

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการรู้จำเสียงพูด

SPEECH RECOGNITION SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 36894
วัน, เดือน, ปี..... 29 ส.ค. 2548

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการรู้จำเสียงพูด

SPEECH RECOGNITION SYSTEM

โดย

นาย นีรุตต์ จุลเดชนันทชัย 40013214

นาย พิเชษฐ์ อยู่ศรี 40013221

อาจารย์ที่ปรึกษา

ผศ. ดร. สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการรู้จำเสียงพูด

ผู้จัดทำ 1. นายนิรุทธ์ จุลเดชนันทชัย

2. นายพิเชษฐ์ อยู่ศรี


..... อาจารย์ที่ปรึกษา

(ผศ.ดร.สุรพันธุ์ เอื้อไพบูลย์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการรู้จำเสียงพูด

SPEECH RECOGNITION SYSTEM

นาย นิรุตต์ จุลเดชนันทชัย 40013214

นาย พิเชษฐ์ อยู่ศรี 40013221

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้

ลงชื่อ.....อาจารย์ที่ปรึกษา

(ผศ. ดร. สุรพันธุ์ เอื้อไพบูลย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการรู้จำเสียงพูด

นาย นิรุตต์ จุลเดชนันทชัย

นาย พิเชษฐ์ อยู่ศรี

ผศ. ดร. สุรพันธุ์ เอื้อไพบูลย์ (อาจารย์ที่ปรึกษา)

ปีการศึกษา 2542

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ เป็นการศึกษาเรื่องการรู้จำเสียงพูด โดยใช้วิธีแบบครบพาทะกัษณ์ นิวัตรเน็ตเวอร์ค เพื่อให้รู้จำเสียงพูดไม่ต่อเนื่อง และขึ้นกับผู้พูด ในการรับเสียงพูดเราจะใช้การ์ดเสียง ในขบวนการปริ-โพรเซสซิง สัญญาณในแกนเวลาของเสียงที่ใช้ทดลอง จะถูกแปลงให้เป็นสัญญาณดิจิตอล แล้วใช้การแปลงฟูเรียร์อย่างรวดเร็ว ร่วมกับการแปลงข้อมูลในแกนความถี่ ให้อยู่ในแกนฮาร์โมนิคของความถี่ เพื่อลดขนาดของข้อมูล และขยายรายละเอียดในช่วงความถี่ที่มีการเปลี่ยนแปลงมาก ได้เป็น สเปกโตรแกรมของเสียง จากนั้นนำไปฝึกสอน และทดสอบกับนิวัตร-เน็ตเวอร์ค ผลลัพธ์ที่ได้จากนิวัตรเน็ตเวอร์ค จะนำไปควบคุม สเต็ปมิ่งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPEECH RECOGNITION SYSTEM

Mr. Nirut Juladechnantachai

Mr. Phichet Yoosri

Assist. Prof. Dr. Suraphun Airphaiboon

Education Year 1999

Abstract

This thesis is the study from Speech recognition by Back Propagation Neural Network for discrete word and speaker dependent voice recognition. For input device Sound card. In pre-processing unit to find voices pattern in harmonic frequency domain by using Fast Fourier Transform. All of voice spectrogram are using for training and testing process in neural network. The result from neural network use control stepping motor.

สารบัญ

	หน้า
บทคัดย่อ	a
Abstract	b
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 กายวิภาคของระบบการเปล่งเสียงของมนุษย์	4
2.2 กระบวนการผลิตเสียงพูด	5
2.3 เสียงพูดของมนุษย์	6
2.4 ความถี่ฟอร์แมนท์	6
2.5 แบบจำลองระบบกำเนิดเสียงพูด	7
2.6 รูปแบบของไฟล์เสียง	8
2.7 การแบ่งช่วงสัญญาณ	10
2.8 การวินโดว์	10
2.9 การวิเคราะห์ฟูรีเยร์	11
2.10 นิเวียร์เน็ตเวอร์ค	14
2.11 สเตปป์มอเตอร์	29
บทที่ 3 การออกแบบและการสร้าง	
3.1 ส่วนปรีโพรเซสซิ่ง	31
3.2 ส่วนการตัดสินใจ	32
3.3 ส่วนควบคุมมอเตอร์	33
บทที่ 4 การดำเนินการและทดลอง	
4.1 การดำเนินการ	34
4.2 ผลการทดลอง	37
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	
5.1 สรุปผลการทดลอง	59
5.2 วิจารณ์ผลการทดลอง	59
5.3 แนวทางการพัฒนา	60
ภาคผนวก	62
กิตติกรรมประกาศ	76
บรรณานุกรม	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1	แสดงบล็อกการทำงานของการแยกแยะสัญญาณเสียง	1
รูปที่ 1.2	แสดงการแบ่งช่วงสัญญาณ	2
รูปที่ 1.3	ตัวอย่างของฟังก์ชันหน้าต่างแบบแฮมมิง ที่ $N=50$	3
รูปที่ 2.1	ภาพตัดขวางแสดงอวัยวะในระบบการพูดของมนุษย์	4
รูปที่ 2.2	รูปกล่องเสียงขณะ (ก) หายใจปกติ (ข) หายใจลึก ๆ (ค) กำลังส่งเสียง (ง) ส่งเสียงกระซิบหรือเสียงแผ่ว	5
รูปที่ 2.3	แผนภาพระบบเสียงพูดของมนุษย์	6
รูปที่ 2.4	(ก) ตัวอย่างรูปคลื่นของสัญญาณเสียงก้อง /a/ (ข) ตัวอย่างรูปคลื่นของสัญญาณเสียงไม่ก้อง /sh/	7
รูปที่ 2.5	แผนภาพกรอบจำลองระบบกำเนิดเสียงเริ่มต้น	8
รูปที่ 2.6	แสดงโครงสร้างเวฟไฟล์ที่มีรูปแบบของไฟล์ RIFF	9
รูปที่ 2.7	แสดงค่าสูงสุด, ค่าต่ำสุด, ค่ากลาง ของรูปแบบการบันทึกแต่ละอัน	9
รูปที่ 2.8	แสดงการแบ่งช่วงสัญญาณ	10
รูปที่ 2.9	แสดงหน่วยสี่เหลี่ยมของการคำนวณตามขั้นตอนวิธีลดทอนทางเวลา	13
รูปที่ 2.10	แสดงวิธีการของ FFT แบบลดทอนทางเวลา (DIT) สำหรับข้อมูลขนาดจุด	13
รูปที่ 2.11	แสดงโครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ	15
รูปที่ 2.12	แสดงไดอะแกรมของนิวรอลที่สร้างขึ้น (Artificial Neuron)	16
รูปที่ 2.13	แสดงโมดูลนิวรอลที่สร้างขึ้นร่วมกับ Activation Function	17
รูปที่ 2.14	แสดงกราฟที่ได้จากสมการซิมมอยด์ลอจิสติกฟังก์ชัน (Sigmoidal logistic function)	17
รูปที่ 2.15	แสดง Hyperbolic Tangent Function	18
รูปที่ 2.16	แสดงลักษณะโครงข่ายประสาทเทียมแบบชั้นเดียว (Single-Layer Neural Networks)	19
รูปที่ 2.17	แสดงไดอะแกรมของ Backpropagation Neural Networks แบบสองชั้น	20
รูปที่ 2.18	แสดงไดอะแกรมของ Two Layer Feed-Forward	23
รูปที่ 2.19	แสดงไดอะแกรมเชื่อมโยงระหว่างชั้นของนิวรอลแบบ 2 ชั้นของ Backpropagation Network	24
รูปที่ 2.20	แสดงไดอะแกรมของการปรับค่า Weight ในชั้น Output layer และ Hidden layer	26
รูปที่ 2.21	ไดอะแกรมแสดงวิธีการหาค่าน้ำหนักที่เหมาะสมของการเทรนนิ่ง โดยใช้อัลกอริทึมแบบแบคพรอบพาเกชัน	27
รูปที่ 2.22	แสดง Flow Chart แสดงกระบวนการเทรนนิ่ง เพื่อหาค่าน้ำหนักที่เหมาะสมในกระบวนการทดสอบ	28
รูปที่ 3.1	แสดงวงจรควบคุมมอเตอร์	33
รูปที่ 4.1	แสดงหน้าจอของโปรแกรม RecFFT.EXE	34
รูปที่ 4.2	แสดงหน้าจอของโปรแกรม UnitScal.EXE	35
รูปที่ 4.3	แสดงหน้าจอของโปรแกรม Neural.EXE	35
รูปที่ 4.4	แสดงหน้าจอของโปรแกรม SndTest.EXE	36
รูปที่ 4.5	เสียง Wave	37
รูปที่ 4.6	เสียง Two	37
รูปที่ 4.7	เสียง Half	38
รูปที่ 4.8	เสียง ซ้าย	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ 4.9 เสียง ขวา	39
รูปที่ 4.10 เสียง Up	39
รูปที่ 4.11 เสียง Down	40
รูปที่ 4.12 เสียง Go	40
รูปที่ 4.13 เสียง Pause	41
รูปที่ 4.14 เสียง Clear	41
รูปที่ 4.15 แสดงสัญญาณเสียงที่อัดเข้ามา	42
รูปที่ 4.16 แสดงสัญญาณเสียงที่พลิกด้านล่างขึ้นมาด้านบน	42
รูปที่ 4.17 แสดงค่าเฉลี่ยของสัญญาณเสียง	43
รูปที่ 4.18 แสดงตำแหน่งของจุดตัด	43
รูปที่ 4.19 แสดงสัญญาณเสียงที่ผ่านการตัดแล้ว	44
รูปที่ 4.20 แสดงสเปกโตรแกรมที่ได้จากการแปลง FFT	44
รูปที่ 4.21 แสดงสเปกโตรแกรมที่ลดข้อมูลในแกนความถี่แล้ว	45
รูปที่ 4.22 แสดงสเปกโตรแกรมที่ลดข้อมูลในแกนเวลาแล้ว	45
รูปที่ 4.23 แสดงสเปกโตรแกรมของเสียงทั้งหมดที่ใช้ในการทดลอง Plot โดยใช้โปรแกรม MatLab	46
รูปที่ 4.24 แสดงสเปกโตรแกรมของเสียง Wave ที่ได้จากการบันทึก	46
รูปที่ 4.25 แสดงสเปกโตรแกรมของเสียง Two ที่ได้จากการบันทึก	47
รูปที่ 4.26 แสดงสเปกโตรแกรมของเสียง Half ที่ได้จากการบันทึก	47
รูปที่ 4.27 แสดงสเปกโตรแกรมของเสียง ซ้าย ที่ได้จากการบันทึก	48
รูปที่ 4.28 แสดงสเปกโตรแกรมของเสียง ขวา ที่ได้จากการบันทึก	48
รูปที่ 4.29 แสดงสเปกโตรแกรมของเสียง Up ที่ได้จากการบันทึก	49
รูปที่ 4.30 แสดงสเปกโตรแกรมของเสียง Down ที่ได้จากการบันทึก	49
รูปที่ 4.31 แสดงสเปกโตรแกรมของเสียง Go ที่ได้จากการบันทึก	50
รูปที่ 4.32 แสดงสเปกโตรแกรมของเสียง Pause ที่ได้จากการบันทึก	50
รูปที่ 4.33 แสดงสเปกโตรแกรมของเสียง Clear ที่ได้จากการบันทึก	51
รูปที่ 4.34 แสดงสเปกโตรแกรมของเสียง Wave ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	51
รูปที่ 4.35 แสดงสเปกโตรแกรมของเสียง Two ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	52
รูปที่ 4.36 แสดงสเปกโตรแกรมของเสียง Half ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	52
รูปที่ 4.37 แสดงสเปกโตรแกรมของเสียง ซ้าย ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	53
รูปที่ 4.38 แสดงสเปกโตรแกรมของเสียง ขวา ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	53
รูปที่ 4.39 แสดงสเปกโตรแกรมของเสียง Up ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	54
รูปที่ 4.40 แสดงสเปกโตรแกรมของเสียง Down ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	54
รูปที่ 4.41 แสดงสเปกโตรแกรมของเสียง Go ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	55
รูปที่ 4.42 แสดงสเปกโตรแกรมของเสียง Pause ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	55
รูปที่ 4.43 แสดงสเปกโตรแกรมของเสียง Clear ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว	56
รูปที่ 4.44 แสดงค่า Sum Square Error ที่ได้จากการเทรน 3000 รอบ	56

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการขับสเตรปมอเตอร์แบบ 2 เฟส	30
ตารางที่ 2.2 แสดงการขับสเตรปมอเตอร์แบบเวฟ	30
ตารางที่ 2.3 แสดงการขับสเตรปมอเตอร์แบบครึ่งสเต็ป	30
ตารางที่ 4.1 แสดงผลการทดสอบนิวัตรเน็ตเวอร์คด้วยข้อมูลเดิมที่ใช้ฝึกสอน ค่าในตารางเป็นค่าเฉลี่ยของเสียงทดสอบเสียงละ 50 ครั้ง	57
ตารางที่ 4.2 แสดงผลการทดสอบนิวัตรเน็ตเวอร์คด้วยเสียงใหม่ของคนเดิมที่ไม่ได้ผ่านการฝึกสอน ค่าในตารางเป็นค่าเฉลี่ยของเสียงทดสอบเสียงละ 50 ครั้ง	58
ตารางที่ 4.3 แสดงผลการทดสอบนิวัตรเน็ตเวอร์คด้วยเสียงใหม่ของคนเดิมที่ไม่ได้ผ่านการฝึกสอน ค่าในตารางเป็นค่าเอาท์พุทของเสียงทดสอบเสียงละ 50 ครั้ง	58



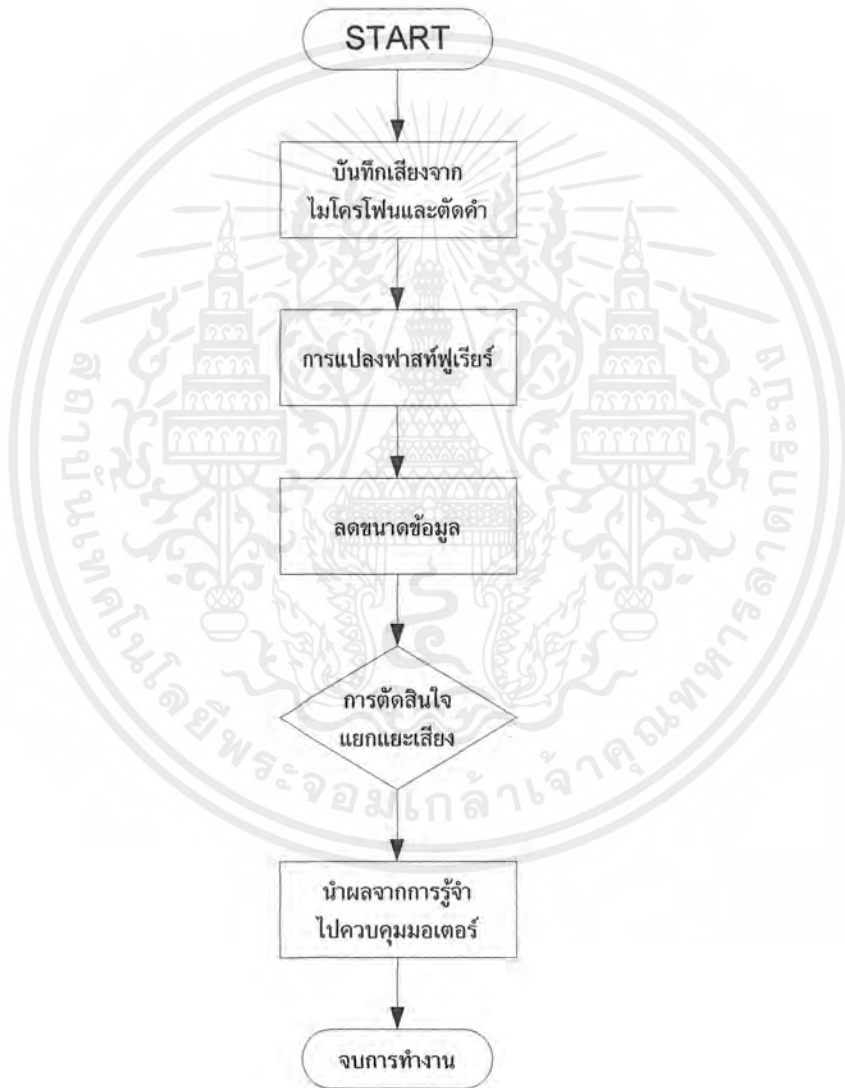
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

การสั่งงานอุปกรณ์ไฟฟ้าด้วยเสียง ก่อให้เกิดประโยชน์แก่นutzerอย่างมาก อันได้แก่ อำนวยความสะดวกสบายในการดำเนินชีวิต สร้างความปลอดภัยในกรณีที่มีความเสี่ยงอันเกิดจากการเข้าใกล้อุปกรณ์ไฟฟ้านั้นๆ การสั่งงานคอมพิวเตอร์ด้วยเสียงแทนการสั่งงานด้วยคีย์บอร์ด อีกทั้งยังสามารถใช้งานทางด้านการรักษาความปลอดภัยได้ด้วย

ปริญญานิพนธ์นี้มีจุดมุ่งหมายที่จะทำการแยกแยะสัญญาณเสียงคำโดดจำนวน 10 คำว่า เป็นเสียงของคำใด แล้วนำมาควบคุมการทำงานของมอเตอร์ โดยมีโครงสร้างการทำงานดังรูปที่ 1.1



รูปที่ 1.1 แสดงบล็อกการทำงานของ การแยกแยะสัญญาณเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนของการบันทึกเสียง

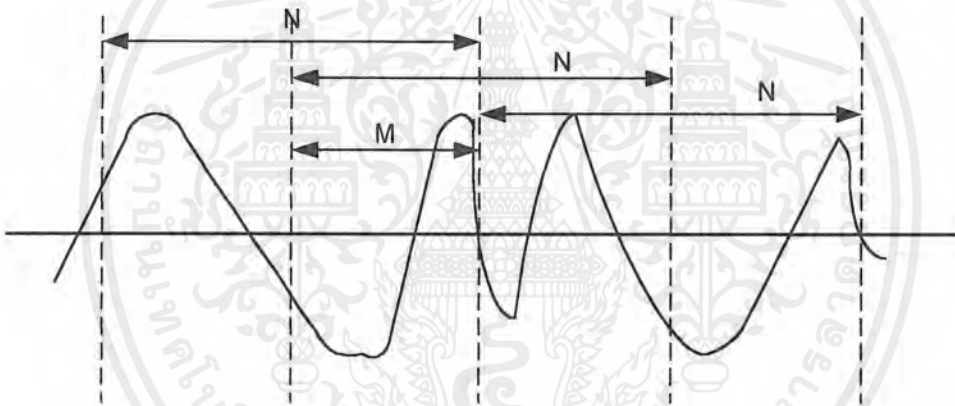
เสียงพูดซึ่งเป็นสัญญาณอนาล็อกจะถูกบันทึกจากไมโครโฟนผ่านทางซาวด์การ์ด (Sound Card) เป็นข้อมูลดิจิทัลของเวฟไฟล์ (Wave File) ด้วยความถี่สุ่ม 8kHz ขนาด 8บิต เป็นเวลา 1วินาที จะได้ข้อมูลจำนวน 8000ตัว แล้วนำมาผ่านการตัดค่าที่ระดับ 3% ของขนาดสูงสุด และตัดข้อมูลให้เหลือจำนวน 4096 (2^{12}) ตัว เพื่อให้เหมาะสำหรับการนำข้อมูลไปแปลงฟาสต์ฟูริเยร์ (Fast Fourier Transform หรือเรียกย่อๆ ว่า FFT)

2. ส่วนการแปลงฟาสต์ฟูริเยร์ทรานสฟอร์ม

- การแบ่งช่วงสัญญาณ (Frame Blocking) ข้อมูลเสียงที่อยู่ในโดเมนเวลาจำนวน 4096 จุดจะถูกตัดมาวิเคราะห์ทีละเฟรมๆ ละ N จุด โดยมีส่วนที่เหลื่อมกัน M จุด จนหมดสัญญาณเสียงที่นำมาวิเคราะห์ การเลือกใช้ค่า N และ M พิจารณาได้ดังนี้

ก) ค่า N เล็กไป จะทำให้ได้สเปกตรัมที่ไม่ละเอียด (Low Resolution) แต่ถ้าค่า N ใหญ่ไปก็จะลดความสามารถในการติดตามการเปลี่ยนแปลงของสเปกตรัมลง จึงไม่เหมาะกับสัญญาณที่มีสเปกตรัมเปลี่ยนแปลงรวดเร็ว

ข) M เล็กไป จะลดความสามารถในการติดตามการเปลี่ยนแปลงของสเปกตรัมลง แต่ถ้า M ใหญ่ไปก็จะสิ้นเปลืองการประมวลผลมาก[7]



รูปที่ 1.2 แสดงการแบ่งช่วงสัญญาณ

ปัญญานิพนธ์นี้เลือกใช้ค่า $N = 128$ จุด , $M = 64$ จุด จะทำให้ได้ผลการแปลง FFT จำนวน 63 เฟรม

- การวินโดว์ (Windowing) เนื่องจากเราใช้วิธีตัดสัญญาณเพื่อมาหาสเปกตรัมเป็นบล็อกๆ ซึ่งเท่ากับเป็นการหาสเปกตรัมของ “บล็อกของสัญญาณ” ไม่ใช่ตัวสัญญาณจริงๆ ที่เข้ามาติดต่อกันไม่มีจุดเริ่มต้นหรือสิ้นสุด การตัดสัญญาณเป็นบล็อกนี้จะทำให้เกิดความคลาดเคลื่อนในสเปกตรัมที่ได้ วิธีลดผลของความคลาดเคลื่อนนี้ทำได้โดยคูณสัญญาณแต่ละบล็อก หรือ $x_i(n)$ ด้วยฟังก์ชันหน้าต่าง (window function) ก่อนที่จะทำการหาสเปกตรัม

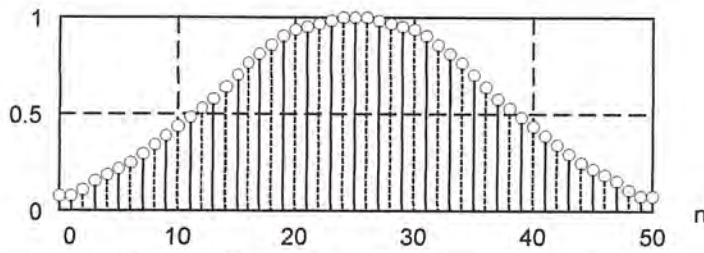
ฟังก์ชันหน้าต่างมีหลายแบบ เป็นฟังก์ชันที่มีลักษณะสมมาตร และมีค่าสูงสุดที่จุดกึ่งกลาง มีความยาวเท่าไรก็ได้ตามต้องการ คือแล้วแต่จะแทนค่าลงในสูตร ตัวอย่างของฟังก์ชันหน้าต่างที่ใช้บ่อยก็คือ หน้าต่างแฮมมิง (Hamming window) ซึ่งมีสมการคือ

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad n = 0, 1, 2, \dots, N-1 \quad (1.1)$$

$$x_{inew}(n) = x_i(n)w(n) \quad (1.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราต้องการ $w(n)$ ที่มีความยาวเท่ากับ 1 บล็อกสัญญาณ ดังนั้น ต้องใช้ N เท่ากับความยาว 1 บล็อกสัญญาณ รูปร่างของหน้าต่างแบบแฮมมิงแสดงอยู่ในรูปที่ 1.3 ซึ่งจะสังเกตเห็นได้ว่า ฟังก์ชันหน้าต่างมีค่าเล็กที่ส่วนต้นและส่วน



ปลาย ซึ่งเป็นส่วนที่ลดผลของการเปลี่ยนแปลงที่จุดเริ่มต้น และจุดสิ้นสุดของบล็อก[7]

รูปที่ 1.3 ตัวอย่างของฟังก์ชันหน้าต่างแบบแฮมมิง ที่ $N=50$

- การลดจำนวนข้อมูล เนื่องจากจำนวนข้อมูลที่ได้จากการแปลง FFT มีจำนวนมากเกินไป ($15 \times 65 = 975$ ตัว) ยังไม่เหมาะสมที่จะนำไปเทรนให้กับนิวรัลเน็ตเวิร์ค จึงต้องมีการลดจำนวนข้อมูล โดยจะใช้วิธีเลือกข้อมูลจากแต่ละเฟรม (1 เฟรมมีข้อมูล 65 ตัว) มาเฟรมละ 17 ตัวเท่าๆ กัน หลักการที่ใช้ในการเลือกข้อมูลนี้คือเลือกข้อมูลในตำแหน่งที่มีค่าเพิ่มขึ้นทีละ 2 เท่าของความถี่ในช่วงก่อน ซึ่งอาจเรียกว่าเป็นความถี่ฮาร์โมนิกของความถี่ในช่วงก่อน (อันว่า ศรี ประโมง 2535:200-203) ทำให้เหลือข้อมูลจำนวน 255 ตัว ข้อมูลที่ได้นี้ ก่อนที่จะนำไปป้อนให้กับนิวรัลเน็ตเวิร์ค จะต้องมีการปรับค่าของข้อมูลให้อยู่ในช่วง 0 ถึง 1 เสียก่อน โดยใช้สมการ Max-Min Scale

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} \quad (1.3)$$

โดยที่ X_{old} = ข้อมูลที่นำมาปรับสเกล
 X_{new} = ข้อมูลที่ถูกปรับสเกลแล้ว
 X_{max} = ข้อมูลที่มีค่ามากที่สุดของสเปกตรัม
 X_{min} = ข้อมูลที่มีค่าน้อยที่สุดของสเปกตรัม

ข้อมูลที่ถูกปรับสเกลแล้ว จะถูกบันทึกอยู่ในรูปของ Text File เพื่อเตรียมที่จะนำไปป้อนให้กับนิวรัลเน็ตเวิร์คต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

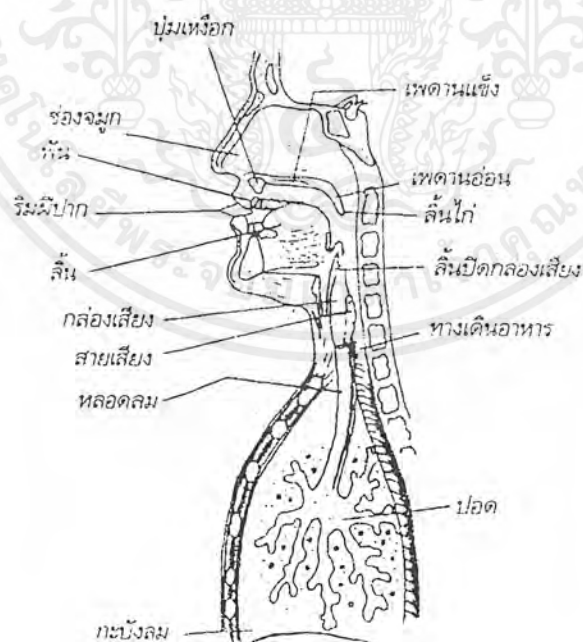
ทฤษฎีและหลักการ

ในปัจจุบัน ความก้าวหน้าทางด้านการประมวลผลสัญญาณต่าง ๆ ได้รู้ดหน้าไปอย่างรวดเร็ว ซึ่งในบางโอกาส สัญญาณอาจถูกประมวลผลในระบบอนาลอก ในบางโอกาสก็ถูกประมวลผลในระบบดิจิทัลด้วยคอมพิวเตอร์ ซึ่งในระบบของการประมวลผลด้วยเครื่องคอมพิวเตอร์ สัญญาณที่ถูกประมวลผลจะถูกแปลงจากสัญญาณอนาลอกที่ต่อเนื่องไปเป็นสัญญาณดิจิทัล มีการสุ่มเอาตัวอย่างสัญญาณเท่านั้นไปประมวล ซึ่งอยู่ในส่วนการทำงานของการ์ดเสียง จากนั้นจะเป็นขั้นตอนการประมวลผลซึ่งมีมากมายหลายวิธี ล้วนแต่มีวัตถุประสงค์ในการประมวลผลสัญญาณเพื่อแยกแยะสัญญาณต่าง ๆ ไปทำประโยชน์ตามต้องการ

2.1 กายวิภาคของระบบการเปล่งเสียงของมนุษย์

จากการศึกษาด้านกายวิภาคศาสตร์ของมนุษย์ (human anatomy) วิชาที่ว่าด้วยเสียงของภาษา (phonetics) และศาสตร์ทางด้านเสียง (acoustics) ช่วยให้เข้าใจขั้นตอนการทำงาน ร่วมกันของอวัยวะต่าง ๆ ในการเปล่งเสียงพูด ตลอดจนลักษณะทางกายภาพของเสียงพูด เพื่อนำมาวิเคราะห์ และสร้างแบบจำลองเลียนแบบเสียงพูดของมนุษย์

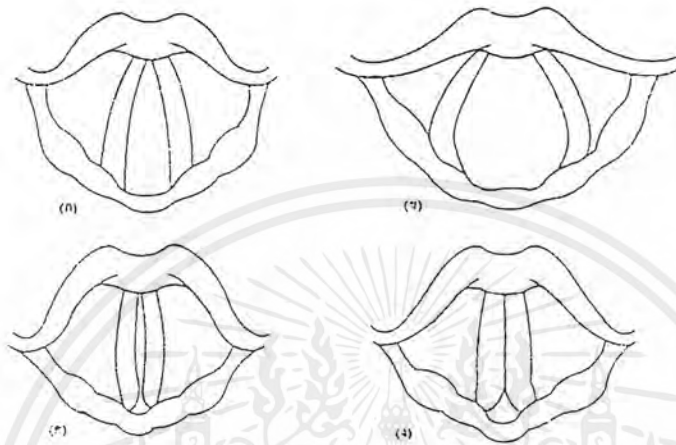
การทำให้เกิดเสียงเป็นหน้าที่หนึ่งของระบบหายใจ การออกเสียงหรือการพูดของมนุษย์แต่ละครั้ง จะต้องมีการทำงานร่วมกันของอวัยวะต่าง ๆ ของร่างกาย ดังรูปที่ 2.1 อันประกอบด้วย



รูปที่ 2.1 ภาพตัดขวางแสดงอวัยวะในระบบการพูดของมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ปอดและกระบังลม ทำหน้าที่สำคัญในการหายใจ และเป็นต้นกำเนิดการไหลของอากาศในกระบวนการผลิตเสียง
2. หลอดลม ทำหน้าที่นำอากาศจากปอดผ่านกล่องเสียง และเป็นอวัยวะที่อยู่ด้านหน้าของหลอดอาหาร
3. กล่องเสียง เป็นอวัยวะพิเศษที่ทำหน้าที่เป็นทางเดินอากาศเวลาหายใจ และเป็นตัวผลิตพัลส์ (pulse) ของอากาศขณะเปล่งเสียง ซึ่งประกอบด้วยเส้นเสียง (vocal cords) และช่องสายเสียง (glottis) รูปร่างของกล่องเสียง และเส้นเสียงในลักษณะต่าง ๆ แสดงในรูปที่ 2.2



รูปที่ 2.2 รูปร่างของกล่องเสียงขณะ (ก) หายใจปกติ (ข) หายใจเข้าลึก ๆ
(ค) กำลังส่งเสียง (ง) ส่งเสียงกระซิบหรือเสียงแผ่ว

4. ช่องปากและส่วนของหลอดอาหารตอนต้น อวัยวะกลุ่มนี้อยู่ต่อจากกล่องเสียง อาจเรียกว่าอวัยวะกำทอนเสียง (vocal tract) ทำหน้าที่กำทอนเสียง โดยให้กำทอนทั้งเสียงที่เกิดจากกล่องเสียง และเสียงที่เกิดภายในช่องปาก ขนาดของอวัยวะกำทอนเสียงขึ้นอยู่กับตำแหน่งของลิ้น ริมฝีปาก ขากรรไกร และเพดานอ่อน และเปลี่ยนแปลงไปตามการออกเสียง
5. โพร่งจมูก เริ่มจากเพดานอ่อนจนถึงรูจมูกทั้งสอง ทำหน้าที่กำทอนเสียงร่วมกับช่องปาก เมื่อมีการเปล่งเสียงที่ออกจากจมูก (nasal sounds) เช่นเสียง /ม/, /น/ และ /ง/ เป็นต้น

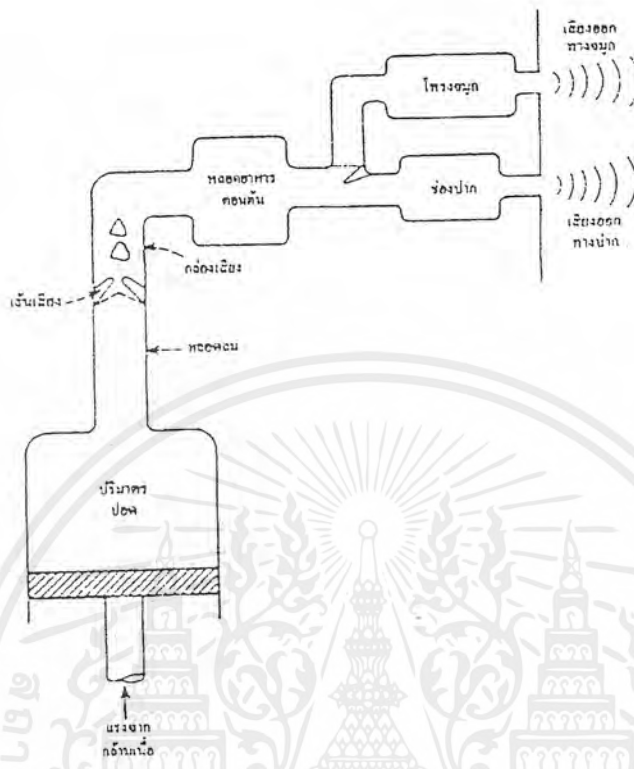
2.2 กระบวนการผลิตเสียงพูด

จากระบบเสียงพูด สามารถแสดงเป็นแผนภาพของระบบกำเนิดเสียง ดังรูปที่ 2.3 ซึ่งสามารถจำแนกกลไกการสร้างเสียงพูดของมนุษย์ได้ 3 แบบ ดังนี้

1. อากาศที่ไหลจากปอดจะถูกมอดูเลต (modulate) โดยการสั่นของเส้นเสียงทำให้เกิดคลื่นเสียงลักษณะคล้ายพัลส์ที่มีคาบเวลาแบบควอไซ (quasi-periodic pulse-like excitation)
2. อากาศที่ไหลจากปอดถูกทำให้ปั่นป่วน ด้วยการบังคับให้ไหลผ่านช่องแคบอันเกิดจากการบีบตัวของอวัยวะในช่องปากทำให้เกิดเสียงลักษณะคล้ายเสียงรบกวน (noise-like excitation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อากาศที่ไหลถูกกัก และเกิดแรงดันอยู่ภายในส่วนของช่องปากที่ปิด จากนั้นจึงปล่อยให้อากาศที่มีแรงดันพุ่งออกไปอย่างรวดเร็ว ทำให้เกิดการกระตุ้นเป็นเสียงในช่วงเริ่มต้น (transient excitation)



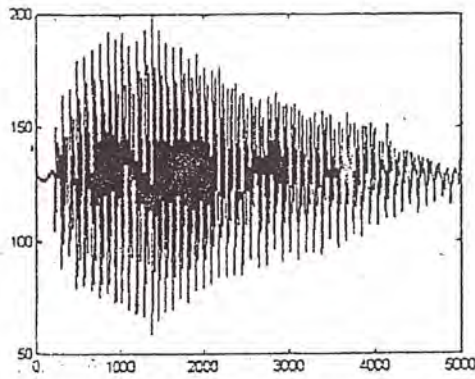
รูปที่ 2.3 แผนภาพระบบเสียงพูดของมนุษย์

2.3 เสียงพูดของมนุษย์

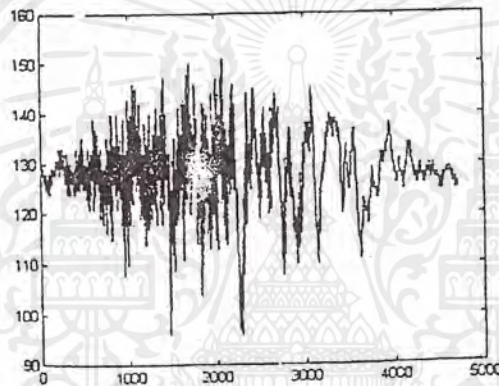
เสียงพูด เป็นคลื่นตามยาว (longitudinal wave) เกิดจากการสั่นของอนุภาคตัวกลางนั้นคือ อากาศ และทิศทางการสั่นของอนุภาค จะอยู่ในทิศเดียวกันกับทิศทางของการเคลื่อนที่ คลื่นเสียงเป็นคลื่นที่เปลี่ยนแปลงไปตามเวลา เสียงพูดแบ่งออกได้เป็น 2 ชนิด ตามการกำเนิดเสียง หรือโหมด (mode) การกระตุ้น คือ

