การพัฒนาเทคนิคเวคเตอร์ควอนไทซ์เซชั่นสำหรับการบีบอัดสัญญาณเสียง

ENHANCEMENT VECTOR QUANTIZATION TECHNIQUE FOR SPEECH COMPRESSION

อรอนงค์ วิริยานุรักษ์นคร

ORNANONG WIRIYANURUKNAKORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

พ.ศ. 2553

การพัฒนาเทคนิคเวคเตอร์ควอนไทซ์เซชั่นสำหรับการบีบอัดสัญญาณเสียง

อรอนงค์ วิริยานุรักษ์นคร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
พ.ศ. 2553

ENHANCEMENT VECTOR QUANTIZATION TECHNIQUE
FOR SPEECH COMPRESSION

ORNANONG WIRIYANURUKNAKORN

A THESIS SUMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF ENGINEERING
IN ELECTRICAL ENGINEERING DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
2010

วิทยานิพนธ์ฉบับนี้เป็นงานวิจัยที่เกิดจากการค้นคว้าและวิจัยขณะที่ข้าพเจ้าศึกษาอยู่ในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ดังนั้นงานวิจัยในวิทยานิพนธ์ฉบับนี้ถือเป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรีและข้อความต่างๆ ในวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอรับรองว่าไม่มีการคัดลอกหรือนำงานวิจัยของผู้อื่นมานำเสนอในชื่อของข้าพเจ้า

นางสาวอรอนงค์ วิริยานุรักษ์นคร

COPYRIGHT © 2010 ลิขสิทธิ์ พ.ศ. 2553
FACULTY OF ENGINEERING คณะวิศวกรรมศาสตร์
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ใบรับรองวิทยานิพนธ์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

หัวข้อวิทยานิพนธ์	การพัฒนาเทคนิคเวคเตอร์ควอนไทซ์เซชั่นสำหรับการบีบอัดสัญญาณเสียง ENHANCEMENT VECTOR QUANTIZATION TECHNIQUE FOR SPEECH COMPRESSION
ชื่อนักศึกษา	นางสาวอรอนงค์ วิริยานุรักษ์นคร
รหัสประจำตัว	114870402009-4
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ดร.จักรี ศรีนนท์ฉัตร
วัน เดือน ปี ที่สอบ	11 กันยายน 2553
สถานที่สอบ	ห้องรวงข้าว ชั้น 7 อาคารเฉลิมพระเกียรติ 80 พรรษา 5 ธันวาคม 2550 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(รองศาสตราจารย์ ดร.สมเกียรติ อุดมหรรษากุล)

.... กรรมการ

(ดร.สุรินทร์ แหงมงาม)

.... กรรมการ

(ดร.อำนวย เรืองวารี)

.... กรรมการ

(ดร.จักรี ศรีนนท์ฉัตร)

....

(ผู้ช่วยศาสตราจารย์ ดร. สมชัย หิรัญวโรดม)

คณบดีคณะวิศวกรรมศาสตร์

หัวข้อวิทยานิพนธ์	การพัฒนาเทคนิคเวคเตอร์ควอนไทซ์เซชั่นสำหรับการบีบอัดสัญญาณเสียง
นักศึกษา	นางสาวอรอนงค์ วิริยานุรักษ์นคร
รหัสประจำตัว	114870402009-4
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า (วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม)
ปีการศึกษา	2553
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ดร. จักรี ศรีนนท์ฉัตร

บทคัดย่อ

การบีบอัดสัญญาณเสียงพูดที่มี bit rate ต่ำๆ นิยมใช้รูปแบบการทำนายค่าพารามิเตอร์ (Linear Predictive Parameters model) หรือการเข้ารหัสแบบเชิงเส้น (Linear Predictive Coder : LPC) ซึ่งพารามิเตอร์นี้จะแสดงถึงค่าคุณลักษณะสเปกตรัมของสัญญาณเสียง (speech spectrum) ช่วงสั้น แต่พารามิเตอร์ LPC จะไม่ค่อยมีประสิทธิภาพสำหรับการควอนไตร์เซชั่น จึงส่งค่าพารามิเตอร์ LPC แปลงเป็นค่าพารามิเตอร์ของ LSF (Line Spectrum Frequencies) หรือ LSP (Line Spectrum Pairs) เพื่อปรับปรุงการเข้ารหัสสัญญาณเสียงพูดให้มีประสิทธิภาพดีขึ้น

ในงานวิจัยนี้ ได้ศึกษาเทคนิควิธีการบีบอัดสัญญาณเสียงพูดภาษาไทย โดยพัฒนาจากการเข้ารหัสสัญญาณเสียงพูดแบบ LPC-10 ด้วยการแปลงเป็นค่าพารามิเตอร์ LSP เพื่อหาคุณลักษณะเด่น

ผลการวิจัยพบว่า การบีบอัดสัญญาณเสียงพูดด้วยเทคนิคนี้สามารถลดจำนวนบิตเรตได้ 5.5-37% โดยลดค่าบิตในพารามิเตอร์ p1-p4 ทั้งในส่วนของสัญญาณเสียง Voice และ Unvoice งานวิจัยในอนาคตจึงมีแนวความคิดที่จะพัฒนาจำนวน bit rate ทางด้านความถี่สูงให้มีค่าเท่ากับสัญญาณเสียงต้นฉบับ โดยใช้หลักการของ Post-Processing

คำสำคัญ : การประมวลผลสัญญาณเสียง , การบีบอัดเสียงพูด , การเข้ารหัสเสียงพูด , สัมประสิทธิ์คู่-เส้นสเปกตรัม , Clustering , KSOFM

Thesis Title :	ENHANCEMENT VECTOR QUANTIZATION TECHNIQUE FOR SPEECH COMPRESSION
Student Name :	Miss Ornanong Wiriyanuruknakorn.
Student ID :	114870402009-4
Degree Award :	Master of Engineering
Study Program :	Electrical Engineering (Electronics and Telecommunication Engineering)
Academic year :	2010
Thesis Advisor :	Dr. Jakkree Srinonchat

ABSTRACT

Linear Predictive Parameters (LPC) Model is wildly used to represent speech future for speech coding. These parameters extract to speech spectrum in short term corresponding that varies from 0 to ∞. The consequence of this technique provide the efficiency quantization less than Line Spectrum Frequencies which takes corresponding between 0 to 1.

This research uses the LPFs to be the speech feature for speech coding system. A Finite State Vector Quantization, which used the Kohonen Self Organizing Feature Map, is used to improve the quantization technique for reducing bit rate of speech parameters.

The results show that the technique can be reduce bit rate to 5.5-37%. However the quality of speech signal is reduced in the term of high frequency when compares with the original speech. In the future work, the post-processing will be research to improve the quality of speech signal in term of high frequency.

Keywords : Speech Processing , Speech Compression , Speech Coding , Line Spectral Pairs , Clustering , KSOFM

กิตติกรรมประกาศ

ขอขอบคุณ ดร.จักรี ศรีนนท์ฉัตร อาจารย์ที่ปรึกษาและอาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่ให้คำแนะนำ ตรวจสอบ ควบคุม และเอาใจใส่ จนวิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณห้องปฏิบัติการวิจัยทางด้านการประมวลผลสัญญาณ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี และห้องปฏิบัติการดิจิตอล มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ ที่ให้การสนับสนุนเครื่องมือและอุปกรณ์การทดลอง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ที่ให้ความช่วยเหลือ ให้ข้อคิดเห็นต่างๆ และให้กำลังใจ ในการทำงานวิจัยตลอดมา

ท้ายที่สุด ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และทุกๆ คนในครอบครัว ผู้เป็นที่รัก ผู้ให้กำลังใจเสมอมา

อรอนงค์ วิริยานุรักษ์นคร

11 กันยายน 2553

สารบัญ

หน้า

บทคัดย่อภาษาไทย ก
บทคัดย่อภาษาอังกฤษ ข
กิตติกรรมประกาศ ค
สารบัญ ง
สารบัญตาราง ฉ
สารบัญรูป ช
รายการคำอธิบายสัญลักษณ์และคำย่อ ญ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 ความมุ่งหมายและวัตถุประสงค์ 2
1.3 ขอบเขตของการศึกษา 2
1.4 ขั้นตอนการศึกษา 2
1.5 ข้อจำกัดของการศึกษา 2
1.6 ประโยชน์ที่คาดว่าจะได้รับ 2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3
2.1 การวิเคราะห์สัญญาณเสียงพูด 3
2.2 โครงข่ายประสาทเทียมอัจฉริยะ 9
2.3 การประเมินคุณภาพของข้อมูลเสียงพูด 19
2.4 งานวิจัยที่เกี่ยวข้อง 19
บทที่ 3 วิธีการดำเนินงานวิจัย 22
3.1การเก็บบันทึกเสียงพูด 23
3.2 การสร้างระบบการบีบอัดสัญญาณเสียงพูด 24
3.3 การออกแบบโครงข่ายประสาทเทียม 30
3.4 การตรวจสอบคุณลักษณะทางสัญญาเสียงพูด 52

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการวิจัย	55
4.1 ผลการทดสอบการสร้างโค้ดบุ๊ค	55
4.2 ผลการทดสอบคุณภาพเสียงเชิงเวลา	60
4.3 ผลการทดสอบคุณภาพเสียงเชิงความถี่	65
4.4 ผลการทดสอบคุณภาพเสียงโดยใช้ SNR	67
4.5 ผลการประเมินคุณภาพเสียงพูด MOS	68
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	70
5.1 สรุปผลการวิจัย	70
5.2 ข้อเสนอแนะ	71
เอกสารอ้างอิง	72
ภาคผนวก	74
ผลงานตีพิมพ์เผยแพร่	74
ประวัติผู้เขียน	85

สารบัญตาราง

ตารางที่	หน้า
2.1 ค่า MOS ที่เหมาะสมกับการใช้งานในระบบต่างๆ	19
2.2 รายละเอียดวิธีการให้คะแนนในการวัด MOS	19
4.1 การแบ่งเฟรมของเสียง	55
4.2 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 128	56
4.3 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 256	57
4.4 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 512	58
4.5 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 1,024	59
4.6 ผลการบีบอัดเสียงพูดเชิงจำนวนตามขนาดโค้ดบุ๊ค	60
4.7 ผลการบีบอัดเสียงพูดเชิงเวลาที่ใช้โค้ดบุ๊ค 128	60
4.8 ผลการบีบอัดเสียงพูดเชิงเวลาที่ใช้โค้ดบุ๊ค 256	61
4.9 ผลการบีบอัดเสียงพูดเชิงเวลาที่ใช้โค้ดบุ๊ค 512	62
4.10 ผลการบีบอัดเสียงพูดเชิงเวลาที่ใช้โค้ดบุ๊ค 1,024	63
4.11ผลการบีบอัดเสียงพูดเชิงเวลาตามขนาดโค้ดบุ๊ค	64
4.12 ผลทดสอบสัญญาณเสียงพูดต้นฉบับกับสัญญาณเสียงพูดที่บีบอัดเชิงความถี่	65
4.13 ค่า PSNR	68
4.14 การประเมินคุณภาพเสียงสูดจากการรับฟัง	69
5.1 เปรียบเทียบผลการบีบอัด LPC-10 แบบเดิมกับแบบใหม่	70

สารบัญรูป

รูปที่	หน้า
2.1 การวางเรียงสลับของรากของพหุนามคู่เส้นสเปกตรัม *P(z)* และ *Q(z)*	9
2.2 ความสัมพันธ์ระหว่างรากของ *A(z)* กับรากของคู่เส้นสเปกตรัม *P(z)* และ *Q(z)*	9
2.3 Multilayer Feed Forward	11
2.4 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบตาราง	12
2.5 ปมประสาทเมตริกซ์ขนาด 2x3	12
2.6 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบหกเหลี่ยม	13
2.7 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบสุ่ม	14
2.8 ปมประสาทข้างเคียง 1 มิติ และ 2 มิติ	15
2.9 ระยะห่างของปมประสาท 2 มิติ	16
2.10 แผนผังเรียนรู้การจัดตัวเอง (Self-organizing Map)	18
3.1 แผนผังขั้นตอนการดำเนินการวิจัย	22
3.2 การตั้งค่าเริ่มต้นบันทึกเสียงในโปรแกรม GoldWave	23
3.3 ตัวอย่างสัญญาณเสียงพูดในกลุ่มผู้พูดที่ใช้ในการวิเคราะห์	24
3.4 การใช้คำสั่งเปิด *.m ไฟล์	24
3.5 การใช้คำสั่งการอ่านไฟล์เสียง	25
3.6 การใช้ตัวแปรเก็บค่า LPC_coeff	25
3.7 การสร้าง Callback Function วิเคราะห์ค่า LPC_coeff	26
3.8 ค่าสัมประสิทธิ์ LPC_coeff ในไฟล์ Excel	27
3.9 ฟังก์ชั่นแปลงค่า LPC เป็น LSP ในไฟล์ main.m	28
3.10 การสร้าง Callback Function วิเคราะห์ค่า LSP_coeff	29
3.11 ค่าสัมประสิทธิ์ LSP_coeff ในไฟล์ Excel	30
3.12 ผังการทำงานของโครงข่ายการจัดการตนเอง	30
3.13 โปรแกรมในการจัดแบ่งเฟรมเสียง	31
3.13 โปรแกรมในการจัดแบ่งเฟรมเสียง (ต่อ)	32
3.14 สัญญาณเสียงสังเคราะห์จากสัมประสิทธิ์ LPC-10	32
3.15 คำสั่งในการพล็อตกราฟ	33
3.16 สัญญาณเสียงต้นฉบับกับสัญญาณเสียงสังเคราะห์ LPC-10	33
3.17 โปรแกรมการสร้างปมประสาทในชั้นลำดับจัดการตนเอง	33
3.17 โปรแกรมการสร้างปมประสาทในชั้นลำดับจัดการตนเอง (ต่อ)	34

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.18	โครงข่ายจัดการตนเองขนาด 32x4 จำนวน 128 ปมประสาท	35
3.19	การดำเนินการฝึกฝนข้อมูลโครงข่ายจัดการตนเอง	35
3.20	จุดสิ้นสุดของขบวนการฝึกฝนข้อมูลโครงข่าย	36
3.21	การจัดรูปแบบโครงข่ายแบบหกเหลี่ยม	36
3.22	การเชื่อมต่อของปมประสาทข้างเคียง	37
3.23	ระยะทางเชื่อมโยงระหว่างปมประสาท	37
3.24	น้ำหนักของปมประสาท	37
3.25	การกระจายตัวของเวกเตอร์ภายในปมประสาท	38
3.26	ตำแหน่งของค่าน้ำหนักของโครงข่ายปมประสาท	38
3.27	ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด	39
3.28	คำสั่งสร้างลำดับชั้นปมประสาทของโค้ดบุ๊ค	39
3.29	คำสั่งในกำหนดจำนวนโค้ดบุ๊คของปมประสาทค่า 128 ค่า	40
3.30	การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 128 ค่า	40
3.31	การเชื่อมต่อของปมประสาทข้างเคียงขนาด 128 ค่า	40
3.32	ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 128 ค่า	41
3.33	น้ำหนักแต่ละปมประสาทจำนวนขนาด 128 ค่า	41
3.34	การกระจายตัวของข้อมูลเวกเตอร์ขนาด 128 ค่า	41
3.35	น้ำหนักตำแหน่งของปมประสาทขนาด 128 ค่า	42
3.36	ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด	42
3.37	การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 256 ค่า	43
3.38	การเชื่อมต่อของปมประสาทข้างเคียงขนาด 256 ค่า	43
3.39	ค่าระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 256 ค่า	43
3.40	ค่าน้ำหนักแต่ละปมประสาทจำนวนขนาด 256 ค่า	44
3.41	ค่าการกระจายตัวของข้อมูลเวกเตอร์ขนาด 256 ค่า	44
3.42	ค่าน้ำหนักตำแหน่งของปมประสาทขนาด 256 ค่า	44
3.43	ค่าข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด	45
3.44	การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 512 ค่า	45
3.45	การเชื่อมต่อของปมประสาทข้างเคียงขนาด 512 ค่า	46
3.46	ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 512 ค่า	46

สารบัญรูป

รูปที่	หน้า
3.47 น้ำหนักแต่ละปมประสาทจำนวนขนาด 512 ค่า	47
3.48 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 512 ค่า	47
3.49 น้ำหนักตำแหน่งของปมประสาทขนาด 512 ค่า	48
3.50 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด	48
3.51 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 1,024 ค่า	49
3.52 การเชื่อมต่อของปมประสาทข้างเคียงขนาด 1,024 ค่า	49
3.53 ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 1,024 ค่า	50
3.54 น้ำหนักแต่ละปมประสาทจำนวนขนาด 1,024 ค่า	50
3.55 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 1,024 ค่า	51
3.56 น้ำหนักตำแหน่งของปมประสาทขนาด 1,024 ค่า	51
3.57 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด	52
3.58 คำสั่งในการแสดงผลในเชิงเวลา	53
3.59 ผลเปรียบเทียบเชิงเวลาสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วย KSOFM	53
3.60 คำสั่งผลตอบสนองความถี่ของสัญญาณเสียงต้นฉบับเปรียบเทียบสัญญาณบีบอัด	54
4.1 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 128	56
4.2 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 256	57
4.3 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 512	58
4.4 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 1,024	59
4.5 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 128	61
4.6 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 256	62
4.7 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 512	63
4.8 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 1,024	64
4.9 เปรียบเทียบผลตอบสนองเชิงความถี่ของสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัด	66
4.10 การวัดค่า PSNR ด้วย Simulink	67
4.11 โปรแกรมปรับค่าขนาดใน workspace สำหรับวัดค่า PSNR	68
5.1 เปรียบเทียบผลการบีบอัด LPC -10 แบบเดิมกับแบบใหม่	70

คำอธิบายสัญลักษณ์และคำย่อ

KSOFM	Kohonen Self Organizing Feature Maps
LPC	Linear Predictive Coder
LSF	Line Spectrum Frequencies
LSP	Line Spectrum Pairs
MOS	Mean Opinion Score
PSNR	Peak Signal to Noise Ratio
SOM	Self Organizing Maps
VQ	Vector Quantization

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระบบสื่อสาร มนุษย์ใช้เสียงพูด (Speech) เป็นสื่อกลางในการติดต่อระหว่างกัน ส่งผลให้เกิดเทคโนโลยีดิจิตอลใหม่ๆ ขึ้นเสมอ ดังนั้น เมื่อต้องการจะส่ง (Transmitted) สัญญาณเสียงพูด (Speech signal) สิ่งที่จะต้องคำนึงถึงคือความกว้าง (Bandwidth) ของช่องสัญญาณสียงพูดและการลดอัตราการส่งข้อมูล (Bit Rate) ของสัญญาณเสียงพูด ในทำนองเดียวกัน การจัดเก็บ (Stored) สัญญาณดิจิตอลต้องมีพื้นที่ว่างเพียงพอในการจัดเก็บสัญญาณเสียงพูดที่รับมาด้วยเช่นกัน ด้วยเหตุนี้จึงต้องทำการลดขนาดข้อมูลของสัญญาณเสียงพูดลง (Reduce bit rate) โดยการบีบอัดสัญญาณเสียงพูด (Speech compression) สิ่งจำเป็นที่ต้องใช้ในการบีบอัดสัญญาณเสียงพูดมักพบได้ในงานวิจัยการเข้ารหัสสัญญาณเสียงพูด (Speech coding) [1] ดังนั้น การทำให้สัญญาณเสียงมีอัตราการส่งข้อมูลต่ำ (Low bit rate) แต่ยังคงคุณภาพของเสียงพูดไว้ จึงเป็นเป้าหมายสำคัญในระบบการสื่อสารทางเสียง มาตรฐานของอัตราการเข้ารหัสสัญญาณเสียงพูด จึงเกิดขึ้นในระบบการสื่อสารหลายมาตรฐาน ทั้งการสื่อสารภายในประเทศและการสื่อสารระหว่างประเทศ

โดยทั่วไปสัญญาณเสียงพูดที่ถูกสร้างขึ้นมาใหม่ (Reconstructed) จากการเข้ารหัสสัญญาณเสียงพูดจะมีความแตกต่างจากสัญญาณเสียงพูดเดิม เนื่องจากการลดอัตราการส่งข้อมูลด้วยการตัดส่วนที่เกิดซ้ำๆ (Redundancy) ของสัญญาณเสียงพูดออกไป แต่ยังคงความถูกต้องของสัญญาณเสียง-พูดต้นฉบับหรือยังคงรักษาไว้ซึ่งคุณภาพและความถูกต้องของข้อมูล

การเข้ารหัสสัญญาณสียงพูดที่มี Bit rate ต่ำๆ นิยมใช้รูปแบบการทำนายค่าพารามิเตอร์ (Linear Predictive Parameters Model) หรือการเข้ารหัสแบบเชิงเส้น (Linear Predictive Coder : LPC) ซึ่งค่าพารามิเตอร์นี้ ได้มาจากการกรองค่าสเปกตรัมของสัญญาณ (Speech spectrum) ช่วงสั้น ในรูปแบบออลโพล (All-poll Model) [2,3] แต่อย่างไรก็ตาม พารามิเตอร์ LPC จะมีประสิทธิภาพไม่ค่อยดีนักสำหรับการควอนไตเซชั่น [4] จึงส่งค่าพารามิเตอร์ LPC ไปแปลงเป็นค่าพารามิเตอร์ของ LSP (Line Spectrum Pairs) หรือ LSF (Line Spectrum Frequencies) [5] เพื่อปรับปรุงการเข้ารหัสสัญญาณเสียงพูดให้มีประสิทธิภาพดีขึ้น

ผู้วิจัยจึงได้ศึกษาเทคนิควิธีการบีบอัดสัญญาณเสียงพูดภาษาไทย โดยพัฒนาจากการเข้ารหัสแบบ LPC-10 แปลงเป็นพารามิเตอร์ LSP ในการหาคุณลักษณะเด่นของเสียงพูด แล้วใช้โครงข่ายประสาทเทียมทำการลดค่าอัตราการส่งสัญญาณเสียงพูด ให้สัญญาณเสียงพูดที่สร้างขึ้นมีคุณลักษณะที่ใกล้เคียงกับสัญญาณเสียงต้นฉบับ

1.2 ความมุ่งหมายและวัตถุประสงค์

1.2.1 เพื่อศึกษาการบีบอัดสัญญาณเสียงพูดภาษาไทย

1.2.2 เพื่อพัฒนากรรมวิธีการเข้ารหัสเพื่อลดจำนวนบิตในการส่งสัญญาณ

1.3 ขอบเขตของการศึกษา

1.3.1 เพิ่มอัตราการบีบอัดได้ไม่น้อยกว่า 30% เมื่อเทียบกับเทคนิค LPC-10

1.3.2 เปรียบเทียบผลกระทบของการลดจำนวนบิตกับคุณภาพของเสียง

1.4 ขั้นตอนการศึกษา

1.4.1 ศึกษาหลักการและทฤษฎีที่เกี่ยวกับการเข้ารหัสและการบีบอัดสัญญาณเสียงพูด รวมทั้งค้นคว้างานวิจัยทั้งภาษาไทยและภาษาต่างประเทศ เพื่อเป็นแนวทางการทดลอง

1.4.2 ศึกษาการใช้งานโปรแกรมโกล์ดเวฟ (Goldwave v5.12) เพื่อใช้บันทึกและตัดต่อเสียง, โปรแกรมMATLAB เพื่อใช้ในการทดลองและประมวลผลค่าตัวแปรต่างๆ

1.4.3 ทำการทดลองโดยเริ่มจากใช้โปรแกรม Goldwave บันทึกเสียงพูดของกลุ่มตัวอย่าง แล้วนำไปประมวลผลหาค่าพารามิเตอร์ LPC และค่า LSP ด้วยโปรแกรม MATLAB แล้วนำค่า LSP ไปใช้ในการสร้าง Codebook ด้วยวิธีการของการจัดการตนเอง (KSOFM)

1.4.4 สรุปผลการวิเคราะห์ข้อมูล

1.4.5 สรุปผลการวิจัยและข้อเสนอแนะ

1.5 ข้อจำกัดของการศึกษา

1.5.1 สำเนียงเสียงพูดของกลุ่มตัวอย่างไม่นำมาพิจารณา

1.5.2 อายุของกลุ่มตัวอย่างไม่นำมาพิจารณา

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้เรียนรู้การบีบอัดสัญญาณเสียงพูด

1.6.2 ได้โปรแกรมการบีบอัดเสียงพูด

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทที่ 2 จะกล่าวถึงทฤษฎีและความรู้ต่างๆ ที่ได้นำมาใช้ ประกอบด้วยการวิเคราะห์สัญญาณเสียงพูดด้วยเทคนิคการทำนายพันธะเชิงเส้น (Linear Predictive) เทคนิคคู่เส้นสเปกตรัม (LSP) หรือความถี่เส้นสเปกตรัม (LSF) เทคนิคการบีบอัดข้อมูลจากการเข้ารหัสสัญญาณเสียงพูดด้วยอัลกอริทึม KSOFM (Kohonen Self Organizing Feature Maps) และงานวิจัยต่างๆ ที่เกี่ยวข้องกับวิทยานิพนธ์นี้

2.1 การวิเคราะห์สัญญาณเสียงพูด

2.1.1 การทำนายพันธะเชิงเส้น

การทำนายพันธะเชิงเส้นเป็นเทคนิคที่สำคัญทางการวิเคราะห์สัญญาณเสียงพูด เนื่องจากมีความแม่นยำสูงในการประมาณค่าพารามิเตอร์ของสัญญาณเสียงพูดเมื่อเทียบกับความเร็วในการประมวลผล หลักการพื้นฐานของการทำนายพันธะเชิงเส้นอาศัยแนวความคิดว่าตัวอย่างสัญญาณเสียงพูดสามารถประมาณค่าได้จากผลรวมของตัวอย่างสัญญาณเสียงพูดจากอดีต [6] การวิเคราะห์หาพารามิเตอร์เพื่อใช้ในการทำนายโดยทั่วไปเรียกว่าการเข้ารหัสการทำนายพันธะเชิงเส้น (LPC) ในด้านการประมวลผลสัญญาณเสียง การเข้ารหัสการทำนายพันธะเชิงเส้นนี้ถูกนำไปใช้ในสองแนวทาง [1] ได้แก่

1) การเข้ารหัสสัญญาณเสียง โดยถูกนำไปใช้เป็นวงจรกรองวิเคราะห์การทำนายพันธะเชิงเส้น (LP Analysis Filter) เพื่อแยกส่วนซ้ำซ้อนของสัญญาณเสียงออก ส่วนที่เหลือเรียกว่าสัญญาณตกค้าง (Residual Signal)

2) การสังเคราะห์สัญญาณเสียงพูด โดยถูกนำไปใช้เป็นวงจรกรองการทำนายพันธะเชิงเส้นผกผัน (Inverse LP Filter) หรือวงจรกรองสังเคราะห์การทำนายพันธะเชิงเส้น (LP Synthesis Filter) โดยที่ฟังก์ชั่นถ่ายโอนของวงจรกรองดังกล่าวแสดงกรอบสเปกตรัมของสัญญาณเสียงพูด วงจรกรองสังเคราะห์การทำนายพันธะเชิงเส้นถูกใช้แสดงแทนช่องทางเดินเสียงของมนุษย์และใช้หาสัญญาณกระตุ้นที่เหมาะสม

ในการวิเคราะห์การเข้ารหัสการทำนายพันธะเชิงเส้นเริ่มต้นจากพิจารณากรอบสัญญาณเสียงที่มีตัวอย่าง N ตัวอย่าง คือ $s_1 , s_2 , \ldots , s_N$ โดยอ้างว่าตัวอย่างสัญญาณปัจจุบันสามารถทำนายได้จากผลรวมของตัวอย่างสัญญาณในอดีต p ตัวอย่าง ดังสมการที่ 2.1

$$\tilde{s}_n = -a_1 s_{n-1} - a_2 s_{n-2} - a_3 s_{n-3} - \ldots - a_p s_{n-p} = -\sum_{k=1}^{p} a_k s_{n-k} \qquad (2.1)$$

เมื่อ $\widetilde{s}_n$ คือ สัญญาณเสียงตัวอย่างปัจจุบัน

s_n คือ สัญญาณเสียงเอาต์พุตตัวอย่างในช่วงเวลานั้นๆ

n คือ จำนวนตัวอย่างที่อยู่ในแต่ละเฟรม โดยที่ $n = 0, 1, \ldots, N-1$

p คือ อันดับของการวิเคราะห์การทำนายพันธะเชิงเส้น

$a_1, a_2, \ldots, a_p$ คือ สัมประสิทธิ์การเข้ารหัสการทำนายพันธะเชิงเส้น

กำหนดให้ e_n แทนค่าผิดพลาดระหว่างค่าจริงและค่าที่ทำนายได้ จะได้ตามสมการที่ 2.2 และสมการที่ 2.3

$$e_n = s_n - \widetilde{s}_n \tag{2.2}$$

$$e_n = s_n + \sum_{k=1}^{p} a_k s_{n-k} \tag{2.3}$$

จากสมการที่ 2.2 จะเห็นว่าค่าสัญญาณ e_n เป็นผลมาจากการลบสัญญาณ $\widetilde{s}_n$ ด้วยสัญญาณ s_n จึงอาจเรียกสัญญาณ e_n นี้ว่าเป็นสัญญาณตกค้างได้ด้วยเช่นกัน และเนื่องด้วยค่าสหสัมพันธ์ช่วงสั้น (Short-term Correlation) ระหว่างตัวอย่างของสัญญาณตกค้างมีค่าต่ำ ดังนั้นประมาณได้ว่ากรอบสเปกตรัมกำลังของสัญญาณตกค้างมีลักษณะเรียบ เมื่อทำการแปลงแซด (Z Transform) สมการที่ 2.3 จะได้ค่าดังสมการที่ 2.4

$$E(z) = A(z) \cdot S(z) \tag{2.4}$$

เมื่อ *S(z)* คือ ผลการแปลงแซดของสัญญาณเสียงพูด

E(z) คือ ผลการแปลงแซดของสัญญาณตกค้าง

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{2.5}$$

เมื่อ *A(z)* คือ วงจรกรองแยกค่าสหสัมพันธ์ช่วงสั้นที่ปรากฏในสัญญาณเสียงพูด

ผลจากสมการที่ 2.5 วงจรกรอง *A(z)* จะได้ค่าสเปกตรัมที่เรียบ เนื่องจาก *E(z)* ประมาณได้ว่ามีสเปกตรัมเรียบ ดังนั้นสามารถออกแบบกรอบสเปกตรัมของสัญญาณช่วงสั้นได้จากการวิเคราะห์การทำนายพันธะเชิงเส้นในรูปแบบออลโพล (All-poll Model) หรือรูปแบบออโตรีเกรสซีฟ (Autoregressive Model) ดังสมการที่ 2.6

$$H(z) = \frac{1}{A(z)} \tag{2.6}$$

วงจรกรอง $A(z)$ เรียกอีกชื่อหนึ่งว่าวงจรกรองผกผัน (Inverse Filter) เนื่องจากเป็นส่วนผกผันของรูปแบบออลโพล $H(z)$ ของสัญญาณเสียงพูด และรากของ $A(z)$ ทำให้เกิดโพลใน $H(z)$ นั่นคือตำแหน่งฟอร์แมนต์ของเสียงที่ได้จากช่องทางเดินเสียงที่มีฟังก์ชั่นถ่ายโอน $H(z)$

