldsymbol{u}_1$ with that of other nodes and $\boldsymbol{u}_1$. We remark that the selected entry may not minimize the interference and is the suboptimal entry.

Algorithm 3 Kd-tree search for the nearest neighbor

1: Start at the root of a Kd-tree codebook with $\boldsymbol{u}_1$.
2: Set $n = 1$ where n is the index of element in an N-dimensional vector.
3: **while** Not a leaf node. **do**
4: **if** $n > N$ **then**
5: $n = 1$
6: **end if**
7: $\boldsymbol{p}$ = current node's vector
8: **if** $p_n > u_{1,n}$ **then**
9: Move to the left-child node.
10: **else**
11: Move to the right-child node.
12: **end if**
13: $n = n + 1$
14: **end while**
15: $\boldsymbol{s}$ = vector from leaf node
16: $D = \|\boldsymbol{s} - \boldsymbol{u}_1\|$
17: $n = 1$
18: $\boldsymbol{s}$ = check_distance(root node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
19: **return** $\boldsymbol{s}$

Performance of the Kd-tree codebook is difficult to evaluate

Algorithm 4 function: $\boldsymbol{s}$ = check_distance(node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)

1: **if** A leaf node **then**
2: $\boldsymbol{s}_c$ = vector of a leaf node
3: $D_c = \|\boldsymbol{s}_c - \boldsymbol{u}_1\|$
4: **if** $D_c < D$ **then**
5: $D = D_c$
6: $\boldsymbol{s} = \boldsymbol{s}_c$
7: **return** $\boldsymbol{s}$
8: **end if**
9: **end if**
10: $D_p = p_n^2 - u_{1,n}^2$
11: **if** $D_p < D$ **then**
12: **if** $n > N$ **then**
13: $n = 1$
14: **end if**
15: **if** $p_n > u_{1,n}$ **then**
16: $n = n + 1$
17: check_distance(left-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
18: **else**
19: $n = n + 1$
20: check_distance(right-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
21: **end if**
22: **else**
23: $n = n + 1$
24: check_distance(left-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
25: check_distance(right-child node, n, $\boldsymbol{u}_1$, D, $\boldsymbol{s}$)
26: **end if**

analytically. Thus, we look to approximate the performance instead. Clearly, RVQ with the exhaustive search is guaranteed to find the optimal entry and therefore, its performance gives an upper bound to that of Kd-tree. The interference power with RVQ codebook is given by

$$\tilde{I}_{\text{rvq}} = \tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \tilde{\boldsymbol{s}}_1 \tag{9}$$

$$= \lambda_1 (\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 + \sum_{i=2}^{N} \lambda_i (\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_i)^2 \tag{10}$$

where $\tilde{\boldsymbol{s}}$ is the quantized signature for user 1 given by (8) and the second equation is obtained by substituting (4). Finding expectation for $\tilde{I}_{\text{rvq}}$ for finite N, K, and B is not tractable due to very complicated distribution expressions. However, as $(N, K, B) \to \infty$, an eigenvalue distribution of $\boldsymbol{S}_1 \boldsymbol{S}_1^\dagger$ converges to a deterministic function and the smallest eigenvalue λ_1 converges to [16]

$$\lambda_1 \longrightarrow \lambda_{\min}^{\infty} = \begin{cases} 0 & : \ 0 \le \bar{K} \le 1 \\ (1 - \frac{1}{\sqrt{\bar{K}}})^2 & : \ \bar{K} > 1 \end{cases}. \tag{11}$$

We have shown in [17] that as $(N, K, B) \to \infty$

$$(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 1 - 2^{-2\bar{B}} \tag{12}$$

almost surely. The large system $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2$ increases with $\bar{B}$. With infinite feedback ($\bar{B} = \infty$), we can relay $\boldsymbol{u}_1$ unquantized and $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 1$. On the other hand, with no feedback, $\tilde{\boldsymbol{s}}$ is randomly selected and $(\tilde{\boldsymbol{s}}_1^\dagger \boldsymbol{u}_1)^2 \to 0$.