1. เสียงก้องหรือเสียงโฆษะ (voiced) เกิดจากการบังคับอากาศให้ผ่านช่องสายเสียงทำให้มีการเปลี่ยนแปลงความตึงหย่อนของเส้นเสียง โดยเส้นเสียงจะสั่น และเกิดเป็นพัลส์ (pulse) ของอากาศไปกระตุ้นอวัยวะกำหนดเกิดเป็นเสียงก้อง ตัวอย่างเสียงก้องได้แก่ เสียงสระ เสียงพยัญชนะ ที่ต้องออกเสียงจากลำคอ (voiced consonants) เช่น เสียง /a/ ดังรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 (ก) ตัวอย่างรูปคลื่นของสัญญาณเสียงก้อง /a/



รูปที่ 2.4 (ข) ตัวอย่างรูปคลื่นของสัญญาณเสียงไม่ก้อง /sh/

2. เสียงไม่ก้องหรืออโหชะ (unvoiced หรือ voiceless) เป็นเสียงที่เกิดในช่องปาก หรือโพรงจมูก โดยอวัยวะภายในช่องปาก ริมฝีปาก ขวางการไหลของอากาศให้ผ่านได้เป็นช่องเล็ก ๆ อากาศจึงไหลผ่านอย่างรวดเร็ว และปั่นป่วนจนกระทั่งสร้างเป็นเสียงรบกวนช่วงความถี่กว้าง (broad-spectrum noise) ตัวอย่างเสียงไม่ก้องได้แก่ เสียงพยัญชนะที่ไม่ได้เกิดจากลำคอ (voiceless consonants) เช่น เสียง /sh/ ดังรูปที่ 2.4 (ข)

2.5 แบบจำลองระบบกำเนิดเสียงพูด

จากภาพรูปที่ 2.5 สามารถแสดงแผนภาพกรอบจำลองระบบกำเนิดเสียงเบื้องต้น ดังรูปที่ 2.5 จากรูปมีการแยกภาคแหล่งกำเนิดสัญญาณกระตุ้น ออกจากส่วนกำหนดเสียง ซึ่งแทนด้วยระบบเชิงเส้นแปรผันตามเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แผนภาพกรอบจำลองระบบกำเนิดเสียงเริ่มต้น

จากรูปที่ 2.5 แหล่งกำเนิดสัญญาณกระตุ้นทำหน้าที่แทนการทำงานของปอด และกล่องเสียงส่วนนี้จะผลิตขบวนพัลส์ที่มีคาบเวลาพิชชณะเปล่งเสียงวอยซ์ และให้กำเนิดเสียงซึ่งคล้ายเสียงรบกวนขณะเปล่งเสียงอโฆยะ (unvoiced) ส่วนที่สองเป็นท่อกำทอนเสียง จะแทนการทำงานของช่องปาก และโพรงจมูก ทำหน้าที่เสมือนตัวกรองสัญญาณ (filter) ที่ยอมให้ความถี่ฟอร์แมนท์ผ่านได้ ซึ่งสามารถแทนด้วยระบบเชิงเส้นแปรผันตามเวลา (time-varying linear system)[6]

2.6 รูปแบบของไฟล์เสียง

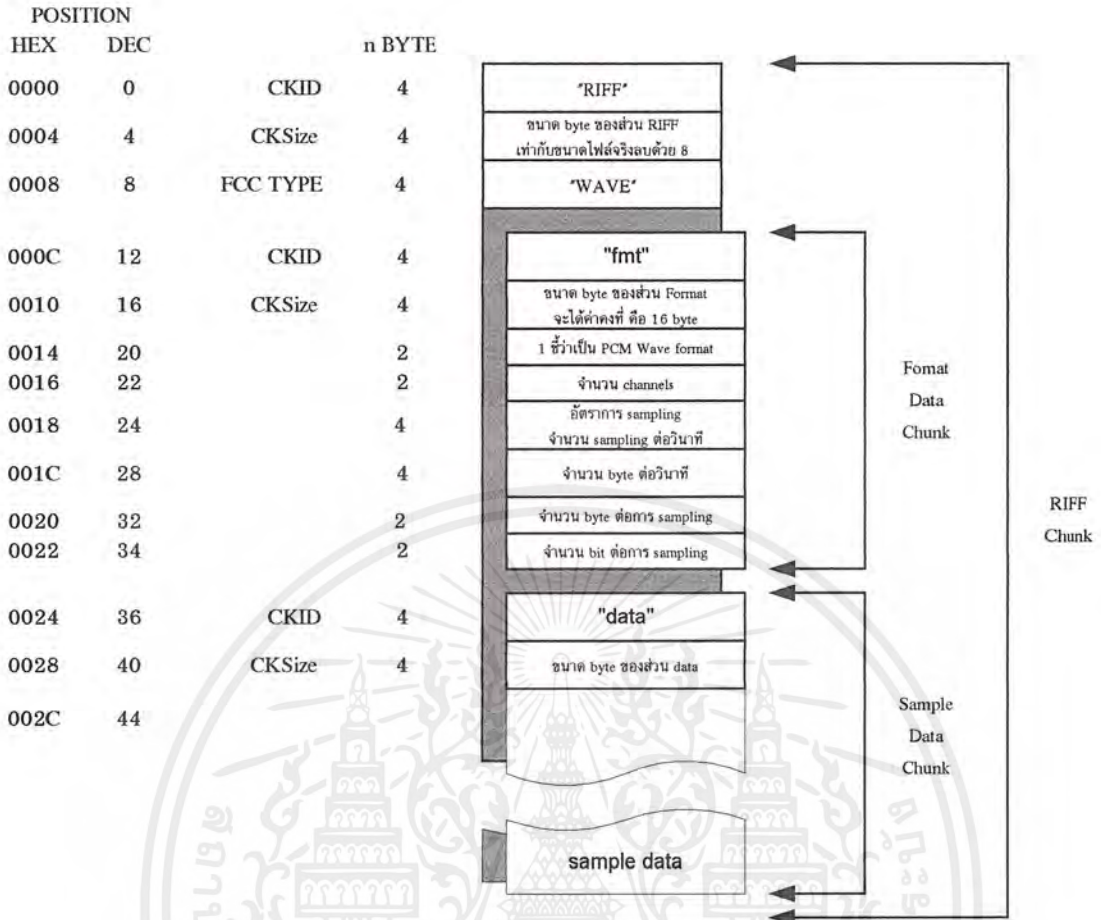
ในการบันทึกเสียงพูดเป็นข้อมูลดิจิทัลโดยใช้การ์ดเสียง จะทำให้ได้ไฟล์ที่มีรูปแบบของไฟล์ RIFF (Resource International File Format) ซึ่งเป็นรูปแบบมาตรฐานที่ใช้กันอย่างกว้างขวาง ไฟล์ RIFF นั้นมีโครงสร้างเชิงซ้อนคือ มีลักษณะเป็นกลุ่ม (Chunk) ซึ่งภายในกลุ่มนี้ยังประกอบด้วยกลุ่มย่อยซึ่งมีโครงสร้างคล้ายกัน ดังแสดงในรูปที่ 2.6

จากรูปที่ 2.6 จะเห็นว่า กลุ่มRIFF (RIFF Chunk) จะเป็นกลุ่มใหญ่ที่สุด ซึ่งประกอบด้วยกลุ่มรูปแบบของข้อมูล (Format Data Chunk) และกลุ่มข้อมูลจากการแซมปลิง (Sample Data Chunk) ซึ่งในแต่ละกลุ่มจะมีโครงสร้างที่ประกอบด้วย

1. ชื่อกลุ่ม (CKID: Chunk Identification) เช่น RIFF, fmt_data ซึ่งมีกฎกำหนดว่าต้องมี 4 ตัวอักษร (FCC : Four Character Code) ถ้ามี 3 ตัวอักษร ก็จะต้องเพิ่มช่องว่าง (Blank) เข้าไปอีก 1 ตัวอักษรให้รวมเป็น 4 ตัวอักษร ในกรณี fmt_

2. ตัวเลขบอกขนาดข้อมูลของกลุ่ม (CKSize : Chunk Size) มีหน่วยเป็นไบต์(Byte) โดยจะไม่รวมไบต์ที่ใช้ไปในการเก็บชื่อกลุ่ม(CKID) แต่จะเก็บตัวเลขบอกขนาดของกลุ่มเองด้วย ตัวเลขบอกขนาดของกลุ่มจะแสดงด้วยข้อมูล 4 ไบต์โดยไบต์ที่มีความสำคัญต่ำสุด(Least Significant Byte) จะถูกเก็บเข้าไปก่อน

3. ข้อมูลของกลุ่ม(CKData : Chunk Data) มีจำนวนไบต์เท่ากับตัวเลขที่แสดงในข้อที่ 2 เช่น ในกลุ่มรูปแบบของข้อมูล(Format Data Chunk) จะมีข้อมูลของกลุ่มเป็นรูปแบบในการบันทึกเสียง ยกตัวอย่างเช่น อัตราการแซมปลิง, จำนวนบิตต่อ 1 แซมเปิล เป็นต้น หรือในกลุ่มข้อมูลจากการแซมปลิง (Sample Data Chunk) จะมีข้อมูลของกลุ่มเป็นแอมพลิจูด(Amplitude) ที่ได้จากการแซมปลิง



รูปที่ 2.6 แสดงโครงสร้างเวฟไฟล์ที่มีรูปแบบของไฟล์ RIFF

รูปแบบการบันทึก	ค่าสูงสุด	ค่าต่ำสุด	ค่าที่ตำแหน่งตรงกลาง
8 บิต พีซีเอ็ม	255 (FFH)	0	128(80H)
16 บิต พีซีเอ็ม	32767(7FFFH)	-32768(-8000H)	0

รูปที่ 2.7 แสดงค่าสูงสุด, ค่าต่ำสุด, ค่ากลาง ของรูปแบบการบันทึกแต่ละอัน

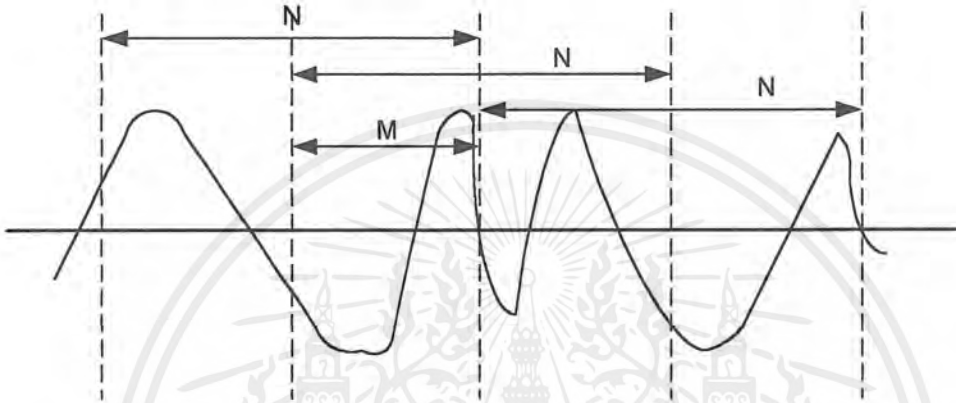
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การแบ่งช่วงสัญญาณ (Frame Blocking)

ข้อมูลเสียงซึ่งอยู่ในโดเมนเวลา จะถูกตัดมาวิเคราะห์ทีละเฟรม เฟรมละ N จุด โดยมีส่วนที่เหลื่อมกัน M จุด จนหมดสัญญาณเสียงที่นำมาวิเคราะห์ การเลือกใช้ค่า N และ M พิจารณาได้ดังนี้

ก) ค่า N เล็กไป จะทำให้ได้สเปกตรัมที่ไม่ละเอียด (Low Resolution) แต่ถ้าค่า N ใหญ่ไปก็จะลดความสามารถในการติดตามการเปลี่ยนแปลงของสเปกตรัมลง จึงไม่เหมาะกับสัญญาณที่มีสเปกตรัมเปลี่ยนแปลงรวดเร็ว

ข) M เล็กไป จะลดความสามารถในการติดตามการเปลี่ยนแปลงของสเปกตรัมลง แต่ถ้า M ใหญ่ไปก็จะสิ้นเปลืองการประมวลผลมาก[7]



รูปที่ 2.8 แสดงการแบ่งช่วงสัญญาณ

ปริญญานิพนธ์นี้เลือกใช้ค่า $N = 128$ จุด , $M = 64$ จุด จะทำให้ได้ผลการแปลง FFT จำนวน 63 เฟรม

2.8 การวินโดว์ (Windowing)

เนื่องจากเราใช้วิธีตัดสัญญาณเพื่อมาหาสเปกตรัมเป็นบล็อกๆ ซึ่งเท่ากับเป็นการหาสเปกตรัมของ “บล็อกของสัญญาณ” ไม่ใช่ตัวสัญญาณจริงๆ ที่เข้ามาติดต่อกันไม่มีจุดเริ่มต้นหรือสิ้นสุด การตัดสัญญาณเป็นบล็อกนี้จะทำให้เกิดความคลาดเคลื่อนในสเปกตรัมที่ได้ วิธีลดผลของความคลาดเคลื่อนนี้ทำได้โดยคูณสัญญาณแต่ละบล็อก หรือ $x_i(n)$ ด้วยฟังก์ชันหน้าต่าง (window function) ก่อนที่จะทำการหาสเปกตรัม

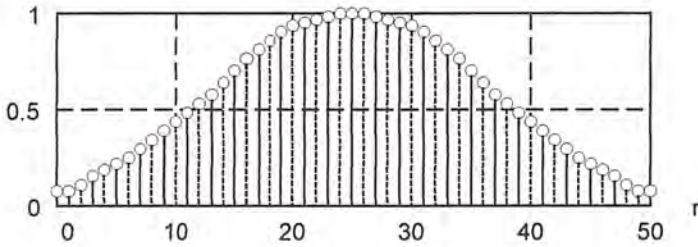
ฟังก์ชันหน้าต่างมีหลายแบบ เป็นฟังก์ชันที่มีลักษณะสมมาตร และมีค่าสูงสุดที่จุดกึ่งกลาง มีความยาวเท่าไรก็ได้ตามต้องการ คือแล้วแต่จะแทนค่าลงในสูตร ตัวอย่างของฟังก์ชันหน้าต่างที่ใช้บ่อยก็คือ หน้าต่างแฮมมิง (Hamming window) ซึ่งมีสมการคือ

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad , \quad n = 0, 1, 2, \dots, N-1 \quad (2.1)$$

$$x_{new}(n) = x_i(n)w(n) \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราต้องการ $w(n)$ ที่มีความยาวเท่ากับ 1 บล็อกสัญญาณ ดังนั้น ต้องใช้ N เท่ากับความยาว 1 บล็อกสัญญาณ รูปร่างของหน้าต่างแบบแฮมมิงแสดงอยู่ในรูปที่ 2.9 ซึ่งจะสังเกตเห็นได้ว่า ฟังก์ชันหน้าต่างมีค่าเล็กที่ส่วนต้นและส่วนปลาย ซึ่งเป็นส่วนที่ลดผลของการเปลี่ยนแปลงที่จุดเริ่มต้น และจุดสิ้นสุดของบล็อก[7]



รูปที่ 2.9 ตัวอย่างของฟังก์ชันหน้าต่างแบบแฮมมิง ที่ $N=50$

2.9 การวิเคราะห์ฟูรีเยร์ (Fourier Analysis)

การวิเคราะห์ฟูรีเยร์ช่วงสั้น เป็นเทคนิคการหาความถี่ที่ใช้กันมานานแล้ว การวิเคราะห์ฟูรีเยร์ให้ตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชันความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งตลอดเวลา ดังนั้นจึงจำเป็นต้องทำการวิเคราะห์ในช่วงสั้น ๆ โดยใช้วินโดว์

การแปลงฟูรีเยร์ช่วงสั้นมีสมการว่า

$$X_n(e^{-j\omega}) = \sum_{m=-\infty}^{\infty} x(m)[\exp(-j\omega m)w(n-m)] \quad (2.3)$$

ในการคำนวณต้องใช้ DFT แทนการแปลงฟูรีเยร์แบบต่อเนื่อง โดยใช้ฟังก์ชันวินโดว์ ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว (N คือ ช่วงเวลาหรือขนาดของวินโดว์ที่ใช้ในการแปลง DFT) ข่าวสารต่างๆ ใน $X_n(e^{-j\omega})$ จะไม่สูญหายไปจากข้อมูลเดิมถ้าการแปลงนั้นสุ่มมาด้วยความถี่สูงเพียงพอ (คือ ช่วงระยะห่างระหว่าง N) และวินโดว์ $w(n)$ ไม่มีจุดสุ่มที่เป็นศูนย์ตลอดช่วง N ตัวแปร N เป็นตัวแปรที่จะต้องระวังมากเป็นพิเศษในการวิเคราะห์ความถี่ช่วงสั้น ถ้าค่าของ N ต่ำจะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำในช่วงสั้น ๆ เท่านั้น ในทางตรงกันข้ามถ้า N มีขนาดใหญ่ จะให้ผลของความละเอียดเวลาที่แย่ในโดเมนเวลา แต่จะทำให้โดเมนความถี่มีความละเอียดสูงกว่า

2.9.1 การแปลงฟาสต์ฟูรีเยร์ (Fast Fourier Transform)

โดยทั่วไปการแปลงฟูรีเยร์ (Discrete Fourier Transform) เป็นการคำนวณที่ใช้เวลาค่อนข้างมาก ลำดับขั้นตอนในการคำนวณ DFT ให้เร็วขึ้น เรียกว่า ฟาสต์ฟูรีเยร์ (Fast Fourier Transform หรือ FFT) โดยการแปลง FFT แบ่งได้เป็น 2 ชนิดใหญ่ ๆ คือ ชนิดลดทอนทางเวลา (Decimation In Time หรือ DIT) และชนิดลดทอนทางความถี่ (Decimation In Frequency หรือ DIF) สำหรับในส่วนนี้จะแสดงเฉพาะชนิดลดทอนทางเวลาซึ่งเกี่ยวกับในโครงงานนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 ขั้นตอนวิธีลดทอนทางเวลา (Decimation In Time หรือ DIT)

วิธีนี้เป็นการจัดแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา $X(n)$ ขนาด N จุดออกเป็น 2 ลำดับ สัญญาณขนาด $N/2$ จุดเท่ากันคือ ลำดับคู่และลำดับคี่ โดยที่ลำดับคู่เกิดจากการเอาลำดับในตำแหน่งคู่มาเรียงกัน ที่เหลือเป็นลำดับคี่ ดังนั้นจะได้

$$\begin{aligned}x_c(m) &= x(2n) & ; m = 0, 1, 2, \dots, (N/2)-1 \\x_o(m) &= x(2n+1) & ; m = 0, 1, 2, \dots, (N/2)-1\end{aligned}\quad (2.4)$$

ถ้าให้ W_N เท่ากับ $\exp(-2j/N)$ จะทำให้การคำนวณ DFT ของลำดับ $x(n)$ ที่ยาว N จุด สามารถเขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N/2-1} x_e(2m)W_N^{2km} + \sum_{m=0}^{N/2-1} x_o(2m+1)W_N^{(2m+1)k} \quad (2.5)$$

โดยที่

$$W_N^2 = \{\exp[j2\pi/N]^2\} = \exp[j2\pi/N/2] = W_{N/2} \quad (2.6)$$

ซึ่ง $W_{N/2}$ เป็นค่า W ลำดับความยาว $N/2$ จุด สมการ 2.5 สามารถเขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N/2-1} x_e(m)W_{N/2}^{km} + \sum_{m=0}^{N/2-1} x_o(m)W_{N/2}^{km} \quad (2.7)$$

การนำผลการแปลง DFT ขนาด 2 จุด จำนวน $N/2$ ภาคมารวมกัน เพื่อให้เป็นการคำนวณ DFT ขนาด N จุด จะต้องมีหลักเกณฑ์ที่ถูกต้องด้วย จากสมการ 2.7 ถ้าเขียนให้อยู่ในช่วง $0 \leq k \leq N/2 - 1$ สามารถเขียนใหม่ได้เป็น

$$\begin{aligned}X(k) &= X_e(k) + (W_N^k)X_o(k) & 0 \leq k \leq N/2 - 1 \\ &= X_e(k - N/2) + (W_N^k)X_o(k - N/2) & 0 \leq k \leq N/2 - 1\end{aligned}\quad (2.8)$$

เทอม W_N^k เรียกว่า ตัวประกอบหมุน (Twiddle Factor) ซึ่งใช้ร่วมกับ DFT ขนาด 2 จุดหรือขนาด $N/2$ จุด ในการนำมาประกอบเป็น DFT ขนาด N จุดได้เหมือนเดิม และจากความสัมพันธ์

$$(W_N)^{k-N/2} = -(W_N)^k$$

จะได้

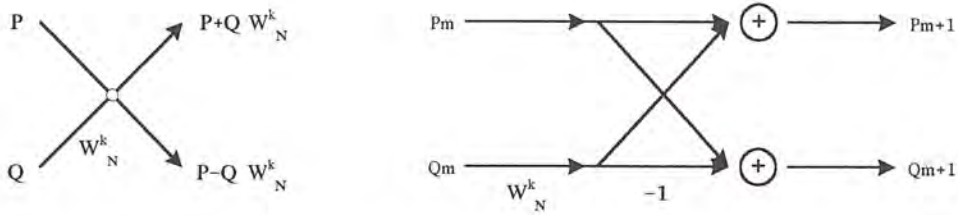
$$\begin{aligned}X(k) &= X_1(k) + (W_N)^k X_2(k) & ; 0 \leq k \leq N/2 - 1 \\ &= X_1(k - N/2) + (W_N)^{k-N/2} X_2(k - N/2) & ; N/2 \leq k \leq N/2 - 1\end{aligned}\quad (2.9)$$

ผลจากสมการนี้สามารถนำไปใช้สร้างหน่วยคำนวณที่เรียกว่า หน่วยผีเสื้อ (Butterfly Unit) โดยมีข้อมูลเข้าคือ A และ B และข้อมูลออก คือ X และ Y เป็น

$$\begin{aligned}X &= A + (W_N)^k B \\ Y &= A + (W_N)^k B\end{aligned}\quad (2.10)$$

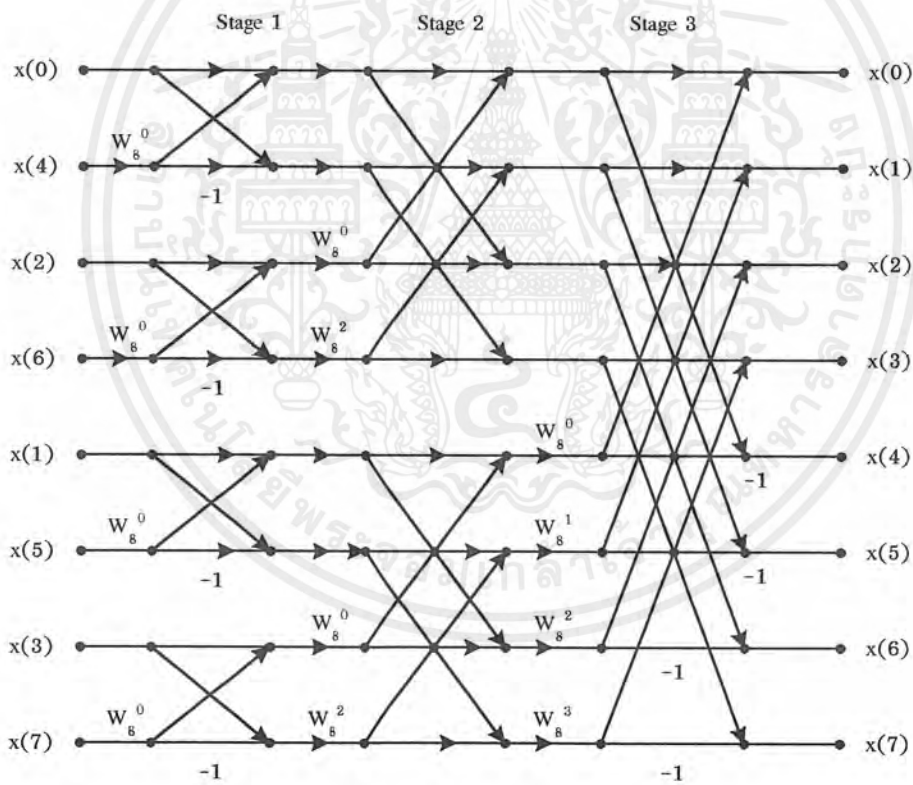
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสามารถเขียนอธิบายแทนด้วยโพลีชาร์ทดังแสดงในรูป 2.9



รูปที่ 2.9 แสดงหน่วยผีเสื้อของการคำนวณตามขั้นตอนวิธีลดทอนทางเวลา

สำหรับตัวอย่างการคำนวณ DFT โดยใช้ FFT แสดงดังรูป 2.10



รูปที่ 2.10 แสดงวิธีการของ FFT แบบลดทอนทางเวลา (DIT) สำหรับข้อมูลขนาดจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 นิวรัลเน็ตเวิร์ค (Neural Networks)

2.10.1 ความรู้เบื้องต้นเกี่ยวกับนิวรัลเน็ตเวิร์ค

ในช่วงระยะเวลา 8-9 ปีที่ผ่านมา ในต่างประเทศมีการตื่นตัวในการวิจัยและพัฒนาเกี่ยวกับโครงข่ายประสาทเทียม (Artificial Neural Networks : ANN) อย่างกว้างขวางทั้งทางด้านทฤษฎีและการประยุกต์ใช้งาน จนกระทั่งถึงปัจจุบันได้มีการประยุกต์นำนิวรัลเน็ตเวิร์คมาใช้ในอุปกรณ์เครื่องใช้ต่างๆ มากขึ้น เช่น เครื่องมือหาปลา (Sonar) ที่มีความฉลาดมากขึ้น เช่น สามารถบอกได้ว่า ฟองปลาที่กำลังตรวจจับอยู่นั้นเป็นปลาชนิดใด จำนวนเท่าไร เครื่องโทรศัพท์แบบที่สามารถเรียกเลขหมายปลายทางให้อัตโนมัติ (Voice Phone) เพียงยกหูแล้วพูดชื่อของผู้ที่จะติดต่อเท่านั้น เครื่องอ่านตัวอักษร (OCR) ที่สามารถเปลี่ยนภาพตัวอักษรให้เป็นรหัสตัวอักษรแบบแอสกี ระบบนักบินอัตโนมัติ (Auto Pilot Aircraft) ระบบการคาดเดาอนาคตจากข้อมูลในอดีต (Forecasting, Prediction) ฯลฯ ซึ่งเครื่องมือและอุปกรณ์ต่างๆ ที่นำเอา ANN มาใช้ช่วยวิเคราะห์นั้น จะมีความฉลาดมากขึ้นและมีระบบความคิดที่มีการทำงานในลักษณะคล้ายกับมนุษย์ [1]

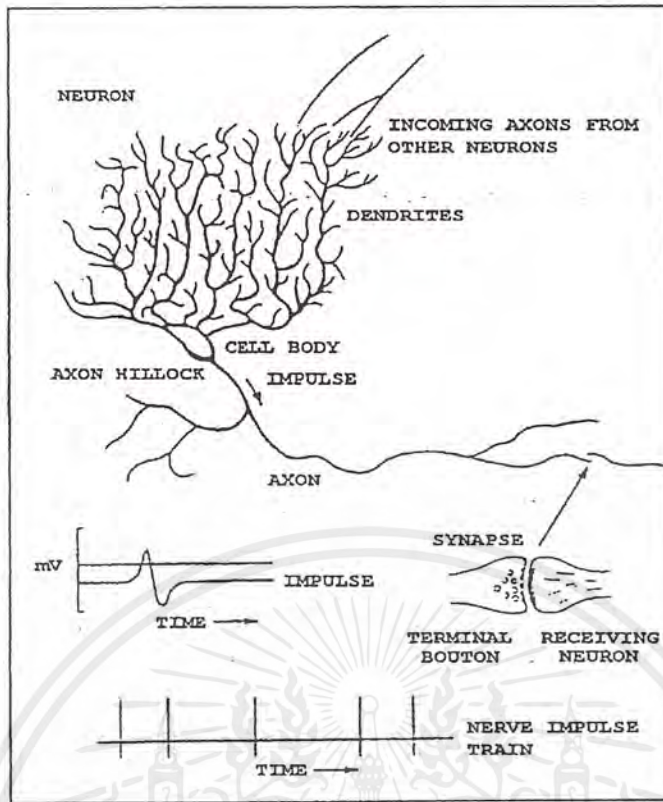
นิวรัลเน็ตเวิร์ค หมายถึง โครงข่ายใยประสาทที่เชื่อมต่อกันระหว่างเซลล์ประสาทจำนวนมากมายมหาศาล มีความสามารถประมวลผลสูงบรรจุอยู่ในสมอง สมองชีวภาพที่เป็นจุดศูนย์กลางการควบคุมกิจกรรมของการดำเนินชีวิต การวิจัยสร้างโครงข่ายประสาทเทียม (Artificial Neural Network) มีแนวคิดเลียนแบบการทำงานของสมองชีวภาพ โดยเรียนรู้และศึกษาการทำงานของสมองชีวภาพเพื่อกำหนดแนวทางสำหรับการสร้างแบบจำลองขึ้นมา แล้วพยายามสมมติฐานลักษณะการทำงานโดยจำลองเป็นโมเดลคณิตศาสตร์ที่มีลักษณะเดียวกันแล้วดำเนินการคำนวณโดยใช้คอมพิวเตอร์

2.10.2 นิวรัลเน็ตเวิร์คชีวภาพ

ระบบคิดคำนึงของมนุษย์ มีโครงสร้างพื้นฐานจากเซลล์สมองที่เรียกว่านิวรัล (Neural) เรียงเป็นชั้น ๆ อย่างซับซ้อนจำนวนมหาศาลประมาณหมื่นล้าน (10^{11}) นิวรัล และอาจมีจุดเชื่อมโยงส่งผ่านจุดเชื่อมโยงภายในถึงพันล้านล้าน (10^{15}) จุด แต่ละนิวรัลจะมีคุณลักษณะแตกต่างกันไปโดยมีการทำงานคล้ายกัน คือ รับเข้า, ประมวลผล, ส่งออก สัญญาณไฟฟ้าเคมีผ่านไปยังนิวรัล ซึ่งจะส่งสื่อสารไปตามระบบของสมอง

จากภาพที่ 2.11 ส่วนแขนงที่ขยายแยกออกไปจากตัวเซลล์ต่อไปยังเซลล์อื่นๆ เพื่อรับสัญญาณ เรียกว่า เดนไดรท์ (Dendrites) จุดรับสัญญาณจากเซลล์อื่น เข้ามายังตัวเซลล์ จะผ่านมาทางจุดเชื่อมต่อที่เรียกว่า ซินแนปส์ (Synapse) ซึ่งแอกซอน (Axon) จะเป็นตัวส่งสัญญาณเอาที่พุด ออกไปยังนิวรอลอื่น จากผลการวิจัยพบว่า แต่ละนิวรอลจะเชื่อมต่อออกไปยังนิวรอลอื่นๆ ซึ่งแต่ละนิวรอลจะมีคุณสมบัติในการเพิ่ม ขยาย หรือลดทอนความเข้มของสัญญาณ บางสัญญาณที่เข้ามาทางเดนไดรท์ ของเซลล์ (ซึ่งมีแขนงมากมาย) อาจสามารถกระตุ้นตัวเซลล์ แต่บางสัญญาณก็อาจจะยับยั้งตัวเซลล์ เนื่องจากเซลล์ประสาทหนึ่งเซลล์ มีเดนไดรท์มาก ฉะนั้น สัญญาณกระตุ้นจากเดนไดรท์ ที่รับเข้ามาจากเซลล์ประสาทอื่นๆ จะถูกนำมารวมกันที่ตัวเซลล์ประสาทที่เซลล์ประสาทจะมีค่าเทรชโฮลด์ (Threshold) ค่าหนึ่งหากผลรวมของสัญญาณไฟฟ้าเคมี (Electrochemical) มีค่ามากกว่า เทรชโฮลด์เซลล์ เซลล์ประสาทก็จะส่งสัญญาณขนาดหนึ่งผ่านทางแอกซอน ไปยังนิวรอลอื่นๆ การจัดเรียงชั้น (Layer) และลักษณะการเชื่อมโยงระหว่างนิวรอลในสมองนั้น มีการจัดเรียงที่ซับซ้อน สอดคล้องกับหน้าที่การทำงานเฉพาะส่วน มีการเจริญเติบโตสัมพันธ์กับสิ่งแวดล้อม และมีการเรียนรู้ตลอดเวลา ซึ่งใช้เวลานานนับปี ดังนั้นจึงยากที่จะสร้างโมเดลขึ้นมา เพื่อเลียนแบบให้มีคุณลักษณะคล้ายสมองชีวภาพได้ทั้งหมด ผลงานที่ได้จากการท้าววิจัยในปัจจุบันเป็นเพียงการจำลอง การเลียนแบบ และการทำงานเฉพาะบางส่วนของโครงข่ายประสาทมาใช้เฉพาะกับงานใดงานหนึ่ง ซึ่งมีการวิจัยลักษณะของโครงข่ายแบบต่างๆ ขึ้นมา โดยแต่ละแบบจะเหมาะกับงานประเภทหนึ่งๆ เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



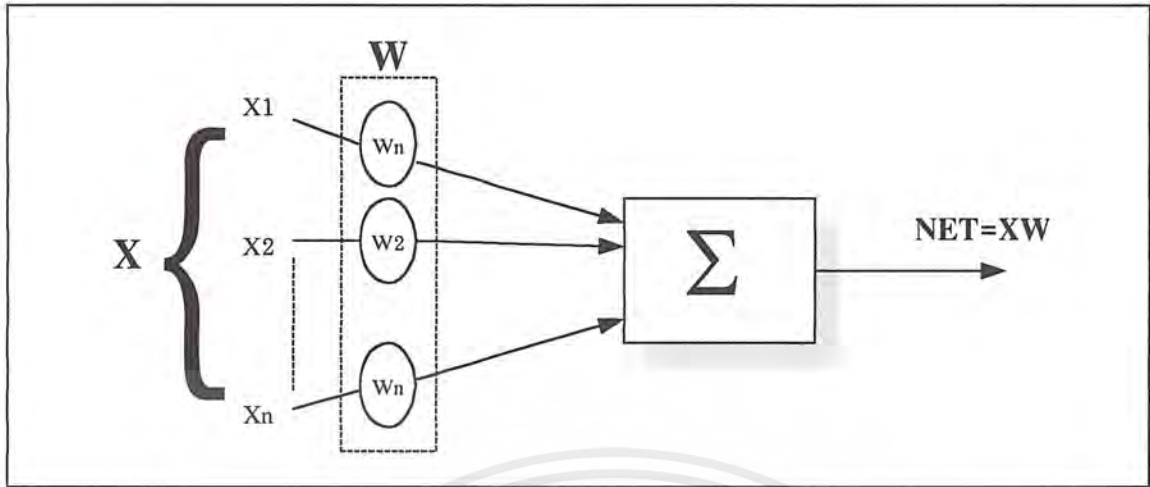
รูปที่ 2.11 แสดงโครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ

2.10.3 โครงข่ายประสาทเทียม (Artificial Neural Networks)