การหากรอบสเปกตรัมกำลังช่วงสั้นของเสียงพูดด้วยวิธีการวิเคราะห์การทำนายพันธะเชิงเส้น คำนวณได้จาก $H(z)$ บนวงกลมหนึ่งหน่วย (Unit Circle) โดยในขั้นแรกต้องหาสัมประสิทธิ์การทำนายพันธะเชิงเส้นของสัญญาณเสียงพูดก่อน โดยปกติหาได้จากการทำให้ค่าผิดพลาดการทำนายพันธะเชิงเส้นทั้งหมดยกกำลังสอง ดังสมการที่ 2.7 มีค่าต่ำที่สุด

$$E = \sum_{n=n_1}^{n_2} e_n^2 \tag{2.7}$$

โดยผลรวมของช่วง n_1 ถึง n_2 ที่คำนวณได้ขึ้นอยู่กับวิธีการที่ใช้ ซึ่งมีอยู่ 2 วิธี [1] ได้แก่

1) วิธีอัตสหสัมพันธ์ (Autocorrelation) ในการวิเคราะห์การทำนายพันธะเชิงเส้นช่วงสั้นสามารถหาได้โดยใช้การวิเคราะห์แบบหน้าต่างสัญญาณเสียงพูดโดยกำหนดให้ตัวอย่างสัญญาณภายนอกหน้าต่างนี้มีค่าเท่ากับศูนย์ ดังสมการที่ 2.8 จึงทำให้ได้ค่าผิดพลาดตามสมการที่ 2.7 มีค่าต่ำสุด

$$\sum_{k=1}^{p} r_{|i-k|} a_k = -r_i \quad \text{เมื่อ } 1 \le i \le p \tag{2.8}$$

เมื่อ r_k คือ ค่าสัมประสิทธิ์อัตสหสัมพันธ์อันดับที่ k ของหน้าต่างสัญญาณเสียง

หาค่า r_k ได้ดังสมการที่ 2.9

$$r_k = \frac{1}{N} \sum_{n=k}^{N} w_n s_n w_{n-k} s_{n-k} \tag{2.9}$$

เมื่อ w_n คือ ฟังก์ชั่นหน้าต่างที่มีระยะเวลา N ตัวอย่าง

การหาค่าสัมประสิทธิ์การเข้ารหัสการทำนายพันธะเชิงเส้นสามารถหาได้จากการแก้สมการที่ 2.8 ซึ่งมีจำนวน p สมการ สมการดังกล่าวเรียกว่าสมการยูล-วอล์กเกอร์ (Yule-Walker) สมการทั้งหมดสามารถเขียนในรูปของเมตริกซ์ได้ดังนี้

$$Ra = -r \tag{2.10}$$

โดยที่

$$R = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ r_2 & r_1 & r_0 & \cdots & r_{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_0 \end{bmatrix} \quad (2.11)$$

$$a = [a_1, a_2, ..., a_p]^T \quad (2.12)$$

$$r = [r_1, r_2, ..., r_p]^T \quad (2.13)$$

เมตริกซ์อัตสหสัมพันธ์ (Autocorrelation Matrix) หรือเมตริกซ์ R นี้มีลักษณะโครงสร้างที่กำหนดให้โพลของวงจรกรองสังเคราะห์การทำนายพันธะเชิงเส้น $H(z)$ อยู่ภายในวงกลมหนึ่งหน่วย (Unit Circle) โครงสร้างลักษณะเช่นนี้จะเรียกว่าเป็นโครงสร้างแบบโทพลิทซ์ (Toeplitz) ดังนั้นจึงรับรองได้ว่าวงจรกรองสังเคราะห์ $H(z)$ ที่ได้จากวิธีอัตสหสัมพันธ์นี้จะเสถียรเสมอ

สำหรับวิธีการคำนวณหาสัมประสิทธิ์การทำนายพันธะเชิงเส้นในสมการที่ 2.10 มีอยู่หลายวิธี และวิธีการหนึ่งที่นิยมใช้คือวิธีการวนซ้ำของเลวินสัน-เดอบิน (Levinson-Durbin Algorithm) ซึ่งแบ่งออกเป็น 4 ขั้นตอน ดังนี้

ขั้นที่ 1 กำหนดค่าเริ่มต้น : $E_0 = R(0)$ และ $a_0 = 0$

ขั้นที่ 2 คำนวณหาค่าสัมประสิทธิ์การสะท้อน (Reflection coefficient)

$$k_m = \frac{R(m) - \sum_{i=1}^{m-1} a_{m-1} R(m-i)}{E_{m-1}} \quad ; \; m = 1,2,3,...,p$$

เมื่อ $R(m)$ และ $R(m-i)$ คำนวณได้จากสมการ

$$R(m) = \sum_{n=m}^{N-1} x(n)x(n-m)$$

ขั้นที่ 3 คำนวณค่าสัมประสิทธิ์ของการทำนายพันธะเชิงเส้น

ให้ $a_m(m) = k_m$

และ $a_m(i) = a_{m-1}(i) - k_m a_{m-1}(m-i) \quad ; \; 1 \le i < m$

ขั้นที่ 4 คำนวณค่าผิดพลาดใหม่

$E_m = (1 - k_m^2)E_{m-1}$

$m = m + 1$

วนซ้ำขั้นที่ 2 ถึง 4 เมื่อ $m < p$

เมื่อ $m = p$ แล้ว $a_i = a_p(i)$

โดยที่ p คืออันดับของค่าสัมประสิทธิ์การทำนายพันธะเชิงเส้น

2) วิธีโคแวเรียนต์ (Covariance Method) ในการวิเคราะห์การทำนายพันธะเชิงเส้น ช่วงของการรวมอยู่ในช่วง $(p+1,\ N)$ ดังนั้นจึงไม่จำเป็นต้องใช้หน้าต่าง การทำให้ค่าผิดพลาดทั้งหมดยกกำลังสองมีค่าต่ำที่สุดหาได้จากสมการจำนวน p สมการ ดังต่อไปนี้

$$\sum_{k=1}^{p} c_{ik} a_k = -c_{i0} \quad \text{เมื่อ } 1 \le i \le p \tag{2.14}$$

โดยที่

$$c_{ik} = \sum_{n=p+1}^{N} s_{n-i} s_{n-k} \tag{2.15}$$

สมการจำนวน p สมการ สามารถเขียนในรูปของเมตริกซ์ได้ดังนี้

$$Ca = -c \tag{2.16}$$

โดยที่

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1p} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2p} \\ c_{31} & c_{32} & c_{33} & \cdots & c_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & c_{p3} & \cdots & c_{pp} \end{bmatrix} \tag{2.17}$$

$$a = [a_1, a_2, \ldots, a_p]^T \tag{2.18}$$

$$c = [c_{10}, c_{20}, \ldots, c_{p0}]^T \tag{2.19}$$

เมตริกซ์ C เรียกว่า เมตริกซ์ซ์โคแวเรียนต์ (Covariance Matrix) และเป็นเมตริกซ์สมมาตร นั่นคือ $c_{1p} = c_{p1}$ แต่เมตริกซ์ไม่ได้มีโครงสร้างเป็นแบบโทพลิทซ์ ดังนั้นการหาสัมประสิทธิ์การเข้ารหัสการทำนายพันธะเชิงเส้นด้วยวิธีนี้จึงมีประสิทธิภาพต่ำกว่าวิธีอัตสหสัมพันธ์ และสัมประสิทธิ์การเข้ารหัสการทำนายพันธะเชิงเส้นที่ได้จากวิธีนี้จึงรับรองไม่ได้ว่าจะให้วงจรกรอง

สังเคราะห์ที่มีความเสถียรเสมอ นอกจากนี้โครงสร้างที่สมมาตรทำให้การคำนวณบางส่วนสามารถใช้เทคนิคการคำนวณให้เร็วขึ้นได้ แต่ยังไม่เร็วเทียบเท่าวิธีเลวินสัน-เดอบิน

2.1.2 คู่เส้นสเปกตรัม

คู่เส้นสเปกตรัมหรือความถี่เส้นสเปกตรัม (LSF) เป็นพารามิเตอร์รูปแบบหนึ่งที่พัฒนามาจากพารามิเตอร์การทำนายพันธะเชิงเส้น เนื่องจากพารามิเตอร์การทำนายพันธะเชิงเส้นในขั้นตอนการประมาณค่าพารามิเตอร์อาจทำให้เกิดความไม่เสถียรของสัญญาณได้ ซึ่งส่งผลต่อคุณภาพของเสียง ในขณะที่พารามิเตอร์คู่เส้นสเปกตรัมมีคุณสมบัติที่เด่นคือ ค่าพารามิเตอร์อยู่ภายในขอบเขตที่จำกัด มีการเรียงลำดับของค่าพารามิเตอร์ และสามารถตรวจสอบเสถียรภาพของวงจรกรองได้ง่าย นอกจากนี้คู่เส้นสเปกตรัมยังแสดงในรูปเชิงความถี่จึงสามารถนำไปใช้ในการหาคุณสมบัติที่แน่นอนในระบบการรับรู้ของคนได้ [1]

ในการคำนวณหาคู่เส้นสเปกตรัมเริ่มต้นจากพหุนามอันดับ M ของวงจรกรองผกผันในเชิงแซดดังสมการ 2.5 โดยทำการแยกส่วนสมการดังกล่าวเป็นพหุนามอันดับ $M+1$ จำนวน 2 พหุนาม ดังสมการที่ 2.20 และสมการที่ 2.21

$$P(z) = A(z) + z^{-(M+1)} A(z^{-1}) \tag{2.20}$$

$$Q(z) = A(z) - z^{-(M+1)} A(z^{-1}) \tag{2.21}$$

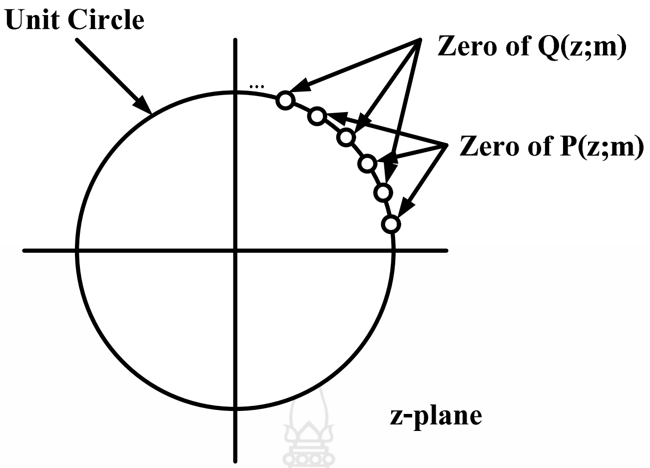
โดยพหุนาม $P(z)$ และ $Q(z)$ มีความสัมพันธ์กับ $A(z)$ ตามสมการที่ 2.22

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{2.22}$$

พหุนาม $P(z)$ และ $Q(z)$ สอดคล้องกับรูปแบบช่องทางเดินเสียงที่ไร้การสูญเสียขณะที่ช่องระหว่างเส้นเสียง (Glottis) ปิดและเปิดตามลำดับ [7] และรากของพหุนาม $P(z)$ และ $Q(z)$ เรียกว่าความถี่เส้นสเปกตรัม โดยพหุนามทั้งสองมีคุณสมบัติดังต่อไปนี้

1) ราก (Zeroes) ทั้งหมดของพหุนาม $P(z)$ และ $Q(z)$ จะตั้งอยู่บนวงกลมหนึ่งหน่วยเสมอ

2) ราก (Zeroes) ของพหุนาม $P(z)$ และ $Q(z)$ จะวางเรียงสลับกันตามลำดับจากน้อยไปหามาก ดังแสดงในรูปที่ 2.1

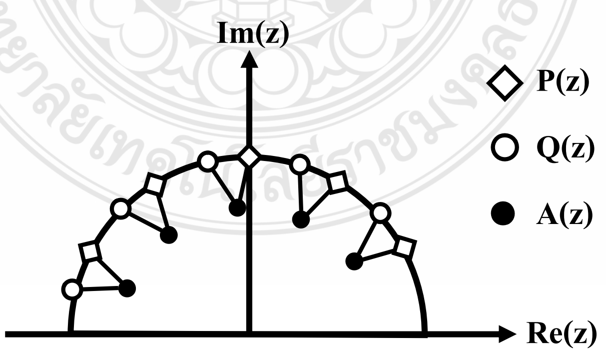


รูปที่ 2.1 การวางเรียงสลับของรากของพหุนามคู่เส้นสเปกตรัม *P(z)* และ *Q(z)* [8]

โดยสามารถแสดงให้เห็นได้ว่าวงจรกรองผกผัน *A(z)* จะมีเฟสต่ำสุด ถ้าคู่เส้นสเปกตรัมของ *A(z)* สอดคล้องกับคุณสมบัติทั้งสองนี้ ดังนั้นวงจรสังเคราะห์การเข้ารหัสการทำนายพันธะเชิงเส้นสามารถยืนยันได้ว่ามีเสถียรภาพ โดยการทำการประมวลพารามิเตอร์การเข้ารหัสการทำนายพันธะเชิงเส้นในรูปแบบคู่เส้นสเปกตรัม [1]

เมื่อพิจารณารากของพหุนามทั้งสองพบว่าพหุนาม *P(z)* และ *Q(z)* มีรากจริงอยู่ที่ -1 และ 1 ตามลำดับ สำหรับรากอื่นๆ อยู่บนวงกลมหนึ่งหน่วยโดยวางเรียงสลับกันตามคุณสมบัติของพหุนามทั้งสอง และรากทั้งสองของพหุนามมีลักษณะเป็นคู่เชิงซ้อนสังยุค ดังนั้นในการเก็บรากของพหุนามเพื่อใช้เป็นพารามิเตอร์จึงเก็บเพียง *M* ค่า

เนื่องจากรากของพหุนาม *A(z)* แสดงตำแหน่งฟอร์แมนต์ของเสียงพูด และพหุนาม *P(z)* และ *Q(z)* สัมพันธ์กับ *A(z)* ตามสมการที่ 2.20 , สมการที่ 2.21 และสมการที่ 2.22 ดังนั้นรากของพหุนามทั้งสองจึงสัมพันธ์กับฟอร์แมนต์ด้วย โดยรากของพหุนาม *A(z)* แต่ละอันจะจับคู่กับรากของพหุนาม *P(z)* และ *Q(z)* อย่างละหนึ่งราก [7] ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ความสัมพันธ์ระหว่างรากของ *A(z)* กับรากของคู่เส้นสเปกตรัม *P(z)* และ *Q(z)* [8]

จากรูปที่ 2.2 รูปวงกลมสีดำจะเป็นรากของพหุนาม *A(z)*, ส่วนรูปสี่เหลี่ยมและรูปวงกลมสีขาวเป็นรากของพหุนาม *P(z)* และ *Q(z)* ตามลำดับ

การเกาะกลุ่มของรากหรือความถี่เส้นสเปกตรัมจำนวน 2 หรือ 3 รากจะกำหนดลักษณะความถี่ฟอร์แมนต์และขนาดแบนด์วิดท์ของฟอร์แมนต์ โดยขึ้นกับความชิดของกลุ่มรากดังกล่าว [1] ถ้ารากชิดกันมากจะทำให้เกิดฟอร์แมนต์และมีแบนด์วิดท์แคบ ในทางตรงข้ามกลุ่มรากจะแสดงถึงสเปกตรัมที่มีแบนด์วิดท์กว้างคือไม่เกิดฟอร์แมนต์ [9] นอกจากนี้คุณสมบัติอีกอย่างหนึ่งของคู่เส้นสเปกตรัมคือความไวทางสเปกตรัมของคู่เส้นสเปกตรัมมีลักษณะเฉพาะที่ คือเมื่อมีการเปลี่ยนแปลงคู่เส้นสเปกตรัมใดๆ จะทำให้เกิดการเปลี่ยนแปลงของสเปกตรัมกำลังของการเข้ารหัสการทำนายพันธะเชิงเส้นเฉพาะบริเวณรอบๆ เท่านั้น ทำให้สามารถทำการประเมินค่า (Quantization) ได้อย่างอิสระโดยไม่มีผล กระทบจากการลดทอนเนื่องจากการประเมินค่าจากสเปกตรัมหนึ่งไปสู่สเปกตรัมอื่นๆ [1]

2.2 โครงข่ายประสาทเทียมอัจฉริยะ (Artificial neural networks)

โครงข่ายประสาทเทียม (Neural networks) เป็นระบบประมวลผลสัญญาณข้อมูลต่างๆ ซึ่งประกอบด้วยจำนวนหน่วยประมวลผลพื้นฐานมากมายที่เรียกว่าปมประสาท (Neurons) มีคุณลักษณะเด่นคือการกระจายข้อมูลแบบขนาน ไม่เป็นเชิงเส้น สามารถเชื่อมโยงกับโครงข่ายภายนอก จัดการตัวเองได้ และประมวลผลข้อมูลได้รวดเร็ว

รูปแบบการสังเคราะห์ปมประสาท เป็นพื้นฐานเบื้องต้นของการออกแบบโครงข่ายประสาท-เทียม โดยค่าผลลัพธ์หรือเอาต์พุต (Output : $Y_{(t)}$) นั้นจะเกิดจากผลรวมของการคูณระหว่างค่าเวกเตอร์อินพุต $x(t)$ และค่าน้ำหนัก $w(t)$

$$Y(t) = f(\sum_{i=1}^{t} x_i w_i) \qquad (2.23)$$

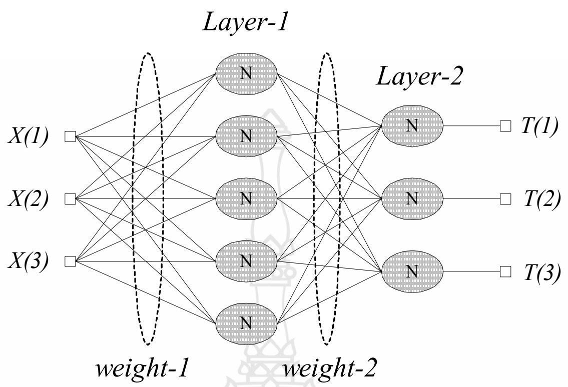
แต่ละสมาชิกของ t ซึ่งเป็นค่าของเวกเตอร์อินพุต $X_{(t)}$ ถูกถ่วงด้วยค่า $W_{(t)}$ ซึ่งเป็นค่าน้ำหนักของเวกเตอร์ W ค่าน้ำหนักอินพุตจะถูกรวมกับฟังก์ชั่นถ่ายโอนขาเข้า พบว่าค่าฟังก์ชั่นถ่ายโอนจะให้ค่าขนาดอยู่ในขอบเขตที่จำกัด ฉะนั้นค่าเอาต์พุตของปมประสาทจึงหาค่าได้ตามขอบเขตที่กำหนด

2.2.1 สถาปัตยกรรมโครงข่าย (Network architectures)

สถาปัตยกรรมของโครงข่ายประสาทเทียม เป็นการเชื่อมต่อกันระหว่างปมประสาทหนึ่งกับปมประสาทหนึ่ง เพื่อที่จะกระจายหรือส่งสัญญาณ และรวมถึงการเชื่อมโยงกับโครงข่ายอื่นๆ โดยปกติแล้วปมประสาทจะถูกจัดเป็นชั้นๆ (Layer) ปมประสาทที่อยู่ในระดับชั้นเดียวกันจะส่งข้อมูลและคุณลักษณะร่วมกัน

สถาปัตยกรรมโครงข่ายประสาทเทียมพื้นฐานประกอบด้วยชั้นของข้อมูลนำเข้า (Input layer) หนึ่งชั้นและเชื่อมต่อไปข้างหน้า (Feed forward) ยังชั้นของข้อมูลขาออก (Output layer)

มากมาย ในกรณีที่โครงข่ายประสาทเทียมที่มีความซับซ้อนจะมีชั้นการทำงานภายใน (Hidden Layer) ตั้งแต่ 1 ชั้นขึ้นไปเพื่อช่วยในการประมวลผลและเชื่อมต่อกับโครงข่ายอื่นๆ ที่อยู่สูงขึ้นไป ทำให้มีประโยชน์มากเมื่อมีจำนวนข้อมูลขาเข้า (Input layer) หลายๆ ชั้น



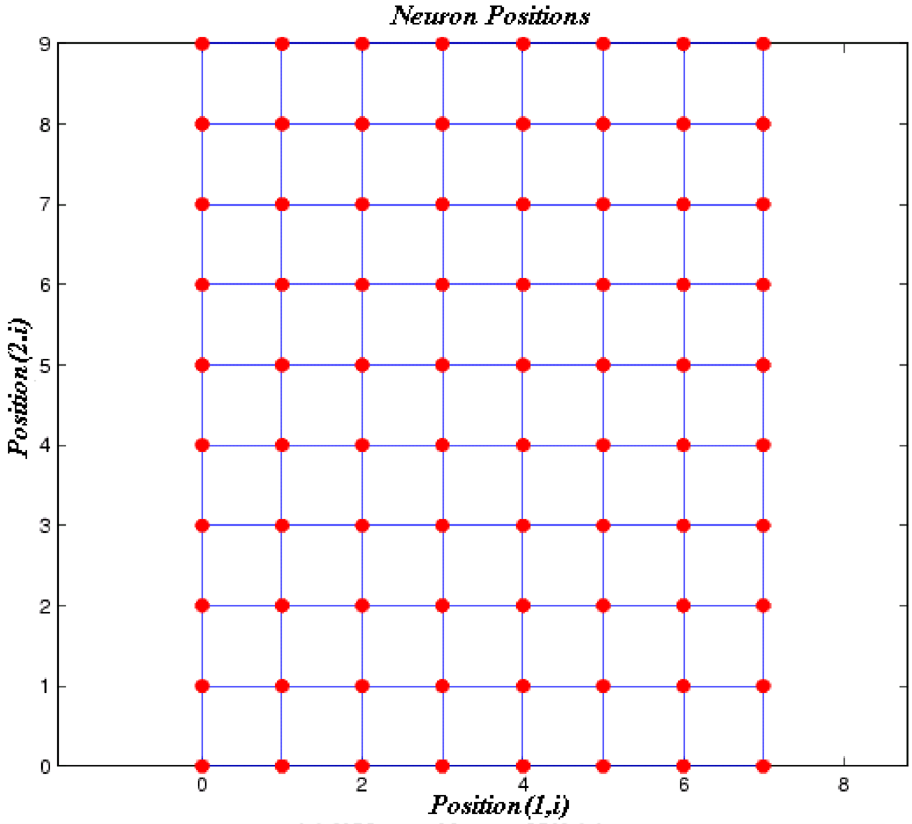
รูปที่ 2.3 Multilayer Feed Forward [10]

โดยปกติแล้วโครงข่ายประสาทเทียมจะมีจำนวนชั้นข้อมูลมากมายหลายชั้น เมื่อนำชั้นมารวมกันก็จะทำให้ได้จำนวนปมประสาทที่แตกต่างกันมาก ในรูปที่ 2.3 แสดงให้เห็นถึงสถาปัตยกรรมโครงข่ายประสาทเทียม 2 ชั้น ซึ่งแต่ละองค์ประกอบมี input vector $x(t)$ เป็นองค์ประกอบ ค่า $t = 1,2,3$ ถูกเชื่อมต่อกับ layer 1 และ output ของ layer 1 จะกลายเป็น input ของ layer 2 ที่เรียกว่า Hidden layers ส่วนชั้นที่อยู่ในตำแหน่งสุดท้ายก็คือ Output Layer ดังนั้นโครงข่ายประสาทเทียมในรูปที่ 2.3 จึงมี 3 เลเยอร์

2.2.2 ลักษณะฟังก์ชั่นโครงสร้าง (Topology function)

รูปแบบการเชื่อมต่อเซลล์ประสาทหรือการจัดเรียงของเซลล์ประสาทของการจัดการตนเอง (SOM) ถูกแบ่งตามลักษณะฟังก์ชั่นโครงสร้างได้ 3 ลักษณะ ได้แก่ ฟังก์ชั่นโครงสร้างในแบบตาราง (Grid topology function) ฟังก์ชั่นโครงสร้างในแบบหกเหลี่ยม (Hexagonal topology function) และฟังก์ชั่นโครงสร้างในแบบสุ่ม (Random topology function) [11]

1) ฟังก์ชั่นโครงสร้างแบบตาราง (Grid topology Function)



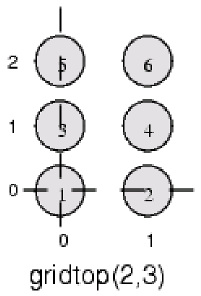
รูปที่ 2.4 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบตาราง

ฟังก์ชั่นโครงสร้างลักษณะตารางจะใช้ปมประสาทที่มีการจัดวางในลักษณะตารางสี่เหลี่ยม เช่น ปมประสาทแบบชุดเมตริกซ์ขนาด 2x3 มีจำนวนสมาชิก 6 ค่า เขียนด้วยคำสั่ง Matlab ตามได้ดังนี้

<u>คำสั่ง</u> pos = gridtop (2,3)

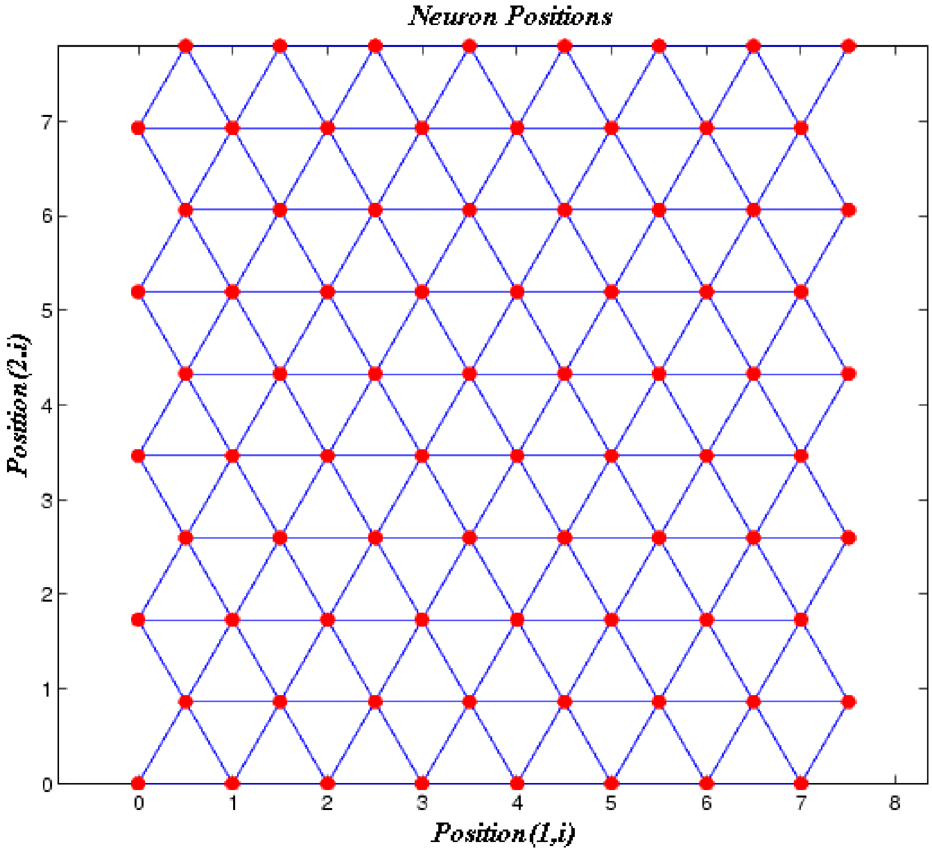
<u>แสดงผล</u>	pos =	0	1	0	1	0	1
		0	0	1	1	2	2

เช่น ปมประสาทที่ 1 อยู่ตำแหน่ง (0,0) ปมประสาทที่ 2 อยู่ตำแหน่ง (1,0)และปมประสาทที่ 3 อยู่ที่ตำแหน่ง (0,1) แสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 ปมประสาทเมตริกซ์ขนาด 2x3

2) ฟังก์ชั่นโครงสร้างลักษณะหกเหลี่ยม (Hexagonal topology function)

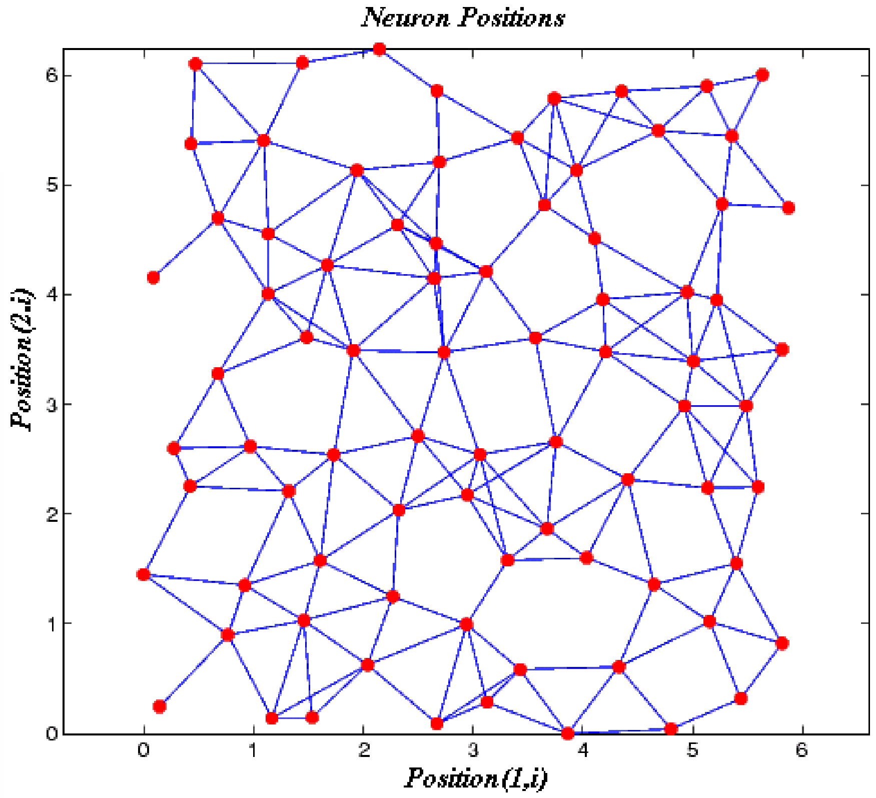


รูปที่ 2.6 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบหกเหลี่ยม

ฟังก์ชั่นโครงสร้างลักษณะหกเหลี่ยมจะใช้จัดการปมประสาทในลักษณะหกเหลี่ยม เช่น การเขียนคำสั่ง hextop เพื่อแสดงสมาชิกของปมประสาทขนาด 2x3 แบบหกเหลี่ยม

<u>คำสั่ง</u>	pos = hextop(2,3)						
<u>แสดงผล</u>	pos =	0	1.0000	0.5000	1.5000	0	1.0000
		0	0	0.8660	0.8660	1.7321	1.7321

3) ฟังก์ชั่นโครงสร้างลักษณะสุ่ม (Random topology function)



รูปที่ 2.7 การเชื่อมต่อเซลล์ประสาทตามฟังก์ชั่นโครงสร้างแบบสุ่ม

ฟังก์ชั่นโครงสร้างลักษณะสุ่มจะใช้จัดวางปมประสาทในลักษณะสุ่มค่าโดยการใช้คำสั่ง randtop เช่น การเขียนคำสั่ง randtop เพื่อแสดงสมมาชิกของปมประสาทขนาด 2x3 แบบสุ่ม