Next, we analyze each term in the sum in (10). Since $\boldsymbol{u}_1$ are orthogonal to all $\boldsymbol{u}_i$, as B increases, $\tilde{\boldsymbol{s}}$ and $\boldsymbol{u}_i$ is getting closer be perpendicular. We can show that

Lemma 1: For large N with fixed $\bar{K}$ and $\bar{B}$,

$$(\tilde{\boldsymbol{s}}_1^{\dagger}\boldsymbol{u}_i)^2 = \frac{1}{N}2^{-2\bar{B}} + \mathcal{O}(\frac{1}{N^2}). \tag{13}$$

A core of the proof is to derive the distribution of $(\tilde{\boldsymbol{s}}_1^{\dagger}\boldsymbol{u}_i)^2$, which is equivalent to a partial surface of the N-dimensional unit hypersphere. We substitute (13) into the sum on the right-hand side of (10) and obtain

$$\sum_{i=1}^{N}\lambda_i(\tilde{\boldsymbol{s}}_1^{\dagger}\boldsymbol{u}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\lambda_i 2^{-2\bar{B}} + \mathcal{O}(\frac{1}{N}). \tag{14}$$

As $(N, K) \to \infty$, it can be shown that [16]

$$\frac{1}{N}\sum_{i=1}^{N}\lambda_i \to \bar{K}. \tag{15}$$

Applying Lemma 1 and combining (10)–(15), we have

Theorem 1: As $(N, K, B) \to \infty$ with fixed $\bar{K}$ and $\bar{B}$,

$$\bar{I}_{\text{rvq}} \to \bar{I}_{\text{rvq}}^{\infty} \tag{16}$$

$$= \begin{cases} \bar{K}2^{-2\bar{B}} & : 0 \le \bar{K} \le 1 \\ (1-\frac{1}{\sqrt{\bar{K}}})^2(1-2^{-2\bar{B}}) + \bar{K}2^{-2\bar{B}} & : \bar{K} > 1 \end{cases} \tag{17}$$

almost surely.

Hence, the associated SINR for RVQ that quantizes $\boldsymbol{u}_1$ is given by

$$\bar{\gamma}_{\text{rvq}}^{\infty} = \frac{1}{\bar{I}_{\text{rvq}}^{\infty} + \sigma_n^2}. \tag{18}$$

Thus, performance of Kd-tree is upper bounded by the result in Theorem 1.

B. Minimizing Interference Power

We modify the objective function in the search algorithm in Algorithms 3 and 4 to be the interference power for a given interfering signature matrix $\boldsymbol{S}_1$. The modified search shown in Algorithms 5 and 6 is in the same spirit as the earlier search algorithm. At each step, we look for the entry, which gives the interference power, which is closest to the minimum.

These algorithms however are not guaranteed to find the entry that minimizes the interference power since the tree we used is constructed with the nearest neighbor criterion. The solution will be suboptimal to the exhaustive search. However, as we will observe from simulation results, this Kd-tree with modified search performs very close to RVQ with exhaustive search.

Again, analyzing the performance of this scheme exactly is not tractable. So, we look to analyze the performance of RVQ. Reference [18] has shown that

$$\hat{I}_{\text{rvq}} = \hat{\boldsymbol{s}}^{\dagger}\boldsymbol{S}_1\boldsymbol{S}_1^{\dagger}\hat{\boldsymbol{s}} \to \hat{I}_{\text{rvq}}^{\infty} \tag{19}$$

and $\hat{I}_{\text{rvq}}^{\infty}$ is a function of $\bar{K}$ and $\bar{B}$, which can be used to approximate the performance of Kd-tree.