การออกแบบสร้างประสาทเทียมนั้น มีสมมติฐานขั้นแรกจากคุณสมบัติ ของระบบประสาทชีวภาพ ดังที่กล่าวมา กล่าวคือ ชุดรับสัญญาณข้อมูล อินพุตของเซลล์ประสาทหนึ่ง ได้จากสัญญาณเอาต์พุตของเซลล์ประสาทอื่นๆ ผ่านทาง ซินแนปส์และเดนไดรต์ ข้อมูลแต่ละค่าที่รับมาจะถูกลดขนาดด้วย ซินแนปติกส์ ซึ่งภายในประกอบด้วยสารเคมีประเภท K^+ , Ca^{++} , Na^+ , Cl^- ซึ่งจะมีลักษณะทางความนำพัลส์ (Pulse) สัญญาณไฟฟ้าเคมีที่แตกต่างกัน (James A. Freeman and David M.Skapura, 1991:8-9) ด้วยเหตุนี้ โมดูลประสาทเทียมที่สร้างขึ้น จะต้องมีการถ่วงน้ำหนักให้กับโมดูล ก่อนที่จะนำเข้าสู่โมดูลประสาทเทียม จุดนี้เรียกว่า น้ำหนักซินแนปติกส์ ปริมาณของข้อมูลที่เข้าสู่นิวรอน จะถูกนำมา รวมกัน และตัดสินใจด้วยระดับความสนใจของนิวรอน (Activation level) แล้วจะส่งเป็นเอาต์พุตออกที่แยกออกไปยัง นิวรอนอื่น ๆ

จากภาพที่ 2.12 แสดงถึงโมดูลที่สร้างขึ้นโดยแนวความคิดจากเซลล์สมองชีวภาพ สัญญาณอินพุต คือ X_1, X_2, \dots, X_n จะถูกป้อนเข้าไปยังนิวรอนที่สร้างขึ้น ซึ่งเปรียบเทียบกับสัญญาณที่ป้อนเข้ายังซินแนปส์ของนิวรอน ชีวภาพ สัญญาณอินพุตนี้จะนำไปคูณกับค่าน้ำหนักแนปติกส์ที่มีค่าตั้งแต่ 0.00-1 (Weight : ค่าที่ใช้ถ่วงน้ำหนัก) W_1, W_2, \dots, W_n ก่อนที่จะเข้าสู่บล็อกซัมเมชัน (Σ : Summation) ซึ่งค่าถ่วงน้ำหนักนี้ จะสอดคล้องกับค่าสเตรงท์ (Strength) ของจุดต่อซินแนปส์ชีวภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงไดอะแกรมของนิวรอนที่สร้างขึ้น (Artificial Neuron)

แต่ละจุด (Single biological synaptic connection) บล็อกซิมเมชันนี้ ก็จะทำหน้าที่สอดคล้องคล้ายกับตัวเซลล์สมองชีวภาพ ผลรวมทางคณิตศาสตร์ของอินพุต และน้ำหนักจะได้เป็นเอาต์พุต เราเรียกว่า เน็ต (NET) ซึ่งเราจะรวมกันในรูปของเวกเตอร์ได้ดังนี้

$$NET = X_1W_1 + X_2W_2 + \dots + X_nW_n \quad (2.11)$$

จะได้

$$NET = XW \quad (2.12)$$

2.10.3 ฟังก์ชันกระตุ้นความสนใจ (Activation Function)

เมื่อได้สัญญาณ NET แล้ว กระบวนการต่อมาที่นิวรอนต้องทำคือตัดสินใจ เราจึงต้องกำหนดฟังก์ชันการตัดสินใจ เพื่อใช้เป็นระดับของการตัดสินใจให้กับนิวรอน เพื่อให้ได้สัญญาณเอาต์พุตของนิวรอนออกมา ซึ่งเชื่อมต่อไปยังนิวรอนตัวอื่น ๆ เป็นโครงข่าย OUT ที่ได้อาจเป็น Simple linear function โดย

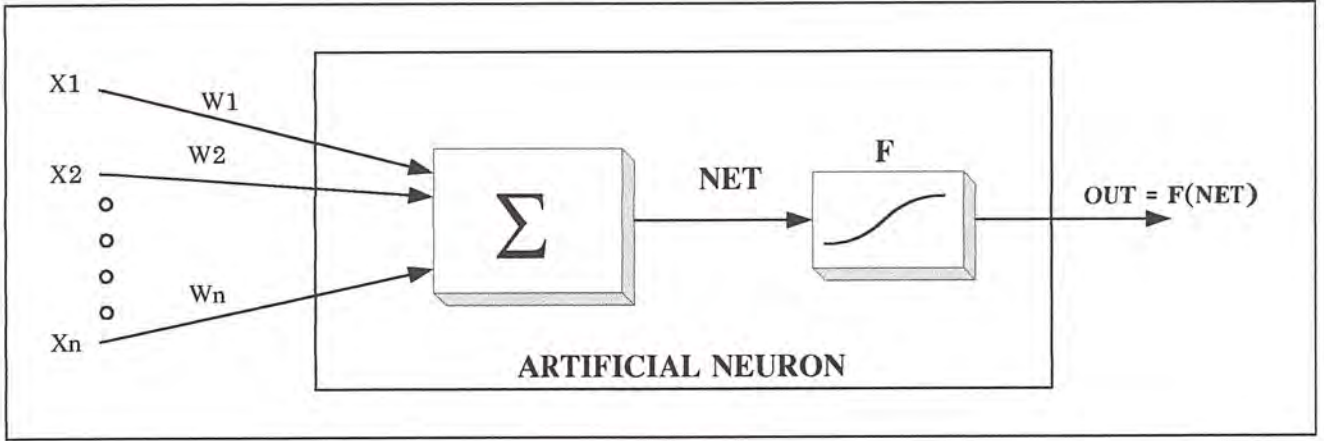
$$OUT = K[NET] \quad (2.13)$$

โดย K เป็นค่าคงที่ ที่เรียกว่า Threshold function
ตัวอย่างเช่น

$$\begin{aligned} OUT &= 1 \quad \text{ถ้า } NET > T \\ OUT &= 0 \quad \text{เมื่อเป็นกรณีอื่น} \end{aligned} \quad (2.14)$$

และ T เป็นค่าเทรชโฮลด์คงที่ หรืออาจเป็น Function อื่น ๆ ที่เลียนแบบคุณสมบัติไม่เป็นเชิงเส้นของเซลล์ประสาทชีวภาพได้อย่างใกล้เคียงกว่า และใช้เป็นฟังก์ชันให้กับโครงข่ายทั่วไปได้

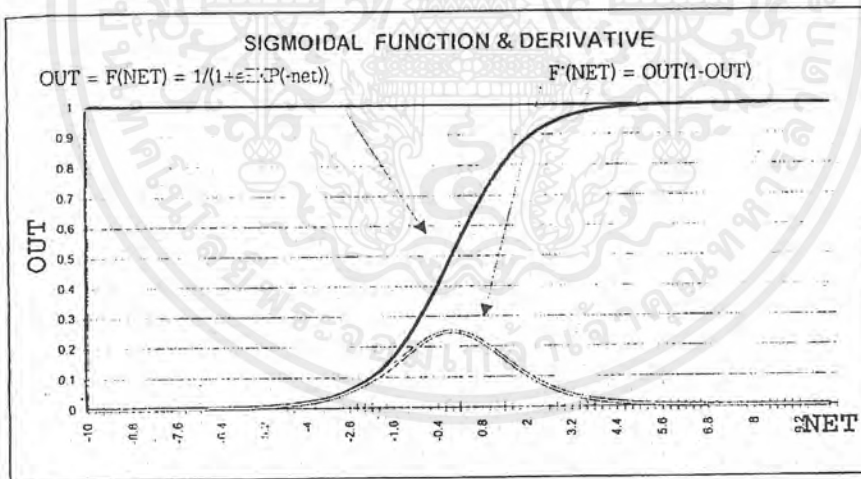
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงโมเดลนิวรอนที่สร้างขึ้นร่วมกับ Activation Function

ในภาพที่ 2.13 บล็อก F จะรับผลที่ได้จาก NET มาสร้างเป็นสัญญาณเอาต์พุตที่ OUT โดยกระบวนการภายในบล็อก F จะบีบช่วงของ OUT ให้อยู่ในขอบเขตจำกัด ตามต้องการ ดังนั้น ค่า OUT จะมีค่าไม่ต่ำกว่าช่วงที่กำหนดโดยค่าของ NET เราเรียกบล็อก F นี้ว่าสแควชชิ่งฟังก์ชัน (Squashing function) และโดยทั่วไป สแควชชิ่งฟังก์ชันที่ใช้เป็นแบบลอจิสติกฟังก์ชัน หรือซิกมอยด์ (Logistic function or “Sigmoid”) ซึ่งมีรูปร่างคล้ายตัว S โดยเขียนเป็นสมการคณิตศาสตร์ได้ดังนี้ คือ

$$F(x) = \frac{1}{(1 + e^{-x})} \tag{2.15}$$

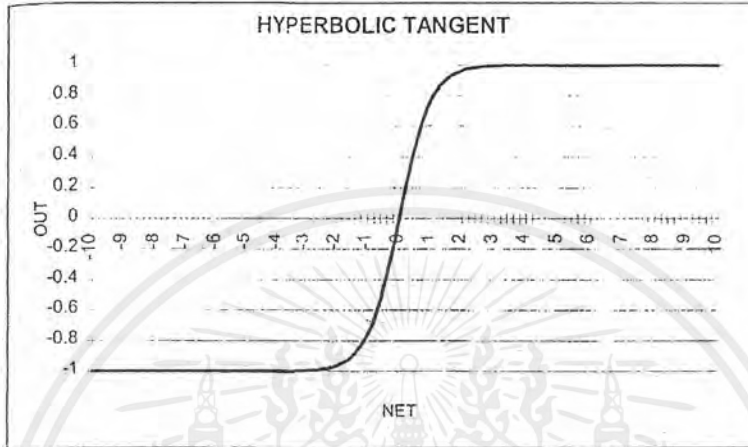


รูปที่ 2.14 แสดงกราฟที่ได้จากสมการซิมมอยด์ลอจิสติกฟังก์ชัน (Sigmoidal logistic function)

ลักษณะของเทรซโฮลฟังก์ชันมีลักษณะเป็น Non-linear function เช่น S-Curve เราจะได้ค่าเอาต์พุต ที่มีความไวต่อสัญญาณอินพุตที่มีขนาดเล็ก ๆ และเฉื่อยต่อสัญญาณแรง ๆ ซึ่งสัญญาณอ่อน ๆ ไปทางบวกเพียงเล็กน้อยก็จะทำให้ OUT ใกล้เคียง “1” กระตุ้นหรือสัญญาณอ่อน ๆ ทางลบเพียงเล็กน้อย ก็จะทำให้ Output ใกล้เคียง “0” (ขยับยั้ง) ขณะที่สัญญาณแรง ๆ ทางบวกก็ยังคงให้ Output ใกล้เคียง “1” และสัญญาณทางลบแรง ๆ ก็คงให้ Output ใกล้เคียง “1” และสัญญาณทางลบแรง ๆ ก็คงให้ Output ใกล้เคียง “0” เช่นกัน คุณลักษณะแบบนี้ เป็นแบบ NON-LINEAR GAIN

ซึ่งคลอสส์เบิร์ก (Grossberg, 1973) พบว่า คุณลักษณะที่เป็น Non-linear gain นี้สามารถแก้ปัญหา Noise-saturation dilemma ได้ และทำให้นิวรอลเทียมที่สร้างขึ้นสามารถทำงานกับขนาดของอินพุตได้กว้างมากขึ้น

ยังมีฟังก์ชันอื่น ๆ อีกคือ ไฮเปอร์โบลิก แทนเจนท์ (Hyperbolic tangent) มันจะมีลักษณะคล้ายกับ Logistic function และนิยมใช้บ่อย ๆ ในการสร้างโมเดลคณิตศาสตร์ การกระตุ้นเร้าความสนใจของเซลล์สมองเทียม ซึ่งมีคุณสมบัติคล้ายชีวภาพของเซลล์สมองคือ $OUT = \text{Tanh}(X)$



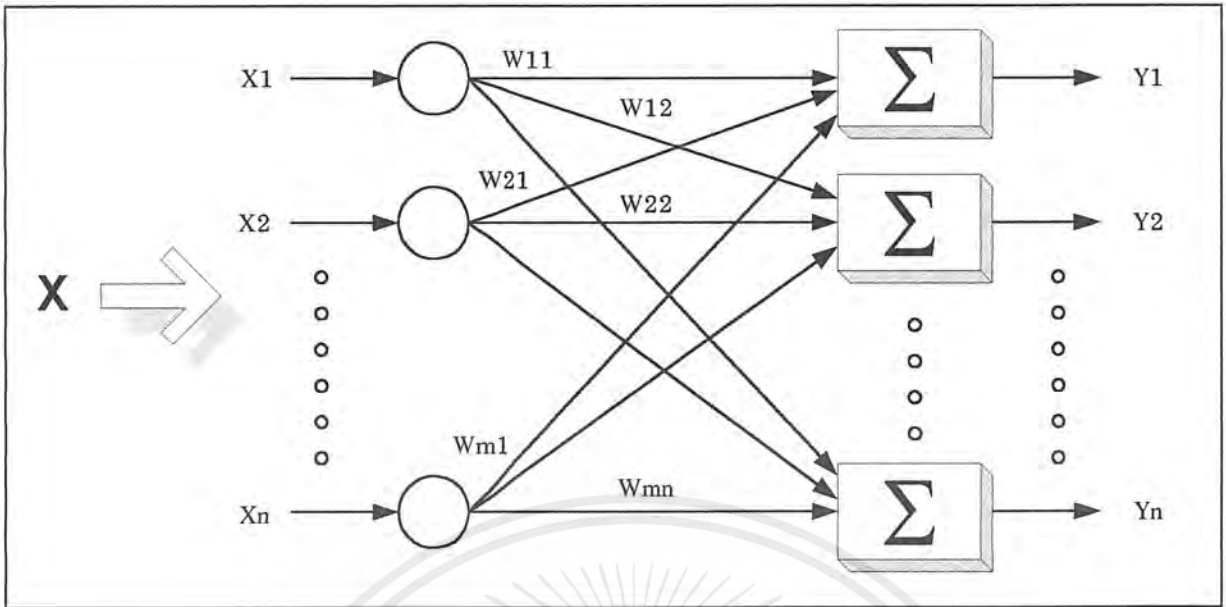
รูปที่ 2.15 แสดง Hyperbolic Tangent Function

จากภาพที่ 2.15 ส่วนที่เหมือนกับ ซิกมอยด์ ลอจิสติก ฟังก์ชัน คือมีลักษณะเป็น S แต่เนื่องจากมันจะมีความสมมาตร จึงให้ OUTPUT อยู่ระหว่าง “-1” ถึง “1” OUTPUT จะเป็น “0” เมื่อ NET เป็น “0” OUTPUT เข้าใกล้ “1” เมื่ออินพุตไปทางบวก และเข้าใกล้ “-1” เมื่อ อินพุตมีทิศทางไปทางลบ

2.10.4 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Artificial Neural Networks)

ที่กล่าวมาจนถึงจุดนี้ เป็นการกล่าวถึงหลักการ และเหตุผลในการสร้างเซลล์ประสาทเทียม เพียงหนึ่งเซลล์ โดยใช้แนวความคิดจากเซลล์ประสาทชีวภาพ การจะนำเซลล์ประสาทเทียมมาใช้งานได้นั้น ต้องใช้เซลล์ประสาทเทียมที่มีคุณลักษณะต่างๆ กัน (ค่า Weight จะทำให้คุณสมบัติของเซลล์ประสาทเทียม แต่ละเซลล์มีคุณสมบัติแตกต่างกันไป) มาเชื่อมโยงเป็นโครงข่ายในลักษณะเดียวกับเซลล์สมองชีวภาพเสียก่อน ซึ่งลักษณะการเชื่อมโยงมีหลายชนิด แต่ละชนิดก็มีคุณลักษณะเด่นที่แตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงลักษณะโครงข่ายประสาทเทียมแบบชั้นเดียว (Single-Layer Neural Networks)

จากภาพที่ 2.16 เป็นโครงข่ายประสาทเทียมแบบชั้นเดียว ที่ประกอบด้วยเซลล์ประสาทเทียมง่าย ๆ หลายๆ ชุด ความสามารถในการคำนวณของโครงข่ายประสาทเทียม ได้มาจากลักษณะการเชื่อมต่อ เป็นโครงข่ายประสาทเทียมโครงข่ายง่าย ๆ เป็นกลุ่มโมดูลประสาทเทียมที่เชื่อมต่อกันเป็นชั้นๆ (Layer) ดังแสดงในภาพที่ 2.17 เป็นโครงข่ายประสาทเทียม แบบชั้นเดียว (Single layer) ที่ประกอบด้วยเอาต์พุทเลเยอร์ (กลุ่มของบล็อกซิมเมชันที่อยู่ทางขวามือ) และอินพุทเลเยอร์ (วงกลมทางซ้ายมือ) โดยไม่พิจารณาอินพุทเลเยอร์ว่าเป็นนิวรอลเลเยอร์ เนื่องจากอินพุทเลเยอร์จะทำหน้าที่เชื่อมต่ออินพุทที่รับมา และส่งออกไปให้ยังแต่ละอินพุทนิวรอลเลเยอร์ (ในที่นี้คือ Output layer) ในชั้นถัดไป โดยแต่ละอินพุทจะถูกคูณโดยค่าน้ำหนักเฉพาะแต่ละอินพุท โครงข่ายประสาทเทียมที่สร้างขึ้นในชั้นแรกไม่ซับซ้อน โดยแต่ละนิวรอลจะได้เอาต์พุทจาก

$$\text{OUT} = \text{Logistic Function} \text{ คูณ (ผลรวมของ Input X กับ Weight)}$$

หรือ

$$\text{OUT} = F(\text{NET}) \quad (2.16)$$

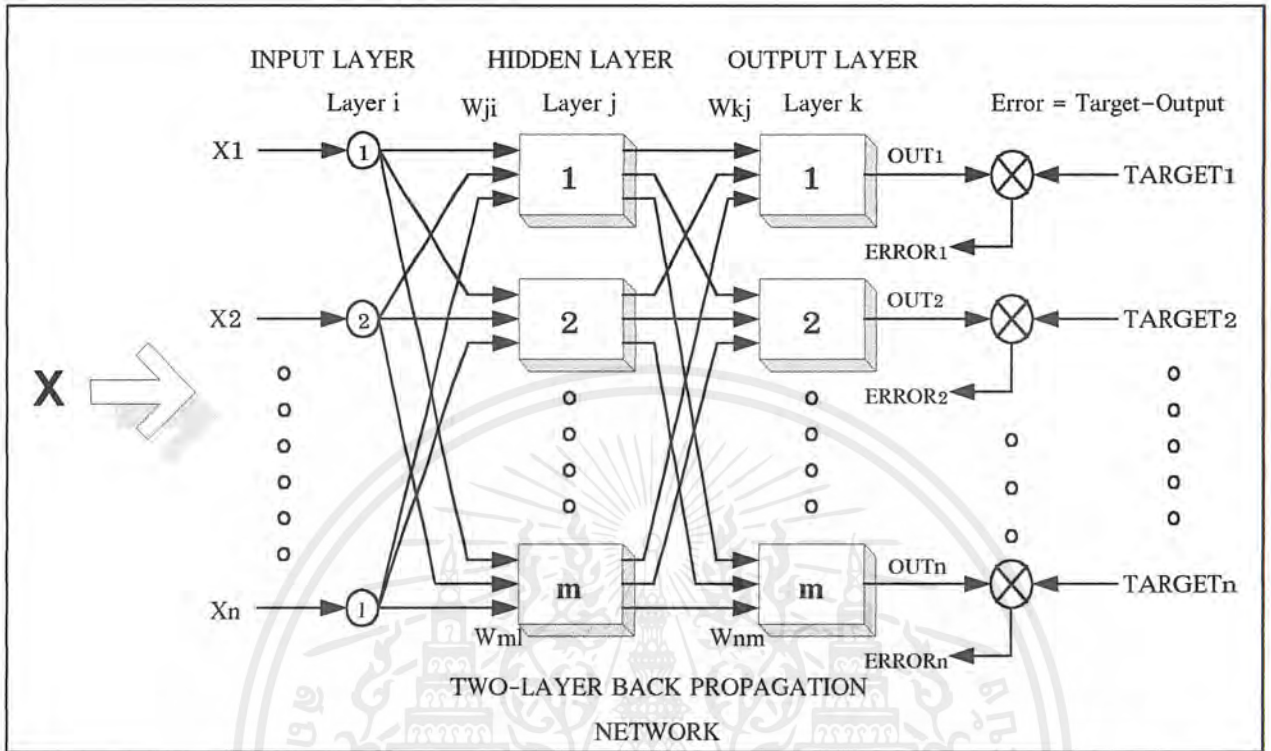
อย่างไรก็ดีลักษณะการเชื่อมโยงระหว่างโครงข่ายไม่ได้มีแบบเดียว การเชื่อมโยงระหว่างเลเยอร์อาจมีการเชื่อมโยงย้อนกลับมาที่อินพุทเลเยอร์อีก ซึ่งโครงข่ายประสาทชีวภาพก็มีลักษณะดังกล่าวเช่นกัน สำหรับค่า Weight ในภาพที่ 2.17 มีวิธีการพิจารณาในรูปของ เมตริกน้ำหนัก (Weight matrix) ซึ่งหากโครงข่ายมีหลายชั้น จะช่วยให้ระบุค่าน้ำหนักได้ง่ายขึ้น และเพื่อหลีกเลี่ยงความสับสนจะกำหนดเป็นโดเมนชั้น (Dimensions) ของเมตริก โดยให้ m แทนจำนวนแถว หรือจำนวนของอินพุท และ n แทนจำนวนของนิวรอล ที่สร้างขึ้น ตัวอย่างเช่น น้ำหนักที่เชื่อมระหว่างอินพุทตัวที่ 4 กับนิวรอลตัวที่ 2 คือ $W_{4,2}$

2.10.5 โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Artificial Neural Networks)

โครงข่ายที่ซับซ้อนจะมีความสามารถในการคำนวณที่ดีขึ้นมันจะเป็นโครงข่ายที่มีโครงสร้างเป็นจินตนาการที่น่าเป็นไปได้โดยการจัดการเชื่อมโยงนิวรอล มีโครงสร้างเป็นชั้นๆ คล้ายส่วนหนึ่งของสมอง และมีการพัฒนาอัลกอริทึมเกี่ยวกับการฝึกสอนให้โครงข่ายแบบหลายชั้นทำงานได้ ตามความต้องการแล้วเมื่อไม่นานมานี้ โครงข่ายแบบหลายชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจจะสร้างจาก กลุ่มของโครงข่ายแบบชั้นเดียวเอาต์พุทของ Layer หนึ่ง จะใช้เป็นอินพุทของ Layer ถัดไป ในภาพที่ 2.17 แสดงเน็ตเวิร์คที่มีการเชื่อมต่อแบบสองชั้น



รูปที่ 2.17 แสดงไดอะแกรมของ Backpropagation Neural Networks แบบสองชั้น

ภาพที่ 2.17 แสดงโครงข่ายประสาทเทียมแบบหลายชั้น ที่ต่อเชื่อมโยงแบบเต็มชั้น ในโครงข่ายแบบหลายชั้น มีการเรียกชื่อชั้นต่างๆ ดังนี้ คือ ชั้นที่ต่อโดยตรงกับอินพุท เรียกว่า อินพุทเลเยอร์ (Input layer) ชั้นนี้จะไม่มีการคำนวณ แต่จะทำหน้าที่ต่อเชื่อมข้อมูลไปยังชั้นถัดไป ชั้นที่อยู่ท้ายสุดทางขวามือเรียกว่า เอาต์พุทเลเยอร์ (Output layer) เป็นชั้นที่โครงข่ายจะให้ผลลัพธ์ ส่วนชั้นที่อยู่ระหว่างอินพุทเลเยอร์ และเอาต์พุทเลเยอร์ จะมีกี่ชั้นก็ตามจะเรียกว่า ฮิดเดนเลเยอร์ (Hidden layer) หากฮิดเดนเลเยอร์ มีหลาย ๆ ชั้น ก็จะมีการตั้งชื่อเฉพาะลงไปให้กับแต่ละชั้น

2.10.6 ฟังก์ชันกระตุ้นความสนใจแบบไม่เป็นเชิงเส้น (The Nonlinear Activation Function)

การนำเอาต์พุทของเลเยอร์หนึ่ง มาเชื่อมกับอินพุทของเลเยอร์ชั้นถัดไป โดยผ่านฟังก์ชันกระตุ้นความสนใจแบบไม่เป็นเชิงเส้น จะทำให้โครงข่ายมีความสามารถในการคำนวณเพิ่มขึ้น (หากไม่ผ่านฟังก์ชันดังกล่าวความสามารถในการคำนวณจะไม่เพิ่มขึ้น และจะมีความสามารถไม่แตกต่างไปจาก Single layer networks) และสามารถกำหนดขอบเขตของเอาต์พุทให้อยู่ในช่วงที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.7 การฝึกสอนให้กับโครงข่ายประสาทเทียม (Training of Artificial Neural Networks)

ค่าน้ำหนัก มีความสัมพันธ์กับอะไร เปลี่ยนแปลงอย่างไร นั่นก็เช่นเดียวกับเด็กที่คลอดออกมา ก็มีสมองแล้ว แต่สมองยังไม่เจริญเติบโตเพียงพอ และยังไม่ได้รับการฝึกสอน และเรียนรู้ เด็กจึงไม่สามารถทำกิจกรรมใดๆ ด้วยตนเอง เว้นแต่กิจกรรมที่ธรรมชาติสร้างมาพร้อมกับการกำเนิดที่เรียกว่า “สัญชาตญาณ” ซึ่งธรรมชาติใส่คุณลักษณะบางอย่างให้เซลล์สมองบางส่วนตั้งแต่ทารกเจริญเติบโตอยู่ในครรภ์มารดา เช่น ระบบควบคุมการหายใจ, ความรู้สึก, การเรียกกร้องเมื่อหิว, การตอบสนองต่อสิ่งเร้า ฯลฯ เด็กจะพัฒนาการเรียนรู้ไปตามขั้นตอน หลังจากนั้นสมองของเขาก็จะได้รับการฝึกสอน และเจริญเติบโตไปพร้อมๆ กัน เซลล์สมองจะได้รับการปรับคุณลักษณะสอดคล้องกับการฝึกสอน และจะเจริญเป็นโครงข่ายสอดคล้องกัน

โครงข่ายประสาทเทียมที่สร้างขึ้นมีลักษณะเช่นเดียวกัน คือ เมื่อสร้างเสร็จ แต่ละเซลล์ประสาทที่สร้างขึ้นมานั้น จะยังไม่มีคุณลักษณะใดเลย เนื่องจากยังไม่มีการกำหนดค่าซินแนปติกส์เฉพาะที่เหมาะสมกับงานที่ต้องการให้กับมัน จึงต้องมีการฝึกสอนเพื่อให้เน็ตเวิร์ค ที่สร้างขึ้นมีคุณลักษณะตามที่ต้องการ การฝึกสอนของโครงข่ายประสาทเทียม จะกระทำโดยการปรับเปลี่ยนค่าน้ำหนักซินแนปติกส์ เพื่อให้โครงข่ายจดจำ แพตเทิร์น ความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตได้ โดยในขั้นแรกอาจกำหนดเป็นค่าสุ่มใด ๆ (Random weight) ก่อนแล้วถึงปรับเปลี่ยนน้ำหนักไปตามอัลกอริทึมสมมติฐาน หลายๆ รอบจนกว่าจะได้เอาต์พุตของเน็ตเวิร์ค เหมือนกับเอาต์พุตที่ต้องการ ในเงื่อนไขความผิดพลาดที่ยอมรับได้

2.10.8 วัตถุประสงค์ของการเทรนนิ่ง (Objective of Training)

เนื่องจากค่าน้ำหนักที่เป็นค่าสุ่มใด ๆ โครงข่ายจึงไม่แสดงคุณลักษณะใดออกมา การฝึกสอน (Training) ให้โครงข่ายก็คือ การปรับค่าน้ำหนักทุกๆ จุด ให้สอดคล้องกับอินพุตหลายๆ แบบ เพื่อให้ได้เอาต์พุตตามต้องการนั่นเอง การฝึกสอนโครงข่าย จะต้องบรรลุถึงกระบวนการเข้าใจพื้นฐานเสียก่อน คือ การเรียนรู้ในโครงข่ายประสาทเทียมนั้นมีขีดจำกัด ปัญหาต่างๆ ผู้ใช้คงต้องแก้ให้มันก่อน แล้วนำผลนั้น ไปอ้างอิงสำหรับการปรับปรุงค่าน้ำหนัก หลังจากปรับน้ำหนักจนได้ค่าผิดพลาดที่เอาต์พุตเทียบกับเป้าหมายน้อยลงเป็นที่พอใจแล้ว โครงข่ายประสาทเทียมนั้นก็พร้อมที่จะวิเคราะห์อินพุต และให้เอาต์พุตตามลักษณะตัวอย่างที่มันเคยเรียนรู้มา การเรียนรู้จะมีการปรับน้ำหนักหลายๆ รอบจนค่าน้ำหนักสอดคล้องกับตัวอย่างหลายๆ ตัวอย่าง และให้เอาต์พุตตามต้องการ พบว่าโครงข่ายได้ตัวอย่างสำหรับการเทรนนิ่งมากๆ โครงข่ายก็จะมีคามแม่นยำสูงขึ้น แต่ก็ใช้เวลาในการเทรนนิ่งเพิ่มขึ้นเช่นกัน หากพิจารณาต่อไปจะพบว่า โครงข่ายประสาทเทียมที่สร้างขึ้นจะมีพฤติกรรมคล้ายกับระบบการเรียนรู้ของมนุษย์มาก เป็นเพราะมีต้นแบบมาจากระบบประสาทชีวภาพนั่นเอง

2.10.9 การเทรนนิ่งแบบควบคุม (Supervised Training)

เทรนนิ่งอัลกอริทึมถูกจัดเป็น 2 ประเภท คือ แบบควบคุม (Supervised training) และแบบอิสระ (Unsupervised training) โดย การเทรนนิ่งแบบควบคุม จะต้องการคู่ของเทรนนิ่งระหว่างอินพุตกับเป้าหมายที่ต้องการ ที่เรียกว่า เทรนนิ่งแพร์ (Training pairs) โครงข่ายจะถูกเทรนไปตามจำนวนของคู่ที่เทรนนิ่ง (จำนวนคู่ของ Input กับ Output ที่ต้องการให้โครงข่ายรู้จัก) เอาต์พุตที่คำนวณได้จากโครงข่ายจะถูกเปรียบเทียบกับความสอดคล้องกับเป้าหมาย ค่าผิดพลาดที่เกิดขึ้นจะถูกป้อนกลับไปยังโครงข่าย และเปลี่ยนแปลงค่าน้ำหนักให้สอดคล้องกับอัลกอริทึม ที่ทำให้น้ำหนักของค่าผิดพลาดที่เกิดขึ้นระหว่างเอาต์พุตกับเป้าหมายโดยเฉลี่ยมีค่าลดต่ำลง ตัวอย่างการเทรนนิ่งแบบนี้ ได้แก่ การเทรนนิ่งแบบแพร่กลับ (Back propagation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.10 การเทรนนิ่งแบบอิสระ (Unsupervised Training)

ถึงแม้ว่า อัลกอริทึมแบบควบคุม (Supervised training) สามารถจะประยุกต์ใช้เพื่อปรับคุณลักษณะของโครงข่ายได้สำเร็จ แต่ก็ยังมีข้อวิจารณ์อยู่ คือ มันเป็นไปอย่างแบบชีวภาพไม่ได้ และยากที่จะเชื่อได้ว่า กลไกการเทรนนิ่งของสมองจะต้องการ การเปรียบเทียบระหว่างคนที่ต้องการกับเอาท์พุทจริง โดยกระบวนการป้อนกลับไปแก้ไขคุณลักษณะของโครงข่าย และถ้าสมมติว่า ถ้าสมองมีกลไกเช่นนี้ ต้องมีผู้หาเอาท์พุทที่ต้องการเพื่อนำมาเป็นเป้าหมายตลอดเวลา และจะเอามาจากที่ใด สรุปคือ ต้องมีผู้คิดเป้าหมายให้กับโครงข่ายก่อน โครงข่ายไม่สามารถคิด และปรับคุณลักษณะได้ก่อนด้วยตนเอง ในทางตรงกันข้ามหากพิจารณาทารกแรกเกิดสมองของเขาสามารถจัดระบบเองได้อย่างไร การเทรนนิ่งแบบระบบของสมอง จนกระทั่งมีการพัฒนาการเทรนนิ่งแบบอิสระนี้ขึ้นราวปี 1984 โดยโคโฮเนน (Kohonen) และบุคคลอื่นๆ โดยได้เสนอแนวคิดที่เป็นการเทรนนิ่งแบบไม่ต้องการเป้าหมาย ไม่มีการตัดสินใจด้วยเหตุผลในอุดมคติมาก่อน ชุดของการเทรนนิ่ง จะมีเพียงอินพุทเวกเตอร์เท่านั้น เทรนนิ่งอัลกอริทึม จะเปลี่ยนแปลงค่าน้ำหนักของโครงข่าย เพื่อสร้างเอาท์พุทที่มีความมั่นคง ยกตัวอย่าง เช่น หากให้โครงข่ายรู้จำภาพหน้าคนหนึ่ง หากภาพหน้าคนคนนั้น เปลี่ยนแปลงไปเล็กน้อย (Image อาจมี Noise รวมอยู่บ้าง) โครงข่ายนั้นก็ยังสามารถบอกได้ว่า คนคนนั้นเป็นคนเดิมเป็นต้น การเทรนนิ่งจะไม่มี การตัดสินใจมาก่อน ไม่มีการกำหนดแบบเอาท์พุทมาก่อน (อาจกล่าวได้ว่าแบบเอาท์พุทจะถูกกำหนดโดยอินพุทเวกเตอร์นั่นเอง) ดังนั้น เอาท์พุทของโครงข่ายก็เช่นกัน ส่วนใหญ่จะถูกแปรรูป ซึ่งจะเข้าใจได้ภายหลังกระบวนการเทรนนิ่ง ดังนั้นจึงไม่สามารถแก้ปัญหาที่เคร่งครัดสำคัญได้ แต่มักนิยมใช้โครงข่ายแบบนี้กับงานง่าย ๆ ประเภทการเปรียบเทียบเอกลักษณ์, รูปแบบที่สัมพันธ์กันระหว่างอินพุท-เอาท์พุท ที่ถูกกำหนดโดยโครงข่าย

2.10.11 วิธีการแก้ปัญหาการฝึกสอน (Training Algorithm)

ส่วนใหญ่แล้วทุกวันนี้ การแก้ปัญหาฝึกสอนของโครงข่ายค่อย ๆ พัฒนาก้าวหน้าขึ้นจากแนวความคิดของ ดี โอ เฮบบ์ (ปี 1961) เขาได้เสนอโมเดลของ การเทรนนิ่งแบบอิสระ (Unsupervised training) ในแบบซินแนปติกสเตรงท์ หรือน้ำหนัก ซึ่งจะเพิ่มขึ้น ถ้าทั้งแหล่งกำเนิด (Input Source) และจุดหมายปลายทาง (Destination) ของนิวรอลได้รับการสนใจ กรณีนี้ถ้ามีการใช้งานทางเส้นนี้บ่อย ๆ ก็จะทำให้ซินแนปติกสเตรงท์ (หรือน้ำหนัก) แข็งแรงขึ้น (เซลล์สมองที่ใช้งานมากบ่อย ๆ ก็จะทำให้ ซินแนปส์ใหญ่ขึ้น การส่งผ่านข้อมูลพัลส์ไฟฟ้าทำได้ดีขึ้น ทำให้มีประสิทธิภาพดีขึ้น เช่น สามารถคิด หรือจดจำได้เร็ว และดีขึ้น) โครงข่ายประสาทเทียมนี้ ใช้การเรียนรู้แบบเฮบบ์ (Hebbian learning) จะเพิ่มค่าน้ำหนักของโครงข่ายอย่างสอดคล้องกับผลคูณของระดับความสนใจของแหล่งกำเนิด และจุดหมายของนิวรอลตามสมการดังนี้

$$W_{ij}(n+1) = W_{ij}(n) + \alpha \text{OUT}_i \text{OUT}_j \quad (2.17)$$

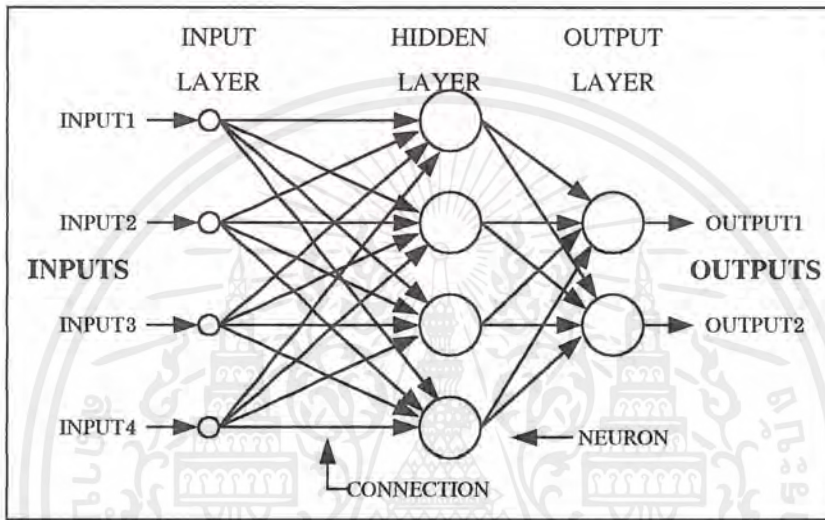
โดย $W_{ij}(n)$ คือค่าน้ำหนัก จากนิวรอล I ไปยังนิวรอล j ก่อนปรับปรุงค่า
 $W_{ij}(n+1)$ คือค่าน้ำหนัก จากนิวรอล I ไปยังนิวรอล j หลังปรับปรุงค่า
 α คือค่าคงที่ของการเรียนรู้ (Learning rate coefficient)
 OUT_i คือเอาท์พุทของนิวรอล I และเป็นอินพุทของนิวรอล j
 OUT_j คือเอาท์พุทของนิวรอล j

โครงข่ายที่มีลักษณะการเทรนนิ่งแบบเฮบบ์นั้น เป็นผลมาจากการพัฒนามาแล้วกว่า 20-30 ปี โดยเฉพาะงานของโรเซนเบลทท์ (Rosenblatt:1962), วิโดว์ (Widrow:1959), วิโดว์และฮอฟฟ์ (Widrow&Hoff:1960) และอีกหลาย ๆ คนที่พยายามพัฒนาระบบการเทรนนิ่งแบบควบคุม ที่สร้างโครงข่ายที่สามารถเรียนรู้แบบของอินพุทได้อย่างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว้างขวาง และมีอัตราการเรียนรู้สูง ที่บรรลุผลได้จากหลักการพื้นฐานของการเรียนรู้ หรือเทรนนิ่งให้กับโครงข่ายเช่น Perceptrons, Hopfieldnets, Backpropagation Networks และ Counter propagation.