คำสั่ง	pos =	randtop (2,3)					
แสดงผล	pos =	0	0.7620	0.6268	1.4218	0.0663	0.7862
		0.0925	0	0.4984	0.6007	1.1222	1.4228

2.2.3 การหาค่าระยะห่างของปมประสาท

ในโครงข่ายประสาทเทียมจะประกอบไปด้วยปมประสาท 2 ลักษณะ ได้แก่

1) ปมประสาทจุดกำเนิด (Home neuron) คือ ปมประสาทที่ใช้เป็นจุดเริ่มต้นหรือจุดศูนย์กลางของการจัดกลุ่มการจัดการตนเอง

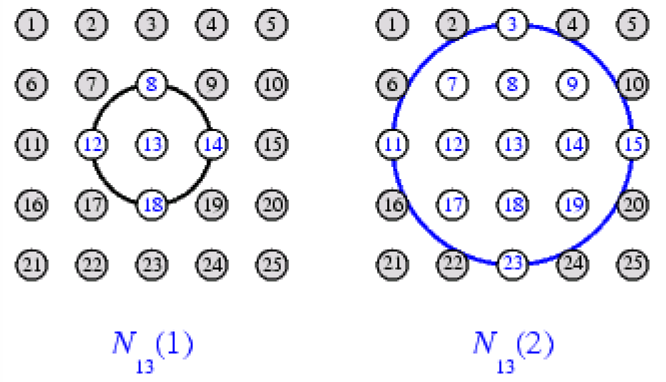
2) ปมประสาทข้างเคียง (Neighborhood neuron) คือ ปมประสาทที่อยู่รอบๆ ปมประสาทจุดกำเนิด

แนวคิดปมประสาทจุดกำเนิดและปมประสาทข้างเคียงแสดงได้ดังรูปที่ 2.8 ก) และ ข) กำหนดให้ N_{13} เป็นปมประสาทจุดกำเนิด จะเห็นว่าปมประสาทข้างเคียง 1 มิติ ได้แก่ ปมประสาทที่ 8,

12, 14, 18 ปมประสาทข้างเคียง 2 มิติของ N_{13} ได้แก่ ปมประสาทที่ 3, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 23 เขียนในรูปแบบของเซตได้ ดังนี้

N_{13} (1) = {8, 12, 14, 18}

N_{13} (2) = {3, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 23}



ก) ข)

รูปที่ 2.8 ก) ปมประสาทข้างเคียง 1 มิติ

ข) ปมประสาทข้างเคียง 2 มิติ

การหาค่าระยะห่างของปมประสาทจะมีวิธีการคำนวณค่าระยะห่างอยู่ 2 วิธี ได้แก่ การหาระยะห่างด้วยวิธี Euclidean และการหาระยะห่างด้วยวิธี Manhattan ซึ่งมีวิธีการดังต่อไปนี้

1) การหาระยะห่างด้วยวิธี Euclidean จะใช้ฟังก์ชั่น dist ในMatlab ในการหาค่าปมประสาทจากตำแหน่งจุดกำเนิด (Home neuron) ไปยังจุดข้างเคียง (Neighborhood) เช่น กำหนดให้ปมประสาทมี 3 ปม เขียนเป็นคำสั่งได้ดังนี้

<u>คำสั่ง</u> Pos2 = [0 1 2 ; 0 1 2]

<u>แสดงผล</u> Pos2 =

```
0    1    2
0    1    2
```

จากนั้นหาระยะห่างจากปมประสาทแต่ละค่าจากความสัมพันธ์ตามคำสั่งต่อไปนี้

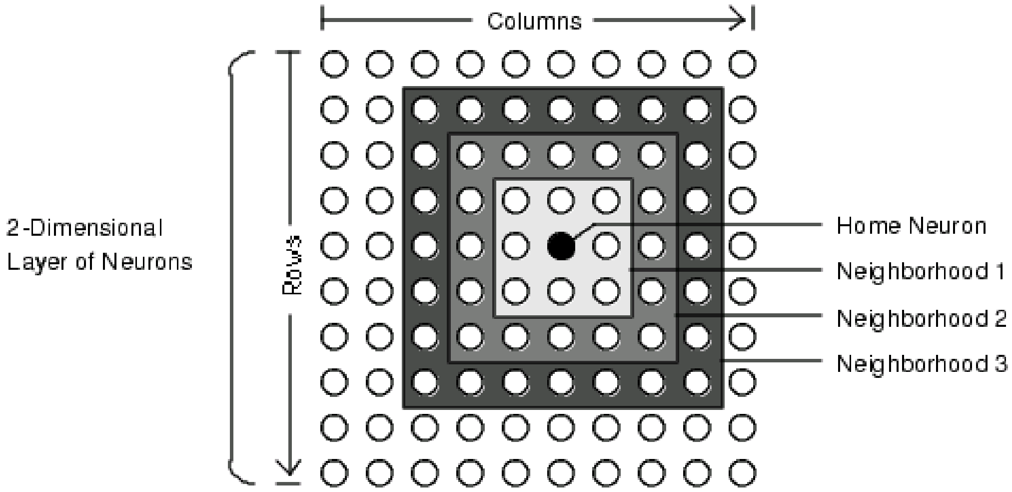
<u>คำสั่ง</u> D2 = dist (pos2)

<u>แสดงผล</u> D2 =

```
0         1.4142    2.8284
1.4142    0         1.4142
2.8284    1.4142    0
```

จะเห็นว่าระยะห่างจากปมประสาทที่ 1 ซึ่งเป็นของตนเองมีค่าเป็น 0 และระยะห่างจากปมประสาทที่ 1 ถึงปมประสาทที่ 2 มีค่าเป็น 1.414 ค่าระยะห่างระหว่างปมประสาทนี้เรียกว่า Euclidean distance

การแสดงค่าระยะห่างของปมประสาทใน 2 มิติ จะวัดระยะจากปมประสาทจุดกำเนิดไปยังจุดปมประสาทที่ต้องการ โดยวิธีเพิ่มค่าระยะทางไปยังปมประสาทข้างงเคียงโดยรอบ



รูปที่ 2.9 ระยะห่างของปมประสาท 2 มิติ

จะเห็นว่าการใช้คำสั่ง dist จะใช้หาค่าระยะทางปมประสาท 1 มิติ แต่ในปมประสาท 2 มิติ จะใช้คำสั่ง boxdist เนื่องจากค่าระยะทางมีค่าสัมพันธ์กับปมประสาทข้างเคียงในลักษณะเมตริกซ์ เช่น กำหนดให้ปมประสาท 2 มิติ และใช้โครงสร้างแบบตารางขนาด 2x3 จะใช้คำสั่งดังนี้

<u>คำสั่ง</u> Pos = gridtop (2,3)

d = boxdist (pos)

<u>แสดงผล</u>

pos =	0	1	0	1	0	1
	0	0	1	1	2	2
d =	0	1	1	1	2	2
	1	0	1	1	2	2
	1	1	0	1	1	1
	1	1	1	0	1	1
	2	2	1	1	0	1
	2	2	1	1	1	0

จากผลการทำงานของโปรแกรมจะพบว่าระยะทางจากปมประสาทที่ 1 ไป 2, 3 และ 4 มีค่าเท่ากับ 1 และระยะทางจากปมประสาทที่ 1 ไป 5 และ 6 มีค่าเท่ากับ 2 ส่วนระยะทางจากปมประสาท 3 และ 4 ไปยังปมประสาทอื่นๆ มีค่าเท่ากับ 1

ในการหาค่าระยะทางเชื่อมโยง (link distance) จากปมประสาทหนึ่งๆ ไปยังปมประสาทอื่น จะต้องใช้วิธีการคำนวณค่าจากชุดของปมประสาทด้วยคำสั่ง linkdist เช่น จงหาค่าของระยะทางเชื่อมโยงของปมประสาท 2 มิติ ขนาด 2x3

<u>คำสั่ง</u> pos = gridtop (2,3)

dlink = linkdist(pos)

<u>แสดงผล</u> dlink =

0	1	1	2	2	3
1	0	2	1	3	2
1	2	0	1	1	2
2	1	1	0	2	1
2	3	1	2	0	1
3	2	2	1	1	0

2) การหาค่าระยะห่างด้วยวิธี Manhattan เป็นการคำนวณค่าระยะทางระหว่างเวคเตอร์ x และ y ซึ่งสามารถหาได้จากความสัมพันธ์ดังต่อไปนี้

D = sum(abs(x-y))

ถ้ากำหนดให้ W1 = [1 2; 3 4; 5 6] และ P1 = [1;1] ค่าระยะทาง Manhattan ที่ได้จะมีค่าดังต่อไปนี้

<u>คำสั่ง</u> W1 = [1 2; 3 4; 5 6]

P1 = [1;1]

Z1 = mandist(W1,P1)

<u>แสดงผล</u> W1 =

1	2
3	4
5	6

P1 =

1

1

Z1 =

1

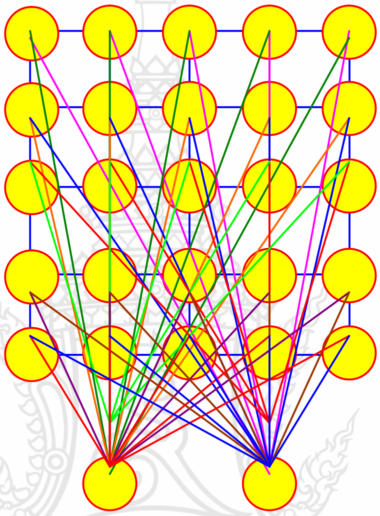
5

2.2.4 โครงข่ายประสาทเทียมแบบจัดการตนเองของโคโฮเนน

KSOFM (Kohonen Self Organizing Feature Maps) เป็นอัลกอริทึ่มหนึ่งของโครงข่ายประสาทเทียม (Neuron network) ซึ่งมีการเรียนรู้ที่สามารถจัดการตนเองโดยจะทำการประมวลผลจัดจำแนก input codevectors เป็นกลุ่มๆ หรือการทำซ้ำข้อมูลเพื่อหาค่าของน้ำหนักของข้อมูลที่มีอยู่ทั้งหมดตามจำนวนกลุ่มที่ต้องการ เมื่อมี input codevector ชุดใหม่เข้ามา ระบบก็จะประมวลผลค่าน้ำหนักของ input codevector ใหม่ ปมประสาทที่มีค่าใกล้เคียงกับ input codevector มากที่สุดจะเรียกว่า Winning neuron โดย Winning neuron นี้จะถูกปรับค่าหรือปรับแต่งให้มีการตอบสนองต่อ

โครงข่ายมากที่สุด และจะส่งผลให้ปมประสาทข้างเคียงหรือ codevector ใกล้เคียงถูกปรับค่าเช่นกัน ซึ่งคุณลักษณะสมบัติการปรับแต่งค่าเหล่านี้ของ KSOFM จะเป็นประโยชน์อย่างมากต่อการประมวลผลข้อูล

โครงสร้างการทำงานของ KSOFM ชั้นแรกทำหน้าที่นำเข้าข้อมูลและจัดส่งข้อมูลให้แก่ neurons ชั้นที่สอง ทุกๆ ปมประสาทระหว่างชั้นจะเชื่อมต่อกันด้วยค่าน้ำหนัก (Weight Vector) ดังนั้นข้อมูลเข้าในชั้นแรกจะถูกส่งไปยัง neurons ในชั้นที่สองเพื่อทำการเปรียบเทียบว่าใกล้เคียงกับค่ากลางกลุ่มใดมากที่สุด แต่ละปมประสาทในชั้นนี้จะมีความสัมพันธ์กันแบบ neighborhood relation การก่อรูปแบบ 2 มิติ จึงเกิดขึ้นดังรูปที่ 2.10 [12]



รูปที่ 2.10 แผนผังเรียนรู้การจัดตัวเอง (Self-organizing Map)

ขั้นตอนการทำงานของ KSOFM

1) กำหนดชุดของข้อมูลที่จะเป็นอินพุต
2) สุ่มค่าเริ่มต้นให้กับค่ากลางของกลุ่ม (cluster center) และกำหนดอัตราการเรียนรู้
3) วัดค่าความคล้ายด้วยวิธี Euclidean โดยการทำซ้ำทุกข้อมูลนำเข้า $1 \leq i \leq N$ ให้หาผู้ชนะจากการเลือกกลุ่มที่มีค่ายูคลิเดียนน้อยที่สุด ค่าความต่างระหว่างข้อมูลนำเข้ากับเวกเตอร์น้ำหนัก
4) คำนวณค่ากลางสำหรับกลุ่มที่เป็น the winner ใหม่
5) ทำซ้ำขั้นตอนที่ 3-4
6) หาระยะห่างระหว่างโหนดนั้นๆ กับ โหนดที่เป็น the winner
7) หาโหนดเพื่อนบ้านที่ใกล้เคียงที่สุด โดยขอบเขตจะลดลงตามเวลา
8) ปรับค่าน้ำหนักของแต่ละโหนด

2.3 การประเมินคุณภาพของข้อมูลเสียงพูด

การประเมินคุณภาพของข้อมูลเสียงพูดโดยใช้วิธีการของ MOS (Mean Opinion Score) เป็นวิธีการหนึ่งที่นิยมใช้เปรียบเทียบระหว่างเสียงพูดต้นแบบกับเสียงพูดที่ผ่านการบีบอัดข้อมูล โดยใช้การรับรู้และความรู้สึกของมนุษย์เป็นเกณฑ์ในการตัดสิน (Subjective measurement)

ตารางที่ 2.1 ค่า MOS ที่เหมาะสมกับการใช้งานในระบบต่าง ๆ

MOS	การใช้งาน
4.5-5.0	Broadcast Quality
4.0-4.5	Network or Toll Quality
3.5-4.0	Communication Quality
2.5-3.5	Synthetic Quality

วิธีการประเมินหรือวัดคุณภาพสียงนั้นจะใช้คนประมาณ 12-24 คน ทดสอบคุณภาพเสียงด้วยการฟัง โดยที่แต่ละคนจะให้คะแนนที่มีค่าอยู่ระหว่าง 1-5 ตามคุณภาพของสัญญาณที่ตัวเองรู้สึก จากนั้นหาค่าเฉลี่ยแต่ละเสียงพูดว่าอยู่ในระดับใด

ตารางที่ 2.2 รายละเอียดวิธีการให้คะแนนในการวัด MOS

คะแนน	คุณภาพของเสียง
5	ดีมาก (คุณภาพเสียงชัดเจนและเข้าใจง่าย)
4	ดี (คุณภาพเสียงดีและเข้าใจง่าย แต่อาจได้ยินเสียงรบกวนบ้าง)
3	พอใช้ (คุณภาพเสียงดีและเข้าใจ ได้แต่อาจต้องการอาศัยความตั้งใจ หรือบางทีต้องขอให้พูดซ้ำ)
2	เลว (คุณภาพเสียงดีและเข้าใจได้ก็ต่อเมื่อมีความตั้งใจมากๆ และบ่อยครั้งที่ต้องขอให้พูดซ้ำ)
1	เลวมาก (ฟังไม่รู้เรื่องเลย)

2.4 งานวิจัยที่เกี่ยวข้อง

งานวิจัยทางด้านการประมวลผลสัญญาณเสียงพูดได้มีการพัฒนามาอย่างหลากหลายในหลายแนวทาง และมีความแตกต่างกันออกไปตามวัตถุประสงค์ของการวิจัยนั้นๆ ในการวิจัยครั้งนี้จึงได้คัดเลือกงานวิจัยที่เกี่ยวข้องและมีความใกล้เคียงกับงานวิจัยที่กำลังศึกษา ดังตัวอย่างต่อไปนี้

1) ในงานวิจัยของ J. Srinonchat [3] ได้ศึกษาเกี่ยวกับโครงสร้างการบีบอัดข้อมูลให้มีค่า bit rate ต่ำลงด้วยการใช้ LPC-10 แบบใหม่ โดยพบว่าค่าพารามิเตอร์ LPC ที่ใช้ทั่วไปจะให้ค่าการ ควอนไตซ์ไม่ค่อยมีประสิทธิภาพ เนื่องจากจะเกิดค่าผิดพลาดขึ้นในกรณีที่ระดับความแตกต่างของสัญญาณเสียงพูดมีระดับต่ำ เพื่อแก้ปัญหาเรื่องความผิดพลาดของการควอนไตซ์ จึงได้นำเสนอวิธีการ โดยเปลี่ยนค่าพารามิเตอร์ LPC เป็นพารามิเตอร์ LSP เพื่อจะสร้าง codevector ของการแบ่งลำดับชั้นใน vector quantization เพื่อที่จะใช้เข้ารหัสสัญญาณเสียงพูดใหม่ เรียกว่า LPC-10 จากผลทดลองพบว่าสามารถลดจำนวนบิตของสัญญาณเสียงพูดในบิตสัญญาณเสียงพูด p1-p4 ได้ 4 บิต และอัตราการส่งข้อมูลลดลง 2-66%

2) ในงานวิจัยของ J. Srinonchat and other. [4] ได้ศึกษาเกี่ยวกับการใช้ตำแหน่งเวกเตอร์ ควอนไตซ์ (VQ) ในการบีบอัดสัญญาณเสียงพูด โดยการแบ่งเสียงพูดออกเป็นเฟรม แต่ละเฟรมมีค่า 30 ms. แล้วนำมาสกัดค่าเป็นสัมประสิทธิ์ LPC และ LSP เพื่อใช้เป็นพารามิเตอร์ในการสร้าง codebook ด้วย KSOFM ซึ่งทำหน้าที่คำนวณค่าและแทนค่าสัมประสิทธ์สัญญาณเสียงพูดลงใน codebook ให้มีความผิดเพี้ยนน้อยสุด ผลจากการทดลองเสียงพูดที่เป็นชายจำนวน 2 คนและหญิง 2 คน พบว่า ค่าผิดพลาดสูงสุดของการบีบอัดค่า LSP ในเสียงพูดชาย 35% และในเสียงพูดหญิง 40% ส่วนค่าผิดพลาดสูงสุดของการบีบอัดค่า LPC ในเสียงพูดชาย 35% และในเสียงพูดหญิง 30% สรุปแล้วการบีบอัดสัญญาณเสียงพูดด้วยเทคนิค LPC-VQ จะใช้ KSOFM และการใช้ Address-codebook จะลดค่า bit rate ได้ 33%

3) ในงานวิจัยของ จักรกฤษ อ่อนชื่นจิตร [7] การนำเสนอหลักการวิเคราะห์แนวทางเดินเสียงในรูปแบบของสัมประสิทธิ์คู่เส้นสเปกตรัม (LSP) ร่วมกับเทคนิคของ Double Clustering เพื่อนำประโยชน์ไปใช้สำหรับการประมวลผลสัญญาณเสียงพูดที่เป็นภาษาไทยคำโดดคือ "ศูนย์", "หนึ่ง", "สอง", "สาม", "สี่", "ห้า", "หก", "เจ็ด", "แปด" และ "เก้า" โดยใช้ข้อมูลเสียงพูดจากผู้ชาย 100 คน และผู้หญิง 100 คน พูดคนละ 3 รอบ จะได้ข้อมูลเสียงพูดละ 300 เสียง นำมาผ่านกระบวนการหาค่าลักษณะสำคัญด้วยสัมประสิทธิ์คู่เส้นสเปกตรัม ที่ 10 อันดับ โดยแบ่งแต่ละสัญญาณเสียงพูดออกเป็น 20 เฟรมเท่ากัน จากนั้นหาตัวแทนของเสียงพูดด้วยวิธีการ Double Clustering แบบอัลกอริทึ่ม K-Means ที่ผ่านการหาค่าลักษณะสำคัญด้วยสัมประสิทธิ์คู่เส้นสเปกตรัมแล้วเปรียบเทียบระยะห่างจากจากตัวแทนสัมประสิทธิ์ของแต่ละเสียงพูด จากผลการทดลองที่ได้จะเห็นว่าเทคนิคที่นำมาใช้นี้สามารถนำไปใช้งานได้ดี มีความแม่นยำค่อนข้างสูง เฉลี่ยประมาณ 92.85% โดยมีค่าความแม่นยำสูงสุดอยู่ที่ 100% ของเสียงพูด "สอง" จากข้อมูลเสียงพูดผู้หญิง และมีค่าความแม่นยำต่ำสุดอยู่ที่ 79% ของเสียงพูด "ห้า" จากข้อมูลเสียงพูดผู้หญิง

4) ในงานวิจัยของ F.K. Soong and B-H. Juang [13] ได้ศึกษาเกี่ยวกับ LSP และการบีบอัดข้อมูลเสียงพูด ซึ่งได้แสดงถึงค่าคุณสมบัติของฟังก์ชั่นเฟสและคุณลักษณะของความถี่ LSP ที่ใช้ในการวิเคราะห์ข้อมูลเสียงพูด การทดลองนี้ใช้ข้อมูลสัญญาณเสียงพูดชายและหญิง จำนวน 37,000 เฟรม แบ่ง 100 เฟรม/วินาที คาบเวลา 20 ms. โดยแบ่ง LPC ออกเป็น 10 ระดับ และใช้ LSP ที่มีค่าความ

แตกต่างของความถี่ 10 ค่า ผลการทดลองพบว่ามีการกระจายตัวของความถี่ LSP ตลอดตามแกนความถี่ ฉะนั้น การที่จะทำให้ bit rate ลดลงจึงต้องทำให้ความแตกต่างของค่า LSP ลดลง ในงานวิจัยนี้สามารถบีบอัดสัญญาณเสียงพูดที่มีค่า bit rate 9.6 kb/s ให้ลดลงได้ด้วยวิธี Multi–pulse LPC เหลือค่า bit rate เพียง 2 kb/s และเมื่อใช้วิธีทางความถี่ด้วย LSP จะมีค่า bit rate เพียง 2.5 kb/s.

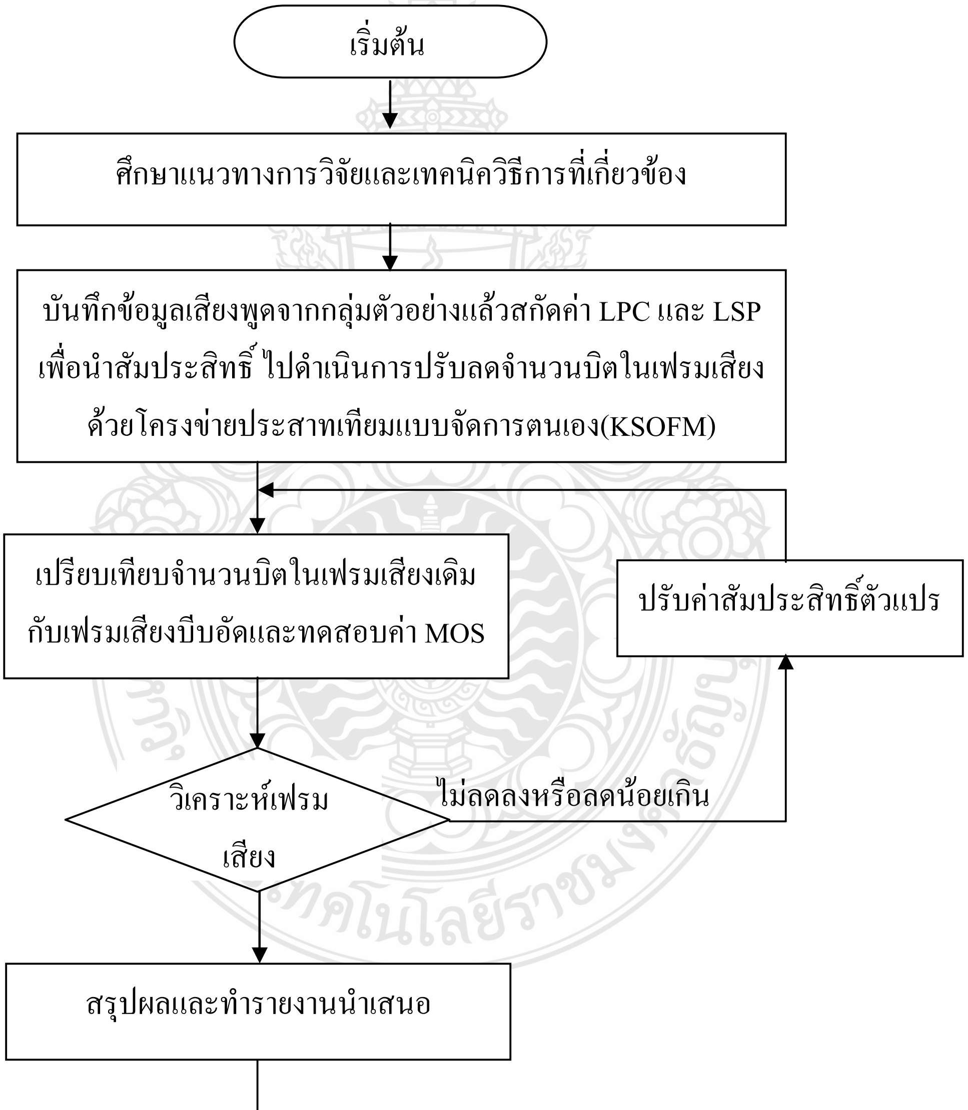
5) ในงานวิจัยของ Won Cheol Lee และ Myung Jin Bae [14] ได้ศึกษาเกี่ยวกับการแสดงค่า LSP แบบใหม่ เพื่อหาค่ากรอบคลื่นของการควอนไตซ์ ในรายงานฉบับนี้ได้เสนอตัวแปรทางสถิติใหม่ของการคำนวณค่า LSP ซึ่งจะให้ประสิทธิภาพการควอนไตซ์ที่ดีกว่า LPC โดยวิธีการทั่วไปแล้วความชัดเจนในการคำนวนค่า LSP จะใช้สำหรับข่าวสารสัญญาณเสียงพูด ในรายงานฉบับนี้เป็นการแสดงค่า LSP ใหม่ที่ชัดเจนที่เป็นสเปกตรัมบริเวณกรอบคลื่น ซึ่งเป็นการลดค่าจำนวนบิตลงโดยไม่เกิดการผิดเพี้ยน ผลของการจำลองชี้ให้เห็นว่าสมการที่ใช้ สามารถลดจำนวน bit rate ให้ผลดีขึ้นด้วยการใช้วิธีของ autocorrelation ลดค่าสัมประสิทธิ์ของ LSP ทำให้พารามิเตอร์ที่จะมีค่า pole และ zero รอบๆ วงกลมหน่วยลดลง

6) ในงานวิจัยของ สุทธี ทับทองดี [15] ได้วิจัยและคิดค้นเทคนิคใหม่เพื่อที่จะนำมาบีบอัดสัญญาณเสียงพูดภาษาไทยบนพื้นฐานของการเข้ารหัสด้วยวิธี LPC-10 (Linear Predictive Coding Order 10) โดยใช้เสียงภาษาไทยมาจากบุคคล 4 คนโดยเป็นผู้หญิง 2 คน และเป็นผู้ชาย 2 คน พูดคนละ 30 นาทีบันทึกลงในคอมพิวเตอร์ สัญญาณเสียงถูกแบ่งออกเป็นเฟรมโดยความยาวของเฟรมที่ใช้อยู่ที่ 240 ตัวอย่าง (Samples) และใช้โปรแกรม MATLAB ออกแบบโมดูลของต้นฉบับ LPC-10 และ K-means จากการทดลองพบว่าเมื่อเปรียบเทียบวิธีบีบอัดสัญญาณเสียงพูดต้นฉบับ LPC-10 และวิธีใหม่ของ LPC-10 จะพบว่าสัญญาณเสียงสังเคราะห์ของวิธีใหม่ LPC-10 สามารถที่จะสร้างเสียงสังเคราะห์ได้ใกล้เคียงกับทางวิธีการต้นฉบับ LPC-10 ซึ่งใช้จำนวนบิตน้อยกว่าวิธีการต้นฉบับ LPC-10

7) ในงานวิจัยของ สุวรรณ รื่นสุคนธ์และจักรี ศรีนนท์ฉัตร [16] ได้นำเสนอระบบการรู้จำเสียงพูดโดยใช้สัมประสิทธิ์คู่เส้นสเปกตรัมร่วมกับแบบจำลองโครงข่ายประสาทเทียมเพื่อจำแนกเสียงพูดตัวเลขภาษาไทยตั้งแต่ “ศูนย์ ถึง เก้า” โดยใช้การวิเคราะห์แนวทางเดินของเสียงแบบสัมประสิทธิ์คู่เส้นสเปกตรัมในการดึงค่าคุณลักษณะเด่นของเสียงพูด และใช้แบบจำลองโครงข่ายประสาทเทียมในการรู้จำ ผลที่ได้พบว่าความแม่นยำของการรู้จำแบบโครงข่ายประสาทเทียมอยู่ที่ 84.20% ของสัญญาณเสียงผู้หญิง ในส่วนของผู้ชายอยู่ที่ 87.40% ของสัญญาณเสียงตัวเลขทั้งหมด ซึ่งให้ผลที่ดีกว่าการใช้สัมประสิทธิ์คู่เส้นสเปกตรัมในการหาระยะห่างของข้อมูลสัญญาณเสียงพูดแต่ละสัญญาณเสียง

บทที่ 3
วิธีการดำเนินงานวิจัย

ในส่วนนี้ได้ศึกษาและนำเสนอเกี่ยวกับเทคนิควิธีการบีบอัดสัญญาณเสียงพูดขั้นสูง โดยมีพื้นฐานมาจาก LPC-10 ซึ่งจะใช้หลักของ LPC และ LSP กำหนดคุณลักษณะของการควอนไทซ์เซชั่นและการเข้ารหัสสัญญาณเสียงพูดในแต่ละเฟรมจะถูกทำให้เกิดค่าสัมประสิทธิ์ของ LSP และการลดค่าอัตราการส่งข้อมูล (Bit Rate) โดยใช้โครงข่ายประสาทเทียมแบบจัดการตนเอง (KSOFM) จากนั้นทำการวิเคราะห์ผลของเฟรมเสียงที่ได้ ซึ่งมีขั้นตอนในการดำเนินงานวิจัยดังรูปที่ 3.1



รูปที่ 3.1 แผนผังขั้นตอนการดำเนินการวิจัย

3.1 การเก็บบันทึกเสียงพูด

การบันทึกสัญญาณเสียงพูดใช้การบันทึกเสียงผ่านไมโครโฟนชนิดคอนเดนเซอร์ลงในคอมพิวเตอร์ส่วนบุคคลในห้องทำงานที่มีสภาพแวดล้อมปกติ (สัญญาณรบกวนโดยรวมไม่เกิน 0 - 0.5 dB) โดยบันทึกเสียงแบบดิจิตอลในระบบโมโน กำหนดให้มีอัตราการสุ่มตัวอย่าง 8,000 Hz/s และมีการแบ่งระดับที่ 8 บิต บันทึกข้อมูลอยู่ในรูปของไฟล์ *.wav โดยมีขั้นตอนการบันทึกเสียงพูดดังนี้

- เตรียมบทความที่ให้ผู้พูดอ่าน โดยใช้เวลาประมาณ 10 นาที
- เตรียมอาสาสมัครผู้พูด 20 คน แบ่งเป็นชาย 10 คน หญิง 10 คน
- ให้ผู้พูดทดลองอ่านบทความเพื่อปรับระดับสัญญาณเสียงอยู่ในช่วง 0.7-10 โวลท์
- ให้ผู้พูดแต่ละคนอ่านบทความที่เตรียมไว้ คนละ 10 นาที บันทึกลงในคอมพิวเตอร์

