การรู้จำเสียงพูดคำไทยด้วยวิธีการเอมเอ็ฟซีซี และโครงข่ายประสาทเทียม

THAI WORD SPEECH RECOGNITION USING MFCC AND ARTIFICIAL NEURAL NETWORKS

จักรพันธ์ จิตรทรัพย์

JAKKAPAN JITSUP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมธารณแทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2551

KMITL-2008-EN-M-030-033

การรู้จำเสียงพูดคำไทยด้วยวิธีการเอมเอ็ฟซีซี และ โครงข่ายประสาทเทียม

THAI WORD SPEECH RECOGNITION USING MFCC AND ARTIFICIAL NEURAL NETWORKS

จักรพันธ์ จิตรทรัพย์

JAKKAPAN JITSUP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2551

KMITL-2008-EN-M-030-033

THAI WORD SPEECH RECOGNITION USING MFCC AND ARTIFICIAL NEURAL NETWORKS

JAKKAPAN JITSUP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

KMITL-2008-EN-M-030-033

หัวข้อวิทยานิพนธ์	การรู้จำเสียงพูดคำไทยด้วยวิธีการเอมเอ็ฟซีซี และโครงข่ายประสาทเทียม
นักศึกษา	นายจักรพันธ์ จิตรทรัพย์
รหัสนักศึกษา	48061051
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2551
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.สมเกียรติ อุดมหรรษากุล

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอหลักการรู้จำเสียงพูดคำไทยเพื่อนำไปใช้ในการควบคุมโปรแกรมเล่นเพลงวินแอมป์ (Winamp player) โดยมีคำสั่งสำหรับการควบคุมโปรแกรมเล่นเพลง 8 คำได้แก่ คำว่า "เปิดเครื่อง" "ปิดเครื่อง" "เพลงก่อนหน้า" "เพลงถัดไป" "เพิ่มเสียง" "ลดเสียง" "เล่นเพลง" และ "หยุดเพลง" ซึ่งอัลกอริทึมในส่วนภาคการเตรียมสัญญาณเสียงเบื้องต้นเพื่อใช้ในการตัดบริเวณหัวท้ายของสัญญาณเสียง (Endpoint detection) จะใช้หลักการหาค่าพลังงาน ในส่วนของการดึงคุณลักษณะเด่นของสัญญาณเสียงจะใช้วิธีการหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient : MFCC) โดยการดึงคุณลักษณะเด่นของสัญญาณเสียงจะกระทำ 2 ส่วน คือ การดึงคุณลักษณะเด่นของสัญญาณเสียงทั้งคำพูดร่วมกับการดึงคุณลักษณะเด่นของสัญญาณเสียงเฉพาะพยางค์แรกของคำพูด และ ในส่วนของการรู้จำ (Recognition) จะใช้ระบบโครงข่ายประสาทเทียม (Artificial Neural Networks) ประเภทเพอเซปตรอนหลายชั้น (Multilayer perceptron : MLP) และ มีการเรียนรู้แบบแพร่กลับ (Backprpagation) เพื่อควบคุมโปรแกรมเล่นเพลงวินแอมป์

Thesis Title	Thai Word Speech Recognition Using MFCC and Artificial Neural Networks
Student	Mr. Jakkapan Jitsup
Student ID.	48061051
Degree	Master of Engineering
Program	Information Engineering
Year	2008
Thesis Advisor	Asst. Prof. Dr. Somkait Udomhunsakul

ABSTRACT

This thesis proposed Thai word speech recognition approach used to control Winamp program. In this thesis focused on 8 words such as "/pèrt/ /krêuaŋ/" (turn on), "/'pit/ /krêuaŋ/" (turn off), "/pleŋ/ /gòn / /nâa/" (previous song), "/pleŋ/ /tàt/ /pai/" (next song), "/pêrm/ /sĭaŋ/" (turn up), "/lót/ /sĭaŋ/" (turn down), "/lên/ /sĭaŋ/" (play song) and "/yòot/ /pleŋ/" (stop the song). The pre-processing algorithm is based on energy of signal for the boundary detection. Since the feature of the first syllable is the main difference feature between each tested-word. Therefore, we propose to use the Mel frequency cepstral coefficient (MFCC) of each word and the first syllable of word as the feature parameter. Next, these features are fed into multilayer perceptron (MLP) neural network with backpropagation learning algorithm for training and identification process. Finally, this result will be applied to control Winamp player.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยความช่วยเหลือชี้แนะในด้านต่างๆ จาก ผศ.ดร. สมเกียรติ อุดมหรรษากุล ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ที่คอยช่วยชี้แนะให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า ซึ่งข้าพเจ้ารู้สึกซาบซึ้งในความช่วยเหลือเป็นอย่างมากทำให้วิทยานิพนธ์นี้สำเร็จลงได้เป็นอย่างดี

ขอขอบพระคุณ รศ.อรลาภ แสงอรุณ ที่ให้ความรู้ด้านโครงข่ายประสาทเทียมและแหล่งข้อมูลในอินเตอร์เน็ตรวมถึงให้เอกสารอื่นๆที่เกี่ยวกับโครงข่ายประสาทเทียม และขอขอบพระคุณ ดร.พิทักษ์ ธรรมวารินที่ให้ยืมหนังสือเกี่ยวกับระบบรู้จำเสียง

ขอขอบพระคุณ บิดา และ มารดาที่ให้ชีวิต คอยดูแล ให้กำลังใจและกำลังทรัพย์ในการศึกษาวิจัยมาโดยตลอด จนทำให้สำเร็จมาถึงจุดนี้ได้

ขอขอบคุณนางสาวอัจฉราภรณ์ สุวรรณรมย์ ที่ให้คำแนะนำรวมถึงเอกสารการใช้โปรแกรม Qnet ทำให้ระบบการรู้จำเสียงในวิทยานิพนธ์นี้มีความแม่นยำยิ่งขึ้น

ขอขอบคุณพี่ ๆ เพื่อน ๆ น้อง ๆ นักศึกษาทุกคนที่ช่วยเหลือในการบันทึกเสียงเพื่อใช้ในการทดลองในวิทยานิพนธ์นี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์นี้ ข้าพเจ้าขอมอบให้แด่ผู้มีพระคุณทุกท่าน

จักรพันธ์ จิตรทรัพย์

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญรูป (ต่อ)

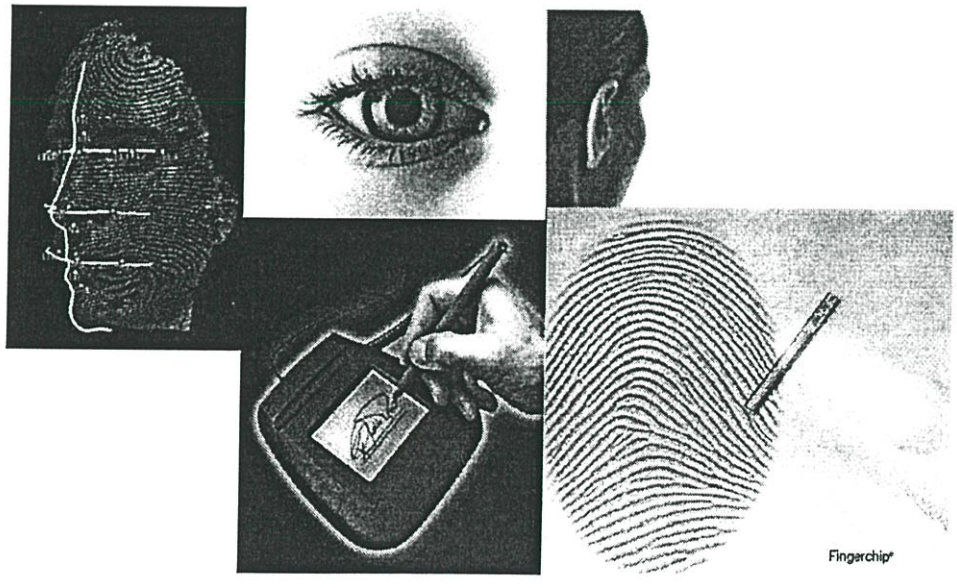
รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการพัฒนาเทคโนโลยีต่างๆ มุ่งเน้นไปที่การสร้างสิ่งอำนวยความสะดวกให้กับมนุษย์ เช่น การประดิษฐ์รถยนต์ทำให้ผู้คนสามารถเดินทางไกลได้ หรือ แม้แต่อุปกรณ์ไฟฟ้าต่างๆ ที่อยู่ภายในบ้านหรือรอบตัวเรา ซึ่งอุปกรณ์ไฟฟ้าเหล่านั้นจะมีวิธีควบคุมการทำงานที่แตกต่างกันไปในแต่ละอุปกรณ์ ซึ่งส่วนใหญ่ผู้ใช้จะต้องควบคุม หรือ สั่งงานผ่านการกดปุ่มบนตัวเครื่อง หรือ ผ่านการกดปุ่มบนรีโหมดคอนโทรล โดยหลายปีที่ผ่านมาจนถึงปัจจุบันได้มีความพยายามที่จะพัฒนาการควบคุมอุปกรณ์ไฟฟ้า หรือ เครื่องจักรต่างๆ ด้วยเสียงพูดของมนุษย์เพื่อเป็นการเพิ่มความสะดวก และ รวดเร็วในการสั่งงานอุปกรณ์ไฟฟ้าเหล่านั้นได้มากยิ่งขึ้น

ในระบบรักษาความปลอดภัยโดยใช้การระบุลักษณะของตัวบุคคล (Biometrics personal identification) ในปัจจุบันที่มีใช้งานได้แก่ การตรวจสอบลายนิ้วมือ (Fingerprints) การตรวจสอบม่านตา (Retinal patterns) การตรวจสอบใบหน้า (Face recognition) หรือ การตรวจสอบความถูกต้องของลายเซ็น (Signature verification) ดังแสดงในรูปที่ 1.1 ซึ่งการนำเอาลักษณะเฉพาะของตัวบุคคลมาใช้ระบุตัวบุคคลนั้นยากต่อการปลอมแปลง เนื่องจากว่าลายนิ้วมือหรือม่านตาของมนุษย์แต่ละคนไม่เหมือนกัน แต่อย่างไรก็ตามการที่จะออกแบบอัลกอริทึมเพื่อใช้ระบุตัวบุคคล ให้มีประสิทธิภาพการตรวจสอบที่แม่นยำ และ มีความน่าเชื่อถือนั้นทำได้ยาก



รูปที่ 1.1 การพิสูจน์เอกลักษณ์ของตัวบุคคลที่มีใช้ในปัจจุบัน

ระบบการรู้จำเสียงพูดสามารถแบ่งออกได้เป็น 2 แบบคือ แบบไม่จำกัดผู้พูด (Speaker Independent) และ แบบจำกัดผู้พูด (Speaker dependent) นอกจากนั้นระบบเหล่านี้ยังแบ่งย่อยเป็นแบบกำหนดคำพูดตายตัว (Text-dependent) และ แบบไม่กำหนดคำพูดตายตัว (Text-independent) ซึ่งในวิทยานิพนธ์นี้จะเป็นแบบกำหนดคำพูดตายตัว (Text-dependent) [1]

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อนำเสนอความสามารถของการรู้จำเสียงพูดคำไทยกับเสียงคำพูดจำนวน 8 คำ ได้แก่ เปิดเครื่อง ปิดเครื่อง เพลงก่อนหน้า เพลงถัดไป เพิ่มเสียง ลดเสียง เล่นเพลง และ หยุดเพลง เพื่อควบคุม และ สั่งงานโปรแกรมเล่นเพลงวินแอมป์ (Winamp player) โดยใช้โปรแกรม Qnet 2000 ซึ่งเป็นโปรแกรมเครื่องมือที่ใช้ในการออกแบบโครงข่ายประสาทเทียม และ สอนระบบโครงข่ายประสาทเทียม และใช้วิธี Mel-Frequency cepstral coefficients (MFCC) ในการดึงคุณลักษณะเด่นของสัญญาณเสียง

1.3 สมมุติฐานของการศึกษา

การรู้จำเสียงพูดคำไทยในวิทยานิพนธ์นี้มุ่งเน้นเพื่อการสั่งงานโปรแกรมเล่นเพลงวินแอมป์โดยที่ไม่ขึ้นกับผู้พูด คือ เสียงพูดที่ไม่ได้อยู่ในกลุ่มในการสอนระบบ (Training) จะยังคงสามารถสั่งงานโปรแกรมวินแอมป์ได้ โดยใช้การดึงคุณลักษณะเด่นด้วยวิธี MFCC กับสัญญาณเสียงทั้งคำ และ สัญญาณเสียงพยางค์แรกของคำพูดมาใช้ในการรู้จำ เนื่องจากสัญญาณเสียงพยางค์แรกของแต่ละคำมีความแตกต่างกันเป็นส่วนใหญ่ คือ

- เปิดเครื่อง
- ปิดเครื่อง
- เพลงก่อนหน้า
- เพลงถัดไป
- เพิ่มเสียง
- ลดเสียง
- เล่นเพลง
- หยุดเพลง

จากกลุ่มข้อมูลเสียงที่พิจารณาด้านบนจะเห็นว่าพยางค์แรกของคำส่วนใหญ่นั้นมีความแตกต่างกันโดยมีพยางค์ที่ซ้ำกันเพียง 1 พยางค์เท่านั้นคือ คำว่า "เพลง" ดังนั้นการดึงคุณลักษณะของเสียงพยางค์แรกนี้จะช่วยให้เปอร์เซ็นต์การรู้จำเสียงพูดนั้นดีขึ้น

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

แนวความคิดของวิทยานิพนธ์นี้ คือ การใช้พยางค์แรกของสัญญาณเสียงมาใช้ในการรู้จำคำพูดร่วมกับการใช้สัญญาณเสียงทั้งคำ และ ในการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) จะพิจารณาลักษณะโดยรวมของคุณลักษณะเด่นของสัญญาณเสียงเพื่อที่จะยังคงสามารถรู้จำเสียงคำพูดของบุคคลอื่นที่พูดคำเดียวกันได้ ซึ่งจะมีคุณลักษณะเด่นของสัญญาณเสียงคล้ายกัน และ สามารถรู้จำได้อย่างถูกต้อง ซึ่งระบบรู้จำเสียงพูด (Speech recognition) โดยทั่วไปจะมีหลักการทำงานที่สามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ๆ คือ

1. การเตรียมสัญญาณเสียงเบื้องต้น (Pre-processing)
2. การดึงคุณลักษณะเด่นของสัญญาณเสียงพูด (Feature extraction)
3. การรู้จำเสียงพูด (Speech recognition)

1.4.1 การเตรียมสัญญาณเสียงเบื้องต้น (Pre-processing)

การเตรียมสัญญาณเสียงเบื้องต้นถือเป็นขั้นตอนแรกและเป็นขั้นตอนที่มีความสำคัญที่สุดขั้นตอนหนึ่ง เนื่องจากขั้นตอนนี้จะทำหน้าที่ปรับสัญญาณเสียงอินพุตที่เข้ามาให้อยู่ในรูปแบบที่เหมาะสมที่จะนำไปใช้ในขั้นตอนการดึงคุณลักษณะเด่น (Feature extraction) และ การรู้จำ (Recognition) ซึ่งโดยปกติขั้นตอนการเตรียมสัญญาณเสียงเบื้องต้นจะมีฟังก์ชั่นการทำงานดังนี้ คือ

1. การกำจัดสัญญาณเสียงรบกวน (Noise) ด้วยการกรองทางความถี่ (Filtering)
2. การตัดหัวและท้ายของสัญญาณเสียง (Endpoint detection) คือ เป็นการตัดบริเวณที่ไม่ใช่ข้อมูลเสียงออกไป เพื่อเป็นการลดปริมาณข้อมูลที่จะต้องนำไปประมวลผลลงได้
3. การแบ่งเสียงในระดับพยางค์

1.4.2 การดึงคุณลักษณะเด่นของสัญญาณเสียงพูด (Feature extraction)

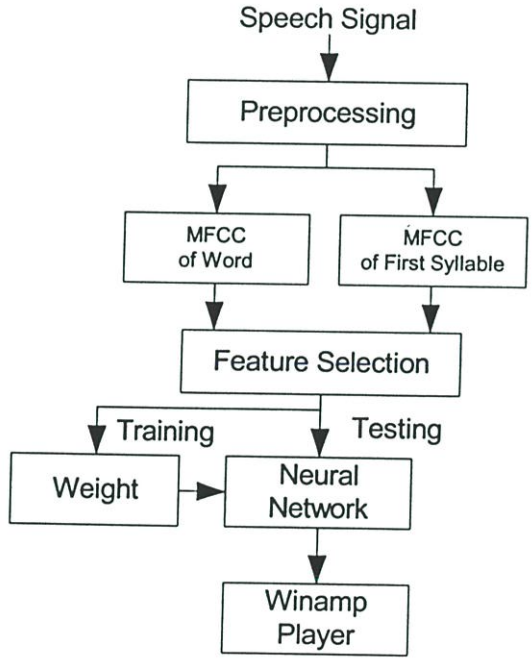
การนำสัญญาณเสียงที่ได้จากภาคการเตรียมสัญญาณเสียงเบื้องต้น (Pre-processing) มาหาคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) ซึ่งอีกนัยหนึ่งก็คือ การเข้ารหัสสัญญาณเสียงเพื่อให้ปริมาณข้อมูลของสัญญาณเสียงนั้นลดลง โดยเป็นการเลือกเก็บข้อมูลของสัญญาณเสียงเฉพาะข้อมูลที่สำคัญ หรือ เป็นข้อมูลที่สามารถบอกความแตกต่างของแต่ละคำพูดได้เท่านั้น ซึ่งทำให้ใช้เวลาในการประมวลผลข้อมูลสัญญาณเสียงนั้นน้อยลงตามไปด้วย

1.4.3 การรู้จำเสียงพูด (Speech recognition)

ข้อมูลสัญญาณเสียงที่ได้จากภาคการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) ในหัวข้อที่ผ่านมาจะนำมาเข้าระบบการรู้จำเสียงพูด (Recognition) ซึ่งจากอดีตที่ผ่านมาได้มีการนำเสนอมาแล้วหลายวิธีด้วยกัน คือ

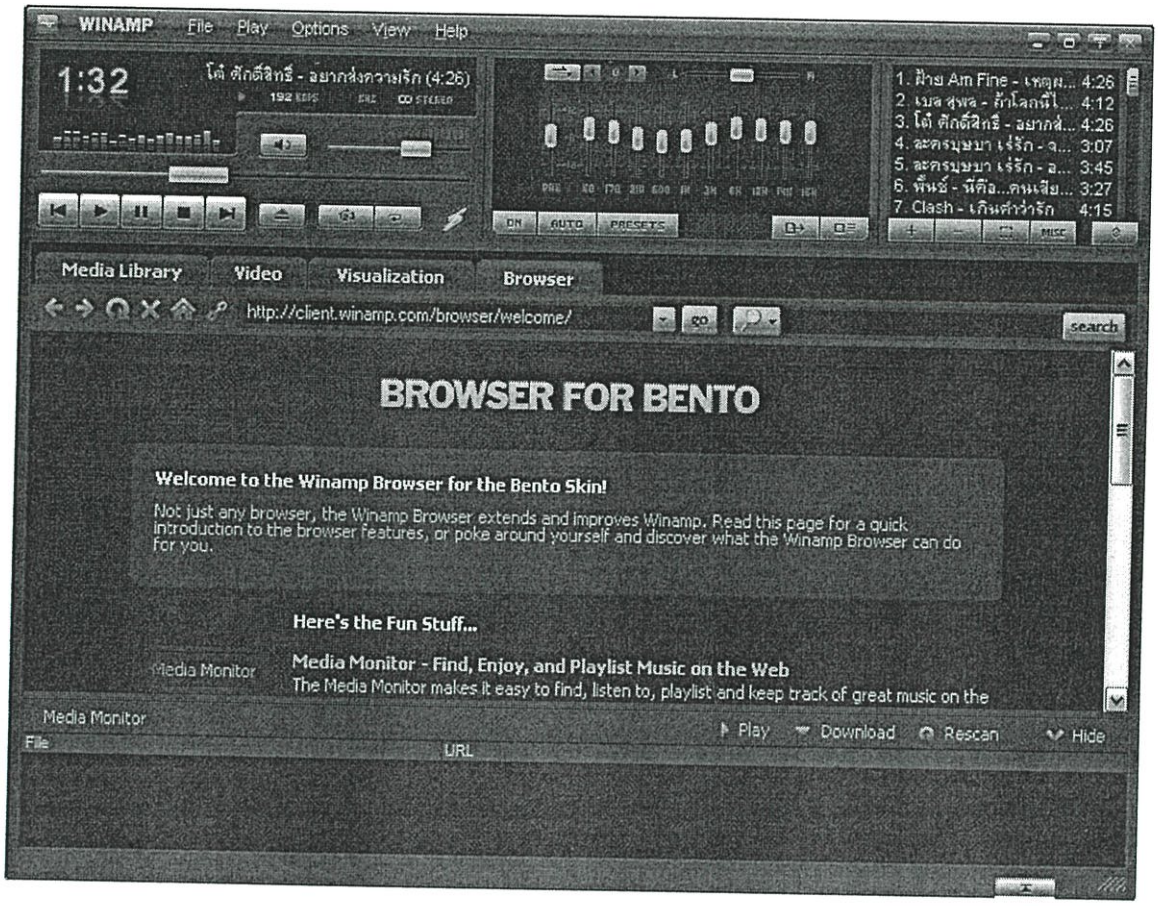
- การรู้จำเสียงโดยใช้ Hidden Markov Model (HMM) [21]
- การรู้จำเสียงโดยใช้ Artificial Neural Networks (ANNS) [22]
- การรู้จำเสียงโดยใช้ Dynamic Time Warping (DTW) [23]
- อื่นๆ

การออกแบบระบบรู้จำเสียงพูดคำไทยในวิทยานิพนธ์นี้จะใช้การดึงคุณลักษณะเด่นด้วยวิธีการหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient) ซึ่งมีชื่อย่อว่า MFCC และ ใช้กระบวนการรู้จำด้วยโครงข่ายประสาทเทียม (Artificial Neural Networks) ประเภทเพอเซปตรอนหลายชั้น (Multilayer Perceptron : MLP) และ ใช้กระบวนการเรียนรู้แบบแพร่กลับ (Backpropagation) ดังแสดงในรูปที่ 1.2



รูปที่ 1.2 โครงสร้างของระบบรู้จำเสียงพูดคำไทย

จากรูปที่ 1.2 การดึงคุณลักษณะเด่นของสัญญาณเสียงจะกระทำ 2 ส่วน คือ 1.) ดึงคุณลักษณะเด่นของทั้งคำพูด และ 2.) ดึงคุณลักษณะเด่นเฉพาะพยางค์แรกของคำพูด จากนั้นจะนำคุณลักษณะเด่นเหล่านี้มารวมกันในบล็อก Feature Selection แล้วนำข้อมูลนี้ไปรู้จำด้วยโครงข่ายประสาทเทียมเพื่อสั่งงานโปรแกรมเล่นเพลงวินแอมป์ดังแสดงในรูปที่ 1.3



รูปที่ 1.3 โปรแกรมเล่นเพลงวินแอมป์ (Winamp player)

1.5 การเปรียบเทียบระหว่างวิธีที่นำเสนอกับวิธีการแบบพื้นฐาน [7]

ศึกษาความสามารถของการรู้จำโดยนำพยางค์แรกของสัญญาณเสียงมาช่วยในการรู้จำ เนื่องจากกลุ่มคำที่พิจารณาในวิทยานิพนธ์นี้มีจำนวนคำ 2 พยางค์ และ 3 พยางค์ ในขณะที่วิธีพื้นฐานจะทำการดึงคุณลักษณะเด่นของสัญญาณเสียงทั้งคำแต่เพียงอย่างเดียว

1.6 ขอบเขตการวิจัย

1. วิทยานิพนธ์ฉบับนี้มุ่งเน้นศึกษาการรู้จำเสียงพูดคำไทยโดยไม่ขึ้นกับผู้พูด เพื่อใช้ในการควบคุมโปรแกรมเล่นเพลงวินแอมป์ (Winamp)

2. ศึกษาวิธีการประมวลผลข้อมูลสัญญาณเสียงด้วยวิธี Mel-Frequency cepstral coefficients (MFCC)

3. ศึกษาแบบจำลองการรู้จำโครงข่ายประสาทเทียมโดยใช้โปรแกรม Qnet 2000

4. ศึกษาการควบคุมโปรแกรมเล่นเพลงวินแอมป์ผ่านทาง Command line

1.7 ขั้นตอนของการศึกษา

1. กำหนดขอบเขตของงานวิจัย และ แนวทางการปฏิบัติงาน
2. ศึกษาการเขียนโปรแกรม MATLAB เพื่อใช้ในการเขียนโปรแกรมตามอัลกอริทึมที่ได้คิดขึ้น
3. ศึกษาระบบการประมวลผลสัญญาณเสียงเบื้องต้น (Pre-processing) ในแบบต่างๆ
4. ศึกษาวิธีการดึงคุณลักษณะเด่นของสัญญาณเสียงพูด (Feature extraction)
5. ศึกษาโปรแกรม Qnet 2000 เพื่อใช้ในการออกแบบโครงข่ายประสาทเทียม
6. ศึกษาวิธีการสั่งงานโปรแกรมเล่นเพลงวินแอมป์บนระบบปฏิบัติการ DOS
7. ศึกษาวิธีการใช้ Batch files เพื่อสั่งงานโปรแกรมเล่นเพลงวินแอมป์
8. ทดสอบอัลกอริทึมที่ได้พัฒนาขึ้น
9. สรุปผลการทดลอง และ ประเมินผล

1.8 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อพัฒนาระบบรู้จำเสียงพูดคำไทยด้วยวิธีใหม่ๆ ต่อจากงานวิจัยในอดีต
2. เพื่อเป็นแนวทางในการพัฒนาระบบรู้จำเสียงในแบบอื่นๆ
3. เพื่อสามารถนำไปใช้ในการสั่งงานโปรแกรมเล่นเพลงวินแอมป์ได้จริง
4. เพื่อเป็นแหล่งข้อมูลสำหรับผู้ที่จะทำวิจัยในอนาคต

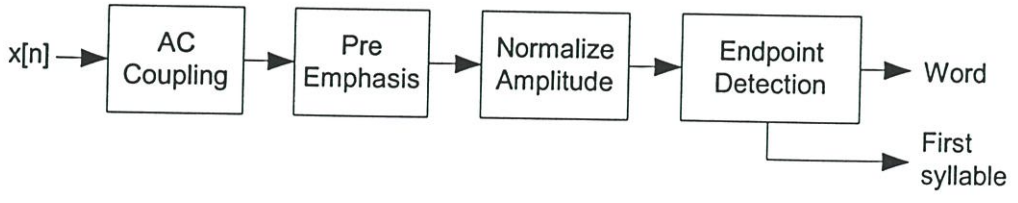
บทที่ 2
การเตรียมสัญญาณเสียงเบื้องต้น

2.1 บทนำ

การเตรียมสัญญาณเสียงเบื้องต้น (Pre-processing) เป็นขั้นแรกสุดที่สัญญาณเสียงจะถูกประมวลผลเพื่อทำให้สัญญาณที่เข้ามาทางอินพุตมีลักษณะพร้อมที่นำไปใช้ในขั้นตอนการดึงคุณลักษณะเด่น (Feature extraction) และ การรู้จำเสียง(Recognition) ต่อไป ซึ่งปกติขั้นตอนการเตรียมสัญญาณเสียงเบื้องต้นจะประกอบไปด้วยขั้นตอนการตัดสัญญาณรบกวน(Noise) ปรับขนาดสัญญาณเสียง(Amplitude Normalization) การตัดหัวท้ายของสัญญาณเสียง(Endpoint Detection) ซึ่งเป็นการตัดบริเวณที่ไม่ใช่สัญญาณเสียง(Unvoiced) ออกจากบริเวณสัญญาณเสียงที่ต้องการ (Voiced) และ อื่นๆ เพื่อให้ปริมาณข้อมูลสัญญาณเสียงนั้นลดลง โดยกระบวนการทั้งหมดในภาคการเตรียมสัญญาณเสียงเบื้องต้นจะได้กล่าวในหัวข้อถัดไป

2.2 การออกแบบระบบ

การออกแบบระบบในส่วนของภาคการเตรียมสัญญาณเสียงเบื้องต้น (Pre-Processing) จะใช้กระบวนการที่ไม่ซับซ้อนมากนัก เนื่องจากสัญญาณเสียงที่ใช้ในการทดลองนั้นอยู่ในห้องปิดที่มีสัญญาณรบกวน (Noise) ไม่สูงมากนัก ดังแสดงในรูปที่ 2.1 ซึ่งสัญญาณเสียงอินพุตจะผ่านกระบวนการเน้นสัญญาณเสียง(Pre-emphasis) และ ถูกลดขนาดข้อมูล โดยตัดบริเวณที่ไม่ใช่เสียงพูดออกไป และ นำเฉพาะบริเวณที่เป็นเสียงพูดเท่านั้นไปใช้งาน ซึ่งสัญญาณเสียงทางเอาท์พุตของภาคการประมวลผลสัญญาณเสียงเบื้องต้นจะได้สัญญาณเสียงออกมา 2 ส่วน คือ 1.) ส่วนของสัญญาณเสียงทั้งคำพูด(Word) และ 2.) ส่วนของสัญญาณเสียงพูดเฉพาะพยางค์แรกของคำ (First syllable) ซึ่งสัญญาณเสียงทั้งสองส่วนนี้จะถูกนำไปใช้ในการหาคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) ในภาคถัดไป



รูปที่ 2.1 บล็อกไดอะแกรมภาคการเตรียมสัญญาณเสียงเบื้องต้น

2.3 การบันทึกสัญญาณเสียง

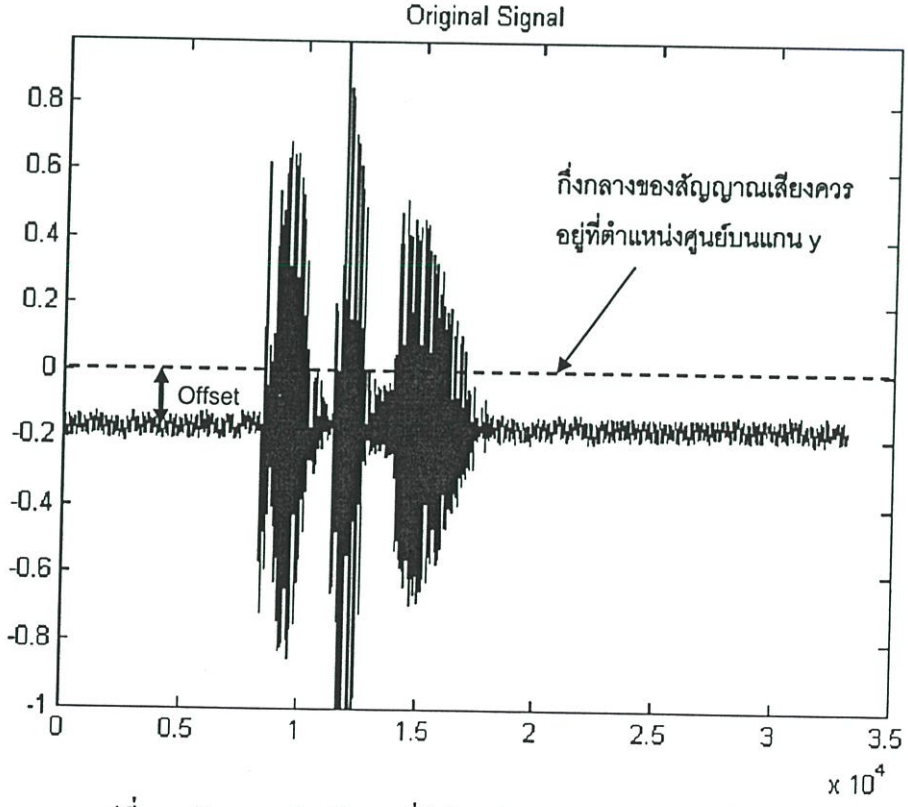
สัญญาณเสียงพูดถูกบันทึกผ่านไมโครโฟนด้วยอัตราการสุ่มสัญญาณ (Sampling rate) เท่ากับ 11.025kHz ขนาด 16 บิต เป็นเวลา 3 วินาทีในแต่ละคำพูด โดยสัญญาณเสียงที่บันทึกได้จะเก็บอยู่ในรูปแบบของไฟล์นามสกุลจุดเวฟ (Wave format)

2.4 ตัวกรองสัญญาณเอซีผ่าน (AC-Coupling)

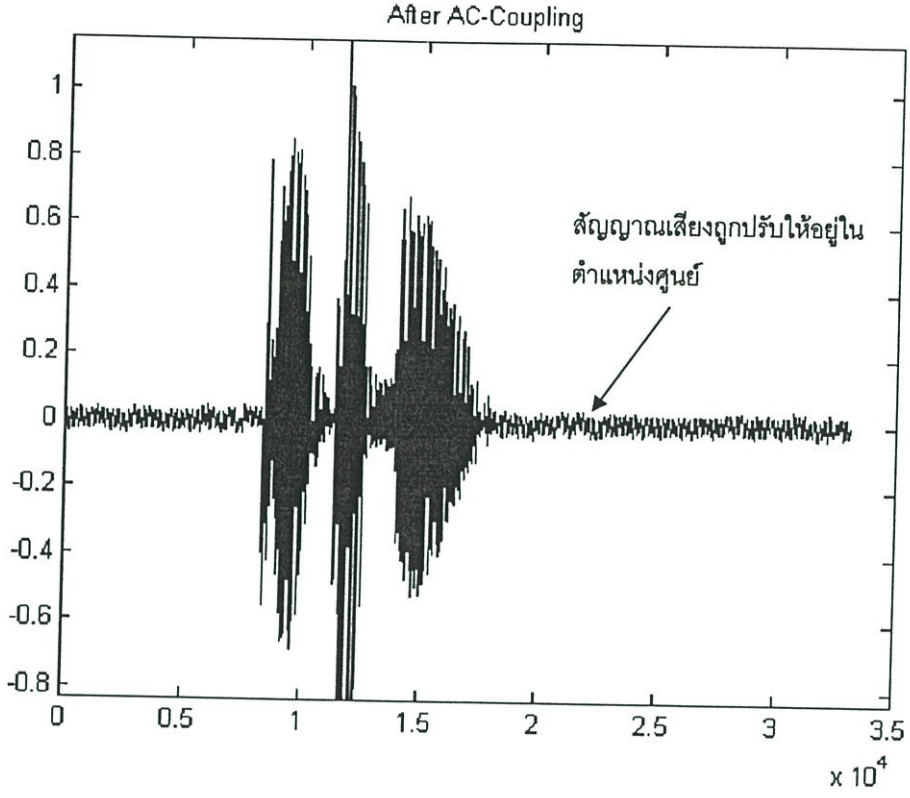
ตัวกรองสัญญาณ AC-Coupling จะทำหน้าที่ปรับระดับสัญญาณเสียงอินพุตให้มีค่าเฉลี่ยของสัญญาณอยู่ที่ศูนย์ [2] โดยการทำงานของ AC-Coupling จะทำการตัดองค์ประกอบของสัญญาณไฟตรง (DC component) ที่รวมอยู่ในสัญญาณเสียงออกไป ดังแสดงในสมการที่ 2.1

$$\tilde{x}(i) = x(i) - \frac{1}{N}\sum_{i=1}^{N} x(i) \tag{2.1}$$

จากสมการที่ 2.1 เป็นการนำสัญญาณเสียงอินพุต $x(i)$ ที่มีจำนวนข้อมูลสัญญาณเสียงเท่ากับ N ไปลบกับสมการหาค่าเฉลี่ยของสัญญาณเสียงอินพุตเดิม ซึ่งผลจะทำให้องค์ประกอบของสัญญาณไฟตรงหายไปดังแสดงในรูปที่ 2.2 และ รูปที่ 2.3



รูปที่ 2.2 สัญญาณเสียงอินพุตที่มีส่วนประกอบของสัญญาณไฟตรง



รูปที่ 2.3 สัญญาณเสียงอินพุตที่ตัดส่วนขององค์ประกอบไฟตรงออก

2.5 พลีเอ็มพาสิท (Pre-emphasis)

