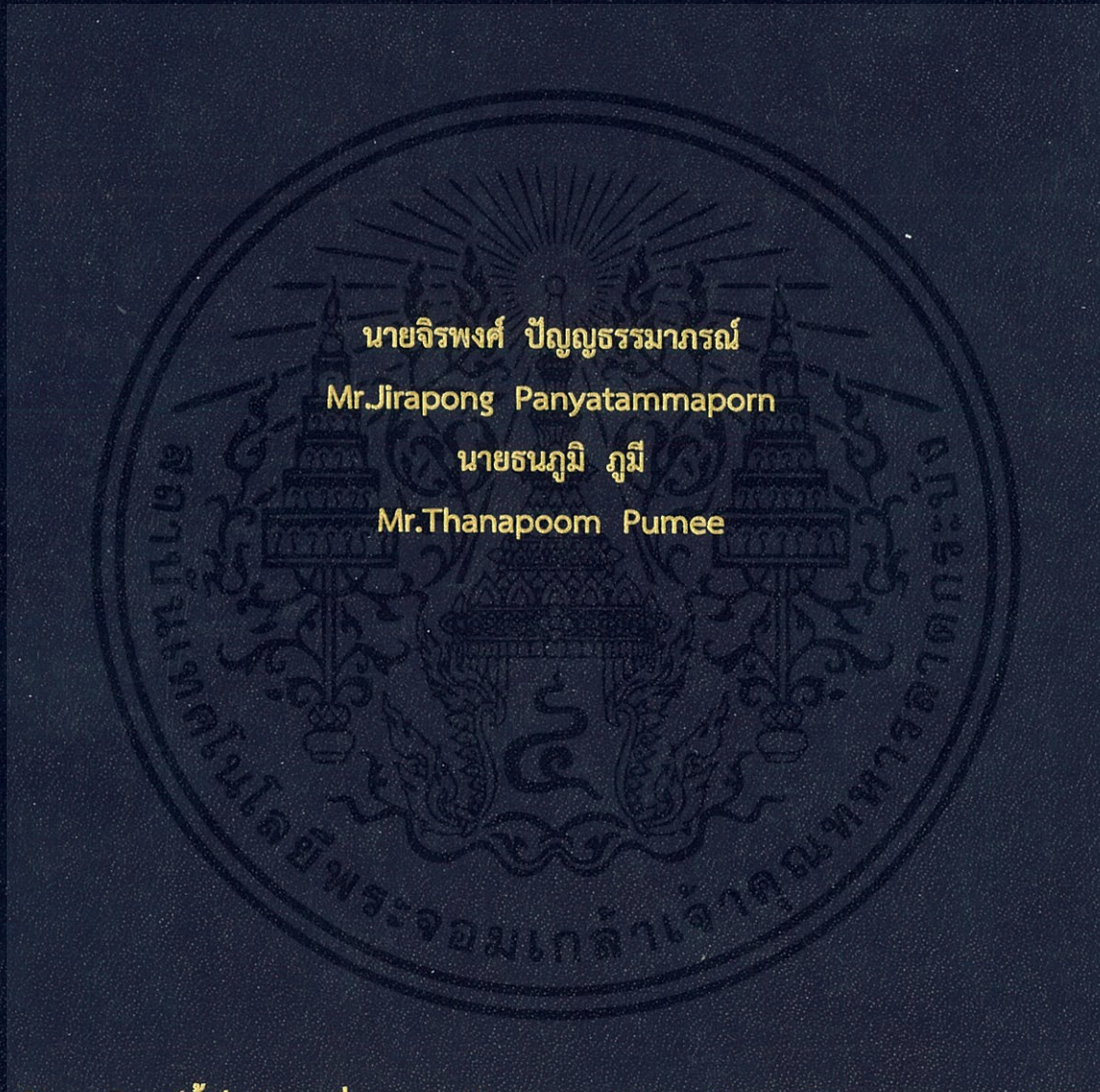


การจำลองเสียงเครื่องดนตรีดนตรีประเภทสาย

Surrogate Instrument



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

การจำลองเสียงเครื่องดนตรีดนตรีประเภทสาย

Surrogate Instrument



โดย

นาย จีรพงศ์ ปัญญธรรมาภรณ์ 58010183
นาย ธนภูมิ ภูมิ 58010511

อาจารย์ที่ปรึกษา
ผศ.ดร. ยุทธนา คัดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง การจำลองเสียงเครื่องดนตรีประเภทสาย

Surrogate Instrument

ผู้จัดทำ นายจิรพงศ์ ปัญญธรรมาภรณ์ 58010183

นายธนภูมิ ภูมิ 58010511

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว


(ผศ.ดร.ยuthนา คิติใจเดียว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การจำลองเสียงเครื่องดนตรีประเภทสาย		
นักศึกษา	นายจิรพงศ์	ปัญญาธรรมาภรณ์	58010183
	นายธนภูมิ	ภูมิ	58010511
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์		
ปีการศึกษา	2561		
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.ยุทธนา คิดใจเดียว		

บทคัดย่อ

วัตถุประสงค์ของโครงการนี้การสร้าง Equalizer ด้วยโครงข่ายประสาทเทียมแบบ Radial Basis Function Artificial ซึ่ง Equalizer จะทำให้การตอบสนองความถี่ของกีตาร์ที่มีคุณภาพต่ำใกล้เคียงกับการตอบสนองความถี่ของกีตาร์ที่มีคุณภาพสูง โครงข่ายประสาทเทียมใน Equalizer ซึ่งจะเรียนรู้จาก Amplitude ของข้อมูล และ เป้าหมายของการเรียนรู้คือ Amplitude ของการตอบสนองความถี่ของกีตาร์ที่มีคุณภาพดี และ Input ของโครงข่ายประสาทเทียมคือ Amplitude ของกีตาร์ที่มีคุณภาพต่ำ หลังจาก Training เสร็จเมื่อส่งสัญญาณผ่าน Equalizer และ กีตาร์คุณภาพต่ำ ตามทฤษฎีแล้วสัญญาณที่ออกมาจากระบบควรมีเอกลักษณ์ของสัญญาณกีตาร์คุณภาพสูง

Project Title	Surrogate Instrument	
Student	Mr. Jirapong	Panyatammaporn
	Mr. Thanapoom	Pumee
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2018	
Thesis Advisor	Assist.Prof.Dr. Yuttana kitjaidure	

ABSTRACT

The purpose of this project is equalizer creating with Radial Basis Function Artificial Neural Network. This equalizer adjusts balance between frequency response of Low quality guitars and High quality guitars. Artificial Neural Network in the equalizer is trained by amplitude. Target for training is amplitude response of High quality guitar and input is amplitude response of Low quality guitar. After finish training then feed signal through equalizer and then feed output of equalizer to Low quality guitar. Theoretically the identity of sound of Low quality guitar should be same with High quality guitar

กิตติกรรมประกาศ

ในการจัดทำโครงการเสร็จสมบูรณ์ได้นั้น ข้าพเจ้าขอขอบพระคุณท่านอาจารย์ที่ปรึกษา ผศ.ดร.ยุทธนา คิตใจเดียว ที่คอยชี้แนะให้ความรู้เกี่ยวกับโครงข่ายประสาทเทียม ให้คำแนะนำและช่วยแก้ไขปัญหาต่าง ๆ และ ชุมชุมโรบอท สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ให้ใช้สถานที่และอุปกรณ์ที่ใช้ในการทำ โครงการ อีกทั้งคณะอาจารย์ทุกท่านที่เกี่ยวข้องในการให้ความรู้แก่คณะผู้จัดทำและนำความรู้นั้นมาประยุกต์ใช้ในโครงการ ขอขอบพระคุณบิดามารดาที่ช่วยเหลือในเรื่องค่าใช้จ่าย ทำให้คณะผู้จัดทำสามารถทำโครงการนี้ให้ สำเร็จลุล่วง สุดท้ายนี้คณะผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจนำผลงานนี้ไปใช้งาน

จิรพงศ์ ปัญญธรรมภรณ์

ธนภูมิ ภูมิ



สารบัญ

หน้า

บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโครงงาน.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบข่ายโครงงาน.....	2
1.4 ผลที่คาดว่าจะได้รับ.....	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	
2.1 โครงสร้างและการตอบสนองที่แตกต่างของโครงสร้างกิตาร์.....	4
2.2 อีควอไลเซอร์ (Equalizer).....	7
2.3 วงจรขยายสัญญาณ.....	9
2.4 Artificial Neural Network.....	11
2.5 Radial Basis Function Neural Network.....	16
บทที่ 3 วิธีการดำเนินการ	
3.1 การเก็บข้อมูลเสียง.....	18
3.2 Zero Padding.....	19
3.3 Hanning Function.....	19

สารบัญ (ต่อ)

หน้า

3.4 Discrete Fourier Transform.....	21
3.5 Normalization.....	21
3.6 การปรับปรุงสัญญาณเสียงในโดเมนความถี่เพื่อให้เหมาะสม.....	22
บทที่ 4 ผลการทดลอง	
4.1 ผลการทดลองจากการเพิ่มข้อมูลเสียง.....	23
4.2 ผลจากการทดลองการเรียนรู้ด้วยโครงข่ายประสาทเทียม.....	24
บทที่ 5 สรุปและวิจารณ์	
5.1 ปัญหาที่เกิดขึ้น.....	29
5.1.1 ด้านฮาร์ดแวร์.....	29
5.1.2 ด้านฮาร์ดแวร์.....	29
5.2 สรุปผลการทดลอง.....	29
บรรณานุกรม.....	30
ภาคผนวก.....	31

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 บล็อกไดอะแกรม ของตอน Training Process.....	3
2.2 บล็อกไดอะแกรม ของตอน Feed Forward.....	3
2.3 โครงสร้างของ Bracing.....	4
2.4 ชนิดของ Bracing.....	5
2.5 ภาพของทราหยที่ตอบสนองความถี่บน Body Guitar.....	5
2.6 ภาพของการตอบสนองความถี่แต่ละโครงสร้าง.....	6
2.7 Equalizer.....	7
2.8 วงจรขยาย.....	9
2.9 LM3886TF.....	10
2.10 Artificial Neural Network.....	11
2.11 Artificial Neural Network.....	12
2.12 โครงสร้าง Artificial Neural Network.....	13
2.13 การทำงานของ Back-Propagation.....	13
2.14 Supervised Learning.....	14
2.15 Unsupervised Learning.....	15
2.16 Radial Basis Function Neural Network.....	16
2.17 การทำงานใน Hidden Layer.....	16
3.1 บล็อกไดอะแกรม ของตอน Training Process.....	18
3.2 ขั้นตอนการบันทึกการตอบสนองของความถี่กีตาร์ทั้งสอง.....	18

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.3 ภาพของสัญญาณข้อมูลหลังผ่านกิตาร์.....	19
3.4 ทำการ Zero Padding ข้อมูล.....	19
3.5 ข้อมูลเพลงทั้งหมด.....	19
3.6 Hanning Function.....	20
3.7 ข้อมูลเพลงเมื่อผ่านการ Hanning Function.....	20
3.8 ภาพข้อมูลเพลงเมื่อผ่าน Discrete Fourier Transform.....	21
3.9 ข้อมูลหลัง Normalization.....	21
4.1 ภาพของข้อมูลที่ได้รับมาปกติ.....	23
4.2 ข้อมูลเมื่อผ่าน Moving Average.....	24
4.3 บล็อกไดอะแกรม เมื่อตรวจผลการทดลอง.....	24
4.4 ภาพของข้อมูลปกติ.....	25
4.5 ภาพของการตอบสนองความถี่ของกิตาร์คุณภาพดี.....	25
4.6 ภาพของการตอบสนองความถี่ของกิตาร์คุณภาพต่ำ.....	26
4.7 ภาพของการตอบสนองความถี่ของกิตาร์คุณภาพต่ำเมื่อข้อมูลนั้นถูกปรับปรุงด้วย RBF.....	26

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

Soundboard guitar หรือ ไม้แผ่นหน้าของกีตาร์ คือส่วนประกอบส่วนหนึ่งของกีตาร์โปร่งซึ่งเป็นส่วนสำคัญที่จะขยายกำลังของคลื่นเสียงที่เกิดจากการสั่นสะเทือนของสายกีตาร์ ซึ่งสถาปัตยกรรมของ Soundboard guitar นั้นมีหลากหลายมากและแต่ละสถาปัตยกรรมนั้นมีการตอบสนองความถี่ที่แตกต่างกัน ทำให้ราคาของกีตาร์นั้นสูงต่ำแตกต่างกันไป ผู้จัดทำเล็งเห็นจุดสนใจตรงนี้จึงต้องการสร้างระบบที่สามารถปรับปรุงข้อมูลเสียงทำให้เสียงกีตาร์ที่มีคุณภาพต่ำมีเสียงคุณภาพใกล้เคียงกีตาร์คุณภาพสูง

ในปี 2015 Lamtharn Hantrakul ได้ทำโครงการเกี่ยวกับ Surrogate violin เพื่อที่ต้องการสร้างเสียงจำลองของไวโอลินให้สมจริงมากขึ้น โดยใช้การประมวลผลสัญญาณดิจิทัลแบบระบบควบคุมป้อนสัญญาณกลับ (Feedback Control Systems) และในปี 2016 Krittika Leartpantulak, Pongnarit Sundarasardula, Pongpinich Thamma นักศึกษาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้พยายามสร้าง Surrogate Guitar โดยการสร้างระบบกรองสัญญาณด้วยโครงข่ายประสาทเทียม ซึ่งผู้จัดทำมีความสนใจอยากศึกษาและปรับปรุงโครงการนี้ โดยการนำโครงข่ายประสาทเทียมสร้าง Equalizer ที่ได้เรียนรู้การตอบสนองความถี่ของ Soundboard guitar และลอกเลียนแบบเอกลักษณ์เฉพาะตัวของกีตาร์โปร่งที่มีราคาสูง และนำ Equalizer มาประมวลผลสัญญาณให้กีตาร์โปร่งราคาต่ำให้เสียงได้คล้ายกับกีตาร์โปร่งราคาสูง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

Equalizer ที่สร้างด้วย Radial basis function neural network เพื่อปรับ amplitude ของการตอบสนองความถี่ของกีตาร์คุณภาพต่ำให้มีการตอบสนองเทียบเท่ากับการตอบสนองความถี่ของกีตาร์คุณภาพสูงได้

1.3 สมมุติฐานของการศึกษา

เมื่อมีข้อมูลผ่านเข้า Equalizer ที่สร้างด้วย Radial basis function neural network นำสัญญาณขาออกของ Equalizer ผ่านวงจรขยายแล้วนำไปผ่านกีตาร์คุณภาพต่ำ การตอบสนองของความถี่ของกีตาร์คุณภาพต่ำจะใกล้เคียงการตอบสนองความถี่ของกีตาร์คุณภาพสูง