3.1.1 อุปกรณ์ที่ใช้ในการจัดเก็บข้อมูลเสียง

อุปกรณ์ที่ใช้ในการจัดเก็บข้อมูลเสียงนั้น จะเป็นอุปกรณ์ที่ได้มาตรฐานสามารถนำมาทำการบันทึกใช้เป็นข้อมูลจริงได้ และทำการทดสอบได้จริง และอุปกรณ์ทั้งหมดอยู่ในสภาพที่สมบูรณ์สามารถใช้งานได้ ซึ่งจะมีรายละเอียดของอุปกรณ์ต่างๆ ดังนี้

1) เครื่องคอมพิวเตอร์ Notebooks รุ่น Intel(R) Pentium(R) core 2 Dual ความเร็วในการประมวลผล 2 GHz ที่มีหน่วยความจำหลัก 2 GB และหน่วยความจำสำรอง 250 GB

2) การ์ดเสียง (Sound Card) SoundMAX ติดตั้งมาพร้อมกับเครื่องคอมพิวเตอร์ Notebooks

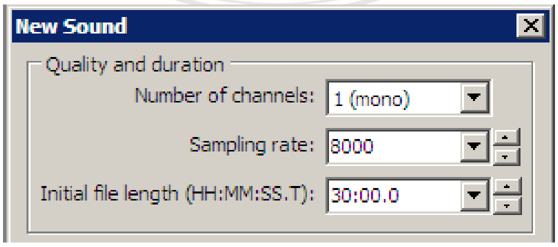
3) ไมโครโฟน ยี่ห้อ AIWA รุ่น DM-H200

4) โปรแกรม GoldWave Version 5.23 ติดตั้งบนระบบปฏิบัติการ WindowsXP

3.1.2 ขั้นตอนในการบันทึกข้อมูลเสียงพูด

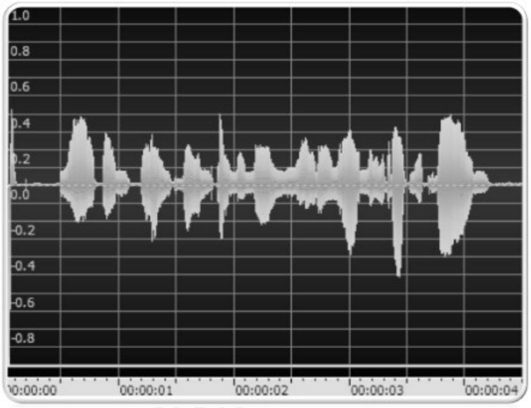
ขั้นตอนการบันทึกสัญญาณเสียงพูดด้วยโปรแกรม GoldWave version 5.5 มีดังนี้

1) กำหนดค่าการบันทึกสัญญาณเสียงพูด กำหนดให้ความแรงของสัญญาณเสียงพูดมีขนาด -1.0 ถึง 1.0 โดยเลือกเมนูคำสั่ง File>New จะปรากฏหน้าต่าง New Sound เป็นการกำหนดคุณภาพเสียงและระยะเวลาการบันทึก กำหนดค่าที่จำนวนช่องสัญญาณเสียงเป็น 1 (mono), จำนวนของการสุ่มค่าตัวอย่าง ตั้งค่าเป็น 8,000 และช่วงระยะเวลาการบันทึก ตั้งค่าเป็น 30:00.0



รูปที่ 3.2 การตั้งค่าเริ่มต้นบันทึกเสียงในโปรแกรม GoldWave

2) การบันทึกสัญญาณเสียงพูดนำเมาส์คลิกที่ปุ่มการเริ่มต้นบันทึก Starts recording เพื่อทำการบันทึกไฟล์เสียงใน Directory ที่กำหนดโดยใช้ผู้ทดสอบอายุระหว่าง 18 ถึง 40 ปี เป็นผู้ชาย 10 คน ผู้หญิง 10 คน เป็นเวลา 30 นาที



รูปที่ 3.3 ตัวอย่างสัญญาณเสียงพูดในกลุ่มผู้พูดที่ใช้ในการวิเคราะห์

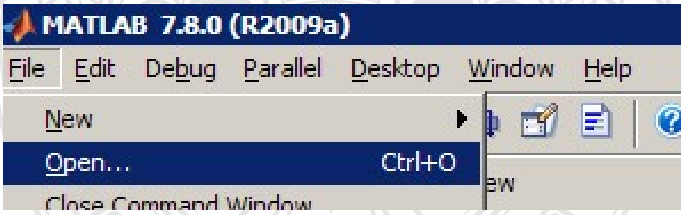
3.2 การสร้างระบบการบีบอัดสัญญาณเสียงพูด

ในการสร้างระบบการบีบอัดสัญญาณเสียงพูด จะต้องหาค่าลักษณะสำคัญของสัญญาณเสียงพูดแต่ละสัญญาณ โดยการสกัดค่าพารามิเตอร์ LPC และ LSP เพื่อนำค่าสัมประสิทธิ์ตัวแทนเสียงที่สกัดได้ไปใช้ในโครงข่ายจัดการตนเองอัลกอริทึ่ม KSOFM ต่อไป

3.2.1 การสกัดค่าพารามิเตอร์ LPC

การสกัดค่าพารามิเตอร์ LPC มีขั้นตอนการปฏิบัติการดังนี้

1) เปิดโปรแกรม MatLab เรียกไฟล์ *.m โดยเลือกเมนูคำสั่ง File>Open



รูปที่ 3.4 การใช้คำสั่งเปิด *.m ไฟล์

2) การอ่านไฟล์เสียงด้วยคำสั่ง wavread เปิดไฟล์ main.m จะปรากฏรหัสคำสั่งในบรรทัดที่ 4 [in_speech,fs,bits] = wavread ('ชื่อไฟล์เสียง'); ให้ใส่ไดเร็คทอรี่และชื่อไฟล์

```
clear
clc
% get wave-file
[in_speech,fs,bits] = wavread('org.wav')
% prepare and preprocess wave-file
aux_speech=in_speech;
in_speech = in_speech - mean(in_speech(:
speechnorm = resample(in_speech,8000,fs)
speech_filt=preprocess(speechnorm,fs);
wavwrite(speech_filt,'original.wav');
```

รูปที่ 3.5 การใช้คำสั่งการอ่านไฟล์เสียง

3) การสกัดค่าพารามิเตอร์ LPC จะเก็บค่าผลลัพธ์ตัวแปรใน LPC_coeff ในรูปของเมตริกซ์ซึ่งจะเขียนคำสั่งในฟังก์ชั่น main.m

```
% Frame variables
sp_size=size(speech);                          %speech size
frame=240;                                     %frame size
frame_numb=fix(sp_size(2)/frame)+1;            %number of frames
speech(1,(frame_numb*frame))=0;                %cutting the end
speechy=speech;                                %speech for windowing
sp_matrix=(reshape(speech,frame,frame_numb));  %speech in matrix
sp_matrix=sp_matrix';                          %transposed speech matrix
LPC_order=10;                                  %order of LPC-Analysis
LPC_coeff=zeros(frame_numb,LPC_order+1);       %LPC coefficients matrix

% window variables
overlap=100;                                   %50 samples on each side
win=hamming(frame+overlap);                    %Hamming window

% Analysis
  LPC_coeff=get_lpc(speechy,frame_numb,frame,win,overlap,LPC_order);
```

รูปที่ 3.6 การใช้ตัวแปรเก็บค่า LPC_coeff

4) การวิเคราะห์ค่าสัมประสิทธิ์ LPC_coeff ในฟังก์ชั่น main.m จะเรียกใช้ฟังก์ชั่น get_lpc โดยเขียนฟังก์ชั่นเรียกใช้ (Callback Function) เป็น m-file อีกไฟล์หนึ่งคือ GET_LPC.m

```
function [LPC_matrix,gaini]=get_lpc(speech,c,b,window,d,e);
```

ฟังก์ชั่น get_lpc

```
% b=framesize
% c=framenumber
% d=overlap
% e=LPC_order
 %function variables
win=window';
coeff=zeros(c,e+1);
 %windowing and lpc analysis
for n=0:c-1
  if n==0
   coeff(n+1,:)=lpc2(speech(n*b+1:(n+1)*b+d/2).* win(1:b+d/2),e);
   weigh(n+1,:)=lpcbwexp(coeff(n+1,:),0.015);        %15 Hz bandwidth
  elseif n>0 & n<c-1
   coeff(n+1,:)=lpc2(speech(n*b+1-d/2:(n+1)*b+d/2).* win,e);
   weigh(n+1,:)=lpcbwexp(coeff(n+1,:),0.015);        %15 Hz bandwidth
  elseif n==c-1
   coeff(n+1,:)=lpc2(speech(n*b+1-d/2:(n+1)*b).* win(1:b+d/2),e);
   weigh(n+1,:)=lpcbwexp(coeff(n+1,:),0.015);        %15 Hz bandwidth expansion
  end
end
 %LPC coefficients matrix
LPC_matrix=weigh;
```

รูปที่ 3.7 การสร้าง Callback Function วิเคราะห์ค่า LPC_coeff

5) การแสดงค่าผลลัพธ์ของตัวแปร LPC_coeff ใน workspace โดยการเก็บในไฟล์ Excel ภายใต้ชื่อ LPC_coeff.xls จะใช้คำสั่ง xlswrite('LPC_coeff.xls',LPC_coeff) โดยป้อนผ่านบรรทัดคำสั่ง (command line)

	B	C	D	E	F	G	H	I	J
1	-0.73444	0.82792	-0.72057	0.725346	-0.55509	0.620613	-0.34503	0.413684	-0.25117
1	-0.74149	0.885377	-0.73744	0.68888	-0.54541	0.665435	-0.34469	0.391633	-0.17533
1	-0.71943	0.682767	-0.49176	0.568462	-0.49532	0.515827	-0.32989	0.40976	-0.21702
1	-0.50678	0.484792	-0.48443	0.496055	-0.33831	0.468145	-0.27401	0.282037	-0.17785
1	-0.88484	0.501346	-0.41733	0.795915	-0.62694	0.452078	-0.28052	0.358983	-0.23584
1	-0.60612	0.499264	-0.45154	0.53975	-0.29729	0.433909	-0.3379	0.343841	-0.03627
1	-0.66181	0.331753	-0.49095	0.480416	-0.36658	0.367415	-0.21966	0.227138	-0.1776
1	-0.49408	0.522851	-0.63089	0.594882	-0.41502	0.485965	-0.34683	0.273647	-0.18304
1	-0.36357	0.547977	-0.51523	0.627349	-0.40135	0.550501	-0.27485	0.355253	-0.13124
1	-0.28803	0.531652	-0.30496	0.555004	-0.32584	0.440342	-0.09645	0.213951	-0.11865
1	-0.29432	0.724186	-0.39794	0.800242	-0.38698	0.541538	-0.15135	0.376676	-0.14649
1	-0.36869	0.503991	-0.55307	0.52863	-0.4933	0.44971	-0.22775	0.270723	-0.1802
1	-1.20636	0.675089	-1.01952	0.963759	-0.63391	0.722875	-0.31897	0.189847	-0.35406

รูปที่ 3.8 ค่าสัมประสิทธิ์ LPC_coeff ในไฟล์ Excel

3.2.2 การสกัดค่าพารามิเตอร์ LSP

การสกัดค่าพารามิเตอร์ LSP มีขั้นตอนการปฏิบัติการดังนี้

1) การสกัดค่าพารามิเตอร์ LSP จะใช้ฟังก์ชั่นในการแปลงค่าสัมประสิทธิ์จาก LPC เป็น LSP (LPC/LSP Conversion) ซึ่งเขียนไว้ในไฟล์ main.m โดยเรียกใช้งานผ่านฟังก์ชั่นเรียกใช้ (Callback Function) คือ ฟังก์ชั่น lpc_lsp ไว้ใน m-file อีกไฟล์หนึ่งคือไฟล์ LPC_LSP.m แสดงดังรูปที่ 3.9

2) การวิเคราะห์ค่าสัมประสิทธิ์ LSP_coeff ในฟังก์ชั่น main.m จะเรียกใช้ฟังก์ชั่น lpc_lsp โดยเขียนฟังก์ชั่นเรียกใช้ (Callback Function) เป็น m-file อีกไฟล์หนึ่งคือ LPC_LSP.M แสดงดังรูปที่ 3.10

3) การแสดงค่าผลลัพธ์ของตัวแปร LSPC_coeff ใน workspace โดยการเก็บในไฟล์ Excel ภายใต้ชื่อ LPC_coeff.xls จะใช้คำสั่ง xlswrite('LSP_coeff.xls',LSP_coeff) โดยป้อนผ่านบรรทัดคำสั่ง (command Line) แสดงดังรูปที่ 3.11

```
% get wave-file
[in_speech,fs,bits] = wavread('org.wav');        % getting from speech.m
% prepare and preprocess wave-file
aux_speech=in_speech;
in_speech = in_speech - mean(in_speech(:));
speechnorm = resample(in_speech,8000,fs);
speech_filt=preprocess(speechnorm,fs);
wavwrite(speech_filt,'original.wav');
speech = speech_filt';
% Frame variables
sp_size=size(speech);                                %speech size
frame=240;                                           %frame size
frame_numb=fix(sp_size(2)/frame)+1;                  %number of frames
speech(1,(frame_numb*frame))=0;                      %cutting the end
speechy=speech;                                      %speech for windowing
sp_matrix=(reshape(speech,frame,frame_numb));        %speech in matrix
sp_matrix=sp_matrix';                                %transposed speech matrix
LPC_order=10;                                        %order of LPC-Analysis
LPC_coeff=zeros(frame_numb,LPC_order+1);             %LPC coefficients matrix
% window variables
overlap=100;                                         %50 samples on each side
win=hamming(frame+overlap);                          %Hamming window
% Analysis
LPC_coeff=get_lpc(speechy,frame_numb,frame,win,overlap,LPC_order);
% lpc/lsp conversion
LSP_coeff=lpc_lsp(LPC_coeff);
```

ฟังก์ชั่นแปลงค่า LPC เป็น LSP

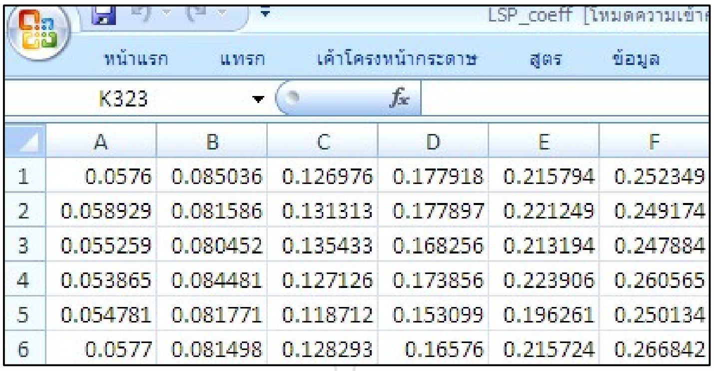
รูปที่ 3.9 ฟังก์ชั่นแปลงค่า LPC เป็น LSP ในไฟล์ main.m

```
function lsp=lpc_lsp(lpc)
```

ฟังก์ชั่นแปลงค่า LP

```
[nf,p1]=size(lpc);
p = p1-1;
p2 = fix(p/2);
d=0.5/pi;
 if rem(p,2)                          % odd order
  for k=1:nf
    aa=[lpc(k,:) 0];
    r = aa + fliplr(aa);
    q = aa - fliplr(aa);
    fr = sort(angle(roots(r)));
    fq = [sort(angle(roots(deconv(q,[1 0 -1])))); 0];
    f = [fr(p2+2:p+1).' ; fq(p2+1:p).'];
    f(p+1) = [];
    lsp(k,:) = d*f(:).';
  end
else
  for k=1:nf
    aa=[lpc(k,:) 0];
    r = aa + fliplr(aa);
    q = aa - fliplr(aa);
    fr = sort(angle(roots(deconv(r,[1 1]))));
    fq = sort(angle(roots(deconv(q,[1 -1]))));
    f = [fr(p2+1:p).' ; fq(p2+1:p).'];
    lsp(k,:) = d*f(:).';
  end
end
```

รูปที่ 3.10 การสร้าง Callback Function วิเคราะห์ค่า LSP_coeff



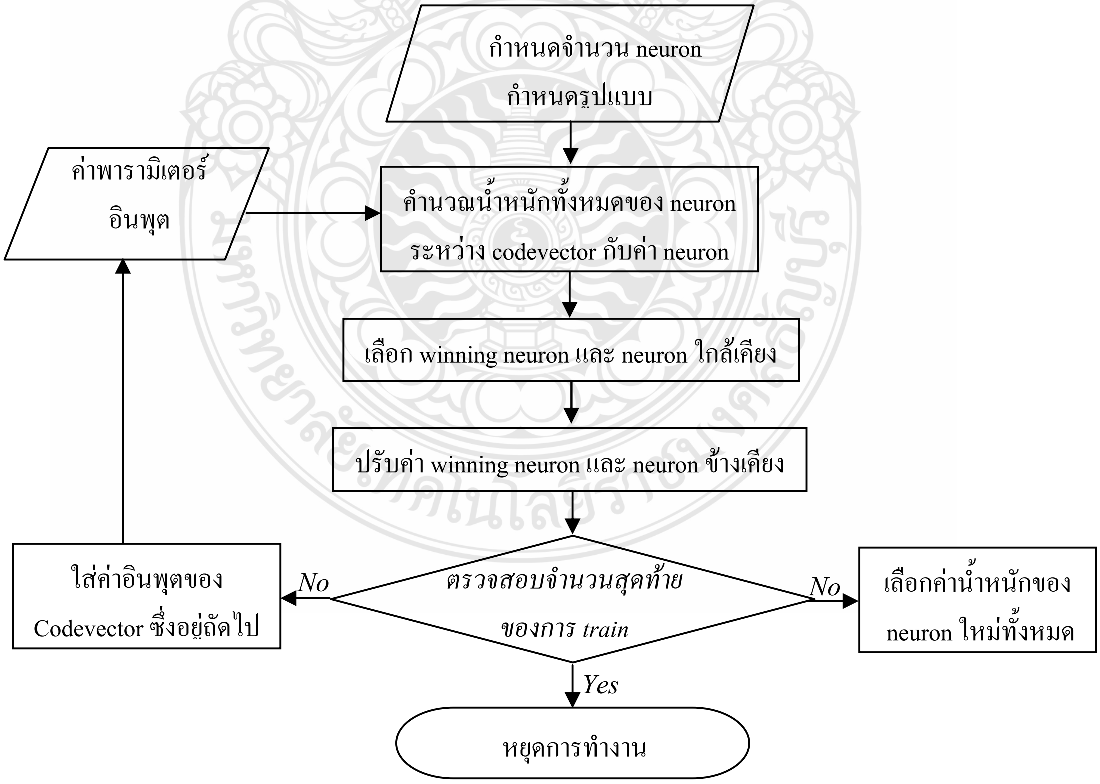
	A	B	C	D	E	F
1	0.0576	0.085036	0.126976	0.177918	0.215794	0.252349
2	0.058929	0.081586	0.131313	0.177897	0.221249	0.249174
3	0.055259	0.080452	0.135433	0.168256	0.213194	0.247884
4	0.053865	0.084481	0.127126	0.173856	0.223906	0.260565
5	0.054781	0.081771	0.118712	0.153099	0.196261	0.250134
6	0.0577	0.081498	0.128293	0.16576	0.215724	0.266842

รูปที่ 3.11 ค่าสัมประสิทธิ์ LSP_coeff ในไฟล์ Excel

3.3 การออกแบบโครงข่ายประสาทเทียม

ในการแบ่งกลุ่มของโค้ดเวกเตอร์อินพุต จะถูกจัดแบ่งเป็นกลุ่มๆ ตามลักษณะรูปแบบที่กำหนดไว้ เวกเตอร์โค้ดอินพุตที่มีลักษณะคล้ายกันจะถูกวางในตำแหน่งเดียวกัน ซึ่งอาจจะใช้ปมประสาทดียวกันหรือปมประสาทที่ใกล้เคียงกันทำให้โค้ดเวกเตอร์อินพุตซ้ำแล้วซ้ำเล่า เมื่อมีโค้ดเวกเตอร์อินพุตใหม่เข้ามายังระบบและค่าเฟสก็จะเริ่มถูกประมวลใหม่เช่นกัน ส่งผลให้ค่าน้ำหนักของโค้ดเวกเตอร์อินพุตที่ถูกเลือก (Winning neuron) จะถูกปรับค่าหรือปรับแต่งให้มีการตอบสนองต่อโครงข่ายมากที่สุด

3.3.1 แผนผังโครงข่ายการจัดการตนเอง



รูปที่ 3.12 ผังการทำงานของโครงข่ายการจัดการตนเอง

การสร้างไฟล์สัญญาณเสียงบีบอัดจากโครงข่ายปมประสาทแบบจัดการตนเองมีขั้นตอนดังนี้

1) เรียกสัมประสิทธิ์สัญญาณเสียงต้นฉบับในแต่ละเฟรมและสัมประสิทธิ์ของปมประสาทในแต่ละเฟรมมาเป็นใช้ในโปรแกรม

2) แปลงค่าสัมประสิทธิ์ปมประสาทในแต่ละเฟรมให้เป็นค่าเมตริกซ์ m x n เพื่อจะนำไปสร้างค่าน้ำหนักเมตริกซ์

3) สร้างสัมประสิทธิ์สัญญาณเสียงบีบอัดในแต่ละเฟรมโดยการนำค่าน้ำหนักเมตริกซ์ในแต่ละเฟรมคูณกับสัมประสิทธิ์สัญญาณเสียงต้นฉบับในแต่ละเฟรม

4) รวมค่าสัมประสิทธิ์สัญญาณเสียงบีบอัดในแต่ละเฟรมเพื่อสร้างสัญญาณเสียงบีบอัดที่ภายในบรรจุข้อมูลข่าวสารที่เหมือนกับสัญญาณเสียงต้นฉบับโดยสัมประสิทธิ์รวมนี้จะใช้เป็นตัวแปรในการเข้ารหัสเสียงแบบ Microsoft Wave

3.3.2 ขั้นตอนการสร้างโครงข่ายจัดการตนเอง

ขั้นตอนการทำงานของโครงข่ายจัดการตนเองมีขั้นตอนการปฏิบัติการดังนี้

1) อ่านไฟล์เสียงต้นฉบับจัดแบ่งเฟรมเสียงและสกัดค่าสัมประสิทธิ์ LPC–10 เพื่อใช้สร้างอินพุตเวกเตอร์ที่ใช้ในการสร้างโครงข่ายจัดการตนเอง โดยใช้โปรแกรม main.m

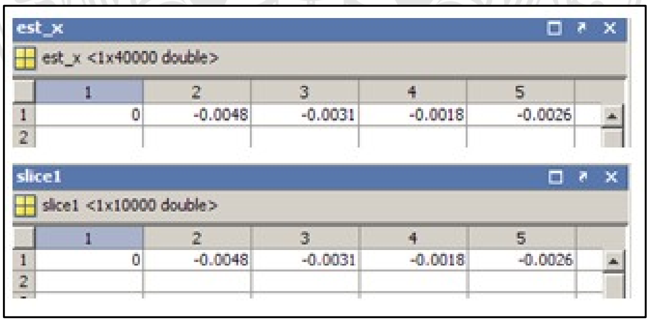
```
clc;                                                    %ล้างจอ
partition = -1:1/64:1;                                  %แบ่งระดับความแรงเป็น 128 ระดับ
codebook(1,1) = -1;                                     %แบ่งกำหนดขนาดสูงสุดด้านลบ
codebook(1,130) = 1;                                    %แบ่งกำหนดขนาดสูงสุดด้านบวก
for i=1:1:128
  codebook(i+1) = (partition(1,i)+partition(1,(i+1)))/2;  %สร้างcodebook
end
[samp, Fs, nbits] = wavread('org.wav');                 %อ่านไฟล์เสียง
[index,quant] = quantiz(samp,partition,codebook);       %ปรับระดับขนาดเสียง 128 ระดับ
s = filter(1,[1 1/2 1/3 1/4],quant);                    %กรองความถี่เสียง
a = lpc(s,10);                                          %สกัดค่าสัมประสิทธิ์ LPC-10
est_x = filter([0 -a(2:end)],1,s);                      %ค่าประมาณเสียงด้วย LPC

for (i=1:1:10000)                                       % แบ่งเฟรมเป็น 8 เฟรม
   slice1(1,i)   = est_x(1,i);                          %เฟรม 1 ถึง 10000
```

รูปที่ 3.13 โปรแกรมในการจัดแบ่งเฟรมเสียง

```
slice2(1,i)   = est_x(1,(i+10000));                    %เฟรม 10001 ถึง 20000
slice3(1,i)   = est_x(1,(i+20000));                    %เฟรม 20001 ถึง 30000
slice4(1,i)   = est_x(1,(i+30000));                    %เฟรม 30001 ถึง 40000
slice5(1,i)   = est_x(1,(i+40000));                    %เฟรม 40001 ถึง 50000
slice6(1,i)   = est_x(1,(i+50000));                    %เฟรม 50001 ถึง 60000
slice7(1,i)   = est_x(1,(i+60000));                    %เฟรม 60001 ถึง 70000
slice8(1,i)   = est_x(1,(i+70000));                    %เฟรม 70001 ถึง 80000
end
   sum = [slice1,slice2,slice3,slice4,...
       slice5,slice6,slice7,slice8];                   %รวมเฟรมเสียงที่เปลี่ยนอัตราสุ่ม
   save('slice1.mat', 'slice1');                       %บันทึกตัวแปรเฟรมเสียงที่ 1
   save('slice2.mat', 'slice2');                       %บันทึกตัวแปรเฟรมเสียงที่ 2
   save('slice3.mat', 'slice3');                       %บันทึกตัวแปรเฟรมเสียงที่ 3
   save('slice4.mat', 'slice4');                       %บันทึกตัวแปรเฟรมเสียงที่ 4
   save('slice5.mat', 'slice5');                       %บันทึกตัวแปรเฟรมเสียงที่ 5
   save('slice6.mat', 'slice6');                       %บันทึกตัวแปรเฟรมเสียงที่ 6
   save('slice7.mat', 'slice7');                       %บันทึกตัวแปรเฟรมเสียงที่ 7
   save('slice8.mat', 'slice8');                       %บันทึกตัวแปรเฟรมเสียงที่ 8
   wavwrite(sum,8000,8,'sumorg.wav');     %สัญญาณเสียงที่ได้จากการประมาณค่า  LPC-10
```

รูปที่ 3.13 โปรแกรมในการจัดแบ่งเฟรมเสียง (ต่อ)



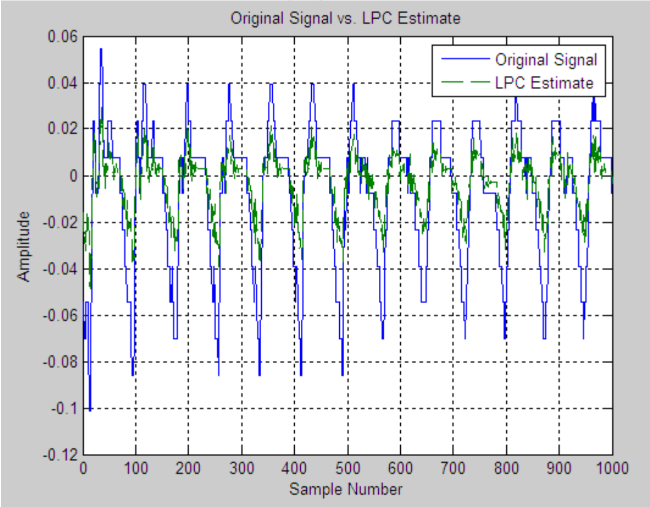
รูปที่ 3.14 สัญญาณเสียงสังเคราะห์จากสัมประสิทธิ์ LPC-10

ค่า est_x เป็นค่าสัมประสิทธิ์สัญญาณเสียงสังเคราะห์ LPC-10 ทั้งเฟรมขนาด 1x 80,000 ส่วนค่า slice1 เป็นค่าสัมประสิทธิ์สัญญาณเสียงสังเคราะห์ LPC-10 ในเฟรมที่1 ขนาด 1x 10,000 และ

การเปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงที่สกัดค่าด้วย LPC-10 จำนวน 1,000 ตัวอย่างที่ตำแหน่ง 4,001 ถึง 5,000 จะแสดงด้วยคำสั่ง plot

```
plot(1:1000,quant(4001:5000),1:1000,est_x(4001:5000),'--');
title('Original Signal vs. LPC Estimate');
xlabel('Sample Number'); ylabel('Amplitude'); grid;
legend('Original Signal','LPC Estimate')
```

รูปที่ 3.15 คำสั่งในการพล็อตกราฟ



รูปที่ 3.16 สัญญาณเสียงต้นฉบับกับสัญญาณเสียงสังเคราะห์ LPC-10

2) กำหนดรูปแบบโครงสร้างของโครงข่ายแบบหกเหลี่ยม เนื่องจากเป็นรูปแบบที่ใช้เวลาในการเข้าถึงข้อมูลได้เร็วที่สุด จากนั้นกำหนดจำนวนปมประสาทให้มี 128 ปม คำสั่งที่ใช้ในการสร้างโครงข่ายคือคำสั่ง newsom , การเรียนรู้โครงข่ายด้วยคำสั่ง train และสร้างปมประสาทด้วยคำสั่ง sim โดยโปรแกรมที่ใช้ในการสร้างโครงข่ายจัดการตนเองด้วยไฟล์ outksofm.m

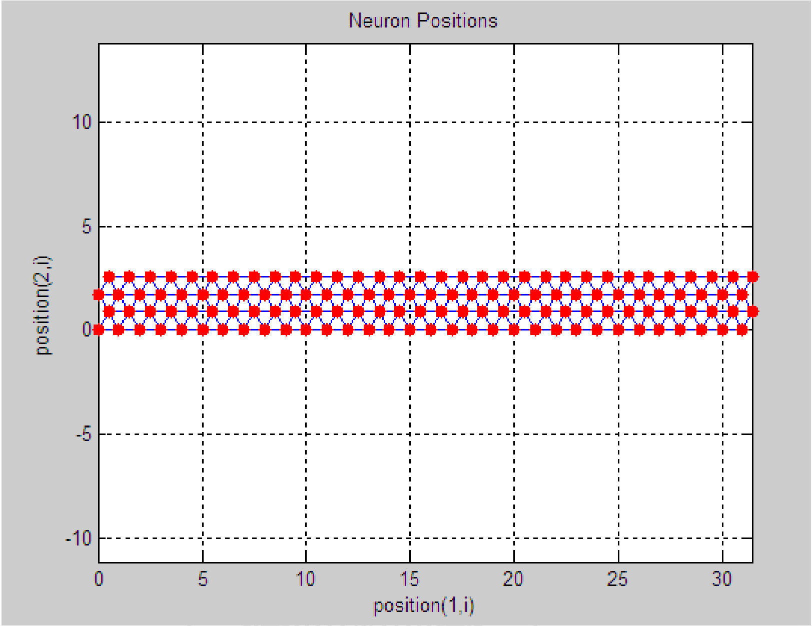
```
clc;                                  %ล้างจอ
disp('Frame1')                        %แสดงข้อความ
load  slice1.mat;                     %เรียกข้อมูลในเฟรมที่ 1ใส่ในตัวแปร
reslice2 = reshape(slice1,25,400)     %เปลี่ยนขนาดเมตริกซ์สียง 25x400
net = newsom(reslice1,[32,4]);        %สร้างโครงข่ายจัดการตนเองขนาด 128
```