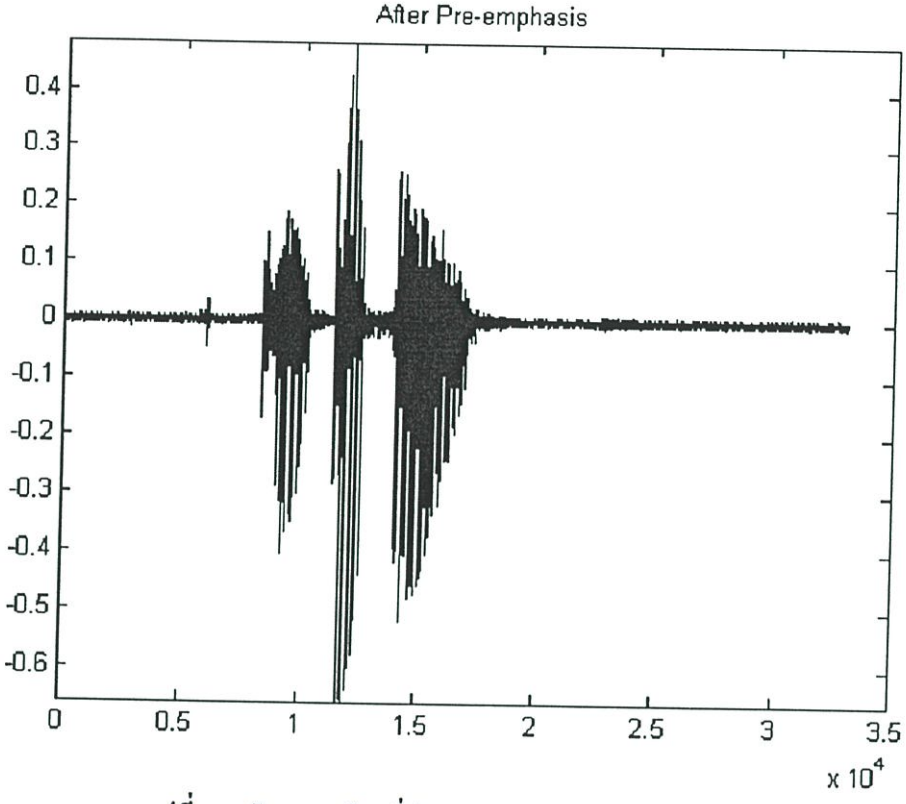
Pre-emphasis คือ ตัวกรองความถี่ชนิด First-order FIR filter ซึ่งทำหน้าที่ชดเชยสเปคตรัมของเสียงพูดมนุษย์ หรือ เป็นการเน้นสัญญาณเสียงในช่วงความถี่สูงไปประมาณ 6 dB/octave ซึ่งตัวกรองชนิดนี้มีผลทำให้แอมพลิจูดของสัญญาณเสียงในส่วนความถี่สูงมีขนาดสูงขึ้น [3] ซึ่งในการวิเคราะห์สัญญาณเสียงด้วยวิธีการหาค่าสัมประสิทธ์เซปสตรัมบนสเกลเมล (MFCC) ส่วนใหญ่จะทำการเน้นสัญญาณเสียงด้วยการทำ Pre-emphasis ก่อนเสมอ โดยสามารถเขียนเป็นฟังก์ชันถ่ายโอน (Transfer function) ได้ดังแสดงในสมการที่ (2.2) และ เขียนเป็นความสัมพันธ์ระหว่างอินพุต กับ เอาท์พุตได้ดังแสดงในสมการที่ (2.3) [4] [5]

$$H(z) = 1 - \beta z^{-1}, \qquad 0.9 \le \beta \le 1.0 \tag{2.2}$$

เมื่อ z^{-1} คือ Delay operator และ กำหนดให้ $\beta = 0.95$

$$s(n) = x(n) - \beta \cdot x(n-1) \qquad \text{เมื่อ } \beta = 0.95 \tag{2.3}$$

เมื่อสัญญาณเสียงผ่านตัวกรอง Pre-emphasis ผลจะทำให้สัญญาณเสียงนั้นมีค่า SNR ที่ดีขึ้น โดยทำให้สัญญาณรบกวนที่ติดมากับสัญญาณเสียงนั้นลดลงดังแสดงในรูปที่ 2.4 เทียบกับรูปที่ 2.3



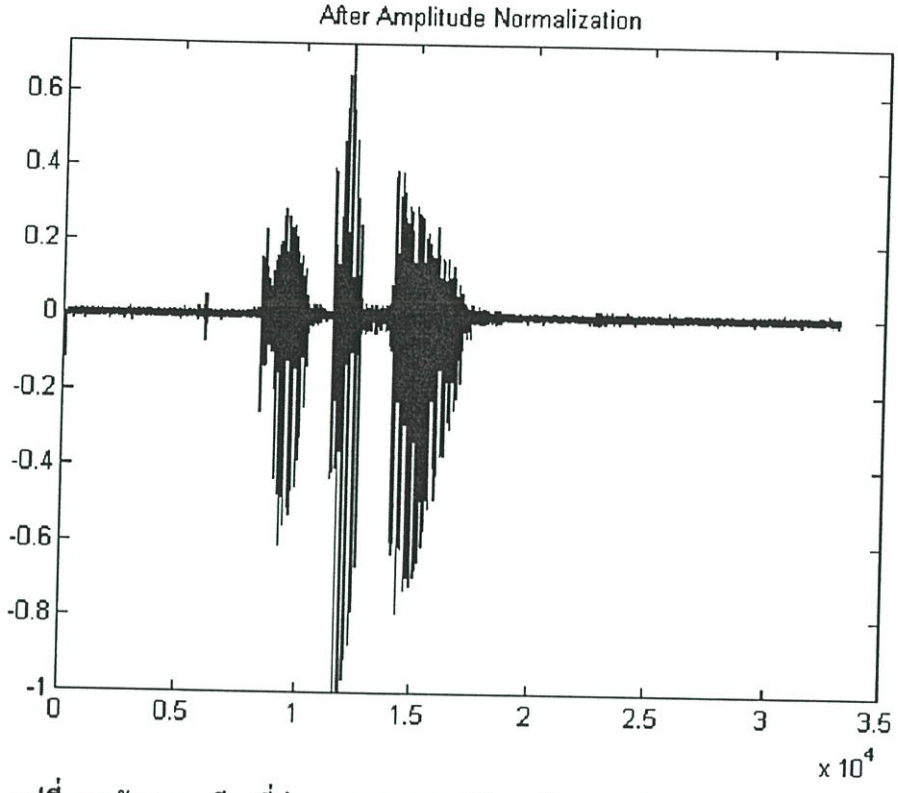
รูปที่ 2.4 สัญญาณเสียงที่ผ่านกระบวนการ Pre-emphasis

เมื่อเปรียบเทียบรูปที่ 2.3 กับ 2.4 จะเห็นว่าสัญญาณเสียงเมื่อผ่านกระบวนการ Pre-emphasis แล้วจะทำให้สัญญาณรบกวนนั้นลดน้อยลงได้ แต่สัญญาณเสียงโดยรวมจะมีขนาดแอมพลิจูดลดลงเช่นกัน ดังนั้น ก่อนที่จะนำสัญญาณเสียงนี้ไปประมวลผลต่อในภาคถัดไปจะต้องทำการปรับขนาดแอมพลิจูดให้สูงขึ้นด้วยการปรับระดับแอมพลิจูดของสัญญาณเสียงให้อยู่ในช่วง -1 ถึง 1

2.6 การปรับระดับแอมพลิจูดของสัญญาณเสียง (Amplitude normalization)

การปรับระดับแอมพลิจูดของสัญญาณเสียงให้อยู่ในช่วง -1 ถึง 1 โดยเมื่อสัญญาณเสียงผ่านภาค Pre-emphasis ขนาดแอมพลิจูดของสัญญาณเสียงโดยรวมจะมีขนาดเล็กลง ดังนั้น การทำงานในส่วนนี้จึงเป็นการปรับขนาดสัญญาณเสียงให้สูงขึ้นอยู่ในระดับ -1 ถึง 1 ดังแสดงในสมการที่ 2.4 และ ผลลัพธ์ของการปรับระดับแอมพลิจูดของสัญญาณเสียงแสดงดังรูปที่ 2.5

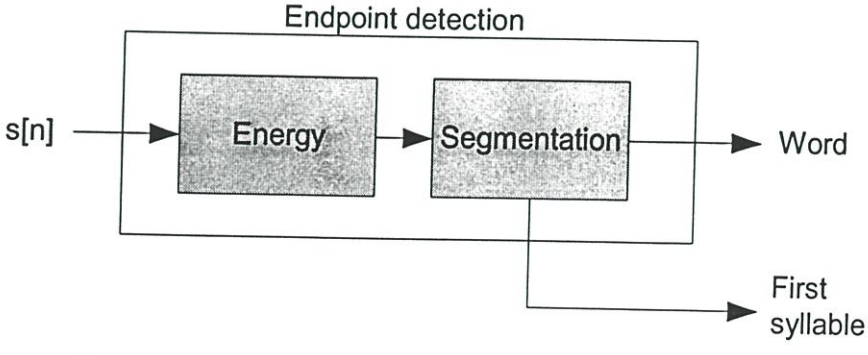
$$\tilde{s}(n) = \frac{s(n)}{\max(|s(n)|)} \tag{2.4}$$



รูปที่ 2.5 สัญญาณเสียงที่ผ่านกระบวนการปรับระดับแอมพลิจูดของสัญญาณเสียง

2.7 การตัดหัวท้ายของสัญญาณเสียง (Endpoint detection)

การตัดหัวท้ายของสัญญาณเสียง (Endpoint detection) [6] [7] คือ การตัดบริเวณที่ไม่ใช่สัญญาณเสียง(Unvoiced) ออกจากสัญญาณเสียงที่ต้องการ (Voiced) ซึ่งโดยปกติจะอยู่ตรงบริเวณส่วนหัว และ ส่วนท้ายของรูปคลื่นสัญญาณเสียง (Speech waveform) ที่บันทึกได้ ซึ่งผลลัพธ์จะทำให้ข้อมูลของสัญญาณเสียงที่จะนำไปประมวลผลในภาคถัดไปนั้นน้อยลง และ ทำให้ระบบรู้จำเสียงนั้นมีการทำงานที่เร็วขึ้น ซึ่งการตัดบริเวณที่ไม่ใช่สัญญาณเสียงออกจากสัญญาณเสียงที่ต้องการมีกระบวนการทำงานดังแสดงในรูปที่ 2.6



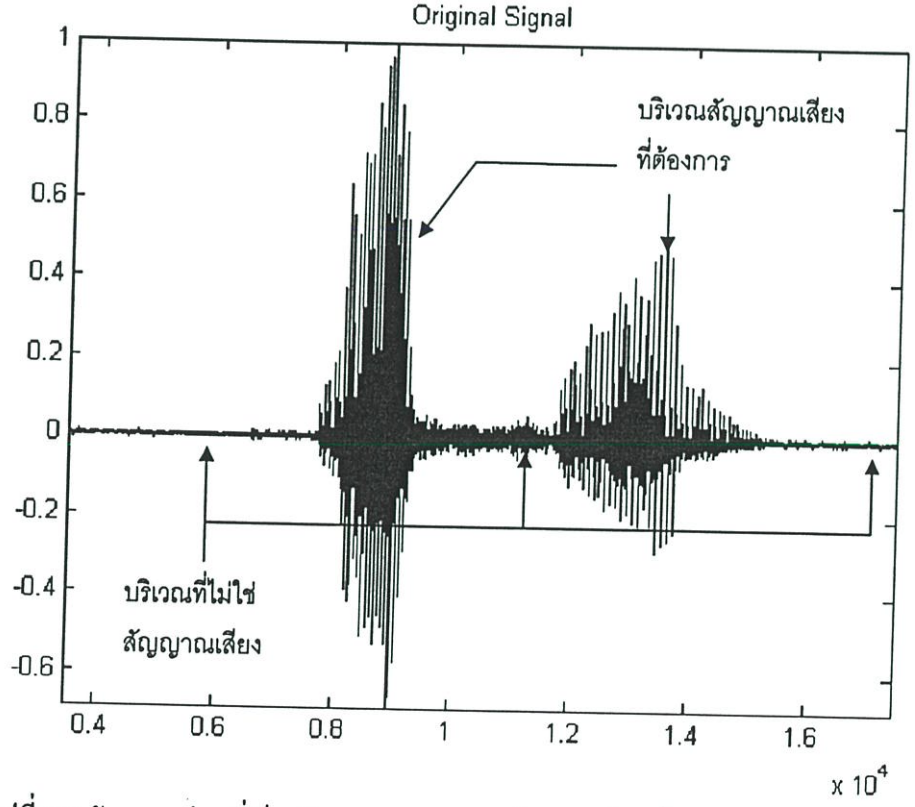
รูปที่ 2.6 บล็อกไดอะแกรมการทำงานของการตัดหัวท้ายของสัญญาณเสียง

จากรูปที่ 2.6 ผลลัพธ์ทางเอาท์พุตของบล็อก Endpoint detection จะถูกแบ่งออกเป็น 2 ส่วน คือ 1.) ส่วนของสัญญาณเสียงทั้งคำ (Word) และ 2.) ส่วนของสัญญาณเสียงเฉพาะพยางค์แรกของคำ (First syllable) โดยวิธีการตัดหัวท้ายของสัญญาณเสียง (Endpoint detection) ที่มีการใช้งานกันโดยทั่วไปจำแนกได้เป็น 3 วิธีคือ

1. การตัดหัวท้ายของสัญญาณเสียงด้วยวิธีหาค่าพลังงาน (Energy) [8]
2. การตัดหัวท้ายของสัญญาณเสียงด้วยวิธีสเปคโตรแกรม (Spectrogram) [6]
3. การตัดหัวท้ายของสัญญาณเสียงด้วยวิธีอัตราการผ่านค่าศูนย์ (Zero-crossing) [9]

2.7.1 การตัดหัวท้ายสัญญาณเสียงด้วยวิธีหาค่าพลังงาน (Energy)

การหาค่าพลังงานของสัญญาณเสียงโดยทั่วไปจะใช้เทคนิคการแบ่งสัญญาณเสียงออกเป็นส่วนย่อย (Frame) จากนั้นนำสัญญาณเสียงแต่ละส่วนย่อยมาหาค่าพลังงาน (Short-time energy) โดยใช้แนวความคิดที่ว่าส่วนที่เป็นสัญญาณเสียง (Voiced) จะเป็นส่วนที่มีระดับพลังงานสูงกว่าส่วนที่ไม่ใช่สัญญาณเสียง (Unvoiced) ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 สัญญาณส่วนที่เป็นเสียง (Voiced) และ สัญญาณส่วนที่ไม่ใช่เสียง (Unvoiced)

เมื่อสัญญาณเสียงผ่านกระบวนการ Pre-emphasis เพื่อเป็นการลดสัญญาณรบกวนลงระดับหนึ่งแล้ว จากนั้นสัญญาณเสียงนี้จะผ่านกระบวนการหาค่าพลังงานเพื่อนำแผนภาพพลังงานไปใช้

ในการหาขอบเขตของคำ (Boundary detection) หรือ ขอบเขตของพยางค์ต่อไป ซึ่งอัลกอริทึมการหาค่าพลังงานสามารถแบ่งออกได้เป็น 5 วิธีดังต่อไปนี้ [8]

1. วิธี Absolute energy คือ การนำสัญญาณเสียงอินพุต $S[i]$มาแบ่งเป็นส่วนย่อย $S_n[i]$ หรือ เฟรมขนาด N จากนั้นนำสัญญาณมาหาค่าสมบูรณ์ และ หาผลบวกของสัญญาณในแต่ละเฟรม ดังแสดงในสมการที่ 2.5

$$E_n = \sum_{i=1}^{N} |S_n[i]| \tag{2.5}$$

2. วิธี Root mean square energy คือ วิธีที่ใช้หลักการของการหาค่าเฉลี่ยของค่าพลังงานในแต่ละเฟรมย่อย โดยสัญญาณเสียงในแต่ละเฟรมจะถูกยกกำลังสอง และ หารากที่สองดังแสดงในสมการที่ 2.6

$$E_n = \left[\frac{1}{N}\sum_{i=1}^{N} S_n^2[i]\right]^{1/2} \tag{2.6}$$

3. วิธี Square energy คือ วิธีนี้จะแบ่งสัญญาณเสียงอินพุตออกเป็นส่วนย่อยขนาด N จากนั้นหาผลรวมของสัญญาณเสียงในแต่ละส่วนย่อยที่ได้ทำการยกกำลังสอง ดังแสดงในสมการที่ 2.7

$$E_n = \sum_{i=1}^{N} S_n^2[i] \tag{2.7}$$

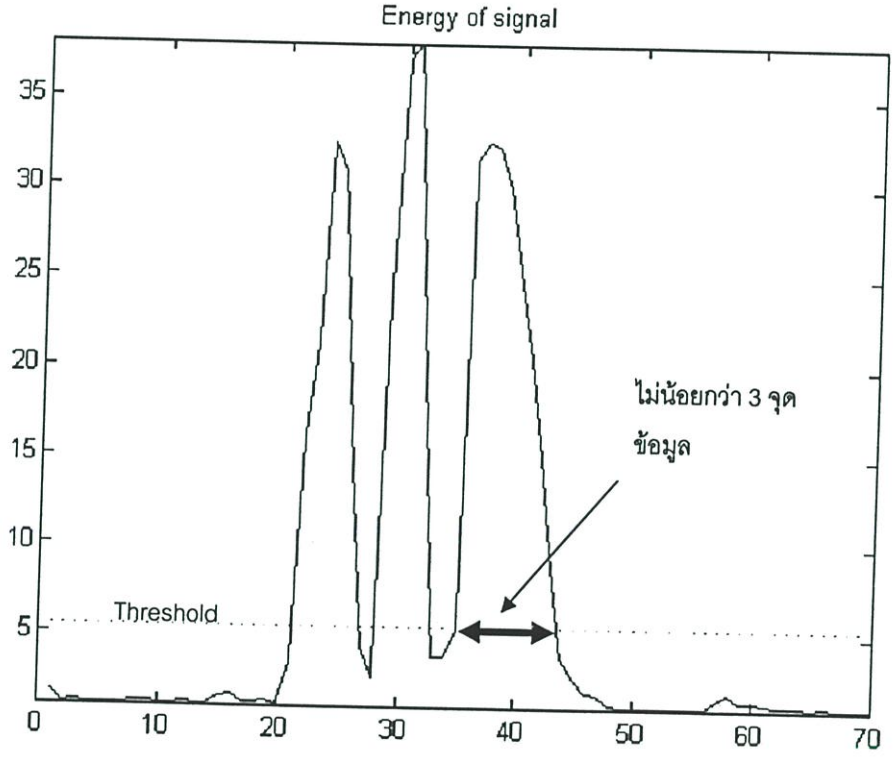
4. วิธี Teager energy คือ วิธีนี้จะใช้ข้อมูลสัญญาณเสียงที่ $S[i+1]$ คูณกับสัญญาณเสียงที่ $S[i-1]$ จากนั้นนำผลลัพธ์มาลบออกจากสัญญาณเสียงที่ยกกำลังสอง ดังแสดงในสมการที่ 2.8 และจากนั้นนำพลังงานนี้ไปหาผลรวมดังแสดงในสมการที่ 2.9

$$E[i] = S^2[i] - S[i+1] \cdot S[i-1] \tag{2.8}$$

$$E_n = \sum_{i=1}^{N} E[i] \tag{2.9}$$

5. วิธี Modified Teager energy คือ วิธีนี้เป็นการพัฒนามาจากวิธี Teager energy ซึ่งจะใช้หลักการหาค่าพลังงานของสัญญาณเสียง จากนั้นค่าพลังงานของสัญญาณเสียงนี้จะถูกถ่วงน้ำหนัก (Weighted) ด้วยผลของการยกกำลังสองของความถี่ของสัญญาณเสียง

ในวิทยานิพนธ์นี้จะใช้การตัดหัวท้ายของสัญญาณเสียง (Endpoint detection) ด้วยวิธีการหาค่าพลังงาน (Energy) แบบ Absolute energy โดยจะกำหนดให้แต่ละเฟรมมีขนาดข้อมูล N = 200 และ ในแต่ละเฟรมย่อยของสัญญาณเสียงจะไม่มีการซ้อนทับ (Overlap) ในแต่ละเฟรมย่อยของสัญญาณเสียง ซึ่งลักษณะของแผนภาพพลังงานของสัญญาณเสียงแสดงให้เห็นดังรูปที่ 2.8



รูปที่ 2.8 พลังงานของสัญญาณเสียงคำว่า "เพลงก่อนหน้า"

เมื่อได้ค่าพลังงานของสัญญาณเสียง จากนั้นการหาขอบเขตของแต่ละพยางค์สามารถทำได้โดยใช้ค่าการตัดสินใจ (Threshold) ซึ่งจะกำหนดไว้ที่ 1/7 ของค่าพลังงานสูงสุด ดังแสดงในรูปที่ 2.8 โดยที่ ณ. ตำแหน่งที่เส้นการตัดสินใจลากผ่านไปบนแผนภาพพลังงานของสัญญาณเสียงจะต้องมีจำนวนข้อมูลในแต่ละพยางค์ไม่น้อยกว่า 3 ข้อมูล เพื่อป้องกันสัญญาณรบกวนที่มีลักษณะปลายแหลม จึงจะถือว่าเป็นพยางค์ของสัญญาณเสียง (Voiced) ซึ่งถ้าน้อยกว่าจะถือว่าไม่ใช่สัญญาณเสียง (Unvoiced) ซึ่งวิธีการตัดหัวท้ายของสัญญาณเสียงสามารถแสดงดังสมการที่ (2.10), (2.11) และ (2.12)

$$Logic(n) = \left[\frac{E(n) - th}{|E(n) - th|} + 1\right] \times \left(\frac{1}{2}\right) \tag{2.10}$$

$$Edge(n) = Logic(n) - Logic(n-1) \tag{2.11}$$

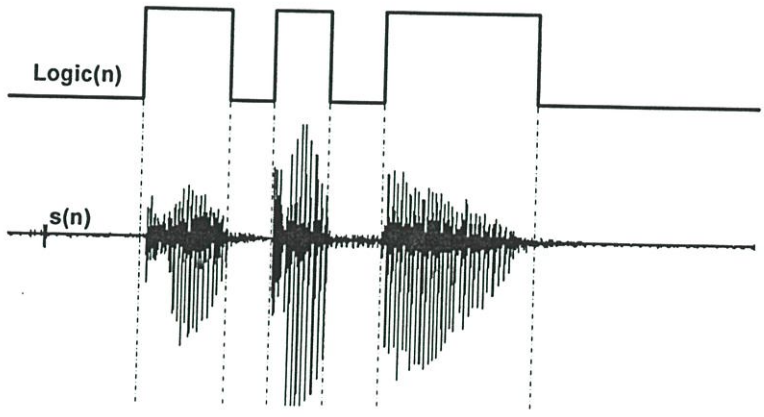
$$Output = Logic(n) \cdot s(n) \tag{2.12}$$

จากสมการที่ 2.10 เอาท์พุต $Logic(n)$ ของสมการนี้จะมีเพียง 2 ค่าคือ '1' หมายถึง บริเวณที่ค่าพลังงานของสัญญาณเสียงนั้นมีค่ามากกว่าค่าการตัดสินใจ (Threshold) และ '0' หมายถึง บริเวณที่ค่าพลังงานของสัญญาณเสียงนั้นมีค่าน้อยกว่าค่าการตัดสินใจ (Threshold) ซึ่งทำให้สามารถหาจุดเริ่มต้น และ จุดปลายของแต่ละพยางค์ได้

ในส่วนของสมการที่ 2.11 จะใช้สำหรับหาตำแหน่งจุดเริ่มต้นของพยางค์ซึ่งจะมีค่า "+1" และ หาตำแหน่งจุดปลายของพยางค์ซึ่งจะมีค่า "-1" ของแต่ละพยางค์ และ จะมีค่าเป็น "0" ที่จุดอื่นๆ ของสัญญาณเสียง

การตัดหัวท้ายของสัญญาณเสียงคำพูด หรือการตัดสัญญาณเสียงในระดับพยางค์ (Syllable) จะใช้สมการที่ (2.12) ในการตัด (Segmentation) โดยถ้าต้องการตัดหัวท้ายของสัญญาณเสียงระดับคำพูด (Word) จะใช้ค่าตำแหน่งขอบขาขึ้นของพยางค์จุดแรก และ ใช้ขอบขาลงของพยางค์จุดสุดท้าย (ซึ่งค่าตำแหน่งขอบนี้จะได้จากสมการที่ 2.11) จากนั้นข้อมูลสัญญาณเสียงที่อยู่ระหว่างตำแหน่งสองค่านี้จะถือเป็น 1 คำพูด ดังแสดงในรูปที่ 2.9

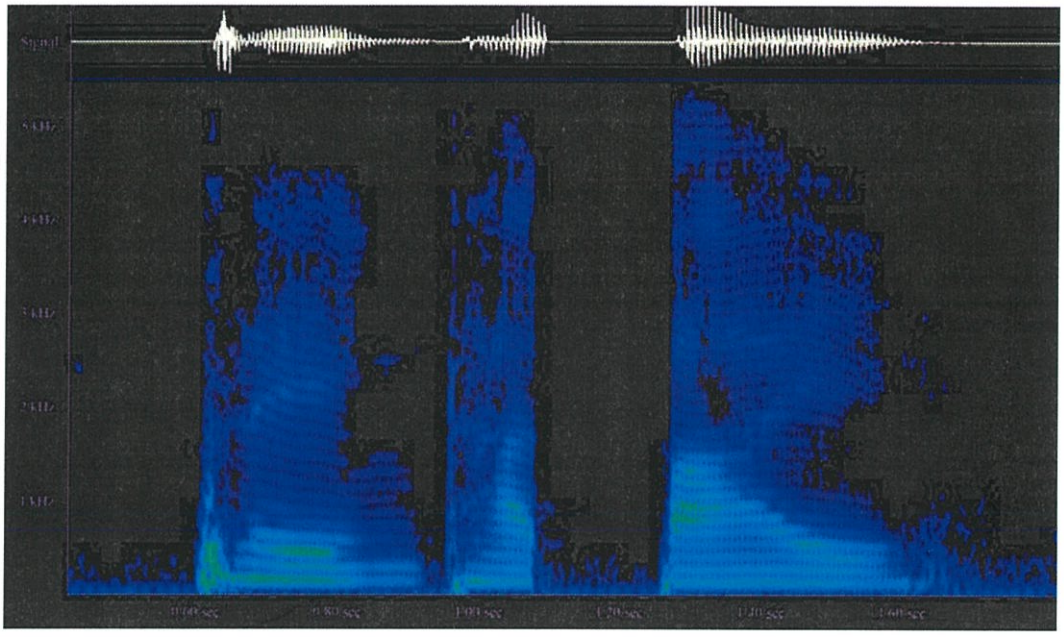
การหาพยางค์แรกของคำพูดจะใช้หลักการเดียวกันกับการตัดหัวท้ายของทั้งคำพูด คือ จะนำตำแหน่งขอบขาขึ้นของพยางค์จุดแรก และ ใช้ตำแหน่งขอบขาลงของพยางค์จุดแรกมาใช้ ซึ่งจะทำให้ได้ข้อมูลเสียงพยางค์ที่ 1 ออกมา



รูปที่ 2.9 การตัดหัวท้ายในระดับพยางค์ของสัญญาณเสียง

2.7.2 การตัดหัวท้ายของสัญญาณเสียงด้วยวิธีหาสเปคโตรแกรม (Spectrogram)

การตัดหัวท้ายของสัญญาณเสียงคำพูดโดยใช้วิธีการสเปคโตรแกรม (Spectrogram) จะเป็นการแสดงแผนภาพในแกนความถี่ และ แกนเวลา โดยจะใช้หลักการแปลงฟูเรียร์เพื่อใช้ในการสร้างแผนภาพสเปคโตรแกรม [6] ซึ่งจะทำให้ทราบว่าสัญญาณเสียงที่ตำแหน่งเวลาต่างๆ มีความถี่เป็นอย่างไร และ เมื่อพิจารณาในแนวแกน z จะทำให้ทราบถึงค่าระดับพลังงานของสัญญาณเสียง ณ. ความถี่นั้น ซึ่งจะบ่งบอกเป็นค่าสีต่างๆ บนแผนภาพดังแสดงในรูปที่ 2.10 โดยภาพที่แสดงเป็นการใช้โปรแกรม Audio Spectrum Analysis ในการวาดแผนภาพสเปคโตรแกรม



รูปที่ 2.10 แผนภาพสเปคโตรแกรม

2.7.3 การตัดหัวท้ายของสัญญาณเสียงด้วยวิธีอัตราการผ่านค่าศูนย์ (Zero-crossing)

หลักการของวิธีอัตราการผ่านค่าศูนย์ (Zero-crossing) คือ การที่รูปคลื่นของสัญญาณเสียงมีการตัดกับแกนเวลา ซึ่งบริเวณที่เป็นสัญญาณเสียง (Voiced) จะมีอัตราการตัดกับแกนเวลาน้อยกว่าบริเวณที่ไม่ใช่สัญญาณเสียง (Unvoiced) ซึ่งบริเวณที่ไม่ใช่สัญญาณเสียงจะมีอัตราการตัดผ่านแกนเวลามากกว่า และ เนื่องจากบริเวณของสัญญาณเสียงส่วนใหญ่จะมีค่าพลังงานอยู่ในช่วงความถี่ต่ำ และ บริเวณที่ไม่ใช่สัญญาณเสียงจะมีพลังงานอยู่ในช่วงความถี่สูง ประกอบกับวิธีอัตราการผ่านค่าศูนย์มีความสัมพันธ์โดยตรงกับความถี่ของสัญญาณเสียง ดังนั้นจึงสามารถนำวิธีการนี้มาหาขอบเขตของคำ หรือ พยางค์เสียงได้ [9] โดยสมการของวิธีอัตราการผ่านค่าศูนย์แสดงดังสมการที่ 2.13

$$Z = \frac{1}{2N} \cdot \sum_{n=1}^{N} \left| Sign(s(n)) - Sign(s(n-1)) \right| \tag{2.13}$$

เมื่อ

$$Sign(s(n)) = \begin{cases} 1 & ; s(n) > 0 \\ 0 & ; Others \end{cases}$$

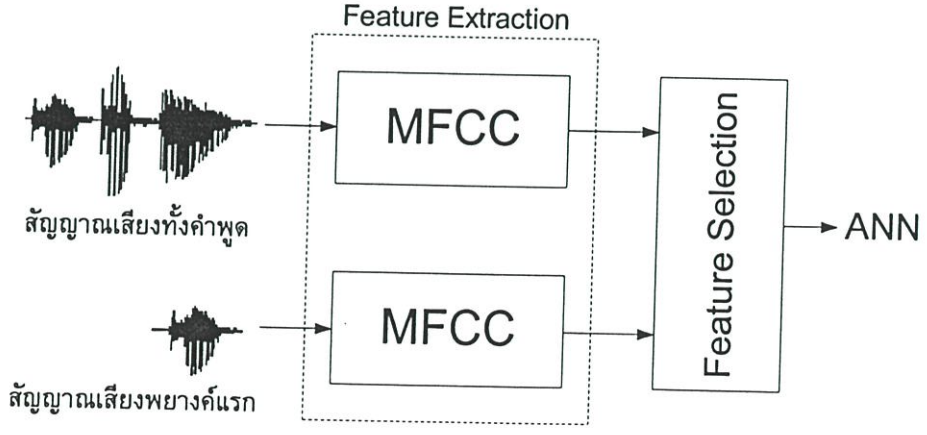
บทที่ 3
การดึงคุณลักษณะเด่นของสัญญาณเสียงพูด

3.1 บทนำ

ปัจจุบันวิธีการดึงคุณลักษณะเด่น (Feature extraction) ของสัญญาณเสียงที่นิยมใช้กันอย่างมาก คือ วิธี Linear predictive cepstral coefficients (LPCC) ซึ่งเป็นวิธีการหาสัมประสิทธิ์เซปสตรัมที่คำนวณมาจากวิธี LPC และ วิธีการหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient) ซึ่งมีชื่อย่อว่า MFCC โดยค่าสัมประสิทธิ์เซปสตรัม (Cepstral parameters) ที่ได้จากวิธี MFCC นี้จะสามารถอธิบายความไม่เป็นเชิงเส้นของการรับรู้ของหูมนุษย์ได้ดี เนื่องจากสเปกตรัมของวิธี MFCC นี้จะให้รายละเอียดที่ดีในช่วงสัญญาณความถี่ต่ำ และ นอกจากนั้นวิธี MFCC นี้ยังมีความทนทานต่อสัญญาณรบกวน (Noise) ได้ดีกว่าวิธี LPCC [10]

3.2 การออกแบบระบบ

กระบวนการดึงคุณลักษณะเด่นของสัญญาณเสียงในวิทยานิพนธ์นี้จะใช้วิธีการหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient) โดยการดึงคุณลักษณะเด่นของสัญญาณเสียงจะกระทำ 2 ส่วน คือ 1.) ดึงคุณลักษณะเด่นของสัญญาณเสียงทั้งคำพูด และ 2.) ดึงคุณลักษณะเด่นของสัญญาณเสียงเฉพาะพยางค์แรกของคำพูด จากนั้นคุณลักษณะเด่นของสัญญาณเสียงที่ได้จะนำไปสู่ขั้นตอนการเลือกคุณลักษณะเด่น (Feature selection) ของสัญญาณเสียงเพื่อนำข้อมูลนี้ไปใช้ในการรู้จำเสียงด้วยโครงข่ายประสาทเทียมต่อไปดังแสดงในรูปที่ 3.1



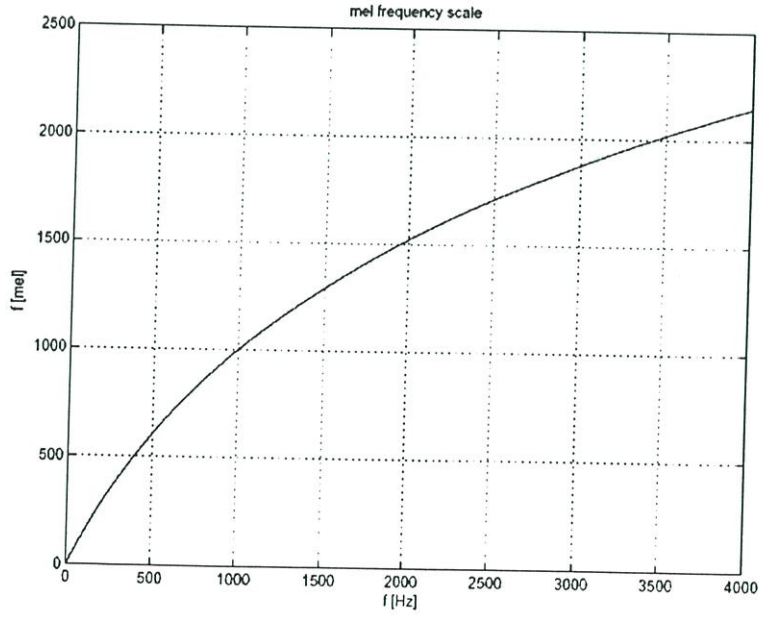
รูปที่ 3.1 บล็อกไดอะแกรมของภาคการดึงคุณลักษณะเด่นของสัญญาณเสียง

3.3 การดึงคุณลักษณะเด่นของสัญญาณเสียงด้วยวิธี MFCC

ขั้นตอนการดึงคุณลักษณะเด่นของสัญญาณเสียงพูด (Feature extraction) [11-14] คือ การแทนสัญญาณเสียงพูดที่มีปริมาณข้อมูลมากๆ ด้วยรูปแบบข้อมูลใหม่ที่มีปริมาณข้อมูลน้อยลง โดยจะให้เหลือเฉพาะข้อมูลที่จำเป็นต่อการนำไปใช้ในการรู้จำเท่านั้น ซึ่งในวิทยานิพนธ์นี้จะใช้วิธีการหาค่าสัมประสิทธ์เซปสตรัมบนสเกลเมล (MFCC) ในการดึงคุณลักษณะเด่นของสัญญาณเสียง ซึ่งวิธีการนี้จะเป็นการเปลี่ยนจากสเกลความถี่ธรรมดาให้เป็นสเกลบนความถี่เมล (Mel Scale) เนื่องจากการรับฟังของหูมนุษย์จะมีความไม่เป็นเชิงเส้น โดยจะให้รายละเอียดของข้อมูลเสียงที่ดีในบางช่วงความถี่เท่านั้นซึ่งจะเป็นช่วงความถี่ต่ำ และการไม่เป็นเชิงเส้นของแกนความถี่นี้จะถูกเรียกว่าสเกลเมล (mel-scale) และ ความถี่ที่วาดอยู่บนสเกลเมลจะถูกเรียกว่าความถี่เมล (mel-frequency) หรือ f_{mel} ซึ่งการปรับจากความถี่บนสเกลความถี่ธรรมดาให้เป็นความถี่เมลสามารถทำได้ดังสมการที่ 3.1

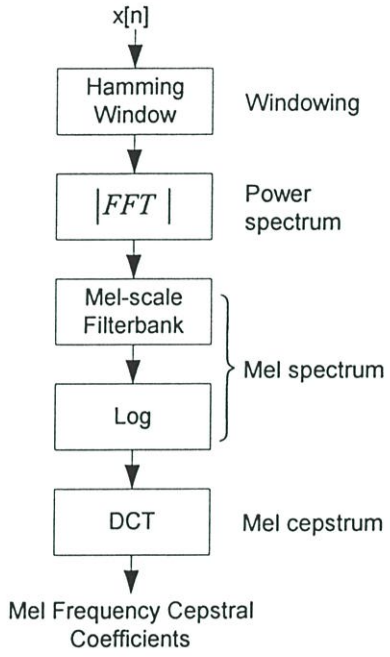
$$f_{mel}(f) = 2595 \cdot \log\left(1 + \frac{f}{700Hz}\right) \tag{3.1}$$

จากสมการที่ 3.1 เมื่อ f คือ ความถี่บนแกนความถี่แบบเชิงเส้นปกติ และ f_{mel} คือ ความถี่บนสเกลเมล ซึ่งสามารถแสดงด้วยกราฟได้ดังรูปที่ 3.2 โดยจากรูปเป็นการแสดงให้เห็นถึงความสัมพันธ์ระหว่างความถี่บนสเกลธรรมดากับความถี่บนสเกลเมลที่มีความไม่เป็นเชิงเส้น