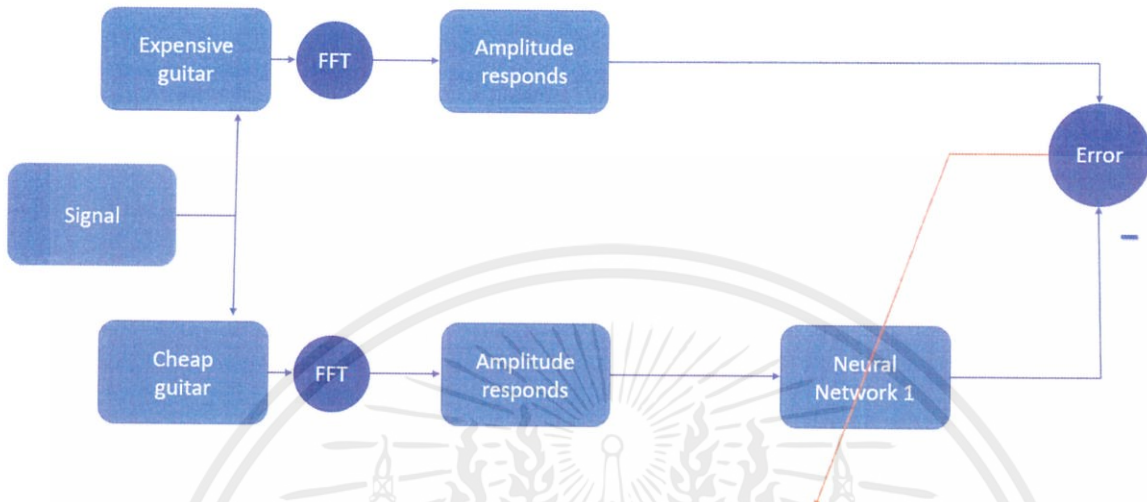
1.4 ขอบเขตการวิจัย

- จำลองการทำงานของ Equalizer ด้วยการ Simulator
- ปรับปรุง Equalizer เพื่อจำลองเสียงกีตาร์คุณภาพสูง
- ใช้ Artificial Neural Network ในการสร้าง Equalizer
- ออกแบบวงจรขยายและวงจรที่ใช้งานร่วมกับการทำงานของ Hardware

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ผลการทดลอง การจำลองโครงงานด้วยโปรแกรม simulator สามารถทำงานได้จริง
- เสียงที่ออกมาจากกีตาร์คุณภาพต่ำ เมื่อผ่าน Equalizer แล้ว มีคุณภาพเสียงใกล้เคียงกับกีตาร์คุณภาพสูง
- ได้รับความรู้เกี่ยวกับการเขียนโปรแกรม MATLAB
- สามารถเข้าใจการวิเคราะห์สัญญาณดิจิทัล เข้าใจอัลกอริทึมต่างๆที่นำมาใช้งาน เข้าใจการออกแบบขั้นตอนการทำงาน และวิธีการทำงานของ Artificial Neural Network ที่นำมาใช้
- สามารถอธิบายให้ผู้สนใจโครงงานเข้าใจความต้องการของผู้จัดทำได้

บทที่ 2 งานวิจัยที่เกี่ยวข้อง



รูปที่ 2.1 Block diagram ของตอน Training Process

จากรูปที่ 1 แสดงการทำงานของ การ Training Process ของ Artificial Neural Network ให้กับ Equalizer เริ่มต้นจากการนำไฟล์เพลงข้อมูลตัวอย่างไปเล่นผ่าน Sound exciter ให้กับกีตาร์คุณภาพดี หลังจากนั้นเก็บสัญญาณจากกีตาร์คุณภาพดี เพื่อนำไฟล์สัญญาณจากกีตาร์คุณภาพดี ไปเล่นผ่าน Equalizer และ Sound exciter ให้กับกีตาร์คุณภาพไม่ดี หลังจากนั้นนำสัญญาณผลต่าง หรือ ค่าความผิดพลาดไปปรับปรุงคุณภาพให้กับ Equalizer เพื่อให้สัญญาณผลต่างหรือค่าความผิดพลาดน้อยที่สุด

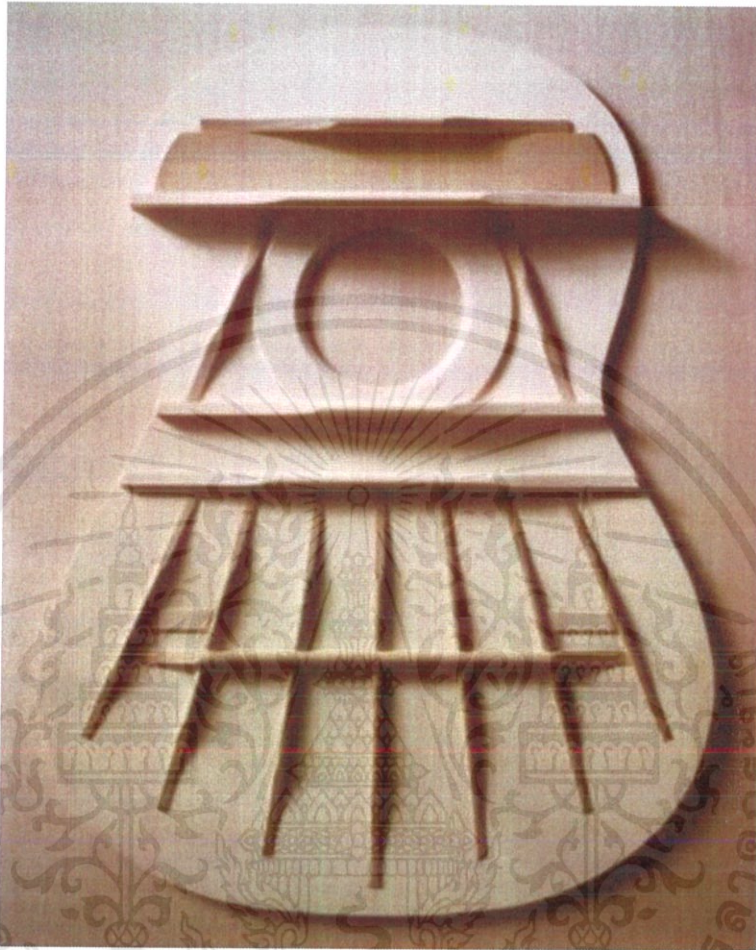


รูปที่ 2.2 Block diagram ของตอนใช้งาน

จากรูปที่ 2 แสดงถึงการใช้งานหลังจากที่ได้ Equalizer ที่ได้รับการปรับปรุงจากการ Training Process โดยการใช้งานจะเริ่มที่นำไฟล์เพลงข้อมูลทดลอง ส่งเข้า Equalizer หลังจากนั้นนำไปเล่นผ่าน Sound exciter ให้กับกีตาร์คุณภาพไม่ดีเก็บสัญญาณให้นำข้อมูลไปสรุปผลการทดลองว่าเสียงที่ได้รับนั้นมีความเหมือนกีตาร์คุณภาพดีมากน้อยแค่ไหน

2.1 โครงสร้างและการตอบสนองที่แตกต่างของโครงสร้างกีตาร์

เหตุที่ต้องเกริ่นถึงเรื่องนี้เนื่องจากเสียงที่แตกต่างของกีตาร์แต่ละตัวนั้นได้รับผลกระทบมาจากโครงสร้างของตัวมันเองและเสียงที่แตกต่างของกีตาร์แต่ละตัวนั้นเป็นที่มาและความสำคัญของโครงงานนี้

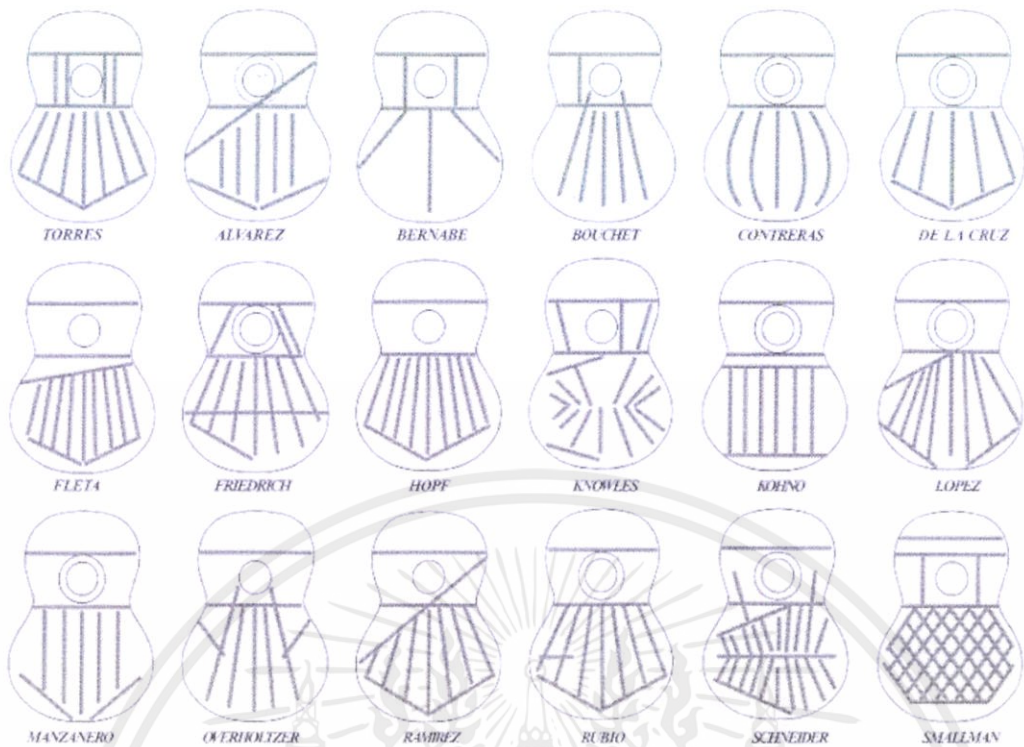


รูปที่ 2. 3 โครงสร้างของ bracing

2.1.1 Bracing Patterns หรือ รูปแบบโครงสร้างของกีตาร์ เป็นปัจจัยสำคัญหลักในการกำหนดทิศทางและคุณภาพของเสียง มีส่วนสำคัญหลักๆอยู่ 2 ประการ

- ช่วยเพิ่มความแข็งแรงให้กับไม้หน้า ไม้หลัง และ ไม้ข้าง
- ช่วยกำหนดลักษณะการสั่นสะท้อนของเสียงที่เปล่งออกมา

การนำ Bracing มาติดตั้งที่แผ่นไม้ของกีตาร์นั้นเริ่มต้นจากทางใต้ของสเปนช่วงปี 1759 ซึ่งในยุคแรกกีตาร์หรือเครื่องดนตรีส่วนใหญ่ยังใช้เพียงสายไนลอนซึ่ง Bracing ที่ถูกพัฒนามาจนเหมาะสมคือ Fan-Bracing และเมื่อเปลี่ยนมาถึงยุคกีตาร์สายโลหะ Fan-Bracing มีความแข็งแรงไม่เพียงพอและไม่สามารถรับกับแรงดึงของสายโลหะได้ดังนั้นจึงมีการคิดค้น Bracing อื่นๆมากมาย

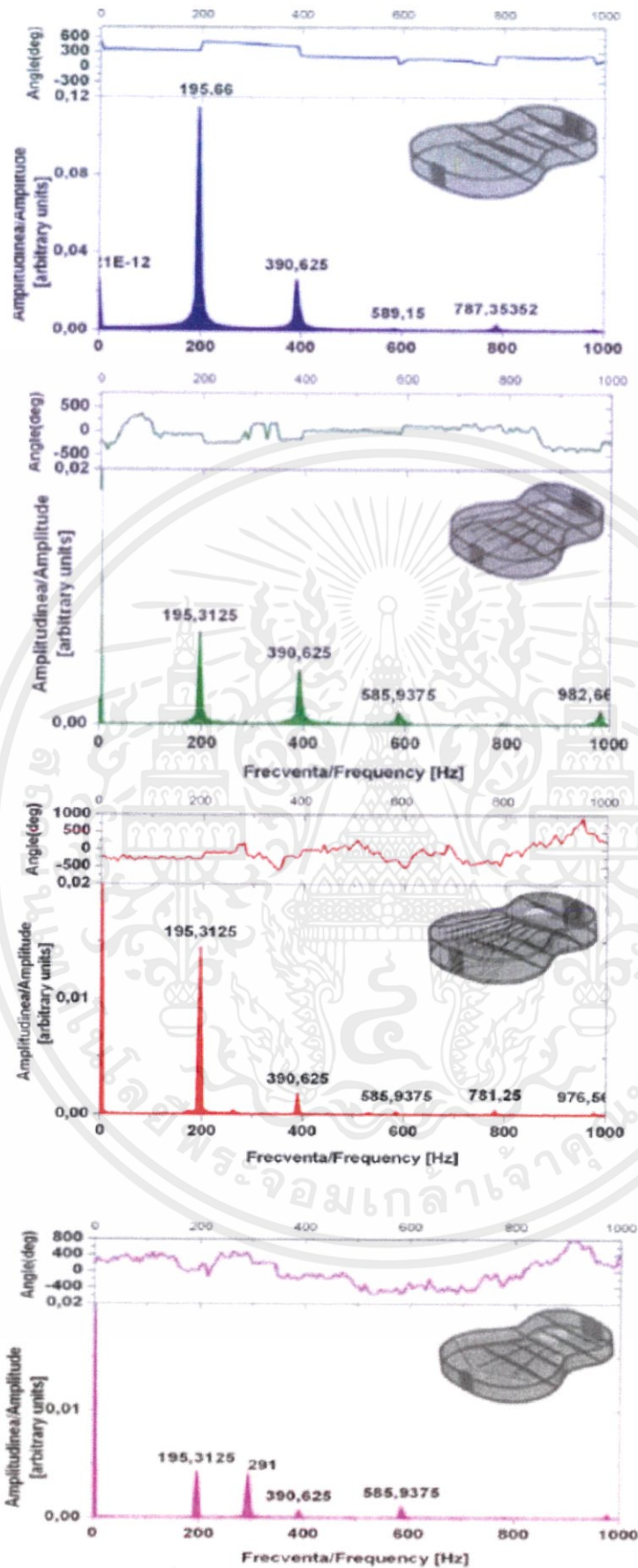


รูปที่ 2.4 Bracing แต่ละชนิด

2.1.2 การตอบสนองความถี่แต่ละ bracing เนื่องจากการคิดค้น bracing นั้นมีการคิดค้นมากมายไม่สามารถนับได้ภาพตัวอย่างต่อไปนี้จึงเป็นตัวอย่างที่คัดมาเฉพาะบางตัวของ bracing เพื่ออธิบายว่าสถาปัตยกรรมแต่ละอันนั้นให้การตอบสนองความถี่ที่แตกต่างกัน

Types of acoustic boxes	110 Hz	720 Hz
 Case 1		
 Case 2		
 Case 3		
 Case 4		

รูปที่ 2.5 ภาพของหลายที่ตอบสนองความถี่บน body guitar ที่ใช้ bracing ต่างกัน

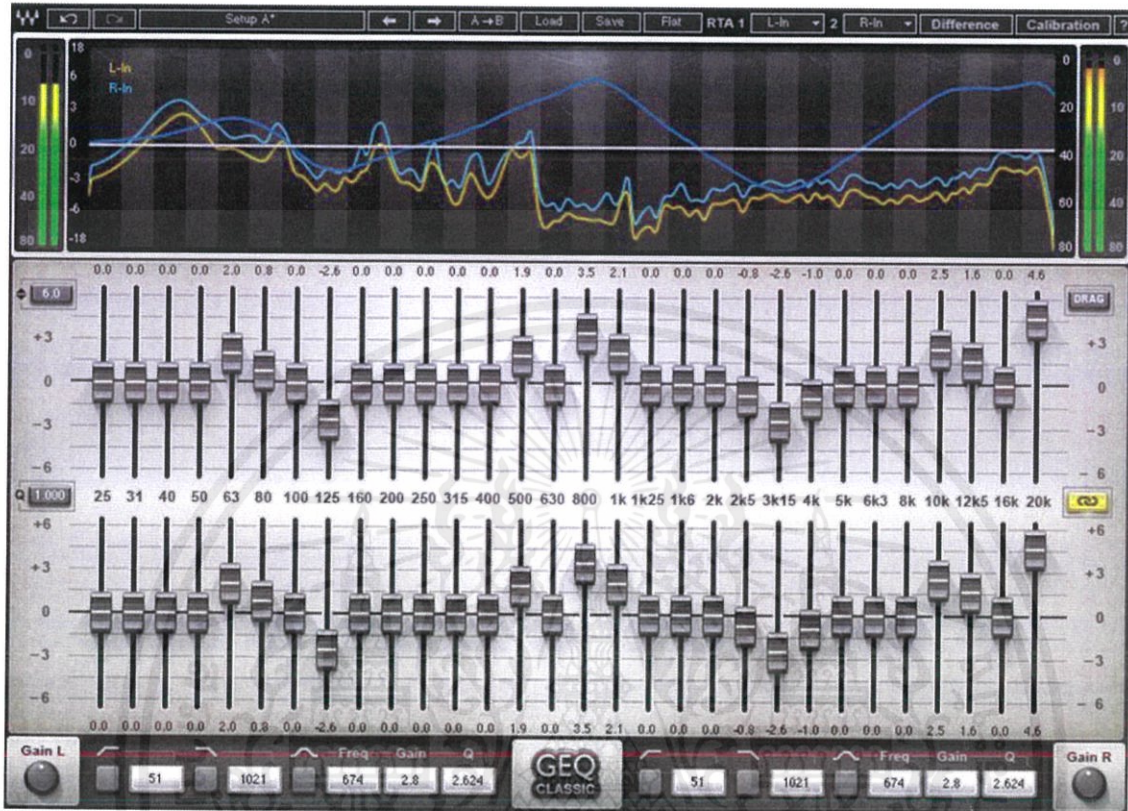


รูปที่ 2. 6 การตอบสนองความถี่แต่ละโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 อีควอลไลเซอร์ (Equalizer)