รูปที่ 3.17 โปรแกรมการสร้างปมประสาทในชั้นลำดับจัดการตนเอง

```
net.trainParam.epochs = 1000;             %กำหนดการเรียนรู้ 1000 ครั้ง
net = train(net,reslice1);                %ฝึกฝนโครงข่าย
a1 = sim(net,reslice1);                   %หาค่าเอาต์พุตปมประสาท
target = full(a1);          %แปลงค่าเอาต์พุตปมประสาทให้เป็นเมตริกซ์ 25x400
for(i=1:1:25)
   result1 = 0
  for(j=1:1:400)
   result1 = result1 + target1(i,j);      %หาอัตราการใช้ข้อมูลในลำดับชั้น
  end
 num1(i,1) = result1;                     %แสดงผลการจัดข้อมูลในลำดับชั้น
end
for(i=1:1:25)
   if (num1(i,1)== 1)
       test1(i,1) = i;                    %หาตำแหน่งที่ต้องการลบข้อมูล
   end
end
bus1 = size(test1)
RT = bus1(1,1).*bus1(1,2);                %กำหนดตำแหน่งที่ต้องการลบ
for(i=1:1:(RT-10))
 if (test1(i,1) ~= 0)
       n1(i,1) = test1(i,1);
       layer(n1(i,1),:) = [];             %ลบข้อมูลในส่วนที่ซ้ำซ้อน
 end
end
len1 = size(layer1);                      %ขนาดข้อมูลแถวในข้อมูลเสียง
layer11 = reshape(layer1,1,(len1(1,1)*len1(1,2)));          %สร้างเมตริกซ์ 1xn
wavwrite(4.0*layer11,8000,8,'layer1.wav')                   % สร้างไฟล์เสียงเฟรมที่ 1
```

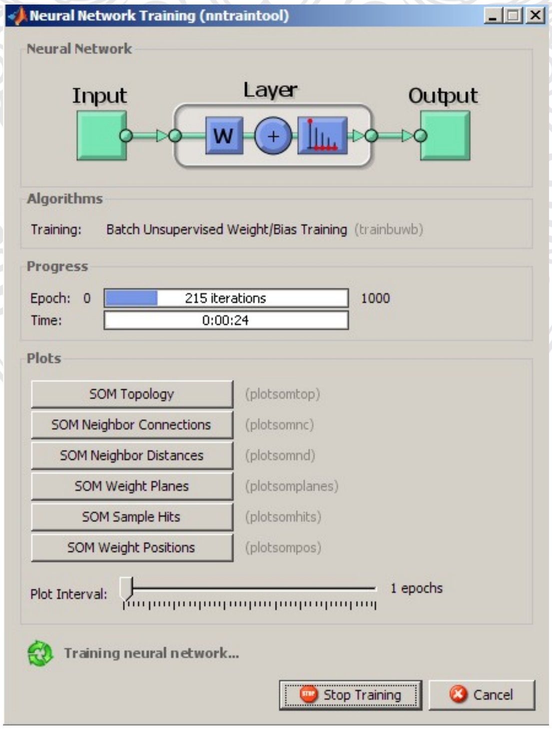
รูปที่ 3.17 โปรแกรมการสร้างปมประสาทในชั้นลำดับจัดการตนเอง (ต่อ)

3) เปิดโปรแกรม MatLab เรียกไฟล์ outksofm.m แล้วเลือก Debug> Run โปรแกรมจะทำการจัดการสร้างโครงข่ายจัดการตนเองแบบหกเหลี่ยมขนาด 32x4 จำนวน 128 ปมประสาท



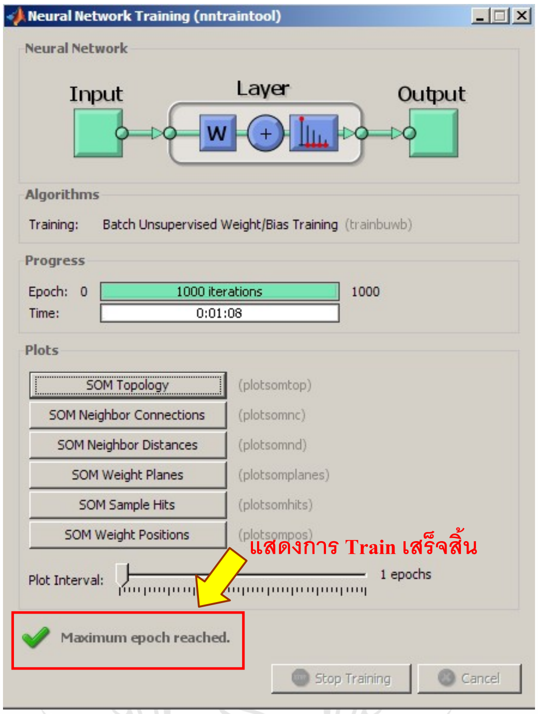
รูปที่ 3.18 โครงข่ายจัดการตนเองขนาด 32x4 จำนวน 128 ปมประสาท

4) กำหนดจำนวนการเรียนรู้โครงข่ายจัดการตนเองจากคำสั่ง train ในโปรแกรม outksofm.m จะดำเนินฝึกฝนจำนวน 1,000 ครั้ง เพื่อปรับค่าน้ำหนัก, หาระยะทางระหว่างปมประสาท



รูปที่ 3.19 การดำเนินการฝึกฝนข้อมูลโครงข่ายจัดการตนเอง

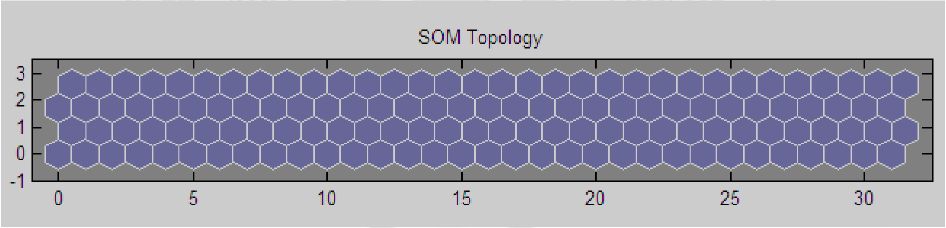
5) การแสดงสถานะการเรียนรู้โครงข่ายโดยเปิดหน้าต่าง Neural Network Training โปรแกรมจะดำเนินการฝึกฝนโครงข่ายรอจนขบวนการสิ้นสุดโดยสังเกตจากตำแหน่งล่างสุดจะปรากฏข้อความว่า Maximum epoch reached



รูปที่ 3.20 จุดสิ้นสุดของขบวนการฝึกฝนข้อมูลโครงข่าย

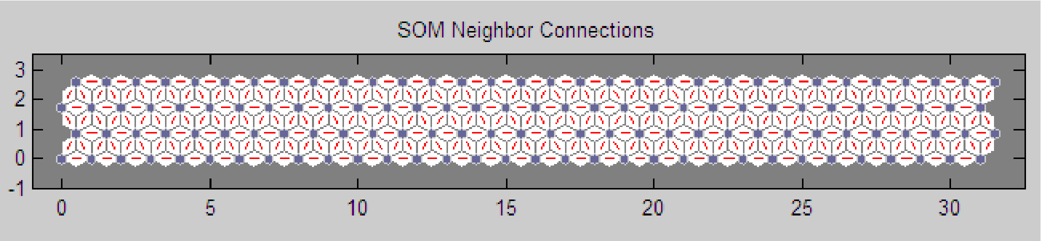
6) การแสดงผลการทำงานของโครงข่ายจัดการตนเองภายหลังการฝึกฝนเสร็จสิ้น มีรายละเอียดดังนี้

- การจัดรูปแบบโครงข่ายจะเปิดหน้าต่าง Neural Network Training และคลิกที่ตำแหน่ง Som Topology



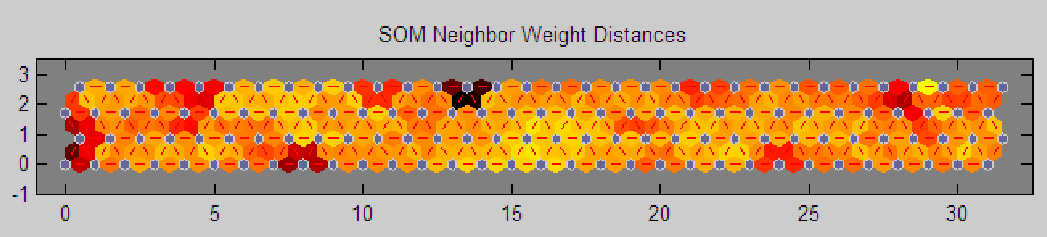
รูปที่ 3.21 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยม

- การเชื่อมต่อระหว่างปมประสาทข้างเคียงจะเปิดหน้าต่าง Neural Network Training และคลิกที่ตำแหน่งของคำสั่ง Som Neighbor Connection



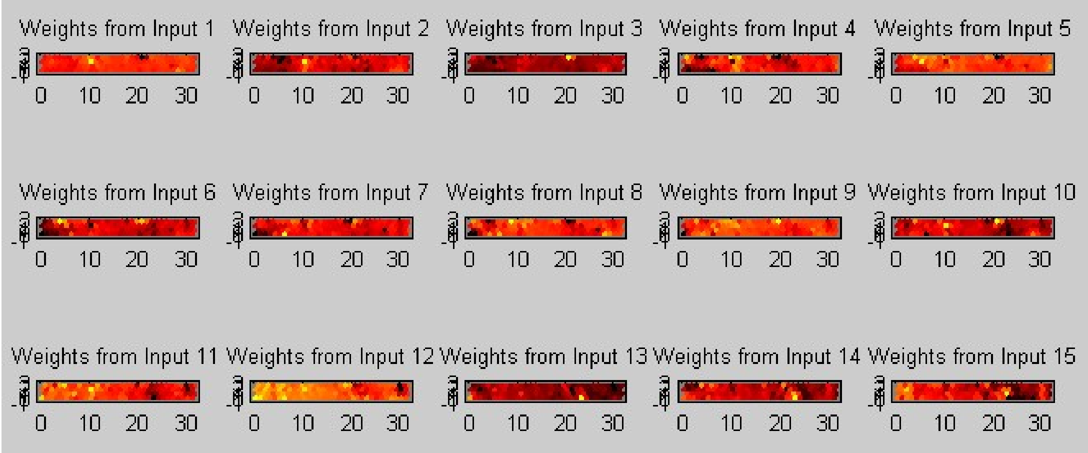
รูปที่ 3.22 การเชื่อมต่อของปมประสาทข้างเคียง

■ ค่าระยะทางการเชื่อมโยงระหว่างปมประสาทจะเปิดหน้าต่าง Neural Network Training และคลิกที่ตำแหน่งของคำสั่ง Som Neighbor distance



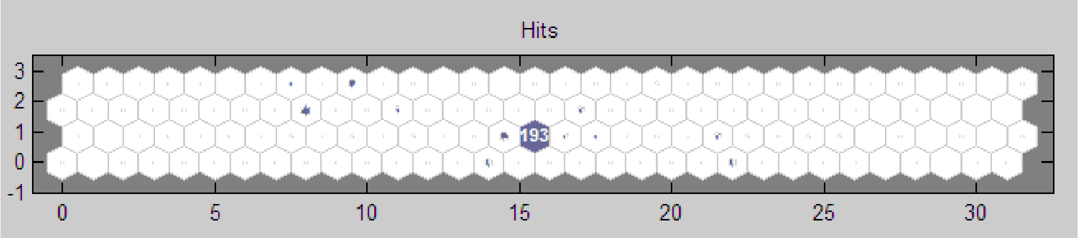
รูปที่ 3.23 ระยะทางเชื่อมโยงระหว่างปมประสาท

■ ค่าน้ำหนักแต่ละปมประสาทจะเปิดหน้าต่าง Neural Network Train และคลิกที่ตำแหน่งของคำสั่ง Som Weight Planes



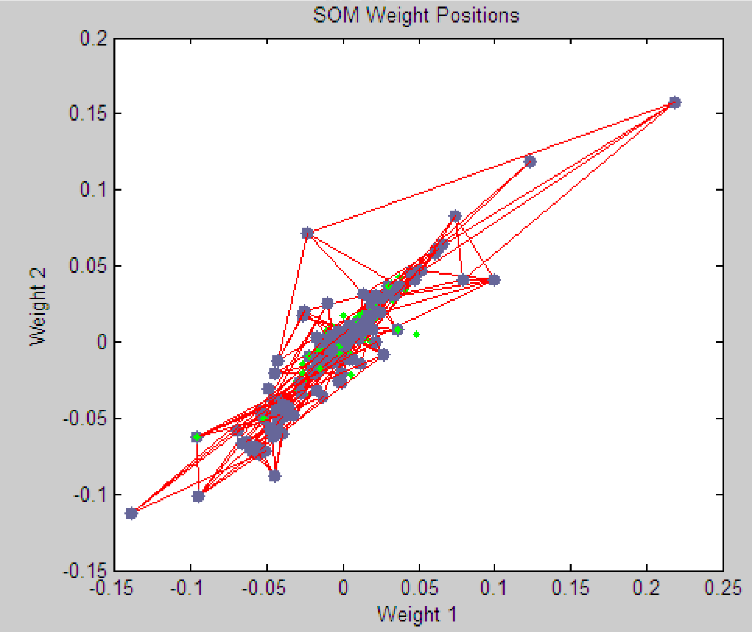
รูปที่ 3.24 น้ำหนักของปมประสาท

■ ค่าการกระจายตัวของข้อมูลเวกเตอร์ในแต่ละปมประสาทจะเปิดหน้าต่าง Neural Network Training และคลิกที่ตำแหน่งของคำสั่ง Som Sample Hits



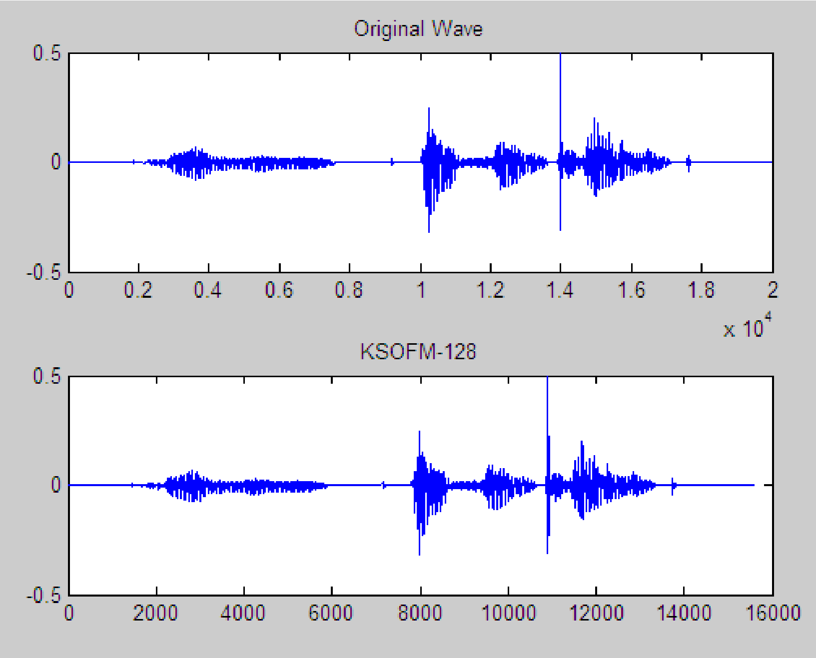
รูปที่ 3.25 การกระจายตัวของเวกเตอร์ภายในปมประสาท

■ ค่าน้ำหนักตำแหน่งของปมประสาทจะเปิดหน้าต่าง Neural Network Training และคลิกที่ตำแหน่งของคำสั่ง Som Sample Hits



รูปที่ 3.26 ตำแหน่งของค่าน้ำหนักของโครงข่ายปมประสาท

■ ค่าข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับจากไฟล์เสียง org.wav ในเฟรม แรกค่าตัวอย่างจำนวน 20,000 ตัวอย่างเมื่อผ่านการบีบอัดสัญญาณเสียงด้วยโครงข่ายการจัดการตนเองจะเหลือเฟรมเสียงจำนวน 14,000 ตัวอย่าง



รูปที่ 3.27 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด

3.3.3 การแบ่งขนาดโค้ดบุ๊คที่ใช้ในโครงข่ายประสาทเทียมแบบจัดการตนเอง

การกำหนดขนาดโค้ดบุ๊คเป็นการกำหนดค่าปมประสาทเอาต์พุตของโครงข่ายจัดการตนเอง โดยขนาดของโค้ดบุ๊คขึ้นอยู่กับการจัดเรียงตัวของลำดับชั้นของโครงข่าย เช่น โค้ดบุ๊คขนาด 128 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x4 โค้ดบุ๊คขนาด 256 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x8 โค้ดบุ๊คขนาด 512 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x16 และโค้ดบุ๊คขนาด 1,024 ต้องสร้างให้ลำดับชั้นมีขนาด 32x32 จะเห็นขนาดของโค้ดบุ๊คจะเป็นตามความสัมพันธ์ของ $n \times 2^n$

ลำดับขั้นตอนการสร้างโค้ดบุ๊คในโครงข่ายประสาทเทียมแบบจัดการตนเองมีดังนี้

- แบ่งเฟรมเสียงต้นฉบับให้มีความยาวเฟรมเหมาะสมกับขบวนการฝึกฝน
- กำหนดขนาดโค้ดบุ๊คเพื่อที่ออกแบบลำชั้นของโครงข่าย
- เขียนคำสั่งสร้างโครงข่ายประสาทเทียมแบบจัดการตนเอง
- ทดสอบการทำงานการกำหนดจำนวนลำดับชั้นและค่าขนาดโค้ดบุ๊ค

1) การสร้างโค้ดบุ๊คขนาด 128 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x4 โดยการกำหนดที่ [D1,D2,...] ของคำสั่ง newsom

```
net = newsom(P,[D1,D2,...],TFCN,DFCN,STEPS,IN)
```

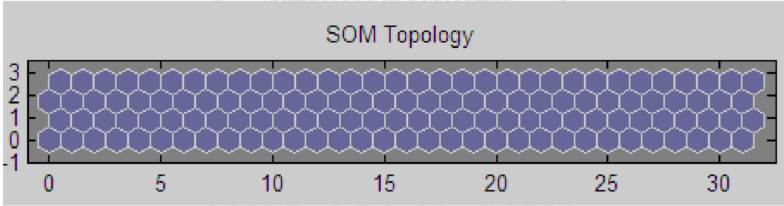
รูปที่ 3.28 คำสั่งสร้างลำดับชั้นปมประสาทของโค้ดบุ๊ค

```
disp('Frame1')
load  slice1.mat;
reslice1 = reshape(slice1,50,400);
net = newsom(reslice1,[32,4]);
net.trainParam.epochs = 1000;
net = train(net,reslice1);
a1 = sim(net,reslice1);
target1 = full(a1);
 layer1 = reslice1;
```

รูปที่ 3.29 คำสั่งในกำหนดจำนวนโค้ดบุ๊คของปมประสาท 128 ค่า

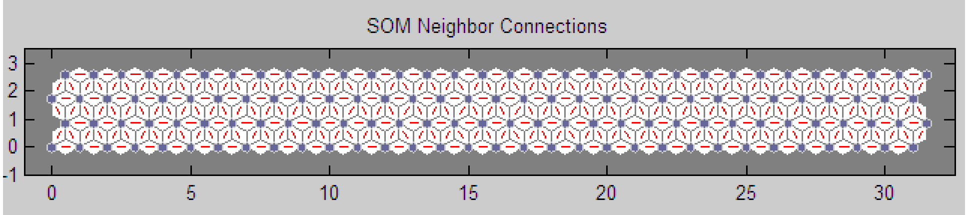
เมื่อมีการฝึกฝนและสร้างการเรียนรู้โครงข่ายจัดการตนเองจะเกิดผลลัพธ์ดังนี้

ก) การจัดโครงสร้างโครงข่ายจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsomtop(net) เพื่อแสดงกราฟการจัดการโครงข่าย



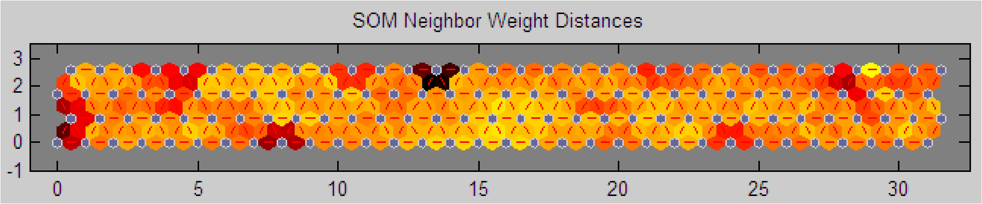
รูปที่ 3.30 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 128 ค่า

ข) การเชื่อมต่อระหว่างปมประสาทข้างเคียงจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsomnc(net) เพื่อแสดงกราฟการเชื่อมโยงโครงข่ายข้างเคียง



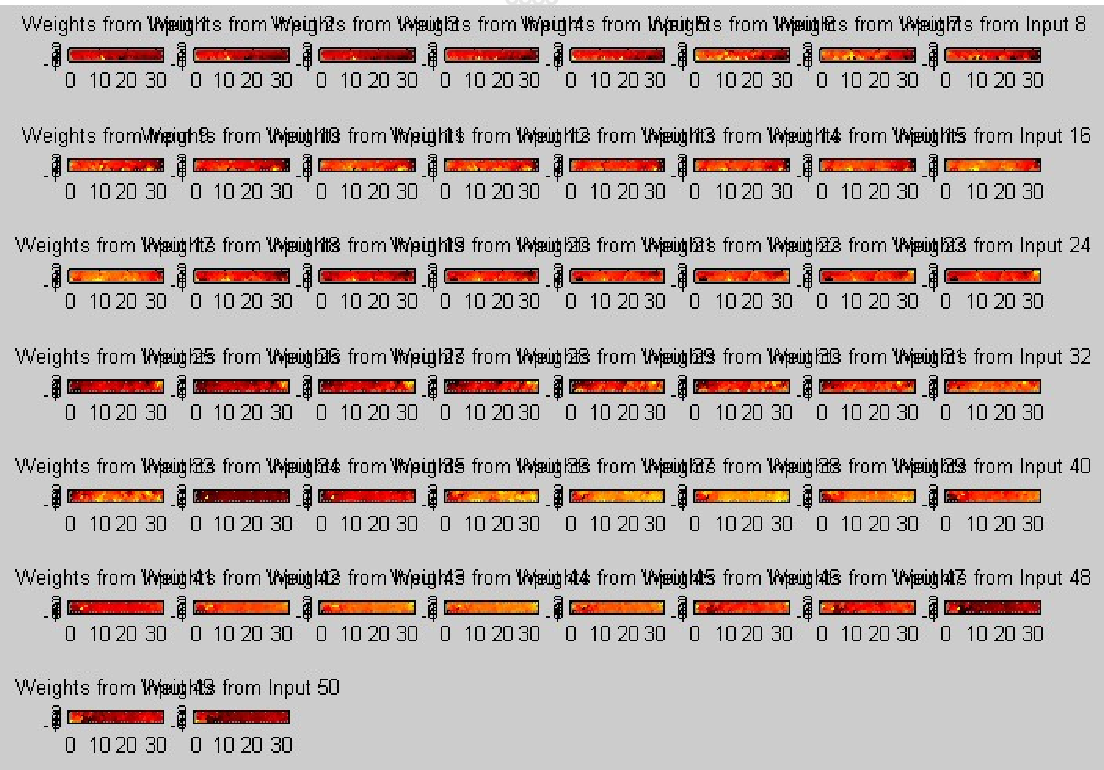
รูปที่ 3.31 การเชื่อมต่อของปมประสาทข้างเคียงขนาด 128 ค่า

ค) ค่าระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsomnd(net) เพื่อแสดงกราฟระยะทางการเชื่อมโยงระหว่างปมประสาท



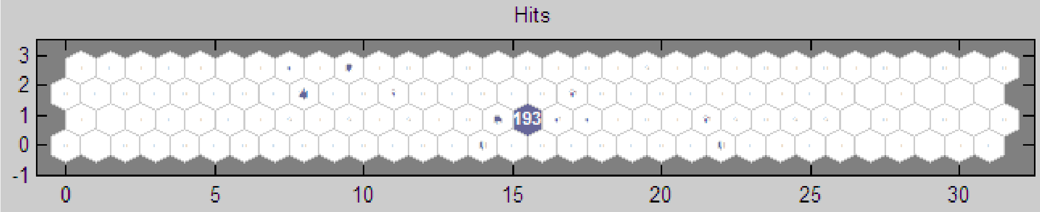
รูปที่ 3.32 ระยะทางการเชื่อมโยงระหว่างปมประสาทขนาด 128 ค่า

ง) ค่าน้ำหนักแต่ละปมประสาทจะเปิดหน้าต่างจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsomplanes(net) เพื่อแสดงกราฟค่าน้ำหนักแต่ละปมประสาท



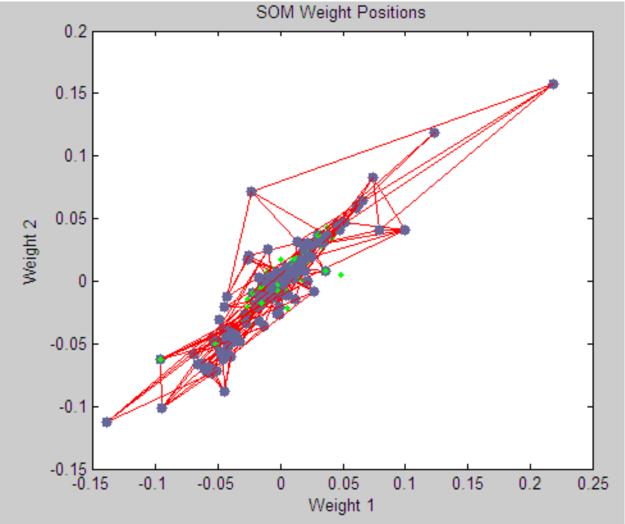
รูปที่ 3.33 น้ำหนักแต่ละปมประสาทจำนวนขนาด 128 ค่า

จ) ค่าการกระจายตัวของข้อมูลเวกเตอร์ในแต่ละปมประสาทจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsomhits(net,reslice1) เพื่อแสดงกราฟค่าการกระจายตัวของข้อมูลเวกเตอร์



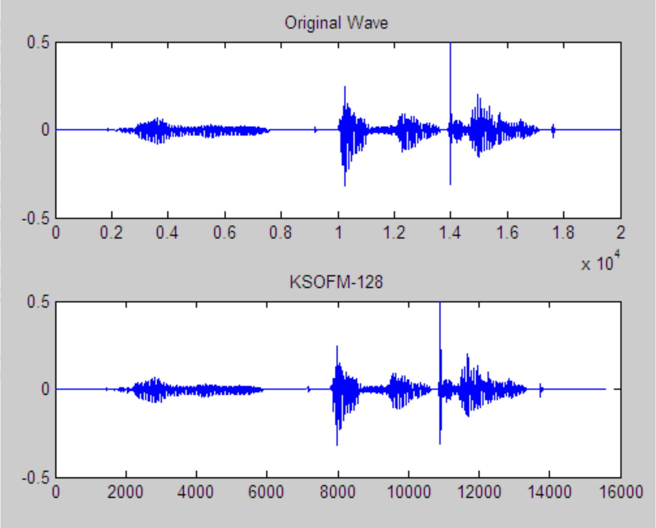
รูปที่ 3.34 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 128 ค่า

ฉ) ค่าน้ำหนักตำแหน่งของปมประสาทจำนวน 128 ค่ามิติขนาด 32x4 จะใช้คำสั่ง >> plotsompos (net,reslice1) เพื่อแสดงกราฟค่าน้ำหนักตำแหน่งของปมประสาท



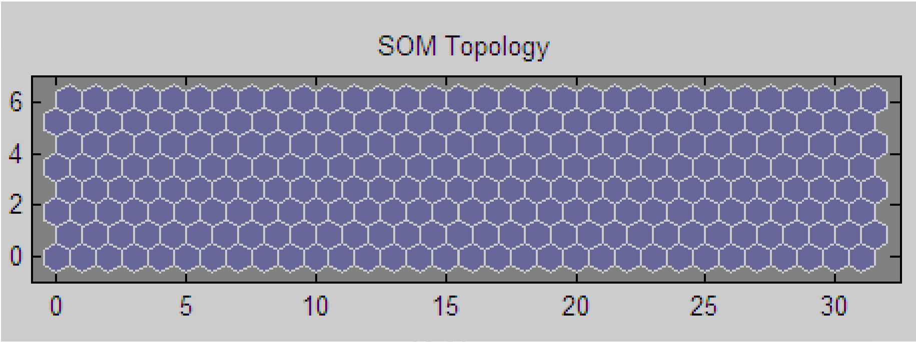
รูปที่ 3.35 น้ำหนักตำแหน่งของปมประสาทขนาด 128 ค่า

ช) ค่าข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับจากไฟล์เสียง org.wav ในเฟรมแรกค่าตัวอย่างจำนวน 20,000 ตัวอย่างเมื่อผ่านการบีบอัดสัญญาณเสียงด้วยโครงข่ายการจัดการตนเองจะเหลือเฟรมเสียงจำนวน 14,000 ตัวอย่าง

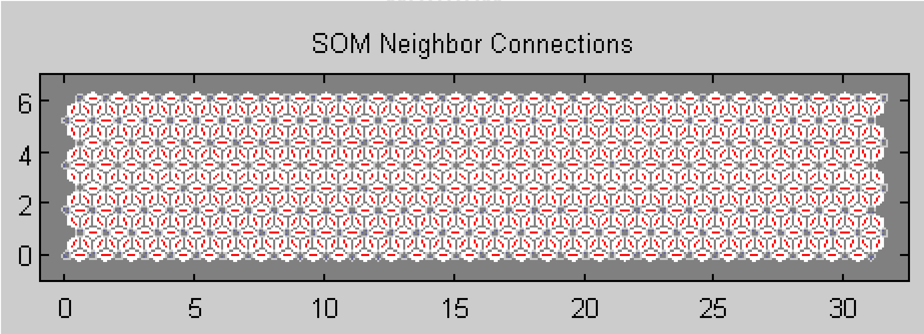


รูปที่ 3.36 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด

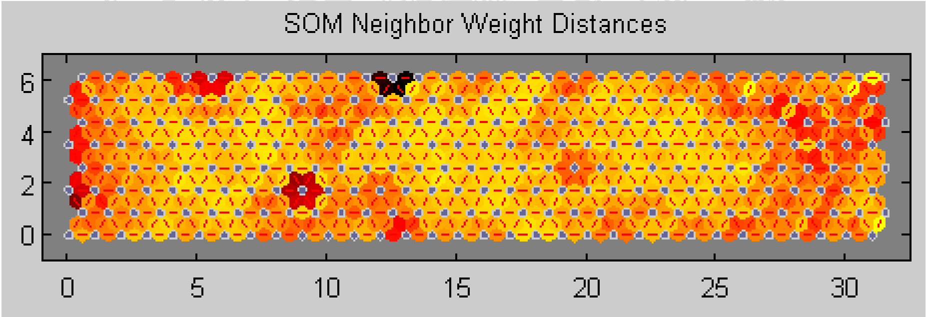
2) การสร้างโค้ดบุ๊คขนาด 256 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x8 โดยการกำหนดที่ [D1,D2,...] ของคำสั่ง newsom และเมื่อมีการฝึกฝนและสร้างการเรียนรู้โครงข่ายจัดการตนเอง เช่นเดียวกันกับข้อ ก) ถึง ช) จะได้ผลลัพธ์ดังนี้