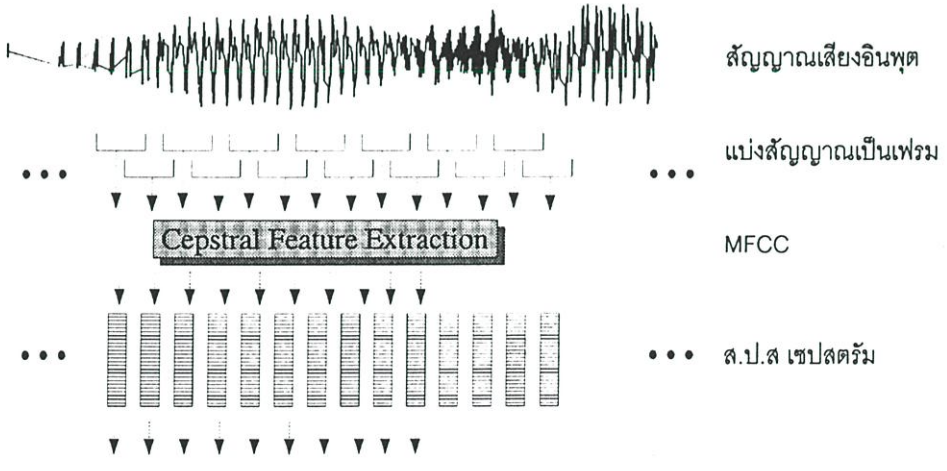


รูปที่ 3.2 กราฟแสดงความสัมพันธ์ทางความถี่ระหว่างสเกลธรรมดากับสเกลเมล (Mel scale)

วิธีการหาค่าสัมประสิทธิ์เซปสตรัมด้วยวิธี MFCC ขั้นตอนแรกจะทำการแบ่งสัญญาณเสียงที่จะนำมาดึงคุณลักษณะเด่นออกเป็นส่วนย่อย หรือ เฟรมจากนั้นนำสัญญาณเสียงแต่ละเฟรมมาแปลงให้อยู่ในโดเมนความถี่ (Frequency domain) ด้วยการแปลงฟูเรียร์ทรานฟอร์ม ซึ่งผลลัพธ์ที่ได้จะนำไปผ่านชุดตัวกรองรูปสามเหลี่ยม (Triangle-shaped windows) จากนั้นจะนำมาผ่านฟังก์ชันลอการิทึม และ ผ่านกระบวนแปลงโคซายน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform : DCT) ดังแสดงในรูปที่ 3.3 และ รูปที่ 3.4



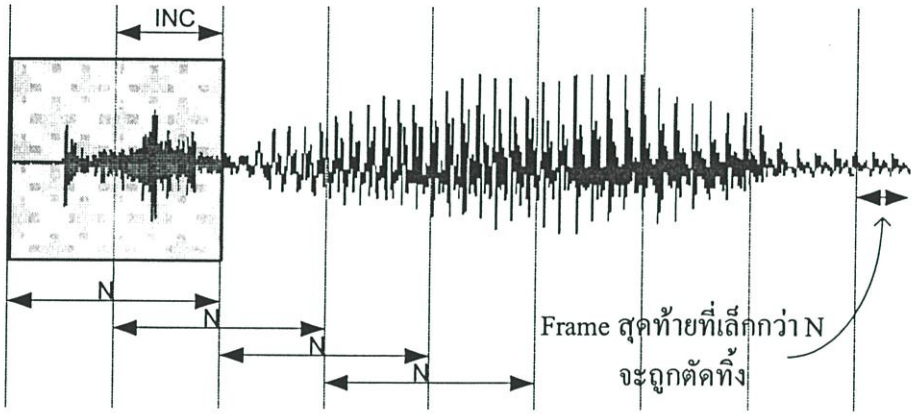
รูปที่ 3.3 บล็อกไดอะแกรมของการหาค่าสัมประสิทธิ์เซปสตรัมด้วยวิธี MFCC



รูปที่ 3.4 ขั้นตอนการหาค่าสัมประสิทธิ์เซปสตรัมด้วยวิธี MFCC

3.3.1 การแบ่งสัญญาณเสียงออกเป็นเฟรมย่อย (Windowing)

สัญญาณเสียงที่ได้มาจากภาคการเตรียมสัญญาณเสียงเบื้องต้นจะถูกแบ่งออกเป็นส่วยย่อย หรือเฟรม โดยปกติแต่ละเฟรมจะถูกแบ่งอยู่ในช่วงขนาด 20 – 30 มิลลิวินาที และ มีการเลื่อนครั้งละ 1/3 ถึง 1/2 ของขนาดเฟรม [11] ซึ่งในวิทยานิพนธ์นี้จะกำหนดให้แต่ละเฟรมจะมีขนาด N= 256 ข้อมูล หรือประมาณ 23มิลลิวินาที และ ทำการเลื่อนเฟรมทีละ N/2 หรือ INC=128 และ ถ้าเฟรมสุดท้ายมีขนาดน้อยกว่า N จะถูกตัดทิ้งดังแสดงในรูปที่ 3.5



รูปที่ 3.5 วิธีการแบ่งสัญญาณเสียงออกเป็นเฟรมย่อย

จุดศูนย์กลางของเฟรมที่ i สามารถหาได้ดังสมการที่ 3.2 โดยกำหนดให้ x(i) คือ สัญญาณเสียงอินพุต และ INC คือ การเลื่อนของเฟรมแต่ละครั้งโดย N คือ ขนาดของเฟรม ซึ่งผลลัพธ์ของค่าจุดศูนย์กลางของเฟรมที่ i จะถูกเก็บไว้ที่ fc(i)

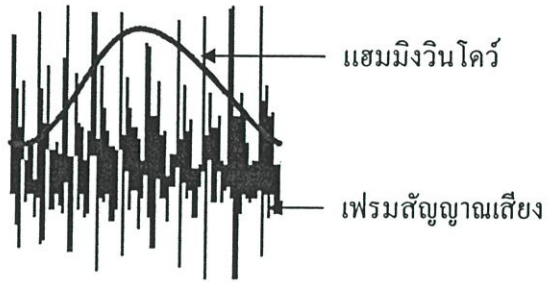
$$fc(i) = x\left((i-1)\cdot INC + \frac{(N+1)}{2}\right), \qquad \text{i=1,2,...} \tag{3.2}$$

จำนวนของเฟรมทั้งหมดที่ถูกแบ่งในสัญญาณเสียงอินพุตจะถูกเก็บไว้ที่ตัวแปร nf ดังแสดงในสมการที่ 3.3

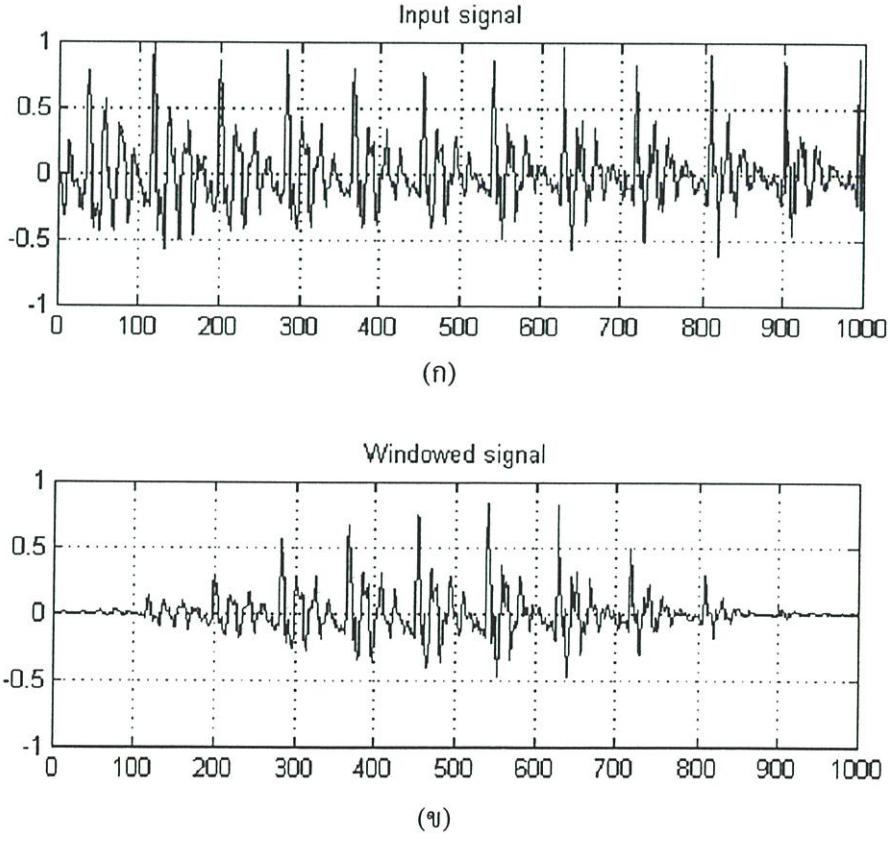
$$nf = \frac{(length\ of\ x) - N + INC}{INC} \tag{3.3}$$

เมื่อทำการแบ่งสัญญาณเสียงออกเป็นเฟรมหรือส่วนย่อยแล้วจากนั้นสัญญาณแต่ละเฟรมจะถูกคูณด้วยแฮมมิงวินโดว์ (Hamming window) ซึ่งสาเหตุที่ใช้แฮมมิงวินโดว์เนื่องจากต้องการให้สัญญาณเสียงในแต่ละเฟรมมีลักษณะที่เป็นคาบ (Periodic) และ ต่อเนื่อง (Continuous) ในตำแหน่ง

จุดเริ่มต้นของเฟรม และ จุดปลายของเฟรม ดังแสดงในรูปที่ 3.6 และ รูปที่ 3.7 โดยเมื่อนำสัญญาณเสียงนี้ไปแปลงฟูเรียร์ทรานฟอร์มจะทำให้สเปกตรัมความถี่มูลฐาน(fundamental frequency) และ ความถี่ฮาร์มอนิก (Harmonics) ของสัญญาณเสียงแสดงออกมาในลักษณะสัญญาณยอดแหลม (narrow peaks) จำนวนมากที่มีระยะห่างใกล้เคียงกัน ซึ่งจะเป็นผลตอบสนองทางความถี่ (Frequency response) แบบที่ต้องการ [11] [25]

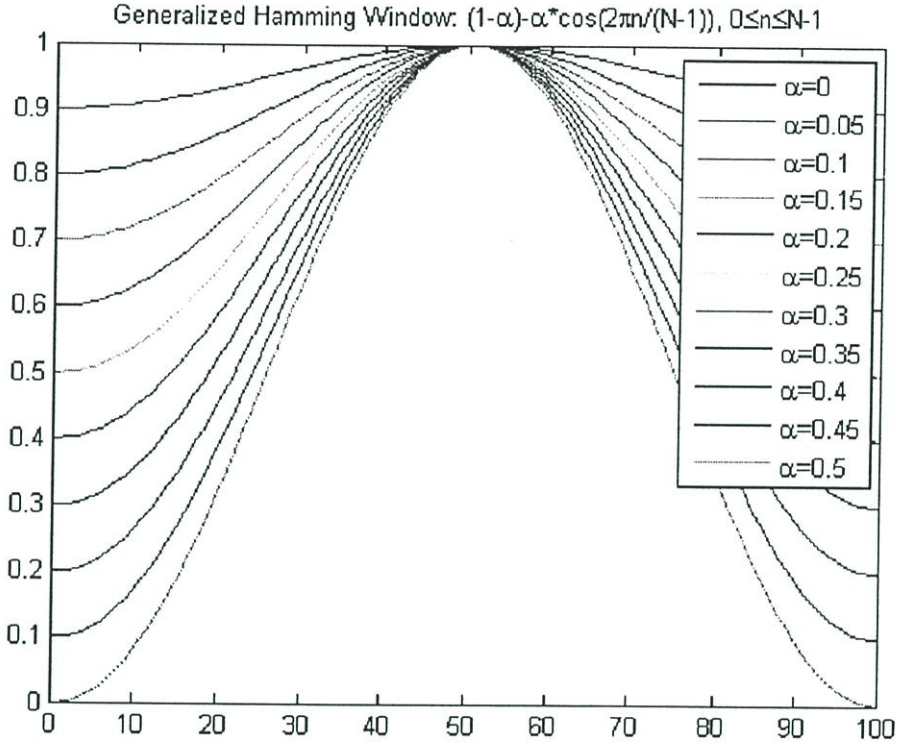


รูปที่ 3.6 การนำแฮมมิงวินโดว์คูณกับเฟรมสัญญาณเสียงอินพุต



รูปที่ 3.7 การตัดสัญญาณเสียงออกเป็นเฟรม (ก) สัญญาณเสียงอินพุต 1 เฟรม (ข) ผลลัพธ์หลังจากที่ใช้แฮมมิงวินโดว์ (Hamming window) คูณกับสัญญาณเสียงอินพุต

สมการแฮมมิงวินโดว์ (Hamming window) แสดงดังสมการที่ 3.4 โดยที่ N คือ ขนาดของวินโดว์ ซึ่งจะเท่ากันกับขนาดสัญญาณเสียงในแต่ละเฟรมย่อย และการเปลี่ยนแปลงค่าอัลฟ่า (Alpha) จะทำให้ความชันของแฮมมิงวินโดว์นั้นเปลี่ยนไปดังแสดงในรูปที่ 3.8 ซึ่งในวิทยานิพนธ์นี้จะกำหนดค่าอัลฟ่าให้เท่ากับ 0.46 ซึ่งค่าอัลฟ่า (Alpha) ที่เลือกใช้นี้เป็นค่าที่นิยมใช้โดยทั่วไปในการวิเคราะห์สัญญาณเสียงดังแสดงในสมการที่ 3.5



รูปที่ 3.8 แฮมมิงวินโดว์ที่การกำหนดค่าอัลฟ่าต่างๆ [11]

$$w(n) = (1-\alpha) - \alpha \cdot \cos\left(\frac{2\pi n}{N-1}\right) \qquad \text{เมื่อ } n=0,1,2\ldots,N-1 \qquad (3.4)$$

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \qquad \text{เมื่อ } n=0,1,2\ldots,N-1 \qquad (3.5)$$

3.3.2 การหาสเปกตรัมเชิงขนาดของสัญญาณเสียง (Power Spectrum)

เมื่อสัญญาณเสียงอินพุตถูกแบ่งออกเป็นเฟรม จากนั้นสัญญาณเสียงทุกเฟรมจะถูกนำมาหาค่าสเปกตรัมเชิงกำลัง (Power spectrum) หรือ สเปกตรัมเชิงแม็กนิจูด (Magnitude spectrum) โดยการแปลงดิสครีตฟูเรียร์ทรานฟอร์ม (Discrete Fourier Transform) หรือ เขียนตัวย่อว่า DFT เพื่อ

แปลงสัญญาณเสียงที่อยู่ในโดเมนทางเวลา (Time domain) ให้ไปอยู่ในโดเมนความถี่ (Frequency domain) ซึ่งเป็นโดเมนเดียวกันกับฟิลเตอร์แบงก์ ซึ่งเมื่อสัญญาณเสียงและฟิลเตอร์แบงก์อยู่ในโดเมนเดียวกันจะทำให้สามารถคูณกันได้โดยตรง ซึ่งการแปลงดิสครีตฟูเรียร์ทรานฟอร์มแสดงได้ดังสมการที่ 3.6 และ 3.7

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}} \tag{3.6}$$

$$S(k) = \left|X(k)\right| = \left|\sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}\right| \tag{3.7}$$

เมื่อกำหนดให้

x(n) คือ ลำดับสัญญาณเสียงทางเวลา

X(k) คือ สเปคตรัมความถี่ของสัญญาณเสียง x(n)

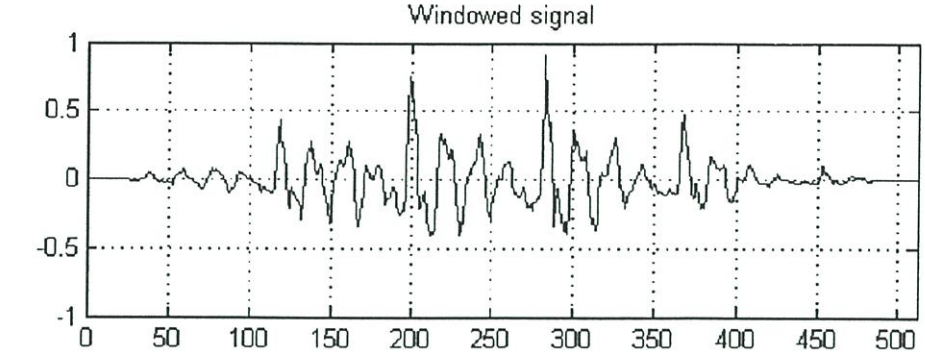
N คือ ขนาดของสัญญาณอินพุต

n คือ ตำแหน่งของสัญญาณเสียงอินพุตซึ่งจะมีค่าตั้งแต่ n = 0 ถึง N-1

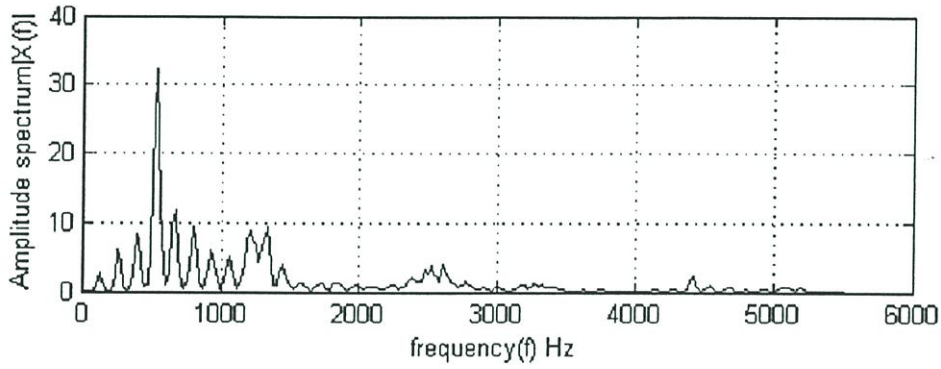
นอกจากนั้นการหาสเปกตรัมเชิงกำลังของสัญญาณเสียงยังสามารถหาได้ด้วยวิธี Autocorrelation ดังแสดงในสมการที่ 3.8 โดยที่ $X(k)$ คือ สัญญาณที่ผ่านการแปลงดิสครีตฟูเรียร์ทรานฟอร์ม และ $X^*(k)$ การทำคอมเพล็กคอนจูเกจของ $X(k)$

$$S(k) = X(k) \cdot X^*(k) \tag{3.8}$$

การแปลงดิสครีตฟูเรียร์ทรานฟอร์ม (Discrete Fourier Transform) เป็นวิธีการแปลงลำดับสัญญาณทางเวลาที่มีขนาดข้อมูล N จุดข้อมูล ให้ได้ค่าผลลัพธ์เป็นค่าสเปคตรัมทางความถี่ที่มีจำนวนจุดข้อมูลเท่ากับ N ตัว แต่การคำนวณ DFT แต่ละครั้งจะต้องมีการคูณตัวเลขทั้งสิ้น $4N^2$ ครั้ง และ มีการบวกตัวเลขอีก N(4N-2) ครั้ง [15] ดังนั้นจะเห็นว่าการแปลง DFT แต่ละครั้งจะต้องใช้เวลาในการคำนวณเพิ่มขึ้นเป็นสัดส่วนโดยตรงกับ N^2 ซึ่งถ้า N มีจำนวนมากการคำนวณในแต่ละครั้งจะต้องใช้เวลานานขึ้นตามไปด้วย ดังนั้น ในทางปฏิบัติจึงใช้การแปลงฟูเรียร์ทรานฟอร์มด้วยวิธี Fast fourier transform หรือเขียนตัวย่อว่า FFT ซึ่งจะทำให้การแปลงฟูเรียร์ทรานฟอร์มนั้นทำงานเร็วขึ้น โดยผลลัพธ์ของการแปลงดิสครีตฟูเรียร์ทรานฟอร์มเพื่อให้ได้สเปกตรัมเชิงกำลัง (Power spectrum) ของสัญญาณเสียงนั้นแสดงให้เห็นดังรูปที่ 3.9



(ก)

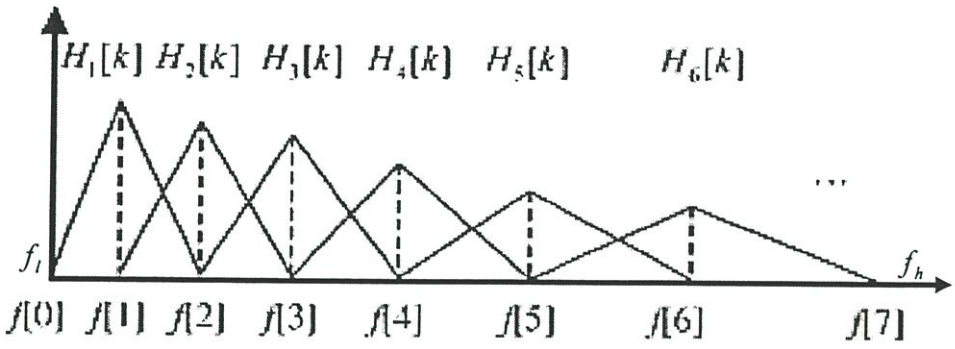


(ข)

รูปที่ 3.9 การแปลงดิสครีตฟูเรียร์ทรานฟอร์ม (ก) เฟรมสัญญาณเสียงอินพุต (ข) ผลลัพธ์ของการแปลงดิสครีตฟูเรียร์ทรานฟอร์ม

3.3.3 การหาค่าสเปกตรัมบนสเกลเมล (Mel Spectrum)

การหาค่าสเปกตรัมบนสเกลเมล คือ การเปลี่ยนสเกลทางความถี่ธรรมดาซึ่งเป็นสเกลเชิงเส้น (Linear frequency scale) ให้เป็นสเกลเมล (Mel scale) ซึ่งเป็นสเกลแบบไม่เป็นเชิงเส้น (Nonlinear frequency scale) ซึ่งสามารถทำได้โดยการนำสัญญาณเสียงที่ผ่านการแปลงดิสครีตฟูเรียร์ทรานฟอร์ม (DFT) คูณกับฟิลเตอร์รูปสามเหลี่ยม (Triangular Filters) โดยผลลัพธ์จะเป็นการคำนวณหาค่าเฉลี่ย (Average spectrum) รอบความถี่กลาง (Center frequency) ของฟิลเตอร์รูปสามเหลี่ยม [16] ดังแสดงในสมการที่ 3.9 ซึ่งฟิลเตอร์นี้เป็นแบนพาสฟิลเตอร์ (Bandpass filters) ซึ่งจะสามารถมีได้ตั้งแต่ 1 ฟิลเตอร์ไปจนถึง M ฟิลเตอร์ (m = 1,2,...M) และ ชุดฟิลเตอร์นี้จะถูกเรียกว่าฟิลเตอร์แบงก์ (Filterbank) ดังแสดงในรูปที่ 3.10 โดยในวิทยานิพนธ์นี้จะกำหนดให้มีฟิลเตอร์รูปสามเหลี่ยมในฟิลเตอร์แบงก์ M = 12 ซึ่งค่านี้เป็นเป็นค่าที่นิยมใช้ในงานทางด้านการรู้จำเสียง (Speech recognition) และจะทำให้สัมประสิทธิ์เซปตรัมบนสเกลเมลทางเอาท์พุตของวิธี MFCC มีสัมประสิทธิ์ในแต่ละเฟรม (N_{Mel}) เท่ากับ 12 ข้อมูล (C1 ถึง C12)



รูปที่ 3.10 ฟิลเตอร์รูปสามเหลี่ยมที่ใช้ในการคำนวณหาสัมประสิทธิ์เซปตรัมบนความถี่เมล

จากฟิลเตอร์รูปสามเหลี่ยมที่แสดงในรูปที่ 3.10 จะเห็นว่าฟิลเตอร์ทางซ้ายมือซึ่งอยู่บริเวณความถี่ต่ำจะมีความกว้างของแถบความถี่ (Bandwidth) ที่แคบกว่าฟิลเตอร์ที่อยู่ทางขวามือ และจะกว้างขึ้นเลื่อยๆ ในบริเวณช่วงความถี่ที่สูงขึ้น สาเหตุที่เป็นเช่นนั้นเนื่องจากต้องการให้สเปกตรัมเชิงกำลัง (Power spectrum) ในช่วงความถี่สูงและความถี่ต่ำมีขนาดที่ใกล้เคียงกัน โดยที่ f_l คือ ความถี่ต่ำสุดของฟิลเตอร์แบงก์ และ f_h คือ ความถี่สูงสุดของฟิลเตอร์แบงก์

$$H_m[k] = \begin{cases} 0 & , k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & , f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & , f[m] \le k \le f[m+1] \\ 0 & , k > f[m+1] \end{cases} \quad (3.9)$$

ผลลัพธ์ที่ผ่านฟิลเตอร์แต่ละตัวจะถูกนำไปผ่านฟังก์ชันลอการิทึมเพื่อให้สเปกตรัมเชิงกำลัง (Power spectrum) อยู่ในสเกลแบบลอการิทึม (log-energy) ดังแสดงในสมการที่ 3.10 โดยกำหนดให้ N คือ จำนวนข้อมูลสัญญาณเสียงที่ได้จากการแปลง DFT และ ตัวแปร m จะมีค่าตั้งแต่ 1 จนถึง N_{Mel}

$$S_{Mel}[m] = \log\left[\sum_{k=0}^{N-1} S[k] \cdot H_m[k]\right] \quad (3.10)$$

3.3.4 การหาค่าเซปสตรัมบนสเกลเมล (Mel Cepstrum)

การหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมลทำได้โดยนำสเปกตรัมเชิงกำลัง (Power spectrum) ที่อยู่ในสเกลแบบลอการิทึม (log-energy) คือ $S_{Mel}[m]$ มาผ่านการแปลงโคซายน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform: DCT) ดังแสดงในสมการที่ 3.11 และ 3.12 [14]

$$c[n] = dct(S_{Mel}[m]) \tag{3.11}$$

$$c[n] = \sqrt{\frac{2}{N_{Mel}}} \cdot \sum_{m=1}^{N_{Mel}} S_{Mel} \cos(\pi n(m - 0.5) / N_{Mel}) \qquad n = 1...C \tag{3.12}$$

เมื่อกำหนดให้

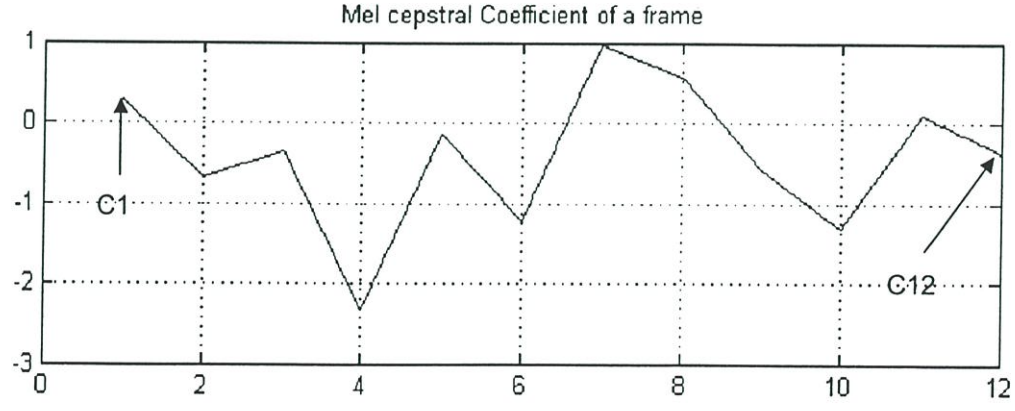
n = 1,..., C

C คือ จำนวนสัมประสิทธิ์เซปสตรัมเอาท์พุต

N_{Mel} = จำนวนของฟิลเตอร์รูปสามเหลี่ยม (Triangular Filters)

การแปลงโคซายน์แบบไม่ต่อเนื่อง (DCT) เป็นการแปลงแบบออทอโกนัลที่มีฟังก์ชันโคไซน์เป็นฐาน ซึ่งมักจะนำไปประยุกต์ใช้ในงานประมวลผลสัญญาณภาพ และ สัญญาณเสียง โดยเฉพาะอย่างยิ่งการเข้ารหัสข้อมูล และ การบีบอัดข้อมูลที่จะสามารถบีบอัดค่าพลังงานของสัญญาณเสียงส่วนใหญ่ไปไว้ในสัมประสิทธิ์ย่านความถี่ต่ำได้ [24]

เมื่อผ่านขั้นตอนการแปลงโคซายน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform) แล้วซึ่งผลลัพธ์ทางเอาท์พุตจะได้เป็นค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient) ซึ่งแต่ละเฟรมจะมีสัมประสิทธิ์เฟรมละ 12 ตัว คือ C1 ถึง C12 ดังแสดงในรูปที่ 3.11



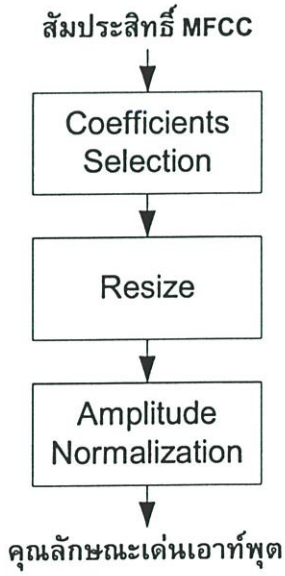
รูปที่ 3.11 สัมประสิทธิ์เซปตรัมเอาท์พุตของ 1 เฟรมสัญญาณเสียง

จากรูปที่ 3.11 เป็นการแสดงให้เห็นถึงสัมประสิทธิ์เซปตรัมบนสเกลเมล 1 เฟรม ซึ่งจะมีสัมประสิทธิ์จำนวน 12 ตัว คือ C1 ถึง C12 ดังนั้น สัญญาณเสียง 1 คำอาจจะมีจำนวนเฟรมเป็นร้อยเฟรมซึ่งข้อมูลคุณลักษณะเด่นของสัญญาณเสียงเหล่านี้มีมากเกินไป ดังนั้นจะต้องมีวิธีในการที่จะเลือกคุณลักษณะเด่นของสัญญาณเสียง (Feature selection) ที่จำเป็นหรือหน้าสนใจไปใช้ในการรู้จำด้วยโครงข่ายประสาทเทียมซึ่งมีจำนวนโหนดอินพุตที่จำกัด

3.4 การเลือกคุณลักษณะเด่นของสัญญาณเสียง

การเลือกคุณลักษณะเด่นของสัญญาณเสียง (Feature selection) คือ กระบวนการจัดเรียงข้อมูล และ ลดขนาดคุณลักษณะเด่นของสัญญาณเสียงลงอีกจากการที่ได้ลดข้อมูลไปแล้วในภาคการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) และ เพื่อให้ปริมาณข้อมูลนั้นมีขนาดพอดีกับทางอินพุตของระบบโครงข่ายประสาทเทียมที่ได้ออกแบบไว้

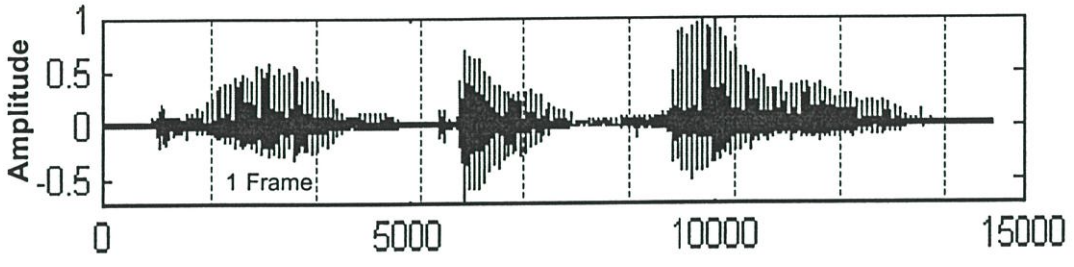
เมื่อได้ทำการดึงคุณลักษณะเด่นของสัญญาณเสียงคำพูด และ สัญญาณเสียงพยางค์แรกของคำพูดออกมาดังที่ได้กล่าวไปแล้วในหัวข้อที่ 3.3 จากนั้นคุณลักษณะเด่นของสัญญาณเสียงเหล่านี้จะถูกเลือก และ ถูกปรับขนาดใหม่ เพื่อให้มีขนาดตามที่ต้องการ โดยมีกระบวนการทำงานดังแสดงในรูปที่ 3.12



รูปที่ 3.12 อัลกอริทึมการเลือกคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction)

เนื่องจากสัมประสิทธิ์เซปสตรัม (Cepstral coefficient) ที่ได้จากภาคการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) นั้นมีจำนวนมากเกินกว่าที่จะใส่ลงในแบบจำลองการรู้จำได้ทั้งหมด ดังนั้น จึงมีกระบวนการเลือกข้อมูลที่สำคัญเท่านั้นไปใช้ในการรู้จำ โดยในกรณีการเลือก

คุณลักษณะเด่นของสัญญาณเสียงของคำพูดทั้งคำจะใช้สัมประสิทธิ์เซปสตรัมได้แก่ C1, C2, C3, C4 ของแต่ละเฟรมสัญญาณเสียง เนื่องจากสัมประสิทธิ์เหล่านี้มีการเปลี่ยนแปลงที่ชัดเจน และ มีขนาดแอมพลิจูดสูงกว่าสัมประสิทธิ์ตัวอื่นๆ โดยจะพิจารณาสัมประสิทธิ์ C1 ของทุกเฟรมมาเรียงต่อกันดังแสดงในรูปที่ 3.13



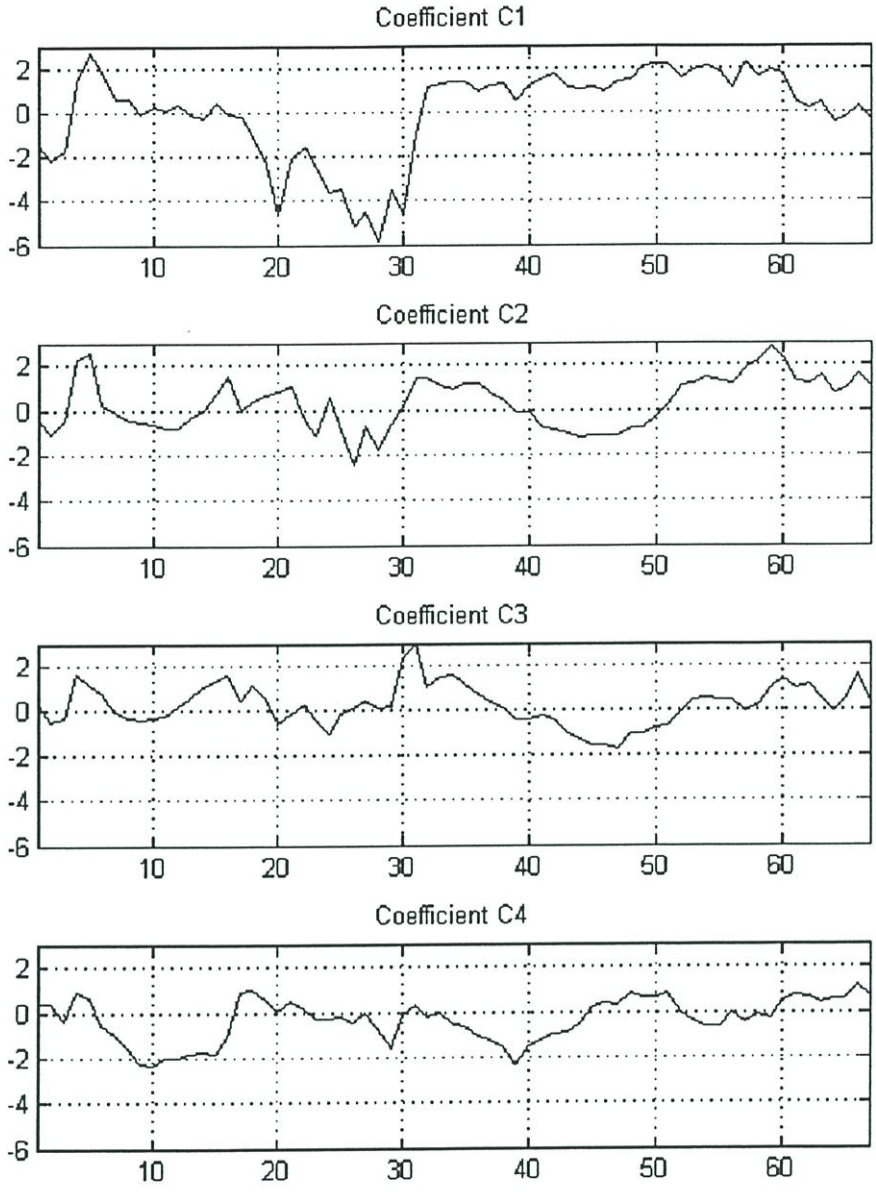
(ก)

(ข)