เนื่องจากโครงงานนี้แท้จริงแล้วคือการสร้าง Equalizer ด้วย Artificial Neural Network จึงจำเป็นอย่างมากที่จะกล่าวถึง Equalizer



รูปที่ 2.7 ภาพของ Equalizer

Equalizer เป็นเครื่องมือที่ทำหน้าที่ เพิ่มหรือลดระดับเสียงในช่วงความถี่ที่ต้องการ เอาไว้ปรับแต่งเสียงโดยตรง การทำงานของ Equalizer นั้นจะเป็นการปรับ เพิ่ม/ลด ความถี่ (Frequency) ของเสียง

Equalizer แบ่งออกเป็น 2 ชนิด

1. Graphic Equalizer ประเภทนี้ถูกออกแบบมาให้ตัดแบ่งย่านความถี่เฉพาะตามที่ถูกออกแบบกำหนดมาจากโรงงานโดยตรง ผู้ใช้งานไม่สามารถปรับเลื่อนค่าย่านความถี่ได้แต่อย่างใด ผู้ใช้งานมีหน้าที่เพิ่มหรือลด Gain ค่าขยายของย่านความถี่ที่ต้องการเท่านั้นเอง

Equalizer ประเภทนี้จึงเหมาะสำหรับการใช้งานควบคุมการกระจายเสียงของลำโพงเป็นส่วนใหญ่ เพื่อปรับปรุงอคูสติกของห้อง ซึ่งนิยมนำเอาไปใช้งานในงานระบบเสียง PA เป็นส่วนมาก หรือนำไปใช้ควบคุมลำโพงแบบขนาดใหญ่ในสตูดิโอเพื่อให้ได้เสียงที่ถูกต้อง เป็นต้น

Equalizer ประเภทนี้ จะแบ่งเป็นความถี่ส่วนใหญ่มีประมาณ 31 ความถี่ ตัวอย่างวิธีการปรับ ถ้าต้องการความถี่ที่ 500 Hz. จะปรับขึ้นหรือลงก็แล้วแต่สามารถ cut หรือ boost ที่ปุ่ม 500 Hz. ได้เลยโดยที่ความถี่ข้างเคียงไม่ว่าจะเกิน 500 Hz. หรือต่ำกว่า 500 Hz. สัญญาณความถี่ที่ cut หรือ boost จะไม่เกี่ยวข้อง

2.Parametric Equalizer ประเภทนี้ถูกออกแบบมาให้สามารถเลื่อนหาย่านความถี่ได้อย่างอิสระ ผู้ใช้งานเป็นผู้กำหนดย่านความถี่ด้วยตัวเอง

Equalizer ประเภทนี้จึงเหมาะสำหรับการใช้เพื่อปรับแต่งเสียงเฉพาะในแต่ละ channel บนมิกเซอร์ และเป็น Equalizer ประจำเครื่องของมิกเซอร์ทุกยี่ห้อ การนำไปใช้งาน Equalizer ประเภทนี้ เช่นไว้ใช้ปรับแต่งกับเสียงร้องหรือเสียงเครื่องดนตรี เป็นต้น

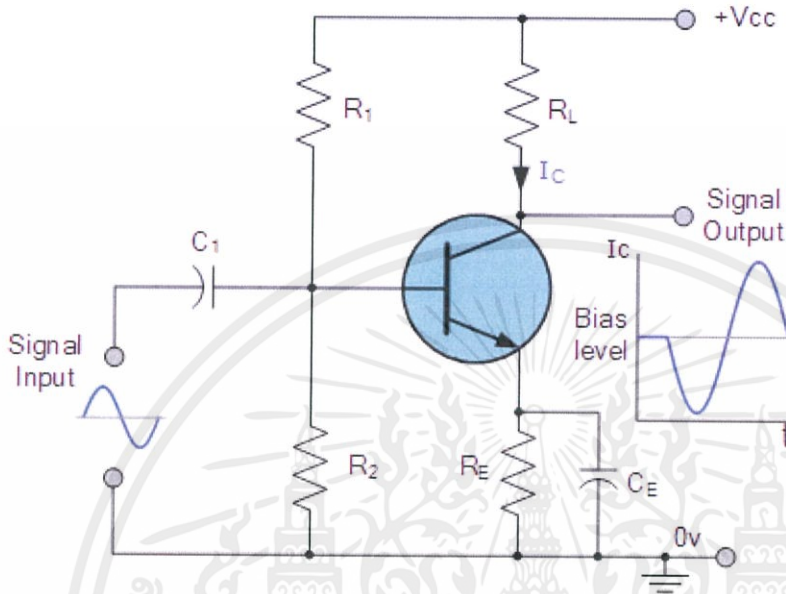
Equalizer ประเภทนี้ วิธีการปรับเหมือนกับ Graphic Equalizer เพียงแต่ความถี่ข้างเคียงจะขยับขึ้นตามตัวอย่าง ปรับความถี่ 500 Hz. ขึ้นในด้านความถี่สูงกว่า 500 Hz. หรือต่ำกว่า 500 Hz. จะขยับขึ้นตามไปด้วยเช่นกัน

ความแตกต่างของอีคิว(EQ) ทั้งสองแบบ อยู่ตรงที่ว่า Graphic Equalizer เป็นแบบกำหนดย่านความถี่ตายตัวมาจากโรงงาน ผู้ใช้งานมีหน้าที่เร่งหรือลด Gain ขยายของย่านความถี่ในแต่ละย่านที่มีมาให้ แต่ Parametric Equalizer เป็นอีคิว(EQ)ที่สามารถเลื่อนหาย่านความถี่ได้อย่างอิสระ ผู้ใช้งานเป็นผู้กำหนดย่านความถี่ด้วยตัวเอง การอีคิวแบบละเอียดสามารถทำได้

ฉะนั้น งานระบบเสียง PA จะนิยมใช้ Graphic Equalizer ใช้ปรับแต่งอคูสติคของระบบเป็นหลัก ส่วน Parametric Equalizer จะนิยมใช้เพื่อปรับแต่งเสียงเฉพาะในแต่ละ channel บนมิกเซอร์ อย่างเช่นใช้ปรับแต่งกับเสียงร้องหรือเสียงเครื่องดนตรี

2.3 วงจรขยายสัญญาณ (Amplifiers)

หลังจากได้ผลลัพธ์จาก Equalizer ก็ต้องทำการขยายสัญญาณเพื่อนำไปขับ Sound Excitor เพื่อพิสูจน์ผลการทดลองต่อไป



รูปที่ 2.8 วงจรขยาย

ในหนังสือวิชา Electronics Engineering 1 บทที่สอง นิยามวงจขยาย (Amplifiers) ไว้ว่า วงจรที่ใช้เพิ่มหรือขยายกำลังไฟฟ้าของสัญญาณ ในการพิจารณาการทำงานของวงจขยายเบื้องต้น มักจะพิจารณาวงจขยายทำหน้าที่เพิ่มขนาดแรงดันไฟฟ้าของสัญญาณหรือกล่าวอีกนัยหนึ่งว่า พิจารณาการทำงานของวงจขยายเป็นลักษณะหรือรูปแบบของวงจขยายแรงดัน (Voltage Amplifiers) ในความเป็นจริงการที่แรงดันไฟฟ้ามีการเปลี่ยนแปลง ย่อมทำให้กระแสไฟฟ้าเปลี่ยนแปลงด้วยวงจขยายอาจถูกพิจารณาการทำงานในรูปแบบของวงจขยายกระแส (Current Amplifiers) ก็ได้ หรือรูปแบบอื่น วงจขยายอาจจะให้ค่าแรงดันไฟฟ้าที่ทางออกมีขนาดต่ำกว่าทางเข้า แต่ค่ากำลังไฟฟ้าที่ออกนั้นยังคงมีค่าสูงกว่ากำลังไฟฟ้าที่ทางเข้าก็สามารถจะเป็นไปได้ ทั้งนี้เนื่องจากค่ากำลังไฟฟ้ายังขึ้นอยู่กับความต้านทานของวงจรอีกด้วย

2.3.1 วงจขยายรูปแบบต่างๆ วงจขยายสามารถที่จะแบ่งการทำงานของวงจขยายเป็นโหมด (Mode) หรือรูปแบบการทำงานได้จากความสัมพันธ์ของสัญญาณที่ทางเข้าและทางออก สมการที่แสดงถึงความสัมพันธ์ระหว่างสัญญาณที่ทางเข้าและทางออกเรียกว่าฟังก์ชันถ่ายโอน (Transfer Function) หรือ ฟังก์ชันการขยาย (Gain Function) หรืออาจจะเรียกสั้นๆว่าอัตราขยาย (Gain)

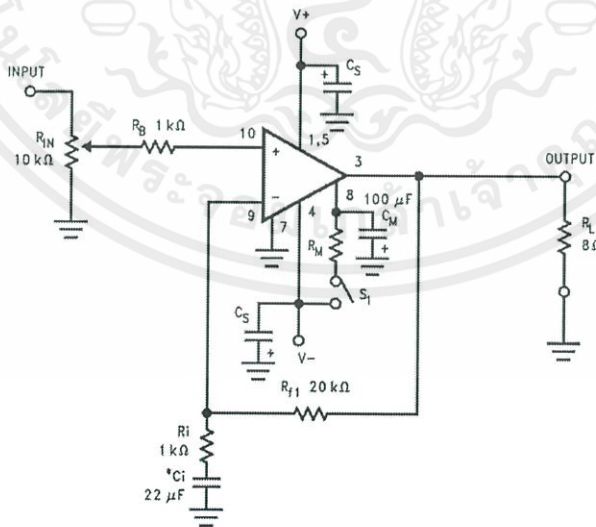
$$\text{อัตราขยาย (Gain)} = \frac{\text{สัญญาณที่ทางออก}}{\text{สัญญาณที่ทางเข้า}} = \frac{x(out)}{x(in)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรขยายที่ติจะต้งให้สัญญาณที่ทางออกมีรูปร่างเหมือนสัญญาณที่ทางเข้า หรือไม่มีความผิดเพี้ยน สมการความสัมพันธ์ระหว่างสัญญาณที่ทางออกกับสัญญาณที่ทางเข้าจะต้งอยู่ในรูปของสมการเส้นตรง อาจถูกเรียกว่า วงจรรขยายเชิงเส้น (Linear Amplifier)

ชนิดของวงจรรขยาย	สัญญาณเข้า x (in)	สัญญาณออก x (out)	ค่าอัตราการขยาย(หน่วย)
วงจรรขยายแรงดัน	V (in)	V (out)	V(out)/V(in)
Voltage Amplifiers			อัตราการขยายแรงดัน
วงจรรขยายกระแส	I (in)	I (out)	I(out)/I(in)
Current Amplifier			อัตราการขยายกระแส
วงจรรขยายความนำ	V (in)	I (out)	I(out)/V(in)
Transconductance			ความนำโอนย้าย (ซีเมนส์)
วงจรรขยายความต้านทาน	I (in)	V (out)	V(out)/I(in)
Transresistance			ความต้านทานโอนย้าย (โอห์ม)

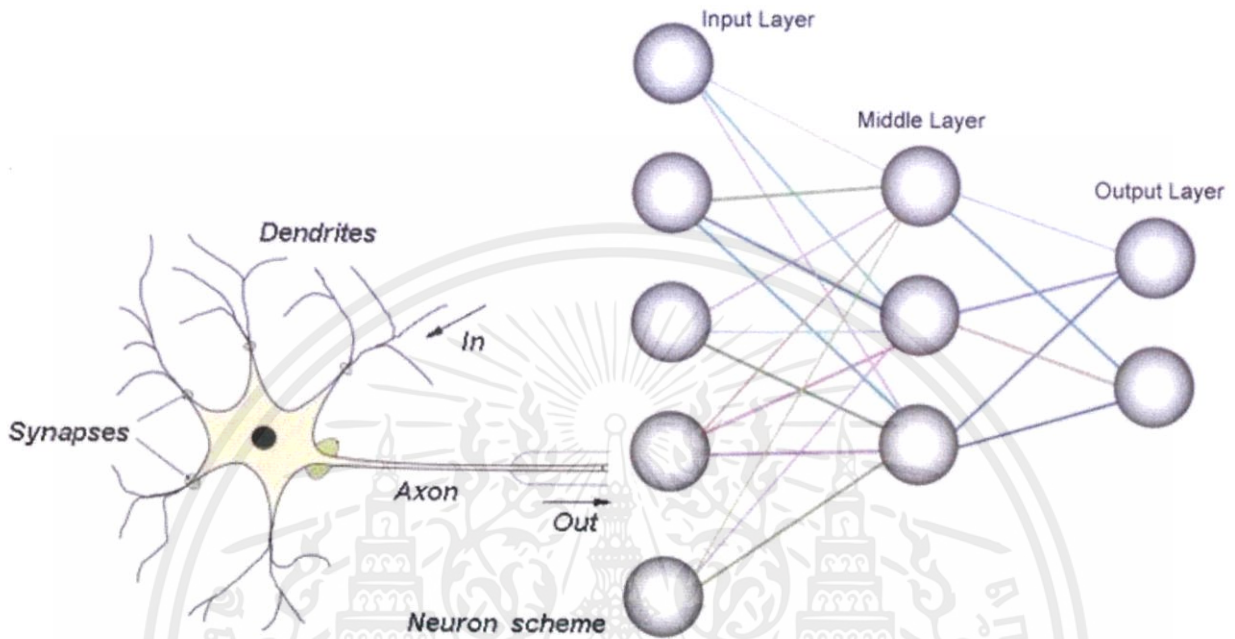
ซึ่งในโครงการนี้ได้หยิบวงจรรวม LM3886TF มาใช้เลยซึ่งมีการต่อวงจรดังนี้



รูปที่ 2.9 LM3886TF

2.4 Artificial Neural Network คืออะไร

Artificial Neural Network หรือชื่อภาษาไทยคือ โครงข่ายประสาทเทียม คือโครงข่ายที่เลียนแบบการทำงานของเซลล์ประสาทในสิ่งมีชีวิต โดยโครงข่ายจะประกอบด้วยเซลล์ประสาทเทียมจำนวนมาก ซึ่งแต่ละเซลล์จะต่อกัน เหมือนๆกับเซลล์ประสาทของสิ่งมีชีวิต