รูปที่ 3.37 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 256 ค่า

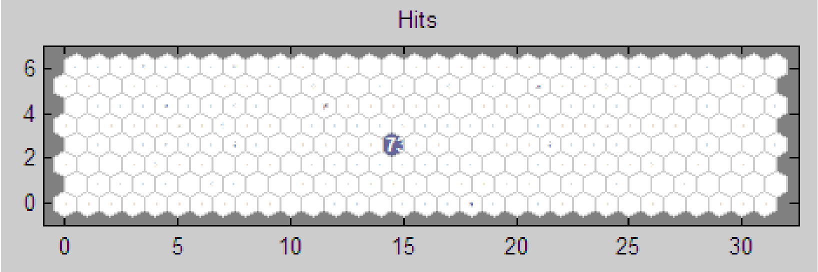


รูปที่ 3.38 การเชื่อมต่อของปมประสาทข้างเคียงขนาด 256 ค่า

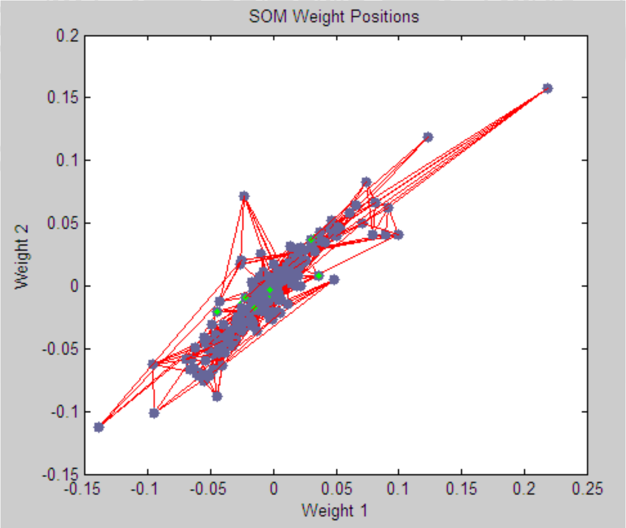


รูปที่ 3.39 ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 256 ค่า

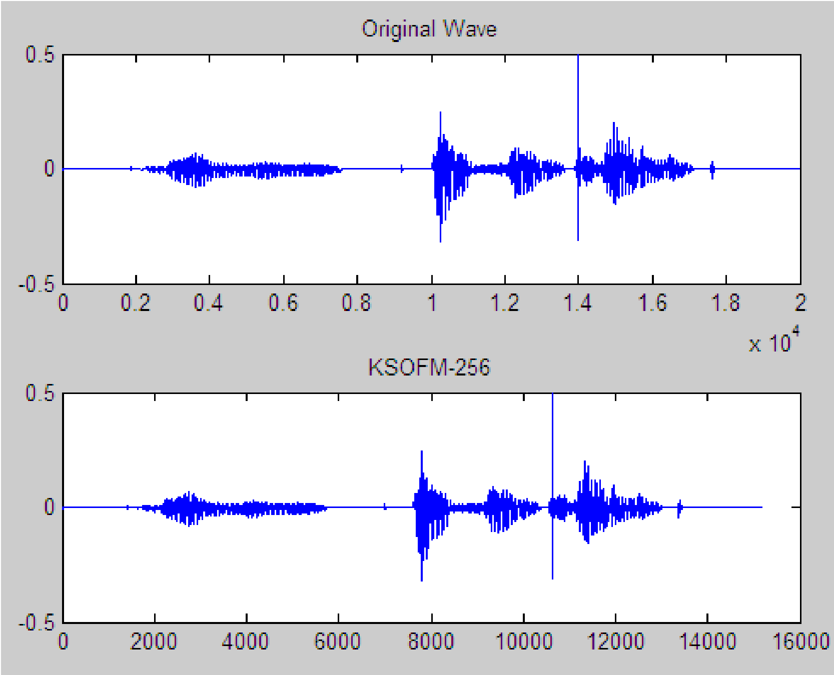
รูปที่ 3.40 น้ำหนักแต่ละปมประสาทจำนวนขนาด 256 ค่า



รูปที่ 3.41 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 256 ค่า

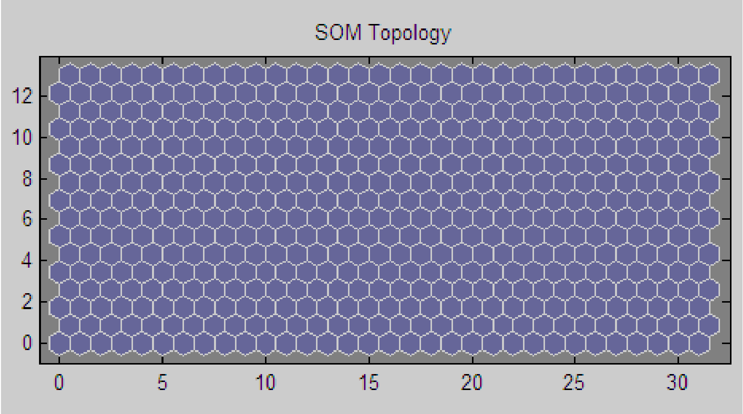


รูปที่ 3.42 น้ำหนักตำแหน่งของปมประสาทขนาด 256 ค่า

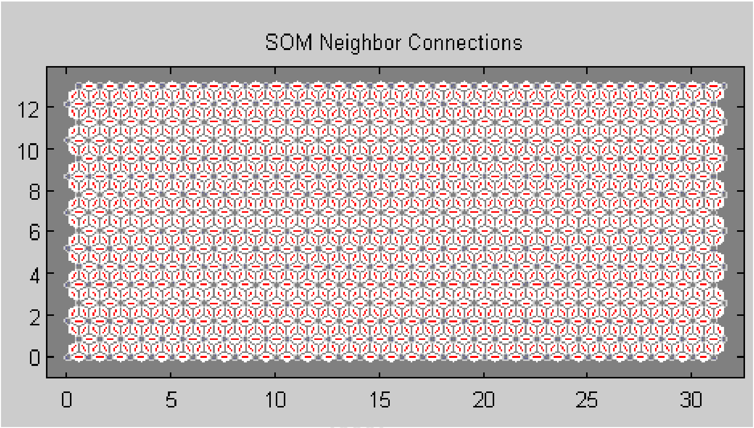


รูปที่ 3.43 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด

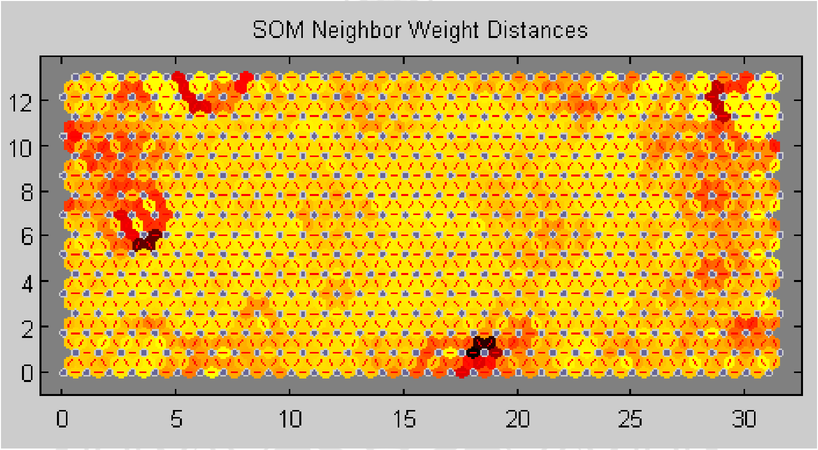
3) การสร้างโค้ดบุ๊คขนาด 512 จะต้องสร้างให้ลำดับชั้นมีขนาด 32x16 โดยการกำหนดที่ [D1, D2, ...] ของคำสั่ง newsom และเมื่อมีการฝึกฝนและสร้างการเรียนรู้โครงข่ายจัดการตนเอง เช่นเดียวกันกับข้อ ก) ถึง ช) จะได้ผลลัพธ์ดังนี้



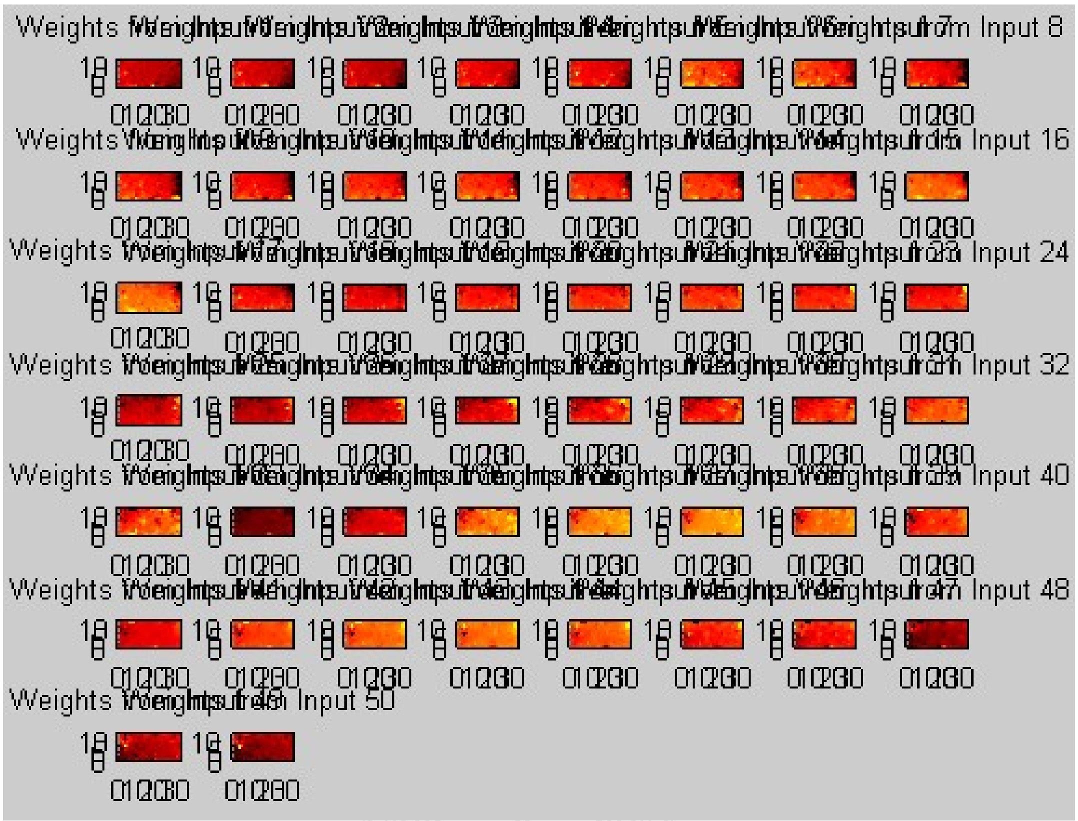
รูปที่ 3.44 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 512 ค่า



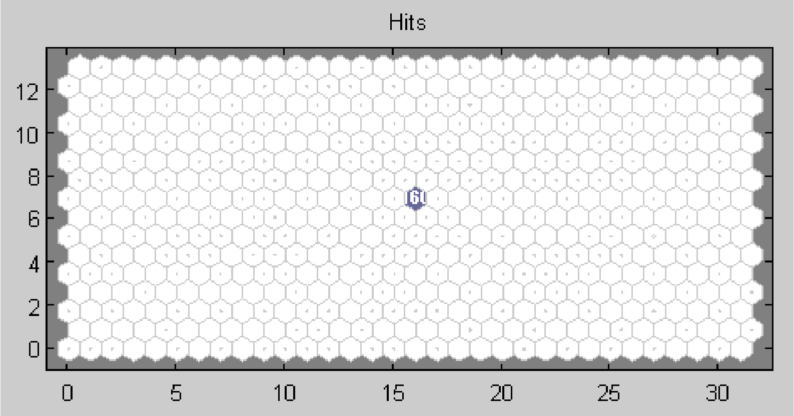
รูปที่ 3.45 การเชื่อมต่อของปมประสาทข้างเคียงขนาด 512 ค่า



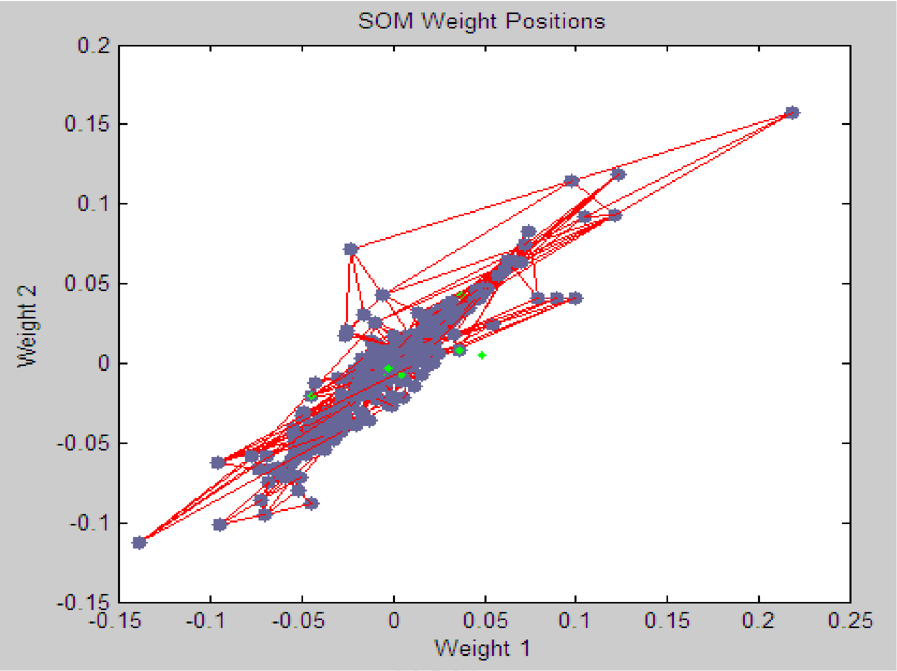
รูปที่ 3.46 ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 512 ค่า



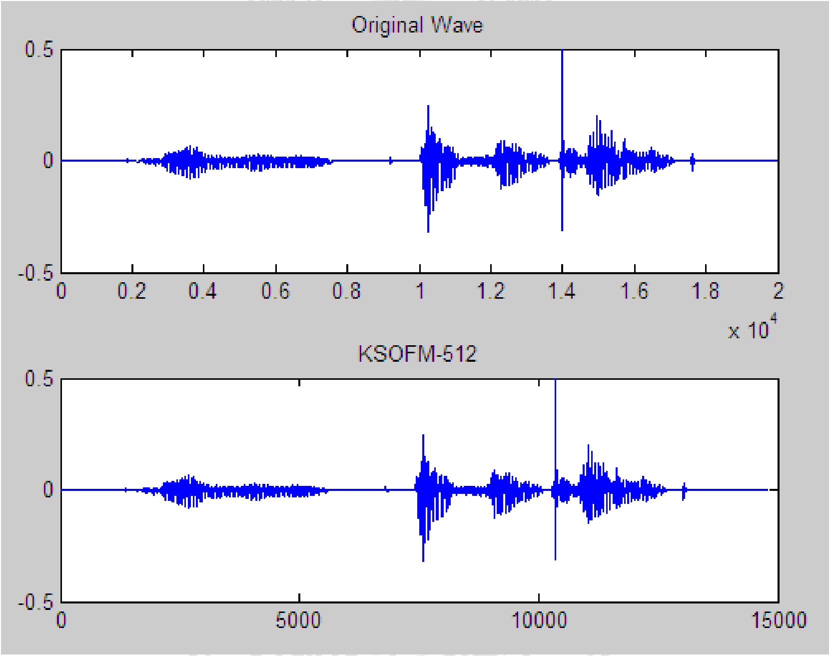
รูปที่ 3.47 น้ำหนักแต่ละปมประสาทจำนวน 512 ค่า



รูปที่ 3.48 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 512 ค่า

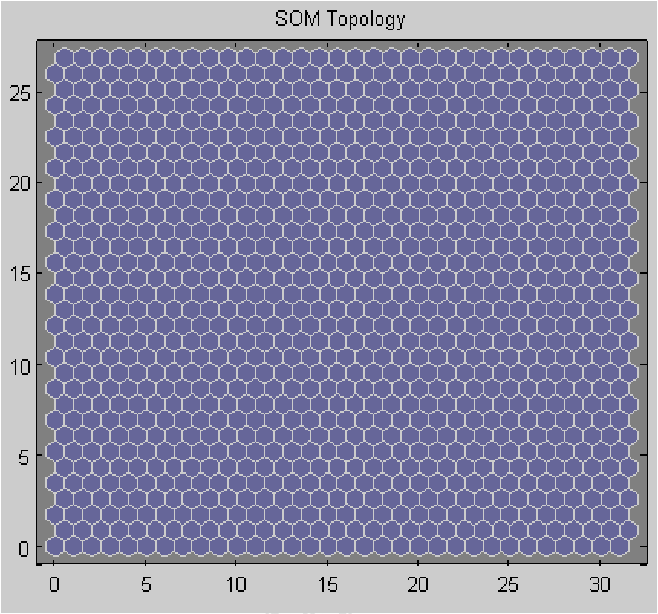


รูปที่ 3.49 น้ำหนักตำแหน่งของปมประสาทขนาด 512 ค่า

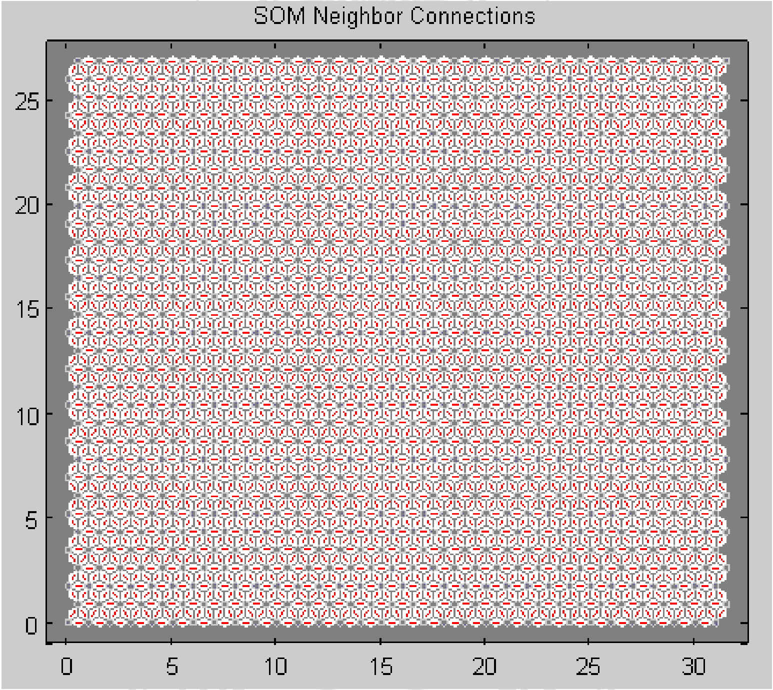


รูปที่ 3.50 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด

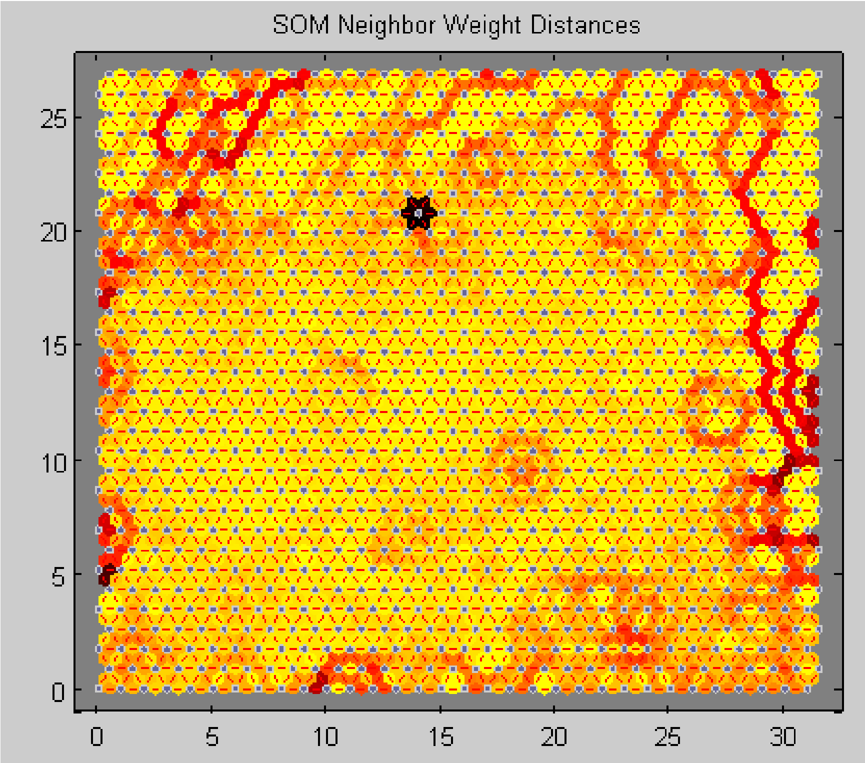
4) การสร้างโค้ดบุ๊คขนาด 1,024 จะต้องสร้างให้ลำดับชั้นมีขนาด 32 x32 โดยการกำหนดที่ [D1,D2,...] ของคำสั่ง newsom และเมื่อมีการฝึกฝนและสร้างการเรียนรู้โครงข่ายจัดการตนเองเช่นเดียวกันกับข้อ 3.3.3.1 ก) ถึง ช) จะได้ผลลัพธ์ดังนี้



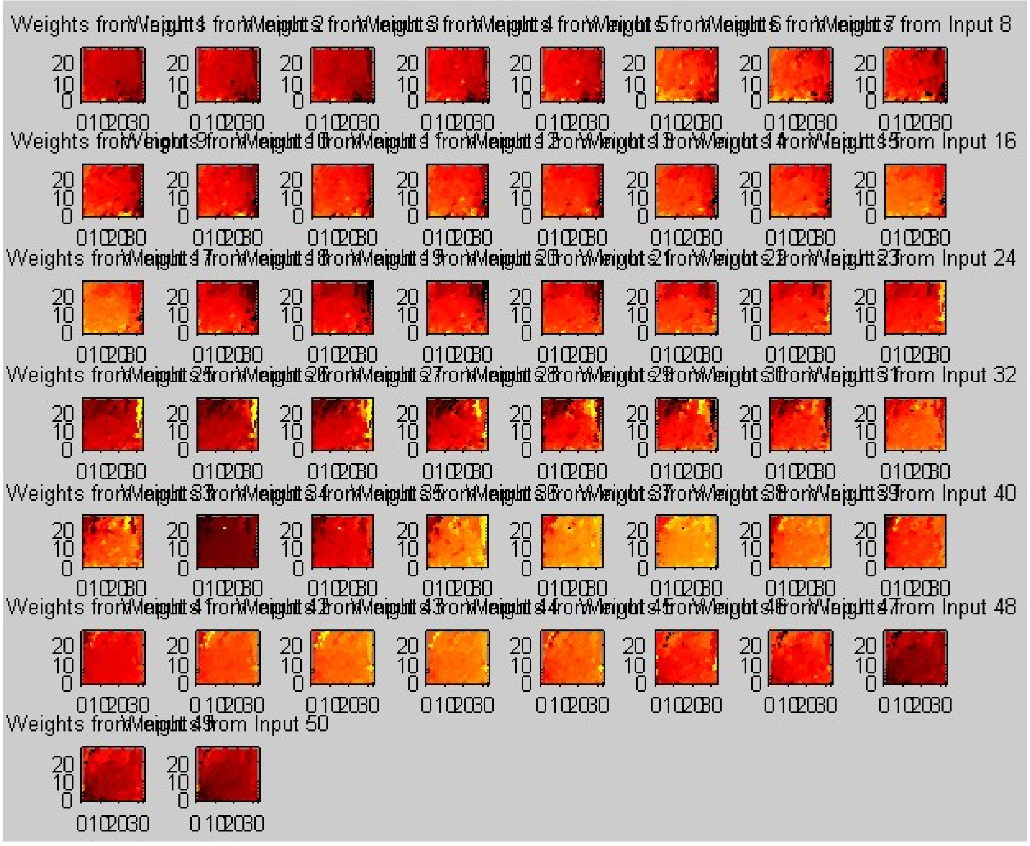
รูปที่ 3.51 การจัดรูปแบบโครงข่ายแบบหกเหลี่ยมขนาด 1,024 ค่า



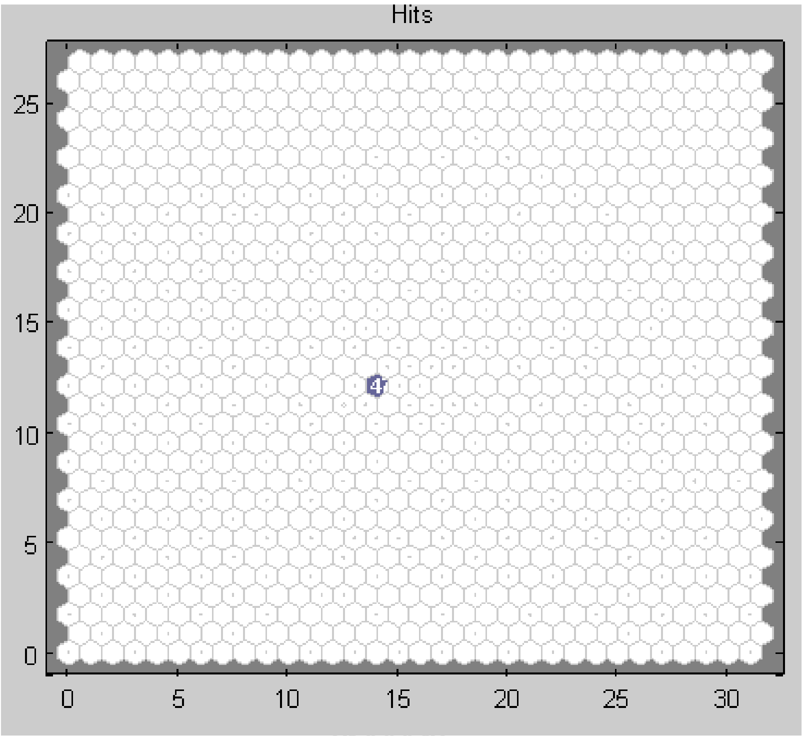
รูปที่ 3.52 การเชื่อมต่อของปมประสาทข้างเคียงขนาด 1,024 ค่า



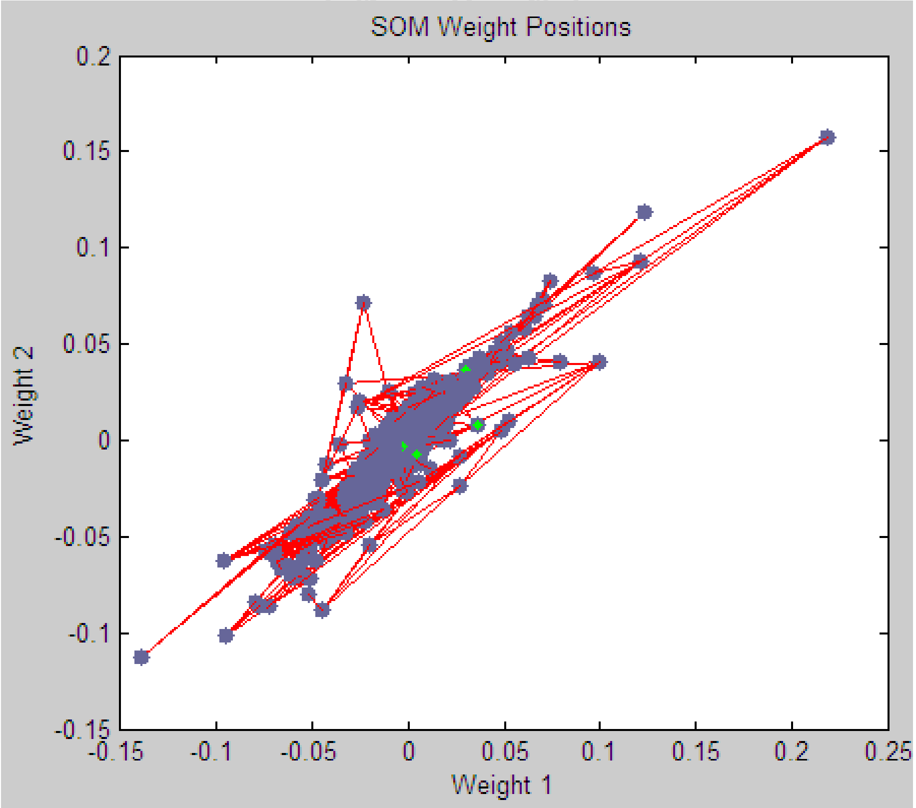
รูปที่ 3.53 ระยะทางการเชื่อมโยงระหว่างปมประสาทจำนวน 1,024 ค่า



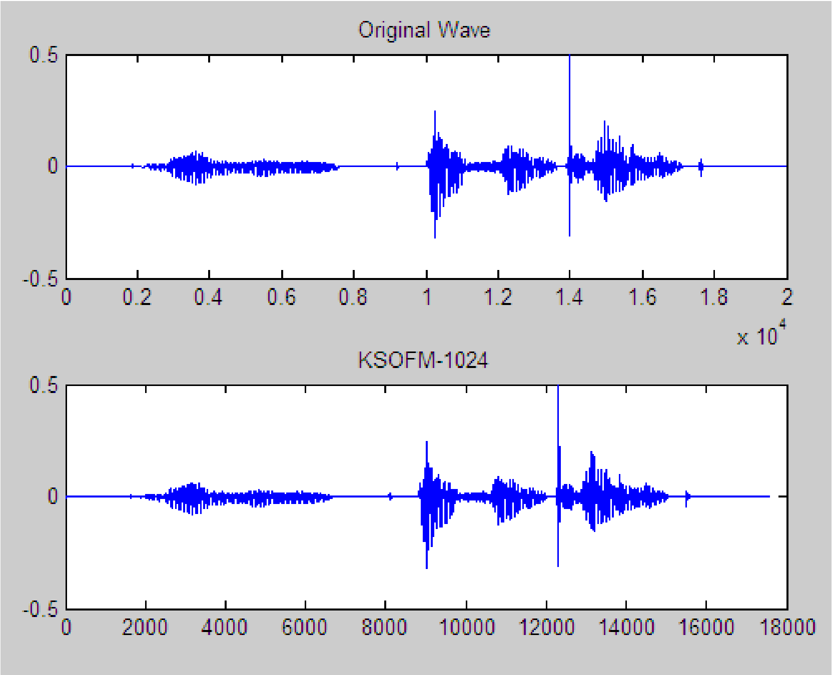
รูปที่ 3.54 น้ำหนักแต่ละปมประสาทจำนวน 1,024 ค่า



รูปที่ 3.55 การกระจายตัวของข้อมูลเวกเตอร์ขนาด 1,024 ค่า



รูปที่ 3.56 น้ำหนักตำแหน่งของปมประสาทขนาด 1,024 ค่า



รูปที่ 3.57 ข้อมูลตัวอย่างของสัญญาณเสียงต้นฉบับเทียบกับสัญญาณเสียงบีบอัด

3.4 การตรวจสอบคุณลักษณะทางสัญญาณเสียงพูด

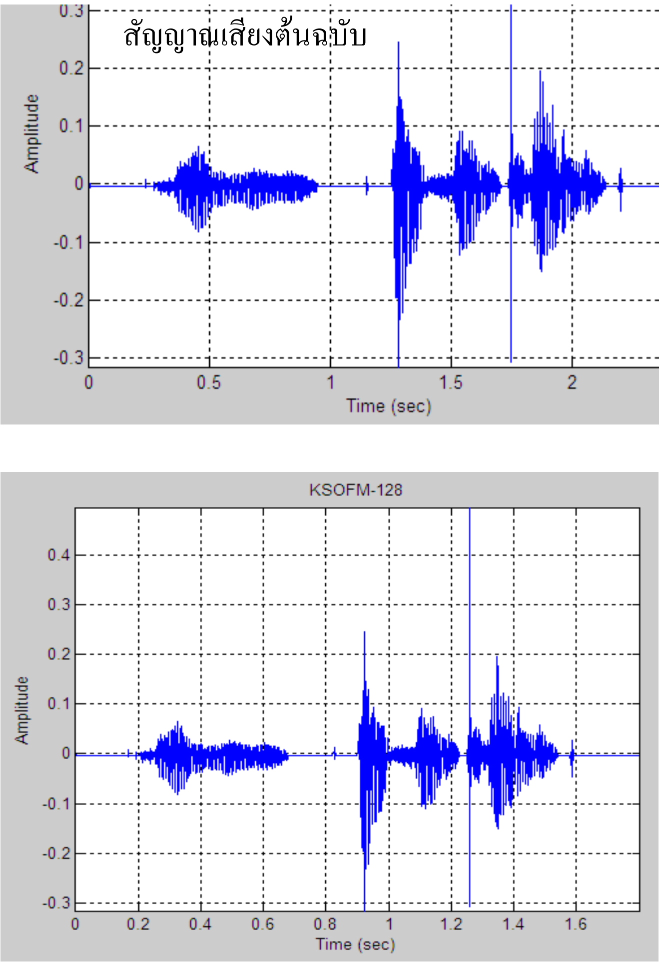
การตรวจสอบคุณภาพของเสียงจะพิจารณาจากการแสดงผลเชิงเวลา (Time Domain) และเชิง-ความถี่ (Frequency Domain) ของสัญญาณเสียงพูดต้นฉบับกับสัญญาณเสียงพพูดที่บีบอัด ดังต่อไปนี้

3.4.1 การตรวจสอบคุณภาพเสียงพูดเชิงเวลา

การตรวจสอบคุณลักษณะเชิงเวลาของสัญญาณเสียงพูดต้นฉบับและสัญญาณเสียงพูดที่บีบอัด จะพิจารณาจากผลตอบสนองเชิงเวลาด้วยกราฟแสดงผลเชิงเวลา พิจารณาที่แกน x เป็นค่าช่วงเวลาสัญญาณ และแกน y เป็นค่าขนาดของสัญญาณ โดยที่การแสดงผลด้วยโปรแกรม Matlab เป็นค่าจำนวนตัวอย่างของสัญญาณเสียงพูดที่เกิดขึ้นพบว่าจำนวนตัวอย่างที่เกิดขึ้นมีความสัมพันธ์กับคาบเวลาการสุ่มตัวอย่างจึงเขียนโปรแกรมที่จะเปลี่ยนค่าจำนวนตัวอย่างที่เกิดขึ้นเป็นค่าคาบเวลาในเกิดสัญญาณเสียงพูด

```
Command Window
>> numSamp = size(slice1)
numSamp = numSamp(1,1)*numSamp(1,2)
Fs = 8000;
t  = 0:1/Fs:numSamp;
idx = 1:numSamp;
plot(t(idx),slice1(idx)); grid; ylabel('Amplitude'); xlabel('Time (sec)');
axis tight;
```

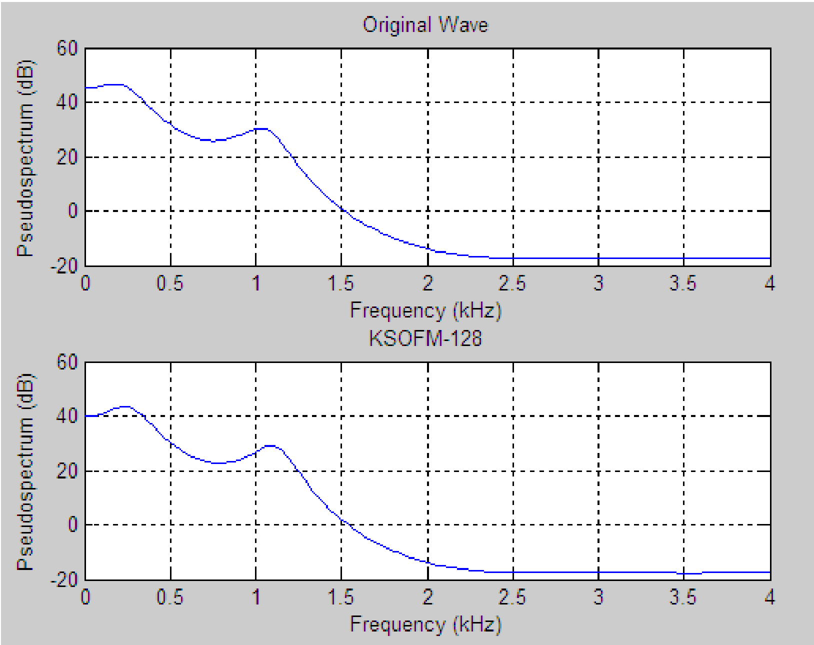
รูปที่ 3.58 คำสั่งในการแสดงผลในเชิงเวลา



รูปที่ 3.59 เปรียบเทียบเชิงเวลาสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วย KSOFM

3.4.2 การตรวจสอบคุณภาพเสียงพูดเชิงความถี่

การตรวจสอบคุณลักษณะทางความถี่ของสัญญาณเสียงพูดต้นฉบับและสัญญาณเสียงพูดที่บีบอัดจะพิจารณาจากผลตอบสนองความถี่ทางสัญญาณแสดงในลักษณะของกราฟในเชิงความถี่ (Frequency domain) โดยจะเขียนโปรแกรม Matlab ที่ประกอบด้วยคำสั่ง pmusic ใช้สร้างสเปคตรัมสัญญาณด้วยโครงสร้างแบบ MUSIC และ คำสั่ง dspdata.pseudospectrum ใช้สร้างองค์ประกอบของสัญญาณข้อมูลที่ประกอบด้วยฮาร์โมนิกของสัญญาณซายน์ โปรแกรมการตรวจผลตอบสนองความถี่จะใช้ในการหาค่าผลตอบสนองของสัมประสิทธิ์การสกัดค่า LPC ที่มีผลต่อการตอบสนองเชิงความถี่ของสัญญาณเสียงต้นฉบับหรือสัญญาณ เสียงที่ผ่านการบีบอัดด้วยโครงข่ายประสาทเทียมจัดการตนเอง



รูปที่ 3.60 ผลตอบสนองความถี่ของสัญญาณเสียงต้นฉบับเปรียบเทียบสัญญาณบีบอัด

บทที่ 4

ผลการวิจัย

ส่วนนี้เป็นการแสดงผลส่วนของข้อมูลทั้งหมดที่ได้ทำการเก็บบันทึกและทำการทดสอบในเรื่องของการสกัดค่าสัมประสิทธิ์ของLPC การจัดกลุ่มเฟรมเสียงเพื่อเปรียบเทียบกลุ่มข้อมูลเฟรมเสียงต้นฉบับกับเฟรมเสียงที่บีบอัด การหาค่าการกระจายตัวข้อมูลสัญญาณเสียงที่บีบอัดในลำดับชั้นของโครงข่ายประสาทเทียม รวมไปถึงข้อมูลที่ใช้ในการประเมินคุณภาพเสียงบีบอัด

4.1 ผลการทดสอบการสร้างโค้ดบุ๊ค

สัญญาณเสียงพูดต้นฉบับที่ใช้ในการทดสอบมีอัตราการสุ่มตัวอย่าง 8 KHz มีขนาดตัวอย่างจำนวน 80,000 ค่าโดยจะนำมาแบ่งเป็น 4 ส่วน แต่ละส่วนมีค่า 20,000 ตัวอย่างดังนี้

ตารางที่ 4.1 การแบ่งเฟรมของเสียง

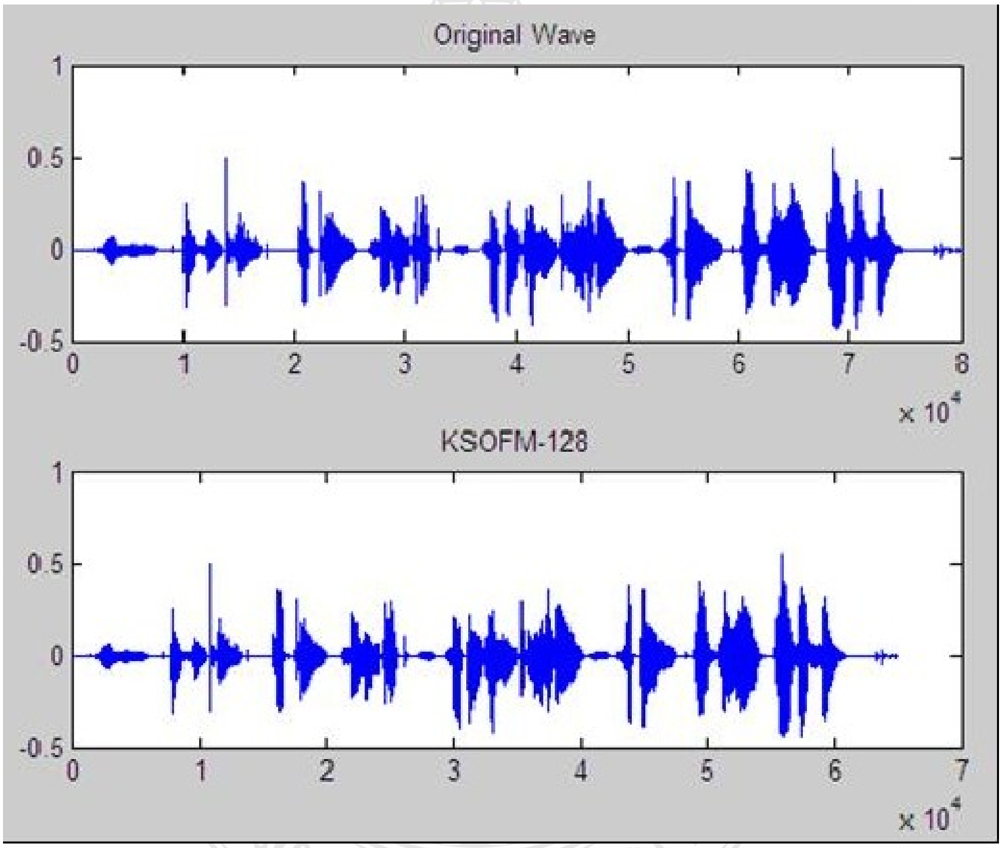
ส่วนที่	จำนวนตัวอย่าง	คาบเวลา
1	1 – 20,000	2.5 second
2	20,001 – 40,000	2.5 second
3	40,001 – 60,000	2.5 second
4	60,001 – 80,000	2.5 second
รวม	1 – 80,000	10 second

4.1.1 ผลการทดสอบโค้ดบุ๊คขนาด 128

การแบ่งสร้างโค้ดบุ๊คขนาด 128 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างของตัวอย่างเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาทเทียมแบบจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.2 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 128

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	ค่าต้นฉบับ	ค่าบีบอัด	
1	20,000	15,600	22%
2	20,000	15,200	24%
3	20,000	16,000	20%
4	20,000	16,400	18%
รวม	80,000	63,200	21%



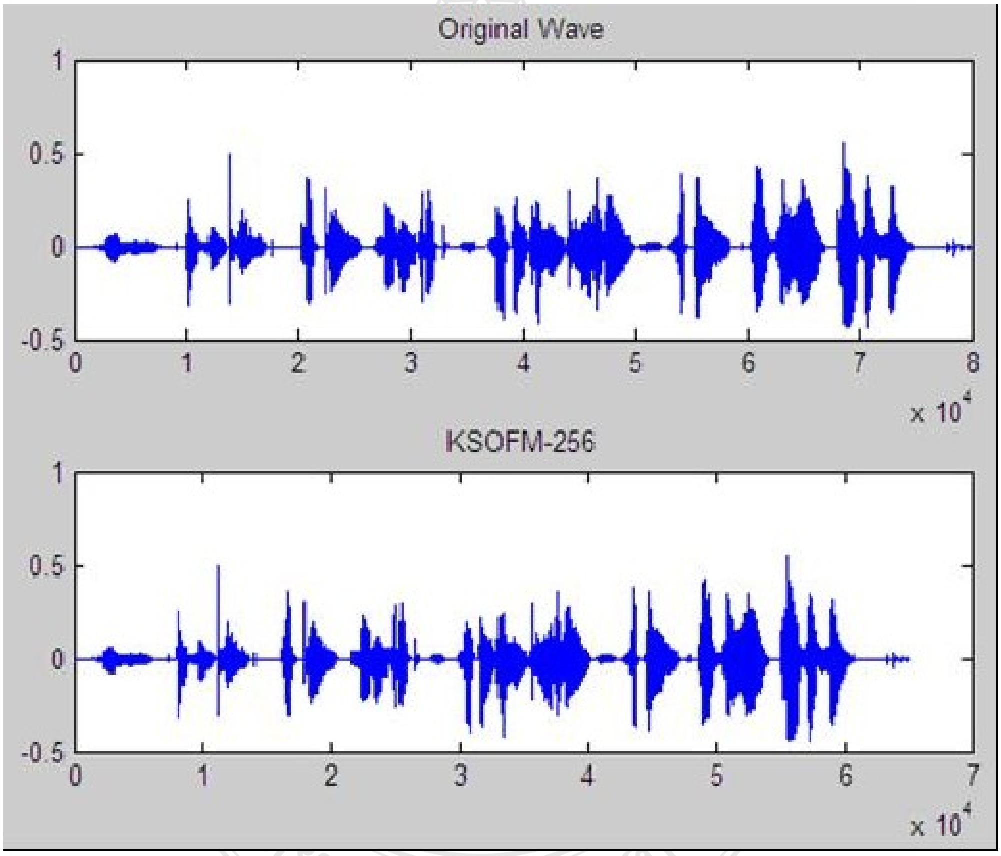
รูปที่ 4.1 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 128

4.1.2 ผลการทดสอบโค้ดบุ๊คขนาด 256

การแบ่งสร้างโค้ดบุ๊คขนาด 128 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างของตัวอย่างเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาทเทียมจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.3 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 256

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	ค่าต้นฉบับ	ค่าบีบอัด	
1	20,000	15,200	24%
2	20,000	15,600	22%
3	20,000	16,000	20%
4	20,000	15,600	22%
รวม	80,000	62,400	22%



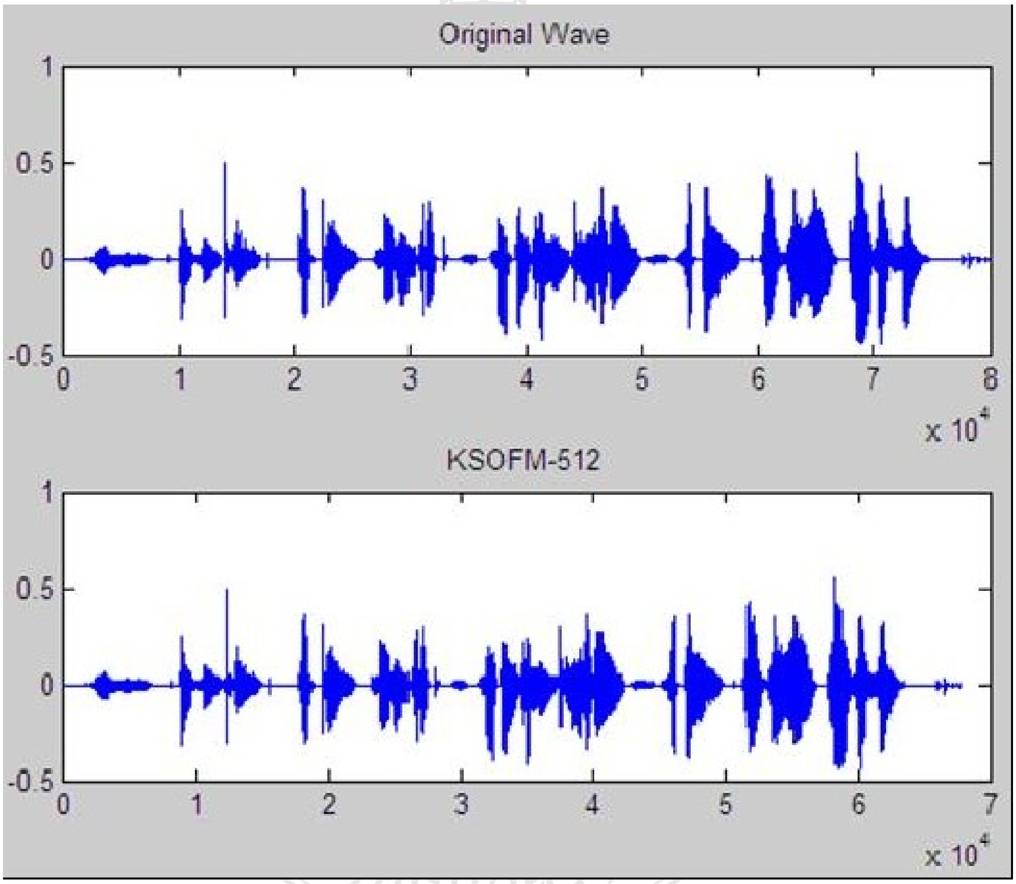
รูปที่ 4.2 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 256

4.1.3 ผลการทดสอบโค้ดบุ๊คขนาด 512

การแบ่งสร้างโค้ดบุ๊คขนาด 512 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างของตัวอย่างเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาทเทียมแบบจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.4 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 512

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	ค่าต้นฉบับ	ค่าบีบอัด	
1	20,000	18,000	10%
2	20,000	17,200	14%
3	20,000	18,000	10%
4	20,000	16,400	18%
รวม	80,000	69,600	13%



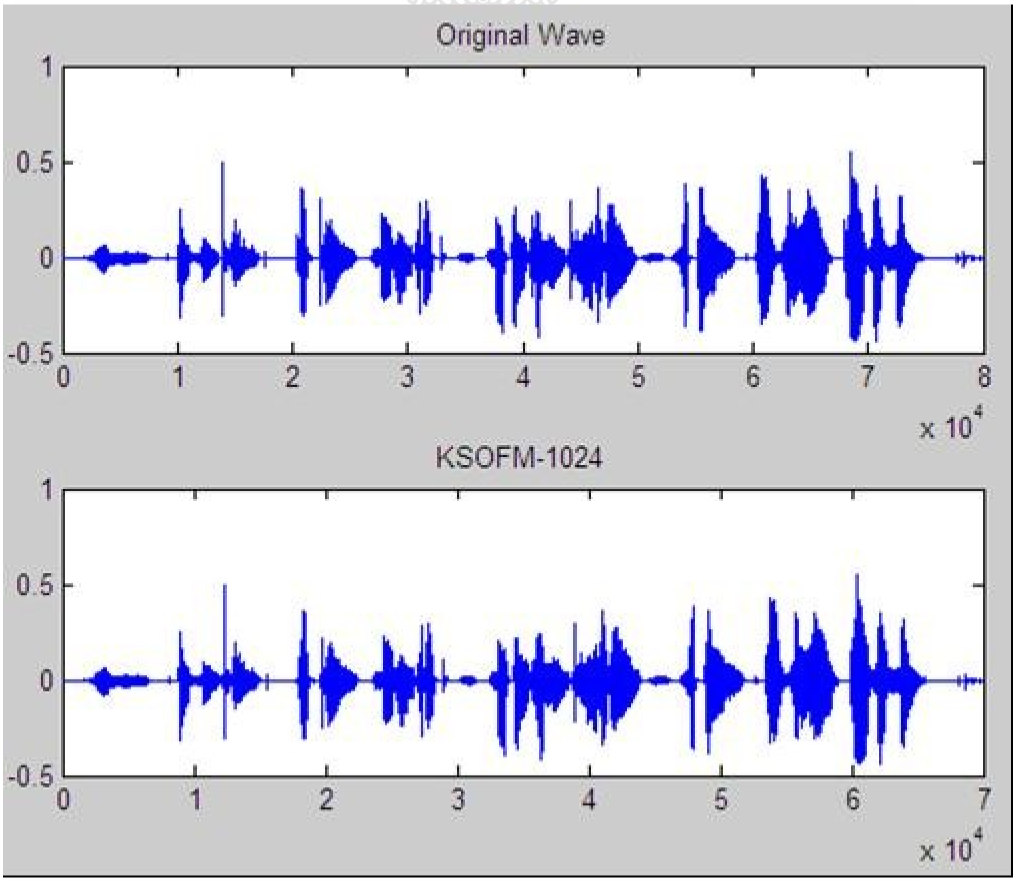
รูปที่ 4.3 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 512

4.1.4 ผลการทดสอบโค้ดบุ๊คขนาด 1,024

การแบ่งสร้างโค้ดบุ๊คขนาด 1,024 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างของตัวอย่างเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาทเทียมแบบจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.5 จำนวนตัวอย่างของการบีบอัดที่ใช้โค้ดบุ๊ค 1,024

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	ค่าต้นฉบับ	ค่าบีบอัด	
1	20,000	17,200	14%
2	20,000	17,600	12%
3	20,000	18,000	10%
4	20,000	16,800	16%
รวม	80,000	69,600	13%



รูปที่ 4.4 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดค่าโค้ดบุ๊ค 1,024

ตารางที่ 4.6 ผลการบีบอัดเสียงพูดเชิงจำนวนตามขนาดโค้ดบุ๊ค

สัญญาณเสียง	จำนวนข้อมูล	% การบีบอัด
ต้นฉบับ	80,000	-
128	63,200	21%
256	62,400	22%
512	69,600	13%
1,024	69,600	13%

จากผลการทดลองการบีบอัดเสียงพูดเชิงจำนวนพบว่า โค้ดบุ๊คขนาด 256 สามารถบีบอัดข้อมูลได้มากที่สุดคือ 22% รองลงมาได้แก่โค้ดบุ๊คขนาด 128 คือ 21% และโค้ดบุ๊คขนาด 512 มีผลการบีบอัดได้เท่ากับโค้ดบุ๊คขนาด 1,024 คือ 13%

4.2 ผลการทดสอบคุณภาพเสียงเชิงเวลา

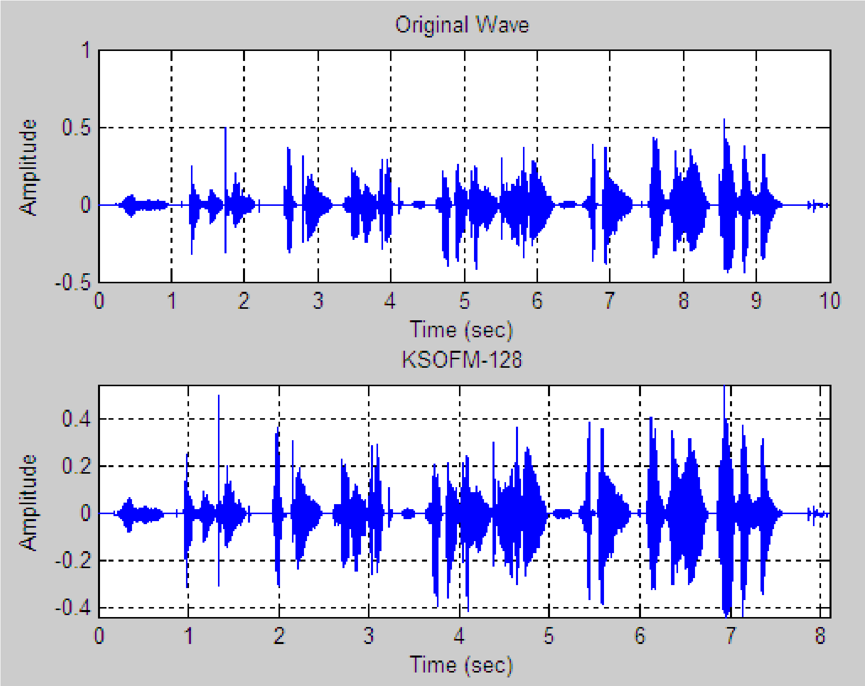
สัญญาณเสียงพูดต้นฉบับที่ใช้ในการทดสอบมีอัตราการสุ่มตัวอย่าง 8 KHz มีคาบเวลาจำนวน 10 วินาที โดยจะนำมาแบ่งเป็น 4 ส่วน แต่ละส่วนมีค่าส่วนละ 2.5 วินาที ดังนี้

4.2.1 ผลการทดสอบคุณภาพเสียงเชิงเวลาของโค้ดบุ๊คขนาด 128

การแบ่งสร้างโค้ดบุ๊คขนาด 128 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างเวลาของเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาท-เทียมแบบจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.7 ผลการบีบอัดเชิงเวลาที่ใช้โค้ดบุ๊คขนาด 128

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	เวลาต้นฉบับ	เวลาบีบอัด	
1	2.5 second	1.95 second	22%
2	2.5 second	2.05 second	18%
3	2.5 second	2.00 second	20%
4	2.5 second	1.95 second	22%
รวม	10 second	7.95 second	20.5%



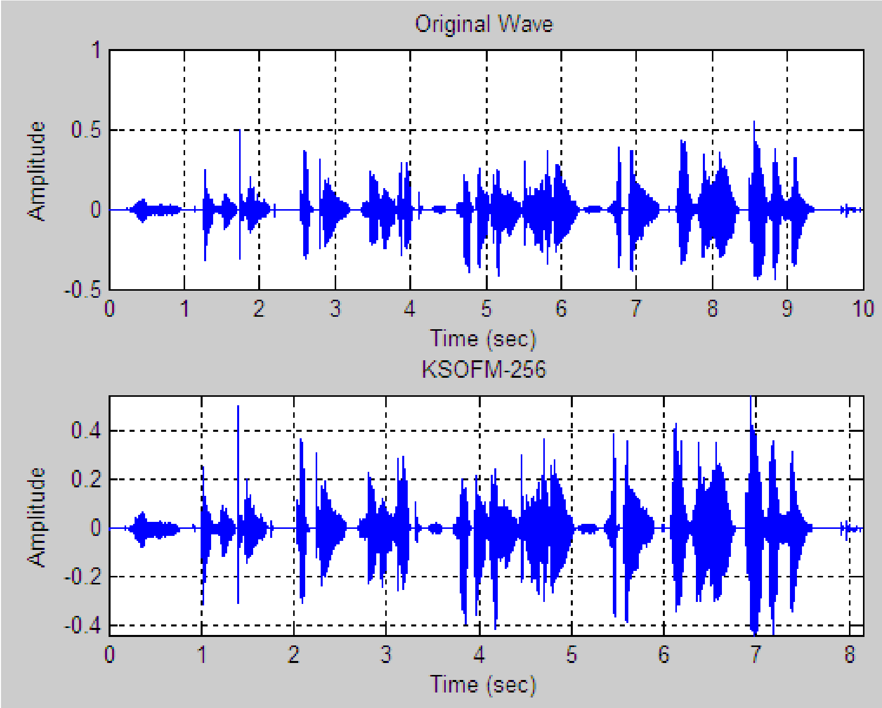
รูปที่ 4.5 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 128

4.2.2 ผลการทดสอบคุณภาพเสียงเชิงเวลาสำหรับโค้ดบุ๊คขนาด 256

การแบ่งสร้างโค้ดบุ๊คขนาด 256 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างเวลาของเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาท-เทียมจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.8 ผลการบีบอัดเชิงเวลาที่ใช้โค้ดบุ๊คขนาด 256

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	เวลาต้นฉบับ	เวลาบีบอัด	
1	2.5 second	1.9 second	24%
2	2.5 second	1.95 second	22%
3	2.5 second	2.0 second	20%
4	2.5 second	1.95 second	22%
รวม	10 second	7.8 second	22%



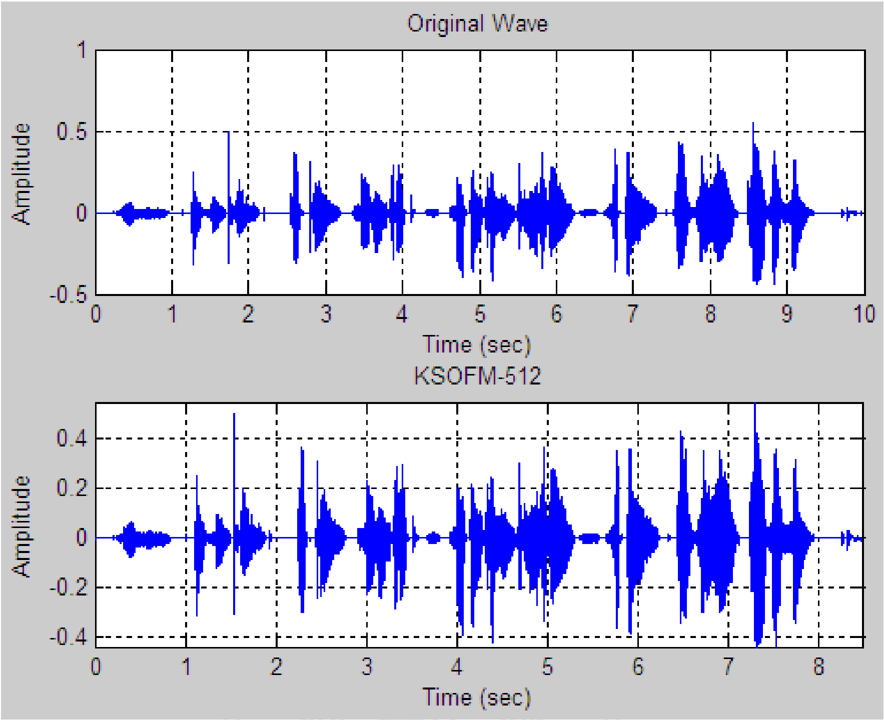
รูปที่ 4.6 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 256

4.2.3 ผลการทดสอบคุณภาพเสียงเชิงเวลาของโค้ดบุ๊คขนาด 512

การแบ่งสร้างโค้ดบุ๊คขนาด 512 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างเวลาของเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาท-เทียมแบบจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.9 ผลการบีบอัดเชิงเวลาที่ใช้โค้ดบุ๊คขนาด 512

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	เวลาต้นฉบับ	เวลาบีบอัด	
1	2.5 second	2.25 second	10%
2	2.5 second	2.15 second	14%
3	2.5 second	2.25 second	10%
4	2.5 second	2.05 second	18%
รวม	10 second	8.70 second	13%