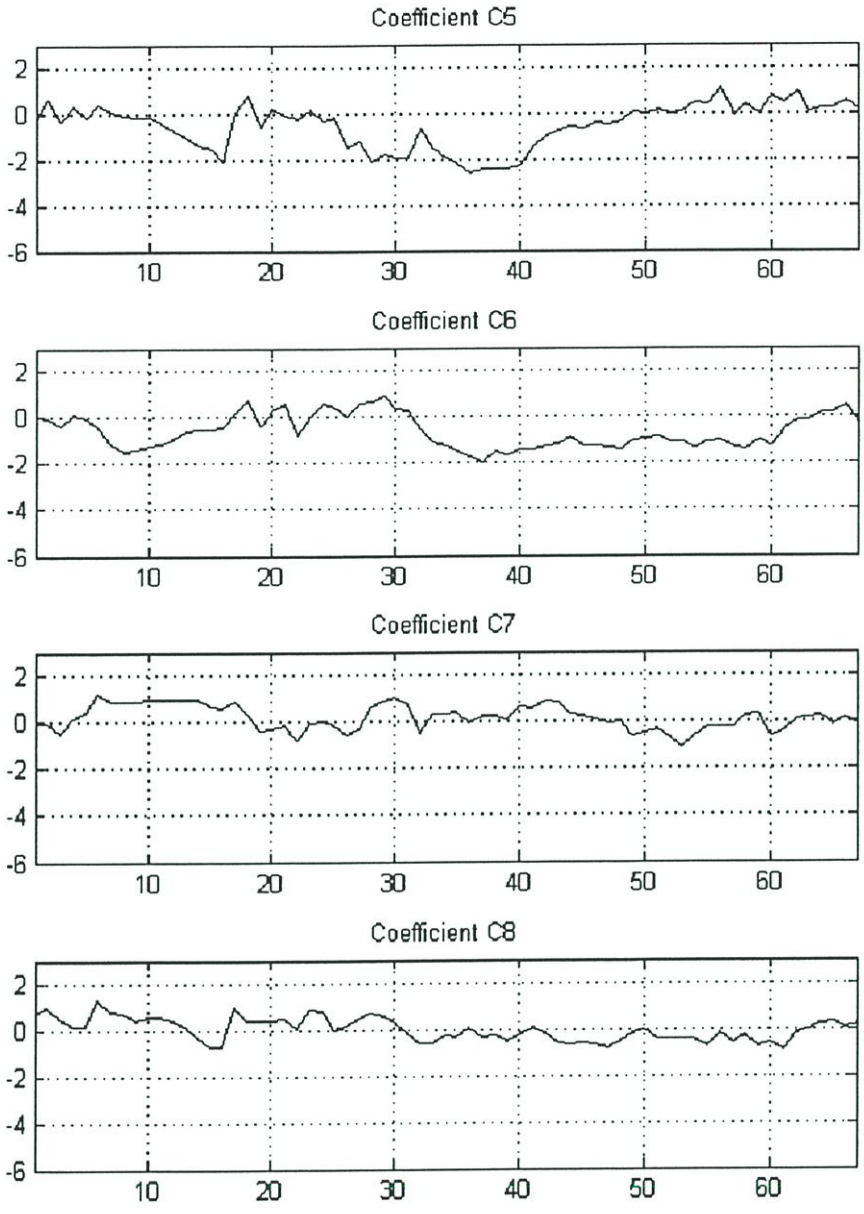
รูปที่ 3.13 การเลือกคุณลักษณะเด่นของคำพูด (ก) สัญญาณเสียงอินพุต (ข) สัมประสิทธิ์เซปสตรัมที่ถูกเรียงในแนวตั้งของแต่ละเฟรมของสัญญาณเสียง

จากรูปที่ 3.13 (ก) คือ สัญญาณเสียงอินพุตโดยเส้นปะที่แสดงหมายถึงเฟรมของสัญญาณเสียงที่ถูกแบ่งเพื่อใช้ในการหาค่าสัมประสิทธิ์เซปสตรัม (1เฟรมจะได้สัมประสิทธิ์เซปสตรัม 12 ตัว คือ C1 ถึง C12) รูปที่ 3.13 (ข) จากรูปเป็นการแสดงให้เห็นถึงสัมประสิทธิ์เซปสตรัมแต่ละเฟรมของสัญญาณเสียงที่วางเรียงกันในแนวตั้ง C1 ถึง C12 และ เส้นแรเงาแนวนอนทั้ง 4 เส้นคือคุณลักษณะเด่นของสัญญาณเสียงที่จะนำไปใช้ในขั้นตอนถัดไป

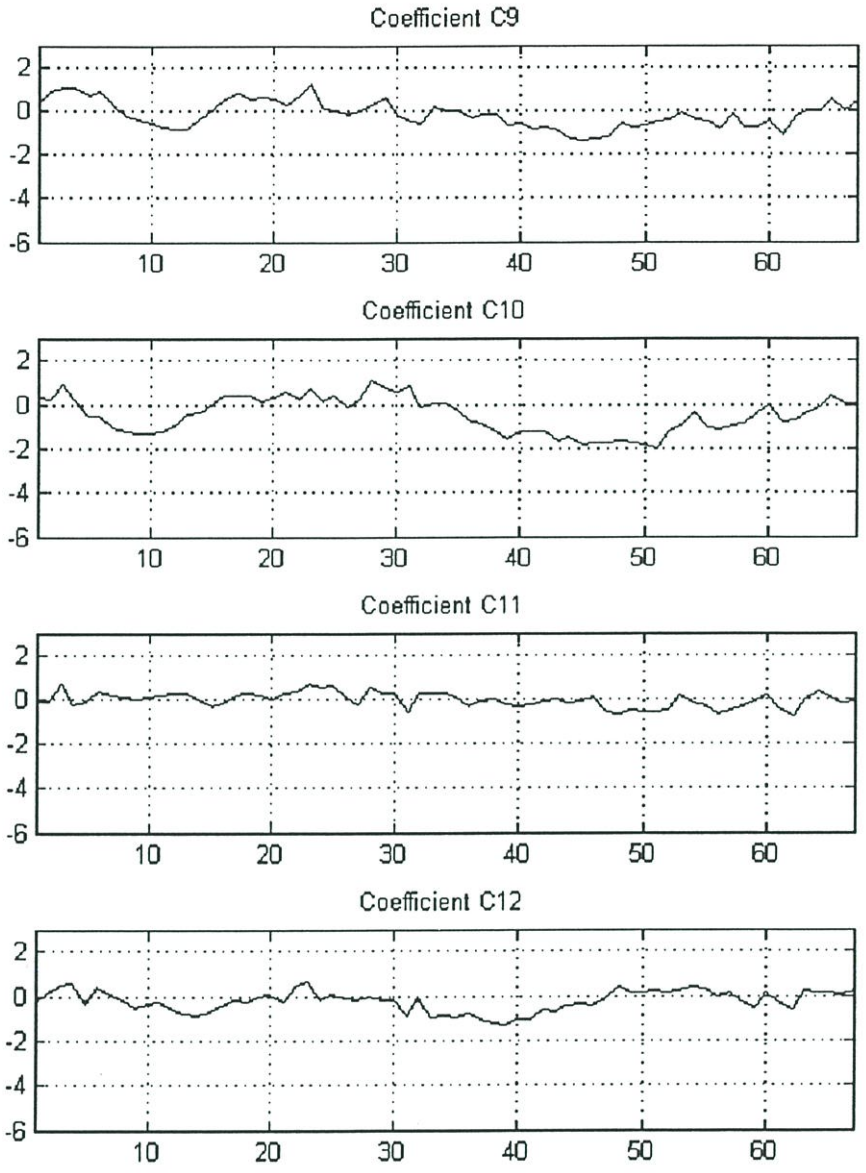
จากรูปที่ 3.12 บล็อก Resize จะทำการเรียงลำดับข้อมูล C1 ของทุกเฟรมเป็นข้อมูลใหม่ 1 ชุด และในทำนองเดียวกันกับ C2, C3 และ C4 ซึ่งจะได้ข้อมูลใหม่อย่างละ 1 ชุดเช่นกัน โดยการเรียงข้อมูลใหม่นี้จะได้เส้นแรเงาทั้งหมด 4 เส้นที่แสดงดังรูปที่ 3.13 (ข) และ สามารถนำข้อมูลคุณลักษณะเด่นที่ได้เรียงใหม่มาวาดกราฟได้ดังแสดงในรูปที่ 3.14 ถึงรูปที่ 3.16 โดยจากรูปทั้งสามเป็นการแสดงให้เห็นว่าสัมประสิทธิ์ตัวแรกๆ เช่น C1 หรือ C2 เมื่อพิจารณาตามเส้นแรเงาจะมีขนาดการเปลี่ยนแปลงที่มากกว่าค่าสัมประสิทธิ์ตัวท้ายๆ เช่น C11 หรือ C12



รูปที่ 3.14 สัมประสิทธิ์เซปสตรัม C1 ถึง C4 ของทุกเฟรมสัญญาณเสียง



รูปที่ 3.15 สัมประสิทธิ์เซปสตรัม C5 ถึง C8 ของทุกเฟรมสัญญาณเสียง



รูปที่ 3.16 สัมประสิทธิ์เซปสตรัม C9 ถึง C12 ของทุกเฟรมสัญญาณเสียง

จากนั้นข้อมูลทั้ง 4 ชุดนี้จะถูกย่อขนาดลงโดย C1 ถึง C3 จะย่อขนาดให้เหลือ 20 ข้อมูล และ C4 จะย่อให้เหลือ 10 ข้อมูล เมื่อนำ C1 ถึง C4 มาเรียงต่อกันจะได้ข้อมูลใหม่ 1 ชุดซึ่งมีข้อมูลรวมเท่ากับ 70 ข้อมูล ดังแสดงในรูปที่ 3.17

C1	C2	C3	C4
20 ข้อมูล	20 ข้อมูล	20 ข้อมูล	10 ข้อมูล

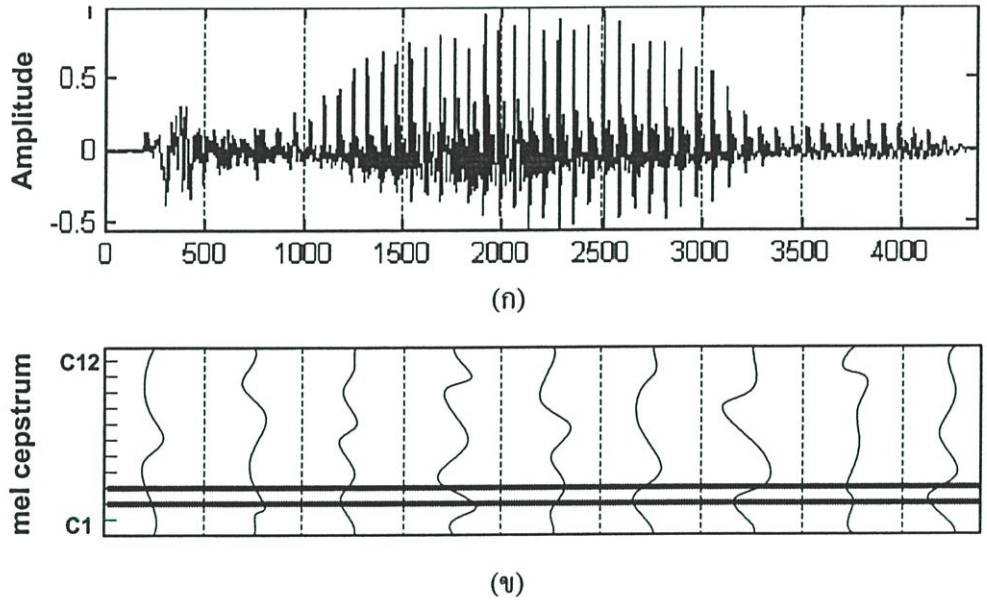
รูปที่ 3.17 การจัดเรียงข้อมูลคุณลักษณะเด่นของคำพูด

จากนั้นนำคุณลักษณะเด่นของสัญญาณเสียงนี้ไปนอร์มอลไลซ์ทางแอมพลิจูด (Amplitude normalization) เพื่อปรับระดับความสูงของสัญญาณเสียงให้อยู่ในช่วง 0 ถึง 1 ดังแสดงในสมการที่ 3.13 และ 3.14

$$f(n) = x(n) - \min(x) \tag{3.13}$$

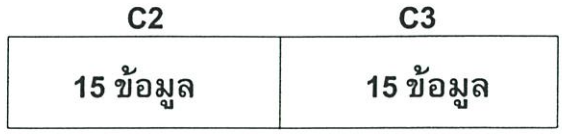
$$y(n) = \frac{f(n)}{|\max(f)|} \tag{3.14}$$

การหาสัมประสิทธิ์เซปสตรัมของสัญญาณเสียงเฉพาะพยางค์แรก นั้นจะมีขั้นตอนการเลือกคุณลักษณะเด่นเช่นเดียวกับขั้นตอนที่ได้ทำกับคำพูด (Word) ดังที่ได้กล่าวไปแล้ว เพียงแต่ในส่วนนี้จะเลือกใช้สัมประสิทธิ์เซปสตรัมได้แก่ C2 และ C3 ตามลำดับดังแสดงในรูปที่ 3.18



รูปที่ 3.18 การเลือกคุณลักษณะเด่นของพยางค์แรกของคำพูด (ก) สัญญาณเสียงอินพุต (ข) ค่าข้อมูลสัมประสิทธิ์เซปสตรัมที่ถูกเรียงในแนวตั้งของแต่ละเฟรมของสัญญาณเสียง

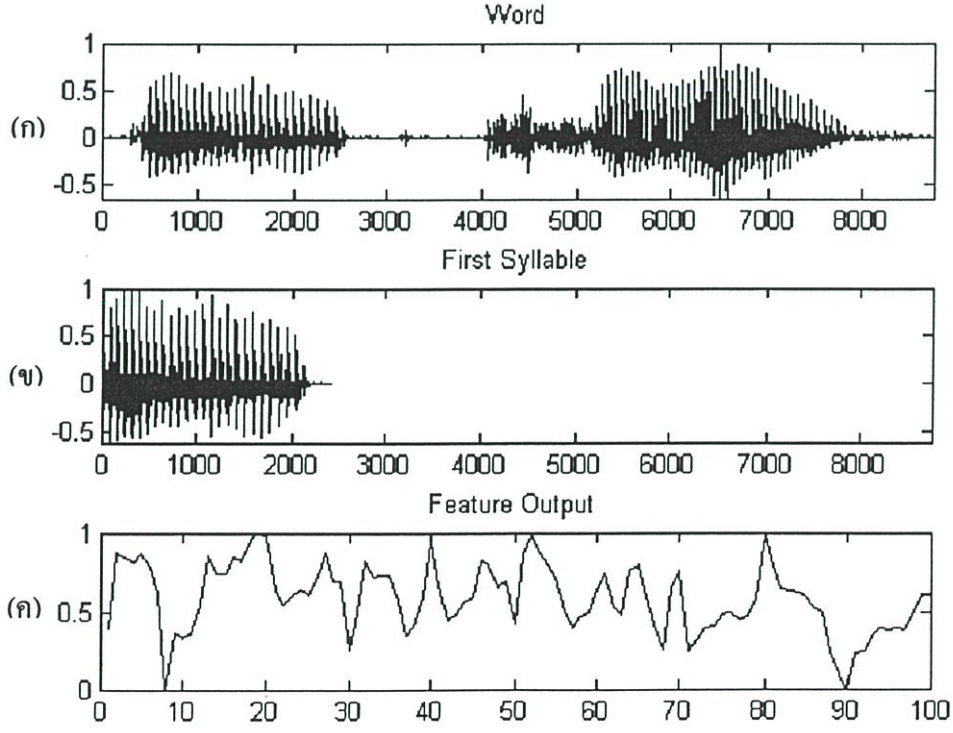
ทำการลดขนาดข้อมูลของสัมประสิทธิ์เซปสตรัม C2, และ C3 ให้เหลืออย่างละ 15 ข้อมูล จากนั้นนำสัมประสิทธิ์เซปสตรัมแต่ละตัวมาเรียงต่อกันผลรวมจะได้ 30 ข้อมูล ดังแสดงในรูปที่ 3.19 จากนั้นนำคุณลักษณะเด่นของสัญญาณเสียงนี้ไปนอร์มอลไลซ์ทางแอมพลิจูด (Amplitude normalization) เพื่อปรับระดับความสูงของสัญญาณเสียงให้อยู่ในช่วง 0 ถึง 1 ดังแสดงในสมการที่ 3.13 และ 3.14



รูปที่ 3.19 การจัดเรียงข้อมูลคุณลักษณะเด่นของพยางค์แรกของคำพูด

เมื่อได้คุณลักษณะเด่นของสัญญาณเสียงของคำพูด และ พยางค์แรกของคำพูดแล้ว จากนั้นจะนำคุณลักษณะเด่นทั้งสองส่วนนี้มารวมกันดังแสดงในรูปที่ 3.20 และ รูปที่ 3.21 ซึ่งจะเป็นข้อมูลอินพุตของระบบการรู้จำด้วยโครงข่ายประสาทเทียม

รูปที่ 3.20 การรวมคุณลักษณะเด่นของคำพูด และ พยางค์แรกของคำพูดเข้าด้วยกัน



รูปที่ 3.21 เอาท์พุตของภาคการเลือกคุณลักษณะเด่นของสัญญาณเสียง (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงเฉพาะพยางค์แรกของคำพูด (ค) คุณลักษณะเด่นของสัญญาณเสียง

เมื่อถึงตรงจุดนี้จะทำให้ได้คุณลักษณะเด่นของสัญญาณเสียง (Feature of signal) ซึ่งได้รวมคุณลักษณะเด่นของสัญญาณเสียงคำพูด และ สัญญาณเสียงเฉพาะพยางค์แรกของคำพูดไว้ด้วยกันแล้ว โดยจะมีจำนวนข้อมูลทั้งหมด 100 ข้อมูล ต่อ 1 สัญญาณเสียง ซึ่งข้อมูลคุณลักษณะเด่นนี้จะกลายเป็นสัญญาณอินพุตให้กับระบบการรู้จำด้วยโครงข่ายประสาทเทียมซึ่งจะมีโหนดอินพุตทั้งหมด 100 โหนดเท่ากับข้อมูลอินพุต

บทที่ 4
โครงข่ายประสาทเทียม

4.1 โครงข่ายประสาทเทียมคืออะไร

โครงข่ายประสาทเทียม (Artificial Neural Network, ANN) [17-20] คือแบบจำลองการประมวลผลข้อมูล ซึ่งมีแรงบันดาลใจมาจากระบบประสาททางชีววิทยา (Biological nervous systems) หรือ สมองของมนุษย์ ซึ่งมีการเรียนรู้จากข้อมูลที่ถูกส่งผ่านเข้าไปทางกิ่งก้านของเซลล์ประสาท (Dendrite) จากนั้นจะถูกคำนวณ และ ส่งผ่านให้กับเซลล์ประสาทอื่นๆ ที่เชื่อมโยงกันเป็นเครือข่ายผ่านทางแกนเซลล์ประสาท (Axon) ซึ่งระบบโครงข่ายประสาทเทียมมักจะนำไปประยุกต์ใช้กับงานเฉพาะทาง เช่น การรู้จำรูปแบบ (Pattern recognition) หรือ การจำแนกประเภทข้อมูล (Data classification) เป็นต้น

4.2 ทำไมถึงใช้โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมมีความสามารถที่โดดเด่นเป็นอย่างมาก คือ มันสามารถที่จะหาความหมายของข้อมูลที่มีลักษณะที่คลุมเครือ หรือ ไม่แน่ชัดได้ ดังนั้นมันจึงถูกนำไปใช้ในการแยกแยะ หรือ สกัดรูปแบบ (Extract patterns) ข้อมูลที่ซับซ้อนหรือใช้ตรวจหาแนวโน้มของข้อมูลซึ่งเป็นการยากที่จะใช้มนุษย์ หรือ เทคนิคการคำนวณอื่นๆ มาใช้ได้ ซึ่งโครงข่ายประสาทเทียมที่มีการสอนแล้ว (Trained) จะเป็นผู้เชี่ยวชาญที่สามารถคิดหรือตัดสินใจตามประเภทของข้อมูลที่มันได้เรียนรู้มาก่อนหน้านี้ได้

4.3 ความสามารถของโครงข่ายประสาทเทียมเทียบกับเครื่องคำนวณโดยทั่วไป

โครงข่ายประสาทเทียมมีวิธีการทำงานเพื่อแก้ไขปัญหาในงานต่างๆ ที่แตกต่างจากเครื่องคำนวณประเภทอื่นๆ ที่เคยมีใช้กันมา และมันสามารถจะแก้ไขปัญหาได้ดีกว่า [17] โดยเครื่องคำนวณธรรมดาจะมีวิธีคำนวณแบบที่ทำตามลำดับขั้นตอน (Algorithmic approach) กล่าวคือ เครื่องคำนวณจะทำตามชุดคำสั่งที่ได้ออกแบบไว้อย่างเป็นลำดับ เพื่อใช้ในการแก้ไขปัญหาหนึ่งๆ ซึ่งถ้านอกเหนือจากนี้ชุดคำสั่งจะไม่สามารถแก้ไขปัญหาได้เลย เพราะข้อจำกัดในการแก้ไขปัญหาของเครื่องคำนวณแบบธรรมดาที่เคยใช้กันมา ซึ่งจะต้องรู้วิธีการแก้ไขปัญหานั้นก่อนแล้วจึงนำมาเขียนเป็นชุดคำสั่ง หรือ โปรแกรมเพื่อใช้ป้องกันความผิดพลาดที่จะเกิดขึ้น

กระบวนการประมวลผลข้อมูลด้วยโครงข่ายประสาทเทียมจะคล้ายคลึงกับการทำงานของสมองมนุษย์ ซึ่งโครงข่ายจะมีจำนวนการเชื่อมต่อระหว่างเซลล์ประสาทจำนวนมาก และ มีการ

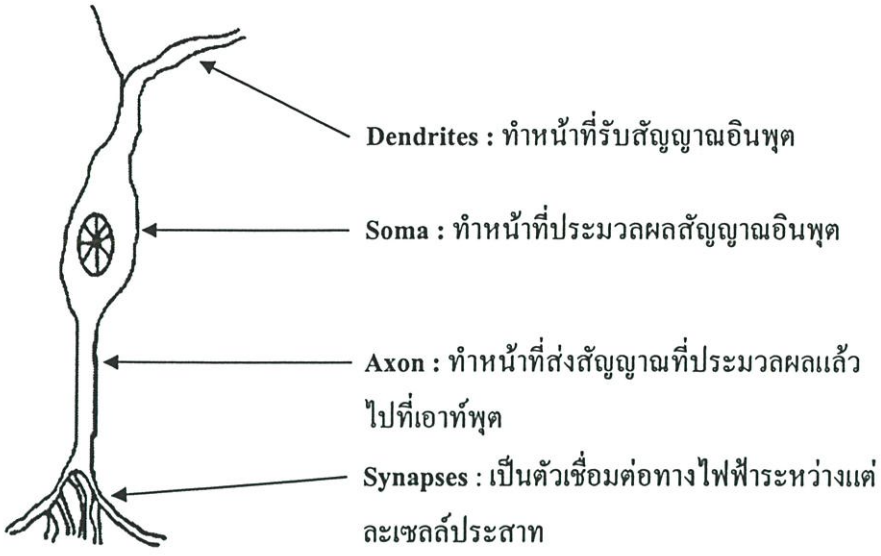
ทำงานแบบขนานในการแก้ไขปัญหาหนึ่งๆ ซึ่งโครงข่ายประสาทเทียมจะเรียนรู้ได้จากตัวอย่างข้อมูลอินพุตที่เข้ามา ซึ่งไม่สามารถที่จะโปรแกรมไว้ก่อนล่วงหน้าได้ โดยข้อมูลอินพุตที่ใช้สอนจะต้องถูกเลือกอย่างระมัดระวังมิฉะนั้นอาจจะทำให้ความสามารถของโครงข่ายนั้นแย่ลง หรือทำงานผิดพลาดได้ ซึ่งข้อเสียของโครงข่ายประสาทเทียม คือ โครงข่ายประสาทเทียมจะต้องหาวิธีการแก้ไขปัญหาด้วยตัวของมันเอง ซึ่งการทำงานของมันอาจจะทำงานผิดพลาดได้

ในทางตรงกันข้ามเครื่องคำนวณแบบธรรมดาจะต้องรู้วิธีการแก้ไขปัญหานั้นๆ ก่อนโดยการเขียนคำสั่งลงในเครื่องคำนวณ ซึ่งโปรแกรมคำสั่งนี้จะเป็นภาษาระดับสูง (High level language program) และ จากนั้นจะถูกแปลงเป็นภาษาเครื่อง (Machine code) ซึ่งเป็นภาษาที่เครื่องคำนวณเข้าใจ

4.4 สมองของมนุษย์มีการเรียนรู้ได้อย่างไร

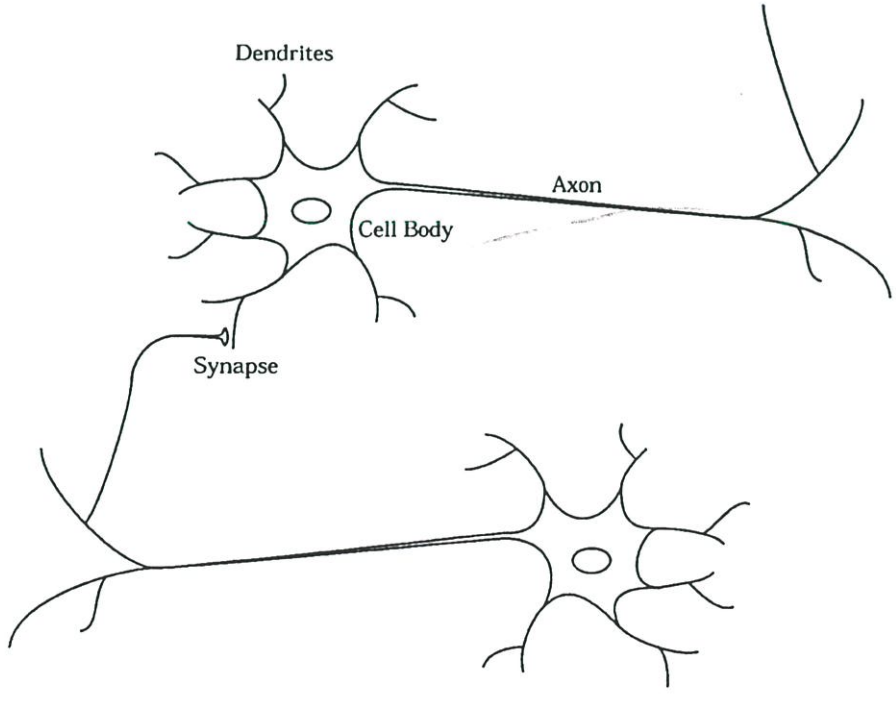
คนส่วนใหญ่ไม่ทราบว่าสมองมีกระบวนการฝึกฝนตัวเองได้อย่างไร ประกอบกับในทางทฤษฎีนั้นมีเนื้อหาที่มากและยากแก่การเรียนรู้ ซึ่งโดยปกติเซลล์ประสาทหนึ่งๆ ภายในสมองของมนุษย์จะประกอบไปด้วย 1. ช่องว่างระหว่างเซลล์ประสาท หรือ Synapse 2. กิ่งก้านของเซลล์ประสาท หรือ Dendrite 3. ลำตัวของเซลล์ประสาท หรือ Cell body และ 4. แกนเซลล์ประสาท หรือ Axon โดยเซลล์ประสาทหนึ่งเซลล์จะรับสัญญาณที่มาจากเซลล์ประสาทอื่นผ่านทางกิ่งก้านของเซลล์ประสาท (Dendrite) จากนั้นเซลล์ประสาทนี้จะส่งสัญญาณไฟฟ้าปลายแหลมที่ถูกสร้างขึ้นจากลำตัวของเซลล์ประสาท (Cell body) ผ่านทางแกนเซลล์ประสาท (Axon) ออกไปให้กับเซลล์ประสาทอื่นหลายพันเซลล์ ซึ่งที่ปลายของแกนเซลล์ประสาท (Axon) จะเรียกว่าช่องว่างระหว่างเซลล์ประสาท (Synapse) ซึ่งช่องว่างระหว่างเซลล์ประสาทจะทำหน้าที่แปลงจากสัญญาณไฟฟ้าในแกนเซลล์ประสาทให้เป็นสัญญาณทางเคมี และ สามารถแปลงกลับไปกลับมาได้ โดยสามารถแสดงดังรูปที่ 4.1 และ รูปที่ 4.2

รูปที่ 4.1 ลักษณะสมองของมนุษย์



รูปที่ 4.2 โครงข่ายประสาททางชีววิทยา

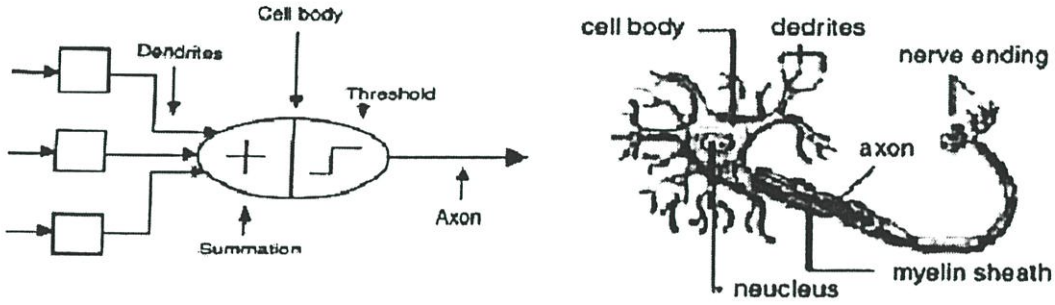
ในส่วนของการเรียนรู้ หรือ Learning จะเกิดขึ้นเมื่อมีผลการเปลี่ยนแปลงที่บริเวณช่องว่างระหว่างเซลล์ประสาท (Synapse) ซึ่งในรูปที่ 4.3 จะเป็นการแสดงให้เห็นถึงเซลล์ประสาทในทางชีววิทยา 2 เซลล์ที่เชื่อมต่อถึงกัน



รูปที่ 4.3 โครงข่ายสมองทางชีววิทยาที่มีการเชื่อมต่อกัน 2 เซลล์

4.5 จากโครงข่ายสมองมนุษย์ไปสู่ระบบโครงข่ายประสาทเทียม

โครงข่ายประสาททางชีววิทยา หรือ โครงข่ายประสาทที่มีอยู่ในสมองของมนุษย์ที่ได้กล่าวไปแล้วในหัวข้อที่ผ่านมา เมื่อทำการลดรายละเอียดการทำงานของมันลง และ นำเฉพาะการทำงานที่สำคัญเท่านั้นมาเขียนเป็นโปรแกรมเพื่อจำลองการทำงาน ซึ่งอย่างไรก็ตามเนื่องจากความรู้เกี่ยวกับเซลล์ประสาททางชีววิทยานั้นยังไม่สมบูรณ์ และ ยังมีข้อจำกัดในเรื่องของการคำนวณ จึงทำให้แบบจำลอง (Model) ที่สร้างขึ้นนั้นยังมีการทำงานที่หยาบ หรือ ไม่ละเอียดเมื่อเทียมกับโครงข่ายประสาทของมนุษย์ ซึ่งรูปที่ 4.4 จะเป็นการอธิบายโครงข่ายประสาททางชีววิทยาเปรียบเทียบกับโครงข่ายประสาทเทียม



(ก) (ข)

รูปที่ 4.4 โครงข่ายประสาทเทียมหนึ่งเซลล์ (ก) โครงข่ายประสาทเทียม (ข) โครงข่ายประสาททางชีววิทยา

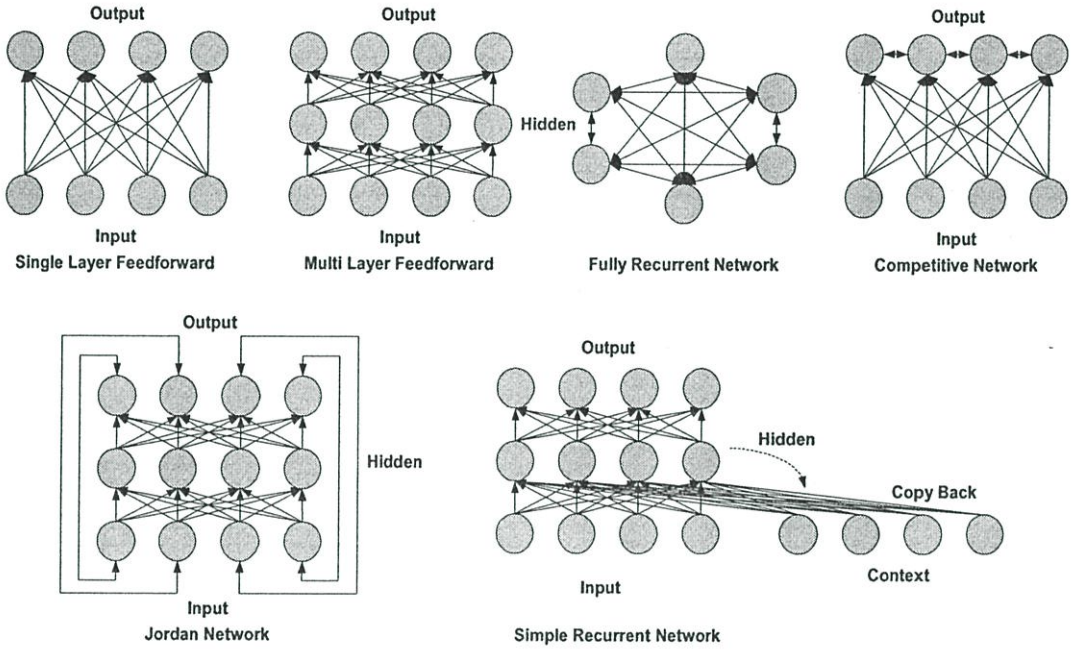
การทำความเข้าในในโครงข่ายประสาทเทียม (Artificial Neural Networks) หรือ การสร้างโครงข่ายประสาทเทียมพื้นฐานนั้นจะมี 3 ส่วนที่ต้องพิจารณา [18] [19] คือ

1. สถาปัตยกรรมของโครงข่ายประสาท (Network architecture)
2. การกำหนดค่าถ่วงน้ำหนัก (Setting the weights) ด้วยกระบวนการเรียนรู้ (learning)
3. ฟังก์ชันกระตุ้น (Activation function)

4.5.1 สถาปัตยกรรมของโครงข่ายประสาท (Neural Architecture)

การออกแบบโครงข่ายประสาทเทียม คือ การกำหนดจำนวนเซลล์ประสาทภายในชั้นของโครงข่าย (Layers) หรือ การกำหนดรูปแบบการเชื่อมต่อกันระหว่างแต่ละเซลล์ในแต่ละชั้น ซึ่งทั้งหมดนี้ถูกเรียกว่า สถาปัตยกรรมของโครงข่าย (Architecture of the net) โดยทุกการเชื่อมต่อระหว่างเซลล์ในแต่ละชั้นของโครงข่ายจะมีค่าถ่วงน้ำหนัก (Weight) คูณอยู่ด้วย โดยถ้าโครงข่ายมีจำนวน 2 ชั้น หรือ มากกว่า 2 ชั้น ซึ่งชั้นที่อยู่ตรงกลางระหว่างชั้นอินพุต (Input layer) กับชั้น

เอาท์พุต (Output layer) จะถูกเรียกว่าชั้นซ่อน หรือ Hidden layer (หนังสือบางเล่มจะไม่นับชั้นอินพุต (Input layer) เป็นชั้นที่ 1 แต่จะนับชั้นซ่อนที่ 1 เป็นชั้นแรกแทน เนื่องจากชั้น Input layer จะไม่มีการคำนวณ) นอกจากนั้นสถาปัตยกรรมโครงข่ายประสาทเทียมยังมีหลายรูปแบบด้วยกัน ตัวอย่างเช่น Feed forward, Feedback, Fully interconnected net, Competitive net และ อื่นๆ ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 สถาปัตยกรรมโครงข่ายประสาทเทียม

โครงข่ายชนิด Feed forward networks จะมีทั้งแบบหนึ่งชั้น (Single layer) และ แบบหลายชั้น (Multiple layer) โดยโครงข่ายชนิดหนึ่งชั้น (Single layer) ชั้นอินพุตจะถูกต่อตรงไปที่ชั้นเอาท์พุต โดยตรงในขณะที่โครงข่ายแบบหลายชั้น ซึ่งชั้นอินพุตจะถูกต่อไปที่ชั้นซ่อนและชั้นซ่อนจะถูกต่อไปที่ชั้นเอาท์พุต

โครงข่ายชนิด Competitive net จะมีความคล้ายคลึงกับโครงข่ายชนิด Single-layered feed forward network แต่จะแตกต่างตรงที่จะมีการลบกันระหว่างโหนดภายในชั้นเอาท์พุต (Output layer) สำหรับโครงข่ายชนิด Recurrent net ซึ่งทุกโหนดในโครงข่ายของมันจะถูกเชื่อมต่อถึงกันหมดทุกโหนด และทุกๆ โหนดจะเป็นได้ทั้งโหนดอินพุต และ โหนดเอาท์พุต โดยลักษณะโครงข่ายแบบนี้จะมีการประมวลผลข้อมูลแบบต่อเนื่องกันไป ซึ่งการประมวลผลในโครงข่ายแบบ Recurrent net นี้จะขึ้นอยู่กับสถานะของโครงข่ายที่เวลาผ่านล่าสุด ดังนั้น ผลตอบสนองของข้อมูลอินพุตปัจจุบันจะขึ้นอยู่กับอินพุตก่อนหน้า

4.5.2 การกำหนดค่าถ่วงน้ำหนัก (Setting the weights)