2.10.12 การประยุกต์ใช้นิวรัลเน็ตเวิร์คกับปริญาณิพนธ์

จากการศึกษาพฤติกรรม การสื่อสารของมนุษย์พบว่า การเรียนรู้ภาษาต้องเรียนรู้จากผู้อื่น ภาษาไม่ใช่สิ่งที่เป็นสัญชาตญาณ ดังนั้น จึงต้องเรียนรู้จากภาษาที่สังคมนั้นกำหนดขึ้น ซึ่งเป็นแพตเทิร์นของภาษาที่บรรพบุรุษได้สร้างขึ้นสั่งสมกันมาก่อน จากเหตุผลดังกล่าว พบว่าสอดคล้องกับอัลกอริทึม การเทรนนิ่งแบบควบคุม (Supervised training) ผู้ทำจึงเลือกใช้วิธี Backpropagation มีดีเลเยอร์เป็นแบบ 2 เลเยอร์ แสดงดังภาพที่ 2.18



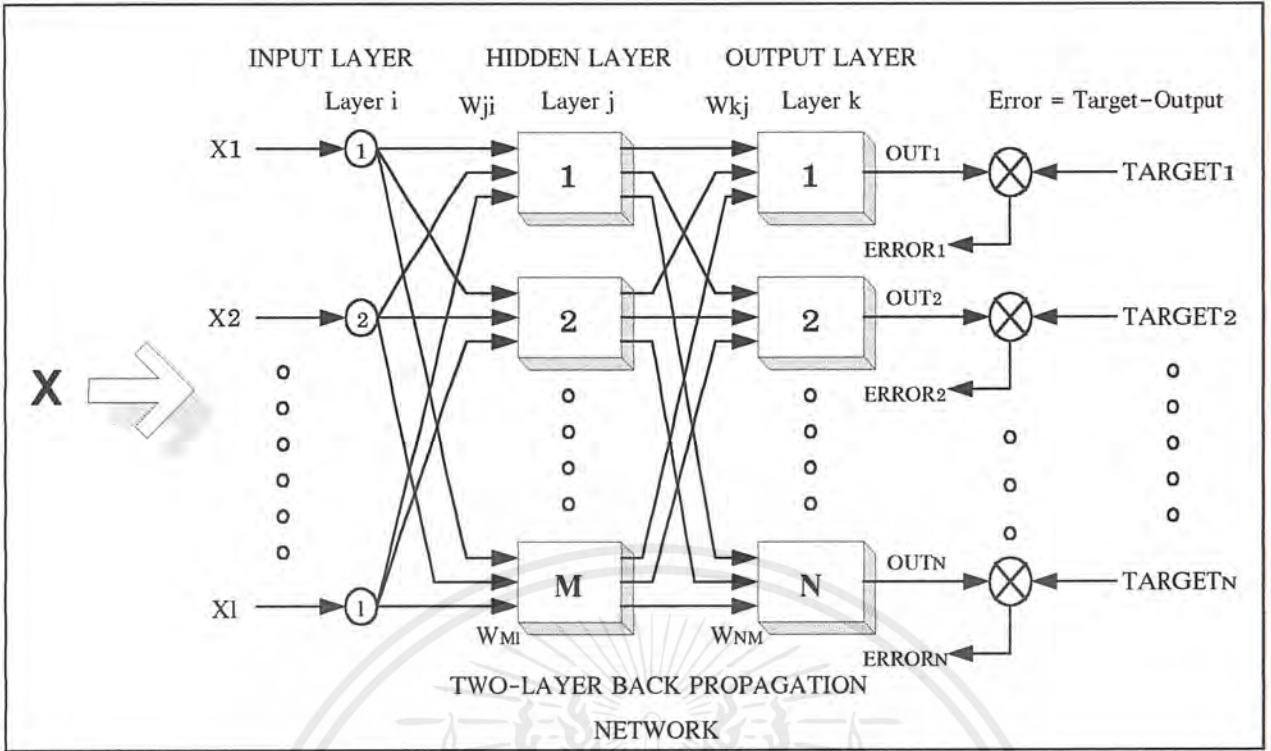
รูปที่ 2.18 แสดงไดอะแกรมของ Two Layer Feed-Forward

พิจารณาภาพที่ 2.18 ประกอบ จะเห็นว่า ประกอบด้วย input array, hidden layer และ output layer ในส่วนของ Input array จะไม่ถือเป็น layer เพราะไม่มีการคำนวณใดๆ แต่มันจะทำหน้าที่เชื่อมโยงข้อมูลให้กับ Hidden layer ทุกๆ นิวรอล output ของ hidden layer ก็จะถูกเชื่อมโยงไปเป็น input ให้กับ output layer ทุกอินพุทของนิวรอลทุกตัว จะมีค่า Weight ประจำตัว ค่า Weight นี้ทำให้ระดับของอินพุทผ่านเข้าไปภายในนิวรอลแต่ละตัวไม่เท่ากัน ซึ่งเป็นตัวกำหนดคุณลักษณะสมบัติของนิวรอลแต่ละตัวในเน็ตเวิร์ค วิธีการกำหนดคุณสมบัติดังกล่าว ทำได้โดยการเทรนนิ่งโดยในขั้นแรก จะสมมุติว่า ค่าแบบสุ่มน้อยๆ ให้ช่วงที่ใช้คือตั้งแต่ -0.5 ถึง $+0.5$ แล้วจึงค่อยๆ ปรับเปลี่ยนค่าน้ำหนักในแต่ละ layer ให้สอดคล้องกับคุณสมบัติที่ต้องการ

2.10.13 กระบวนการเทรนนิ่ง

กระบวนการเทรนนิ่ง คือ กระบวนการปรับคุณลักษณะของนิวรัลเน็ตเวิร์ค ให้มีคุณสมบัติตามที่ต้องการ การเทรนนิ่งของอัลกอริทึม Backpropagation เป็นแบบ Supervised Training แบ่งเป็น 2 ส่วนคือ Forward pass และ Reversed pass การเทรนนิ่งจะกระทำทั้ง Forward pass และ Reverse pass สลับกันไป จนกว่าค่าผิดพลาดระหว่าง Output กับ Target จะมีค่าต่ำกว่าจุดที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงไดอะแกรมเชื่อมโยงระหว่างชั้นของนิวรอลแบบ 2 ชั้นของ Backpropagation Network

FORWARD PASS

จากภาพ 2.19 อินพุต X ตั้งแต่ X_1 ถึง X_I จะถูกคูณด้วย Weight matrix ผลรวมของอินพุตเมื่อคูณกับค่า Weight ของแต่ละนิวรอลเรียกว่า เน็ต (NET) เขียนเป็นสมการได้คือ

$$NET_j = X_1 W_{j1} + X_2 W_{j2} + \dots + X_I W_{ji} \quad (2.18)$$

หรือ

$$NET_j = \sum_{i=1}^I X_i W_{ji} \quad (2.19)$$

- โดย
- X คือ ค่าที่รับเข้ามาของ Input node
 - W คือ ค่าน้ำหนักประจำตัวของแต่ละอินพุตโนด
 - I คือ หมายเลขโนดโดยเริ่มที่โนด 1 ถึง โนด I
 - J คือ หมายเลขนิวรอลใน Hidden layer ตั้งแต่นิวรอลที่ 1 ถึง M
 - K คือ หมายเลขนิวรอลใน Output layer ตั้งแต่นิวรอลที่ 1 ถึง N

และทำนองเดียวกัน

$$NET_k = \sum_{j=1}^M OP_j W_{kj} \quad (2.20)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ค่า NET ของแต่ละนิวรอนแล้ว จะปรับให้ได้ Output อยู่ในช่วงที่ต้องการเป็นค่าที่จะนำไปยับยั้ง หรือ กระตุ้นนิวรอนตัวถัดไปโดย Squashing Function ที่ใช้เป็น Logistic Function หรือ Sigmoid จะได้ตามสมการ 2.21 และ 2.22

$$OP_j = F_j(NEF_j) = \frac{1}{1 + e^{-(net/\theta_j)}} \quad (2.21)$$

$$OP_k = F_k(NEF_k) = \frac{1}{1 + e^{-(net/\theta_k)}} \quad (2.22)$$

กำหนด

$$\theta_j = 0.05 \text{ และ } \theta_k = 0.05$$

θ_j และ θ_k คือ ค่าที่กำหนดความชันของ Sigmoid curve

OP_j คือ ค่า Output ที่ Neuron j ของ Hidden Layer (เป็น Input ของ Output Layer)

OP_k คือ ค่า Output ที่ Neuron k ของ Output Layer (เป็น Output ของ Output Layer)

การคำนวณหาค่า Output จะเริ่มที่ Hidden layer ที่ชั้น j ก่อน เมื่อได้ค่า output ของ Hidden layer ครบทุกตัวแล้ว จึงคำนวณหาค่า Output ที่ Output layer โดย Input ของ Output layer ก็คือ Output ของ Hidden layer นั้นเอง โดยใช้การคำนวณทำนองเดียวกันกับ Hidden layer Output ที่ได้จะนำไปเปรียบเทียบกับ Target ที่ต้องการเกิดเป็น Error ขึ้น ค่า Error นี้ จะถูกนำไปใช้ร่วมกับค่าดิฟเฟอเรนเชียล ของ $F(F')$ ใน Output layer เพื่อปรับค่า Weight ให้กับ Output layer ดังจะกล่าวต่อไปในส่วนของ Reverse pass

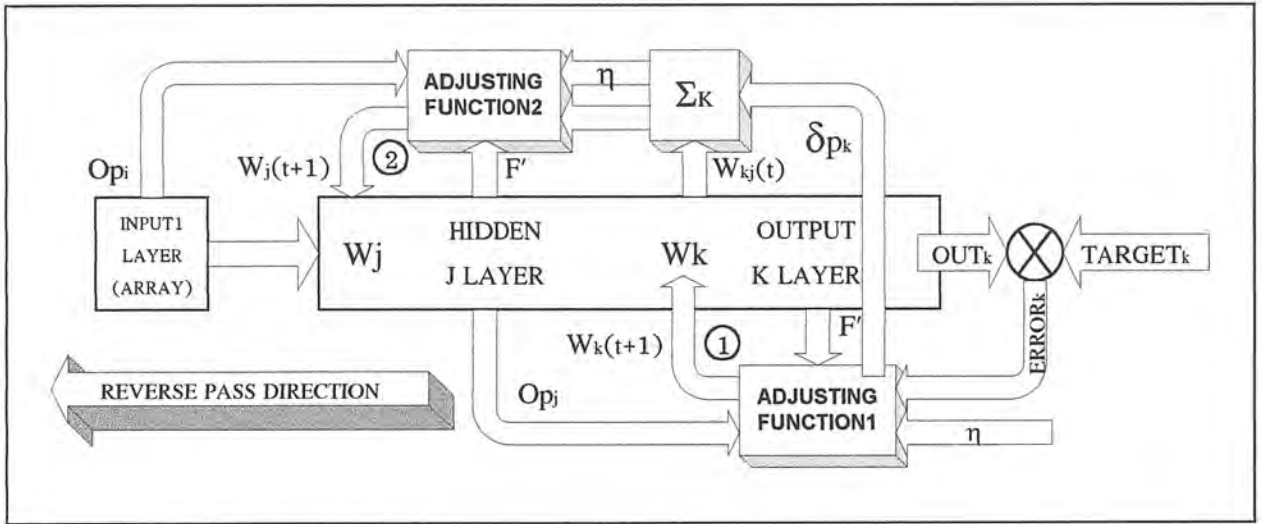
REVERSE PASS

เป็นส่วนที่ป้อนกลับค่า Error(Target-Output) จากเอาท์พุทเลเยอร์ย้อนกลับมาปรับปรุงค่าน้ำหนักที่เลเยอร์ก่อนหน้า สำหรับนิวรอนเน็ตเวิร์คที่มีมากกว่า 1 เลเยอร์ การปรับปรุงค่าน้ำหนักชั้นถัดถอยมา จะใช้ค่าน้ำหนักเดิมของเลเยอร์ที่สูงกว่า ร่วมกับตัวร่วมทางด้าน Output ของเลเยอร์มาคำนวณ (เนื่องจากไม่มีค่า Error มาใช้เหมือน Output layer) ดังจะกล่าวในรายละเอียดดังต่อไปนี้

จากภาพที่ 2.20 การปรับเปลี่ยนค่าน้ำหนักใหม่ จะเป็นไปตามทิศทางของลูกศร และได้ข้อมูลจากค่า Error ที่ Output layer เป็นต้นกำเนิดเพื่อกำหนดทิศทางปรับค่าน้ำหนักให้นิวรอนเน็ตเวิร์ค มีคุณลักษณะสมบัติตามต้องการลำดับขั้นตอนของ Reverse pass พอสรุปได้ดังนี้คือ

1. นำ Output ที่ได้จาก Output Layer ไปลบออกจาก Target ได้เป็น error
2. นำค่า Error มาคูณกับ F' ที่ได้จาก Output layer เรียกว่า δp_k
3. นำค่า δp_k คูณกับ OP_j แล้วคูณกับ η (อัตราการเรียนรู้ 0.01-1)
4. ค่า Weight ค่าใหม่ที่ได้ คือ ค่า Weight เดิมบวกกับผลลัพท์ ที่ได้จากข้อ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงไดอะแกรมของการปรับค่า Weight ในชั้น Output layer และ Hidden layer

พอเขียนเป็นสมการใหม่ได้คือ

$$W_{kj}(t+1) = W_{kj}(t) + \eta \delta p_k O_{pj} \quad (2.23)$$

เพราะว่า $\eta \delta p_k O_{pj} = \Delta W_{kj}$ (2.24)

ฉะนั้น $W_{kj}(t+1) = W_{kj}(t) + \Delta W_{kj}$ (2.25)

เมื่อ $W_{kj}(t)$ คือ ค่าน้ำหนักเดิมก่อนปรับ
 $W_{kj}(t+1)$ คือ ค่าน้ำหนักหลังปรับ
 η คือ อัตราการเรียนรู้
 O_{pj} คือ Output ของ Layer j
 δp_k คือ ค่าที่ได้จาก Error คูณกับดิฟเฟอเรนเชียลของฟังก์ชัน F

การปรับค่าของ Layer ถัดมา (Hidden layer) มีลักษณะคล้ายคลึงกับ Output layer คือ

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta p_j O_{pi} \quad (2.26)$$

เพราะว่า $\eta \delta p_j O_{pi} = \Delta W_{ji}$ (2.27)

ฉะนั้น $W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}$ (2.28)

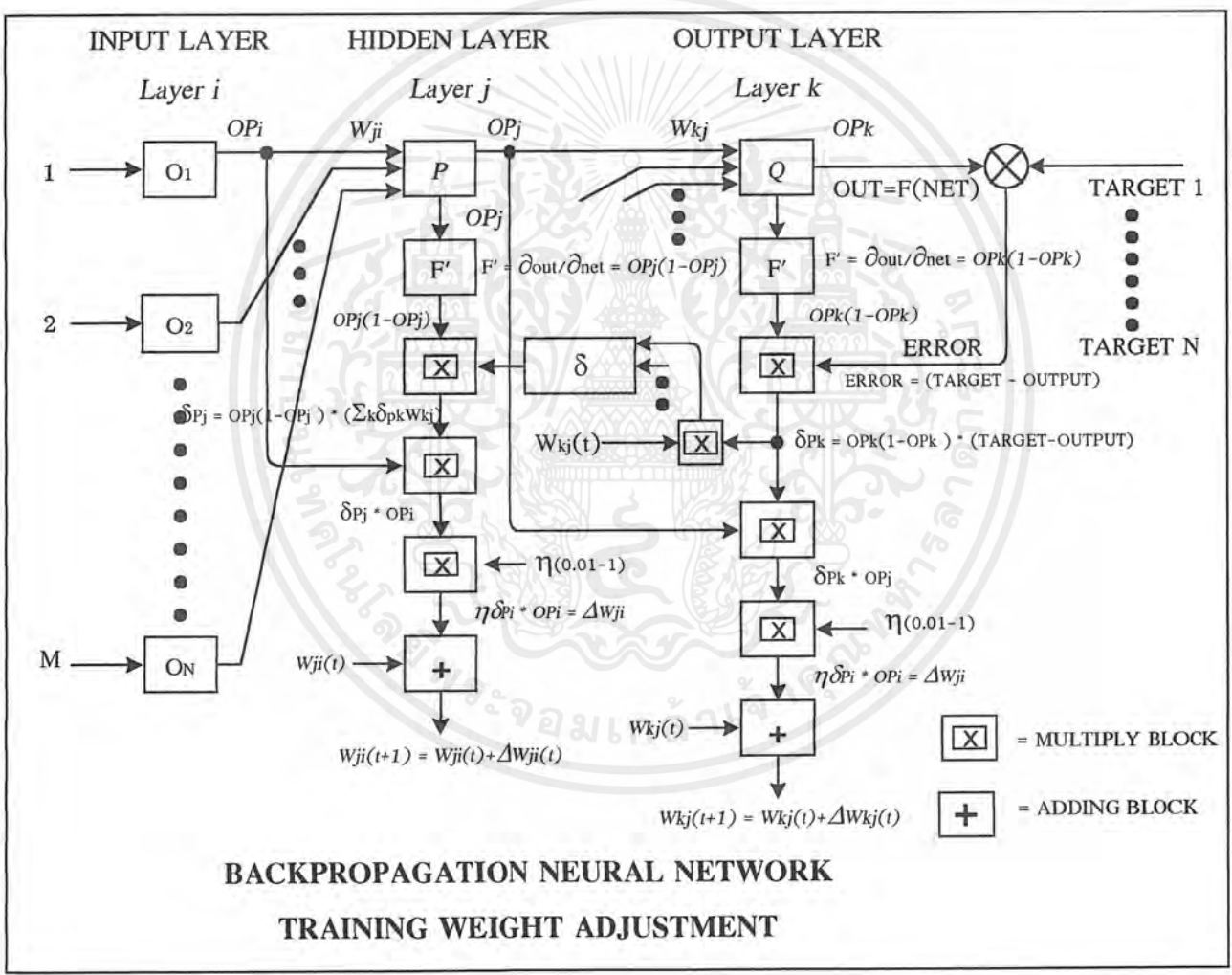
สมการที่ 2.27 แตกต่างกับสมการ 2.24 ตรงการได้มาซึ่ง δp_j เพราะไม่มีค่า Output กับ Target ที่ Hidden layer ค่าดังกล่าวสามารถทดแทนได้ด้วย $\sum_k (\delta p_k W_{kj})$

โดย $\delta p_j = O_{pj}(1-O_{pj})(\sum_k (\delta p_k W_{kj}))$ (2.29)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อ $W_{ji}(t)$ คือ ค่าน้ำหนักเดิมก่อนปรับ
- $W_{ji}(t+1)$ คือ ค่าน้ำหนักหลังปรับ
- η คือ อัตราการเรียนรู้
- OP_i คือ ข้อมูลที่ได้จาก Input Layer
- δ_{pj} คือ ค่าควบคุมให้น้ำหนักเปลี่ยนแปลงเข้าสู่จุดที่ดีที่สุด

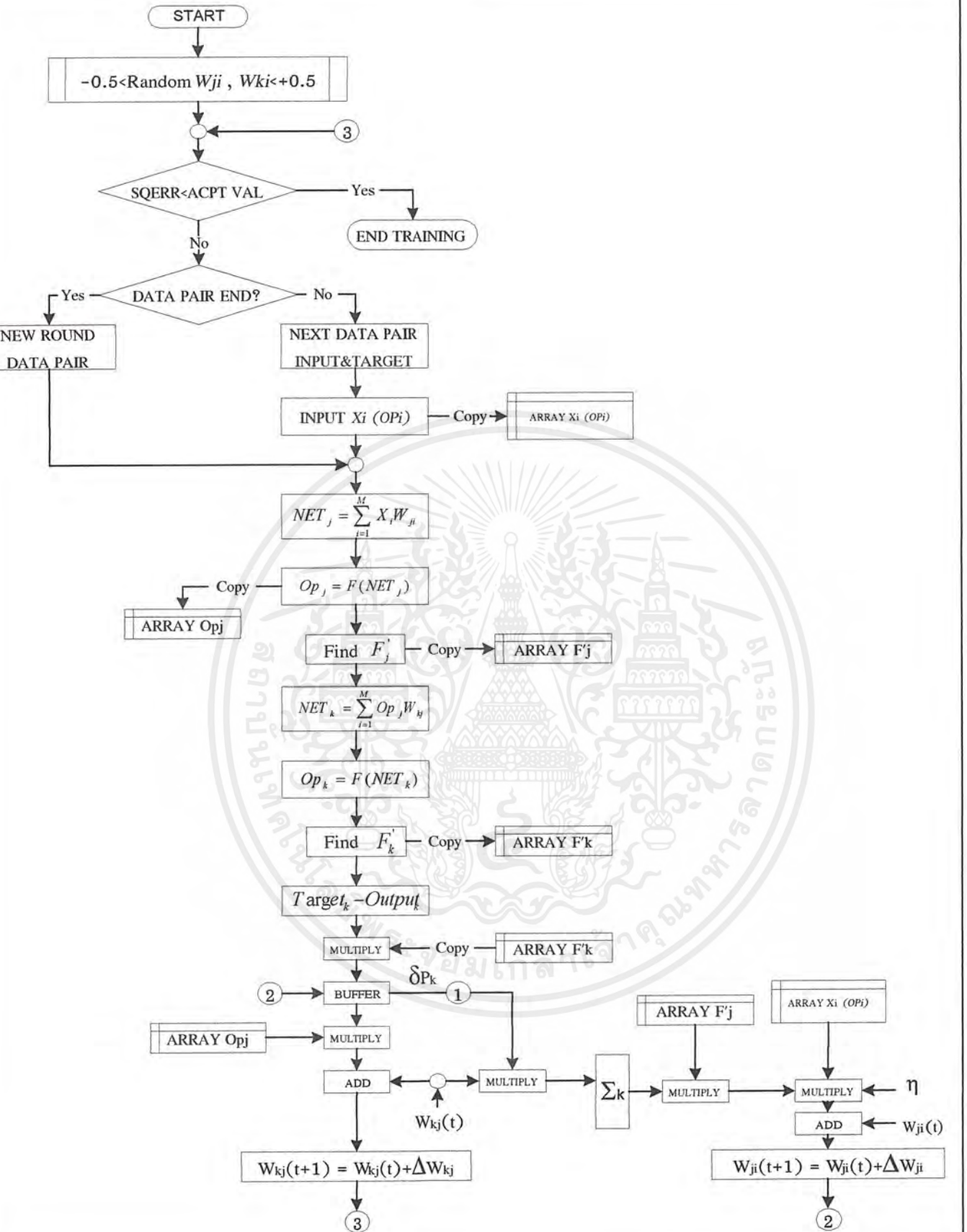
การปรับค่าน้ำหนักจะปรับที่ละเลเยอร์จาก Output layer กลับไปยัง Input layer เมื่อเสร็จสิ้นก็จะกลับสู่ส่วนของ Forward pass อีกครั้ง เพื่อหา Output มาเปรียบเทียบกับ Target และดำเนินการปรับค่าน้ำหนักในส่วนของ Reverse pass เช่นนี้สลับกันไปจนกว่าค่า Error ที่ได้จะลดลงต่ำกว่าค่าที่ต้องการจึงหยุดกระบวนการ ค่าน้ำหนักที่ได้จะเป็นค่าที่เหมาะสมสำหรับทุกคู่ของ Input และ Target ซึ่งพร้อมที่จะนำไปใช้ในระบบการตรวจสอบ การปรับค่าน้ำหนักเทรนนิ่งของ Backpropagation Networks พออธิบายได้ด้วยไดอะแกรมดังนี้



รูปที่ 2.21 ไดอะแกรมแสดงวิธีการหาค่าน้ำหนักที่เหมาะสมของการเทรนนิ่ง โดยใช้อัลกอริทึมแบบแบคพรอบพาคั่น

และพอเขียนเป็น Flow Chart ได้ดังภาพที่ 2.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 แสดง Flow Chart แสดงกระบวนการเทรนนิ่ง เพื่อหาค่าน้ำหนักที่เหมาะสมในกระบวนการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.1.4 เงื่อนไขการหยุดการฝึกสอน

การกำหนด เงื่อนไขการหยุดฝึกสอนทำได้สองกรณีคือ

1. เมื่อผลรวมของค่าผิดพลาดเฉลี่ยระหว่างเอาต์พุตกับเป้าหมายทั้งหมด (SSQERR) ลดลงน้อยกว่าค่าที่กำหนด โดยกำหนดสมการหาค่าผิดพลาดเฉลี่ยของเอาต์พุตทั้งหมดดังนี้

$$SSQERR = \frac{1}{2} \sum (Target - Output)^2 \quad (2.30)$$

SSQERR คือ ค่า Sum Square error ที่ได้จาก Output neurons เทียบกับ Target ใช้สำหรับบ่งชี้ผลของการฝึกสอนว่าเพียงพอกับการใช้งาน หรือเงื่อนไขที่กำหนดหรือไม่

2. จำนวนรอบของการฝึกสอนดำเนินไปถึงค่าที่ตั้งไว้ ซึ่งจำนวนรอบที่เหมาะสมได้จากการทดลอง (ในการทดลองจำนวนรอบที่เหมาะสมอยู่ในช่วง 2,500-5,000 รอบ)[1]

2.11 สเต็ปป์มอเตอร์

สเต็ปเปอร์มอเตอร์ เป็นมอเตอร์ชนิดพิเศษ ที่มีการหมุนของแกนเป็นแบบสเต็ป โดยแต่ละ สเต็ปของการหมุนจะถูกกำหนดเป็นค่ามุมคงที่ เช่นมอเตอร์มีค่าคงที่ในการหมุนสเต็ปละ 1.8 องศา การหมุนให้ได้ครบ 1 รอบ หรือ 360 องศาจึงต้องทำการหมุนเป็นจำนวน 200 สเต็ป เป็นต้น สเต็ปเปอร์มอเตอร์สามารถแบ่งออกได้เป็น 2 ชนิดคือ

1. แบบไบโพลาร์ (bipolar motors) ภายในมอเตอร์จะมีขดลวดภายในจำนวน 2 ขด เท่านั้น การควบคุมให้มอเตอร์หมุนเปลี่ยนทิศทางทำได้โดยการเปลี่ยนทิศทางกระแสที่ป้อนให้กับขดลวดตามจังหวะที่เหมาะสม มอเตอร์ประเภทนี้จะมีสายต่อใช้งานเพียง 4 สาย แต่สำหรับโครงการนี้ไม่สามารถต่อใช้งานกับมอเตอร์ชนิดนี้ได้

2. แบบยูนิโพลาร์ (unipolar motors) มอเตอร์ประเภทนี้จะมีการแท็ป (tap) ที่จุดกลางของขดลวดภายในทั้ง 2 ขด ทำให้สามารถแบ่งขดลวดภายในออกได้เป็น 4 ขด สายเชื่อมต่อภายนอกอาจมีได้ตั้งแต่ 5, 6 หรือ 8 สาย อย่างไรก็ตาม ในการใช้งานจริงสำหรับโครงการนี้จะต้องนำสายที่เป็นแท็ปกลางของขดลวดมาลัดวงจรถึงกันเสมอ แล้วต่อเข้ากับแหล่งจ่ายไฟบวก

การควบคุมให้สเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์เกิดการหมุน จะต้องทำการป้อนกระแสไฟเข้าสู่ขดลวดตามลำดับที่เหมาะสม ซึ่งโดยทั่วไปแบ่งออกได้เป็น 3 วิธีคือแบบ 2 เฟส, แบบเวฟ และแบบครึ่งสเต็ป โดยจะต่อเนื่องกันและเป็นรูปแบบซ้ำๆ กันในแต่ละรอบตามลำดับ ในตารางที่ 2.1-2.3 ส่วนทิศทางหมุนจะขึ้นอยู่กับลักษณะการเปลี่ยนของลำดับที่ป้อนให้กับมอเตอร์

- แบบ 2 เฟส (Two phase sequence) การขับสเต็ปเปอร์มอเตอร์แบบนี้ จะมีการป้อนกระแสไฟเข้าสู่ขดลวดจำนวน 2 ขด ในขณะเดียวกัน ทำให้การขับเคลื่อนวิธีนี้ สามารถให้แรงบิด (torque) ได้สูง ดังแสดงลักษณะการทำงานในตารางที่ 2.1

- แบบเวฟ (Wave sequence) การขับมอเตอร์แบบนี้ จะมีการป้อนกระแสไฟเข้าสู่ขดลวดเพียง 1 ขด ในเวลาใดเวลาหนึ่งเท่านั้น วิธีนี้จะทำให้การหมุนเป็นไปอย่างราบรื่นกว่าแบบ 2 เฟส แต่ก็สามารถให้แรงบิดได้ต่ำกว่า ดังแสดงลักษณะการทำงานในตารางที่ 2.2

- แบบครึ่งสเต็ป (Half step sequence) เป็นวิธีการขับมอเตอร์ที่ผสมระหว่างแบบปกติ และแบบเวฟ โดยวิธีนี้จะทำให้มอเตอร์ทำการหมุนได้ครึ่งละครึ่งสเต็ป จากค่ามุมต่อสเต็ปตามปกติ เช่นจากเดิมหมุนสเต็ปละ 3.6 องศา (100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็ปต่อรอบ) จะเหลือเพียงหมุนสเต็ปละ 1.8 องศา (200 สเต็ปต่อรอบ) แม้ว่าการขับเคลื่อนจะมีข้อดีตรงที่สามารถเพิ่มความละเอียดในการหมุนได้ แต่แรงบิดที่ได้ก็มีค่าไม่สูงมากนัก เนื่องจากในบางช่วงอาจมีการป้อนกระแสไฟเข้าสู่มอเตอร์เพียง 1 ขดเท่านั้น ดังแสดงลักษณะการทำงานในตารางที่ 2.3

นอกจากวิธีการในการขับเคลื่อนสเต็ปมอเตอร์แล้ว อัตราการสเต็ป (step rates) ก็ถือได้ว่าเป็นเรื่องที่สำคัญเช่นกัน เนื่องจากสเต็ปมอเตอร์เป็นอุปกรณ์ทางกลชนิดหนึ่ง การใช้อัตราการสเต็ปที่เร็วเกินไป โดยการป้อนกระแสไฟกระตุ้นในสเต็ปต่อไปเร็วเกินไป ก่อนที่การเคลื่อนที่ในสเต็ปแรกยังไม่สมบูรณ์ อาจทำให้มอเตอร์ทำงานผิดพลาดได้ เช่น ไม่เกิดการหมุนขึ้นเลย, เกิดการสั่นแทนที่จะหมุน, หมุนแต่เอาแน่เอานอนไม่ได้กลับไปกลับมา หรือหมุนผิดทิศทาง เป็นต้น

ตารางที่ 2.1 แสดงการขับเคลื่อนสเต็ปมอเตอร์ แบบ 2 เฟส

สเต็ปที่	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

ตารางที่ 2.2 แสดงการขับเคลื่อนสเต็ปมอเตอร์ แบบเวฟ

สเต็ปที่	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

ตารางที่ 2.3 แสดงการขับเคลื่อนสเต็ปมอเตอร์ แบบครึ่งสเต็ป

สเต็ปที่	A	B	C	D
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

การทำงานของปริณิษณานิพนธ์นี้ประกอบด้วยด้วย 3 ส่วนหลัก คือ

- ส่วนปรีโพรเซสซิง (Pre-Processing)
- ส่วนการตัดลีนใจ
- ส่วนควบคุมมอดเตอร์

โปรแกรมทั้งหมดเขียนด้วยภาษาซี บนโปรแกรม Borland C++ Builder version 3.0

3.1 ส่วนปรีโพรเซสซิง (Pre-Processing)

- ส่วนของการอัดเสียง

ส่วนนี้จะทำหน้าที่รับเสียงเข้ามาจากไมโครโฟน ผ่านทางการ์ดเสียงโดยใช้คอมโปเนนท์ TMediaPlayer ของ C++ Builder เป็นตัวควบคุมการอัดเสียง (ข้อดีของการใช้คอมโปเนนท์เป็นตัวอัดเสียงก็คือสามารถใช้กับการ์ดเสียงรุ่นใดก็ได้ไม่ขึ้นกับฮาร์ดแวร์ และไดร์เวอร์) เมื่อกดปุ่ม Record โปรแกรมจะเริ่มอัดเสียงเป็นเวลา 1 วินาที แล้วจะแสดงรูปสัญญาณเสียงที่อัดเข้าไปในโดเมนเวลา เพื่อให้สามารถตรวจสอบได้ว่าเสียงของคำที่อัดเข้าไปอยู่ในช่วง 1 วินาทีหรือไม่ ถ้ารูปสัญญาณเสียงยังไม่ดี สามารถอัดซ้ำได้โดยกดปุ่ม Record อีกครั้ง ก็จะเริ่มอัดเสียงใหม่ เสียงที่อัดมาจะถูกเก็บในรูปแบบของเวฟไฟล์ (*.wav) ด้วยความถี่สุ่ม 8000 Hz ขนาด 8 บิต แบบ Unsigned char

- ส่วนของการตัดเสียง

เสียงที่เข้ามาจะมีจุดเริ่มต้นของเสียงแต่ละเสียงไม่เท่ากัน เมื่อนำไปแปลง FFT จะทำให้ได้สเปกตรัมในตำแหน่งที่ไม่แน่นอน เป็นผลให้การตัดลีนใจผิดพลาด และข้อมูลเสียงที่สุ่มด้วยความถี่ 8000 Hz ในเวลา 1 วินาที จะมีข้อมูลจำนวน 8000 ตัว ซึ่งมากเกินไปที่จะนำไปป้อนให้นิวรัลเน็ตเวิร์ค ดังนั้นจึงต้องมีการตัดเสียงให้จุดเริ่มต้นของเสียงอยู่ในตำแหน่งเดียวกันและตัดข้อมูลที่เป็นเสียงเงียบออกให้เหลือข้อมูลทั้งหมด 4096 ตัว (2^{12}) เพื่อให้เหมาะสมที่จะนำไปแปลง FFT โดยขั้นตอนการตัดเสียงมีดังนี้

1. นำข้อมูลเสียงที่เก็บอยู่ในรูป Unsigned Char ซึ่งจะมีค่าอยู่ระหว่าง ถึง 255 โดยค่ากลางคือ 128 มาหาค่าสัมบูรณ์ของผลต่างระหว่าง ค่ากลางกับค่าของข้อมูลเสียง ซึ่งเขียนเป็นสมการได้ดังนี้

$$\text{DataAbs} = |128 - \text{Data}| \quad (3.1)$$

ผลลัพธ์ที่ได้ เปรียบเสมือนกับการนำเอาสัญญาณเสียงมาพับครึ่งตามแกน x ค่าที่ได้จะอยู่ในช่วง 0 - 128 ถ้าค่าเท่ากับ 0 แสดงว่าขณะนั้น ไม่มีสัญญาณเสียง และถ้าค่าเท่ากับ 128 แสดงว่าขณะนั้นสัญญาณเสียงมีขนาดสูงที่สุด

2. นำข้อมูลที่ได้ออกมาหาค่าเฉลี่ยทีละ 300 จุดข้อมูล

3. ตรวจสอบจุดเริ่มต้นของข้อมูล ซึ่งจะใช้จุดตัดที่ 3% ของขนาดสูงสุด คือ จะสุ่มตรวจจากจุดเริ่มต้นไปเรื่อย ๆ โดยแต่ละจุดจะห่างกัน 300 จุดข้อมูล ถ้าตรวจพบว่าข้อมูลมีขนาดมากกว่า 4 ติดกัน 3 ครั้ง จะถือว่าจุดนั้น เป็นจุดเริ่มต้นของเสียง การตัดเสียงจะเริ่มจากจุดก่อนที่จะพบเสียง 300 จุดข้อมูล ไปจนครบจำนวน 4096 ตัว ข้อมูลจะถูกเก็บในรูปแบบ Unsigned char เหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนการแปลง FFT

การแปลง FFT จะใช้คอมโปเนนท์ Fourier ของ C++ Builder ซึ่งเป็นแชร์แวร์ ดาวน์โหลดมาจาก www.lohninger.com โดยจะนำข้อมูลมาแปลงบล็อกละ 128 จุดข้อมูล เนื่องจากสัญญาณเสียงมีสัญญาณรบกวนมปนอยู่มาก จึงต้องใช้ เทคนิคเพื่อปรับเส้นสเปกตรัม ให้เรียบและถูกต้องมากขึ้นโดยการนำสเปกตรัมที่ได้จากบล็อกที่อยู่ติด ๆ กันมา "เฉลี่ย" กัน สมมติว่าใช้ K บล็อกมาทำการเฉลี่ย จะได้สเปกตรัมที่เกิดจากการเฉลี่ยคือ

$$P_{avg}(f) = \frac{1}{K} \sum_{i=1}^K P_{xi}(f) \quad (3.2)$$

สัญญาณแต่ละบล็อกจะมีส่วนเหลื่อมกันเพื่อเพิ่มประสิทธิภาพในการติดตามสเปกตรัมที่เปลี่ยนแปลงในสัญญาณ โดยมีส่วนที่เหลื่อมกัน 64 จุด และใช้สเปกตรัมที่ได้จาก 4 บล็อกมาเฉลี่ยกัน (K=4) เทคนิคที่ใช้นี้เป็นเทคนิคที่ใช้ประมาณค่าสเปกตรัมของกำลังที่เรียกว่า Averaging Modified Periodogram หรือวิธีของ Welch (การประมวลผลสัญญาณดิจิทัลเบื้องต้น: หน้า 81) เมื่อแปลง FFT จนหมดข้อมูลจะได้ค่าของสเปกโตรแกรมขนาด 15 x 65 ข้อมูล (3 บล็อกสุดท้ายตัดทิ้ง) จะเห็นว่าข้อมูลมีขนาดมากเกินไป ยังไม่เหมาะที่จะนำไปเทรนให้กับนิวรัลเน็ตเวิร์ค จึงต้องมีการลดจำนวนข้อมูลในแกนความถี่ลง โดยเลือกเอาข้อมูลบางตัวที่มีความถี่เพิ่มขึ้นทีละ 2 เท่าของความถี่ในช่วงก่อน ซึ่งอาจเรียกได้ว่า เป็นความถี่ฮาร์โมนิก ของความถี่ในช่วงก่อน เนื่องจากสเปกตรัมของสัญญาณเสียงพูดจะสังเกตเห็นความแตกต่างได้ง่ายในแกนความถี่ฮาร์โมนิก (ธินา ศรีประโมง 2537 : 199-204) ซึ่งปริยญาทิพนธ์นี้ได้เลือกข้อมูลในแกนความถี่ออกมา 17 ตัว เป็นข้อมูลของความถี่ต่าง ๆ ดังนี้ คือ 187.5, 250, 312.5, 375, 500, 625, 750, 875, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2875 และ 3250 Hz ซึ่งเป็นข้อมูลตัวที่ 3, 4, 6, 8, 10, 12, 14, 16, 18, 20, 24, 28, 32, 36, 40, 46 และ 52 ของข้อมูลในแกนความถี่ 65 ตัว จะทำให้เราได้สเปกตรัมที่มีข้อมูล 15 x 17 ตัว ซึ่งเหมาะที่จะนำไปป้อนให้นิวรัลเน็ตเวิร์คต่อไป

3.2 ส่วนการตัดสินใจ

จะใช้นิวรัลเน็ตเวิร์คแบบแบ็คพรอพพาเกชัน ในการตัดสินใจเสียงพูดที่เข้ามาว่าเป็นเสียงของคำใดใน 10 คำ คือ wave, two, half, ช้าย, ขาว, up, down, go, pause, clear ขึ้นตอนของการฝึกสอน จะใช้เสียงในการฝึกสอนของผู้พูดคนเดียว เสียงละ 50 ครั้ง 10 เสียง รวมเป็น 500 เสียง โดยมีรายละเอียดโมดูลของนิวรัลเน็ตเวิร์คดังนี้

INPUT NEURONS	=	255
HIDDEN NEURONS	=	50
OUTPUT NEURONS	=	10
LEARNING RATE	=	0.1

เทรนนิ่ง 3000 รอบ ใช้เวลา 1 ชั่วโมง จะได้ไฟล์ของค่าน้ำหนักออกมาเพื่อนำไปใช้ในการทดสอบ ในขั้นตอนของการทดสอบ จะรับเสียง จากไมโครโฟนเข้ามาแปลง FFT แล้วป้อนให้นิวรัลเน็ตเวิร์คที่ผ่านการฝึกสอนมาแล้ว จะได้ค่าออกมาจากทั้ง 10 เอาท์พุทของนิวรัลเน็ตเวิร์ค มีค่าอยู่ในช่วง 0-1 ซึ่งเอาท์พุทของค่าที่นิวรัลเน็ตเวิร์ครู้จำได้จะมีค่ามากกว่า เอาท์พุทตัวอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

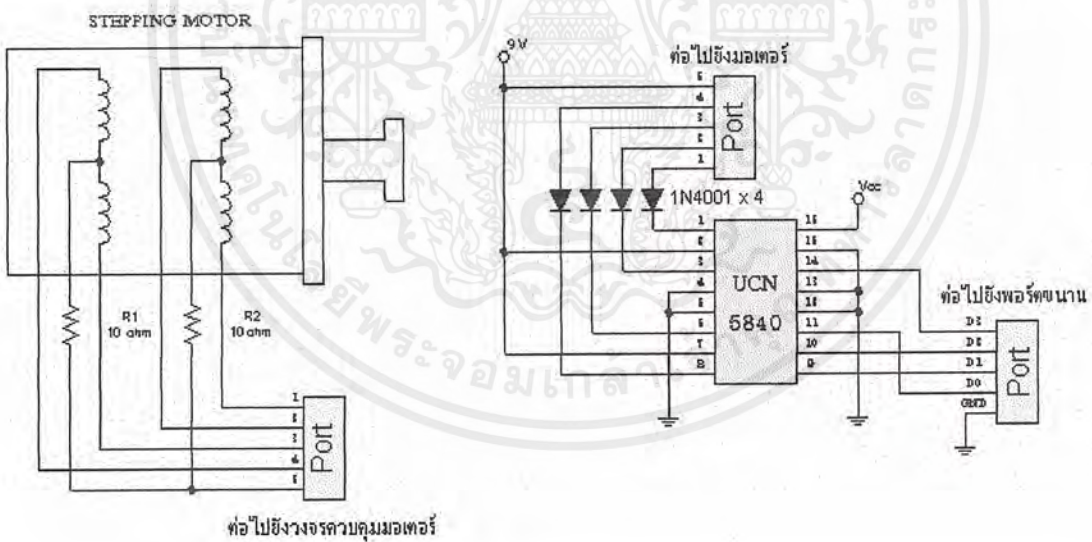
3.3 ส่วนควบคุมมอเตอร์

จะใช้ IC สำเร็จรูปเบอร์ UCN 5804B ในการควบคุม ข้อดีของ IC ตัวนี้ก็คือสามารถควบคุมได้ง่ายโดยการป้อนพัลส์ให้ 1 ลูกจะทำให้มอเตอร์หมุน 1 สเต็ป และสามารถควบคุมแบบการหมุนและทิศทางการหมุนได้ด้วยการให้ลอจิกที่ขาควบคุม

คำสั่งที่ใช้ควบคุมมี 10 คำสั่ง คือ

1. Wave ใช้กำหนดแบบการหมุนให้เป็นแบบ Wave-Drive
2. Two ใช้กำหนดแบบการหมุนให้เป็นแบบ Two-Phase
3. Half ใช้กำหนดแบบการหมุนให้เป็นแบบ Half-Step
4. ช้าย ใช้กำหนดทิศทางการหมุน ให้หมุนทวนเข็มนาฬิกา
5. ขวา ใช้กำหนดทิศทางการหมุน ให้หมุนตามเข็มนาฬิกา
6. Up ใช้เพิ่มความเร็วในการหมุน
7. Down ใช้ลดความเร็วในการหมุน
8. Go ใช้สั่งให้มอเตอร์เริ่มหมุน
9. Pause ใช้สั่งให้มอเตอร์หยุดหมุน
10. Clear ใช้สั่งให้มอเตอร์หมุนกลับไปตำแหน่งเริ่มต้นและกลับไปใช้ค่าเริ่มต้น

สัญญาณที่ใช้ควบคุมมอเตอร์จะต่อมาจากพอร์ตขนาน โดยใช้สายสัญญาณ 4 เส้น การต่อวงจรควบคุมมอเตอร์แสดงดังรูป



รูปที่ 3.1 แสดงวงจรควบคุมมอเตอร์

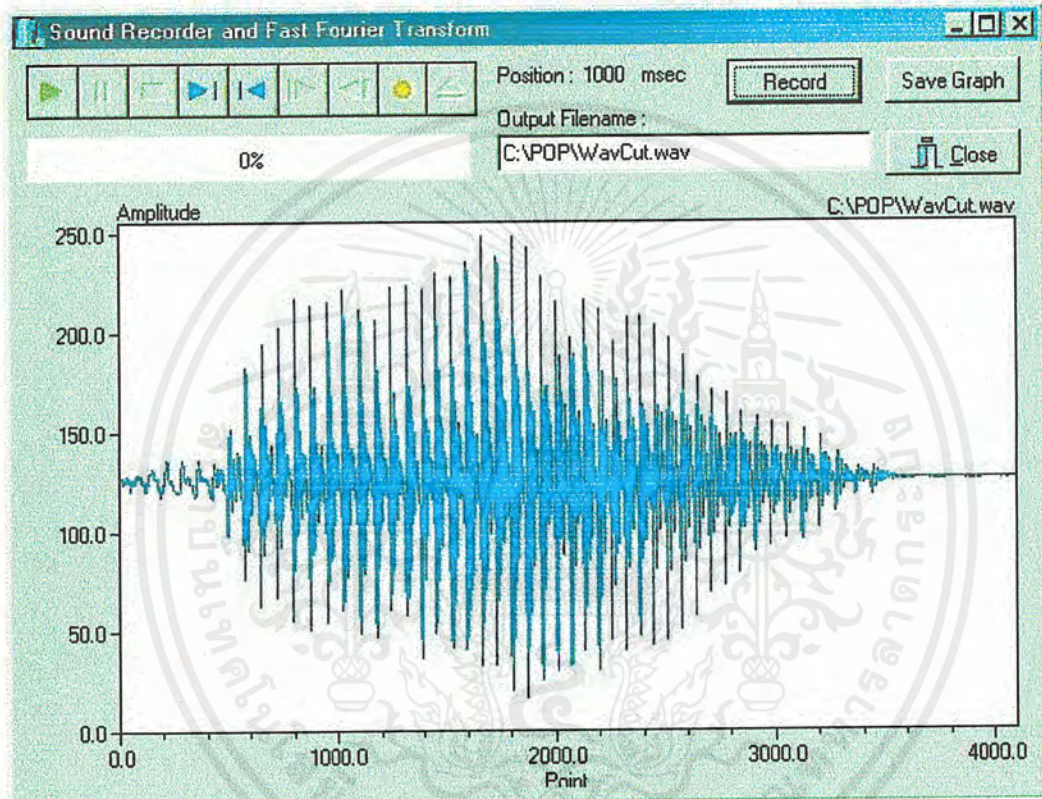
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การดำเนินการและทดลอง

4.1 การดำเนินการ

4.1.1 ขั้นตอนการฝึกสอน

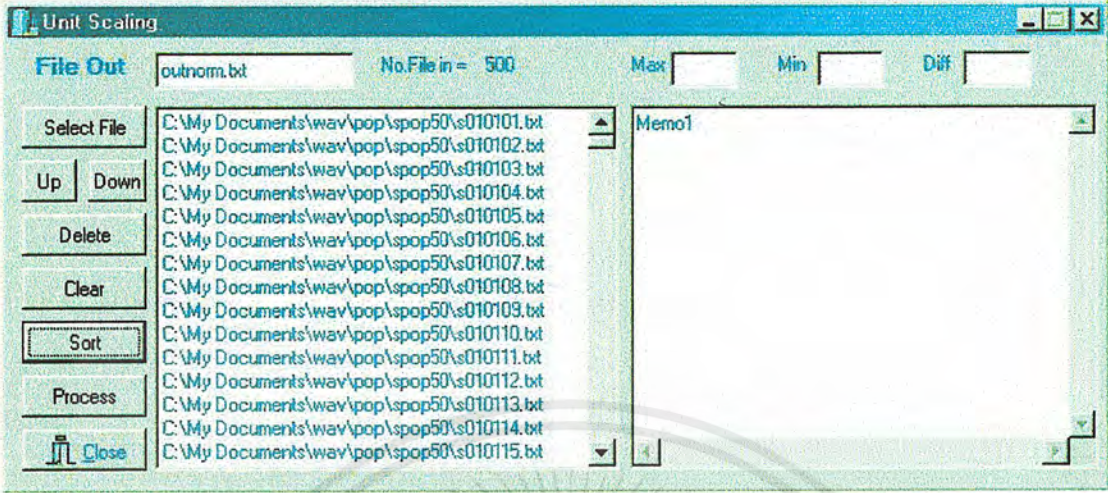


รูปที่ 4.1 แสดงหน้าจอของโปรแกรม RecFFT.EXE

เริ่มจากการอัดเสียงเพื่อนำไปใช้ในการฝึกสอน โดยใช้โปรแกรม RecFFT.EXE วิธีการอัดเสียง ทำได้โดย คลิกที่ปุ่ม Record แล้วรอประมาณ 0.3 วินาที จึงเริ่มแปลงเสียง เพื่อให้เสียงอยู่ในช่วงกลางของไฟล์ เมื่อโปรแกรมรับเสียงเข้าไปแล้วจะแสดงภาพสัญญาณเสียง ให้ดูบนกราฟ การแปลงเสียงควรจะกะเวลาให้พอดีไม่ให้เร็วไปหรือช้าไป เพราะเสียงอาจจะอยู่นอกช่วงเวลาการอัด และควรให้ความดังของเสียงอยู่ในระดับกลาง ๆ ไม่เบาหรือดังเกินไป การแปลงเสียงควรให้เป็นธรรมชาติมากที่สุดและไม่ควรให้มีเสียงลมที่เกิดจากการเปิดริมฝีปากเข้าไปด้วย ก่อนแปลงเสียงควรจะเขยอริมฝีปากเอาไว้เล็กน้อย ถ้าต้องการฟังเสียงที่อัดเข้าไปทำได้โดยคลิกที่ปุ่ม play เมื่อได้เสียงที่พอใจแล้ว ก็เปลี่ยนชื่อไฟล์เพื่ออัดเสียงอื่นต่อไป จนครบ 10 เสียง ๆ ละ 50 ครั้ง การทดลองนี้จะใช้เสียงของคน ๆ เดียวเพื่อความสะดวก ไฟล์ที่ได้จากโปรแกรม RecFFT.EXE จะเป็นข้อมูลสเปกโตรแกรม อยู่ในรูปของตัวเลข Integer จำนวน 255 ตัว ซึ่งจะเก็บอยู่ในไฟล์เท็กซ์ เมื่ออัดเสียงเสร็จแล้ว ก็นำไฟล์เท็กซ์ของเสียงทุกเสียงมาทำการปรับค่าให้มีค่าอยู่ระหว่าง 0-1 โดยใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม UnitScal.EXE ข้อมูลของเสียงทุกเสียงที่ปรับค่าแล้วจะเก็บรวมไว้ในไฟล์เท็กซ์เดียวกันตามลำดับ ไฟล์เท็กซ์นี้จะเป็นไฟล์อินพุตเพื่อใช้ฝึกสอนให้นิวรัลเน็ตเวิร์ค



รูปที่ 4.2 แสดงหน้าจอของโปรแกรม UnitScal.EXE

การจะฝึกสอนให้นิวรัลเน็ตเวิร์ค จะต้องมีการเตรียมข้อมูล โดยจะเก็บไว้ในไฟล์เท็กซ์ ไฟล์นี้จะสร้างโดยใช้โปรแกรม Excel ภายในไฟล์จะประกอบด้วยตัวเลข 0 และ 1 ขนาด $M \times N$ โดย M คือ จำนวนแถว ซึ่งจะเท่ากับจำนวนเสียงที่นำมาเทรน และ N คือจำนวนหลัก ซึ่งจะเท่ากับจำนวนเอาต์พุตของนิวรัลเน็ตเวิร์ค ถ้าเสียงอินพุตเป็นเสียงของคำไหน คำ 1 จะอยู่ที่หลักของคำนั้น ส่วนหลักอื่น ๆ จะเป็น 0 จากการทดลองนี้ คำเป้าหมายจะมีจำนวน 500×10 ตัว

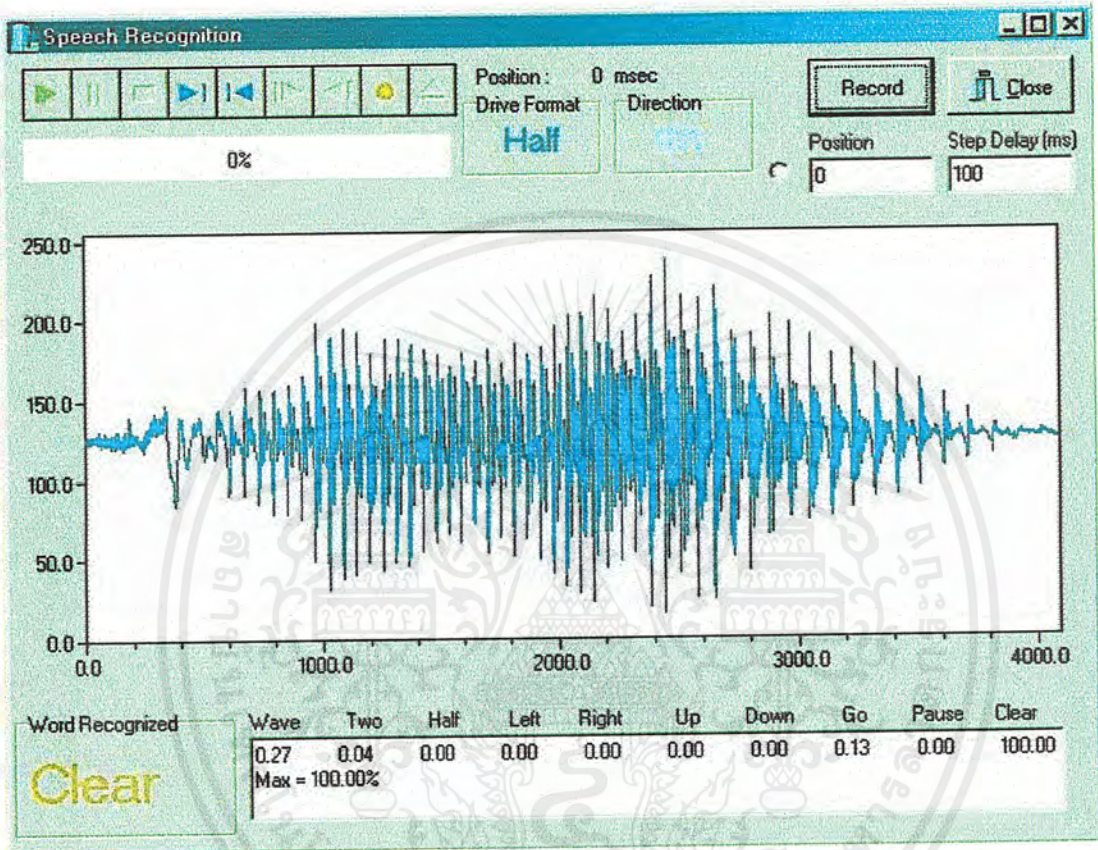


รูปที่ 4.3 แสดงหน้าจอของโปรแกรม Neural.EXE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ไฟล์อินพุตและไฟล์เป้าหมายแล้ว ก็พร้อมที่จะเริ่มทำการฝึกสอนโดยกำหนดรอบของการฝึกสอนเท่ากับ 3000 รอบ จะใช้เวลาประมาณ 1 ชั่วโมง สาเหตุที่กำหนดให้หยุดสอนที่ 3000 รอบ เพราะค่า sum square error จะลดลงอย่างช้า ๆ หลังจาก 1500 รอบ เมื่อฝึกสอนเสร็จแล้วจะได้ไฟล์เอาต์พุต 2 ไฟล์ คือ ไฟล์ค่าน้ำหนัก และไฟล์ค่าผิดพลาด ไฟล์น้ำหนักที่ได้นี้จะเก็บไว้ใช้ในกระบวนการทดสอบต่อไป

4.1.2 ขั้นตอนการทดสอบ



รูปที่ 4.4 แสดงหน้าจอของโปรแกรม SndTest.EXE

ในการทดสอบจะใช้โปรแกรม SndTest.EXE ก่อนการใช้งานจะต้องกำหนดชื่อไฟล์น้ำหนักที่ได้จากการฝึกสอนให้มันก่อน เมื่อต้องการทำการทดสอบก็ทำโดยการคลิกที่ปุ่ม Record จากนั้นก็เปล่งเสียง แบบเดียวกับที่ทำการบันทึกเสียง แล้วโปรแกรมก็จะทำการประมวลผล เสร็จแล้วจะแสดงคำที่รู้จักได้ออกมาและควบคุมมอเตอร์ให้ทำงานตามคำสั่ง

4.1.3 ส่วนควบคุมมอเตอร์

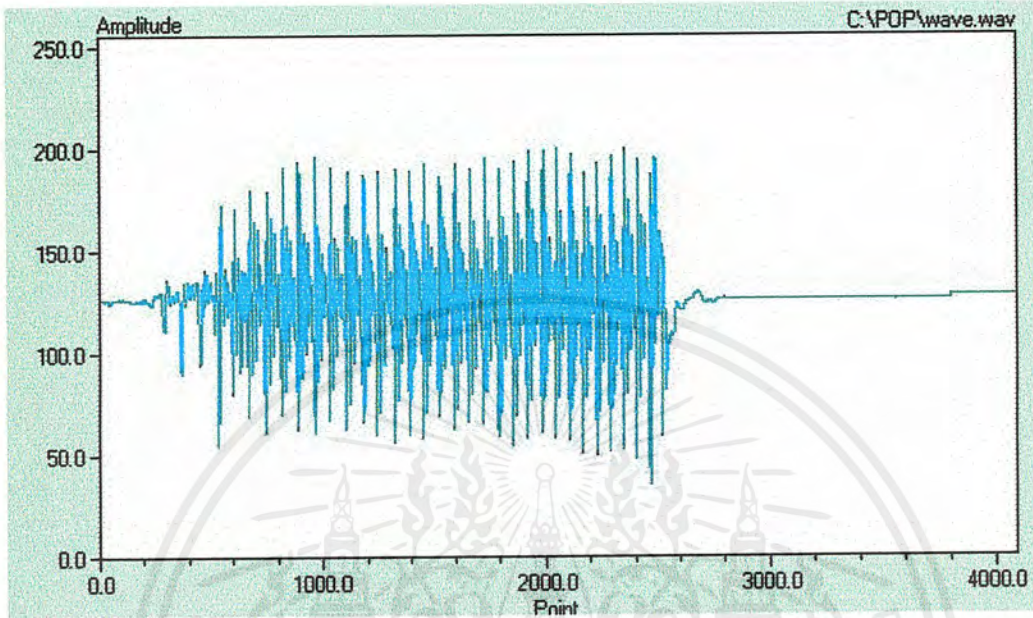
ส่วนควบคุมมอเตอร์จะรับค่าผลลัพธ์จากโปรแกรมนิวรัลเน็ตเวอร์คซึ่งเป็นค่าตัวเลข 0-9 มาควบคุมมอเตอร์ โดยจะกำหนดค่าเริ่มต้นให้มอเตอร์ดังนี้

- รูปแบบการหมุน = half
- ทิศทางการหมุน = ขวา
- ความเร็ว = 0.5 วินาที/step

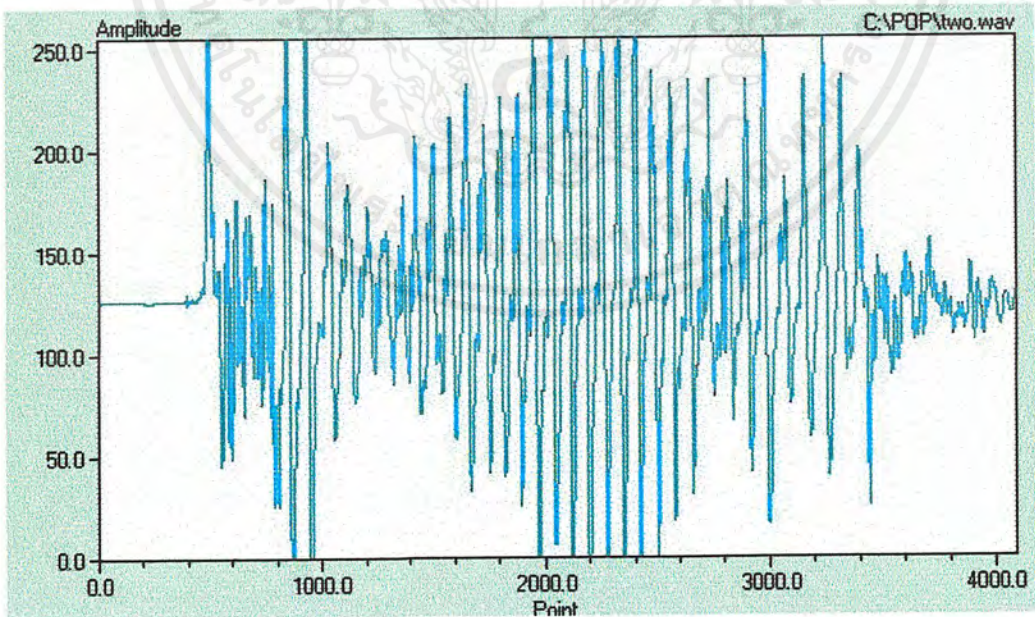
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

4.2.1 ตัวอย่างเสียงที่ใช้ในการทดลองแสดงในโดเมนเวลา

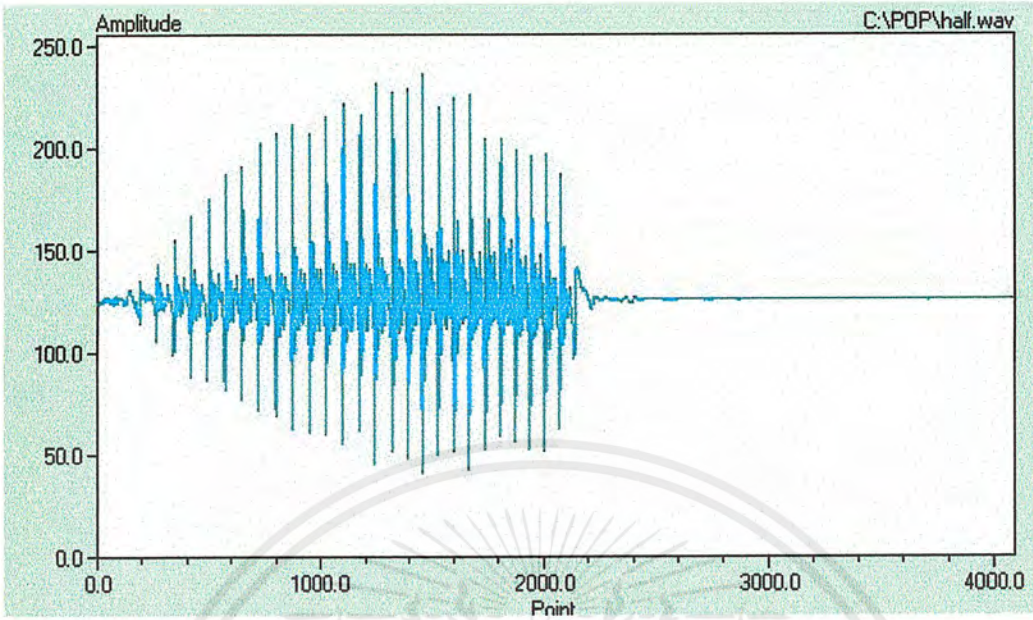


รูปที่ 4.5 เสียง Wave

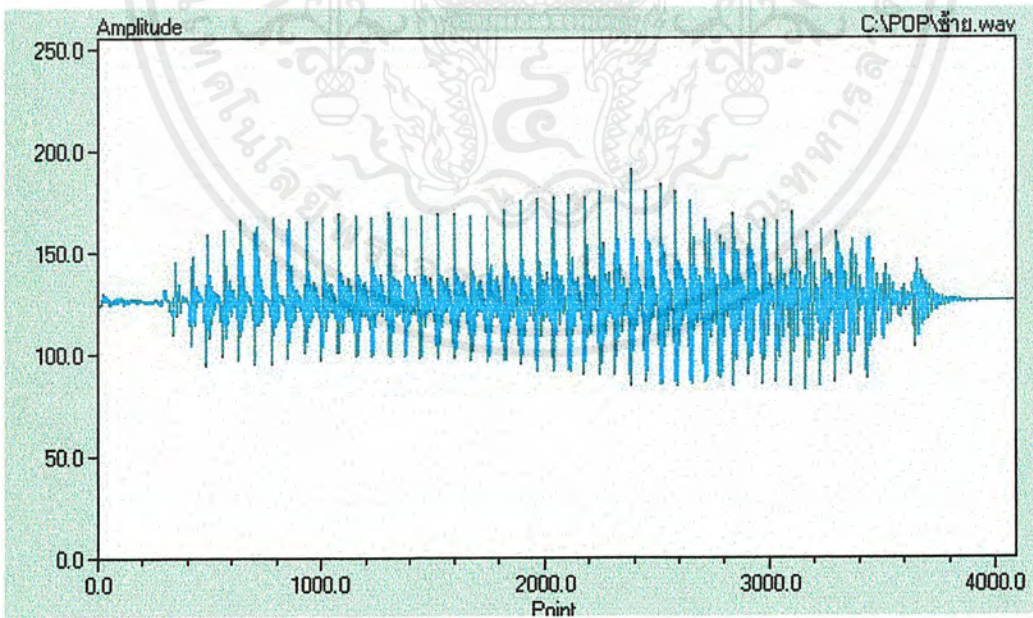


รูปที่ 4.6 เสียง Two

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

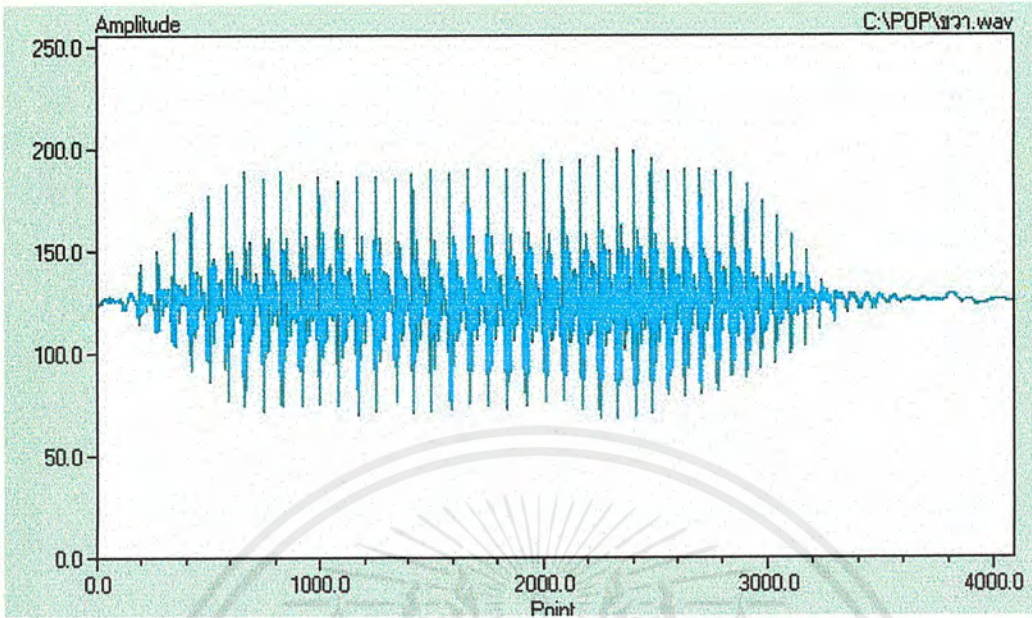


รูปที่ 4.7 เสียง Half

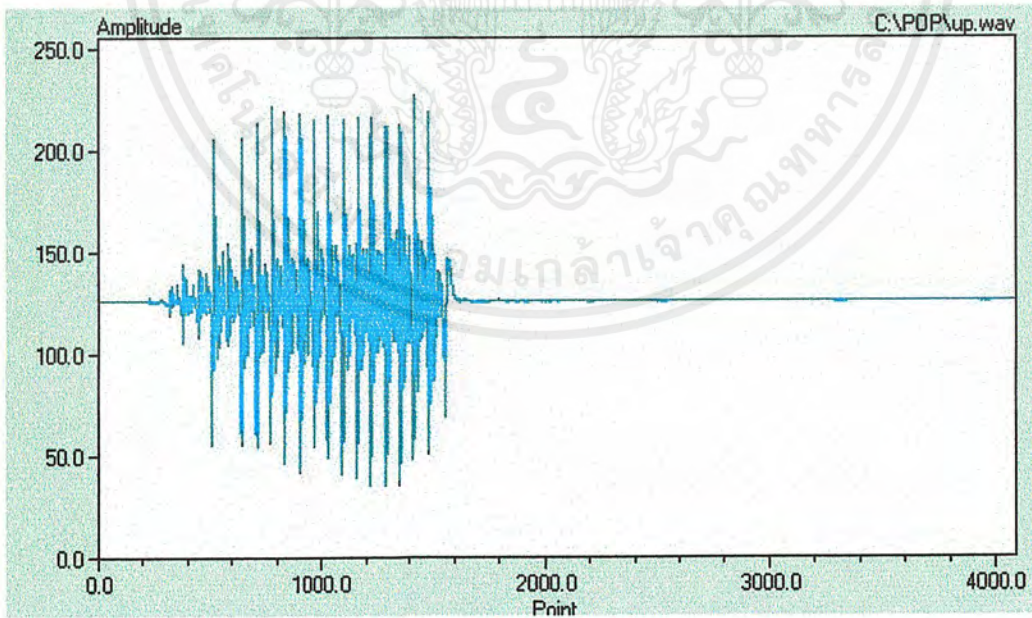


รูปที่ 4.8 เสียง ซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

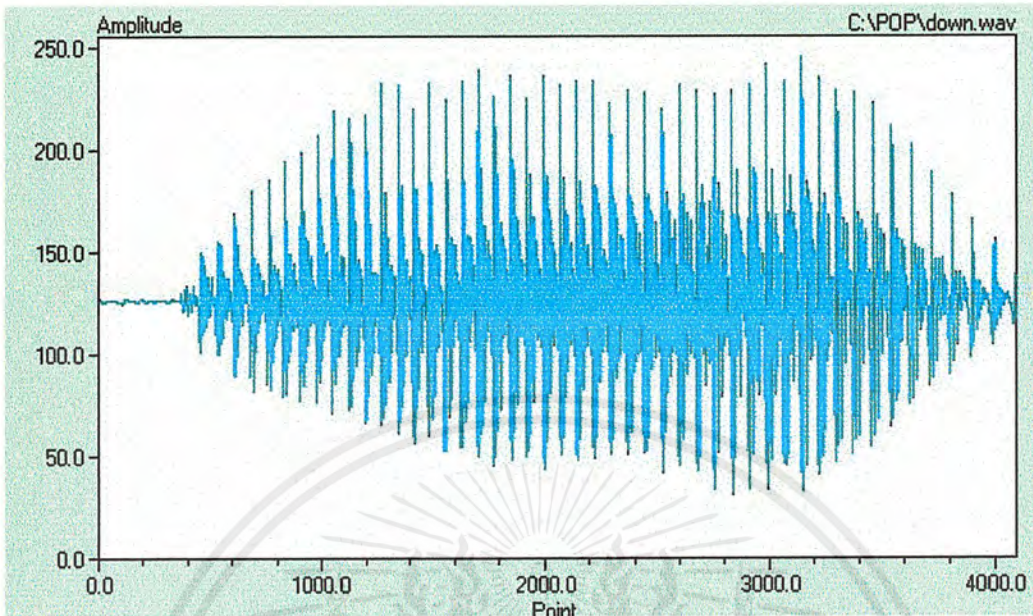


รูปที่ 4.9 เสียง ขวา

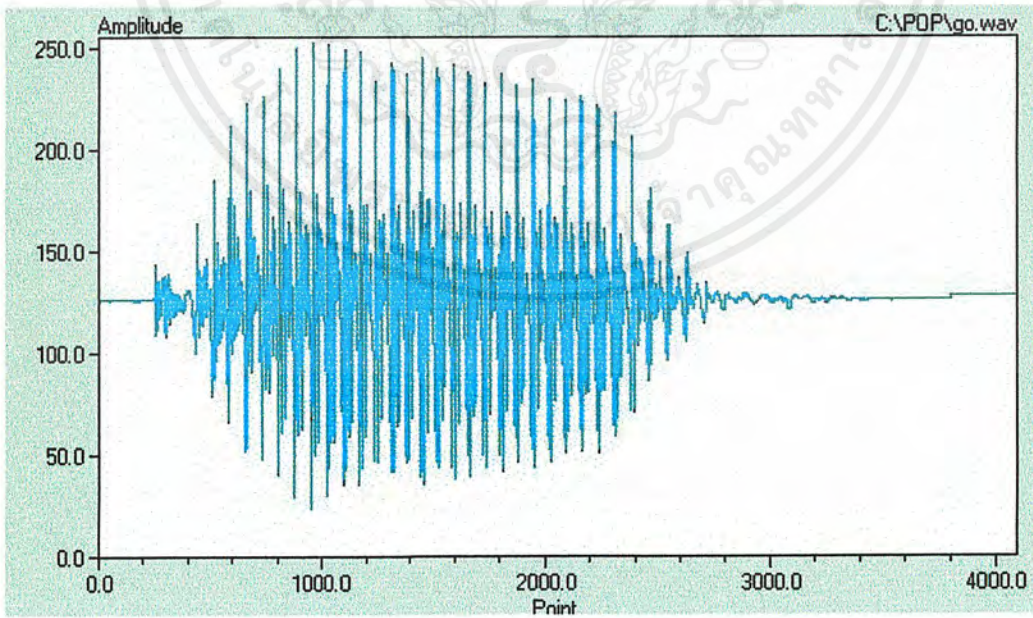


รูปที่ 4.10 เสียง Up

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

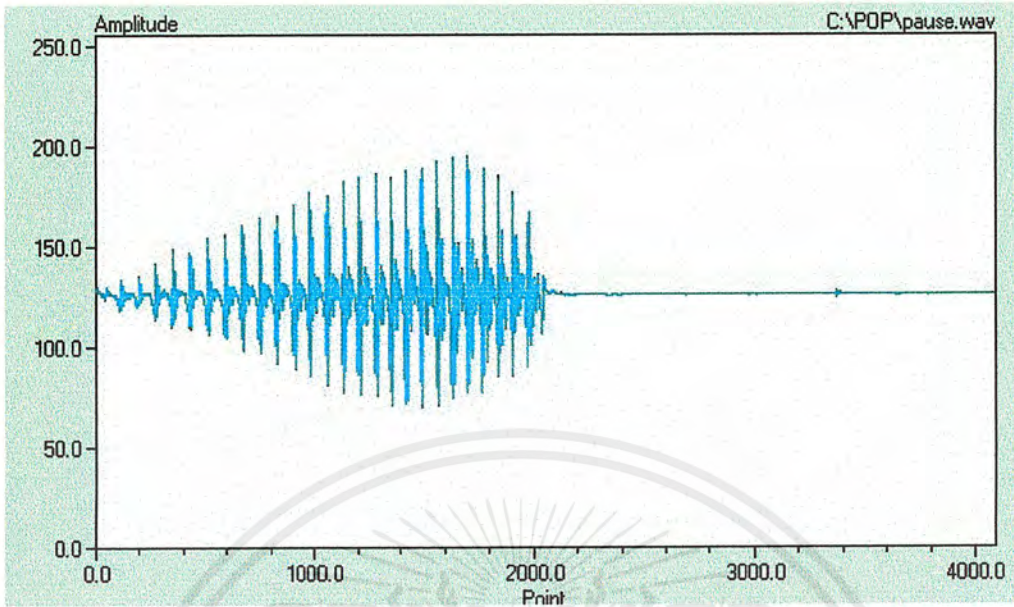


รูปที่ 4.11 เสียง Down

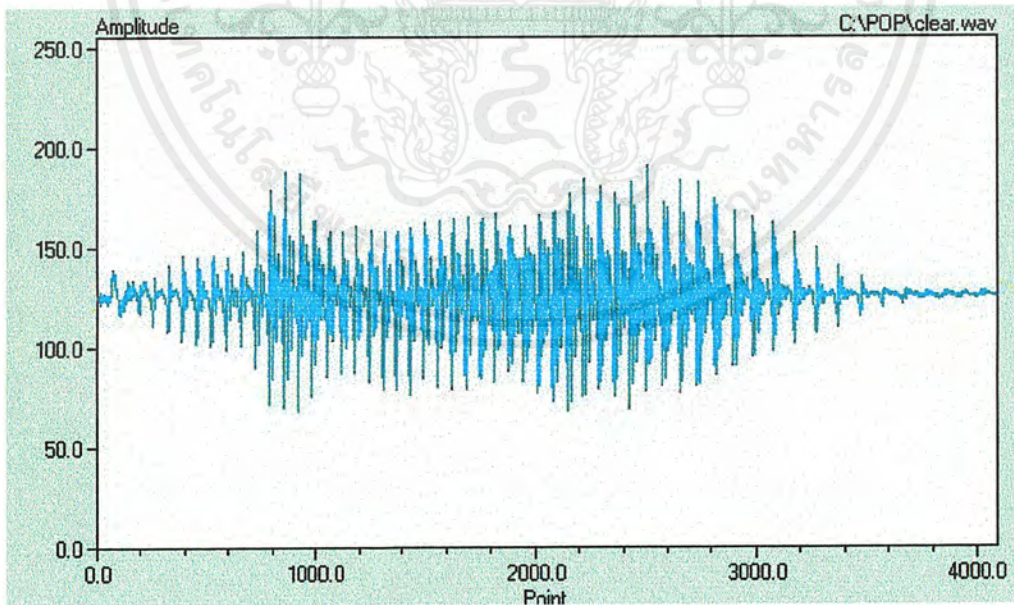


รูปที่ 4.12 เสียง Go

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 เสียง Pause

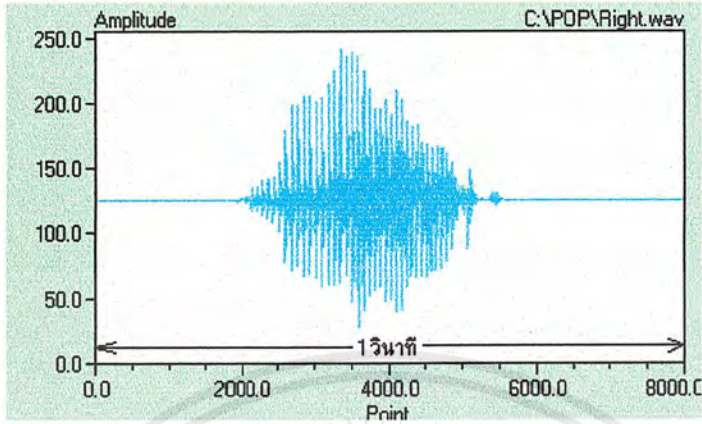


รูปที่ 4.14 เสียง Clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 แสดงขั้นตอนการทดลองส่วน Pre-processing

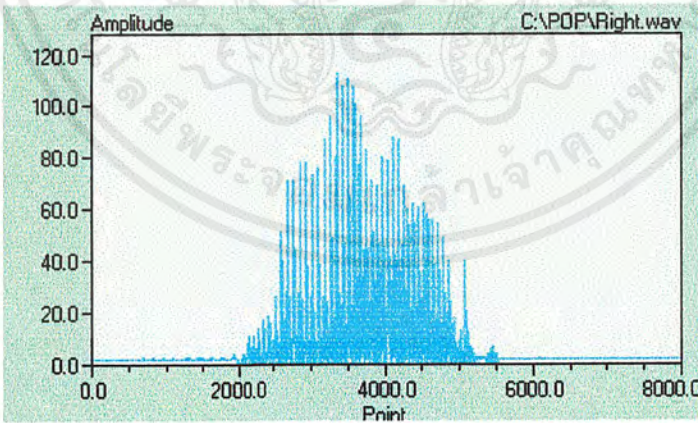
1. อัดเสียงเข้ามา 1 วินาที โดยใช้ Component TMediaPlayer



รูปที่ 4.15 แสดงสัญญาณเสียงที่อัดเข้ามา

2. หาจุดเริ่มต้นของเสียง แล้วตัดข้อมูลให้เหลือ 4096 จุด โดยมีขั้นตอนดังนี้

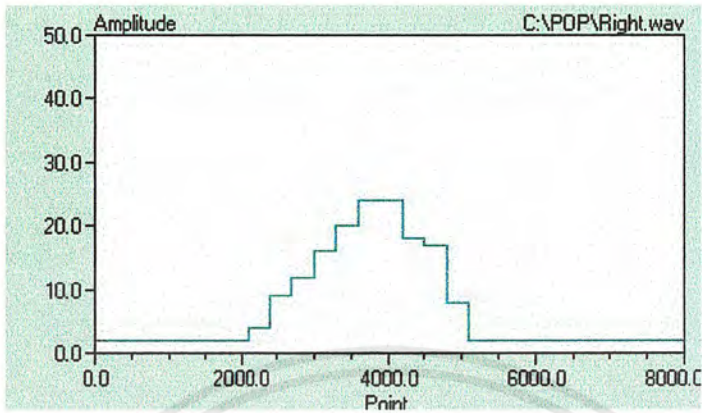
2.1 พลิกลสัญญาณเสียงที่มีค่าต่ำกว่า 128 ขึ้นมาด้านบน



รูปที่ 4.16 แสดงสัญญาณเสียงที่พลิกด้านล่างขึ้นมาด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

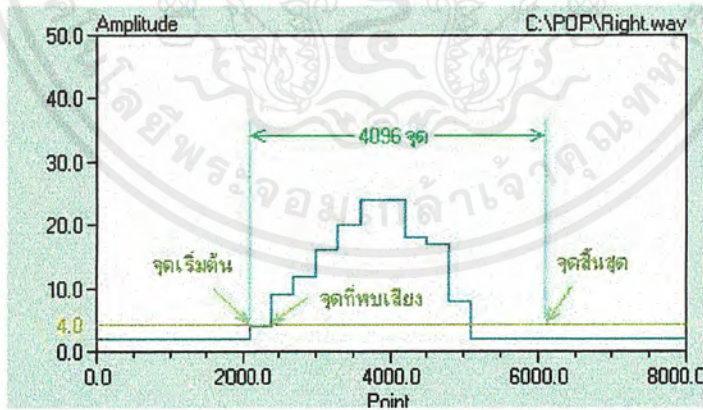
2.2 หาค่าเฉลี่ยของเสียงทีละ 300 จุด



รูปที่ 4.17 แสดงค่าเฉลี่ยของสัญญาณเสียง

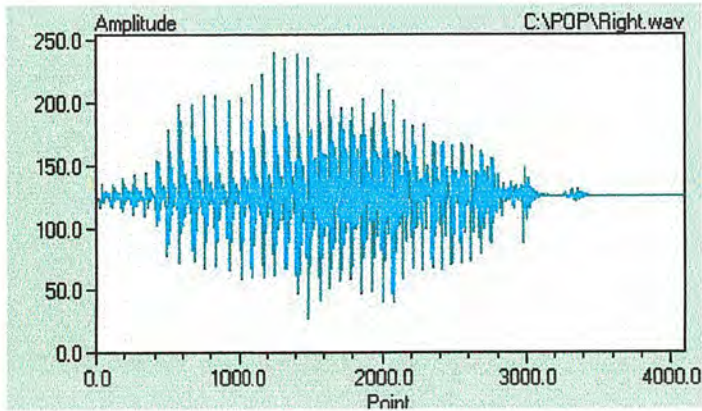
2.3 หากจุดที่สัญญาณมีค่าเฉลี่ยมากกว่า 4 ติดกัน 3 ช่วงการสุ่ม(สุ่มช่วงละ 300 จุด)

2.4 เมื่อพบจุดเริ่มต้นของเสียงแล้ว จะได้ช่วงของข้อมูลเสียงที่จะนำไปแปลง FFT โดยจะมีจุดเริ่มต้นที่จุดก่อนที่จะพบเสียง 300 จุด และจุดสิ้นสุดจะนับจากจุดเริ่มต้นไป 4096 จุด



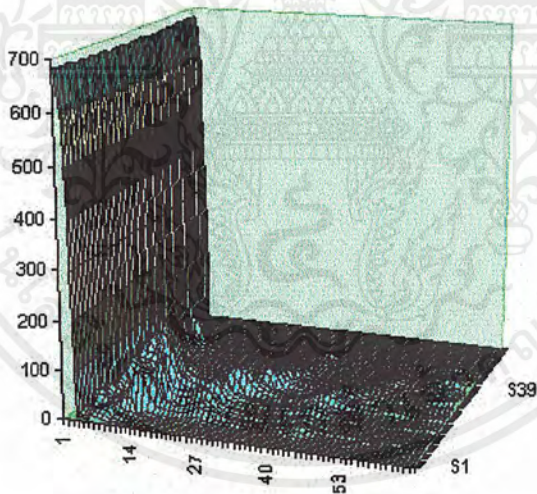
รูปที่ 4.18 แสดงตำแหน่งของจุดตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงสัญญาณเสียงที่ผ่านการตัดแล้ว

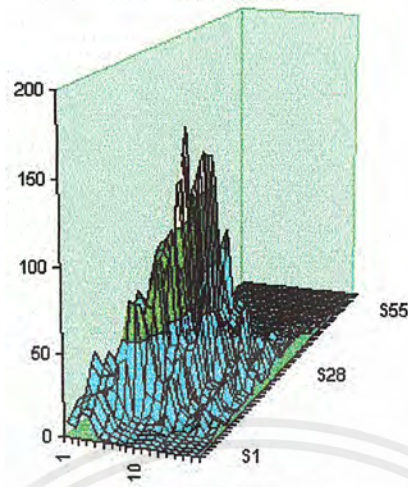
3. นำข้อมูลเสียงไปแปลง FFT โดยแบ่งเป็นบล็อกละ 128 จุด เหลื่อมช่วงละ 64 จุด ใช้ Hamming Window จะได้สเปกโตรแกรม ขนาด 63 ดีกรี x 65 จุด



รูปที่ 4.20 แสดงสเปกโตรแกรมที่ได้จากการแปลง FFT

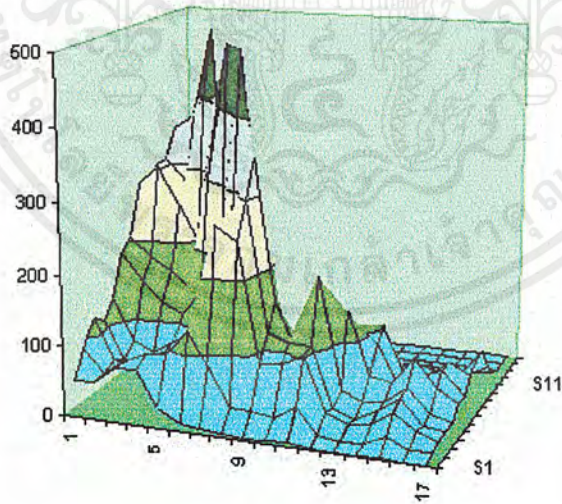
4. เลือกข้อมูลในแกนความถี่มา 17 จุด ทุกๆ ดีกรี โดยเลือกตัวที่ 3, 4, 5, 6, 8, 10, 12, 14, 16, 20, 24, 28, 32, 36, 40, 46 และ 52 จะได้สเปกโตรแกรม ขนาด 63 ดีกรี x 17 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 แสดงสเปกโตรแกรมที่ลดข้อมูลในแกนความถี่แล้ว

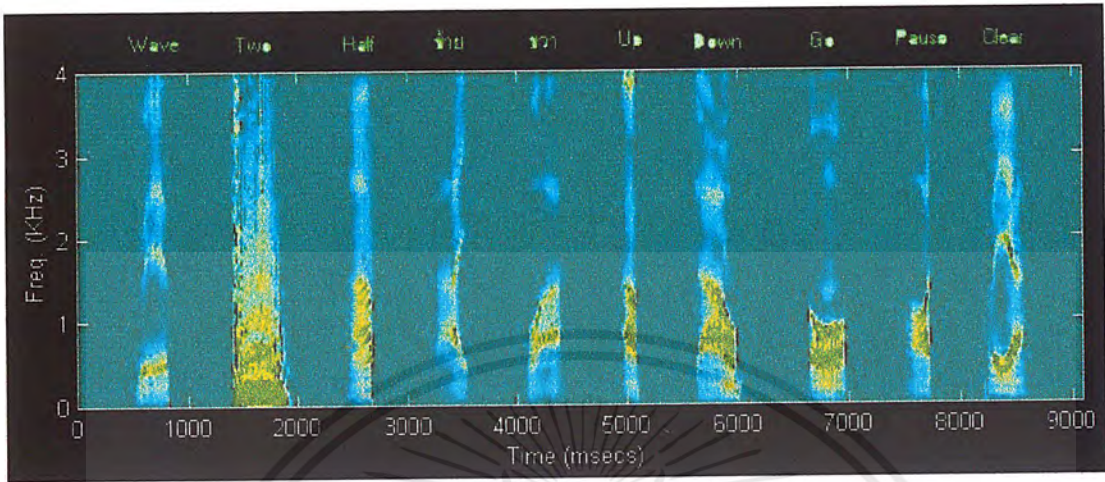
5. ลดข้อมูลของสเปกโตรแกรมในแกนเวลา โดยหาค่าเฉลี่ยทีละ 4 ดิสครีท จะได้สเปกโตรแกรม ขนาด 15 ดิสครีท x 17 จุด (3 ดิสครีทสุดท้ายตัดทิ้ง)



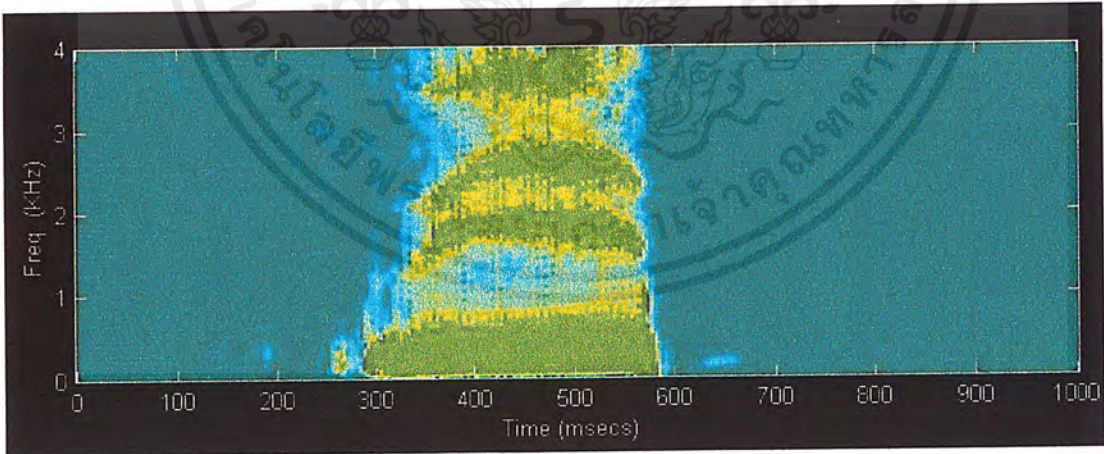
รูปที่ 4.22 แสดงสเปกโตรแกรมที่ลดข้อมูลในแกนเวลาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 รูปแสดงสเปกโตรแกรมของเสียงพูดที่ใช้ในการทดลอง

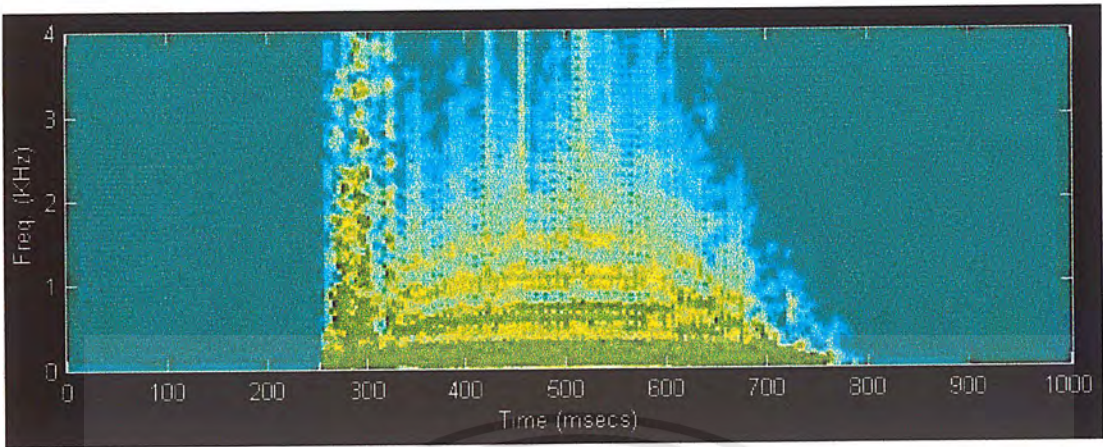


รูปที่ 4.23 แสดงสเปกโตรแกรมของเสียงทั้งหมดที่ใช้ในการทดลอง Plot โดยใช้โปรแกรม MatLab

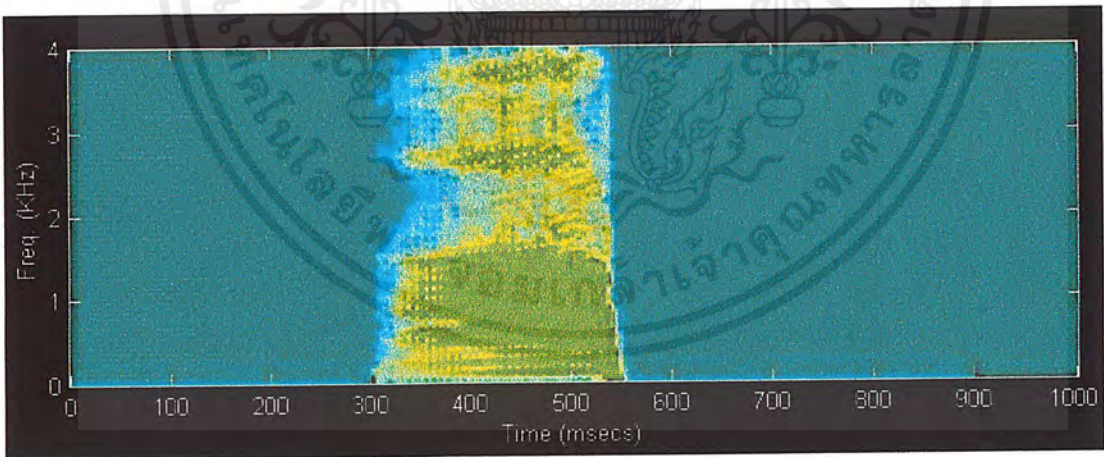


รูปที่ 4.24 แสดงสเปกโตรแกรมของเสียง Wave ที่ได้จากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

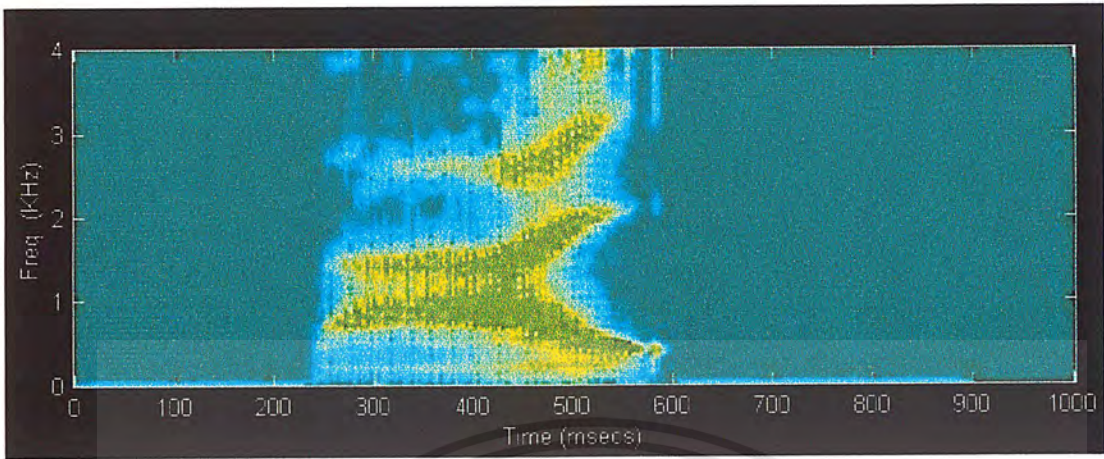


รูปที่ 4.25 แสดงสเปกโตรแกรมของเสียง Two ที่ได้จากการบันทึก

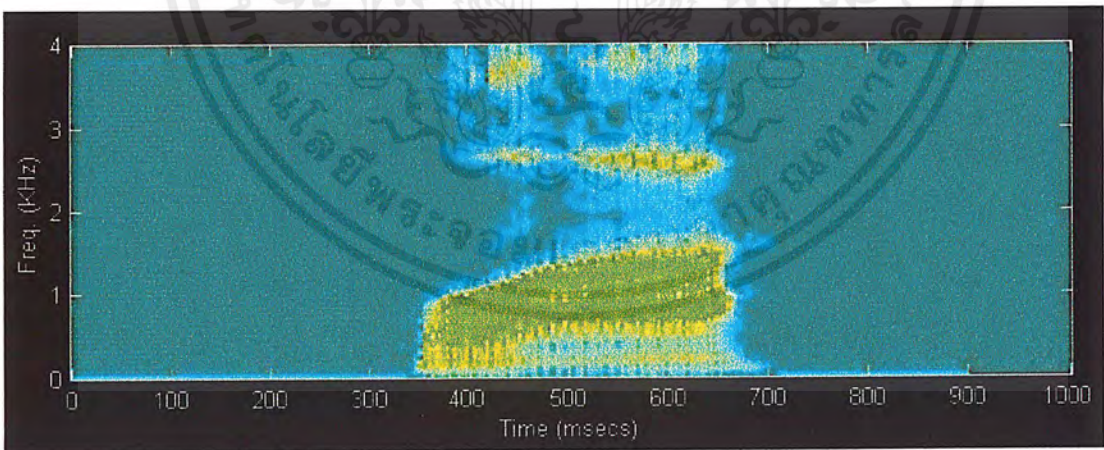


รูปที่ 4.26 แสดงสเปกโตรแกรมของเสียง Half ที่ได้จากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

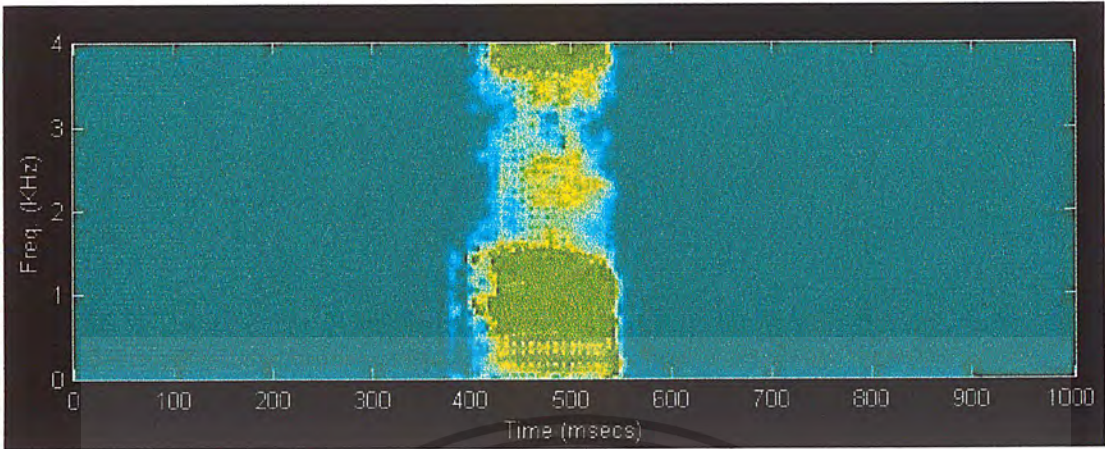


รูปที่ 4.27 แสดงสเปกโตรแกรมของเสียง ช่าย ที่ได้จากการบันทึก

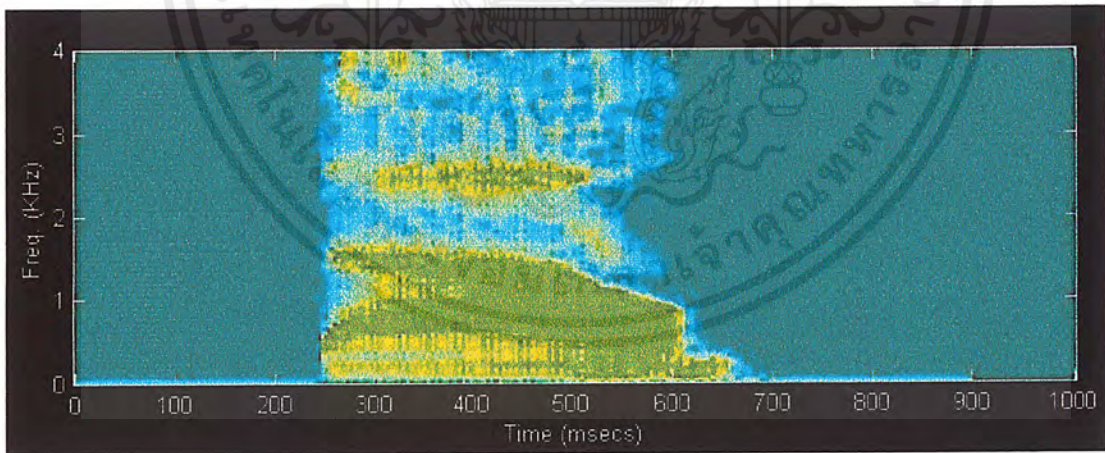


รูปที่ 4.28 แสดงสเปกโตรแกรมของเสียง ขาว ที่ได้จากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

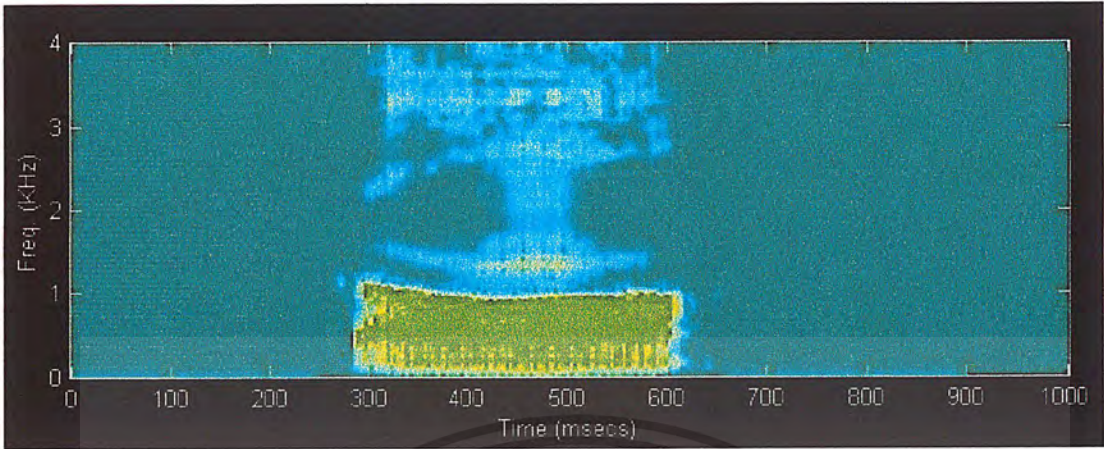


รูปที่ 4.29 แสดงสเปกโตรแกรมของเสียง Up ที่ได้จากการบันทึก

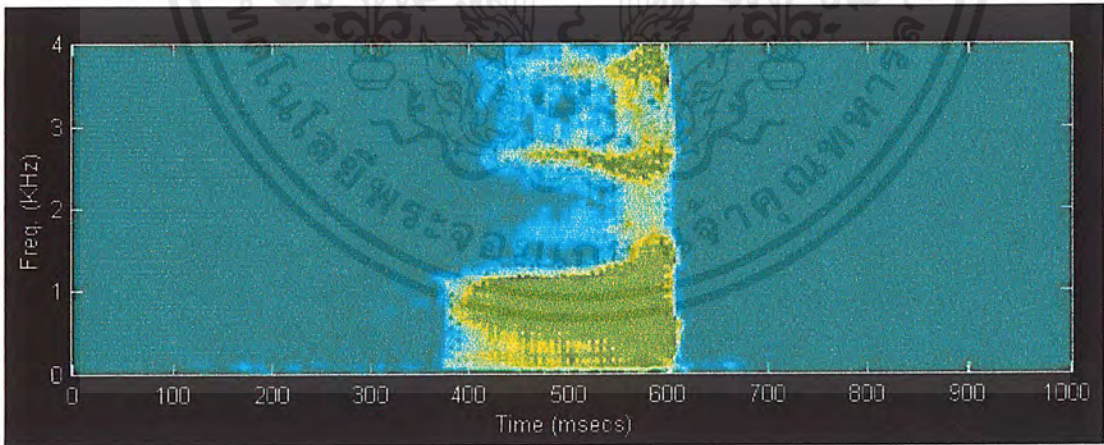


รูปที่ 4.30 แสดงสเปกโตรแกรมของเสียง Down ที่ได้จากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

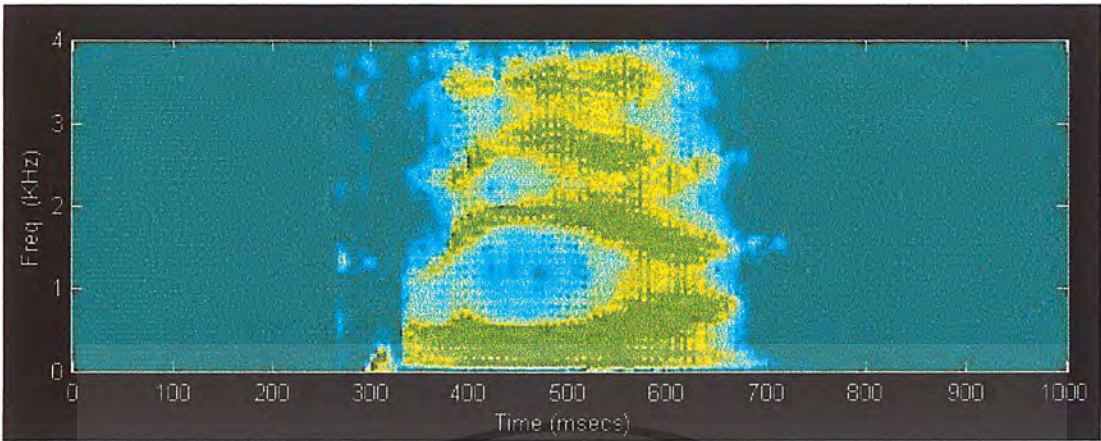


รูปที่ 4.31 แสดงสเปกโตรแกรมของเสียง Go ที่ได้จากการบันทึก

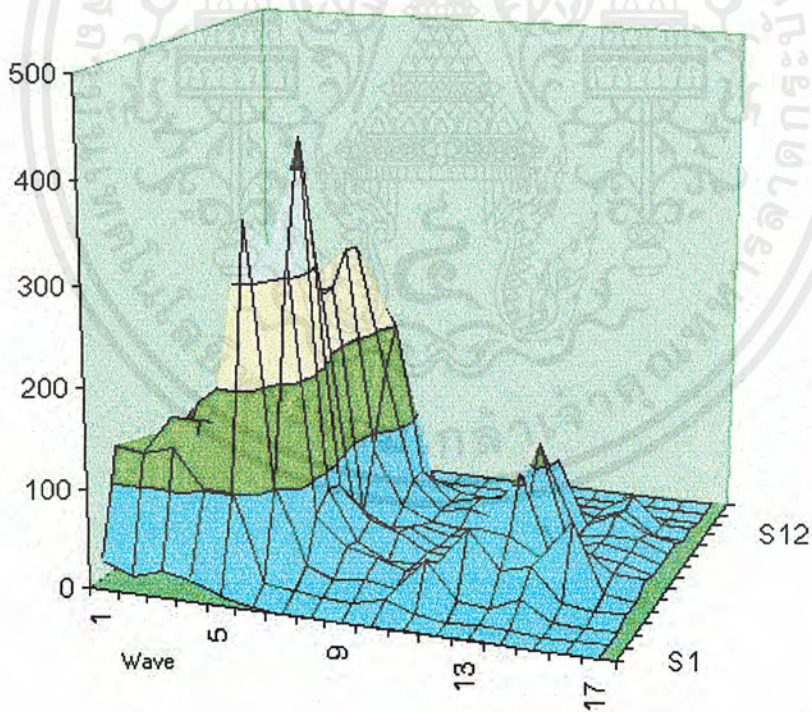


รูปที่ 4.32 แสดงสเปกโตรแกรมของเสียง Pause ที่ได้จากการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

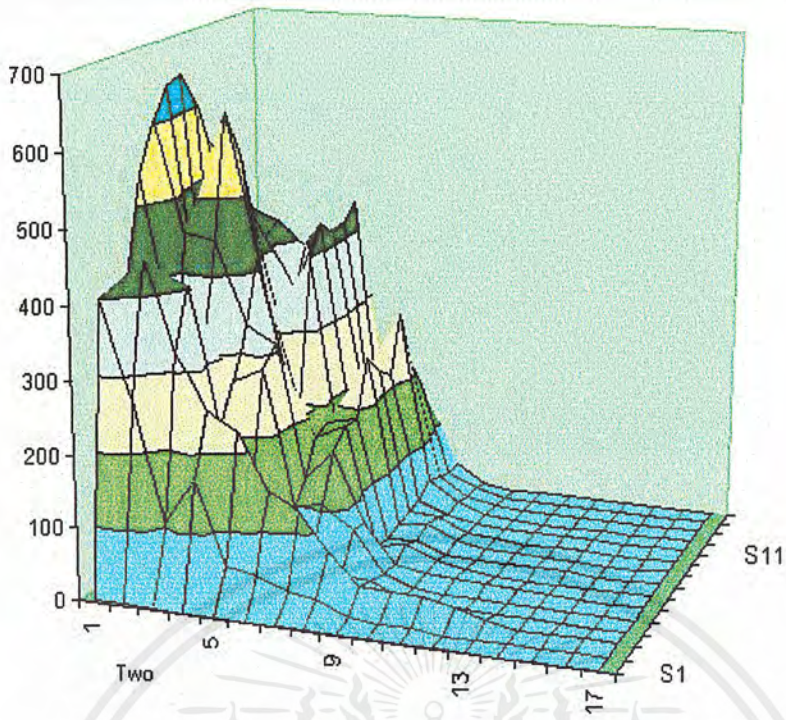


รูปที่ 4.33 แสดงสเปกโตรแกรมของเสียง Clear ที่ได้จากการบันทึก

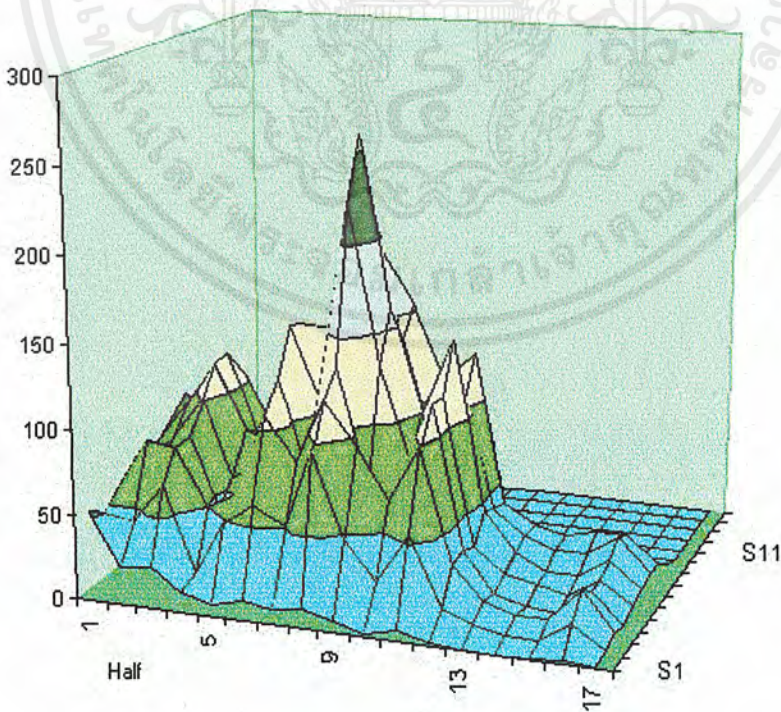


รูปที่ 4.34 แสดงสเปกโตรแกรมของเสียง Wave ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

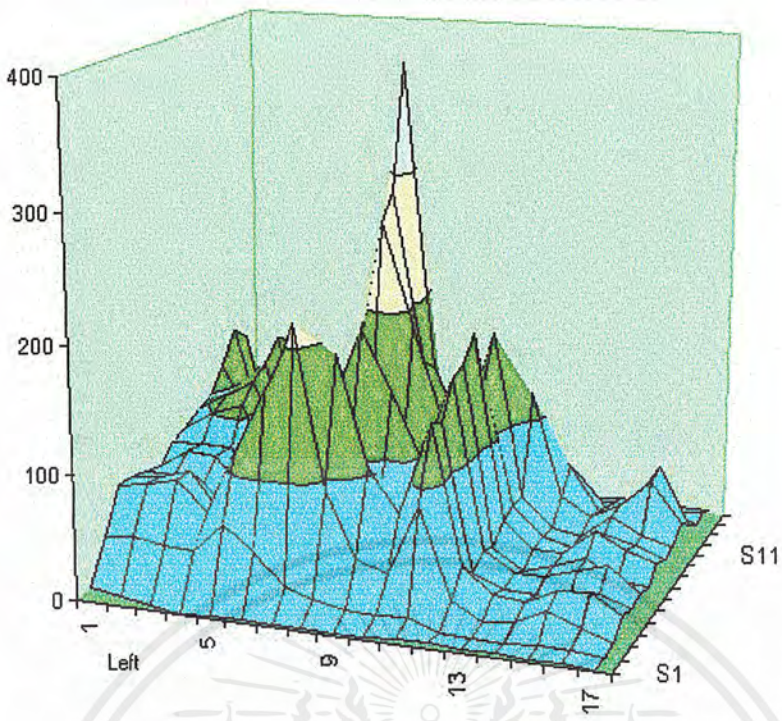


รูปที่ 4.35 แสดงสเปกโตรแกรมของเสียง Two ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

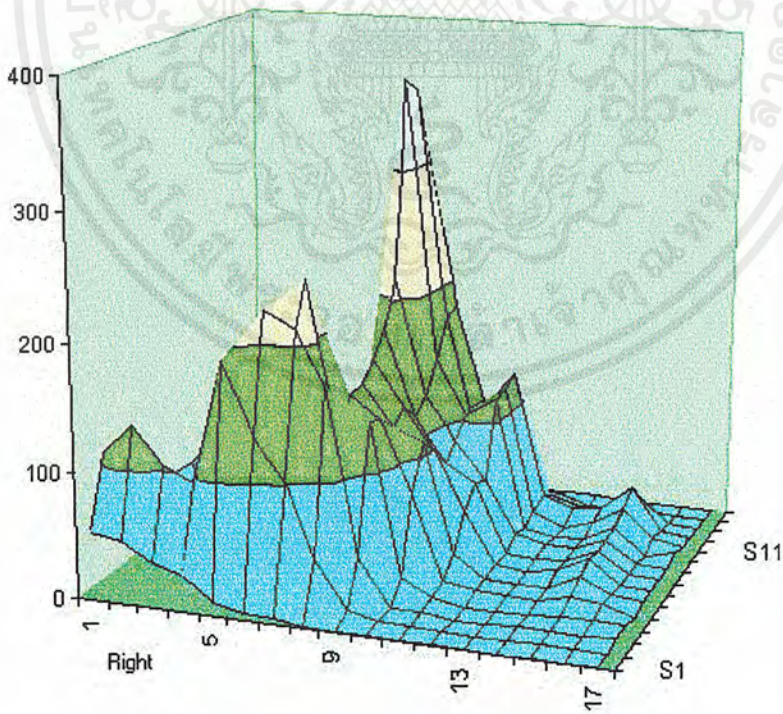


รูปที่ 4.36 แสดงสเปกโตรแกรมของเสียง Half ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

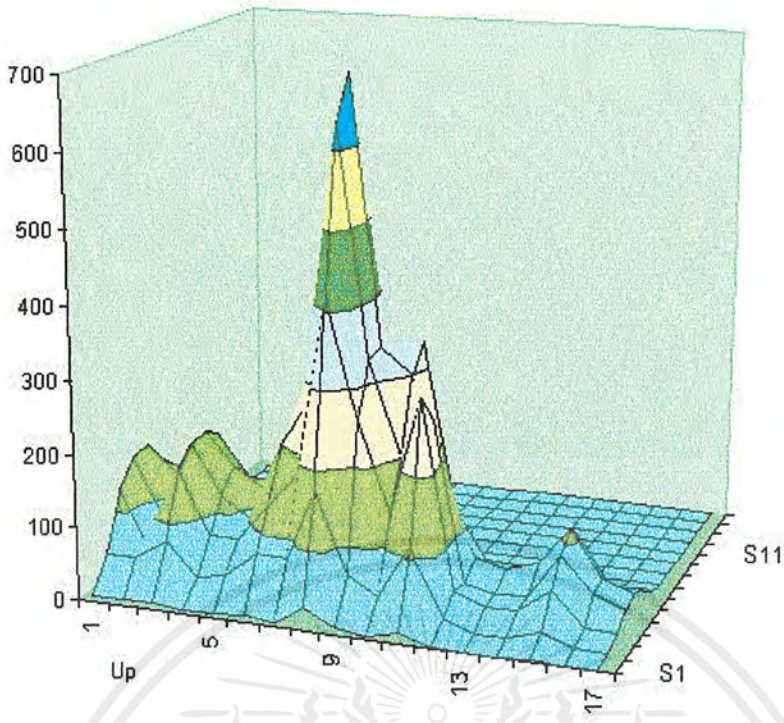


รูปที่ 4.37 แสดงสเปกโตรแกรมของเสียง ชาย ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

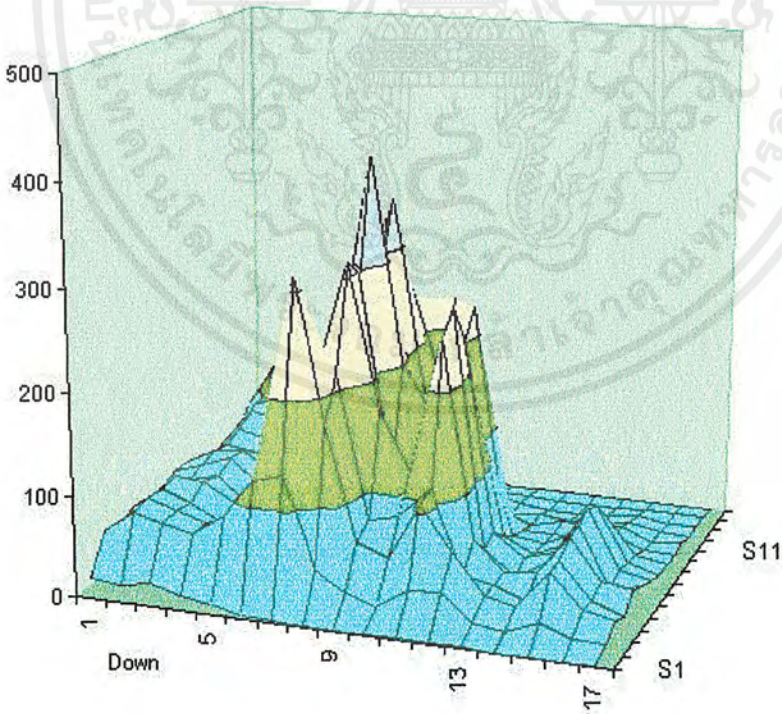


รูปที่ 4.38 แสดงสเปกโตรแกรมของเสียง ขาว ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

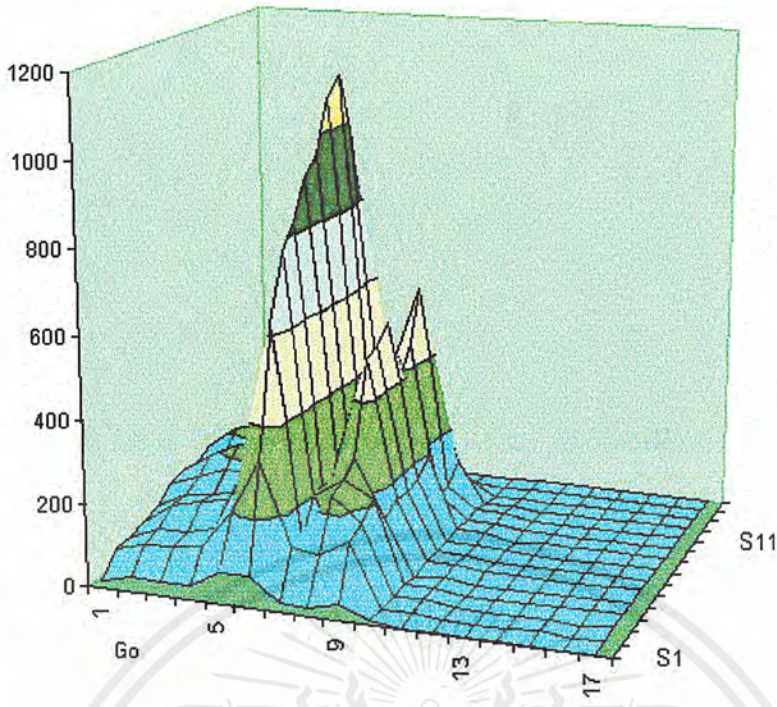


รูปที่ 4.39 แสดงสเปกโตรแกรมของเสียง Up ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

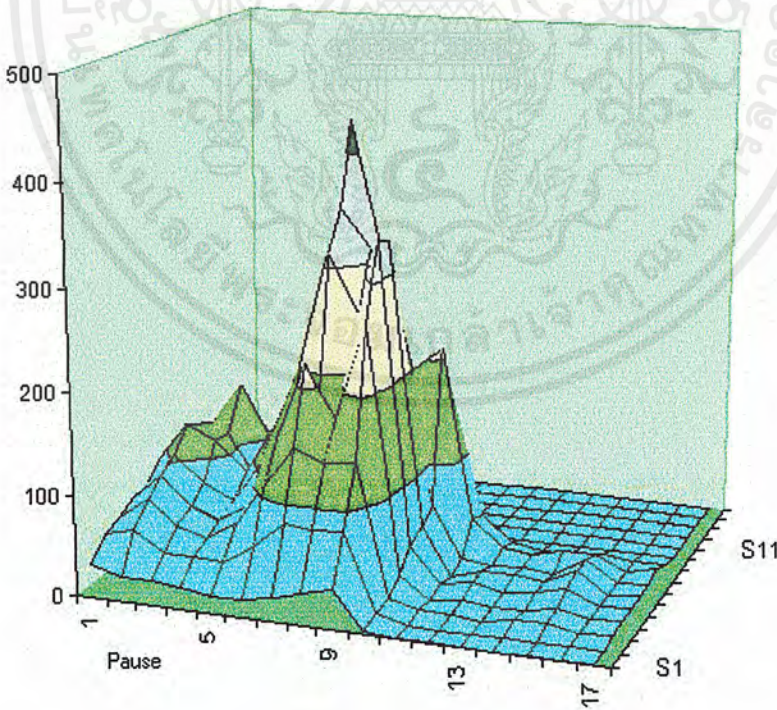


รูปที่ 4.40 แสดงสเปกโตรแกรมของเสียง Down ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

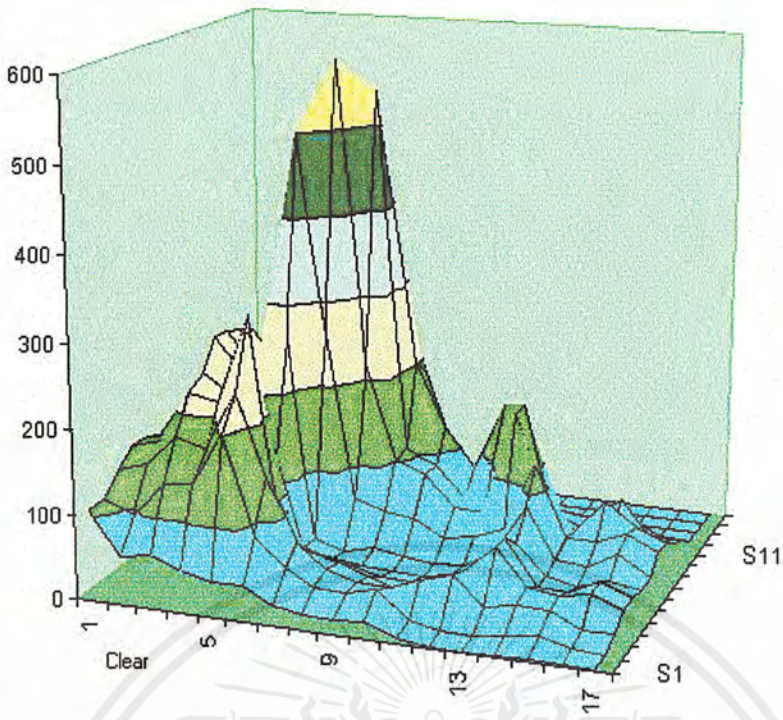


รูปที่ 4.41 แสดงสเปกโตรแกรมของเสียง Go ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว



รูปที่ 4.42 แสดงสเปกโตรแกรมของเสียง Pause ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

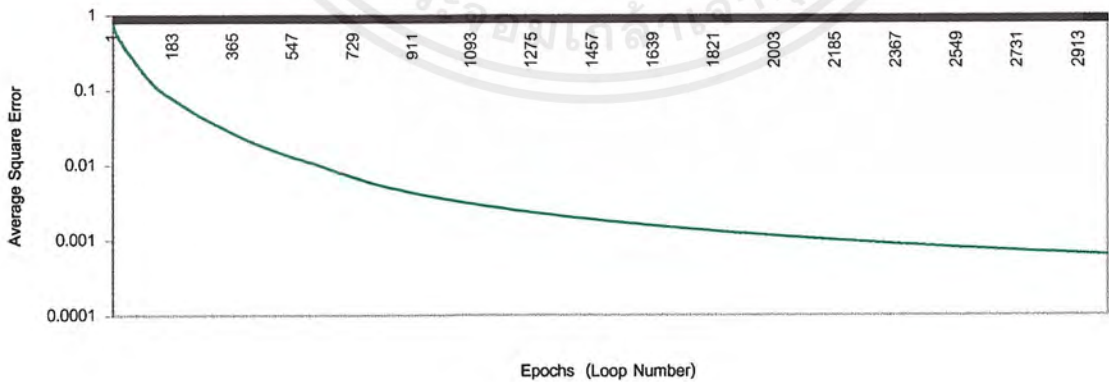
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.43 แสดงสเปกโตรแกรมของเสียง Clear ที่ผ่านการแปลง FFT และลดขนาดข้อมูลแล้ว

4.2.4 ค่า Sum Square Error

Training mode with Average Error



รูปที่ 4.44 แสดงค่า Sum Square Error ที่ได้จากการเทรน 3000 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอบที่ 1	=	0.616597
รอบที่ 1000	=	0.002279
รอบที่ 2000	=	0.000808
รอบที่ 3000	=	0.000469

4.2.5 ผลการทดสอบการรู้จำ

ตารางที่ 4.1

ตารางแสดงผลการทดสอบนิรวัลเน็ตเวิร์ค ด้วยข้อมูลเดิมที่ใช้ฝึกสอน
ค่าในตารางเป็นค่าเฉลี่ยของเสียงทดสอบเสียงละ 50 ครั้ง

เสียง ทดสอบ	เอาท์พุท									
	Wave	Two	Half	Left	Right	Up	Down	Go	Pause	Clear
Wave	0.9966	0.0015	0.0017	0.0001	0.0005	0.0000	0.0000	0.0022	0.0000	0.0027
Two	0.0009	0.9984	0.0002	0.0021	0.0002	0.0004	0.0000	0.0000	0.0000	0.0012
Half	0.0023	0.0029	0.9878	0.0042	0.0041	0.0064	0.0000	0.0008	0.0052	0.0048
Left	0.0000	0.0038	0.0017	0.9896	0.0035	0.0012	0.0047	0.0007	0.0010	0.0026
Right	0.0000	0.0014	0.0003	0.0019	0.9906	0.0000	0.0034	0.0019	0.0050	0.0010
Up	0.0010	0.0012	0.0050	0.0007	0.0016	0.9943	0.0033	0.0000	0.0007	0.0008
Down	0.0000	0.0002	0.0000	0.0041	0.0025	0.0010	0.9941	0.0018	0.0026	0.0015
Go	0.0027	0.0000	0.0005	0.0002	0.0004	0.0000	0.0001	0.9968	0.0013	0.0004
Pause	0.0000	0.0004	0.0019	0.0020	0.0061	0.0010	0.0020	0.0032	0.9931	0.0012
Clear	0.0012	0.0015	0.0009	0.0016	0.0004	0.0000	0.0004	0.0008	0.0000	0.9942

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2

ตารางแสดงผลการทดสอบนิรวัลเน็ตเวอร์ค ด้วยเสียงใหม่ของคนเดิมที่ไม่ได้ผ่านการฝึกสอน
ค่าในตารางเป็นค่าเฉลี่ยของเสียงทดสอบเสียงละ 50 ครั้ง

เสียง	เอ้าท์พุท									
	Wave	Two	Half	Left	Right	Up	Down	Go	Pause	Clear
Wave	0.9893	0.0020	0.0000	0.0004	0.0001	0.0303	0.0000	0.2501	0.0000	0.1676
Two	0.0003	1.0000	0.0000	0.0000	0.0057	0.0000	0.0000	0.0000	0.0000	0.0019
Half	0.0003	0.0005	0.7651	0.1799	0.0003	0.0005	0.0013	0.0008	0.0139	0.0003
Left	0.0046	0.0002	0.0103	0.9490	0.0002	0.0001	0.0027	0.0005	0.0027	0.0005
Right	0.0000	0.0006	0.0001	0.0009	0.9441	0.0001	0.0042	0.0013	0.0188	0.0000
Up	0.0126	0.0001	0.0170	0.0013	0.0102	0.8922	0.0008	0.0000	0.0027	0.0011
Down	0.0000	0.0006	0.0000	0.0022	0.0155	0.0003	0.8008	0.0003	0.0383	0.0077
Go	0.0029	0.0001	0.0009	0.0001	0.0125	0.0000	0.0000	0.9625	0.0001	0.0042
Pause	0.0001	0.0003	0.0380	0.0034	0.0192	0.0046	0.0004	0.0005	0.8713	0.0000
Clear	0.0832	0.0004	0.0000	0.0001	0.0000	0.0000	0.0000	0.0027	0.0000	0.9235

ตารางที่ 4.3

ตารางแสดงผลการทดสอบนิรวัลเน็ตเวอร์ค ด้วยเสียงใหม่ของคนเดิมที่ไม่ได้ผ่านการฝึกสอน
ค่าในตารางเป็นค่าเอ้าท์พุทของเสียงทดสอบเสียงละ 50 ครั้ง

เสียง	เอ้าท์พุท										% ความ
	Wave	Two	Half	Left	Right	Up	Down	Go	Pause	Clear	ถูกต้อง
Wave	49							1			98
Two		50									100
Half			43	6						1	86
Left				50							100
Right					49				1		98
Up	1					49					98
Down					1		49				98
Go								50			100
Pause									50		100
Clear	2									48	96
										เฉลี่ย	97.4
										สูงสุด	100
										ต่ำสุด	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

1. ผลการรู้จำอยู่ในระดับที่น่าพอใจ เพราะเป็นการรู้จำเสียงของคน ๆ เดียว
2. เสียงของคำว่า Half มีผลการรู้จำต่ำที่สุด เนื่องจากสเปกโตรแกรมของเสียง Half มีลักษณะเด่นต่างจากเสียงอื่นน้อย
3. การออกเสียงในขั้นตอนการทดสอบ ต้องออกเสียงให้ใกล้เคียงกับเสียงที่ใช้ในการฝึกสอน ระบบจึงจะสามารถรู้จำได้ดี
4. เสียงที่ใช้ในการฝึกสอนจะต้องมีความหลากหลาย เพื่อเพิ่มความยืดหยุ่นในการรู้จำ แต่เสียงของคำเดียวกันจะต้องไม่แตกต่างกันมากนัก เพราะอาจจะทำให้การฝึกสอนผิดพลาดได้
5. ถ้าใช้ตัวอย่างเสียงในการทดลองมาก ๆ จะทำให้อัตราการรู้จำสูงขึ้น แต่ก็ต้องใช้เวลาในการฝึกสอนนานมากขึ้นด้วย
6. ถ้าจำนวนรอบของการฝึกสอนมากขึ้น ค่าผิดพลาดก็จะลดลง แต่เมื่อถึงจุด ๆ หนึ่ง ค่าผิดพลาดก็จะเริ่มลดลงอย่างช้า ๆ เนื่องจากค่าเวกซ์เริ่มเข้าใกล้ค่าที่เหมาะสมแล้ว
7. ถ้าทดสอบด้วยเสียงของคนที่ไม่ได้ใช้ในการฝึกสอน ระบบจะรู้จำได้บ้างเป็นบางคำ มีเปอร์เซ็นต์ความถูกต้องต่ำ
8. ในการแปลงเสียง ถ้ามีลมที่เกิดจากการเป่าริมฝีปากเข้าไปมาก จะทำให้การรู้จำผิดพลาด ดังนั้นก่อนการแปลงเสียง ควรเขี่ยลมฝีปากเอาไว้เล็กน้อย
9. สัญญาณรบกวนที่มีขนาดต่ำ ๆ จะไม่มีผลต่อการรู้จำของระบบมากนัก แต่ถ้าสัญญาณรบกวนมีขนาดใหญ่ จะทำให้การตัดเสียงผิดพลาด ทำให้รูปของสเปกโตรแกรมคลาดเคลื่อน

5.2 วิจารณ์ผลการทดลอง

อัตราการรู้จำจะขึ้นอยู่กับปัจจัยต่าง ๆ ดังนี้

1. ความสามารถในการหาคุณลักษณะที่แตกต่างกันของเสียงที่ใช้ในการทดลองของส่วนปริโปรเซส- ซึ่ง ถ้าสามารถทำให้เห็นความแตกต่างของเสียงที่ต่างกันได้มากเท่าไร อัตราการรู้จำก็จะมากขึ้นเท่านั้น เราอาจเพิ่มอัตราการรู้จำได้โดยเปลี่ยนตัวเลือกในแกนความถี่ใหม่ ให้สอดคล้องกับลักษณะเด่นของคำที่ใช้
2. ความแตกต่างกันของคำที่ใช้ ถ้าใช้คำที่มีเสียงแตกต่างกันมากขึ้นอัตราการรู้จำก็จะมากขึ้นตาม
3. ลักษณะการแปลงเสียงของผู้ที่ใช้ฝึกสอน ว่าสามารถแปลงเสียงของคำแต่ละคำที่ต่างกันได้อย่างมากเท่าใด ถ้าแปลงเสียงได้ต่างกันมาก อัตราการรู้จำก็จะมาก
4. เสียงที่ใช้ในการฝึกสอน ถ้าเสียงที่เป็นคำเดียวกันมีความแตกต่างกันน้อย ในการฝึกสอนจะได้ค่าผิดพลาด (Sum Square Error) ลดลงอย่างรวดเร็ว เมื่อทดสอบด้วยเสียงเดิมที่ใช้ในการฝึกสอนจะได้ค่าเอาท์พุทของคำที่ถูกต้องสูง แต่ถ้าทดสอบด้วยเสียงที่แตกต่างจากเสียงที่ใช้ในการฝึกสอนมาก ความผิดพลาดของเอาท์พุทก็จะมากด้วย ดังนั้นเพื่อความถูกต้องของเอาท์พุท จึงควรใช้เสียงในการฝึกสอนที่มีความแตกต่างกันพอสมควร เพื่อให้สามารถครอบคลุมการเปลี่ยนแปลงในการแปลงเสียงของคำ ๆ เดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

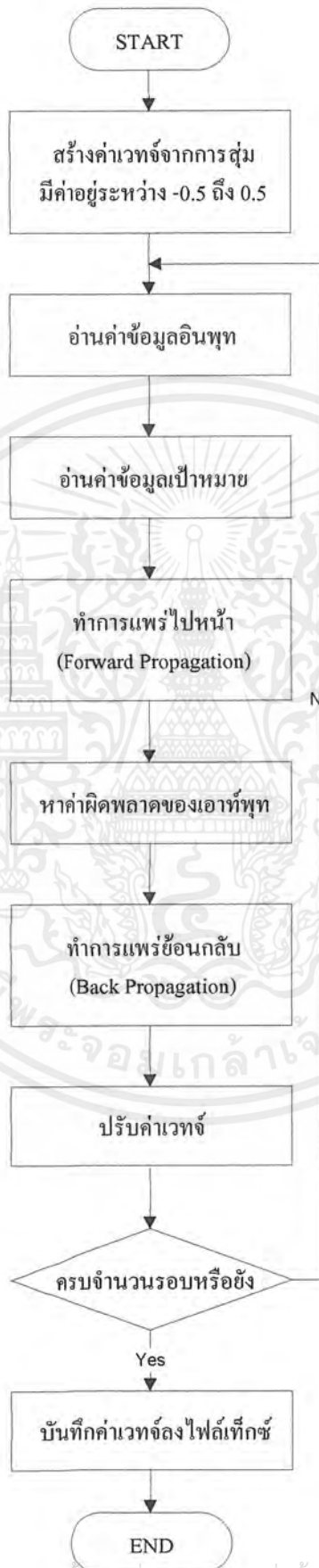
5.3 แนวทางการพัฒนา

1. พัฒนาการรับเสียงให้สามารถตรวจจับเสียงที่เข้ามาได้โดยอัตโนมัติ ไม่ต้องใช้วิธีการกดปุ่ม อาจทำได้โดยการเขียนโปรแกรมติดต่อกับการ์ดเสียง ให้การ์ดเสียงเริ่มทำการบันทึกเสียงโดยอัตโนมัติเมื่อสัญญาณเสียงที่เข้ามาทางไมโครโฟนมีขนาดของสัญญาณสูงกว่าระดับที่กำหนดไว้
2. พัฒนาการตัดเสียงให้แม่นยำขึ้น โดยอาจจะตัดเสียงจากสเปกโตรแกรมแทนการตัดเสียงในโดเมนเวลา เพราะถ้ามีสัญญาณรบกวนเข้ามาก่อนที่จะปลั่งเสียง การตัดเสียงในโดเมนเวลาจะเกิดความผิดพลาดได้มาก เมื่อนำข้อมูลเสียงไปแปลงสเปกโตรแกรม จะทำให้ได้สเปกโตรแกรมที่มีตำแหน่งที่ไม่แน่นอน และการวิเคราะห์โดยใช้นิวรัลเน็ตเวิร์ค ถ้าตำแหน่งของสเปกโตรแกรมไม่ตรงกัน จะทำให้การวิเคราะห์ผิดพลาด
3. พัฒนาการลดข้อมูลก่อนที่จะป้อนเข้านิวรัลเน็ตเวิร์คให้มีประสิทธิภาพมากขึ้น โดยการลดข้อมูลของสเปกโตรแกรมในแกนเวลาอาจใช้วิธีการประมาณค่า(Interpolation)แทนการหาค่าเฉลี่ย ส่วนในแกนความถี่อาจจะเปลี่ยนค่าตัวเลือกให้เหมาะสมกับเสียงที่ใช้มากขึ้น
4. พัฒนาส่วนของนิวรัลเน็ตเวิร์ค โดยอาจเพิ่มชั้นของ Hidden Layer ให้มากขึ้น เพื่อเพิ่มประสิทธิภาพในการคำนวณ จำนวนชั้นที่เหมาะสมจะทราบได้จากการทดลอง ซึ่งบางครั้งจำนวนชั้นที่เพิ่มขึ้นอาจจะไม่ทำให้ได้ผลที่ดีขึ้นเสมอไป จำนวนที่เหมาะสมของแต่ละงานจะแตกต่างกันไป
5. พัฒนาให้สามารถรู้จำคำพูดได้มากขึ้น ให้ใช้กับคำพูดเป็นประโยคได้
6. นำระบบการรู้จำไปประยุกต์ใช้กับงานอื่นๆ เช่น ใช้แยกแยะบุคคล, ใช้เปิด-ปิดสวิทช์, ควบคุมอุปกรณ์ไฟฟ้า, ควบคุมหุ่นยนต์, ประยุกต์ใช้กับคนพิการ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของโปรแกรมทดสอบเสียง Neural.exe



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของโปรแกรมทดสอบเสียง SndTest.exe