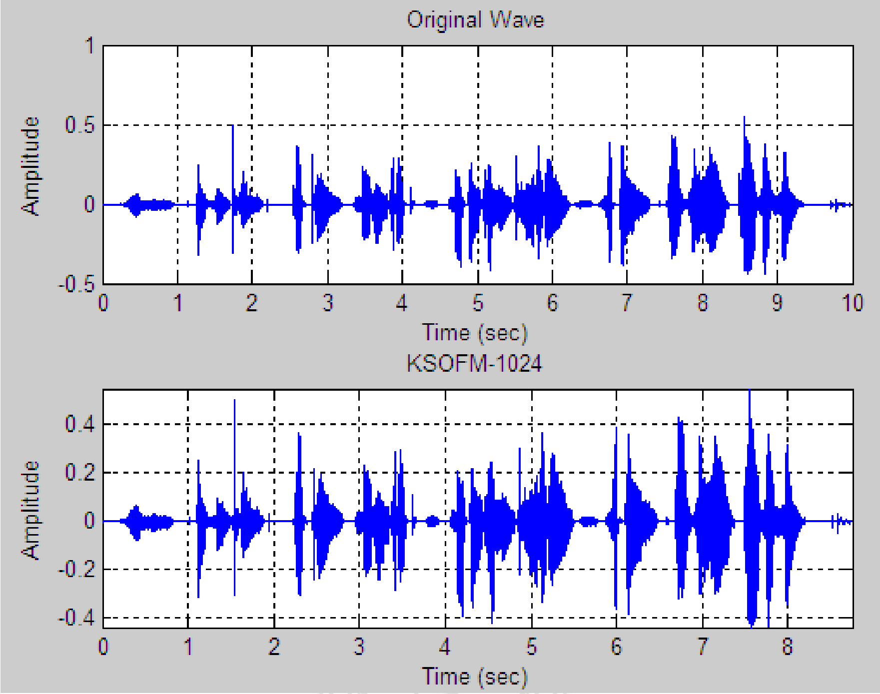
รูปที่ 4.7 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 512

4.2.4 ผลการทดสอบคุณภาพเสียงเชิงเวลาของโค้ดบุ๊คขนาด 1,024

การแบ่งสร้างโค้ดบุ๊คขนาด1,024 เพื่อจะตรวจสอบค่าผลการบีบอัดในแต่ละส่วนของเฟรมเสียงต้นฉบับและผลของความแตกต่างเวลาของเสียงที่เกิดขึ้นจากการบีบอัดด้วยโครงข่ายประสาทเทียมจัดการตนเอง มีผลปรากฏดังนี้

ตารางที่ 4.10 ผลการบีบอัดเชิงเวลาที่ใช้โค้ดบุ๊คขนาด 1,024

ส่วนที่	ค่าจำนวนตัวอย่าง		% การบีบอัด
	เวลาต้นฉบับ	เวลาบีบอัด	
1	2.5 second	2.15 second	14%
2	2.5 second	2.20 second	12%
3	2.5 second	2.25 second	10%
4	2.5 second	2.10 second	16%
รวม	10 second	8.70 second	13%



รูปที่ 4.8 เปรียบเทียบสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัดด้วยโค้ดบุ๊คขนาด 1,024

ตารางที่ 4.11 ผลการบีบอัดเสียงพูดเชิงเวลาตามขนาดโค้ดบุ๊ค

สัญญาณเสียง	เวลาที่ใช้	% การบีบอัด
ต้นฉบับ	10 วินาที	-
128	7.9 วินาที	20.5%
256	7.8 วินาที	22%
512	8.7 วินาที	13%
1,024	8.7 วินาที	13%

จากผลการทดลองการบีบอัดเสียงพูดเชิงเวลาพบว่า โค้ดบุ๊คขนาด 256 สามารถบีบอัดข้อมูลได้มากที่สุดคือ 22% รองลงมาได้แก่โค้ดบุ๊คขนาด 128 คือ 20.5% และโค้ดบุ๊คขนาด 512 มีผลการบีบอัดได้เท่ากับโค้ดบุ๊คขนาด 1,024 คือ 13%

จึงสรุปผลการบีบอัดเชิงจำนวนและเวลาได้ว่า การบีบอัดข้อมูลเสียงพูดจะแปรผกผันกับขนาดโค้ดบุ๊ค นั่นคือโค้ดบุ๊คที่มีขนาดใหญ่หรือมีปมประสาทมากจะบีบอัดข้อมูลได้น้อย เนื่องจากโค้ดบุ๊คขนาดใหญ่จะมีปมประสาทมากทำให้สามารถเก็บค่าข้อมูลหรือรักษาคุณลักษณะเสียงพูด

ต้นฉบับไว้ได้มาก ความสามารถในการบีบอัดจึงน้อย แต่จะมีคุณภาพเสียงดีหรือใกล้เคียงกับเสียงต้นฉบับ

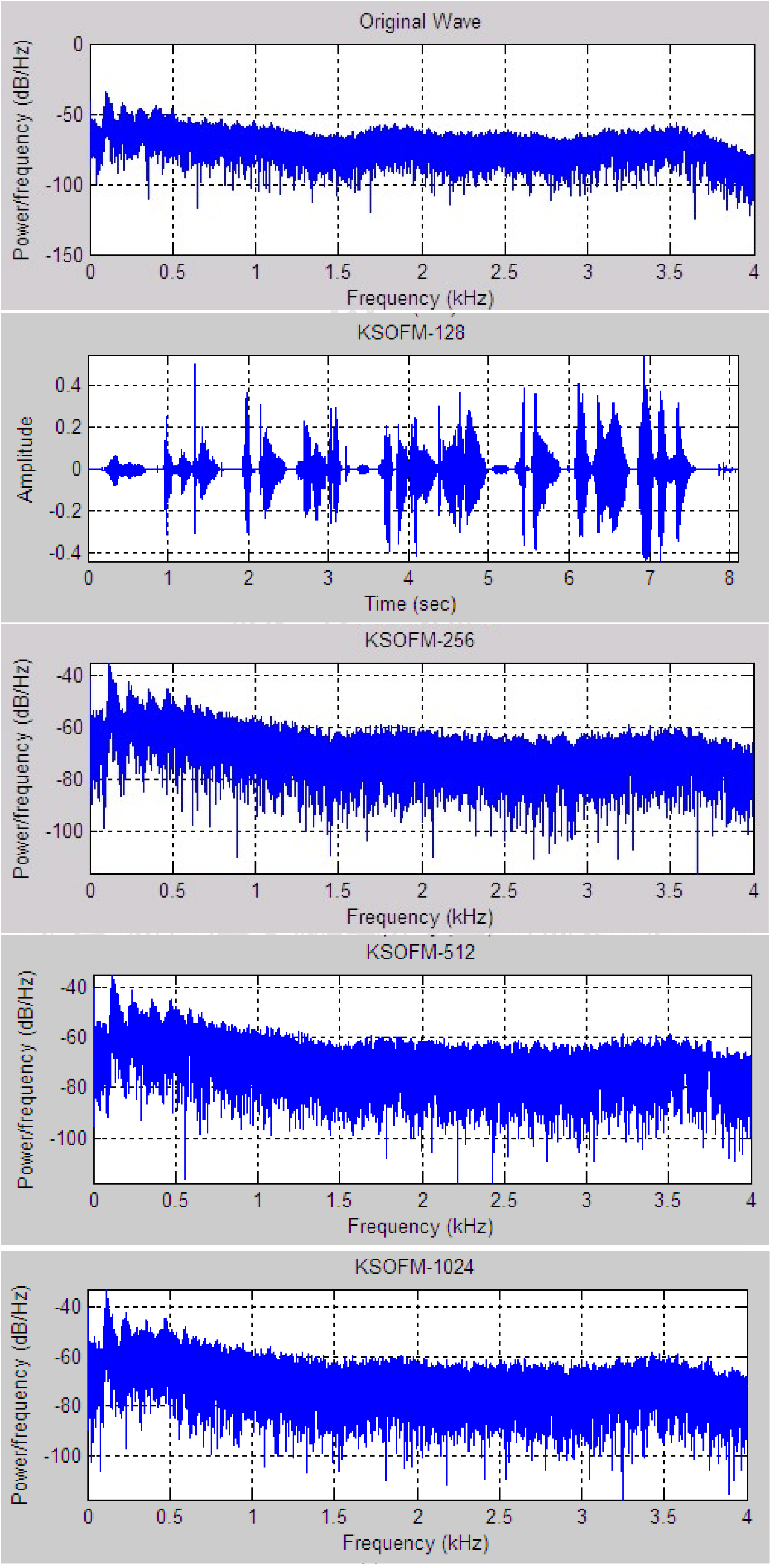
4.3 ผลการทดสอบคุณภาพเสียงเชิงความถี่

การทดสอบผลตอบสนองความถี่สัญญาณเสียงพูดต้นฉบับกับสัญญาณเสียงพูดที่บีบอัดด้วยโครงข่ายประสาทเทียมจัดการตนเองโดยในการทดสอบจะจัดแบ่งผลการทดสอบเป็น 4 ค่า ซึ่งจะประกอบด้วยสัญญาณเสียงพูดต้นฉบับและสัญญาณเสียงพูดบีบอัดประกอบด้วยค่าของโค้ดบุ๊คที่ใช้ในการบีบอัดขนาด 128 256 512 และ 1,024 ปรากฏผลดังนี้

ตารางที่ 4.12 ผลทดสอบสัญญาณเสียงพูดต้นฉบับกับสัญญาณเสียงพูดที่บีบอัดเชิงความถี่

สัญญาณเสียง	ผลตอบสนองความถี่								
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	kHz
ต้นฉบับ	-50	-57	-67	-61	-64	-67	-60	-83	dB
บีบอัดด้วยโค้ดบุ๊ค 128	-45	-57	-61	-60	-62	-64	-61	-68	dB
บีบอัดด้วยโค้ดบุ๊ค 256	-47	-56	-60	-59	-62	-63	-61	-68	dB
บีบอัดด้วยโค้ดบุ๊ค 512	-49	-57	-64	-68	-67	-67	-64	-68	dB
บีบอัดด้วยโค้ดบุ๊ค 1,024	-50	-60	-61	-64	-64	-63	-63	-68	dB

จากผลการทดลองการบีบอัดสัญญาณเสียงพูดเชิงความถี่ตั้งแต่ 0.5-4.0 kHz พบว่าผลตอบสนองทางความถี่ของสัญญาณเสียงพูดที่ผ่านการบีบอัดโดยโค้ดบุ๊คขนาด 256 มีผลตอบสนองความถี่ดีที่สุดเกือบทุกย่านความถี่ ยกเว้นที่ความถี่ 0.5 kHz โค้ดบุ๊คขนาด 128 จะตอบสนองความถี่ได้ดีกว่าที่สุด และที่ความถี่ 4 kHz โค้ดบุ๊คทั้ง 4 ขนาดมีผลตอบสนองทางความถี่เท่ากัน จึงสรุปผลการบีบอัดเชิงความถี่ได้ว่าขนาดของโค้ดบุ๊คที่ต่างกันจะมีผลตอบสนองทางความถี่ต่อการบีบอัดสัญญาณเสียงพูดที่ความถี่สูงๆ ได้เท่ากันหรือใกล้เคียงกันมาก


Original Wave
Power/frequency (dB/Hz)
0
-50
-100
-150
0 0.5 1 1.5 2 2.5 3 3.5 4
Frequency (kHz)
KSOFM-128
Amplitude
0.4
0.2
0
-0.2
-0.4
0 1 2 3 4 5 6 7 8
Time (sec)
KSOFM-256
Power/frequency (dB/Hz)
-40
-60
-80
-100
0 0.5 1 1.5 2 2.5 3 3.5 4
Frequency (kHz)
KSOFM-512
Power/frequency (dB/Hz)
-40
-60
-80
-100
0 0.5 1 1.5 2 2.5 3 3.5 4
Frequency (kHz)
KSOFM-1024
Power/frequency (dB/Hz)
-40
-60
-80
-100
0 0.5 1 1.5 2 2.5 3 3.5 4
Frequency (kHz)

รูปที่ 4.9 เปรียบเทียบผลตอบสนองเชิงความถี่ของสัญญาณเสียงต้นฉบับกับสัญญาณบีบอัด

4.4 ผลการทดสอบคุณภาพของเสียงโดยใช้ SNR

การหาคุณภาพของสัญญาณเสียงพูดโดยวัดค่า PSNR (Peak Signal to Noise Ratio) เป็นการหาค่าอัตราส่วนสัญญาณเสียงพูดต่อสัญญาณรบกวนโดยทำการวัดเปรียบเทียบค่าระหว่างสัญญาณเสียงต้นฉบับกับสัญญาณเสียงบีบอัด ค่าของ PSNR ที่มีค่าสูงจะบ่งบอกถึงคุณภาพการบีบอัดที่ดี ค่าของ PSNR จะเป็นตามความสัมพันธ์ของสมการดังนี้

$$PSNR = 10\log_{10}\left\{\frac{\sum_{n=0}^{M-1} s^2(n)}{\sum_{n=0}^{M-1}(s(n)-s'(n))^2}\right\}$$

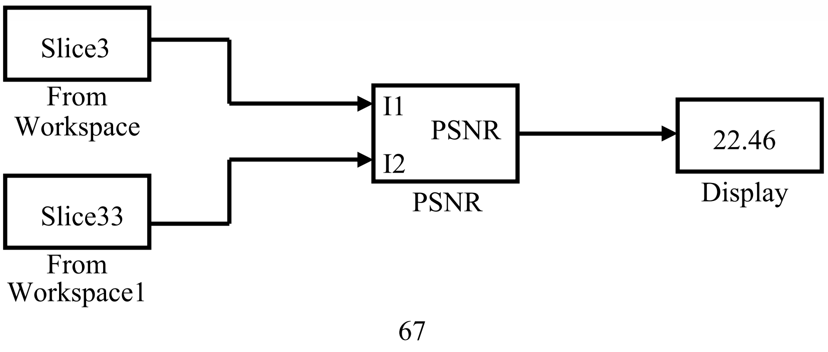
เมื่อ S(n) คือ สัญญาณเสียงพูดที่ไม่ได้ผ่านการบีบอัด

S'(n) คือ สัญญาณเสียงพูดที่ผ่านการบีบอัด

ในงานวิจัยนี้แบ่งสัญญาณเสียงพูดต้นฉบับขนาด 80,000 ตัวอย่างเป็น 4 ส่วนโดยแต่ละส่วนมีค่า 20,000 ตัวอย่างโดยนำสัญญาณเสียงพูดต้นฉบับผ่านเข้าโครงข่ายประสาทเทียมแบบจัดการตนเองเพื่อจัดแบ่งปมประสาทขาออกดังนี้

1) ปมประสาทขาออกขนาด 128 ค่าลำดับชั้น 32x4
2) ปมประสาทขาออกขนาด 256 ค่าลำดับชั้น 32x8
3) ปมประสาทขาออกขนาด 512 ค่าลำดับชั้น 32x16
4) ปมประสาทขาออกขนาด 1,024 ค่าลำดับชั้น 32x32

โดยผลการวัดค่า PSNR จะใช้ Toolbox ใน Simulink เป็นเครื่องมือวัดค่าโดยนำค่าสัญญาณเสียงพูดต้นฉบับที่เก็บในตัวแปร est_x และ ค่าสัญญาณเสียงพูดบีบอัดที่เก็บในตัวแปร Sumsound ที่อยู่ใน workspace มาใช้เป็นส่วนของอินพุตของไดอะแกรมในบล็อก PSNR ร่วมโปรแกรมปรับขนาดใน workspace



รูปที่ 4.10 การวัดค่า PSNR ด้วย Simulink

```
Sizeslice3 = size(slice3)
Sizelayer33 = size(layer33)
Sizeslice3 = Sizeslice3(1,1)*Sizeslice3(1,2)
Sizelayer33 = Sizelayer33(1,1)*Sizelayer33(1,2)
for(i=1:1:(Sizeslice3-Sizelayer33))
    layer33(1,(Sizelayer33+i)) = 0;
end
```

รูปที่ 4.11 โปรแกรมปรับค่าขนาดใน workspace สำหรับวัดค่า PSNR

ตารางที่ 4.13 ค่า PSNR

ค่าจำนวนตัวอย่าง	ค่า PSNR ขนาด 128	ค่า PSNR ขนาด 256	ค่า PSNR ขนาด 512	ค่า PSNR ขนาด 1,024
ส่วนที่ 1 : 1-20,000	30.18	30.42	29.86	29.96
ส่วนที่ 2 : 20,001-40,000	23.28	23.2	23.07	22.91
ส่วนที่ 3 : 40,001-60,000	22.02	22.02	21.84	21.99
ส่วนที่ 4 : 60,001-80,000	19.47	19.57	19.36	19.36
ค่าผลรวม 80,000	22.36	22.46	22.36	22.17

จากผลการทดลองหาคุณภาพของสัญญาณเสียงพูดโดยวัดค่า PSNR พบว่า ค่า PSNR ของโค้ดบุ๊คขนาด 256 มีค่ามากที่สุดคือ 22.46 รองลงมาได้แก่โค้ดบุ๊คขนาด 128 ซึ่งมีค่า PSNR เท่ากันกับโค้ดบุ๊คขนาด 512 คือ 20.36 และโค้ดบุ๊คขนาด 1,024 มีค่า PSNR น้อยที่สุดคือ 22.17

4.5 ผลการประเมินคุณภาพเสียงพูด (MOS)

การประเมินคุณภาพเสียงพูดจะใช้การรับรู้และความรู้สึกของผู้ฟังเป็นเกณฑ์ในการตัดสิน โดยการฟังเสียงพูดที่ผ่านการบีบอัดแล้วเปรียบเทียบความชัดเจนกับเสียงพูดต้นฉบับแล้วทำการประเมินคุณภาพด้วยวิธีการของ MOS

ตารางที่ 4.14 การประเมินคุณภาพเสียงจากการรับฟัง

กลุ่ม	ค่าระดับการประเมินคุณภาพเสียงจากการรับฟัง (MOS)				
ทดสอบ	ต้นฉบับ	128	256	512	1,024
กลุ่ม 1	4.82	2.52	3.33	3.53	4.15
กลุ่ม 2	4.76	2.14	2.93	3.36	3.73
กลุ่ม 3	4.74	2.33	3.66	3.91	4.04
กลุ่ม 4	4.88	2.47	3.46	3.74	3.92
เฉลี่ย	4.80	2.36	3.35	3.64	3.96

จากผลการการประเมินคุณภาพเสียงพูดด้วย MOS พบว่าโค้ดบุ๊คขนาด 1,024 มีค่าเฉลี่ย MOS มากที่สุดคือ 3.96 รองลงมาได้แก่โค้ดบุ๊คขนาด 512 โค้ดบุ๊คขนาด 256 และโค้ดบุ๊คขนาด 128 โดยมีค่าเฉลี่ย MOS คือ 3.64, 3.35 และ 2.36 ตามลำดับ

จึงสรุปผลการการประเมินคุณภาพเสียงพูดด้วย MOS ได้ว่า คุณภาพของเสียงบีบอัดจะแปรผันตามขนาดของโค้ดบุ๊ค นั่นคือโค้ดบุ๊คที่มีขนาดใหญ่หรือมีปมประสาทมากจะสามารถเก็บค่าข้อมูลหรือรักษาคุณลักษณะเสียงพูดต้นฉบับไว้ได้มาก คุณภาพเสียงจึงใกล้เคียงกับเสียงต้นฉบับ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ในงานวิจัยนี้ ได้พัฒนาเทคนิคควอนไทซ์เซชั่นสำหรับการบีบอัดสัญญาณเสียงพูดภาษาไทย โดยมีพื้นฐานมาจากวิธีการบีบอัดแบบ LPC-10 ซึ่งจะใช้ค่าพารามิเตอร์ LSP ที่ได้มาจากการแปลงค่าของพารามิเตอร์ LPC ไปเป็นข้อมูลส่งเข้าสู่ระบบโครงข่ายประสาทเทียมแบบจัดการตนเอง เพื่อทำการบีบอัดข้อมูลเสียงพูดหรือลดจำนวนบิตเรตให้น้อยลงกว่าเสียงต้นฉบับ แล้วประเมินระดับคุณภาพเสียงที่ผ่านการบีบอัดด้วยวิธีการของ MOS

5.1 สรุปผลการวิจัย

จากการที่ศึกษาและวิจัยในส่วนที่ผ่านมา พบว่าในการสกัดค่าพารามิเตอร์ LPC และ LSP มาประยุกต์ใช้บนโมดูลของต้นฉบับ LPC-10 มีส่วนทำให้ลดจำนวนบิตของสัญญาณเสียงต้นฉบับลงได้ ดังผลการเปรียบเทียบค่าบิตเรตที่ลดลงได้ดีกว่าเดิมในตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบผลการบีบอัดของ LPC-10 แบบเดิมกับแบบใหม่

<table>
<tr><th rowspan="3">parameters</th><th colspan="4">Number of bit per frame</th><th colspan="2">Reduce bit rate</th></tr>
<tr><th colspan="2">Original LPC-10</th><th colspan="2">New LPC-10</th><th rowspan="2">ค่าบิตที่ลดลง</th><th rowspan="2">ค่าบิตที่เพิ่มขึ้น</th></tr>
<tr><th>Voiced</th><th>Unvoiced</th><th>Voiced</th><th>Unvoiced</th></tr>
<tr><td>Pitch</td><td>7</td><td>7</td><td>7</td><td>7</td><td>-</td><td>-</td></tr>
<tr><td>Gain</td><td>5</td><td>5</td><td>5</td><td>5</td><td>-</td><td>-</td></tr>
<tr><td>p_1</td><td>5</td><td>5</td><td rowspan="2">7</td><td rowspan="2">7</td><td rowspan="2">6</td><td rowspan="2">-</td></tr>
<tr><td>p_2</td><td>5</td><td>5</td></tr>
<tr><td>p_3</td><td>5</td><td>5</td><td rowspan="2">7</td><td rowspan="2">7</td><td rowspan="2">6</td><td rowspan="2">-</td></tr>
<tr><td>p_4</td><td>5</td><td>5</td></tr>
<tr><td>p_5</td><td>4</td><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>p_6</td><td>4</td><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>p_7</td><td>4</td><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>p_8</td><td>4</td><td>-</td><td>4</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>p_9</td><td>3</td><td>-</td><td>4</td><td>-</td><td>-</td><td>1</td></tr>
<tr><td>p_{10}</td><td>2</td><td>-</td><td>4</td><td>-</td><td>-</td><td>2</td></tr>
<tr><td>Total</td><td>54</td><td>54</td><td>51</td><td>34</td><td>26</td><td>3</td></tr>
</table>

จากตารางที่ 5.1 แสดงให้เห็นผลการบีบอัดในวิธีการของ LPC-10 แบบเดิมและแบบใหม่เปรียบเทียบกัน จะเห็นได้ว่าการบีบอัดด้วยวิธีของ LPC-10 แบบใหม่สามารถลดบิตเรตลงได้ 5.5-37% ทั้งนี้เนื่องจากค่าพารามิเตอร์ p1-p4 เดิมมีจำนวนบิตรวมกันคือ 20 บิต แต่เมื่อใช้วิธีการบีบอัดในงานวิจัยนี้พบว่าสามารถบิตของพารามิเตอร์ p1-p2 ได้ 6 บิต และลดบิตของพารามิเตอร์ p3-p4 ได้อีก 6 บิต รวมจำนวนบิตที่ลดลงได้ 12 บิต โดยแยกสัญญาณที่สามารถลดได้ของ Voiced คือ 6 บิต และ Unvoiced อีก 6 บิต

ทั้งนี้ การบีบอัดด้วยวิธีของ LPC-10 แบบใหม่พบว่า ณ บริเวณความถี่สูงของสัญญาณเสียงที่ผ่านการบีบอัดจะถูกลดทอนคุณลักษณะเสียงพูดลงไปมากจนทำให้คุณภาพเสียงพูดที่ประเมินด้วยการฟังไม่ชัดเจน จึงได้เพิ่มบิตในส่วนของ Voiced ที่พารามิเตอร์ p9-p10 รวมจำนวน 3 บิต เพื่อใช้ในการเก็บค่าคุณลักษณะของเสียงพูดเพิ่มมากขึ้น ทำให้คุณภาพของเสียงที่บีบอัดชัดเจนขึ้นนั่นเอง

5.2 ข้อเสนอแนะ

จากผลการทดสอบผลตอบสนองความถี่และผลการทดสอบระดับการรับฟัง พบว่า noise ในบริเวณที่มีการสิ้นสุดของเสียงพูดที่เป็นจุดเชื่อมต่อระหว่าง Voiced กับ Unvoiced นั้น หากการบีบอัดมีการเพิ่มจำนวน order ของพารามิเตอร์ LPC มากขึ้น จะเกิดการกระเพื่อมของสัญญาณเสียงพูดบริเวณรอยต่อนั้นมากขึ้นตาม ส่งผลให้จำนวนบิตเพิ่มขึ้นด้วย จึงขอเสนอแนะเพื่อการพัฒนาไว้ดังนี้

1) กรณีที่ต้องการเพิ่ม order ของ LPC ควรต้องใช้วงจรกรองความถี่ Post-Processing ช่วยลดสัญญาณเสียงที่กระเพื่อมบริเวณความถี่สูงให้น้อยลงหรือกรอง noise ทิ้งไปนั่นเอง

2) เพิ่มจำนวนบิตเพื่อชดเชยคุณลักษณะเสียงพูดหรือเพื่อสามารถเก็บคุณลักษณะเสียงพูดไว้ได้มากขึ้น แต่บิตเรตก็จะลดลงได้น้อยลงเช่นกัน

เอกสารอ้างอิง

[1] W.B. Kleijn and K.K.Paliwal, Speech Coding and Synthesis, Elsevier Science B.V., 1995.

[2] W.T.K Wong, J. Joe, K. Joe, L. Joe, and M. Joe, “Low rate speech coding for telecommunications,” BT Technol. J., 1996. 14(1): p. 28 – 43.

[3] J. Srinonchat, “New Technique to Reduce Bit Rate of LPC-10 Speech Coder” TENCON, 2006. 2006 IEEE Region 10, Nov. 2006, p. 1-4

[4] J. Srinonchat et al., “Address Vector Quantisation applied to Speech Coding,” IEEE International Symposium on Signal Processing and Information Technology, 2003, p.745–748.

[5] M. Hasegawa-Johnson, Line spectral frequencies are poles and zeros of the glottal driving-point impedance of a discrete matched-impedance vocal tract model, J. Acoust. Soc. Amer. 108 (July 2000) 457-460.

[6] L.R. Rabiner and R.W. Schafer, “Digital Processing of Speech Signal”, Prentice Hall, 1979.

[7] คณะนักวิจัยและพัฒนาระบบระบุผู้พูดสำหรับภาษาไทย, “การระบุผู้พูดด้วย LPC และ DTW สำหรับภาษาไทย”, เอกสารประกอบการประชุมวิชาการ NECTEC, 30 มีนาคม – 1 เมษายน 2542. หน้า 238-246.

[8] จักรกฤษ อ่อนชื่นจิตร, "การวิเคราะห์แนวทางเดินเสียงพูดในรูปแบบของสัมประสิทธิ์คู่เส้นสเปกตรัมร่วมกับ Double Clustering", วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2551.

[9] J.R. Deller, J.G. Proakis and J.H. Hansen, “Discrete-Time Processing of Speech Signals”, 2000.

[10] O. Wiriyanuruknakorn and J. Srinonchat, "A Finite State Vector Quantizer for New Bit Rate Speech Compression" IEEE International on Signal Processing, Communications and Networking 2008, ICSCN '08, Jan. 2008, p.255-259.

[11] MATLAB Manual and MATLAB help
Available : http://www.mathworks.com/help/toolbox/nnet/, 2009

[12] ขนิษฐา รบไพรี, ขวัญนภา พิมพ์ชารีย์ และโอฬาริก สุรินต๊ะ, “การรู้จำตัวอักษรลายมือเขียนภาษาไทยด้วยอัลกอริทึมการแบ่งกลุ่มข้อมูลด้วยตัวเอง”, เอกสารประกอบการประชุมวิชาการ NCSIT, 14 กุมภาพันธ์ 2552.

[13] F.K. Soong and B-H. Juang, “Linear spectrum Pair (LSP) and speech data compression”, in Proc. IEEE Acoustics Speech and Signal Proc, March 1984, San diego,CA, vol. 1, 1984. pp. 1.10.1-1.10.4.

[14] Won Cheol Lee, Myung Jin Bae, “A new line spectrum pair representation for spectral envelope quantization” MILCON’96 Conference Proceedings. IEEE Volume 2, 21-24 Oct.,1996 Page(s):658-662.

[15] สุทธี ทับทองดี, "การพัฒนาเทคนิคสำหรับการบีบอัดสัญญาณเสียงพูดภาษาไทยในมาตรฐาน LPC-10", วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2551

[16] สุวรรณ รื่นสุคนธ์และจักรี ศรีนนท์ฉัตร, “การรู้จำเสียงพูดภาษาไทยโดยใช้สัมประสิทธิ์คู่เส้นสเปกตรัมร่วมกับโครงข่ายประสาทเทียม”, International Joint Conference on Computer Science and Software Engineering (JCSSE2009), Vol 6, No.2, 2009. pp 341 - 346.

ภาคผนวก

ผลงานวิจัยตีพิมพ์เผยแพร่

ประวัติผู้เขียน

ชื่อ – นามสกุล	นางสาวอรอนงค์ วิริยานุรักษ์นคร
วัน เดือน ปีเกิด	2 พฤษภาคม 2514
ที่อยู่	39/1159 หมู่บ้านกรีนการ์เด้นท์โฮม ถ.รังสิต-นครนายก ต.บึงน้ำรักษ์ อ.ธัญบุรี จ.ปทุมธานี 12110
ประวัติการศึกษา	สำเร็จการศึกษา...

- ครุศาสตรอุตสาหกรรมบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม จากสถาบันเทคโนโลยีราชมงคล เมื่อปี พ.ศ. 2540
- วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า (อิเล็กทรอนิกส์) จากมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี เมื่อ พ.ศ.2544
- ครุศาสตรอุตสาหกรรมมหาบัณฑิต สาขาไฟฟ้า จากสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ เมื่อปี พ.ศ.2545

ประวัติการทำงาน

พ.ศ. 2535-2536	ตำแหน่ง Asst.Editor (Studio Grammy) บริษัท แกรมมี่ เอนเตอร์เทนเม้นท์ (มหาชน) จำกัด
พ.ศ. 2536-2538	ตำแหน่ง Senior Tape Operator (Post Production) บริษัท อิลูร่า (ประเทศไทย) จำกัด
พ.ศ. 2538-2540	ตำแหน่ง Producer (Cutting House) บริษัท ฟินิโต้ จำกัด
พ.ศ. 2541-2546	ตำแหน่งหัวหน้าฝึกอบรมและหลักสูตร, หัวหน้าวิจัยและประเมินผล สำนักเทคโนโลยีสารสนเทศ (ศูนย์กลางคลองหก) สถาบันเทคโนโลยีราชมงคล
พ.ศ.2546-2548	ตำแหน่งอาจารย์ประจำสาขาวิชาเทคนิคศึกษา วิทยาเขตเทเวศร์ สถาบันเทคโนโลยีราชมงคล
พ.ศ.2548-2551	ตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะวิศวกรรมศาสตร์และสถาปัตยกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ
พ.ศ.2551-ปัจจุบัน	ตำแหน่งอาจารย์ประจำสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

ผลงานวิจัย

O. Wiriyanuruknakorn and J. Srinonchat, *"A Finite State Vector Quantizer for New Bit Rate Speech Compression"* IEEE-International Conference on Signal processing, Communications and Networking (ICSCN'08), Jan 4-6, 2008. pp.255-259