วิธีในการกำหนดค่าถ่วงน้ำหนัก (Weights) ให้กับโครงข่ายประสาทเทียมเพื่อทำให้ผลลัพธ์ทางเอาท์พุตนั้นตรงกับข้อมูลอินพุตที่ต้องการ ซึ่งกระบวนการในการกำหนดค่าถ่วงน้ำหนักนี้ คือ การเรียนรู้ (Learning) หรือ การสอน (Training) ซึ่งกระบวนการเรียนรู้มี 2 ประเภทหลักๆ คือ

1. Supervised Training คือ การเรียนรู้แบบมีผู้สอน ซึ่งการสอนแบบนี้จะต้องมีคู่ข้อมูลของการปรับสอน คือ ข้อมูลอินพุต และ ข้อมูลเป้าหมาย (Target) โดยค่าเอาท์พุตที่คำนวณได้จากข้อมูลอินพุตจะถูกเปรียบเทียบกับข้อมูลเป้าหมาย (Target) โดยกระบวนการปรับสอนจะต้องกำหนดเงื่อนไขการหยุดการปรับสอนว่าจะใช้วิธีใด เช่น การนับจำนวนรอบการปรับสอนว่าจะกำหนดกี่รอบ หรือ อีกวิธีหนึ่งคือ การพิจารณาจากจำนวนความผิดพลาด (Error) ที่ได้จากการเปรียบเทียบระหว่างค่าเอาท์พุตที่คำนวณได้จากข้อมูลอินพุต กับ ข้อมูลเป้าหมาย (Target) ซึ่งถ้าค่าความผิดพลาด (Error) นี้น้อยกว่าค่าที่กำหนดถึงจะหยุดการสอน และ วิธีการหาค่าความผิดพลาดมีหลายชนิดด้วยกันเช่น Least mean square (LMS) หรือ Mean square error (MSE) เป็นต้น และวิธีการเรียนรู้แบบ Supervised training ที่มีการใช้งานทั่วไปได้แก่ Hebb, Back propagation เป็นต้น ซึ่งในวิทยานิพนธ์นี้จะใช้กระบวนการเรียนรู้แบบ Supervised Training และ ใช้เงื่อนไขการหยุดสอนแบบนับจำนวนรอบ

2. Unsupervised Training คือ โครงข่ายที่มีการเรียนรู้แบบไม่มีผู้สอน หรือ การเรียนรู้ด้วยตัวเอง (Self-learning networks) ซึ่งการสอนแบบนี้ต้องการข้อมูลอินพุตแต่เพียงอย่างเดียว โดยที่ไม่จำเป็นต้องกำหนดข้อมูลเป้าหมาย (Target) ขึ้นมาก่อน ซึ่งโครงข่ายการเรียนรู้แบบนี้จะมีกลไกการทำงานที่เหมือนกับสมองมนุษย์ กล่าวคือ โครงข่ายมีการเรียนรู้ และ ทำงานไปพร้อมๆ กันในเวลาเดียวกัน หรือมีการทำงานแบบ On-line ในขณะที่การสอนแบบ Supervised training จะต้องทำการสอนระบบด้วยค่าข้อมูลอินพุต และ ค่าคำตอบที่ต้องการ (Target value) จากนั้นจึงกลับมาทำงานโหมดปกติ ซึ่งการทำงานแบบนี้เรียกว่าเป็นทำงานแบบ Off-line

ปกติวิธีการสอนแบบ Unsupervised training จะมีความซับซ้อน และ สร้างยากกว่าการสอนระบบแบบ Supervised training ซึ่งวิธีการสอนระบบแบบไม่มีผู้สอน (Unsupervised Training) มีหลายชนิดด้วยกัน ได้แก่ Hebbian training, Competitive training เป็นต้น

4.5.3 ฟังก์ชันกระตุ้น (Activation function)

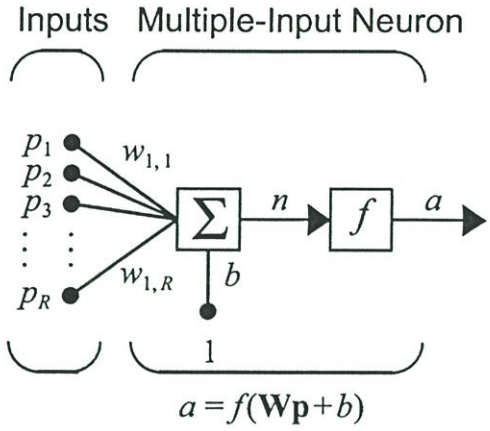
ฟังก์ชันกระตุ้น (Activation function) หรือ ในหนังสือบางเล่มใช้คำว่า Transfer function ซึ่งฟังก์ชันกระตุ้นนี้ใช้คำนวณผลตอบสนองทางเอาท์พุต (Output response) ของแต่ละเซลล์ประสาท โดยฟังก์ชันนี้มีทั้งที่ให้ผลการตอบสนองเป็นแบบเชิงเส้น (Linear) และ เป็นแบบที่ไม่เป็นเชิงเส้น (Nonlinear) ซึ่งการเลือกใช้งานจะแล้วแต่ความเหมาะสม ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ฟังก์ชันกระตุ้นของโครงข่ายประสาทเทียมชนิดต่างๆ

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons		compet

4.6 โครงข่ายแบบเพอร์เซปตรอน (Single Layer Perceptron)

โครงข่ายเพอร์เซปตรอนเป็นโครงข่ายชนิด 1 ชั้น (Single-layer perceptron) และ เป็นโครงข่ายที่ง่ายที่สุดในการนำไปใช้ในการแยกประเภทสิ่งต่างๆ แบบเชิงเส้น หรือ ใช้ในกระบวนการรู้จำ ซึ่งโครงข่ายอย่างง่ายที่สุดจะประกอบไปด้วย 1 เซลล์ประสาทที่สามารถปรับค่าถ่วงน้ำหนักได้ (Weight) และ ค่าไบอัส (Bias) ดังแสดงในรูปที่ 4.6 เมื่อโครงข่ายถูกสอนแล้วจะสามารถแยกแยะสิ่งต่างๆ ออกเป็น 2 ประเภทได้ ซึ่งนี่คือข้อจำกัดของการทำงานแบบ 1 เซลล์ [19]



รูปที่ 4.6 เซลล์ประสาทแบบหลายอินพุต

จากรูปที่ 4.6 เป็นการแสดงให้เห็นถึงโครงข่ายประสาท 1 เซลล์แบบหลายอินพุต โดยตัวแปร R คือ จำนวนโหนดอินพุต ซึ่งทุกโหนดในชั้นอินพุต (โดยปกติชั้นอินพุตจะไม่ถูกนับเป็น 1 ชั้น) จะถูกเชื่อมต่อถึงกันทั้งหมดกับชั้นเอาท์พุต โดยหนึ่งเซลล์จะมีไบอัส (Bias : b) หนึ่งตัวซึ่งปกติจะมีค่าเป็น '1' และ จะถูกบวกรวมไปกับผลคูณระหว่างข้อมูลอินพุตกับค่าถ่วงน้ำหนัก ซึ่งสามารถอธิบายเป็นสมการได้ดังสมการที่ 4.1

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \cdots + w_{1,R}p_R + b \tag{4.1}$$

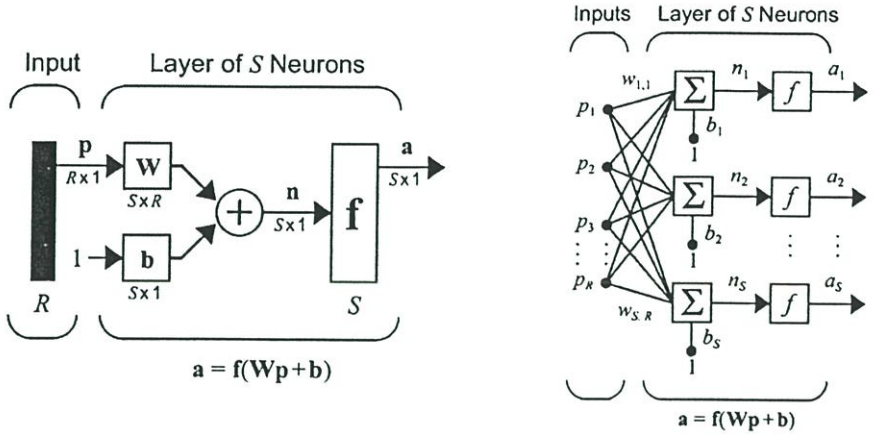
จากสมการที่ 4.1 สามารเขียนให้อยู่ในรูปเมตริกได้ดังดังสมการที่ 4.2

$$n = Wp + b \tag{4.2}$$

เมื่อเมตริก W คือ เมตริกของค่าถ่วงน้ำหนัก และ เมตริก p คือ เมตริกข้อมูลอินพุต จากนั้นสมการเอาท์พุตสามารถแสดงได้ดังสมการที่ 4.3

$$a = f(Wp + b) \tag{4.3}$$

เมื่อ f คือ ฟังก์ชันกระตุ้น (Activation function) ซึ่งมีหลายชนิดด้วยกันดังแสดงในตารางที่ 4.1 สำหรับโครงข่ายที่มีจำนวนเซลล์ประสาทจำนวนมากการวาดผังโครงข่ายจะแสดงดังรูปที่ 4.7



(ก) (ข)

รูปที่ 4.7 โครงข่ายประสาทเทียมแบบ 1 ชั้น (ก) การเขียนโครงข่ายแบบย่อ (ข) การเขียนโครงข่ายแบบแสดงการเชื่อมโยงทั้งหมด

ในการออกแบบโครงข่ายที่มีจำนวนโหนดอินพุต และ จำนวนชั้น (Layer) มากๆ การเขียนแผนภาพดังแสดงในรูปที่ 4.7 (ข) จะทำให้ภาพโครงข่ายนั้นดูซับซ้อนยุ่งเหยิงไปด้วยเส้นการเชื่อมต่อ ดังนั้น จึงสามารถวาดแผนภาพใหม่ได้ดังแสดงในรูปที่ 4.7 (ก) ซึ่งจะเป็นการอธิบายแผนภาพด้วยตัวแปรเมตริกโดยจะทำให้จำนวนเส้นการเชื่อมต่อและจำนวนตัวแปรที่ต้องใส่ลงในแผนภาพนั้นน้อยลง

จากรูปที่ 4.7 (ก) ตัวแปรเมตริกอินพุต p จะมีขนาดเท่ากับ Rx1 โดย R คือ จำนวนโหนดอินพุต ส่วนเมตริก W จะมีขนาด SxR โดย S คือ จำนวนเซลล์ประสาทใน 1 ชั้น และ เมตริก b จะมีขนาด Sx1 ซึ่งสามารถแสดงดังสมการที่ 4.4

$$a = f\left(\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \vdots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} + b\right) \tag{4.4}$$

4.7 กฎการเรียนรู้แบบเพอร์เซปตรอน (Percepron learning rule)

การสอนระบบเพื่อให้ระบบเริ่มการเรียนรู้จะเริ่มจากกำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนักและ ค่าไบอัสให้เป็นศูนย์ ซึ่งอันที่จริงเทคนิค หรือ วิธีการกำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนักมีหลายวิธีด้วยกัน เช่น Fuzzy systems, Genetic Algorithm และ อื่นๆ ซึ่งมันจำเป็นสำหรับการกำหนดค่าอัตราการเรียนรู้ (Learning rate) ซึ่งค่านี้จะอยู่ในช่วง 0 ถึง 1 จากนั้นข้อมูลอินพุตจะถูก

คำนวณโดยการนำค่าถ่วงน้ำหนักคูณกับข้อมูลอินพุต และ นำผลลัพธ์นี้บวกกับค่าไบอัส ซึ่งเมื่อข้อมูลอินพุตถูกคำนวณแล้วจะผ่านฟังก์ชันกระตุ้น (Activation function) และ จะได้ค่าเอาท์พุตของโครงข่ายออกมา จากนั้นค่าเอาท์พุตนี้จะถูกเปรียบเทียบกับค่าข้อมูลเป้าหมาย (Target) ซึ่งถ้า 2 ค่ามีความแตกต่างระบบจะทำการปรับค่าถ่วงน้ำหนักโดยใช้กฎการเรียนรู้แบบเพอร์เซปตรอน (Perceptron learning rule) ซึ่งอัลกอริทึมนี้สามารถใช้ได้กับข้อมูลที่พุตที่เป็น Binary และ Bipolar โดยขั้นตอนการสอนระบบมีขั้นตอนดังนี้ คือ

ขั้นที่ 1 : กำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนัก และ ไบอัส โดยปกติจะกำหนดให้เป็นศูนย์ และ กำหนดพารามิเตอร์อัตราการเรียนรู้ (Learning rate, α) ให้อยู่ในช่วง 0 ถึง 1

ขั้นที่ 2 : ในกรณีที่เงื่อนไขการหยุดสอนนั้นยังไม่ถูกต้องจะต้องทำขั้นตอนที่ 3 ถึง 7

ขั้นที่ 3 : ในการเปรียบเทียบข้อมูลระหว่างอินพุตกับข้อมูลเป้าหมายจะทำขั้นตอนที่ 4 ถึง 6

ขั้นที่ 4 : กำหนดชนิดของฟังก์ชันกระตุ้น (Activation function)

ขั้นที่ 5 : นำผลคูณระหว่างข้อมูลอินพุตกับค่าถ่วงน้ำหนักไปเข้าฟังก์ชันกระตุ้นดังแสดงในสมการที่ 4.5 และ สมการที่ 4.6

$$n = Wp + b \tag{4.5}$$

$$a = f(Wp + b) \tag{4.6}$$

ขั้นที่ 6 : ค่าถ่วงน้ำหนัก (Weight) และ ค่าไบอัส (bias) จะถูกเปลี่ยนค่าใหม่ (Update) ถ้าข้อมูลเป้าหมายไม่เท่ากับข้อมูลทางเอาท์พุตที่คำนวณได้ ดังแสดงในรูปที่ 4.8

If $t \neq a$ และค่าของอินพุต p_R ไม่เท่ากับศูนย์

$$W_{1,R(new)} = W_{1,R(old)} + \alpha \cdot t \cdot p_R$$

$$b_{(new)} = b_{(old)} + \alpha \cdot t$$

else

$$W_{1,R(new)} = W_{1,R(old)}$$

$$b_{(new)} = b_{old)}$$

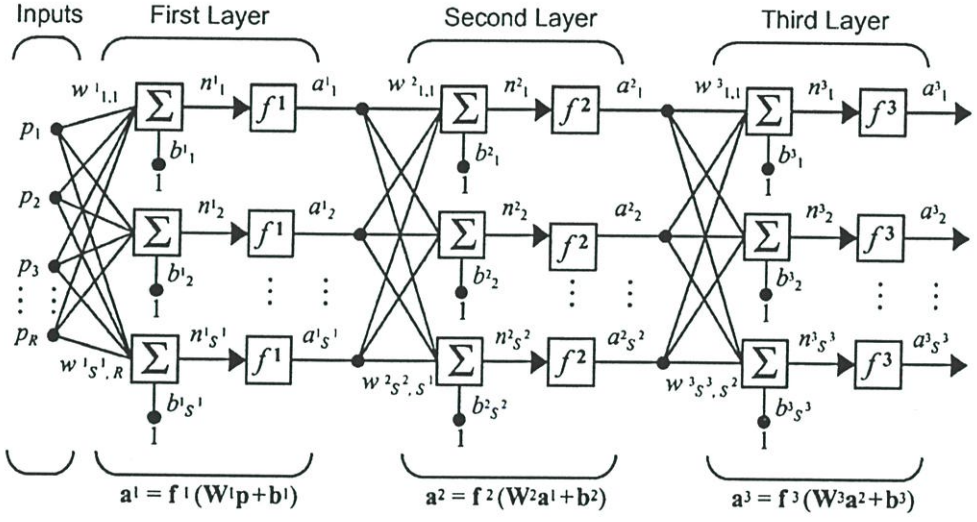
end

รูปที่ 4.8 คำสั่งเงื่อนไขในการปรับค่าถ่วงน้ำหนัก

ขั้นที่ 7 : ทดสอบเงื่อนไขเพื่อจบการสอนระบบ

4.8 โครงข่ายแบบเพอร์เซปตรอนหลายชั้น (Multilayer Perceptrons)

โครงข่ายประสาทเทียมแบบ Multilayer Perceptrons หรือ MLP คือ โครงข่ายประสาทเทียมแบบมีหลายชั้น ซึ่งแต่ละชั้นจะมีเมตริกค่าถ่วงน้ำหนัก (Weight matrix) แทนด้วยตัวอักษร W และ มีเวคเตอร์ไบอัส (Bias vector) แทนด้วยตัวอักษร b เป็นของตัวเองในแต่ละชั้น (Layer) โดยโครงข่ายประสาทเทียมประเภทนี้สามารถจำแนกรูปแบบของข้อมูลอินพุตในแบบไม่เป็นเชิงเส้น (Nonlinear separable) ได้ซึ่งโครงข่ายประสาทเทียมประเภทนี้จะมีวิธีการเรียนรู้แบบแพร่กลับ (Backpropagation algorithm) โดยชั้นที่หนึ่งจะเรียกว่า ชั้นอินพุต (Input layer) และ ชั้นสุดท้ายจะเรียกว่า ชั้นเอาท์พุต (Output layer) โดยชั้นที่อยู่ระหว่างชั้นอินพุต และ ชั้นเอาท์พุตจะเรียกว่าชั้นซ่อน (Hidden layer) ซึ่งสามารถมีได้หลายชั้นดังแสดงในรูปที่ 4.9



รูปที่ 4.9 โครงข่ายประสาทเทียมแบบ 3 ชั้น

จากรูปที่ 4.9 ตัวเลขยกกำลังบนตัวแปรภาษาอังกฤษจะหมายถึงชั้นของโครงข่ายประสาทเทียม ดังนั้น เมตริกค่าถ่วงน้ำหนักของชั้นแรกจะสามารถเขียนได้ คือ W^1 และ เมตริกค่าถ่วงน้ำหนักของชั้นที่สองจะสามารถเขียนได้ คือ W^2 โดยมีจำนวนโหนดอินพุตเท่ากับ R และ ตัวแปร s^1 คือ จำนวนเซลล์ประสาทในชั้นที่ 1 และ ตัวแปร s^2 คือ จำนวนเซลล์ประสาทในชั้นที่ 2

ค่าเอาท์พุตของโครงข่ายประสาทเทียมชั้นที่ 1 จะเป็นข้อมูลอินพุตของโครงข่ายประสาทเทียมชั้นที่ 2 และ เอาท์พุตของโครงข่ายประสาทเทียมชั้นที่ 2 จะเป็นข้อมูลอินพุตของโครงข่ายประสาทเทียมชั้นที่ 3 ดังนั้น จะเห็นว่าโครงข่ายประสาทเทียมชั้นที่ 2 สามารถมองเป็นโครงข่ายประสาทเทียม 1 ชั้นได้ ดังแสดงในสมการที่ 4.7 ถึง 4.10

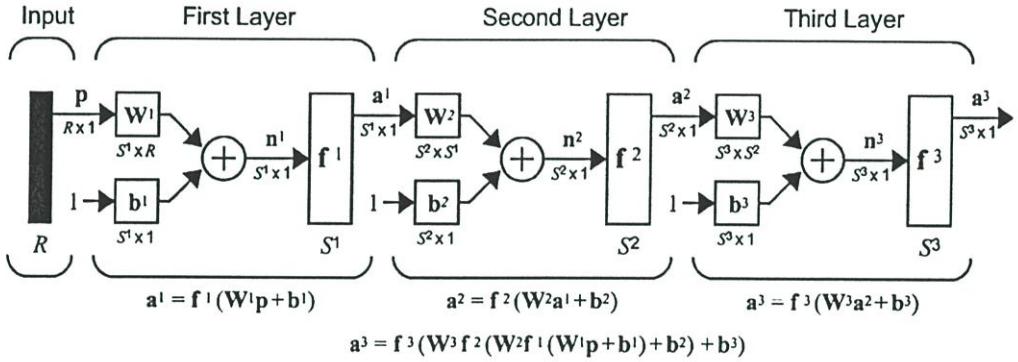
$$a^1 = f^1(W^1 p + b^1) \tag{4.7}$$

$$a^2 = f^2(W^2 a^1 + b^2) \tag{4.8}$$

$$a^3 = f^3(W^3 a^2 + b^3) \tag{4.9}$$

$$a^3 = f^3\left(W^3 f^2\left(W^2 f^1\left(W^1 p + b^1\right) + b^2\right) + b^3\right) \tag{4.10}$$

จากรูปที่ 4.9 ที่ได้แสดงไว้ทางด้านบนเป็นโครงข่ายประสาทเทียมแบบ 3 ชั้นซึ่งสามารถทำการเขียนแบบย่อได้ดังแสดงในรูปที่ 4.10



รูปที่ 4.10 สัญลักษณ์โครงข่ายประสาทเทียมแบบ 3 ชั้นแบบย่อ

ในส่วนของค่าไบอัส (Bias) ซึ่งปกติผู้ออกแบบโครงข่ายประสาทเทียมจะสามารถกำหนดให้มีในโครงข่าย หรือ ไม่มีในโครงข่ายก็ได้ ซึ่งค่าไบอัสนี้จะเป็นตัวแปรพิเศษที่เพิ่มเติมขึ้นมา ซึ่งจะทำให้โครงข่ายประสาทเทียมที่ออกแบบนั้นมีความสามารถในการรู้จำเพิ่มมากขึ้น

4.9 อัลกอริทึมการเรียนรู้แบบแพร่กลับ (Backpropagation learning Algorithm)

Backpropagation learning algorithm คือ อัลกอริทึมแบบหนึ่งที่ใช้สำหรับสอนโครงข่ายประสาทเทียมชนิดเพอร์เซปตรอนหลายชั้น (Multilayer Perceptrons: MLP) ซึ่งได้กล่าวไปแล้วในหัวข้อที่ผ่านมา ซึ่งเอาท์พุตของชั้นหนึ่งจะกลายเป็นข้อมูลอินพุตในชั้นถัดไป หรือ เรียกกระบวนการนี้ว่า Feed Forward ซึ่งสามารถแสดงดังสมการที่ 4.11 ซึ่งเป็นสมการในรูปทั่วไป

$$a^{m+1} = f^{m+1}\left(W^{m+1} a^m + b^{m+1}\right), \quad \text{เมื่อ m} = 0, 2, \ldots, \text{M-1}, \tag{4.11}$$

เมื่อ M คือ จำนวนของชั้นในโครงข่ายประสาทเทียม ซึ่งเซลล์ประสาท หรือ โหนดในชั้นที่ 1 จะรับข้อมูลอินพุตมาจากภายนอก

อัลกอริทึมแบบแพร่กลับ (Backpropagation algorithm) ที่ใช้ในการสอนระบบ (Training) จะทำหน้าที่ป้อนกลับค่าผิดพลาดจากชั้นเอาท์พุตกลับมาปรับปรุงค่าถ่วงน้ำหนักในชั้นก่อนหน้าทีละชั้นต่อไปเรื่อยๆ จนถึงชั้นที่ 1 โดยสำหรับโครงข่ายประสาทเทียมที่มีจำนวนชั้นมากกว่า 1 ชั้น (โดยไม่นับชั้นอินพุต) การปรับค่าถ่วงน้ำหนักในชั้นซ่อน (Hidden layer) จะใช้ค่าถ่วงน้ำหนักเดิมของชั้นที่สูงกว่ามาร่วมด้วย เมื่อเสร็จแล้วกระบวนการจะกลับไปสู่การทำ Feed forward propagation อีกครั้ง ซึ่งในกระบวนการสอนระบบสามารถแบ่งขั้นตอนออกได้เป็น 4 ส่วน คือ

1. กำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนัก
2. ทำกระบวนการ Feed forward propagation
3. ทำกระบวนการ Back Propagation
4. ทำการเปลี่ยนแปลงค่าถ่วงน้ำหนัก และ ค่าไบอัส

4.9.1 กำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนัก

ขั้นที่ 1 : กำหนดค่าถ่วงน้ำหนักเริ่มต้นให้กับโครงข่าย (Initialization of weights) ซึ่งปกติจะใช้ค่าจากการสุ่มซึ่งเป็นค่าเล็กๆ

ขั้นที่ 2 : ถ้าการตรวจสอบเงื่อนไขการหยุดสอน (Stopping condition) ยังไม่ถูกต้องให้ทำขั้นตอนที่ 3 – 10 ซ้ำจนกว่าเงื่อนไขจะถูกต้อง

ขั้นที่ 3 : ขั้นตอนการสอนระบบทำ ขั้นตอนที่ 4 – 9

4.9.2 กระบวนการ Feed Forward Propagation

ขั้นที่ 4 : เมื่อรูปแบบของข้อมูลอินพุต (Input patterns) จะถูกเก็บอยู่ในรูปแบบของ Array ดังแสดงในสมการที่ 4.12 โดยเมื่อ P คือ จำนวนชุดข้อมูลอินพุต (Pattern sequence number) และ N คือ จำนวนโหนดอินพุต หรือ ความยาวของข้อมูล ซึ่งข้อมูลอินพุตนี้จะอยู่ในช่วง 0 ถึง 1

$$X_{P,n} = (X_{P,1}, X_{P,2}, X_{P,3}, \cdots X_{P,N}) \qquad 4.12)$$

ขั้นที่ 5 : จากนั้นข้อมูลอินพุต $X_{P,n}$ จะถูกส่งต่อไปให้กับชั้นซ่อนใดๆ ที่อยู่ถัดไปซึ่งชั้นซ่อนนี้จะสามารถมีได้หลายชั้น ดังแสดงในสมการที่ 4.13

$$Y_{P,L,J} = X_{P,L-1} \cdot W_{J,L} + B_{J,L} \qquad (4.13)$$

เมื่อ $Y_{P,L,J}$ คือ ผลของการคำนวณของโหนดที่ J และ เลเยอร์ที่ L และ ชุดข้อมูลอินพุตที่ P โดยในที่นี้จะนับชั้นอินพุต (Input layer) เป็นชั้นที่ 1

ขั้นที่ 6 : จากนั้นผลลัพธ์ทางเอาท์พุตของทุกโหนด คือ $Y_{P,L,J}$ ในชั้นที่ L จะผ่านฟังก์ชันกระตุ้น (Activation function) แล้วกลายเป็นข้อมูลอินพุตในชั้นถัดไป ดังแสดงในสมการที่ 4.14

$$X_{P,L} = \left(f(Y_{P,L,1}), f(Y_{P,L,2}), \cdots, f(Y_{P,L,K})\right) \tag{4.14}$$

เมื่อ K คือ จำนวนโหนดทั้งหมดในชั้นที่ L ซึ่งค่าเอาท์พุตของการคำนวณนี้จะกลายเป็นอินพุตในชั้นถัดไปเรื่อยๆ จนถึงชั้นสุดท้ายคือชั้นเอาท์พุต (Output layer) ซึ่งจะกำหนดให้ตัวแปรชื่อ $X_{P,O}$

4.9.3 ทำกระบวนการ Back Propagation

ขั้นที่ 7 : เมื่อถึงจุดนี้จะทำให้ได้ข้อมูลเอาท์พุต ($X_{P,O}$) และ ข้อมูลเป้าหมาย (Target) แทนด้วย $T_{(P)}$ จากนั้นทำการหา error ของชั้นเอาท์พุต ($E_{(P)}$) ดังแสดงในสมการที่ 4.15

$$E_{(P,O)} = \left(T_P - X_{(P,O)}\right) \cdot X'_{(P,O)} \tag{4.15}$$

เมื่อ $X'_{(P,O)}$ สามารถแสดงดังสมการที่ 4.16

$$X'_{(P,O)} = \left(f'\left(Y_{(P,O,1)}\right), f'\left(Y_{(P,O,2)}\right), \ldots, f'\left(Y_{(P,O,K)}\right)\right) \tag{4.16}$$

เมื่อ f' คือ การทำ First derivative ของฟังก์ชันกระตุ้น (Transfer function)

ขั้นที่ 8 : ในกรณีการหาความผิดพลาด (error) ในชั้นซ่อนที่ L ใดๆ ของโหนดที่ J สามารถทำได้ดังแสดงในสมการที่ 4.17

$$E_{(P,L,J)} = X'_{(P,L,J)} \cdot \sum\left(E_{(P,L+1,K)} \cdot W_{(K,L+1,J)}\right) \tag{4.17}$$

เมื่อ K แทนโหนดที่ K ในเลเยอร์ที่ L+1 ซึ่งเมื่อทำสมการที่ 4.17 นี้แล้วผลลัพธ์ของค่าความผิดพลาด (error) ที่ได้ คือ $E_{(P,L)}$ ซึ่งเป็นค่าความผิดพลาดของชั้นซ่อนที่ L ใดๆ

4.9.4 การเปลี่ยนแปลงค่าถ่วงน้ำหนัก และ ค่าไบอัส

ขั้นที่ 9 : จากนั้นค่าถ่วงน้ำหนักของแต่ละโหนดจะถูกเปลี่ยนแปลง (updated) ซึ่งค่าถ่วงน้ำหนักใหม่ของโหนดที่ J และ ชั้นที่ L สามารถแสดงดังสมการที่ 4.18

$$W_{(J,L)}T+1 = W_{(J,L)}T + \eta\sum\left(E_{(P,L,J)} \cdot X_{(P,L-1)}\right) + \alpha\left(W_{(J,L)}T - W_{(J,L)}T-1\right) \qquad (4.18)$$

เมื่อกำหนดให้ η คือ อัตราการเรียนรู้ (Learning rate) และ กำหนดให้ α คือ Momentum factor และ กำหนดให้ T คือ จำนวนรอบที่ทำซ้ำ (Iteration cycle) โดยค่าถ่วงน้ำหนักที่มีการเปลี่ยนแปลงนั้นจะถูกคำนวณจากค่าถ่วงน้ำหนักในรอบ (cycle) การคำนวณก่อนหน้าที่ถูกคูณด้วยค่า Momentum factor

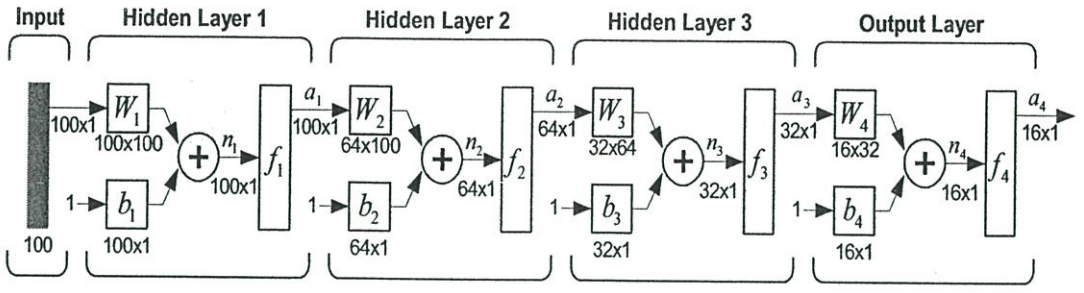
ขั้นที่ 10 : ตรวจสอบเงื่อนไขการหยุดสอนระบบ (Training)

4.5 การนำโครงข่ายประสาทเทียมไปประยุกต์ใช้ประโยชน์

จากหลักการของโครงข่ายประสาทเทียม ซึ่งเป็นความพยายามจำลองการทำงานของสมองมนุษย์ ได้มีการนำไปประยุกต์ใช้งานต่างๆ มากมายซึ่งได้แก่ การจดจำลายมือ พิสูจน์เอกลักษณ์ลายเซ็น การจดจำใบหน้า การประมาณค่าฟังก์ชันหรือการประมาณความสัมพันธ์ต่างๆ งานจัดหมวดหมู่และแยกแยะสิ่งของ การพยากรณ์อากาศ การพยากรณ์หุ้น และ อื่นๆ

4.6 การออกแบบระบบรู้จำ

วิทยานิพนธ์นี้ใช้ระบบโครงข่ายประสาทเทียม (Artificial Neural Network) ในการรู้จำเสียงพูดคำไทยที่ใช้สำหรับการควบคุมโปรแกรม Winamp โดยอินพุตของระบบโครงข่ายประสาทเทียมจะเป็นข้อมูลที่ได้มาจากภาคการเลือกคุณลักษณะเด่น (Feature selection) ซึ่งในวิทยานิพนธ์นี้จะใช้โครงข่ายประเภทเพอเซปตรอนหลายชั้น (Multilayer Perceptron) และ ใช้กระบวนการเรียนรู้แบบแพร่กลับ (Backpropagation) โดยออกแบบจะใช้โครงข่ายจำนวน 5 ชั้น (Layer) โดยมีชั้นอินพุต (Input layer) 1 ชั้น ชั้นซ่อน (Hidden layer) 3 ชั้น และ มีชั้นเอาท์พุต (Output layer) 1 ชั้น ซึ่งแต่ละชั้นมีจำนวนโหนดเท่ากับ 100, 100, 64, 32, 16 ตามลำดับ [2] โดยแต่ละโหนดเลือกใช้ฟังก์ชันกระตุ้น (Activation function) แบบ Log-sigmoid [19] เพื่อให้ค่าทางเอาท์พุตอยู่ในช่วง 0 – 1 ซึ่งการออกแบบโครงข่ายประสาทเทียมแสดงดังรูปที่ 4.11

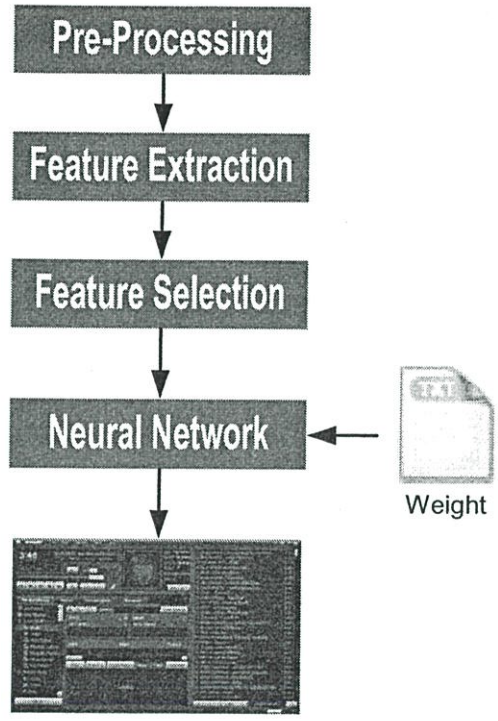


รูปที่ 4.11 แสดงการออกแบบโครงข่ายประสาทเทียมที่ใช้ในวิทยานิพนธ์

จากรูปที่ 4.11 เป็นการออกแบบโครงข่ายประสาทเทียมเพื่อแสดงให้เห็นความสัมพันธ์ระหว่างจำนวนเซลล์ประสาทในแต่ละชั้น และ ขนาดของข้อมูลทางเอาท์พุตของแต่ละชั้น โดยในวิทยานิพนธ์นี้จะใช้โปรแกรม Qnet 2000 [20] มาใช้ในการสอนระบบ (Training) ซึ่งโปรแกรมนี้มีความสามารถในการสอนระบบโครงข่ายประสาทเทียมที่มีจำนวนโหนด หรือ จำนวนชั้นของโครงข่ายหลายชั้นได้ ซึ่งจะทำให้ผู้พัฒนาโปรแกรมสามารถที่จะแก้ไขโครงสร้างของโครงข่ายประสาทเทียม และ สอนระบบใหม่ได้อย่างรวดเร็วยิ่งขึ้น ซึ่งผลจะทำให้ได้ค่าถ่วงน้ำหนัก (Weight) ออกมาและนำไปใช้ในโปรแกรมที่กำลังพัฒนาต่อไป โดยสามารถอธิบายขั้นตอนการทำงานได้ดังรูปที่ 4.12

รูปที่ 4.12 กระบวนการสอนระบบด้วยโปรแกรม Qnet 2000

กระบวนการสอนระบบ (Training) ด้วยโปรแกรม Qnet 2000 ซึ่งผลลัพธ์ที่ได้จะเป็นค่าถ่วงน้ำหนักที่เก็บอยู่ในรูปของไฟล์นามสกุล ".TXT" จากนั้นในโหมดการทำงานปกติซึ่งเป็นโปรแกรมหลักจะนำไฟล์นามสกุล ".TXT" นี้โหลดเข้าสู่โปรแกรมเพื่อกระจายค่าถ่วงน้ำหนักไปยังชั้น (Layer) ต่างๆ ในโครงข่ายประสาทเทียมดังแสดงในรูปที่ 4.13



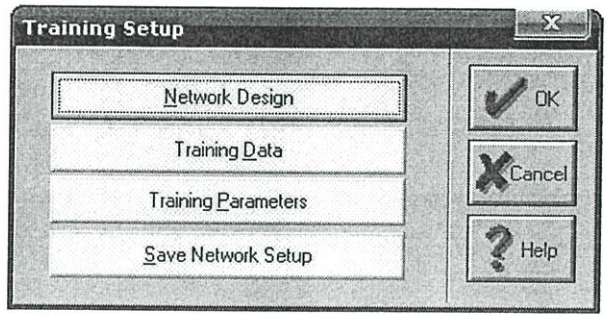
รูปที่ 4.13 ขั้นตอนการทำงานของโปรแกรมหลักที่ใช้สำหรับสั่งงานโปรแกรมเล่นเพลงวินแอมป์