2.10 Neural Network และ Artificial Neural Network

โดยในธรรมชาติแล้วสมองของมนุษย์จะประกอบด้วยเซลล์ประสาทจำนวน 84 พันล้านเซลล์โดยแต่ละเซลล์จะเชื่อมต่อกันด้วย axon เมื่อมีสิ่งเร้าจากสิ่งแวดล้อมเกิดขึ้นกับอวัยวะต่างๆในร่างกาย ก็จะเกิดสัญญาณไฟฟ้าขึ้นแล้วส่งผ่านไปผ่านเซลล์ประสาทแต่ละตัวเพื่อตอบสนองต่อสภาวะนั้นๆ

สิ่งนี้เกิดขึ้นกับโครงข่ายประสาทเทียมเช่นกัน โดยที่สิ่งเร้าที่นั่นคืออินพุตของโครงข่ายและจะแพร่ผ่านเซลล์ประสาทเทียมทุกตัวในโครงข่ายนั้นๆ โดยเซลล์ประสาทแต่ละตัวจะตอบสนองต่ออินพุตที่แตกต่างกัน เช่นเดียวกับโครงข่ายประสาทของสิ่งมีชีวิต นอกจากนี้ ยังสามารถรับและจดจำสารสนเทศในรูปแบบที่เป็นประสบการณ์ได้ ทำให้สามารถ เชื่อมโยงข้อเท็จจริงทั้งหลายเข้าด้วยกันเพื่อหาข้อสรุป และใช้ประสบการณ์ที่จัดเก็บไว้มาเรียนรู้และทำความเข้าใจว่า ข้อเท็จจริงใหม่ที่ได้รับเข้ามามีความเกี่ยวข้องกันอย่างไร เพื่อทำการปรับปรุงองค์ความรู้ให้ มีความทันสมัยเพื่อประโยชน์ในอนาคต

2.4.1 องค์ประกอบและโครงสร้างการทำงาน

1.) ข้อมูลนำเข้า (Input) ข้อมูลนำเข้าจะถูกจำแนกตามคุณลักษณะ (Attribute) เช่น ถ้าปัญหาที่ระบบประสาทเสมือนจะต้องตัดสินใจคือ การอนุมัติเงินกู้ว่าจะให้ผ่านหรือไม่ ข้อมูล นำเข้า ก็จะถูกจำแนกเป็นคุณลักษณะ กล่าวคือ ระดับรายได้ และอายุ เป็นต้น ข้อมูลนำเข้า นอกจากจะเป็นข้อความแล้ว ยังสามารถเป็นรูปภาพ หรือเสียงก็ได้ แต่อาจจะต้องผ่านการแปลง ให้เป็นสัญลักษณ์หรือตัวเลขเพื่อให้เครื่องสามารถทำความเข้าใจได้ก่อน จากนั้นก็จะเข้าสู่การ ทำงานที่แท้จริงของระบบ

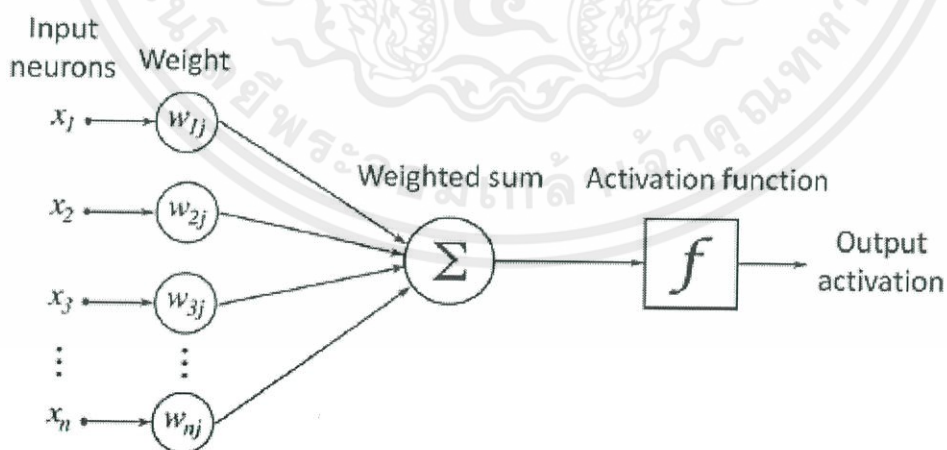
ประสาทเสมือนที่เริ่มต้นด้วยการนำข้อมูลเข้ามาให้น้ำหนัก (weight) ของความสัมพันธ์ระหว่างข้อมูล นำเข้าเหล่านั้นใน เลเยอร์แรกภายใต้ขอบเขตของระบบ

2.) น้ำหนัก (Weight) เป็นส่วนประกอบที่สำคัญของระบบโครงข่ายประสาท เนื่องจากเป็นส่วนที่ใช้หาน้ำหนักของความสัมพันธ์ระหว่างข้อมูลนำเข้า ว่าข้อมูลนำเข้าใดมีความสัมพันธ์กับข้อมูลนำเข้าอื่นในระดับใด ซึ่งจะทำให้สามารถเชื่อมโยงไปหาข้อสรุปได้ด้วย การลองผิดลองถูกในความสัมพันธ์แต่ละแบบ และเก็บไว้เป็นแบบแผนหรือรูปแบบ (pattern) ของประสบการณ์เพื่อการเรียนรู้ของโครงข่าย

3.) ฟังก์ชันการรวม (Summation Function) เป็นโครงข่ายที่ทำหน้าที่ในการรวมค่า น้ำหนัก ที่ได้จากโครงข่ายในเลเยอร์ input เพื่อสรุปผลความสัมพันธ์ระหว่างข้อมูลนำเข้า รอ การแปลงเป็นสารสนเทศที่มีความหมายในเลเยอร์ต่อไป

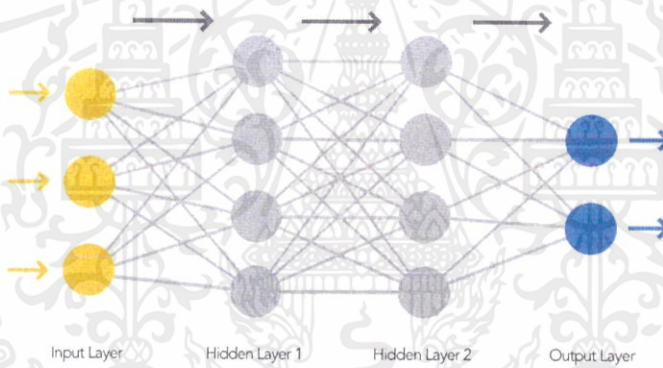
4.) ฟังก์ชันการแปลง (Transformation Function) เป็นโครงข่ายที่ทำหน้าที่ในการ ประสาน (integrate) สารสนเทศที่ผ่านการประมวลผลจากโครงข่ายในเลเยอร์ต่างๆแล้วทำการ แปลง (Transform) ให้กลายเป็นสารสนเทศที่สื่อความหมาย และเป็นประโยชน์ต่อการนำไปใช้ ได้เพื่อส่งออกไปเป็นผลลัพธ์ (Output)

5.) ผลลัพธ์ (Output) ผลลัพธ์ที่ได้จากโครงข่ายประสาทเสมือน จะหมายถึงแนวทาง ในการแก้ไขปัญหา เช่น ปัญหาการอนุมัติเงินกู้ว่าผู้กู้จะผ่านการอนุมัติหรือไม่ “ผลลัพธ์” ที่ผู้ใช้จะได้รับคือ “อนุมัติ” หรือ “ไม่อนุมัติ” ซึ่งโครงข่ายประสาทเสมือนจะใช้สัญลักษณ์แทน คำตอบทั้งหมด ผลลัพธ์ที่ได้จากโครงข่ายหนึ่ง สามารถเป็นข้อมูลนำเข้า (input) ของอีกโครงข่ายหนึ่งได้ ทั้งนี้ เพื่อเป็นข้อมูลนำเข้าของการตัดสินใจแก้ไขปัญหาอื่น เช่น ผลลัพธ์ที่ได้จากการอนุมัติเงินกู้ อาจจะนำไปใช้เป็นข้อมูลนำเข้าเพื่อการอนุมัติสินเชื่อที่อยู่อาศัยได้



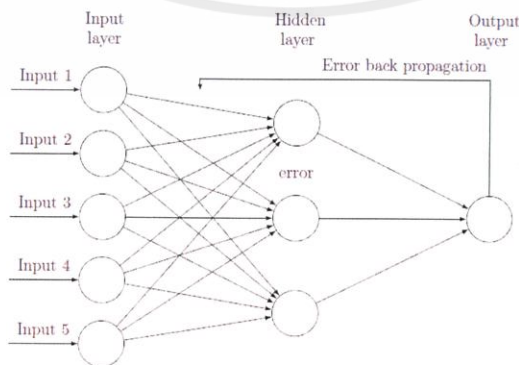
รูปที่ 2.11 ภาพของ Neural Network

2.4.2 หลักการทำงาน สำหรับในคอมพิวเตอร์ Neurons ประกอบด้วย input และ output เหมือนกัน โดย จำลองให้ input แต่ละอันมี weight เป็นตัวกำหนดน้ำหนักของ input โดย neuron แต่ละ หน่วยจะมีค่า threshold เป็นตัวกำหนดว่าน้ำหนักรวมของ input ต้องมากขนาดไหนจึงจะ สามารถส่ง output ไปยัง neurons ตัวอื่นได้ เมื่อนำ neuron แต่ละหน่วยมาต่อกันให้ทำงาน ร่วมกันการทำงานนี้ในทางตรรกแล้วก็จะ เหมือนกับปฏิกิริยาเคมีที่เกิดในสมอง เพียงแต่ใน คอมพิวเตอร์ทุกอย่างเป็นตัวเลขเท่านั้นเอง สิ่งสำคัญคือเรา ต้องทราบค่า weight และ threshold สำหรับสิ่งที่เราต้องการเพื่อให้ คอมพิวเตอร์รู้จัก ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้ โดยการสอนให้มันรู้จัก pattern ของสิ่งที่เราต้องการให้ มันรู้จัก เรียกว่า "back propagation" ซึ่งเป็นกระบวนการย้อนกลับของการรู้จัก ในการฝึก feed-forward neural networks จะมี การใช้อัลกอริทึมแบบ back-propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของ เครือข่าย (network weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ ได้รับ (output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่า ความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนัก คะแนนต่อไป



รูปที่ 2.12 แสดงโครงสร้าง neural network

Back-propagation เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครือข่ายประสาทวิธึหนึ่งที นิยมใช้ใน multilayer perceptron เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้เหมาะสม โดยการปรับค่านี้อจะขึ้นกับความแตกต่างของค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ



รูปที่ 2.13 การทำงาน Back-Propagation

ขั้นตอนของ Back-propagation Algorithm มีดังนี้

1. กำหนดค่าอัตราเร็วในการเรียนรู้ (rate parameter: r)
2. สำหรับแต่ละตัวอย่างอินพุตให้ทำตามขั้นตอนต่อไปนี้จนกว่าได้ระดับ performance ที่ต้องการ
 - คำนวณค่าเอาต์พุตโดยใช้ค่าน้ำหนักเริ่มต้นซึ่งอาจได้จากการสุ่ม
 - คำนวณค่า β : แทนประโยชน์ที่จะได้รับสำหรับการเปลี่ยนค่าเอาต์พุตของแต่ละโหนด
 - ในชั้นเอาต์พุต (Output Layer)

$$\beta z = dz - oz \quad (2.1)$$

เมื่อ dz = ค่าเอาต์พุตที่ต้องการ

oz = ค่าเอาต์พุตที่คำนวณได้

- ในชั้นซ่อน (Hidden Layer)

$$\beta_j = W_{ij} \times O_k \times (1 - O_k) \beta_k \quad (2.2)$$

เมื่อ W_{ij} = น้ำหนักของเส้นเชื่อมระหว่างชั้นที่ j กับ k

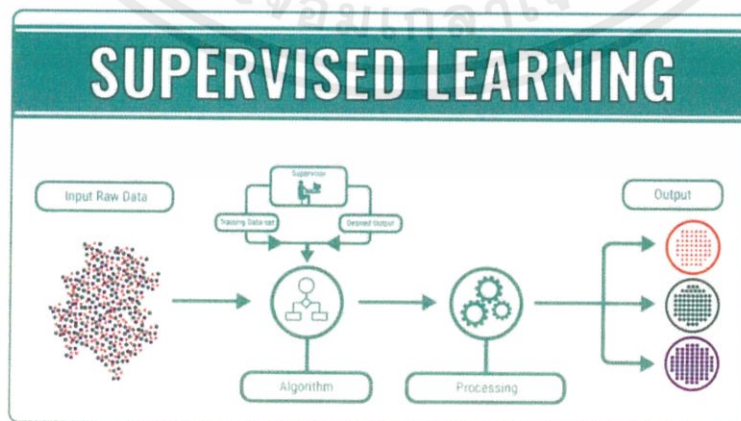
- คำนวณค่าน้ำหนักที่เปลี่ยนแปลงไปสำหรับในทุกน้ำหนัก ด้วยสมการต่อไปนี้

$$\Delta W_{ij} = r \times o_i \times o_j (1 - o_j) \times \beta_j \quad (2.3)$$

- เพิ่มค่าน้ำหนักที่เปลี่ยนแปลง สำหรับตัวอย่างอินพุตทั้งหมด และเปลี่ยนค่าน้ำหนัก

2.4.3 การเรียนรู้สำหรับโครงข่ายประสาทเทียม

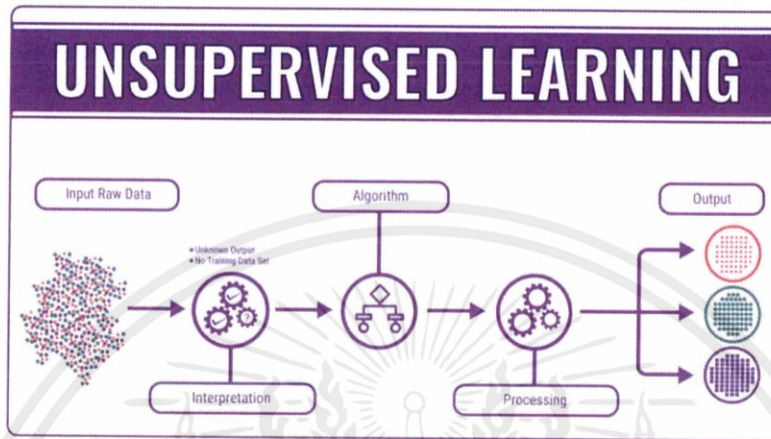
1) Supervised Learning การเรียนแบบมีการสอน เป็นการเรียนที่มีการตรวจคำตอบเพื่อให้วงจรข่ายปรับตัวชุดข้อมูลที่ใช้สอนวงจรข่ายจะมีคำตอบไว้คอยตรวจสอบดูว่าวงจรข่ายให้คำตอบที่ถูกหรือไม่ถ้าตอบไม่ถูกวงจรข่ายจะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้น (เปรียบเทียบกับคน เหมือนกับการสอนนักเรียนโดยมีครูผู้สอนคอยแนะนำ)



รูปที่ 2.14 Supervised Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Unsupervised Learning การเรียนแบบไม่มีการสอนเป็นการเรียนแบบไม่มีผู้แนะนำไม่มีการตรวจสอบคำตอบถูกหรือผิด วงจรข่ายจะจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล ผลลัพธ์ที่ได้วงจข่ายจะสามารถจัดหมวดหมู่ของข้อมูลได้(เปรียบเทียบกับคนเช่นการที่เราสามารถแยกแยะพันธุ์พืช พันธุ์สัตว์ตามรูปร่างลักษณะของมันได้เองโดยไม่มีใครสอน)