Algorithm 5 Kd-tree search for the minimum interference

1: Start at the root of a Kd-Tree codebook with interfering signature matrix $\boldsymbol{S}_1$.
2: Set $n = 1$ where n is the index of element in an N-dimensional vector.
3: **while** Not a leaf node **do**
4: **if** $n > N$ **then**
5: $n = 1$
6: **end if**
7: $\boldsymbol{p}_L^{(n)} = [0 \quad 0 \ \ldots \ p_{L,n+1} \ \ldots \ 0 \quad 0]^{\dagger}$ where $\boldsymbol{p}_L$ is the vector of the left-child node.
8: $\boldsymbol{p}_R^{(n)} = [0 \quad 0 \ \ldots \ p_{R,n+1} \ \ldots \ 0 \quad 0]^{\dagger}$ where $\boldsymbol{p}_R$ is the vector of the right-child node.
9: **if** $\boldsymbol{p}_L^{(n)\dagger}\boldsymbol{S}_1\boldsymbol{S}_1^{\dagger}\boldsymbol{p}_L^{(n)} < \boldsymbol{p}_R^{(n)\dagger}\boldsymbol{S}_1\boldsymbol{S}_1^{\dagger}\boldsymbol{p}_R^{(n)}$ **then**
10: Move to the left child node.
11: **else**
12: Move to the right child node.
13: **end if**
14: $n = n + 1$
15: **end while**
16: $\boldsymbol{s}$ = vector from leaf node.
17: $I = \boldsymbol{s}^{\dagger}\boldsymbol{S}_1\boldsymbol{S}_1^{\dagger}\boldsymbol{s}$
18: $n = 1$
19: $\boldsymbol{s}$ = check_interference(root node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
20: **return** $\boldsymbol{s}$

IV. Numerical Results

Fig. 1 shows the SINR for user 1 versus normalized feedback bits $\bar{B}$ for loads $\bar{K} = 0.5$ and $\bar{K} = 1.2$, and SNR = 10 dB. Simulation results shown are for RVQ and Kd-tree codebooks with closest in angle and nearest neighbor criterions, respectively. As expected, SINR increases with $\bar{B}$. User 1 can achieve 7 dB and 4 dB with only one feedback bit for $\bar{K} = 0.5$ and $\bar{K} = 1.2$ as compared to 3 dB and -1 dB for zero feedback. Also shown is the asymptotic performance for RVQ with closest in angle criterion with $(N, K, B) \to \infty$. We can see that the large system limit approximates the performance of a small-size system very well. We expect a gap between simulation and large system results to narrow as N increases.

Figs. 2 and 3 show the performance of RVQ with closest in angle, and Kd-tree and PAM with nearest neighbor. All three performs approximately the same for a given feedback except when $\bar{B} < 1$, PAM codebook is not valid. We note that for $\bar{B}$ between 1 and 2, PAM performs a bit worse than the other two do. In Fig. 3, we show a number of inner product computations required for all three codebooks to locate the selected entries for a given SINR. We see that RVQ with exhaustive search requires significantly larger number of computations. For SINR at 7 dB, the complexity of RVQ is four orders of magnitude larger than that of Kd-tree and PAM. For low to moderate SINR, PAM codebook is the least complex while for the other range, it is unclear. Due to finite storage and computational power, we do not have simulation

Algorithm 6 function: s = check_interference(node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)

1: **if** Leaf node. **then**
2: $\boldsymbol{s}_c$ = vector from leaf node.
3: $I_c = \boldsymbol{s}_c^\dagger \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{s}_c$
4: **if** $I_c < I$ **then**
5: $I = I_c$
6: $\boldsymbol{s} = \boldsymbol{s}_c$
7: **return** $\boldsymbol{s}$
8: **end if**
9: **end if**
10: $\boldsymbol{p}_c^{(n)} = [0 \quad 0 \ \ldots \ p_{c,n} \ \ldots \ 0 \quad 0]^\dagger$ where $\boldsymbol{p}_c$ is the vector of the current node.
11: $I_p = \boldsymbol{p}_c^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_c^{(n)}$
12: **if** $I_p < I$ **then**
13: **if** $n > N$ **then**
14: $n = 1$
15: **end if**
16: **if** $\boldsymbol{p}_L^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_L^{(n)} < \boldsymbol{p}_R^{(n)\dagger} \boldsymbol{S}_1 \boldsymbol{S}_1^\dagger \boldsymbol{p}_R^{(n)}$ **then**
17: $n = n + 1$
18: check_interference(left-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
19: **else**
20: $n = n + 1$
21: check_interference(right-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
22: **end if**
23: **else**
24: $n = n + 1$
25: check_interference(left-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
26: check_interference(right-child node, n, $\boldsymbol{S}_1$, I, $\boldsymbol{s}$)
27: **end if**

results for larger $\bar{B}$ for both RVQ and Kd-tree. For large feedback, Kd-tree needs significantly larger storage for code entries while PAM does not have this issue.

In Fig. 4, we have SINR from three quantization schemes: RVQ and Kd-tree with minimizing interference and PAM with nearest neighbor for $\bar{K} = 0.5$ and SNR $= 10$ dB. We see that Kd-tree performs almost the same as RVQ does while PAM with nearest neighbor performs significantly worse for smaller $\bar{B}$. At $\bar{B} = 1$, a gap between quantizing to minimize interference and to find the nearest neighbor is as large as 5 dB. We also show the large system performance for RVQ derived by [18], which seems to approximate performance of a finite-size system very closely.

We also examine computational complexity of each quantization schemes with Fig. 5. The PAM codebook, which finds the nearest neighbor to the optimal eigenvector is the least complex. However, from Fig. 4, its performance is the worst for a given feedback bits. So, what codebook to use depends on the system need and requirement.

V. Conclusions

We have presented different quantization codebooks to quantize the user signature in reverse-link CDMA. Our main focus is Kd-tree codebook, which was motivated by the need

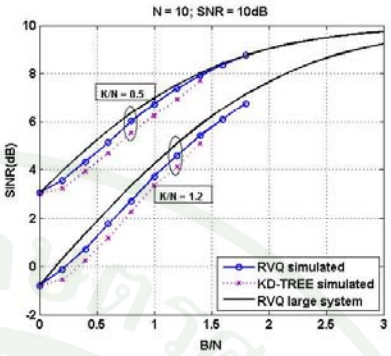


Fig. 1. SINR versus normalized feedback bits are shown for RVQ with closest in angle criterion and Kd-tree with nearest neighbor criterion.

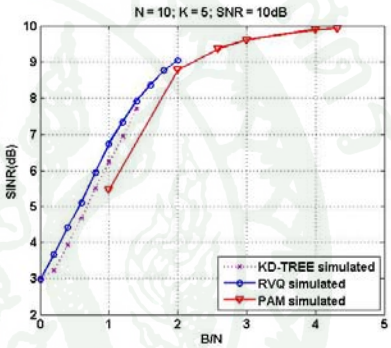


Fig. 2. SINR versus normalized feedback bits are shown for RVQ with closest in angle criterion, and Kd-tree and PAM with nearest neighbor criterion.

to lessen search complexity. Kd-tree codebook was shown to perform close to RVQ with significantly less search complexity. We also derived the performance upper bound for Kd-tree with nearest neighbor. The bound was shown to be a good approximation for the actual performance. Numerical examples show tradeoff between performance and complexity for different codebooks. RVQ with exhaustive performs best for a given feedback while PAM is the least complex. Kd-tree is a compromise. What codebook to use will depend on system's needs and requirements.

We can straightforwardly extend our model to include fading and other channel characteristics. We expect to see similar performance behaviors for all three codebooks. Here we assume that the receiver knows signatures of other users (or CSI) perfectly. The performance will degrade with the estimation of CSI or interference information.

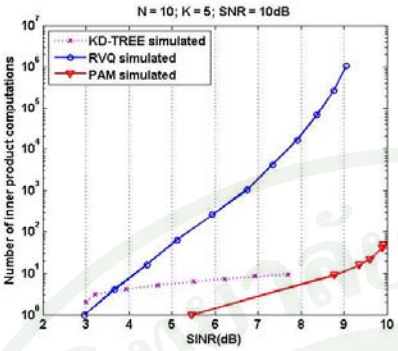


Fig. 3. A number of inner product computations is displayed for RVQ with closest in angle criterion, and Kd-tree and PAM with nearest neighbor criterion for a given SINR.

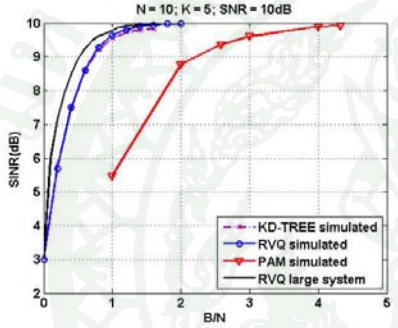


Fig. 4. SINR versus normalized feedback bits are shown for RVQ and Kd-tree with minimizing interference, and PAM with nearest neighbor criterion.

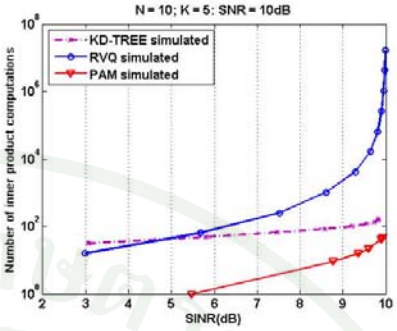


Fig. 5. A number of inner product computations is displayed for RVQ and Kd-tree with minimizing interference and PAM with nearest neighbor criterion for a given SINR.

REFERENCES

[1] W. Santipach and M. L. Honig, "Signature optimization for CDMA with limited feedback," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3475–3492, Oct. 2005.

[2] D. J. Love and R. W. Heath, Jr., "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2745, Oct. 2003.

[3] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple antenna systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2562–2579, Oct. 2003.

[4] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, "Efficient use of side information in multiple antenna data transmission over fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1423–1436, Oct. 1998.

[5] J. C. Roh and B. D. Rao, "Transmit beamforming in multiple-antenna systems with finite rate feedback: A VQ-based approach," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1101–1112, Mar. 2006.

[6] W. Santipach and M. L. Honig, "Signature optimization for DS-CDMA with limited feedback," in *Proc. IEEE Int. Symp. on Spread-Spectrum Tech. and Appl. (ISSSTA)*, Prague, Czech Republic, Sep. 2002, pp. 180–184.

[7] D. J. Love, R. W. Heath, Jr., W. Santipach, and M. L. Honig, "What is the value of limited feedback for MIMO channels?" *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 54–59, Oct. 2004.

[8] D. J. Ryan, I. V. L. Clarkson, I. B. Collings, D. Guo, and M. L. Honig, "Qam and psk codebooks for low-complexity limited feedback mimo beamforming," *IEEE Trans. Commun.*, vol. 57, no. 4, pp. 1184–1196, Apr. 2009.

[9] K. Mamat and W. Santipach, "Multiuser signature quantization with tree-structured codebook in DS-CDMA," in *Proc. ECTI-CON*, vol. 2, Pattaya, Thailand, May 2009, pp. 870–873.

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer, 1991.

[11] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory*, vol. 6, pp. 7–12, Mar. 1960.

[12] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[13] J. L. Bentley, "Multidimisional divide and conquer," *Communications of the ACM*, vol. 22, no. 4, pp. 214–229, 1980.

[14] ——, "K-d trees for semidynamic point sets," in *Proc. the Sixth Annual Symposium on Computational Geometry*, Berkley, California, 1990, pp. 187 – 197.

[15] S. Verdú and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 622–640, Mar. 1999.

[16] A. M. Tulino and S. Verdú, "Random matrix theory and wireless communications," *Foundations and Trends in Communications and Information Theory*, vol. 1, no. 1, pp. 1–182, 2004.

[17] W. Santipach and M. L. Honig, "Capacity of a multiple-antenna fading channel with a quantized precoding matrix," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1218–1234, Mar. 2009.

[18] W. Dai, Y. Liu, and B. Rider, "The effect of finite rate feedback on CDMA signature optimization and MIMO beamforming vector selection," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3651–3669, Aug. 2009.

ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นายกฤษฎา มามาตร
วัน เดือน ปี ที่เกิด	1 มีนาคม 2527
สถานที่เกิด	อ. สามชุก จ. สุพรรณบุรี
ประวัติการศึกษา	วศ. บ. (วิศวกรรมไฟฟ้า) มหาวิทยาลัยบูรพา
ตำแหน่งหน้าที่การงานปัจจุบัน	นิสิตปริญญาโท มหาวิทยาลัยเกษตรศาสตร์ บางเขน
สถานที่ทำงานปัจจุบัน	อาคารวิศวกรรมศาสตร์ 60 ปี ชั้น 5 ห้อง 505/7
ผลงานดีเด่นและรางวัลทางวิชาการ	- ได้รับการตีพิมพ์ผลงานเรื่อง “Multiuser Signature Quantization With Tree-Structured Codebook in DS-CDMA” ใน ECTI CON 2009 International Conference, pp 870-873, May 2009, Pattaya , Thailand. - ได้รับการตีพิมพ์ผลงานเรื่อง “Kd-tree codebook for limited feedback CDMA” ในงาน ICT 2010 International Conference, April 2010, Doha Qatar.
ทุนการศึกษาที่ได้รับ	สถาบันวิจัยและพัฒนาอุตสาหกรรมโทรคมนาคม (TRIDI)