โปรแกรม Qnet2000 [13] คือ โปรแกรมเครื่องมือสำหรับโปรแกรมเมอร์ หรือ นักวิจัยที่มีการใช้งานระบบโครงข่ายประสาทเทียมแบบหลายชั้น (MLP) โดยโปรแกรมนี้จะทำหน้าที่สอนระบบ (Training) โครงข่ายประสาทเทียม ซึ่งภายในโปรแกรม Qnet2000 ผู้พัฒนาโปรแกรมสามารถออกแบบโครงข่ายประสาทเทียมให้มีจำนวนเซลล์ประสาทในแต่ละชั้น (Layer) ได้ไม่จำกัด และ มีจำนวนชั้นไม่เกิน 9 ชั้น นอกจากนั้นยังสามารถเลือกประเภทของฟังก์ชันกระตุ้น (Transfer function) ได้หลายแบบด้วย คือ

1. Sigmoid function
2. Gaussian function
3. Hyperbolic Tangent function
4. Hyperbolic Secant function

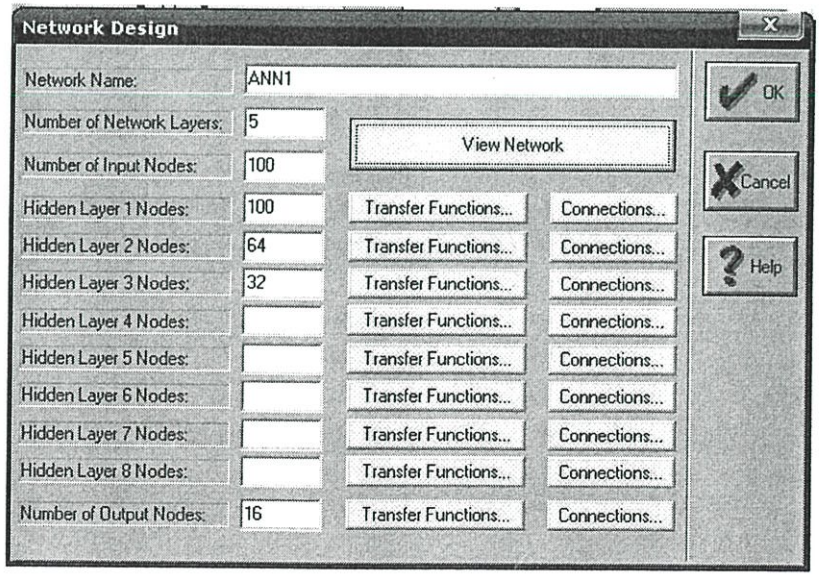
ในการใช้งานโปรแกรม Qnet 2000 เพื่อให้ได้ค่าถ่วงน้ำหนัก (Weight) มานั้นจะต้องมีการออกแบบโครงข่าย รวมทั้งเลือกชนิดของฟังก์ชันกระตุ้น (Activation function) ภายในโปรแกรม Qnet 2000 ให้ตรงกับที่ได้ออกแบบไว้ในโปรแกรมหลักซึ่งจะต้องออกแบบโครงข่ายด้วยการเขียนโปรแกรมด้วยตัวเอง ซึ่งการใช้งานโปรแกรม Qnet 2000 มีขั้นตอนการทำงานดังนี้ คือ

1. เปิดโปรแกรม Qnet 2000 จากนั้นเลือกกดที่ปุ่ม New เพื่อสร้างการออกแบบใหม่ดังแสดงในรูปที่ 4.14



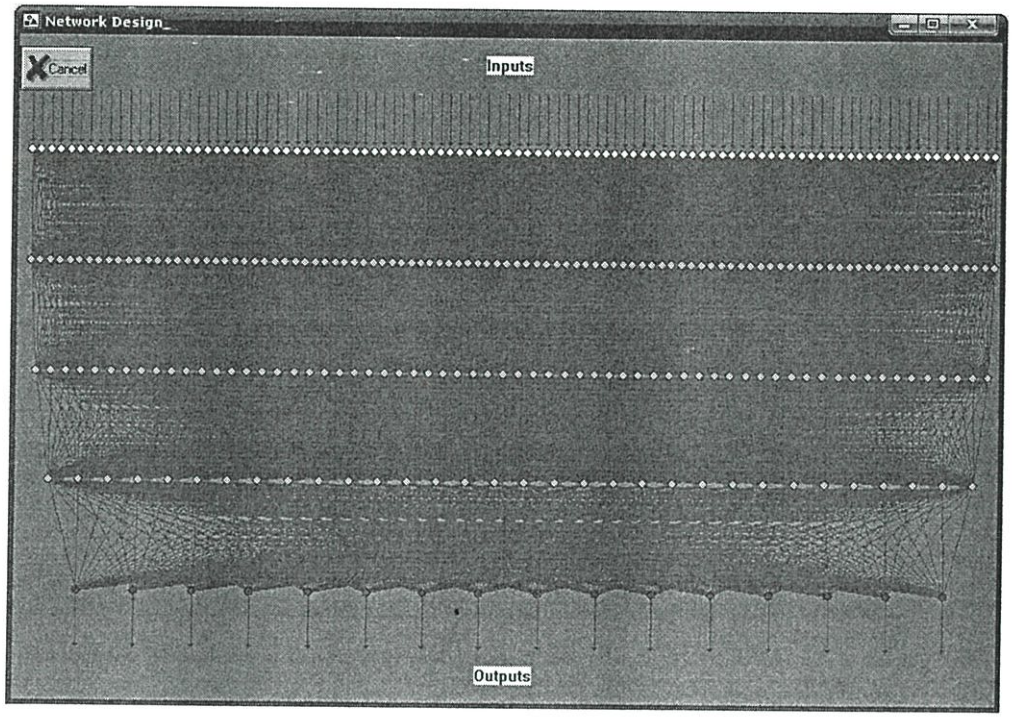
รูปที่ 4.14 หน้าต่าง Training Setup

2. กด Network Design เพื่อทำการออกแบบโครงข่าย ซึ่งจะต้องเหมือนกับที่ได้เขียนโปรแกรมไว้ในโปรแกรมหลักดังแสดงในรูปที่ 4.15 โดยในหน้าต่างนี้จะสามารถกำหนดจำนวนชั้นของโครงข่ายกำหนดจำนวนโหนด หรือ จำนวนเซลล์ประสาทในแต่ละชั้น และ สามารถกำหนดฟังก์ชันกระตุ้นได้



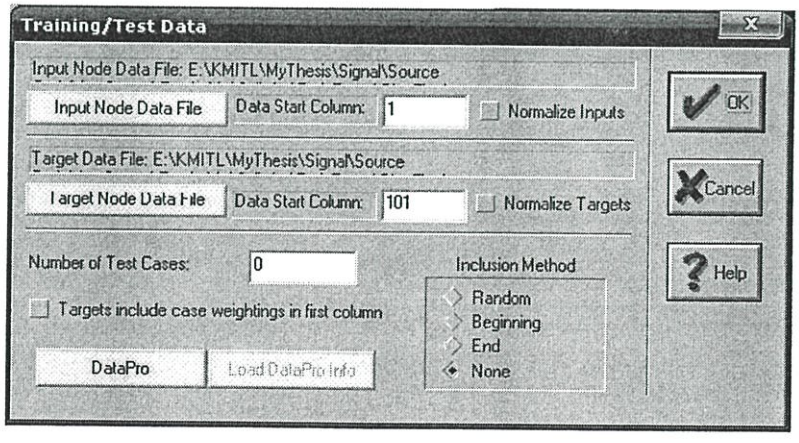
รูปที่ 4.15 หน้าต่าง Network Design ใช้สำหรับออกแบบโครงข่ายประสาทเทียม

เมื่อกดปุ่ม View Network จะทำให้สามารถดูแผนภาพของโครงข่ายประสาทเทียมที่ได้ออกแบบไว้ได้ดังแสดงในรูปที่ 4.16



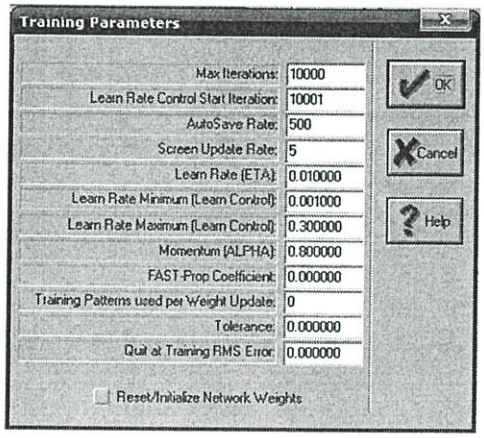
รูปที่ 4.16 การออกแบบโครงข่ายประสาทเทียมด้วยโปรแกรม Qnet 2000

3. เมื่อออกแบบโครงข่ายประสาทเทียมแล้ว จากนั้นจะทำการกำหนดข้อมูลอินพุต และข้อมูลเป้าหมาย (Target) ให้กับโครงข่ายประสาทเทียมโดยกดปุ่ม Training Data ในหน้าต่าง Training Setup ดังแสดงในรูปที่ 4.17 ซึ่งข้อมูลนี้จะอยู่ในรูปไฟล์นามสกุล ".TXT"



รูปที่ 4.17 การกำหนดข้อมูลอินพุต และ ข้อมูลเป้าหมาย (Target) ให้กับโปรแกรม Qnet 2000

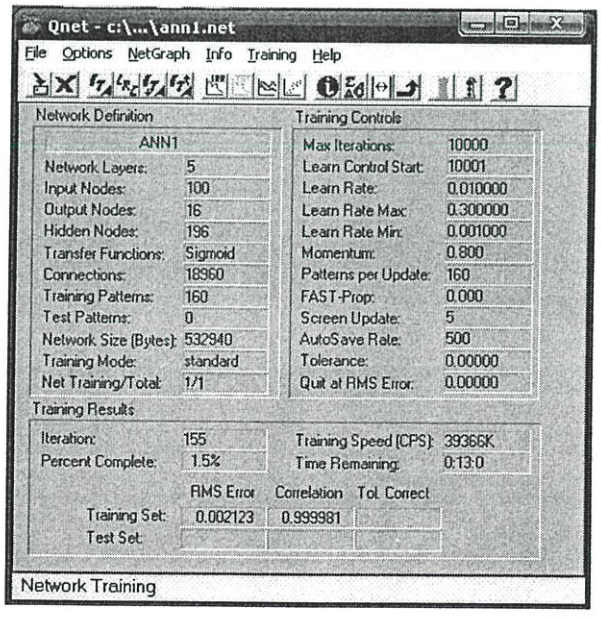
4. จากนั้นจะทำการกำหนดพารามิเตอร์สำหรับการสอนระบบ (Training parameters) โดยกดที่ปุ่ม Training Parameters ในหน้าต่าง Training Setup ซึ่งในหน้าต่าง Training Parameters จะสามารถกำหนดพารามิเตอร์ได้หลายอย่าง เช่น จำนวนรอบในการสอนระบบซึ่งกำหนดไว้ที่ 10,000 รอบ อัตราการเรียนรู้ (Learning rate) และ อื่นๆ เป็นต้น ดังแสดงในรูปที่ 4.18



รูปที่ 4.18 การกำหนดพารามิเตอร์ให้กับระบบการสอนในโปรแกรม Qnet 2000

5. ทำการ Save การออกแบบระบบ และ การกำหนดค่าพารามิเตอร์ต่างๆ ด้วยการกดปุ่ม Save Network Setup ที่อยู่ในหน้าต่าง Training Setup

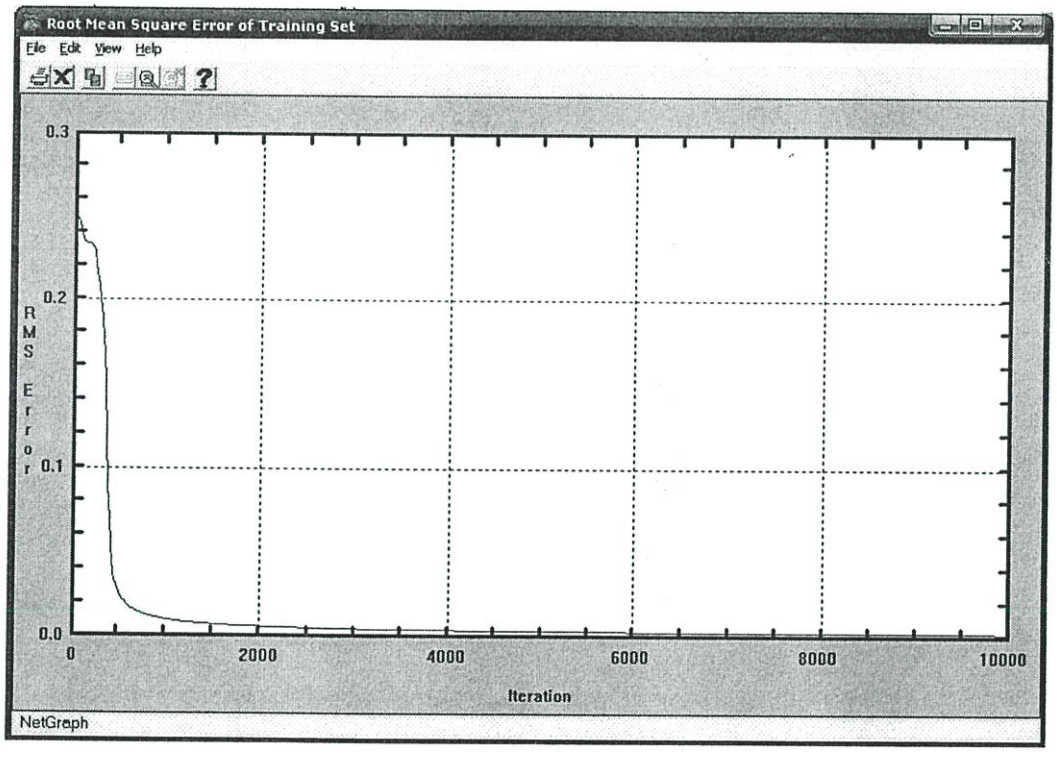
6. กดปุ่ม OK ในหน้าต่าง Training Setup จะส่งผลให้โปรแกรม Qnet เริ่มสอนระบบ



รูปที่ 4.19 หน้าต่างการทำงานขณะโปรแกรม Qnet 2000 สอนระบบ (Training)

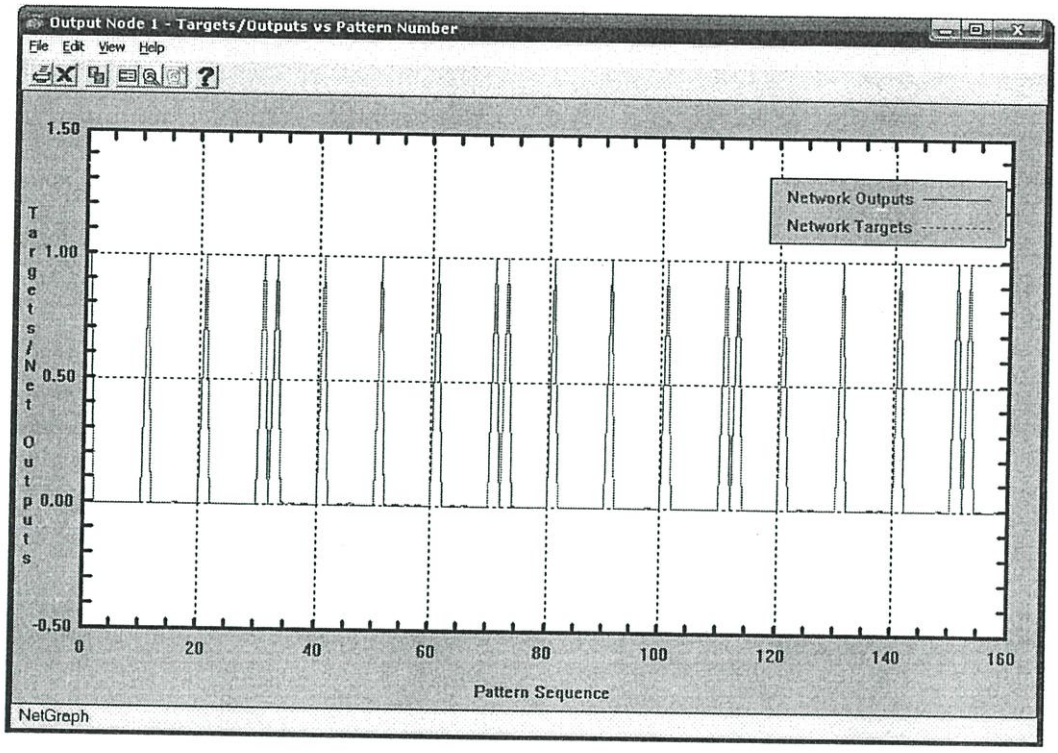
7. เมื่อทำการสอนระบบด้วยโปรแกรม Qnet 2000 เสร็จแล้วจะทำให้ได้ค่าถ่วงน้ำหนักออกมา ซึ่งผู้พัฒนาโปรแกรมสามารถตรวจสอบ Error ในแต่ละรอบของการสอนระบบได้ดังแสดงในรูปที่ 4.20 ซึ่งจากรูปที่ 4.20 จะเห็นว่าค่า Error ในแต่ละรอบนั้นจะลดลงอย่างต่อเนื่องและเมื่อถึงรอบที่ประมาณ 2,000 จะเห็นว่า Error นั้นเริ่มหยุดนิ่ง โดยการวัดค่า Error ในโปรแกรม Qnet 2000 นี้จะใช้วิธี RMS error (Root Mean Square Error)

จากรูปที่ 4.20 จะเห็นว่าสามารถกำหนดรอบของการสอนระบบใหม่ให้น้อยลงได้ โดยควรจะกำหนดไว้ที่ 5,000 รอบก็เพียงพอ ซึ่งจากเดิมกำหนดไว้ที่ 10,000 รอบ ในส่วนของเวลาที่ใช้ในการสอนระบบ (Training) จะขึ้นอยู่กับจำนวนข้อมูลอินพุต และ จำนวนรอบที่ใช้ในการสอนระบบ ซึ่งถ้ามีข้อมูลอินพุต 80 ข้อมูล และ กำหนดจำนวนรอบของการสอนระบบไว้ที่ 10,000 รอบจะใช้เวลาในการสอนระบบประมาณ 5 นาที



รูปที่ 4.20 หน้าต่างแสดงความผิดพลาดด้วยวิธี RMS

8. ในการตรวจสอบความถูกต้องของการสอนระบบอีกอย่างหนึ่งที่ต้องพิจารณา คือ ผลลัพธ์ทางเอาท์พุตจริง เมื่อเทียบกับค่าเป้าหมาย (Target) ดังแสดงในรูปที่ 4.21



รูปที่ 4.21 ผลลัพธ์ทางเอาท์พุตของการสอนระบบด้วยโปรแกรม Qnet 2000

จากรูปที่ 4.21 เป็นการแสดงค่าเอาท์พุตของโหนด 1 (ทั้งหมดมี 16 โหนดเอาท์พุต) ในทุกรูปแบบของข้อมูลอินพุต ซึ่งถ้าการสอนระบบนั้นมีประสิทธิภาพที่ดีเส้นสีแดง (เส้นประ) และ เส้นสีเขียว (เส้นทึบ) จะต้องทับกันสนิด และ ในการดูเอาท์พุตของโหนดที่ 2 จนถึง โหนดที่ 16 ทำได้โดยปิดหน้าต่างของโหนดที่ 1

บทที่ 5
การสั่งงานโปรแกรมเล่นเพลงวินแอมป์

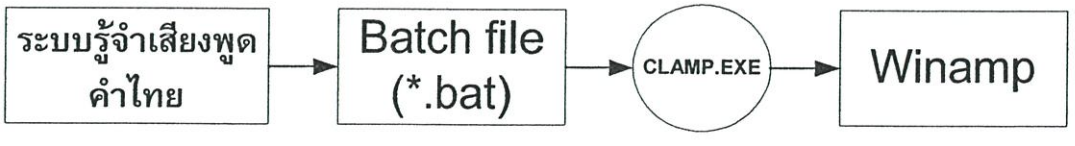
5.1 บทนำ

โปรแกรมประยุกต์ที่ผู้ใช้ (User) ส่วนใหญ่รู้จักเป็นอย่างดีได้แก่โปรแกรม Photoshop โปรแกรม Adobe Acrobat โปรแกรม Windows Media Player และ อื่นๆ นั้นผู้ใช้จะสามารถใช้งานมันได้ผ่านการดับเบิลคลิกที่ไอคอนบนหน้าจอคอมพิวเตอร์ ซึ่งถือว่าเป็นวิธีการเข้าทางหน้าบ้าน แต่สำหรับบางแอพพลิเคชันโปรแกรมนั้นจะถูกพัฒนามาให้มีทางเข้าหลังบ้านด้วย ซึ่งช่องทางนี้ส่วนใหญ่ผู้ที่ใช้จะเป็นโปรแกรมเมอร์ หรือ นักพัฒนาโปรแกรมต่างๆ ซึ่งจะต้องรู้ข้อมูลทางเทคนิคในการใช้งานเป็นอย่างดี และ โดยปกติทางเข้านี้ผู้ใช้โดยทั่วไป (User) จะไม่สามารถใช้งานได้ ซึ่งในวิทยานิพนธ์นี้จะใช้ช่องทางนี้ในการควบคุมโปรแกรมเล่นเพลงวินแอมป์ (Winamp)

การนำระบบการรู้จำเสียงพูดคำไทยที่ได้สร้างขึ้นมาทดสอบ และ ใช้งานจริงกับการควบคุมโปรแกรมเล่นเพลงวินแอมป์ (Winamp) นั้นจะต้องมีโปรแกรมอีกตัวหนึ่งซึ่งทำหน้าที่เป็นตัวกลางระหว่างโปรแกรมรู้จำเสียงพูดคำไทยที่พัฒนาขึ้นด้วยโปรแกรม MATLAB กับโปรแกรมเล่นเพลงวินแอมป์ นั้นก็คือ โปรแกรม CLAMP ซึ่งจะทำหน้าที่เป็นตัวกลางในการเชื่อมระหว่างโปรแกรม MATLAB กับ โปรแกรมเล่นเพลงวินแอมป์

5.2 หลักการควบคุมโปรแกรมวินแอมป์

ในการควบคุมโปรแกรมวินแอมป์เพื่อให้โปรแกรมทำการเปิด หรือ ปิด การทำงานโดยที่ไม่ต้องดับเบิ้ลคลิกที่ไอคอนของโปรแกรมวินแอมป์บนหน้าจอคอมพิมเตอร์ สามารถทำได้โดยการสั่งงานมันผ่านการเขียนคำสั่งบน DOS (Command Line) หรือ การสร้าง Batch file ซึ่งคำสั่งภายใน Batch file จะเป็นการเขียนคำสั่ง DOS เพื่อเชื่อมต่อไปที่โปรแกรม CLAMP เพื่อสั่งงานโปรแกรมวินแอมป์อีกทีหนึ่งตามที่ต้องการซึ่งมีการบวนการดังแสดงในรูปที่ 5.1



รูปที่ 5.1 การสั่งงานโปรแกรมวินแอมป์โดยผ่าน Patch file

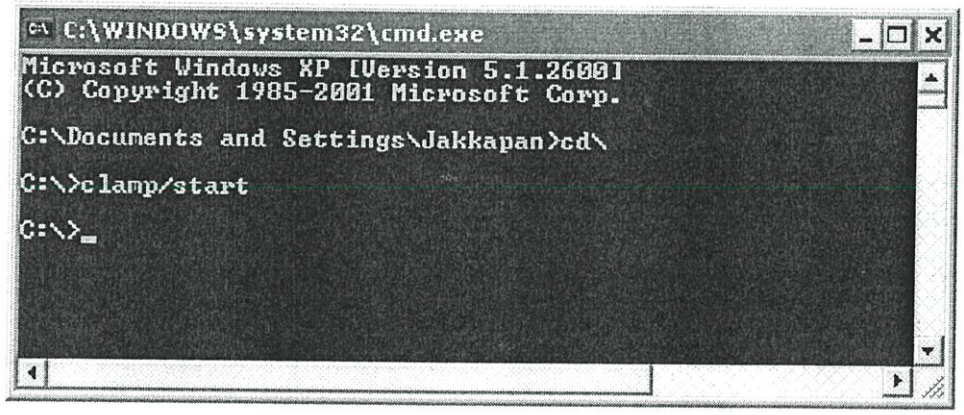
5.3 Batch file

Batch file คือ ไฟล์ที่สร้างจากโปรแกรมเอกสารต่างๆ เช่น โปรแกรม Notepad, WordPad หรือ EditPlus จากนั้นเขียนคำสั่ง DOS และ ทำการเซฟด้วยนามสกุล ".bat" ซึ่งการใช้งาน Batch file นั้นสามารถทำได้เพียงแค่ดับเบิลคลิกที่ไฟล์นี้ ซึ่งจะทำให้คำสั่ง DOS ใน Batch file นี้ทำงานตามที่ได้เขียนคำสั่งเอา ไว้ข้างใน

5.4 โปรแกรม CLAMP

โปรแกรม CLAMP คือ โปรแกรมขนาดเล็กที่นิยมนำไปใช้ในการสั่งงานโปรแกรมวินแอมป์ผ่านทาง Command line (การเข้าถึงโปรแกรมด้วยการพิมพ์คำสั่ง DOS) โดยโปรแกรมนี้จะทำงานโดยลำพัง และ ไม่ต้องมีกระบวนการติดตั้งโปรแกรมใดๆ รวมถึงไม่ต้องใช้ Plug-in ของโปรแกรมวินแอมป์ด้วย ซึ่งการติดตั้งเพียงแค่นำไฟล์ CLAMP.EXE ไปวางตำแหน่งใดๆ บนเครื่องคอมพิวเตอร์ จากนั้นเขียนคำสั่ง DOS โดยชี้ไปที่ตำแหน่งไฟล์ CLAME.EXE นี้ แล้วตามด้วยคำสั่งสำหรับควบคุมการทำงานของโปรแกรมวินแอมป์ดังแสดงในรูปที่ 5.2

- ให้ทำการคัดลอกไฟล์ CLAMP.exe ไปวางไว้ที่ไดร์ C:\>
- พิมพ์คำสั่งบนโปรแกรม DOS คือ C:\>CLAMP/start
- โปรแกรมวินแอมป์จะเปิดขึ้นมา ซึ่งแสดงว่าการใช้คำสั่งนั้นถูกต้อง



รูปที่ 5.2 ตัวอย่างการสั่งให้เปิดโปรแกรมวินแอมป์ด้วยการพิมพ์คำสั่งบน DOS

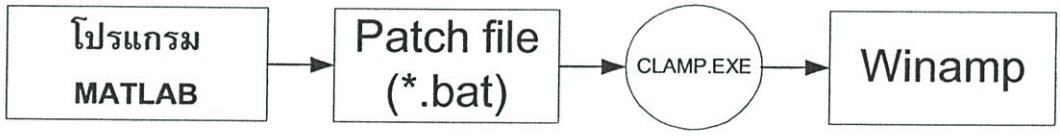
ในส่วนของคำสั่งที่ใช้ควบคุมโปรแกรมวินแอมป์นั้นมีจำนวนมาก ซึ่งสามารถดูได้จากเวปไซต์ http://membres.lycos.fr/clamp/ และ ในตารางที่ 5.1 จะเป็นคำสั่งที่ใช้งานในวิทยานิพนธ์นี้

ตารางที่ 5.1 คำสั่งการควบคุมโปรแกรมวินแอมป์ของโปรแกรม CLAMP

คำสั่ง	ความหมาย
START	เปิดโปรแกรมวินแอมป์
QUIT	ปิดโปรแกรมวินแอมป์
PREV	สั่งให้โปรแกรมวินแอมป์เล่นเพลงก่อนหน้า
NEXT	สั่งให้โปรแกรมวินแอมป์เล่นเพลงถัดไป
VOLUP	สั่งให้โปรแกรมวินแอมป์เพิ่มเสียง
VOLDN	สั่งให้โปรแกรมวินแอมป์ลดเสียง
PLAY	สั่งให้โปรแกรมวินแอมป์เล่นเพลง
PLAYPAUSE	สั่งให้โปรแกรมวินแอมป์หยุดเพลง

5.5 การเชื่อมระหว่างโปรแกรม MATLAB กับโปรแกรม CLAMP

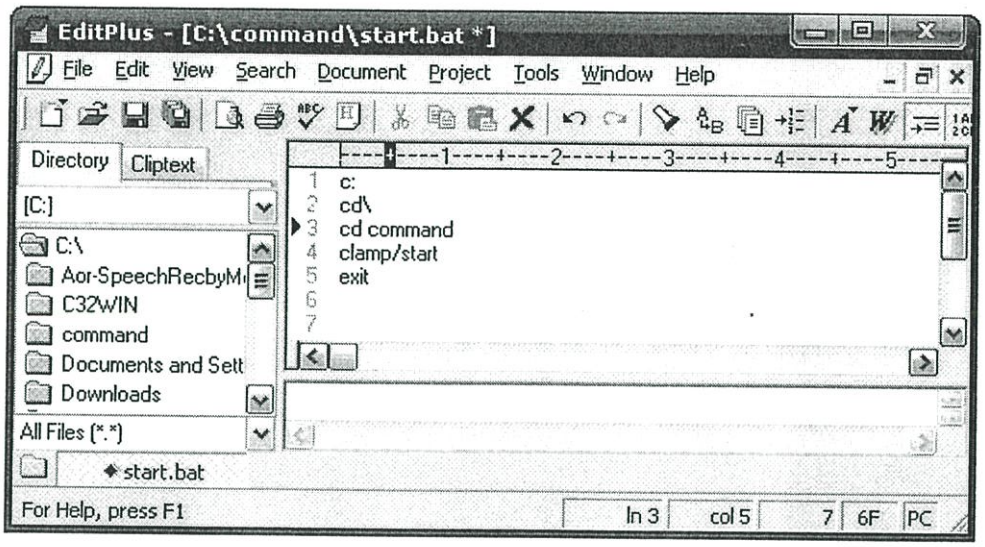
จากหัวข้อที่ผ่านได้ทำการศึกษาการควบคุมโปรแกรมวินแอมป์โดยการพิมพ์คำสั่งบนโปรแกรม DOS เพื่อเรียกใช้คำสั่งของโปรแกรม CLAMP มาแล้ว ซึ่งในหัวข้อนี้จะนำเสนอวิธีการสั่งงานโปรแกรมวินแอมป์เพื่อเรียกใช้คำสั่งของโปรแกรม CLAMP โดยที่ไม่ต้องพิมพ์คำสั่งบนโปรแกรม DOS แต่จะสั่งงานโปรแกรมวินแอมป์ผ่านทาง Batch file แทนและจากนั้นจะเขียนโปรแกรม MATLAB เพื่อเรียก Batch file นี้ไปใช้งานอีกครั้งหนึ่ง ซึ่งมีขั้นตอนดังแสดงในรูปที่ 5.3



รูปที่ 5.3 ขั้นตอนการสั่งงานโปรแกรมวินแอมป์ด้วยโปรแกรม MATLAB

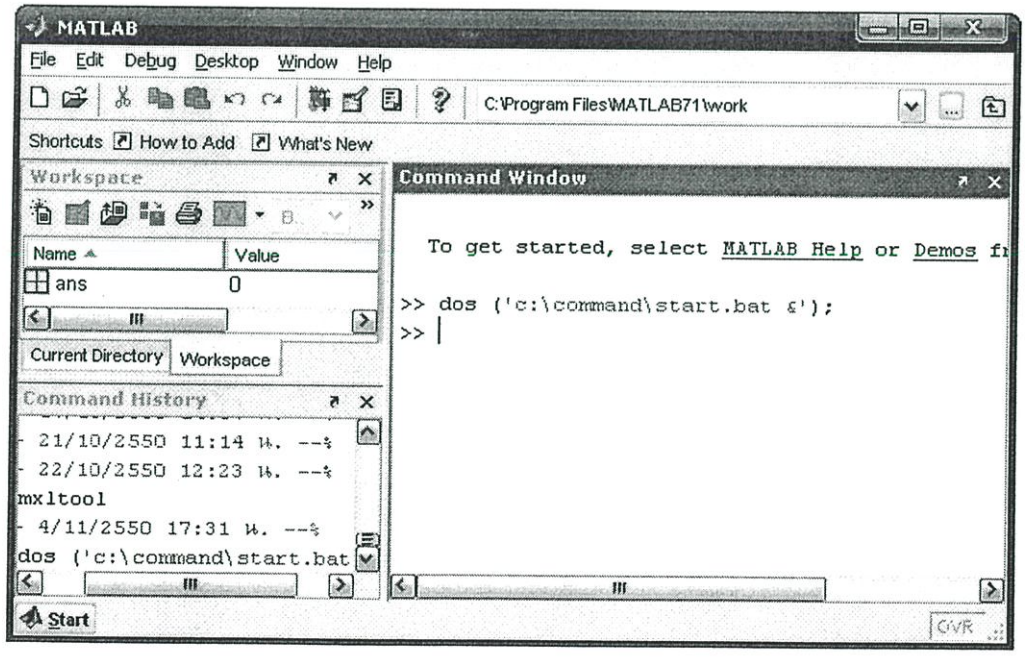
จากรูปที่ 5.3 จะเห็นว่าเมื่อทำการรันโปรแกรมระบบรู้จำเสียงพูดคำไทยที่เขียนด้วยโปรแกรม MATLAB จนถึงภาคเอาท์พุตของระบบโครงข่ายประสาทเทียม ซึ่งในแต่ละคำตอบของโครงข่ายประสาทเทียมจะใช้คำสั่งเรียก Batch file ให้ทำงานเพื่อเชื่อมต่อไปที่โปรแกรม CLAMP เพื่อสั่งงานโปรแกรมวินแอมป์ ซึ่งมีขั้นตอนดังนี้คือ

- ให้ทำการคัดลอกไฟล์ CLAMP.exe ไปวางไว้ที่ไดร์ C:\>command\
- ทำการสร้าง Batch file โดยพิมพ์คำสั่งดังแสดงในรูปที่ 5.4 จากนั้นบันทึกเป็น start.bat เก็บไว้ที่ตำแหน่ง C:\>command\



รูปที่ 5.4 การสร้างคำสั่ง "เปิดเครื่อง" บน Batch file

จากรูปที่ 5.4 Batch file ที่บรรจุคำสั่ง DOS นี้จะทำหน้าที่เข้าไปสั่งให้โปรแกรม CLAMP ส่งคำสั่ง "start" ไปสั่งให้โปรแกรมวินแอมป์เปิดขึ้นมาแทนการพิมพ์คำสั่งบนโปรแกรม DOS โดย สำหรับคำสั่งอื่นๆ สามารถทำได้โดยนำคำสั่งในตารางที่ 5.1 มาแทนในบรรทัดที่ 4 ของรูปที่ 5.4 จากนั้นทดลองสั่งงานโปรแกรมวินแอมป์ด้วยโปรแกรม MATLAB โดยพิมพ์คำสั่งดังรูปที่ 5.5



รูปที่ 5.5 การพิมพ์คำสั่งบนโปรแกรม MATLAB เพื่อเรียก Batch file

เมื่อคำสั่งนี้ทำงานจะทำให้โปรแกรมวินแอมป์ทำงานโดยเป็นการสั่งงานจากโปรแกรม MATLAB แทนการพิมพ์บนโปรแกรม DOS

บทที่ 6
ผลการทดลอง

6.1 บทนำ

การทดลองนี้มีจุดประสงค์เพื่อทดสอบความสามารถในการรู้จำเสียงพูดคำไทยเพื่อควบคุมโปรแกรมเล่นเพลงวินแอมป์กับคำประสม (Compound word) จำนวน 8 คำ ซึ่งได้แก่คำว่า "เปิดเครื่อง" "ปิดเครื่อง" "เพลงก่อนหน้า" "เพลงถัดไป" "เพิ่มเสียง" "ลดเสียง" "เล่นเพลง" และ "หยุดเพลง" โดยทำการทดสอบกับบุคคลเพศชายจำนวน 9 คน และ เพศหญิงจำนวน 9 คน มีอายุระหว่าง 19 – 25 ปี โดยได้เก็บบันทึกเสียงพูดของแต่ละคนคำละ 10 ครั้ง รวมเป็น 1,440 คำ ซึ่งสัญญาณเสียงที่พูดจากไมโครโฟนมีการสุ่มสัญญาณ (Sampling rate) เท่ากับ 11.025kHz จำนวน 16 บิต และ มีการบันทึกเสียงครั้งละ 3 วินาที นอกจากนั้นในการบันทึกเสียงคำพูดของแต่ละคนจะบันทึกเสียงทุกคำในช่วงเวลาเดียวกัน

ในส่วนของกลุ่มข้อมูลเสียงที่ใช้ในการสอนระบบการรู้จำ (Training) จะใช้เสียงพูดของผู้ชาย 1 คน และ ผู้หญิง 1 คน โดยแต่ละคนพูดคำละ 5 ครั้งในการสอนระบบ และ อยู่นอกกลุ่มข้อมูลเสียงที่จะนำมาใช้ทดสอบระบบการรู้จำ

6.2 องค์ประกอบของระบบที่ใช้ในการทดลอง

6.2.1 องค์ประกอบทางฮาร์ดแวร์

1. คอมพิวเตอร์ Intel Pentium4 2.66GHz, RAM 1GHz
2. การ์ดเสียง (Sound card) ซึ่งสำหรับการทดลองนี้ใช้ของบริษัท Creative
3. ไมโครโฟนของบริษัท Labtech รุ่น Labtech Verse 504