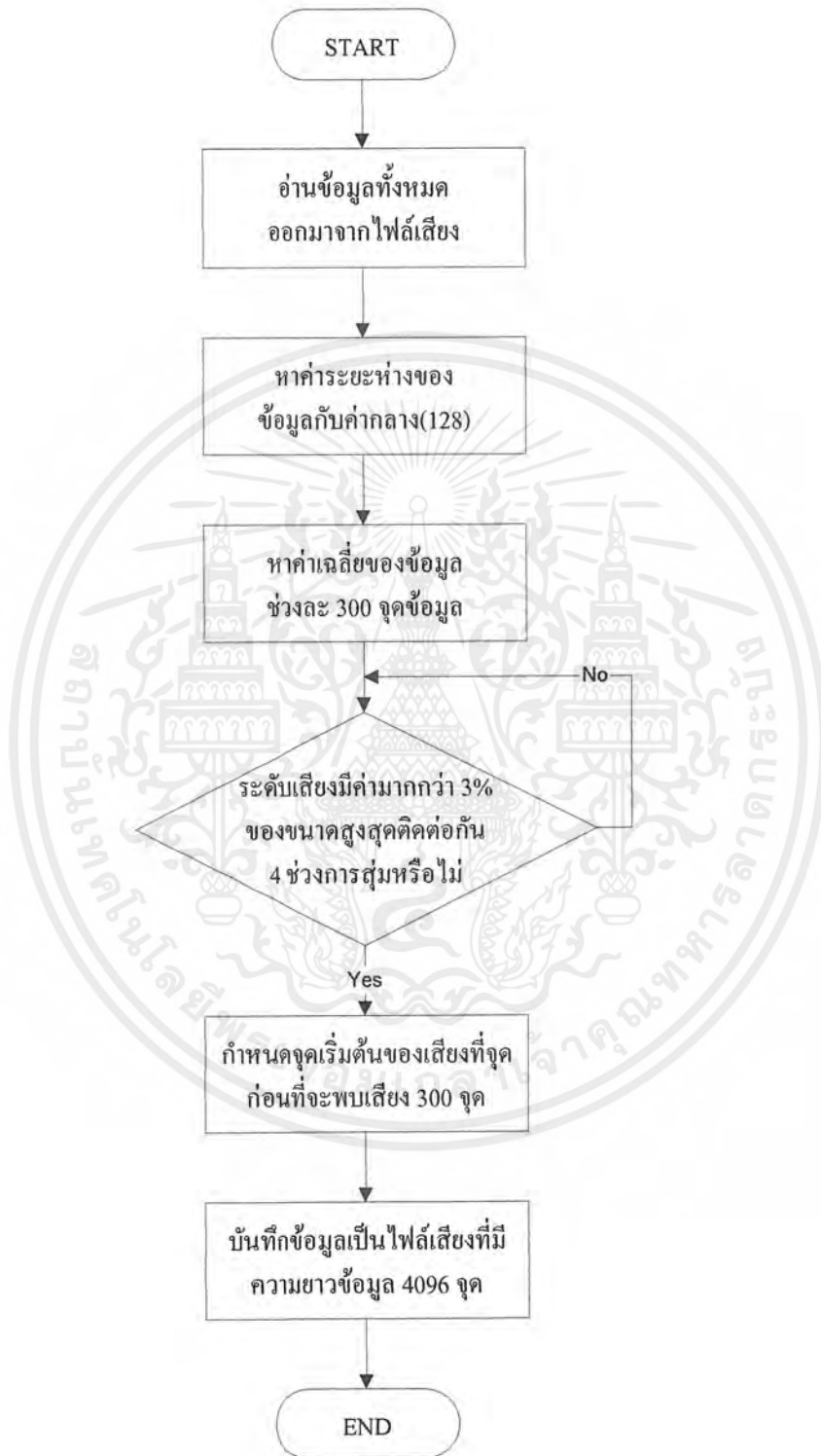
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของส่วน Pre-processing



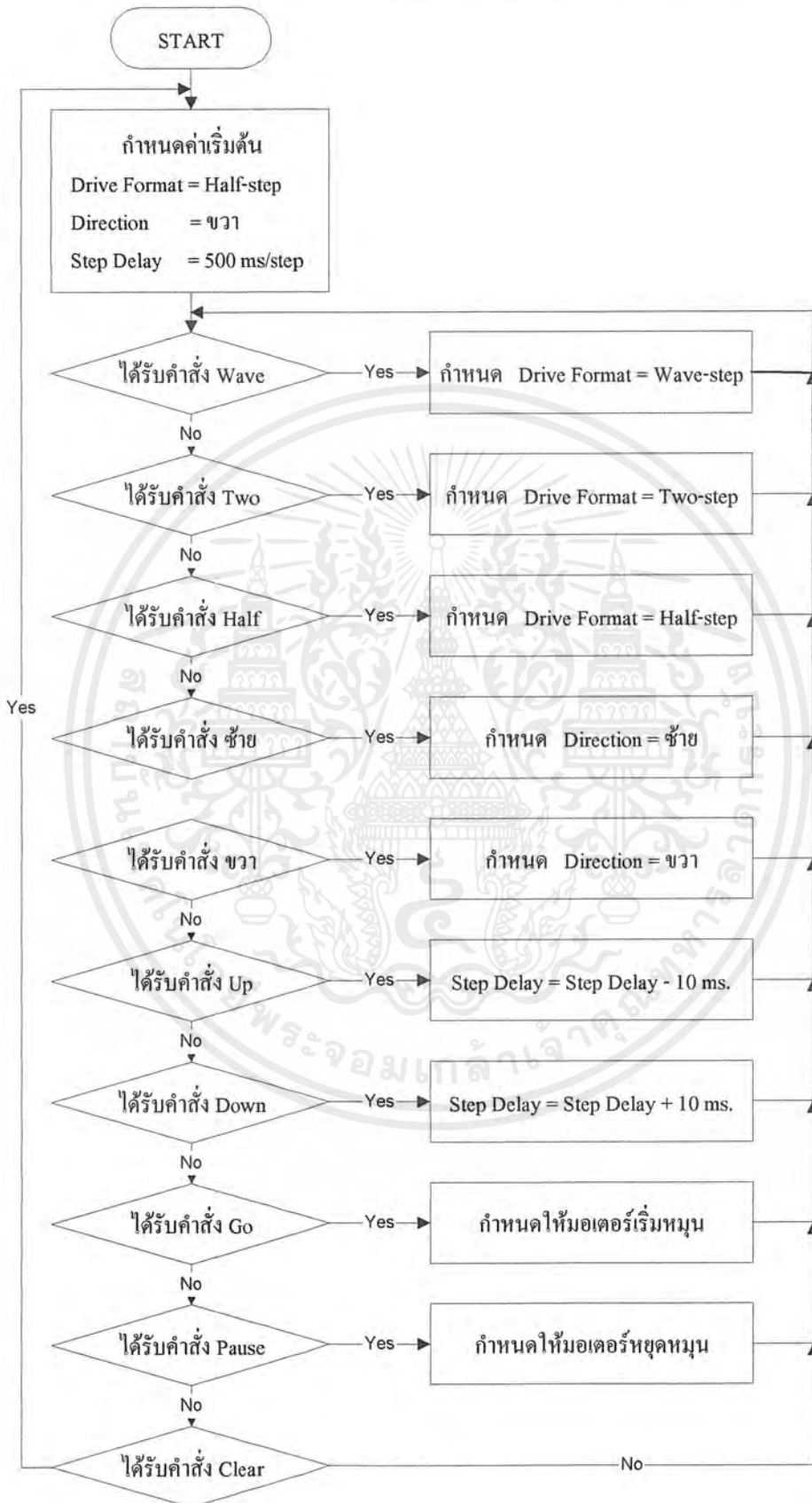
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของส่วนการตัดเสียง



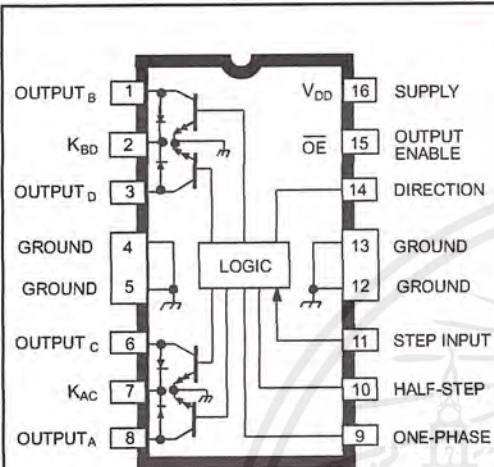
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของส่วนควบคุมมอเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804

**BiMOS II UNIPOLAR
STEPPER-MOTOR TRANSLATOR/DRIVER**

Dwg. W-194

Note that the UCN5804B (dual in-line package) and UCN5804LB (small outline IC package) are electrically identical and share a common pin number assignment.

ABSOLUTE MAXIMUM RATINGS

Output Voltage, V_{CE}	50 V
Output Sustaining Voltage, $V_{CE(sus)}$	35 V
Output Sink Current, I_{OUT}	1.5 A
Logic Supply Voltage, V_{DD}	7.0 V
Input Voltage, V_{IN}	7.0 V
Package Power Dissipation, P_D	See Graph
Operating Temperature Range, T_A	-20°C to +85°C
Storage Temperature Range, T_S	-55°C to +150°C

Combining low-power CMOS logic with high-current and high-voltage bipolar outputs, the UCN5804B and UCN5804LB BiMOS II translator/drivers provide complete control and drive for a four-phase unipolar stepper-motor with continuous output current ratings to 1.25 A per phase (1.5 A startup) and 35 V.

The CMOS logic section provides the sequencing logic, DIRECTION and OUTPUT ENABLE control, and a power-ON reset function. Three stepper-motor drive formats, wave-drive (one-phase), two-phase, and half-step are externally selectable. The inputs are compatible with standard CMOS, PMOS, and NMOS circuits. TTL or LSTTL may require the use of appropriate pull-up resistors to ensure a proper input-logic high.

The wave-drive format consists of energizing one motor phase at a time in an A-B-C-D (or D-C-B-A) sequence. This excitation mode consumes the least power and assures positional accuracy regardless of any winding imbalance in the motor. Two-phase drive energizes two adjacent phases in each detent position (AB-BC-CD-DA). This sequence mode offers an improved torque-speed product, greater detent torque, and is less susceptible to motor resonance. Half-step excitation alternates between the one-phase and two-phase modes (A-AB-B-BC-C-CD-D-DA), providing an eight-step sequence.

The bipolar outputs are capable of sinking up to 1.5 A and withstanding 50 V in the OFF state (sustaining voltages up to 35 V). Ground-clamp and flyback diodes provide protection against inductive transients. Thermal protection circuitry disables the outputs when the chip temperature is excessive.

Both devices are rated for operation over the temperature range of -20°C to +85°C. The UCN5804B is supplied in a 16-pin dual in-line plastic batwing package with a copper lead frame and heat-sinkable tabs for improved power dissipation capabilities; the UCN5804LB is supplied in a 16-lead plastic SOIC batwing package with a copper lead frame and heat-sinkable tabs.

FEATURES

- 1.5 A Maximum Output Current
- 35 V Output Sustaining Voltage
- Wave-Drive, Two-Phase, and Half-Step Drive Formats
- Internal Clamp Diodes
- Output Enable and Direction Control
- Power-ON Reset
- Internal Thermal Shutdown Circuitry

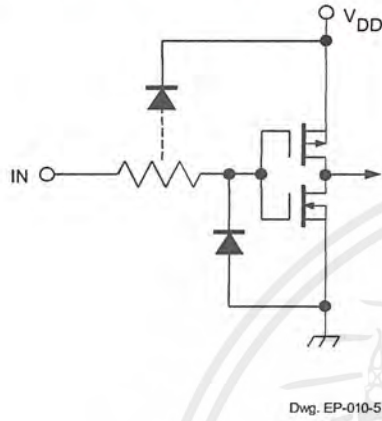
Always order by complete part number, e.g., **UCN5804B**.



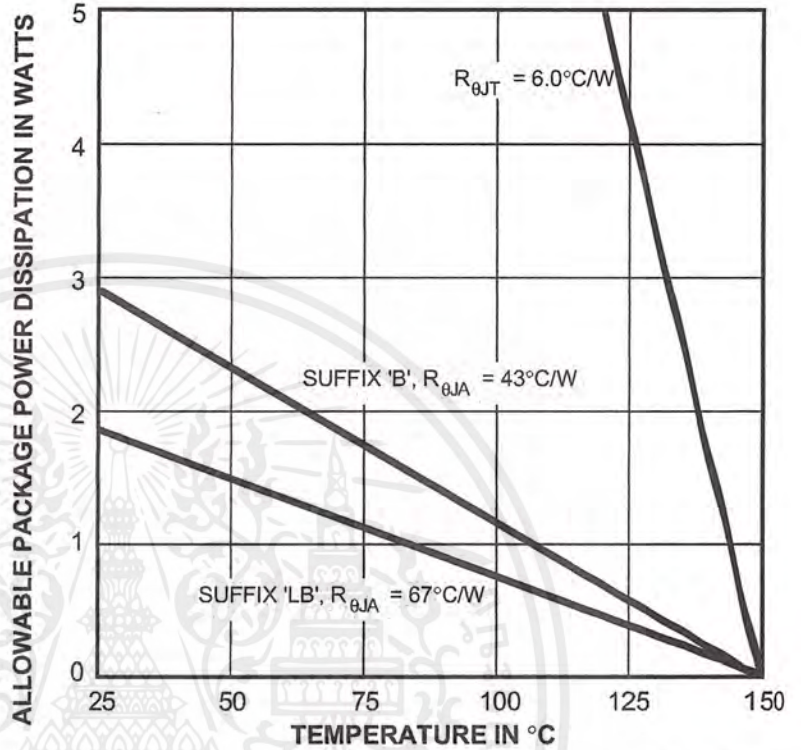
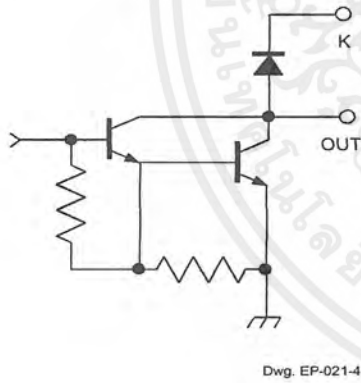
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804
BiMOS II UNIPOLAR
STEPPER-MOTOR
TRANSLATOR/DRIVER

TYPICAL INPUT CIRCUIT



TYPICAL OUTPUT DRIVER



Dwg. GP-049-2

TRUTH TABLE

Drive Format	Pin 9	Pin 10
Two-Phase	L	L
One-Phase	H	L
Half-Step	L	H
Step-Inhibit	H	H



115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000
 Copyright © 1987, 1998, Allegro MicroSystems, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

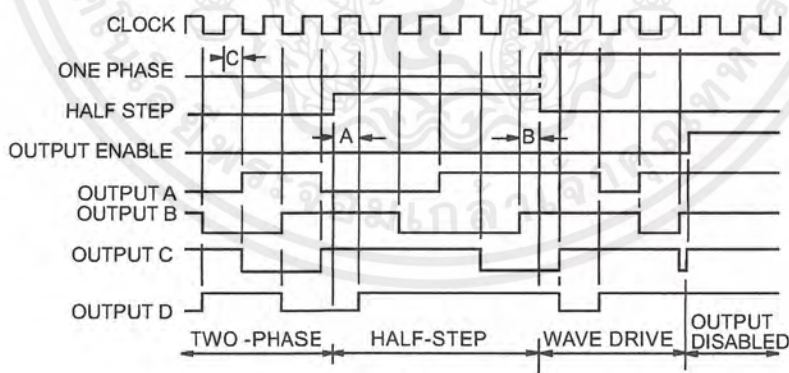
5804

BiMOS II UNIPOLAR STEPPER-MOTOR TRANSLATOR/DRIVER

ELECTRICAL CHARACTERISTICS at $T_A = 25^\circ\text{C}$, $T_J \leq 150^\circ\text{C}$, $V_{DD} = 4.5\text{ V to }5.5\text{ V}$
(unless otherwise noted).

Characteristic	Symbol	Test Conditions	Limits			Units
			Min.	Typ.	Max.	
Output Leakage Current	I_{CEX}	$V_{OUT} = 50\text{ V}$	—	10	50	μA
Output Sustaining Voltage	$V_{CE(sus)}$	$I_{OUT} = 1.25\text{ A}$, $L = 3\text{ mH}$	35	—	—	V
Output Saturation Voltage	$V_{CE(SAT)}$	$I_{OUT} = 700\text{ mA}$	—	1.0	1.2	V
		$I_{OUT} = 1\text{ A}$	—	1.1	1.4	V
		$I_{OUT} = 1.25\text{ A}$	—	1.2	1.5	V
Clamp Diode Leakage Current	I_R	$V_R = 50\text{ V}$	—	10	50	μA
Clamp Diode Forward Voltage	V_F	$I_F = 1.25\text{ A}$	—	1.5	3.0	V
Input Current	$I_{IN(1)}$	$V_{IN} = V_{DD}$	—	0.5	5.0	μA
	$I_{IN(0)}$	$V_{IN} = 0.8\text{ V}$	—	-0.5	-5.0	μA
Input Voltage	$V_{IN(1)}$	$V_{DD} = 5\text{ V}$	3.5	—	5.3	V
	$V_{IN(0)}$		-0.3	—	0.8	V
Supply Current	I_{DD}	2 Outputs ON	—	20	30	mA
Turn-Off Delay	t_{ON}	50% Step Inputs to 50% Output	—	—	10	μs
Turn-On Delay	t_{OFF}	50% Step Inputs to 50% Output	—	—	10	μs
Thermal Shutdown Temperature	T_J		—	165	—	$^\circ\text{C}$

TIMING CONDITIONS



Dwg. W-110A

- A. Minimum Data Set Up Time 100 ns
 B. Minimum Data Hold Time 100 ns
 C. Minimum Step Input Pulse Width 3.0 μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804 BiMOS II UNIPOLAR STEPPER-MOTOR TRANSLATOR/DRIVER

APPLICATIONS INFORMATION

Internal power-ON reset (POR) circuitry resets OUTPUT_A (and OUTPUT_D in the two-phase drive format) to the ON state with initial application of the logic supply voltage. After reset, the circuit then steps according to the tables.

The outputs will advance one sequence position on the high-to-low transition of the STEP INPUT pulse. Logic levels on the HALF-STEP and ONE-PHASE inputs will determine the drive format (one-phase, two-phase, or half-step). The DIRECTION pin determines the rotation sequence of the outputs. Note that the STEP INPUT must be in the low state when changing the state of ONE-PHASE, HALF-STEP, or DIRECTION to prevent erroneous stepping.

All outputs are disabled (OFF) when OUTPUT ENABLE is at a logic high. If the function is not required, OUTPUT ENABLE should be tied low. In that condition, all outputs depend only on the state of the step logic.

During normal commutation of a unipolar stepper motor, mutual coupling between the motor windings can force the outputs of the UCN5804B below ground. This condition will cause forward biasing of the collector-to-substrate junction and source current from the output. For many L/R applications, this substrate current is high enough to adversely affect the logic circuitry and cause misstepping. External series diodes (Schottky are recommended for increased efficiency at low-voltage operation) will prevent substrate current from being sourced through the outputs. Alternatively, external ground clamp diodes will provide a preferred current path from ground when the outputs are pulled below ground.

Internal thermal protection circuitry disables all outputs when the junction temperature reaches approximately 165°C. The outputs are enabled again when the junction cools down to approximately 145°C.

WAVE-DRIVE SEQUENCE

Half Step = L, One Phase = H				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

TWO-PHASE DRIVE SEQUENCE

Half Step = L, One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	ON
1	ON	OFF	OFF	ON
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON

HALF-STEP DRIVE SEQUENCE

Half Step = H, One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

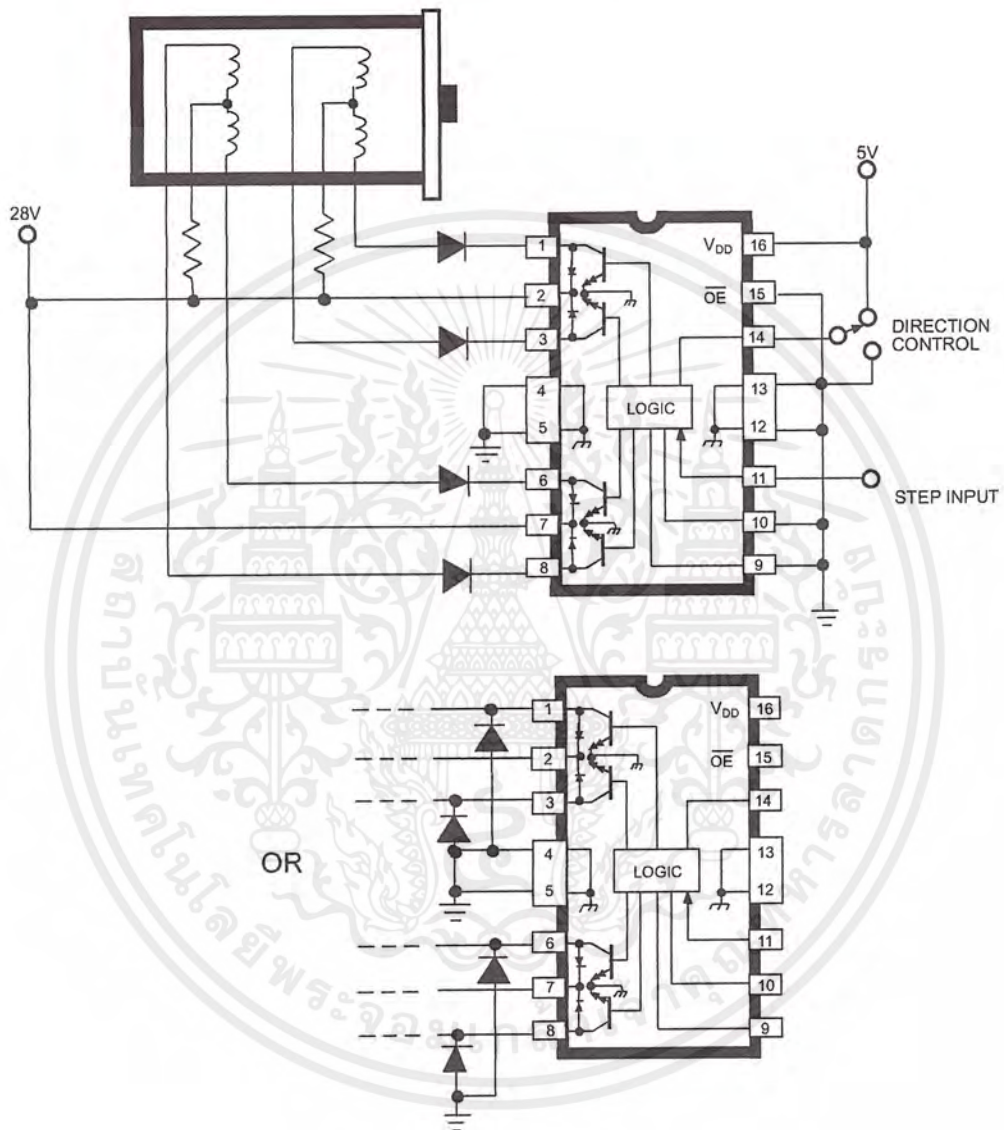


115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804
BiMOS II UNIPOLAR
STEPPER-MOTOR
TRANSLATOR/DRIVER

TYPICAL APPLICATION
L/R Stepper-Motor Drive

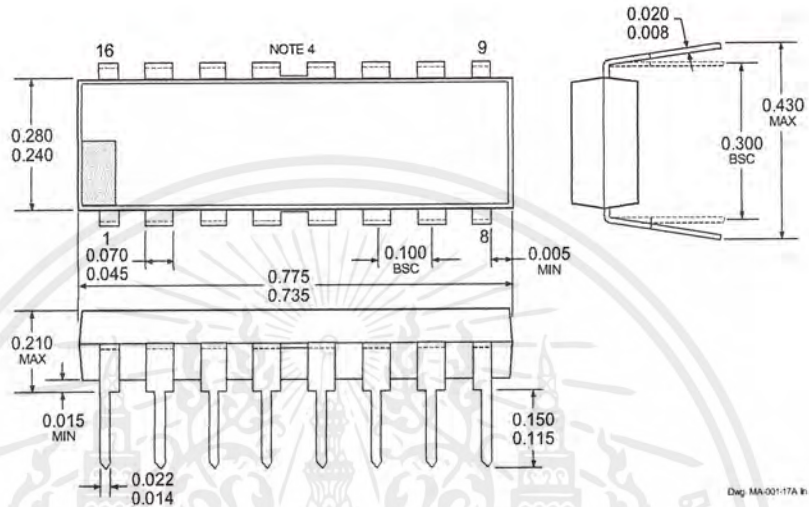


Dwg. EP-029A

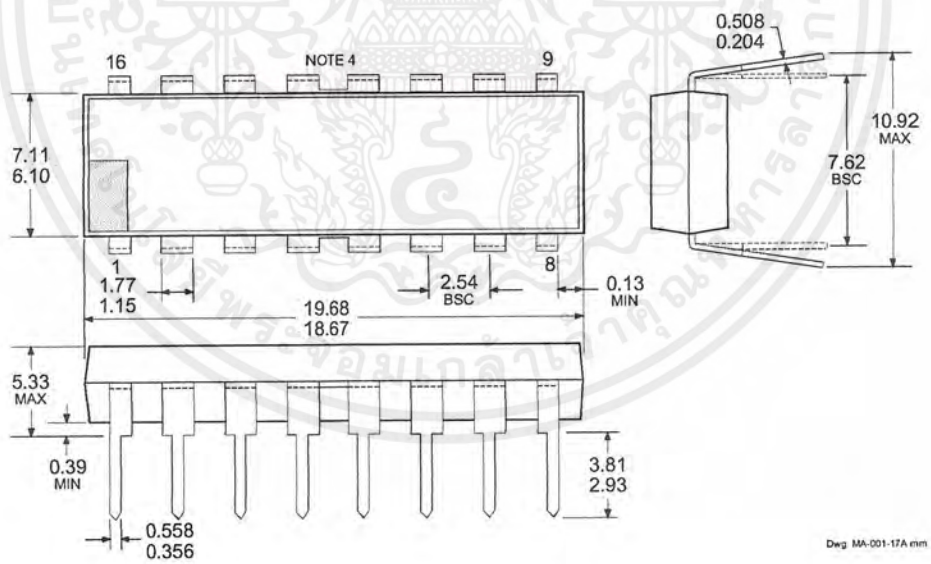
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804
BiMOS II UNIPOLAR
STEPPER-MOTOR
TRANSLATOR/DRIVER

UCN5804B
Dimensions in Inches
(controlling dimensions)



Dimensions in Millimeters
(for reference only)



- NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
 2. Lead spacing tolerance is non-cumulative
 3. Lead thickness is measured at seating plane or below.
 4. Webbed lead frame. Leads 4, 5, 12, and 13 are internally one piece.

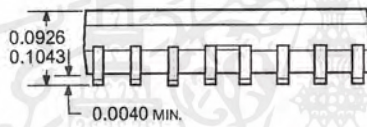
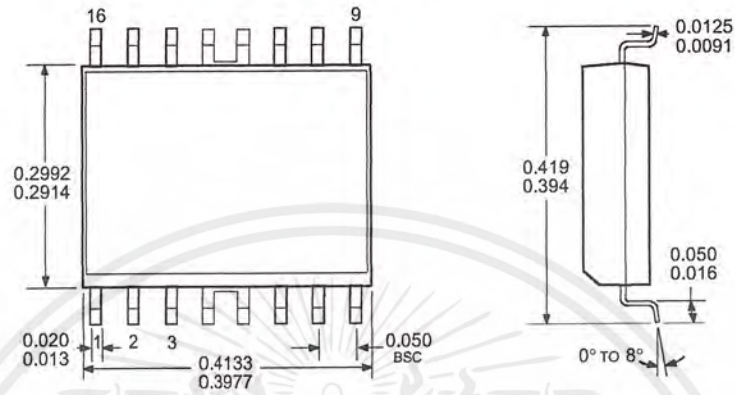


115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000

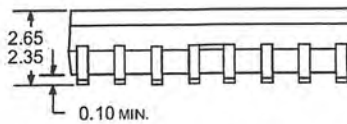
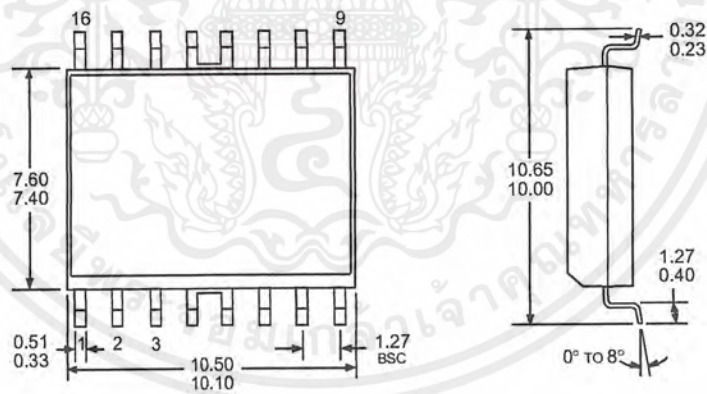
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804
BiMOS II UNIPOLAR
STEPPER-MOTOR
TRANSLATOR/DRIVER

UCN5804LB
Dimensions in Inches
 (for reference only)



Dimensions in Millimeters
 (controlling dimensions)



- NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
 2. Lead spacing tolerance is non-cumulative
 3. Lead thickness is measured at seating plane or below.
 4. Webbed lead frame. Leads 4, 5, 12, and 13 are internally one piece.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804
BiMOS II UNIPOLAR
STEPPER-MOTOR
TRANSLATOR/DRIVER

MOTOR DRIVERS SELECTION GUIDE

Function	Output Ratings *		Part Number †
INTEGRATED CIRCUITS FOR BRUSHLESS DC MOTORS			
3-Phase Controller/Drivers	±2.0 A	45 V	2936 and 2936-120
Hall-Effect Latched Sensors	10 mA	24 V	3175 and 3177
2-Phase Hall-Effect Sensor/Controller	20 mA	25 V	3235
Hall-Effect Complementary-Output Sensor	20 mA	25 V	3275
2-Phase Hall-Effect Sensor/Driver	900 mA	14 V	3625
2-Phase Hall-Effect Sensor/Driver	400 mA	26 V	3626
Hall-Effect Complementary-Output Sensor/Driver	300 mA	60 V	5275
3-Phase Back-EMF Controller/Driver	±900 mA	14 V	8902-A
3-Phase Controller/DMOS Driver	±4.0 A	14 V	8925
3-Phase Back-EMF Controller/Driver	±1.0 A	7 V	8984
INTEGRATED BRIDGE DRIVERS FOR DC AND BIPOLAR STEPPER MOTORS			
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	2916
PWM Current-Controlled Dual Full Bridges	±1.5 A	45 V	2917 and 2918
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	2919
Dual Full-Bridge Driver	±2.0 A	50 V	2998
PWM Current-Controlled Full Bridge	±2.0 A	50 V	3952
PWM Current-Controlled Full Bridge	±1.3 A	50 V	3953
PWM Current-Controlled Microstepping Full Bridges	±1.5 A	50 V	3955 and 3957
PWM Current-Controlled Dual Full Bridge	±800 mA	33 V	3964
PWM Current-Controlled Dual Full Bridge	±650 mA	30 V	3966 and 3968
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	6219
OTHER INTEGRATED CIRCUIT & PMCM MOTOR DRIVERS			
Unipolar Stepper-Motor Quad Driver	1.8 A	50 V	2544
Unipolar Stepper-Motor Translator/Driver	1.25 A	50 V	5804
Unipolar Stepper-Motor Quad Drivers	1 A	46 V	7024 and 7029
Unipolar Microstepper-Motor Quad Driver	1.2 A	46 V	7042
Voice-Coil Motor Driver	±500 mA	6 V	8932-A
Voice-Coil Motor Driver	±800 mA	16 V	8958
Voice-Coil (and Spindle) Motor Driver	±350 mA	7 V	8984

* Current is maximum specified test condition, voltage is maximum rating. See specification for sustaining voltage limits or over-current protection voltage limits. Negative current is defined as coming out of (sourcing) the output.

† Complete part number includes additional characters to indicate operating temperature range and package style.

Allegro MicroSystems, Inc. reserves the right to make, from time to time, such departures from the detail specifications as may be required to permit improvements in the design of its products. The information included herein is believed to be accurate and reliable. However, Allegro MicroSystems, Inc. assumes no responsibility for its use; nor for any infringements of patents or other rights of third parties which may result from its use.



115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงได้ด้วยความช่วยเหลือจากหลายๆ ด้าน โดยเฉพาะ ผศ.ดร.สุรพันธุ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา ที่ได้ช่วยแนะนำ ให้คำปรึกษา และชี้แนะแนวทางวิธีการ ซึ่งเป็นประโยชน์อย่างมาก ในการทำปริญญาานิพนธ์ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ อาจารย์เสรี ปานซาง ที่ได้ให้ข้อมูลเรื่อง Neural Network และได้ให้คำปรึกษาในหลายๆ เรื่องเป็นอย่างดี ผู้จัดทำมีความซาบซึ้งเป็นอย่างยิ่ง

ขอขอบคุณ รุ่นพี่ปริญญาโท ที่ให้คำแนะนำเกี่ยวกับโปรแกรมต่างๆ เพื่อนๆ ที่ให้ตัวอย่างเสียง และน้องๆ ที่ให้ข้อมูล

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา ที่ให้การสนับสนุน และคอยให้กำลังใจ จนปริญญาานิพนธ์นี้ สำเร็จลงด้วยดี

นิรุตต์ จุลเดชนันทชัย
พิเชษฐ อยู่ศรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] เสรี ปานชาง, การรู้จำเสียงพูดคำไทยแบบไม่ขึ้นกับผู้พูดด้วยนิวรัลเน็ตเวิร์ค, วิทยานิพนธ์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ.2540
- [2] เสรี ปานชาง และ ชม กัมปาน, การรู้จำเสียงพูดคำไทยเฉพาะบุคคลด้วยนิวรัลเน็ตเวิร์ค, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่18, ชลบุรี: มหาวิทยาลัยมหานคร, 982-987, พ.ศ.2538
- [3] อันวา ศรีประโมง, การวิเคราะห์เสียงพูดภาษาไทยในแกนความถี่ฮาร์โมนิค, วิทยานิพนธ์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ.2538
- [4] อันวา ศรีประโมง และชม กัมปาน, การวิเคราะห์เสียงภาษาไทยในแกนความถี่ฮาร์โมนิค, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่15, พ.ศ.2535
- [5] อันวา ศรีประโมง, การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม, กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีมหานคร, พ.ศ.2539
- [6] ทศเวท วีระวัฒน์, การรู้จำเสียงคำไทยเฉพาะบุคคล, วิทยานิพนธ์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ.2541
- [7] พรชัย ภววงศ์ศักดิ์, การประมวลผลสัญญาณดิจิทัลเบื้องต้น, กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีมหานคร, พ.ศ.2542
- [8] นุกูล กระจาย, การเขียนโปรแกรมในดอสและวินโดวส์ด้วยบอร์แลนด์ C++ 5.0, กรุงเทพฯ: บริษัทซีเอ็ดยูเคชั่นจำกัด, พ.ศ.2540
- [9] อบศ.นท.ดร.สุวรรณ ศรีเกษม, MATLAB เพื่อการแก้ปัญหาทางวิศวกรรม, กรุงเทพฯ: สำนักพิมพ์มหาวิทยาลัยรังสิต, พ.ศ.2538
- [10] ระพีพัฒน์ เพ็ญศิริ และสมชาย จิตะพันธ์กุล, การรู้จำเสียงตัวเลขภาษาไทยโดยไม่ขึ้นต่อผู้พูดโดยการใช้ไดนามิกโทมวาร์บิง, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่18, ชลบุรี: มหาวิทยาลัยมหานคร, 977-981, พ.ศ.2538
- [11] ระพีพัฒน์ เพ็ญศิริ, การรู้จำเสียงพูดตัวเลขไทยโดยไม่ขึ้นต่อผู้พูดโดยการใช้ไดนามิกโทมวาร์บิง, วิทยานิพนธ์จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ.2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้