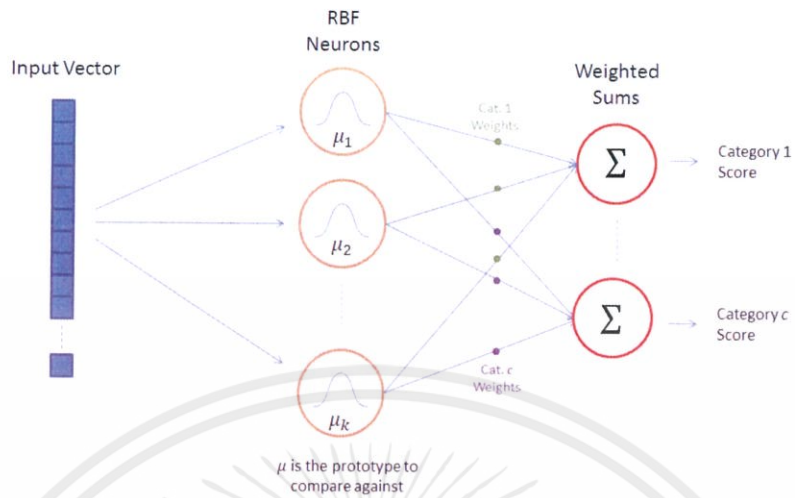


รูปที่ 2.15 Unsupervised Learning

2.4.4 ประโยชน์ของโครงข่ายประสาทเสมือน

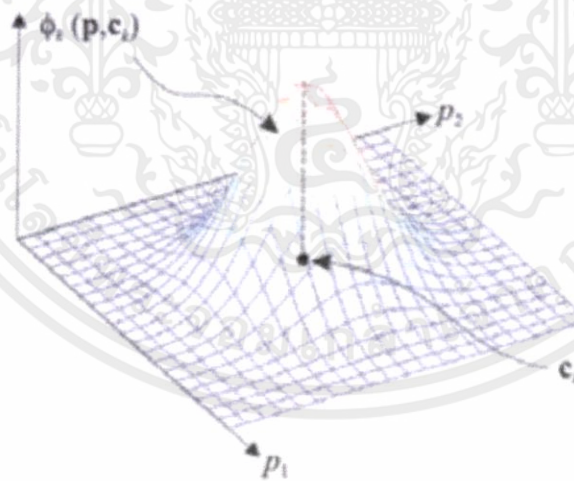
- 1.) เกิดข้อผิดพลาดได้ยาก (Fault Tolerance) หากระบบโครงข่ายประสาทเสมือน ประกอบไปด้วยโครงข่ายที่ใช้ในการประมวลผลมากมายหลายโครงข่าย ความผิดพลาดที่เกิดขึ้น จากเพียงหนึ่งหรือสองโครงข่ายจะไม่ทำให้ทั้งระบบเกิดข้อผิดพลาดได้
- 2.) ความสามารถในการหาเหตุผล (Generalization) เมื่อระบบโครงข่ายประสาทเสมือนได้รับข้อมูลนำเข้าที่ไม่ครบถ้วนหรือไม่เพียงพอต่อการหาข้อสรุป หรือได้รับข้อเท็จจริงที่ไม่เคยได้รับมาก่อน ระบบจะสามารถลำดับการเชื่อมโยงข้อเท็จจริงจนสามารถหาข้อสรุปและ เหตุผลได้
- 3.) ความสามารถในการปรับเปลี่ยน (Adaptability) โครงข่ายประสาทเสมือน สามารถเรียนรู้สภาพแวดล้อมใหม่ได้ ดังนั้นเมื่อมีเหตุการณ์ใหม่ๆ เข้าสู่ระบบก็จะสามารถ ปรับเปลี่ยนหรือปรับปรุงองค์ความรู้ให้ทันสมัยตามเหตุการณ์ใหม่นั้น
- 4.) ความสามารถในการพยากรณ์ (forecasting Capability) โครงข่ายประสาทเสมือนสามารถนำข้อมูลทางสถิติเดิมที่มีอยู่ในระบบ มาใช้คาดการณ์หรือพยากรณ์ข้อมูลในอนาคตได้

2.5 Radial Basis Function Neural Network



รูปที่ 2.16 Radial Basis Function Neural Network

โครงข่ายเรเดียลเบซิสฟังก์ชัน (Radial Basis Function Network หรือ RBF Network) เป็นโครงข่ายแบบ Feed Forward Network ประเภทหนึ่งที่ได้รับการยอมรับว่ามีประสิทธิภาพสูงโครงข่ายหนึ่ง RBF นั้นมี Hidden Layer เพียงแค่ 1 ชั้น



รูปที่ 2.17 การทำงานใน Hidden Layer

สำหรับแต่ละนิวรอนในชั้นซ่อน ค่าระยะทางยูคลิดระหว่างเวกเตอร์จุดศูนย์กลาง C_k กับเวกเตอร์อินพุต p จะถูกคำนวณ เอาต์พุตของนิวรอนใน Hidden Layer นี้จะได้จากฟังก์ชัน $\phi_k(\cdot)$ ซึ่งเป็นฟังก์ชันแบบไม่เชิงเส้นสุดท้ายแล้วเอาต์พุตของโครงข่ายจะได้จากผลรวมของค่าน้ำหนักประสาทกับเอาต์พุตของนิวรอน

จาก Hidden Layer ตัวอย่างของฟังก์ชัน $\phi_k(\cdot)$ ที่ใช้ในโครงข่าย RBF เช่นฟังก์ชันเกาส์เซียน (Gaussian Function)

$$\text{Gaussian Function RBF} \quad \phi(p) = e^{-(p\sigma)^2}$$

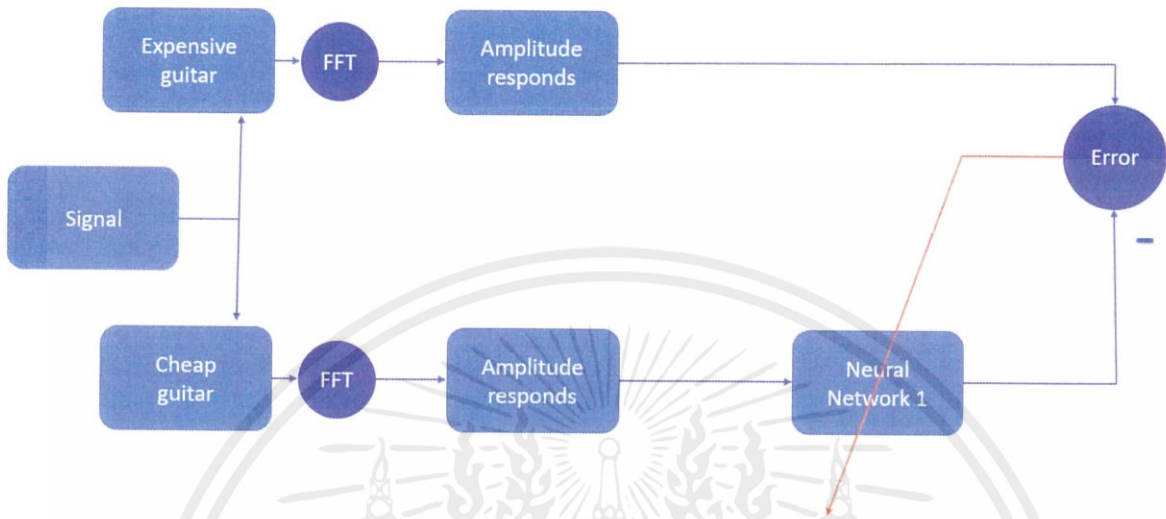
โดยฟังก์ชันเกาส์เซียนเป็นฟังก์ชันที่นิยมใช้ใน RBF มากที่สุดซึ่งรูปร่างของฟังก์ชันเกาส์เซียนแสดงได้ดังรูปที่ 3 โดยที่พารามิเตอร์ σ เป็นตัวควบคุมความกว้างของ RBF หรือเรียกว่าพารามิเตอร์การกระจาย (spread parameter)

เวกเตอร์ศูนย์กลาง C_k ของ Hidden Layer ที่ k จะรับอินพุตจากเวกเตอร์ p ที่มีมิติเท่ากับ N พารามิเตอร์ σ_k ทำหน้าที่ควบคุมความกว้างของแต่ละ RBF โดยปกติแล้ว ถ้าเวกเตอร์อินพุต p มีระยะห่างจากจุดศูนย์กลาง C_k มากขึ้น กล่าวคือ $\|p - c_k\|$ มีค่ามากขึ้น ค่าที่ได้จากฟังก์ชัน ϕ_k จะลดลง ดังรูปที่ 3 โดยพื้นที่ของฟังก์ชัน ϕ เรียกว่าสนามรับ (receptive field) ของนิวรอนนั้นๆ ϕ จากทุก Hidden Layer โดยปกติแล้วเวกเตอร์จุดศูนย์กลาง C_k จะถูกเลือกจากปริภูมิของเวกเตอร์อินพุต ที่ซึ่งจะต้องมีจำนวนเวกเตอร์จุดศูนย์กลาง (หรือจำนวนนิวรอนในชั้นซ่อน) เพียงพอและครอบคลุมปริภูมิของอินพุตได้

2.5.2 การฝึกสอนโครงข่าย RBF การฝึกสอนโครงข่าย RBF นั้นได้มีผู้นำเสนอไวหลายรูปแบบ ในลักษณะเดียวกันกับการฝึกสอนโครงข่ายประสาทเทียมแบบอื่น ๆ การฝึกสอน เครือข่าย RBF ก็คือการค้นหาพารามิเตอร์ของเครือข่ายซึ่งประกอบไปด้วย

- ค่า W
- เวกเตอร์จุดศูนย์กลางของ Hidden Layer
- ค่าพารามิเตอร์การกระจาย

บทที่ 3 วิธีการดำเนินการ



รูป 3.1 บล็อกไดอะแกรมของโครงงานช่วง Training Process

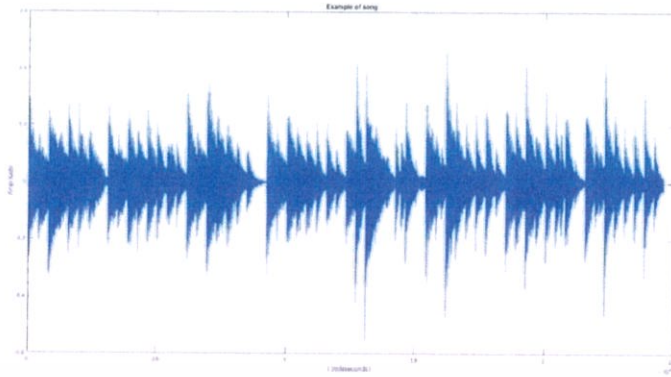
จากรูปที่ 3.1 วิธีดำเนินการแรกที่จะต้องทำคือการเก็บข้อมูลของการตอบของเสียงของกีตาร์ทั้งสองตัว

3.1 การเก็บข้อมูลเสียงก่อนนำไป Training Process



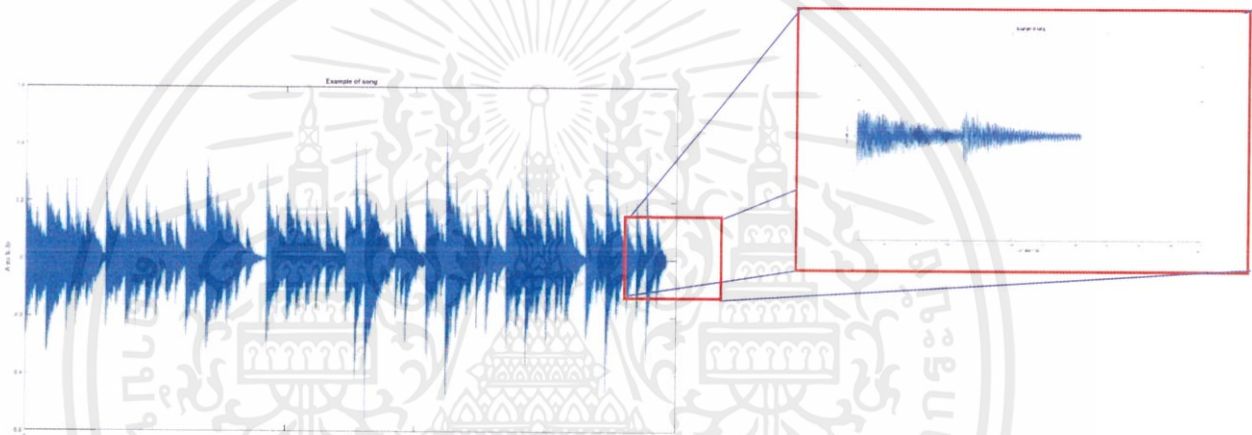
รูปที่ 3.2 ขั้นตอนการบันทึกการตอบสนองความถี่ของกีตาร์ทั้งสอง

โดยการนำคอมพิวเตอร์เครื่องที่ 1 เล่นไฟล์เพลงขับเพลงด้วย Sound exciter ใส่บริเวณ Bridge ของกีตาร์ และนำคอมพิวเตอร์อีกเครื่องบันทึกการตอบสนองของกีตาร์ด้วย Piezo pickup เพื่อเป็นการเก็บข้อมูลการตอบสนองความถี่ของกีตาร์ทั้งสอง ซึ่งต้องทำการเก็บข้อมูลเพลงหลายๆเพลงเพื่อเตรียมตัวนำไปใช้งานต่อไป



รูปที่3.3 ภาพของสัญญาณข้อมูลหลังผ่านกีตาร์

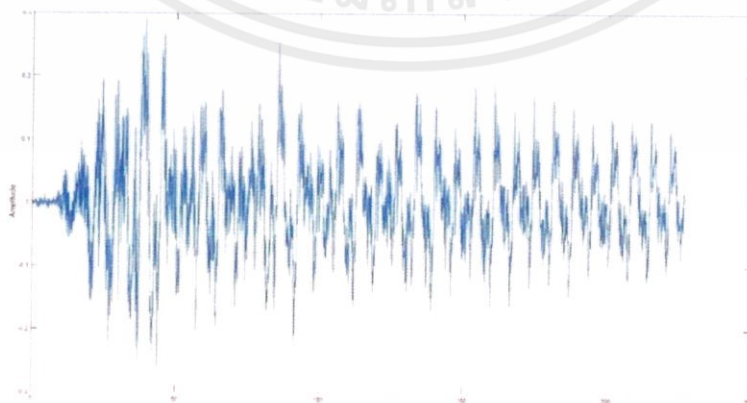
3.2 การเตรียมข้อมูลเสียงก่อนนำไป Training Process ด้วยการ Zero padding



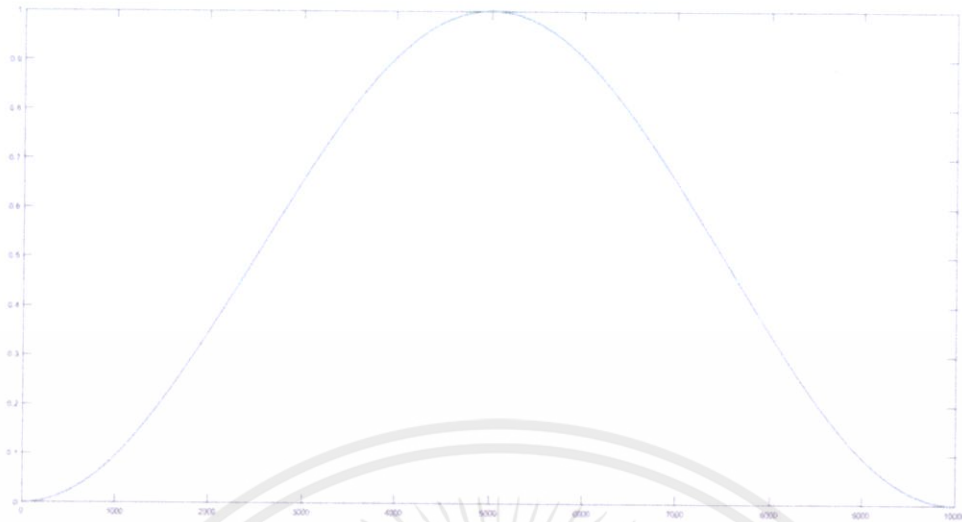
รูปที่3.4 ทำการ Zero padding ข้อมูล

การ zero padding เป็นการเพิ่มข้อมูลด้วย 0 ให้กับข้อมูล เพื่อให้ข้อมูลที่เจ้านำไปประมวลผลสามารถนำไปพิจารณาในกรอบให้เต็มกรอบพิจารณา(หารด้วย10000ลงตัวเพราะกรอบพิจารณามีขนาด10000) เพื่อนำเป็นข้อมูลในลำดับต่อไป

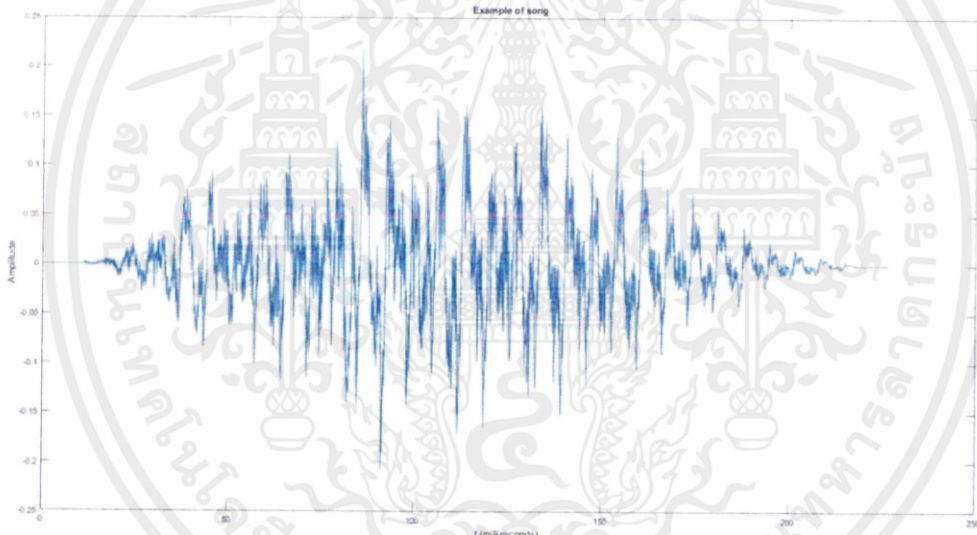
3.3 การนำข้อมูลใส่ใน Windows ด้วย Hanning Function



รูปที่ 3.5 ข้อมูลเพลงทั้งหมด



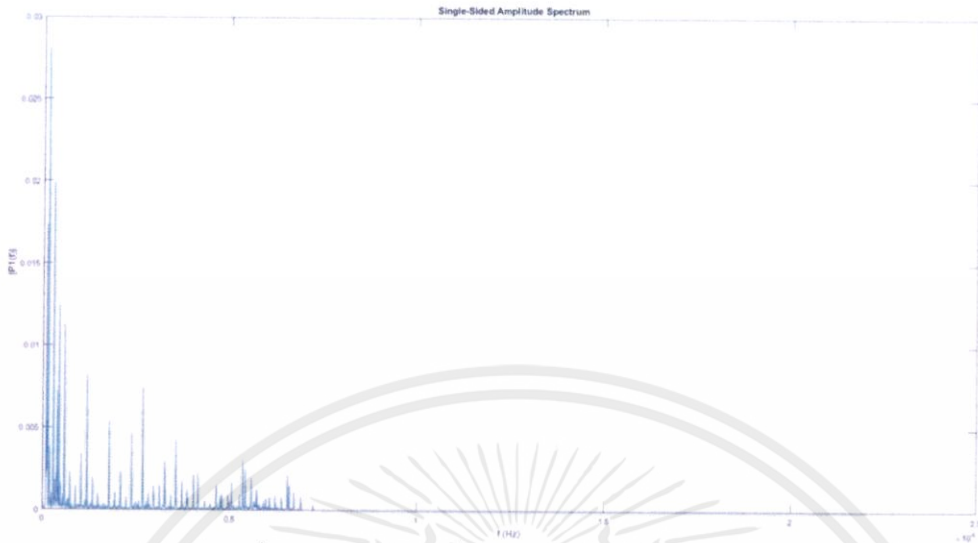
รูปที่ 3.6 Hanning Function



รูปที่ 3.7 ข้อมูลเมื่อผ่านการ Hanning Function

ซึ่งการเก็บข้อมูลลงไปในแต่ละ Windows นั้นจะใช้ Hanning Function เพื่อลดการเกิดเหตุการณ์ Gibb phenomenon บริเวณขอบของ Windows ซึ่งจะทำให้การขยับการ Crop ข้อมูลทีละ 5000 หรือครึ่งหนึ่งของจำนวน Windows

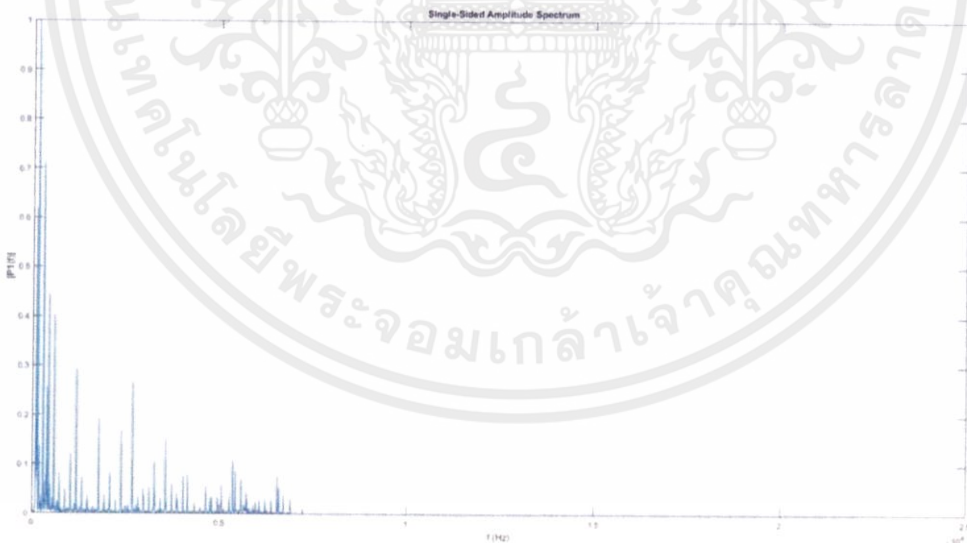
3.4 ทำการข้อมูลแต่ละ Windows ด้วย Discrete Fourier Transform



รูปที่ 3.8 ภาพข้อมูลเมื่อผ่าน Discrete Fourier Transform

เมื่อได้ข้อมูลเพลงมาใส่ Windows (ที่มีขนาด Windows1อันมี 10000 ข้อมูล จำนวนWindowsนั้นแล้วแต่ความสั้นความยาวแต่ละเพลงที่นำไป Training) จับข้อมูลทั้งหมดไปเข้าสมการ Discrete Fourier Transform ได้ข้อมูลดังในรูปภาพ 3.6

3.5 Normalization ข้อมูลก่อนที่เข้า Training process



รูปที่ 3.9 ข้อมูลหลัง Normalization

ทำได้โดยการหาขนาด Amplitude สูงสุด และนำมาหารข้อมูลทุกตัวเพื่อให้ข้อมูลทุกตัวอยู่ในช่วง 0.00 – 1.00

3.6 การปรับปรุงสัญญาณเสียงในโดเมนความถี่เพื่อให้เหมาะสมสำหรับการสั่นหน้าไม้ของกีตาร์

ผู้จัดทำได้เลือกใช้เครื่องมือในโปรแกรม MATLAB 2016 ในการสร้างโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) โดยผู้จัดทำได้เลือกใช้ Radial Basis Function Neural Network (RBF) เนื่องจากมีคุณสมบัติคล้ายกับ Equalizer โดยสถาปัตยกรรมของ RBF จะมีลักษณะคล้ายกับ ANN คือ มีชั้นอินพุต (Input layer) , ชั้นซ่อน (Hidden layer) และ ชั้นเอาต์พุต (Output layer) แต่จะแตกต่างตรงที่ชั้นซ่อนจะมี Activation Function ที่ไม่เป็นเส้นตรงอยู่ โดยในชั้นงานนี้เลือกใช้ Gaussian Basis Function ซึ่งเป็นที่นิยมที่สุด

โดยผู้จัดทำได้ทำการแบ่งข้อมูลของเสียงเป็นช่วง ช่วงละ 10000 ตำแหน่ง (หรือประมาณ 226.76 มิลลิวินาที) โดยใช้ Hanning Window เป็นฟิลเตอร์ในการแบ่งข้อมูล เพื่อป้องกันความถี่สูงที่ขอบของข้อมูล

อินพุตของโครงข่ายนั้นคือ ความถี่ของเสียงที่เล่นจากกีตาร์คุณภาพต่ำ โดยผู้จัดทำได้ทำการแปลงข้อมูลเสียงในโดเมนเวลา 10000 ตำแหน่ง ด้วย Fast Fourier Transform ได้ผลลัพธ์เป็นสัญญาณในโดเมนความถี่ 10,000 ตำแหน่ง แต่เนื่องจากผลลัพธ์ที่ได้มีลักษณะสมมาตร ผู้จัดทำจึงได้ทำการลดข้อมูลที่ใช้ในการสอนโครงข่ายเหลือเพียง 5001 ตำแหน่ง ซึ่งเป็นจำนวนที่เพียงพอสำหรับการแปลงข้อมูลเอาต์พุตกลับไปอยู่ในโดเมนเวลา

เอาต์พุตของโครงข่ายนั้นจะมี 5,001 ตำแหน่งเช่นกันกับอินพุต โดย Target ที่ใช้คือเสียงของกีตาร์คุณภาพดี

หลังจากที่เสร็จสิ้นขั้นตอนการสอนโครงข่ายแล้ว ในการใช้งานจริง ผู้จัดทำจะใช้เสียงดิบเป็นอินพุตของโครงข่าย แล้วจึงนำเอาต์พุตของโครงข่ายไปสั่นหน้าไม้ของกีตาร์คุณภาพต่ำเพื่อกำเนิดเสียง โดยสมมติฐานคือเสียงที่ออกมาจากกีตาร์คุณภาพต่ำ จะมีคุณลักษณะที่ใกล้เคียงกับกีตาร์คุณภาพสูง

บทที่ 4

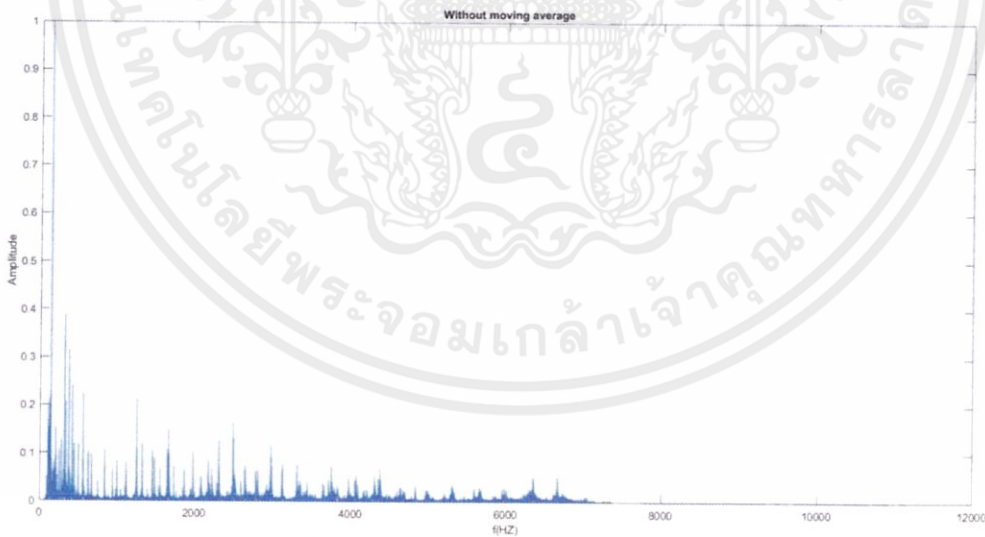
ผลการทดลอง

บทที่ 4 จะกล่าวถึงผลการทดลองหลังจากได้ทำการปรับปรุงสัญญาณเสียงในโดเมนความถี่เพื่อให้เหมาะสมสำหรับการสั่นหน้าไม้ของกีตาร์ การทดลองที่ได้กล่าวไว้ข้างต้นโดยผู้ทำการทดลองได้นำข้อมูลเข้าเรียนรู้ในโครงข่ายระบบประสาทเทียมและได้มีการปรับค่าพารามิเตอร์ต่าง ๆ โดยผลการทดลองกระบวนการมีดังนี้

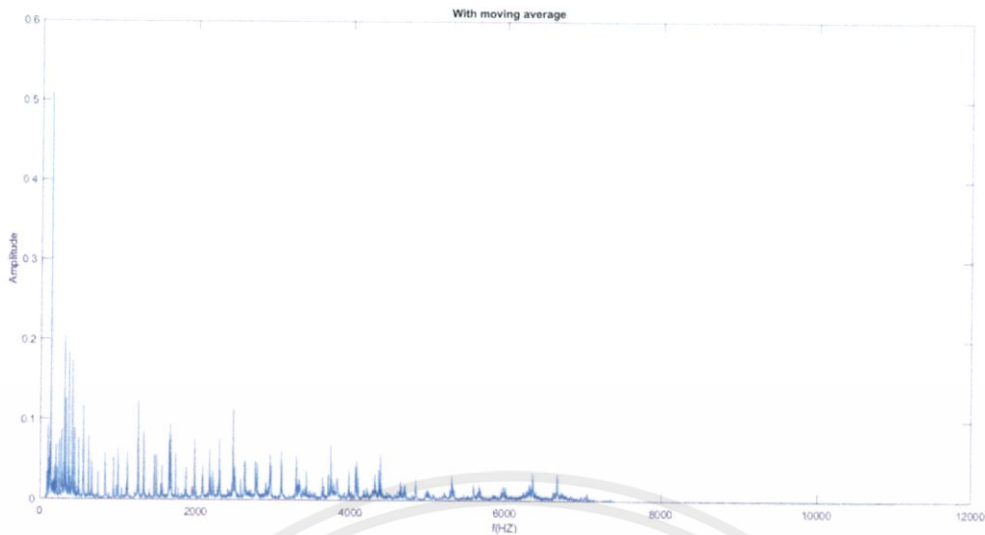
1. ผลการทดลองจากทดลองการเพิ่มข้อมูลเสียง
2. ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม

4.1 ผลการทดลองจากการเพิ่มข้อมูลเสียง

ผู้จัดทำทดลองได้พบว่า ยิ่งเพิ่มข้อมูลที่เรียนเพิ่มขึ้นมากเท่าไรระบบโครงข่ายประสาทเทียมยิ่งหาความสัมพันธ์ของการเปลี่ยนแปลงของกีตาร์ได้เพิ่มขึ้นแต่ถ้าข้อมูลน้อยเกินไปอาจจะทำ Neural ที่ได้ เกิดการจำแต่ข้อมูลที่ Training (ซึ่งเป็นผลลัพธ์ที่ไม่ควรเกิดขึ้น) ข้อมูลที่ได้รับจำเป็นต้องเชื่อถือได้และเป็นข้อมูลที่แม่นยำ ไม่ควรมีสัญญาณรบกวนขณะเก็บข้อมูล อาจจะสามารถช่วยทำให้ข้อมูลดีขึ้นโดยการใช้ Low pass Filter ข้อมูลอีกชั้นเช่น Moving Average , การหาค่ากลางของข้อมูลที่ละชุด ฯลฯ



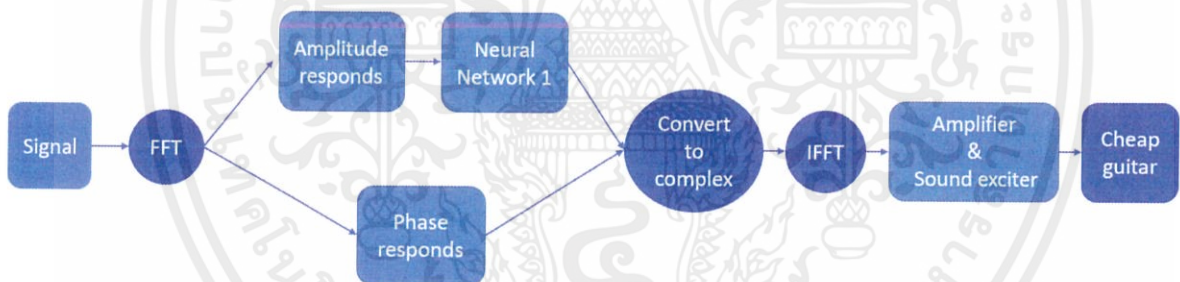
รูปที่ 4.1 ภาพของข้อมูลที่ได้รับมาแบบปกติ



รูปที่ 4.2 ข้อมูลเมื่อผ่าน Moving Average

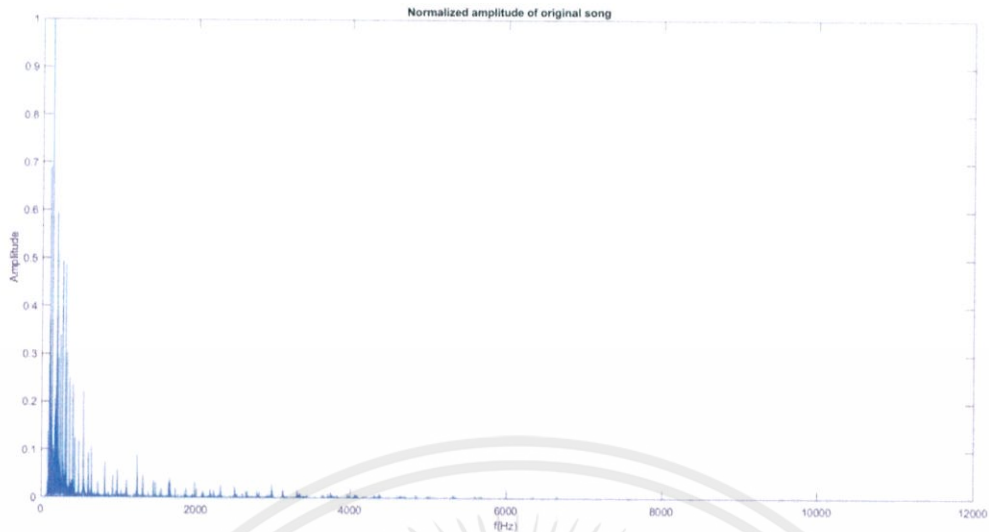
4.2 ผลการทดลองจากการเรียนรู้ด้วยโครงข่ายประสาทเทียม

หลังจาก Training process นำ Neural Network นำมา Feed Forward ตาม Blog diagram



รูปที่ 4.3 Blog Diagram เมื่อตรวจผลการทดลอง

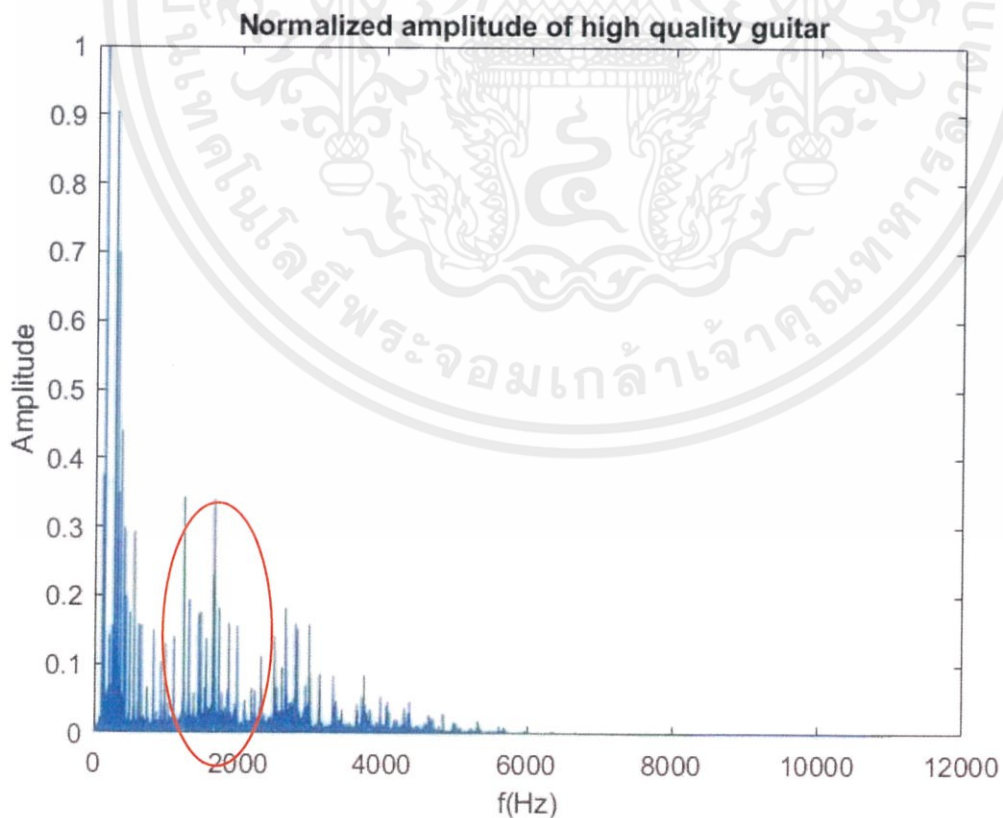
Blog diagram นี้แสดงการถึงขั้นตอนพิสูจน์ผล นำไฟล์เพลงเข้า FFT นำข้อมูล Amplitude ผ่าน Blog Neural Network หลังจากนั้นแปลงกลับด้วย IFFT ผ่าน Amplifier เข้า exciter เขย่าแผ่นไม้ของกีตาร์คุณภาพต่ำ



รูปที่ 4.4 ภาพของข้อมูลที่ได้รับมาแบบปกติ

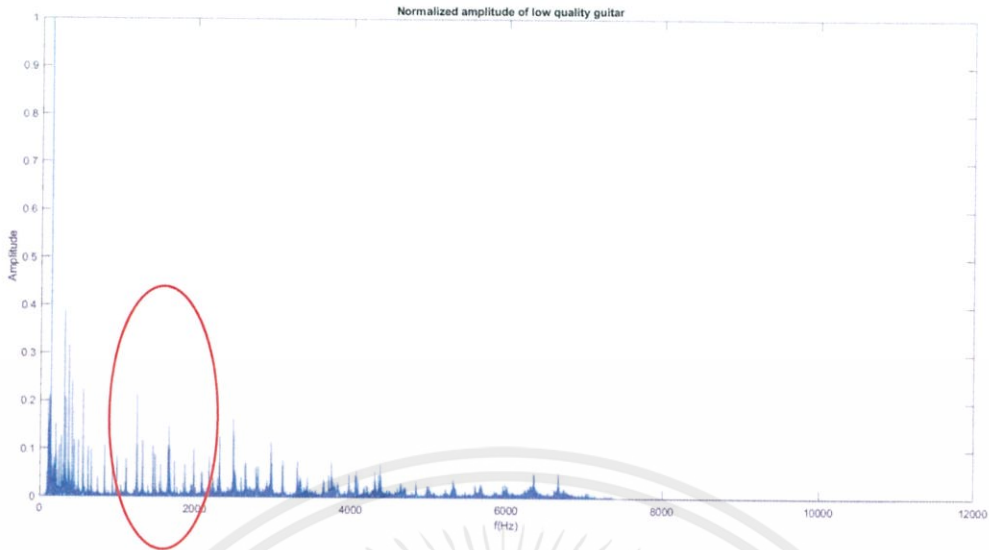
ภาพที่ 4.4 แสดงให้เห็น ภาพของข้อมูลเพลงใด ๆ ในโดเมนความถี่ ซึ่งได้ทำการเตรียมข้อมูลก่อน Feed Forward เหมือนที่เตรียม ก่อนเข้า Training Process ในบทที่ 3

การทดสอบทำได้โดยการนำข้อมูลเพลงใด ๆ เล่นให้กับกีตาร์คุณภาพสูง แล้วบันทึกกราฟเอาไว้เพื่อรอการเปรียบเทียบกับเอาต์พุตของ Neural Network



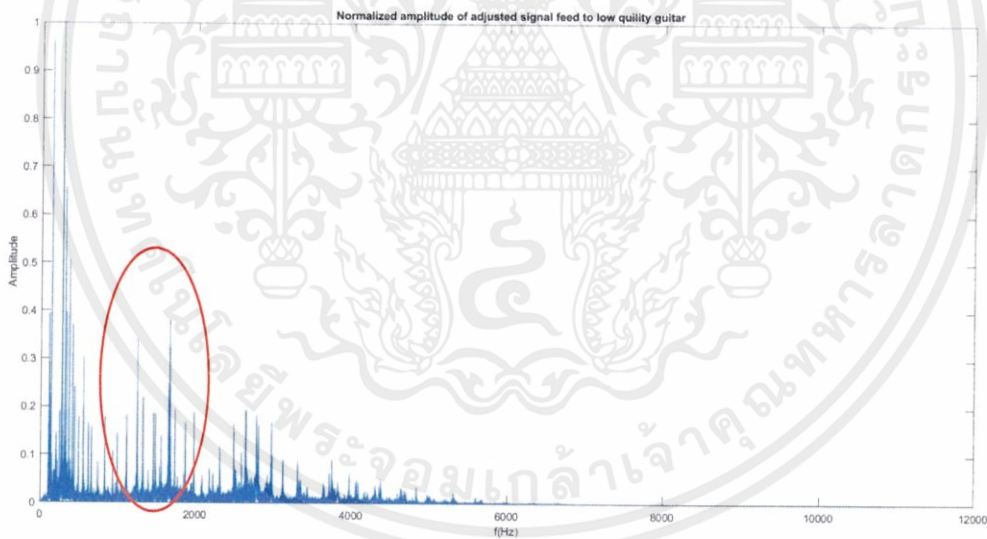
รูปที่ 4.5 ภาพของการตอบสนองความถี่ของกีตาร์คุณภาพดี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



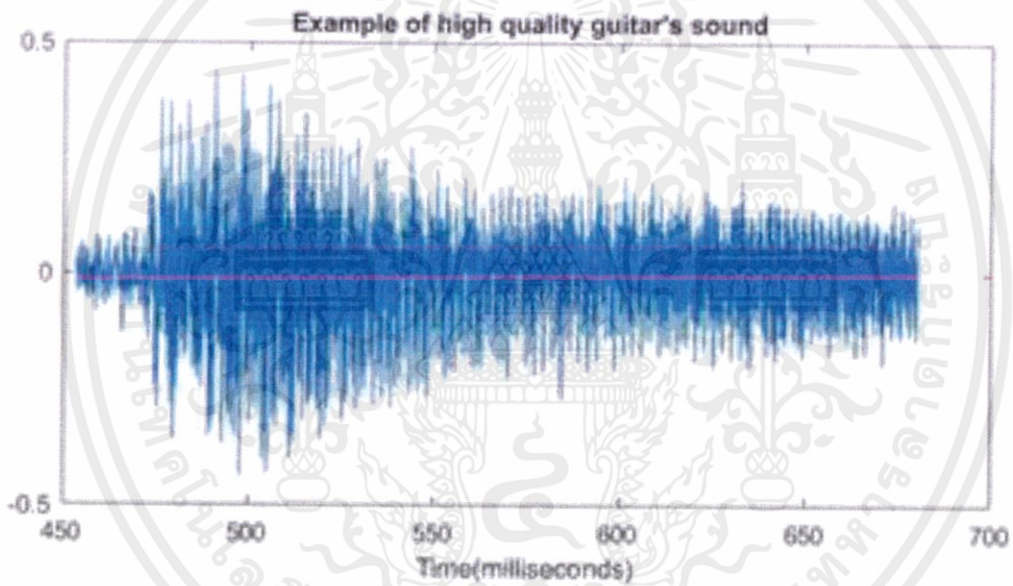
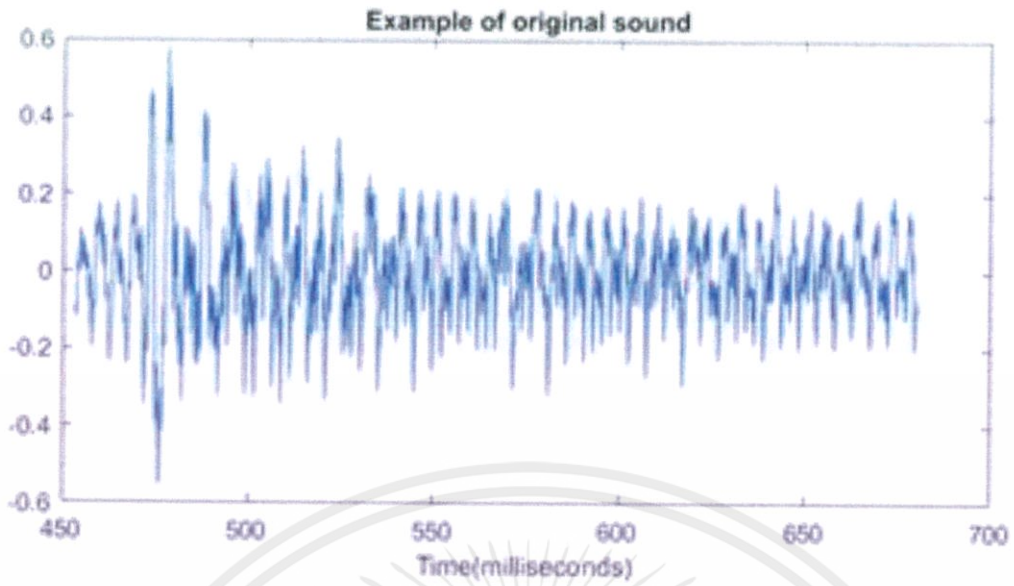
รูปที่ 4.6 ภาพของการตอบสนองความถี่ของกีตาร์คุณภาพต่ำ

ดูความแตกต่างของ การตอบสนองความถี่ กีตาร์แต่ละตัว ทั้งที่มีคุณภาพสูงและคุณภาพที่ต่ำกว่า จะเห็นได้ว่ากีตาร์ทั้งสองตัวนั้นมี การตอบสนองความถี่ที่แตกต่างกันอย่างเห็นได้ชัดเจน และเมื่อทำตาม Blog diagram แล้วดูผลลัพธ์

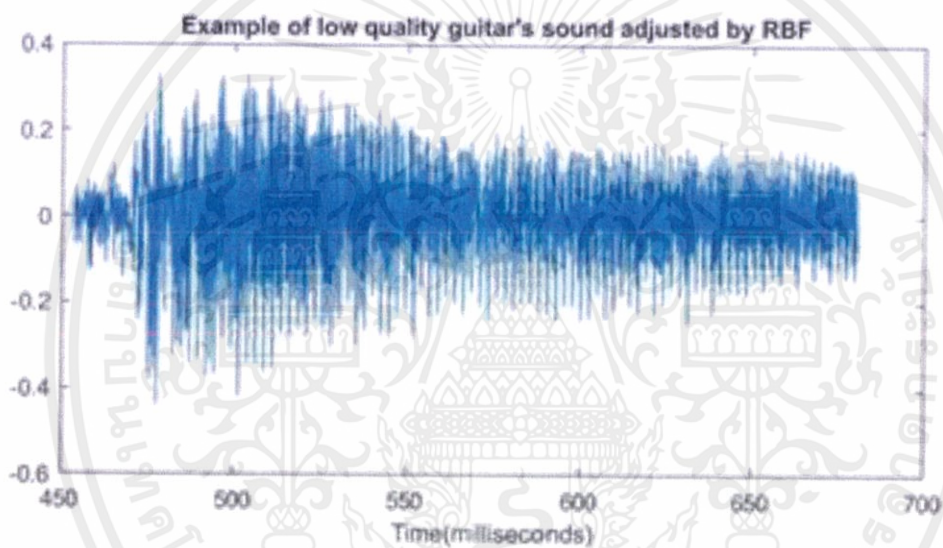
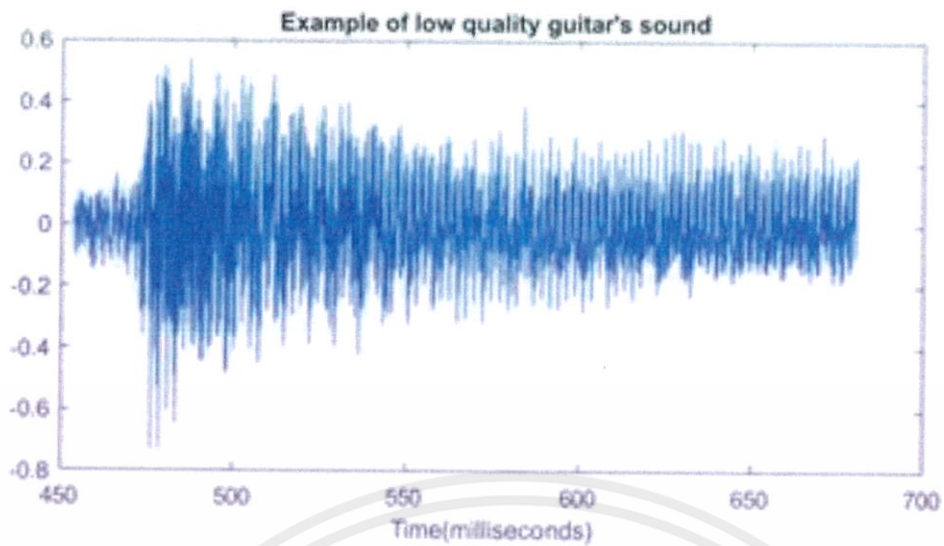


รูปที่ 4.7 ภาพของการตอบสนองความถี่ของกีตาร์คุณภาพต่ำเมื่อข้อมูลนั้นถูกปรับปรุงด้วย RBF Neural network Equalizer

จะเห็นได้ว่า เมื่อนำข้อมูลเอาต์พุตมาเล่นผ่าน กีตาร์คุณภาพต่ำ จะเห็นการตอบสนองความถี่ใกล้เคียงกับ การตอบสนองความถี่ของกีตาร์คุณภาพสูงมากขึ้น หลังจากนั้นแปลงกลับ Time Domain ด้วย IFFT



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อเล่นเป็นเพลงก็ยังสามารถฟังออกมาเป็นเพลงได้โดยข้อมูลไม่เสียหาย ซึ่งเป็นผลที่หน้าพอใจให้กับการทดลองครั้งนี้อย่างมาก

บทที่ 5

สรุปและวิจารณ์

5.1 ปัญหาที่เกิดขึ้น

5.1.1 ด้านฮาร์ดแวร์ (Hardware)

เนื่องการทดลองครั้งนี้มีการอัดบันทึกเสียงเพื่อนำข้อมูลไป Training Process ซึ่งเครื่องมือในการบันทึกควรจะมีคุณภาพที่ดีและสภาพแวดล้อมในการอัดที่ดี เพื่อข้อมูลที่มีประสิทธิภาพมากขึ้นอีกทั้งยังรวมถึงฮาร์ดแวร์การเก็บข้อมูลเสียงของคอมพิวเตอร์ (ADC ของคอมพิวเตอร์ ควรที่จะมีคุณภาพมากกว่านี้) อาจจะสามารถแก้ไขได้โดยการบันทึกเสียงทั้งหมดที่ห้องอัดเสียงโดยเฉพาะ โครงสร้างในการยึดระหว่างที่ออกแบบไม่ควรไปขัดขวางการสั่นของแผ่นไม้กีดาร์

5.1.2 ด้านซอฟต์แวร์ (Software)

คอมพิวเตอร์ และ โปรแกรมในการ Training ควรเลือกใช้ให้เหมาะสมซึ่งผู้จัดทำเคยทำงานด้านนี้ผ่านโปรแกรม Python มีความเห็นว่าโปรแกรม Python ความอิสระในการใช้งานมากกว่าโปรแกรม MatLab แต่โปรแกรม MatLab นั้นมีความสะดวกสบายในการใช้งานมากขึ้นแต่มีขีดจำกัดในการใช้งาน การเก็บข้อมูลนั้นควรมีข้อมูลจำนวนมาก ๆ เพื่อป้องกัน การจำเฉพาะข้อมูล Training Process

5.2 สรุปผลการทดลอง

จากผลการทดลองที่ได้พบว่า สัญญาณที่ได้จากโครงข่ายประสาทเทียม (Artificial Neural Network) สามารถช่วยให้คุณลักษณะของเสียงที่ออกมาจากกีตาร์คุณภาพต่ำมีความใกล้เคียงกับกีตาร์คุณภาพสูงได้ดีพอสมควร

การใช้ Hanning Filter ในการแบ่งตัวอย่างของสัญญาณในทางเวลาสามารถลดทอนความสูงที่ขอบของฟิลเตอร์ที่ไม่ต้องการได้ ทำให้ข้อมูลที่ใช้ในการสอนโครงข่ายซึ่งเป็นโดเมนความถี่มีความถูกต้องมากขึ้น การเลือกใช้ Radial Basis Function Neural Network เนื่องจากสถาปัตยกรรมของโครงข่ายประพุดิตนคล้ายกับ Equalizer สามารถช่วยเพิ่มประสิทธิภาพของชิ้นงานได้เป็นอย่างดี โครงงานชิ้นนี้ถือว่ามีแนวโน้มที่ดีในการเข้าใกล้กับทฤษฎีหรือสมมุติฐานที่ตั้งไว้ แต่ถ้ามีเวลาในการเก็บข้อมูลเพื่อ Training Process มากขึ้นหรือปรับปรุงข้อมูลที่ใช้ Training อาจจะทำให้ผลลัพธ์ดีขึ้นกว่าครั้งนี้

เอกสารอ้างอิง

[1] <https://www.mathworks.com/>

[2] <https://cargocollective.com/lh-hantraku/Surrogate-Soundboards-Mark-I>

[3] <https://cargocollective.com/lh-hantraku/Surrogate-Soundboards-Mark-II>



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data preprocessing.

```
wav_file = scan_wav('C:\Users\USER\Desktop\Project');

nb_wav = size(wav_file , 2);

car = cell(1 , nb_wav / 3);
mar = cell(1 , nb_wav / 3);
org = cell(1 , nb_wav / 3);

j = 1;
k = 1;
l = 1;

for i = 1 : nb_wav
    if string(wav_file{i}(end - 6:end - 4)) == string('car')
        car{j} = wav_file{i};
        j = j+1;
    elseif string(wav_file{i}(end - 6:end - 4)) == string('mar')
        mar{k} = wav_file{i};
        k = k+1;
    else
        org{l} = wav_file{i};
        l = l+1;
    end
end

sound_car = cell(nb_wav / 3 , 1);
sound_mar = cell(nb_wav / 3 , 1);
sound_org = cell(nb_wav / 3 , 1);

for i = 1 : nb_wav / 3
    file_name = car{i};
    [y , Fs] = audioread(file_name);
    yMono = sum(y, 2) / 2;
    Size = size(yMono);
    Size = Size(1);
    for j = 1 : Size
        sound_car{i}{j} = yMono(j);
    end

    file_name = mar{i};
    [y , Fs] = audioread(file_name);
    yMono = sum(y, 2) / 2;
    Size = size(yMono);
    Size = Size(1);
    for j = 1 : Size
        sound_mar{i}{j} = yMono(j);
    end

    file_name = org{i};
    [y , Fs] = audioread(file_name);
    yMono = sum(y, 2) / 2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Size = size(yMono);
Size = Size(1);
for j = 1 : Size
    sound_org{i}{j} = yMono(j);
end
end

wd = transpose(hanning(10000));
car_amp = zeros(1 , 5001);
mar_amp = zeros(1 , 5001);
org_amp = zeros(1 , 5001);

car_phs = zeros(1 , 5001);
mar_phs = zeros(1 , 5001);
org_phs = zeros(1 , 5001);

for i = 1 : nb_wav / 3
    fprintf('%d' , i);
    fprintf('\n');
    fprintf('car \n');

    dummy = sound_car{i};
    dum_size = size(dummy , 2);
    rem = mod(dum_size , 10000);
    for j = 1 : 10000-rem
        dummy{dum_size + j} = 0;
    end

    dummy_amp = zeros(1 , 5001);
    dummy = cell2mat(dummy);
    dum_size = size(dummy , 2);
    for j = 1 : 5000 : dum_size - 9999
        dummy2 = fft(wd .* dummy(j : j +9999));
        dummy2 = dummy2(1 : 5001);
        dummy3 = abs(dummy2);
        dummy_phs = angle(dummy2);
        dummy_amp = cat(1 , dummy_amp , dummy3);
        car_phs = cat(1 , car_phs , dummy_phs);
    end
    dummy_amp = dummy_amp(2:end , :);
    max_ = max(dummy_amp);
    max_ = max(max_);
    dummy_amp = dummy_amp / max_;
    car_amp = cat(1 , car_amp , dummy_amp);

    fprintf('mar \n');

    dummy = sound_mar{i};
    dum_size = size(dummy , 2);
    rem = mod(dum_size , 10000);
    for j = 1 : 10000-rem
        dummy{dum_size + j} = 0;
    end

    dummy_amp = zeros(1 , 5001);

```

```

dummy = cell2mat(dummy);
dum_size = size(dummy , 2);
for j = 1 : 5000 : dum_size - 9999
    dummy2 = fft(wd .* dummy(j : j +9999));
    dummy2 = dummy2(1 : 5001);
    dummy3 = abs(dummy2);
    dummy_phs = angle(dummy2);
    dummy_amp = cat(1 , dummy_amp , dummy3);
    mar_phs = cat(1 , mar_phs , dummy_phs);
end
dummy_amp = dummy_amp(2:end , :);
max_ = max(dummy_amp);
max_ = max(max_);
dummy_amp = dummy_amp / max_;
mar_amp = cat(1 , mar_amp , dummy_amp);

fprintf('org \n');

dummy = sound_org{i};
dum_size = size(dummy , 2);
rem = mod(dum_size , 10000);
for j = 1 : 10000-rem
    dummy{dum_size + j} = 0;
end

dummy_amp = zeros(1 , 5001);
dummy = cell2mat(dummy);
dum_size = size(dummy , 2);
for j = 1 : 5000 : dum_size - 9999
    dummy2 = fft(wd .* dummy(j : j +9999));
    dummy2 = dummy2(1 : 5001);
    dummy3 = abs(dummy2);
    dummy_phs = angle(dummy2);
    dummy_amp = cat(1 , dummy_amp , dummy3);
    org_phs = cat(1 , org_phs , dummy_phs);
end
dummy_amp = dummy_amp(2:end , :);
max_ = max(dummy_amp);
max_ = max(max_);
dummy_amp = dummy_amp / max_;
org_amp = cat(1 , org_amp , dummy_amp);

end

car_amp = car_amp(2:end , :);
mar_amp = mar_amp(2:end , :);
org_amp = org_amp(2:end , :);

car_phs = car_phs(2:end , :);
mar_phs = mar_phs(2:end , :);
org_phs = org_phs(2:end , :);

car_amp = transpose(car_amp);
mar_amp = transpose(mar_amp);
org_amp = transpose(org_amp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function wav_file = scan_wav(directory)
    listing = dir(directory);
    dim_list = size(listing);
    dim_list = dim_list(1);

    file_name = cell(1 , dim_list);
    wav_file = cell(1 , dim_list);

    for i = 1 : dim_list
        file_name{i} = char(listing(i).name);
        if strlength(file_name{i}) < 5
            file_name{i} = 'Wrong';
        end
    end

    for i = 1 : dim_list
        a = char(file_name{i});
        a = a(end-2 : end);

        if string(a) ~= string('wav')
            file_name{i} = 'Wrong';
        end
    end

    j = 1;
    for i = 1 : dim_list
        a = file_name{i};
        if string(a) ~= string('Wrong')
            wav_file{j} = file_name{i};
            j = j + 1;
        end
    end

    idx = 1;
    for i = 1 : dim_list
        if class(wav_file{dim_list - i + 1}) ~= string('double')
            idx = dim_list - i + 1;
            break
        end
    end

    wav_file = wav_file(1:idx);
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Feed Forward Process.

```
[y , Fs] = audioread('Track_9_o.wav');
yMono = sum(y, 2) / 2;

song = transpose(yMono);

rem = mod(size(song , 2) , 10000);

song = cat(2 , song , zeros(1 , 10000 - rem));

wd = transpose(hanning(10000));

song_piece = zeros(1 , 10000);

len_data = size(song , 2);

for i = 1 : 5000 : len_data - 9999
    dummy = wd .* song(:, i:i+9999);
    song_piece = cat(1 , song_piece , dummy);
end

song_piece = song_piece(2:end , :);
song_fre = fft(song_piece , [] , 2);
song_fre = song_fre(:, 1:5001);

song_amp = abs(song_fre);
song_phs = angle(song_fre);

max_amp = max(song_amp);
max_amp = max(max_amp);
song_amp = song_amp / max_amp;

song_amp = transpose(song_amp);

y = RBF_amp_4(song_amp);

y = transpose(y);

y_complex = complex(y .* cos(song_phs) , y .* sin(song_phs));

y_complex(:, 1) = abs(y_complex(:, 1));
y_complex(:, 5001) = abs(y_complex(:, 5001));

half = conj(y_complex(:, 5000:-1:2));

y_complex = cat(2 , y_complex , half);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y_time = ifft(y_complex , [] , 2);  
  
song_new = zeros(1 , len_data);  
  
j = 1;  
for i = 1 : 5000 : len_data - 9999  
    fprintf('%d : %d' , i , j);  
    fprintf('\n');  
    song_new(i:i+9999) = song_new(i:i+9999) + y_time(j , :);  
    j = j + 1;  
end  
  
song_new = song_new * 1000;  
  
audiowrite('Track_9_RBF.wav' , song_new , 44100);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้