6.2.2 องค์ประกอบทางซอฟแวร์

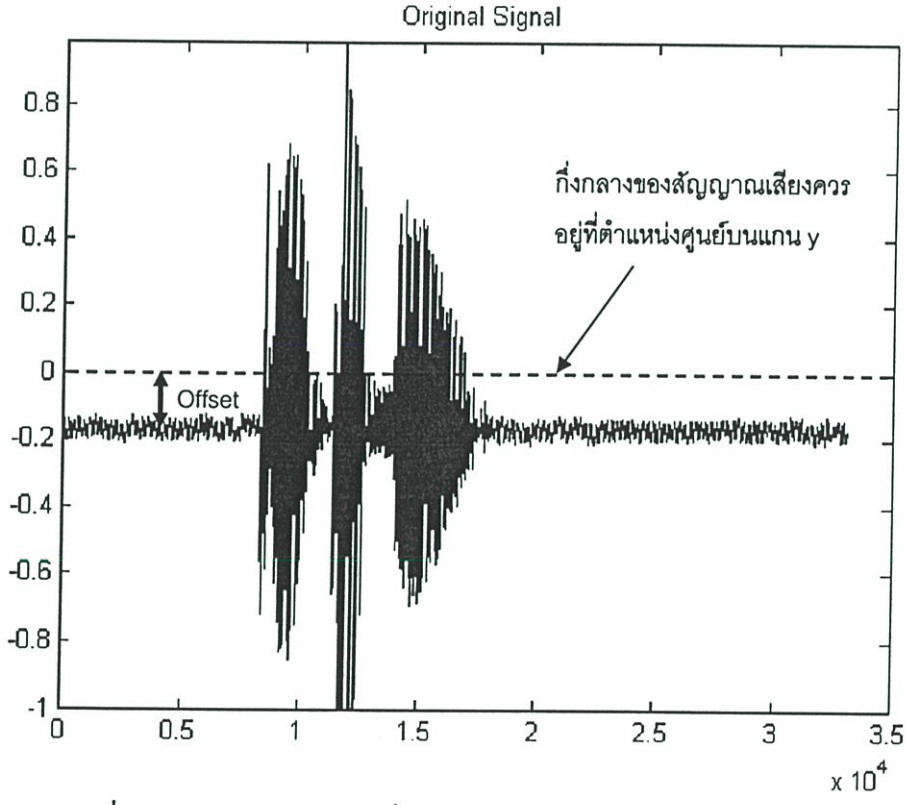
1. ระบบปฏิบัติการ Window XP
2. โปรแกรม MATLAB 7.1
3. โปรแกรม Qnet 2000
4. โปรแกรม CLAMP 1.11
5. โปรแกรม Winamp 5.5

6.3 การทดลองในภาคการเตรียมสัญญาณเสียงเบื้องต้น

ในระบบรู้จำเสียงนั้นกระบวนแรกที่จะต้องถูกทำงานเป็นอันดับแรก คือ ส่วนของการเตรียมสัญญาณเสียงเบื้องต้น (Pre-processing) ซึ่งภาคการทำงานนี้จะทำหน้าที่ลดสิ่งแปลกปลอมหรือ สัญญาณส่วนที่ไม่ต้องการออกไปจากสัญญาณเสียงพูด เช่นสัญญาณรบกวน (Noise) รวมถึงการตัดสัญญาณเสียงบริเวณที่ไม่ใช่คำพูดออกไป เป็นต้น

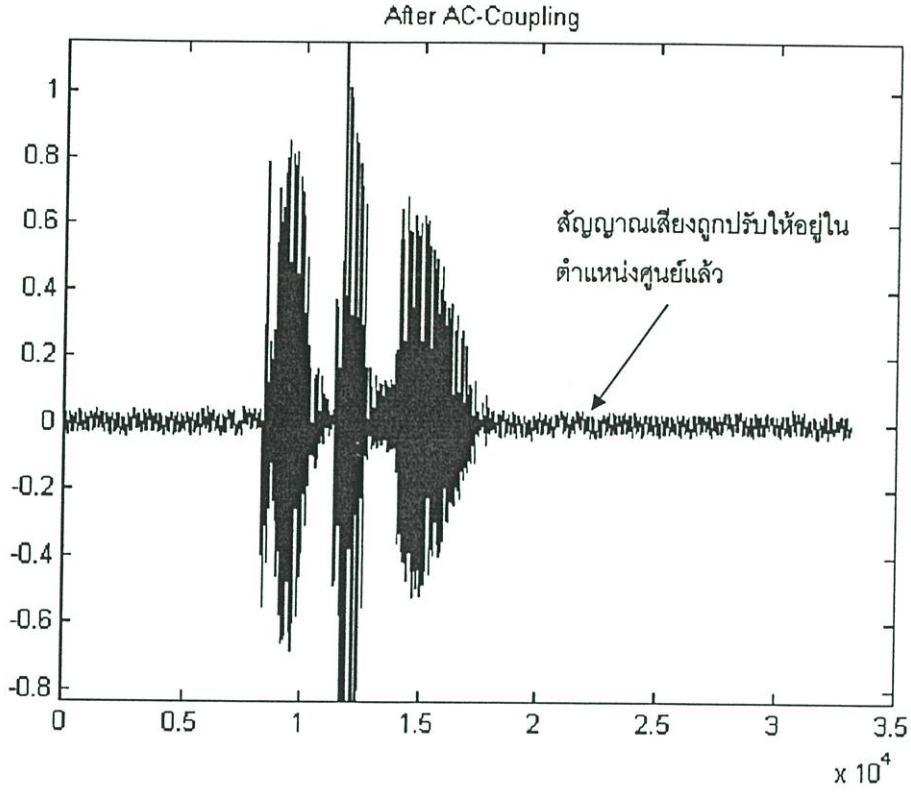
6.3.1 การทดสอบกระบวนการ AC-Coupling

กระบวนการ AC-Coupling ทำหน้าที่กรองสัญญาณเสียงให้เฉพาะสัญญาณ AC ผ่านเท่านั้น และจะกันไม่ให้ส่วนประกอบของสัญญาณไฟตรง DC เข้ามาได้ ซึ่งสัญญาณเสียงที่มีลักษณะของส่วนประกอบสัญญาณไฟตรง DC แสดงดังรูปที่ 6.1



รูปที่ 6.1 สัญญาณเสียงอินพุตที่มีส่วนประกอบของสัญญาณไฟตรง DC

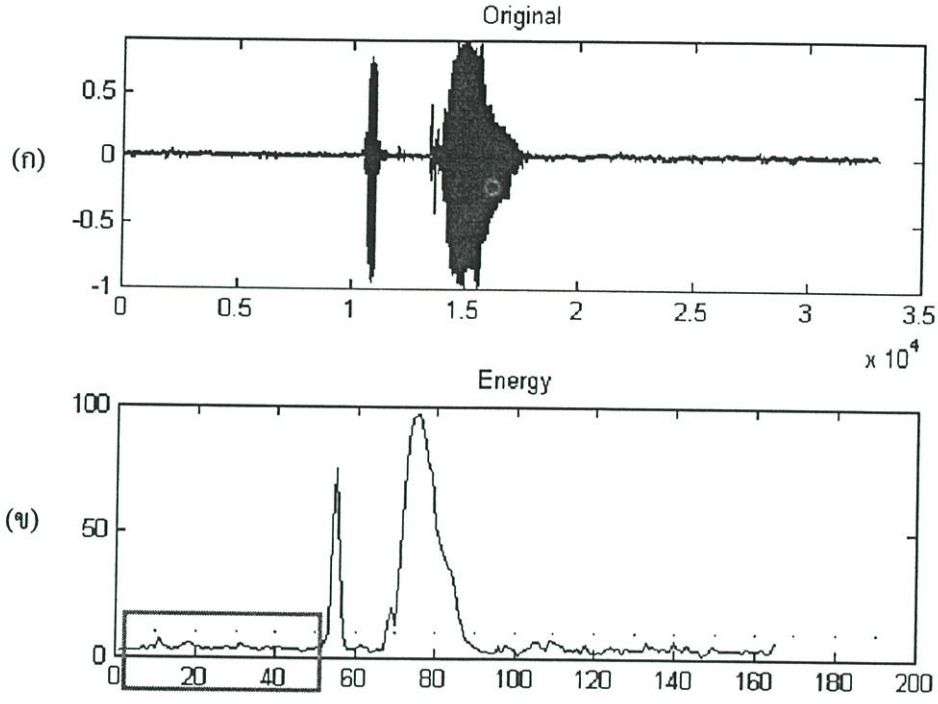
จากรูปที่ 6.1 เมื่อนำสัญญาณเสียงที่มีองค์ประกอบของสัญญาณไฟตรง DC มาผ่านกระบวนการทำ AC-Coupling ผลลัพธ์จะทำให้ส่วนประกอบของสัญญาณไฟตรง DC นั้นหาย และทำให้ระดับของสัญญาณเสียงมาอยู่ที่ระดับศูนย์ดังแสดงในรูปที่ 6.2



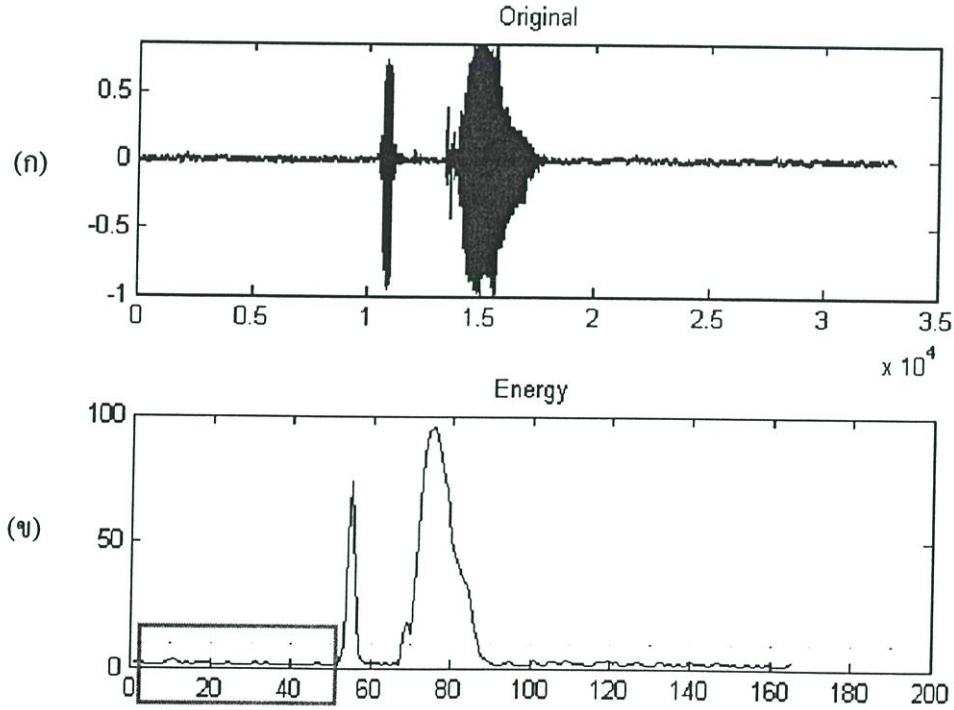
รูปที่ 6.2 สัญญาณเสียงอินพุตที่ถูกตัดส่วนขององค์ประกอบไฟตรงออกแล้ว

ถ้าปล่อยให้สัญญาณเสียงที่มีส่วนประกอบของสัญญาณไฟตรง DC ผ่านเข้าไปยังส่วนการหาค่าพลังงานจะส่งผลให้ค่าระดับพลังงานของบริเวณที่ไม่ใช่สัญญาณเสียงนั้นสูงขึ้น และ ในบางครั้งอาจจะมีผลต่อการตัดสินใจในกระบวนการตัดหัวท้าย (Endpoint detection) ของสัญญาณเสียงคำพูดได้ ดังแสดงในรูปที่ 6.3

จากรูปที่ 6.3 เป็นการแสดงให้เห็นว่าถึงแม้สัญญาณเสียงจะดูเหมือนว่าไม่มีส่วนประกอบของสัญญาณไฟตรง DC แต่เมื่อทำการหาค่าพลังงานของสัญญาณเสียงออกมาจะเห็นความแตกต่างระหว่างสัญญาณเสียงที่ผ่านการทำ AC-Coupling มาก่อนดังแสดงในรูปที่ 6.4 กับสัญญาณเสียงที่ไม่ได้ผ่านการทำ AC-Coupling ดังแสดงในรูปที่ 6.3 ซึ่งจากรูปจะเห็นว่าบริเวณที่ไม่ใช่สัญญาณเสียงนั้นมีพลังงานที่สูงพอสมควรเมื่อเทียบกับรูปที่ 6.4



รูปที่ 6.3 การหาพลังงานของเสียงคำพูดโดยไม่ผ่านกระบวนการ AC-Coupling (ก) สัญญาณเสียงอินพุต (ข) พลังงานของสัญญาณเสียง



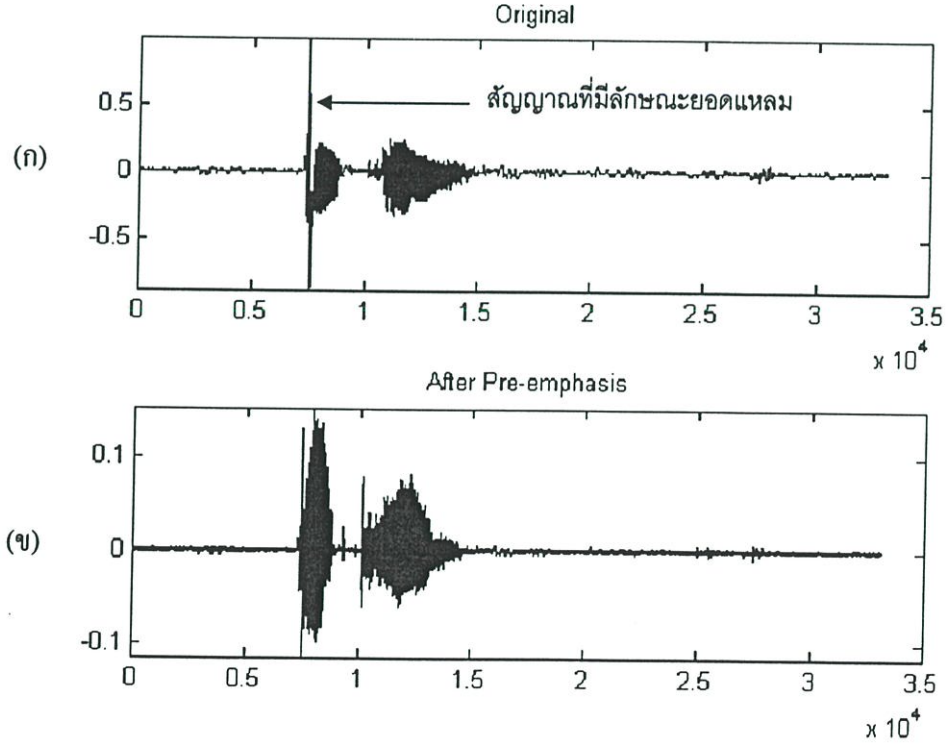
รูปที่ 6.4 พลังงานของเสียงคำพูดโดยผ่านกระบวนการ AC-Coupling (ก) สัญญาณเสียงอินพุต (ข) พลังงานของสัญญาณเสียง

6.3.2 การทดสอบกระบวนการ Pre-emphasis

Pre-emphasis คือ ตัวกรองชนิดความถี่สูงผ่าน (High-pass filter) ซึ่งตัวกรองชนิดนี้ทำหน้าที่เน้นสัญญาณเสียงในช่วงความถี่สูงทำให้แอมพลิจูดของสัญญาณเสียงนั้นมีขนาดสูงขึ้น และจะเป็นการปรับปรุงคุณภาพสัญญาณเสียงซึ่งทำให้ SNR ของสัญญาณเสียงนั้นดีขึ้น

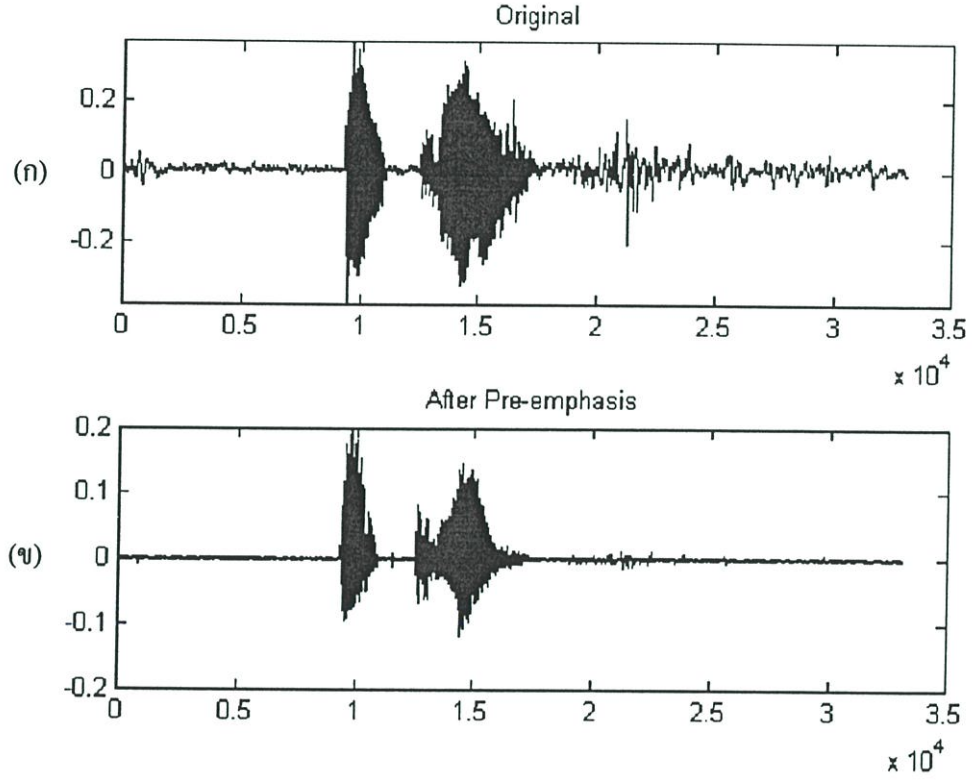
ปกติการพูดผ่านไมโครโฟนจะทำให้มีลมที่ออกจากปากไปกระแทกกับไมโครโฟนจึงทำให้เกิดเสียงที่มีลักษณะเป็นยอดแหลมปะปนเข้ามากับสัญญาณเสียงพูดด้วย ซึ่งสัญญาณเสียงรบกวนที่เกิดจากกรณีนี้ส่วนใหญ่จะอยู่ในช่วงความถี่ต่ำ ดังนั้น สัญญาณรบกวนในส่วนนี้จะสามารถถูกกำจัดออกไปได้ ดังแสดงในรูปที่ 6.5 ซึ่งจากการทดลองจะเห็นว่าเมื่อสัญญาณเสียงอินพุตผ่านกระบวนการ Pre-emphasis แล้วจะสามารถกำจัดเสียงลมที่กระแทกกับไมโครโฟนออกไปได้พอสมควร แต่ก็จะทำให้ขนาดแอมพลิจูดของสัญญาณเสียงทั้งคำพูดนั้นเล็กลงไปด้วยดังแสดงในรูปที่ 6.5 (ข) ดังนั้น เมื่อผ่านขั้นตอนการทำ Pre-emphasis นี้แล้วสัญญาณเสียงจะถูกปรับขนาดแอมพลิจูดให้มีขนาดใหญ่ดังเดิมอีกครั้งด้วยการทำนอร์มอลไลซ์ทางแอมพลิจูด

จากผลการทดลองจะสังเกตเห็นว่าสัญญาณเสียงก่อนการทำ Pre-emphasis ดังแสดงในรูปที่ 6.5 (ก) กับหลังทำ Pre-emphasis ดังแสดงในรูปที่ 6.5 (ข) จะมีความแตกต่างกันพอสมควร แต่อย่างไรก็ตามองค์ประกอบหลัก และ ความหมายของเสียงยังคงเหมือนเดิม



รูปที่ 6.5 การนำ Pre-emphasis มาช่วยลดสัญญาณรบกวนจากเสียงลมที่กระแทกกับไมโครโฟน
(ก) สัญญาณเสียงอินพุต (ข) สัญญาณเสียงหลังจากผ่านกระบวนการ Pre-emphasis

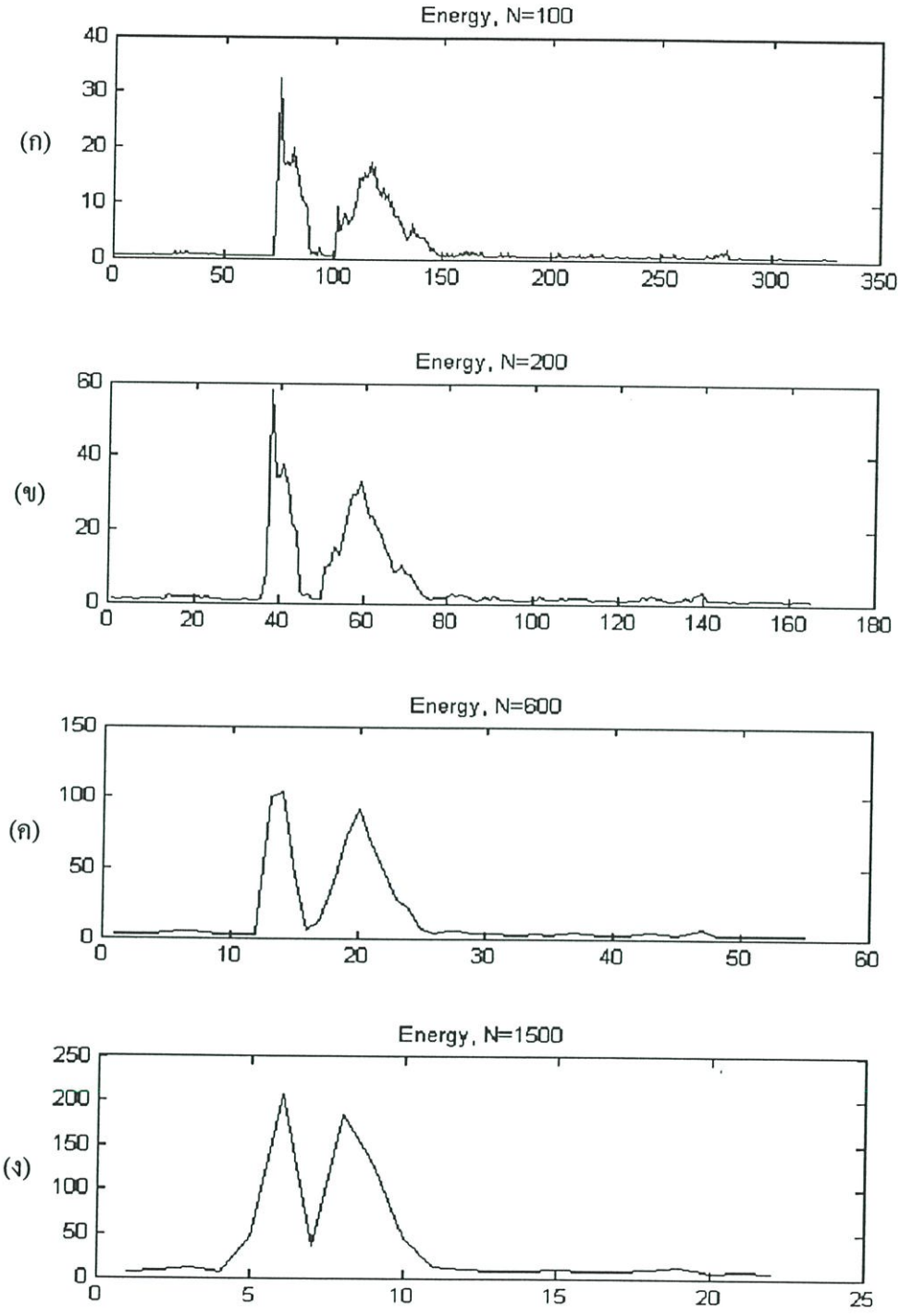
สำหรับอีกกรณีหนึ่งคือ เสียงลมหายใจทางตอนท้ายของสัญญาณเสียงคำพูดดังแสดงในรูปที่ 6.6 (ก) ซึ่งสัญญาณเสียงรบกวนนี้จะเป็นสัญญาณเสียงที่เกิดจากลมหายใจที่ออกมาจากจมูกของผู้พูด ซึ่งถือว่าเป็นสัญญาณรบกวนที่มีขนาดใหญ่ โดยเมื่อนำสัญญาณนี้ไปหาค่าพลังงานจะทำให้พลังงานตรงบริเวณสัญญาณรบกวนนี้มีขนาดสูงจนอาจจะตรวจพบว่าเป็น 1 พยางค์ได้เลย และในรูปที่ 6.6 (ข) คือสัญญาณเสียงที่ผ่านกระบวนการ Pre-emphasis ซึ่งจะลดสัญญาณรบกวนนี้ได้



รูปที่ 6.6 การนำ Pre-emphasis มาช่วยลดสัญญาณรบกวนจากเสียงลมหายใจ (ก) สัญญาณเสียงที่มีสัญญาณรบกวน (ข) สัญญาณเสียงที่ผ่านกระบวนการ Pre-emphasis

6.3.3 การทดสอบในส่วนของการหาค่าพลังงาน

การหาค่าพลังงานของสัญญาณเสียงเป็นวิธีการหนึ่งที่ใช้ในการหาตำแหน่งของคำ หรือพยางค์ของสัญญาณเสียง ซึ่งจะใช้หลักการแบ่งสัญญาณเสียงออกเป็นส่วนย่อย หรือ เป็นเฟรม จากนั้นนำสัญญาณแต่ละเฟรมนี้มาหาผลรวม ซึ่งสิ่งที่ต้องพิจารณาในการหาค่าพลังงานของสัญญาณเสียงคือการกำหนดขนาดของเฟรมสัญญาณเสียงซึ่งจะมีผลต่อลักษณะของพลังงานของสัญญาณเสียงดังแสดงในรูปที่ 6.7

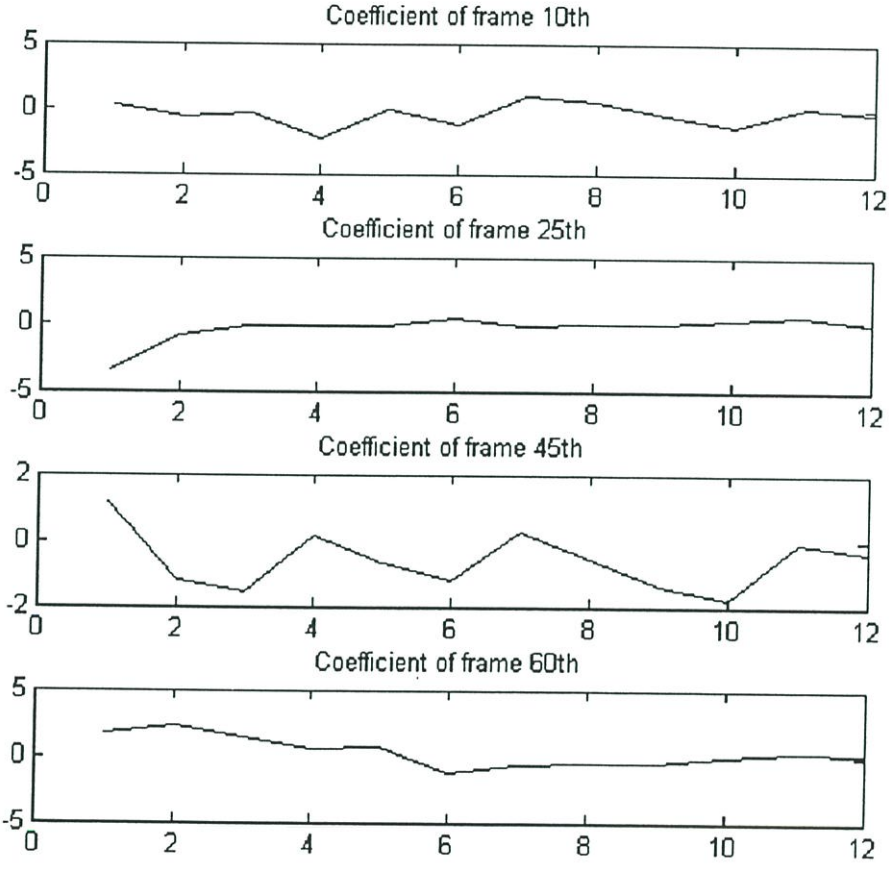


รูปที่ 6.7 ผลของการปรับเปลี่ยนขนาดเฟรมของสัญญาณเสียงด้วยค่าที่ต่างกัน

จากรูปที่ 6.7 จะเห็นว่ายิ่งกำหนดขนาดส่วนย่อยของแต่ละเฟรมมากยิ่งขึ้นจะทำให้รายละเอียดของพลังงานของสัญญาณเสียงนั้นน้อยลง ซึ่งถ้ากำหนดให้แต่ละเฟรมย่อยมีขนาดมากเกินไปจะทำให้ค่าพลังงานตรงบริเวณรอยต่อของแต่ละพยางค์นั้นยกสูงขึ้นจนถึงระดับเส้นการตัดสินใจ (Threshold) ได้ ซึ่งจะส่งผลให้สัญญาณเสียง 2 พยางค์ถูกมองเป็น 1 พยางค์ได้

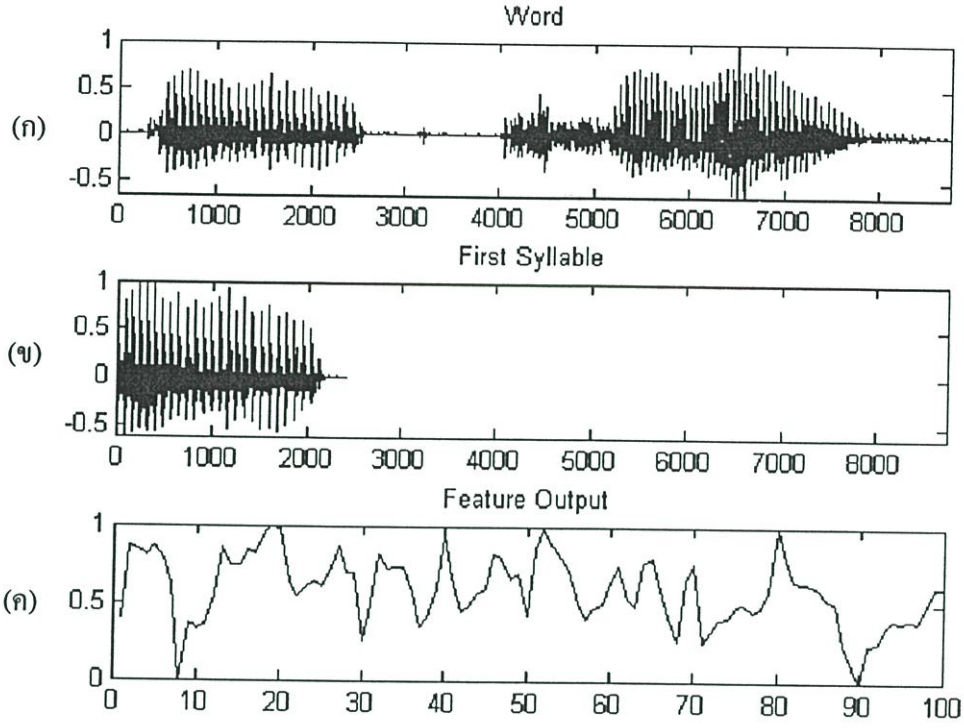
6.4 การทดลองในส่วนของภาคการดึงคุณลักษณะเด่นของสัญญาณเสียง

ในวิทยานิพนธ์นี้จะใช้การดึงคุณลักษณะเด่นของสัญญาณเสียงด้วยวิธี Mel-Frequency cepstral coefficients (MFCC) โดยจะแบ่งสัญญาณเสียงออกเป็นส่วนย่อย หรือ เฟรมขนาด N = 256 ข้อมูล หรือ ประมาณ 23 มิลลิวินาที โดยแต่ละเฟรมที่ผ่านการดึงคุณลักษณะเด่นด้วยวิธี MFCC จะได้ สัมประสิทธิ์เซปสตรัม จำนวน 12 ตัว คือ C1 ถึง C12 ดังแสดงในรูปที่ 6.8

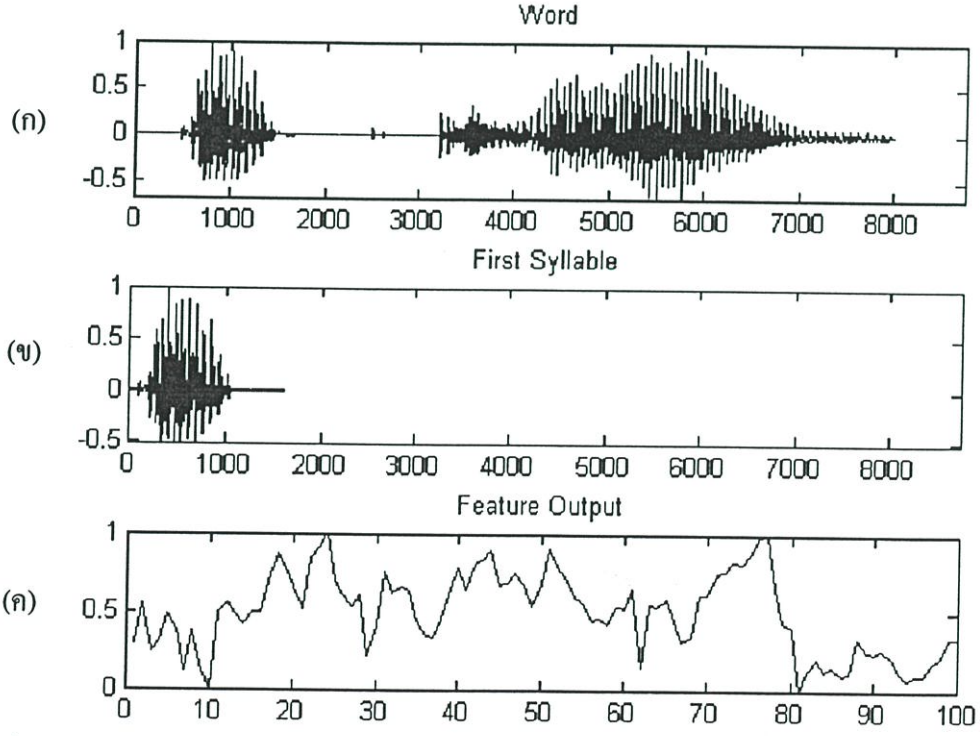


รูปที่ 6.8 สัมประสิทธิ์เซปตรัมของเฟรมที่ 10 25 45 และ 60 ตามลำดับ

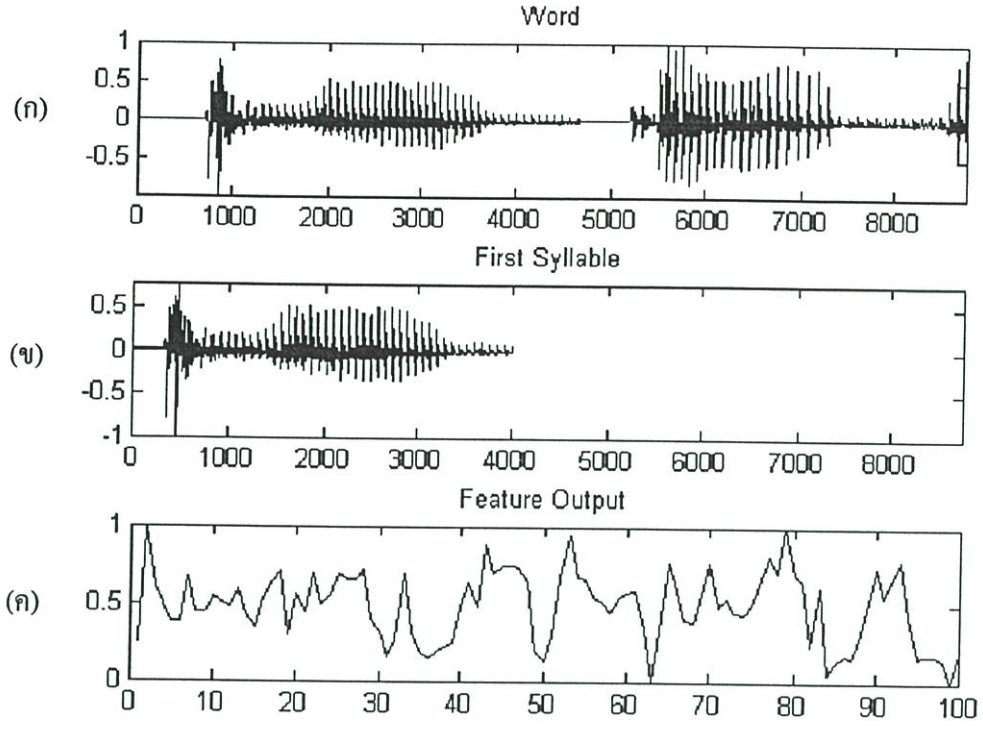
การดึงคุณลักษณะเด่นของสัญญาณเสียงจะกระทำ 2 ส่วนคือ 1.) ส่วนของทั้งคำพูด และ 2.) ส่วนของเฉพาะพยางค์แรก ซึ่งคุณลักษณะเด่นที่ได้จะเป็นดังแสดงในรูปที่ 6.8 จากนั้นข้อมูลคุณลักษณะเด่นของสัญญาณเสียงที่ได้ทั้ง 2 ส่วนนี้จะนำมารวมกันดังที่ได้กล่าวรายละเอียดไว้แล้วในบทที่ 3 ซึ่งคุณลักษณะเด่นของสัญญาณเสียงที่ได้จะเป็นอินพุตของภาคการรู้จำด้วยโครงข่ายประสาทเทียม (Neural network) ดังแสดงรูปที่ 6.9 ถึง รูปที่ 6.16



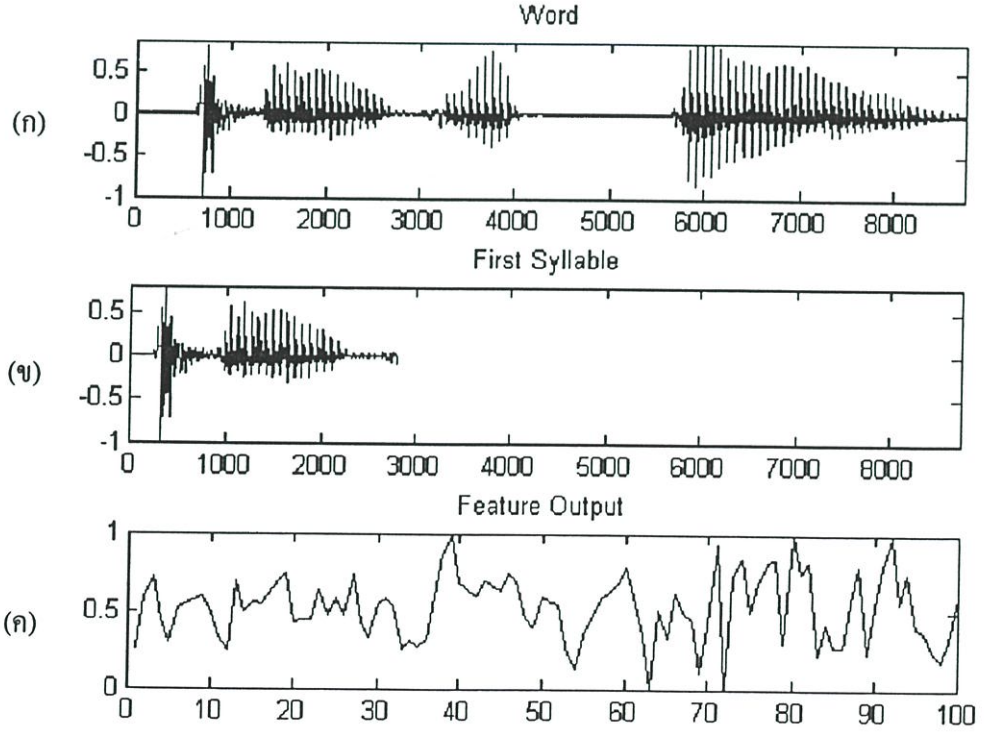
รูปที่ 6.9 คุณลักษณะเด่นของคำพูด "เปิดเครื่อง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



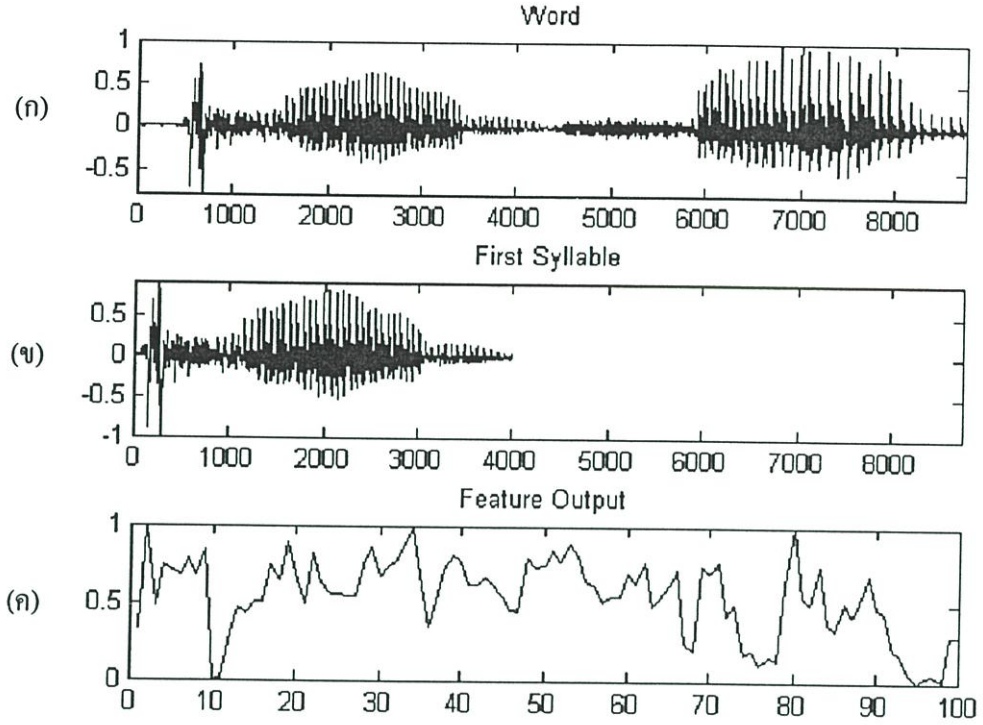
รูปที่ 6.10 คุณลักษณะเด่นของคำพูด "ปิดเครื่อง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



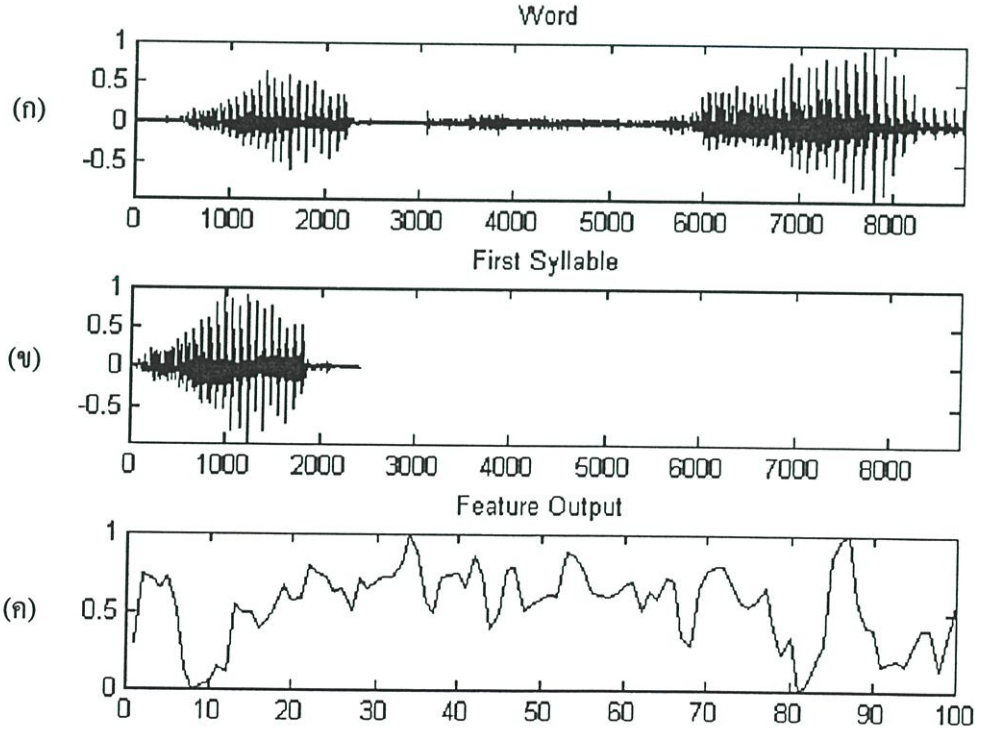
รูปที่ 6.11 คุณลักษณะเด่นของคำพูด "เพลงก่อนหน้า" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



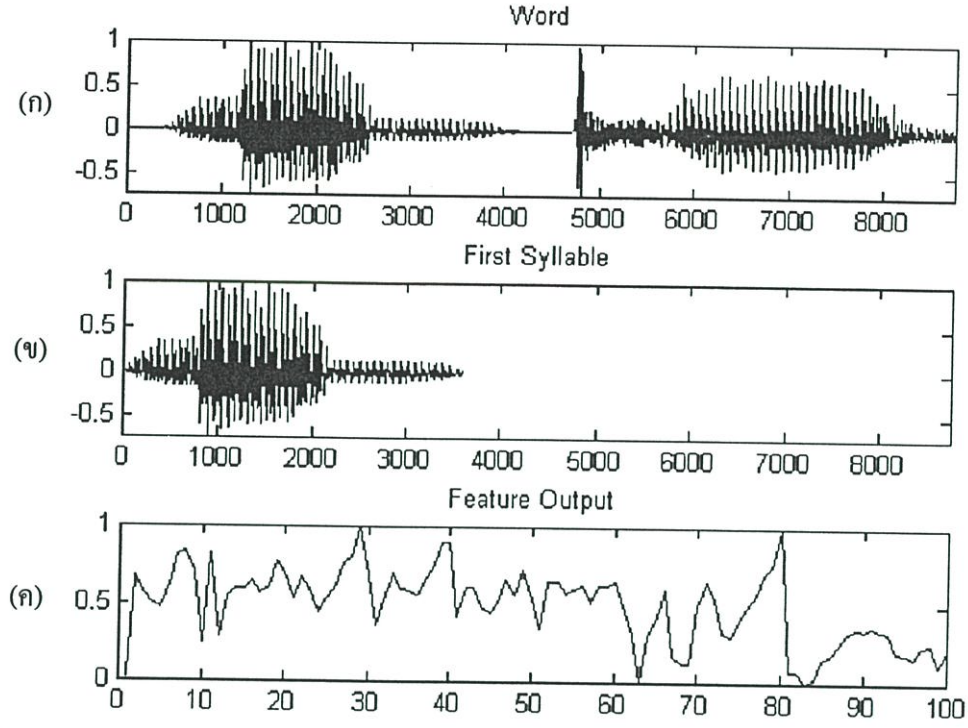
รูปที่ 6.12 คุณลักษณะเด่นของคำพูด "เพลงถัดไป" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



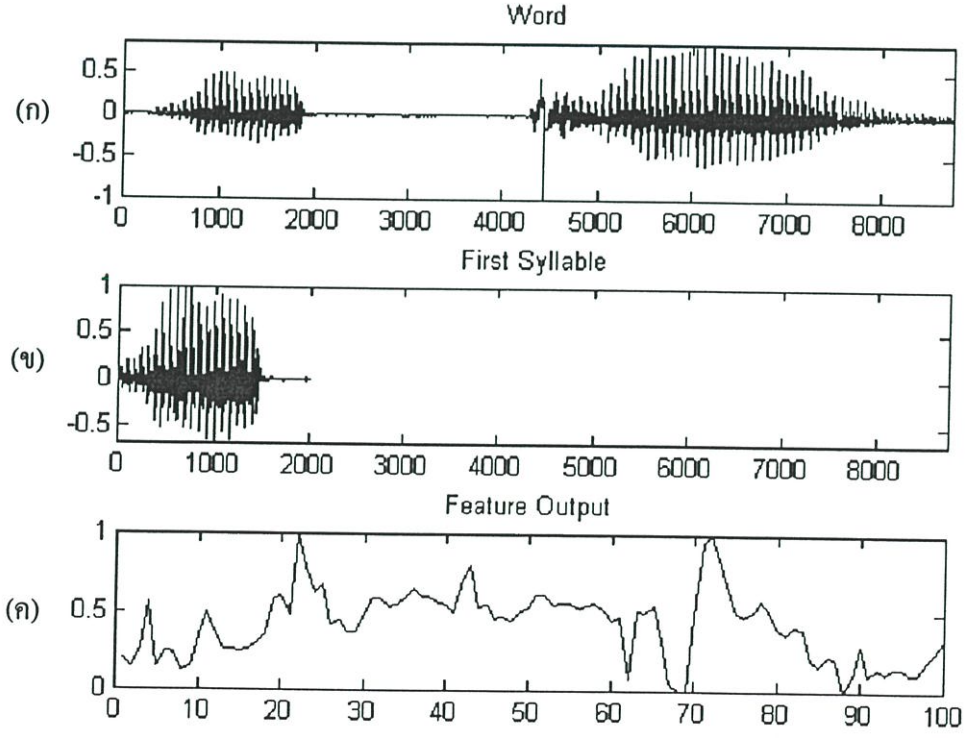
รูปที่ 6.13 คุณลักษณะเด่นของคำพูด "เพิ่มเสียง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



รูปที่ 6.14 คุณลักษณะเด่นของคำพูด "ลดเสียง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



รูปที่ 6.15 คุณลักษณะเด่นของคำพูด "เล่นเพลง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม



รูปที่ 6.16 คุณลักษณะเด่นของคำพูด "หยุดเพลง" (ก) สัญญาณเสียงคำพูด (ข) สัญญาณเสียงพยางค์แรก (ค) ข้อมูลคุณลักษณะเด่นอินพุตของโครงข่ายประสาทเทียม

6.5 การทดลองในส่วนของภาคการรู้จำ

การทดสอบระบบการรู้จำเสียงพูดคำไทย เพื่อใช้ในการควบคุมโปรแกรมเล่นเพลงวินแอมป์นั้นแสดงดังตารางที่ 6.1 ซึ่งการทดลองนี้ทำงานครอบคลุมเสียงคำพูดจำนวน 8 คำ ได้แก่คำว่า "เปิดเครื่อง" "ปิดเครื่อง" "เพลงก่อนหน้า" "เพลงถัดไป" "เพิ่มเสียง" "ลดเสียง" "เล่นเพลง" และ "หยุดเพลง" โดยทำการทดสอบกับบุคคลเพศชายจำนวน 9 คน และ เพศหญิงจำนวน 9 คน มีอายุระหว่าง 19 – 25 ปี โดยได้เก็บบันทึกเสียงพูดของแต่ละคนคำละ 10 ครั้ง รวมเป็น 1,440 คำ และใช้ผู้สอนระบบเป็นผู้ชาย 1 คน และ ผู้หญิง 1 คน โดยแต่ละคนพูดคำละ 5 ครั้ง

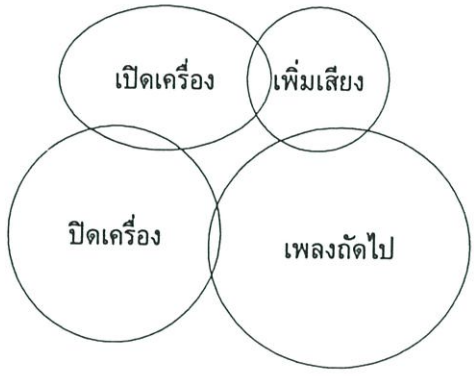
การสอนระบบ (Training) ด้วยโปรแกรมเครื่องมือ Qnet 2000 โดยมีข้อมูลอินพุตเท่ากับ 80 ข้อมูล โดยแบ่งเป็นเสียงบุคคลเพศชาย 1 คนมีข้อมูลอินพุต 40 ข้อมูล และ มีข้อมูลอินพุตเป็นเสียงบุคคลเพศหญิง 1 คนมีข้อมูลอินพุต 40 ข้อมูล เมื่อทำการสอนระบบจนเสร็จจะใช้เวลาประมาณ 5 นาที โดยมีการกำหนดจำนวนรอบของการสอนระบบเท่ากับ 10,000 รอบ

ตารางที่ 6.1 ผลการทดลองวิธีใหม่เปรียบเทียบกับวิธีเก่า

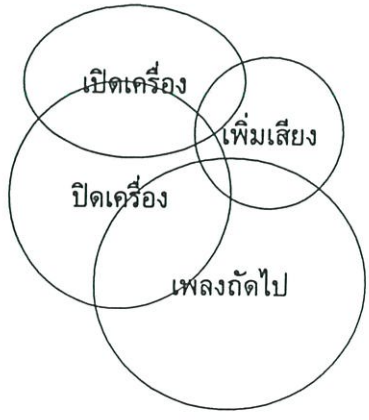
คำพูด	ชาย		หญิง	
	วิธีเก่า	วิธีใหม่	วิธีเก่า	วิธีใหม่
"เปิดเครื่อง"	86.67%	88.89%	88.89%	94.44%
"ปิดเครื่อง"	68.89%	95.56%	90.00%	90.00%
"เพลงก่อนหน้า"	91.11%	94.44%	94.44%	96.67%
"เพลงถัดไป"	92.22%	83.33%	91.11%	88.89%
"เพิ่มเสียง"	92.22%	93.33%	93.33%	93.33%
"ลดเสียง"	72.22%	88.89%	83.33%	94.44%
"เล่นเพลง"	88.89%	86.67%	95.56%	96.67%
"หยุดเพลง"	78.89%	84.44%	90.00%	92.22%

การทดลองเป็นการเปรียบเทียบการทำงานระหว่างวิธีเก่าเทียบกับวิธีใหม่ซึ่งเป็นวิธีที่นำเสนอในวิทยานิพนธ์นี้ โดยเมื่อวิธีใหม่ คือ การหาคุณลักษณะเด่นสัญญาณเสียงของทั้งคำพูดร่วมกับพยางค์แรกของคำพูดแล้วนำมารวมกัน และในส่วนของวิธีเก่า คือ การหาคุณลักษณะเด่นของสัญญาณเสียงทั้งคำพูดเพียงอย่างเดียว โดยผลลัพธ์จากการทดลองจะได้ว่าวิธีใหม่มีเปอร์เซ็นต์ความถูกต้องของการรู้จำที่สูงกว่าวิธีเก่าที่ใช้การดึงคุณลักษณะเด่นกับคำพูดแต่เพียงอย่างเดียว โดยวิธีใหม่ให้เปอร์เซ็นต์ความถูกต้องถึง 91.38% ในขณะที่เปอร์เซ็นต์ความถูกต้องของวิธีเก่ามีเพียง 87.36%

จากผลการทดลองในตารางที่ 6.1 จะเห็นว่าวิธีเก่าสามารถรู้จำสัญญาณเสียงบางคำได้ดีกว่าวิธีใหม่ ซึ่งได้แก่ “เพลงถัดไป” และ “เล่นเพลง” สำหรับผู้ชาย และ “เพลงถัดไป” สำหรับผู้หญิง แต่อย่างไรก็ตามเปอร์เซ็นต์โดยรวมการรู้จำของวิธีใหม่ก็ยังทำได้ดีกว่าวิธีเก่า ซึ่งการเลือกข้อมูลที่จะนำมาใช้สอนระบบโครงข่ายประสาทเทียมนั้นถือว่าเป็นสิ่งที่สำคัญมาก ซึ่งถ้าเลือกข้อมูลได้ดีจะส่งผลให้การแยกแยะด้วยโครงข่ายประสาทเทียมนั้นแม่นยำมากขึ้น และ เนื่องคุณลักษณะเด่นของสัญญาณเสียงของคำพูด 1 คำที่พูดซ้ำกันในแต่ละครั้งจะได้ลักษณะของคุณลักษณะเด่นที่ไม่เหมือนกัน แต่จะมีลักษณะที่คล้ายคลึงกัน ดังนั้น ความไม่แน่นอนนี้จะทำให้สัญญาณเสียงคำพูดคำอื่นๆ อาจจะมีคุณลักษณะเด่นที่ซ้ำหรือคล้ายกับคุณลักษณะเด่นของอีกคำหนึ่งได้ ดังแสดงในรูปที่ 6.17



(ก)



(ข)

รูปที่ 6.17 การเลือกคุณลักษณะเด่นของสัญญาณเสียง (ก) การเลือกคุณลักษณะเด่นของสัญญาณเสียงที่ดี (ข) การเลือกคุณลักษณะเด่นของสัญญาณเสียงที่ไม่ดี

จากการทดสอบนำอัลกอริทึมการรู้จำสัญญาณเสียงคำพูดนี้ไปควบคุมโปรแกรมเล่นเพลงวินแอมป์เวอร์ชัน 5.5 ผ่านโปรแกรม CLAMP ที่ทำหน้าที่เป็นไดร์เวอร์หรือเป็นตัวกลางระหว่างโปรแกรมที่พัฒนาขึ้นกับโปรแกรมเล่นเพลงวินแอมป์ ซึ่งผลการเชื่อมต่อนั้นสามารถควบคุมโปรแกรมเล่นเพลงวินแอมป์ได้เป็นอย่างดี

บทที่ 7
สรุปผลการวิจัยและข้อเสนอแนะ

7.1 สรุปผลการศึกษาระบบการรู้จำเสียงพูดคำไทยเพื่อควบคุมโปรแกรมวินแอมป์

วิทยานิพนธ์ฉบับนี้นำเสนอกระบวนการรู้จำเสียงพูดคำไทยเพื่อใช้ในการควบคุมโปรแกรมเล่นเพลงวินแอมป์ (Winamp player) ซึ่งคำพูดที่ใช้ในการสั่งงานจะเป็นแบบสองพยางค์และแบบสามพยางค์ โดยใช้วิธีการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature Extraction) ด้วยวิธีการหาค่าสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient : MFCC) และ ใช้การรู้จำด้วยแบบจำลองโครงข่ายประสาทเทียม (Neural Network) ซึ่งในส่วนของการดึงคุณลักษณะเด่นของสัญญาณเสียง (Feature extraction) จะกระทำกับสัญญาณเสียงคำพูดทั้งคำ และกับเฉพาะพยางค์แรกของคำพูด ซึ่งพบว่าผลการรู้จำได้ 91.38 เปอร์เซ็นต์ เทียบกับวิธีที่ใช้การดึงคุณลักษณะเด่นของสัญญาณเสียงคำพูดเพียงอย่างเดียวได้ 87.36 เปอร์เซ็นต์ โดยใช้เสียงของผู้สอนระบบเป็นเพศชาย 1 คน และ เพศหญิง 1 คน

7.2 ข้อเสนอแนะและแนวทางในการพัฒนา

จากผลการทดลองจะเห็นว่าการรู้จำสัญญาณเสียงบางคำพูดยังทำได้ไม่ดีนักซึ่งสามารถเสนอแนวทางในการพัฒนาได้ดังนี้ คือ

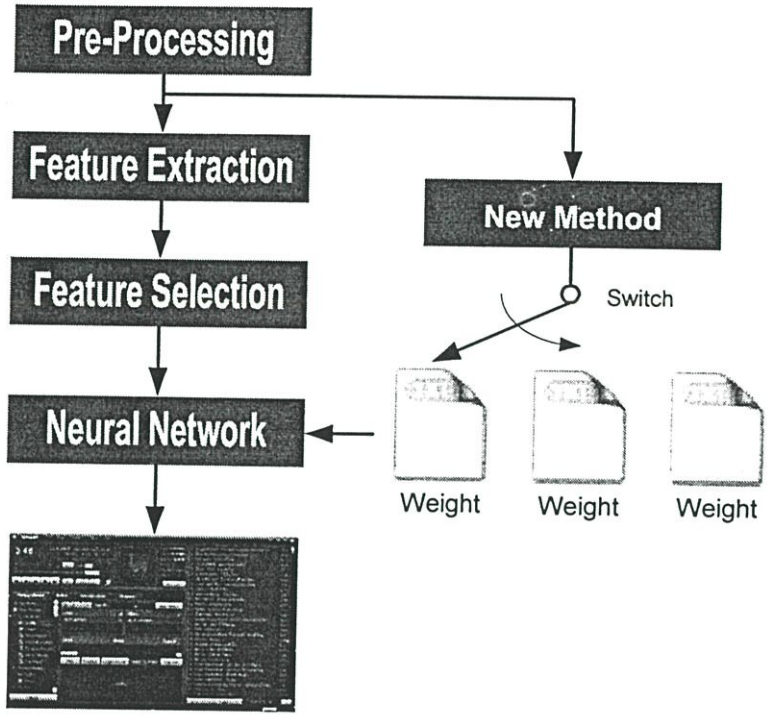
1. เพื่อลดปัญหาเกี่ยวกับลมที่จะมากระทบกับไมโครโฟนขณะที่ทำการพูดบันทึกเสียงควรจะมีวัสดุกรองเสียงที่ทำจากฟองน้ำหุ้มไว้ที่ไมโครโฟนเสมอ

2. ทดลองเปลี่ยนการดึงคุณลักษณะของสัญญาณเสียง (Feature Extraction) ให้เป็นวิธี Delta cepstrum ซึ่งจะทำให้ได้คุณลักษณะเด่น (Feature) ของสัญญาณเสียงใหม่ๆ มาใช้ในการดึงคุณลักษณะเด่นของสัญญาณเสียง

3. ปรับการเลือกคุณลักษณะเด่นของสัญญาณเสียง (Feature Selection) ให้เป็นแบบอื่น

4. ทดลองเปลี่ยนแบบจำลองการรู้จำเป็นแบบอื่นนอกเหนือจากการใช้แบบจำลองโครงข่ายประสาทเทียม

5. ทำชุดค่าถ่วงน้ำหนักให้เป็นหลายชุดเพื่อให้สามารถรองรับผู้พูดที่ไม่ได้อยู่ในกลุ่มเสียงที่ใช้สอน (Training) ระบบได้มากขึ้น โดยหาวิธีใดๆ เพื่อใช้สำหรับการเลือกชุดค่าถ่วงน้ำหนักดังแสดงในรูปที่ 7.1



รูปที่ 7.1 ระบบที่สามารถเลือกชุดค่าถ่วงน้ำหนักได้

เอกสารอ้างอิง

[1] ชัย วุฒิวิวัฒน์ชัย, สุทัศน์ แซ่ตั้ง และ วารินทร์ อัจฉริยกุลพร. "ความก้าวหน้าของการพัฒนาระบบระบุผู้พูดภาษาไทย" วารสารวิชาการเนคเทค, ฉบับที่ 2, No.7 . มีนาคม 2543. หน้า 24-35

[2] ชาญชัย นุชนารถ, ปณิธาน จันทรอุดร และ วิทยา ฤกษ์สำเร็จ, 2544. "การทำอาร์ดแวร์ระบบรู้จำเสียงภาษาไทยระยะที่ 1", ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

[3] Yang S., Er M.J., and Gao Y. "A High Performance Neural-Networks-Based Speech Recognition System.", **1st International Joint Conference on Neural Networks.,** Vol.2, 2001. Pp. 1527 -1531

[4] Ethnicity group. "**Cepstrum method.**" [Online]. Available : http://www.owlnet.rice.edu/~elec532/PROJECTS98/speech/cepstrum/cepstrum.html. 1998.

[5] Wang Y. and Guan L. "An Investigation of Speech-Based Human Emotion Recognition." **IEEE 6th Workshop on Multimedia Signal Processing.,** Sept. 2004. Pp. 15-18

[6] Gold, B. and Morgan, N. 1999. **Speech and Audio Signal Processing.** Danvers : John Wiley & Sons, Inc.

[7] Mueen F, Ahmed A, Sanaullah, Gaba A. "Speech Recognition Using Artificial Neural Networks." **ISCON apos.,** vol.1, issue 16-17, Aug. 2002. Pp. 99-102

[8] Jittiwarangkul N., Jitapunkul S., Luksaneeyanavin S., Ahkuputra V. and Wutiwiwatchai C. "Thai Syllable segmentation for Connected Speech Based on Energy." **IEEE Asia-Pacific Conference on Circuits and Systems.,** Nov. 1998. Pp. 169 – 172

[9] ประณิต อ่อนไสว และ พรชัย ธรรมานุรัตพันธุ์, 2542. "การจดจำเสียงพูด.", ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[10] Yao M., Hu J. and Gu Q. "A Mixed Parameter Method Based on MFCC and Fractal Dimension for Speech Recognition." **IEEE International Conference on Information Acquisition.,** Aug. 2006. Pp. 1144-1146

[11] **"11-2 MFCC."** [Online]. Available : http://neural.cs.nthu.edu.tw/jang/books/audioSignalProcessing/11.2-speechFeatureMfcc.asp?title=11-2%20MFCC

[12] Gemello R., Albesano D. and Mana F. "Multi-Source Neural Networks for Speech Recognition." **IJCNN apos.,** vol.5 Oct.1999. Pp. 2946 – 2949

[13] Milner B. and Shao X. "Clean speech reconstruction from MFCC vectors and fundamental frequency using an integrated front-end", **Speech Communication.,** Vol. 48, Issue 6, June. 2006. Pp.697-715

[14] Ricotti L.P. "Multitapering and a wavelet variant of MFCC in speech recognition." **IEE Proceeding, Image & Signal Processing.,** vol. 152, issue 1, Feb. 2005. Pp. 29-35

[15] ลัญฉกร วุฒิสิทธิกุลกิจ. 2547. **พื้นฐานกรรมวิธีสัญญาณดิจิทัล.** กรุงเทพฯ : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

[16] Burileanu C. and Popescu V. "Isolated Word Recognition Engine Implemented on Java ™ Platform." **Proc. ECIT2004.,** July. 2004. Pp.1-18

[17] Stergiou C. and Siganos D. **"Neural Networks."** [Online]. Available : http://www.doc. ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

[18] Sivanandam, S.N. Sumathi, S. Deepa, S.N. 2006. **Introduction to Neural Networks Using Matlab 6.0.** New Delhi : Tata McGraw-Hill.

[19] Hagan, M.T. Demuth, H.B. Beale, M. 1996. **Neural network design.** Boston : PWS Publishing Company.

[20] Vesta Services, Inc. **"Qnet 2000."** [Online]. Available : http://www.qnetv2k.com/Qnet 2000Manual/contents2000.htm

[21] Picone, J. "Continuous speech recognition using hidden Markov models." **ASSP Magazine.**, vol. 7, issue 3, July. 1990. Pp.26-41

[22] Polur, P.D. Ruobing Zhou. Jun Yang. Adnani, F. and Hobson, R.S. "Isolated speech recognition using artificial neural networks." **Proceedings of the 23rd Annual International Conference of the IEEE.**, vol. 2, Oct. 2001. Pp. 1731-1734

[23] Vuckovic, V. "Dynamic time-warping method for isolated speech sequence recognition." **TELSIKS 2001.**, Vol. 1, Sept. 2001, Pp.257-260

[24] Blinn, J.F. "What's that deal with the DCT?" **IEEE Computer Graphics & Applications.**, vol. 13, issue 4, July. 1993. Pp.78 – 83

[25] Rabiner, L. and Juang, B.H. 1993. **Fundamentals of speech recognition.** Prentice-Hall International.

ภาคผนวก ก
ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] จักรพันธ์ จิตรทรัพย์ อุทัย ศรีธีระวิโรจน์ และ สมเกียรติ อุดมหรรษากุล. "การแยกพยางค์เสียงพูดภาษาไทยโดยใช้การแปลงเวฟเล็ต" การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 3 (NCCIT 2007), ฉบับที่ 1, 25-26 พฤษภาคม 2550. หน้า 293-298.

[2] J. Jitsup, U-thai Sritheeravirojana and S. Udomhunsakul, "Thai Word Segmentation of Human Speech Using Stationary Wavelet Transform," Proceeding of the 2007 Asia-Pacific Conference on Communications (APCC 2007), no. 1, pp 29-32, 18-20 October 2007.

[3] จักรพันธ์ จิตรทรัพย์ และ สมเกียรติ อุดมหรรษากุล. "การรู้จำเสียงพูดคำไทยด้วยวิธีการเอ็มเอ็ฟซีซี และ โครงข่ายประสาทเทียม" การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30 (EECON-30), ฉบับที่ 2, 25-26 ตุลาคม 2550. หน้า 833 – 836.

[4] S. Chanthamenavong, J. Jitsup and S. Udomhunsakul, "Improved Syllable Segmentation of Connected LAO Digit Speech Using Fundamental Frequency," Proceeding of the Joint International Conference on Information Communication Technology (JICT 2007), pp 158-161, 19-22 December 2007.

ภาคผนวก ข

คำสั่งที่มีใช้ในโปรแกรม CLAMP.EXE

โปรแกรม CLAMP คือ โปรแกรมที่ทำหน้าที่สั่งงานโปรแกรมวินแอมป์ผ่านทาง Command line หรือ การพิมพ์คำสั่งบน DOS โดยโปรแกรมนี้ทำงานโดยลำพัง (Standalone) คือ ไม่ต้องมีกระบวนการติดตั้งโปรแกรม ไม่ต้องมีโปรแกรมอื่นๆภายนอก เช่น DLL หรือ Winamp plug-in ซึ่งโปรแกรมนี้สามารถดาว์นโหลดได้จากทาง http://membres.lycos.fr/clamp/ ซึ่งเป็นโปรแกรมฟรีแวร์ โดยคำสั่งต่างๆ ที่สามารถควบคุมโปรแกรมวินแอมป์ได้มีดังนี้ คือ

คำสั่งในการควบคุมการเปิดปิดโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
START	Start Winamp
QUIT	Exit Winamp
RESTART	Restart Winamp

คำสั่งในการควบคุมโปรแกรมวินแอมป์ที่ใช้งานบ่อย

คำสั่ง	ลักษณะการทำงาน
PLAY	Play (current file) - Quits Stopped or Pause mode
STOP	Stop playing
STOPFADE	Stop playing with fadout
STOPAFTER	Stop playing after current track (returns now, stops later)
PAUSE	Toggle pause mode
PAUSE ON|OFF	Sets pause mode NEW

คำสั่ง	ลักษณะการทำงาน
PLAYPAUSE	*Same as PAUSE*
NEXT	Play next song
PREV	Play previous song
FWD	Forward 5 seconds
FORWARD	*Same as above*
REW	Rewind 5 seconds
REWIND	*Same as above*
RESTART	Restart current track from beginning (not working with Winamp 2)
JUMP <time>	Seek to <time> (in millisecs) NEW

คำสั่งที่ใช้กำหนดโหมดการทำงานของโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
REPEAT	Toggle Repeat mode
SWREPEAT	*Same as above*
REPEAT ON	Set Repeat mode ON
REPEAT=1	*Same as above*
REPEAT OFF	Set Repeat mode OFF
REPEAT=0	*Same as above*
REPEAT STATUS	Query REPEAT status (ON, OFF)

คำสั่ง	ลักษณะการทำงาน
GETREPEAT	*Same as above*
RANDOM	Toggle Random mode
RANDOM ON	Set Random mode ON
RANDOM=1	*Same as above*
RANDOM OFF	Set Random mode OFF
RANDOM=0	*Same as above*
RANDOM STATUS	Query RANDOM status (ON, OFF)
GETSHUFFLE	*Same as above*

คำสั่งควบคุม Playlist ในโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
PLADD <file>	Add file(s) to end of playlist (like drag-n-drop)
LOAD <file>	*Same as above*
PLCLEAR	Clear Playlist
CLEAR	*Same as above*
PL	Show/Hide Winamp Playlist window
PLWIN	*Same as above*
PLPOS	Query Playlist position (requires Winamp 2.05+)
PLFIRST	Play first item of playlist

คำสั่ง	ลักษณะการทำงาน
PLLAST	Play last item of playlist
PLSET <num>	Set current playlist item (note this does not interfere with curring playing, if needed, use /PLAY after to go to this item)
LOADNEW <file>	Same as /PLCLEAR /PLADD <file>
LOADPLAY <file>	*Shortcut for /PLCLEAR /PLADD <file> /PLAY* NEW
PLSAVE <file>	Saves current playlist to <file> (as a M3U file) NEW

คำสั่งควบคุมโวลุ่มในโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
VOLUP [X]	Volume up
VOLDN [X]	Volume down
VOLSET <value>	Volume set (scale 0-255)
VOL=<value>	Volume set (scale 0-100) NEW
VOLMAX	Volume max
VOLMIN	Volume min (no sound)

คำสั่งควบคุมโวลุ่มของ Windows

คำสั่ง	ลักษณะการทำงาน
WAV MUTE ON	Mutes speaker NEW
WAV MUTE OFF	Unmutes speaker NEW
WAV VOLGET	Prints current Windows volume as two figures (left speaker, right speaker) NEW

คำสั่ง	ลักษณะการทำงาน
WAV VOLSET <value>	Sets volume (for both speakers) on a 0-65535 scale NEW
WAV VOLSET MIN	Sets volume to zero (for both speakers) NEW
WAV VOLSET MAX	Sets volume to maximum (for both speakers) NEW

คำสั่งควบคุม Equalizer ของโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
EQWIN	Toggle Eq window (Works with Classic skins only)
EQINFO	Query Eq parameters (10 bands, Preamp, Status, Autoload)
EQSET <parms>	Set Eq parameters (Same format as EQINFO)
EQSTATUS	Toggle Eq status (ON / OFF)
EQSTATUS ON	Set Eq status ON
EQSTATUS OFF	Set Eq status OFF

คำสั่งควบคุมการแสดงโปรแกรมวินแอมป์บนWindows

คำสั่ง	ลักษณะการทำงาน
ONTOP	Toggle Always On Top option
MAINWIN	Toggle Main Window (Show / Hide)
MINIMIZE	Minimize Winamp

คำสั่งจัดการกับ Skin ของโปรแกรมวินแอมป์

คำสั่ง	ลักษณะการทำงาน
SKINGET	Display name of current skin NEW
OPGET	Display name of output plug-in NEW

คำสั่งอื่นๆ

คำสั่ง	ลักษณะการทำงาน
CDPLAY	Play CD

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายจักรพันธ์ จิตรทรัพย์
วัน เดือน ปีเกิด	27 พฤศจิกายน 2522
ที่อยู่	หมู่บ้านพล.รพศ.1 320/53 หมู่8 ต.นิคมสร้างตนเอง อ.เมือง จ.ลพบุรี 15000
ประวัติการศึกษา	
พ.ศ. 2545	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
พ.ศ. 2550	วิศวกรรมศาสตรมหาบัณฑิต สาชาวิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง